

FINDROUTE, APLICACIÓN MÓVIL PARA EL APOYO EN EL PROCESO DE SELECCIÓN DE RUTA DE TRANSPORTE PÚBLICO EN EL MUNICIPIO DE IPIALES.



Universidad de **Nariño**

**DANY MAURICIO VILLOTA MUÑOZ
MIGUEL ANGEL CABRERA VELASQUEZ**

**PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE NARIÑO
IPIALES
MAYO 2018**

FINDROUTE, APLICACIÓN MÓVIL PARA EL APOYO EN EL PROCESO DE SELECCIÓN DE RUTA DE TRANSPORTE PÚBLICO EN EL MUNICIPIO DE IPIALES.

**DANY MAURICIO VILLOTA MUÑOZ
MIGUEL ANGEL CABRERA VELASQUEZ**

Trabajo de grado presentado como requisito parcial para optar el título de Ingenieros de Sistemas

Ing. JORGE ALBEIRO RIVERA ROSERO

ASESOR

**PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE NARIÑO
IPIALES
MAYO 2018**

NOTA DE RESPONSABILIDAD

“La universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Artículo 13, Acuerdo No. 005 del 26 de enero de 2010, emanado por el Honorable Consejo Académico de la Universidad de Nariño.

“Las ideas y conclusiones aportadas en el siguiente trabajo son responsabilidad exclusiva de los autores.”

Artículo 1ro del Acuerdo No. 324 de octubre 11 de 1966 emanado del Honorable Consejo Directivo de la Universidad de Nariño.

NOTA DE ACEPTACIÓN

Firma del jurado

Firma del jurado

Firma del asesor

Firma del co-asesor

AGRADECIMIENTOS

Al ingeniero Jorge Rivera, nuestro asesor por brindar su tiempo y ayuda incondicional en el desarrollo de nuestro trabajo de grado, aportando con sus conocimientos en los temas más importantes y necesarios para el proyecto, revisiones periódicas y por las ideas que permiten crear productos novedosos y valiosos para la comunidad.

A nuestros profesores tanto de la universidad de Nariño como de otras instituciones por apoyarnos y brindarnos sus conocimientos durante cada semestre tanto dentro y fuera de clases, permitiéndonos crecer en nuestra formación académica e integral y así ser mejores personas cada día.

A nuestros compañeros por ser un soporte durante el transcurso de nuestra carrera, compartiendo historias, conocimientos, anécdotas y sobre todo por enseñarnos lo valioso que es el trabajo en grupo para conseguir nuestras metas.

Agradecimientos especiales a mis padres,
Ulises Cabrera y Beatriz Velásquez, los
cuales me brindan su apoyo incondicional
en cada momento y son los que me
impulsan a seguir adelante para hacer
realidad mis sueños.

A mis compañeros y amigos que siempre
están en la lucha por el conocimiento y
nos impulsan para terminar exitosamente
nuestro proyecto, siendo nuestros guías y
personas claves que aportan nuevas ideas
Gracias.

Miguel Ángel Cabrera

Agradezco a mi madre Yolanda
Muñoz Ortega la mayor inspiración
en mi vida, por ser mi principal
fortaleza y fuerza quien ha luchado
constantemente enseñándome a
crecer personalmente apoyándome
en todo momento.

A mi novia Michele Revelo Solarte
quien me ha brindado su amor y
compromiso estando mi lado en los
momentos más difíciles y ha sido un
apoyo incondicional en mi
formación.

A mis amigos por creer en mí y en
darme fuerzas para dar cada día lo
mejor de mí.

Dany Mauricio Villota

RESUMEN

FindRoute es un grupo de tres aplicaciones una orientada a la web y dos a dispositivos móviles las cuales fueron desarrolladas para facilitar el uso del sistema de transporte público del municipio del Ipiales, así como también permitieron a los habitantes y extranjeros de la ciudad conocer las rutas de los buses.

Con la aplicación móvil FindRoute los usuarios propietarios de un dispositivo móvil con sistema operativo Android pueden observar las rutas y sub-rutas de los buses de transporte público, agregar rutas a favoritos y también obtener una sugerencia de ruta en el caso de no saber que ruta se debe utilizar. Alternativo a esto se presentaron dos aplicaciones, una web que facilitó la administración de rutas y conductores y una móvil la cual fue utilizada para determinar la ubicación del conductor.

Para el desarrollo de las aplicaciones se utilizaron herramientas como APIs de Google, sensores GPS de los dispositivos móviles y bibliotecas PolyUtil para obtener un producto valioso para el usuario.

Las pruebas de funcionamiento de FindRoute se realizaron en un entorno de prueba con usuarios que simularon diferentes roles como conductores, administradores, usuarios y se realizaron encuestas para dar soporte a los resultados obtenidos.

ABSTRACT

FindRoute is a group of three applications, two web and one mobile, which have been developed to facilitate the use of the public transport system of the Ipiales municipality as well as to allow the habitants and foreigners of the city to know the bus routes.

With the FindRoute mobile application, users who own a mobile device with an Android operating system can observe the routes, sub-routes of public transport buses, add routes to favorites and also obtain a route suggestion in case they do not know which route must be used. Alternative, two applications were presented, one web that facilitates the administration of routes and drivers and one mobile which is used to determine the location and route that the driver follows.

For the development of applications were used tools as APIs of Google, sensors GPS of the mobile device and PolyUtil library for the product useful the user.

The tests of operation of FindRoute were performed in a test environment with user simulated different roles as drivers, administrators, users and were made surveys to give support to the results obtained.

CONTENIDO

	Pág.
INTRODUCCIÓN	5
1.DESARROLLO DEL PROYECTO	25
1.1 Fase I – Planeación	25
1.1.1 Iteraciones	25
1.1.2 Identificación de requerimientos	25
1.1.3 Historias de usuario.	26
1.1.3.1 FindRoute conductor.....	26
1.1.3.2 Findroute administrador	27
1.1.3.3 FindRoute usuario.....	31
1.2 Fase II - Diseño.....	34
1.2.1 Arquitectura de las aplicaciones	34
1.2.1.1 Aplicación web	34
1.2.1.2 Aplicación móvil	34
1.2.2 Casos de uso	35
1.2.2.1 Casos de uso aplicación conductor.....	35
1.2.2.2 Casos de uso aplicación administrador.....	38
1.2.2.3 Casos De Uso Aplicación Usuario	43
1.2.3 Diagrama entidad relación	46
1.2.4 Diagrama de clases	47
1.2.4.1 Diagrama de clases conductor.....	47
1.2.4.2 Diagrama de clases administrador.....	48
1.2.4.3 Diagrama de clases usuario.....	49
1.3 Codificación	50
1.3.1 Desarrollo aplicación móvil usuario.....	50

1.3.2 Componente para el procesamiento de coordenadas.....	50
1.4 Pruebas.....	57
1.4.1 Pruebas funcionales.....	57
1.4.2 Pruebas funcionales de la sugerencia de ruta	60
1.4.3 Análisis de resultados de las pruebas realizadas a la función de sugerencia de rutas.....	64
1.4.4 Pruebas de estrés.....	64
1.4.5 Prueba de estrés 1. 10 usuarios simultáneos	64
1.4.6 Prueba de estrés 2. 100 usuarios simultáneos	67
1.4.7 Prueba de estrés 3. 1000 usuarios simultáneos	68
2.RESULTADOS OBTENIDOS.....	70
2.1 Requerimientos de funcionamiento.....	72
2.1.1 Usuario.....	72
2.1.2 Requerimientos de hardware.	72
2.2 Descripción de funcionamiento de aplicación FindRoute.....	72
2.2.1 Interfaz de inicio de sesión.....	72
2.2.2 Interfaz dar permisos	73
2.2.3 Interfaz de selección de rutas o sugerencias	75
2.2.4 Interfaz para observar la ruta seleccionada	76
2.2.5 Interfaz de ítems de aplicación	77
2.2.6 Interfaz de selección de sub-ruta.....	77
2.2.7 Interfaz para visualizar sub-ruta en mapa	78
2.2.8 Interfaz de sugerencia.....	79
2.3 Descripción de funcionamiento de aplicación FindRoute Conductor	80
2.3.1 Interfaz inicio de sesión.....	80
2.3.2 Interfaz visualizar ubicación conductor en el mapa.....	81
2.3.3 Interfaz de ítems de menú	81

2.4 Descripción de funcionamiento de aplicación FindRoute Administrador.....	82
2.4.1 Interfaz inicio de sesión.....	82
2.4.2 Interfaz principal.....	83
2.4.3 Interfaz Administrar conductores	84
2.4.4 Interfaz listar conductores	84
2.4.5 Interfaz modificar datos conductor	85
2.4.6 Interfaz buscar conductores.....	85
2.4.7 Interfaz administrar rutas	85
2.4.8 Interfaz graficar sub-ruta.....	86
2.4.9 Interfaz listar rutas.....	87
2.4.10 Interfaz modificar sub-ruta	87
3.CONCLUSIONES	88
4.RECOMENDACIONES	89
BIBLIOGRAFIA.....	90
ANEXOS.....	92

LISTA DE FIGURAS

Pág.

Figura 1. Arquitectura y componente de la plataforma Android. (Developers, Arquitectura de la plataforma Android, s.f.).....	13
Figura 2. Ventas mundiales de teléfonos inteligente (Gartner, 2017).	18
Figura 3. Gráfico de distribución de versiones de Android.....	20
Figura 4. Diagrama de casos de uso aplicación conductor.....	36
Figura 5. Diagrama de casos de uso administrador.....	38
Figura 6. Diagrama de casos de uso usuario.....	43
Figura 7. Diagrama entidad relación.	46
Figura 8. Diagrama de clases conductor.....	47
Figura 9. Diagrama de clases administrador.....	48
Figura 10. Diagrama de clases usuario.	49
Figura 11. Código de cálculo de rumbo en grados.	51
Figura 12. Gráfico de orientaciones.	51
Figura 13. Código de cálculo de orientaciones.	52
Figura 14. Código de cálculo de concurrencias.	52
Figura 15. Código de cálculo de orientación de ruta.....	53
Figura 16. Código de cálculo de orientación destino.	53
Figura 17. Código de decodificación de ruta encriptado.	54
Figura 18. Código para graficar ruta.	55
Figura 19. Código de obtención de coordenadas de conductores.....	56
Figura 20. Código de actualización de coordenadas.	57
Figura 21. Sugerencia universidad de Nariño centro.	60
Figura 22. Sugerencia Centro comercial gran plaza centro.	61
Figura 23. Sugerencia Centro Rumichaca.	62
Figura 24. Sugerencia Totoral Batallón de Ipiales.	63
Figura 25. Configuración de peticiones.....	65
Figura 26. Configuración de petición HTTP.	65
Figura 27. Resultados de prueba 1	66
Figura 28. Grafica de resultados de prueba 1.....	66
Figura 29. Resultado de peticiones correctas de prueba 1.....	67
Figura 30. Resultados de prueba 2.....	68
Figura 31. Resultados de prueba 3.....	69
Figura 32. Interfaz de inicio de sesión.	73
Figura 33. Interfaz de petición de permisos.	74
Figura 34. Interfaz de selección de rutas o sugerencias.....	75

Figura 35. Interfaz para observar la ruta seleccionada.	76
Figura 36. Interfaz de ítems de aplicación.	77
Figura 37. Interfaz de selección de sub-ruta.	77
Figura 38. Interfaz para visualizar sub-rutas en el mapa.	78
Figura 39. Interfaz de sugerencia de rutas.	79
Figura 40. Interfaz de inicio de sesión.	80
Figura 41. Interfaz visualizar ubicación conductor en el mapa.	81
Figura 42. Interfaz de ítems de menú.	82
Figura 43. Inicio de sesión.	83
Figura 44. Interfaz principal.	83
Figura 45. Interfaz conductores.	84
Figura 46. Interfaz listar conductores.	84
Figura 47. Interfaz modificar datos conductor.	85
Figura 48. Interfaz buscar conductores.	85
Figura 49. Interfaz administrar rutas.	86
Figura 50. Interfaz graficar sub-ruta.	86
Figura 51. Interfaz listar rutas.	87
Figura 52. Interfaz modificar sub-ruta.	87
Figura 53. Escala de valoración.	92
Figura 54. Encuesta pregunta número 1.	93
Figura 55. Encuesta pregunta número 2.	94
Figura 56. Encuesta pregunta número 3.	94
Figura 57. Encuesta pregunta número 4.	95
Figura 58. Encuesta pregunta número 5.	95
Figura 59. Encuesta pregunta número 6.	96
Figura 60. Encuesta pregunta número 7.	97
Figura 61. Encuesta pregunta número 8.	97
Figura 62. Encuesta pregunta número 9.	98
Figura 63. Encuesta pregunta número 10.	98

LISTA DE TABLAS

Pág.

Tabla 1. Descripción de historia de usuario.	21
Tabla 2. Descripción de requisitos no funcionales.	22
Tabla 3. HURF 1; Escoger ruta.....	26
Tabla 4. HURF 2; Autenticar conductor.	26
Tabla 5. HURF 3; Minimizar aplicación.....	26
Tabla 6. HURF 4; El usuario/conductor puede buscar las rutas.	27
Tabla 7. HURF 5; Como actualizar mi ubicación.	27
Tabla 8. HURNF 1; La interfaz será fácil de usar.....	27
Tabla 9. HURF 6; Iniciar sesión.	27
Tabla 10. HURF 7; Consultar ubicación de conductores.	28
Tabla 11. HURF 8; Registrar rutas.....	28
Tabla 12. HURF 9; Registrar conductores.	28
Tabla 13. HURF 10; Consultar información conductor.....	29
Tabla 14. HURF 11; Modificar datos de conductor.	29
Tabla 15. HURF 12; Eliminar datos de conductor.....	29
Tabla 16. HURF 13; Consultar información de rutas.	30
Tabla 17. HURF 14; Eliminar datos de ruta.	30
Tabla 18. HURF 15; Modificar datos de ruta.....	30
Tabla 19. HURNF 2; El administrador podrá cerrar sesión.....	31
Tabla 20. HURNF 3; La aplicación no permitirá ingresar datos erróneos.	31
Tabla 21. HURF 16; Iniciar sesión.	31
Tabla 22. HURF 17; Escoger rutas.	32
Tabla 23. HURF 18; Agregar a favoritos.	32
Tabla 24. HURF 19; Escoger sub-rutas.	32
Tabla 25. HURF 19; Escoger ruta de favoritos.	32
Tabla 26. HURNF 4; consultar sugerencia.....	33
Tabla 27. HURNF 5; La interfaz será fácil de manejar.....	33
Tabla 28. HURNF 6; La aplicación es escalable.....	33
Tabla 29. HURNF 7; Sistema operativo de la aplicación.	33
Tabla 30. Plantilla de caso de uso.	35
Tabla 31. Casos de uso aplicación conductor.....	36
Tabla 32. CUC #01 Seleccionar ruta a seguir.....	36
Tabla 33. CUC #02 Autenticar conductor.....	37
Tabla 34. CUC #03 Minimizar aplicación.	37
Tabla 35. Casos de uso aplicación administrador.....	38
Tabla 36. CUA #01 Autenticar ingreso.....	39

Tabla 37. CUA #02 Registrar conductores.....	39
Tabla 38. CUA #03 Consultar información de conductores.	39
Tabla 39. CUA #04 Modificar información de conductores.	40
Tabla 40. CUA #05 Eliminar conductores.	40
Tabla 41. CUA #06 Consultar ubicación de conductores en el mapa.	41
Tabla 42. CUA #07 Registrar rutas	41
Tabla 43. CUA #08 Consultar información de rutas.	41
Tabla 44. CUA #09 Modificar información de conductores.	42
Tabla 45. CUA #10 Eliminar información de rutas.	42
Tabla 46. Casos de uso aplicación usuario.....	43
Tabla 47. CUU #01 Autenticar con google.....	43
Tabla 48. CUU #02 Seleccionar ruta a seguir.....	44
Tabla 49. CUU #03 Seleccionar sugerencia de ruta.	44
Tabla 50. CUU #04 Escoger una sub-ruta.	44
Tabla 51. CUU #05 Agregar una sub-ruta a favoritos.....	45
Tabla 52. CUU #06 Escoger una sub-ruta favorita.	45
Tabla 53. CDP 1: Inicio de sesión aplicación web.	57
Tabla 54. CDP 2: Registro de conductores aplicación web.	58
Tabla 55. CDP 3: Registro de rutas aplicación web.	58
Tabla 56. CDP 4: Inicio de sesión aplicación conductor.	58
Tabla 57. CDP 5: Inicio de sesión aplicación usuario.	59
Tabla 58. CDP 6: Registro de favoritos aplicación usuario.	59
Tabla 59. Lista de pruebas funcionales.	59

MARCAS REGISTRADAS

PHP: es una marca registrada de The PHP Group © 2001-2015.

JAVA: es una marca registrada de ORACLE Corporation.

WINDOWS 10®: es una línea de sistemas operativos registrada por Microsoft Corp.

APACHE: es una marca registrada de The Apache Software Foundation © 2015.

ANDROID OPEN SOURCE PROYECT: es una marca registrada de Google Inc.

UML: marca registrada de OMG, consorcio dedicado al cuidado y el establecimiento de diversos estándares orientados a objetos.

INTRODUCCIÓN

El uso de dispositivos inteligentes en la sociedad ha tenido un gran incremento en los últimos años y es una de las soluciones principales para satisfacer diferentes necesidades, facilitando al usuario realizar gran parte de sus tareas cotidianas de manera organizada, eficiente y eficaz.

El sistema de transporte público urbano es un factor que está presente en la vida cotidiana de las personas, estos sistemas tienen diversas complicaciones, algunas de ellas afectan directamente al usuario, causándole inconvenientes al momento de esperar o elegir el bus adecuado para dirigirse a un destino deseado.

El sistema de transporte público del municipio del Ipiales también tiene procesos por mejorar, además, siendo este municipio lugar de frontera con Ecuador, implica que haya un gran número de extranjeros, quienes en su mayoría desconocen las rutas y los vehículos de transporte público de la ciudad.

En este proyecto se buscó brindar al usuario una herramienta para facilitar la elección del bus que cumpla la ruta más próxima a su lugar de destino, de igual manera una herramienta para que el conductor pueda seleccionar la ruta que va a seguir para ese día.

En el rol del administrador se planteó brindar una herramienta web que permita realizar modificaciones con respecto a los datos del conductor y las rutas que pueden tomar los buses colectivos en el municipio de Ipiales.

Para el desarrollo de las aplicaciones móviles se utilizó el SDK de Android, estas se apoyan en APIs de Google, el GPS para leer las coordenadas actuales y el lugar de destino del usuario para identificar la ruta más conveniente para él. Además, le permitió visualizar en el mapa la ubicación de los buses actualizándose cada 6 segundos y la distancia aproximada entre la posición del usuario y un segmento de la ruta.

La metodología que se utilizó para el desarrollo de este proyecto es la metodología XP (Programación Extrema), la cual brinda un desarrollo iterativo e incremental que nos permite asegurar la calidad de las herramientas a desarrollar.

Para la recolección de los datos se tuvo presente las historias de usuario por parte de clientes que utilizan el servicio de transporte público, algunas empresas de buses colectivos y conductores.

Contextualización de la comunidad

Ipiales, municipio del departamento de Nariño se encuentra ubicado sobre la frontera con Ecuador, y colinda con municipios como Pupiales, Gualmatan, El Contadero, Aldana, Puerres, Córdoba, Potosí, entre otros; por lo cual, un gran número de habitantes deciden ir a Ipiales a visitar los lugares turísticos, centros comerciales, restaurantes, etc. Esto permite que el comercio en el municipio de Ipiales se incremente y se construyan nuevos hoteles, barrios, centros comerciales, todo para el beneficio de la comunidad. (DÍAZ, La primavera económica de Ipiales Nariño, 2016)

Además, el transporte público en la zona urbana y rural de Ipiales ha mejorado en los últimos años, creando nuevas empresas de buses colectivos y vehículos pequeños de transporte público, los cuales llegan a la mayoría de barrios y sectores de Ipiales e incrementaron las horas de trabajo de los conductores, para favorecer a estudiantes, turistas y personas que requieren del servicio de transporte público.

El mejoramiento en el transporte público, permite a los usuarios ahorrar una parte del dinero que invierten al utilizar el servicio de taxis en donde el valor del servicio es más alto, dependiendo del lugar destino del usuario. (Ipiales, 2016)

Debido al uso masivo de dispositivos móviles, es muy probable que por lo menos una persona por familia tenga acceso a un dispositivo móvil. (Dinero.com, 2018)

Descripción del problema

Planteamiento del problema

Debido a que Ipiales está ubicado en un lugar de frontera donde existe gran concurrencia de habitantes del municipio y también de países como Ecuador, surgen falencias en el uso del transporte público, específicamente en el uso de los buses colectivos. Una de las tantas falencias que existen es la falta de tableros que muestren las rutas que toman los buses y herramientas informáticas que sugieran rutas y buses adecuados, que llevarían a los usuarios a su lugar de destino.

En muchas ocasiones los extranjeros y habitantes del municipio de Ipiales no saben que bus de transporte publico utilizar o no conocen el tiempo de espera para la llegada del bus y esto puede ocasionar pérdida de tiempo e incluso inseguridad.

En estos momentos el municipio de Ipiales se encuentra en un estado de cambio tecnológico debido al incremento de dispositivos móviles con altas capacidades.

Si las cosas siguen así, podría suceder que los extranjeros visiten con menos frecuencia el municipio de Ipiales o que aumente la inseguridad en la ciudad, esto causaría que el municipio empiece a decaer económicamente y que se utilice el servicio de taxi en ciertos horarios donde esperar los buses públicos es peligroso. (RADIO, 2016)

Por lo tanto, se requirió un método que facilite a las personas escoger el bus indicado para llegar a su lugar destino de manera fácil y efectiva, además de tratar de que las empresas de transporte público brinden un servicio más eficiente.

Formulación del problema

¿Cómo se facilitó al usuario de transporte público del municipio de Ipiales la elección de una ruta de bus adecuada?

Sistematización del problema

- ¿Cómo monitorear y administrar los datos básicos de los buses y conductores de transporte público en el municipio de Ipiales?
- ¿Cómo establecer una sugerencia de rutas para que el usuario pueda llegar a su lugar destino?
- ¿Cuál es la manera de saber que tan funcional y efectiva es FindRoute?

Objetivos

Objetivo general. Aportar al sistema de transporte público en el municipio de Ipiales mediante el desarrollo de una aplicación móvil bajo plataforma Android, la cual por medio del uso del GPS sugiere al usuario la ruta de buses de transporte público más conveniente para llegar a su lugar destino. Mediante el uso de GPS en el municipio de Ipiales.

Objetivos específicos

- Desarrollar un paquete de tres aplicaciones dos móviles y una web. Una app móvil para el conductor, con la cual se determina su ubicación y la ruta que sigue, una app móvil para que el usuario pueda conocer las rutas, la ubicación de los buses y obtener una sugerencia de la posible ruta para llegar a su lugar destino, un aplicativo web para el administrador, donde se podrá observar el recorrido y las coordenadas de los buses de transporte público en el municipio de Ipiales, como también gestionar los datos básicos de las rutas y el conductor del bus.
- Crear un componente para el procesamiento de coordenadas donde se reciba y procese la posición geográfica de los buses y el lugar de origen y destino del usuario, para así establecer la ruta sugerida.
- Determinar el nivel de satisfacción de la aplicación mediante entornos de prueba.

