

**Prototipo software para el posicionamiento GPS de buses de transporte masivo.
Caso: Metrolínea**

Cristian Yesid Fraija Castillo

Trabajo de Grado para optar el título de Ingeniería de Sistemas

Director

Urbano Eliécer Gómez Prada,

**Magister en ingeniería - Área informática y ciencias de la
computación**

Universidad Industrial de Santander

Facultad de Ingenierías Físico-Mecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2020

Tabla de contenido

Introducción 14

1. Descripción del proyecto..... 14

1.1. Planteamiento y justificación del problema 14

1.2. Justificación..... 16

1.3. Metodología 16

1.3.1. Fase de diseño y redacción de requerimientos. 17

1.3.2. Primer prototipo. 18

1.3.3. Segundo prototipo. 18

1.3.4. Tercer prototipo. 19

2. Objetivos 20

2.1. Objetivo general 20

2.2. Objetivos específicos..... 20

3. Marco de referencia 21

3.1. Estado del arte 21

3.1.1. EasyTaxi..... 21

3.1.2. UBER. 21

3.1.3. Transmisitp | Android | IOS. 22

3.1.4. TAPPSI..... 22

3.1.5. MOOVIT..... 22

3.2. Marco Contextual 23

3.2.1. BRT..... 23

3.2.2. Metrolínea 23

3.2.3. API..... 24

3.2.4. Framework..... 24

3.2.5. Geolocalización. 24

3.2.6. Sistema de información. 25

3.2.7. Frontend..... 25

3.2.8. Backend. 25

3.3. Marco legal 25

3.3.1. Google Maps API Key. 25

3.4. Marco tecnológico 26

3.4.1. Mean Stack. 26

3.4.2. Visual Studio Code..... 26

3.4.3. Bootstrap. 27

3.4.4. Materialized..... 27

3.4.5. API de Google Maps. 27

3.4.6. Directions API.....	27
3.4.7. API de geolocalización.....	27
3.4.8. Node Js.	28
3.4.9. Express.	28
3.4.10. MongoDB.....	28
3.4.11. Mongoose.	28
3.4.12. Angular.....	28
3.4.13. Angular route guards.....	29
3.4.14. Postman.....	29
3.4.15. Lucidchart.....	29
3.4.16. Google Analytics.....	29
3.4.17. Autenticación basada en JWT.....	30
3.4.18. GitHub.....	30
3.4.19. Heroku.....	30
3.4.20. MongoDB Atlas.....	30
3.4.21. Bcrypt.....	31
4. Resultados	32
4.1. Requerimientos funcionales y no funcionales.....	32
4.1.1. Requerimientos funcionales.....	32
4.1.2. Requerimientos no funcionales.....	33

4.2. Diseño del prototipo	34
4.2.1. Casos de uso.	34
4.2.2. Diagrama de actividades.	39
4.2.3. Diagrama de clases	44
4.2.4. Arquitectura software.	45
4.3. Implementación	46
4.3.1. Implementación prototipos.....	47
4.4. Validación	59
4.4.1. Validación desde los perfiles del software – Administrador.....	59
4.4.2. Validación desde los módulos del software - Conductor	60
4.4.3. Validación desde los módulos del software – Invitado	61
4.4.4. Validación de la experiencia de usuario	62
4.5. Google Analytics.....	66
5. Conclusiones	71
6. Recomendaciones	72
Referencias Bibliográficas	73
Apéndices	77

Lista de tablas

Tabla 1. *Lista de requerimientos funcionales.* 32

Tabla 2. *Lista de requerimientos no funcionales.* 34

Tabla 3. *Descripción de actores del prototipo.* 35

Tabla 4. *rutas guardadas en base de datos.* 60

Tabla 5. *Registrar usuario* 77

Tabla 6. *Gestionar usuario.* 77

Tabla 7. *Login usuario.* 77

Tabla 8. *Logout usuario.* 78

Tabla 9. *Crear rutas.* 78

Tabla 10. *Gestionar rutas.* 78

Tabla 11. *Crear buses.* 78

Tabla 12. *Gestionar buses.* 79

Tabla 13. *Asignar buses a ruta.* 79

Tabla 14. *Gestionar asignaciones de rutas.* 79

Tabla 15. *Enviar la ubicación mediante GPS.* 79

Tabla 16. *Visualizar ruta seleccionada en mapa (Google Maps).* 79

Tabla 17. *Filtrar búsqueda de ruta.* 80

Tabla 18. *Observar la vista principal.* 80

Tabla 19. *Observar rutas.* 80

Tabla 20. *Acceder a una ruta.* 80

Tabla 21. *Filtrar buses por ruta.* 81

Tabla 22. *Asignar conductor a un bus.* 81

Tabla 23. <i>Gestionar asignaciones de conductores.</i>	81
Tabla 24. <i>Diseñar una interfaz gráfica de usuario.</i>	81
Tabla 25. <i>Operar con varios usuarios al tiempo.</i>	81
Tabla 26. <i>Operar con varios buses y rutas al tiempo.</i>	82
Tabla 27. <i>Actualizar información en la base de datos.</i>	82
Tabla 28. <i>Actualizar información en tiempo real, la posición de los buses en mapa y base de datos.</i>	82
Tabla 29. <i>Compatibilidad del prototipo.</i>	82
Tabla 30. <i>Login usuario.</i>	83
Tabla 31. <i>Logout usuario.</i>	83
Tabla 32. <i>Acceder a rutas.</i>	84
Tabla 33. <i>Ver una ruta específica.</i>	84
Tabla 34. <i>Ubicación de buses en mapa.</i>	84
Tabla 35. <i>Filtrar rutas desde el navbar.</i>	85
Tabla 36. <i>Compartir ubicación GPS a su bus asociado.</i>	85
Tabla 37. <i>Dejar de Compartir ubicación a su bus asociado.</i>	86
Tabla 38. <i>Acceder a vista administrador.</i>	86
Tabla 39. <i>Acceder a vista Registrar.</i>	87
Tabla 40. <i>Crear ruta.</i>	87
Tabla 41. <i>Crear bus.</i>	88
Tabla 42. <i>Registrar usuario.</i>	88
Tabla 43. <i>Editar ruta.</i>	89
Tabla 44. <i>Editar bus.</i>	89

Tabla 45. *Editar conductor.* 90

Tabla 46. *Eliminar ruta.* 90

Tabla 47. *Eliminar bus.* 91

Tabla 48. *Eliminar conductor.* 91

Tabla 49. *Filtrar rutas.* 92

Tabla 50. *Filtrar buses.* 92

Tabla 51. *Filtrar conductores.* 93

Tabla 52. *Asignar ruta a un bus.* 93

Tabla 53. *Asignar conductor a un bus.* 94

Tabla 54. *Ver información de un bus.* 94

Lista de Figuras

Figura 1. Diagrama de casos de uso principal del prototipo. 36

Figura 2. Diagrama de casos de uso de la interacción con la interfaz web. 37

Figura 3. Diagrama de casos de uso de gestionar conductores, rutas buses y paradas. 38

Figura 4. Diagrama de actividades ver rutas y buses asociados a esta en el mapa. 39

Figura 5. Diagrama de actividades de la funcionalidad enviar la posición de un conductor a su bus asociado. 40

Figura 6. Diagrama de actividades de la funcionalidad Gestionar rutas para usuarios de tipo administrador..... 41

Figura 7. Registrar usuarios y gestionar usuarios de tipo conductor para usuarios de tipo administrador..... 42

Figura 8. Crear y gestionar buses para usuarios de tipo administrador..... 43

Figura 9. Diagrama de clases del prototipo. 44

Figura 10. Modelo-Vista-Controlador del prototipo. 46

Figura 11. Ejemplo del código en el entorno de programación Visual Studio Code y el Framework Angular. 47

Figura 12. Visualizar rutas. 48

Figura 13. Login usuario. 49

Figura 14. Register usuario..... 49

Figura 15. Vista administrador. 51

Figura 16. Vista conductor sin mostrar posición..... 52

Figura 17. Vista conductor Mostrando posición. 52

Figura 18. Vista agregar/editar ruta. 53

Figura 19. Vista agregar/editar bus. 54

Figura 20. Vista asignar ruta a bus. 55

Figura 21. Vista asignar conductor a bus. 55

Figura 22. Vista ruta seleccionada..... 56

Figura 23. Vista Home 57

Figura 24. Vista gestionar paradas. 58

Figura 25. Vista agregar / eliminar ruta a parada. 58

Figura 26. Resultados del primer ítem evaluado 63

Figura 27. Resultados del segundo ítem evaluado. 63

Figura 28. Resultados del tercer ítem evaluado..... 64

Figura 29. Resultados del cuarto ítem evaluado..... 64

Figura 30. Resultados del quinto ítem evaluado. 64

Figura 31. Resultados del sexto ítem evaluado. 64

Figura 32. Resultados del séptimo ítem evaluado. 65

Figura 33. Resultados del octavo ítem evaluado. 65

Figura 34. Resultados del noveno ítem evaluado. 65

Figura 35. Resultados del décimo ítem evaluado..... 66

Figura 36. Google Analytics Data. 67

Figura 37. Obtención de usuarios..... 67

Figura 38. Horario y ubicación de acceso a la página web. 68

Figura 39. Páginas que visitan los usuarios..... 69

Figura 40. Informe general de audiencia. 69

Lista de Apéndices

Apéndices A. Requerimientos software..... 77
Apéndices B. Casos de uso..... 83
Apéndices C Implementación prototipos..... 95

Resumen

Título: Prototipo software para el posicionamiento GPS de buses de transporte masivo. Caso: Metrolínea.

Autor: Cristian Yesid Fraija Castillo

Palabras clave: Transporte masivo, posicionamiento GPS, desarrollo de software, prototipo software.

Objetivo General: Desarrollar un prototipo software en entorno web para la generación de informes del posicionamiento de rutas y ubicación en tiempo real a través de GPS de buses de transporte masivo.

Descripción: Las ciudades seguirán creciendo, sobre todo en los países en desarrollo. Actualmente, el 55% de las personas en el mundo vive en ciudades. Según un nuevo informe de la ONU (Organización de Naciones Unidas), se estima que en el 2050 esta proporción aumentará hasta un 13%. El gran reto por parte de las ciudades es el desarrollo sostenible, el cual dependerá cada vez más de la forma apropiada del crecimiento urbano (Noticias ONU, 2018), la capacidad de transformar sus sistemas y el uso de la tecnología para optimizar el uso de sus recursos. El desarrollo de este prototipo pretende brindar información de valor agregado para el sistema de buses de transporte masivo.

La alternativa de transporte masivo de pasajeros en áreas urbanas que se ha implementado en Latinoamérica es el modelo conocido como BRT (Buses de transporte Rápido), una de las diversas problemáticas del sistema de información que se maneja actualmente es la falta de información por parte de sus usuarios. Mediante este trabajo y un prototipo software se pretende aportar a un manejo eficiente de la información en el sistema de transporte actual de Bucaramanga, el tener acceso a datos, como la posición de los buses del sistema en tiempo real, generando un valor agregado.

* Trabajo de Grado

** Facultad Fisicomecánicas Escuela de Sistemas Director Urbano Eliécer Gómez Prada

Abstract

Title: software prototype for GPS positioning of mass transit buses. Case: Metrolínea.

Authors: Cristian Yesid Fraija Castillo

Keywords: Mass transport, GPS positioning, software development, software prototype.

General Objective: Develop a software prototype in web environment for the generation of reports of the positioning of routes and location in real time through GPS of mass transit buses.

DESCRIPTION: Cities will continue to grow, especially in developing countries. Currently, 55% of people in the world live in cities. According to a new report by the UN (United Nations Organization), it is estimated that in 2050 this proportion will increase up to 13%. The great challenge on the part of cities is sustainable development, which will depend increasingly on the way appropriate urban growth (Noticias ONU, 2018), the ability to transform their systems and the use of technology to optimize the use of their resources. The development of this prototype aims to provide value-added information for the mass transit bus system.

The alternative of mass passenger transport in urban areas that has been implemented in Latin America is the model known as BRT (Fast Transport buses), one of the various problems of the information system currently being handled is the lack of information by your users. Through this work and a software prototype it is intended to contribute to an efficient management of information in the current transport system of Bucaramanga, having access to data, such as the position of the buses of the system in real time, generating added value.

* Bachelor Thesis

** Facultad Fisicomecánicas Escuela de sistemas Director Urbano Eliécer Gómez Prada

Introducción

El transporte masivo en áreas urbanas ha sido la medida optada por la mayoría de los gobiernos en Latinoamérica para darle solución a la problemática de trasladar un flujo masivo de usuarios en zonas urbanas, el modelo BRT (Buses de transporte rápido) es un sistema basado en buses, diseñado para mejorar el flujo de pasajeros y se caracteriza por transitar generalmente por carriles exclusivos. Este modelo consta de muchas ventajas y fortalezas, desde sus costos de construcción e implementación hasta menores tiempos de viaje mientras se encuentran en carreteras con carriles exclusivos, así también tiene muchas problemáticas a mejorar, una de ellas es la desinformación por parte de los usuarios y la dificultad de llevar o informar cambios en sus rutas de manera oportuna.

Con el fin de generar valor agregado e incluso llegar a solucionar una problemática a el sistema de transporte masivo: **caso Metrolínea**, se propone un entorno web el cual permitirá a los usuarios localizar mediante GPS la posición en tiempo real de cada bus del sistema. Además de conocer las rutas y una sección de alerta en donde se encontrará de manera oportuna cualquier cambio que sea realizado a las rutas de cada uno de los buses.

1 Descripción del proyecto

1.1 Planteamiento y justificación del problema

Con el cambio de las economías regionales, la sustitución de la agricultura por servicios y la industria, las ciudades en América Latina crecieron de forma rápida y desordenada (60's y 70's). Con el crecimiento de las ciudades y su población se generaron problemas de movilidad los desplazamientos eran cada vez más largos y la precaria red vial se tradujo en congestión. Simultáneamente llegaron problemas de contaminación del aire, accidentes de tráfico y disminución de la actividad física de las personas—todos con serios problemas para la salud. El desarrollo urbano informal también hizo difícil la creación de espacios públicos para la recreación y el deporte, los cuales son parte fundamental de un desarrollo humano integral [11].

Adoptar medidas que países desarrollados habían tomado para solucionar estos problemas resultaban inviables por las dificultades financieras en esos momentos, se necesitaba una solución de rápida implantación, de bajos costos y alto impacto.

El modelo de transporte basado en buses de tránsito rápido (BRT) es el modelo que encajaba perfectamente como una solución viable en muchas ciudades, su velocidad de implantación hacía que el tiempo de planeamiento y puesta en marcha fuese corto, los costos de construcción, mantenimiento y operación eran considerablemente más bajos que otras alternativas (como transporte sobre rieles o cableado) y su conectividad de rutas era más eficiente gracias a los buses alimentadores.

A pesar de sus ventajas, aún este modelo cuenta con muchas falencias, en algunos casos por falta de inversión y en otros por el escaso manejo de la información que poseen, para este último, uno de los ejemplos más comunes es la falta de conocimiento del tiempo que tarda un bus en

llegar a determinada parada ya que para los usuarios del sistema de transporte masivo, el conocer la posición y el tiempo estimado que tardará en llegar un bus a la parada o ubicación deseada sería de gran utilidad a la hora de toma de decisión, les permitiría programarse con anticipación (saber si llegan a tiempo o esperan al siguiente bus), esto hace necesario el uso de una herramienta la cual como función principal permita conocer la posición en tiempo real y un tiempo estimado de llegada a cierta parada, entre otras características.

1.2 Justificación

Algunos medios de transporte actuales ya cuentan con el rastreo mediante GPS, la API de Google Maps y sus dispositivos móviles, esto les permite estimar la ubicación y el tiempo de espera. Lo cual genera una sensación de confort y confianza hacia los usuarios, y a su vez también genera la necesidad de implementar un sistema similar en el sistema de transporte de la ciudad.

El presente trabajo pretende aportar un prototipo de software el cual intentará ayudar al mejorar falencias en el modelo de transporte actual en Bucaramanga, con un manejo adecuado de la información el sistema estaría generando un valor agregado para sus usuarios. Especialmente dirigido a los usuarios de buses o Transmilenio.

1.3 Metodología

Para el diseño y la estructuración del prototipo de software, el cual pretende aportar información extra a usuarios al momento de usar buses de transporte, se empleará el enfoque de desarrollo de software llamado prototipado evolutivo el cual permite un mayor entendimiento del propósito, diseño y requerimientos del sistema, así se obtendrá un sistema basado en lo que el usuario realmente necesita.

Se crearán “bocetos” o prototipos rápidos los cuales permitan mostrarle al usuario las funcionalidades más destacadas del software y de acuerdo con la realimentación obtenida por este, se continuará desarrollando el software evolucionando de acuerdo con los requerimientos y corrección oportuna de errores presentados al momento de las pruebas directas. Así involucraremos al usuario en las diferentes fases del proceso. Esto se convertirá en un ciclo que se repetirá hasta obtener un sistema robusto y funcional del prototipo de software.

Las fases por seguir para el desarrollo del prototipo de software serán las siguientes:

1.3.1 Fase de diseño y redacción de requerimientos.

Se realizará la obtención de requerimientos funcionales y no funcionales de acuerdo con los requisitos y características que debe cumplir el prototipo de software.

Después de haber analizado a mano y papel los requisitos que se tienen para nuestro sistema, Se diseñaran diferentes diagramas los cuales permitan armar el cascarón bajo el cual se estará implementando el código más adelante, tales como diagrama de casos de usos, entidad-relación, base de datos, actividades.

1.3.2 Primer prototipo.

Se utilizará el diseño realizado en la fase anterior como base para la construcción nuestro primer prototipo, durante esta fase se implementará y documentará el inicio de nuestro prototipo de software, el registro e inicio de sesión de los diferentes roles en este. Se crearán las funcionalidades de agregar y ver rutas, a las cuales se les podrá agregar paradas o estaciones en caso de ser un usuario administrador.

Se implementará las funcionalidades de gestionar buses, rutas, paradas y usuarios, también las vistas necesarias para su ágil y correcto funcionamiento, además de las vistas donde el usuario invitado pueda observar las rutas y sus buses, una vista de inicio del prototipo, las vista que permita el inicio de sesión de los usuarios.

1.3.3 Segundo prototipo.

Se continuará con el diseño del primer prototipo, se arreglarán las falencias encontradas y se mejorará el diseño, de acuerdo con las criticas obtenidas por los usuarios.

Se añadirán las funcionalidades de poder ver el mapa de la ciudad con las respectivas paradas y buses de cada una de las paradas, las cuales podrán ser observadas solo una a la vez.

Se implementará un mapa con ayuda de las APIs de Google Maps donde se ubicarán las diferentes estaciones y paradas de acuerdo con la ruta seleccionada. Luego se añadirá este mapa al segundo prototipo de este software.

1.3.4 Tercer prototipo.

Se continuará con el diseño del primer prototipo, se arreglarán las falencias encontradas y se mejorará el diseño, de acuerdo con las críticas obtenidas por los usuarios.

Se insertará la funcionalidad de buscar la ruta con la que el usuario final podrá ingresar una ubicación de destino, y en el mapa se mostrará una trayectoria desde su ubicación actual hasta una parada o estación cercana donde podrá tomar la ruta que lo llevará a su destino, la cual también será indicada.

La opción por parte del conductor de mostrar su posición mediante GPS, la cual será compartida con su bus asociado y luego será visualizada en el mapa en tiempo real, y su pestaña de administración.

2 Objetivos

2.1 Objetivo general

Desarrollar un prototipo software en entorno web para la generación de informes del posicionamiento de rutas y ubicación en tiempo real a través de GPS de buses de transporte masivo.

2.2 Objetivos específicos

- Identificar y especificar un conjunto de requisitos que conlleve a un diseño de una aplicación para la comunidad de usuarios de transporte masivo.
- Diseñar, en UML (casos de uso, diagramas de actividades) las funcionalidades principales del prototipo, y arquitectura de software y la estructura de datos para el prototipo a partir de los requerimientos propuestos.
- Implementar el prototipo de software a partir de los diseños propuestos.
- Validar por medio de un conjunto de pruebas software el prototipo desarrollado.

3 Marco de referencia

3.1 Estado del arte

A continuación, se describen algunas de las aplicaciones relacionadas con este prototipo.

3.1.1 EasyTaxi.

La aplicación conecta a los taxistas y pasajeros, permitiéndoles experimentar un viaje rápido, conveniente y seguro, con tan solo tocar un botón. Easy Taxi está disponible para dispositivos Android, iOS, Windows Phone y BlackBerry, así como para clientes B2B a través de las soluciones Easy Taxi PRO y Easy Taxi Corporate. La aplicación permite visualizar la ubicación de los diferentes taxis asociados a está, muestra los más cercanos al usuario que lo solicita, también calcula la ruta, el recorrido y el tiempo aproximado del punto de partida hasta el punto de llegada (EasyTaxi, 2020).

3.1.2 UBER.

Es una aplicación para compartir viajes rápidos y fiables en cuestión de minutos, de día o de noche. No hace falta aparcar ni esperar taxis o autobuses. Con Uber solo tienes que pulsar un botón para pedir un viaje y puedes pagarlo fácilmente con tarjeta de crédito o en efectivo en determinadas ciudades. La aplicación permite visualizar la ubicación de los diferentes conductores de UBER, muestra los más cercanos al usuario que lo solicita, también calcula la ruta, el recorrido, el tiempo y el costo de la carrera aproximado del punto de partida hasta el punto de llegada (Uber, 2020).

