



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Aplicaciones con Tecnología Internet



Desarrollo de una Aplicación Móvil para el Monitoreo de Rutas de Transportes Escolares

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por los Bachilleres:

Josué G. Sánchez R.
C.I: 24.757.111
josueguitar2@gmail.com

José G. Yvimas B.
C.I: 23.712.348
jyvimas@gmail.com

Para optar al título de Licenciado en Computación

Tutores:
Prof. Antonio Russoniello
Prof. Dedaniel Urribarri

Caracas, octubre 2018



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Aplicaciones con Tecnología Internet

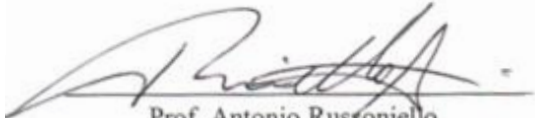
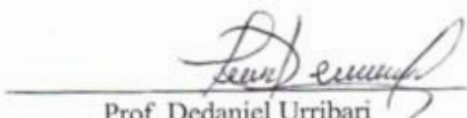
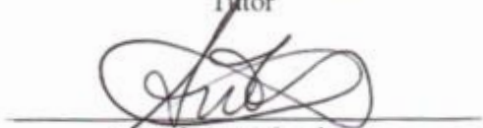
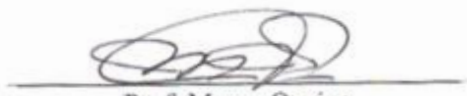


ACTA DEL VEREDICTO

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por los Bachilleres Josué Sánchez (C.I.: 24.757.111) y José Yvimas (C.I.: 23.712.348), con el título **“Desarrollo de una Aplicación Móvil para el Monitoreo de Rutas de Transportes Escolares”**, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 25 de octubre de 2018, a la 11:00 am, para que sus autores lo defendieran en forma pública, en el aula de Internet II, mediante una exposición oral de su contenido, y luego de la cual respondieron satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas el 25 octubre de 2018, dejándose también constancia de que actuó como coordinador del Jurado el Profesor Tutor Antonio Russoniello.

 Prof. Antonio Russoniello Tutor	 Prof. Dedaniel Urribari Tutor
 Prof. Ana Morales Jurado Principal	 Prof. Mercy Ospina Jurado Principal

Agradecimientos

En el transcurso de mi carrera universitaria he recibido el apoyo, consejo, compañía, inspiración y cariño de muchas personas. Con el aporte de cada uno de ellos he logrado cumplir un objetivo que tenía propuesto hace mucho tiempo y además he alcanzado muchas otras metas que ni siquiera imaginé que podría conseguir. Dedico el presente Trabajo Especial de Grado a cada uno de ellos, siempre estaré totalmente agradecido.

Principalmente agradezco a Dios y a la Virgen María, por cada una de las oportunidades que han puesto en mi vida. Cada reto que me han planteado he logrado superarlo y obtener nuevas experiencias que permiten mi crecimiento como persona. Ellos siempre están presentes en mis éxitos y agradezco profundamente sus bendiciones.

Agradezco a mi mamá, Carmen Reyes por su inmenso amor, por ser mi consejera, mi mejor amiga, estar siempre en todos los aspectos de mi vida, por motivarme a siempre alcanzar lo que me proponga y decirme siempre “que no hay limite a menos que tú te lo pongas”, por enseñarme valores, educarme y a ser una gran persona. Gracias infinitas por ser mi mama, te dedico cada uno de mis logros porque tú siempre serás partes de ellos. TE AMO MAMA.

Agradezco a mi papa José Sánchez, mi tía Glileida Pachano, mis primas que más que primas son mis hermanas Daleigly Hernández y Hergly Hernández, mi abuela Elba Pachano que su amor por mí no tiene límite. Gracias por ser mi núcleo, las personas con que siempre puedo contar, las personas que han visto mi crecimiento y han hecho su aporte en la construcción de cada uno de mis logros en mi vida, por ser mi familia, la familia no siempre es perfecta, todos tenemos muchas diferencias y personalidades que contrastan pero la familia no se elige y yo jamás pensaría en tener otra familia donde no estén ustedes presentes. La familia será siempre lo más importante en mi vida.

A la Ilustre Universidad Central de Venezuela, que tanto le ha dado al país, orgulloso me siento de formar parte de ésta casa de estudios y de representarla. A la Facultad de Ciencias y a todo el personal que allí labora, incluyendo a todos los profesores de los que he tenido la oportunidad de recibir clases, por sus enseñanzas, por sus conocimientos y por su sabiduría. En especial, gracias mis tutores Antonio Russoniello y Dedaniel Urribari, por la guía y orientación brindada a lo largo de la investigación.

A mi mejor amiga Stefania Martínez, por ser mi consejera, mi psicóloga, mi confidente, mi chofer, por su inmenso amor. Tenemos más de 16 años conociéndonos donde los dos hemos crecido y cambiado muchísimo y nuestra amistad se fortalece aún más.

A mi amigo Jonathan Rojas tenemos una amistad de prácticamente toda la vida en la cual me regalas toda tu amistad y apoyo incondicional. Nosotros no somos amigos sino hermanos.

A mi amigo Leymer Rosales estudiamos juntos en el colegio, somos vecinos, el destino nos hizo presentar hasta la prueba de admisión juntos en el mismo escritorio donde logramos nuestra meta para ese momento, te agradezco tu amistad y valoro mucho lo que aportas en mi vida y como no darte el crédito de que este Trabajo Especial de Grado es la producción de una idea que tuvimos hace muchos años en nuestras discusiones de inventos que mejoran la sociedad.

A mis amigos Carlos Álvarez, Fernando Piñeiro, Fernando Lozano y Ramsés Fernández por su amistad, apoyo y consejos que me han dado. Por estar en varias experiencias de mi vida. A los que no he mencionado no los olvido solo que sino serian infinitos mis agradecimientos. A todos los quiero mucho y siempre les deseo éxito en todas sus metas como ustedes siempre hacen conmigo.

A cada uno de los compañeros que conseguí a lo largo de la carrera, gracias por el apoyo y ayuda que me brindaron. Un compañero a destacar es Francisco Iannini el cual ha estado compartiendo conmigo en toda la carrera y aunque no logramos graduarnos juntos sé que estas apoyándome siempre.

Agradezco a mi compañero José Yvimas cursamos media carrera juntos, decidimos ser preparadores de la misma materia, nos apoyamos, nos esforzamos en nuestras evaluaciones y crecimos mucho como profesionales y como personas en estos años, decidimos desarrollar un Trabajo Especial de Grado el cual nos permitió cumplir ese objetivo del que tanto anhelamos. Gracias por todo el apoyo que siempre me has brindado, por tu esfuerzo y dedicación en todos nuestros trabajos juntos y por tu amistad.

Por último agradezco a Gabriela Porras y Jennifer Cádiz por ser de las grandes personas que me ha permitido conocer la universidad, ambas a su manera siempre me apoyan en todo el esfuerzo que aplico para cumplir mis objetivos. Gracias infinitas.

Josue Sánchez.

Agradecimientos

Durante todo el transcurso de mi vida universitaria, desde que comencé estudiando la Licenciatura en Química por un año y otros cinco años en Computación, he contado con el apoyo incondicional de muchas personas que me han brindado comprensión, compañía, inspiración y cariño para el logro de cada una de mis metas y la finalización exitosa del presente trabajo y de mi carrera como profesional.

Agradezco principalmente a Dios y al Doctor José Gregorio Hernández por cada una de sus bendiciones. Por cada uno de los retos y dificultades que me han ayudado a superar y las oportunidades de éxito que han puesto en mi camino.

A la mejor mamá de todas, mi mamá Luz Briceño, por su inmenso amor, por ser mi confidente, mi consejera, mi mejor amiga, y el único y verdadero amor de mi vida. Por siempre estar para mí en todo momento, por preocuparse hasta en lo más mínimo, por mover el cielo y la tierra de ser necesario para ayudarme a solucionar mis problemas. Por ser mi ejemplo a seguir e inspiración para ser el mejor en todo lo que me proponga. Gracias infinitas, por demostrarme que siempre estarás para mí, por eso le pido a Dios que te cuide siempre, que te dé mucha salud y vida para que sigas viendo los logros y éxitos de tus hijos. Todo esto es por ti y para ti ¡Te amo como a nadie en este mundo!

A mi papá José Yvimas, a mis hermanas Cecilia y Jhorianny por siempre brindarme su amor, cariño, compañía, apoyo, consejos y por siempre estar presentes en todo momento ¡Los quiero mucho!

A mis abuelos Ángel Briceño y Zoraida Cruz, mis tíos Ana Briceño, Eva Briceño y José Millán por todo su cariño, apoyo y por estar presente en cada uno de los momentos importantes de mi vida. A mis primos Nestor Gelvis y Andrifer Tovar que a pesar que estén lejos siempre me han apoyado y han estado pendiente de mis logros ¡Los quiero mucho!

A mi mejor amiga Alina Materano por ser una persona especial en mi vida durante más de siete años, por su apoyo, consejos, cariño, por siempre estar para mí y acompañarme en cada uno de mis momentos importantes. Gracias por cruzarte en mi camino y por tu valiosa amistad ¡Te amo!

A mi bro, mi amigo y compañero de trabajo Jesús Materano, por tu amistad, apoyo y consejos que me has dado. Por permitir desarrollarme profesionalmente a tu lado, por darme la ambición de siempre ser los mejores y hacer lo mejor, y establecernos metas y logros para el futuro ¡Te quiero!

A mi profesora y tutora de tesis de bachillerato María Gabriela Carrasco, por todo su apoyo desde que salí de quinto año. Por siempre estar presente en los momentos de logros importantes en mi desarrollo como profesional ¡La quiero mucho!

A todos mis compañeros que conseguí a lo largo de mi carrera universitaria con los cuales compartí tanto buenos como malos momentos, gracias por el apoyo y ayuda que me brindaron en sus respectivos momentos. Especial mención a mi compañero Josue Sánchez, con el cual he compartido más de la mitad de mi carrera universitaria. Gracias por tu amistad, por tu apoyo, esfuerzo y dedicación en cada uno de los trabajos y evaluaciones que compartimos juntos.

A la Ilustre Universidad Central de Venezuela, la mejor universidad de nuestro país, mi segunda casa. Orgullosamente me siento ser egresado y de representar a la casa que vence las sombras. A la Facultad de Ciencias y a cada uno de los profesores con los que tuve la oportunidad de recibir sus conocimientos, consejos, enseñanzas y experiencias que permitieron desarrollar al profesional que soy hoy en día. Gracias a mis tutores Antonio Russoniello y Dedaniel Urribari, que además de recibir clase por parte de ellos, aceptaron el reto de guiar el desarrollo del presente trabajo y brindar el apoyo y la orientación necesaria. Agradecimiento especial al profesor Miguel Astor por sus consejos y recomendaciones que sirvieron como mejoras de la investigación.

José Yvimas.

Resumen

Título:

Desarrollo de una Aplicación Móvil para el Monitoreo de Rutas de Transportes Escolares

Autores:

Josué G. Sánchez R.

José G. Yvimas B.

Tutores:

Prof. Antonio Russoniello

Prof. Dedaniel Urribari

Durante los últimos años los dispositivos móviles han avanzado en su evolución y producido enormes cambios en la sociedad y en la manera en que transcurre el día a día, en gran parte se debe, a las distintas funcionalidades tecnológicas que incluyen y las diversas aplicaciones que están a disposición de los usuarios. En el caso de estudio de los transportes escolares el uso de tecnología puede mejorar la experiencia del usuario sobre el servicio prestado y esto puede observarse como una gran oportunidad de negocio. El servicio de transporte escolar es utilizado por una cantidad considerable de representantes y muchas veces estos desean saber sobre la ubicación y estado tanto de su representado como del autobús donde viajan los mismos.

En este Trabajo Especial de Grado se estudió y examinó diferentes tecnologías, herramientas, lenguajes y enfoques para el desarrollo de una solución tecnológica para transportes escolares que posea como objetivo principal un sistema de geolocalización capaz de posicionar el transporte escolar durante sus jornadas laborales.

La solución tecnológica mencionada abarca una aplicación móvil utilizando el enfoque de desarrollo híbrido, capaz de ser utilizada en los dispositivos móviles con sistema Android y iOS. La aplicación móvil cuenta con diferentes roles de acceso para cada uno de los tipos de usuarios permitidos y así mismo con un administrador que permite la carga de contenido, la creación de usuarios y demás funciones administrativas. Además, la aplicación ofrece un mapa en tiempo real donde se visualiza la posición del transporte, la parada del usuario en cuestión y un tiempo aproximado de llegada a dicho punto, notificaciones de difusión amplia en caso de accidentes o inconvenientes que deban ser informado a los usuarios, notificaciones personales cuando el transporte arriba a la parada designada del usuario.

Se hizo uso de las tecnologías GPS y GSM para la localización del autobús con un hardware desarrollado en una placa Arduino Uno, por medio de este hardware ubicado en el interior del autobús se envían las coordenadas de geolocalización del autobús. Por otro lado, se implementó una funcionalidad de rastreo de contingencia

para ser usada desde el dispositivo móvil del chofer del transporte, esto en caso de que el hardware presente inconvenientes.

Finalmente, como resultado de la investigación y del proceso de desarrollo se puede decir que se realizaron de manera satisfactoria los prototipos funcionales de la aplicación móvil. En el caso del hardware planteado en un comienzo no se pudo realizar la integración correcta debido a defectos en uno de los dispositivos utilizados. No obstante, gracias a la funcionalidad de rastreo de contingencia se cumplió con el objetivo del prototipo de hardware.

Palabras Claves: Aplicación Móvil, Placa de desarrollo, Geolocalización, Transporte escolar, GPS, GSM, Aplicación híbrida.

Tabla de Contenido

Índice de Figuras	12
Índice de Tablas	15
1. Introducción	16
2. El Problema	17
2.1 Planteamiento del Problema	17
2.2 Justificación del Problema	17
2.3 Objetivos	18
2.3.1 Objetivo General	18
2.3.2 Objetivos Específicos	18
2.4 Alcance	19
3. Antecedentes	20
3.1 School Bus Tracker - Conjoinix	20
3.2 Closer-Parents (School Bus Tracker)	21
3.3 Próximo Bus Barcelona	22
3.4 School Bus Tracker - Elitech Lab	23
3.5 ALZRASTREO	24
4. Dispositivos Móviles	26
5. Desarrollo de Aplicaciones Móviles	27
5.1 Aplicaciones Híbridas	27
5.2 IONIC	28
6. Herramientas de Desarrollo	29
6.1 Application Programming Interface (API)	29
6.1.1 Leaflet	29
6.1.2 Open Route Service	29
6.2 Bases de Datos	30
6.2.1 Tipos	30
6.3 Firebase	31
6.3.1 Características	31
6.3.2 Firebase Authentication	31
6.3.3 Cloud Firestore	31
6.3.4 Cloud Functions	32
6.3.5 Cloud Storage	32

7.	Geolocalización	33
7.1	Geolocalización vs Georeferenciación	33
8.	Hardware para Prototipos	34
8.1	Tarjetas o Placas de Desarrollo	34
8.1.1	Arduino	34
8.2	Módulo GPS	35
8.3	Módulo GSM	36
8.4	SMS Gateway	37
9.	Metodologías Ágiles de Desarrollo de Software	38
9.1	Scrum	38
10.	Marco Metodológico	40
10.1	Adaptación de la Metodología de Desarrollo	40
10.1.1	Eventos	41
10.1.2	Tecnologías Utilizadas	42
11.	Marco Aplicativo	45
11.1	Arquitectura de la Solución	45
11.2	Planificación de las Actividades del Proyecto	47
11.3	Análisis General de la Aplicación Móvil	53
11.3.1	Artefactos y Diseño de Interfaces de la Aplicación Móvil	54
11.3.2	Instalación y Configuración del Ambiente de Trabajo de la Aplicación	64
11.3.3	Desarrollo de la Aplicación Móvil	68
11.3.4	Despliegue de la Aplicación Móvil	82
11.3.4.1	Despliegue en Sistemas Android	82
11.3.4.1	Despliegue en Sistemas iOS	84
11.4	Análisis General del Servicio en la Nube	85
11.4.1	Instalación y Configuración del Ambiente de Trabajo para la Cloud Function	85
11.4.2	Desarrollo de la Cloud Function	87
11.5	Análisis General del Prototipo de Hardware	89
11.5.1	Acoplamiento y Configuración de los Dispositivos de Hardware	89
11.5.2	Desarrollo de Software para Prototipo de Hardware	90
11.6	Configuración de SMS Gateway	92
12.	Pruebas y Análisis de Resultados	94
12.1	Pruebas de Aceptación	94
12.1	Pruebas de Funcionalidad	100

12.1.1	Estado del Estudiante	100
12.1.2	Mensajes del Chofer	102
12.1.3	Geolocalización	104
13.	Conclusiones y Trabajos Futuros	109
	Referencias	112

Índice de Figuras

Figura 3.1: Pantalla de School Bus Tracker - Conjoinix sobre la información actual de la posición y velocidad del autobús.....	20
Figura 3.2: Pantalla de Closer Parents que muestra la funcionalidad de cambio de locación de parada del autobús	22
Figura 3.3: Pantalla de Próximo Bus Barcelona que indica el tiempo estimado de llegada del autobús a sus respectivas paradas en la línea PL Espanya	23
Figura 3.4: Pantalla de School Bus Tracker que muestra la lista de estado actual de los estudiantes (En el bus, Ausente, Llegó a su hogar).....	24
Figura 4.1: Dispositivos móviles.....	26
Figura 8.1: Arduino Uno.....	35
Figura 8.2: Adafruit Ultimate GPS Breakout V3.....	36
Figura 8.3: Prototipo GSM SIM800L	36
Figura 8.4: Esquema de flujo de un SMS Gateway	37
Figura 9.1: Flujo Metodología Scrum	39
Figura 10.1: Tablero que refleja un Sprint Backlog List del proyecto en el Sprint 1 ..	41
En el caso de los diagramas UML y de navegación serán explicados con mayor detalle en el capítulo de Marco Aplicativo.	41
Figura 10.2: Tablero de reportes de Dailys Meeting en Google Keep	42
Figura 11.1: Esquema de Diseño de Solución	45
Figura 11.2: Prototipo de hardware para el proyecto.....	46
Figura 11.3: Diagrama de Casos de Uso de Primer y Segundo Nivel de la Aplicación Móvil	56
Figura 11.4: Diagrama de Navegación para usuario Administrador	57
Figura 11.5: Diagrama de Navegación para usuario Representante	57
Figura 11.6: Diagrama de Navegación para usuario Chofer.....	58
Figura 11.7: Diseño de Splash Screen de la Aplicación Móvil.....	58
Figura 11.8: Diseño de la Pantalla de Inicio de Sesión de la Aplicación Móvil.....	59
Figura 11.9: Diseño de la Pantalla de Bienvenida de la Aplicación Móvil	59
Figura 11.10: Diseño del Menú de cada tipo de usuario de la Aplicación Móvil (Administrador, Representante y Chofer).....	60
Figura 11.11: Diseño de la vista Mensajes del Chofer de la Aplicación Móvil.....	60
Figura 11.12: Diseño de la vista de creación de estudiantes y cuentas de usuarios de representantes y chofer de la Aplicación Móvil	61
Figura 11.13: Diseño de la vista de Estado de mi representado de la Aplicación Móvil	61
Figura 11.14: Diseño de la vista de Ubicación del Autobús de la Aplicación Móvil...	62
Figura 11.15: Diseño de la vista de Perfil y Editar Perfil de la Aplicación Móvil	62
Figura 11.16: Diseño de la vista de Mensajes a los Padres de la Aplicación Móvil ..	63
Figura 11.17: Diseño de la vista de Estado de los Alumnos de la Aplicación Móvil ..	63
Figura 11.18: Comando para la instalación del Framework Ionic	64
Figura 11.19: Comando para la creación de un proyecto en Ionic.....	64
Figura 11.20: Proyecto de la aplicación móvil en la plataforma Github.....	65
Figura 11.21: Creación de un proyecto en Firebase.....	65
Figura 11.22: Productos Firebase para el Desarrollo	66

Figura 11.23: Creación Base de Datos Cloud Firestore	66
Figura 11.24: Pantalla principal de la base datos Firestore	67
Figura 11.25: Comando de instalación de la librería de AngularFire2	67
Figura 11.26: Inicialización de la librería AngularFire2 en el módulo principal de la aplicación móvil	68
Figura 11.27: Estructura de la carpeta src de la Aplicación Móvil.....	69
Figura 11.28: Ejemplo de documentos de las colecciones de la Base de Datos	71
Figura 11.29: Pantalla de Inicio de Sesión	72
Figura 11.30: Pantallas de Inicio por cada tipo de usuario	72
Figura 11.31: Pantalla para crear estudiantes en la base de datos	73
Figura 11.32: Pantalla para crear usuarios de tipo Representante	74
Figura 11.33: Pantalla para crear usuarios de tipo Chofer	74
Figura 11.34: Pantalla de Mensajes del chofer (Administrador y Representante)	75
Figura 11.35: Pantalla para ingresar la dirección de parada	76
Figura 11.36: Pantalla de Estado de mi representado.....	77
Figura 11.37: Pantalla de Ubicación del Autobús.....	78
Figura 11.38: Pantallas de Perfil y Editar Perfil	79
Figura 11.39: Pantallas de Mensajes a representantes y Mensajes predeterminados	80
Figura 11.40: Pantalla de Estado de los estudiantes.....	81
Figura 11.41: Pantalla de Rastreo de contingencia	82
Figura 11.42: Comando para agregar la plataforma Android a la estructura del proyecto.....	83
Figura 11.43: Archivo config.xml de la Aplicación Móvil	83
Figura 11.44: Comando para la compilación y despliegue en el emulador	83
Figura 11.45: Comando para la compilación y generación de archivo ejecutable optimizado para producción	84
Figura 11.46: Comando para el despliegue de la aplicación móvil en un emulador de dispositivos iOS	84
Figura 11.47: Aplicación ejecutándose en el emulador de iOS nativo de Ionic.....	85
Figura 11.48: Instalación Cloud Functions en el proyecto	86
Figura 11.49: Comando para autenticar la herramienta de Firebase.....	86
Figura 11.50: Configuración de la herramienta Cloud Functions y selección de proyecto Firebase a utilizar	86
Figura 11.51: Selección de JavaScript como lenguaje para escribir la Cloud Function	86
Figura 11.52: Importación de módulos necesarios e inicialización de la app en el desarrollo de Cloud Function	87
Figura 11.53: Implementación de la Cloud Function saveLocation.....	88
Figura 11.54: Despliegue de la Cloud Function saveLocation.....	88
Figura 11.55: Instalación del IDE de Arduino en Windows	89
Figura 11.56: Importación de librerías requeridas	90
Figura 11.57: Asignación de pines a los módulos	90
Figura 11.58: Función setup encargada de inicializar cada uno de los módulos	90
Figura 11.59: Función getCoordinates encargada de obtener las coordenadas del dispositivo.....	91
Figura 11.60: Función sendMessage encargada de enviar las coordenadas vía SMS	91

Figura 11.61: Función principal del prototipo de hardware encargada ejecutar toda la lógica del negocio	92
Figura 11.62: Configuración de url a retransmitir SMSSync	92
Figura 11.63: Palabras claves designadas para la transmisión de SMS	93
Figura 12.1: Pregunta 1 – Prueba de Aceptación.....	95
Figura 12.2: Pregunta 2 – Prueba de Aceptación.....	96
Figura 12.3: Pregunta 3 – Prueba de Aceptación.....	96
Figura 12.4: Pregunta 4 – Prueba de Aceptación.....	97
Figura 12.5: Pregunta 5 – Prueba de Aceptación.....	97
Figura 12.6: Pregunta 6 – Prueba de Aceptación.....	98
Figura 12.7: Pregunta 7 – Prueba de Aceptación.....	98
Figura 12.8: Pregunta 8 – Prueba de Aceptación.....	99
Figura 12.9: Pregunta 9 – Prueba de Aceptación.....	99
Figura 12.10: Pregunta 10 – Prueba de Aceptación.....	100
Figura 12.11: Estado de los estudiantes – Prueba de Funcionalidad	101
Figura 12.12: Estado de mi representado – Prueba de Funcionalidad	102
Figura 12.13: Alerta de mensaje vacío – Prueba de Funcionalidad.....	103
Figura 12.14: Sin mensajes del chofer – Prueba de Funcionalidad.....	103
Figura 12.15: Listado de mensajes del chofer – Prueba de Funcionalidad.....	104
Figura 12.16: Esquema de ruta – Prueba de Geolocalización.....	105
Figura 12.17: Resultado de prueba de los tres días – Prueba de Geolocalización .	108

Índice de Tablas

Tabla 5.1: Ventajas/desventajas de las aplicaciones híbridas	28
Tabla 7.1: Geolocalización vs Georeferenciación	33
Tabla 8.1: Especificaciones Arduino UNO, Leonardo y YUN	34
Tabla 11.1: Sprint 1 – Josue Sánchez	47
Tabla 11.2: Sprint 1 – José Yvimas	48
Tabla 11.3: Sprint 2 – Josue Sánchez	48
Tabla 11.4: Sprint 2 – José Yvimas	49
Tabla 11.5: Sprint 3 – Josue Sánchez	49
Tabla 11.6: Sprint 3 – José Yvimas	50
Tabla 11.7: Sprint 4 – Josue Sánchez	50
Tabla 11.8: Sprint 4 – José Yvimas	51
Tabla 11.9: Sprint 5 – Josue Sánchez	51
Tabla 11.10: Sprint 5 – José Yvimas	52
Tabla 11.11: Sprint 6– Josue Sánchez	52
Tabla 11.12: Sprint 6 – José Yvimas	53
Tabla 12.1: Tiempos estimados – Prueba de Geolocalización	106
Tabla 12.2: Primer día de pruebas – Prueba de Geolocalización	107
Tabla 12.3: Tercer día de pruebas – Prueba de Geolocalización	107

1. Introducción

Los avances de la tecnología aumentan a pasos gigantes, y en la actualidad existen una gran cantidad de soluciones tecnológicas que les permiten a las personas mejorar su día a día. Muchos de estos avances tecnológicos se enfocan en *dispositivos móviles*, los cuales cuentan con un sinnúmero de aplicaciones que solucionan operaciones que antes exigían una mayor inversión de tiempo y que permiten potenciar la comunicación.

Actualmente, las distintas funcionalidades tecnológicas que incluyen diversas aplicaciones disponibles para los usuarios, generan nuevas tendencias y modelos de comportamiento que cada vez se hacen más relevantes para las organizaciones de diversos sectores de actividad, ganando grandes oportunidades de negocio si se adaptan a ellas. Por ello, una de las organizaciones que deben apostar a los recursos tecnológicos son los transportes escolares los cuales no solo deben cumplir el esquema tradicional de ser el medio de transporte de un estudiante a su hogar, sino mejorar la experiencia de los representantes, brindar un servicio donde existan medios de comunicación usables, información relevante del transporte, del estudiante y del viaje en sí mismo, incorporando el uso de los recursos tecnológicos.

A los representantes que utilicen el servicio del transporte escolar, se les garantiza llevar a sus representados a la institución desde una determinada parada y el retorno a la misma luego de que haya terminado la jornada escolar. Pero cuando su representado viaja en el transporte escolar cada padre y/o representante no tiene conocimiento alguno de la situación y localización del autobús, de cuánto tiempo tardará en llegar su representado al destino, así como otra información relevante sobre el mismo.

En base a lo anterior, el objetivo principal de esta investigación es detallar y analizar diversas tecnologías que favorezcan el desarrollo de una aplicación móvil que permita a cada uno de los representantes que utilicen el transporte escolar visualizar información sobre la localización del autobús donde viaja su representado, tiempo estimado de viaje y así como también recibir notificaciones acerca del estado de la unidad.

2. El Problema

En el presente capítulo se describe y plantea el problema observado en los transportes escolares y los representantes que utilizan dicho servicio, junto con la justificación por la cual esos problemas pueden ser solucionados con la implementación de aplicaciones y uso de herramientas tecnológicas, los objetivos que se desean alcanzar, y el alcance del proyecto.

2.1 Planteamiento del Problema

En la actualidad, son muchas las comunidades escolares que cuentan con el servicio de transporte para sus estudiantes, y algunos padres o representantes tienen la necesidad de usar esta alternativa ya que su trabajo u otras actividades no les permiten llevar y/o traer a sus representados.

Durante el viaje de los estudiantes los representantes no poseen información sobre la localización del transporte, ni de ninguno de los otros aspectos que pueden influir en el uso del mismo, como lo son: el tráfico, si el representado se encuentra en dicho transporte, fallas en la unidad que puedan ocurrir, entre otros.

Los transportes escolares cuentan con muchos usuarios donde cada uno de ellos es transportado a diferentes locaciones asignadas por sus representantes. Debido a la cantidad de usuarios y los problemas de tráfico en la ciudad, surge el problema que afecta a los representantes el cual consiste en no tener conocimiento de donde se encuentra su representado, no saber sobre el estado de la unidad, los posibles percances que puedan ocurrir como lo son los accidentes automovilísticos o eventos desafortunados y en la mayoría de ocasiones no existe un método de comunicación con el encargado de la unidad de transporte escolar. Estos son parte de los problemas expuestos por Shahid Mansuri [1] cofundador de YellowSoft, una empresa de desarrollo de software con proyectos en el área de rastreo de autobuses escolares y reserva de taxis. También se presenta la situación en la que los representantes desean conocer un tiempo aproximado de llegada de su representado a la parada asignada, por lo cual pueden caer en la necesidad de esperar en la parada de encuentro más tiempo de lo que ellos estiman que debe llegar el transporte.