Justificación

El uso del transporte público en el municipio de Ipiales ha incrementado en los últimos años, por lo cual existen actualmente dos empresas de buses colectivos, donde la mayoría transitan por diferentes rutas dentro del municipio. Por tal motivo no todas las personas conocen las rutas que toman los buses y el tiempo que se demora en pasar el bus por la ruta más cercana a la posición del usuario, así que conocer las rutas por las cuales transitan los buses, permite que las personas del municipio como de diferentes lugares del país e incluso extranjeros ahorren su dinero al utilizar el sistema de transporte público.

Por otro lado, la mayoría de personas cuentan con algún dispositivo móvil con sistema operativo Android y planes de internet, así que, el uso de aplicaciones Android sigue incrementado en los últimos años, punto a favor para este proyecto en el cual, la herramienta directa de interacción con el usuario son las aplicaciones móviles que ofrecen a las personas información instantánea con respecto a la ubicación de buses colectivos y brinda sugerencias de las rutas que más se adecuan al lugar de destino. (Pulzo.com, s.f.)

La falta de información con respecto a las rutas que siguen los buses provocó que las empresas de transporte público pierdan clientes y causa que las personas gasten más dinero al utilizar taxis públicos, los cuales son más costosos.

Por estas razones fue pertinente el desarrollo de una aplicación web y dos aplicaciones móviles bajo plataforma Android, que brindaron una solución práctica en el proceso de selección de rutas de transporte público, donde el usuario pudo recibir sugerencias de las rutas más convenientes para llegar a su lugar de destino y la posición geográfica de cada bus, igualmente brindar a los administradores o personas a cargo de las empresas de buses colectivos un mayor control de los horarios y rutas que toman los conductores para cada día.

Marco referencial

Marco conceptual

En este proyecto de grado se contempló el desarrollo de 3 aplicaciones, que permiten dar una sugerencia de la mejor ruta de bus colectivo para que el usuario pueda llegar a su lugar deseado, se desarrolló dos aplicaciones móviles bajo plataforma Android, una app para el conductor el cual envía su posición geográficamente, otra app para el usuario en donde mira la rutas que siguen los buses y la ruta sugerida; Una aplicación web para el administrar los datos del conductor.

A continuación, se especifican los conceptos necesarios para tener claridad de los temas que se van abordar dentro del proyecto.

Sistema operativo Android. Sistema operativo, diseñado especialmente para dispositivos móviles como Smartphone y tabletas, el cual provee interfaces de comunicación entre dispositivos y usuario, es desarrollado por Google. (Ulmeher, s.f.)

Lenguaje de programación. Es un lenguaje diseñado para describir el conjunto de acciones, símbolos y reglas consecutivas que un equipo debe ejecutar. (EcuRed, s.f.)

Metodología ágil. Son una serie de técnicas para la gestión de proyectos, manejan un ciclo de vida iterativo e incremental de pocas semanas, para entregar prototipos pequeños y funcionales.(Randstad, 2017)

Aplicación web. Una aplicación web es una herramienta informática a la cual se puede acceder desde cualquier navegador ya sea una red local o a través de internet; es utilizada por empresas, personas para prestar servicios o vender productos.

Una página web tiene un diseño, en el cual se puede organizar el contenido que tendrá e información necesaria para que el usuario comprenda para que esta hecha y pueda interactuar con ella. (Neosoft, 2018)

Servidor web. Es un software que utiliza el protocolo de transferencia de hipertexto, HTTP (Hypertext Transfer Protocol), está diseñado para transferir hipertextos, páginas web o paginas HTML (HyperText Markup Languaje), para mostrar los archivos que forman páginas Web a los usuarios, en respuesta a sus solicitudes, que son reenviadas por los clientes HTTP de sus computadoras o dispositivos inteligentes. (Rouse, s.f.)

Base de datos relacional. Está compuesta por varias tablas, cada una tiene un nombre exclusivo y un conjunto de columnas y registros, para relacionarse con otra tabla lo hacen por medio de las claves primarias y las claves foráneas (o ajenas). (Rouse, Base de datos relacional, s.f.)

Android Studio. Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelloJ IDEA (ambiente de desarrollo integrado para el desarrollo de programas informáticos).

Además posee un editor de código, emulador con varias funciones, un entorno unificado en el que se puede realizar desarrollos para todos los dispositivos Android. Utiliza el lenguaje de programación Java y un lenguaje de marcado XML (eXtensible Markup Language) para desarrollar la interfaz. (Developers, Conoce Android Studio, 2018)

Lenguajes de codificación. A continuación, se hace una breve descripción de los lenguajes de programación que utiliza el sistema operativo Android para el desarrollo de las aplicaciones móviles.

Java: Es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. (ORACLE, s.f.)

Hay muchas aplicaciones y sitios web que no funcionan hasta que se instale java; La principal característica de java, es de qué se trata de un lenguaje orientado a

objetos diseñado para ser multiplataforma y poder ser empleado el mismo programa en diversos sistemas operativos.

Xml: según la organización W3C, es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML, pero su función principal es describir datos y no mostrarlos como es el caso de HTML.

XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

En el caso de Android, XML es utilizado para declarar los elementos de la interfaz de usuario, puesto que este cuenta con etiquetas simples que permiten definir el diseño y atributos de esta. (W3C, s.f.)

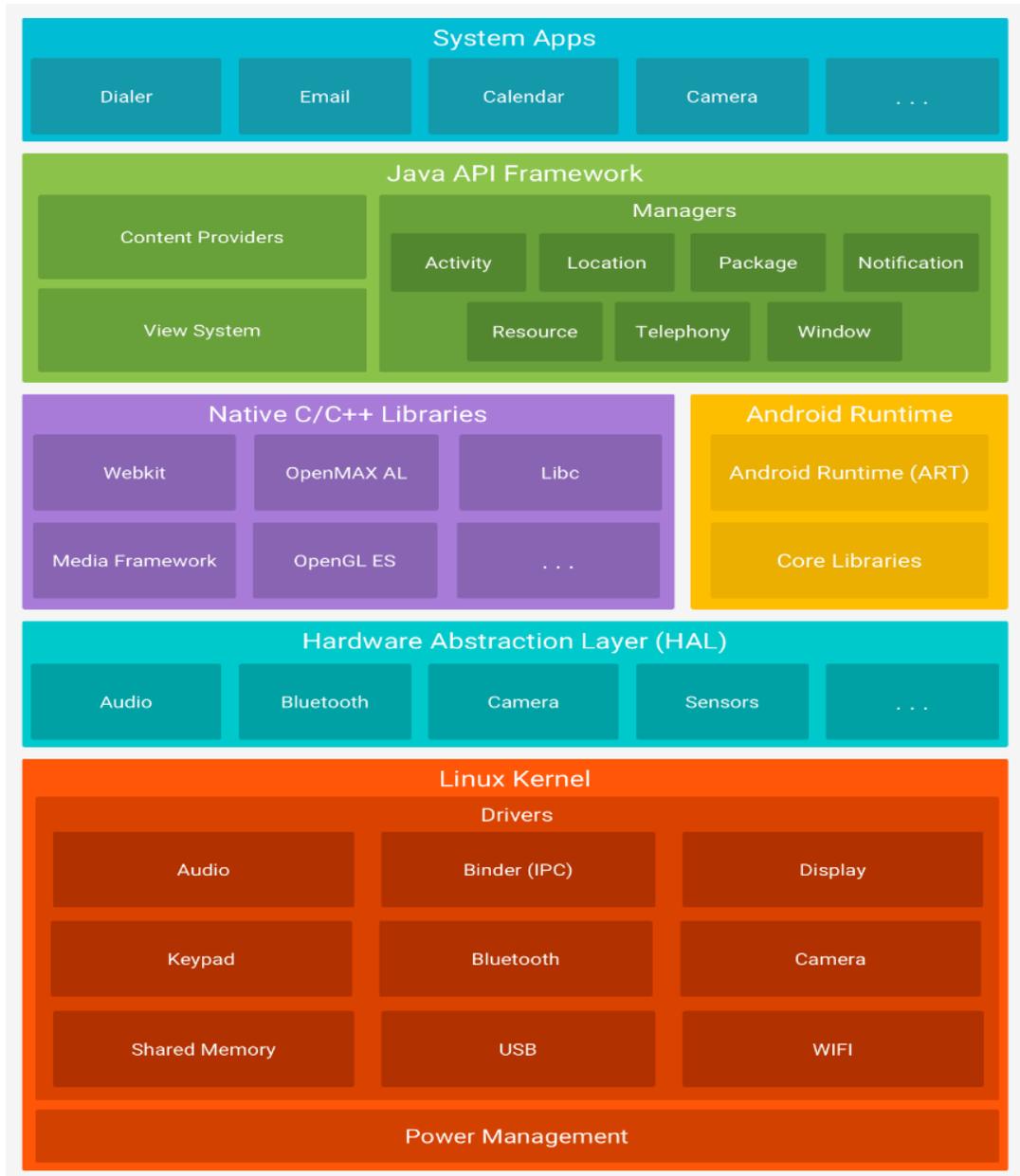
Motor de base de datos. El motor de base de datos por defecto que utiliza el sistema operativo Android es SQLITE el cual se describe a continuación.

SQLITE: es un motor de base de datos transaccional, que no tiene necesidad de usar servidor ni mucho menos configuraciones para su uso. SQLite lee y escribe directamente en archivos de disco ordinarios, el formato de archivos de la base de datos es multiplataforma, lo que permite copiar libremente una base de datos entre sistemas de 32 y 64 bits.

Al ser un motor de bases de datos liviano y poderoso se lo ha utilizado para almacenar información dentro de dispositivos inteligentes, reproductores de música, etc.

Arquitectura de plataforma Android. A continuación, se muestran los componentes principales de la plataforma Android. (Véase figura 3) de developers Android. (Developers, Arquitectura de la plataforma Android, s.f.)

Figura 1. Arquitectura y componente de la plataforma Android. (Developers, Arquitectura de la plataforma Android, s.f.)



Kernel de Linux. La plataforma de Android está basada en un núcleo de Linux, permite que Android aproveche funciones de seguridad claves y al mismo tiempo, permite a los fabricantes de dispositivos desarrollar controladores de hardware para un kernel conocido. Además, se encarga de gestionar los servicios básicos del sistema, tales como: la gestión de memoria, de procesos, Entrada/salida, red, etc.

Capa de abstracción de hardware (HAL). Brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al (framework de la Java API) de nivel más alto, funciona como una interfaz entre el software y el hardware del sistema mediante varios módulos de biblioteca, como módulos de cámara o bluetooth. Las aplicaciones no acceden directamente al hardware, sino que lo hacen a la capa abstracta provista por la HAL, luego el sistema Android carga el módulo de biblioteca para encontrar el componente adecuado de hardware.

Tiempo de ejecución de Android. Para los dispositivos con Android 5.0 o posteriores, cada app ejecuta sus propios procesos con sus propias instancias de las máquinas virtuales que ejecuta ART en dispositivos de memoria baja; ejecutando archivos DEX, formato de código de bytes diseñado para Android.

Para los dispositivos con Android menor de 5.0 quien era encargado del tiempo de ejecución era Dalvik, por tal motivo si tu aplicación se ejecuta normalmente en ART lo hará en Dalvik.

Bibliotecas. La mayoría de bibliotecas nativas para Android están escritas en C o C++, componentes y servicios centrales como el ART (Tiempo de ejecución de Android) y la HAL (Capa de abstracción de hardware). Mediante las APIs del framework de java se puede hacer uso de algunas de estas bibliotecas nativas.

Algunas de las bibliotecas más Utilizadas:

Cloud Storage API de CloudRail: su principal característica es que combina funciones comunes de los sistemas de almacenamiento en la nube, posee métodos como subir archivos, descargar archivos, crear directorios, copiar, mover y borrar.

Retrofit: es utilizada para realizar peticiones a un API REST, la función principal es convertir una API REST en una sencilla interfaz de java.

Roboguice: una herramienta que facilita mucho a la hora de la programación, mediante inyecciones de código. Los que permite es cargar las vistas sin necesidad de utilizar los findViewById.

Leanback v17: proporciona una API para admitir la creación de interfaces de usuario en dispositivos de Tv, además de ofrecer varios widgets importantes para una app de Tv.

Biblioteca de compatibilidad Custom Tabs: proporciona una API para admitir la adición y manejo de pestañas personalizadas en tus app, admite varias clases como CustomTabsService y CustomTabsCallback. (Developers, Funciones de la biblioteca de compatibilidad, s.f.)

OpenGL/ES: se trata de una librería gráfica, puede manipular graficas en 3D y ser utilizada en el momento de que haya hardware que soporte gráficos 3D.

Jnigraphics: expone una interfaz basada en C que permite el acceso confiable de código nativo a búferes de pixeles de los objetivos bitmap de java. (Developers, Android NDK Native API, s.f.)

Para el desarrollo de las aplicaciones móviles tanto para el conductor como para el usuario se describen a continuación las bibliotecas que permitirán el correcto funcionamiento de las aplicaciones.

Google-Gson: es una librería desarrollada por Google, su función es convertir objetos Java en JSON y se puede usar para convertir una cadena JSON a un objeto java, Gson puede trabajar con objetos arbitrarios de java, incluidos objetos de los que no tiene el código fuente. (inder123, 2015)

Biblioteca de compatibilidad Design: proporciona una API para admitir la adición de componentes y patrones de material design a las app como los paneles laterales de navegación, los botones de acción flotante (FAB), las pestañas.

Biblioteca de compatibilidad Multidex: proporciona compatibilidad con la compilación de apps mediante varios archivos Dalvik Ejecutable (DEX).

SQLite: mencionada anteriormente, se la va a utilizar para guardar datos en los dispositivos móviles.

Biblioteca appcompat v7: agrega compatibilidad para el patrón de diseño de la interfaz de usuario de la barra de acciones, también compatibilidad con las implementaciones de interfaz de usuario de material design.

Biblioteca de utilidades de la Google Maps Android Api: es una biblioteca de clases de código abierto que resulta útil para varias aplicaciones; en el caso del presente proyecto se la utiliza para decodificar polilíneas, graficarlas en el mapa, calcular la distancia en metro de puntos en el mapa, etc.

Framework de la Java API. Son las API's que se utilizan para hacer uso de las funciones del sistema operativo Android, igualmente son las bases necesarias para comenzar a desarrollar aplicaciones Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares como los siguientes, cuya descripción se basa en la siguiente referencia: (Developers, Android NDK Native API, s.f.)

- **Sistema de vista:** Puedes usar para compilar la IU de una app; se incluyen listas, cuadrículas, cuadro de texto, botones.
- **Administrador de recursos:** Facilita el acceso a recursos sin código, como Strings localizadas, gráficos y archivos de diseño.
- **Administrador de notificaciones:** Permiten que todas las aplicaciones muestren alertas personalizadas en la barra de estado.
- **Administrador de actividades:** Administra el ciclo de vida de las aplicaciones y proporcionan una pila de retroceso de navegación común.
- **Proveedores de contenido:** Permiten que las aplicaciones accedan a datos desde otras apps, como las aplicaciones de contactos, o compartan sus propios datos.
- **Administrador de ubicaciones:** Proporciona la ubicación geográfica del dispositivo Android a través de GPS.
- **Sistema de firebase:** Conjunto de herramientas que permiten iniciar sesión con cuenta de google.

Apps del sistema. Incluye las aplicaciones bases del dispositivo para correo electrónico, mensajería SMS, calendarios, entre otras, además de aquellas que el usuario ha instalado posteriormente. Las aplicaciones incluidas en la plataforma no tienen un estado inicial con respecto a las aplicaciones que el usuario instala, una app externa puede convertirse con una aplicación para reproducir música, video, etc.

Herramientas de aplicación web. A continuación, se describen las herramientas utilizadas para realizar la aplicación web.

PHP: según web oficial de PHP, creado especialmente para desarrollo de páginas web, es un lenguaje de código abierto, el código es ejecutado en el servidor, el cual genera código HTML y lo envía al cliente, el cual recibe el resultado. (The PHP Group, 2001)

Ajax: según IBM, está compuesta por un conjunto de tecnologías, las cuales permiten crear un nuevo enfoque al desarrollo de aplicaciones web. Su principal ventaja es que permite al usuario de la aplicación web interactuar con una página web con la facilidad de que esta no se vuelva a cargar completamente, solo con partes de la página de recarga y renovación. (Center, s.f.)

JavaScript: es uno de los lenguajes de programación que en la actualidad posee gran acogida por los desarrolladores de software, debido a sus funcionalidades como la facilidad de trabajar en diferentes sistemas operativos o en un navegador preferido, tiene la posibilidad de usar prototipos, para no usar clases, para el uso de herencia, es orientado a objetos y eventos, es súper liviano, etc. En definición JavaScript es muy útil y practico. (Caballero, s.f.)

Bootstarp: es un framework el cual fue presentado en el año 2012, y debido a los beneficios que presta para los desarrolladores de software ha tenido una buena aceptación durante estos últimos años. Entre sus principales beneficios están, la agilidad con que se puede crear las interfaces en nuestras páginas web, la adaptación de pantalla del dispositivos con el que se acceda a la página ya sea por medio de computador, dispositivo móvil, televisor, etc. Está diseñado para CSS y JavaScript, creado para facilitar la organización de contenido. (Robledano, 2015)

Notepad: es un editor de código fuente, utilizado para diferentes lenguajes de programación, es gratuito, sirve para ordenar código de forma sencilla. Es una herramienta indispensable al momento de iniciar el desarrollo de un software.

Desarrollo en dispositivos móviles. En la actualidad, el desarrollo de aplicaciones para dispositivos móviles es una tendencia que ha sufrido un crecimiento natural, existen muchos desarrolladores que trabajan para realizar aplicaciones que sean verdaderamente útiles para los usuarios y por lo tanto muchas aplicaciones que están en el mercado son gratuitas y otras tienen un valor monetario.

Un factor presente en el crecimiento de aplicaciones móviles es el gran auge que han tenido los Smartphones o teléfonos inteligentes y los diversos dispositivos inteligentes como Tablet o smartwatch, debido a la facilidad de manejo y movilidad que le brindan al usuario, además de las muchas funcionalidades que permiten al usuario realizar tareas en un menor tiempo y sin muchos costos.

Según datos publicados por Gartner, empresa consultora y de investigación de las tecnologías de la información con sede en Stamford, Connecticut, Estados Unidos, con respecto al primer trimestre de 2017, en este periodo se vendieron un total de 380 millones de unidades, lo que supone una subida del 9,1% con respecto a los 348 millones vendidos en el mismo periodo del año pasado. (gartner, 2017)

En cuanto a plataformas o sistemas operativos(OS) para dispositivos móviles, Gartner ha publicado unos datos que apuntan a que las plataformas que siguen en lo alto que son Android y Apple iOS representan el 98,4% de la participación del mercado total, pero sin duda Android permanece como líder con un 80,7% y Apple iOS con un 17,7%.

Como se puede observar en la figura 1, se muestran las ventas mundialistas de teléfonos inteligentes a usuarios finales por sistema operativo en el 4T16 (miles de unidades), liderando Android.

Figura 2. Ventas mundiales de teléfonos Inteligentes. (gartner, 2017)

Sistema Operativo	Unidades 4T16	Cuota de mercado 4T16 (%)	Unidades 4T15	Participación de mercado del 4T15 (%)
Android	352,669.9	81.7	325.394.4	80.7
iOS	77,038.9	17.9	71,525.9	17.7
Windows	1,092.2	0.3	4,395.0	1.1
Mora(buscar que es)	207.9	0.0	906.9	0.2
Otros sistemas operativos	530.4	0.1	887.3	0.2
Total	431,539.3	100.0	402,109.4	100.0

Por lo anterior se elige Android como sistema operativo móvil para el desarrollo de las aplicaciones móviles, tanto para el conductor como para el usuario.

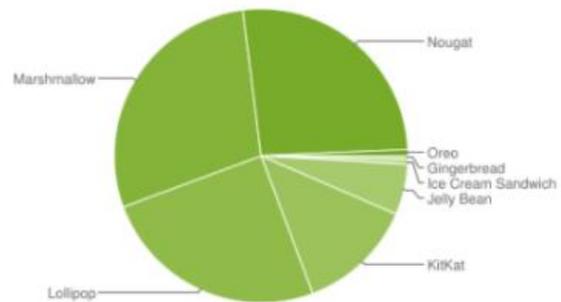
Android. Sistema operativo diseñado principalmente para dispositivos móviles con pantalla táctil, tales como Smartphones o Tablet, está basado en código libre y podría decirse que se trata del sistema que propone una interfaz donde se encuentra todas las aplicaciones de un teléfono o tableta.

Android no ha parado de evolucionar desde el lanzamiento de su primera versión en febrero de 2009 donde Google lanza la versión 1.1; con el pasar del tiempo surgieron nuevas actualizaciones y sacaron nuevas versiones de Android las cuales reciben el nombre en inglés de diferentes postres, siguiendo orden alfabético.

En la figura 2 se muestra datos del mes de enero de 2018 con diferentes versiones de Android que están presentes en el mercado, con el porcentaje de dispositivos que usan las diferentes versiones. Con los datos de la figura se puede observar cuales versiones son las más utilizadas y concluimos que la versión 6.0 tiene mayor distribución en los dispositivos, por lo tanto es necesario realizar las aplicaciones móviles para que funcionen en diferentes versiones de Android.

Figura 3. Gráfico de distribución de versiones de Android. (García, 2017)

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.5%
4.1.x	Jelly Bean	16	1.9%
4.2.x		17	2.9%
4.3		18	0.8%
4.4	KitKat	19	12.8%
5.0	Lollipop	21	5.7%
5.1		22	19.4%
6.0	Marshmallow	23	28.6%
7.0	Nougat	24	21.1%
7.1		25	5.2%
8.0	Oreo	26	0.5%
8.1		27	0.2%



Otras herramientas

Jmeter: herramienta diseñada para hacer simulaciones a cualquier recurso de software, diagnosticando el comportamiento de una aplicación en condiciones de producción. En este caso se realiza una consulta a la base de datos, inicio de sesión, tiempos de respuesta al realizar consultas al servidor por parte de varios usuarios a la vez.

Start-UML: herramienta para modelamiento en los estándares UML (Lenguaje Unificado de Modelado), sirve para realizar diagramas de casos de uso, diagrama de clases, diagrama de relación entidad entre otros.

Marco teórico

En el marco teórico se explican las metodologías, métodos, investigaciones, etc. Explicadas desde una perspectiva personal, las cuales fueron utilizadas para la elaboración del proyecto FindRoute.

Metodología XP

La metodología a seguir es XP (Programación Extrema), la cual propone una metodología basada en simplicidad y agilidad, además permite un desarrollo con ciclos cortos (llamados iteraciones) e incrementales, permitiendo realizar correcciones en cada iteración y generar productos funcionales al final del desarrollo de cada iteración, el trabajo es realizado por dos estudiantes, lo cual permite mayor avance en cada iteración.

Durante la elaboración del proyecto, los procesos y actividades desarrollados se realizaron en 4 iteraciones de aproximadamente 5 semanas, cada iteración complementa a las siguientes y se propuso un entregable por cada iteración.

Historias de usuario.