3.1.3 Transmisitp | Android | IOS.

Con esta aplicación se accede a la información de cada una de las rutas del Sistema Integrado de Transporte de Bogotá, incluyendo estaciones, buses, horarios, alimentadores y mapas. Incluye una completa guía de las rutas del Transmilenio, Sitp urbano, complementario y especial, además, visualiza mapas y tablas. Con el GPS de los dispositivos, ubica la estación más cercana o los puntos de recarga de las tarjetas para abordar a las estaciones y vehículos (MinTIC, 2017).

3.1.4 TAPPSI.

Es una aplicación móvil creada en Colombia para teléfonos celulares que permite pedir un taxi asociado a esa empresa ingresando la ubicación actual para que el servicio llegue al lugar exacto de forma rápida y segura. También se puede acceder a la información del conductor como celular, placas, nombre y lugar de recorrido del taxi para mayor seguridad. Inicialmente, las personas que no contaran con un celular inteligente o con internet en sus móviles, no podían usar el servicio. Sin embargo, recientemente la compañía desarrolló una versión web para todos aquellos que no cuentan con un teléfono inteligente o con internet móvil disponible (Wikipedia.com, 2020).

3.1.5 MOOVIT.

Moovit es una aplicación de transporte público y un servicio de mapeo desarrollado por la compañía de software israelí Moovit Inc. La aplicación incluye planificación de viajes, horarios de salida y llegada en vivo, itinerarios actualizados, mapas de paradas y estaciones locales, alertas de servicios, y avisos que podrían afectar algún viaje (Wikipedia.com, 2020).

3.2 Marco Contextual

A continuación, algunos conceptos que fueron relevantes y ayudaron en el transcurso de la realización de este proyecto.

3.2.1 BRT

Es un sistema de transporte público masivo basado en autobuses, diseñado específicamente con servicios e infraestructuras para mejorar el flujo de pasajeros. Tiene como objetivo combinar la capacidad y la velocidad del tren ligero o del metro, junto con la flexibilidad, menor costo y la simplicidad de un sistema de autobuses. Los sistemas BRT se caracterizan por transitar generalmente por carriles exclusivos en zonas urbanas (Wikipedia.org, 2020).

3.2.2 Metrolínea

El Sistema Integrado de Transporte Masivo Metrolínea, SITM, es un sistema de transporte que entró en funcionamiento en febrero de 2010, con un periodo de un mes de prueba para que la ciudadanía fuera conociendo el sistema en el área metropolitana de Bucaramanga.

Fue concebido para mejorar la movilidad vehicular de la ciudad y su área metropolitana, además de reducir la contaminación de la ciudad y municipios circunvecinos. Se basa en el concepto de autobús de tránsito rápido, con paradas fijas en estaciones exclusivas (Metrolinea.gov.co, 2020).

3.2.3 API

Es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones. Permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero (RedHat, 2020).

3.2.4 Framework

Se podría traducir aproximadamente como marco de trabajo, es el esquema o estructura que se establece y que se aprovecha para desarrollar y organizar un software determinado. Esta definición, algo compleja, podría resumirse como el entorno pensado para hacer más sencilla la programación de cualquier aplicación o herramienta actual (Neoattack, 2020).

3.2.5 Geolocalización.

La geolocalización consiste en obtener la ubicación geográfica de un objeto como puede ser un teléfono móvil, un coche o una calle. Para ello se puede utilizar diferentes métodos como por ejemplo comprobar el código postal de una carta, la dirección IP de un equipo o el sistema GPS de nuestro teléfono móvil. Para obtener la ubicación geográfica aproximada de un smartphone se utiliza un sistema de posicionamiento global. El sistema está formado por una red de satélites geoestacionarios que dan cobertura a toda la Tierra. Para obtener la ubicación el dispositivo se conecta como mínimo con 3 satélites, de estos satélites recibe un identificador y la hora de cada uno ellos. El dispositivo calcula el tiempo que tarda en llegar la señal desde los satélites y gracias al retardo o delay resultante se obtiene la ubicación por medio de la triangulación (OSI.es, 2016).

3.2.6 Sistema de información.

Es un conjunto de componentes que interaccionan entre sí para alcanzar un fin determinado, el cual es satisfacer las necesidades de información de dicha organización. Estos componentes pueden ser personas, datos, actividades o recursos materiales en general, los cuales procesan la información y la distribuyen de manera adecuada (Pérez Porto & Gardey, 2008).

3.2.7 Frontend.

Es la parte de un sitio web que interactúa con los usuarios, por eso decimos que está del lado del cliente (Chapaval, 2018).

3.2.8 Backend.

Es la parte que se conecta con la base de datos y el servidor que utiliza dicho sitio web, por eso decimos que el backend corre del lado del servidor (Chapaval, 2018).

3.3 Marco legal

3.3.1 Google Maps API Key.

Para utilizar los productos de Google Maps Platform, la facturación debe estar habilitada en su cuenta de Google y todas las solicitudes deben incluir una clave API válida. La clave API es un identificador único que se utiliza para autenticar las solicitudes asociadas con su proyecto para fines de uso y facturación (Google Maps, 2020).

El uso de “Directions API”, un servicio de Google Maps, implica un cobro de 5 USD cada 1000 peticiones a esta API. Debido a la gran cantidad de peticiones realizadas en las pruebas y que el cobro por esta ascendiera 5USD/día mientras se encontraba activo este servicio se ha decidido mantener habilitada esta función solo en los momentos que sea necesario, como en futuras pruebas y al momento de mostrar la funcionalidad del prototipo. Deshabilitar esta función implica que ya no se podrán calcular rutas ni ser mostradas en el mapa de Google Maps.

3.4 Marco tecnológico

A continuación, se describe el conjunto de herramientas usadas en el desarrollo e implementación del proyecto.

3.4.1 Mean Stack.

Es el conjunto de tecnologías necesarias para el desarrollo de todas las capas de una aplicación web con JavaScript. Está compuesto por cuatro de las más importantes tecnologías de la industria: MongoDB, Express, Angular y Node.js (Cortés Pineda, 2018) .

3.4.2 Visual Studio Code.

Es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP) y tiempos de ejecución (VisualStudioCode, 2020).

3.4.3 Bootstrap.

Es un kit de herramientas de código abierto o framework para desarrollar con HTML, CSS y JS (Bootstrap, 2020).

3.4.4 Materialized.

Es un Framework para desarrolladores Frontend, que agiliza el trabajo para maquetar y crear vistas en un proyecto, este Framework trabaja bajo el concepto y diseño de Material Design de Google. Este Framework nos ofrece un set de utilidades y componentes listos para implementarlos en nuestro proyecto (BlogNubeColectiva, 2018).

3.4.5 API de Google Maps.

Se utiliza para mostrar los mapas de Google en nuestras páginas webs.

3.4.6 Directions API.

Servicio de Google Maps que permite el acceso a las rutas de conducción, ciclismo, caminata y transporte público con la API de indicaciones utilizando una solicitud HTTP. Los waypoints ofrecen la capacidad de alterar una ruta a través de una ubicación específica. Especifique los orígenes, destinos y puntos de referencia como cadenas de texto (por ejemplo, "Chicago, IL" o "Darwin, NT, Australia") o como coordenadas de latitud / longitud (Google.com, 2020).

3.4.7 API de geolocalización.

Servicio de Google Maps que permite encontrar una ubicación y un radio de precisión basados en información de cosas como torres de celulares y puntos de acceso Wifi que un cliente móvil puede detectar con la API de geolocalización. Esto se usa principalmente cuando el GPS no es posible o apropiado (Google.com, 2020).

3.4.8 Node Js.

Es un entorno JavaScript de lado de servidor que utiliza un modelo asíncrono y dirigido por eventos, está diseñado para construir aplicaciones en red escalables (NodeJS, 2020).

3.4.9 Express.

Es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles (ExpressJS, 2020).

3.4.10 MongoDB.

Es una base de datos orientada a documentos. Esto quiere decir que, en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON (Rubenfa, 2014).

3.4.11 Mongoose.

Es un marco de JavaScript que se usa comúnmente en una aplicación Node.js con una base de datos MongoDB, proporciona una increíble cantidad de funcionalidades para crear y trabajar con esquemas (Munro, 2017).

3.4.12 Angular.

Es un marco de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones de una página eficientes y sofisticadas (Angular.io, 2020), Un framework para aplicaciones web

desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página (Wikipedia.org, 2020).

3.4.13 Angular route guards.

Los protectores de navegación del enrutador angular permiten otorgar o eliminar el acceso a ciertas partes de la navegación (Alligator.io, 2017). Decisión tomada mediante un boolean que es retornado de una clase que implementa la interfaz guard.

3.4.14 Postman.

Está compuesto por diferentes herramientas y utilidades que permiten realizar tareas diferentes dentro del mundo API REST: creación de peticiones a APIs internas o de terceros, elaboración de tests para validar el comportamiento de APIs, posibilidad de crear entornos de trabajo diferentes (con variables globales y locales), y todo ello con la posibilidad de ser compartido con otros compañeros (Redondo, 2017).

3.4.15 Lucidchart.

Es un espacio de trabajo visual que combina diagramas, visualización de datos y colaboración para acelerar el entendimiento e impulsar la innovación (Lucidchart.com, 2020).

3.4.16 Google Analytics.

Es una solución analítica cuantitativa para medir la cantidad de clics, indicar el origen de los visitantes e informar sobre la navegación de los usuarios. También permite establecer objetivos para su sitio web (Chardonneau, 2017).

3.4.17 Autenticación basada en JWT.

Es un estándar abierto (RFC-7519) basado en JSON para crear un token que sirva para enviar datos entre aplicaciones o servicios y garantizar que sean válidos y seguros. El caso más común de uso de los JWT es para manejar la autenticación en aplicaciones móviles o web. Para esto cuando el usuario se quiere autenticar manda sus datos de inicio de la sesión al servidor, este genera el JWT y se lo manda a la aplicación cliente, luego en cada petición el cliente envía este token que el servidor usa para verificar que el usuario este correctamente autenticado y saber quién es (Sergio, 2017).

3.4.18 GitHub.

Es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador (Wikipedia.org, 2020).

3.4.19 Heroku.

Se utiliza para el despliegue de una aplicación. Además, permite manejar los servidores y sus configuraciones, escalamiento y la administración (Celis, 2018).

3.4.20 MongoDB Atlas.

Es la base de datos como servicio que permite implementar, utilizar y escalar una base de datos de MongoDB en la nube con tan solo unos pocos clics (MongoDB.com, 2020).

3.4.21 Bcryp.

Es una función de hashing de contraseñas diseñado por Niels Provos y David Maxieres, basado en el cifrado de Blowfish. Lleva incorporado un valor llamado salt, que es un fragmento aleatorio que se usará para generar el hash asociado a la contraseña, y se guardará junto con ella en la base de datos. Así se evita que dos contraseñas iguales generen el mismo hash y los problemas que ello conlleva (Vicente, 2017).

4 Resultados

4.1 Requerimientos funcionales y no funcionales.

Estos requerimientos fueron obtenidos en la primera fase de la creación del prototipo software de acuerdo con los requisitos y características que debe cumplir el prototipo de software.

4.1.1 Requerimientos funcionales.

Las actividades del software identificadas en el análisis de las validaciones realizadas a distintas personas.

Tabla 1. *Lista de requerimientos funcionales.*

#	Nombre	Acción	Actor
RF01	Registrar usuario	Añadirá un usuario a la base de datos, el cual puede ser de tipo administrador o tipo conductor	Administrador
RF02	Gestionar usuario	Permite registrar, editar e inhabilitar usuarios de tipo conductor	Administrador
RF03	Login usuario	Permite a un usuario acceder a su cuenta.	Administrador. Conductor del bus
RF04	Logout usuario	Permite a un usuario salir de su cuenta.	Administrador Conductor del bus
RF05	Agregar ruta	Permite añadir nuevas rutas.	Administrador
RF06	Gestionar rutas	Permite editar, filtrar e inhabilitar rutas.	Administrador
RF07	Agregar bus	Permite crear buses.	Administrador
RF08	Gestionar buses	Permite visualizar, filtrar editar e inhabilitar buses.	Administrador
RF09	Asignar buses a ruta	Permite asignar un bus a una determinada ruta.	Administrador
RF10	Gestionar asignaciones de ruta	Permite cambiar la ruta actual de un bus por una diferente.	Administrador

#	Nombre	Acción	Actor
RF11	Enviar la ubicación mediante GPS	Permite transmitir en tiempo real la ubicación del bus	Conductor del bus
RF12	Visualizar ruta seleccionada en mapa (Google Maps)	Permite ver los diferentes buses asignados a una ruta en el mapa.	Invitado, administrador, Conductor del bus
RF13	Filtrar búsqueda de ruta	Permite buscar mediante caracteres una ruta en un arreglo de rutas.	Invitado, administrador, conductor del bus
RF14	Observar la vista principal	Permite visualizar la vista principal.	Invitado, administrador, conductor del bus
RF15	Observar rutas	Permite visualizar la vista con las diferentes rutas.	Invitado, administrador, conductor del bus
RF16	Acceder a una ruta	Permite visualizar las características de una ruta en específico.	Invitado, administrador, conductor del bus
RF17	Filtrar buses por ruta	Buscar buses asociados a una ruta específica.	Invitado, administrador, conductor del bus
RF18	Gestionar paradas	Permite editar, filtrar e inhabilitar paradas, además, observar un informe de estas.	Administrador
RF19	Asignar conductor a un bus.	Asigna un usuario de tipo conductor a bus.	Administrador
RF20	Gestionar asignaciones de conductores	Permite cambiar el conductor actual de un bus por uno diferente.	Administrador
RF21	Gestionar ubicación de buses en mapa	Actualizar información en tiempo real, la posición de los buses en mapa y base de datos.	Conductor
RF22	Asignar/Quitar rutas a paradas	Permite asignar o quitar rutas a una parada.	Administrador

4.1.2 Requerimientos no funcionales.

A continuación, se presentan las necesidades y restricciones que se pudieron captar de las validaciones realizadas a los usuarios.

Las especificaciones de cada uno de los requerimientos no funcionales se encuentran consignados en la sesión de Apéndices A.

Tabla 2. *Lista de requerimientos no funcionales.*

Identificador (RNF#)	Nombre
RNF01	Diseñar una interfaz gráfica de usuario
RNF02	Operar con varios usuarios al tiempo.
RNF03	Operar con varios buses y rutas al tiempo.
RNF04	Actualizar información en la base de datos.
RNF05	Compatibilidad del prototipo con internet Explorer y Google Chrome.

4.2 Diseño del prototipo

Se presenta el diseño del prototipo siguiendo los requerimientos mencionados en las tablas 1 y 2, los cuales fueron requeridos para realizar los casos de uso, diagramas de actividades, diagramas clases y arquitectura software.

4.2.1 Casos de uso.

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema (sistemas.com, 2020).

4.2.1.1 Actores.

Un actor es una entidad externa al sistema que se modela y que puede interactuar con él; los actores en este prototipo son presentados en la tabla 3.

Tabla 3. *Relación de los actores con cada uno de los requerimientos.*

USUARIO	DESCRIPCIÓN	RF#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	Administrador	Este usuario puede iniciar y cerrar sesión como administrador. Teniendo las funcionalidades principales del prototipo.	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X
2	Conductor	Este usuario puede iniciar y cerrar sesión como conductor de bus. Además, tiene la funcionalidad especial de compartir o dejar de compartir su ubicación en tiempo real a su bus asociado, la cual podrán ver los demás actores.			X	X							X	X	X	X	X	X	X					X
3	Invitado	No debe registrarse, y no tiene la necesidad de iniciar sesión. Puede acceder al prototipo y visualizar la ubicación de los diferentes buses filtrándolos por rutas, así como observar la vista principal.												X	X	X	X	X	X					

Una vez determinados los usuarios del software se prosiguió a identificar las funcionalidades principales del software realizar los diagramas de casos de uso siguientes. A continuación, se muestra el caso de uso principal del prototipo.

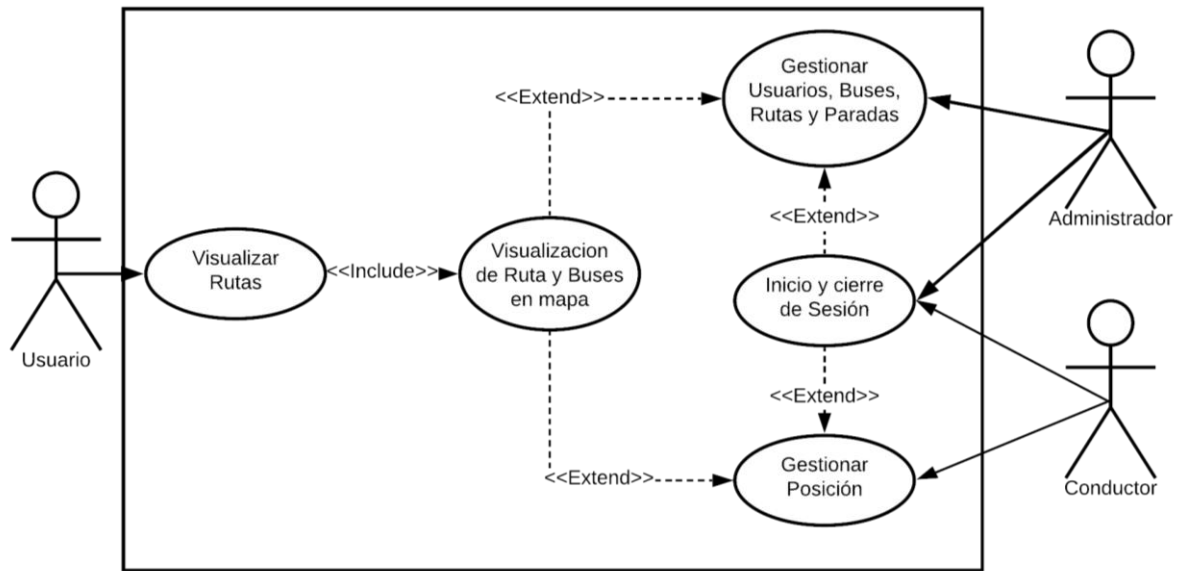


Figura 1. Diagrama de casos de uso principal del prototipo.

A continuación, se muestra el caso de uso interacción con la interfaz web, el cual muestra las funciones principales del prototipo.

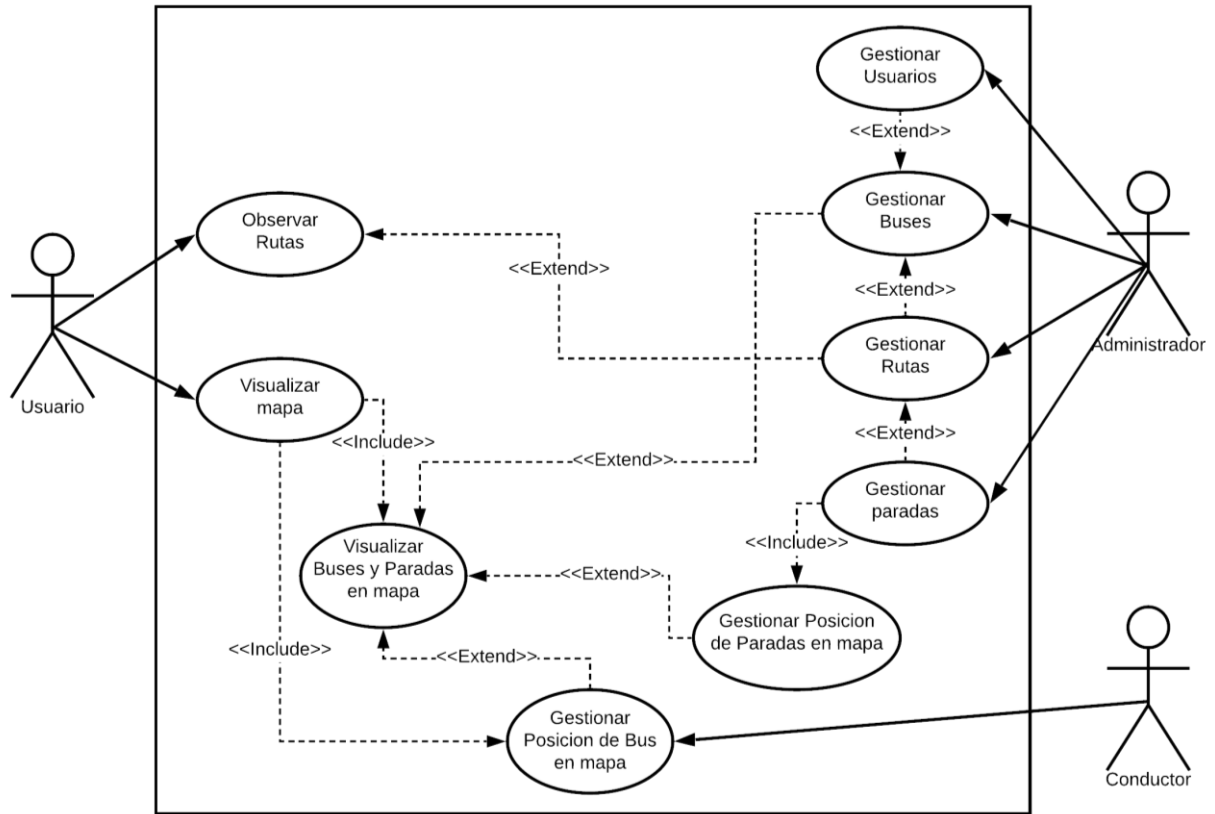


Figura 2. Diagrama de casos de uso de la interacción con la interfaz web.