2.2 Justificación del Problema

Surge la necesidad de desarrollar una solución tecnológica que pueda resolver la problemática anteriormente planteada y que sea accesible por todos los representantes que utilicen el servicio del transporte escolar y posean un dispositivo móvil inteligente. Tomando en consideración los hechos anteriores y apoyándose de que hoy en día existe un gran número de personas que tienen teléfonos inteligentes según Gartner Inc, se desarrollará una aplicación móvil para el monitoreo de las rutas del transporte escolar, en la cual podrán consultar en tiempo real la localización del transporte, si el usuario al que se representa se encuentra en dicha unidad, tiempo

estimado de llegada a su destino, alertas o notificaciones al usuario para eventos específicos. Adicionalmente se desea que la aplicación funcione mediante tecnología GPS y tengan un diseño atractivo usable para el público.

2.3 Objetivos

En esta sección se definen los objetivos que se quieren conseguir con la realización del trabajo propuesto.

2.3.1 Objetivo General

Desarrollar una aplicación móvil para monitorear las rutas de transportes escolares, que permita a los padres o representantes mantenerse informados sobre la localización de la unidad donde viajan sus representados mediante la tecnología GPS.

2.3.2 Objetivos Específicos

1. Definir los requerimientos funcionales y no funcionales en el desarrollo de una aplicación móvil para el monitoreo de transportes escolares.
2. Desarrollar un software que permita transmitir a través de la red GSM, las coordenadas actuales del transporte haciendo uso de un hardware con tecnología GPS.
3. Implementar una arquitectura y establecer los modelos para el uso de una base de datos no relacional basados en colecciones y documentos.
4. Configurar y establecer los lineamientos para la comunicación con el SMS Gateway y la retransmisión de mensajes hacia el servicio en la nube.
5. Desarrollar un servicio en la nube que reciba información de coordenadas y las almacene en la base de datos.
6. Construir una aplicación móvil que consuma servicios de geolocalización para mostrar la ubicación en tiempo real y estimar el tiempo de llegada del transporte escolar.
7. Realizar pruebas de aceptación y pruebas de funcionalidades para la aplicación móvil.

2.4 Alcance

La aplicación móvil híbrida a desarrollar contará con la funcionalidades de monitoreo del transporte escolar, visualización mediante un mapa la ubicación del transporte escolar en tiempo real, mantener la asistencia de los estudiantes en la unidad, tiempo estimado de llegada a su destino y notificaciones al usuario en caso de surgir inconvenientes.

El desarrollo de la aplicación se realizará en ambientes controlados, tomando como caso de estudio un vehículo propio que representará el transporte escolar y una cantidad de 5 de estudiantes simulados, para así obtener resultados para futuros análisis y actualizaciones del proyecto.

La aplicación tendrá como límite un transporte con un máximo de 10 estudiantes y el recorrido no puede superar las 2 horas, estos son los límites establecidos para garantizar el funcionamiento del sistema.

Además, se realizarán pruebas de aceptación de la aplicación móvil y pruebas de funcionalidad enfocadas en aquellas funcionalidades consideradas con más importancia por parte de los autores del proyecto.

3. Antecedentes

Este capítulo tiene como objetivo presentar antecedentes o trabajos relacionados con la presente investigación. Donde dichos trabajos poseen enfoques muy parecidos a la aplicación móvil del actual caso de estudio, además de esto, cuentan con características y funcionalidades que se esperan contemplar en el desarrollo de la misma. Por otro lado, sirven de apoyo para los autores, ya que les permiten obtener una serie de información útil mediante el análisis de las funciones que implementan y de los resultados obtenidos por cada trabajo, esto con el fin de encontrar los requerimientos necesarios para desarrollar una aplicación que cumpla con la función para la que fue desarrollada.

3.1 School Bus Tracker - Conjoinix

School Bus Tracker es una aplicación móvil desarrollada para dispositivos móviles con el objetivo de permitir a los padres poder realizar un seguimiento a sus hijos con la ayuda del posicionamiento vía GPS instalando en el autobús escolar.

Esta aplicación fue desarrollada por Conjoinix una empresa de desarrollo de soluciones tecnológicas en la India con un gran portafolio de servicios expuestos al mercado.

- Incluir identificador para cada autobús del colegio.
- Proporcionar ubicación actual del autobús con su velocidad actual tal como se puede observar en la Figura 3.1.
- Notificación cuando el transporte se encuentra cercano a la parada del usuario.



Figura 3.1: Pantalla de School Bus Tracker - Conjoinix sobre la información actual de la posición y velocidad del autobús

Esta aplicación es la primera que se encuentra en el ranking de aplicaciones de la Google Play Store cuando se busca por las palabras claves school, bus, tracker, fue seleccionada por la gran cantidad de críticas positivas que posee la aplicación, funcionalidades similares a las que se desean implementar como el inicio simple de los representantes a la aplicación y el informar de manera constante sobre el posicionamiento del autobús con el uso de tecnología GPS. Un punto a destacar que influyó a los autores a seleccionar dicho trabajo como una guía de análisis, es que la aplicación contiene un dispositivo GPS y un acelerómetro encargado de suministrar la data a los servidores de la aplicación.

3.2 Closer-Parents (School Bus Tracker)

El sistema Closer-Parents ayuda a la administración de la escuela a monitorear y rastrear los movimientos de sus autobuses para garantizar la seguridad de los estudiantes durante el viaje desde la escuela hasta sus respectivos hogares. Esta nueva tecnología proporciona características que conectan a los padres con la ruta del autobús mediante la recepción de notificaciones antes de que el autobús llegue a casa y al abordar y bajarse del autobús, proporcionando un mecanismo de comunicación rápido e instantáneo con los padres.

Esta aplicación es desarrollada por Trackware LLC se encuentra en su versión 2.9.7 y se mantiene en constante actualizaciones, cuenta con una cantidad considerable de descargas y posee las siguientes funcionalidades:

- Los padres pueden designar la parada del estudiante con el uso de coordenadas.
- Listado de estudiantes.
- Velocidad del autobús.
- Tiempo estimado de llegada a la parada designada.

Los autores decidieron tomar como inspiración la funcionalidad que poseen los representantes de seleccionar una parada mediante el uso de coordenadas geográficas (Figura 3.2) y presentar en el trabajo una nueva implementación de dicha funcionalidad.

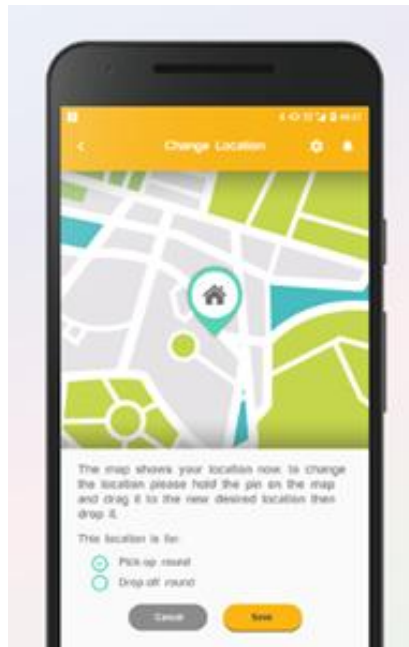


Figura 3.2: Pantalla de Closer Parents que muestra la funcionalidad de cambio de locación de parada del autobús

3.3 Próximo Bus Barcelona

La comunidad de Barcelona en España cuenta con una aplicación para la consulta del tráfico y estado actual de los buses. Se encuentra desarrollada para dispositivos Android¹ y iOS con gran aceptación por la comunidad dado la gran cantidad de descargas y comentarios que posee en las Tiendas de descarga.

Esta aplicación permite consultar cuánto tiempo falta para que el bus llegue a una determinada parada, en todas las paradas de Barcelona y cercanías. Solo se necesita introducir el código de la parada (cada parada está debidamente identificada por un código unívoco) y se podrá consultar cuánto tiempo falta para que las diferentes líneas lleguen a una parada, así como sus recorridos. Además cuenta con una funcionalidad de búsqueda sin necesidad de colocar el código de la parada en caso de deterioro o daño de dicha parada.

Las funcionalidades más relevantes son:

- Ver las paradas de bus y tranvía más cercanas.
- Ver qué líneas pasan por cada parada.
- Ver el recorrido de cada línea y los enlaces con otras líneas.
- Guardar paradas favoritas para que te sea más fácil consultarlas.
- Consultar los billetes y tarifas de transporte público.
- Añadir un widget a tu pantalla de inicio para que ni tan siquiera se tenga que abrir la aplicación.

¹ <https://play.google.com/store/apps/details?id=cat.ereza.properbusbcn>



Figura 3.3: Pantalla de Próximo Bus Barcelona que indica el tiempo estimado de llegada del autobús a sus respectivas paradas en la línea PL Espanya

Esta aplicación se diferencia a las demás aplicaciones que se presentan en este trabajo debido a que el contexto de la misma no son los autobuses escolares, sino el transporte público. Sin embargo fue seleccionada por los autores ya que cuenta con un diseño sencillo, práctico, usable y ciertas funcionalidades como el rastreo y monitoreo de rutas el cual es la función principal del trabajo propuesto.

3.4 School Bus Tracker - Elitech Lab

School Bus Tracker² es una aplicación creada por un equipo de desarrolladores Ingleses debido a la necesidad de tener un mayor control sobre los estudiantes en sus trayectos desde su hogar hacia el colegio. El Software y diseño es exclusivo, propiedad del Grupo tecnológico Elitech Lab – diseñado y desarrollado con la asistencia, colaboración, y feedback de uno de los colegios privados más prestigiosos del Reino Unido.

Permite conectar a los padres con el sistema de transporte escolar de sus hijos, lo que les permite localizarlos, al igual que a el autobús escolar, en todo momento durante su ruta y viceversa.

Esta aplicación funciona con un software de geolocalización y comunicación dinámica personalizada y customizada para cada colegio y cuenta con un back office

² <https://schoolbustrackerapp.com/>

(una plataforma de gestión) que proporciona un control total de todo el transporte escolar y la comunicación entre padres y el colegio. Se encuentra disponible en el mercado de los teléfonos y tabletas con SO tanto Android como iOS.

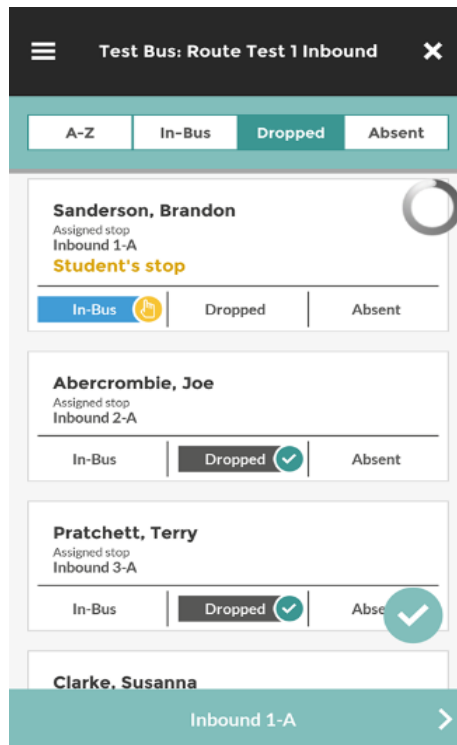


Figura 3.4: Pantalla de School Bus Tracker que muestra la lista de estado actual de los estudiantes (En el bus, Ausente, Llegó a su hogar)

School Bus Tracker se destaca de otras aplicaciones por el estado de cada estudiante al momento de las rutas escolares donde cada estudiante puede tener 3 estados (ausente, en el transporte y entregado a su hogar), estos estados van cambiando de acuerdo a los sucesos que ocurren en la ruta escolar. Esta funcionalidad fue del agrado de los autores y decidieron realizar una implementación similar en su propuesta tecnológica.

3.5 ALZRASTREO

ALZRASTREO es un trabajo especial de grado realizado en la UCV por la Lic. Carla Telleria. Es una aplicación móvil basada en geolocalización que tiene como objetivo principal permitir la conocer la ubicación de pacientes que padecen la enfermedad de Alzheimer por medio de servicios de ubicación de los dispositivos móviles, además de la definición de zonas geográficas consideradas seguras, como su casa o lugar de trabajo, con el fin de notificarles a los guardianes pertinentes sus paraderos en caso de que los pacientes se encuentren fuera de algún lugar seguro, y de esta manera proveer asistencia en caso de posibles emergencias.

Esta aplicación se construida con el Framework Ionic y utilizando Apache Cordova como principal tecnología, las cuales son de interés para los autores de este proyecto ya que son tecnologías híbridas que permiten su despliegue en multiplataforma con un único código.

ALZRASTREO se destaca por el uso de un mapa en tiempo real con el uso de la tecnología Google Maps para ubicar a los pacientes en todo momento indicando la rutas seguras que el paciente debe utilizar para su traslado. Esta funcionalidad fue del agrado de los autores y será tomada en cuenta para las funcionalidades a implementarse en dicho proyecto.

4. Dispositivos Móviles

En la actualidad los dispositivos móviles, están causando disrupción, se escucha siempre los términos “Mobile First”, “Responsive Design”, “PWA”. Un dispositivo móvil (Figura 4.1) puede definirse como un computador o micro-ordenador que es lo suficientemente ligero como para ser transportado por una persona, y que dispone de la capacidad de batería suficiente como para poder funcionar de forma autónoma [2], que cumple con las siguientes 4 características:

- Movilidad
- Capacidad de procesamiento y memoria.
- Tamaño reducido.
- Comunicación inalámbrica a Internet.

Existen una gran cantidad de dispositivos móviles como lo son: los teléfonos inteligentes, relojes inteligentes, calculadoras, videoconsolas, tabletas, cámaras digitales, entre otros.

Para el presente trabajo se utilizarán únicamente los *teléfonos inteligentes* y las *tabletas*. Para ello se trabajará con el término generalizado de *dispositivo móvil* para referirse indistintamente a cualquiera de estos dispositivos mencionados anteriormente.

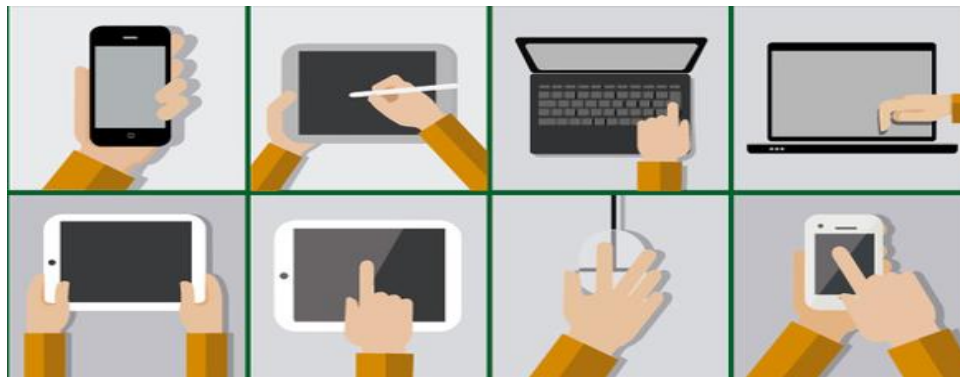


Figura 4.1: Dispositivos móviles

5. Desarrollo de Aplicaciones Móviles

Antes de tocar el desarrollo de aplicaciones móviles, se debe tener una definición sobre qué es una aplicación móvil. Una aplicación móvil es un programa de computadora diseñado y desarrollado para ser utilizado en un dispositivo móvil tales como lo son las tabletas, teléfonos inteligentes, relojes inteligentes, TV inteligentes, entre otros. Dichas aplicaciones pueden utilizar los recursos y funcionalidades que poseen los diferentes dispositivos, se encuentran generalmente disponibles en plataformas de distribuciones de aplicaciones como lo es la App Store y la Google Play Store.

El conjunto de tareas para crear las aplicaciones se les llama “Desarrollo de aplicaciones móviles”, el cual se define como el proceso en el que un software para realizar determinada tarea es desarrollado para dispositivos móviles. Estas aplicaciones pueden venir pre-instaladas en los teléfonos desde su manufactura, ser descargadas por los usuarios desde las distintas plataformas móviles (Android, iOS, Windows Phone), o ser accedidas a través de la web desde un navegador [3].

Escoger un enfoque de desarrollo para una aplicación móvil implica tomar en consideración muchos aspectos tales como el presupuesto, tiempo del proyecto, funcionalidades, tipo de usuarios a los que va dirigido la aplicación, conocimientos del equipo de desarrollo, entre otros. El enfoque seleccionado para el desarrollo del proyecto son las aplicaciones híbridas que presentan en el siguiente punto.

5.1 Aplicaciones Híbridas

Estas aplicaciones son aquellas que combinan tanto el desarrollo nativo como web, por lo tanto son diseñadas en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un framework que permite adaptar la vista web a cualquier vista de un dispositivo móvil. En otras palabras, no son más que una aplicación construida para ser utilizada o implementada en distintos sistemas operativos móviles, tales como, iOS, Android o Windows Phone, evitando la tarea de crear una aplicación para cada sistema operativo. De esta manera, una aplicación híbrida puede ser adaptada a múltiples plataformas móviles sin crear nuevos códigos, pero ajustándose a algunos cambios operacionales para cada uno de ellos [4].

Los programadores escriben una aplicación web con la ayuda de algún Framework (Ionic, NativeScript, Cordova), luego que es finalizada utilizan un empaquetador que cumple la función de intermediario y traduce todas las instrucciones de forma que el SO pueda entenderlas al momento de ejecutar la aplicación.

En la Tabla 5.1 [5] que se muestra a continuación se detallan una serie de ventajas y desventajas sobre el enfoque de híbrido para la aplicaciones móviles.

Ventajas	Desventajas
<ul style="list-style-type: none"> • Es posible distribuirla en las tiendas de iOS y Android. • Instalación nativa pero construida con JavaScript, HTML y CSS. • El mismo código base para múltiples plataformas. • Acceso a parte del hardware del dispositivo. 	<ul style="list-style-type: none"> • Experiencia del usuario más propia de la aplicación web que de la app nativa. • Diseño visual no siempre relacionado con el sistema operativo en el que se muestre.

Tabla 5.1: Ventajas/desventajas de las aplicaciones híbridas

5.2 IONIC

IONIC es un Framework gratuito y de Open Source que sirve de herramienta para el desarrollo de aplicaciones móviles híbridas basadas en HTML5, CSS, JS y optimizado con Angular.

Específicamente Ionic está construido con Angular y SASS (SASS es un preprocesador de CSS), que provee de una serie de componentes con los que permite crear aplicaciones para distintas plataformas móviles.

Ionic permite desarrollar aplicaciones híbridas, esto quiere decir que una aplicación puede ser exportada para que pueda correr en distintas plataformas móviles Android, IOS, Windows Phone u otras, esto es una gran ventaja para los desarrolladores web ya que no necesitan aprender otro lenguaje nativo para el desarrollo de aplicaciones móviles. En cuanto a rendimiento no tiene nada que envidiar a las App nativas, ya que al estar optimizado con Angular posee de una gran capacidad de procesar información. Es fácil de comprender su estructura y su funcionamiento, su estructura está basada en el modelo MVC (Modelo Vista Controlador) [6].

Además, Ionic permite desarrollar y construir en HTML, CSS, y JavaScript, todo esto es interpretado por Cordova que construye la aplicación para la plataforma destino. Por otro lado, Cordova es una plataforma de desarrollo Open Source, para construir aplicaciones nativas usando HTML, CSS y JavaScript. Prácticamente lo que hace es agarrar el código y transformar en lenguaje nativo móvil. Córdoba ofrece muchos plugins de los que se puede disponer, plugins que permite acceder a las propiedades de la cámara, un plugin que permita leer códigos QR, plugins que permiten enviar documentos a imprimir a una impresora y una infinidad de plugin disponibles.

6. Herramientas de Desarrollo

En los capítulos anteriores se ha hablado sobre los dispositivos móviles, su auge en el mercado, los enfoques de desarrollo de aplicaciones móviles (aplicaciones nativas, web e híbridas). Sin embargo en el presente capítulo se tratará una serie de herramientas y tecnologías que son necesarias al momento del desarrollo del software para el actual proyecto.

Se debe comprender y tener un mayor grado de detalle en ciertas tecnologías y herramientas de desarrollo que siguen a continuación. La aplicación móvil a desarrollarse se estructurará bajo una arquitectura Cliente-Servidor, se comunicará mediante el uso de APIs y librerías de terceros a una base de datos, se mantendrá la gestión de sistemas de geolocalización y se utilizarán las tecnologías básicas y fundamentales del desarrollo web como lo son el HTML, CSS y JS.

6.1 Application Programming Interface (API)

La interfaz de programación de aplicaciones (API) es una parte fundamental de la industria informática, se encuentran en la mayoría de arquitecturas de aplicaciones en la actualidad. Se puede definir una API como una interfaz para un componente de software que puede ser invocado a distancia a través de una comunicación de red que usa tecnologías basadas en estándares [7].

Una aspecto a destacar es que existen muchos tipos de API como las de dispositivo o SO y las API remotas. Las API de dispositivo permiten el acceso a datos o funciones proporcionados por un dispositivo o sistema operativo localmente. Mientras que las API remotas son una forma de permitir que un programa de consumo acceda a datos o servicios de un proveedor programa a través de la red de comunicaciones de forma controlada.

A continuación se presentarán dos API's de código abierto que serán utilizadas para el manejo de la geolocalización del proyecto.

6.1.1 Leaflet

Leaflet es una biblioteca de código abierto de JavaScript ampliamente utilizada para crear aplicaciones de mapeo web. Lanzada por primera vez en 2011, es compatible con la mayoría de las plataformas móviles y de escritorio, y es compatible con HTML5 y CSS3 [8]. Además leaflet cuenta con una gran cantidad de plugins que permiten incorporar muchas más funcionalidades e interacciones con los mapas.

Leaflet permite diseñar mapas de manera simple, óptima y usable con pocas líneas de código, posee una documentación amplia y una comunidad muy activa.

6.1.2 Open Route Service

La plataforma Open Route Service ofrece a los desarrolladores varias formas de incorporar las diferentes API de los servicios que proveen a las páginas web. Es de

código abierto con una versión limitada gratis y unas versiones pagas configurables de acuerdo a lo recursos que necesite la aplicación. Entre los productos o API que ofrece se encuentran [9]:

- Direcciones: crea instrucciones detalladas sobre rutas en sus aplicaciones para automóviles, camiones, diferentes perfiles de bicicleta, caminatas.
- Distancia-Tiempo Matriz: utiliza matrices que permiten calcular distancias de muchos a muchos con los tiempos de las rutas entre distancia de una manera óptima.
- Geodecodificación: la geodecodificación transforma una descripción de una ubicación, como el nombre de un lugar, la calle o el código postal, en una descripción normalizada de la ubicación con una geometría de puntos.

6.2 Bases de Datos

Una base de datos es un “almacén” que permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. Una definición más precisa sería “una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular [10].

Luego de conocer el concepto de una base de datos, es necesario saber qué es un Sistema Manejador de Bases de Datos (SMBD), el cual es un conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministra a los distintos tipos de usuarios los medios necesarios para describir y manipular los datos almacenados en la BD, garantizando su seguridad.

6.2.1 Tipos

Existen diferentes tipos pero se resaltarán 2 grupos, las bases de datos relacionales y las bases de datos no relacionales:

- Base de datos relacional: una base de datos relacional es una base de datos que cumple con el modelo relacional, el cual permite establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas [11].
- Base de datos no relacionales: es un sistema de gestión de bases de datos que difieren del modelo clásico de Sistema de Gestión de Bases de Datos Relacionales en aspectos importantes, siendo el más destacado que no usan SQL como lenguaje principal de consultas. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN, ni garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad), y habitualmente escalan bien horizontalmente. Los sistemas NoSQL se denominan a veces "no sólo SQL" para subrayar el hecho de que también pueden soportar lenguajes de consulta de tipo SQL [12].

En el siguiente punto se presentará una herramienta de Google la cual ofrece la Base de Datos que será utilizada en el proyecto.

6.3 Firebase

Firebase es una plataforma en la nube para el desarrollo de aplicaciones web y móviles desarrollada por James Tamplin y Andrew Lee en 2011 y adquirida por Google en 2014 [13].

6.3.1 Características

- Se encuentra disponible para diferentes plataformas (Android, iOS y Web).
- Compila aplicaciones de forma rápida, sin administrar la infraestructura.
- Respaldo de Google y confianza de aplicaciones reconocidas.
- Posee una consola con los productos que ofrece, los cuales funcionan perfectamente por sí solos pero funcionan aún mejor juntos.
- Tiene disponible una buena documentación, incluyendo ejemplos o tutoriales.
- No es totalmente gratis, existe un plan gratuito, pero presenta una serie de limitaciones en espacio de almacenamiento y conexiones simultaneas.

6.3.2 Firebase Autenticacion

La mayoría de las aplicaciones necesitan identificar a los usuarios, por ello, este producto de Firebase busca facilitar la creación de sistemas de autenticación seguros, a la vez que mejora la experiencia de incorporación y acceso para los usuarios finales. Proporciona una solución de identidad de extremo a extremo, compatible con cuentas de correo electrónico y contraseñas, autenticación telefónica, Google, Twitter, Facebook, acceso a través de GitHub, entre otros [14].

Permite una implementación muy rápida. Se puede configurar todo el sistema de autenticación de una aplicación en menos de 10 líneas de código.

6.3.3 Cloud Firestore

Cloud Firestore es una base de datos de documentos NoSQL flexible, escalable y en la nube que permite almacenar, sincronizar y consultar fácilmente datos para aplicaciones webs y móviles [15].

Posee una estructura de datos basada en colecciones y documentos. Se pueden crear jerarquías para almacenar datos relacionados y recuperar fácilmente los datos que se necesitan mediante consultas expresivas.

Permite sincronizar automáticamente los datos de una aplicación en distintos dispositivos a través de agentes de escucha en tiempo real. Notifica los cambios en los datos a medida que ocurren para que se puedan crear experiencias colaborativas y aplicaciones en tiempo real fácilmente.

Cloud Firestore también se integra con Firebase Authentication para ofrecer una autenticación de usuario intuitiva y sencilla. Gracias a lo anterior se puede restringir el acceso a los datos en función de los datos de identidad del usuario.

6.3.4 Cloud Functions

Este producto de Firebase permite ejecutar código de backend para dispositivos móviles sin administrar servidores. Cloud Functions son funciones de JavaScript con un propósito único que se ejecutan en un entorno de Node.js seguro y administrado, solo se ejecutan cuando se emite un evento específico bajo observación. Permite crear funciones que se activan con productos de Firebase, como cambios en colecciones en Cloud Firestore, accesos de usuarios nuevos a través de Auth o directamente con solicitudes HTTPS [16].

Cloud Functions mantienen la privacidad y la seguridad de la lógica de programación. En muchos casos, la lógica de la aplicación se controla mejor en el servidor con el fin de evitar alteraciones del lado del cliente, por ello las Cloud Functions están completamente aisladas del cliente. Por otro lado, solo se necesita del comando: *“firebase deploy...”* para poder implementar el código de la función en los servidores de Firebase.

6.3.5 Cloud Storage

Cloud Storage se creó para programadores de aplicaciones que necesitan almacenar y publicar contenido generado por los usuarios, como fotos o vídeos. Es un producto de Firebase que ofrece un servicio de almacenamiento de objetos potente, simple y rentable construido para la escala de Google [17]. Los SDK de Firebase para Cloud Storage agregan la seguridad de Google a las operaciones de carga y descarga de archivos para las aplicaciones de Firebase, sin importa la calidad de la red. Se pueden utilizar dichos SDK para almacenar imágenes, audio, vídeo y otros tipos de contenido generado por el usuario. En el servidor, se puede usar *Google Cloud Storage* para acceder a los archivos almacenados.

7. Geolocalización

La geolocalización es un término esencial para el desarrollo del proyecto y se define como un conjunto de técnicas que permiten determinar la posición geográfica de un elemento (un ordenador, un teléfono móvil o cualquier dispositivo capaz de ser detectado) en el mundo real y hacer uso de esa información [18]. Esta tecnología requiere de la perfecta sincronización entre hardware y software, es necesario un dispositivo con GPS o conexión a Internet y un software que permita hacer uso de ellos.

La posición que se obtiene es expresada mediante coordenadas geográficas (latitud y longitud). La latitud mide el ángulo entre cualquier punto y el ecuador. Las líneas de latitud se llaman paralelos y son círculos paralelos al ecuador en la superficie de la Tierra. Por otro lado, la longitud se encarga de medir el ángulo a lo largo del ecuador desde cualquier punto de la Tierra. Se acepta que Greenwich en Londres es la longitud 0 en la mayoría de las sociedades modernas [19].

7.1 Geolocalización vs Georeferenciación

Ya teniendo claro el concepto de geolocalización es necesario señalar la diferencia que existe entre geolocalización y georeferenciación para no cometer el error de tratar ambos conceptos por igual aunque a priori puedan parecer lo mismo.

La georeferenciación es un proceso por el cual se dota de un sistema de referencia de coordenadas terreno a una imagen digital que originariamente se encuentra en coordenadas píxel, es decir, es una técnica de posicionamiento espacial de una entidad en una localización geográfica única. Aplicando lo anterior a un ejemplo, la georeferenciación de una parcela es la técnica utilizada para situar con exactitud los límites del terreno. En la Tabla 7.1 se puede apreciar las diferencias que existen entre estos dos términos.

