

**DESARROLLO DE SOFTWARE Y HARDWARE PARA EL
DISPOSITIVO BIGO SAFE EN LA EMPRESA CINNOV
S.A.S. EN LA CIUDAD DE BOGOTÁ**

Autor:

Brayan Daniel Sarmiento Iscala

Director:

Cristhian Ivan Riaño Jaimes

TRABAJO DE GRADO

Junio, 2022

UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍA Y ARQUITECTURA
2022

Índice

1. Introducción	9
1.1. RESUMEN	9
1.2. ABSTRACT	9
1.3. JUSTIFICACIÓN	10
1.4. CONTEXTO Y CONDICIONES DEL ENTORNO DE DESARROLLO	10
1.4.1. Ingreso a la empresa y conocimientos básicos.	10
1.4.2. Capacitaciones y adaptación de entorno para desarrollar las tareas.	11
1.5. PRODUCTO DE LA PASANTÍA	11
1.5.1. Entorno laboral	11
1.5.2. Desarrollo de nuevo dispositivo	12
1.6. DEFINICIÓN DEL PROBLEMA	12
1.7. CONTEXTO Y CONDICIONES DEL PROBLEMA	12
1.7.1. Actualización de los diversos paquetes, plugins y librerías de la aplicación móvil	12
1.7.2. Actualización en el firmware de Bigo para permitir recibir indicaciones de	
GPS enviadas a través de la aplicación móvil	12
1.7.3. Desarrollo de un nuevo dispositivo de seguridad vial	13
1.8. OBJETIVOS	13
1.8.1. Objetivo general	13
1.8.2. Obejtivo específico	13
2. Estructura del documento	14
2.1. CAPITULO I, Introducción	14
2.2. CAPITULO II, Marco Teórico y Estado del Arte	14
2.3. CAPITULO III, Metodología y plan de trabajo	14
3. Estado del Arte y Marco Teórico	15
3.1. IOT	15
3.2. BIG DATA	15
3.3. DISEÑO ASISTIDO POR COMPUTADORA (CAD)	15
3.4. DISEÑO ELECTRÓNICO	16
3.5. FRAMEWORK	16
3.6. ARQUITECTURA MVC	16
3.7. PROGRAMACIÓN ORIENTADA A OBJETOS	17
3.8. ESTADÍSTICAS NACIONALES SOBRE VÍCTIMAS DE SINIESTROS	17
3.9. CIUDADES INTELIGENTES	18
3.10. INGENIERÍA DE SOFTWARE	18
3.11. DESARROLLO DE SOFTWARE	18
3.12. DESARROLLO MÓVIL	19
3.13. APOYO A PROYECTOS DE INGENIERÍA TI REQUERIDOS POR LA ACADÉ-	
MIA INGLÉS PARA TODOS-IPT	19
3.14. Lumiback, sistema de códigos visuales para ciclistas	29
3.15. Implementación de un sistema de navegación para ciclistas, que le permita llegar a	
una dirección	33
4. Metodología y plan de trabajo	38
4.1. Formación introductoria	38
4.1.1. Capacitación inicial	38
4.1.2. Introducción a Git y GitHub	39
4.1.3. Introducción a la Prograamción Orientada a Objetos POO	40
4.1.4. Introducción los servicios de AWS	43
4.2. Mejoras tecnológicas	44
4.3. Actualización de paquetes y librerías	45
4.4. Actualización visual de la aplicación móvil	47
4.5. NAVI	50
4.6. Manual de Usuario	55

5. Modelado funcional	60
6. Resultados	62
6.1. Requerimientos de Diseño	62
6.2. Mejoras Tecnológicas	63
6.3. Aplicación Móvil	64
6.4. Actualización Visual Aplicación Móvil	68
6.5. Navi	81
6.6. Manual de Usuario	84
7. Conclusiones	91

Índice de figuras

1.	Lenguajes de programación orientados a objetos.	21
2.	Lenguajes de programación.	22
3.	Aplicación realizada durante la presentación del curso.	23
4.	Vista login.	24
5.	Vista registro.	25
6.	Vista olvidar contraseña.	25
7.	Convocatorias vigentes y vencidas.	26
8.	Vista Login.	26
9.	Lista de niveles.	27
10.	Roles dentro de la plataforma IPT.	28
11.	Vista index.	28
12.	Vista index.	29
13.	Vista index.	29
14.	Representación de un ciclista urbano con casco y sin chaleco reflectivo	30
15.	Ciclista incumpliendo las normas vigentes en el código de transito	31
16.	Ciclistas en un puente de Bogotá	31
17.	Ciclista en una acera de Bogotá	32
18.	Bocetos del ciclista con los diseños de Lumibac	32
19.	Prototipo Lumibac	33
20.	Diseño del sistema de navegación	34
21.	Arquitectura de la aplicación	34
22.	Ilustración del envío de datos de la aplicación al sistema electrónico del chaleco	35
23.	Montaje y soldadura de los elementos en la placa	35
24.	Montaje de elementos electrónicos en el chaleco	36
25.	Interfaz de la ventana ruta	37
26.	Clases abstractas y herencia simple	41
27.	Mixin	41
28.	Herencia múltiple mediante uso de mixin.	42
29.	Instancia de la clase Duck.	42
30.	Amazon Cognito.	43
31.	Amazon S3.	44
32.	Vista de Dispositivos	45
33.	Vista de Mapa	46
34.	Slider inicial	47
35.	Slider inicial	48
36.	Convocatoria: RETO CANDADO BICI	50
37.	Dimensiones de Navi	51
38.	Partes Navi	52
39.	Vista lateral de Navi	52
40.	Vista posterior de Navi	53
41.	Manual de Usuario	55
42.	Controles de Navi	56
43.	Controles de Navi	57
44.	Indicaciones Iniciales	58
45.	Advertencia de Uso	59
46.	Diagrama IDEF0 Aplicación	60
47.	Diagrama IDEF0 Navi	61
48.	Requerimientos de Diseño de la Aplicación Móvil	62
49.	Vista Login	64
50.	Vista Estado Dispositivos	65
51.	Vista Rutas	66
52.	Vista Control Navi Manual	67
53.	Vista Login Actualizada	68
54.	Slider inicial	69
55.	Slider inicial	70
56.	Slider inicial	71
57.	Vista Mis Dispositivos	72

58.	Vista Perfil	73
59.	Vista Calibración Sensor	74
60.	Vista Calibración Sensor	75
61.	Vista Calibración Sensor	76
62.	Vista Notificaciones Navi	77
63.	Vista Registrar Dispositivo	78
64.	Vista Editar Perfil	79
65.	Vista Registrar Vehículo	80
66.	Funciones Navi	81
67.	Vista Lateral Navi	82
68.	Carcasa Navi	82
69.	Vista Posterior Navi	83
70.	Vista Cúpula Navi	83
71.	Portada Manual de Usuario	84
72.	Contenido Caja	85
73.	Diseño Dispositivo	86
74.	Primero Pasos	87
75.	Controles de Navi	88
76.	Controles de Navi	89
77.	Mensaje de Seguridad	90

1. Introducción

1.1. RESUMEN

En el presente trabajo de grado se describe cómo se desarrollaron las prácticas como ingeniero de desarrollo de software, desarrollo móvil y diseño electrónico, las cuales fueron realizadas de manera virtual dentro de la empresa CiNNOV ubicada en la ciudad de Bogotá. Donde uno de los objetivos fue identificar una serie de requerimientos que permitieran realizar una actualización en el *firmware* utilizado en el dispositivo Bigo y de esta manera conseguir que los usuarios tuviesen una mejor experiencia. Con esta actualización se logró establecer una comunicación por Bluetooth con la aplicación móvil desarrollada previamente por la empresa. Paralelamente, le fueron asignados al practicante una serie de rutas de aprendizaje en las áreas de desarrollo móvil, de tal manera que tuviese conocimientos en áreas como el *frontend*, *backend*, consumo de API's, *Cloud Computing* y desarrollo y consumo de API REST.

Con los conocimientos necesarios, gracias a la ruta de aprendizaje realizada por el practicante, se logró la actualización de los paquetes y librerías de la aplicación móvil, consiguiendo así el control del dispositivo luminoso Bigo por medio de la API de *Google Maps* y el protocolo de comunicación de bajo consumo; BLE. Posteriormente, se realizó un rediseño de las principales vistas de la aplicación mediante la implementación de un gestor de estados más robusto pensado para la escalabilidad de las aplicaciones móviles. Como último objetivo, el practicante estuvo a cargo del desarrollo de un nuevo dispositivo electrónico llamado Navi, cuya finalidad fue complementar el señalizador luminoso Bigo y participar en la convocatoria RETO CANDADO BICI de la ciudad de Bogotá. Para ello, fue necesario realizar todo el diseño del circuito electrónico en una PCB de un diámetro máximo de 45 [mm]. El *firmware* del microprocesador se desarrolló en el lenguaje C++ y al igual que Bigo, Navi es controlado por medio de la aplicación móvil para generar las señalizaciones al actor vial.

PALABRAS CLAVE: Desarrollo de Software, Desarrollo Móvil, Firmware, Frontend, Backend, App, AWS.

1.2. ABSTRACT

In this degree work, it is described how the practices as a software development engineer, mobile development and electronic design were developed, which were carried out virtually within the CiNNOV company located in the city of Bogotá. Where one of the objectives was to identify a series of requirements that would allow an update to be made in the *firmware* used in the Bigo device and in this way ensure that users have a better experience. With this update, it was possible to establish communication via Bluetooth with the mobile application previously developed by the company. At the same time, the practitioner was assigned a series of learning routes in the areas of mobile development, in such a way that he had knowledge in areas such as *frontend*, *backend*, API consumption, *Cloud Computing* and REST API development and consumption. With the necessary knowledge, thanks to the learning path carried out by the practitioner, it was possible to update the packages and libraries of the mobile application, thus achieving control of the Bigo luminous device through the *Google Maps* API and low-power communication protocol; BLE. Subsequently, a redesign of the main views of the application was carried out by implementing a more robust state manager designed for the scalability of mobile applications. As a last objective, the intern was in charge of developing a new electronic device called Navi, whose purpose was to complement the

Bigo light signaling device and participate in the call CHALLENGE BIKE LOCK of the city of Bogotá. For this, it was necessary to carry out the entire design of the electronic circuit on a PCB with a maximum diameter of 45 [mm]. The *firmware* of the microprocessor was developed in the C++ language and, like Bigo, Navi is controlled by means of the mobile application to generate the signals to the road actor.

Keywords: Software Development, Mobile Development, Firmware, Frontend, Backend, App, AWS.

1.3. JUSTIFICACIÓN

Para 2018 Bogotá contaba con 835 mil usuarios de bicicleta, pero esta cifra ha aumentado hasta un 40 % [1], debido a la pandemia del Covid-19 dado que esta modalidad de transporte evita el contacto que se presencia dentro de los servicios de transporte público, además de ser un medio económico durante la crisis económica a raíz de la pandemia. Pero las estadísticas indican que al menos el 70 % de los accidentes de ciclistas son ocasionados por la falta de visibilidad y tan solo el 30 % de los biciusuarios cuentan con luces reflectoras [2]. Bigo es un *hardware* que nace a raíz de un accidente debido a la falta de visibilidad, este dispositivo es un indicador direccional cuyas luces led tienen la potencia para ser visibles en cualquier tipo de clima. El control se realiza mediante un giroscopio que va acoplado en el casco del biciusuario que permite indicar la dirección a tomar con la inclinación de la cabeza. El dispositivo Bigo se complementará con una aplicación móvil que le permitirá a los biciusuarios conocer todas las ciclovías disponibles en la ciudad y todos los reportes que hagan los usuarios sobre daños en la vía, accidentes, robo o violencia hacia la mujer serán proporcionados a la Secretaría de Movilidad de la ciudad. Debido a las grandes cantidades de datos, se implementarán técnicas de análisis de Big Data para poder interpretar toda la información proporcionada por los biciusuarios para generar mejores respuestas al problema de seguridad vial para ciclistas. Y gracias a la integración de tecnología IoT se crean mayores avances en la transición hacia las ciudades inteligentes, donde Bogotá ya se encuentra dentro de esta categoría, gracias a toda la información aportada por los usuarios de bicicleta y los sistemas de comunicación con los que estos interactúan.