A continuación, se muestra el caso de uso gestionar conductores, rutas y buses, el cual muestra como el usuario de tipo administrador puede crear, registrar, editar e inhabilitar estos.

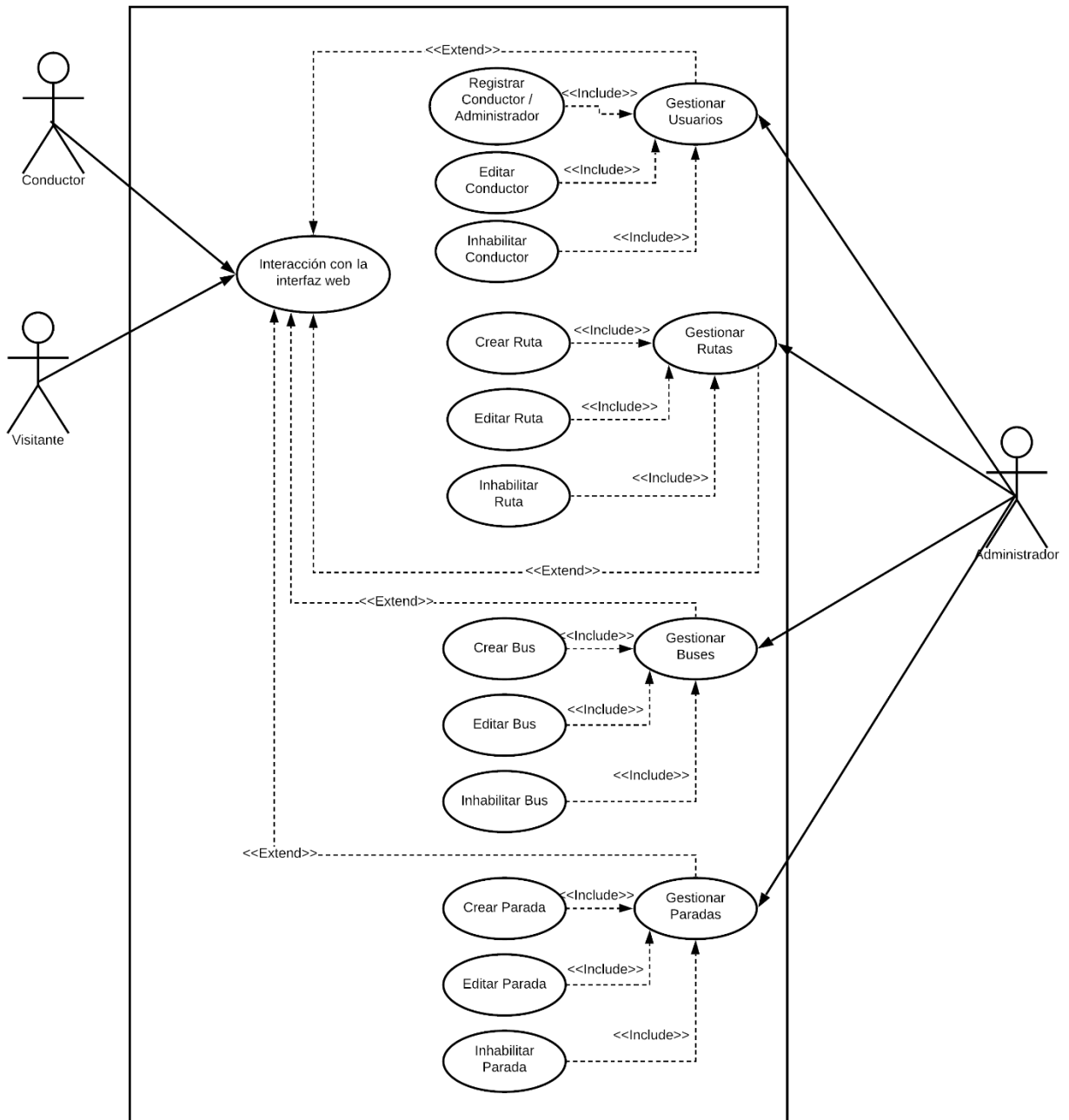


Figura 3. Diagrama de casos de uso de gestionar conductores, rutas buses y paradas.

4.2.2 Diagrama de actividades.

A continuación, se presenta el diagrama de actividades general de la funcionalidad interactuar con la interfaz web para todos los usuarios. Consiste en toda la funcionalidad final la cual está dirigida a los usuarios de tipo invitado. administradores y conductores también pueden acceder a ella y otras funciones que serán especificadas en los siguientes diagramas de actividades.

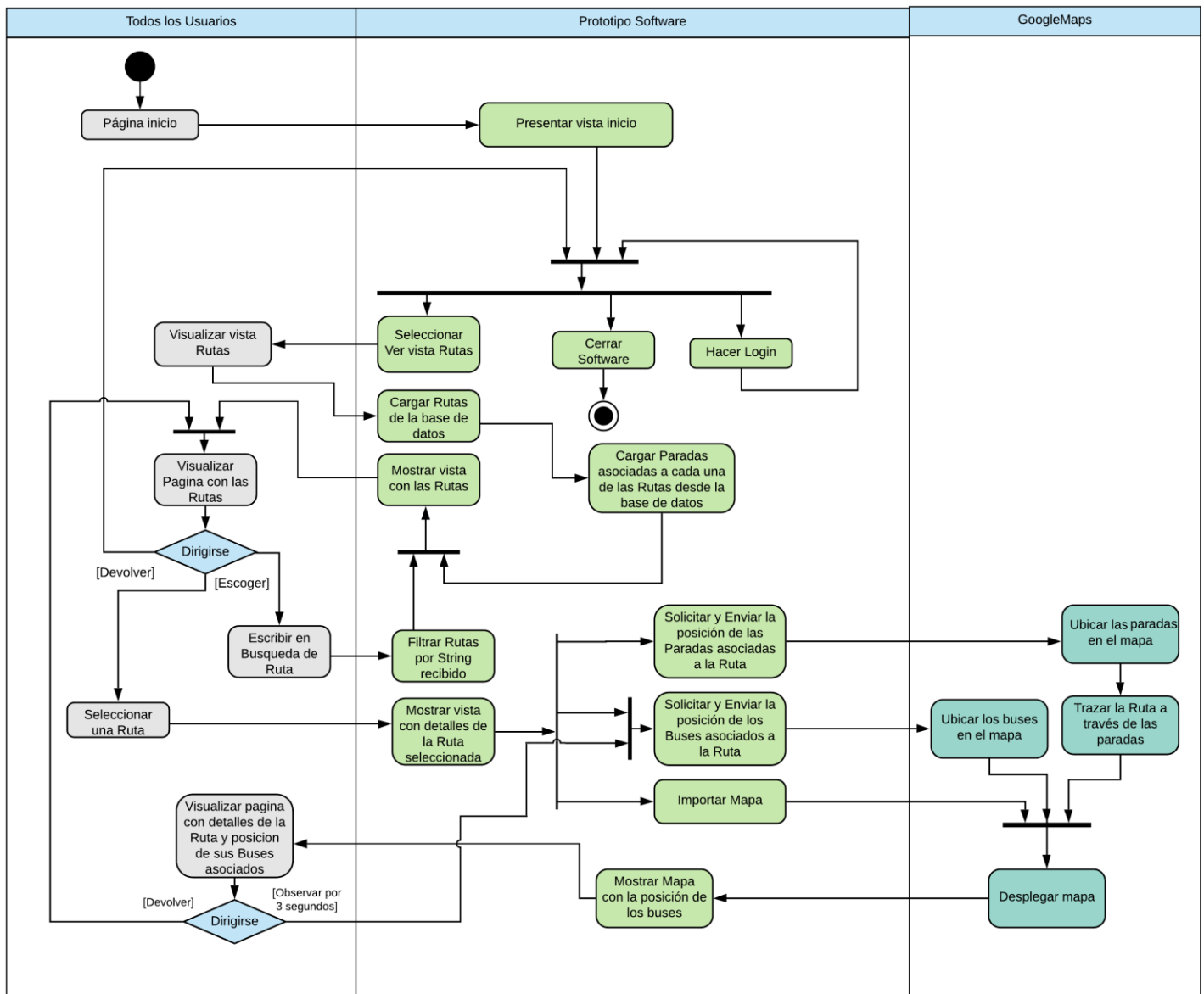


Figura 4. Diagrama de actividades ver rutas y buses asociados a esta en el mapa.

El siguiente diagrama muestra la funcionalidad principal del usuario conductor, enviar su posición mediante GPS a su bus asociado. Esta funcionalidad es única para el usuario de tipo conductor de bus.

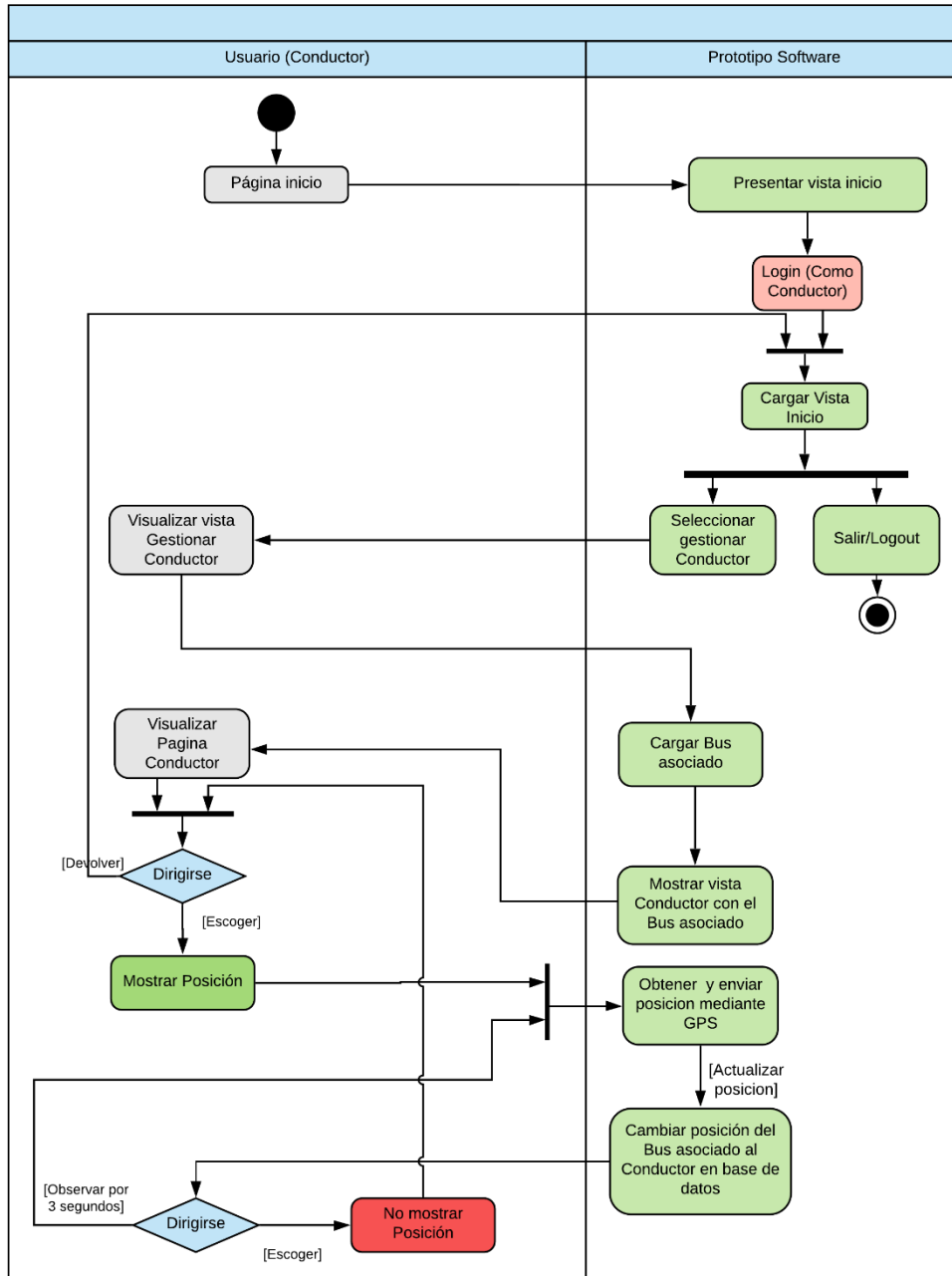


Figura 5. Diagrama de actividades de la funcionalidad enviar la posición de un conductor a su bus asociado.

El siguiente diagrama muestra la funcionalidad de los usuarios administrador, crear y gestionar rutas. Esta funcionalidad es única para usuarios de tipo administrador.

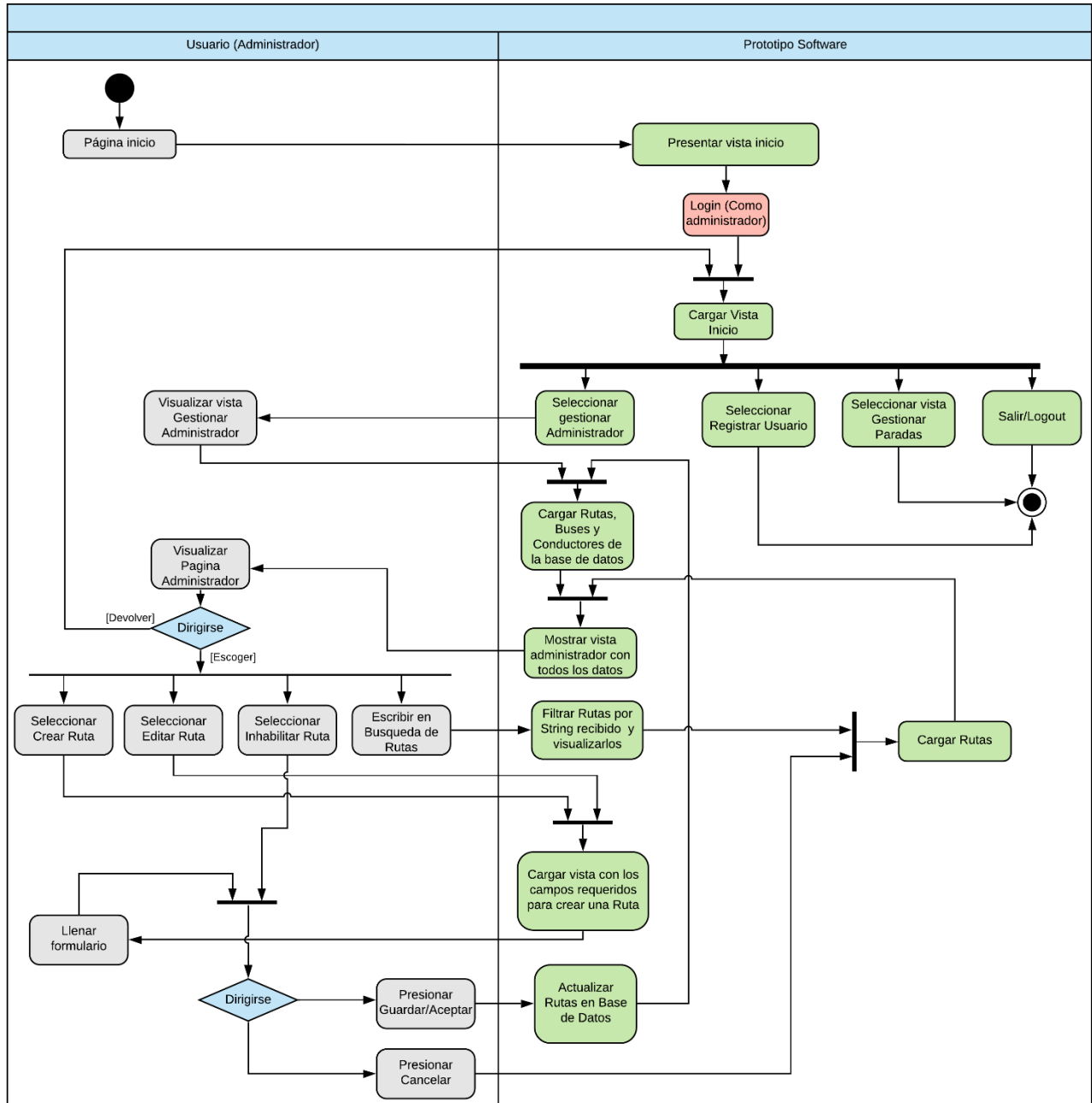


Figura 6. Diagrama de actividades de la funcionalidad Gestionar rutas para usuarios de tipo administrador.

El siguiente diagrama muestra la funcionalidad de los usuarios administrador, crear y gestionar usuarios de tipo conductor. Esta funcionalidad es única para usuarios de tipo administrador.

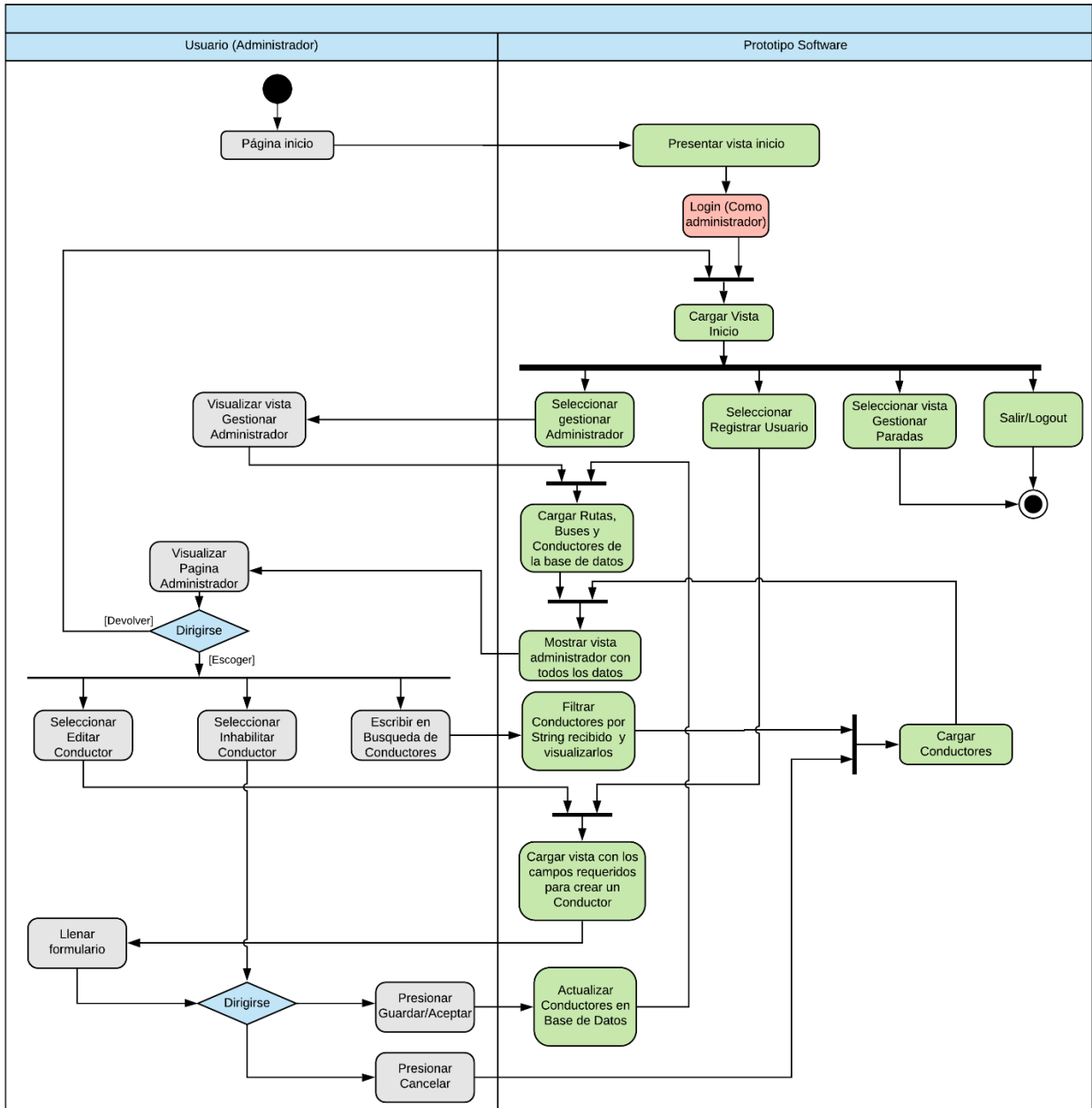


Figura 7. Registrar usuarios y gestionar usuarios de tipo conductor para usuarios de tipo administrador.

El siguiente diagrama muestra la funcionalidad de los usuarios administrador, crear y gestionar usuarios de tipo conductor. Esta funcionalidad es única para usuarios de tipo administrador.