Geolocalización	Georeferenciación
Localiza un objeto/dispositivo en el mapa y en tiempo real.	Sitúa puntos concretos en un mapa.
Permite ofrecer diferentes funciones a partir de la ubicación exacta que se obtiene.	Algunas aplicaciones permiten obtener vistas aéreas de las ubicaciones y navegar a través de ellas.
Google Maps: permite localizar un dispositivo, es decir, acceder a la ubicación exacta y ofrecer las diferentes funciones de la aplicación a partir de esto.	Google Earth: es un sistema de georeferenciación que permite situar en el mapa puntos concretos de la geografía.

Tabla 7.1: Geolocalización vs Georeferenciación

8. Hardware para Prototipos

A continuación se presentarán una serie de dispositivos de hardware con los cuales se pueden realizar prototipos mediante el acoplamiento de los mismos, esto con el fin de aprovechar las funcionalidades de cada uno de los dispositivos y utilizarlas para el desarrollo del proyecto. Se mostrará un tipo de placa o tarjeta de desarrollo y distintos módulos (GPS y GSM) los cuales pueden realizar la conexión y comunicación con dicha placa.

8.1 Tarjetas o Placas de Desarrollo

Las tarjetas o placas de desarrollo de hardware, son útiles al realizar cualquier tipo de sistemas, proyectos académicos como también proyectos privados. Cada tarjeta cuenta con su propio lenguaje de programación o un sistema operativo. La tarjeta de desarrollo seleccionada para el proyecto es la Arduino UNO.

8.1.1 Arduino

Arduino es una plataforma de código abierto basada en microcontrolador ATmega los cuales destacan por su potencia con 4 a 256 KB de memoria flash programable, encapsulados de 28 a 100 pines, conjunto de instrucciones extendido (multiplicación y direccionamiento de programas mayores) y amplio conjunto de periféricos. Existen una gran variedad de modelos Arduino, a continuación (Tabla 8.1) se presentan los 3 con mayor demanda en el mercado para realizar una comparativa entre ellos.

Características	Arduino UNO	Arduino Leonardo	Arduino YUN
Microcontrolador	ATmega328P	ATmega32u4	ATmega32u4
Pines digitales	14	20	20
Pines analógicos	6	12	12
Memoria flash	32KB	32KB	32KB

Tabla 8.1: Especificaciones Arduino UNO, Leonardo y YUN

Las placas Arduino (Figura 8.1) pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, activar un motor, encender un LED y publicar algo en línea. Puede decirle a su tablero qué hacer enviando un conjunto de instrucciones al microcontrolador en el tablero. Para hacerlo, se utiliza el lenguaje de programación Arduino y el software Arduino (IDE) [20].

Arduino se puede utilizar para desarrollar elementos autónomos, o bien conectarse a otros dispositivos o interactuar con otros programas, para interactuar tanto con el hardware como con el software. Sirve tanto para controlar un elemento, pongamos por ejemplo un automóvil puede ser monitoreado con una placa arduino mediante unos módulos o sensores GPS que se incorporan a dicha placa y así transmitir las coordenadas de posicionamiento de dicho automóvil en intervalos de tiempo constantes.

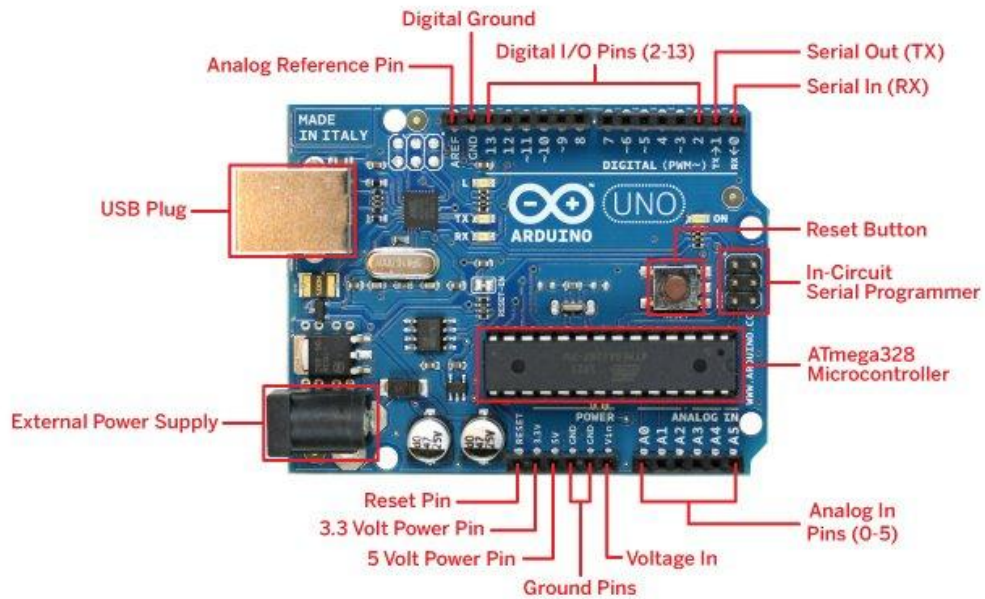


Figura 8.1: Arduino Uno

8.2 Módulo GPS

El módulo o receptor GPS es aquel hardware que permite obtener la posición geográfica longitud y latitud con una precisión de unos metros, usando la tecnología GPS. Como caso de estudio se tomará el módulo Adafruit Ultimate GPS Breakout V3 (Figura 8.2) en su versión 3.

Este dispositivo se basa en el chipset MTK3339, un módulo de GPS de alta calidad que puede rastrear hasta 22 satélites en 66 canales, tiene un excelente receptor de alta sensibilidad (seguimiento de -165 dB) y una antena incorporada. Puede hacer hasta 10 actualizaciones de ubicación por segundo para el registro o seguimiento de alta velocidad y alta sensibilidad. El consumo de energía es increíblemente bajo, solo 20 mA durante la navegación [21].

Este módulo se puede conectar de forma fácil a una placa de desarrollo Arduino o Intel Galileo. Los pines a utilizar son los pines de data serial TX y RX, y los pines de potencia VIN y GND, los cuales se explican a continuación:

- TX: pin que transmite datos desde el módulo GPS al microcontrolador o computadora al que esté conectado.
- RX: pin que puede ser usado para enviar datos al GPS.
- VIN: entrada de poder o alimentación, debe ser conectada a 3-5VDC.
- GND: tierra de potencia y señal.

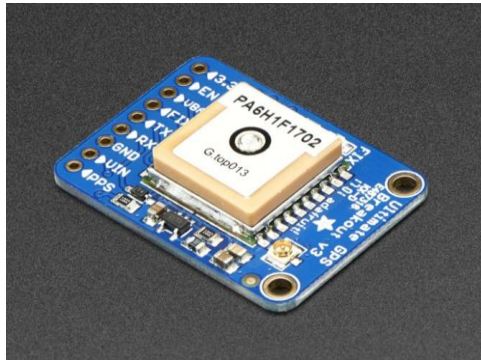


Figura 8.2: Adafruit Ultimate GPS Breakout V3

8.3 Módulo GSM

El GSM (Global System for Mobile Communications) es el sistema de comunicaciones que más se utiliza en teléfonos móviles. La primera funcionalidad de esta tecnología es la transmisión de voz, pero también permite la transmisión de datos (SMS, Internet). En Europa, el estándar GSM usa las bandas de frecuencia de 900MHz y 1800MHz. Sin embargo, en los Estados Unidos se usa la frecuencia de 1900MHz. El estándar GSM soporta un tráfico máximo de 9.6kbps la cual es muy baja para la transmisión de paquetes de gran tamaño. Esto hace pensar en el estándar GPRS.

El GPRS (General Packet Radio Service) es una extensión del GSM basada en la transmisión por paquetes que ofrece un servicio más eficiente para las comunicaciones de datos, especialmente en el caso del acceso a Internet. La velocidad máxima en teoría del GPRS es de 171 kbps, aunque la práctica es bastante más pequeña pero superior a la del GSM.

Caso de estudio Módulo GSM SIM800L

El módulo SIM800L (Figura 8.3) es un módulo GSM/GPRS Quad-Band que trabaja con frecuencias de 850/900/1800/1900 MHz. Este módulo de telefonía móvil permite añadir voz, SMS y datos (TCP/IP, HTTP, etc...) al proyecto que se esté realizando [22].



Figura 8.3: Prototipo GSM SIM800L

8.4 SMS Gateway

Para poder hacer una definición más concisa de un SMS Gateway, se debe conocer las definiciones de SMS y Gateway, por lo cual se procederá a definir ambos ítems:

- **SMS:** el servicio de mensajes cortos o servicio de mensajes simples, más conocido como SMS (por las siglas del inglés Short Message Service), es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos, conocidos como mensajes de texto, entre teléfonos móviles.
- **Gateway:** la pasarela (en inglés gateway) o puerta de enlace es el dispositivo que actúa de interfaz de conexión entre aparatos o dispositivos, y también posibilita compartir recursos entre dos o más computadoras. Su propósito es traducir la información del protocolo utilizado en una red inicial, al protocolo usado en la red de destino.

Ahora que se conocen dichos conceptos, se puede definir pasarela SMS o puerta de enlace por SMS (del inglés SMS Gateway) como un medio de envío o recibo de mensajes de texto (SMS) usando una red de telecomunicaciones. Esta puerta de enlace es usada por entidades y organizaciones para enviar notificaciones masivas a teléfonos móviles que no están conectados a Internet a través de un software.

En la Figura 8.4 se muestra un esquema de cómo es la interacción de una aplicación con un sistema de SMS Gateway para transmisión de mensajes a teléfonos. Una aplicación que se encuentra ejecutándose en la computadora realiza el envío o la recepción de mensajes a través de una puerta de enlace (SMS Gateway), el cual a su vez se comunica con la red de operadoras telefónicas para el envío o recepción de mensajes en el dispositivo móvil.



Figura 8.4: Esquema de flujo de un SMS Gateway

9. Metodologías Ágiles de Desarrollo de Software

Las metodologías ágiles son una serie de metodologías de desarrollo de software basadas en el desarrollo iterativo e incremental. Una metodología ágil proporciona un marco para apoyar el diseño, desarrollo y pruebas de soluciones de software a lo largo de su ciclo de vida. Este tipo de metodologías promueven respuestas rápidas y flexibles al cambio mediante la promoción de la planificación adaptativa, la identificación de requisitos colaborativos, así como el desarrollo gradual de soluciones. A continuación se presenta la metodología ágil que será utilizada para el desarrollo del proyecto.

9.1 Scrum

La metodología scrum nace del análisis de cómo se desarrollaban los productos principales de las grandes compañías de Japón, Ikuro Nonaka e Hirotaka Takeuchi son los investigadores que deciden crear dicha metodología de acuerdo a los resultados obtenidos.

Scrum define un conjunto de prácticas y roles que se pueden tomar como un punto de inicio al definir el proceso de desarrollo que se ejecutará durante un proyecto. Scrum se basa en los siguientes principios ágiles:

- Colaboración estrecha y continua con el cliente.
- Predisposición y respuesta al cambio.
- Desarrollo incremental con entregas funcionales frecuentes.
- Comunicación verbal directa entre los implicados en el proyecto.
- Motivación y responsabilidad de los equipos por la auto-gestión, auto-organización y compromiso.
- Simplicidad, no se utilizan artefactos innecesarios en la gestión del proyecto.

Scrum define unos roles de usuarios para mantener la organización y el desempeño del proyecto. Los roles principales son:

- Propietario del producto: Representa a todos los interesados en el producto final.
- Equipo: Responsable de transformar la pila del sprint en un incremento de la funcionalidad de software.
- Scrum Manager: Responsable del proceso de Scrum.
- Stakeholders: Personas que hacen posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su desarrollo.

Scrum define el desarrollo mediante periodos de tiempo llamados Sprint. Se puede definir un Sprint como el periodo en el cual se lleva a cabo un trabajo en sí. Es recomendable que la duración de los sprint sea constante y definida por el equipo de trabajo de acuerdo a su experiencia [23].

De acuerdo la Figura 9.1 se observa como es el flujo del Scrum, el cual cuenta con:

- Product Backlog: lista ordenada de todo lo que podría necesitarse en el producto.
- Sprint Backlog: Pila de actividades que deben realizarse.
- Sprint Backlog correspondiente al equipo: Son las actividades que el equipo debe desarrollar en dicho sprint.

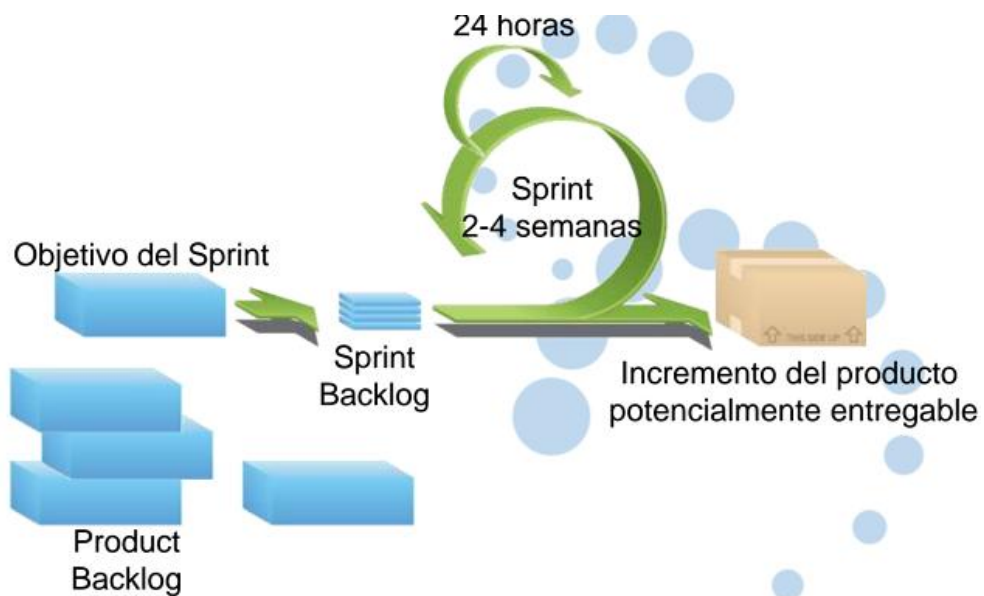


Figura 9.1: Flujo Metodología Scrum

Se espera que al final del sprint exista una nueva funcionalidad demostrable a los interesados, la cual es presentada en la reunión de retrospectiva, en ella se discuten y aclaran todas las actividades que debieron ser realizadas en el sprint cual es su estatus actual y en caso de no ser completadas, se explican qué sucedió al respecto. La reunión de retrospectiva se realiza una vez finalizado cada sprint y tiene un promedio de duración de 4 a 8 hrs.

10. Marco Metodológico

Entendiendo que para lograr el desarrollo de una solución tecnológica de la forma estructurada, planificada y controlada posible es necesario tener y seguir una metodología de desarrollo de software. La metodología que fue implementada en la elaboración de este proyecto se describe en este capítulo, así como el conjunto de herramientas y tecnologías que fueron utilizadas para alcanzar los objetivos planteados que se describen en el Capítulo 2.

10.1 Adaptación de la Metodología de Desarrollo

Para el proceso de desarrollo de la aplicación móvil y el componente de hardware se implementó una metodología de desarrollo ágil llamada **Scrum**. Esta metodología es incremental e iterativa y se caracteriza por ser muy adaptable, siendo este último punto importante de recalcar, ya que existen aspectos definidos sobre esta metodología de desarrollo de software que no se adaptaron a las características y necesidades de los autores en éste proyecto por lo que fueron modificados u omitidos.

Para la gestión y monitoreo de todas las actividades relacionadas con la metodología, se utilizó de principio a fin una aplicación web llamada **Trello**, la cual es una herramienta flexible y con muchas características que permiten estructurar las actividades, asignar tiempos de desarrollo y responsables, para así mantener un control de las actividades necesarias para el cumplimiento del proyecto.

Los autores decidieron por cada Sprint crear un Board en el cual se colocarán todas las actividades que deben realizarse en dicho sprint. Al final de cada Sprint se realizó la reunión respectiva donde se observó el avance de las actividades y si se cumplieron con los objetivos.

Para la adaptación de la metodología los autores Josué Sánchez y José Yvimas asumieron los roles de Product Owner, Scrum Master y Scrum Team. Luego de definir dichos roles se procederá a detallar los artefactos y herramientas que servirán de apoyo para el desarrollo del proyecto.

Para el proyecto se utilizarán herramientas que son propuestas por la metodología Scrum como los Product Backlog List y los Sprint Backlog List y se remplazarán otras como las historias de usuario por diagramas UML y diagramas de navegación.

El Product Backlog List será implementado como un listado de requerimientos funcionales y no funcionales que se utilizarán para el desarrollo de la aplicación móvil.

El Sprint Backlog List será adaptado como una lista de tareas a realizar para cumplir con los requerimientos establecidos. Las tareas son establecidas y asignadas por el Scrum Team, el Scrum Master y el Product Owner.

Como ejemplo en la Figura 10.1 se puede visualizar una imagen tomada desde la aplicación Trello que refleja cómo fue llevado a cabo en el proyecto los distintos Sprint Backlog List.

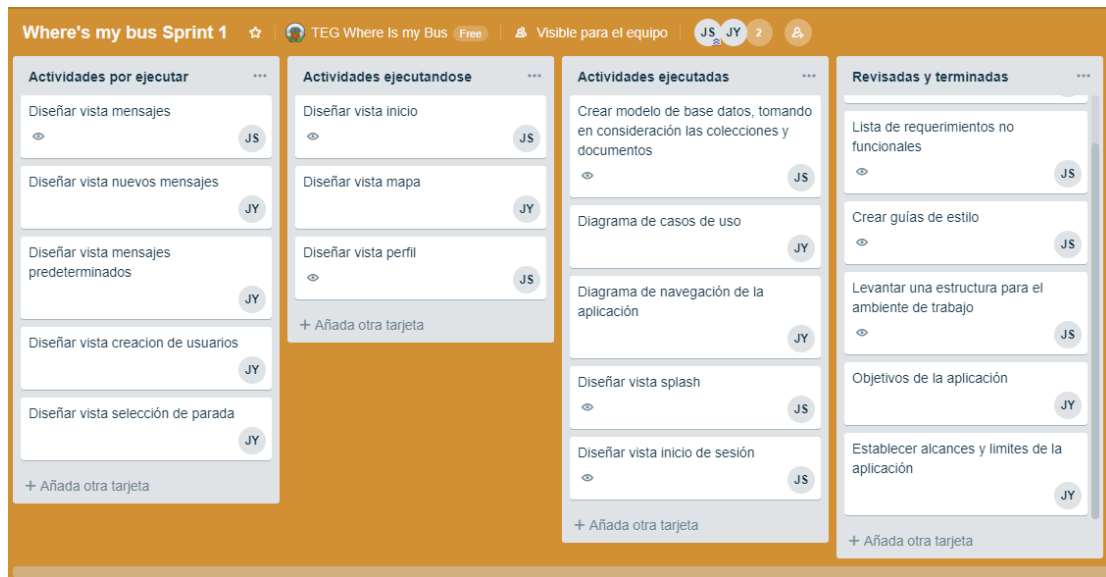


Figura 10.1: Tablero que refleja un Sprint Backlog List del proyecto en el Sprint 1

En el caso de los diagramas UML y de navegación serán explicados con mayor detalle en el capítulo de Marco Aplicativo.

10.1.1 Eventos

Los eventos de la metodología Scrum que serán contemplados en el proyecto fueron:

- **Sprint:** para el desarrollo del proyecto se estableció que la duración de los sprints sería de 2 semanas cada uno, entre los días lunes y viernes con un tiempo de trabajo estimado en 4 horas diarias mínimas por parte de cada uno de los autores.
- **Sprint Planning Meeting:** se llevarán a cabo los días domingo, justo el día antes de comenzar un nuevo sprint por medio de herramientas informáticas que permiten la comunicación a distancia a través de internet como Skype que es una aplicación que permite la comunicación por voz, texto y vídeo.
- **Daily Scrum:** durante el transcurso del proyecto los Dailys Meeting serán registrados y almacenados en Google Keep, el cual es una libreta virtual (Figura 10.2).

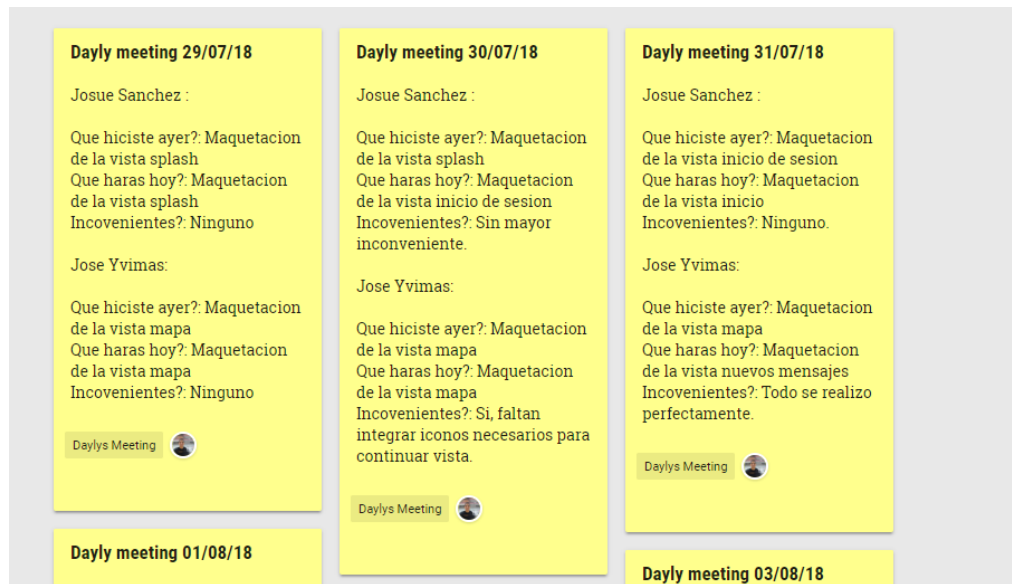


Figura 10.2: Tablero de reportes de Dailys Meeting en Google Keep

- **Sprint Review Meeting:** al igual que los Sprint Planning Meeting se llevarán a cabo los días domingo por medio de la ya mencionada herramienta informática Skype.

10.1.2 Tecnologías Utilizadas

En la presente sección se explicarán brevemente las tecnologías y herramientas de desarrollo utilizadas para llevar a cabo las aplicaciones en las que se centra la presente investigación:

- **Arduino IDE:** es un entorno de desarrollo y en él se realiza la programación de cada una de las placas de Arduino. También es compatible con placas Intel Galileo.
- **Cordova:** una plataforma que permite construir aplicaciones nativas usando HTML, CSS y JavaScript. Transforma el código en lenguaje nativo móvil. Además ofrece muchos plugins de los que se puede disponer. Fue utilizada para construir la aplicación a las plataformas Android y iOS.
- **Draw.io:** una plataforma que permite construir diagramas UML con facilidad y su exportación en distintos formatos.
- **Firebase Authentication:** producto de Firebase que permite la fácil integración de mecanismos de autenticación de usuarios. Utilizado en el proyecto para el registro y autenticación de usuarios.
- **Firebase Cloud Functions:** es un producto de Firebase que permite ejecutar código backend, sin necesidad de tener que poseer una infraestructura. Se

utilizó como una plataforma en la nube en donde se reciben las coordenadas del transporte para su procesamiento y almacenamiento.

- **Firestore Cloud Storage:** servicio que permite subir y descargar archivos de forma segura para las aplicaciones de Firestore, independientemente de la calidad de la red. Puede ser usado para almacenar imágenes audio, vídeo y otro contenido generado por el usuario.
- **Firestore:** es una base de datos NoSQL orientada a colecciones y documentos con la característica de agentes de escucha en tiempo real. Fue utilizada esta tecnología como la base de datos del proyecto.
- **Git:** software de control de versiones que permite el mantenimiento de aplicaciones cuando éstas tienen una gran cantidad de archivos de código fuente. Es ampliamente usado a nivel mundial por lo eficiente y confiable que es la herramienta. Utilizado en el desarrollo de la aplicación móvil.
- **Ionic:** framework gratuito de open source que sirve de herramienta para el desarrollo de aplicaciones móviles híbridas basadas en HTML, CSS y JS optimizado con Angular. Utilizado en el proyecto para desarrollar la aplicación móvil híbrida.
- **JavaScript:** lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Es el lenguaje principal utilizado por el framework Ionic que se utilizó en el desarrollo del proyecto.
- **Leaflet:** librería de código abierto de JavaScript ampliamente utilizada para crear aplicaciones de mapeo web. Es compatible con la mayoría de plataformas móviles, de escritorio y las tecnologías web. Utilizada en el proyecto como librería para la creación y renderización de mapas.
- **Módulo Adafruit Ultimate GPS Breakout:** es un módulo de GPS de alta calidad que puede rastrear hasta 22 satélites en 66 canales, tiene un excelente receptor de alta sensibilidad (seguimiento de -165 dB) y una antena incorporada. Fue el módulo GPS utilizado para la obtención de las coordenadas del transporte escolar.
- **Módulo GSM SIM800L:** es un módulo GSM/GPRS Quad-Band que trabaja con frecuencias de 850/900/1800/1900 MHz. Este módulo de telefonía móvil permite añadir voz, SMS y datos (TCP/IP, HTTP). El cual se utilizó para la transmisión de coordenadas desde el prototipo de hardware hacia el servidor en la nube.
- **Open Route Service (ORS):** es una plataforma de código abierto que ofrece múltiples servicios de geolocalización a los desarrolladores para su integración con aplicaciones web y móviles. Se utilizó el API de direcciones y

distancia-tiempo matriz para obtener estimados de tiempo entre localizaciones.

- **SMSSync:** es una aplicación para dispositivos móviles que permite crear una puerta de enlace SMS (SMS Gateway) para el envío de los mensajes entrantes (SMS) hacia una URL configurada (servidor web).
- **Tarjeta Arduino UNO:** plataforma de código abierto basada en microcontrolador ATmega, con memoria flash de 256 KB, 14 pines digitales, 6 pines analógicos y amplio conjunto de periféricos.
- **Trello:** software de administración de proyectos el cual fue utilizado como herramienta para la estructuración de responsabilidades y tareas con la metodología Scrum.
- **UML:** es el lenguaje unificado de modelado que proporciona una serie de diagrama que permiten describir todas las partes de un sistema, la interacción con los usuarios y otros sistemas.
- **Visual Studio Code:** editor de código fuente disponible para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Fue el IDE utilizado para el desarrollo de la aplicación móvil.

11. Marco Aplicativo

En este capítulo se explicará el diseño de la solución a través de los sprints definidos en base a la adaptación de la metodología Scrum, descrita en el capítulo anterior. Las actividades de cada sprint se definieron en base a las necesidades que surgían a medida que se avanzaba en el desarrollo del proyecto y en base a un estimado de horas que llevaría realizar cada una de esas actividades hasta cubrir todas las actividades que conforman el sprint.

11.1 Arquitectura de la Solución

Antes de iniciar con la explicación del desarrollo del proyecto, se presentará a continuación en la Figura 11.1 una arquitectura general de la solución tecnológica para así tener un entendimiento de cada uno de los módulos que componen este trabajo.

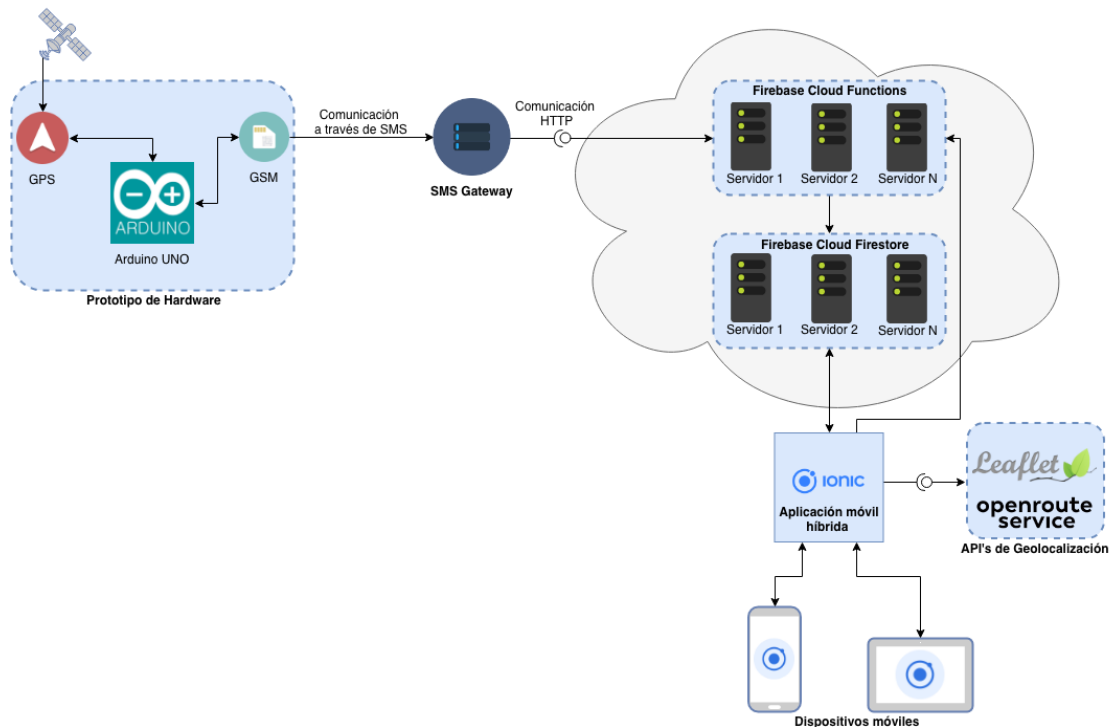


Figura 11.1: Esquema de Diseño de Solución

- Aplicación móvil híbrida: desarrollada en Ionic, contiene toda la lógica que se requiere para mostrar las interfaces que cumplan con todas las funcionalidades esperadas. Se comunica con la base de datos Cloud Firestore utilizando sus operaciones CRUD (Create, Read, Update, Delete) en las diferentes colecciones. Además, utiliza funciones de Leaflet para el manejo de mapas y Open Route Service (ORS) para las rutas y cálculos de tiempo desde un punto a otro.
- Módulo de contingencia: el cual podrá ser activado por el chofer de la unidad en caso de presentarse un inconveniente con el prototipo de hardware. Este módulo utilizará el dispositivo móvil del chofer para la transmisión de coordenadas hacia el servidor en la nube.