1.4. CONTEXTO Y CONDICIONES DEL ENTORNO DE DESARROLLO

Para poner en contexto al lector, vamos a dividirlo en dos partes de forma que sea fácil entender la estructura del presente trabajo de grado. En la primera fase se habla de los requerimientos que fueron dados por parte de la empresa como practicantes de ingeniería y en la segunda fase se tratan las tareas que fueron realizadas durante el desarrollo de las prácticas.

1.4.1. Ingreso a la empresa y conocimientos básicos.

Una vez abierta la convocatoria, comprobando que como ingeniero mecatrónico cumplía con el perfil de ingeniero de prácticas que estaban solicitando para dar solución a una serie de requerimientos que tenía la empresa, realizo la aplicación por medio de la plataforma de *Linkedin*. Posterior a esto inicia el proceso de entrevistas a modo de filtro para ser parte del equipo de desarrollo de la empresa. En esta entrevista se hace una evaluación tanto psicológica, realizada por parte de la

psicóloga de la empresa, así como una componente técnica realizada por el CEO de la empresa, esto con la intención de comprobar que contaba con los conocimientos técnicos necesarios y requeridos para la realización de las diferentes tareas. Finalizado el proceso de entrevistas recibí la confirmación de aceptación en el equipo de prácticas y una segunda reunión con el CEO de la empresa estuvo dirigida a dar aclaración de todas las tareas iniciales que tuve como ingeniero de prácticas, así como la posibilidad de realizar tareas que surgieron durante el proceso de prácticas dentro de la empresa. Como aclaración, todo el desarrollo de las prácticas se dio de manera virtual.

1.4.2. Capacitaciones y adaptación de entorno para desarrollar las tareas.

Dado el enfoque de la empresa, era necesario tener una capacitación inicial para que el practicante pueda comprender la dimensión de la problemática de movilidad, principalmente para ciclistas, en la ciudad de Bogotá y principales carreteras del país. Dado que no solo se dedica a producir tecnología, sino a cumplir un papel social en donde se busca concientizar a los diferentes actores viales de tener mejores prácticas de movilidad y disminuir los accidentes en vías y carreteras del país. Por otro lado, era necesario tener unas bases en áreas de programación orientada a objetos, nunca vista durante el tiempo de estudio, así como en el manejo de plataformas como *GIT*, que permiten a los desarrolladores mantener un control de versiones, así como la posibilidad de realizar un proyecto de forma colaborativa entre más de un desarrollador. Otro componente importante a manejar era la programación de los microprocesadores integrados en la familia ESP dado que toda la tecnología desarrollada por la empresa, en el ámbito de *hardware*, se encuentra desarrollada con este chip. Para el desarrollo de la aplicación móvil era necesario tener conocimientos en el lenguaje *Dart* y su *framework* de desarrollo; *Flutter*.

1.5. PRODUCTO DE LA PASANTÍA

Todo el desarrollo de las prácticas me permitió aplicar diversas áreas aprendidas durante todo el proceso de estudio como el diseño electrónico de PCB, diseño de *firmware* de control para la programación de microprocesadores, diseño de algoritmos en diversos lenguajes de programación y haciendo uso de paradigmas de programación como la programación funcional y la programación orientada a objetos. Por otro lado, la integración con un ambiente laboral interdisciplinario, por las diversas áreas que integran una empresa, permitió vivir la experiencia de un ambiente laboral real, cumpliendo con responsabilidades y tiempos de entrega.

1.5.1. Entorno laboral

La empresa cuenta con diversas áreas que la componen: desarrollo e investigación, diseño, publicidad, marketing, contabilidad. Esto permitió, durante el proceso de prácticas, tener un primer acercamiento a lo que es un ambiente laboral real interdisciplinario. Aprender a trabajar de forma colaborativa es de suma importancia al momento de estar dentro de una empresa, esto permite tener un buen ambiente laboral, mejorando el rendimiento y productividad de la empresa, así como un mejor desempeño como trabajador.

1.5.2. Desarrollo de nuevo dispositivo

Uno de los mayores retos que me fue asignado como tarea durante el proceso de prácticas dentro de la empresa fue el desarrollo de un nuevo dispositivo de seguridad vial para diferentes actores que ahora hace parte de la familia CiNNoV. Para ello fue necesario realizar el diseño electrónico de la placa PCB que se acoplara a un diseño específico realizado por el departamento de diseño, para su funcionamiento se desarrolló el *firmware* del microprocesador montado en el chip ESP 32 en el lenguaje de C++ dada su facilidad a la hora de programar la familia ESP, cumpliendo con las expectativas de funcionamiento esperadas por la empresa.

1.6. DEFINICIÓN DEL PROBLEMA

Debido a la necesidad de crecimiento tecnológico del país y con intención de mejorar el producto ya existente, así como la creación de nuevos dispositivos de seguridad vial donde se implementasen las últimas tecnologías y que al mismo tiempo fuesen un complemento del dispositivo Bigo es que existe la oportunidad de integrar un equipo completo de ingeniero en condición de practicantes, quienes cumplieren con el perfil tanto en conocimientos técnicos, como en personas capaces de resolver problemas de manera ágil, innovadores y con dotes de liderazgo que permitiesen a la empresa ser más competitiva en el mercado nacional y mejorar sus procesos ya existentes.

1.7. CONTEXTO Y CONDICIONES DEL PROBLEMA

1.7.1. Actualización de los diversos paquetes, plugins y librerías de la aplicación móvil

Debido a la constante actualización del lenguaje *Dark* y del *framework Flutter*, es importante mantener las aplicaciones siempre en sus últimas versiones, debido a que todos los paquetes y librerías también se actualizan de forma constante para la corrección de errores, mejor implementación en las diversas plataformas donde es compilada la aplicación y también la seguridad del código. Por eso, esta fue una tarea inicial realizada durante el proceso de prácticas, todo se realizó previo a su publicación en la tienda de aplicaciones del sistema *Android*.

1.7.2. Actualización en el firmware de Bigo para permitir recibir indicaciones de GPS enviadas a través de la aplicación móvil

Inicialmente, Bigo fue desarrollado para recibir las indicaciones de giro por medio de un sensor de movimiento que se instala sobre el casco del usuario, pero con intención de complementar y ampliar su funcionalidad, este debía conectarse por medio de conexión inalámbrica por radiofrecuencia, para el caso se usó *Bluetooth*, a la aplicación móvil y poder recibir las indicaciones enviadas por medio de la API de Google Maps implementada en la aplicación. Logrando así, una mejora en el funcionamiento del Bigo, pues ya no depende únicamente del uso de un sensor de movimiento, ni de la indicación del usuario, mejorando su autonomía.

1.7.3. Desarrollo de un nuevo dispositivo de seguridad vial

Dada la necesidad de acercar la tecnología a la mayor cantidad de usuarios posibles a un precio asequible para todo tipo de usuario final, nace la necesidad y la idea de desarrollar un nuevo dispositivo que hace parte de la familia CiNNoV junto con Bigo, quienes también se complementan para hacer a los usuarios más visibles en las vías. Para ellos, se desarrolla NAVI, dispositivo presentado en la convocatoria RETO CANDADO BICI, promovido por diferentes actores del distrito de la ciudad de Bogotá como la Secretaría de Desarrollo Económico, Fitic, BogoTech y la Alcaldía Mayor de Bogotá. El ganador de dicha convocatoria recibió recursos para impulsar el desarrollo tecnológico.

1.8. OBJETIVOS

1.8.1. Objetivo general

- Desarrollar tareas consecuentes a construir un dispositivo con componentes de electrónica y programación.

1.8.2. Obejtivo específico

- Crear una lista de requerimientos de diseño, partiendo de los requisitos definidos para el proyecto.
- Plantear mejoras tecnológicas en el dispositivo, partiendo de los requerimientos de diseño.
- Aplicar las mejoras que surgieron a partir de los requerimientos de diseño.
- Testear las diferentes funcionalidades proyectadas en los dispositivos.
- Documentar esquemas, funcionamiento, manual de uso, etc.

2. Estructura del documento

2.1. CAPITULO I, Introducción

Este primer capítulo sirve para poder iniciar al lector en la problemática a resolver a lo largo del presente trabajo de grado, así como la contextualización de todo el trabajo, a nivel de ingeniería, que debió desarrollarse para poder culminar el cometido de las prácticas de una manera satisfactoria y también a nivel social, de donde surge la problemática y las soluciones dadas en este trabajo de grado.

2.2. CAPITULO II, Marco Teórico y Estado del Arte

En este capítulo se le da al lector los conceptos básicos necesarios para entender completamente el desarrollo del trabajo, así como una breve y resumida vista general de lo que es el estado de la técnica actual en relación con la temática desarrolladas durante el presente trabajo de grado.

2.3. CAPITULO III, Metodología y plan de trabajo

En este capítulo se da a conocer al lector cómo se llevaron a cabo el cumplimiento de los objetivos y desarrollos de las tareas durante las prácticas realizadas en la empresa, esto permite entender el orden en que fueron desarrolladas las actividades y el orden secuencial llevado a cabo.

3. Estado del Arte y Marco Teórico

En este apartado se hace una descripción de los conceptos principales utilizados durante el desarrollo del libro para que el lector tenga una mejor comprensión. Además, una serie de proyectos relacionados que funcionaron como apoyo a la investigación del tema propuesto en el presente trabajo de grado.

3.1. IOT

El IoT o Internet de las Cosas, por sus siglas en inglés, es un concepto que hace referencia a la interconexión entre cualquier dispositivo e internet. Esta conexión permite que exista una red gigante que hace posible la interacción entre objetos o dispositivos y las personas, donde los datos de áreas como la educación, transporte, comercio, etc son compartidos. Estos datos se recopilan de los sensores de cada dispositivo y son almacenados en una base de datos, normalmente alojada en la nube, donde se procesan solo los datos de interés para obtener una respuesta a necesidades específicas. De esta manera es posible generar modelos de *Machine Learning*, aplicados a los datos organizados, para ayudar en la toma inteligente de decisiones ayudando a que los procesos sean más eficientes gracias al análisis de los datos en tiempo real. [3]

3.2. BIG DATA

El término de *Big Data* se puede simplificar en toda aquella cantidad de información que no puede ser procesada por los métodos tradicionales de análisis estadístico, pues el costo de mantener grandes cantidades de información en bases de datos es muy elevado, aunque no exista una cantidad específica de datos para ser considerados como *Big Data*, se sabe que diariamente se producen 2.5 quintillones de *bytes*. La integración de modelos de Inteligencia Artificial, como el *Machine Learning*, permite crear modelos predictivos y una toma de decisiones inteligentes para mejorar los procesos, así como las investigaciones, pues las técnicas de análisis de *Big Data* permite que los investigadores puedan obtener resultados más precisos debido a que existe un mayor número de información y una mayor velocidad de análisis de los mismos, gracias a las diversas plataformas que han surgido durante los últimos años. La variedad de estos datos es bastante amplia también y estos pueden provenir de cualquier tipo de sensores, sistemas de GPS, equipos industriales, redes sociales, dispositivos móviles, vehículos, transporte, población, etc. y deben ser analizados en tiempo real, pues el avance de los sistemas IoT han aumentado la velocidad con que las empresas reciben toda esta información y el *Cloud Computing* ha ampliado posibilidades del *Big Data*. [4]