Debido a que se ha elegido la metodología Ágil XP, los requisitos están definidos a través de historias de usuario y seleccionados para identificar tanto los requisitos funcionales como los requisitos no funcionales.

Por lo regular las historias de usuario responden a tres preguntas.

- ¿Quién se beneficia?
- ¿Qué se quiere?
- ¿Cuál es el beneficio?

Además, las historias de usuario son cortas e independientes una de otra y se escriben tal y como las pronuncia el usuario. Esto permite desarrollar productos cada cierto tiempo y permite que en cada iteración haya la posibilidad de agregar, modificar o quitar historias de usuario, según el desarrollo del proyecto.

Se siguió el siguiente formato para cada historia de usuario:

Tabla 1. Descripción de historia de usuario.

# - Nombre de la historia	
Como [usuario] quiero [Obtener algo] para [Conseguir algo]	
Prioridad:	1 – 10 (min - max)
Iteración:	1 – 4

Criterio de aceptación
Criterio que hace que se cumpla la historia de usuario correctamente.

Dado que el proyecto consta de tres aplicaciones fue necesario realizar tres conjuntos de historias de usuario para las aplicaciones, obteniendo requisitos funcionales y no funcionales para cada aplicación.

Se realizaron entrevistas a estudiantes, trabajadores, conductores de buses, personas que trabajan en la parte administrativas de algunas empresas de buses y personas del municipio en general; Todo esto para obtener requisitos que permitan el buen desarrollo de las aplicaciones.

Los requisitos no funcionales son aquellos que no describen comportamientos funcionales a realizar, son características que se han de cumplir para que la aplicación funcione correctamente, se utilizó el siguiente formato para describir los requisitos no funcionales.

Tabla 2. Descripción de requisitos no funcionales.

- Nombre del requisito no funcional
Descripción :
Descripción del requisito

Servidor web. La comunicación de los datos se realiza mediante el modelo cliente servidor y utiliza protocolos, concretamente HTTP, el servidor web está pendiente de peticiones HTTP, las cuales son ejecutadas por un cliente HTTP o más conocidos como navegador web.

Tipos de servidores web

Según la página de ecured hay cuatro tipos de servidores los cuales son: (EcuRed, Servidor Web, s.f.)

Servidores basados en procesos: consiste en obtener una duplicación del proceso de ejecución, un diseño basado en procesos es cuando el proceso principal espera la llegada de una nueva conexión y en ese momento se duplica, creando una copia exacta.

Servidores basados en hilos (Threads): es más usado que el de procesos, debido a que la creación de un hilo no es tan costosa como la de un proceso, además que hereda muchas de las características de los servidores basados en procesos, varios hilos de un proceso pueden compartir datos entre ellos.

Servidores basados en sockets no bloqueantes o dirigidos por eventos: en este tipo de servidores su funcionamiento se basa en la utilización de lecturas y escrituras asíncronas sobre sockets, su objetivo principal es observar el estado de un grupo de sockets asociados a cada una de las conexiones.

Servidores implementados en el kernel: la función de este servidor es que mediante el movimiento de código de espacio de usuario a espacio de Kernel se intenta acelerar la velocidad de un servidor web.

Para la elaboración de nuestro proyecto es necesario almacenar los datos en un servidor para luego ser uso de ellos desde diferentes dispositivos.

Servidor bplaced. Para la aplicación se utilizó un servidor gratuito bplaced el cual brinda el servicio de ftp con 2 gigabyte de almacenamiento para subir los archivos mediante el programa gratuito Filezilla, crear 8 usuario, los cuales pueden ingresar al servidor ftp, un nombre de dominio gratuito asociado al servidor bplaced o permite asignarle un nombre de dominio privado a la página, permite administrar 8 base de datos mysql dentro del servidor y administrar 8 base de datos en Postgres.

Para obtener los beneficios Pro es necesario pagar 3 Euros cada mes y brinda, 16 cuentas ftp, 16 bases de datos mysql, 16 bases de datos Postgres, administración de archivos sin necesidad de filezilla, la información fue obtenida de la página del servidor bplaced. (bplaced.net, s.f.)

API'S de Google. Son necesarias para el desarrollo de aplicaciones, las cuales facilitan gran parte del trabajo, son gratuitas para el desarrollo de una gran variedad de casos en donde se la aplica, pero con algunos límites, los cuales tienen un precio monetario para acceder a ellos. Para nuestro caso se utilizarán API's totalmente gratuitas y es necesario obtener una llave que se la puede obtener en la página. (Developers, Android NDK Native API, s.f.)

Las API'S utilizadas tanto para el desarrollo de las aplicaciones móviles como para el desarrollo de la aplicación web se describen a continuación:

- **Google Maps JavaScript API:** permite colocar mapa personalizados con imágenes o contenidos propios en aplicaciones web.
- **Google Maps Android API:** Permite agregar un mapa personalizado a una aplicación móvil, configurando la inclinación, giro, zoom y desplazamiento.

1. DESARROLLO DEL PROYECTO

La metodología citada anteriormente presenta unas fases, las cuales se desarrollaron durante la elaboración del proyecto, a continuación se describe cada fase con su respectiva explicación.

1.1 Fase I – Planeación

1.1.1 Iteraciones. Se realizaron 4 iteraciones, las cuales se explican detalladamente a continuación.

La primera iteración estuvo centrada en el desarrollo de la aplicación móvil para los conductores de los buses de transporte público de Ipiales, la cual envía coordenadas de la ubicación del conductor y nombre de ruta que eligió a la aplicación web, que sería la segunda iteración.

La segunda iteración se centró en el desarrollo de la aplicación web para administradores, la cual permite almacenar, modificar, observar, eliminar datos básicos del conductor y rutas de transporte público.

La tercera iteración se centró en realizar el componente para el procesamiento de coordenadas, en esta iteración es necesario obtener la posición del conductor, del usuario, lugar destino del usuario, de las rutas, para establecer una sugerencia de una posible ruta que el usuario pueda optar para llegar a su lugar destino.

Con los resultados obtenidos en el componente para el procesamiento de coordenadas siguió la construcción de la aplicación para los usuarios de los buses de transporte público de Ipiales, que fue la cuarta iteración.

1.1.2 Identificación de requerimientos. Para la identificación de requerimientos fue necesario realizar unas breves entrevistas con las personas implicadas en los procesos que maneja el transporte público en el municipio de Ipiales, conociendo las necesidades que tiene los usuarios, los conductores de los buses y los administradores que manejan las rutas e información de los conductores; con esto se definieron los requisitos funcionales y requisitos no funcionales que llevaron las aplicaciones a desarrollar.

1.1.3 Historias de usuario.

Según la información citada anteriormente con respecto a las historias de usuario, se elaboraron las siguientes historias, las cuales fueron proporcionadas por conductores, usuarios, administradores de las empresas de buses.

1.1.3.1 FindRoute conductor

Requisitos funcionales

Tabla 3. HURF 1; Escoger ruta.

1 – Escoger ruta	
Como Conductor/usuario quiero escoger la ruta que voy a seguir para cada día.	
Prioridad:	10
Iteración:	1
Criterio de aceptación	
El conductor tiene que escoger la ruta que va a seguir para cada día.	

Tabla 4. HURF 2; Autenticar conductor.

2 – Autenticar conductor	
Como Conductor/ usuario quiero iniciar sesión para que lleven el registro de las rutas que elijo	
Prioridad:	9
Iteración:	1
Criterio de aceptación	
El conductor tiene que iniciar sesión con los datos dados por el administrador y escoger ruta para entrar a la aplicación.	

Tabla 5. HURF 3; Minimizar aplicación.

4 – Minimizar aplicación	
Como Conductor/usuario quiero poder salir de la aplicación para abrir otras aplicaciones como Facebook	
Prioridad:	8
Iteración:	1

Criterio de aceptación
El conductor puede minimizar la aplicación y seguir enviando las coordenadas de su ubicación mientras abre otras aplicaciones.

Tabla 6. HURF 4; El usuario/conductor puede buscar las rutas.

5 – El Usuario/conductor puede buscar las rutas
Descripción :
La aplicación brinda la opción de seleccionar ruta

Tabla 7. HURF 5; Como se actualiza la ubicación.

6– como se actualiza la ubicación
Descripción :
La aplicación permite que el GPS envíe datos al servidor cada cierto tiempo

Requisitos no funcionales

Tabla 8. HURNF 1; La interfaz será fácil de usar.

1 – La interfaz será fácil de usar
Descripción :
La aplicación es muy sencilla de utilizar, puesto que tiene funciones precisas para el conductor

1.1.3.2 Findroute administrador

Requisitos funcionales

Tabla 9. HURF 6; Iniciar sesión.

1 – Iniciar sesión	
Como Administrador/usuario quiero iniciar sesión cada vez que ingrese a la página para tener seguridad al momento de ingresar a la página.	
Prioridad:	10

Iteración:	2
Si el administrador tiene el usuario y contraseñas correctas puede acceder a la página web.	

Tabla 10. HURF 7; Consultar ubicación de conductores.

2 – Consultar ubicación de conductores	
Como Administrador/usuario quiero observar los conductores activos en el mapa para tener control de buses en movimiento	
Prioridad:	8
Iteración:	2
Criterio de aceptación	
En el mapa de la página se muestra los conductores activos su ubicación actualizada cada cierto tiempo.	

Tabla 11. HURF 8; Registrar rutas.

3 – Registrar rutas	
Como Administrador/usuario quiero trazar las rutas de los buses para tener la seguridad que siguen los lugares específicos.	
Prioridad:	9
Iteración:	2
Criterio de aceptación	
La página web utiliza una interfaz agradable e intuitiva para que el administrador grafique las rutas junto a las sub-rutas.	

Tabla 12. HURF 9; Registrar conductores.

4 – Registrar conductores	
Como Administrador/usuario quiero ingresar los datos de los conductores y buses para asignarle un usuario y contraseña al conductor.	
Prioridad:	9
Iteración:	2
Criterio de aceptación	
El administrador puede ingresar los datos necesarios de los	

conductores y buses a cargo, además de asignarles un usuario para que inicien sesión en la aplicación móvil para los conductores.

Tabla 13. HURF 10; Consultar información conductor.

5 – Consultar información conductor	
Como Administrador/usuario quiero tener a disposición datos del conductor para saber en cualquier momento información de algún conductor en específico.	
Prioridad:	8
Iteración:	2
Criterio de aceptación	
El administrador puede visualizar una lista donde se encuentra toda la información de los conductores como también visualizar en tiempo real la ubicación de cada conductor.	

Tabla 14. HURF 11; Modificar datos de conductor.

6 – Modificar datos de conductor	
Como Administrador/usuario quiero tener el poder de modificar o actualizar datos del conductor para tener la posibilidad de modificar datos en caso que el conductor cambie de bus o datos personales.	
Prioridad:	8
Iteración:	2
Criterio de aceptación	
La página permite al administrador modificar todos los datos del conductor, en caso de necesitarlo.	

Tabla 15. HURF 12; Eliminar datos de conductor.

7 – Eliminar datos de conductor	
Como Administrador/usuario quiero tener la posibilidad de eliminar el registro de un conductor para realizar este proceso en caso que el conductor sea despedido o reemplazado.	
Prioridad:	8
Iteración:	2

Criterio de aceptación	
La página permite al administrador eliminar el registro de un conductor en cualquier momento	

Tabla 16. HURF 13; Consultar información de rutas.

8 – Consultar información de rutas	
Como Administrador/usuario quiero tener la posibilidad de consultar los datos de las rutas para observar si otra ruta se pueda agregar.	
Prioridad:	8
Iteración:	2
Criterio de aceptación	
El administrador puede consultar los datos de las rutas y sub-rutas en cualquier momento deseado.	

Tabla 17. HURF 14; Eliminar datos de ruta.

9– Eliminar datos de ruta	
Como Administrador/usuario quiero tener la posibilidad de eliminar cualquier ruta para agregar otras o reemplazarla.	
Prioridad:	8
Iteración:	2
Criterio de aceptación	
La aplicación web permite al administrador eliminar las rutas que ya no sean de utilidad.	

Tabla 18. HURF 15; Modificar datos de ruta.

10 - Modificar datos de ruta	
Como Administrador/usuario quiero tener el poder de modificar o actualizar datos de las rutas para tener la posibilidad de modificar datos alguna sub-ruta, en caso de que la vía esté cerrada por varios meses.	
Prioridad:	8
Iteración:	2
Criterio de aceptación	
La página permite al administrador modificar cualquier datos con	

respecto a las rutas o sub-rutas

Requisitos no funcionales

Tabla 19. HURNF 2; El administrador podrá cerrar sesión.

1 – El administrador podrá cerrar sesión
Descripción :
La aplicación brinda la opción de cerrar sesión, la cual está en alguna parte de la pagina

Tabla 20. HURNF 3; La aplicación no permitirá ingresar datos erróneos.

2 – la aplicación permitirá ingresar datos erróneos
Descripción :
Los campos donde se ingresan los datos de conductores o rutas estarán validados para ingresar correctamente los datos, además tiene textos con ayuda de como ingresar los datos

1.1.3.3 FindRoute usuario

Requisitos funcionales

Tabla 21. HURF 16; Iniciar sesión.

1– Iniciar sesión	
Como usuario puedo crear cuenta de usuario para ingresar a la aplicación.	
Prioridad:	10
Iteración:	3
Criterio de aceptación	
Al momento de iniciar la aplicación por primera vez ingresar con cuenta de Google solamente.	

Tabla 22. HURF 17; Escoger rutas.

5 – Escoger rutas	
Como usuario quiero seleccionar la ruta para ver cuál es la ruta que me favorece.	
Prioridad:	10
Iteración:	3
Criterio de aceptación	
Al inicio de la aplicación se muestran una lista de rutas y al seleccionar una se muestran sus sub-rutas.	

Tabla 23. HURF 18; Agregar a favoritos.

4 – Agregar a favoritos	
Como usuario quiero guardar las sub-rutas que más observo para ahorrar tiempo cuando vuelva a ingresar a la aplicación.	
Prioridad:	7
Iteración:	3
Criterio de aceptación	
El usuario puede guardar en una lista de favoritos las sub-rutas que el mas frecuente.	

Tabla 24. HURF 19; Escoger sub-rutas.

5 – Escoger sub-rutas	
Como usuario quiero seleccionar sub-rutas para observar en el mapa descripción y posición de la sub-ruta	
Prioridad:	9
Iteración:	3
Criterio de aceptación	
Se permite seleccionar las sub-rutas individualmente para observarlas en el mapa y observar sus datos principales.	

Tabla 25. HURF 19; Escoger ruta de favoritos.

6 – Escoger ruta de favoritos

Como usuario quiero escoger una sub-ruta que seleccioné como favoritos para ahorrarme tiempo y llevar un control de las rutas que frecuente.	
Prioridad:	7
Iteración:	3
Criterio de aceptación	
La aplicación guarda una lista de las sub-rutas que el usuario guarde como favoritas.	

Requisitos no funcionales

Tabla 26. HURNF 4; consultar sugerencia.

1– Consultar sugerencia
Descripción :
La aplicación permite ingresar a la sugerencia por dos opciones, escogiendo ruta o directamente por un botón de sugerencia.

Tabla 27. HURNF 5; La interfaz será fácil de manejar.

2 – La interfaz será fácil de manejar
Descripción :
La interfaz tiene la información concreta y necesaria.

Tabla 28. HURNF 6; La aplicación es escalable.

3 – La aplicación es escalable
Descripción :
La aplicación ha de poder agregar nuevas funcionalidades sin afectar al resto de funciones.

Tabla 29. HURNF 7; Sistema operativo de la aplicación.

4 – Sistema operativo de la aplicación
Descripción :
La aplicación ha de estar adaptada para sistema operativo Android

1.2 Fase II - Diseño

1.2.1 Arquitectura de las aplicaciones

1.2.1.1 Aplicación web. La aplicación WEB permite al administrador la (inserción, modificación, eliminación y visualización) de datos del conductor y las rutas de los buses. El diseño fue realizado con una arquitectura Cliente-Servidor debido a que algunos procesos realizan peticiones las cuales proporcionan los recursos solicitados mediante el uso de un servicio de bases de datos; Por lo tanto se basa en el modelo de la arquitectura en tres capas.

Capa de Presentación: los componentes de esta capa solicitan y reciben los servicios de otros componentes, además propone una interfaz visual en donde el cliente puede observar la información recibida.

En esta parte es donde las aplicaciones dan a conocer al usuario final la información acerca de las rutas, buses, sugerencias, etc.

Capa de Negocios: sirve como un intermediario o puente entre la capa de presentación y la capa de datos, su principal función es verificar la solicitud del cliente, para posteriormente llevar a cabo una transacción en una base de datos y traer la respuesta a la solicitud del cliente; Esto sirve para evitar que el cliente tenga acceso directo a la base de datos.

Capa de Datos: es donde se encuentran los datos, permite insertar, modificar, borrar, consultar datos; las tareas básicas que se realizan.

En este caso, la base de datos fue creada en postgresSQL y consta de varias tablas necesarias para el funcionamiento de las aplicaciones a realizar.

1.2.1.2 Aplicación móvil. Las aplicaciones móviles las cuales permiten al usuario realizar consultas de las rutas que siguen los buses, datos y ubicación de los conductores fueron diseñadas con base en el patrón de diseño (MVC) para el sistema operativo Android (aplicación nativa), facilitando la separación de los datos y la lógica de negocios entre la interfaz de usuario y la aplicación.

Modelo: Contiene una representación de los datos que maneja el sistema y sus mecanismos de persistencia.

Vista: Es la interfaz de usuario las cuales muestran la información al cliente y está diseñada con las herramientas ofrecidas por Android Studio.

Controlador: Su función es ser intermediario entre el modelo y la vista, para gestionar el flujo de información entre ellos y así obtener los datos para utilizarlos según las necesidades de cada uno.

1.2.2 Casos de uso. Un caso de uso es una descripción de los pasos o actividades que se deben realizar para llevar a cabo un proceso. A continuación, se encuentra el listado de los casos de uso de la aplicación de conductor, cada aplicación tiene su lista de casos de uso y cada caso de uso tiene su identificador que empieza con las letras CU seguidas por el número de caso de uso. Además, cada caso de uso está asociado a un actor, bien sea el administrador del sistema, el conductor o el usuario de la aplicación.

Se siguió la siguiente plantilla para describir cada uno de los casos de uso, siguiendo siempre una misma estructura:

Tabla 30. Plantilla de caso de uso.

CU #00	
Actor:	Actor o actores implicados en el caso de uso
Prioridad:	1 – 10 (min - Max)
Precondición:	Condición a satisfacer antes de empezar el caso de uso
Extensiones:	Alternativas y tratamiento de situaciones insólitas
Descripción	
Secuencia de acciones e interacciones principales entre el actor y la aplicación	

1.2.2.1 Casos de uso aplicación conductor

Tabla 31. Casos de uso aplicación conductor.

IDENTIFICADOR	NOMBRE	ACTOR
CU1	Seleccionar ruta a seguir	Conductor
CU2	Autenticar conductor	
CU3	Minimizar aplicación	

Figura 1. Diagrama de casos de uso aplicación conductor.

Modelo de casos de uso conductor

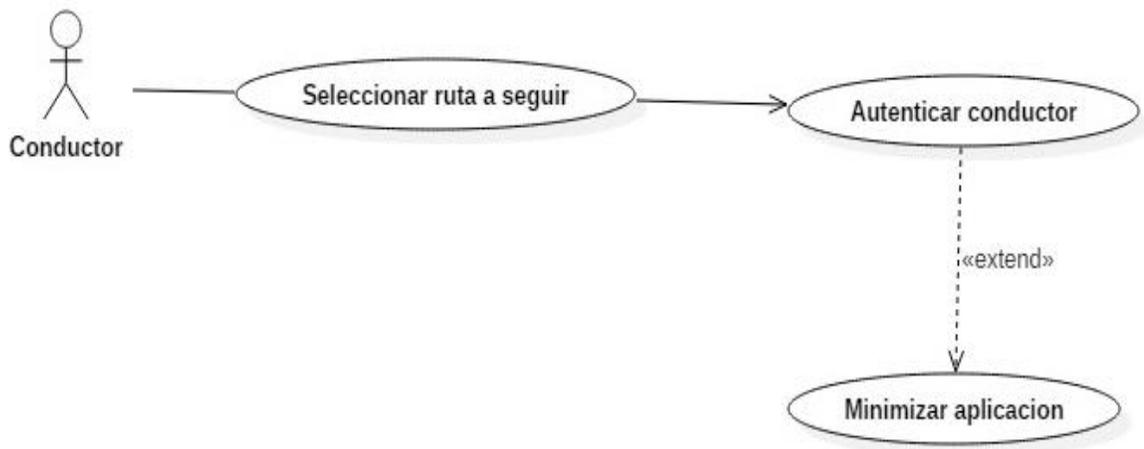


Tabla 32. CUC #01 Seleccionar ruta a seguir.

CU #01 Seleccionar ruta a seguir	
Actor:	Conductor
Prioridad:	10
Precondición:	Tener conexión a internet
Extensiones:	Si no hay conexión con el servidor, se pide encender datos o WIFI
Descripción	

El dispositivo Android establece conexión con el servidor principal y descarga la lista de rutas en el teléfono, el dispositivo muestra la lista y el conductor selecciona una ruta.

Tabla 33. CUC #02 Autenticar conductor.

CU #02 Autenticar conductor	
Actor:	Conductor
Prioridad:	9
Precondición:	Tener una ruta seleccionada
Extensiones:	Si no se ha seleccionado ninguna ruta, se muestra un mensaje de se debe seleccionar ruta y no se deja pasar a la siguiente interfaz.
Descripción	
El conductor ingresa usuario, contraseña y selecciona ingresar, el dispositivo se conecta con el servidor y valida si los datos ingresados son correctos, de ser así la aplicación cambia el estado del conductor como activo y envía a la siguiente interfaz.	

Tabla 34. CUC #03 Minimizar aplicación.

CU #03 Minimizar aplicación	
Actor:	Conductor
Prioridad:	8
Precondición:	Haber iniciado sesión correctamente
Extensiones:	ninguna
Descripción	
El conductor selecciona en el menú de opciones minimizar la aplicación, el dispositivo móvil envía constantemente las coordenadas del conductor en segundo plano y a su vez se muestra un mensaje de aplicación trabajando en segundo plano.	

1.2.2.2 Casos de uso aplicación administrador

Tabla 35. Casos de uso aplicación administrador.

IDENTIFICADOR	NOMBRE	ACTOR
CU1	Autenticar ingreso	Administrador
CU2	Registrar conductores	
CU3	Consultar información de conductores	
CU4	Modificar información de conductores	
CU5	Eliminar conductores	
CU6	Consultar ubicación de conductores en el mapa	
CU7	Registrar rutas	
CU8	Consultar información de rutas	
CU9	Modificar información de rutas	
CU10	Eliminar información de rutas	

Figura 2. Diagrama de casos de uso administrador.