11.2 Planificación de las Actividades del Proyecto

El desarrollo de la solución tecnológica planteada en el presente trabajo de investigación se realizó en un total de 6 Sprints. Para entender con mayor claridad como fue el proceso de desarrollo y las tareas que se propusieron y se realizaron en cada iteración, a continuación, se describe la planificación de cada sprint para cada autor que resultaba de las reuniones conocida como Sprint Planning Meeting de la metodología Scrum que se llevaban a cabo antes de comenzar cada sprint.

Sprint	Sprint 1
Responsable	Josue Sánchez
Fecha	1 de Julio a 13 de Julio
Actividades	<p>Entregar los siguientes artefactos:</p> <ul style="list-style-type: none"> ● Lista de requerimientos funcionales y no funcionales. ● Guías de estilo. ● Modelo de base de datos. ● Estructura del ambiente de trabajo. <p>Se deben entregar los siguientes diseños:</p> <ul style="list-style-type: none"> ● Logo. ● Vista Splash. ● Vista inicio de sesión. ● Vista de Inicio. ● Vista perfil. ● Vista mensajes.

Tabla 11.1: Sprint 1 – Josue Sánchez

Sprint	Sprint 1
Responsable	José Yvimas
Fecha	1 de Julio a 13 de Julio
Actividades	<p>Entregar los siguientes artefactos:</p> <ul style="list-style-type: none"> • Objetivos. • Establecer el alcance y los límites permitidos de recursos. • Diagrama de casos de uso. • Diagrama de navegación de la aplicación. • Establecer comportamiento de comunicación con Hardware. <p>Se deben entregar los siguientes diseños:</p> <ul style="list-style-type: none"> • Vista mapa. • Vista nuevos mensajes. • Vista mensajes predeterminados. • Vista estado del estudiante. • Vista de creación de usuarios. • Vista selección de parada.

Tabla 11.2: Sprint 1 – José Yvimas

Sprint	Sprint 2
Responsable	Josue Sánchez
Fecha	15 de Julio a 27 de Julio
Actividades	<p>Se debe revisar y modificar las vistas:</p> <ul style="list-style-type: none"> • Vista mapa. • Vista nuevos mensajes. • Vista mensajes predeterminados. • Vista estado del estudiante. • Vista de creación de usuarios. • Vista selección de parada. <p>Se debe instalar y configurar el ambiente de trabajo:</p> <ul style="list-style-type: none"> • Visual Studio Code. • Git. • Firebase. • Ionic. • Android Studio. • Cordova.

Tabla 11.3: Sprint 2 – Josue Sánchez

Sprint	Sprint 2
Responsable	José Yvimas
Fecha	15 de Julio a 27 de Julio
Actividades	<p>Se debe revisar y modificar las vistas:</p> <ul style="list-style-type: none"> • Logo. • Vista Splash. • Vista inicio de sesión. • Vista de Inicio. • Vista perfil. • Vista mensajes. <p>Mensajes Se debe instalar y configurar el ambiente de trabajo:</p> <ul style="list-style-type: none"> • Visual Studio Code. • Git. • Firebase. • Ionic. • Xcode. • Arduino IDE.

Tabla 11.4: Sprint 2 – José Yvimas

Sprint	Sprint 3
Responsable	Josue Sánchez
Fecha	29 de Julio a 10 de Agosto
Actividades	<p>Maquetar las siguientes vistas:</p> <ul style="list-style-type: none"> • Vista Splash. • Vista inicio de sesión. • Vista de Inicio. • Vista perfil. • Vista mensajes. <p>Incluir la iconografía, paleta de colores y tipografía seleccionada al proyecto. Adicionalmente se integrará la comunicación con la nube de Firebase.</p>

Tabla 11.5: Sprint 3 – Josue Sánchez

Sprint	Sprint 3
Responsable	José Yvimas
Fecha	29 de Julio a 10 de Agosto
Actividades	<p>Maquetar las siguientes vistas:</p> <ul style="list-style-type: none"> • Vista mapa. • Vista nuevos mensajes. • Vista mensajes predeterminados. • Vista estado del estudiante. • Vista de creación de usuarios. • Vista Selección de parada. <p>Incluir la iconografía, paleta de colores y tipografía seleccionada al proyecto. Adicionalmente se integrará la comunicación con la nube de Firebase</p>

Tabla 11.6: Sprint 3 – José Yvimas

Sprint	Sprint 4
Responsable	Josue Sánchez
Fecha	12 de Agosto a 24 de Agosto
Actividades	<p>Se deben integrar e implementar las funciones necesarias para las vistas:</p> <ul style="list-style-type: none"> • Vista Splash. • Vista inicio de sesión. • Vista de Inicio. • Vista perfil. • Vista mensajes. <p>Desarrollar modulo para comunicación con las API's de OpenRouteService y Leaflet.</p>

Tabla 11.7: Sprint 4 – Josue Sánchez

Sprint	Sprint 4
Responsable	José Yvimas
Fecha	12 de Agosto a 24 de Agosto
Actividades	<p>Se deben integrar e implementar las funciones necesarias para las vistas:</p> <ul style="list-style-type: none"> • Vista mapa. • Vista nuevos mensajes. • Vista mensajes predeterminados. • Vista estado del estudiante. • Vista de creación de usuarios. • Vista Selección de parada. <p>Desarrollar modulo para la obtención de datos a través de la Nube de Cloud Functions.</p>

Tabla 11.8: Sprint 4 – José Yvimas

Sprint	Sprint 5
Responsable	Josue Sánchez
Fecha	26 de Agosto a 7 de Septiembre
Actividades	<p>Desarrollar software para el prototipo de hardware (Obtención de datos a través de modulo GPS).</p> <p>Desarrollo de comunicación con endpoint expuesto en la nube de Firebase (Cloud Functions).</p> <p>Almacenamiento en base de datos de coordenadas recibidas por el hardware.</p> <p>Adicionalmente se debe realizar revisión y ajustar la aplicación móvil en caso de encontrar desperfectos.</p>

Tabla 11.9: Sprint 5 – Josue Sánchez

Sprint	Sprint 5
Responsable	José Yvimas
Fecha	26 de Agosto a 7 de Septiembre
Actividades	<p>Desarrollar software para el prototipo de hardware (Envió de datos a través de módulo GSM).</p> <p>Desarrollo de comunicación con endpoint expuesto en la nube de Firebase (Cloud Functions).</p> <p>Almacenamiento en base de datos de coordenadas recibidas por el hardware.</p> <p>Adicionalmente se debe realizar revisión y ajustar la aplicación móvil en caso de encontrar desperfectos.</p>

Tabla 11.10: Sprint 5 – José Yvimas

Sprint	Sprint 6
Responsable	Josue Sánchez
Fecha	9 de Septiembre a 21 de Septiembre
Actividades	<p>Realizar despliegue de la aplicación en dispositivos Android y en emuladores de dispositivos Android.</p> <p>Realizar pruebas de la funcionalidad de localización del transporte.</p> <p>Realizar pruebas de aceptación de la aplicación.</p> <p>Afinar y corregir módulo de geolocalización.</p> <p>Corregir bugs de la aplicación móvil.</p>

Tabla 11.11: Sprint 6– Josue Sánchez

Sprint	Sprint 6
Responsable	José Yvimas
Fecha	9 de Septiembre a 21 de Septiembre
Actividades	<p>Realizar despliegue de la aplicación en dispositivos iOS y en emuladores de dispositivos iOS.</p> <p>Realizar pruebas de la funcionalidad de localización del transporte.</p> <p>Realizar pruebas de aceptación de la aplicación</p> <p>Afinar y corregir módulo de geolocalización.</p> <p>Corregir bugs de la aplicación móvil.</p>

Tabla 11.12: Sprint 6 – José Yvimas

11.3 Análisis General de la Aplicación Móvil

La aplicación móvil presenta cada una de las funcionalidades usables por los distintos usuarios de la misma, fue desarrollada haciendo uso del framework Ionic en su versión 4, el cual como se ha mencionado anteriormente permite crear aplicaciones móviles híbridas multiplataforma utilizando HTML5, CSS, JS y Cordova como base. En el caso de la persistencia de los datos de la aplicación se usó el SMBD Cloud Firestore de Firebase.

Cada autor trabajaba en el desarrollo de la aplicación de forma local haciendo uso de la herramienta Git para el control de versiones y el trabajo colaborativo sobre el código de la aplicación. Debido al uso de distintos productos de Firebase se necesitaba hacer uso de conexión a Internet en todo momento para poder observar el correcto funcionamiento de la aplicación. Posteriormente se efectuó la construcción de la aplicación mediante Cordova para ser usada y probada en dispositivos Android y iOS, ya que al inicio se probaba la aplicación mediante el navegador web y emuladores de plataformas móviles.

Las funcionalidades que ofrece esta aplicación dependen del rol del usuario que la use. Existen tres tipos de usuario: administradores, representantes y choferes. De forma general, si es el usuario es un administrador puede gestionar perfiles de representantes, estudiantes y choferes, así como también poder leer mensaje/notificaciones que envíe el chofer. Si el usuario es un chofer solo puede

enviar mensajes y cambiar el estado que poseen los estudiantes en la unidad. Por último, si el usuario es un representante puede leer mensajes del chofer, consultar la ubicación del autobús y el tiempo estimado de llegada a su parada, ver el estado de su representado en el autobús y ver su perfil.

11.3.1 Artefactos y Diseño de Interfaces de la Aplicación Móvil

El análisis, diseño y desarrollo de la aplicación móvil transcurre entre el Sprint 1 hasta el Sprint 4. Dentro de la metodología Scrum no se expresa que debe usarse UML, sin embargo es ampliamente usado dentro de los proyectos de desarrollo de software orientado a objetos [24].

Durante dicho Sprint se llevó a cabo la evaluación y análisis que permitió la selección de una serie de artefactos que si bien no forman parte propia de la metodología Scrum son bastante comunes en otras metodologías de la ingeniería de software, y estos son de gran utilidad durante la fase de desarrollo. Además de esto, los autores poseen conocimientos previos durante lo largo de la carrera sobre gran variedad de diagramas UML. También se realizaron los diseños preliminares de las interfaces de usuario que tendría la aplicación móvil.

Como primer paso se determinaron los principales requerimientos funcionales y no funcionales para lograr la construcción de la solución móvil. Donde dichos requerimientos fueron tomados por los autores del proyecto como la Product Backlog List, artefacto que toma en cuenta la Scrum. Los requerimientos son los que se muestran a continuación:

- Requerimientos Funcionales:
 1. Todos los usuarios de la aplicación podrán iniciar sesión con sus datos en el sistema.
 2. Los usuarios con rol de administrador podrán crear la información de cada uno de los estudiantes.
 3. Los administradores podrán crear usuarios con rol de representantes y choferes.
 4. Los usuarios administradores y representantes podrán ver los mensajes/notificaciones enviadas por el chofer del autobús.
 5. Los usuarios con rol de chofer podrán enviar mensajes/notificaciones a los administradores y representantes.
 6. Se debe realizar un modo de contingencia en cual se utilice el dispositivo móvil de chofer como emisor constante de las coordenadas geográficas del autobús.

7. Los choferes podrán cambiar el estado a cada uno de los estudiantes del autobús dependiendo el caso (Ausente, En el transporte y Llegó a su hogar).
 8. Los usuarios con el rol de chofer, podrán activar el modo de rastreo de contingencia.
 9. Los usuarios con rol de representante podrán consultar el estado de su representante.
 10. Los usuarios representantes podrán seleccionar la dirección física de parada donde serán buscados/dejados sus representados.
 11. Los representantes podrán ver su perfil y si así lo quieren pueden editar los campos permitidos.
 12. La lógica de la aplicación permitirá poder ubicar el autobús en el mapa y calcular el tiempo estimado de llegada a cada una de las paradas asignadas, permitiendo a cada uno de los representantes poder consultar toda esta información.
- Requerimientos No Funcionales:
 1. Se desarrollará una aplicación fácil de cargar.
 2. La paleta de colores designada que representará a la aplicación serán naranja, gris y blanco.
 3. Se utilizará una iconografía adecuada que facilite al usuario la comprensión de las funcionalidades del sistema.
 4. La aplicación contará con mensajes de retroalimentación al usuario para indicar el estado del sistema en cada momento.
 5. Se utilizará una tipografía que permita una fácil lectura para todo tipo de usuarios y ofrezca la sensación de estabilidad, confianza y determinación

Se realizó un diagrama UML en donde se puede contemplar todos los posibles casos de uso de la solución. Para dicho diagrama se tomaron en cuenta cada uno de los roles existentes en la misma, es decir, se presentan 3 usuarios en el diagrama: "Administrador", "Representante" y "Chofer". En la Figura 11.3 se puede apreciar el diagrama de los casos de uso de primer nivel y segundo nivel de la aplicación móvil.

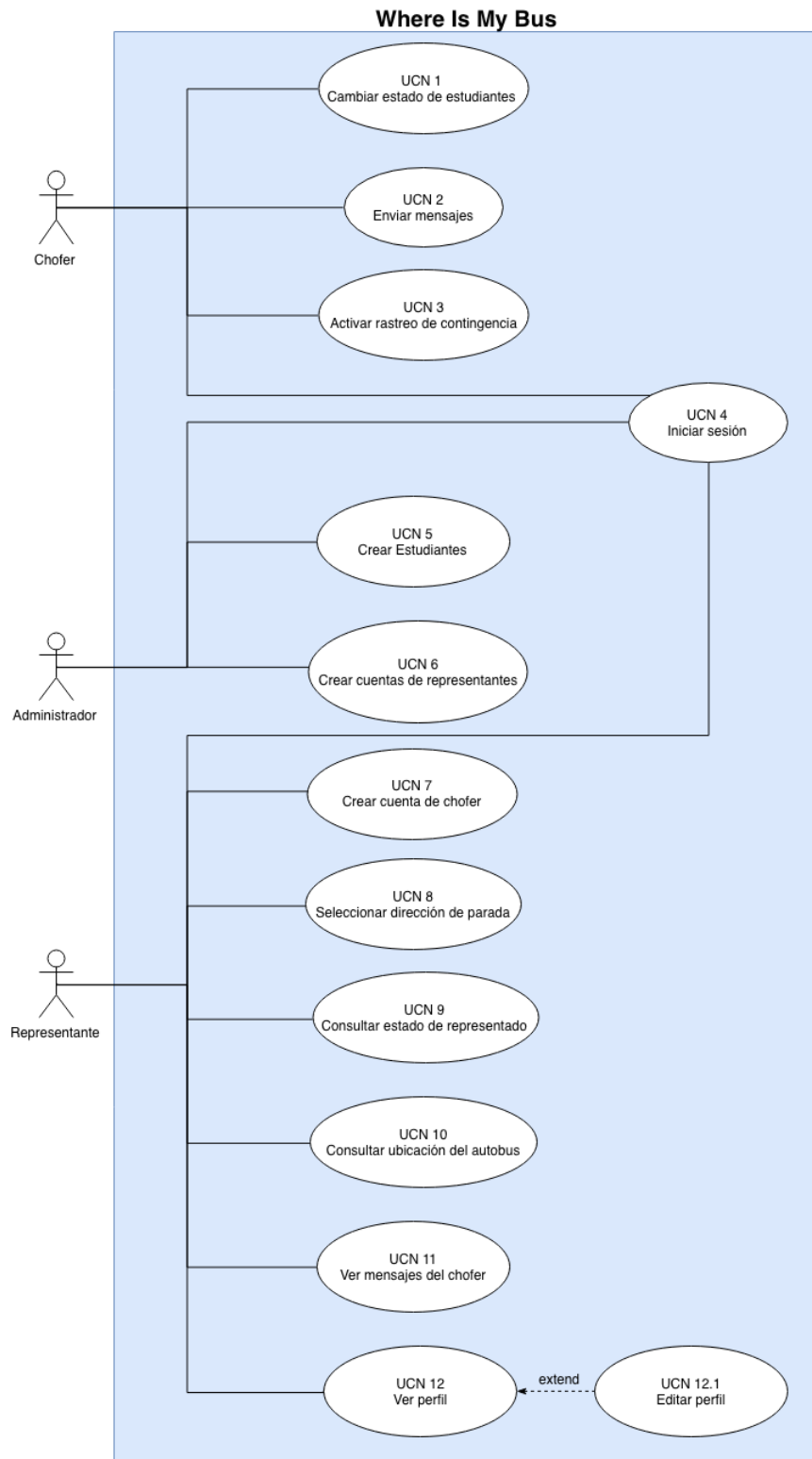


Figura 11.3: Diagrama de Casos de Uso de Primer y Segundo Nivel de la Aplicación Móvil

Por otro lado, se realizaron tres diagramas de navegación para cada uno de los tipos de usuarios de la aplicación. En las Figura 11.4, Figura 11.5 y Figura 11.6 se presentan los diagramas de navegación de los usuarios con rol de Administrador, Representante y Chofer respectivamente.

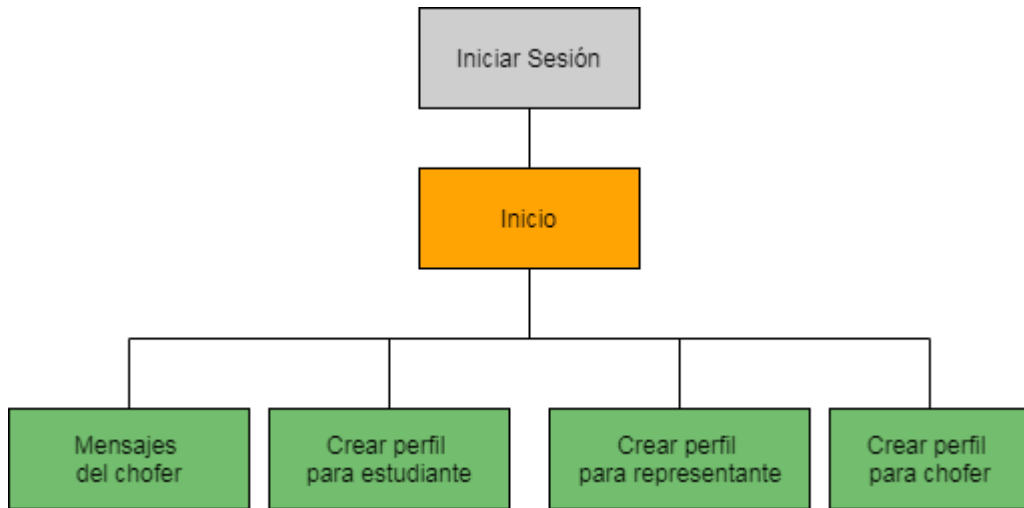


Figura 11.4: Diagrama de Navegación para usuario Administrador

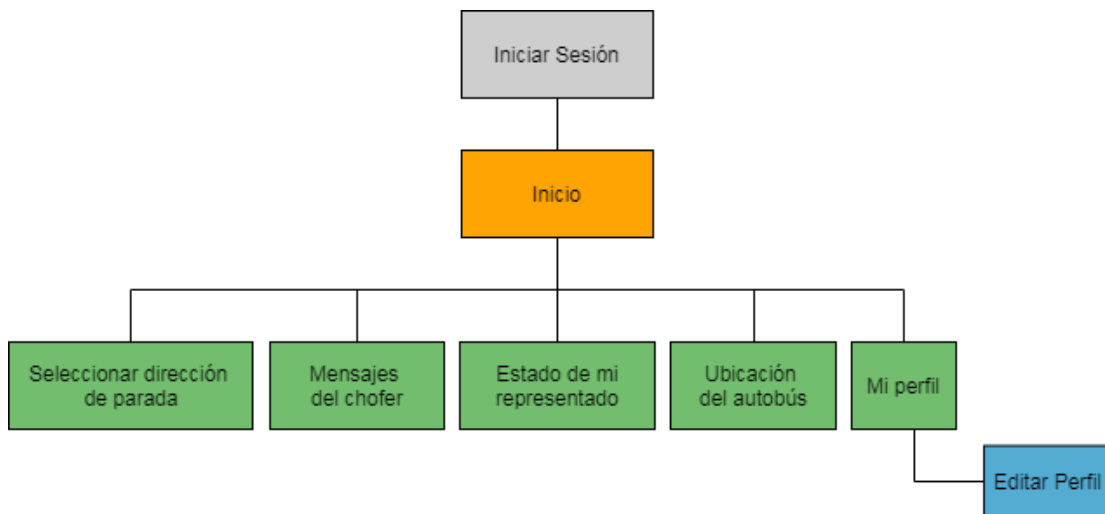


Figura 11.5: Diagrama de Navegación para usuario Representante

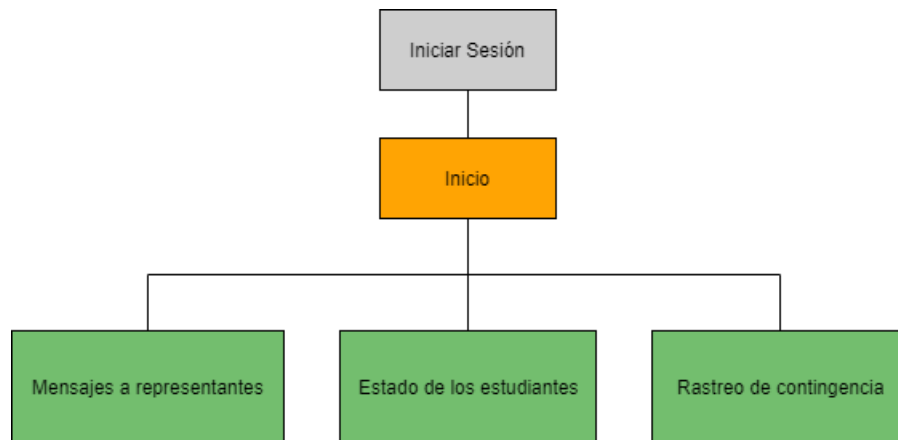


Figura 11.6: Diagrama de Navegación para usuario Chofer

A continuación, se mostrarán algunos prototipos de interfaz de usuario para la aplicación móvil desarrollados con la herramienta Adobe Illustrator CC.

En la Figura 11.7 se puede apreciar el Splash Screen de la aplicación, la cual muestra el logo del proyecto, el cual posee un ícono que mezcla la idea de un autobús ubicado en el mapa y a su vez posee un fondo inspirado en el valle de Caracas.



Figura 11.7: Diseño de Splash Screen de la Aplicación Móvil

En la Figura 11.8 se puede visualizar la vista de inicio de sesión para todos los usuarios que cuenta la aplicación. Esta vista viene inmediatamente luego de la vista Splash si el usuario no ha iniciado sesión.



Figura 11.8: Diseño de la Pantalla de Inicio de Sesión de la Aplicación Móvil

Cuando un representante entra por primera vez a su cuenta se le mostrará una vista de bienvenida que le solicitará la dirección que será la parada donde se buscará o se dejará a su representado (Figura 11.9).



Figura 11.9: Diseño de la Pantalla de Bienvenida de la Aplicación Móvil

En la Figura 11.10 se puede apreciar el menú de la aplicación para los usuarios con rol de Administrador, Representante y Chofer respectivamente, con cada una de sus opciones permitidas.

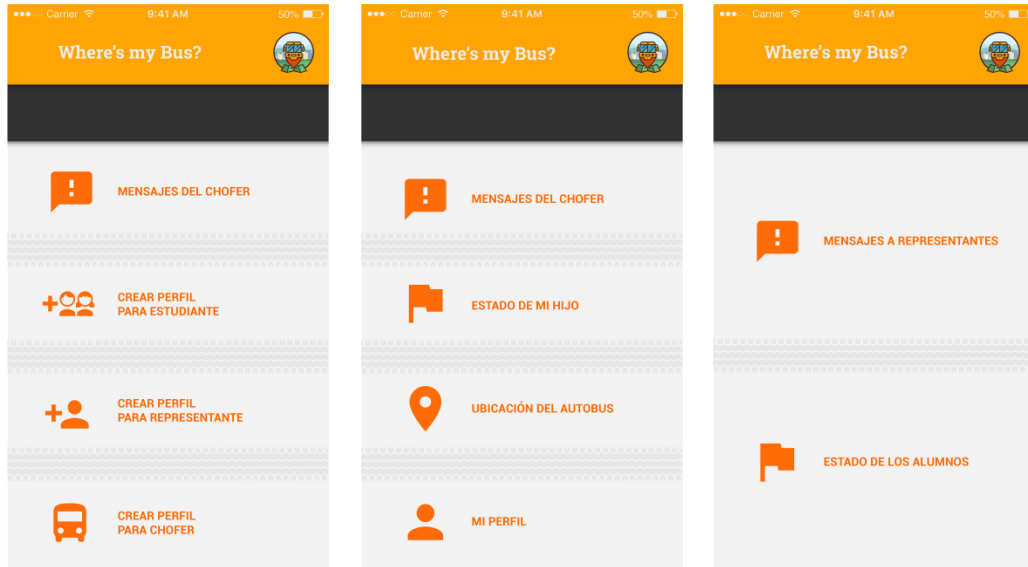


Figura 11.10: Diseño del Menú de cada tipo de usuario de la Aplicación Móvil (Administrador, Representante y Chofer)

La función del menú de Mensajes del Chofer es la misma para los usuarios con rol de Administrador y Representante. En la Figura 11.11 se puede apreciar dicha vista, donde se refleja cada uno de los mensajes enviados por el Chofer de la unidad con su respectiva fecha y hora.



Figura 11.11: Diseño de la vista Mensajes del Chofer de la Aplicación Móvil

El usuario Administrador es el encargado de crear a cada uno de los estudiantes con su respectiva información en la Base de Datos, así como también crear las cuentas de cada representante y del chofer. En la Figura 11.12 se puede observar cada una de las vistas que permiten realizar lo antes mencionado.

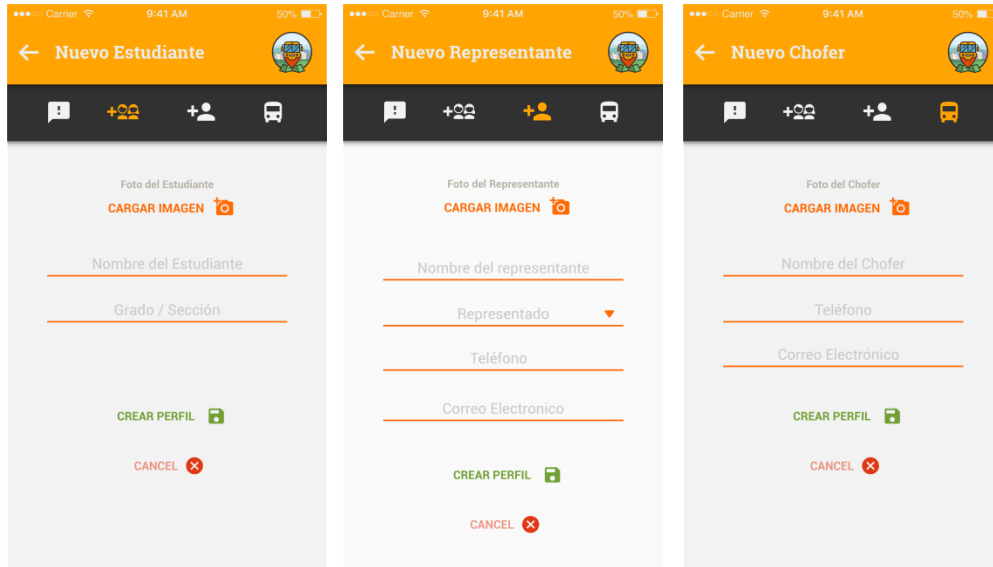


Figura 11.12: Diseño de la vista de creación de estudiantes y cuentas de usuarios de representantes y chofer de la Aplicación Móvil

En la Figura 11.13 se puede apreciar la pantalla en donde el representante puede verificar el en qué estado se encuentra su representado. Los tres estados existentes son: Ausente, En el transporte o Llegó a su hogar.

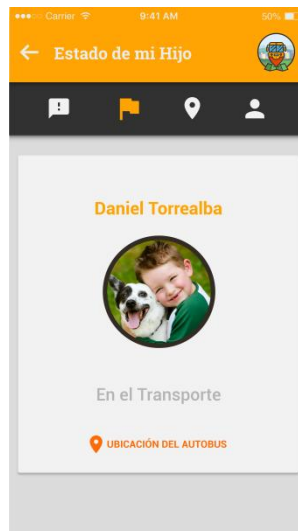


Figura 11.13: Diseño de la vista de Estado de mi representado de la Aplicación Móvil

Los representantes también tienen la opción de Ubicación del Autobús (Figura 11.14), donde podrán consultar la localización actual de la unidad escolar, así como también el tiempo estimado de llegada a la parada de cada representante.