3.3. DISEÑO ASISTIDO POR COMPUTADORA (CAD)

El Diseño Asistido por Computadora o CAD, por sus siglas en inglés, es una técnica que hace referencia al uso de computadoras como ayuda en la creación y análisis de diseños y su aplicación es variada, pues se utiliza en diferentes campos que van desde la ingeniería, arquitectura, medicina, etc., gracias a la posibilidad de realizar diseños tanto en 2D como en 3D. Los *software* de diseño CAD permiten la creación de gráficos, sistemas virtuales, simulaciones tridimensionales, de modelos dinámicos y técnicas de análisis que permiten dar una visión completa del comportamiento de una

pieza o producto antes de que este sea llevado a producción, de esta manera se obtienen mejores resultados y reduciendo los tiempos de producción. Esta técnica de diseño se ha venido integrando con los procesos de automatización industrial para tener una mejor conexión entre las etapas de diseño y producción, apoyadas por la metodología del Diseño Concurrente. [5]

3.4. DISEÑO ELECTRÓNICO

El Diseño Electrónico es un campo de la electrónica que permite la creación de circuitos, cuyo propósito es la solución de problemas específicos mediante el uso de tres tipos de tecnologías; *hardware*, *software* y *firmware*. Desde mediados del siglo XIX el diseño electrónico ha tenido una evolución constante que ha permitido responder a las demandas y necesidades técnicas, así como ir a la par con el desarrollo de los semiconductores. El diseño ha podido avanzar gracias a la aparición de nuevas tecnologías a lo largo del tiempo, empezando por la aparición del transistor que permitió la sustitución de los tubos de vacío y mejorando las técnicas de diseño ya existentes. Cuando aparecieron los circuitos integrados tuvieron una gran relevancia en el diseño de circuitos digitales y un cambio en la concepción de lo que se entendía por Diseño Electrónico y en el momento en que los microprocesadores aparecen, el diseñador obtiene una mayor libertad a la hora de realizar un diseño, pues su función de transferencia es independiente de su estructura y esto permite la personalización del diseño, sumado al uso de tecnologías de *software* y *firmware* se puede tener una mayor versatilidad en los diseños. [6]

3.5. FRAMEWORK

Se puede entender como *Framework*, o marco de trabajo, a todo conjunto de estructuras reales o conceptuales que se encuentran construidas sobre un lenguaje de programación y permiten al desarrollador abordar problemas específicos sin tener qué preocuparse por la estructuración ni la codificación de código, permitiendo así un mayor control sobre el desarrollo de un programa, de esta manera el desarrollador solo debe centrarse en las funcionalidades de alto nivel de una aplicación. A diferencia de las bibliotecas, donde es el lenguaje de programación el encargado de hacer el llamado de la biblioteca, en un *Framework* es este quien se encarga de hacer el llamado al lenguaje. En campos como la Inteligencia Artificial encontramos marcos de trabajo como *PyTorch*, que está destinado a la Visión Artificial y al procesamiento del lenguaje natural (PNL) o *TensorFlow*. En áreas como el desarrollo móvil se encuentra Flutter, que permite el desarrollo multiplataforma mediante la codificación de una sola lógica de programación para todas las plataformas que este *Framework* soporta. [7]

3.6. ARQUITECTURA MVC

Al momento de realizar un proyecto que implique el uso de *frameworks* nos podemos encontrar con el problema de la falta de estándares de monitoreo y administración de los proyectos, pero existen diversos modelos de patrones de diseño que permiten la estructuración de los proyectos de tal manera que se permita el desarrollo de una manera más rápida y eficiente. La arquitectura MVC o Modelo-Vista-Controlador permite separar la estructura de una aplicación en tres componentes; el Modelo hace referencia a la lógica y a la semántica del proyecto, la Vista se relaciona con toda la interacción con el usuario haciendo uso de una sintaxis concreta para definir el comportamiento que

el usuario percibe de la aplicación y el Control se encarga de gestionar todos los cambios presentes en un View y las acciones del usuario para seguir manteniendo la coherencia entre la Vista y el Modelo. Este tipo de arquitectura se implementa en diversos tipos de plataformas de desarrollo como el desarrollo web y el desarrollo móvil gracias a su adaptabilidad y su eficacia. El separar la estructura de un proyecto en tres componentes diferentes permite realizar diversas modificaciones a un solo componente sin comprometer el funcionamiento de la aplicación. [8]

3.7. PROGRAMACIÓN ORIENTADA A OBJETOS

La Programación Orientada a Objetos es un paradigma de programación donde los objetos permiten modelar el mundo representado como estructuras de datos que permiten definir propiedades y comportamientos del modelo. Los objetos cuentan con un estado, un comportamiento y una identidad: Estado: “conjunto de propiedades del objeto y valores actuales de esas propiedades.” Comportamiento: modo en que el objeto actúa y reacciona ante los mensajes que se le envían (con posibles cambios en su estado). Viene determinado por la clase a la que pertenece el objeto. Identidad: propiedad que distingue a unos objetos de otros (nombre único de variable) . Todos los atributos, operaciones, relaciones, métodos y servicios de un objeto se encuentran abstraídos en lo que se conoce como Clase que permite crear instancias de estas, heredando así todas sus propiedades y los objetos instanciados por la misma clase comparten el mismo comportamiento. [9]

3.8. ESTADÍSTICAS NACIONALES SOBRE VÍCTIMAS DE SINIESTROS

El Observatorio Nacional de Seguridad Vial (ONSV) presentó su último informe el mes de julio donde se presentan las estadísticas sobre las víctimas por siniestros viales ocurridas en lo que va del año 2021 hasta la fecha de publicación de dicho informe. Las estadísticas indican un aumento del 10.45 % en los fallecidos por siniestros viales en comparación con los últimos 5 años y los usuarios de motocicleta representan un 59.7 % de estos fallecidos lo que representa un aumento del 23.17 % en comparación con los últimos 5 años, mientras que los usuarios de bicicleta representan el 6.6 %. La mayor cantidad de fallecimientos se encuentran ubicados entre los grupos de edades para adultos jóvenes y adultos con un 37.2 % y un 33.4 % respectivamente, aunque el rango de edad con mayor fallecidos para los usuarios de bicicleta se encuentra los 65 y más, mientras que el rango de edad con mayor número de fallecidos para usuarios de moto se encuentra entre los 20-25 años. En este informe se puede observar que el momento de ocurrencia de los siniestros viales se presenta en horas entre las 18:00 y las 20:59 y otro gran porcentaje de ocurrencia se encuentra entre las 21:00 y las 23:59, lo que indica que la mayoría de los siniestros viales ocurren en horas de la noche debido a los problemas de visibilidad que tienen los usuarios en las vías. Tanto para los usuarios de moto como para los usuarios de bicicleta las clases de siniestro que presentan un mayor porcentaje como causa de los fallecimientos son el choque y el volcamiento con un 51 % y 9.4 % respectivamente en comparación con otras clases de siniestros. Entre los objetos de colisión con mayor porcentaje como responsables de los choques se encuentran los objetos fijos con un 26.6 %, los transportes de carga con un 22.5 % y los transportes individuales con un 22.4 %. Para el caso de solo lesionados las cifras varían un poco y para los usuarios de moto se tiene que representan un 62.4 % de todos los casos de lesionados en las vías, mientras que los ciclistas representan el 8.47 % de estos casos. Y los grupos de edades con mayor porcentaje de involucrados en estos casos siguen siendo los mismos, un 45.2 % para el caso de los adultos jóvenes y un 35.9 % para los adultos. A diferencia de los casos

por fallecimiento, los usuarios lesionados tienen una hora de ocurrencia mayor entre las 15:00 y las 17:59. [10]

3.9. CIUDADES INTELIGENTES

Las estimaciones que tiene la Organización de Naciones Unidas indican que al menos un 70 % de la población vivirá dentro de una ciudad inteligente para el año 2050. Podemos definir a una ciudad inteligente como una ciudad donde la calidad de vida está relacionada de forma directa con el uso de la tecnología. “El propósito final de una Smart City (ciudad inteligente) es alcanzar una gestión eficiente en todas las áreas de la ciudad como el urbanismo, infraestructuras, transporte, servicios, educación, sanidad, seguridad pública, energía, satisfaciendo a las necesidades de la urbe y de sus ciudadanos”. Donde los rasgos más relevantes que podemos encontrar es una mejor gestión del espacio urbano, un mayor uso de energías renovables que permitan disminuir los niveles de CO₂ y una mayor integración con la tecnología y los sistemas inteligentes, de esta manera se pueden conseguir ciudades más sostenibles. [11]

3.10. INGENIERÍA DE SOFTWARE

La ingeniería de *software* juega un papel muy importante en el desarrollo de los países debido a la dependencia que tienen estos de los complejos sistemas informáticos que permiten el normal desarrollo de las actividades de la industria y otras áreas, así como de diversas infraestructuras. Debido a las características de la ingeniería de *software*, esta se tomó como referencia para la realización de este trabajo ya que nos permite el desarrollo de sistemas de software abstractos y complejos de una forma ágil gracias a sus diversas metodologías, técnicas y herramientas que nos permiten tener el control sobre el desarrollo de sistemas altamente complejos. De esta forma podemos reducir el esfuerzo que se requiere para la producción de estos sistemas, gracias a los métodos de especificación, diseño e implementación que se han venido desarrollando y perfeccionando a lo largo de los años, ya que cada vez se requiere del desarrollo de sistemas más complejos, minimizando los tiempos de producción y haciendo uso de técnicas eficaces de testeo e implementación. Para ello, se realiza un proceso de software el cual hace referencia a un conjunto de actividades: especificación del *software*, desarrollo del *software*, validación del *software*, evaluación del *software*. [12]

3.11. DESARROLLO DE SOFTWARE

Actualmente para el desarrollo de *software* se utiliza una estructura o un proceso específico que tienen en cuenta el ciclo de vida de un producto de *software*, según sean las necesidades de aplicación o los requisitos del cliente. Para facilitar la tarea del desarrollo existen diversas metodologías con una serie de pasos en específico que nos permiten reunir todos los requerimientos, facilitando el proceso de desarrollo. Primeramente, se deben tener todos los requerimientos del cliente, los cuáles se reúnen durante la fase de análisis donde el desarrollador debe ser capaz de hacer que el cliente pueda ser guiado en todos los requerimientos. Una vez finalizado el proceso de análisis se deben definir las características necesarias del sistema mediante la fase de especificación donde se pretende establecer cuál debe ser el comportamiento del sistema, de esta forma se puede pasar a la fase de implementación la cual hace referencia al desarrollo final del *software*. [13]

Actualmente el desarrollo de *software* se encuentra en su mayoría dominado por la programación orientada a objetos como paradigma de programación, gracias a que este tipo de abstracción del mundo real representa muchas ventajas en comparación con otros paradigmas como la programación funcional. Este paradigma utilizado actualmente se trabaja con diferentes tipos de arquitectura que permiten al desarrollador la creación de *software* completamente robusto y de calidad.