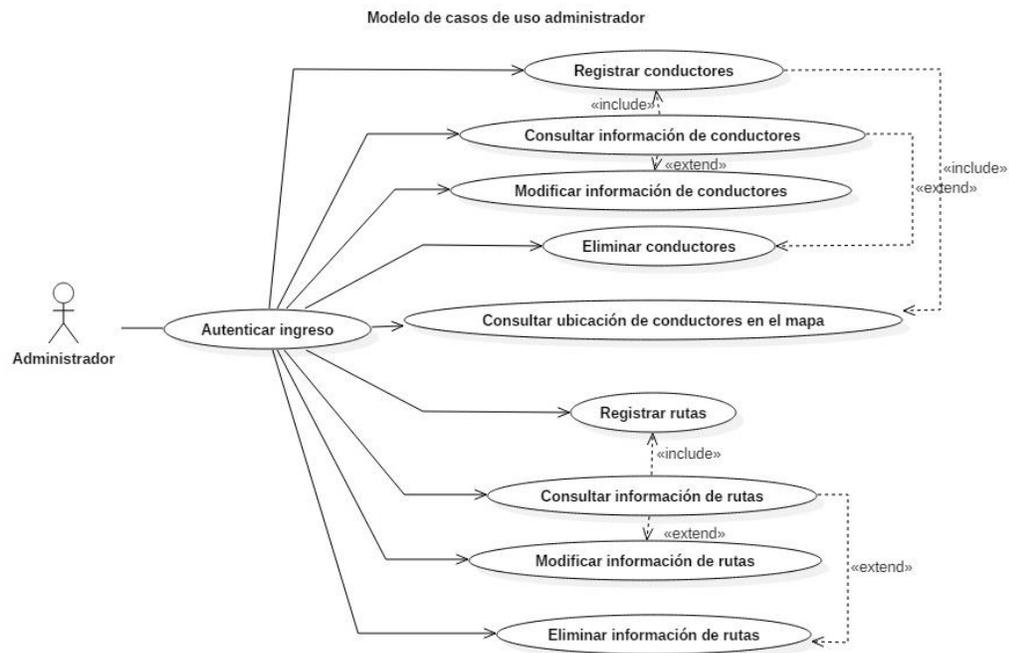


Tabla 36. CUA #01 Autenticar ingreso.

CU #01 Autenticar ingreso	
Actor:	Administrador
Prioridad:	10
Precondición:	Ninguna
Extensiones:	Si los datos ingresados no son correctos se muestra mensaje de error de conexión.
Descripción	
El Administrador ingresa usuario, contraseña y selecciona ingresar, el dispositivo se conecta con el servidor y valida los datos ingresados, de ser así deja pasar a la siguiente interfaz.	

Tabla 37. CUA #02 Registrar conductores.

CU #02 Registrar conductores	
Actor:	Administrador
Prioridad:	9
Precondición:	Haber iniciado sesión
Extensiones:	Si hay un conductor registrado con el mismo nombre de usuario anteriormente registrado se muestra un mensaje de error al registrar el conductor.
Descripción	
El Administrador ingresa los datos del conductor a registrar y selecciona registrar conductor, el programa realiza las validaciones pertinentes, registra el conductor y devuelve a la interfaz anterior.	

Tabla 38. CUA #03 Consultar información de conductores.

CU #03 Consultar información de conductores	
Actor:	Administrador
Prioridad:	8
Precondición:	Tener un conductor registrado
Extensiones:	Ninguna.
Descripción	

El Administrador selecciona la pestaña de mostrar información de conductores, la aplicación devuelve una lista con la información de todos los conductores registrados.

Tabla 39. CUA #04 Modificar información de conductores.

CU #04 Modificar información de conductores	
Actor:	Administrador
Prioridad:	9
Precondición:	Tener un conductor registrado
Extensiones:	Si hay un conductor registrado con el mismo nombre de usuario anteriormente se muestra un mensaje de error al registrar el conductor.
Descripción	
El Administrador selecciona la pestaña de modificación de conductores, la aplicación devuelve una lista con información de todos los conductores registrados, el administrador selecciona un conductor, automáticamente se abre una nueva interfaz donde se puede ingresar los nuevos datos del conductor.	

Tabla 40. CUA #05 Eliminar conductores.

CU #05 Eliminar conductores	
Actor:	Administrador
Prioridad:	8
Precondición:	Tener un conductor registrado
Extensiones:	Si el conductor se encuentra en movimiento no es posible eliminarlo.
Descripción	
El Administrador selecciona en la pestaña de lista de conductores el ítem De eliminar que se encuentra sobre cada conductor, que al realizar esta acción el conductor queda eliminado y la aplicación redirecciona a la lista de conductores.	

Tabla 41. CUA #06 Consultar ubicación de conductores en el mapa.

CU #06 Consulta ubicación de conductores en el mapa	
Actor:	Administrador
Prioridad:	8
Precondición:	Tener un conductor registrado
Extensiones:	El conductor debe estar en movimiento para poder ser visualizado en el mapa.
Descripción	
<p>El Administrador selecciona en la pestaña de lista de conductores el ítem De localización que se encuentra sobre cada conductor, que al realizar esta acción la aplicación redirecciona una interfaz con un mapa centrándose en la ubicación del conductor. Alternativo a esto también se puede observar en la página principal los conductores activos.</p>	

Tabla 42. CUA #07 Registrar rutas

CU #07 Registrar rutas	
Actor:	Administrador
Prioridad:	9
Precondición:	Haber iniciado sesión
Extensiones:	Si hay una ruta registrada con el mismo nombre se muestra un mensaje de error al registrar la ruta.
Descripción	
<p>El Administrador ingresa los datos de la ruta y selecciona cuantas sub-rutas tiene la ruta creada, el programa redirecciona a una nueva interfaz con los campos necesarios para registrar las nuevas sub-rutas, el administrador ingresa los datos y selecciona registrar ruta, la aplicación realiza las validaciones pertinentes y registra la ruta con sus sub-rutas, finalmente envía a la interfaz anterior.</p>	

Tabla 43. CUA #08 Consultar información de rutas.

CU #08 Consultar información de rutas	
Actor:	Administrador
Prioridad:	8
Precondición:	Tener una ruta registrada

Extensiones:	Ninguna.
Descripción	
El Administrador selecciona la pestaña de listar rutas, la aplicación devuelve una lista con información de todas las rutas registradas. Al seleccionar el ítem de más información se muestra una nueva interfaz con más información de la ruta seleccionada.	

Tabla 44. CUA #09 Modificar información de conductores.

CU #09 Modificar información de rutas	
Actor:	Administrador
Prioridad:	9
Precondición:	Tener una ruta registrada
Extensiones:	Si hay una ruta registrada con el mismo nombre de otra ruta se muestra un mensaje de error.
Descripción	
El Administrador selecciona en la pestaña de lista de rutas el ítem De modificar que se encuentra sobre cada ruta, lo cual permite cambiar el nombre de la ruta, pero si se quiere cambiar la información de las sub-rutas se debe seleccionar más información de la ruta, en esta nueva lista se encuentra la opción de modificar una a una las sub-rutas y también se encuentra la opción de agregar más sub-rutas.	

Tabla 45. CUA #10 Eliminar información de rutas.

CU #10 Eliminar información de rutas	
Actor:	Administrador
Prioridad:	9
Precondición:	Tener una ruta registrada
Extensiones:	Ninguno.
Descripción	
El Administrador selecciona en la pestaña de lista de rutas el ítem De eliminar que se encuentra sobre cada ruta. Pero si se quiere eliminar una sub-ruta se debe seleccionar el ítem de más información, donde se encuentra la opción de eliminar las sub-rutas, después de eliminar el programa redirecciona a la lista de rutas.	

1.2.2.3 Casos De Uso Aplicación Usuario

Tabla 46. Casos de uso aplicación usuario.

IDENTIFICADOR	NOMBRE	ACTOR
CU1	Autenticar con Google	Usuario
CU2	Seleccionar ruta a seguir	
CU3	Seleccionar sugerencia de ruta	
CU4	Escoger una sub-ruta	
CU5	Agregar una sub-ruta a favoritos	
CU6	Escoger una sub-ruta favorita	

Diagrama de Casos de uso – Usuario

Figura 3. Diagrama de casos de uso usuario.

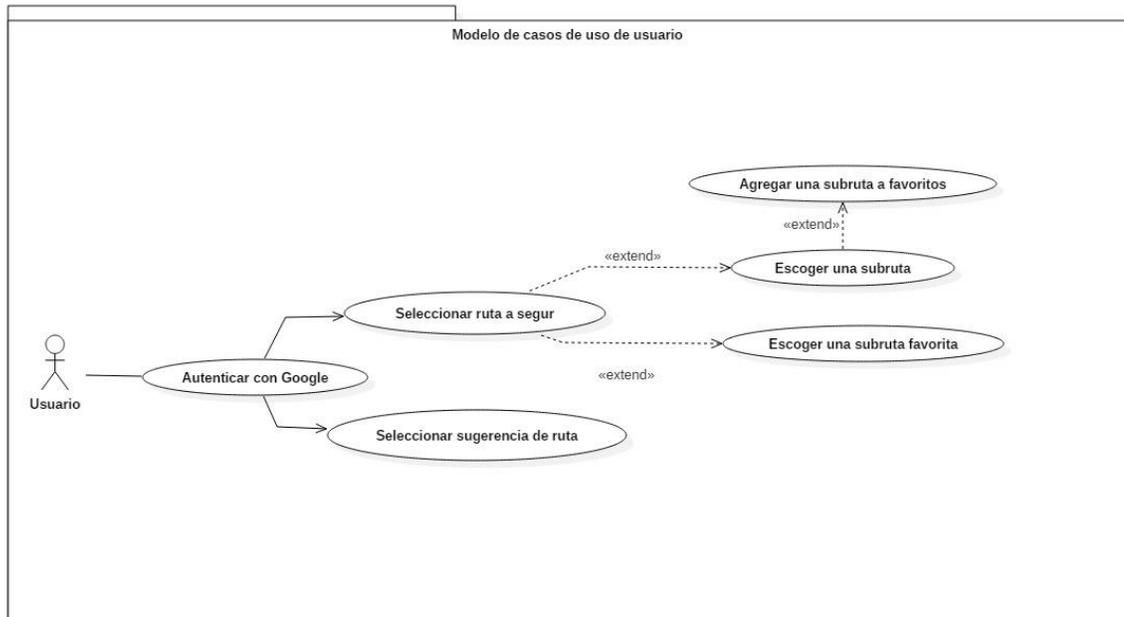


Tabla 47.CUU #01 Autenticar con google.

CU #01 Autenticar con Google	
Actor:	Usuario
Prioridad:	10
Precondición:	Tener una conexión a internet
Extensiones:	La aplicación pide encender WIFI o red de datos.

Descripción	
El usuario Inicia sesión directamente con Google, puede hacerlo creando un nuevo usuario o utilizando uno que se encuentre en el dispositivo Android.	

Tabla 48. CUU #02 Seleccionar ruta a seguir.

CU #02 Seleccionar ruta a seguir	
Actor:	Usuario
Prioridad:	10
Precondición:	Debe haber por lo menos una ruta descargada en el teléfono.
Extensiones:	Ninguna.
Descripción	
El usuario selecciona una ruta en la lista desplegable del menú principal, la aplicación redirecciona a la pantalla siguiente y muestra la ruta seleccionada con los buses ubicados en el mapa.	

Tabla 49. CUU #03 Seleccionar sugerencia de ruta.

CU #03 Seleccionar sugerencia de ruta	
Actor:	Usuario
Prioridad:	9
Precondición:	Tener encendido el GPS
Extensiones:	Se muestra un mensaje si no es posible encontrar la ubicación del usuario
Descripción	
El usuario selecciona el botón de sugerencia en la interfaz principal, la aplicación redirecciona a la pantalla siguiente y pide que seleccione un punto de destino en el mapa, la aplicación muestra la ruta sugerida con los buses ubicados en el mapa.	

Tabla 50. CUU #04 Escoger una sub-ruta.

CU #04 Escoger una sub-ruta	
Actor:	Usuario
Prioridad:	8
Precondición:	Haber seleccionado una ruta

Extensiones:	Ninguna
Descripción	
El usuario selecciona una sub-ruta del submenú, la aplicación muestra la sub-ruta seleccionada en el mapa con más detalles.	

Tabla 51. CUU #05 Agregar una sub-ruta a favoritos.

CU #05 Agregar una sub-ruta a favoritos	
Actor:	Usuario
Prioridad:	7
Precondición:	Haber escogido una ruta, tener conexión de red
Extensiones:	Si no hay conexión a internet, se le muestra un mensaje al usuario que no es posible agregar a favoritos sin conexión.
Descripción	
Al seleccionar una sub-ruta le aparece al usuario la opción de agregarla a favoritos, el usuario selecciona el ítem de favoritos en la aplicación y la sub-ruta se agrega a favoritos.	

Tabla 52. CUU #06 Escoger una sub-ruta favorita.

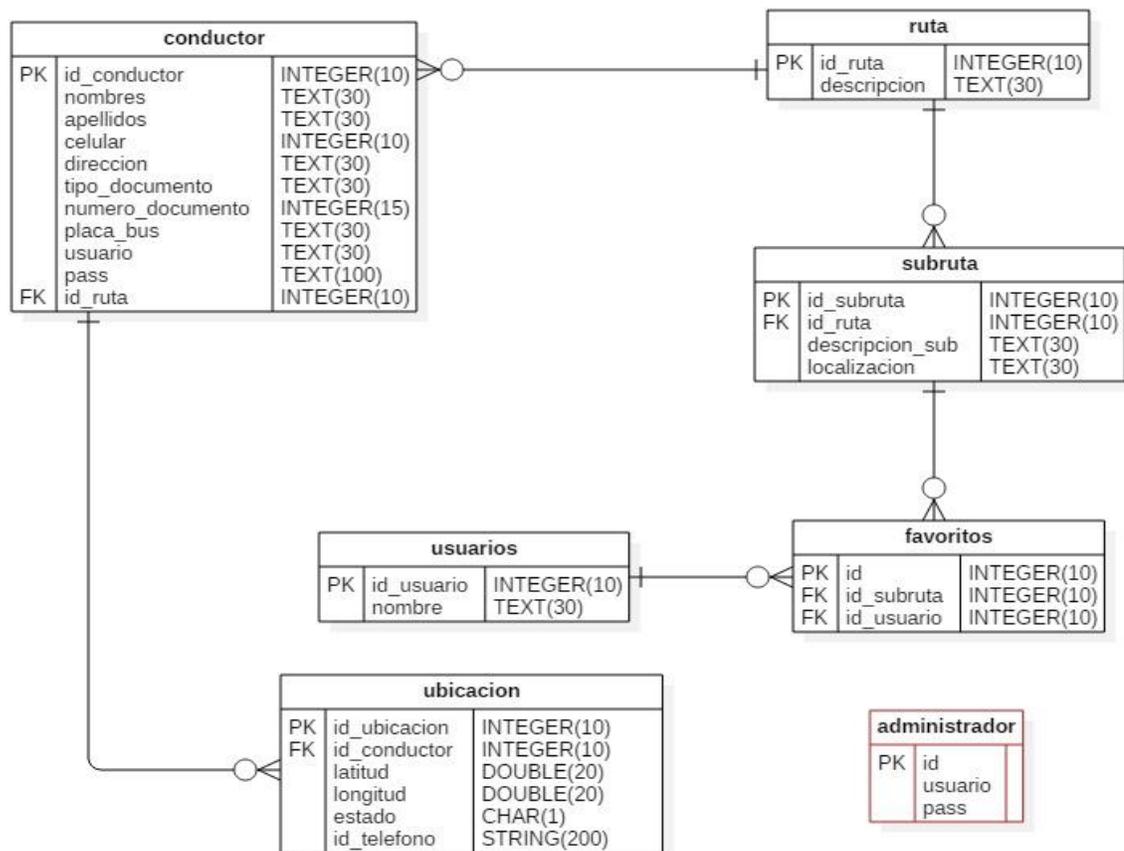
CU #06 Escoger una sub-ruta favorita	
Actor:	Usuario
Prioridad:	9
Precondición:	Haber Seleccionado una ruta
Extensiones:	Si no se ha agregado ninguna ruta a favoritos se muestra un menú en blanco.
Descripción	
El usuario selecciona una sub-ruta favorita del submenú, la aplicación muestra la sub-ruta seleccionada en el mapa con más detalles.	

1.2.3 Diagrama entidad relación. Es una herramienta para el modelado de datos el cual facilita representar las entidades fundamentales de un sistema, así como también sus interrelaciones y propiedades.

El siguiente diagrama presenta el diseño del modelo relacional de la base de datos utilizada en conjunto para las tres aplicaciones, mostrando como las tablas se relacionan entre sí.

Se ha seleccionado únicamente datos básicos de los conductores y solamente se guardaron datos necesarios de los usuarios para el correcto funcionamiento de la aplicación.

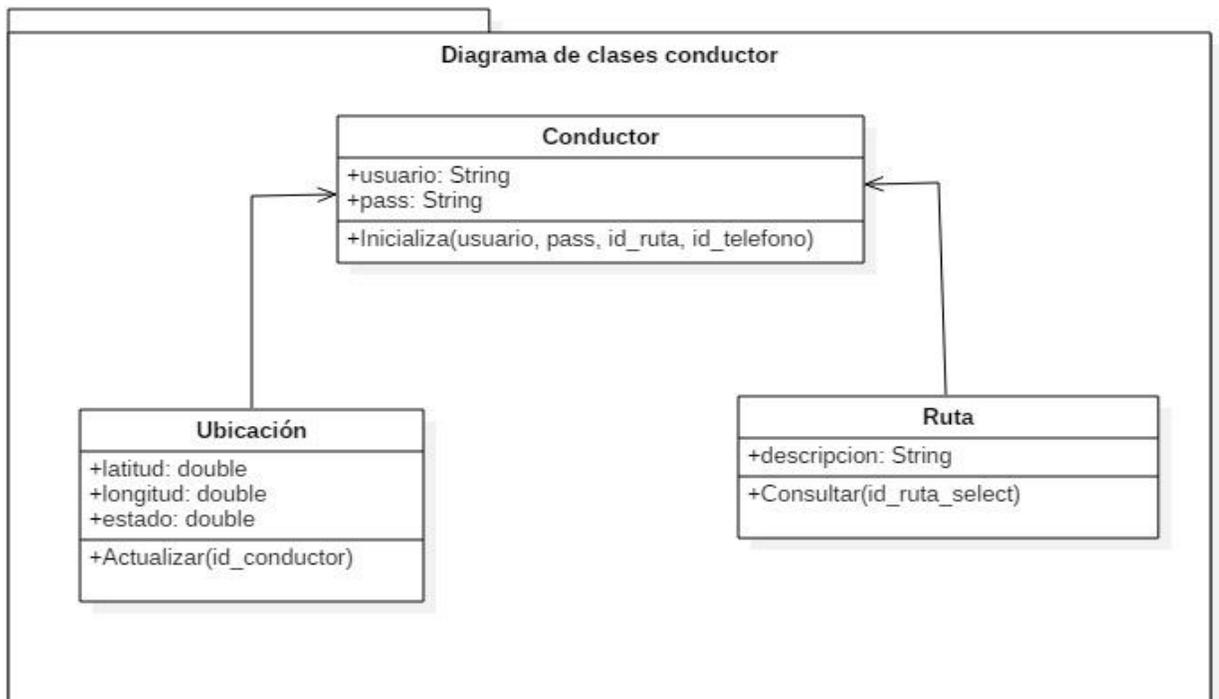
Figura 4. Diagrama entidad relación.



1.2.4 Diagrama de clases. Los siguientes diagramas describen la estructura del sistema mostrando las clases del mismo y las relaciones entre sus objetos. Ya que las tres aplicaciones funcionan por separado se ha optado por realizar uno para cada aplicación.

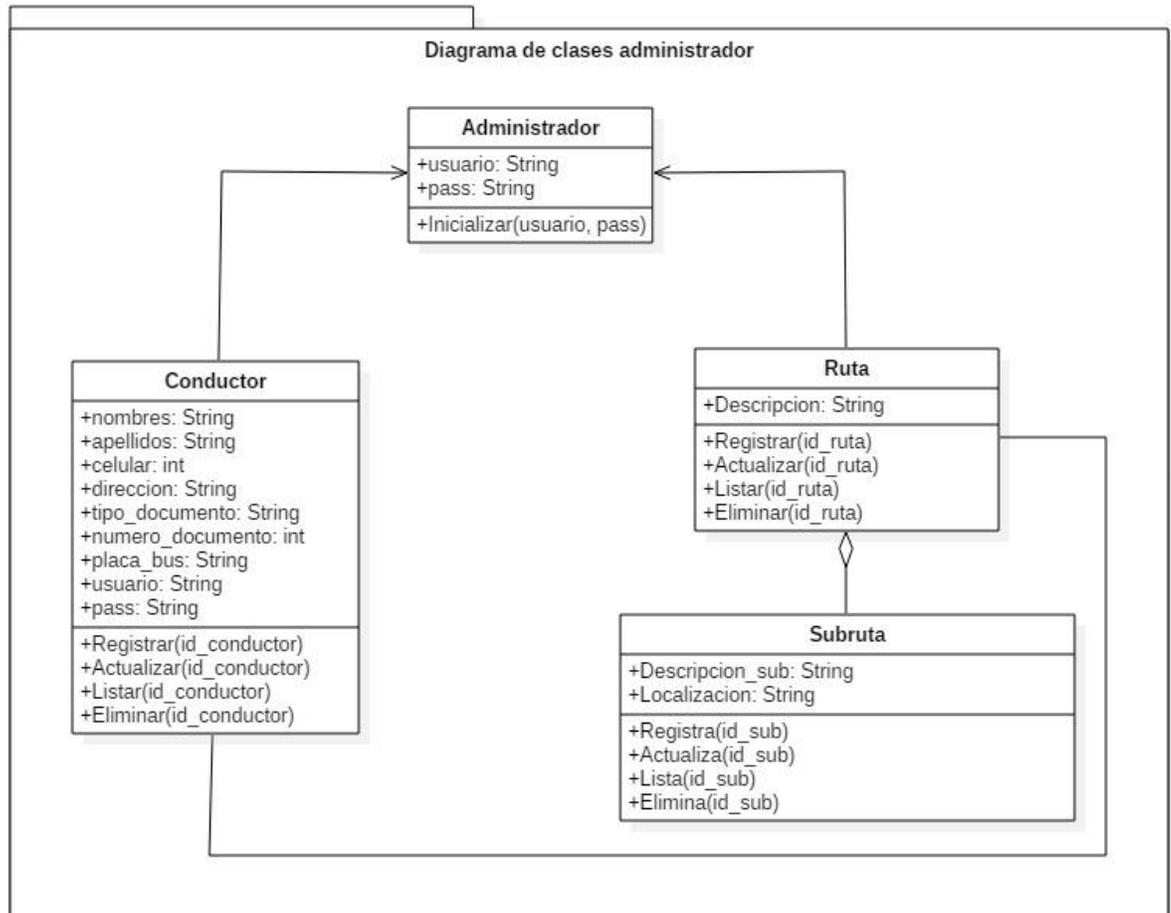
1.2.4.1 Diagrama de clases conductor

Figura 5. Diagrama de clases conductor.