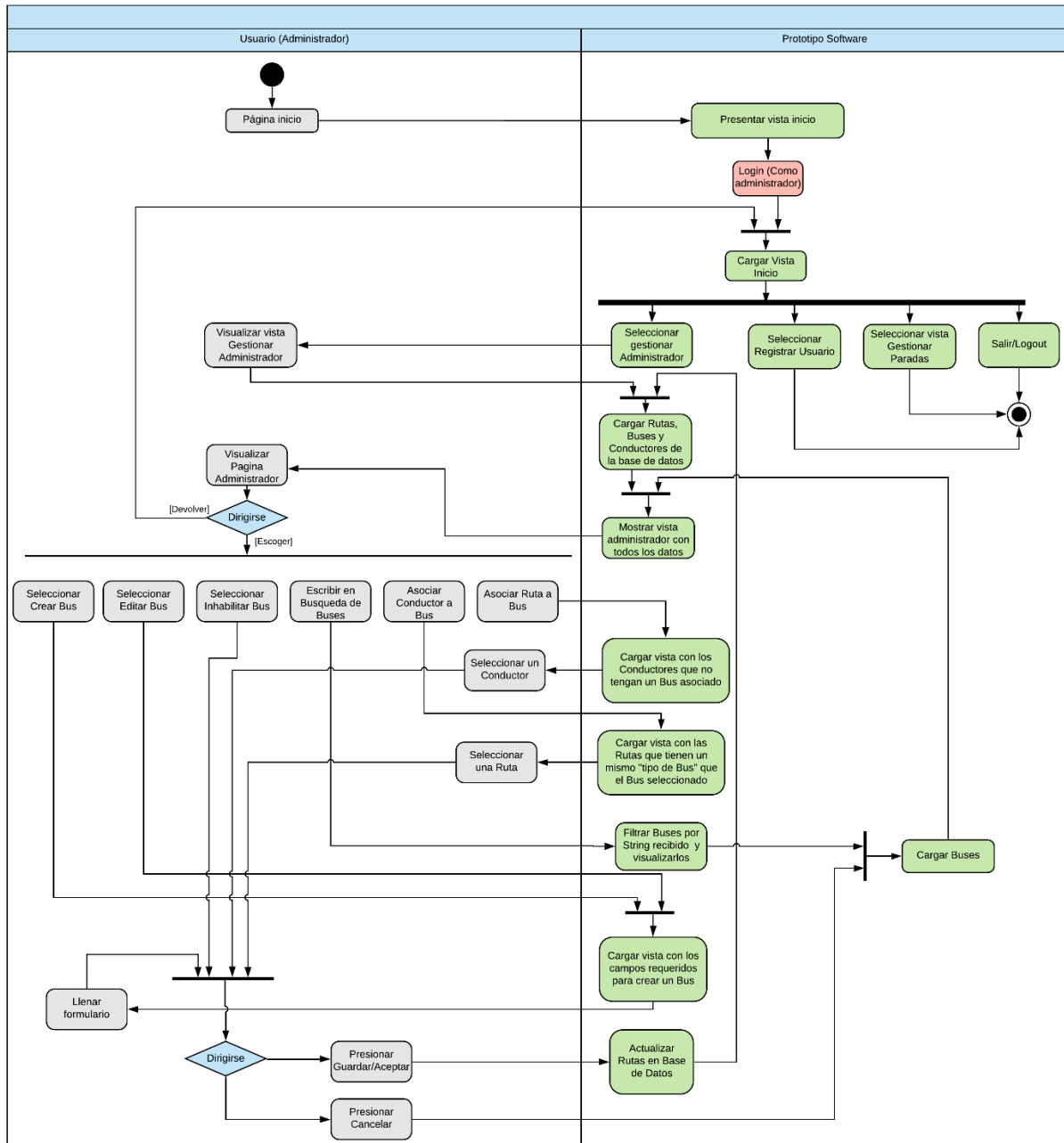


Figura 8. Crear y gestionar buses para usuarios de tipo administrador.

4.2.3 Diagrama de clases

Es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos (Wikipedia.org, 2020).

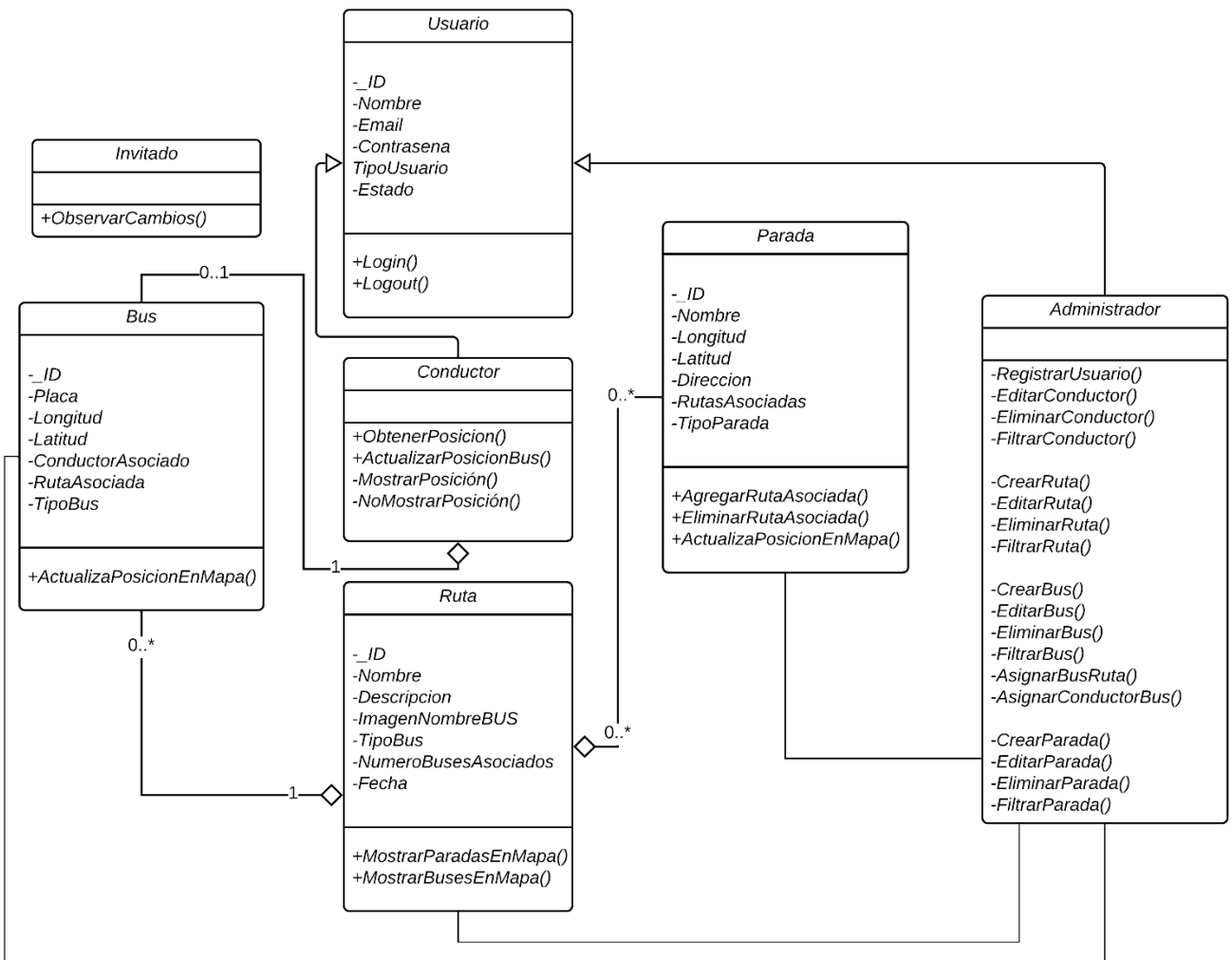


Figura 9. Diagrama de clases del prototipo.

4.2.4 Arquitectura software.

Es un conjunto de patrones que pueden ser utilizados como guía o marco de referencia para crear, diseñar, evaluar o construir un software.

4.2.4.1 Patrón de diseño Modelo -Vista-Controlador (MVC).

El patrón de diseño de modelo-vista-controlador (MVC) especifica que una aplicación consta de un modelo de datos, de información de presentación y de información de control. El patrón requiere que cada uno de estos elementos esté separado en distintos objetos.

- *El modelo.* Contiene únicamente los datos puros de aplicación; no contiene lógica que describe cómo pueden presentarse los datos a un usuario.
- *La vista.* Presenta al usuario los datos del modelo. La vista sabe cómo acceder a los datos del modelo, pero no sabe el significado de estos datos ni lo que el usuario puede hacer para manipularlos.
- *El controlador.* Está entre la vista y el modelo. Escucha los sucesos desencadenados por la vista (u otro origen externo) y ejecuta la reacción apropiada a estos sucesos. En la mayoría de los casos, la reacción es llamar a un método del modelo. Puesto que la vista y el modelo están conectados a través de un mecanismo de notificación, el resultado de esta acción se reflejará automáticamente en la vista (*IBM.com, 2020*).

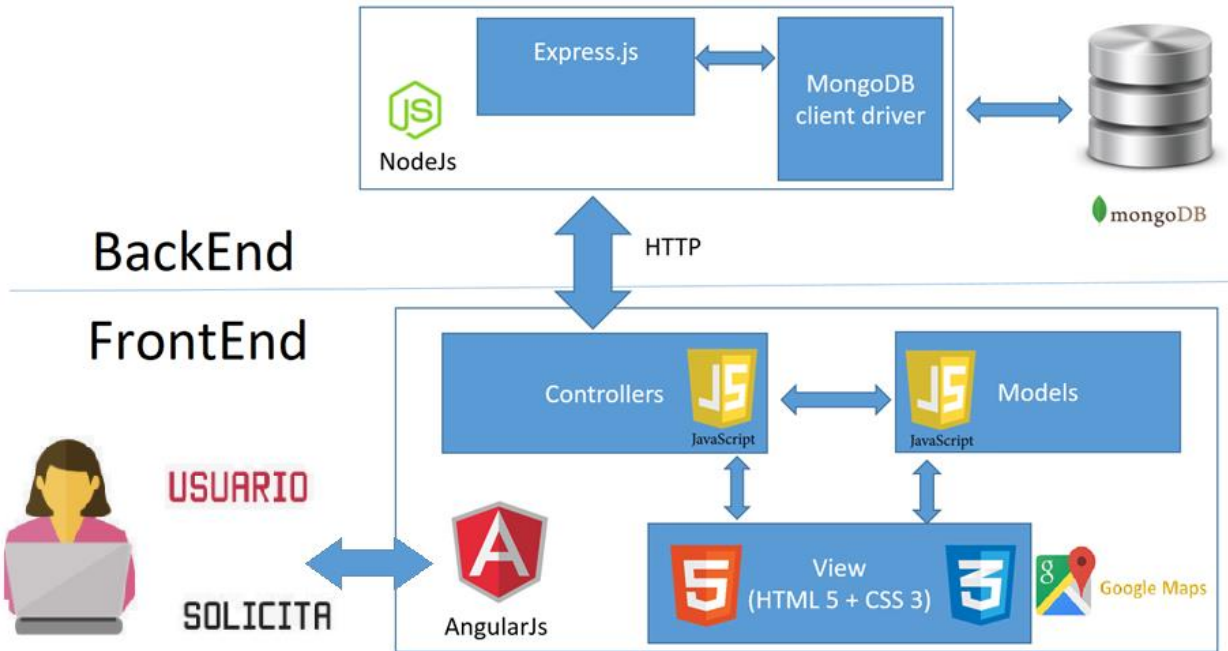


Figura 10. Modelo-Vista-Controlador del prototipo.

Adaptado de (Guerrero, 2019)

4.3 Implementación

En la figura 11 se aprecia una muestra del código del prototipo, en donde se observa a la izquierda, la barra de elementos que componen el proyecto, organizada según el framework Angular. En la carpeta “Components” se aprecian las clases que fueron expresadas en la Figura 9 junto con otras clases necesarias para la interfaz web.

Además, en el panel de codificación se nota un estándar de programación en la forma de codificar: líneas 1-7 la importación de modelos, servicios, clases y componentes, líneas 9-13 es el decorator que convierte una clase a un componente de Angular, en el se define el nombre del componente (línea 10) y su HTML (línea 11) y CSS (línea 12) asociado, declaración variables globales en el componente en las línea 17 y 18, el “Constructor” y sus atributos en la línea 20, el

metodo “ngOnInit” obligado a definir al implementar la interfaz “OnInit” en lineas 23-25 y algunos metodos en las siguientes lineas de codigo, estandar que se cumplio en todo el documento.

```

1 import { NgModule } from '@angular/core';
2 import { Parada } from 'src/app/models/parada';
3 import { RutasService } from 'src/app/services/rutas.service';
4 import { Ruta } from 'src/app/models/ruta';
5
6 @Component({
7   selector: 'app-paradas',
8   templateUrl: './paradas.component.html',
9   styleUrls: ['./paradas.component.css']
10 })
11 export class ParadasComponent implements OnInit {
12
13   Rutas: Ruta[];
14   RutasSinParada: string[] = [];
15
16   constructor(public paradaServices: ParadasService, public rutasService: RutasService) {
17   }
18
19   ngOnInit() {
20     this.cargarBaseDatos();
21   }
22
23   async cargarBaseDatos() {
24     this.rutasService.getRutas();
25     this.paradaServices.getParadas();
26     this.Rutas = this.rutasService.getRutas2();
27   }
28
29   obtenerRutasSinParada() {
30     this.RutasSinParada = [];
31     let agregar: boolean;
32     const array = this.paradaServices.selectedParada.rutasAsociadas;
33     for (const ruta of this.rutasService.getRutas2()) {
34       agregar = true;
35       for (const nombreRutaParada of array) {
36         if (nombreRutaParada === ruta.nombre) {
37           agregar = false;
38         }
39       }
40     }
41   }
42 }

```

Figura 11. Ejemplo del código en el entorno de programación Visual Studio Code y el Framework Angular.

4.3.1 Implementación prototipos.

El prototipo final se encuentra en servidor Heroku y su base de datos en MongoDB Atlas, se puede acceder a este mediante el siguiente enlace: <https://inmetro.herokuapp.com/>

Prototipo uno. Para este prototipo se crearon los diferentes modelos y diagramas necesarios para el diseño del prototipo (Diagramas de actividades, de clase y casos de uso, los cuales han sido

mostrados anteriormente), Además del navbar del prototipo y las primeras vistas de este, como lo son:

Vista rutas: Permite visualizar las rutas existentes mediante cartas, mostrando así un pequeño resumen de estas, y creando un efecto visual agradable.



Figura 12. Visualizar rutas.

Vista Login: Permite a usuarios de tipo administrador y conductor el ingreso al prototipo, mostrando nueva información pertinente al tipo de usuario.

Figura 13. Login usuario.

Vista Register. Permite el registro de nuevos usuarios al prototipo, vista a la que solo puede acceder un usuario de tipo administrador.

Figura 14. Register usuario.

Junto con las vistas también fue diseñado el Backend necesario para que estas funcionaran.

Prototipo dos. Para este prototipo se crearon las vistas:

Vista administrador. Es activada solo cuando un Administrado es logueado, en esta esté usuario puede visualizar, crear, editar e inhabilitar rutas, buses y usuarios de tipo conductor. Además, puede realizar la asignación de rutas a un bus y/o conductor a un bus y observar la información de un bus.

Administrador · Cristian F Inicio Rutas Administrador Registrar Buscar Ruta Buscar Salir

Rutas [Nueva Ruta](#)

Nombre Texto a buscar [Buscar](#)

Nombre	Descripcion	Imagen Ruta	# De Buses	Fecha	Tipo Bus	Operaciones
P10	Kra 15 - E. San Mateo - E. Chorreras		3	2012-01-01	Padron	
P8	Cra 33 - E. UIS - Cra 33		1	02-02-2020	Padron	
P13	Lorem ipsum Lorem ipsum Lorem ipsum		0	2012-01-01	Padron	
T1	Lorem ipsum		1	2012-01-02	Articulado	
AB1	Lorem ipsum		1	2012-01-01	Alimentador	
AB2	Lorem ipsum		0	2012-01-02	Alimentador	

Buses [Nuevo Bus](#)

Placa Texto a buscar [Buscar](#)

Placa	Tipo Bus	Ruta Asociada	Conductor Asociado	Coordenadas	Operaciones
TTT 797	Padron	P10	Cristian	0, 0, 0	
ACX 055	Alimentador	AB1	Camilo Torres	0, 0, 0	
FRT 123	Padron	P10		0, 0, 0	
CVC 444	Articulado	T1	Carolina Castillo	0, 0, 0	
FGF	Padron	P8	Andres	0, 0, 0	
TLU 325	Padron	P10		0, 0, 0	

Conductores

Nombre Texto a buscar [Buscar](#)

Nombre	Email	Contraseña	Tipo	Activo	Operaciones
Cristian	cristian	\$2a\$10\$5VGZkfs80EVku4cb1J2huFcpvx8K83VaVykV.o6296W9uC5TD5SW	Conductor	false	
Camilo Torres	Camilo	\$2a\$10\$dXGhTsAmFeEJUMV19qDYruPlvmxgX4Rj2km1rZSEe2pP8qEjB115q	Conductor	false	
Carolina Castillo	Carolina	\$2a\$10\$BN1FzSWA1.wH2vuKK4IMOUo2R6cyyStBYLDgKbaZTYhvcIDmt17e	Conductor	false	
Andres	Andres	\$2a\$10\$voKx2zsHd10VYKUYwVEvuzpluaBPaiHEVzRW.YwZ/HI3M3ocLWYO	Conductor	false	

Figura 15. Vista administrador.

Vista conductor. Es activada solo cuando un conductor es logueado, esta vista permite al usuario de tipo conductor Conocer cuál es su bus asociado (si lo tiene), además de permitir habilitar la opción de mostrar o no su ubicación la cual será transferida a su bus asociado.



Figura 16. Vista conductor sin mostrar posición.



Figura 17. Vista conductor Mostrando posición.

Vista agregar/editar ruta. Permite agregar y editar rutas, en caso de editar una ruta, al abrirse esta vista son cargados los datos de la ruta a editar.

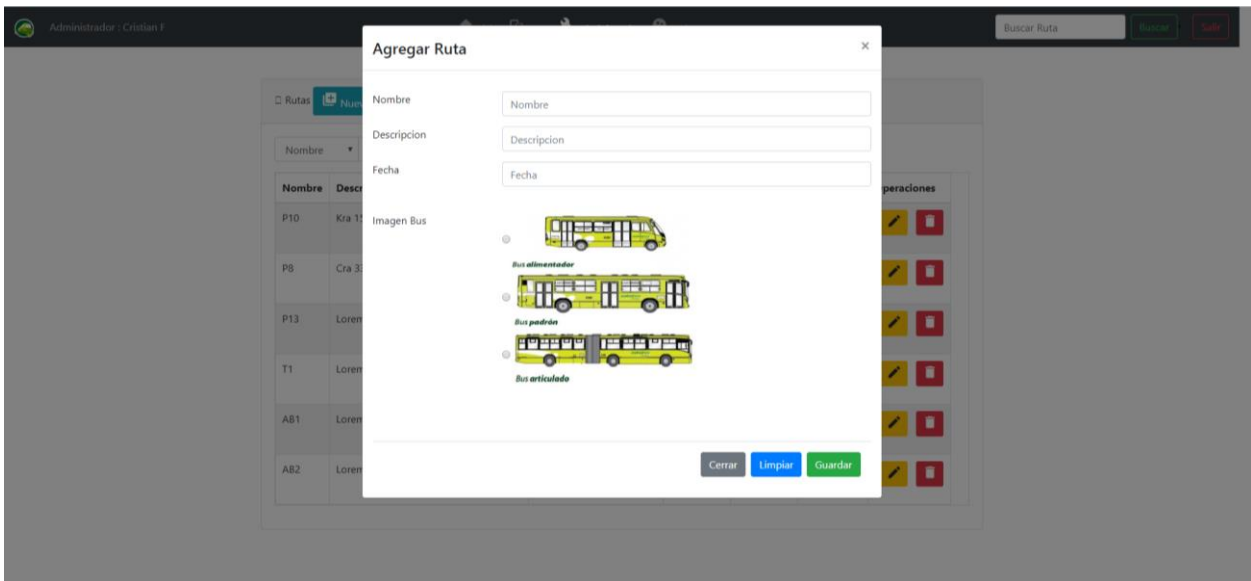


Figura 18. Vista agregar/editar ruta.

Vista agregar/editar bus. Permite agregar y editar buses, en caso de editar un bus, al abrirse esta vista son cargados los datos del bus a editar.

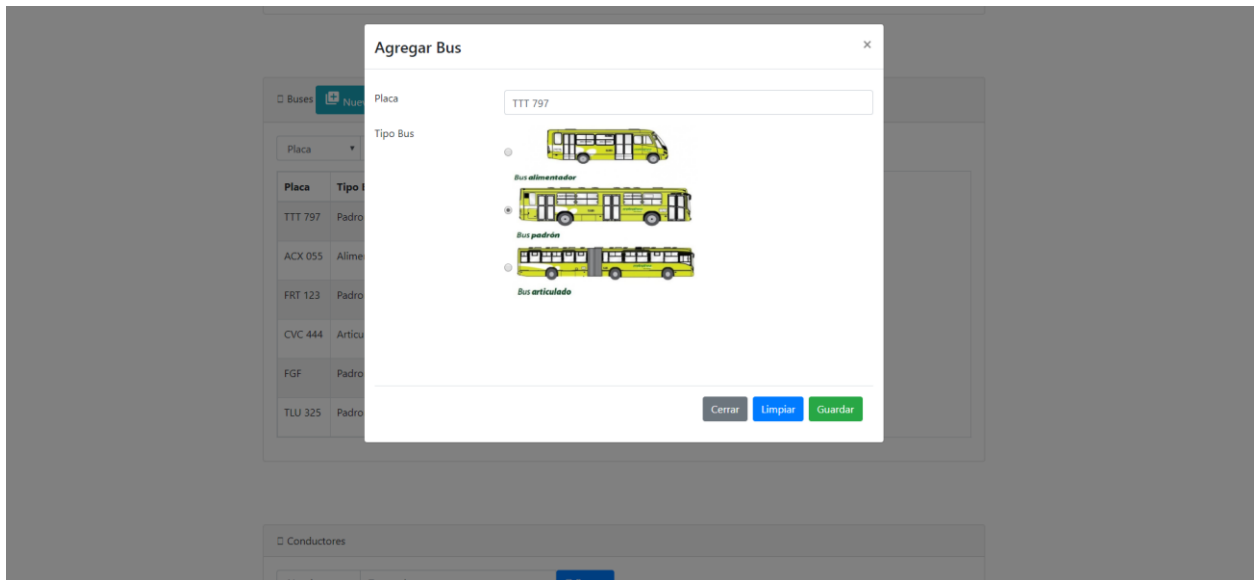


Figura 19. Vista agregar/editar bus.