3.12. DESARROLLO MÓVIL

Desde la presentación de la tienda de aplicaciones por parte de Apple en julio de 2008, el mundo del desarrollo móvil ha venido creciendo de forma exponencial a través de los años gracias al gran interés que existe por parte del mundo económico, empresarial y científico en el campo de los dispositivos móviles. El desarrollo móvil tiene como base el desarrollo de *software*, por lo que puede hacer uso de las mismas metodologías y arquitecturas que se encuentran presentes en el mundo del desarrollo de *software*. Dada la naturaleza de los sistemas operativos existentes en el mercado de los dispositivos móviles, el desarrollo móvil se puede catalogar de diferentes maneras. Primero tenemos el desarrollo móvil nativo que permite desarrollar aplicaciones móviles dirigidas a un sistema operativo en específico partiendo del uso de un lenguaje de programación soportado por dicho sistema; en el caso de Android el desarrollo se hace a partir del uso de *Kotlin* como lenguaje nativo de Android, sin embargo para poder desarrollar aplicaciones dirigidas a IOS se debe hacer uso del lenguaje oficial *Swift*. Para facilitar el desarrollo ambos sistemas operativos proporcionan su propio entorno de desarrollo (IDE). Sin embargo, con los años se ha hecho para las compañías cada vez más complicado realizar el desarrollo de la misma aplicación más de una vez para poder ser implementada en diferentes plataformas, por eso nace el término de desarrollo móvil multiplataforma, el cual permite realizar el desarrollo de una aplicación móvil que puede ser ejecutada en más de un sistema operativo, de esta forma se reducen notoriamente los tiempos y costos que implican en el desarrollo de una aplicación que esté disponible en todas las plataformas. En este caso tenemos diferentes *frameworks* como pueden ser: *Flutter*, *React Native*, *Xamarin.Forms*, etc. Para nuestro caso de estudio se hizo uso del *frameworks* de flutter el cual está escrito en el lenguaje Dart y permite el desarrollo de aplicaciones móviles multiplataforma. [14]

3.13. APOYO A PROYECTOS DE INGENIERÍA TI REQUERIDOS POR LA ACADEMIA INGLÉS PARA TODOS-IPT

El presente capítulo recopila un conjunto de aportes relacionados a la temática de trabajo de grado. Por tanto, se expondrán 3 estudios que fundamentan la temática principal ligadas a las prácticas desarrolladas (Ingeniería TI). En [15], se describe el desarrollo de una práctica empresarial relacionada con el perfil de ingeniero de desarrollo de *software* y auxiliar de áreas de sistemas. Dicha práctica se ejecutó en la ciudad de Cúcuta, Norte de Santander, sede principal de la Academia Ingles Para Todos (IPT). El objetivo de los procesos descritos en este documento radica en la correcta funcionalidad de las múltiples plataformas digitales que dispone la empresa para facilitar la interacción con sus estudiantes. Por ejemplo, la página web oficial de la compañía, plataforma de estudio para los estudiantes, plataforma de realización de exámenes de las unidades según la temática establecida en el cronograma, aplicaciones, entre muchos otros. Inicialmente, se le asignaron pequeñas rutas de aprendizaje al aprendiz con el fin de llevar a cabo su introducción en áreas de desarrollo de *software*, incluyendo conocimientos de *Frontend* y *Backend*, bases de datos, manejos

de APIs (Interfaz de programación de aplicaciones, en inglés *Application Programming Interfaces*), entre otros. Para las actividades se requirió la implementación de un nuevo diseño del sitio web de la compañía, que incluyen mejoras en el servidor por parte del equipo de ingenieros de la academia. Para esto, fue necesario desarrollar una plataforma que contenga las diversas actividades académicas asignadas por el tutor y que además posibilite la conexión de manera eficiente a los estudiantes.

Posteriormente, se propone el desarrollo de una aplicación móvil como uno de los proyectos más importantes solicitados por la empresa. Esto debido a la necesidad de contar con una app con finalidades académicas, socializándose de manera dinámica como una modalidad divertida dirigida a toda la comunidad de estudiantes de la academia (sin restricción de edad). Por otra parte, se presentan algunas actividades realizadas por la academia y la Universidad Francisco de Paula Santander que involucran las pruebas de suficiencia de inglés EKT (*English Knowledge Test*).

Esta prueba representa un requisito para el grado universitario en el cual el estudiante debe aprobar el examen asignado por la institución educativa en la que se encuentra. En este sentido, el equipo de ingenieros de la academia cumple la labor de garantizar el correcto funcionamiento de la plataforma en donde se presentan las pruebas de suficiencia de inglés, incluyendo el buen uso de las bases de datos. Una vez definido el argumento base del trabajo de grado, se procede entonces a describir el desarrollo de los aspectos más importantes contenidos en las actividades realizadas. Para esto, se presentarán los objetivos específicos:

- Realizar de manera prolongada formación introductoria al desarrollo de software para apoyar los proyectos generales de la empresa.
- Desarrollar en conjunto plataforma y página web propia de la empresa.
- Contribuir en el desarrollo de una aplicación móvil académica en Flutter con didáctica de juegos.
- Mantener en buen funcionamiento la plataforma y base de datos para pruebas EKT de la Universidad Francisco de Paula Santander.

Para el cumplimiento del primer objetivo, se llevó a cabo una capacitación inicial para contextualizar la magnitud de los proyectos, plataformas digitales, servicios ofrecidos y entornos de desarrollo que dispone la empresa. La revisión de las herramientas tecnológicas implementadas por la empresa sirve como un enfoque necesario en el panorama del aprendiz. Esta capacitación se realiza con la presencia de todos los integrantes que hacen parte de la comunidad de la empresa, conociendo de esta manera las funciones que desempeña cada uno. Durante el desarrollo de esta sesión, se lleva a cabo una exposición de las metodologías implementadas para el trabajo en equipo, con la finalidad de reducir la falta de organización y aumentando la eficiencia a través de técnicas de codificación rápidas. En adición a esto, se plantea la clasificación de tareas y el uso de repositorios para la unificación y puesta en marcha de proyectos.

En cuanto a la ruta de aprendizaje introductoria y preparativa para el aprendiz, se dispone una primera sección de aprendizaje con un curso profesional de *Git* y *GitHub*. *Git* es conocido mundialmente por ser el control de versiones más moderno disponible en la actualidad, permitiendo el monitoreo en tiempo real de los cambios realizados en un proyecto, la Tabla.1.2. ilustra los comandos principales utilizados en *Git*.

Comandos Principales	
Comando	Descripción
Git init	Lo usamos para determinar la carpeta en la que vamos a trabajar.
Git status	lo usamos para saber si tenemos un archivo añadido o borrado en nuestro proyecto, para saber en la rama en la que estamos y si tenemos commits.
Git add	Es para añadir un archivo a nuestra rama seguidamente ponemos entre comillas el nombre de nuestro archivo o poner un punto para añadir todos los archivos de nuestra carpeta. .
Git rm	Lo usamos para borrar un archivo que hayamos añadido, para eliminarlo por completo de nuestra rama usamos git rm – cached.
Git commit	Se usa para añadir un commit a nuestra rama, tambien podemos ponerle un -m seguidamente ponemos entre comillas nuestro mensaje.
Git config	Muestra configuraciones de git tambien podemos usar –list para mostrar la configuración por defecto de nuestro git y si añadimos –show-origin inhales nos muestra las configuraciones guardadas y su ubicación.

GitHub por su parte, facilita la creación de proyectos abiertos para su implementación por medio de herramientas y aplicaciones.

Por otro lado, se requirió de un curso titulado Introducción a la terminal y línea de comandos como base para la formación de un desarrollador. Así como también los cursos Algoritmos y pensamiento lógico, y Buenas prácticas para escritura de código. La programación orientada a objetos resulta en un pilar para el desarrollo de software debido a la eficiencia y versatilidad en la programación. Entre sus características, se destaca el Encapsulamiento, la Abstracción, La Herencia y el Polimorfismo. En la figura 1 y figura 2 se ilustran algunos lenguajes de programación orientados a objetos.



Figura 1: Lenguajes de programación orientados a objetos.

Fuente: [15]



Figura 2: Lenguajes de programación.

Fuente: [15]

Para la segunda sección de aprendizaje del practicante, se contemplan cursos relacionados a la Introducción a PHP con Composer dado a que PHP es un lenguaje de programación adaptable al desarrollo web del lado del servidor. Por otro lado, de la Introducción a *Laravel* para generar la capacidad de manejar eficazmente el *framework* anteriormente mencionado. Además, el curso de Fundamentos de bases de datos para la comprensión del funcionamiento de un proyecto con un elevado flujo de datos. Finalmente, los cursos SQL prácticos, Manejo de datos en Laravel con Eloquent, curso de API Rest y curso de Postman para complementar la ruta de aprendizaje orientada a PHP con el desarrollo de reglas para que las aplicaciones interactúen entre sí.

Por último, en la tercera sección de aprendizaje, se presentan los cursos relacionados al desarrollo Frontend para generar las funcionalidades de interfaz de usuario. Primeramente, se describe el curso de HTML y CSS práctico para la estructuración de información y el estilo en páginas web, sitio web o plataformas. Seguidamente, se propone el curso Sistemas de diseño para desarrolladores con el objetivo de que el aprendiz adquiriera la capacidad de crear sistemas de diseño, dominar la instalación de un sistema de diseño, aplicar flujos de trabajo, entre otras aplicaciones. Posteriormente, el curso de Frontend Developer para reforzar los conocimientos adquiridos durante la presentación del curso HTML y CSS práctico. En él, se introduce una nueva tecnología denominada responsive design como metodología para adaptar un sitio web a cualquier dispositivo.

El curso de Flutter le brinda al aprendiz las herramientas necesarias para crear aplicaciones de manera rápida, con diseños llamativos y con la característica de ser multiplataforma; al ser un SDK (kit de desarrollo *software*) creado por Google, es posible crear diseños de interfaces nativas en iOS y Android. En la figura 3 se observa la aplicación realizada durante la presentación del curso. El curso *Flutter* avanzado contribuye al practicante una profundización de los conocimientos adquiridos en el curso básico de *Flutter*. Por otra parte, el curso de *Typescript* con Angular permitirá al aprendiz hacer una profundización en los conocimientos relacionados al uso de *JavaScript* y así mismo, adicionarlos al desarrollo de proyectos en *Angular*.

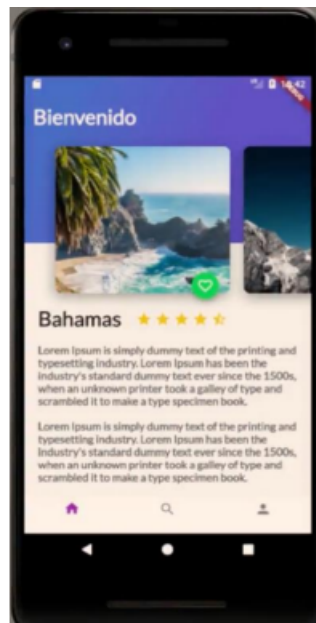


Figura 3: Aplicación realizada durante la presentación del curso.

Fuente: [15]

Finalmente, el Fundamentos de Angular y curso Profesional de Angular le permitirán al practicante culminar la ruta de aprendizaje relacionada a este *framework*. *Angular* es un *framework* creado por Google con múltiples herramientas: PWA, CLI, formularios, etc. El equipo de ingeniería de la empresa ha elegido Angular como *framework* principal debido a su robustez y aceptable curva de aprendizaje.

Una vez efectuada una breve revisión de los cursos introductorios, se procede a socializar el desarrollo de una plataforma web para la presentación de las pruebas de suficiencia de inglés solicitada por la universidad Francisco de Paula Santander. Lo anterior basado en mejorar la experiencia de los estudiantes al momento de realizar la prueba, ofreciendo un servicio optimizado a la universidad. El desarrollo de la plataforma parte de un listado de funcionalidades requeridas por la empresa de acuerdo a un convenio entre la universidad y la academia.

Para esto, se inicia con la elaboración de historias de usuarios por parte del equipo de desarrollo para posteriormente emprender la ejecución del proyecto mediante la metodología cpp's para división de tareas. La plataforma se desarrolla con una vista de login, olvido de contraseña y registro de usuarios nuevos. Asimismo, una vista del dashboard principal que hará parte de Home y presentará un carrusel de imágenes con todas las convocatorias disponibles a la fecha. En cuanto a los Grupos EKT, es una vista destinada al usuario docente y usuario supervisor en el que se divide el listado de los estudiantes por secciones horarias. El desarrollo de la vista para los puntajes EKT se llevó a cabo de la misma manera que los grupos EKT, su diferencia radica en la visualización de las preguntas correctas o incorrectas que llevan los estudiantes. La figura 4, figura 5 y figura 6 presentan la vista login, register y forgot-password. La figura 7 ilustra la previsualización de las convocatorias.