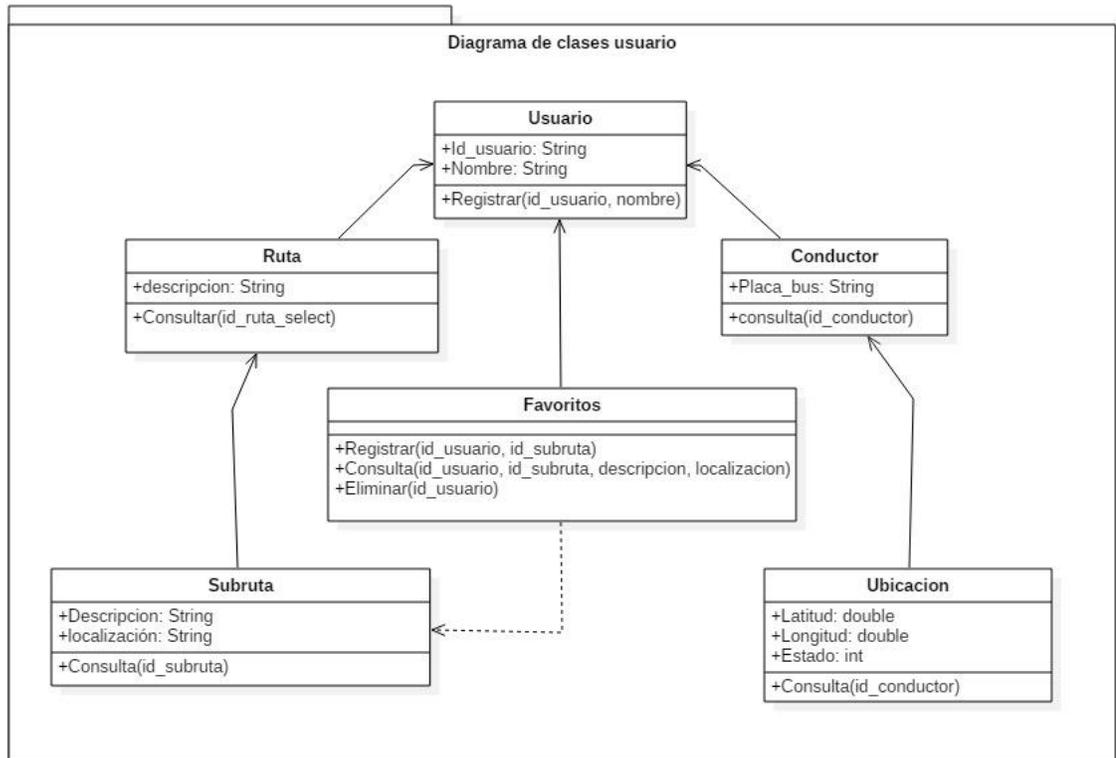
1.2.4.2 Diagrama de clases administrador

Figura 6. Diagrama de clases administrador.



1.2.4.3 Diagrama de clases usuario

Figura 7. Diagrama de clases usuario.



1.3 Codificación

1.3.1 Desarrollo aplicación móvil usuario

Para elaborar una sugerencia de rutas posibles para llegar a un lugar, fue necesario tener la ubicación actual del usuario, ubicación destino de usuario y coordenadas de cada punto de las rutas, para así realizar los cálculos necesarios para determinar una posible ruta que el usuario puede o no elegir para llegar a su lugar destino.

Primero, fue necesario saber la orientación que tiene el lugar destino con respecto al lugar actual del usuario, con esta información se recorre cada ruta obteniendo su respectiva orientación de cada punto con respecto al origen de la ruta y así se determinó la orientación que más se repite con la distancia mayor de recorrido, luego se compara las dos orientaciones, al ser iguales se prosigue comparando la distancia que existe entre el punto origen del usuario con cada punto de la ruta para determinar la distancia menor, de igual manera se determinó la distancia menor entre el punto destino del usuario y cada punto de la ruta, luego sumamos las distancias y se guarda el valor en una variable; Continuamos con la siguiente ruta, comparando de que tenga la misma orientación para determinar la distancia menor, al finalizar comparamos las distancias y seleccionamos la distancia menor para así graficar la ruta correspondiente a esa distancia, la cual es la sugerencia para el usuario.

A continuación se explican algunas de las secciones del código más relevantes en el desarrollo de algunos procesos de la aplicación FindRoute como es el módulo de procesamiento de coordenadas para generar una sugerencia de rutas y graficarlas en el mapa, actualización de coordenadas de ubicación de conductor en el mapa cada 6 segundos y otros procesos fundamentales para el funcionamiento de la aplicación.

1.3.2 Componente para el procesamiento de coordenadas. Para iniciar fue necesario tener una biblioteca de utilidades de google maps api para realizar algunos procesos, algunas clases importantes que se va a utilizar son.

SphericalUtil (): Es una utilidad geométrica esférica, con la cual se puede computar distancias, áreas y rumbos teniendo como datos básicos latitud y longitud de un punto.

Para este proyecto se utilizó **computeDistanceBetween ()**, la cual devuelve la distancia en metros entre diversos puntos de la ruta y la ubicación del usuario. Las coordenadas utilizadas detallan la latitud y longitud de cada ubicación.

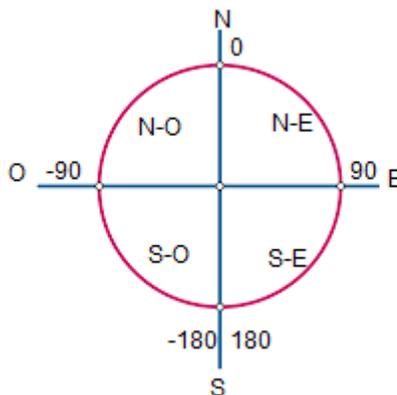
Para la sugerencia se utilizó **computeHeading ()**, el cual devuelve el rumbo, en grados entre dos coordenadas de latitud y longitud. El rango del valor de los grados esta entre -180 a 180.

Figura 8. Código de cálculo de rumbo en grados.

```
public static double computeHeading(LatLng from, LatLng to) {  
    // http://williams.best.vwh.net/avform.htm#CrS  
    double fromLat = toRadians(from.latitude);  
    double fromLng = toRadians(from.longitude);  
    double toLat = toRadians(to.latitude);  
    double toLng = toRadians(to.longitude);  
    double dLng = toLng - fromLng;  
    double heading = atan2(  
        sin(dLng) * cos(toLat),  
        cos(fromLat) * sin(toLat) - sin(fromLat) * cos(toLat) * cos(dLng));  
    return wrap(toDegrees(heading), -180, 180);  
}
```

Para identificar la orientación según los grados obtenidos, se utilizó el siguiente grafico.

Figura 9. Gráfico de orientaciones.



Calculando los grados del punto origen de la ruta con cada punto que contiene la ruta y encontrando orientaciones y distancia que se repiten para así calcular un

patron de la orientacion que sigue, donde n-e : norestre, n-o: noroeste, s-e: sureste, s-o: suroeste.

Figura 10. Código de cálculo de orientaciones.

```
for (int p= 1; p< decodedPath.size(); p++)
{
    punto_siguiete=decodedPath.get(p);

    /**
     * Con respecto al plano cartesiano y posiciones de los puntos
     * cardinales y se define orientación del destino usuario.
     */
    double metros =
SphericalUtil.computeDistanceBetween(decodedPath.get(0),decodedPath.get(p)
)/10;

    res_origen_lat=computeHeading(punto_origen,punto_siguiete);

    if (res_origen_lat>=0 && res_origen_lat <= 90)
    {
        estado_ruta="n-e";

    }else
    if (res_origen_lat>=90 && res_origen_lat <=180)
    {
        estado_ruta="s-e";
    }else
    if (res_origen_lat<=0 && res_origen_lat>=-90)
    {
        estado_ruta="n-o";
    }else
    if (res_origen_lat<=-90 && res_origen_lat>=-180)
    {
        estado_ruta="s-o";
    }
}
```

Determina la distancia en metros del recorrido con cada orientacion, para luego encontrar la de mayor distancia.

Figura 11. Código de cálculo de concurrencias.

```
if (estado_ruta=="n-e")
{
    m1=m1+metros;
}
if (estado_ruta=="n-o")
{
    m2=m2+metros;
}
```

```

if (estado_ruta=="s-e")
{
    m3=m3+metros;
}
if (estado_ruta=="s-o")
{
    m4=m4+metros;
}

```

Así se determinó la orientación que sigue la ruta, e:este o n:norte.

Figura 12. Código de cálculo de orientación de ruta.

```

if (m1>m2 && m1>m3 && m1>m4)
{
    nombre="e";
}
if (m2>m1 && m2>m3 && m2>m4)
{
    nombre="o";
}
if (m3>m2 && m3>m1 && m3>m4)
{
    nombre="e";
}
if (m4>m2 && m4>m3 && m4>m1)
{
    nombre="o";
}

```

Para calcular la orientación destino del usuario se utilizó **computeHeading ()**, con las coordenadas de la posición origen y destino del usuario, así se determinó el estado destino para compararla con la orientación de cada ruta.

Figura 13. Código de cálculo de orientación destino.

```

prue=computeHeading(coordenadas, destino);
/**
 * Con respecto al plano cartesiano y posiciones de los puntos
 * cardinales, se define la orientación del destino del usuario.
 */if (prue>=0 && prue <= 90)
{
    estado_destino="e";
}else
if (prue>=90 && prue <=180)
{

```

```

    estado_destino="e";
}else
if (prue<=0 && prue>=-90)
{

    estado_destino="o";
}else
if (prue<=-90 && prue>=-180)
{

    estado_destino="o";
}
}

```

Comparando la orientacion destino del usuario y orientacion de cada sub-ruta y si son iguales se calcula las distancia menor de cada sub-ruta, para almacenarla en una variable y comparar con la distancia menor del resto de sub-rutas con la misma orientacion y asi determinar cual es la sub-ruta mas conveniente para que el usuario pueda optar en utilizarla.

PolyUtil (): Es util para graficar polilneas y poligonos obteniendo cordenadas de latitud y longitud codificados o decodificados.

Para iniciar fue necesesario crear un poligono o polilinea en Google Maps, el cual almacenó coordenadas de latitud y longitud como una cadena codificada. En este proyecto la polilinea la crea el administrador de la pagina web con ayuda de Google Maps y almacena la cadena en la base de datos, que posteriormente sera utilizada para graficar la ruta en el mapa.

En la aplicación para el usuario se utilizó **decode()** para decodificar la ruta y guardar cada punto en una lista, en la cual puede hacer uso de cada una de las coordenadas ya sea para graficar partes de la ruta con diferente color y tamaño o solo trazar secciones de la ruta en el mapa.

Figura 14. Código decodificación de ruta encriptado.

```

public static List<LatLng> decode(final String encodedPath) {
    int len = encodedPath.length();
    final List<LatLng> path = new ArrayList<LatLng>();
    int index = 0;
    int lat = 0;
    int lng = 0;

    while (index < len) {
        int result = 1;
        int shift = 0;
        int b;

```

```

do {
    b = encodedPath.charAt(index++) - 63 - 1;
    result += b << shift;
    shift += 5;
} while (b >= 0x1f);
lat += (result & 1) != 0 ? ~(result >> 1) : (result >> 1);

result = 1;
shift = 0;
do {
    b = encodedPath.charAt(index++) - 63 - 1;
    result += b << shift;
    shift += 5;
} while (b >= 0x1f);
lng += (result & 1) != 0 ? ~(result >> 1) : (result >> 1);

path.add(new LatLng(lat * 1e-5, lng * 1e-5));
}

return path;
}

```

Para trazar la ruta en el mapa, fue necesario recorrer cada punto de la lista decodificada y se dio color y tamaño y se guardó en una variable tipo PolylineOptions, que posteriormente será graficada en el mapa.

Figura 15. Código para graficar ruta.

```

PolylineOptions optionss = new PolylineOptions();
/**
 * Se Recorre caa punto de la ruta para asignarle color, tamaño.
 */
for (int i = 0; i < decoded_graficar.size(); i++) {
    point = decoded_graficar.get(i);
    optionss.width(10);
    optionss.color(Color.GREEN);
    optionss.add(point);
}

```

Para colocar los marcadores de los buses en el mapa y que se actualizen cada cierto tiempo fue necesario que la aplicación del conductor esté activa y enviando sus coordenadas al servidor, el cual almacena los datos y posteriormente el aplicativo para el usuario realiza peticiones al servidor y trae las coordenadas para graficar el marcador cada 6 segundos según la ruta que haya escogido el conductor para seguir ese día.

En el aplicativo movil para el usuario se realiza una petición por medio de un archivo php que obtiene las coordenadas de los conductores y los guarda en variables para enviarlas al metodo agregarMarcadorC.

Figura 16. Código de obtención de coordenadas de conductores.

```
@Override
public void doInBackground() {
    try {
        respStr = "";
        LinkedHashMap<String, String> nameValuePairs = new
LinkedHashMap<String, String>();
        /**
         * Envía el id_ruta para realizar una consulta y traer los conductores
         con su identificador.
         */
        nameValuePairs.put("id_ruta_select", id_ruta);
        mRequest = new MRequest(new URL(getResources().getString(R.string.url)
+ "/obtener.php"), "");
        respStr = mRequest.execute(nameValuePairs);

    } catch (Exception e) {
        respStr = "Para esta opción es necesario estar conectado a internet";
        e.printStackTrace();
    }
}
@Override
public void onPostExecute() {
    try {
        /**
         * JSONArray con los datos de conductor.
         */
        JSONArray arr = new JSONArray(respStr);
        for (int i = 0; i < arr.length(); i++) {
            JSONObject obj = arr.getJSONObject(i);
            latc = Double.parseDouble(obj.getString("latitud"));
            lngc = Double.parseDouble(obj.getString("longitud"));
            placa = (String) obj.get("placa_bus");
            /**
             * Se envía los datos de conductor para agregar el marcador en mapa */
            agregarMarcadorC(latc, lngc, placa);
        }

    } catch (JSONException e) {
        respStr = "Para esta opción es necesario estar conectado a internet";
        e.printStackTrace();
    }
}
}
```

Obteniendo las coordenadas del posicionamiento geografico de los buses se pone los marcadores en el mapa y esto se repite cada cierto tiempo para actualizar la ubicación de los conductores.

Figura 17. Código de actualización de coordenadas.

```
private void agregarMarcadorC(double lat_bus, double lng_bus, String
placa_bus) {
    LatLng coordenadas_bus = new LatLng(lat_bus, lng_bus);
    marcador1 = mMap.addMarker(new MarkerOptions()
        .position(coordenadas_bus)
        .title("Bus colectivo")
        .snippet("Placa # :"+placa_bus)

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.ic_directions_bus_b
lack_24dp)));
}
```

1.4 Pruebas

Las pruebas funcionales se utilizan para verificar que las aplicaciones desarrolladas cumplan con los requerimientos funcionales y para localizar posibles errores no contemplados durante su desarrollo.

Se realizaron pruebas a la sugerencia de rutas donde se procede a ensayar diferentes tipos de ubicaciones para verificar si hay incongruencias en su funcionamiento.

También se realizó una serie de pruebas de estrés para probar la capacidad que tiene el servidor BPLACED de ejecutar múltiples peticiones de ingreso.

En este caso las pruebas se realizaron ejecutando la consulta que se utiliza para obtener la ubicación de los conductores y mostrarlos en pantalla, ya que las tres aplicaciones realizan la misma tarea y trabajan conjuntamente con un mismo web service no hay necesidad de realizarlas por separado.

1.4.1 Pruebas funcionales

Tabla 53. CDP 1: Inicio de sesión aplicación web.

Nombre del caso de prueba: Inicio de sesión aplicación web
Descripción: Se validan los datos de ingreso del administrador a la aplicación web.
Prerrequisitos: Tener asignado un usuario y contraseña.

Pasos: Ingresar usuario, ingresar contraseña, seleccionar el botón Ingresar.
Resultado esperado: El administrador tiene acceso a la aplicación web.
Resultado obtenido: Ingreso exitoso.

Tabla 54. CDP 2: Registro de conductores aplicación web.

Nombre del caso de prueba: Registro de conductores aplicación web
Descripción: Se registran los conductores para su uso en las diferentes aplicaciones.
Prerrequisitos: Haber iniciado sesión en la aplicación web.
Pasos: Ingresar los datos del conductor a registrar y seleccionar el botón Registrar conductor .
Resultado esperado: Conductor registrado correctamente.
Resultado obtenido: Registro de conductor exitoso.

Tabla 55. CDP 3: Registro de rutas aplicación web.

Nombre del caso de prueba: Registro de rutas aplicación web
Descripción: Se registran las rutas para su uso en las diferentes aplicaciones.
Prerrequisitos: Haber iniciado sesión en la aplicación web.
Pasos: Ingresar los datos de la ruta a registrar, ingresar los datos de las sub-rutas que contiene la ruta, agregar las ubicaciones de las sub-rutas registradas y seleccionar el botón Registrar ruta .
Resultado esperado: Ruta registrada correctamente.
Resultado obtenido: Registro de ruta exitoso.

Tabla 56. CDP 4: Inicio de sesión aplicación conductor.

Nombre del caso de prueba: Inicio de sesión aplicación conductor
Descripción: Se validan los datos de ingreso del conductor a la aplicación del conductor.
Prerrequisitos: Tener un conductor registrado y tener asignado un usuario y contraseña.
Pasos: Ingresar nombre usuario, ingresar contraseña, seleccionar una ruta

a seguir y seleccionar el botón Ingresar .
Resultado esperado: El conductor tiene acceso a la interfaz de ubicaciones.
Resultado obtenido: Acceso correcto.

Tabla 57. CDP 5: Inicio de sesión aplicación usuario.

Nombre del caso de prueba: Inicio de sesión aplicación usuario
Descripción: Se obtienen los datos del usuario que instala la aplicación.
Prerrequisitos: Tener conexión a internet.
Pasos: Seleccionar cualquier cuenta de Google que se encuentre en el dispositivo Android o crear una nueva cuenta.
Resultado esperado: El usuario tiene acceso a la aplicación FindRoute.
Resultado obtenido: Acceso exitoso.

Tabla 58. CDP 6: Registro de favoritos aplicación usuario.

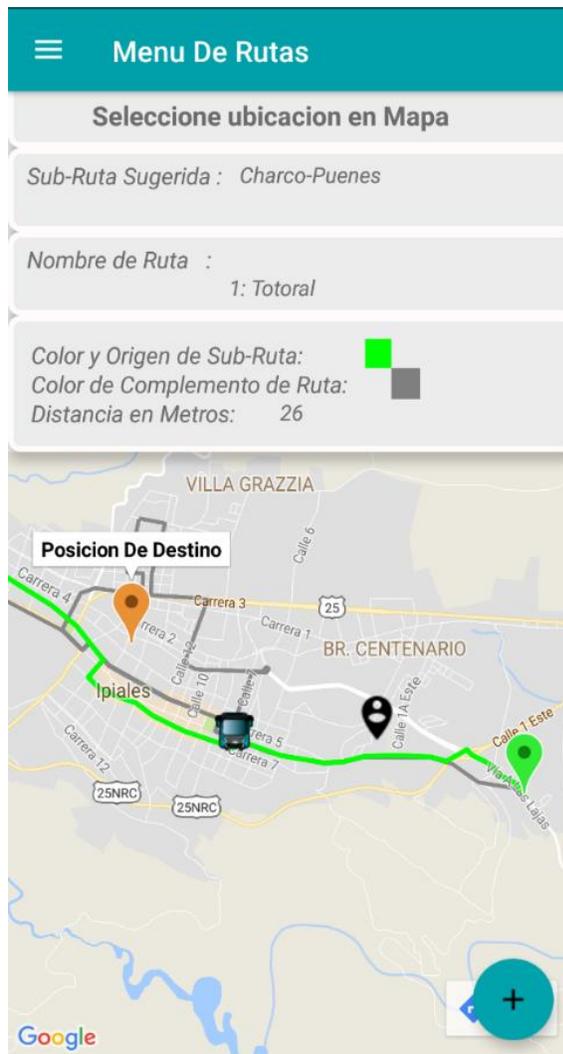
Nombre del caso de prueba: Registro de favoritos aplicación usuario
Descripción: Se agregan o se eliminan rutas favoritas del dispositivo Android.
Prerrequisitos: Tener conexión a internet.
Pasos: Abrir cualquier ruta, elegir una sub-ruta y seleccionar el icono de favoritos en la interfaz del mapa.
Resultado esperado: La ruta se ha agregado a favoritos exitosamente.
Resultado obtenido: Se ha añadido la ruta a favoritos.

Tabla 59. Lista de pruebas funcionales.

Id	Nombre	Cumple	No cumple
CDP 1	Inicio de sesión aplicación web	X	
CDP 2	Registro de conductores aplicación web	X	
CDP 3	Registro de rutas aplicación web	X	
CDP 4	Inicio de sesión aplicación conductor	X	
CDP 5	Inicio de sesión aplicación usuario	X	
CDP 6	Registro de favoritos aplicación usuario	X	

1.4.2 Pruebas funcionales de la sugerencia de ruta. Se probaron diferentes ubicaciones de origen y de destino en la ciudad de Ipiales para demostrar la eficacia de la función de sugerencia de rutas y también se realizó un análisis de resultados obtenidos.

Figura 18. Sugerencia universidad de Nariño centro.



En la figura 21 se puede apreciar que el usuario se encuentra en la universidad de Nariño y ha seleccionado una ubicación de destino en el centro de la ciudad.

En este caso la aplicación sugiere la ruta 1: totoral con la sub-ruta charco Puenes, en esta prueba la ruta-sub-ruta respuesta es la más óptimas para viajar al lugar de destino del usuario.

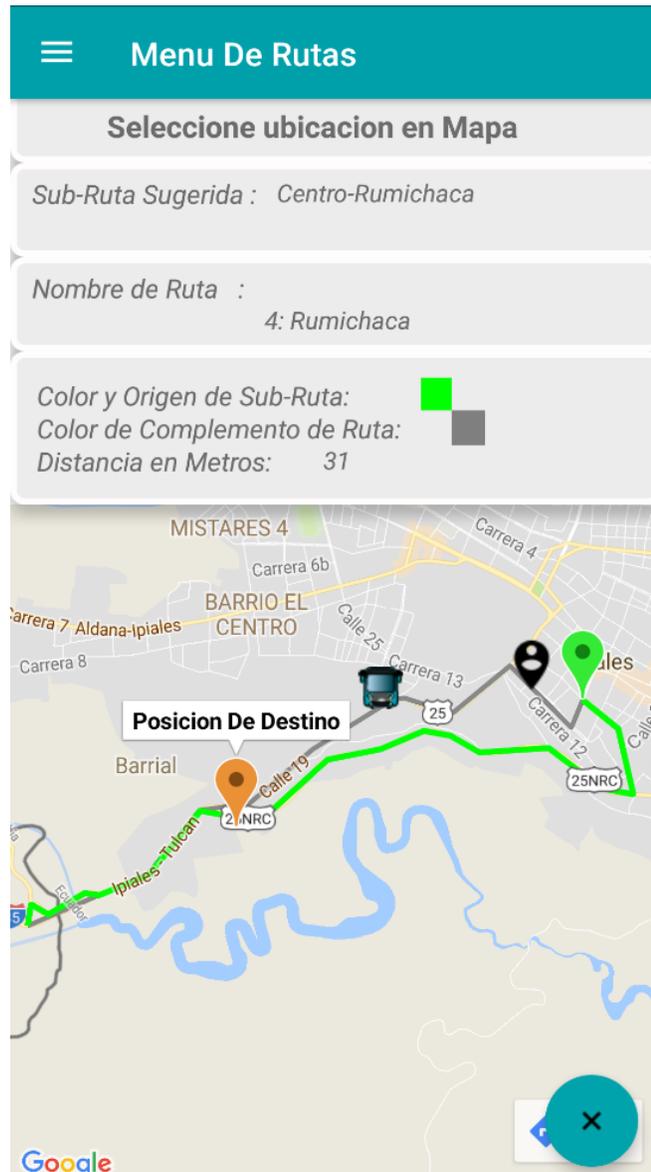
Figura 19. Sugerencia Centro comercial gran plaza centro.