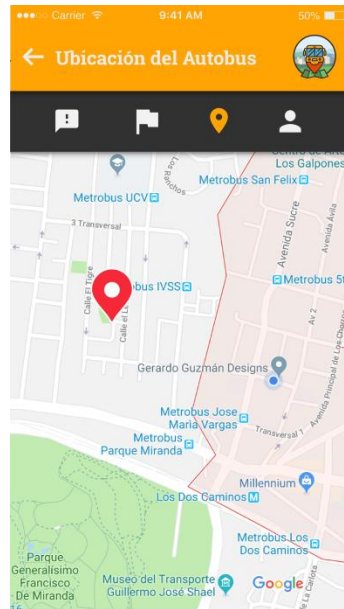


Figura 11.14: Diseño de la vista de Ubicación del Autobús de la Aplicación Móvil

Los representantes tienen la funcionalidad de ver su perfil y también editarlo. En la Figura 11.15 se presentan ambas vistas.

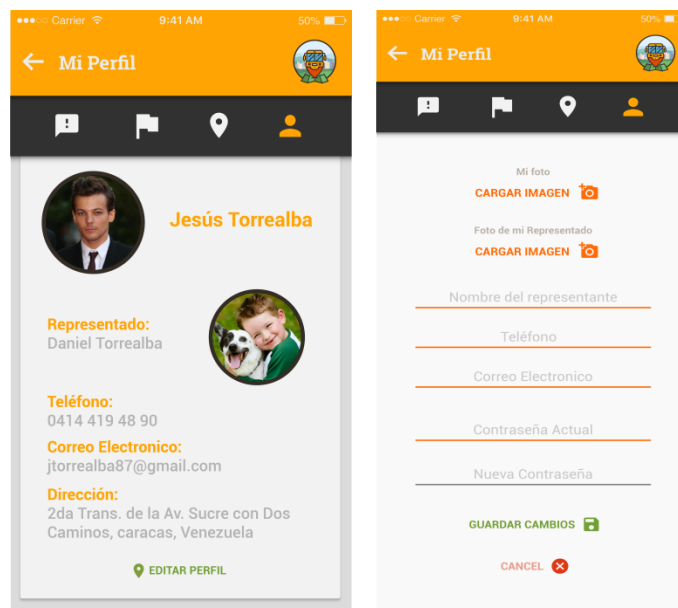


Figura 11.15: Diseño de la vista de Perfil y Editar Perfil de la Aplicación Móvil

El tipo de usuario Chofer puede enviar mensajes/notificaciones desde la sección Mensajes a los Padres (Figura 11.16), dichos mensajes pueden ser vistos por los Representantes y Administradores. El chofer tiene mensajes ya predeterminados.



Figura 11.16: Diseño de la vista de Mensajes a los Padres de la Aplicación Móvil

La otra funcionalidad del Chofer es mantener actualizado el estado de los estudiantes. Cuando estos no asisten tienen estado “Ausente”, si están en la unidad tienen estado “En el transporte” y al ser dejado en su parada tienen estado “Llegó a su hogar”. En la Figura 11.17 se puede apreciar la vista de estado de los estudiantes.

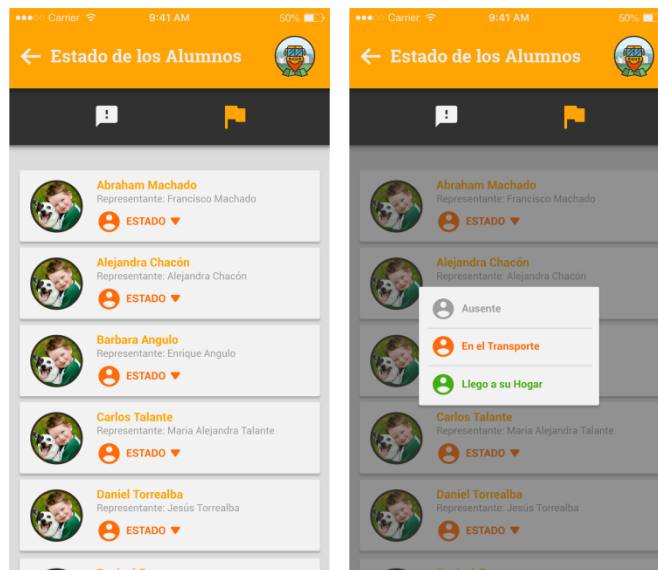


Figura 11.17: Diseño de la vista de Estado de los Alumnos de la Aplicación Móvil

11.3.2 Instalación y Configuración del Ambiente de Trabajo de la Aplicación

En el Sprint 2 se comenzó con la instalación y configuración de las tecnologías y herramientas a utilizar para el desarrollo de la aplicación móvil, en donde se destacan Visual Studio Code, Firebase, Ionic, Git y Cordova.

Durante los primeros sprints del desarrollo, la aplicación móvil fue trabajada por cada autor en servidores locales y emuladores en sus respectivos computadores personales y se utilizaba la herramienta Git para el control de versiones y el trabajo colaborativo se realizara de manera óptima. Uno de los autores utilizaba el sistema operativo Windows 10 y otro el sistema macOS High Sierra Versión 0.13.6.

Para poder iniciar el desarrollo de la aplicación móvil con el uso del Framework de desarrollo híbrido Ionic, el cual posee un Command Line Interface (CLI) que debe ser instalado para su posterior uso. Esta herramienta fue descargada de la página oficial haciendo uso de una serie de comandos que deben ser ejecutados en la línea de comandos del computador, en ambos computadores la instrucción para la instalación del CLI es la que se muestra en la Figura 11.18.

```
josue@LAPTOP-SMOKOMMJ MINGW64 ~/Desktop/whereIsMyBus (master)
$ npm install -g ionic
```

Figura 11.18: Comando para la instalación del Framework Ionic

El CLI de Ionic ofrece una amplia cantidad de herramientas de desarrollo y opciones que sirven de ayuda en el desarrollo. Una vez instalado se procedió a crear el proyecto utilizando el CLI de Ionic con el comando que se aprecia en la Figura 11.19.

```
josue@LAPTOP-SMOKOMMJ MINGW64 ~/Desktop/whereIsMyBus (master)
$ ionic start whereIsMyBus blank --type=angular
```

Figura 11.19: Comando para la creación de un proyecto en Ionic

Con este comando se finalizó la creación del proyecto, ahora la siguiente tarea que se realizó fue subir dicho proyecto a Git.

La herramienta Git se instaló descargando un instalador para SO Windows desde su página oficial y de manera similar la página principal ofrece un instalador para dispositivos iOS. Se configuró el proyecto haciendo uso de la plataforma Github, el cual es un servicio web de control de versiones y desarrollo de software colaborativo

basado en Git. En la Figura 11.20 se muestra la pantalla de inicio del proyecto en la plataforma Github.

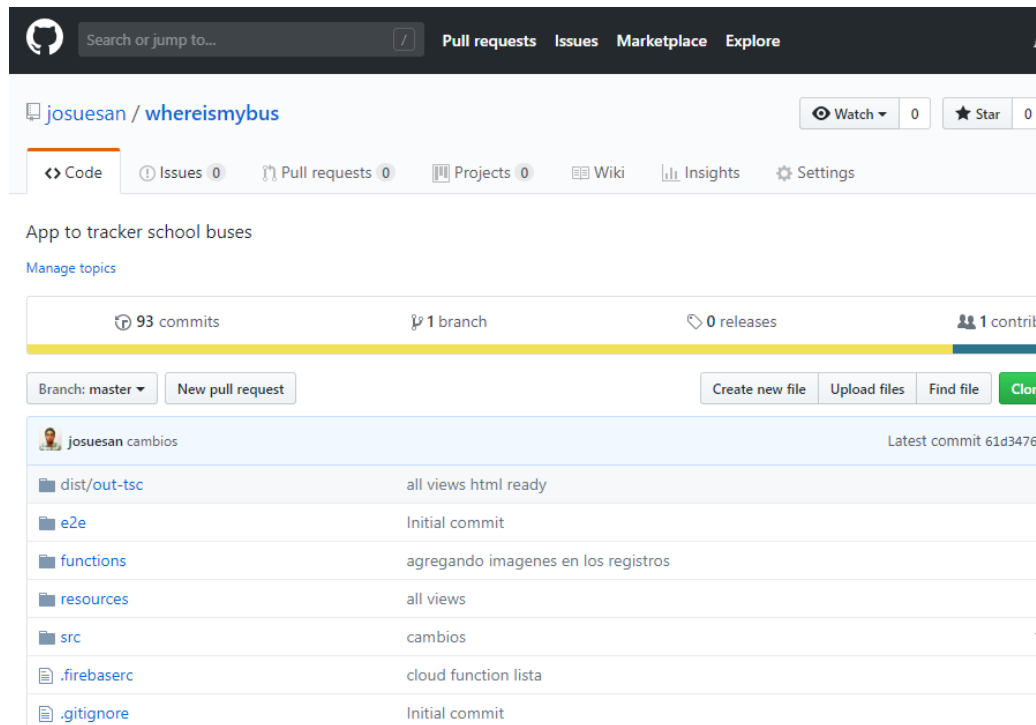


Figura 11.20: Proyecto de la aplicación móvil en la plataforma Github

Posteriormente se procedió a crear un proyecto en Firebase (Figura 11.21) desde <https://console.firebase.google.com>, con el fin de obtener las credenciales para poder hacer uso de los productos necesarios para el desarrollo del proyecto.

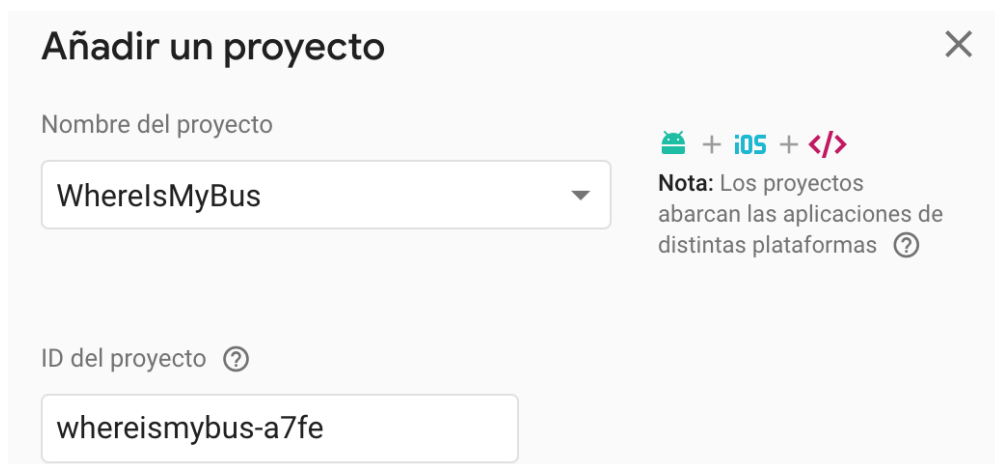


Figura 11.21: Creación de un proyecto en Firebase

Una vez creado el proyecto se presenta un menú con los productos que ofrece Firebase (Figura 11.22) para Desarrollo, de igual forma se señala cuáles de dichos productos fueron utilizados para del proyecto.

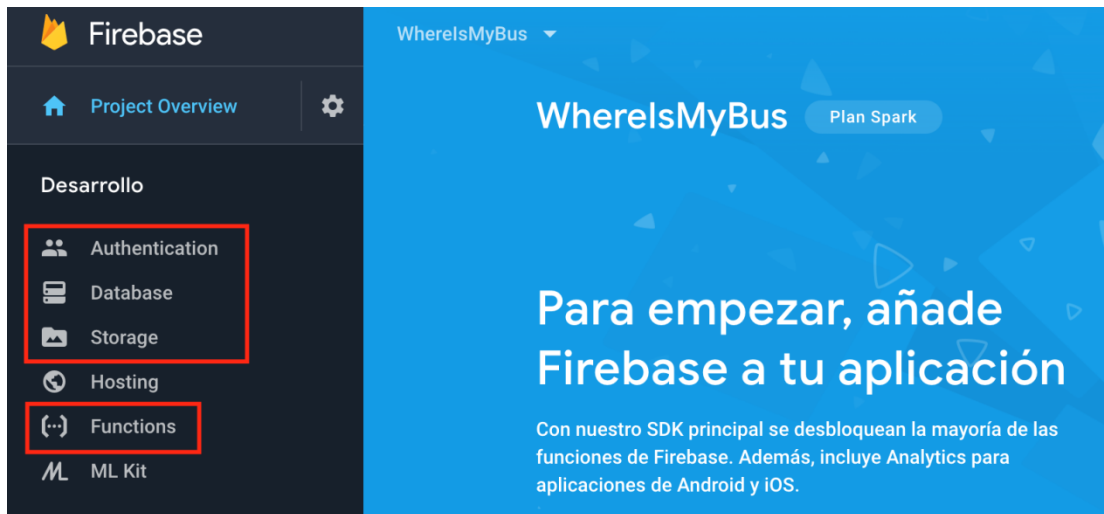


Figura 11.22: Productos Firebase para el Desarrollo

En la sección de Database se creó la Base de Datos Cloud Firestore para el almacenamiento de la data del proyecto (Figura 11.23).

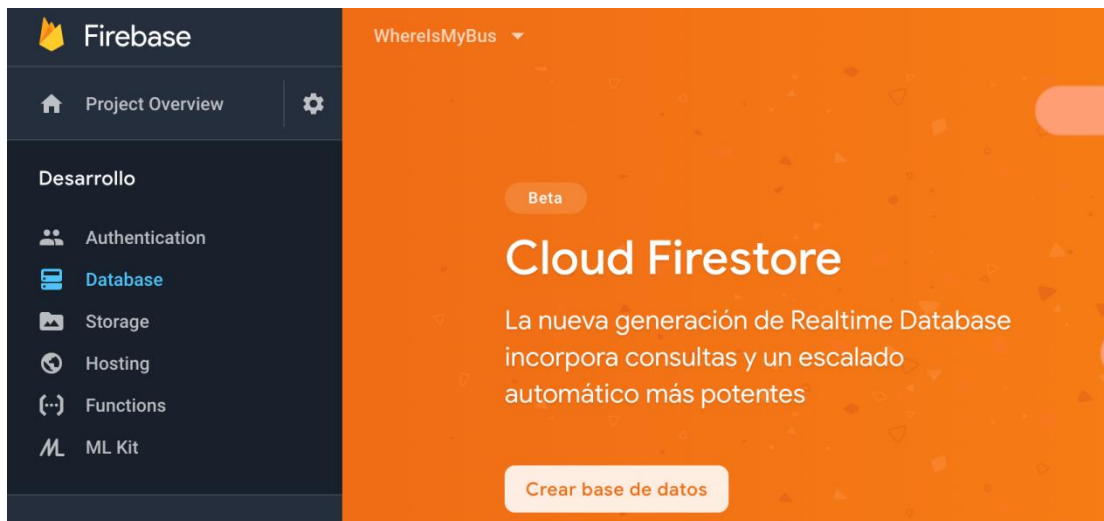


Figura 11.23: Creación Base de Datos Cloud Firestore

A continuación se presenta la Base de Datos lista para comenzar a almacenar colecciones y documentos (Figura 11.24).

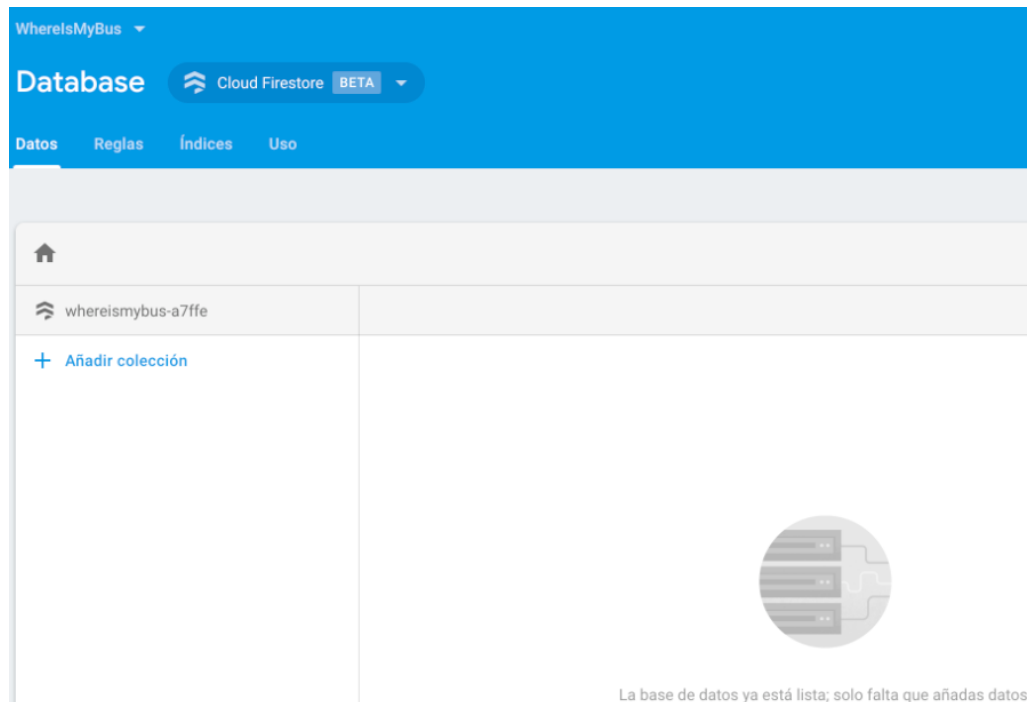


Figura 11.24: Pantalla principal de la base datos Firestore

Para finalizar la instalación y configuración del ambiente de trabajo, se procedió a instalar la librería de AngularFire2, es la librería oficial de Angular para trabajar con Firebase de manera sencilla y simple con el uso de TypeScript y demás complementos de Angular.

En la Figura 11.25 se muestra el comando de instalación de la librería.

```
josue@LAPTOP-SMOKOMMJ MINGW64 ~/Desktop/whereIsMyBus (master)
$ npm install firebase @angular/fire --save
```

Figura 11.25: Comando de instalación de la librería de AngularFire2

Una vez culminado el proceso de instalación, se debe integrar con la aplicación y colocar las credenciales del proyecto creado previamente en Firebase. Para esto se debe en el archivo App.Module de Angular importar los módulos de inicialización y demás complementos que Firebase necesita para ejecutarse correctamente tal como se puede observar en la Figura 11.26.

```

@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [
    BrowserModule,
    IonicModule.forRoot(),
    AppRoutingModule,
    AngularFireModule.initializeApp(firebaseConfig, 'WhereIsMyBus'),
    AngularFireDatabaseModule,
    AngularFireAuthModule,
    AngularFirestoreModule.enablePersistence(),
    AngularFireStorageModule,
    HttpClientModule
  ],
  providers: [

```

Figura 11.26: Inicialización de la librería AngularFire2 en el módulo principal de la aplicación móvil

11.3.3 Desarrollo de la Aplicación Móvil

El desarrollo de todos los módulos, componentes y vistas de la aplicación móvil se llevó a cabo entre los Sprints 2 y 4. Posterior a esos Sprints se hicieron correcciones generales a algunas funcionalidades de la aplicación a lo largo de los Sprints restantes según las pruebas que se realizaban y las necesidades que surgían durante el desarrollo del prototipo de hardware.

Para entender cómo fue el proceso de desarrollo haciendo uso del Framework Ionic, se debe tomar en cuenta que este Framework busca la creación de aplicaciones usando el patrón de diseño basado en componentes, por lo cual este representó para los autores una guía para la estructuración y la arquitectura de la aplicación.

La aplicación móvil fue dividida en 4 módulos importantes que son:

- Módulo de páginas: en este módulo se encuentran todos los archivos que corresponden a las páginas de cada sección de la aplicación móvil, por lo general las páginas solo cuentan con el diseño de la misma y la petición de datos a los servicios para su posterior uso en los componentes que dicha página utilice.
- Módulo de componentes: en este módulo de la aplicación se encuentran todos los archivos que corresponden a los diferentes componentes de la

aplicación. Un componente es un conjunto de elementos que se repiten en varias secciones de una página, por lo cual se desea crear uno solo que pueda ser instanciado en diferentes secciones con diferentes datos según sea el caso.

- Módulo de servicios: en este módulo se encuentran todos los servicios que la aplicación utiliza, llámese servicio al consumo de información a API's externas o a la base de datos. El objetivo de crear servicios es que en ellos sea el único punto donde se puedan realizar todas las operaciones CRUD de la aplicación.
- Módulo de interfaces: es un módulo en el cual se colocan las interfaces que representan diferentes modelos de la aplicación, se utilizan para dar consistencia a los objetos utilizados.

En la Figura 11.27 se puede apreciar la estructura de carpetas de *src* (source de la aplicación) la cual incluye cada uno de los módulos que fueron mencionados anteriormente.

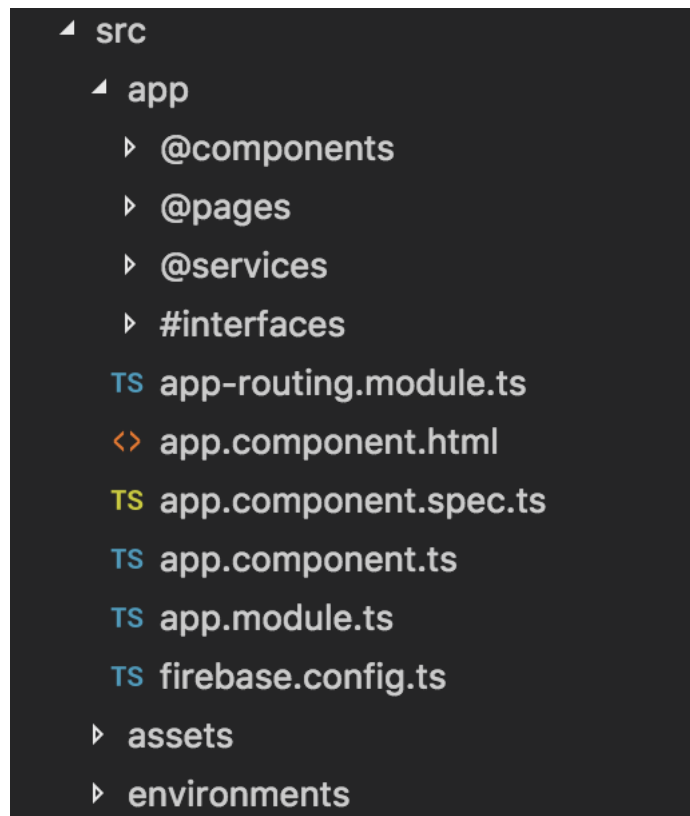


Figura 11.27: Estructura de la carpeta *src* de la Aplicación Móvil

Colecciones y documentos utilizados

A medida que se iba avanzando el desarrollo de la aplicación, fueron surgiendo las distintas colecciones y documentos que serían utilizados para el almacenamiento de la data en la Base de Datos NoSQL Cloud Firestore. Las colecciones utilizadas en el proyecto fueron las siguientes:

- **locations:** contiene la información de coordenadas (latitud y longitud) del autobús, así como también la fecha y hora en la que fue obtenida cada coordenada.
- **messages:** encargada de almacenar la información de cada uno de los mensajes/notificaciones enviadas por el chofer. Contiene el texto del mensaje, id del chofer que lo envió y la fecha en que fue enviado.
- **students:** almacena la información de todos los estudiantes que usan el transporte escolar. Contiene un id único por cada estudiante, el grado/sección al que pertenece, el estado que posee y la foto del mismo.
- **users:** esta colección guarda la información de los distintos tipos de usuarios que pueden utilizar la aplicación. Tanto los usuarios con rol de Representante y Chofer poseen un id único, nombre, correo electrónico, teléfono, rol y foto.

También los representantes poseen el id que los asocia con su representado, un parámetro de tipo booleano el cual indica si es la primera vez que ingresa a la aplicación y pueden añadir de forma dinámica los campos de latitud y longitud que serán los que representarán la dirección de parada de su representado. En el caso del usuario Admin, solo necesitan tener un id único, correo electrónico y el rol que los identifique.

En la Figura 11.28 se presentan las colecciones explicadas anteriormente con un ejemplo de documento de cada una.

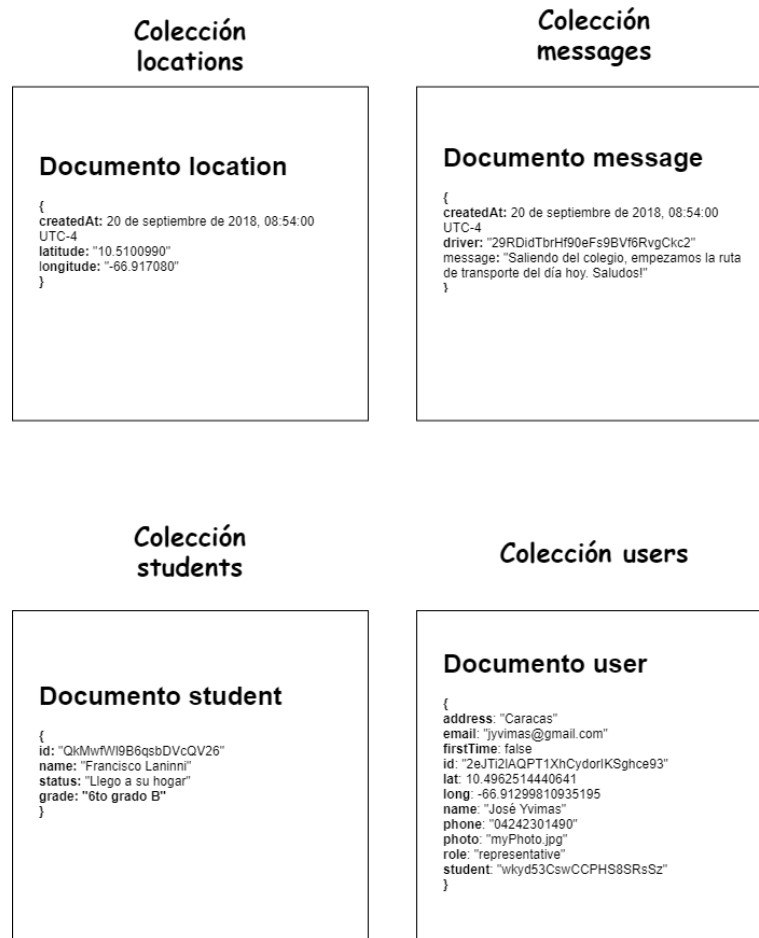


Figura 11.28: Ejemplo de documentos de las colecciones de la Base de Datos

A continuación se presentarán las pantallas finales de la aplicación móvil así como también una explicación resumida del funcionamiento de cada una.

En la Figura 11.29 se puede apreciar la pantalla de login o inicio de sesión. La lógica de esta pantalla fue desarrollada gracias a Firebase Authentication, producto encargado de recibir las credenciales ingresadas por el usuario, permitiendo así mantener la sesión si la información es igual a la que se encuentra almacenada. Para iniciar sesión con Firebase Authentication existen muchos métodos, los cuales fueron mencionados en el Capítulo 5, para el proyecto el inicio de sesión seleccionado viene dado por correo electrónico y contraseña.

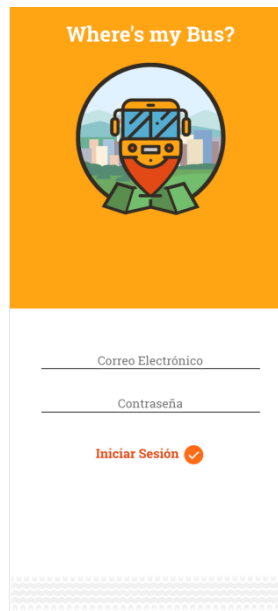


Figura 11.29: Pantalla de Inicio de Sesión

El home o inicio de cada uno de los usuarios de la aplicación viene dado por el rol que posee cada uno. Para ello, luego que el inicio de sesión es exitoso se obtiene la información del usuario en cuestión desde la Cloud Firestore con el fin de verificar el rol que posee. En la Figura 11.30 se muestran los tres tipos de pantallas de inicio existentes en la aplicación: Administrador, Representante y Chofer.



Figura 11.30: Pantallas de Inicio por cada tipo de usuario

Las cuentas de los usuarios de tipo administrador son creadas directamente por los autores del proyecto desde la consola de Firebase.

El usuario administrador tiene la función de crear los estudiantes en la base de datos. Para ello ingresa los datos del estudiante desde la pantalla que se muestra en la Figura 11.31. La data de cada estudiante es almacenada en documentos de la colección “students” en la Cloud Firestore. Cada estudiante posee un id único.



Figura 11.31: Pantalla para crear estudiantes en la base de datos

El administrador también es el encargado de crear las cuentas de los usuarios con rol de Representante (Figura 11.32) y Chofer (Figura 11.33). La lógica de ambas funcionalidades es similar, debido a que cada una hace uso de Firebase Authentication para crear las cuentas con un correo y una contraseña aleatoria (enviada al mismo correo), y uso de la Cloud Firestore para almacenar la data del usuario obtenida desde los formularios haciendo referencia al mismo id del usuario creado con Firebase Authentication. La información de los usuarios es almacenada en documentos de la colección “users”.

Para la creación de representantes es necesario que esté creado su representado con anterioridad para poder asociar su id al documento del representante.

Algo que no se mencionó anteriormente, es el almacenamiento de las fotos. Para este caso se hace uso del Cloud Storage de Firebase, el cual almacena las imágenes

y devuelve una url para el acceso de la misma. La url se guarda en el campo *photo* del documento de los usuarios o estudiantes según sea el caso.