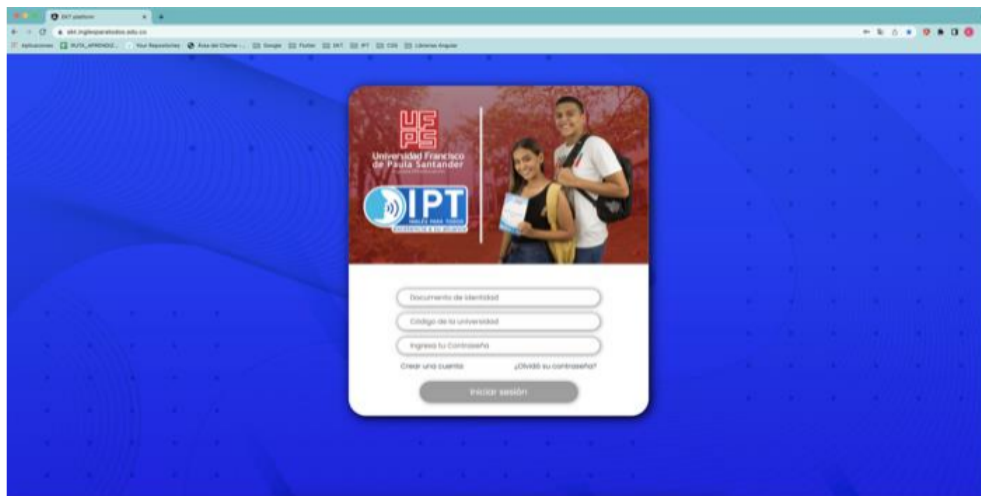


Figura 4: Vista login.

Fuente: [15]

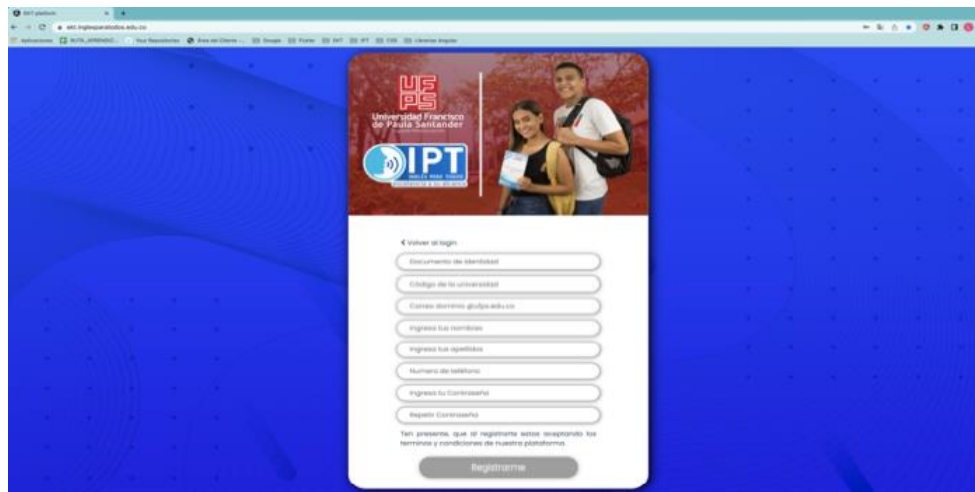


Figura 5: Vista registro.
Fuente: [15]



Figura 6: Vista olvidar contraseña.
Fuente: [15]

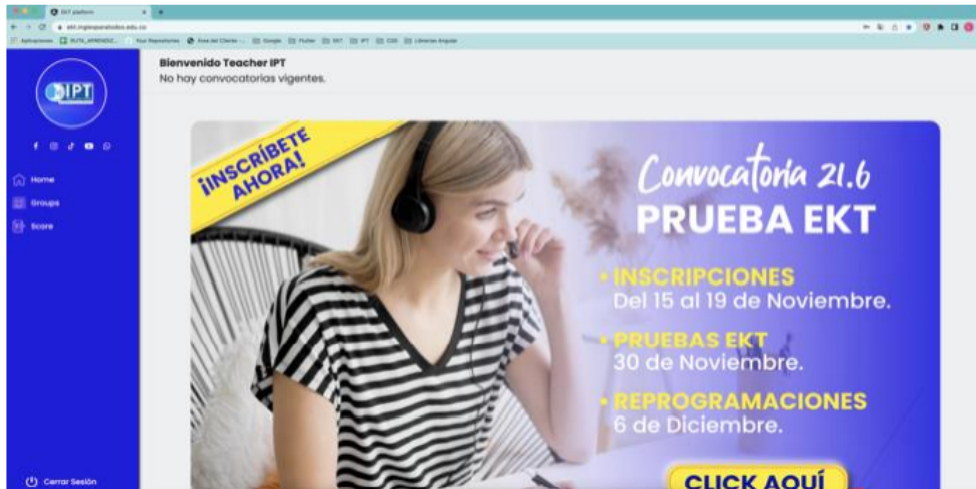


Figura 7: Convocatorias vigentes y vencidas.

Fuente: [15]

En lo que respecta al proyecto IPT World – App móvil, se optó por elegir el *framework Flutter* en el que adicionalmente se realizan tres particiones de corazón de la aplicación: Componentes, núcleos y páginas. Por último, se presenta una vista de mapa, vista de pre-nivel, vista de pre-mapa y vista de registro. La figura 8 se presenta la vista login de la aplicación, la figura 9 ilustra la vista principal con los niveles.

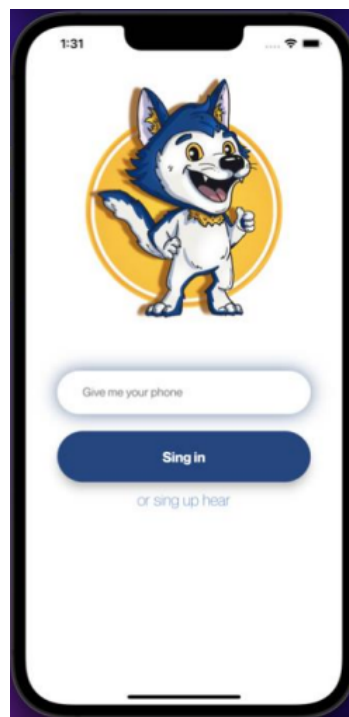


Figura 8: Vista Login.

Fuente: [15]

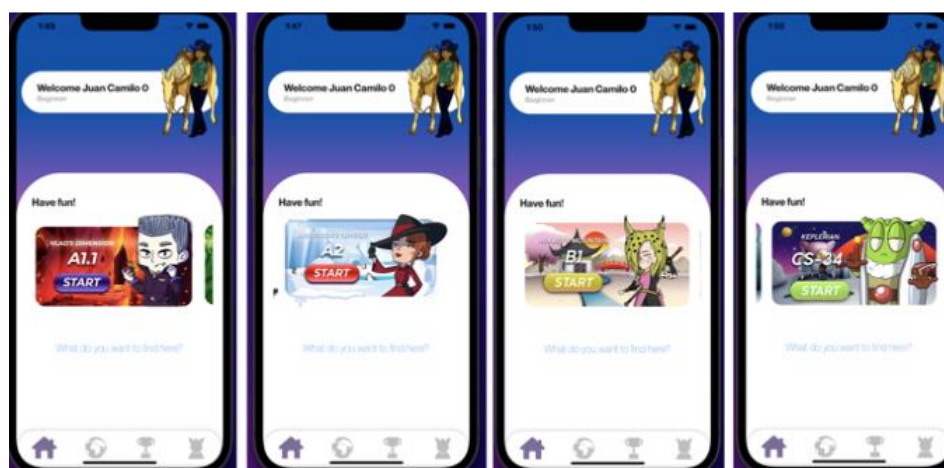


Figura 9: Lista de niveles.

Fuente: [15]

Para el proyecto IPT Platform, se requería el desarrollo de una plataforma web con la capacidad de dividir cada dependencia de la empresa y procesar a su vez los flujos de negocio en función del departamento seleccionado. Además de la creación de usuarios, asignaciones de roles y la realización de actividades de manera manual. Para ello, se emplea la metodología Scrum con hitos relacionados al Daily, CheckStatus, Sprint planning y Sprint. De la misma manera, se inicializa con las Historias de Usuarios clasificada por departamentos. Seguidamente, la presentación del Release 1.0 haciendo referencia a la primera versión de la plataforma con la presentación de los roles observados en la figura 10.

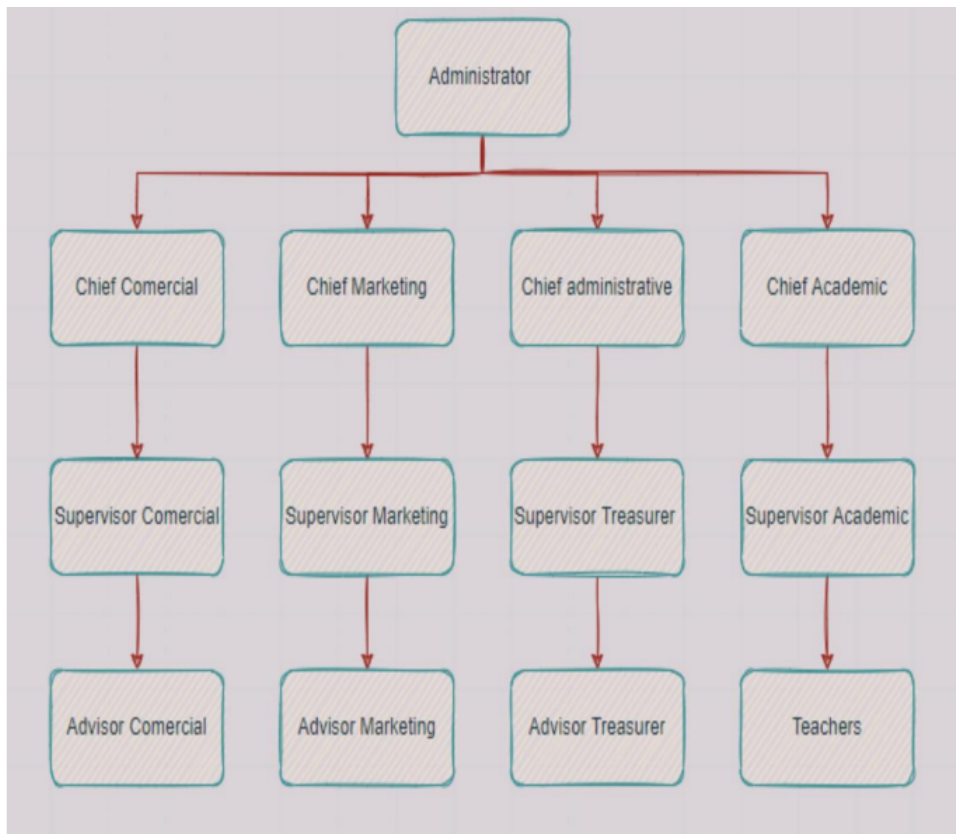


Figura 10: Roles dentro de la plataforma IPT.

Fuente: [15]

Para el desarrollo del proyecto, se emplea la metodología atomic design, manteniendo la misma estructura para los componentes que constituyen cada vista. La figura 11, figura 12 y figura 13 presentan la página principal de la plataforma.



Figura 11: Vista index.