En la figura 22 se puede apreciar que el usuario se encuentra en el centro comercial gran plaza y ha seleccionado una ubicación de destino en el centro de la ciudad.

En este caso la aplicación sugiere la ruta 1: Totoral con la sub-ruta Puenes Totoral, en esta prueba la ruta-sub-ruta respuesta también es la más óptima para viajar al lugar de destino del usuario.

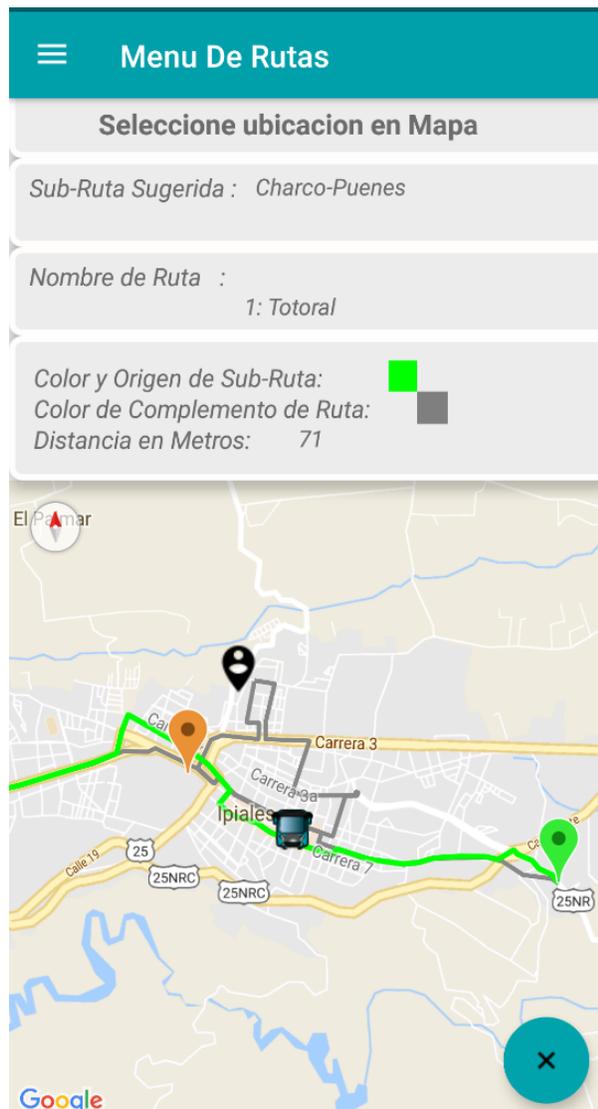
Figura 20. Sugerencia Centro Rumichaca.



En la figura 23 se puede apreciar que el usuario se encuentra en el Centro de la ciudad y ha seleccionado una ubicación de destino en la cercanía a Rumichaca,

En este caso la aplicación sugiere la ruta 4: Rumichaca con la sub-ruta Centro Rumichaca, en esta prueba la ruta-sub-ruta respuesta también es la más óptima para viajar al lugar de destino del usuario

Figura 21. Sugerencia Totoral Batallón de Ipiales.



En la figura 24 se puede apreciar que el usuario se encuentra en el barrio Totoral y ha seleccionado una ubicación de destino cerca al batallón de Ipiales.

En este caso la aplicación sugiere la ruta 1: Totoral con la sub-ruta Charco-Puenes, a pesar de que la sub-ruta sugerida pasa por el lugar de destino del usuario, en este caso se puede observar que no es la ruta-sub-ruta más óptima.

1.4.3 Análisis de resultados de las pruebas realizadas a la función de sugerencia de rutas. Después de las pruebas realizadas se obtienen unos resultados efectivos en la sugerencia mostrando en la mayoría de los casos la mejor ruta posible para el usuario.

Como se puede observar la sugerencia de ruta también está programada para detectar cuáles son las direcciones de la ruta.

Por ejemplo en la figura 22 se puede observar de que a pesar de que la sub-ruta más cercana al usuario y a su destino es la de Rumichaca Centro esta es descartada automáticamente ya que la dirección resultante entre el usuario y el destino es diferente a la dirección de esa sub-ruta, por lo tanto se toma como sugerencia optima la ruta que tenga la misma dirección entre el usuario y el destino como también la ruta-sub-ruta más cercana.

En la sugerencia: Totoral Batallón de Ipiales, figura 24, se puede observar que la ruta seleccionada por la aplicación no es la más óptima pero si es una ruta que llevaría al usuario a su lugar de destino. Al haber tantas ubicaciones factibles en la aplicación es posible que la ruta sugerida no sea la más óptima en todos los casos

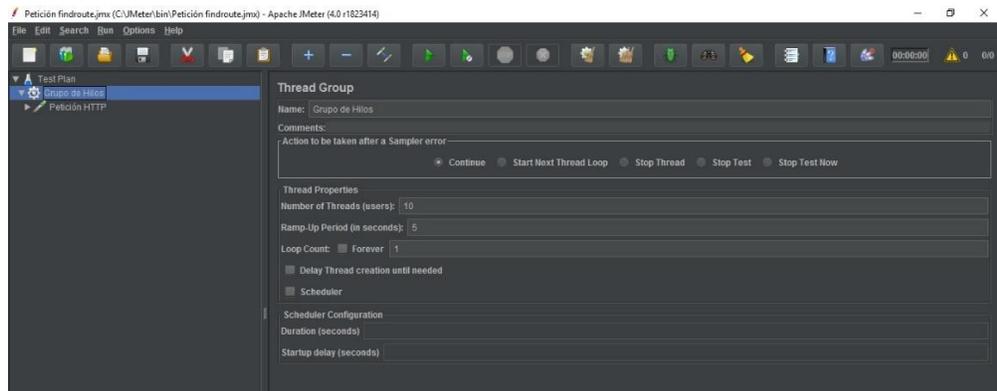
Finalmente el usuario puede optar por no utilizar la ruta sugerida, o también se puede verificar si la ruta sugerida es la más óptima observando el complemento de la ruta o comparando el origen de la ruta con el destino seleccionado.

1.4.4 Pruebas de estrés

Para realizar las pruebas de estrés se utilizó el programa JMeter 4.0 con el cual se puede simular una carga de peticiones de múltiples usuarios utilizando las aplicaciones.

1.4.5 Prueba de estrés 1. 10 usuarios simultáneos

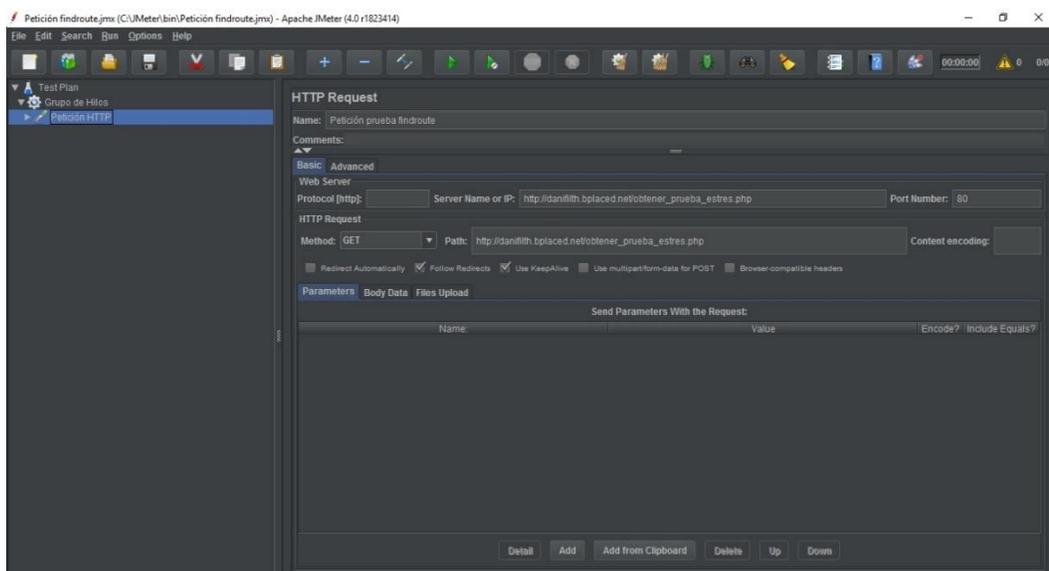
Figura 22. Configuración de peticiones.



En la figura 25 se muestra la configuración de peticiones que se utilizó para esta prueba, también se selecciona el tiempo y las repeticiones.

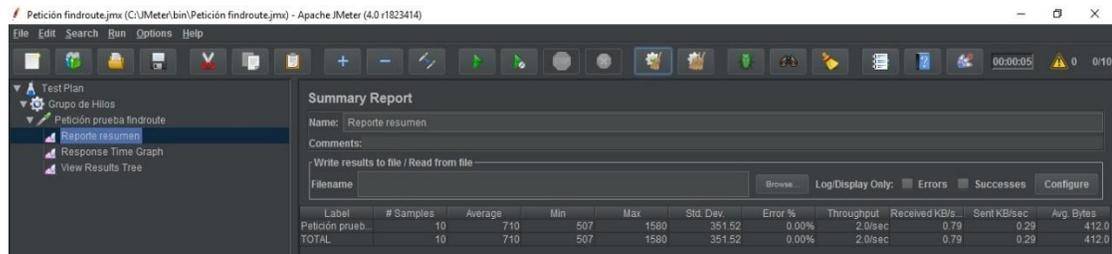
- Number of threads (users): cantidad de usuarios simultáneos que se probaron, en este caso se va a empezar por 10 usuarios.
- Ramp-Up Period (in seconds): tiempo que tardan todos los usuarios en realizar las peticiones, se ha seleccionado 5 segundos lo cual dice que se van a realizar 2 peticiones cada segundo.
- Loop count: es la cantidad de repeticiones que se van a realizar, por ejemplo si se selecciona 2 serían 20 peticiones en 10 segundos, pero en este caso se ha optado por no realizar ninguna repetición.

Figura 23. Configuración de petición HTTP.



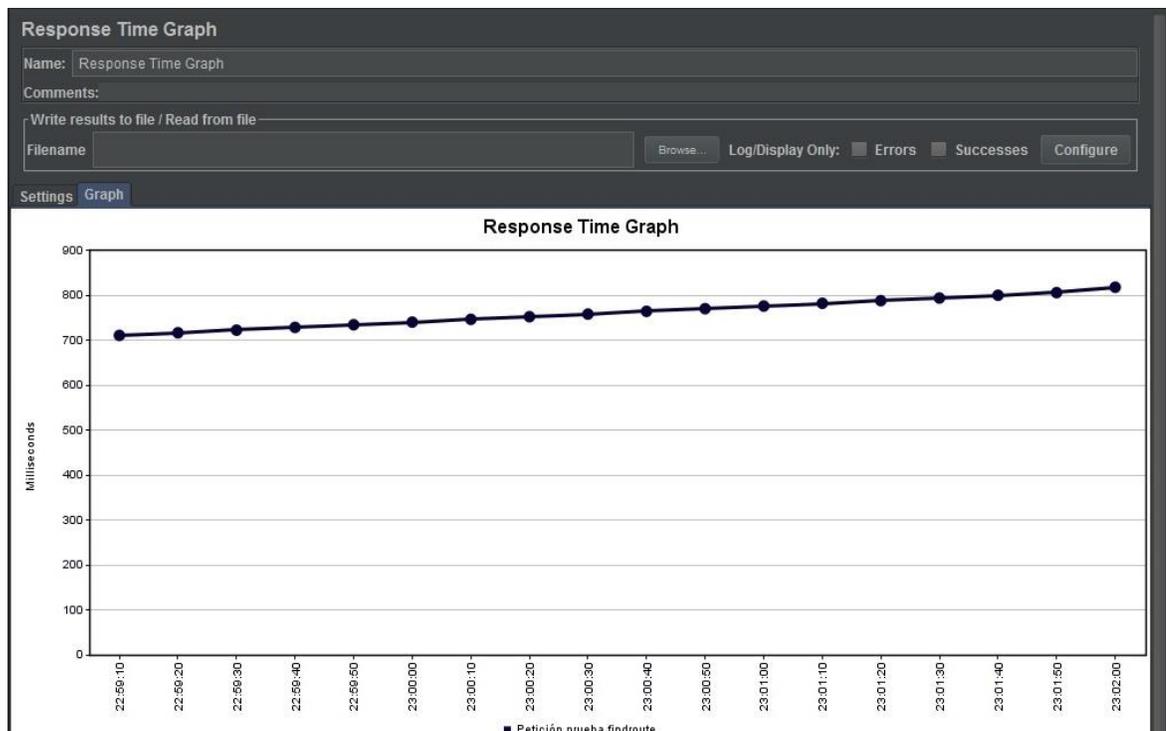
En la figura 26 se muestra la configuración de peticiones HTTP, aquí se ingresó la dirección IP donde se encuentra alojado el servidor y se procede a ejecutar la prueba.

Figura 24. Resultados de prueba 1



En la figura 27 se muestran los resultados obtenidos por la primera prueba, aquí se aprecia cuantas peticiones simultaneas se han realizado, un promedio de respuesta, la desviación estándar, el porcentaje de error y el rendimiento de peticiones sobre segundo.

Figura 25. Grafica de resultados de prueba 1.



En la figura 28 se puede apreciar en grafica la velocidad de respuesta de cada petición en milisegundos.

Figura 26. Resultado de peticiones correctas de prueba 1.

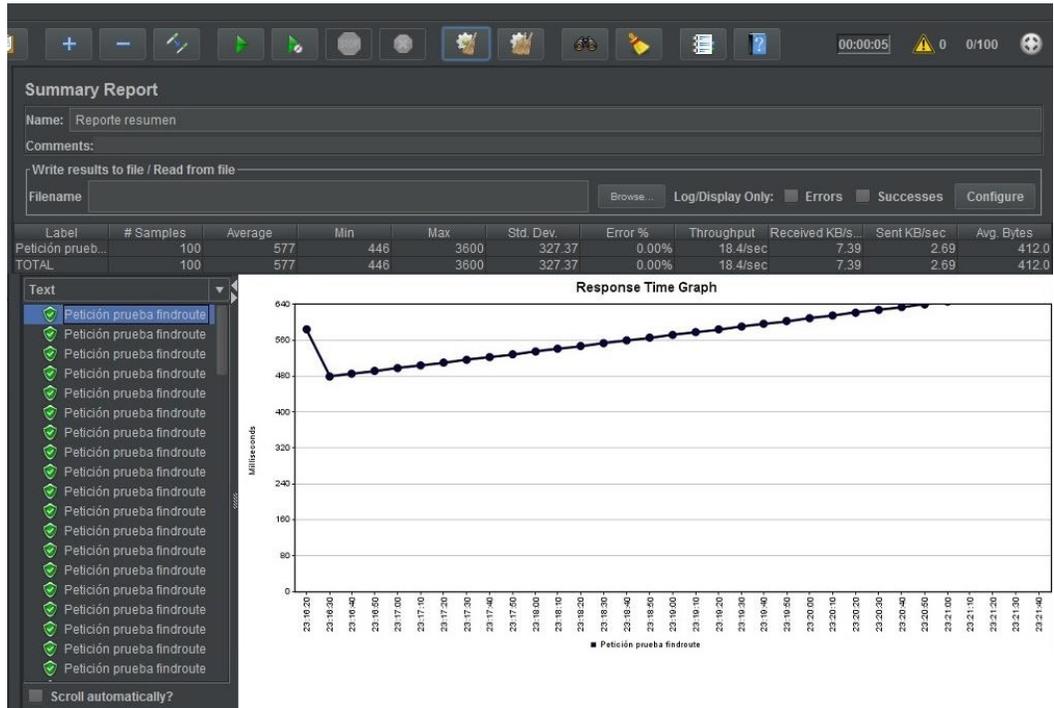


En la figura 29 se puede apreciar cuantas peticiones al servidor han resultado correctas y también la respuesta que reciben las aplicaciones para poder localizar los conductores en el mapa.

Prueba de estrés 1. Análisis de resultados. Según los resultados obtenidos en la primera prueba se apreció que el tiempo de respuesta promedio de cada petición es de 700 milisegundos y que con 10 peticiones no se obtiene ninguna respuesta errónea por parte del servidor, por lo tanto este se encuentra en óptimas condiciones para soportar 10 usuarios simultáneos.

1.4.6 Prueba de estrés 2. 100 usuarios simultáneos

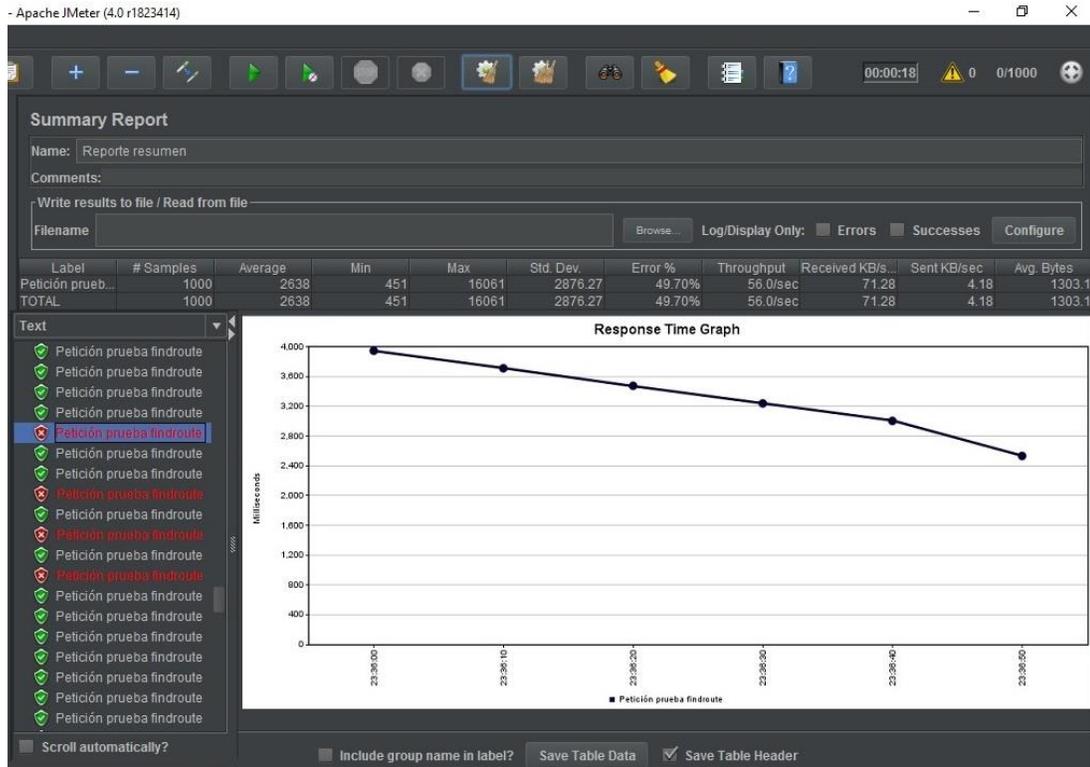
Figura 27. Resultados de prueba 2.



Prueba de estrés 2. Análisis de resultados. Según los resultados obtenidos en la segunda prueba se apreció que el tiempo de respuesta promedio de cada petición es de 577 milisegundos, esto se encuentra cerca del mismo promedio que la primera prueba y que con 100 peticiones no se obtiene ninguna respuesta errónea por parte del servidor, por lo tanto este se encuentra en óptimas condiciones para soportar 100 usuarios simultáneos.

1.4.7 Prueba de estrés 3. 1000 usuarios simultáneos

Figura 28. Resultados de prueba 3.



Prueba de estrés 3. Análisis de resultados. Según la prueba realizada la sobrecarga del servidor está provocando resultados erróneos en algunas peticiones, se puede apreciar que el tiempo de respuesta promedio de cada petición es de 2638 milisegundos algo mucho más tardío que las anteriores pruebas y que con la gran cantidad de peticiones simultaneas se obtiene un 49.70% de respuestas erróneas, pero las aplicaciones están programadas para actualizar la ubicación de los conductores cada 6 segundos por lo tanto si se recibe una respuesta errónea simplemente la ubicación de los conductores se actualizara en la siguiente petición.

Prueba de estrés análisis de resultados global. Las pruebas realizadas demuestran que se han obtenido unas estadísticas favorables en las tres pruebas de estrés ya que la gran concurrencia de usuarios simultáneos no afectaría la experiencia del usuario final.

2. RESULTADOS OBTENIDOS

Las aplicaciones desarrolladas brindaron a la comunidad de Ipiales una alternativa útil y necesaria a las personas que utilizan el transporte público, especialmente para extranjeros, padres de familia, empleados entre otros los cuales no conocen las rutas que siguen los buses, posición del bus en cierto momento y distancia desde su posición a la ruta sugerida.

Después de haber realizado el análisis, selección de tecnologías a utilizar para llevar a cabo el proyecto, la metodología a seguir, se logró el desarrollo de dos aplicaciones móviles, una para el conductor el cual selecciona la ruta que va a seguir en cada día, una aplicación para el usuario el cual puede observar las rutas, su posición actual, la posición del conductor y recibir una sugerencia de la ruta más favorable para llegar a su lugar de destino. El desarrollo de una aplicación web para que el administrador el cual puede ingresar, modificar, eliminar, listar la posición de los conductores y las rutas que siguen los buses.

ANÁLISIS DE RESULTADOS DE ENCUESTA PRUEBA DE USABILIDAD FINDROUTE

Análisis de encuesta figura 54 de anexos: En los resultados obtenidos, se detalló que las personas encuestadas tienen facilidad para ingresar a la aplicación, por lo tanto el inicio a la aplicación no tiene grandes problemas para mejoras futuras.

Análisis de encuesta figura 55 de anexos: En los resultados obtenidos, se detalló que la gran mayoría de personas encuestadas no tienen dificultada para seleccionar y observar una ruta en el mapa, por lo tanto la interfaz en la que se realizan estos procesos tiene facilidad para ser entendida.

Análisis de encuesta figura 56 de anexos: Igual que en la pregunta anterior, los resultados obtenidos, detallaron que la gran mayoría de personas encuestadas no tienen dificultada para seleccionar y observar una sub-ruta en el mapa, pero existe un grupo de personas que no entiende como observar las sub-rutas debido a que están en otra pestaña, por lo tanto las interfaces en las que se realizan estos procesos tienen algunas complicaciones mínimas que fácilmente pueden ser atendidas.

Análisis de encuesta figura 57 de anexos: Los resultados obtenidos, detallaron que la gran mayoría de personas encuestadas pueden visualizar los datos y posición de buses de una manera fácil, aunque existen un porcentaje que tiene dudas al momento de interpretar los datos de los buses.

Análisis de encuesta figura 58 de anexos: Los resultados obtenidos, indicaron que un porcentaje de personas encuestadas guardan las sub-rutas fácilmente, pero faltan cosas por mejorar para que los niveles de facilidad aumenten.