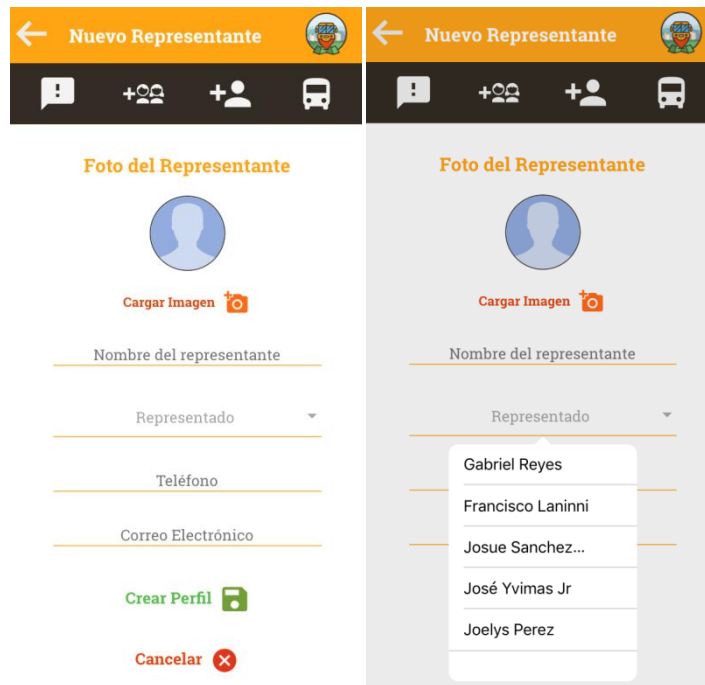


Figura 11.32: Pantalla para crear usuarios de tipo Representante

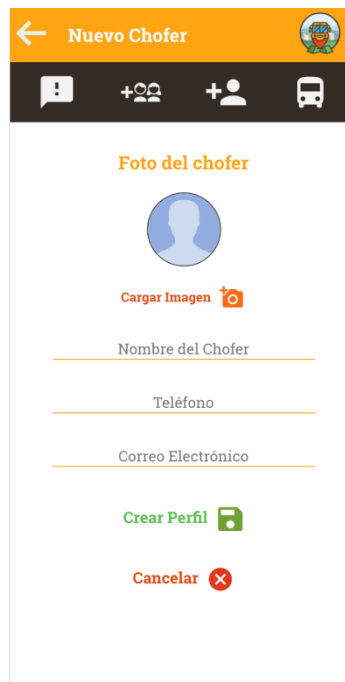


Figura 11.33: Pantalla para crear usuarios de tipo Chofer

Los usuarios administradores y representantes poseen una funcionalidad en común: Mensajes del chofer, en esta pantalla se muestran los mensajes que son enviados por el chofer. Para realizar lo anterior se obtienen todos los documentos que se encuentran en la colección “messages” y se busca la información del usuario que lo envió. En la Figura 11.34 se puede apreciar las pantallas para los Administradores y Representantes, la única variación son las opciones del menú de cada una.

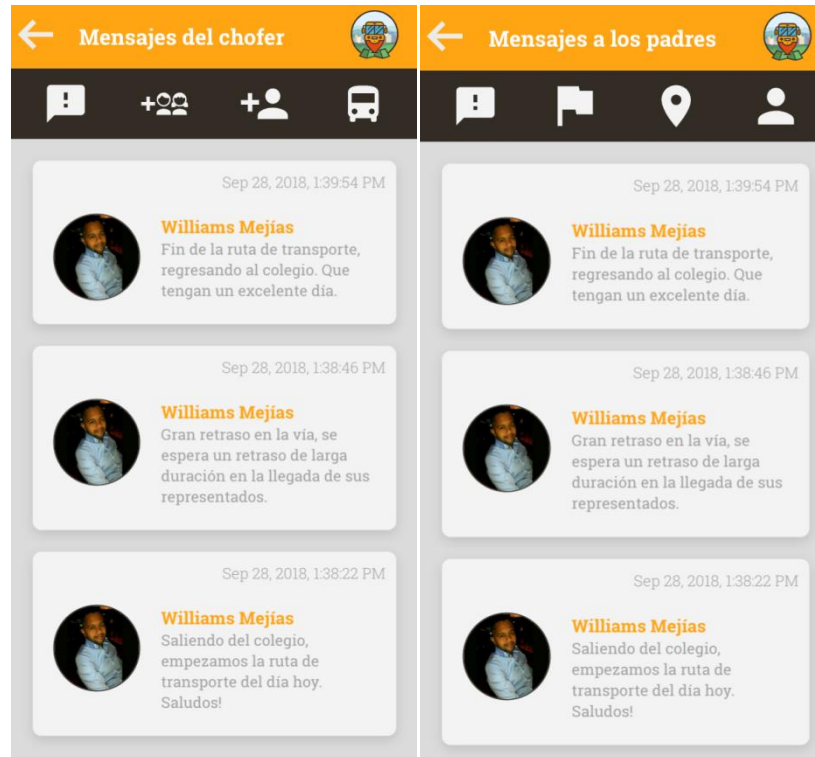


Figura 11.34: Pantalla de Mensajes del chofer (Administrador y Representante)

Cuando los representantes inician sesión por primera vez en la aplicación móvil se les presenta una pantalla de bienvenida donde deben ingresar la dirección de parada de su representado (Figura 11.35). Para esta funcionalidad se implementaron dos formas de realizar un registro de parada exitoso, las cuales se explicaran brevemente a continuación:

- **Selección de parada en mapa:** se incluyó un mapa en esta vista con la intención que el usuario navegue en el mapa, localice el punto de parada y luego añada un marcador. Internamente la aplicación obtiene las coordenadas geográficas de ese marcador y se almacena en base de datos como la parada del estudiante dentro de la colección “users” en los campos de latitud y longitud.
- **Obtención de posición actual:** para usuarios no tan experimentados en el uso de mapas o usuarios que desean un método más rápido de selección de

parada, se incluyó esta funcionalidad la cual utiliza el GPS que posee el dispositivo móvil para obtener las coordenadas de su posición actual y posteriormente ser almacenada en base datos como se mencionó anteriormente.



Figura 11.35: Pantalla para ingresar la dirección de parada

Los representantes pueden observar el estado que posee su representado (Figura 11.36), este estado puede tomar el valor de: “En el transporte” si aún se encuentra en autobús, “Ausente” si no asistió a clases y “Llegó a su hogar” cuando ya llegó a su destino. Para obtener el estado se consulta el documento del representado. Además dicho estado es actualizado por el chofer de la unidad.

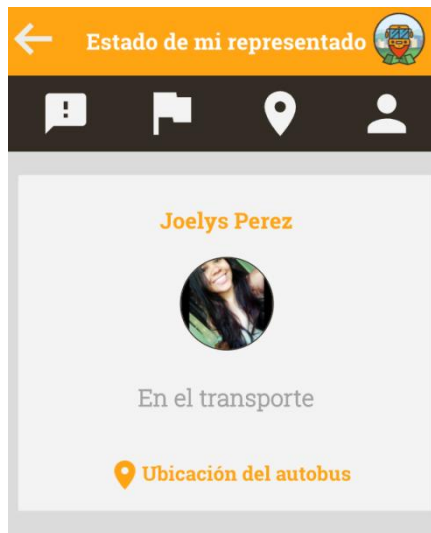


Figura 11.36: Pantalla de Estado de mi representado

La funcionalidad más importante o la principal en la que se basa el proyecto es la de la Ubicación del autobús. Esta función solo puede ser accedida desde los usuarios con rol de representante.

Para esta funcionalidad se utilizó un mapa apoyado en la tecnología Leaflet, el cual permite mostrar diferentes marcadores, realizar acercamiento a zonas específicas, transiciones en el mapa de acuerdo a los parámetros especificados en su API. Como guía para los usuarios se utilizaron 2 marcadores uno azul el cual representa el transporte escolar y estará en constante movimiento por el mapa de acuerdo a la posición del autobús actual y un marcador naranja el cual representa la posición de la parada designada de ese usuario, por lo tanto siempre esta estático en el mapa.

Para poder renderizar los cambios de posición del marcador del autobús en tiempo real se utilizaron elementos observables (comúnmente llamado en las aplicaciones de Firebase como onSnapshot) a la colección "locations" de la base de datos Cloud Firestore. Un observable es un elemento que está en escucha de cambios en cierto objeto (los documentos de la base de datos en el presente caso de estudio) y cuando un cambio ocurre este realiza un aviso el cual puede ser utilizado por los desarrolladores para incluir su lógica de programación deseada.

Para este caso de estudio cuando el observable notificaba un cambio en la base de datos se procedía a realizar una petición al API de OpenRouteService con la última posición del autobús y la parada designada por ese usuario, obteniendo como respuesta el tiempo aproximado de llegada del autobús y la distancia entre el autobús y la parada en cuestión. De igual forma se actualiza la posición del marcador que representa el autobús por cada nuevo documento que se almacene en la colección "locations".

Un punto a destacar de esta funcionalidad es que cuando el chofer marca que el estado del estudiante es “Llegó a su hogar” se procede a eliminar el observable y a indicar al usuario en dicha vista que su representado ya se encuentra en la parada designada.

En la Figura 11.37 se visualizan las pantallas de Ubicación del autobús cuando el representado tiene estado “En el transporte” y cuando posee estado de “Ausente” o “Llegó a su hogar”.

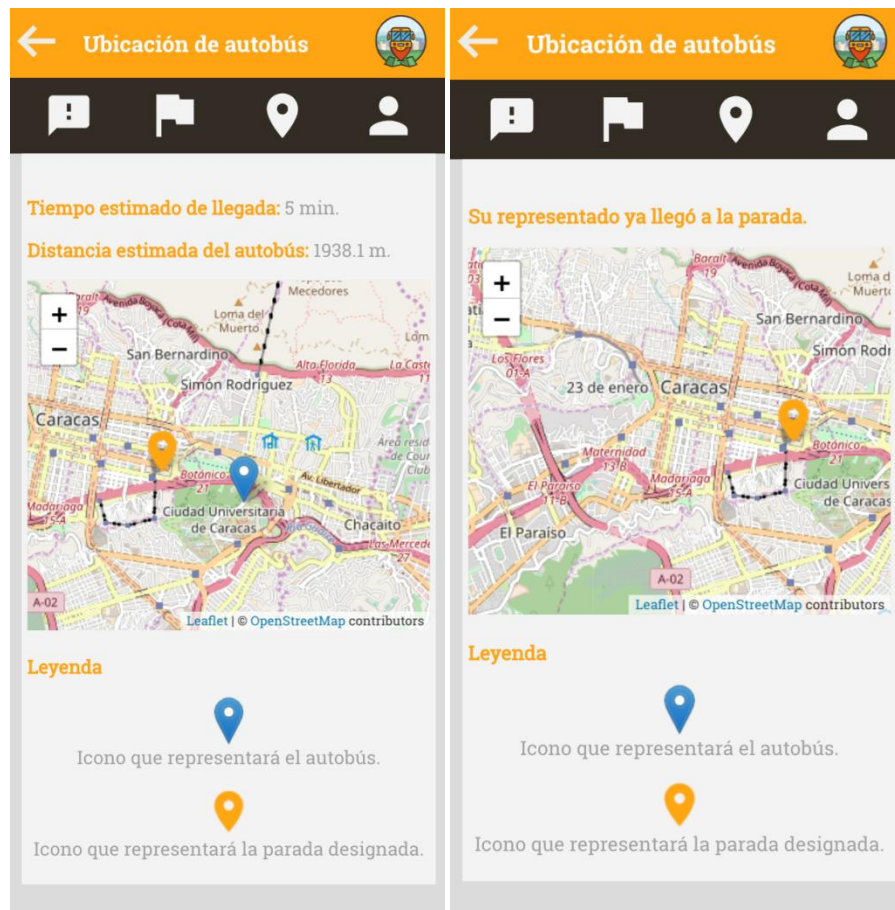


Figura 11.37: Pantalla de Ubicación del Autobús

Existe una función de perfil y editar perfil (Figura 11.38), solo disponible para los representantes. En el perfil del usuario se le muestra toda su información, así como también parte de la información de su representado. Los únicos datos que pueden ser editados son las fotos (representante y representado), número de teléfono y la contraseña. Para obtener toda esta información es necesario consultar el documento del representante en la colección users y posteriormente con el id del estudiante asociado a dicho representante se consulta el documento del representado en la colección students.

A la hora de editar el perfil se almacena la nueva información en los respectivos documentos referenciándolos mediante su id, todo esto con ayuda de la función update de Cloud Firestore. En caso de editar la foto se utilizaría nuevamente la Cloud Storage para su almacenamiento como se mencionó anteriormente.

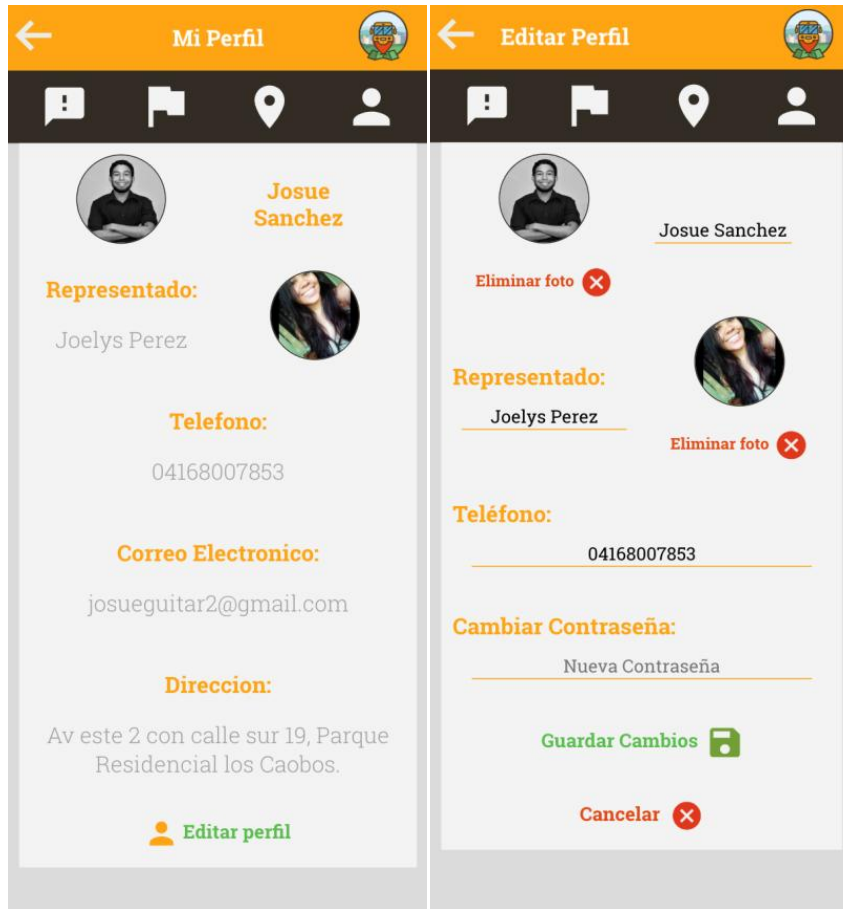


Figura 11.38: Pantallas de Perfil y Editar Perfil

El usuario tipo Chofer solo posee tres funcionalidades en la aplicación móvil. Una es la de enviar mensajes a representantes (Figura 11.39), los cuales también pueden ser visto por los usuarios administradores. El chofer escribe su mensaje o selecciona alguno de los que ya posee predeterminados, estos mensajes son agregados como un documento nuevo en la colección "messages" junto a la fecha y hora en la que fue enviado y el usuario chofer que lo envió.

Cuando el chofer envía un mensaje, este puede ser visto por cada uno de los representantes. Además posee la opción de limpiar el historial de los mensajes que se encuentran almacenados, para ello se eliminan todos los documentos que se encuentran en la colección con ayuda de la función delete de Cloud Firestore.



Figura 11.39: Pantallas de Mensajes a representantes y Mensajes predeterminados

La segunda funcionalidad del chofer es la de Estado de los estudiantes (Figura 11.40). Los únicos estados que deben existir al comenzar una jornada del servicio de transporte son “Ausente” y “En el transporte”, por ello el chofer debe pasar la asistencia de los estudiantes que ingresan al autobús. Cuando sea el viaje de retorno a la parada de los representados, el chofer debe asignarle el estado de “Llegó a su hogar” a cada de uno estudiante que vaya dejando durante el trayecto.

Para esta pantalla se obtiene la información de todos los estudiantes desde la colección “students” y con el id de cada estudiante se consulta la información de los representantes en la colección “users”. Cuando el chofer cambia el estado de algún estudiante, se actualiza su información mediante la función update de Cloud Firestore referenciando el id del estudiante en la colección “students”.

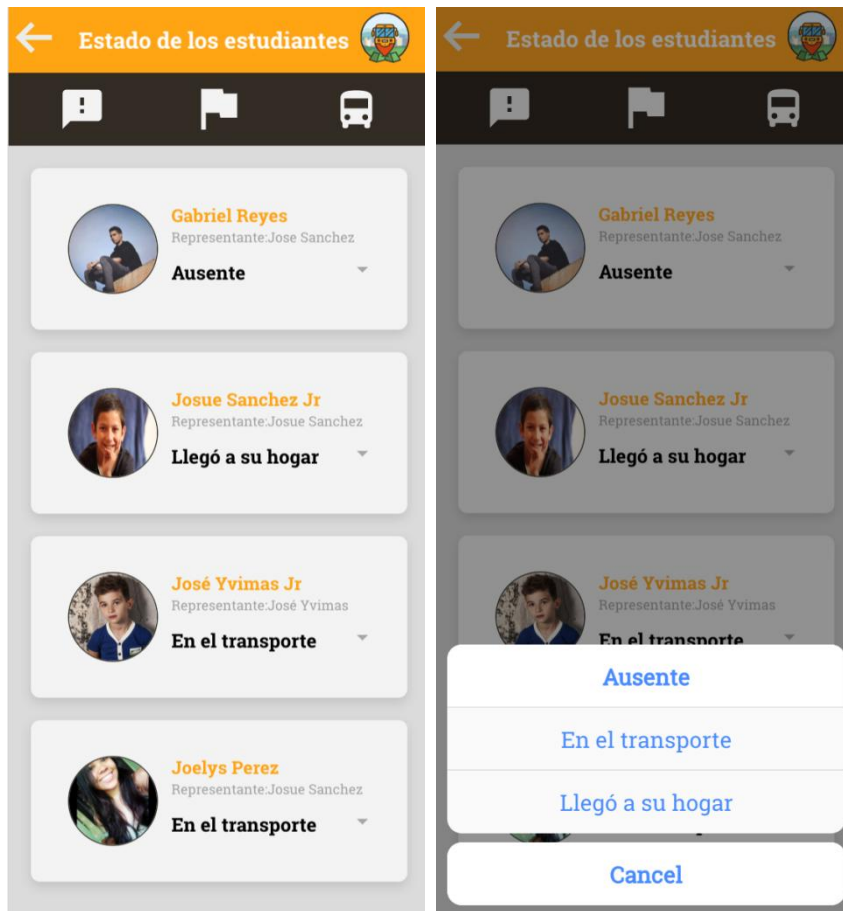


Figura 11.40: Pantalla de Estado de los estudiantes

La tercera funcionalidad del chofer es la de activar rastreo de contingencia (Figura 11.41). Esta funcionalidad está desarrollada como medida de precaución en caso de que el prototipo de hardware falle u ocurra un inconveniente con el mismo. Por eso, esta funcionalidad permite obtener la posición geográfica del autobús mediante el envío de coordenadas utilizando el dispositivo móvil del chofer, una vez finalizada la emergencia puede anular el rastreo de contingencia presionando un botón.

Para esta pantalla se desarrolló de manera simple se procede a tener un botón centrado con los colores rojo o verde de acuerdo al estado de esta funcionalidad. Además incluyendo un texto que explica al usuario de menor experticia qué función se ejecutará.

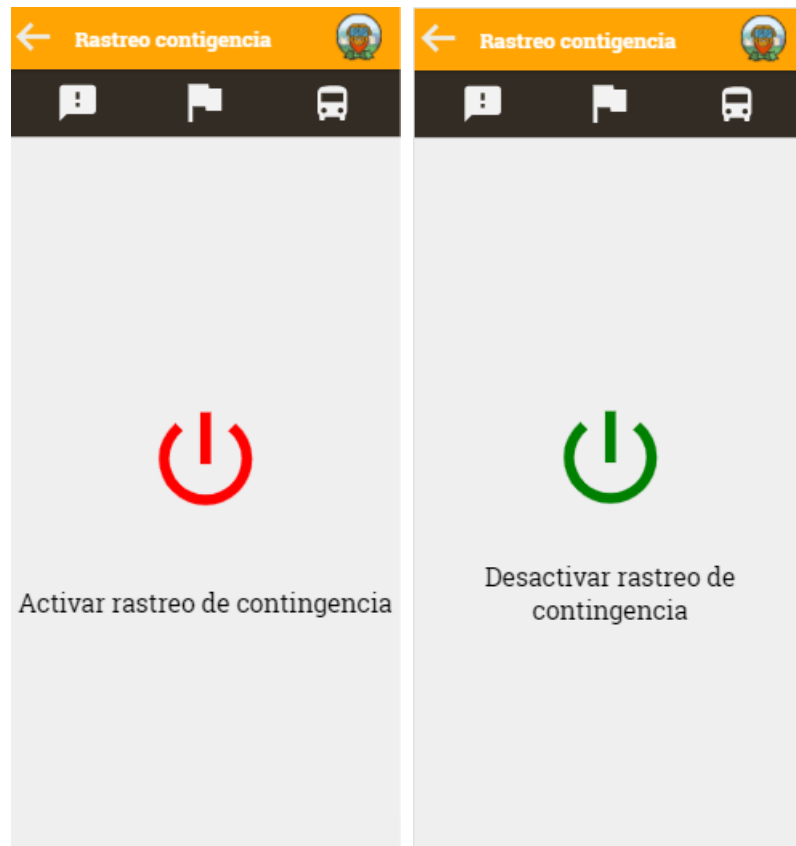


Figura 11.41: Pantalla de Rastreo de contingencia

11.3.4 Despliegue de la Aplicación Móvil

Una vez finalizado el desarrollo de la aplicación móvil se despliega la aplicación móvil en las plataformas Android y iOS para poder realizar pruebas de aceptación y de funcionalidades. Estas actividades se realizaron en el Sprint 6.

11.3.4.1 Despliegue en Sistemas Android

Para el despliegue en dispositivos Android, el CLI de Ionic junto con las funcionalidades de Cordova sirven de soporte para realizar esta tarea de manera simple, sencilla y rápida se puede crear la apk del proyecto.

Como principales requerimientos antes de realizar el despliegue es que el computador debe tener instalado:

- JDK de Java.
- Android Studio.

- Actualizar y poseer la última versión del Gradle.

Los autores del proyecto ya contaban con dichos requerimientos en sus computadores y procedieron a realizar el despliegue de la aplicación realizando los siguientes pasos:

- Se debe agregar la plataforma Android en el directorio de trabajo y crear las múltiples imágenes (Favicon y Splash Screen) para los dispositivos con tecnología Android (Figura 11.42).

```
josue@LAPTOP-SMOKOMMJ MINGW64 ~/Desktop/whereIsMyBus (master)
$ ionic cordova prepare android
```

Figura 11.42: Comando para agregar la plataforma Android a la estructura del proyecto

- El archivo config.xml (Figura 11.43) del proyecto contiene los metadatos de la aplicación y es necesario que cuente con un id propio de la aplicación y el nombre de los autores con una breve descripción. Por lo tanto se procedió a modificar este archivo y colocar los datos necesarios.

```
config.xml x
1 <?xml version='1.0' encoding='utf-8'?>
2 <widget id="com.teg.whereIsMyBus" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
3   <name>whereIsMyBus</name>
4   <description>TEG App to tracker school buses.</description>
5   <author email="JJSY@whereismybus.com" href="https://github.com/josuesan/whereismybus">Where is my bus Team</author>
6   <content src="index.html" />
7   <access origin="*" />
```

Figura 11.43: Archivo config.xml de la Aplicación Móvil

Una de las ventajas de Cordova es que posee un comando que crea una compilación del proyecto para la plataforma Android y posteriormente su archivo ejecutable. Adicionalmente se despliega un emulador el cual debe estar previamente configurado en el IDE de Android Studio. En la Figura 11.44 se observa el comando de compilación y despliegue en emuladores.

```
josue@LAPTOP-SMOKOMMJ MINGW64 ~/Desktop/whereIsMyBus (master)
$ ionic cordova run android -l
```

Figura 11.44: Comando para la compilación y despliegue en el emulador

- Como último paso para lograr una compilación para producción se debe incluir los prefijos “prod” y “release” los mismos generan un archivo ejecutable eliminando todas las opciones de debug y minimizan el código resultante. En la Figura 11.45 se aprecia el comando para realizar dicha tarea.

```
josue@LAPTOP-SMOKOMMJ MINGW64 ~/Desktop/whereIsMyBus (master)
$ ionic cordova build android --prod --release
```

Figura 11.45: Comando para la compilación y generación de archivo ejecutable optimizado para producción

11.3.4.1 Despliegue en Sistemas iOS

Los autores del proyecto pese a poseer un computador con sistema operativo macOS, presentaron inconvenientes en conseguir un dispositivo móvil para realizar dicho despliegue en el mismo, por lo cual recurrieron a realizar despliegue de la aplicación con el uso de emuladores virtuales de dispositivos iOS.

Particularmente Ionic CLI tiene integrado una herramienta llamada Ionic Lab la cual crea unos emuladores de distintas plataformas y despliega la aplicación al ejecutarse el emulador.

En la Figura 11.46 se puede observar el comando utilizado para el despliegue del emulador y en la Figura 11.47 como la aplicación se visualiza ya con un emulador de dispositivos iOS.

```
josue@LAPTOP-SMOKOMMJ MINGW64 ~/Desktop/whereIsMyBus (master)
$ ionic lab
```

Figura 11.46: Comando para el despliegue de la aplicación móvil en un emulador de dispositivos iOS

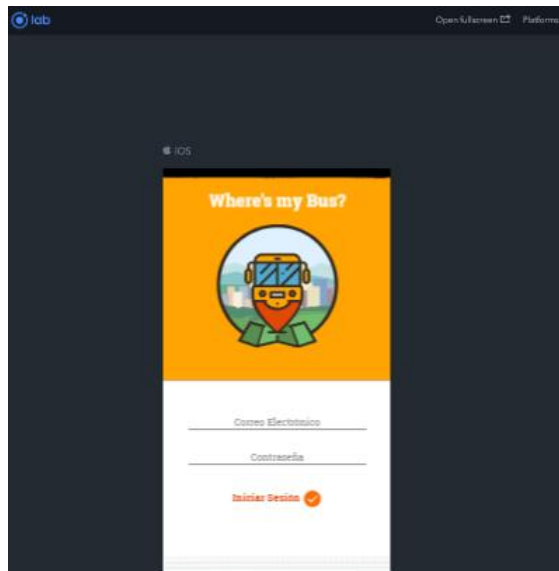


Figura 11.47: Aplicación ejecutándose en el emulador de iOS nativo de Ionic

11.4 Análisis General del Servicio en la Nube

El software de servicio en la nube tiene la función de permitir la comunicación unidireccional desde el prototipo de hardware hacia la base de datos usada para este proyecto, fue desarrollado haciendo uso de las Cloud Functions, producto que ofrece Firebase para correr código de backend en la nube.

La funcionalidad de este servicio en la nube es la recibir las coordenadas de ubicación del autobús desde el hardware y posteriormente almacenarlo en una colección la base de datos NoSQL Cloud Firestore de Firebase. Esta función es consumida desde SMSsync al momento de recibir un nuevo mensaje en la pasarela SMS.

11.4.1 Instalación y Configuración del Ambiente de Trabajo para la Cloud Function

En el Sprint 4 se procedió a la instalación, configuración y creación de las Cloud Functions de Firebase en la ruta de la aplicación móvil.

Es necesario un entorno Node.js para escribir funciones y Firebase CLI a fin de implementar funciones en el tiempo de ejecución de Cloud Functions. A continuación, en la Figura 11.48 se presentan los comandos utilizados para la instalación del producto a través de npm.

```
npm install firebase-functions@latest firebase-admin@latest --save
npm install -g firebase-tools
```

Figura 11.48: Instalación Cloud Functions en el proyecto

Posteriormente, se realizaron una serie de pasos para inicializar el proyecto de la Cloud Function. Se accedió a través del navegador y se autenticó la herramienta de Firebase (Figura 11.49).

```
MacBook-Pro-de-Jose:whereismybus joseyvimas$ firebase login
Already logged in as jyvimas@gmail.com
```

Figura 11.49: Comando para autenticar la herramienta de Firebase

Dentro de la ruta del proyecto (whereismybus), se ejecuta el comando `firebase init functions`. La herramienta ofrece una opción para instalar las dependencias con npm. Además se seleccionó el proyecto de Firebase por defecto, que será utilizado para ese directorio. En la Figura 11.50 se puede apreciar lo mencionado anteriormente.

```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Select a default Firebase project for this directory:
  [don't setup a default project]
  Proyecto-DAM (projecto-dam-94893)
  Instatrends (instatrends-aada0)
> WhereIsMyBus (whereismybus-a7ffe)
  BettermeBootcamp (bettermebootcamp)
  Instatrends-dev (instatrends-dev-dcfc8)
```

Figura 11.50: Configuración de la herramienta Cloud Functions y selección de proyecto Firebase a utilizar

Luego de seleccionar el proyecto de Firebase, la herramienta ofrece dos opciones de lenguajes para la creación de las funciones: *JavaScript* y *TypeScript* (Figura 11.51).

```
? Select a default Firebase project for this directory: WhereIsMyBus (whereismybus-a7ffe)

=== Functions Setup

A functions directory will be created in your project with a Node.js
package pre-configured. Functions can be deployed with firebase deploy.

? What language would you like to use to write Cloud Functions? (Use arrow keys)
> JavaScript
  TypeScript
```

Figura 11.51: Selección de JavaScript como lenguaje para escribir la Cloud Function

11.4.2 Desarrollo de la Cloud Function

Después de completar las tareas de instalación y configuración, se procedió a comenzar a agregar código en el directorio fuente. En el archivo *index.js* es donde se escriben el código de las funciones para ello se importan los módulos de Cloud Functions y el SDK de Admin con declaraciones *require* de Node y se inicializa una instancia de la app desde la cual se pueden realizar cambios a la Cloud Firestore (Figura 11.52).

```
// Las Cloud Functions por Firebase SDK para crear funciones en la nube y desencadenantes de configuración.
const functions = require('firebase-functions');

// El SDK del administrador de Firebase para acceder a la base de datos Cloud Firestore de Firebase.
const admin = require('firebase-admin');
admin.initializeApp();
```

Figura 11.52: Importación de módulos necesarios e inicialización de la app en el desarrollo de Cloud Function

Luego de importar los módulos se implementó la función *saveLocation* la cual permite ser consumida desde la pasarela SMS a través de la retransmisión de los mensajes recibidos en dicha pasarela. La comunicación entre la pasarela SMS y la función en la nube se realiza mediante peticiones POST, gracias a que las funciones en nube cuentan con una estructura *https onRequest* que ofrece Firebase functions. El hardware envía un mensaje con la información de la latitud y la longitud actual del autobús hacia la pasarela y luego está lo retransmite a la función en la nube la cual recibe ambos parámetros en dos constantes *lat* y *lng*, y almacena esos datos en Cloud Firestore en la colección *location* con su respectiva fecha y hora obtenida desde Firebase la cual permitirá mantener un orden entre los documentos agregados en dicha colección.