Fuente: [15]

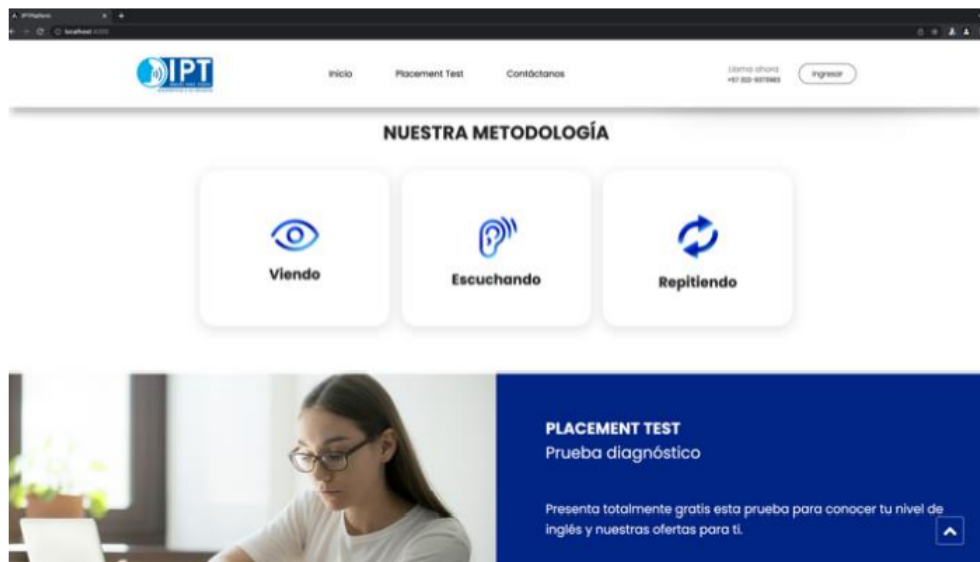


Figura 12: Vista index.

Fuente: [15]

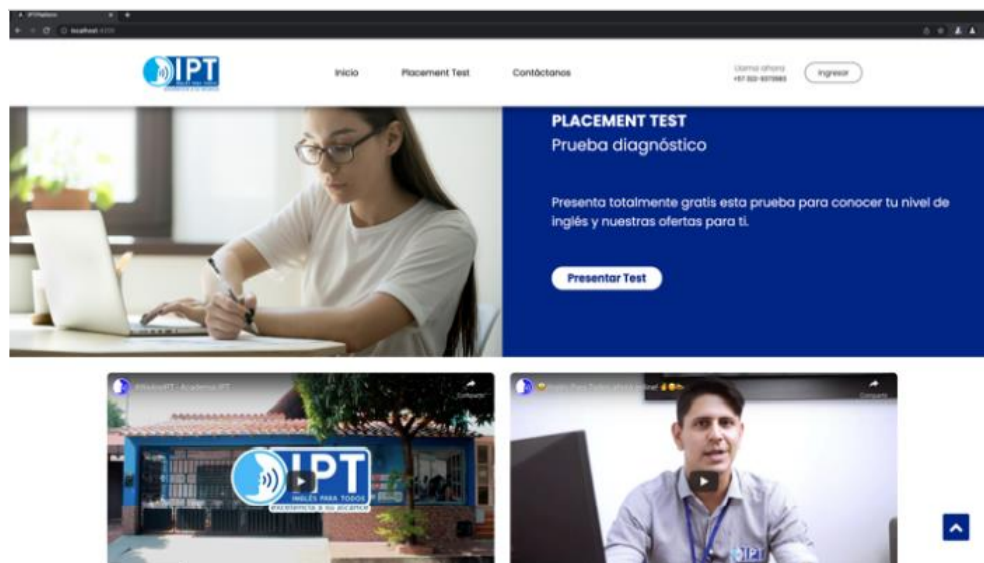


Figura 13: Vista index.

Fuente: [15]

3.14. Lumiback, sistema de códigos visuales para ciclistas

Otro ejemplo, en [16], se presenta el proyecto denominado Lumiback concebido a partir de las problemáticas relacionadas con los biciusuarios o ciclistas al sufrir inconvenientes durante sus trayectos hasta el trabajo, universidades, escuelas e incluso, en los trayectos destinados a prácticas deportivas, hobby o distracción. Esto debido a la falta de visibilidad de los usuarios de bicicletas en las rutas de tránsito o vías de transporte, ocasionando quejas por parte de los demás actores al no lograr detectar o visibilizar a simple vista los movimientos que harán los pedalistas en sus trayectos.

Por otro lado, es debido mencionar que los ciclistas cuentan con un kit de seguridad y visibilidad en orden que incluyen señales visuales. Sin embargo, este método resulta en un completo acertijo para los demás conductores viales. De hecho, para algunos ciclistas no expertos en el tema se les dificulta saber el propósito de cada señal. Por tanto, se pretende concretar un punto de partida para el desarrollo de este proyecto teniendo como punto de referencia la cultura tradicional de sistema visuales en automóviles y su integración con el sistema de señales implementados por los ciclistas expertos. Resulta conveniente especificar la gran variedad de productos relacionados con la identificación del ciclista en las vías. Por ejemplo, luces reflectoras o intermitentes cuya función se limita a indicar la ubicación del ciclista, omitiendo funcionalidades para que los conductores estén atentos a los movimientos del usuario. Es importante considerar las estadísticas y estudios de la Universidad Libre abiertas al público por medio de la Revista Semana, ya que revelan el incremento en el uso de la cicla en la ciudad de Bogotá. Durante el año 2018, el número de trayectos diarios ascendió a 835.000, en donde los viajes de menor trayecto (menos de 5 km) cubrían el 65 % de los viajes en bicicletas. De acuerdo a las conclusiones dadas por el director del Observatorio de Logística y Movilidad de la Universidad Nacional, el ciudadano residente en la ciudad de Bogotá prefiere movilizarse en cicla dadas las deficiencias en el sistema de transporte público de la ciudad. La figura 14 representa a un ciclista urbano utilizando casco, sin chaleco reflectivo.

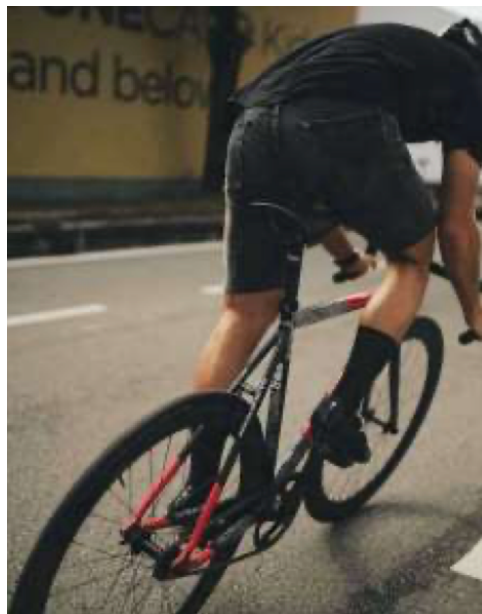


Figura 14: Representación de un ciclista urbano con casco y sin chaleco reflectivo

Fuente: [16]

Según las estadísticas de la Universidad Libre, 6 de cada 10 ciclistas emplean los elementos de seguridad en trayectos. Dichos elementos, incluyen un chaleco reflectivo con un casco de seguridad. Lo cual refiere a que el 60 % usan los elementos de seguridad de manera adecuada según las normativas vigentes. Lo anterior en conjunto con la falta de comprensión de los ciudadanos al seguir la normatividad de tránsito resultas en un 70 % de accidentes ocurridos en las vías por falta de visibilidad al no usar los implementos necesarios o cumplir con los estándares de la ley para la identificación de estos actores de tránsito. La figura 15 presenta a un ciclista incumpliendo los estándares de la ley descritos en el código de tránsito.



Figura 15: Ciclista incumpliendo las normas vigentes en el código de tránsito

Fuente: [16]

Asimismo, el 24 % de estos accidentes corresponden a accidentes con automóviles mientras que el resto de este porcentaje se concentra en peatones, motocicletas, buces, tractomulas, entre otros. La figura 16 expone la toma de algunos ciclistas en un puente de Bogotá debido a que la ciclo ruta no cubre cierto porcentaje del recorrido.



Figura 16: Ciclistas en un puente de Bogotá

Fuente: [16]

Por otro lado, la cifra de accidentes viales relacionados con usuarios desplazándose con medios de transporte de dos ruedas viene en ascenso desde el año 2017, incrementando en un 10 % a 15 % aproximadamente cada año. Esta cifra resulta preocupante en comparación con otros países europeos. Por ejemplo, haciendo una breve comparación entre Bogotá y España, en Bogotá se

presentan más accidentes viales y ciclistas fallecidos que en toda España. La figura 17 ilustra a un ciclista en una acera de Bogotá, evento ocasionado por un accidente vial.



Figura 17: Ciclista en una acera de Bogotá
Fuente: [16]

Con base en los problemas mencionados anteriormente, el trabajo de grado titulado “LUMIBACK: SISTEMA DE CÓDIGOS VISUALES PARA CICLISTAS” refiere a un sistema objetual diseñado para la parte lumbar del ser humano (cinturera) con adaptabilidad para diferentes tipos de percentiles. Este dispone de un control de luces que se encenderán mediante comandos que el usuario ingresará por medio de botones para la identificación del movimiento predecible del pedalis- ta urbano. Proporcionando una serie de códigos visuales que podrán reconocer los demás usuarios que estén en la vista frontal y lateral del ciclista. La figura 18 presenta los bocetos del ciclista con los diseños de Lumibac.



Figura 18: Bocetos del ciclista con los diseños de Lumibac
Fuente: [16]

En cuanto a los aspectos técnicos, se encuentran los siguientes insumos:

- Switch controlador de encendido o apagado.
- Cables 2x22.
- Termoencogible 15 mm.
- Módulos 125 de luz amarilla cálida (x10).
- DMódulos Cob de luz roja (x4).

El prototipo final es presentado en la figura 19.



Figura 19: Prototipo Lumibac
Fuente: [16]

3.15. Implementación de un sistema de navegación para ciclistas, que le permita llegar a una dirección

Finalmente, en [17], se expone el proceso de diseño e implementación de un sistema de navegación para ciclistas compuesto por una aplicación móvil y un chaleco. Inicialmente, se indican las razones que incentivaron la ejecución del proyecto, los fundamentos teóricos, dispositivos y herramientas tecnológicas para su fabricación. Seguidamente, se menciona el tipo de investigación y la metodología implementada para el cumplimiento de los objetivos propuestos. Posteriormente, se detallan los resultados los cuales contienen una explicación de la arquitectura general del sistema de navegación, el desarrollo de la aplicación móvil, exposición del proceso de construcción del sistema electrónico del chaleco y los resultados de las pruebas de funcionamiento. Por último, se presentan las conclusiones y recomendaciones para futuros trabajos con características similares. La figura 20 y figura 21 ilustran el diseño del sistema de navegación y la arquitectura de la aplicación.