Análisis de encuesta figura 59 de anexos: Los resultados obtenidos indicaron que la gran mayoría de persona les pareció muy fácil la manera en que recibieron la sugerencia, pero existen personas a las cuales no entendieron con claridad la sugerencia de las rutas. Por tal motivo existen procesos para mejorar en un futuro.

Análisis de encuesta figura 60 de anexos: La gráfica indica que la gran mayoría de encuestados se les facilitó seleccionar la ubicación destino por medio de un marcador, pero a la vez existen encuestados que no pueden seleccionar precisamente el lugar de destino, cosas por mejorar en un futuro.

Análisis de encuesta figura 61 de anexos: La gráfica muestra datos que indican que fue fácil interactuar por las opciones o ítems que posee la aplicación.

Análisis de encuesta figura 62 de anexos: La gráfica indica que la gran mayoría de encuestados se les facilitó utilizar la aplicación para recibir una sugerencia de rutas en el municipio de Ipiales.

Los resultados obtenidos al realizar la encuesta, demuestran un alto nivel de satisfacción por parte de los usuarios al utilizar la aplicación FindRoute, además indican que fue de gran utilidad y las sugerencias por parte de ellos sirven para mejoras, las cuales permitieron que la aplicación se vaya perfeccionando con el pasar del tiempo.

2.1 Requerimientos de funcionamiento

2.1.1 Usuario. Las aplicaciones van dirigidas a personas con las capacidades necesarias e interés para manejar un dispositivo inteligente, ya sean Tablet o teléfonos inteligentes que cuenten con sensor GPS preinstalado y acceso a internet para algunas de las funciones; las aplicaciones tienen un diseño de interfaz amigable e interactivo, para que el usuario pueda manejar las aplicaciones de una manera fácil y rápida.

Requerimientos de software. Para la creación de las aplicaciones móviles fue necesario la instalación de Android Studio 2.3.3 el cual es el entorno de desarrollo para la plataforma del sistema operativo Android, para la base de datos se utilizó PostgreSQL

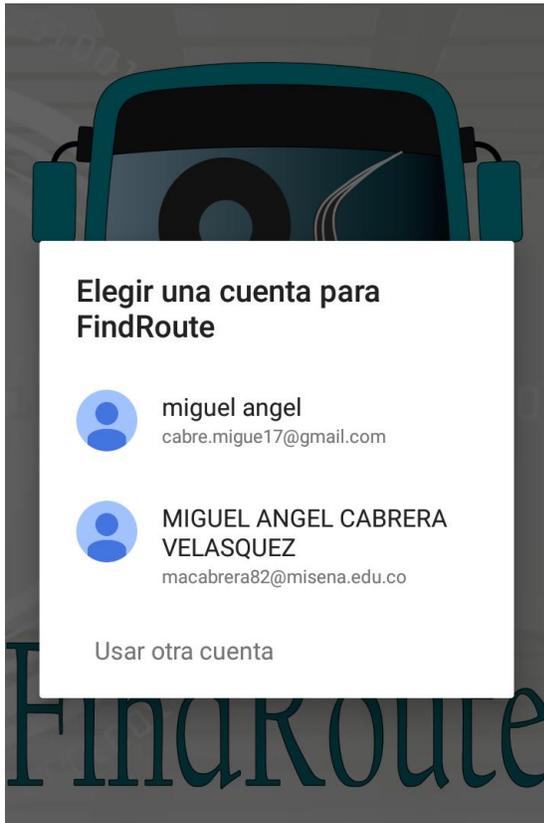
La aplicación web fue desarrollada en php, el framework bootstrap y la tecnología Ajax.

2.1.2 Requerimientos de hardware. Para la instalación de las Aplicaciones móviles se requiere de un dispositivo inteligente, ya sea Tablet, celular con sistema operativo Android en su versión mínima de 4.0 y un espacio disponible de memoria interna de 30 megabytes y memoria ram de 1 Gigabyte.

2.2 Descripción de funcionamiento de aplicación FindRoute

2.2.1 Interfaz de inicio de sesión

Figura 29. Interfaz de inicio de sesión.



En la figura 32 se muestra la interfaz donde se pide seleccionar un correo de cuenta de google para ingresar a la aplicación o usar otra cuenta de las que se muestran en el cuadro de mensajes.

2.2.2 Interfaz dar permisos

Figura 30. Interfaz de petición de permisos.



En la figura 33 permite a la aplicación acceda a funciones del dispositivo inteligente y pueda funcionar correctamente.

2.2.3 Interfaz de selección de rutas o sugerencias

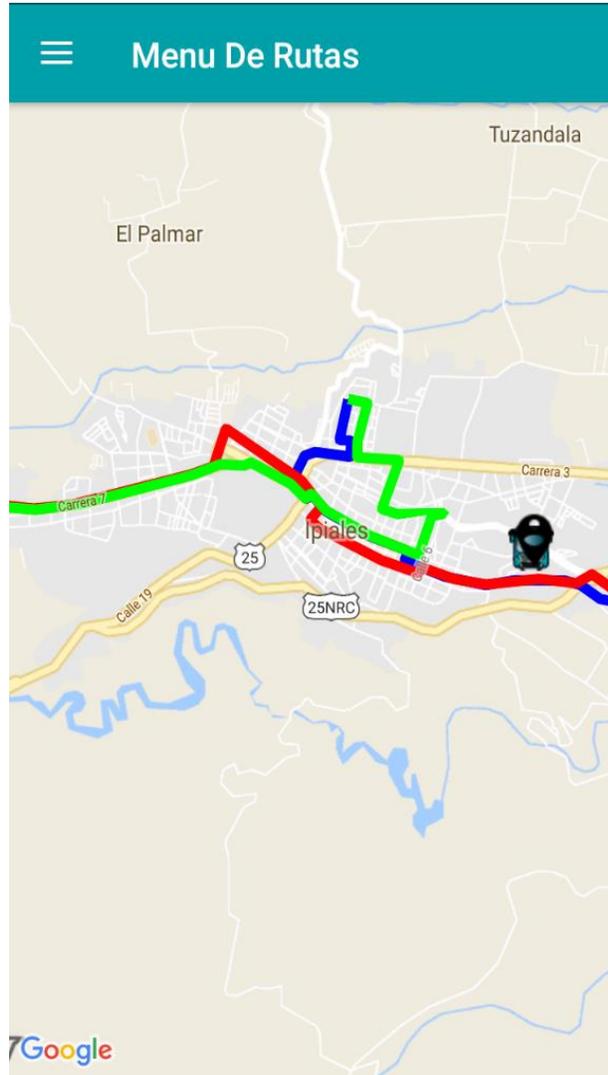
Figura 31. Interfaz de selección de rutas o sugerencias.



En la figura 34 se muestra una lista de rutas que posteriormente se pueden observar en el mapa o un botón de sugerencia para ir directamente a la pestaña y recibir una sugerencia de la posible ruta a seguir para llegar al lugar destino.

2.2.4 Interfaz para observar la ruta seleccionada

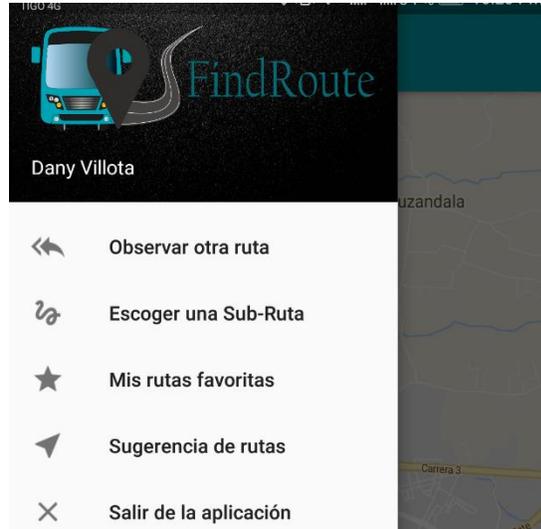
Figura 32. Interfaz para observar la ruta seleccionada.



En la figura 35 se muestra la ruta completa seleccionada, donde se grafica con diferentes colores las sub-rutas, además los buses que siguen dicha ruta y la ubicación del usuario.

2.2.5 Interfaz de ítems de aplicación

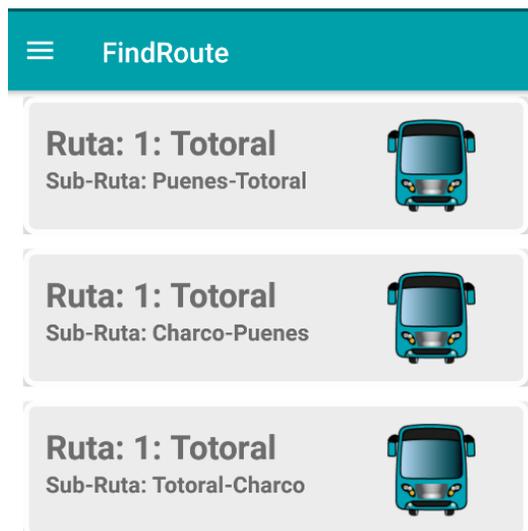
Figura 33. Interfaz de ítems de aplicación.



En la figura 36 se muestran las opciones que tiene la aplicación donde están observar otra ruta, escoger una sub-ruta, mis rutas favoritas, sugerencia de rutas y salir de la aplicación.

2.2.6 Interfaz de selección de sub-ruta

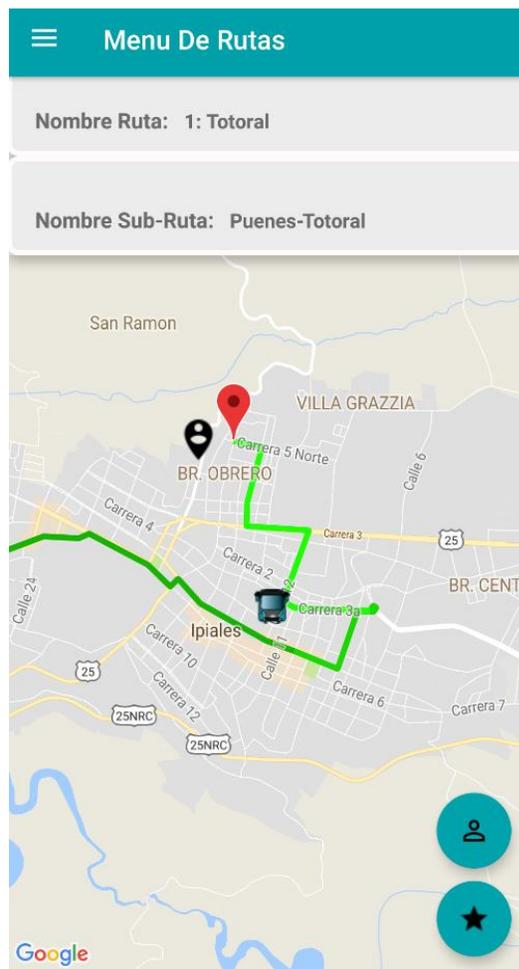
Figura 34. Interfaz de selección de sub-ruta.



En la figura 37 Al seleccionar una ruta completa se listan las sub-rutas que la componen y al escoger una sub-ruta se dirige al mapa, mostrando su ubicación y los buses que están recorriendo dicha ruta.

2.2.7 Interfaz para visualizar sub-ruta en mapa

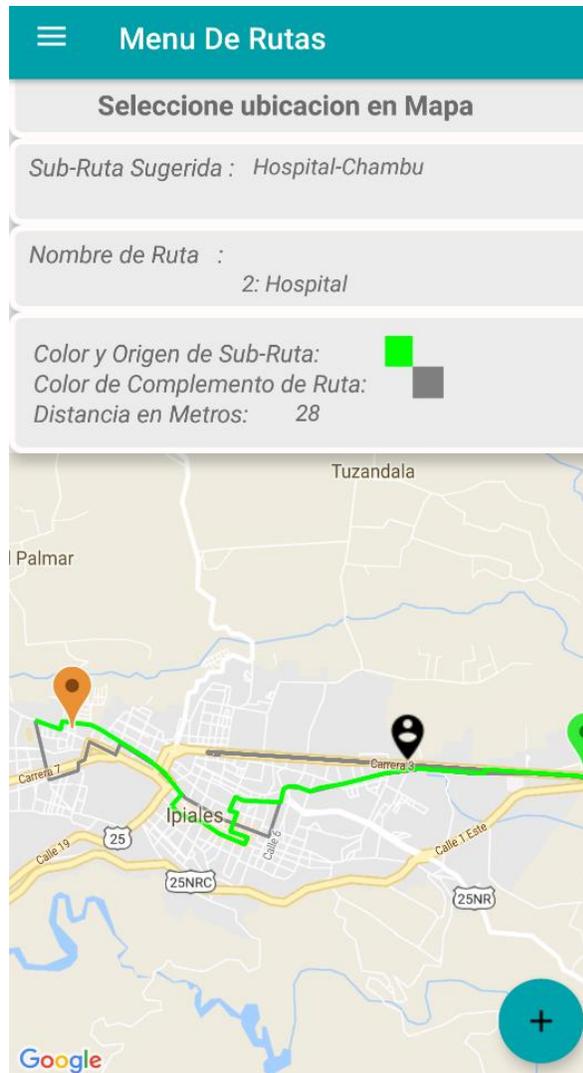
Figura 35. Interfaz para visualizar sub-rutas en el mapa.



En la figura 38 se muestra la sub-ruta seleccionada con colores y marcadores para indicar el origen y final de la sub-ruta, además se muestra un icono para agregar la sub-rutas a favoritos y un icono el cual muestra una ventana de información.

2.2.8 Interfaz de sugerencia

Figura 36. Interfaz de sugerencia de rutas.



En la figura 39 se muestra un mapa donde se ubica un marcador con la ubicación destino del usuario y según eso se grafica una ruta conveniente para llegar al lugar marcado y se muestra el nombre de la ruta y sub-ruta graficada.

En la parte de abajo aparece un icono el cual muestra la distancia promedio de la ubicación del usuario a la ruta sugerida.

2.3 Descripción de funcionamiento de aplicación FindRoute Conductor

2.3.1 Interfaz inicio de sesión

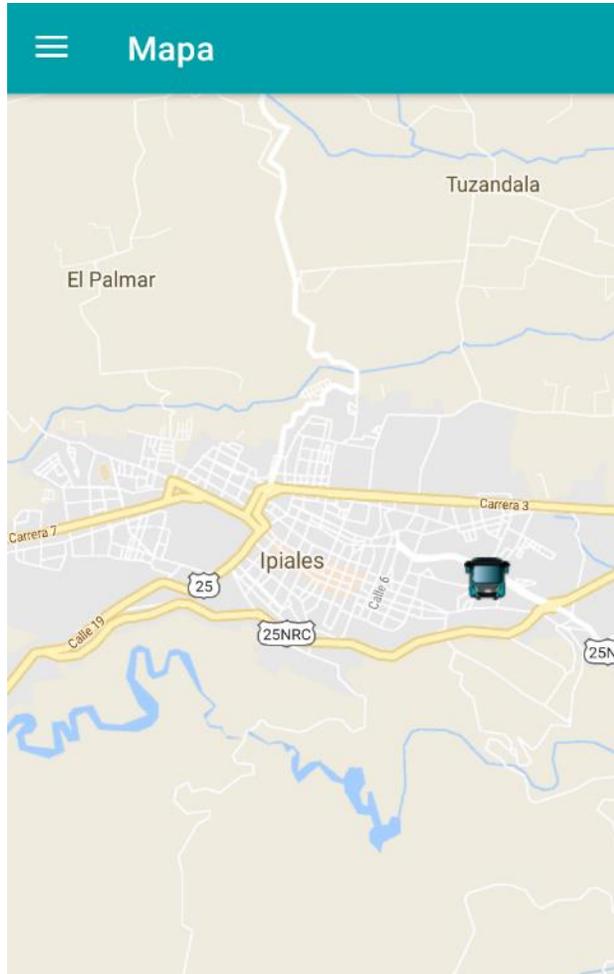
Figura 37. Interfaz de inicio de sesión.



En la figura 40 se debe ingresar el usuario, el cual es asignado por el administrador y seleccionar una ruta a seguir.

2.3.2 Interfaz visualizar ubicación conductor en el mapa

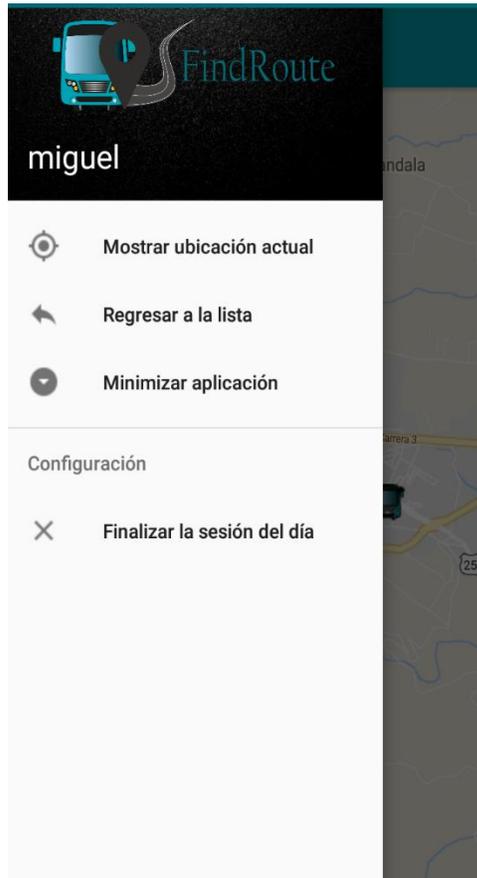
Figura 38. Interfaz visualizar ubicación conductor en el mapa.



En la figura 41 se puede observar la ubicación del conductor.

2.3.3 Interfaz de ítems de menú

Figura 39. Interfaz de ítems de menú.



En la figura 42 se puede observar las opciones que tiene la aplicación del conductor.

Donde una opción es para observar la ubicación actual del conductor, otra opción es para seleccionar otra ruta, otra opción es minimizar la aplicación y seguir realizando otras funciones del dispositivo inteligente mientras sigue enviando coordenadas al servidor y la última opción es para cerrar sesión y finalizar el día.

2.4 Descripción de funcionamiento de aplicación FindRoute Administrador

2.4.1 Interfaz inicio de sesión

Figura 40. Inicio de sesión.



En la figura 43 se muestra una ventana para iniciar sesión e ingresar a la aplicación.

2.4.2 Interfaz principal

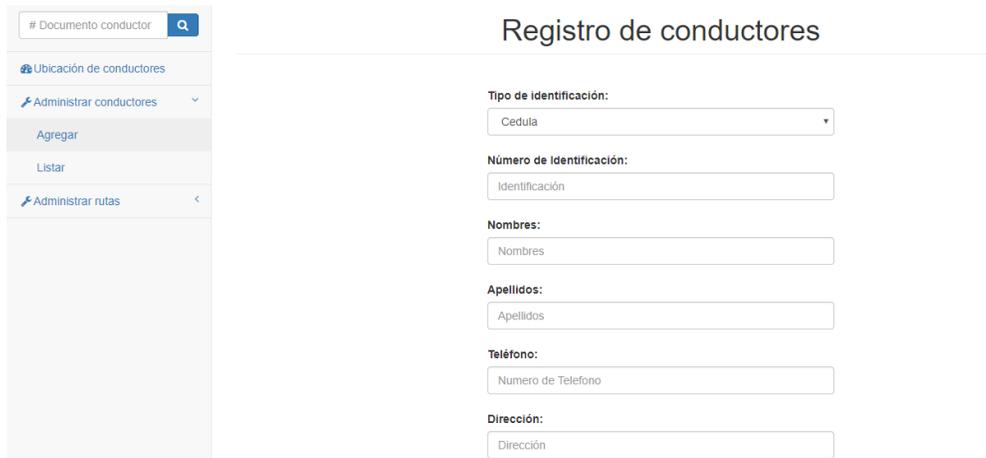
Figura 41. Interfaz principal



En la figura 44 se muestra un mapa con la ubicación de los buses que están trabajando en el momento, se actualiza cada 6 segundos. En la parte izquierda aparece un menú para actualizar mapa de conductores, administrar datos de conductor y administrar datos de las rutas. En la parte derecha aparece el usuario para cerrar sesión.

2.4.3 Interfaz Administrar conductores

Figura 42. Interfaz conductores.



Documento conductor

Ubicación de conductores

Administrar conductores

Agregar

Listar

Administrar rutas

Registro de conductores

Tipo de identificación:

Número de identificación:

Nombres:

Apellidos:

Teléfono:

Dirección:

En la figura 45 se muestra la opción de agregar un conductor, ingresando sus datos básicos en las casillas.

2.4.4 Interfaz listar conductores

Figura 43. Interfaz listar conductores.



FindRoute Bienvenido dany

Documento conductor

Ubicación de conductores

Administrar conductores

Agregar

Listar

Administrar rutas

Lista de conductores

Numero de identificación	Nombres	Apellidos	Teléfono	Dirección	Placas del bus	Estado	Mostrar ubicación en el mapa	Modificar	Eliminar
1234321	Jesus	Torres	2212321	calle 11	JOP123	Detenido			
121224454545	Miguel Angel	Cabrera Velasquez	3019001	saguaran	cdfg56t	Detenido			
132456	Pepe	Burbano	123	calle 12	AHW-125	Detenido			
123431	Dany	Villota	137478	calle 18 4N-55	KHM-123	En movimeinto		El conductor debe estar detenido	El conductor debe estar detenido
10972721	Juan	Guerrero	7728121	caller12	ALCA-15	Detenido			

En la figura 46 se muestra una lista de condutores con sus datos personales, además se observa si están recorriendo una ruta o están quietos, si aparece un icono al presionarlo se dirige a una ventana con la ubicación del conductor. Igualmente se muestran dos botones para eliminar y modificar los conductores.

2.4.5 Interfaz modificar datos conductor

Figura 44. Interfaz modificar datos conductor.

Documento conductor

Ubicación de conductores

Administrar conductores

Agregar

Listar

Administrar rutas

Modificar conductores

Tipo de identificación:

Número de identificación:

Nombres:

Apellidos:

Teléfono:

Dirección:

En la figura 47 se muestra un formulario para modificar los datos del conductor.

2.4.6 Interfaz buscar conductores

Figura 45. Interfaz buscar conductores.

FindRoute Bienvenido d

Documento conductor

Ubicación de conductores

Administrar conductores

Administrar rutas

Lista de conductores

Numero de identificación	Nombres	Apellidos	Teléfono	Dirección	Placas del bus	Estado	Mostrar ubicación en el mapa	Modificar	Eliminar
1234321	Jesus	Torres	2212321	calle 11	JOP123	Detenido	EL conductor debe estar en movimiento		

En la figura 48 se muestra la interfaz para buscar un conductor ingresando el número de identificación y si existe algún conductor con la identificación ingresada se muestra en pantalla los datos del conductor.

2.4.7 Interfaz administrar rutas

Figura 46. Interfaz administrar rutas.

FindRoute

Documento conductor

Ubicación de conductores

Administrar conductores

Administrar rutas

Agregar

Listar

Registro de rutas

Ingrese el nombre de la ruta:

Nombre de ruta

Seleccione el número de sub-rutas:

1

Continuar

En la figura 49 se muestra un formulario para ingresar nuevas rutas, ingresando el nombre y el número de sub-rutas y presionando el botón continuar se dirige hacia otro formulario.