Si el almacenamiento es correcto se retorna OK 200 y si ocurre algún se retorna Internal Error 500. En la Figura 11.53 se puede apreciar el código implementado de la Cloud Function.

```

exports.saveLocation = functions.https.onRequest((request, response) => {
  return cors(request, response, () => {
    const message = request.body.message.split("&");
    const lat = message[0].split("=")[1];
    const lng= message[1].split("=")[1];

    return admin.firestore()
      .collection("location")
      .add({
        latitude: lat,
        longitude: lng,
        createdAt: admin.firestore.FieldValue.serverTimestamp(),
      })
      .then((docRef) => {
        const code = {
          code: 200,
          status: "complete"
        }
        return response.status(200).set('Content-Type', 'application/json').send(code);
      })
      .catch((err) => {
        return response.status(500).send(err)
      })
  });
});

```

Figura 11.53: Implementación de la Cloud Function saveLocation

Para poder ejecutar o consumir la función saveLocation es necesario hacer un despliegue de la misma en la nube de Firebase. Para ello es necesario ejecutar el comando que se muestra en la Figura 11.54.

```

MacBook-Pro-de-Jose:functions joseyvimas$ firebase deploy --only functions

=== Deploying to 'whereismybus-a7ffe'...

i deploying functions
i functions: ensuring necessary APIs are enabled...
✓ functions: all necessary APIs are enabled
i functions: preparing functions directory for uploading...
i functions: packaged functions (40.63 KB) for uploading
✓ functions: functions folder uploaded successfully
i functions: creating Node.js 6 function saveLocation(us-central1)...
✓ functions[saveLocation(us-central1)]: Successful create operation.
Function URL (saveLocation): https://us-central1-whereismybus-a7ffe.cloudfunctions.net/saveLocation

✓ Deploy complete!

```

Figura 11.54: Despliegue de la Cloud Function saveLocation

Con la URL de *saveLocation* generada por la CLI, dicha función puede ser consumida desde mediante una petición POST, la cual debe recibir como parámetro un *message* el cual contendrá la información sobre la latitud y la longitud a almacenar. La URL de la Cloud Function saveLocation se muestra a continuación:

<https://us-central1-whereismybus-a7ffe.cloudfunctions.net/saveLocation/>

Al ser consultada dicha URL, si todo es exitoso, en la colección *location* de Cloud Firestore se almacenaría un nuevo documento que contendrá un parámetro *latitude* y un parámetro *longitude* con los valores obtenidos en el parámetro *message* de la petición, además de la hora en la que se ejecute la función.

11.5 Análisis General del Prototipo de Hardware

El prototipo de hardware que tiene como responsabilidad la de obtener las coordenadas actuales del vehículo y transmitir las hacia un servicio, fue desarrollado utilizando módulos de GSM y GPS conectados a una placa de Desarrollo Arduino Uno.

La funcionalidad de este prototipo es la de recibir coordenadas geográficas del transporte donde esté conectado mediante el módulo GPS y luego transmitir las hacia una pasarela SMS, para este caso de estudio se utilizó SMSsync la cual retransmitirá el mensaje hacia la Cloud Function *saveLocation* haciendo uso de peticiones POST a la url de la misma, esto con la ayuda del módulo GSM.

11.5.1 Acoplamiento y Configuración de los Dispositivos de Hardware

Para la conexión de los módulos GPS y GSM con la placa Arduino Uno se usarán los puertos de comunicación serial como fue explicado en el Capítulo 7 en la sección de Hardware para Prototipos.

Es necesario instalar el IDE de Arduino para poder implementar código en la placa Arduino Uno. Para ello se debe descargar la última versión del IDE de Arduino desde: <http://arduino.cc/en/Main/Software>.

El IDE fue instalado en Windows, donde los pasos de instalación son muy sencillos. En la Figura 11.55 se puede apreciar algunos de los pasos de instalación.

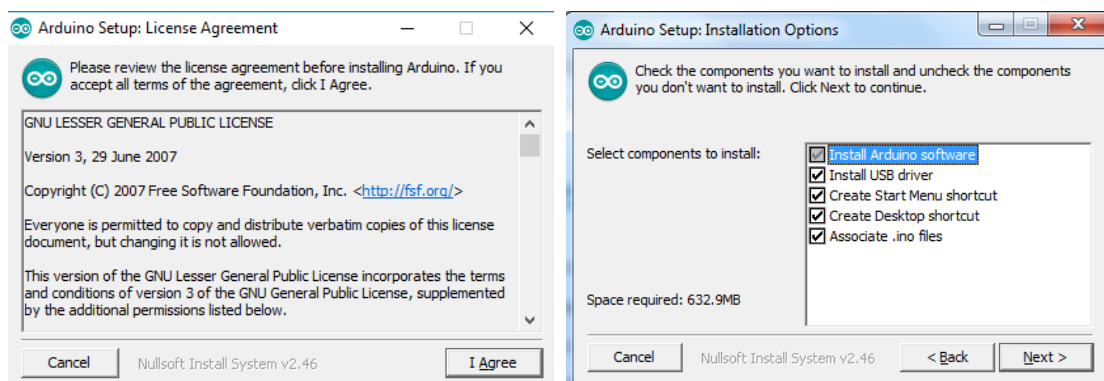


Figura 11.55: Instalación del IDE de Arduino en Windows

11.5.2 Desarrollo de Software para Prototipo de Hardware

Después de completar la configuración, se procedió a desarrollar el código con ayuda del IDE de Arduino. Lo primero que debe realizar es la importación de las librerías `SoftwareSerial` utilizada para el manejo de data serial, y la librería `TinyGPS++` la cual permite el uso correcto de las interfaces del módulo GPS. A continuación se muestra la importación de dichas librerías (Figura 11.56).

```
teg_whereismybus
1 #include <SoftwareSerial.h>
2 #include <TinyGPS++.h>
```

Figura 11.56: Importación de librerías requeridas

Una vez incluida las librerías, el siguiente paso es escoger los pines RX y TX donde se conectarán dichos módulos, en el presente caso estudio el módulo GPS está conectado a los pines 9 y 8 y el módulo GSM a los pines 10 y 11 (Figura 11.57).

```
4 //Conexión serial de los dispositivos
5 SoftwareSerial ss(9, 8);
6 SoftwareSerial sim(10, 11);
```

Figura 11.57: Asignación de pines a los módulos

Posteriormente se implementó una función `setup` que es la encargada de realizar todas configuraciones y comandos necesarios para garantizar que tanto el módulo GPS y GSM funcionen adecuadamente. En la Figura 11.58 se observa el código implementado en la función `setup`.

```
19 void setup() {
20   //delay de 7 segundos para asegurarse de que los módulos reciban la señal
21   delay(7000);
22
23   //GPS INIT
24   ss.begin(4800);
25
26   //GSM INIT
27   sim.begin(9600);
28   delay(1000);
29 }
```

Figura 11.58: Función `setup` encargada de inicializar cada uno de los módulos

Luego se crearon dos funciones principales del prototipo, la primera encargada de la obtención de las coordenadas geográficas (Figura 11.59) y la segunda cumple la

función de enviar dichas coordenadas mediante un mensaje de texto a un SMS Gateway para su posterior transmisión a la Cloud Function (Figura 11.60).

```
38 void getCoordinates() {
39     //Si han llegado caracteres del GPS,
40     //se envían al objeto TinyGPS
41     while (ss.available() > 0)
42         gps.encode(ss.read());
43
44     //Se obtiene la nueva ubicación
45     //y se almacena en las variables
46     Latitude = (gps.location.lat(), 6);
47     Longitude = (gps.location.lng(), 6);
48 }
```

Figura 11.59: Función getCoordinates encargada de obtener las coordenadas del dispositivo

```
50 void sendMessage() {
51     //Establecer el módulo GSM en modo texto
52     sim.println("AT+CMGF=1");
53     delay(1000);
54
55     //Número de teléfono móvil para enviar el mensaje
56     sim.println("AT+CMGS=\"" + number + "\"\r");
57     delay(1000);
58
59     //Enviando mensaje
60     String SMS = "wimb lat=" + Latitude + "&lon=" + Longitude;
61     sim.println(SMS);
62     delay(100);
63
64     sim.println((char)26); // Código ASCII de CTRL+Z
65     delay(1000);
66     _readSerial();
67 }
```

Figura 11.60: Función sendMessage encargada de enviar las coordenadas vía SMS

En la Figura 11.61 se puede apreciar la función principal del sistema que se ejecutará cada 30 segundos y es la encargada realizar la llamada a la función `getCoordinates` y `sendMessage` expuestas anteriormente.

```
32 void loop() {  
33   getCoordinates();  
34   sendMessage();  
35   delay(30000);  
36 }
```

Figura 11.61: Función principal del prototipo de hardware encargada ejecutar toda la lógica del negocio

11.6 Configuración de SMS Gateway

Se ingresó en el panel administrativo de la aplicación SMSSync³ el cual permite configurar los endpoints donde se retransmitirán los SMS. A continuación en la Figura 11.62 se aprecia cómo fue configurada la transmisión del SMS hacia la url "`https://us-central1-whereismybus-a7ffe.cloudfunctions.net/saveLocation/`" que es la función en la nube designada para almacenar la localización del autobús.

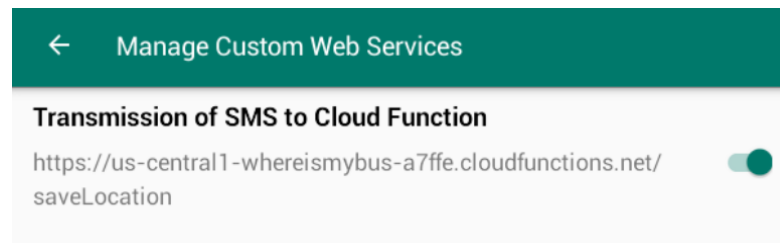


Figura 11.62: Configuración de url a retransmitir SMSSync

Adicionalmente este servicio permite la inclusión de palabras claves para poder filtrar los SMS no deseados a transmitir, por lo cual se decidió utilizar la palabra clave "wimb" al comienzo de todos los mensajes SMS que serán transmitidos tal como se aprecia en la Figura 11.63.

³ https://play.google.com/store/apps/details?id=org.addhen.smssync&hl=es_VE

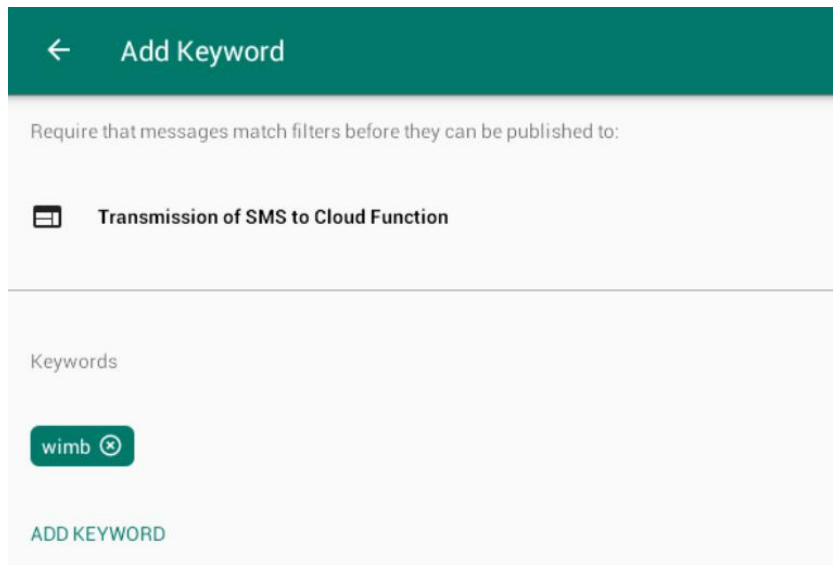


Figura 11.63: Palabras claves designadas para la transmisión de SMS

Para finalizar los autores escogieron un formato que debe poseer un mensaje correcto para su transmisión a la Cloud Function. Un ejemplo del mismo sería el siguiente:

wimb lat=10.496251&lon=-66.912998

12. Pruebas y Análisis de Resultados

En el presente capítulo se presentan las pruebas realizadas a la aplicación móvil y al prototipo de hardware, así como también los resultados obtenidos, permitiendo así evaluar el nivel de aceptación del proyecto desarrollado y si cumple con los objetivos y funciones específicas para los cuales fue diseñado.

12.1 Pruebas de Aceptación

Antes de que un software comience a utilizarse y entrar a la fase conocida como *producción* es necesario evaluar si las capacidades, soluciones y ventajas que ofrece frente una problemática planteada realmente justifican su uso.

En base a la solución tecnológica general del presente proyecto, la aplicación móvil es la que se encuentra dirigida a los usuarios finales, por ello se sometió a una evaluación para determinar y verificar si la misma cumple con los requerimientos establecidos en el diseño.

Las pruebas de aceptación fueron realizadas mediante una encuesta técnica a un grupo seleccionado de usuarios finales comprendido por veinte (20) personas de diferentes géneros, edades, nivel social y nivel de instrucción con el fin de conocer la opinión general sobre el uso y funcionamiento de la aplicación móvil.

Es necesario aclarar que a pesar de que existen tres tipos de usuarios en la aplicación móvil, estas pruebas fueron realizadas con el enfoque del rol de **representante**, esto debido a que la problemática que resuelve el proyecto afecta principalmente a este tipo de usuario.

En la lista que se muestra a continuación se presentan las dieciocho (18) preguntas que fueron formuladas en la encuesta:

1. ¿Considera que la distribución de los elementos es adecuada de tal manera que ningún momento se observan pantallas sobrecargadas de elementos?
2. ¿Considera adecuada la navegación en la aplicación tomando en cuenta la presencia permanente del botón de regreso y botones de navegación entre vistas?
3. ¿Le forma en qué se muestran metáforas en los textos de la aplicación móvil es adecuada?
4. ¿El menú de la aplicación siempre está presente y muestra sus opciones de forma legible?
5. ¿Los títulos de cada vista describen de manera concisa que funcionalidad u objetivo tiene dicha vista?

6. ¿Qué nivel de dificultad considera que tuvo el seleccionar la dirección de parada?
7. ¿Qué tan útil le pareció la funcionalidad de envío de mensajes del conductor?
8. ¿Se muestra de manera legible el tiempo estimado de llegada del representado?
9. ¿Considera que el mapa de ubicación en tiempo real del autobús cumple adecuadamente con la premisa de mostrar un ícono que representa al autobús y un ícono que representa la parada designada?
10. ¿Recomendaría esta aplicación para ser usada por la institución educativa de su representado o el de algún familiar?

Esta prueba fue realizada en dos iteraciones. En la primera iteración se obtuvieron resultados que motivaron a los autores realizar correcciones y/o mejoras en la aplicación móvil. Una vez finalizadas dichas correcciones se realizó otra iteración con los mismos participantes obteniendo una mejora en los resultados.

A continuación se presentarán las Figuras 12.1 hasta la Figura 12.8, las cuales representan los resultados obtenidos de la segunda iteración por cada una de las preguntas en su respectivo orden.

1 ¿Considera que la distribución de los elementos es adecuada de tal manera que en ningún momento se observan pantallas sobrecargadas de elementos?

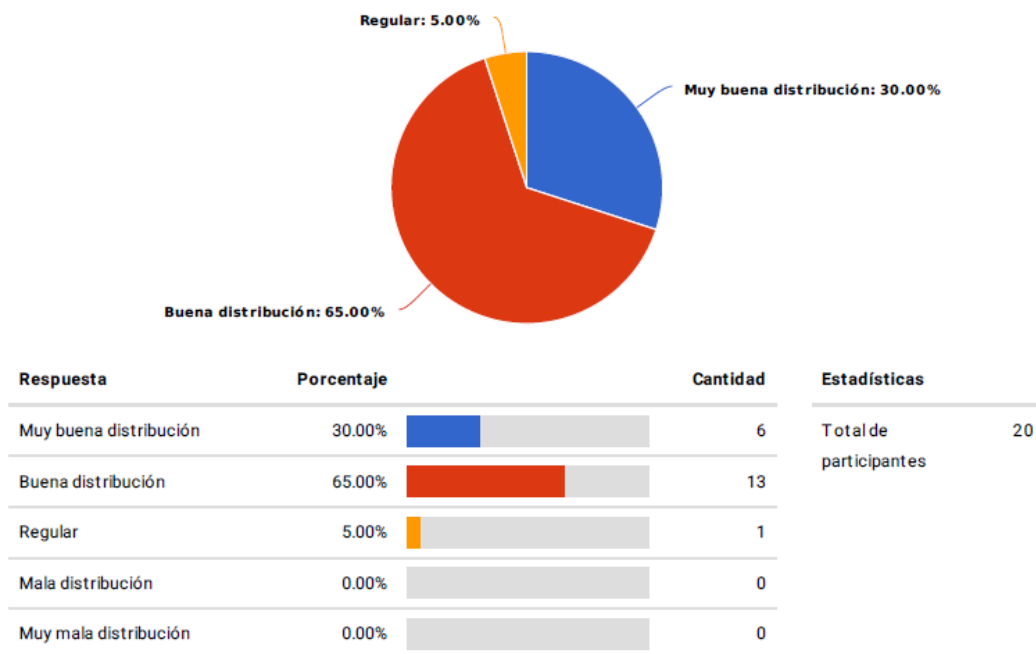
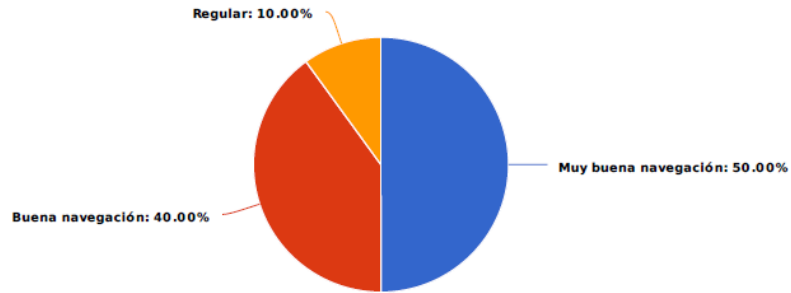


Figura 12.1: Pregunta 1 – Prueba de Aceptación

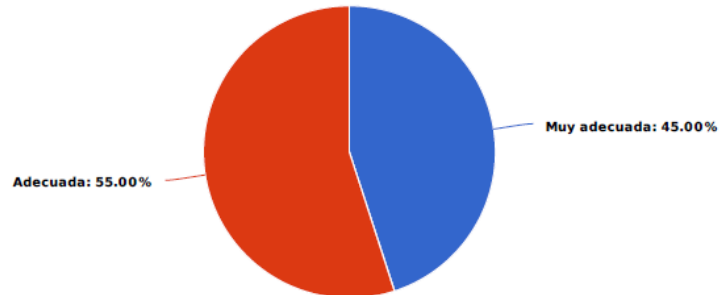
2 ¿Considera adecuada la navegación en la aplicación tomando en cuenta la presencia permanente del botón de regreso y botones de navegación entre vistas?



Respuesta	Porcentaje	Cantidad	Estadísticas
Muy buena navegación	50.00%	10	Total de participantes 20
Buena navegación	40.00%	8	
Regular	10.00%	2	
Mala navegación	0.00%	0	
Muy mala navegación	0.00%	0	

Figura 12.2: Pregunta 2 – Prueba de Aceptación

3 ¿La forma en que se muestran metáforas en los textos de la aplicación móvil es adecuada?



Respuesta	Porcentaje	Cantidad	Estadísticas
Muy adecuada	45.00%	9	Total de participantes 20
Adecuada	55.00%	11	
Regular	0.00%	0	
Mal adecuada	0.00%	0	
Inadecuada	0.00%	0	

Figura 12.3: Pregunta 3 – Prueba de Aceptación

4 ¿El menú de la aplicación siempre está presente y muestra sus opciones de forma legible?

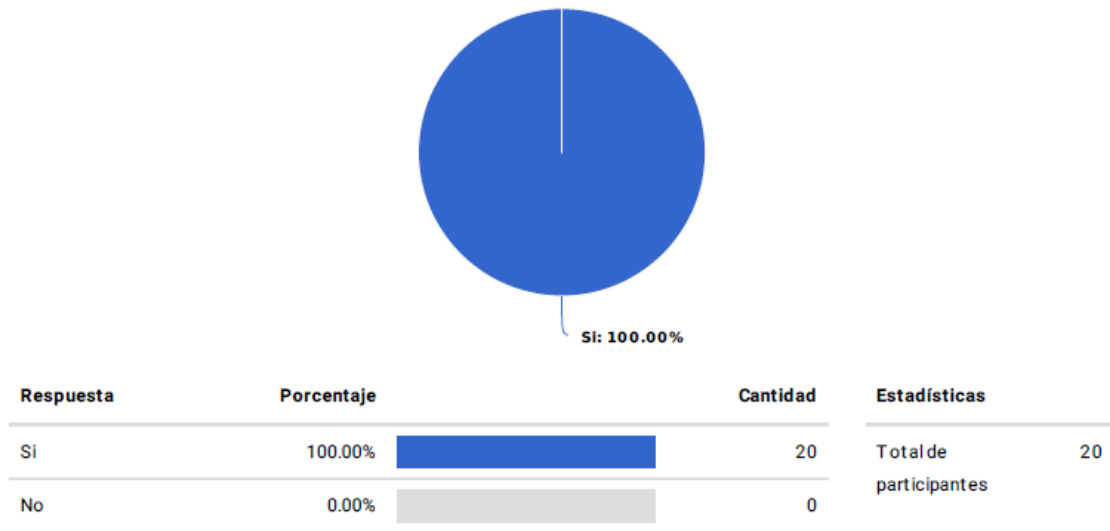


Figura 12.4: Pregunta 4 – Prueba de Aceptación

5 ¿Los títulos de cada vista describen de manera concisa que funcionalidad u objetivo tiene dicha vista?

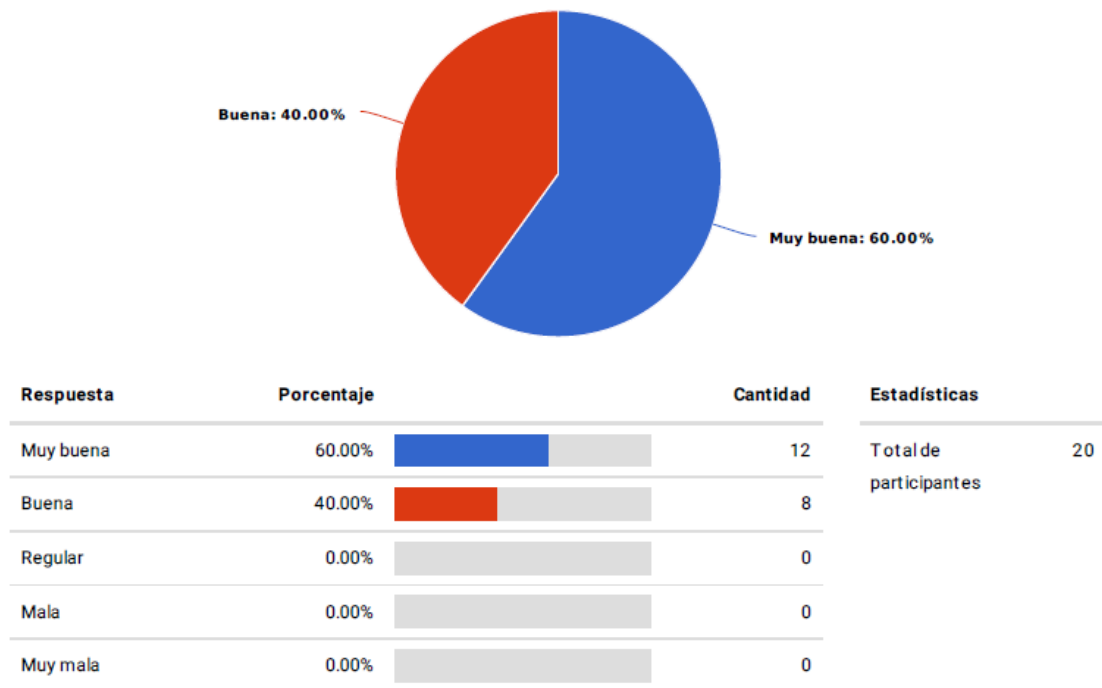
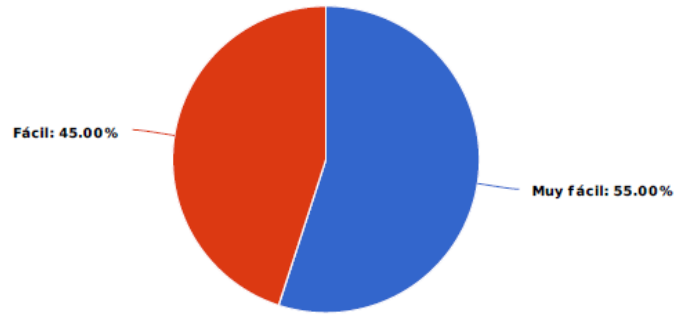


Figura 12.5: Pregunta 5 – Prueba de Aceptación

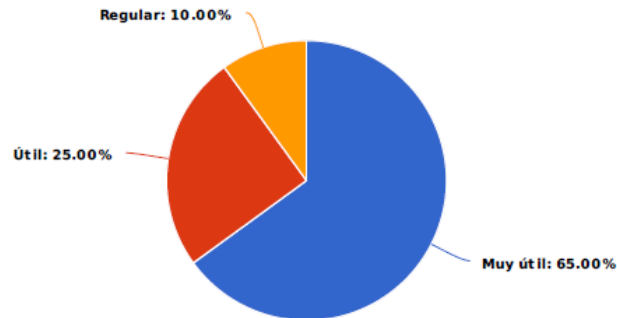
6 ¿Qué nivel de dificultad considera que tuvo el seleccionar la dirección de parada?



Respuesta	Porcentaje	Cantidad	Estadísticas
Muy fácil	55.00%	11	Total de participantes 20
Fácil	45.00%	9	
Regular	0.00%	0	
Difícil	0.00%	0	
Muy difícil	0.00%	0	

Figura 12.6: Pregunta 6 – Prueba de Aceptación

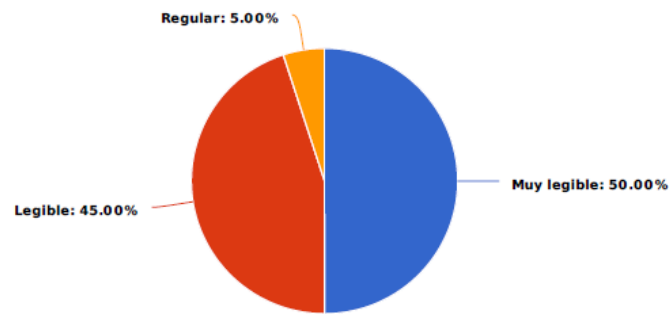
7 ¿Que tan útil le pareció la funcionalidad de envío de mensajes del conductor?



Respuesta	Porcentaje	Cantidad	Estadísticas
Muy útil	65.00%	13	Total de participantes 20
Útil	25.00%	5	
Regular	10.00%	2	
Poco útil	0.00%	0	
Inútil	0.00%	0	

Figura 12.7: Pregunta 7 – Prueba de Aceptación

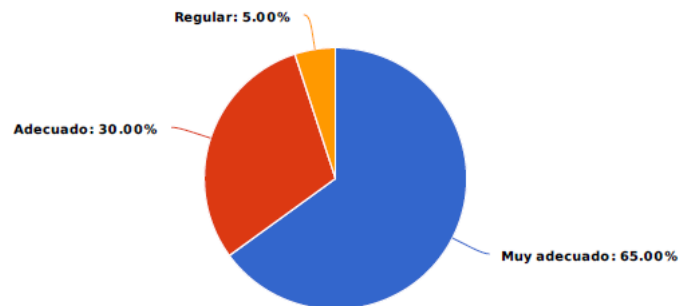
8 ¿Se muestra de manera legible el tiempo estimado de llegada del representado?