Vista asignar ruta a bus. Permite la asignación de una ruta a un bus. Esta vista solo muestra las rutas que tienen el mismo “tipo de bus” que el bus. En este caso el “tipo de bus” del bus con placa “CVC 444” es “Articulado” por lo cual la vista solo muestra la ruta T1 que su “tipo de bus” también es “Articulado” (ver figura 21).

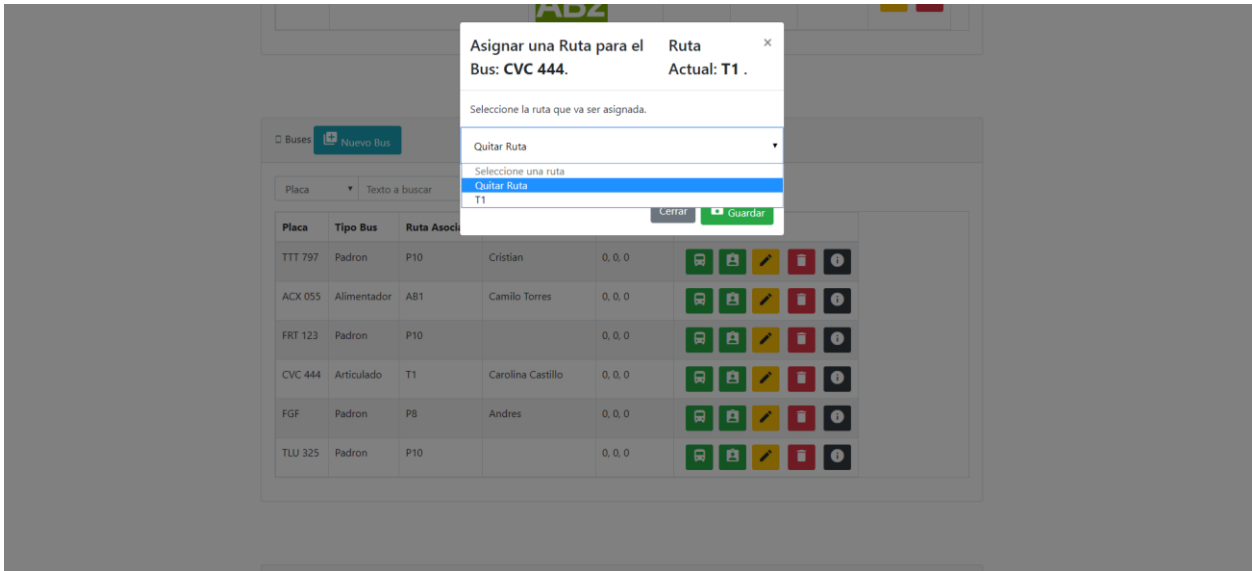


Figura 20. Vista asignar ruta a bus.

Vista asignar conductor a bus. Permite la asignación de un conductor a un bus. Esta vista solo muestra los conductores que aún no tienen asignado un bus.

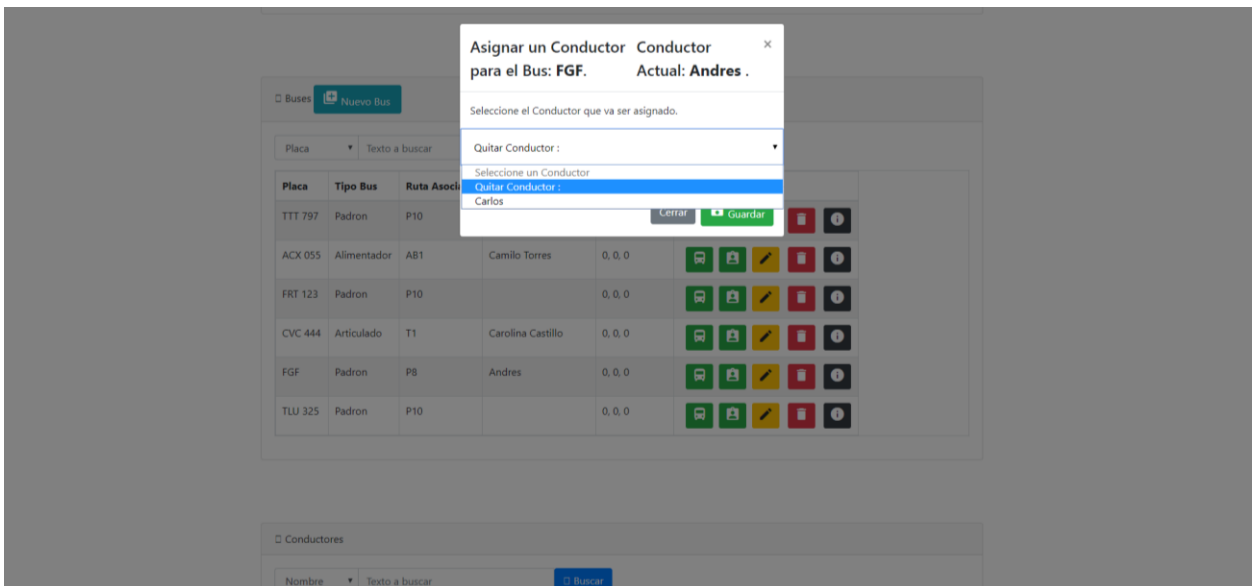


Figura 21. Vista asignar conductor a bus.

Prototipo tres. Para este prototipo se implementó una mejora a la funcionalidad de las vistas anteriores y a la interfaz en algunas de estas. Además, se crearon las vistas:

Vista ruta seleccionada. Permite la visualización de una ruta específico, la cual muestra los buses asociados a esta y su posición en caso de estar activos, también se observa el mapa en el cual se encuentra la trayectoria de la ruta y sus paradas.

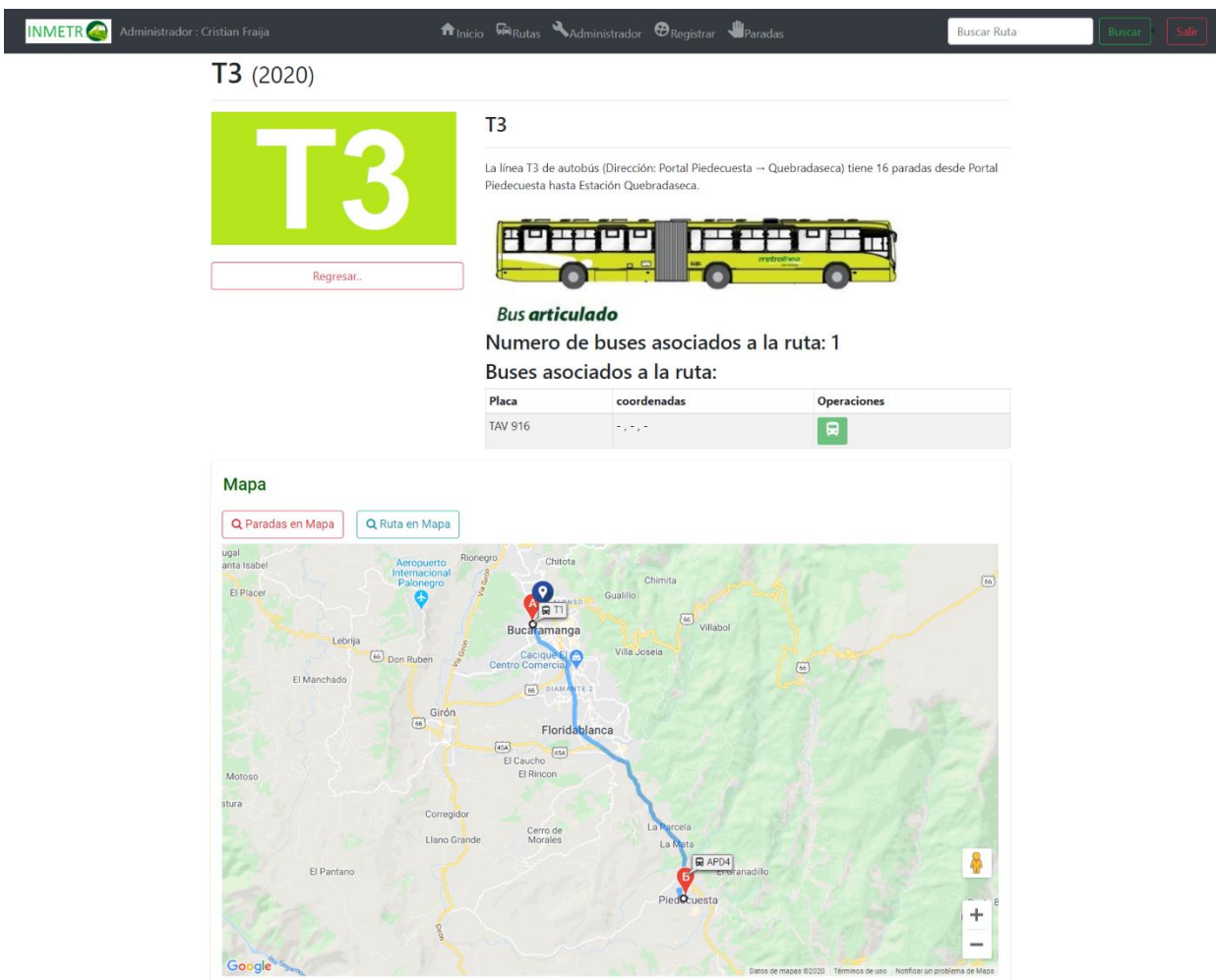


Figura 22. Vista ruta seleccionada.

Vista Home. Presenta el inicio de la interfaz web y también tiene la funcionalidad de buscar una ruta junto con indicaciones para llegar a un destino dado por el usuario.



Figura 23. Vista Home

Vista paradas. En esta vista el usuario administrador gestiona las paradas y les asigna rutas. Presenta informes de las paradas.

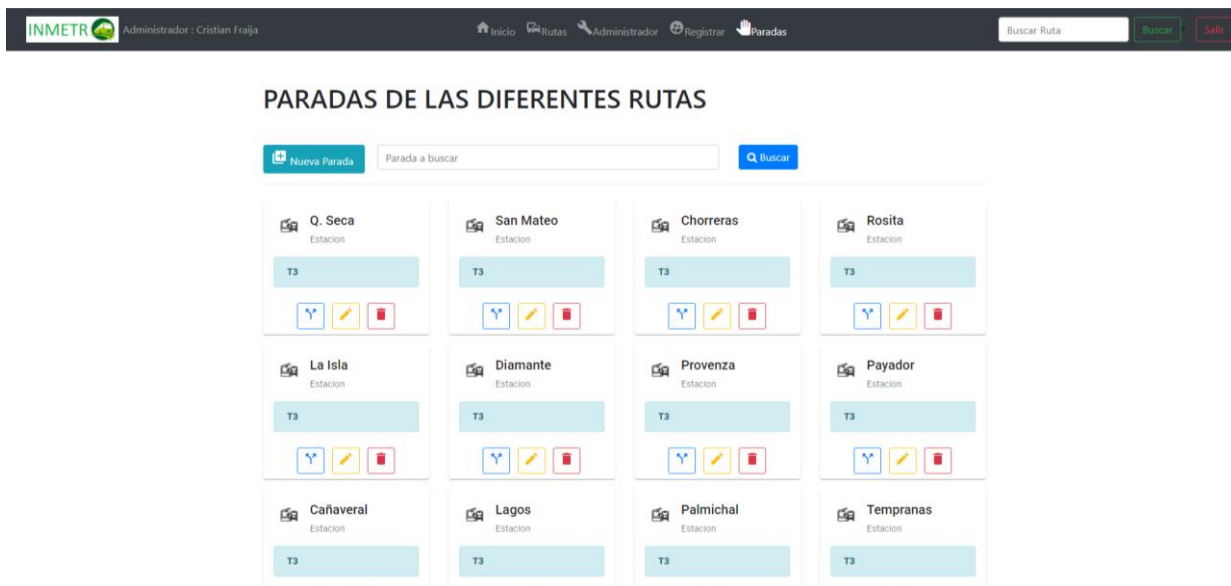


Figura 24. Vista gestionar paradas.

Vista agregar / eliminar ruta a parada. En esta vista el administrador puede asignar / eliminar una o más rutas a una parada seleccionada.

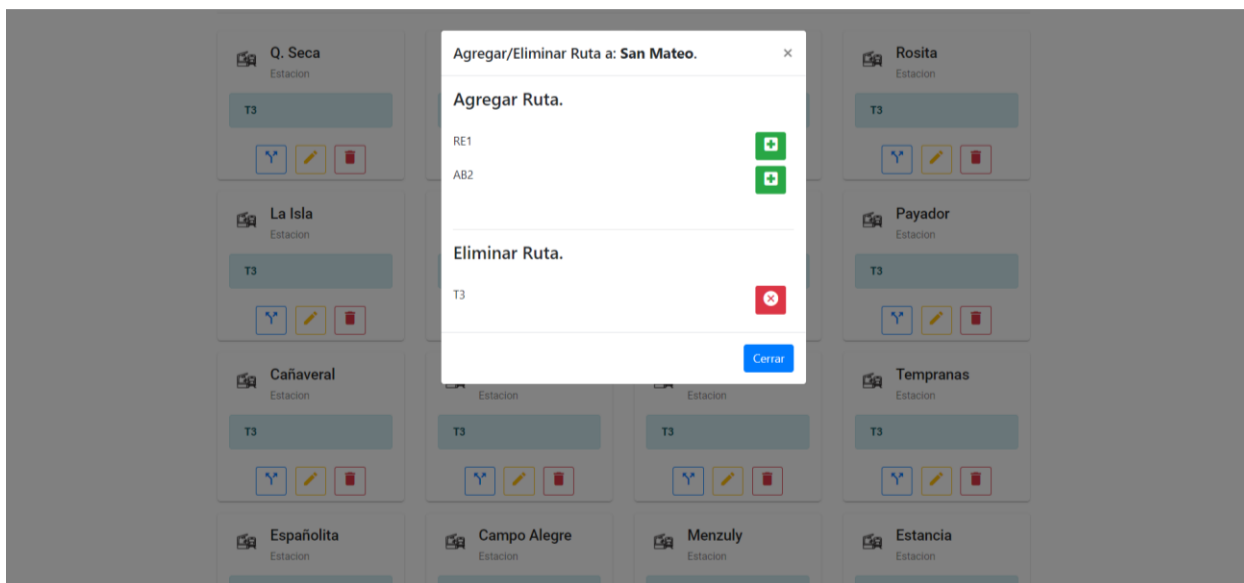


Figura 25. Vista agregar / eliminar ruta a parada.

4.4 Validación

A continuación, es presentada la validación de los módulos principales para los 3 tipos de usuarios del prototipo software.

4.4.1 Validación desde los perfiles del software – Administrador

A continuación, son presentadas las validaciones para los requerimientos impuestos al usuario administrador.

1. Login administrador.

Los usuarios de tipo administrador pueden hacer Login y Logout en el prototipo software, digitando su correo y contraseña. Al hacer esto se muestra que pueden acceder a nuevas funciones de acuerdo con el tipo de usuario.

2. Gestionar buses y conductores.

Se creo un usuario de tipo administrador y con él se inició sesión y se probaron las funcionalidades: registrar, editar e inhabilitar buses y usuarios de tipo conductor, además de asignar rutas y conductores a buses. Todas funcionaron de acuerdo con lo esperado.

3. Gestionar rutas y paradas.

A partir de la indagación que se hizo con funcionarios de Metrolínea, se averiguó en detalle la información de rutas, buses y paradas, y se procedió a registrar la información de 3 rutas. Se inició sesión como usuario de tipo administrador y se ingresaron los datos de la tabla 4¹.

Tabla 4. *Rutas guardadas en base de datos.*

Ruta	Paradas	Bus
1 RE1	Parque Estación UIS, Boulevard Carrera 17, Estación San Mateo, Estación La Isla, Estación De Transferencia Provenza, Estación Payador, Estación Molinos (Occidental), Estación De Transferencia Cañaverall, Estación Lagos, Estación Palmichal, Estación La Españolita, Portal Piedecuesta, San Carlos, San Francisco, Estadio Villa Concha, San Luis, Puente La Normal, Cabecera Del Llano, Eds El Molino, Lomas Del Viento, Eds Barlovento, Entrada Cisnes, Cisnes (Entrada), Carrera 2w Calle 24b, Carrera 2w Calle 23b, Portal Del Valle.	TAV 945
2 AB2	El Porvenir, Estación Provenza, CC Cacique, Gratamira, Megamall, Estadio Alfonso López, Parque Estación UIS, carrera 30 hasta Quebradaseca, HUS.	XWE 103
3 T3	Portal Piedecuesta, Estación La Españolita, Estación Campoalegre, Estación Palmichal, Estación Menzulí, Estación Estancia, Estación Lagos, Estación De Transferencia Cañaverall, Estación Payador, Estación De Transferencia Provenza, Estación Diamante, Estación La Isla, Estación La Rosita, Estación Chorreras, Estación San Mateo, Estación Quebradaseca.	TAV 916

4.4.2 Validación desde los módulos del software - Conductor

A continuación, son presentadas las validaciones para los requerimientos impuestos al usuario conductor.

1. Login conductor.

¹ 1 los datos del usuario administrador se presentarán para la prueba cuando se soliciten.

Los usuarios de tipo conductor pueden hacer Login y Logout en el prototipo software, digitando su correo y contraseña. Al hacer esto se muestra que pueden acceder a nuevas funciones de acuerdo con el tipo de usuario.

2. Gestionar ubicación de buses en mapa.

Se registró un usuario de tipo conductor, se le asignó un bus y una ruta y se pidió a un usuario que ingresara al prototipo software como este conductor, que se subiera a un bus y que fuese reportando la ubicación mediante el prototipo y también vía telefónica para así verificar en el software que efectivamente se estaba mostrando la posición donde él iba. Este paso se realizó en varias ocasiones con diferentes usuarios y rutas, y corresponde a la funcionalidad “Mostrar ubicación de su bus asociado”, la cual al ser ejecutada por el usuario conductor muestra su posición para que los demás usuarios puedan observarla en el mapa correspondiente a su ruta.

Una vez finalizada esta prueba se procede a verificar la funcionalidad “Dejar de mostrar su posición a su bus asociado” la cual anula la funcionalidad anterior. Los resultados obtenidos fueron los esperados.

4.4.3 Validación desde los módulos del software – Invitado

A continuación, son presentadas las validaciones para los requerimientos impuestos al usuario invitado.

1. Búsqueda de ruta.

Para finalizar se realizó la prueba de la funcionalidad principal del software: con las rutas y sus paradas asociadas ya cargadas. Se pidió a 3 usuarios en la “UIS” y 2 usuarios en el “parque los niños” que ingresaran al prototipo software como invitados y se les solicitó que escribieran el lugar a donde se dirigían obteniendo como respuesta indicaciones en el mapa para llegar a su lugar de destino.

4.4.4 Validación de la experiencia de usuario

La usabilidad es la capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones (ISO25000.com, 2020).

Se realizó el siguiente formulario, que consta de 10 preguntas, con la finalidad de determinar la usabilidad del prototipo.

1. ¿Que tan útil encuentra este software a la hora de usar Metrolínea?
2. ¿Utilizaría este software frecuentemente?
3. ¿Es fácil usar el software?
4. ¿El software brinda información pertinente y de manera adecuada?
5. ¿Los tiempos de respuesta al acceder a las diversas funciones son adecuadas?
6. ¿Es intuitivo acceder a la información que se requiere?
7. ¿Es fácil reconocer rápidamente en la interfaz las acciones más relevantes?
8. ¿Encuentra agradable la interfaz gráfica?
9. ¿La información puede ser leída de acuerdo con su tamaño y contraste entre colores?
10. ¿Disfruta o le complace hacer uso de Inmetro?

Este formulario fue entregado a cada uno de los usuarios después de interactuar con el prototipo software, ellos debían responder la prueba con una puntuación de 1 a 5 de acuerdo con la tabla 5.

Tabla 5. *Tabla guía para las respuestas del usuario.*

1	2	3	4	5
Muy mal, muy insatisfecho/a nunca	Mal, poco satisfecho/a, rara vez	Bien, suficiente, adecuado, satisfecho/a, algunas veces	Bastante bien, bastante satisfecho/a, frecuente	Muy bien, muy adecuado, muy satisfecho/a, siempre

Luego de las validaciones por los usuarios, los resultados obtenidos fueron los siguientes:

1. ¿Qué tan útil encuentra este software a la hora de usar Metrolínea?