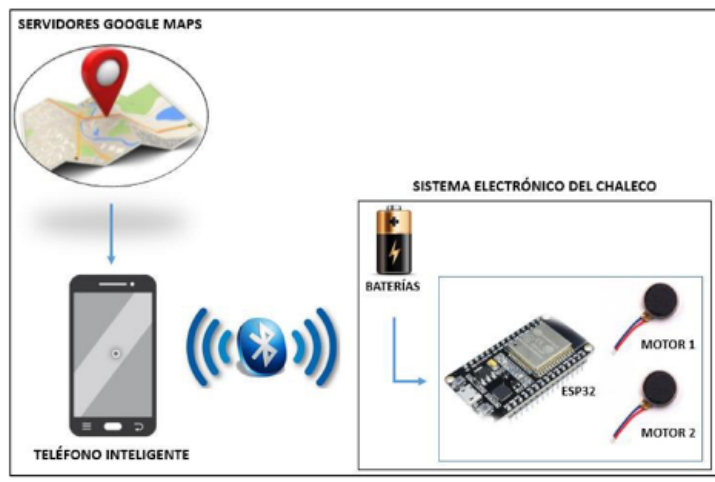


Figura 20: Diseño del sistema de navegación

Fuente: [17]

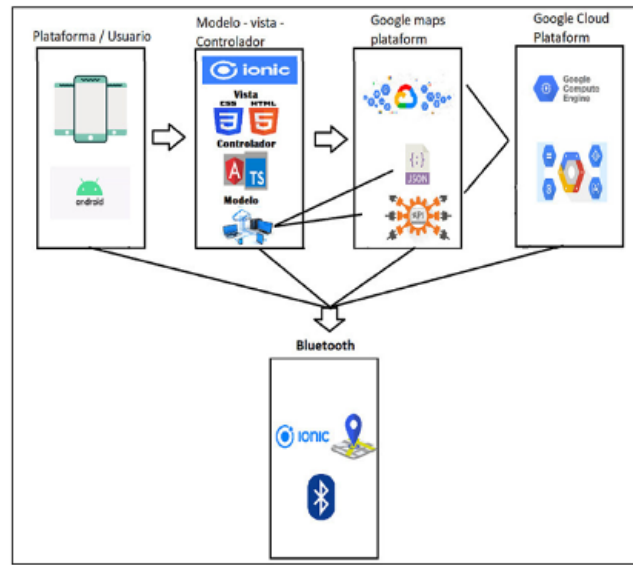


Figura 21: Arquitectura de la aplicación

Fuente: [17]

La necesidad en el desarrollo de dicho proyecto radica en el inconveniente que presentan algunos usuarios de bicicleta en conocer que rutas o calles tomar para llegar a su destino. Si bien existen múltiples herramientas que posibilitan la navegación entre un lugar origen y destino, estas requieren del uso del teléfono móvil para observar o escuchar las indicaciones de dirección, ocasionando que el usuario pierda la percepción del entorno que lo rodea. Lo anterior resulta en posibles causas de accidentes viales, ya que el uso del celular no solo afecta al ciclista, si no, al tránsito en general. Partiendo de estas consideraciones, el proyecto plantea el desarrollo de una aplicación móvil encargada de obtener datos de navegación para posteriormente transmitirlo al sistema electrónico del chaleco mediante comunicación *Bluetooth*. De esta forma, se ofrece la posibilidad de que el usuario pueda seguir una ruta sin la necesidad de utilizar los sentidos de la vista o el oído. La figura 22 presenta

una ilustración relacionada al envío de datos de la aplicación al sistema electrónico del chaleco.

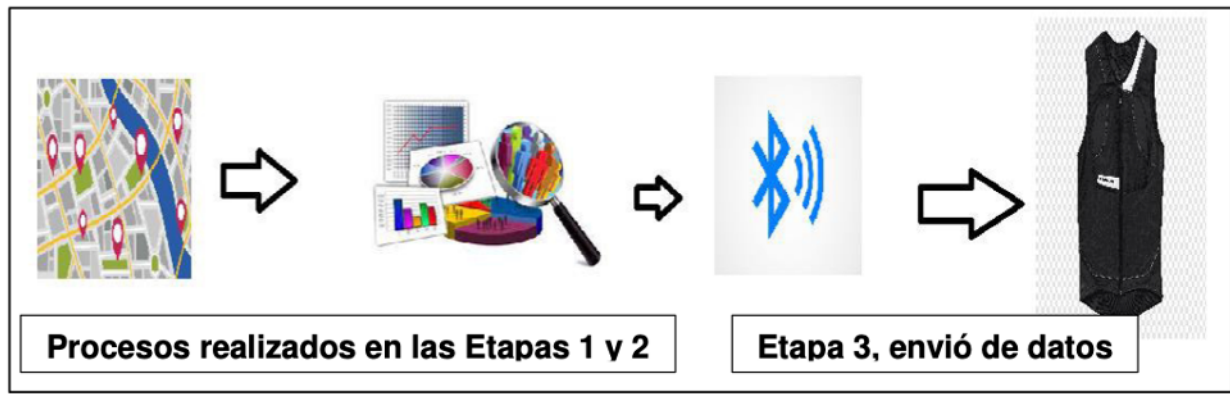


Figura 22: Ilustración del envío de datos de la aplicación al sistema electrónico del chaleco

Fuente: [17]

Por otro lado, el diseño y construcción del chaleco integra un sistema embebido ESP32 encargado de recibir los datos de la aplicación móvil y enviar las señales de activación a los motores de vibración ubicados en la parte frontal del chaleco (lado izquierdo o derecho), proporcionando las indicaciones de navegación al usuario a través de vibraciones. La figura 23 presenta el montaje y soldadura de los elementos electrónicos en la placa, la figura 24 presenta el montaje de los elementos electrónicos en el chaleco y la figura 25 muestra la interfaz de la ventana ruta en la aplicación móvil.

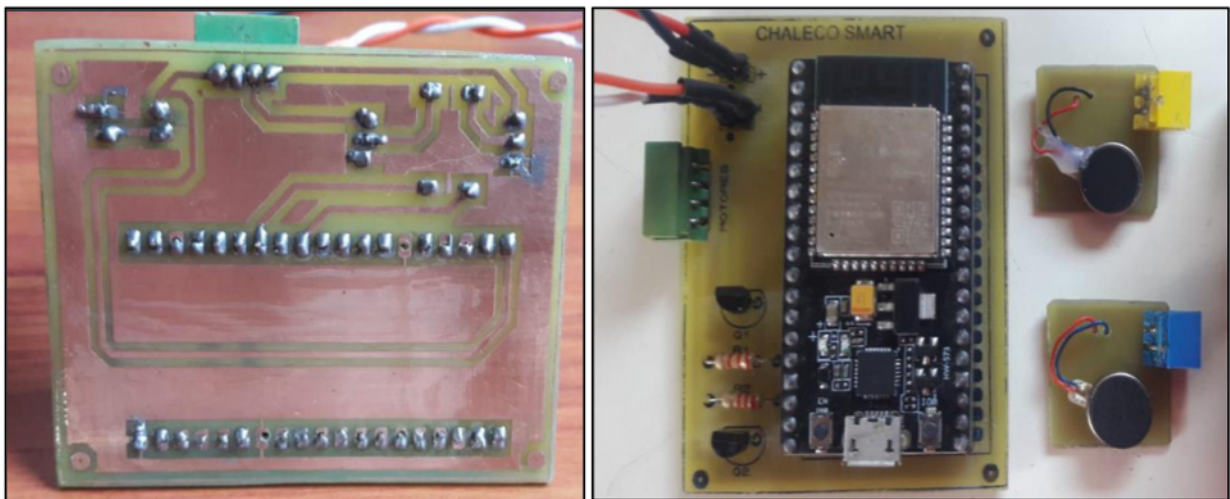


Figura 23: Montaje y soldadura de los elementos en la placa

Fuente: [17]



Figura 24: Montaje de elementos electrónicos en el chaleco
Fuente: [17]

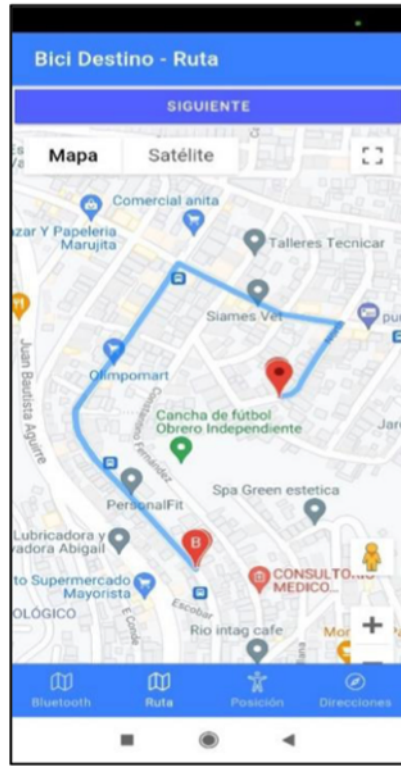


Figura 25: Interfaz de la ventana ruta

Fuente: [17]

En resumen, la tesis titulada “Implementación de un sistema de navegación para ciclistas, que le permita llegar a una dirección” contempla un proyecto que abarca el desarrollo de una aplicación móvil encargada de recopilar los datos de navegación de servidores como *Google Maps* para posteriormente enviarlos a un sistema electrónico integrado en un chaleco, este cuenta con un sistema embebido ESP32 como dispositivo central. De esta manera, el dispositivo recibe los datos enviados por la aplicación móvil para comandar motores de vibración localizados en la parte frontal del chaleco, incluyendo tanto su lado izquierdo como el derecho. Proporcionando indicaciones de navegación al usuario mientras se dirige a su destino por medio de una serie de vibraciones.

4. Metodología y plan de trabajo

4.1. Formación introductoria

En esta sección se desarrolla una etapa introductoria del trabajo realizado dentro del plan de prácticas, para ello se describe, inicialmente, el proceso de capacitación y aprendizaje que se dio previo al desarrollo de todos los proyectos asignados por parte de la empresa dentro de las prácticas realizadas. Las capacitaciones son necesarias para el correcto desarrollo de toda actividad dentro de una empresa o área que comprenda la realización de proyectos importantes, ya sea de pequeñas o grandes dimensiones, esto con el fin de llegar a obtener los mejores resultados que permitan el avance y desarrollo progresivo de la compañía. Para mejorar la etapa de capacitación es necesario el planteamiento de una ruta de aprendizaje.

4.1.1. Capacitación inicial

Dado el enfoque de la empresa y su área de trabajo, se debió tener una capacitación inicial para poder entender la magnitud de las necesidades y requerimientos de la empresa para poder llevar a cabo los proyectos y diversas tareas asignadas dentro del plan de prácticas, de tal manera que el aprendiz esté en la completa capacidad de cumplir con todas las responsabilidades que le sean asignadas. Inicialmente, se dio una capacitación donde participaron todos los integrantes de prácticas del área de tecnología y desarrollo que se encarga de desarrollar todos los proyectos y requerimientos de la empresa. Dado el contexto de movilidad y la integración de la tecnología desarrollada por la empresa para mejorar la visibilidad de los diferentes actuadores viales en las vías y carreteras del país: ciclistas, motociclistas, patinadores, personas con discapacidad que hacen uso de dispositivos auxiliares de movilidad (silla de ruedas), etc.; con el desarrollo social y avance tecnológico del país, se da inicio con unas charlas dadas por la Secretaría de Movilidad de la ciudad de Bogotá con los diferentes actores involucrados en los procesos de seguridad vial, con el fin de mejorar y proyectar el uso de sistemas de transporte alternativos dentro de la ciudad. Dentro de estas charlas, se dio a conocer todo el compromiso que tiene la alcaldía con el sector de movilidad y la inversión destinada a mejorar las diversas falencias que se presentan en dicho sector, así como el compromiso con aportar al progreso del área tecnológica del país con el propósito de hacer de la ciudad una ciudad tecnológica, que sea capaz de competir con los países que ya tienen más experiencia en estas áreas de la tecnología.

Una vez finalizadas las charlas que se dieron entre la Secretaría de Movilidad y los diferentes actores que aportan a la movilidad, se capacita al cuerpo del área de tecnología en diversas áreas como el Big Data por parte de un empleado de la compañía colombiana Kiwi Campus, la cuál es una empresa de servicios de delivery con presencia en Estados Unidos que hace uso de un robot llamado Kiwi para realizar dichas entregas. En esta charlas se da a conocer la importancia del uso de los datos para mejorar la experiencia del usuario a partir del procesamiento de toda la información con técnicas de *Big Data* en plataformas de *Cloud Computing*. Todos estos datos procesados permiten mejorar el desempeño y autonomía del Kiwibot gracias al aprendizaje que se obtiene de los datos que se van recopilando. Además, gracias a esta charla se obtiene información sobre diversos sistemas de mapas de tipo open source, alternativos a la API de *Google Maps*, para poder minimizar los dentro de una compañía que requiera el uso de mapas, para este caso hablamos de tecnologías como *OpenStreetMap* el cual es un proyecto colaborativo completamente abierto hecho en tecnologías

como *javascript*, que permite el uso de mapas en proyectos donde sea necesario y al ser licencia Open Source se mantiene completamente actualizado por toda la comunidad.