Respuesta	Porcentaje	Cantidad	Estadísticas
Muy legible	50.00%	10	Total de participantes 20
Legible	45.00%	9	
Regular	5.00%	1	
Poco legible	0.00%	0	
Ilegible	0.00%	0	

Figura 12.8: Pregunta 8 – Prueba de Aceptación

9 ¿Considera que el mapa de ubicación en tiempo real del autobús cumple adecuadamente con la premisa de mostrar un ícono que representa al autobús y un ícono que representa la parada designada?



Respuesta	Porcentaje	Cantidad	Estadísticas
Muy adecuado	65.00%	13	Total de participantes 20
Adecuado	30.00%	6	
Regular	5.00%	1	
Mal adecuado	0.00%	0	
Inadecuado	0.00%	0	

Figura 12.9: Pregunta 9 – Prueba de Aceptación

10 ¿Recomendaría esta aplicación para ser usada por la institución educativa de su representado o el de algún familiar?

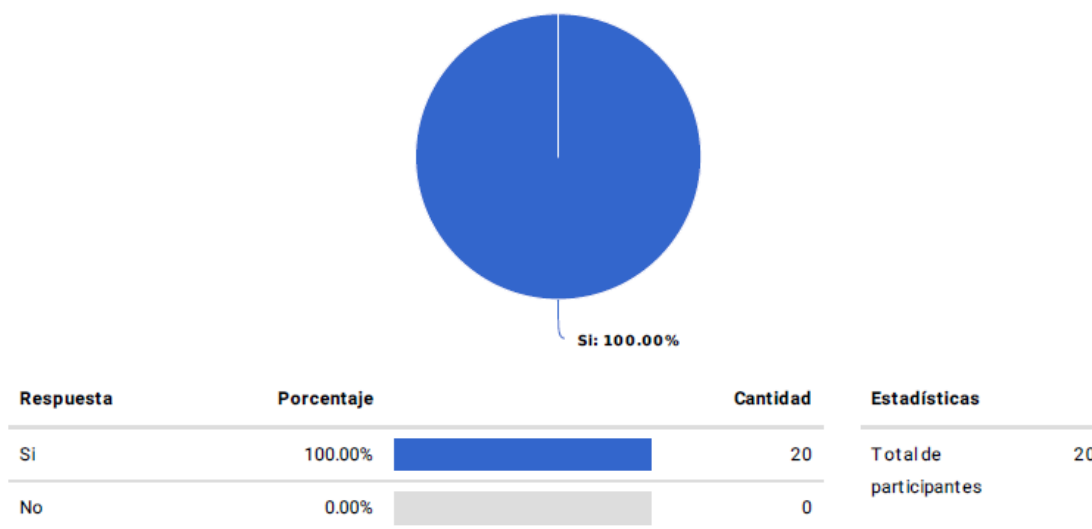


Figura 12.10: Pregunta 10 – Prueba de Aceptación

Luego del análisis de cada uno de los gráficos resultantes por parte de los autores del proyecto, de manera general, se puede evidenciar que las respuestas proporcionadas por las personas que usaron la aplicación móvil fueron positivas, por lo cual se puede decir que la misma cumple con los requerimientos funcionales establecidos y con el alcance definido en un principio.

Sin embargo, en algunas de las preguntas formuladas se obtuvo un porcentaje de respuestas regulares que, si bien no representa la mayoría de las opiniones, es importante tomar en cuenta para llevar a cabo mejoras o correcciones en trabajos futuros o en caso de proceder a pasar el proyecto a una fase de producción.

12.1 Pruebas de Funcionalidad

12.1.1 Estado del Estudiante

Para realizar pruebas en la aplicación móvil sobre el estado del estudiante, los autores fijaron todos los posibles casos que pueden presentar el estado del estudiante. Un estudiante posee 3 estados (Ausente, Llegó a su hogar, En el autobús) como se han explicado con anterioridad en este documento. Estos estados únicamente pueden ser cambiados por un usuario con el rol chofer mediante la pantalla del listado de estudiantes como se aprecia en la Figura 12.11.

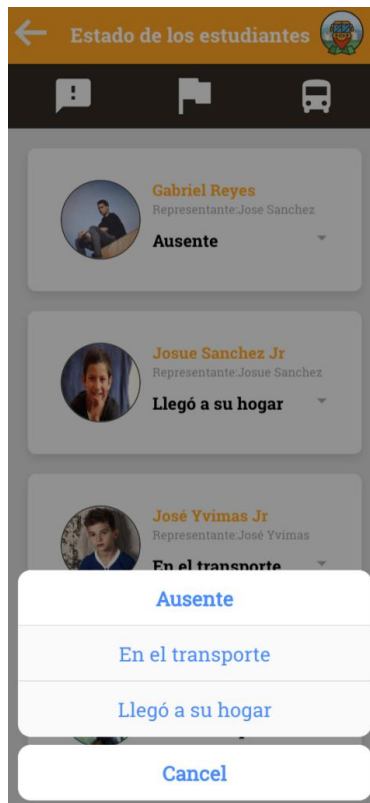


Figura 12.11: Estado de los estudiantes – Prueba de Funcionalidad

Al momento de que el chofer realiza un cambio del estado, este se actualiza instantáneamente en la base de datos, por consecuente en la pantalla del estado de mi representado (esta pantalla es visible para un usuario de tipo representante) se verá reflejado dicho cambio.

Un aporte importante a mencionar es que todo estudiante siempre posee un estado al momento de su creación por el administrador, se le asigna por defecto el estado Llegó a su hogar.

Se realizó una prueba iniciando sesión con un usuario chofer y se asignaron todos los valores posibles del estado del estudiante y a su vez verificando su correcta renderización en la pantalla que observa el representante, garantizando así la consistencia de la información en todo momento de la aplicación móvil. A continuación se muestran todos los posibles estados que un representante podrá observar en dicha pantalla (Figura 12.12).

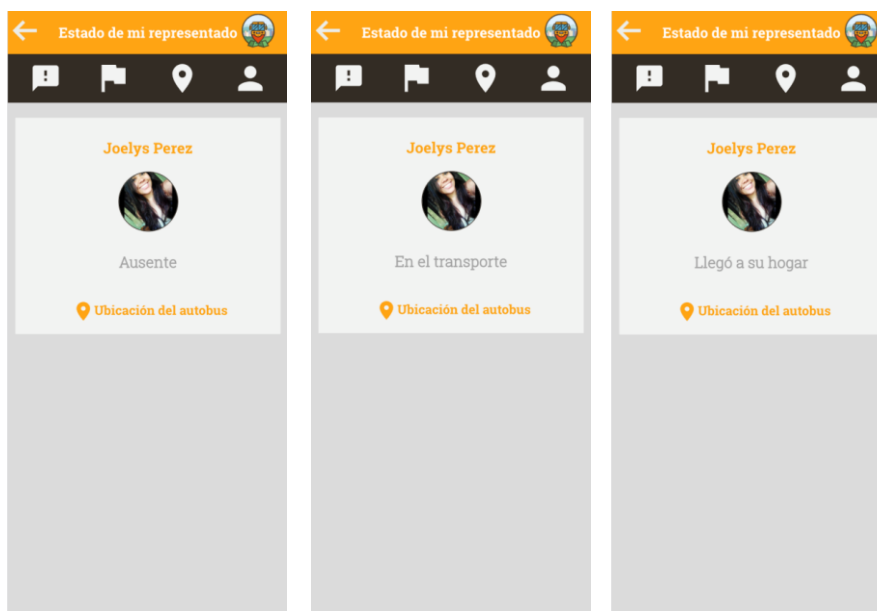


Figura 12.12: Estado de mi representado – Prueba de Funcionalidad

Luego de realizada la prueba de funcionalidad del estado del representado los autores observaron que el estado del representado fue consistente en todo momento y de acuerdo a los casos planteados no mostró fallas, por lo que se afirma que la prueba fue satisfactoria.

12.1.2 Mensajes del Chofer

Para esta prueba se enfocó en la funcionalidad que posee el chofer de enviar mensajes a los representantes que se encuentren en la aplicación móvil. Un chofer puede enviar cualquier tipo de mensaje sin límites de caracteres o símbolos a los representantes, solo no se permite el envío de mensajes sin algún carácter escrito. De llegar ocurrir este caso la aplicación desplegará una alerta informando al chofer que no puede enviar mensajes en blanco, tal como se aprecia en la Figura 12.13



Figura 12.13: Alerta de mensaje vacío – Prueba de Funcionalidad

Con respecto a los representantes tienen 2 estados validos en la aplicación el primero es cuando no poseen mensajes y se muestra un texto indicando dicha información como se observa en la Figura 12.14.

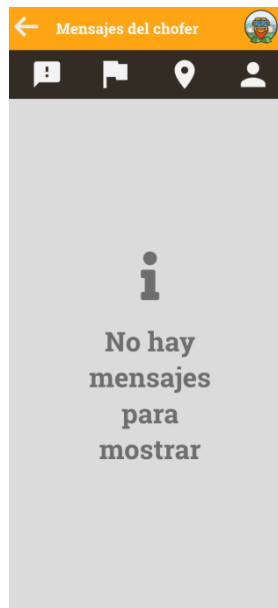


Figura 12.14: Sin mensajes del chofer – Prueba de Funcionalidad

El otro estado posible es el de mostrar todo el listado de los mensajes existentes enviados por el chofer de la unidad tal como se aprecia en la Figura 12.15.



Figura 12.15: Listado de mensajes del chofer – Prueba de Funcionalidad

Esta prueba consistió de iniciar sesión con un usuario chofer y probar los 2 casos existentes de enviar un mensaje (un mensaje vacío y uno con información) obteniendo los valores esperados como se mostraron anteriormente en este capítulo. De igual forma se observó los cambios que se presentan en un usuario con el rol representante al momento de desplegarse nuevos mensajes.

Luego de realizada la prueba, los autores observaron que para el caso de los usuarios con rol de chofer la aplicación permite correctamente el envío de mensajes, a menos que el mismo se encuentre vacío y en este caso la aplicación muestra un mensaje de retroalimentación para indicar dicho error. En cambio, para los usuarios representantes se muestra de manera consistente un mensaje informando que no hay mensajes por el momento o un listado con todos los mensajes existentes. Debido a esto, se afirma que se cumplieron satisfactoriamente los casos planteados de envío de diferentes mensajes y la renderización adecuada de los mensajes para los usuarios.

12.1.3 Geolocalización

Para realizar pruebas en la aplicación móvil sobre el monitoreo en tiempo real del transporte escolar con el uso de la tecnología GPS, para este presente trabajo, se decidió utilizar un vehículo propio en donde se utilizó una cuenta con rol de chofer

con el fin de utilizar el rastreo de contingencia para la obtención y transmisión de coordenadas. Además se contó con una cantidad de 5 personas a bordo del vehículo y 5 personas adicionales que realizaban el rol de representantes.

Todos estos usuarios fueron creados y registrados en la aplicación para realizar las pruebas correctamente. Los representantes se encontraban conectados a la aplicación observando el mapa y prestando mayor atención en los tiempos aproximados de llegada y la localización del vehículo en el mapa en cuestión.

La intención de esta prueba era verificar si los tiempos aproximados mostrados por la aplicación eran correctos al momento del vehículo llegar a la parada designada y a su vez verificar si en el mapa de la aplicación se podía observar con claridad como el vehículo hacía el recorrido. El mapa se actualizaba en intervalos de 30 segundos.

Un punto muy importante al momento de realizar las pruebas fue el hecho de los recursos a consumir ya que al estar utilizando tecnologías Open Source en sus modalidades gratis se tienen ciertos requerimientos que cumplir como lo son:

- Un límite de peticiones al API de Open Route Service 2500 peticiones al día.
- Un límite de cambios en el mapa provisto por Leaflet en intervalos de tiempo de 1 minuto.
- La duración de la batería del dispositivo móvil del chofer.
- La cantidad de datos de navegación del dispositivo móvil del chofer para poder transmitir las coordenadas a la Cloud Function.

Los autores previamente a las pruebas realizaron un esquema de la ruta que tomará el vehículo incluyendo cada una de las paradas designadas como se aprecia en la Figura 12.16.

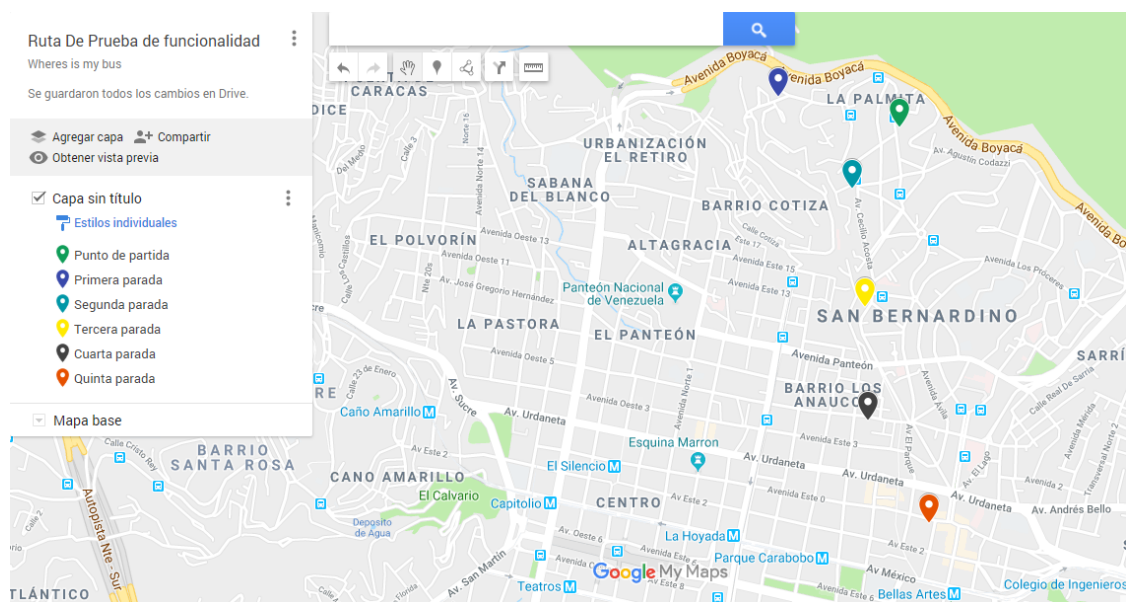


Figura 12.16: Esquema de ruta – Prueba de Geolocalización

Además los autores repitieron esta ruta 10 veces para poder conseguir tiempos promedios estimados de duración del viaje, estos serán utilizados al momento de comparar si los resultados obtenidos en las pruebas son correctos o no.

Adicionalmente hay que destacar que este recorrido se realizó en días de baja afluencia de tráfico y en horas de la mañana, donde esta zona es poco transitada, el recorrido completo se puede realizar en promedio de 15 minutos. A continuación en la Tabla 12.1 se observan los tiempos estimados.

Paradas	Tiempo promedio de llegada	Tiempo promedio de recepción del estudiante y cambio de estado en la aplicación	Tiempo total estimado
Primera parada (187 Calle Parque Forestal, Caracas 1010, Distrito Capital, Venezuela)	3 minutos	1 minuto	4 minutos
Segunda parada (Metrobus IUTI, Av. López Méndez, Caracas 1011, Distrito Capital, Venezuela)	5 minutos	1 minuto	6 minutos
Tercera parada (Metrobus Diagnóstico, Caracas 1010, Distrito Capital, Venezuela)	8 minutos	1 minuto	9 minutos
Cuarta parada (Avenida Este 5, Caracas 1011, Distrito Capital, Venezuela)	10 minutos	1 minuto	11 minutos
Quinta parada (Avenida Este 0, Caracas 1011, Distrito Capital, Venezuela)	14 minutos	1 minuto	15 minutos

Tabla 12.1: Tiempos estimados – Prueba de Geolocalización

Luego de tener los tiempos estimados se aplicaría la prueba con la aplicación para verificar que tan cercanos o no están los tiempos estimados que arroja en relación al tiempo promedio estimado calculado por los autores.

Para poder evaluar los resultados, los autores decidieron tomar como un resultado correcto que el tiempo real de llegada no difiera al tiempo promedio estimado de la tabla anterior en al menos 2 minutos. Otro resultado correcto es que el marcador del vehículo siempre estuvo presente en el mapa y en constante movimiento en los intervalos asignados.

En contra parte un resultado incorrecto es aquel que difiere del tiempo estimado en más de 2 minutos o cuando el indicador del mapa no cumplió el funcionamiento esperado.

La prueba de geolocalización se tuvo que repetir 3 veces debido a que el primer día los autores tuvieron muchos resultados incorrectos que los motivaron a ajustar ciertas secciones de los módulos de la aplicación.

En la Tabla 12.2 se aprecian los resultados del primer día de pruebas y en la Tabla 12.3 se muestran los resultados del tercer día de hacer las pruebas de geolocalización en la ruta mencionada anteriormente.

Paradas	1	2	3	4	5
Tiempo real que tardó en llegar a la parada.	2 min	6 min	10.5 min	13 min	18 min
Tiempo estimado por la aplicación que tardó en llegar a la parada.	0 min	5 min	10 min	15 min	10 min
Valor estimado por los autores como correcto	4 min	6 min	9 min	11 min	15 min
Estado del marcador del transporte	Presente	Ausente	Presente	Presente	Presente
Resultado	Incorrecto	Incorrecto	Correcto	Incorrecto	Incorrecto

Tabla 12.2: Primer día de pruebas – Prueba de Geolocalización

Paradas	1	2	3	4	5
Tiempo real que tardó en llegar a la parada.	3 min	6 min	9 min	12 min	16.5 min
Tiempo estimado por la aplicación que tardó en llegar a la parada.	3 min	5 min	9 min	11 min	18 min
Valor estimado por los autores como correcto	4 min	6 min	9 min	11 min	15 min
Estado del marcador del transporte	Presente	Presente	Presente	Presente	Presente
Resultado	Correcto	Correcto	Correcto	Correcto	Incorrecto

Tabla 12.3: Tercer día de pruebas – Prueba de Geolocalización

En la Figura 12.17 se muestra un gráfico de barras con el porcentaje general de resultados correctos e incorrectos de las pruebas que se realizaron en los 3 días utilizando la ruta designada.

Resultados de prueba de funcionalidad de geolocalización

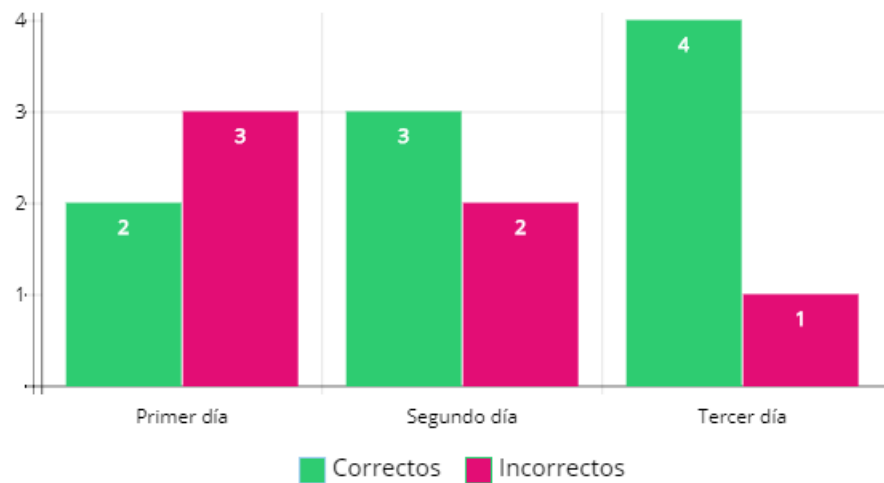


Figura 12.17: Resultado de prueba de los tres días – Prueba de Geolocalización

Finalmente se puede apreciar que luego de los ajustes realizados por los autores se notó una mejoría con respecto a los resultados que arroja la aplicación en el tiempo estimado de llegada y el estado del marcador de transporte. Por lo cual se puede concluir que al utilizar la base datos en tiempo real Cloud Firestore y la comunicación con el API de Open Route Services y Leaflet es una opción factible para implementar funcionalidades de geolocalización pese a las limitantes que posee, ya que se obtuvo una cantidad superior de resultados correctos con respecto a los resultados incorrectos.

13. Conclusiones y Trabajos Futuros

El transporte escolar es un servicio que ofrecen algunas comunidades escolares a los representantes para que sus representados sean llevados hacia la institución y de regreso a su hogar. Durante el viaje surge el problema en donde los representantes no poseen conocimiento de donde se encuentra su estudiante, el estado del mismo y cualquier otra eventualidad referente a la unidad o al tráfico de la ciudad.

En el presente trabajo, se detallaron conceptos, herramientas y tecnologías que sirvieron para establecer una base sólida en la ejecución del proyecto, siendo importantes y necesarias en el desarrollo de la aplicación móvil, la cual permite a los representantes de los estudiantes de instituciones educativas poder monitorear y tener información referente al mismo, como el estatus de su representado en la unidad de transporte (asistencia), tiempo estimado de llegada a su destino y mensajes/notificaciones en caso de inconvenientes.

A pesar de los problemas que surgieron durante el desarrollo del proyecto y teniendo en cuenta los resultados finales, se puede afirmar que se cumplió el objetivo general propuesto, ya que los autores lograron realizar una aplicación móvil con el uso de tecnologías de desarrollo híbrido que a su vez consumían API's de terceros para ciertas funcionalidades, una comunicación en tiempo real con la base de datos, el despliegue de funciones en la nube que almacenaban los datos pertinentes en la base de datos y la realización de un sistema capaz de estar en un autobús escolar y enviar las coordenadas del mismo hacia la nube.

Mediante las reuniones realizadas por los autores del proyecto se logró determinar cada uno de los requerimientos funcionales y no funcionales de la aplicación móvil, para luego ser utilizados en el desarrollo de la aplicación móvil y posteriormente validados con la realización de pruebas de aceptación.

En el caso del segundo objetivo específico fue necesario crear el prototipo de hardware, encargado de obtener las coordenadas geográficas actuales de la unidad de transporte utilizando un módulo GPS conectado a la placa Arduino y una vez obtenidas dichas coordenadas son enviadas vía SMS utilizando el módulo GPS hacia un SMS Gateway que retransmitirá la información hacia el servicio en la nube. En el desarrollo de este módulo se presentaron limitantes como las baterías necesarias para la alimentación de los módulos y desperfectos en uno de los módulos utilizados, por lo cual no se logró realizar una integración completa del prototipo de hardware con la aplicación y el SMS Gateway. Como solución, los autores desarrollaron la funcionalidad de rastreo de contingencia lo cual permitió mitigar las fallas presentadas con dicho prototipo y lograr cumplir con la misión del mismo.

Se logró instalar y configurar una base de datos que tuviera agentes de escucha en tiempo real como Cloud Firestore para el almacenamiento de la posición del autobús y demás información utilizando 4 colecciones para almacenar toda la información del sistema (users, students, locations, messages). Esto luego se validó con las pruebas de funcionalidades ya que estas requieren de la persistencia de la base de datos.

También se logró la retransmisión de SMS a través de un software SMSsync el cual es un SMS Gateway capaz de obtener la información de un SMS y enviar la información a una URL previamente asignada, con el fin de poder enviar las coordenadas transmitidas por el prototipo de hardware hacia la nube.

Así mismo se logró realizar la instalación y configuración del entorno de Cloud Functions, donde se despliegan funciones codificadas por un usuario y fácilmente accesibles por una URL. Se creó una función que se encarga de recibir por parámetros las coordenadas geográficas del autobús para almacenarlas en la colección locations de la base de datos Cloud Firestore. Posteriormente se validó si el funcionamiento era correcto al realizar pruebas de geolocalización en las que es indispensable esta funcionalidad.

Se logró realizar el desarrollo de la aplicación móvil con el Framework Ionic V4 con la finalidad de tener un software híbrido compatible para las plataformas iOS y Android, apoyado en el uso de productos de Firebase para la creación de la base de datos, autenticación, servicios en la nube y almacenamiento de archivos. De igual forma se utilizó funciones de Leaflet para el manejo de mapas y de Open Route Service (ORS) para las rutas, cálculos de tiempo y distancia desde un punto a otro.

Se realizaron pruebas de aceptación y de ciertas funcionalidades consideradas importantes en la aplicación. Para la prueba de aceptación, luego de observar los resultados de la primera iteración, los autores decidieron realizar las correcciones y mejoras necesarias con el fin de realizar una nueva iteración. En los resultados de la segunda iteración se observó una mejoría, reflejando que las correcciones realizadas mejoraron la experiencia del usuario.

En el caso de las pruebas de funcionalidades, para la prueba de geolocalización luego de haber analizado los resultados obtenidos se determinaron una serie de errores y eventos no esperados en la funcionalidad de geolocalización, los cuales fueron solventados mediante la configuración correcta de las herramientas Open Route Service y Leaflet. Luego de realizadas las pruebas de funcionalidad de mensajería de la aplicación y el estado del representado, se observó que la aplicación mantiene un estado correcto aún en los casos bordes, logrando así verificar cada una de las funcionalidades primordiales de la aplicación.

Aunque hubo ciertas dificultades en cuanto a conseguir los equipos necesarios para el desarrollo de la aplicación y el prototipo de hardware, estas fueron superadas en el proceso del desarrollo. De esta manera se puede concluir de forma positiva que la aplicación móvil desarrollada cumple con los objetivos planteados y satisface los requerimientos funcionales que fueron descritos.

No obstante como trabajos futuros, se propone la experiencia contenida en este trabajo considerando adicionalmente:

- Incorporar baterías para la alimentación de la placa de desarrollo junto con sus módulos y realizar las respectivas pruebas de integración en un transporte escolar.

- Usar herramientas pagas para la geolocalización o que no posean muchas limitaciones, para así aumentar la cantidad de representados y mejorar la precisión de la data obtenida.
- Notificación push al representante cuando su representado se encuentre cerca de su parada.
- Líneas de marcaje de la ruta tomada por el conductor de la unidad.
- Permitir la comunicación del prototipo de hardware mediante el uso de GPRS o SMS de acuerdo a cual es la mejor alternativa para transmisión de los datos teniendo en cuenta la cobertura telefónica.
- Sección donde se le indique al chofer cuál representado posee alguna discapacidad, para que así esté atento y preste una mayor colaboración.

Referencias

- [1] S. Mansuri, «Yelowsoft,» 2018. [En línea]. Available: <https://www.yelowsoft.com/blog/top-6-reasons-why-school-bus-tracking-system-is-a-must-have-for-academic-institutes/>.
- [2] C. T. Moro, «Dispositivos Móviles y Multimedia,» 2009. [En línea]. Available: multimedia.uoc.edu.
- [3] «Desarrollo de Aplicaciones Móviles,» 2018. [En línea]. Available: <http://estudiowam.com/desarrollo-de-aplicaciones-moviles/>.
- [4] «Aplicaciones Híbrida: ¿Qué son y cómo usarlas?,» 2018. [En línea]. Available: <https://www.nextu.com/blog/aplicaciones-hibridas-que-son-y-como-usarlas/>.
- [5] «Los 3 tipos de aplicaciones móviles: ventajas e inconvenientes,» 2014. [En línea]. Available: <https://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/>.
- [6] L. Huanca, «Desarrollo de aplicaciones móviles híbridas con Ionic,» Tarija, Bolivia, 2017.
- [7] «What is an API?,» 3scale, 2012.
- [8] «Leaflet,» 2018. [En línea]. Available: <https://leafletjs.com>.
- [9] «openrouteservice,» 2018. [En línea]. Available: <https://openrouteservice.org/services/>.
- [10] «¿Qué son las bases de datos?,» 2007. [En línea]. Available: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>.
- [11] E. Suárez, «¿Qué es una base de datos relacional?,» Agosto 2008. [En línea]. Available: <http://www.uprh.edu/adem/Base%20de%20datos%20relacional.pdf>.
- [12] A. Lith y J. Mattsson, «Investigating storage solutions for large data,» Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden, 2010.
- [13] «Firebase,» 2018. [En línea]. Available: <https://firebase.google.com>.
- [14] «Firebase Authentication,» 2018. [En línea]. Available: <https://firebase.google.com/docs/auth>.
- [15] «Cloud Firestore,» 2018. [En línea]. Available: <https://firebase.google.com/docs/firestore/>.
- [16] «Cloud Functions para Firebase,» 2018. [En línea]. Available: <https://firebase.google.com/docs/functions/>.
- [17] «Cloud Storage,» 2018. [En línea]. Available: <https://firebase.google.com/docs/storage/>.
- [18] R. Sánchez, «Aplicación de geolocalización Form IGP iOS/Android,» 2012.
- [19] «Latitud y longitud,» 2000. [En línea]. Available: <http://www.educaplus.org/game/latitud-y-longitud>.
- [20] «What is Arduino?,» 2018. [En línea]. Available: <https://www.arduino.cc/en/Guide/Introduction>.

- [21] «Adafruit Ultimate GPS Breakout - 66 channel w/10 Hz updates - Version 3,» 2018. [En línea]. Available: <https://www.adafruit.com/product/746>.
- [22] R. De La Camara, «Arduino + módulo GSM/GPRS: monitorización, automatización y gestión remota en un viñedo.,» Universitat Oberta de Catalunya, Barcelona, España, 2017.
- [23] L. Chacón, «Scrum,» Patrones de Diseño, Facultad de Ciencias, UCV.
- [24] S. Bustos, «Metodología Scrum aplicada a través de un software de consulta de las listas OFAC y ONU de Topbls en las Pyme Colombianas,» Universidad Nacional Abierta y a Distancia, Bogota, Colombia, 2014.