2.4.8 Interfaz graficar sub-ruta

Figura 47. Interfaz graficar sub-ruta.

Centrar mapa en:

Mapa Satélite

Latitud: 0.82870

Longitud: -77.63798

Lista de ubicaciones

- (0.82492,-77.63764)
- (0.82488,-77.63708)
- (0.82565,-77.63669)
- (0.82694,-77.63614)
- (0.82750,-77.63742)
- (0.82788,-77.63841)
- (0.82870,-77.63798)

Ingrese descripción de la sub-ruta 1:

Descripción subruta

En la figura 50 se muestra la interfaz para graficar la sub-ruta, ingresar nombre de sub-ruta y guardar la sub-ruta para continuar con otra según el número de sub-rutas ingresado.

2.4.9 Interfaz listar rutas

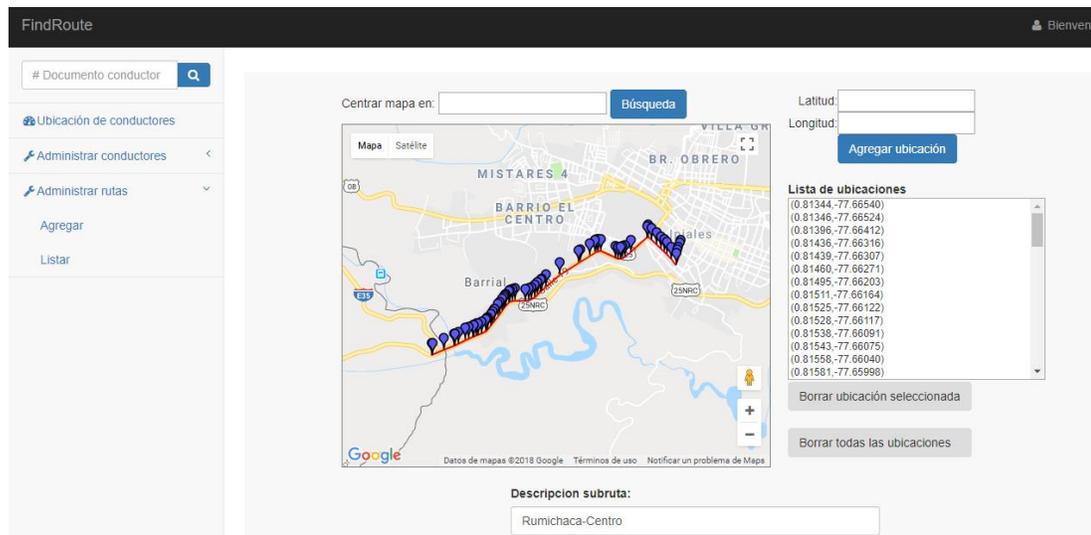
Figura 48. Interfaz listar rutas.



En la figura 51 se muestra una lista de rutas ingresadas y opciones para más información, modificar y eliminar sub-rutas.

2.4.10 Interfaz modificar sub-ruta

Figura 49. Interfaz modificar sub-ruta.



En la figura 52 se muestra un interfaz para modificar los datos de la sub-ruta y guardar la nueva sub-ruta.

3. CONCLUSIONES

La aplicación FindRoute planteó una opción de búsqueda y selección de rutas que siguen los buses de una manera diferente a los métodos tradicionales, obteniendo información detallada y fácil de entender.

El uso de las aplicaciones fue de gran utilidad para las personas de Ipiales como también para visitantes extranjeros o de otros lugares del país en el momento de que deseen utilizar el sistema de transporte público de la ciudad.

El desarrollo de la aplicación FindRoute ha sido enfocado hacia el municipio de Ipiales, sin embargo es posible adaptarla fácilmente con el fin de operar en el sistema de transporte público de cualquier ciudad de Colombia.

FindRoute utilizó una base de datos interna la cual, permitió el acceso a la mayoría de funciones de la aplicación de usuario, sin necesidad de conexión a internet.

4. RECOMENDACIONES

Analizar los estándares de cada empresa para la selección de los datos de usuarios y conductores.

Al graficar las rutas, Los puntos seleccionados deben estar cercanos entre si ya que estos harán que la sugerencia sea lo más efectiva posible.

Utilizar una base de datos diferente por cada empresa de transporte público donde se implemente la aplicación con el fin de tener más orden con los usuarios registrados y evitar sobrecarga del servidor.

Dar indicaciones pertinentes a los conductores sobre el manejo de la aplicación al momento en que estos se encuentren activos en la ruta.

Modificar la ubicación donde se centran las aplicaciones según la ciudad donde esta sea implementada.

Se podría incluir trabajos futuros como una ayuda, donde el conductor pueda informar cambios en los recorridos, fallas técnicas del bus o aplicación móvil.

Tener conexión a internet para el rastreo de buses en la aplicación y para la búsqueda de la ubicación es recomendable tener el dispositivo móvil en un área despejada.

En caso de ser implementadas las aplicaciones, el transporte público sufrirá cambios en su mayoría favorables, incrementando el uso de los buses por parte de los usuarios, y obteniendo un monitoreo de la ubicación e información de los conductores por parte de los administradores de las empresas de transporte público.

BIBLIOGRAFIA

- bplaced.net. (s.f.). *www.bplaced.net*. Obtenido de <https://www.bplaced.net/>
- Caballero, J. G. (s.f.). *¿Qué es JavaScript?* Obtenido de <https://devcode.la/blog/que-es-javascript/>
- Center, I. K. (s.f.). *¿Qué es Ajax?* Obtenido de https://www.ibm.com/support/knowledgecenter/es/SS8PJ7_9.1.2/com.ibm.e.tools.webtoolscore.doc/topics/cajax.html
- Developers, A. (2018). *Conoce Android Studio*. Obtenido de <https://developer.android.com/studio/intro/?hl=es-419>
- Developers, A. (s.f.). *Android NDK Native API*. Obtenido de https://developer.android.com/ndk/guides/stable_apis.html?hl=es-419
- Developers, A. (s.f.). *Arquitectura de la plataforma Android*. Obtenido de <https://developer.android.com/guide/platform/?hl=es-419>
- Developers, A. (s.f.). *Funciones de la biblioteca de compatibilidad*. Obtenido de <https://developer.android.com/topic/libraries/support-library/features.html>
- DÍAZ, C. (07 de 05 de 2016). *La primavera económica de Ipiales Nariño*. Obtenido de <https://www.dinero.com/opinion/columnistas/articulo/la-primavera-economica-de-ipiales-narino-por-camilo-diaz/223428>
- DÍAZ, C. (07 de 05 de 2016). *www.dinero.com*. Obtenido de [www.dinero.com: https://www.dinero.com/opinion/columnistas/articulo/la-primavera-economica-de-ipiales-narino-por-camilo-diaz/223428](https://www.dinero.com/opinion/columnistas/articulo/la-primavera-economica-de-ipiales-narino-por-camilo-diaz/223428)
- Dinero.com. (15 de 01 de 2018). *Por cada colombiano hay 1,2 líneas de telefonía móvil*. Obtenido de <https://www.dinero.com/pais/articulo/boletin-trimestral-de-las-tic-a-septiembre-de-2017/254148>
- EcuRed. (s.f.). *Lenguaje de programación*. Obtenido de https://www.ecured.cu/Lenguaje_de_Programaci%C3%B3n
- EcuRed. (s.f.). *Servidor Web*. Obtenido de http://www.ecured.cu/Servidor_web
- García, D. (13 de 11 de 2017). *Android 6.0 Marshmallow sigue al frente en la última foto de la fragmentación de Android*. Obtenido de <https://andro4all.com/2017/11/distribucion-android-noviembre-2017>
- gartner. (2017). *gartner.com*. Obtenido de <https://www.gartner.com/en>

- inder123. (24 de 11 de 2015). *Gson User Guide*. Obtenido de <https://github.com/google/gson/blob/master/UserGuide.md>
- Ipiales, R. (03 de 09 de 2016). *Supertaxis del Sur inicia a operar en rutas urbanas de Ipiales*. Obtenido de <http://www.radioipiales.co/2016/03/supertaxis-del-sur-inicia-a-operar-en-rutas-urbanas-de-ipiales/>
- Neosoft. (08 de 01 de 2018). *¿Qué es una aplicación Web?* Obtenido de <https://www.neosoft.es/blog/que-es-una-aplicacion-web/>
- ORACLE. (s.f.). *¿Qué es la tecnología java y para que la necesito?* Obtenido de https://www.java.com/es/download/faq/whatis_java.xml
- Pulzo.com, R. d. (s.f.). *Uso de los smartphones en Colombia ya es mayor al 50% de la población, según Asomóvil*. Obtenido de <http://www.asomovil.org/uso-de-los-smartphones-en-colombia-ya-es-mayor-al-50-de-la-poblacion-segun-asomovil/>
- RADIO, C. (22 de 01 de 2016). *Aumentó un 70% la delincuencia en Ipiales*. Obtenido de http://caracol.com.co/emisora/2016/01/22/pasto/1453466289_703507.html
- Randstad. (17 de 05 de 2017). *qué es la metodología ágil, y por qué es tan popular en ti*. Obtenido de https://www.randstad.cl/tendencias360/archivo/que-es-la-metodologia-agil-y-por-que-es-tan-popular-en-ti_1463/
- Robledano, A. (27 de 05 de 2015). *Tutorial Bootstrap 3: Introducción e instalación*. Obtenido de <https://openwebinars.net/blog/tutorial-bootstrap-3-introduccion-e-instalacion/>
- Rouse, M. (s.f.). *Base de datos relacional*. Obtenido de <https://searchdatacenter.techtarget.com/es/definicion/Base-de-datos-relacional>
- Rouse, M. (s.f.). *Servidor Web*. Obtenido de <https://searchdatacenter.techtarget.com/es/definicion/Servidor-Web>
- The PHP Group. (2001). *¿Qué es PHP?* Obtenido de <http://php.net/manual/es/intro-whatism.php>
- Ulmeher, J. M. (s.f.). *Android, ¿qué es y cómo funciona?* Obtenido de <https://www.ibertronica.es/blog/tutoriales/android-sistema-operativo/>
- W3C. (s.f.). *W3C*. Obtenido de <https://www.w3.org/standards/xml/core>

ANEXOS

Anexo A. Encuesta prueba de usabilidad: Findroute

Se realizó una serie de preguntas a un grupo de 50 personas, entre ellas estudiantes, trabajadores, padres de familia del municipio de Ipiales. La encuesta consta de 9 preguntas de selección y una pregunta abierta, con las cuales se determinó el fácil uso de las aplicaciones, la acogida que tendrán las aplicaciones en caso de ser implementadas por parte de la comunidad.

La encuesta fue desarrollada con ayuda de los formatos de encuestas de google y fue distribuida a los correos de personas conocidas, las cuales muy amablemente llenaron los formatos y enviaron las respuestas, con ayuda de google se generan los gráficos de las respuestas para posteriormente analizar los datos obtenidos.

Las siguientes preguntas deberán ser respondidas de acuerdo a la siguiente escala de valoración.

Figura 50. Escala de valoración.

Muy difícil	Difícil	Fácil	Muy fácil
1	2	4	5

1. ¿Cómo fue para usted el ingreso a la aplicación?_____
2. ¿Cómo fue el proceso de selección y observación de la ruta completa en el mapa?_____
3. ¿Cómo fue el proceso de selección y observación de las sub-rutas en el mapa?_____
4. ¿Cómo fue para usted la manera de visualizar la posición y datos de los buses en el mapa?_____
5. ¿Cómo fue el proceso de guardar sub-rutas a favoritos ?_____
6. ¿Qué tan fácil es comprender el proceso de sugerencia de rutas que brinda la aplicación?_____

7. ¿Qué le pareció, la manera de escoger la dirección destino en la sugerencia de ruta por medio de un marcador en el mapa?_____
8. ¿Cómo fue la navegación por los diferentes ítems de la aplicación FindRoute?_____
9. En general, ¿cómo considera el uso de la aplicación móvil?_____
10. ¿Tiene alguna sugerencia para mejoras de la aplicación móvil?
 Sí_____ No_____

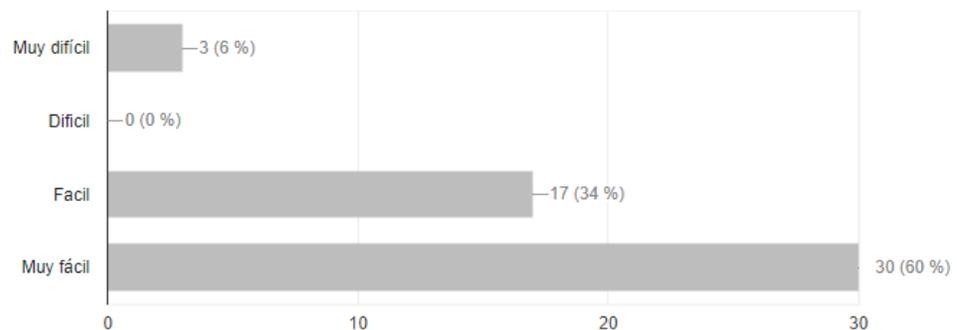
Si la respuesta es sí, ¿Cuál?

Encuesta pregunta número 1

Figura 51. Encuesta pregunta número 1.

1. ¿Cómo fue para usted el ingreso a la aplicación?

0 de 50 respuestas correctas



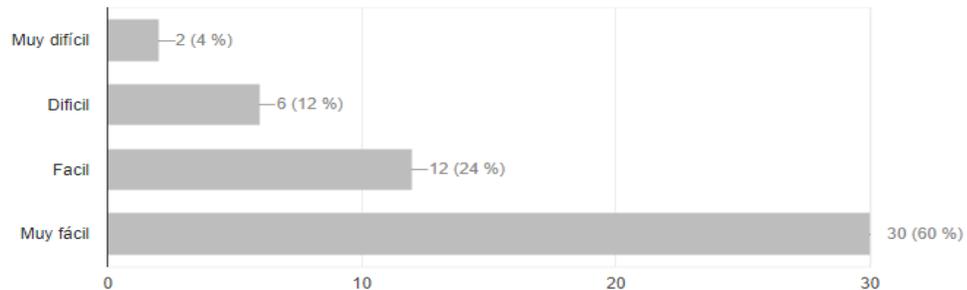
Descripción: En la figura 54 muestra los porcentajes de facilidad del ingreso a la aplicación, donde el 60% de personas encuestadas señala que es muy fácil ingresar a la aplicación, el 34% señala que es fácil ingresar a la aplicación, el 6% señala que es muy difícil y el 0% que es difícil ingresar a la aplicación.

Encuesta pregunta número 2

Figura 52. Encuesta pregunta número 2.

2. ¿Cómo fue el proceso de selección y observación de la ruta completa en el mapa? 

0 de 50 respuestas correctas



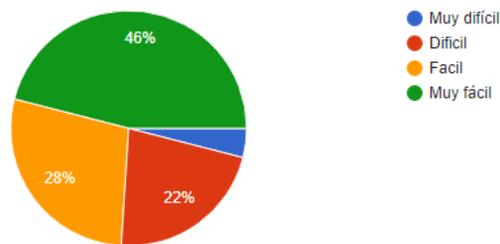
Descripción: En la figura 55 muestra los porcentajes de facilidad del proceso de selección y observación de la ruta en el mapa, según las respuestas de los encuestados, donde el 60% de personas encuestadas indica que es muy fácil, el 24% señala que es fácil, el 12% que es difícil y el 4% que es muy difícil.

Encuesta pregunta número 3

Figura 53. Encuesta pregunta número 3.

3. ¿Cómo fue el proceso de selección y observación de las subrutas en el mapa?

50 respuestas



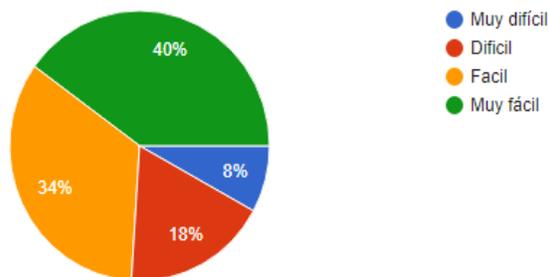
Descripción: En la figura 56 muestra los porcentajes de facilidad del proceso de selección y observación de las sub-rutas en el mapa, según las respuestas de los encuestados, donde el 46% de personas encuestadas indica que es muy fácil, el 28% señala que es fácil, el 22% que es difícil y un porcentaje mínimo señala que es muy difícil.

Encuesta pregunta número 4

Figura 54. Encuesta pregunta número 4.

4. ¿Cómo fue para usted la manera de visualizar la posición y datos de los buses en el mapa?

50 respuestas



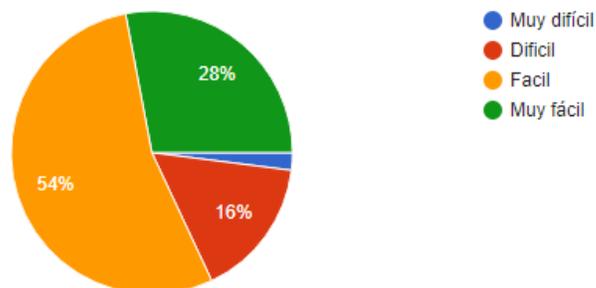
Descripción: En la figura 57 muestra los porcentajes de facilidad al momento de visualizar la posición y datos de los buses, según las respuestas de los encuestados el 40% de personas encuestadas indica que es muy fácil, el 34% señala que es fácil, el 18% que es difícil y el 8% señala que es muy difícil.

Encuesta pregunta número 5

Figura 55. Encuesta pregunta número 5.

5. ¿Cómo fue el proceso de guardar subrutas a favoritos ?

50 respuestas



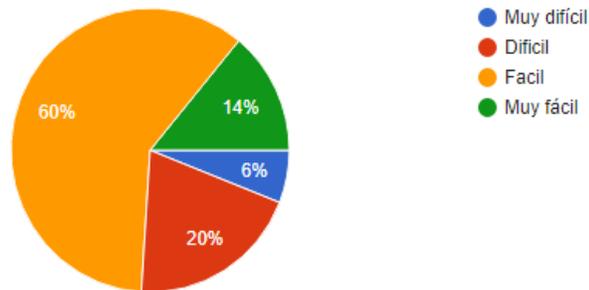
Descripción: En la figura 58 muestra los porcentajes de facilidad al momento de guardar las sub-rutas a favoritos, según las respuestas de los encuestados el 28% de personas encuestadas indica que es muy fácil, el 54% señala que es fácil, el 16% que es difícil y un mínimo porcentaje señala que es muy difícil.

Encuesta pregunta número 6

Figura 56. Encuesta pregunta número 6.

6. ¿Qué tan fácil es comprender el proceso de sugerencia de rutas que brinda la aplicación?

50 respuestas



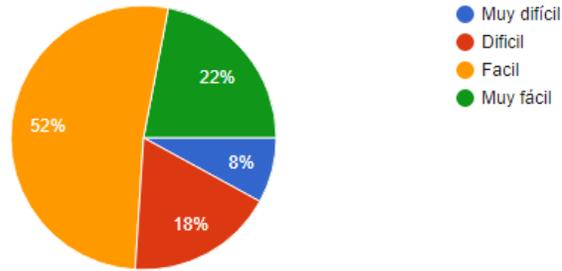
Descripción: En la figura 59 muestra los porcentajes de facilidad al momento de comprender la sugerencia recibida, según las respuestas de los encuestados el 60% de personas encuestadas indica que es muy fácil, el 20% señala que es fácil, el 14% que es difícil y el 6% señala que es muy difícil.

Encuesta pregunta numero 7

Figura 57. Encuesta pregunta número 7.

7. ¿Qué le pareció, la manera de escoger la dirección destino en la sugerencia de ruta por medio de un marcador en el mapa?

50 respuestas



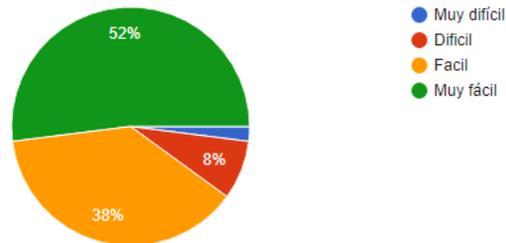
Descripción: En la figura 60 muestra los porcentajes de facilidad al momento de seleccionar el lugar destino por medio de un marcador en el mapa, según las respuestas de los encuestados el 22% de personas encuestadas indica que es muy fácil, el 52% señala que es fácil, el 18% que es difícil y el 8% señala que es muy difícil.

Encuesta pregunta número 8

Figura 58. Encuesta pregunta número 8.

8. ¿Cómo fue la navegación por los diferentes ítems de la aplicación FindRoute?

50 respuestas



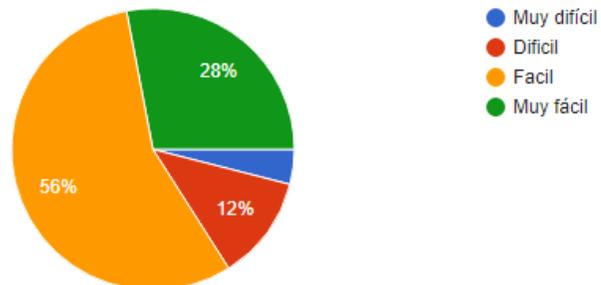
Descripción: En la figura 61 muestra los porcentajes de facilidad al momento de interactuar por los diferentes ítems de la aplicación, según las respuestas de los encuestados el 52% de personas encuestadas indica que es muy fácil, el 38% señala que es fácil, el 8% que es difícil y un mínimo porcentaje señala que es muy difícil navegar por los diferentes ítems de la aplicación.

Encuesta pregunta numero 9

Figura 59. Encuesta pregunta número 9.

9. En general, ¿cómo considera el uso de la aplicación móvil?

50 respuestas

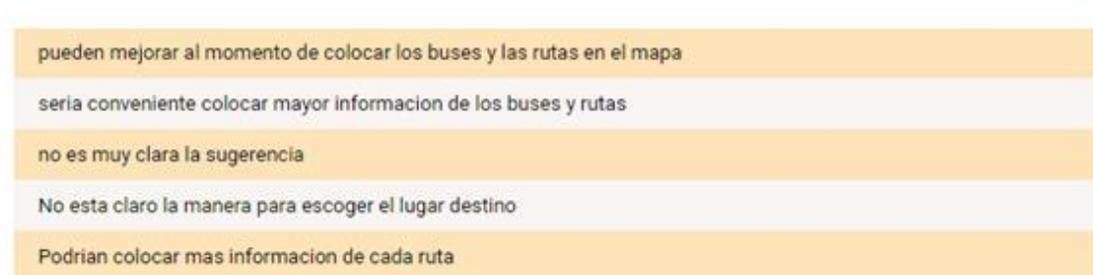


Descripción: En la figura 62 muestra los porcentajes de facilidad al momento de usar la aplicación, según las respuestas de los encuestados el 28% de personas encuestadas indica que es muy fácil, el 56% señala que es fácil, el 12% que es difícil y un mínimo porcentaje señala que es muy difícil usar la aplicación.

Encuesta pregunta número 10

Figura 60. Encuesta pregunta número 10.

10. ¿Tiene alguna sugerencia para mejoras de la aplicación móvil?, Si la respuesta es sí, ¿Cuál?



Descripción: En la figura 63 muestra las respuestas de las sugerencias que brindan los usuarios como también las mejoras que se le puede hacer a la aplicación en un futuro.