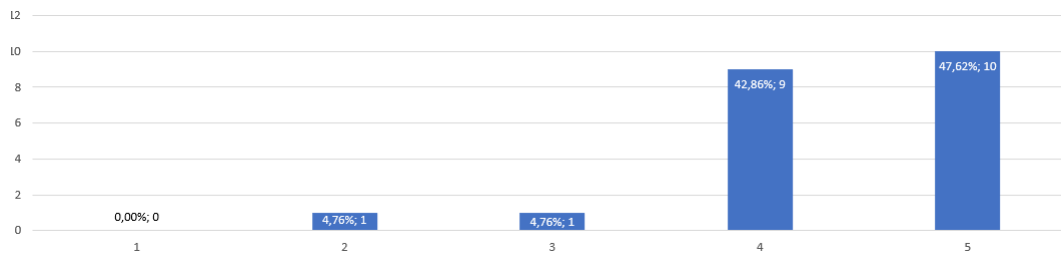


Figura 26. Resultados del primer ítem evaluado

2. ¿Utilizaría este software frecuentemente

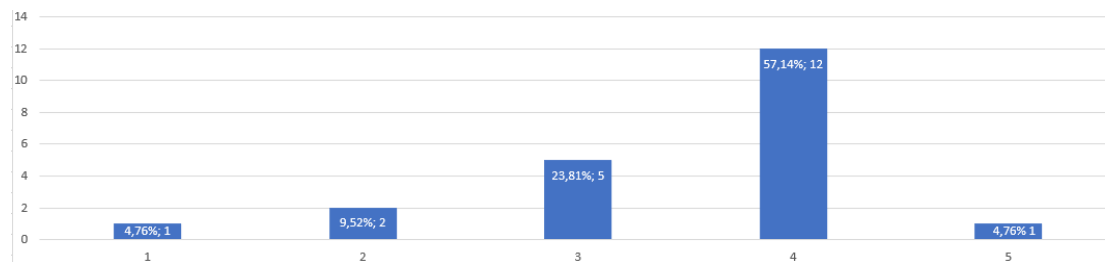


Figura 27. Resultados del segundo ítem evaluado.

3. ¿Es fácil usar el software?

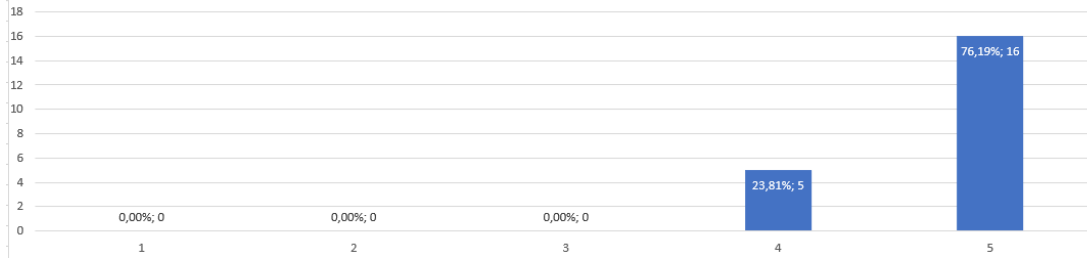


Figura 28. Resultados del tercer ítem evaluado.

4. ¿El software brinda información pertinente y de manera adecuada?

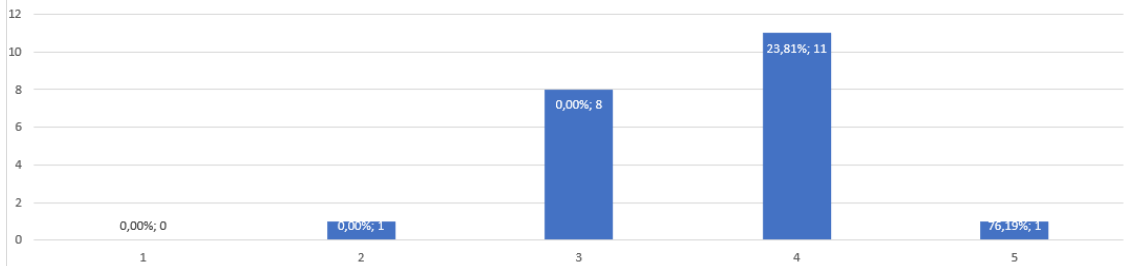


Figura 29. Resultados del cuarto ítem evaluado.

5. ¿Los tiempos de respuesta al acceder a las diversas funciones son adecuadas?

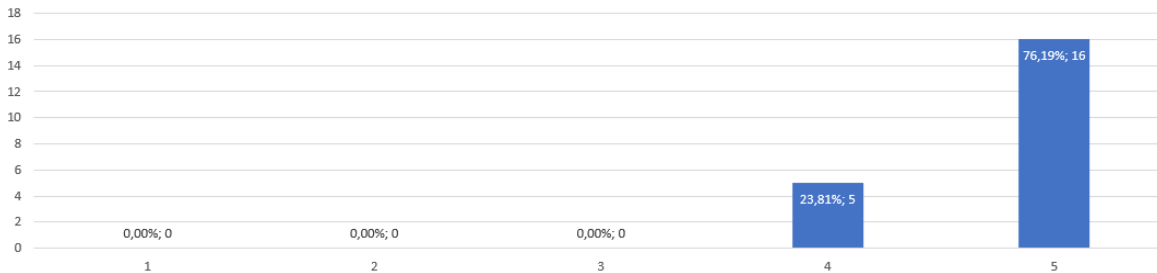


Figura 30. Resultados del quinto ítem evaluado.

6. ¿Es intuitivo acceder a la información que se requiere?

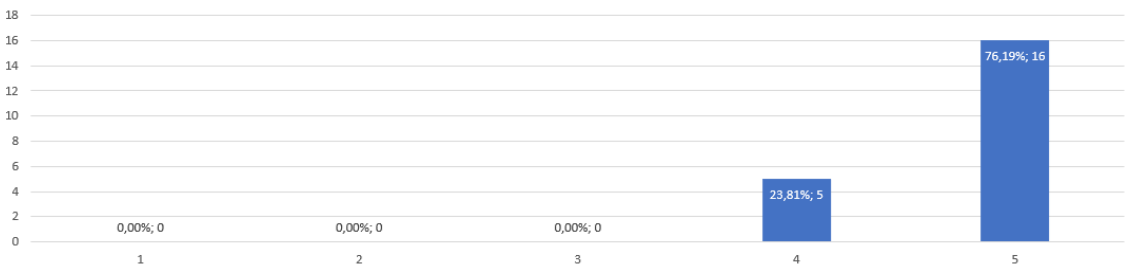


Figura 31. Resultados del sexto ítem evaluado.

7. ¿Es fácil reconocer rápidamente en la interfaz las acciones más relevantes?

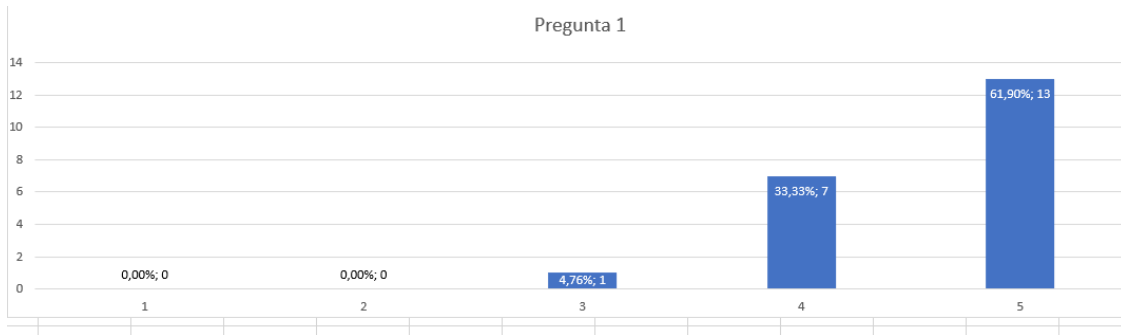


Figura 32. Resultados del séptimo ítem evaluado.

8. ¿Encuentra agradable la interfaz gráfica?

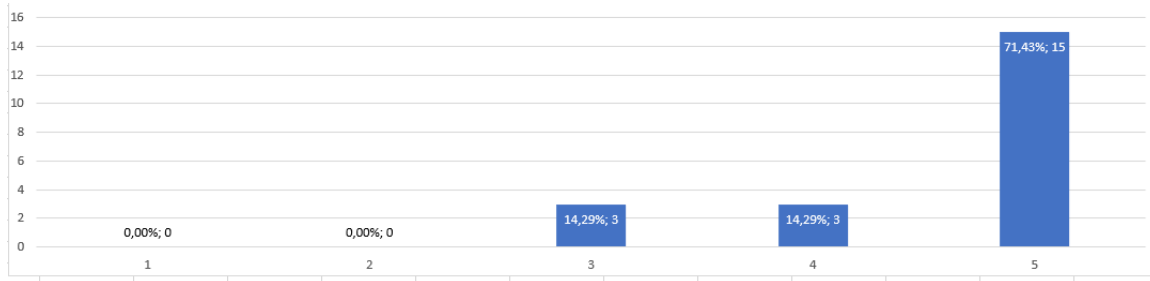


Figura 33. Resultados del octavo ítem evaluado.

9. ¿La información puede ser leída de acuerdo con su tamaño y contraste entre colores?

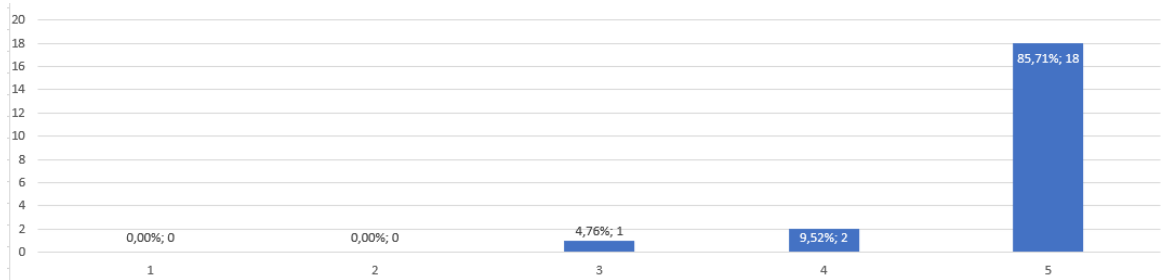


Figura 34. Resultados del noveno ítem evaluado.

10. ¿Disfruta o le complace hacer uso de Inmetro?

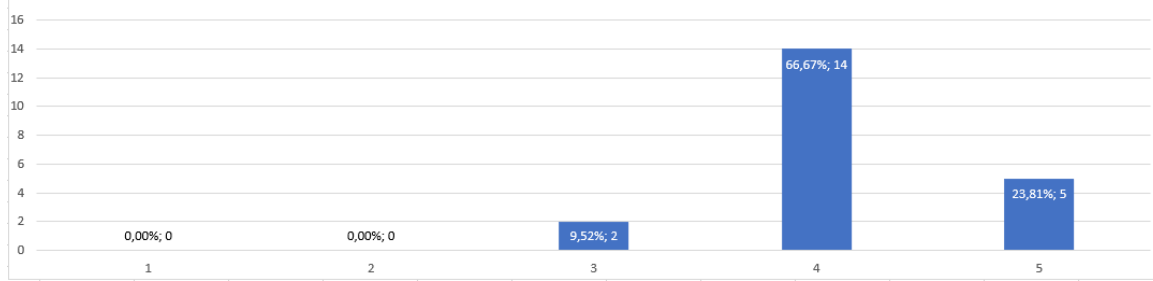


Figura 35. Resultados del décimo ítem evaluado.

4.5 Google Analytics

Es utilizada para monitorear el estado de un sitio web, básicamente se trata de un servicio de análisis web con el que puedes obtener diferentes tipos de información de un sitio, como su número de visitantes o la tasa de conversiones, todos ellos en tiempo real.

Con esta información es posible tomar las mejores decisiones sobre qué aspectos cambiar o mejorar, o qué cosas nuevas implementar para el logro de los objetivos del sitio (Castro, 2020).

En la figura 12, se observa la vista principal de Google Analytics la cual nos brinda información sobre: “usuarios”, que aumenta su cuenta cada vez que un nuevo dispositivo entra en la página web, “sesiones”, en donde una sesión hace referencia al periodo de tiempo en el un usuario interactúa con el sitio web con una inactividad menor a 30 minutos (Tras 30 minutos de inactividad cualquier página visualizada por ese usuario será tratada como una nueva sesión), “porcentaje de rebote”, que es una métrica que mide cuántos usuarios visitan una web y la abandonan directamente sin visitar ninguna página más de la misma, y la “duración promedio de la sesión”

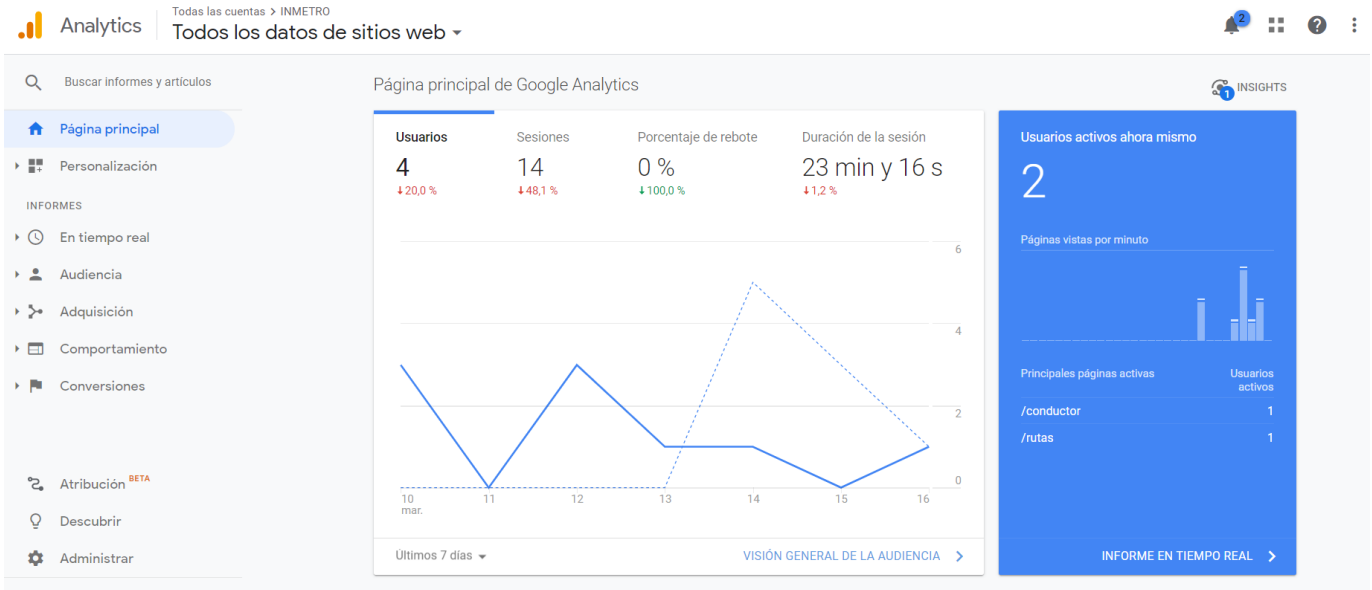


Figura 36. Google Analytics Data.

En la figura 13, observamos el número de usuarios que ingresan día a día y la forma como los usuarios están accediendo al prototipo, en este caso solo hay tráfico directo, que significa que ingresan digitando la URL directamente en el navegador.

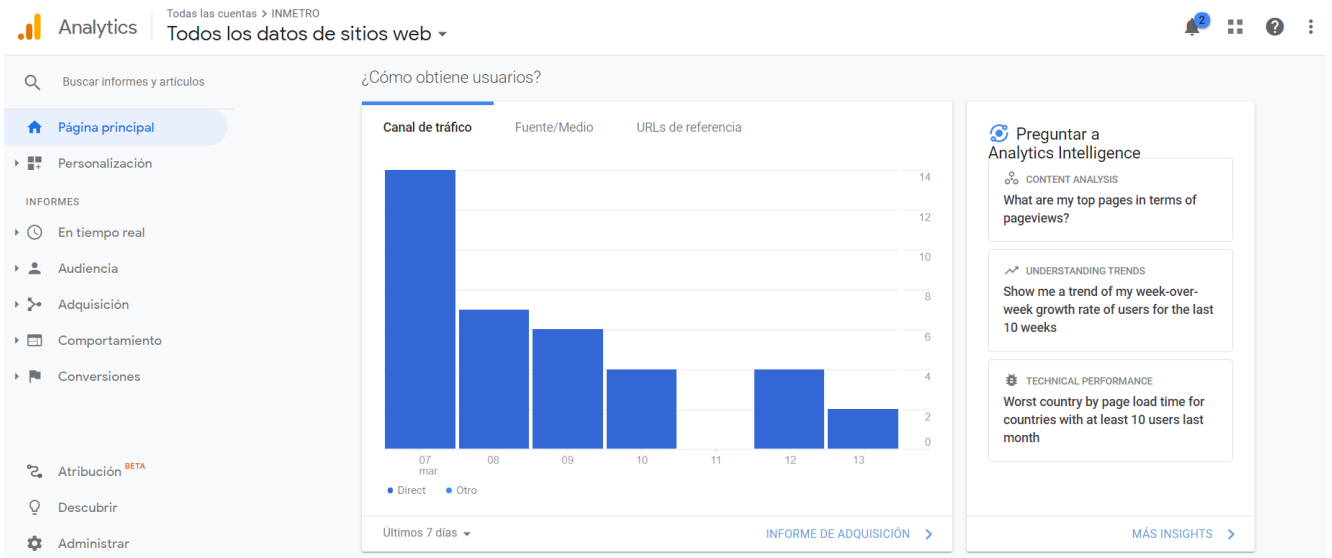


Figura 37. Obtención de usuarios.

Google Analytics también tiene la funcionalidad de recopilar información de los horarios en que los usuarios visitan el sitio web, desde que país se accedió y el tipo de dispositivo que usaron para entrar. Observar figura 14.

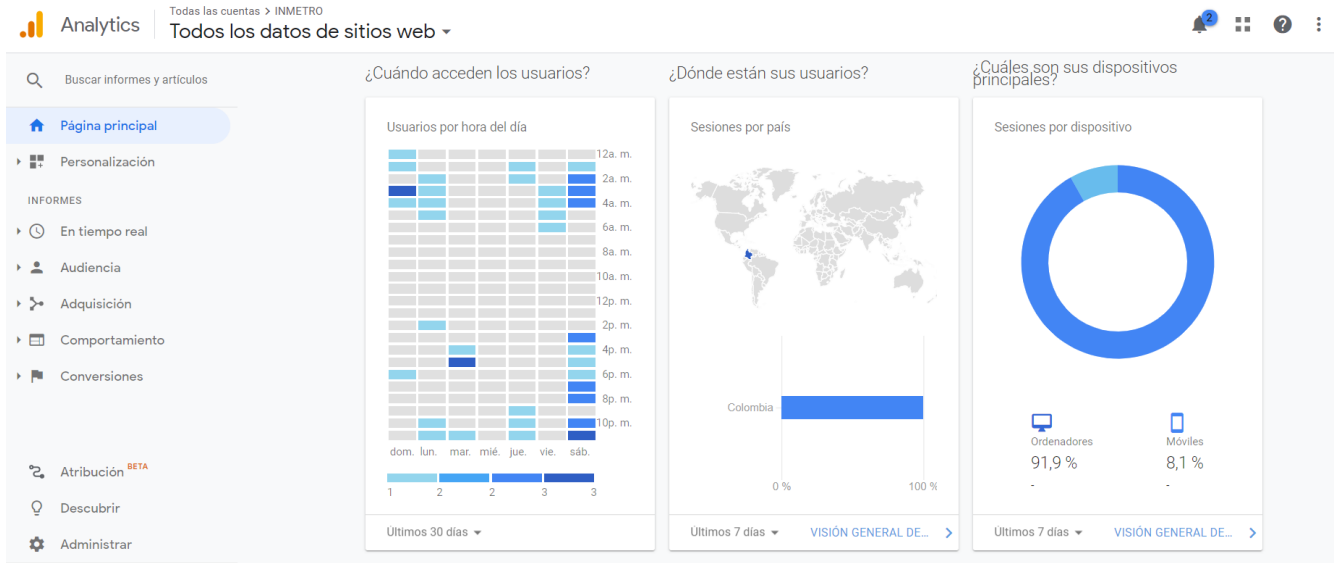


Figura 38. Horario y ubicación de acceso a la página web.

En la figura 15, se observan las vistas o componentes que más visitan los usuarios.

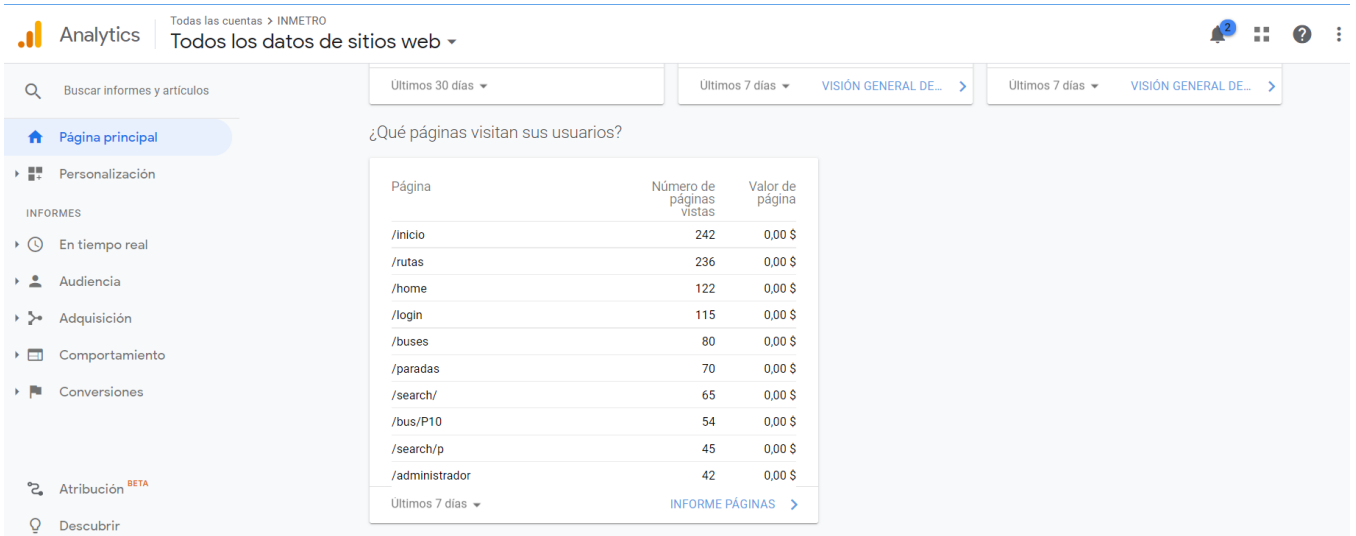


Figura 39. Páginas que visitan los usuarios.

Por otra parte, podemos observar informes más detallados de cada una de las métricas mencionadas anteriormente.

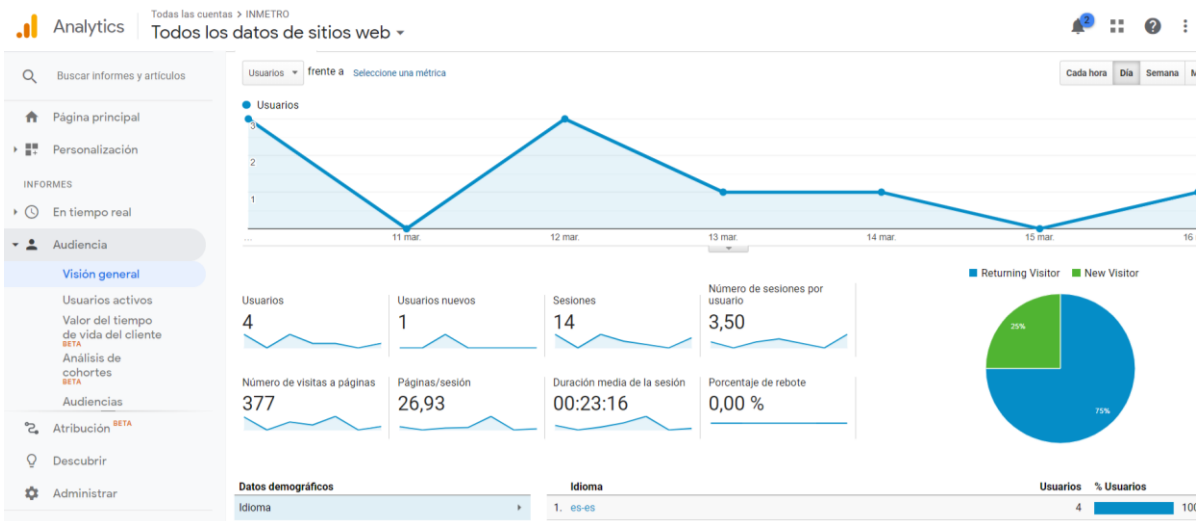


Figura 40. Informe general de audiencia.

Y por medio de la implementación en el código, gracias a los eventos que se pueden programar con Google Analytics, es posible registrar las búsquedas que realizan los usuarios, hacia donde se dirigen con mayor frecuencia.

5 Conclusiones

Este trabajo se convierte en un prototipo software que permite disminuir las falencias en el modelo de transporte actual en Bucaramanga ya que hace un manejo adecuado de la información, al permitirle al usuario ubicarse y ubicar buses geográficamente según la Ruta que elija. Además, permite ingresar un destino y el prototipo calcula una Ruta desde su ubicación actual la cual es trazada con indicaciones en el mapa de Google.

La validación del prototipo permitió corregir detalles de funcionamiento o alcance del prototipo tales como: El solo mostrar las rutas y su trayectoria no era suficiente, la inclusión de la funcionalidad que permite la búsqueda de una Ruta dado un origen y destino era indispensable para el ajuste del software a las necesidades del usuario.

Los diagramas UML ayudaron a la comunicación con el director y facilitó la comprensión del alcance del proyecto con los usuarios que hicieron las pruebas. Cuando dejaba el proyecto, revisar los diferentes diagramas ahorra tiempo a la hora de retomar. Además, se pudo corregir la funcionalidad en la que el usuario conductor compartía su ubicación a los demás usuarios antes de ser programada.

La solución presentada a este problema requirió la aplicación de conceptos aprendidos a lo largo de la carrera, el uso de diferentes frameworks y el estudio de nuevas tecnologías, en las cuales la curva de aprendizaje fue favorable. Generando una experiencia de usuario amigable.

6 Recomendaciones

Incorporar un tiempo estimado de un bus a una parada específica, de tal forma que los usuarios puedan conocer cuánto tiempo tardará en llegar un bus a una parada.

Realizar una nueva versión de la interfaz gráfica que mejore la experiencia de usuario.

Este proyecto sirve de base o guía para nuevos proyectos que van trabajar con la API de Google Maps, Geolocalización y Angular, un ejemplo de estos podría ser la creación de una aplicación para el rastreo de rutas escolares, en las que los padres puedan acceder desde su celular y observar en tiempo real la ubicación de las rutas asociadas a sus hijos y un plus sería acceder a una cámara dentro del bus en la que se pueda observar lo que sucede dentro de este. Otro proyecto podría ser la geolocalización de vehículos para evitar o contraatacar el robo de estos.

Implementar el prototipo para que este sea una aplicación móvil. Gracias a que se trabajó con Angular y con el responsive de Bootstrap este prototipo se podría convertir en una PWA (Progressive Web Application) que permite a una página web ser instalada como una aplicación en un teléfono móvil.

Implementar una vista para el usuario de tipo administrador en la cual él pueda observar todos los buses Activos al mismo tiempo en el mapa.

Referencias Bibliográficas

- Alligator.io. (7 de Mayo de 2017). *Angular Router: Using Route Guards* ← Alligator.io. Obtenido de Guards: <https://cutt.ly/6yxII0f>
- Angular.io. (Febrero de 2020). *Angular*. Obtenido de <https://angular.io/>
- BlogNubeColectiva. (1 de Agosto de 2018). *Que es Materialize CSS y tus Primeros pasos con esta Herramienta | Blog Nube Colectiva*. Obtenido de Materialize: <https://cutt.ly/DyxII2F>
- Bootstrap. (Febrero de 2020). *Bootstrap · La biblioteca HTML, CSS y JS más popular del mundo*. Obtenido de <https://getbootstrap.com/>
- Castro, J. (Febrero de 2020). *Google Analytics ¿Qué es y para qué sirve?* Obtenido de <https://cutt.ly/HyxIGt8>
- Celis, R. (2018). *¿Qué es Heroku y para qué me sirve?* Obtenido de <https://cutt.ly/EyxIHkx>
- Chapaval, N. (2018). *Qué es Frontend y Backend*. Obtenido de <https://cutt.ly/GyxIKFF>
- Chardonneau, R. (2017). *Google Analytics: analice el tráfico para orientar sus acciones de marketing web - Ronan Chardonneau - Google Libros*. Obtenido de Google Analytics: <https://cutt.ly/GyxIZF5>
- Cortés Pineda, Y. L. (2018). *Qué es MEAN: desarrollo full-stack en JavaScript*. Obtenido de Full-Stack en JavaScript: <https://cutt.ly/KyxIXXC>
- EasyTaxi. (Febrero de 2020). *Easy Taxi, una app de Cabify - Aplicaciones en Google Play*. Obtenido de <https://cutt.ly/FyxIbLO>
- ExpressJS. (Febrero de 2020). *Express - Infraestructura de aplicaciones web Node.js*. Obtenido de <https://expressjs.com/es/>

Google Maps. (Febrero de 2020). *Overview / Maps JavaScript API / Google Developers*. Obtenido de Maps JavaScript API: <https://cutt.ly/iyxIY3I>

Google.com. (Febrero de 2020). *APIs y servicios - INMETRO - Google Cloud Platform*. Obtenido de Directions API: <https://console.cloud.google.com/apis/library/directions-backend.googleapis.com?q=dire&id=c6b51d83-d721-458f-a259-fae6b0af35c5&project=proyecto-ing-quimica>

Google.com. (Febrero de 2020). *APIs y servicios - INMETRO - Google Cloud Platform*. Obtenido de API de geolocalización: <https://cutt.ly/GyxIz77>

Guerrero. (17 de Abril de 2019). *Aspectos Basicos que debes Saber de MVC*. Obtenido de <https://cutt.ly/4yxIV2A>

IBM.com. (Febrero de 2020). *Patrón de diseño de modelo-vista-controlador*. Obtenido de <https://cutt.ly/kyxIUGq>

ISO25000.com. (Febrero de 2020). *Usabilidad*. Obtenido de ISO25000: <https://cutt.ly/IyxISCK>

Lucidchart.com. (Febrero de 2020). *Software de diagramación en línea y solución visual*. Obtenido de Lucidchart: <https://www.lucidchart.com/pages/es>

Metrolinea.gov.co. (Febrero de 2020). *Metrolínea | Metrolínea*. Obtenido de Metrolínea: <https://cutt.ly/lyxIEss>

MinTIC. (10 de Enero de 2017). *MinTIC presenta las aplicaciones colombianas más exitosas del 2016 - Ministerio de Tecnologías de la Información y las Comunicaciones*. Obtenido de <https://cutt.ly/qyxIRvj>

MongoDB.com. (Febrero de 2020). *MongoDB Atlas: La mejor forma de utilizar MongoDB en la nube 1 | MongoDB*. Obtenido de <https://cutt.ly/IyxITid>

Munro, J. (9 de Octubre de 2017). *Una introducción a Mongoose para MongoDB y Node.js*.

Obtenido de Mongoose: <https://cutt.ly/ayxINuM>

Neoattack. (2020). *¿Qué es un Framework y para que sirve? - Neo Wiki | NeoAttack*. Obtenido de

¿Qué es un Framework y para que sirve?: <https://cutt.ly/UyxIh5H>

NodeJS. (Febrero de 2020). *Acerca | Node.js*. Obtenido de Node.js: <https://nodejs.org/es/about/>

Noticias ONU. (2018). *Las ciudades seguirán creciendo, sobre todo en los países en desarrollo |*

ONU DAES | Naciones Unidas Departamento de Asuntos Económicos y Sociales.

Obtenido de <https://cutt.ly/gyxIQXB>

OSI.es. (20 de Septiembre de 2016). *Geolocalización: virtudes y riesgos | Oficina de Seguridad*

del Internauta. Obtenido de Geolocalización: <https://cutt.ly/jyxInXC>

Pérez Porto, J., & Gardey, A. (2008). *Definición de sistema de información - Qué es, Significado*

y Concepto. Obtenido de Definición de sistema de información: <https://cutt.ly/VyxI1t5>

RedHat. (2020). *Redhat.com*. Obtenido de <https://cutt.ly/eyxIONU>

Redondo, F. (2017). *Postman: gestiona y construye tus APIs rápidamente - Paradigma*. Obtenido

de Postman: <https://cutt.ly/3yxIOHM>

Rubenfa. (3 de Febrero de 2014). *MongoDB. Qué es, cómo funciona y cuándo podemos usarlo (o*

no). Obtenido de MongoDB: <https://cutt.ly/YyxI967>

S. D. (2017). *Introducción a JSON Web Tokens (JWT)*. Obtenido de <https://cutt.ly/KyxIDEW>

sistemas.com. (Febrero de 2020). *Definición de Casos de uso - Significado y definición de Casos*

de uso. Obtenido de <https://cutt.ly/lyxIvp7>

Uber. (Febrero de 2020). *Uber - Aplicaciones en Google Play*. Obtenido de <https://cutt.ly/VyxIA3t>

VisualStudioCode. (Febrero de 2020). *Documentation for Visual Studio Code*. Obtenido de

<https://code.visualstudio.com/docs>

- Vicente, D. (9 de Marzo de 2017). Encriptación de password en NodeJS y MongoDB: bcrypt. Obtenido de <https://cutt.ly/wyxIFzL>
- Wikipedia.com. (Febrero de 2020). *Moovit* - *Wikipedia, la enciclopedia libre*. Obtenido de <https://cutt.ly/HyxIT2K>
- Wikipedia.com. (Febrero de 2020). *Tappsi* - *Wikipedia, la enciclopedia libre*. Obtenido de <https://cutt.ly/lyxIAhY>
- Wikipedia.org. (21 de Enero de 2020). *Angular (framework)* - *Wikipedia, la enciclopedia libre*. Obtenido de <https://cutt.ly/8yxIk2x>
- Wikipedia.org. (Febrero de 2020). *Autobús de tránsito rápido* - *Wikipedia, la enciclopedia libre*. Obtenido de Autobús de tránsito rápido: <https://cutt.ly/dyxIcwR>
- Wikipedia.org. (Febrero de 2020). *Diagrama de clases* - *Wikipedia, la enciclopedia libre*. Obtenido de Diagrama de clases: <https://cutt.ly/eyxIv20>
- Wikipedia.org. (Febrero de 2020). *GitHub* - *Wikipedia, la enciclopedia libre*. Obtenido de <https://cutt.ly/WyxImPn>

Apéndices

Apéndices A. Requerimientos software.

A continuación, se presentan las tablas con las respectivas especificaciones de los requerimientos funcionales:

1. Requerimientos del software.

Tabla 6. *Registrar usuario*

Requerimiento funcional	
Identificador (RF#)	RF01
Nombre	Registrar usuario
Prioridad	Alta
Descripción	Los usuarios administradores podrán registrar usuarios de tipo administrador y de tipo conductor registrando los siguientes datos: Nombre, Correo electrónico, Contraseña y escoger tipo de usuario.

Tabla 7. *Gestionar usuario.*

Requerimiento funcional	
Identificador (RF#)	RF02
Nombre	Gestionar usuario
Prioridad	Alta
Descripción	Los usuarios administradores podrán editar e inhabilitar usuarios de tipo conductor, para editarlos se requerirán los siguientes datos: Nombre, Correo electrónico y Contraseña.

Tabla 8. *Login usuario.*

Requerimiento funcional	
Identificador (RF#)	RF03
Nombre	Login usuario
Prioridad	Alta

Descripción	Permite a los usuarios entrar a su cuenta. Dependiendo del tipo de usuario logueado se activarán funcionalidades asociadas a este.
--------------------	--

Tabla 9. *Logout usuario.*

Requerimiento funcional	
Identificador (RF#)	RF04
Nombre	Logout usuario
Prioridad	Alta
Descripción	Permite a los usuarios salir de su cuenta. Dependiendo del tipo de usuario logueado se desactivarán funcionalidades asociadas a este. El usuario activo al hacer Logout será Invitado.

Tabla 10. *Crear rutas.*

Requerimiento funcional	
Identificador (RF#)	RF05
Nombre	Crear rutas
Prioridad	Alta
Descripción	Los usuarios administradores podrán crear rutas registrando los siguientes datos: Nombre, Descripción, Fecha y Seleccionar Imagen bus. La Imagen ruta se relacionará de acuerdo con el Nombre.

Tabla 11. *Gestionar rutas.*

Requerimiento funcional	
Identificador (RF#)	RF06
Nombre	Gestionar rutas
Prioridad	Alta
Descripción	Los usuarios administradores podrán editar e inhabilitar rutas, para editar una ruta se requerirán los siguientes datos: Nombre, Descripción, Fecha y Seleccionar Imagen bus.

Tabla 12. *Crear buses.*

Requerimiento funcional	
Identificador (RF#)	RF07
Nombre	Crear buses
Prioridad	Alta
Descripción	Los usuarios administradores podrán crear buses registrando los siguientes datos: Placa y seleccionando el Tipo bus.

Tabla 13. *Gestionar buses.*

Requerimiento funcional	
Identificador (RF#)	RF08
Nombre	Gestionar buses
Prioridad	Alta
Descripción	Los usuarios administradores podrán editar e inhabilitar buses, para editar un bus se requerirán los siguientes datos: Placa y seleccionando el Tipo bus.

Tabla 14. *Asignar buses a ruta.*

Requerimiento funcional	
Identificador (RF#)	RF09
Nombre	Asignar buses a ruta
Prioridad	Alta
Descripción	Los usuarios administradores podrán Asignar una ruta específica a un bus seleccionado.

Tabla 15. *Gestionar asignaciones de rutas.*

Requerimiento funcional	
Identificador (RF#)	RF10
Nombre	Gestionar asignaciones de rutas
Prioridad	Alta
Descripción	Los usuarios administradores podrán editar o eliminar una asignación de una ruta a un bus. Para editarla se requerirá el siguiente dato: Nueva ruta.

Tabla 16. *Enviar la ubicación mediante GPS.*

Requerimiento funcional	
Identificador (RF#)	RF11
Nombre	Enviar la ubicación mediante GPS
Prioridad	Alta
Descripción	Los usuarios conductores, en caso de estar asociados a un bus, podrán compartir o dejar de compartir su ubicación en tiempo real con este. Posición la cual tendrán acceso los demás tipos de usuario.

Tabla 17. *Visualizar ruta seleccionada en mapa (Google Maps).*

Requerimiento funcional	
Identificador (RF#)	RF12

Nombre	Visualizar ruta seleccionada en mapa (Google Maps)
Prioridad	Alta
Descripción	Todos los usuarios, al seleccionar una ruta, los buses asociados a dicha ruta y que, a su vez, el conductor asociado este compartiendo la ubicación, se podrán visualizar en el mapa Google Maps. La posición se actualizará cada 3 segundos.

Tabla 18. *Filtrar búsqueda de ruta.*

Requerimiento funcional	
Identificador (RF#)	RF13
Nombre	Filtrar búsqueda de ruta
Prioridad	Baja
Descripción	Todos los usuarios, Podrán filtrar mediante un texto las rutas.

Tabla 19. *Observar la vista principal.*

Requerimiento funcional	
Identificador (RF#)	RF14
Nombre	Observar la vista principal
Prioridad	Alta
Descripción	Todos los usuarios, Podrán acceder a la vista principal.

Tabla 20. *Observar rutas.*

Requerimiento funcional	
Identificador (RF#)	RF15
Nombre	Observar rutas
Prioridad	Alta
Descripción	Todos los usuarios, Podrán acceder a la vista rutas.

Tabla 21. *Acceder a una ruta.*

Requerimiento funcional	
Identificador (RF#)	RF16
Nombre	Acceder a una ruta
Prioridad	Alta
Descripción	Todos los usuarios, Podrán acceder a una ruta específica y podrán ver sus detalles, tales como: Nombre, Descripción, Fecha e imagen de Tipo bus, el número de buses asociados a esta ruta, los buses asociados a la ruta y los buses asociados a la ruta en el mapa.

Tabla 22. Filtrar buses por ruta.

Requerimiento funcional	
Identificador (RF#)	RF17
Nombre	Filtrar buses por ruta
Prioridad	Baja
Descripción	Todos los usuarios, Podrán filtrar buses por ruta al acceder a una ruta especifica se mostrarán solo los buses asociado a la ruta.

Tabla 23. *Asignar conductor a un bus.*

Requerimiento funcional	
Identificador (RF#)	RF18
Nombre	Asignar conductor a un bus
Prioridad	Alta
Descripción	Los usuarios administradores podrán Asignar un conductor específico a un bus seleccionado.

Tabla 24. *Gestionar asignaciones de conductores.*

Requerimiento funcional	
Identificador (RF#)	RF19
Nombre	Gestionar asignaciones de conductores
Prioridad	Alta
Descripción	Los usuarios administradores podrán editar o eliminar una asignación de un conductor a un bus. Para editarla se requerirá el siguiente dato: Nuevo conductor.

Tabla 25. *Diseñar una interfaz gráfica de usuario.*

Requerimiento no funcional	
Identificador (RNF#)	RNF01
Nombre	Diseñar una interfaz gráfica de usuario.
Prioridad	Alta
Características	Se debe de diseñar una interfaz gráfica que sea agradable y una sencilla usabilidad, para así permitir a los usuarios interactuar con la aplicación.

Tabla 26. *Operar con varios usuarios al tiempo.*

Requerimiento no funcional	
-----------------------------------	--

Identificador (RNF#)	RNF02
Nombre	Operar con varios usuarios al tiempo.
Prioridad	Alta
Características	La posibilidad de uso de la aplicación por varios usuarios al tiempo hace que surja esta necesidad. El prototipo debe funcionar con varios usuarios con los diferentes roles al tiempo.

Tabla 27. Operar con varios buses y rutas al tiempo.

Requerimiento no funcional	
Identificador (RNF#)	RNF03
Nombre	Operar con varios buses y rutas al tiempo.
Prioridad	Alta
Características	El prototipo de operar con varios buses y rutas las cuales serán accedidas en cualquier momento por cualquier número de usuarios.

Tabla 28. Actualizar información en la base de datos.

Requerimiento no funcional	
Identificador (RNF#)	RNF04
Nombre	Actualizar información en la base de datos.
Prioridad	Alta
Características	La información modificada por parte de los usuarios administrador debe ser actualizada en base de datos y así mismo en el servidor.

Tabla 29. Actualizar información en tiempo real, la posición de los buses en mapa y base de datos.

Requerimiento no funcional	
Identificador (RNF#)	RNF05
Nombre	Actualizar información en tiempo real, la posición de los buses en mapa y base de datos.
Prioridad	Alta
Características	La ubicación compartida por parte de los usuarios de tipo conductor debe ser actualizada en base de datos y servidor en tiempo real.

Tabla 30. Compatibilidad del prototipo.

Requerimiento no funcional	
Identificador (RNF#)	RNF05
Nombre	Compatibilidad del prototipo.
Prioridad	Media

Características El prototipo debe ser compatible en los diferentes navegadores como: Google Chrome, Firefox e Internet Explorer. Para así abarcar la mayor cantidad de usuarios.

Apéndices B. Casos de uso.

A continuación, se representan los casos de uso debidamente detallados:

Tabla 31. *Login usuario.*

Caso de uso: Login usuario	
Actor Principal	Usuario administrador y conductor
Descripción	Iniciar sesión con un usuario existente en el prototipo.
	Paso Acción
	1 El usuario ingresa al prototipo
	2 El usuario hace clic en el botón “Iniciar sesión”
	3 El usuario digita su correo y contraseña
	4 El usuario hace clic en “Ingresar”
	5 El prototipo valida la información ingresada
	6 El prototipo muestra la página principal para administrador o para conductor.
Excepción 1.	Paso Acción
	5 Los datos ingresados no corresponden a ningún usuario registrado
	5 El prototipo muestra un mensaje, notificando que los datos ingresados no son correctos

Tabla 32. *Logout usuario.*

Caso de uso: Logout usuario	
Actor Principal	Usuario administrador y conductor
Descripción	Cerrar sesión con un usuario anteriormente logueado.
	Paso Acción
	1 Hacer clic en el botón “Salir”
	2 El prototipo cierra sesión del usuario activo, eliminando datos en el localStorage.
	3 El prototipo muestra la página principal

Tabla 33. *Acceder a rutas.*

Caso de uso: Acceder a rutas	
Actor Principal	Usuario Invitado, administrador y conductor.
Descripción	Ver todas las rutas.
Paso	Acción
1	El usuario ingresa al prototipo
2	El usuario hace clic en el botón “rutas”
3	El prototipo carga la vista rutas
4	El prototipo carga todas las rutas de la base de datos
5	El prototipo muestra cada una de las rutas resumidas en pequeñas cartas.

Tabla 34. *Ver una ruta específica.*

Caso de uso: Ver una ruta específica	
Actor Principal	Usuario Invitado, administrador y conductor
Descripción	Acceder a una ruta para ver su información y buses asociados.
Paso	Acción
1	El usuario ingresa al prototipo
2	El usuario hace clic en el botón “rutas”
3	El prototipo carga la vista rutas
4	El prototipo carga todas las rutas de la base de datos
5	El prototipo muestra cada una de las rutas resumidas en pequeñas cartas.
6	El usuario hace clic en el botón “Ver más...” de la carta de la ruta que quiere acceder
7	El prototipo carga la vista de ruta específica
8	El prototipo carga la información de la ruta seleccionada

Tabla 35. *Ubicación de buses en mapa.*

Caso de uso: Ubicación de buses en mapa	
Actor Principal	Usuario Invitado, administrador y conductor
Descripción	Ver buses asociados a una ruta en el Google Maps.
Paso	Acción
1	El usuario ingresa al prototipo
2	El usuario hace clic en el botón “Rutas”
3	El prototipo carga la vista rutas
4	El prototipo carga todas las rutas de la base de datos
5	El prototipo muestra cada una de las rutas resumidas en pequeñas cartas.
6	El usuario hace clic en el botón “Ver más...” de la carta de la ruta que quiere acceder
7	El prototipo carga la vista de ruta específica

8	El prototipo carga la información de la ruta seleccionada
9	El usuario se dirige a la parte inferior de la vista ruta específica
10	El usuario observa la posición de los buses asociados a la ruta seleccionada en el mapa
11	El usuario hace clic en el botón “Ubicar en mapa” de un bus
12	El prototipo ubica la posición del bus seleccionado como centro del mapa

Tabla 36. *Filtrar rutas desde el navbar.*

Caso de uso: Filtrar rutas desde el navbar	
Actor Principal	Usuario Invitado, administrador y conductor
Descripción	Filtrar rutas mediante un texto.
	Paso Acción
	1 El usuario ingresa al prototipo
	2 El usuario hace clic en el cuadro de texto “Buscar ruta”
	3 El usuario escribe el nombre o parte del nombre de la ruta a buscar
	4 El usuario hace clic en el botón “Buscar”
	5 El prototipo carga la vista rutas
	6 El prototipo carga todas las rutas relacionadas al texto escrito en el cuadro de texto “Buscar ruta”
	7 El prototipo muestra cada una de las rutas cargadas
Excepción 1.	Paso Acción
	5 El prototipo no encuentra ninguna ruta relacionada con el texto en el cuadro de texto “Buscar ruta”
	5 El prototipo muestra un mensaje informando que no existen resultados con el texto ingresado

Tabla 37. *Compartir ubicación GPS a su bus asociado.*

Caso de uso: Compartir ubicación GPS a su bus asociado	
Actor Principal	Usuario conductor
Descripción	Cambia en base de datos la posición del bus asociado al conductor, permitiendo a los demás usuarios ver la posición de este bus en el mapa
	Paso Acción
	1 El usuario debe hacer Login con su usuario de tipo conductor
	2 El prototipo habilita la vista Conductor para este usuario
	3 El conductor hace clic en el botón “Conductor”
	4 El prototipo carga la vista Conductor, junto con la información de su bus asociado

	5	El conductor hace clic en el botón “Mostrar Posición”
	6	El prototipo accede a la ubicación GPS del usuario conductor.
	7	El prototipo muestra en cuadros de texto la latitud, longitud y sentido en cuadros de texto.
	8	El prototipo actualiza en base de datos la latitud, longitud y sentido del bus asociado al conductor con los datos tomados de su GPS.
	9	El prototipo se mantiene escuchando los cambios de posición y actualiza estos datos en base de datos cada 3 segundos
Excepción 1.	Paso	Acción
	4	El conductor no tiene bus asociado
	5	El conductor no puede compartir su ubicación
Excepción 2.	Paso	Acción
	6	El GPS se encuentra desactivado
	6	El prototipo no puede acceder a la ubicación
	7	El prototipo informa que el GPS esta desactivado

Tabla 38. *Dejar de Compartir ubicación a su bus asociado.*

Caso de uso: Dejar de Compartir ubicación a su bus asociado		
Actor Principal	Usuario conductor	
Descripción	Paso	Acción
	1	El usuario debe hacer Login con su usuario de tipo conductor
	2	El conductor se encuentra compartiendo su ubicación a su bus asociado
	3	El conductor hace clic en el botón “No Mostrar Posición”
	4	El prototipo actualiza en base de datos la latitud, longitud y sentido del bus asociado al conductor con los 0 en todos los campos.
	5	El prototipo deja de escuchar los cambios de posición y de actualizarlos en base de datos.
Excepción 1.	Paso	Acción
	2	El conductor no se encontraba compartiendo la ubicación
	2	El prototipo no habilita el botón “No mostrar posición”

Tabla 39. *Acceder a vista administrador.*

Caso de uso: Acceder a vista administrador		
Actor Principal	Usuario administrador	
Descripción	Loguearse como administrador y acceder a la vista administrador	
	Paso	Acción

	1	El usuario debe hacer Login con su usuario de tipo administrador
	2	El prototipo habilita la vista Administrador para este usuario
	3	El administrador hace clic en el botón “Administrador”
	4	El prototipo carga la vista Administrador, junto con todas las rutas, los buses y los usuarios de tipo conductor
Excepción 1.	Paso	Acción
	1	Los datos ingresados no corresponden a un usuario de tipo administrador
	2	El prototipo no habilita la vista Administrador

Tabla 40. *Acceder a vista Registrar.*

Caso de uso: Acceder a vista Registrar		
Actor Principal	Usuario administrador	
Descripción	Loguearse como administrador y acceder a la vista Registrar	
	Paso	Acción
	1	El usuario debe hacer Login con su usuario de tipo administrador
	2	El prototipo habilita la vista Registrar para este usuario
	3	El administrador hace clic en el botón “Registrar”
	4	El prototipo carga la vista Registrar
Excepción 1.	Paso	Acción
	1	Los datos ingresados no corresponden a un usuario de tipo administrador
	2	El prototipo no habilita la vista Registrar

Tabla 41. *Crear ruta.*

Caso de uso: Crear ruta		
Actor Principal	Usuario administrador	
Descripción	Crea una ruta y la guarda en base de datos	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el botón “Nueva ruta”
	3	El prototipo carga la vista Crear ruta
	4	El administrador llena los datos requeridos
	5	El prototipo valida que todos los campos sean llenados
	6	El administrador hace clic en “Guardar”
	7	El prototipo guarda la nueva ruta en base de datos
	8	El prototipo muestra la vista Administrador con la nueva ruta
Excepción 1.	Paso	Acción
	6	El administrador hace clic en “Cerrar”

Excepción 2.	Paso	Acción
	6	El administrador hace clic en “Limpiar”
Excepción 2.	Paso	Acción
	1	Caso de uso: Acceder a vista administrador terminó en una excepción

Tabla 42. *Crear bus.*

Caso de uso: Crear bus		
Actor Principal	Usuario administrador	
Descripción	Crea un bus y lo guarde en base de datos	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el botón “Nuevo bus”
	3	El prototipo carga la vista Crear bus
	4	El administrador llena los datos requeridos
	5	El prototipo valida que todos los campos sean llenados
	6	El administrador hace clic en “Guardar”
	7	El prototipo guarda el nuevo bus en base de datos
	8	El prototipo muestra la vista Administrador con el nuevo bus
Excepción 1.	Paso	Acción
	6	El administrador hace clic en “Cerrar”
Excepción 2.	Paso	Acción
	6	El administrador hace clic en “Limpiar”
Excepción 2.	Paso	Acción
	1	Caso de uso: Acceder a vista administrador terminó en una excepción

Tabla 43. *Registrar usuario.*

Caso de uso: Registrar usuario		
Actor Principal	Usuario administrador	
Descripción	Crea un usuario y lo guardo en base de datos	
	Paso	Acción
	1	Caso de uso: Acceder a vista Registrar
	2	El administrador llena los datos requeridos
	3	El prototipo valida que todos los campos sean llenados y que sean validos
	4	El administrador hace clic en “Registrar”
	5	El prototipo guarda el nuevo usuario en base de datos
	6	El prototipo muestra un mensaje de confirmación de creación del nuevo usuario
Excepción 1.	Paso	Acción
	4	El administrador hace clic en otra pestaña

Excepción 2.	Paso	Acción
	3	El prototipo valida que algún campo no fue llenado y o no es válido.

Tabla 44. *Editar ruta.*

Caso de uso: Editar ruta		
Actor Principal	Usuario administrador	
Descripción	Edita una ruta y la guarda en base de datos	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el botón “Editar ruta”
	3	El prototipo carga la vista Editar ruta con los datos de la ruta a editar
	4	El administrador cambia los datos pertinentes
	5	El prototipo valida que todos los campos sean llenados
	6	El administrador hace clic en “Guardar”
	7	El prototipo edita la ruta en base de datos
	8	El prototipo muestra la vista Administrador con la ruta editada
Excepción 1.	Paso	Acción
	6	El administrador hace clic en “Cerrar”
Excepción 2.	Paso	Acción
	6	El administrador hace clic en “Limpiar”
Excepción 2.	Paso	Acción
	1	Caso de uso: Acceder a vista administrador terminó en una excepción

Tabla 45. *Editar bus.*

Caso de uso: Editar bus		
Actor Principal	Usuario administrador	
Descripción	Edita un bus y lo guarda en base de datos	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el botón “Editar bus”
	3	El prototipo carga la vista Editar bus con los datos del bus a editar
	4	El administrador cambia los datos pertinentes
	5	El prototipo valida que todos los campos sean llenados
	6	El administrador hace clic en “Guardar”
	7	El prototipo edita el bus en base de datos
	8	El prototipo muestra la vista Administrador con el bus editado
Excepción 1.	Paso	Acción

	6	El administrador hace clic en “Cerrar”
Excepción 2.	Paso	Acción
	6	El administrador hace clic en “Limpiar”
Excepción 2.	Paso	Acción
	1	Caso de uso: Acceder a vista administrador terminó en una excepción

Tabla 46. *Editar conductor.*

Caso de uso: Editar conductor		
Actor Principal	Usuario administrador	
Descripción	Edita un conductor y lo guarda en base de datos	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el botón “Editar conductor”
	3	El prototipo carga la vista Editar conductor con los datos del conductor a editar
	4	El administrador cambia los datos pertinentes
	5	El prototipo valida que todos los campos sean llenados
	6	El administrador hace clic en “Guardar”
	7	El prototipo edita el conductor en base de datos
	8	El prototipo muestra la vista Administrador con el conductor editado
Excepción 1.	Paso	Acción
	6	El administrador hace clic en “Cerrar”
Excepción 2.	Paso	Acción
	6	El administrador hace clic en “Limpiar”
Excepción 2.	Paso	Acción
	1	Caso de uso: Acceder a vista administrador terminó en una excepción

Tabla 47. *Eliminar ruta.*

Caso de uso: Eliminar ruta		
Actor Principal	Usuario administrador	
Descripción	Elimina una ruta de la base de datos	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el botón “Eliminar ruta”
	3	El prototipo carga la vista Eliminar ruta con el nombre de la ruta a eliminar
	4	El administrador hace clic en el botón “Eliminar”
	5	El prototipo elimina la ruta en base de datos
	6	El prototipo muestra la vista Administrador sin la ruta eliminada

Excepción 1.	Paso	Acción
	4	La ruta tiene uno o más buses asociados
	4	El prototipo detecta que tiene buses asociados y no permite eliminarla
	4	El prototipo informa que se debe desvincular primero la ruta de sus buses y luego ser eliminada
Excepción 2.	Paso	Acción
	3	El administrador hace clic en “Cerrar”

Tabla 48. *Eliminar bus.*

Caso de uso: Eliminar bus		
Actor Principal	Usuario administrador	
Descripción	Elimina un bus de la base de datos	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el botón “Eliminar bus”
	3	El prototipo carga la vista Eliminar bus con la placa del bus a eliminar
	4	El administrador hace clic en el botón “Eliminar”
	5	El prototipo elimina el bus en base de datos
	6	El prototipo muestra la vista Administrador sin el bus eliminado
Excepción 1.	Paso	Acción
	3	El administrador hace clic en “Cerrar”

Tabla 49. *Eliminar conductor.*

Caso de uso: Eliminar conductor		
Actor Principal	Usuario administrador	
Descripción	Elimina un conductor de la base de datos	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el botón “Eliminar conductor”
	3	El prototipo carga la vista Eliminar conductor con el nombre del conductor a eliminar
	4	El administrador hace clic en el botón “Eliminar”
	5	El prototipo elimina el conductor en base de datos
	6	El prototipo muestra la vista Administrador sin el conductor eliminado
Excepción 1.	Paso	Acción
	4	El conductor tiene un bus asociado
	4	El prototipo detecta que el conductor tiene un bus asociado y no permite eliminarlo

	4	El prototipo informa que se debe desvincular primero el conductor de su bus y luego ser eliminado
Excepción 2.	Paso	Acción
	3	El administrador hace clic en “Cerrar”

Tabla 50. *Filtrar rutas.*

Caso de uso: Filtrar rutas		
Actor Principal	Usuario administrador	
Descripción	Filtrar rutas mediante un texto	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el cuadro de texto “Buscar ruta”
	3	El administrador escribe el nombre o parte del nombre de la ruta a buscar
	4	El administrador hace clic en el botón “Buscar”
	5	El prototipo carga todas las rutas relacionadas al texto escrito en el cuadro de texto “Buscar ruta” en la vista administrador
Excepción 1.	Paso	Acción
	5	El prototipo no encuentra ninguna ruta relacionada con el texto en el cuadro de texto “Buscar ruta”
	5	El prototipo muestra un mensaje informando que no existen resultados con el texto ingresado

Tabla 51. *Filtrar buses.*

Caso de uso: Filtrar buses		
Actor Principal	Usuario administrador	
Descripción	Filtrar buses mediante un texto	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el cuadro de texto “Buscar bus”
	3	El administrador escribe el nombre o parte del nombre del bus a buscar
	4	El administrador hace clic en el botón “Buscar”
	5	El prototipo carga todos los buses relacionados al texto escrito en el cuadro de texto “Buscar bus” en la vista administrador
Excepción 1.	Paso	Acción
	5	El prototipo no encuentra ningún bus relacionado con el texto en el cuadro de texto “Buscar bus”

5	El prototipo muestra un mensaje informando que no existen resultados con el texto ingresado
---	---

Tabla 52. *Filtrar conductores.*

Caso de uso: Filtrar conductores	
Actor Principal	Usuario administrador
Descripción	Filtrar conductores mediante un texto
	Paso Acción
	1 Caso de uso: Acceder a vista administrador
	2 El administrador hace clic en el cuadro de texto “Buscar conductor”
	3 El administrador escribe el nombre o parte del nombre del conductor a buscar
	4 El administrador hace clic en el botón “Buscar”
	5 El prototipo carga todos los conductores es relacionados al texto escrito en el cuadro de texto “Buscar conductor” en la vista administrador
Excepción 1.	Paso Acción
	5 El prototipo no encuentra ningún conductor relacionado con el texto en el cuadro de texto “Buscar conductor”
	5 El prototipo muestra un mensaje informando que no existen resultados con el texto ingresado

Tabla 53. *Asignar ruta a un bus.*

Caso de uso: Asignar ruta a un bus	
Actor Principal	Usuario administrador y conductor
Descripción	Se asigna una ruta a un bus que no tenga ruta asignada y en caso de tenerla esta es reemplazada
	Paso Acción
	1 Caso de uso: Acceder a vista administrador
	2 El administrador hace clic en el botón “Asignar ruta” del bus que desea asignarle o cambiarle su ruta
	3 El prototipo carga la vista Asignar ruta a bus con el nombre del bus, ruta actual si tiene y con un listado de las rutas con el mismo tipo de bus (Padrón, articulado, alimentador) que el bus seleccionado.
	4 El administrador selecciona la ruta a asignar
	5 El administrador hace clic en el botón “Guardar”
	6 El prototipo asigna la ruta al bus
	7 El prototipo carga la vista Administrador con el bus y su nueva ruta asignada
Excepción 1.	Paso Acción
	4 El administrador selecciona “Quitar ruta”

	5	El administrador hace clic en el botón “Guardar”
	6	El prototipo quita la ruta al bus
	7	El prototipo carga la vista Administrador con el bus y sin ruta asignada
Excepción 2.	Paso	Acción
	3	El administrador hace clic en “Cerrar”

Tabla 54. *Asignar conductor a un bus.*

Caso de uso: Asignar conductor a un bus		
Actor Principal	Usuario administrador y conductor	
Descripción	Se asigna un conductor a un bus que no tenga conductor asignado y en caso de tenerlo este es reemplazado	
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el botón “Asignar conductor” del bus que desea asignarle o cambiarle su conductor
	3	El prototipo carga la vista Asignar conductor a bus con el nombre del bus, conductor actual si tiene y con un listado de los conductores disponibles.
	4	El administrador selecciona el conductor a asignar
	5	El administrador hace clic en el botón “Guardar”
	6	El prototipo asigna el conductor al bus
	7	El prototipo carga la vista Administrador con el bus y su nuevo conductor asignado
Excepción 1.	Paso	Acción
	4	El administrador selecciona “Quitar conductor”
	5	El administrador hace clic en el botón “Guardar”
	6	El prototipo quita el conductor al bus
	7	El prototipo carga la vista Administrador con el bus y sin conductor asignado
Excepción 2.	Paso	Acción
	3	El administrador hace clic en “Cerrar”

Tabla 55. *Ver información de un bus.*

Caso de uso: Ver información de un bus		
Actor Principal	Usuario administrador y conductor	
Descripción		
	Paso	Acción
	1	Caso de uso: Acceder a vista administrador
	2	El administrador hace clic en el botón “Info” del bus que desea ver la información
	3	El prototipo carga la vista con la información del bus

Apéndices C Selección de las herramientas

Para crear este prototipo se utilizó stackMEAN, Stack es un conjunto de tecnologías de un desarrollador para crear un cierto tipo de aplicaciones en este caso se utilizó MEAN.

MongoDB (Base de datos). Encargado de dar persistencia y almacenar los datos. Es una base de datos no SQL, puesto que no es fuertemente tipada, el tamaño y la velocidad de envío de los datos se ven beneficiados.

Angular (Interfaz).

También brinda un entorno de desarrollo ordenado el cual cuenta con componentes servicios, modelos, rutas, entre otros.

NodeJS (Servidor). Es un entorno de ejecución de JS. Si queremos crear una aplicación en el sistema operativo, necesitamos que el sistema entienda este lenguaje de alguna manera, para esto utilizamos **NodeJS**.

Express (Servidor). Es un Framework del lado del Backend de NodeJS, el cual nos permite utilizar métodos existentes para no tener que crear nuestro Backend desde 0, lo cual sería una tarea tediosa

la razón de usar todas estas tecnologías justas es que todas utilizan JavaScript, por lo tanto, vamos a utilizar el mismo lenguaje tanto en la base de datos como en el servidor y en la interfaz web. Lo cual reduce gran cantidad de tiempo y mejora la curva de aprendizaje.