Posterior a todas estas charlas de capacitación inicial sobre al área de movilidad y seguridad para diferentes tipos de actores viales, el cual es el enfoque de la expresa, se da inicio con una ruta de aprendizaje de temas necesarios para el desarrollo de software, en este caso enfocado directamente a dispositivos móviles así como el desarrollo de *firmware* para microprocesadores de la familia ESP-32. Así como los conceptos básicos para el manejo de estados y versiones de la mano del repositorio *GitHub*, los conceptos básicos para desarrollo de código bajo el paradigma de la programación orientada a objetos utilizado en el lenguaje Dark, el cual es el motor del *framework Flutter*, necesario para el desarrollo de la aplicación móvil; usado gracias a la facilidad que ofrece para escribir código destinado a más de una plataforma móvil: Android, IOS.

4.1.2. Introducción a Git y GitHub

Utilizando el control de versiones *Git*, que permite crear diferentes copias de un mismo código para el manejo de diferentes versiones de un proyecto y que de esta manera los desarrolladores sean capaces, no sólo de administrar el proyecto de una forma ordenada, sino que también son capaces de comparar un mismo archivo o fichero en diferentes versiones del mismo proyecto sin alterar su funcionalidad en modo *release*, *GitHub* es uno de los repositorios y *software* de control de versiones más utilizado por los desarrolladores gracias a todas las herramientas que integra la facilidad de crear código de forma colaborativa entre diferentes desarrolladores, proyectos de código abierto, aplicaciones, etc. Permite la restauración de versiones anteriores, fusión de cambios, creación de diferentes ramas de un mismo proyecto haciendo más fácil la depuración de errores de código y bugs.

La plataforma cuenta con dos tipos de repositorios; públicos y privados. Todos los proyectos de código abierto y donde toda la comunidad pueda hacer aportes al desarrollo del código sin restricciones no tiene costo, en caso de querer crear y alojar un proyecto de forma privada para que no todo el mundo tenga acceso al mismo, GitHub ofrece una forma de pago. Estas son unas de las herramientas que ofrece este *software* para diferentes tipos de enfoques en proyectos.

Otra de las grandes herramientas que ofrece esta plataforma es la posibilidad de crear una documentación completa del código del código mediante una *Wiki* para mantener toda la información disponible sobre funcionalidades del código así como todos los cambios que pueden tener cada una de las versiones del proyecto. Permite, además, la creación de discusiones alrededor del proyecto para que todos los desarrolladores activos en el mantenimiento del proyecto puedan dejar mejoras, corrección de errores, optimización de código, etc. [18]

Para el control de versiones de código usando *Git*, la plataforma ofrece una gran variedad de comandos para los desarrolladores que permite un rápido manejo de versiones de un proyecto. En la tabla 1 se especifican los principales comandos *Git* [19]:

Country List	
Comando	Descripción
Git init	Comando que permite crear un repositorio dentro de un entorno local del desarrollador.
Git clone	Comando para clonar un proyecto que se encuentra en un repositorio remoto a un repositorio local.
Git add	Comando para agregar todos los archivos, o uno en específico, al área de trabajo del repositorio local.
Git commit	Comando para confirmar los cambios realizados a un proyecto dentro de un repositorio local.
Git rm -r	Comando para eliminar archivo o carpeta.
Git branches	Comando para listar todas las ramas de un proyecto o para crear una nueva rama.
Git push	Comando para enviar todos los cambios de un repositorio local a un repositorio remoto.
Git checkout	Comando para cambiar a una rama en específico.
Git merges	Comando para fusionar dos ramas.
Git pull	Comando para bajar todos los cambios de un repositorio remoto a un repositorio local.

4.1.3. Introducción a la Programación Orientada a Objetos POO

La programación orientada a objetos es uno de los paradigmas más utilizados actualmente en la programación para el desarrollo de *software*, gracias a todas las ventajas que ofrece sobre paradigmas más antiguos como la programación funcional. Dentro de la POO todas las operaciones se almacenan dentro de unidades lógicas llamadas objetos y a su vez, toda la información se contiene dentro atributos que pertenecen a dicho objeto y hace referencia a las propiedades, mientras que toda la funcionalidad lógica del objeto se almacena dentro métodos, que se puede entender como funciones haciendo la comparación con la programación funcional; sin embargo, no se puede llegar a decir que son lo mismo.

Dart es un lenguaje que soporta este paradigma de programación, facilitando los tiempos de escritura y desarrollo de código. Como la mayoría de los lenguajes de programación que soportan este paradigma *Dart* permite la herencia de clases, la creación de clases abstractas y la creación de interfaces, sin embargo *Dart* no soporta la herencia múltiple por lo que es necesario usar algo propio del lenguaje llamado *mixin*, que son una solución sencilla para poder implementar la herencia múltiple de otro forma. En la figura 26 se ve un ejemplo de clases abstractas en lenguaje *Dart* y en la figura 27 se ve un ejemplo de *mixin*. usado para resolver la herencia múltiple.



Figura 26: Clases abstractas y herencia simple
Fuente: Autor

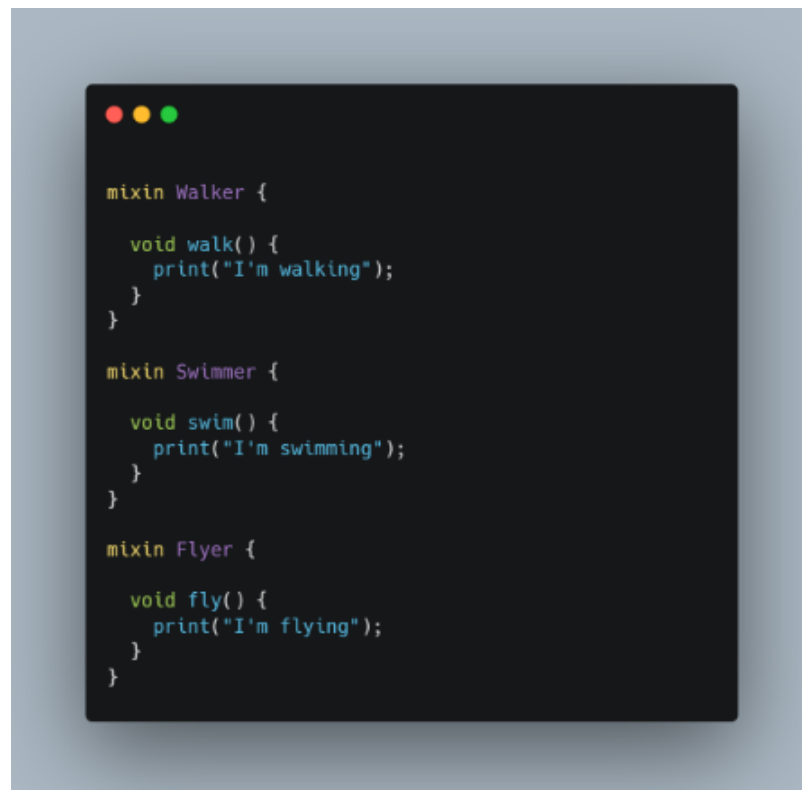
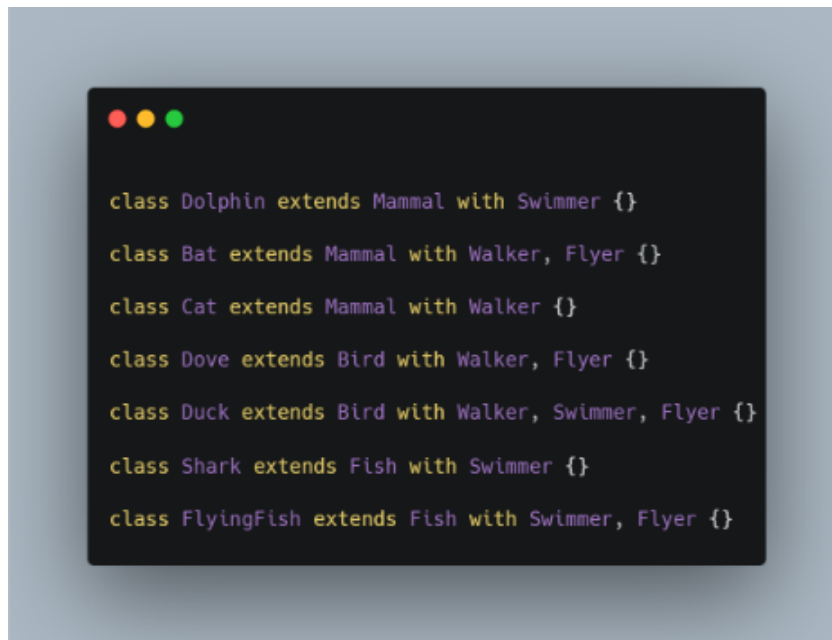


Figura 27: Mixin
Fuente: Autor



```

class Dolphin extends Mammal with Swimmer {}
class Bat extends Mammal with Walker, Flyer {}
class Cat extends Mammal with Walker {}
class Dove extends Bird with Walker, Flyer {}
class Duck extends Bird with Walker, Swimmer, Flyer {}
class Shark extends Fish with Swimmer {}
class FlyingFish extends Fish with Swimmer, Flyer {}

```

Figura 28: Herencia múltiple mediante uso de mixin.

Fuente: Autor

En este ejemplo sencillo realizado durante el tiempo de aprendizaje se hace uso de las principales funciones que ofrece la programación orientada a objetos dentro del lenguaje *Dart*, en este caso se hace uso de las clases abstractas que no pueden ser instanciadas sino heredadas. Primero se crea una superclase padre llamada *Animal* y todos sus atributos y métodos son heredados en las clases hijas para definir cada uno de los tipos de animales, para poder hacer uso de la herencia múltiple se crean los *mixin* que definen cada una de las habilidades que tienen los animales: nadar, volar, caminar. De esta forma puedo crear una clase que tenga todos los atributos y métodos de la clase mamífero y a su vez agregarle todos los métodos y atributos que pueda llegar a tener este animal como se aprecia en la figura 28. Un ejemplo es la clase *Duck*, pues este animal cuenta tanto con la habilidad de caminar como con las habilidades de volar y nadar.



```

Duck duck = Duck()
duck.fly()

```

Figura 29: Instancia de la clase *Duck*.

Fuente: Autor

De esta forma se puede crear una instancia de la clase *Duck*, y acceder a todos los métodos y propiedades, tanto de la superclase *Animal* como de la clase hija *Bird* y a su vez de los *mixin*

creados para implementar la herencia múltiple. Para ello, nos referimos a la figura 29

4.1.4. Introducción los servicios de AWS

Para la primera versión de la aplicación móvil, toda la arquitectura del *backend* se encontraba montada originalmente en los servicios de AWS de Amazon, esto debido a que la empresa contaba con un crédito por una suma total de 15000 dólares, los cuales debían ser utilizados en los servicios de dicha plataforma. Uno de los servicios que ofrece Amazon AWS es Amazon Cognito, que es un servicio para la autenticación y autorización de usuarios dentro de plataformas como lo pueden ser páginas web, aplicaciones web o aplicaciones móviles. Es una solución rápida y práctica que permite minimizar los tiempos de desarrollo pues no es necesario crear una estructura completa pues esta ya se encuentra creada. Para su conexión con el *framework* de Flutter se hizo uso de un plugin de código abierto mantenido por la comunidad que permitió establecer una conexión entre los servidores de AWS a través de Amazon Cognito y la aplicación móvil. De esta forma el usuario podría registrarse haciendo uso de cualquiera de sus redes sociales. En la figura 30 se ve la plataforma de Amazon Cognito en la Web.

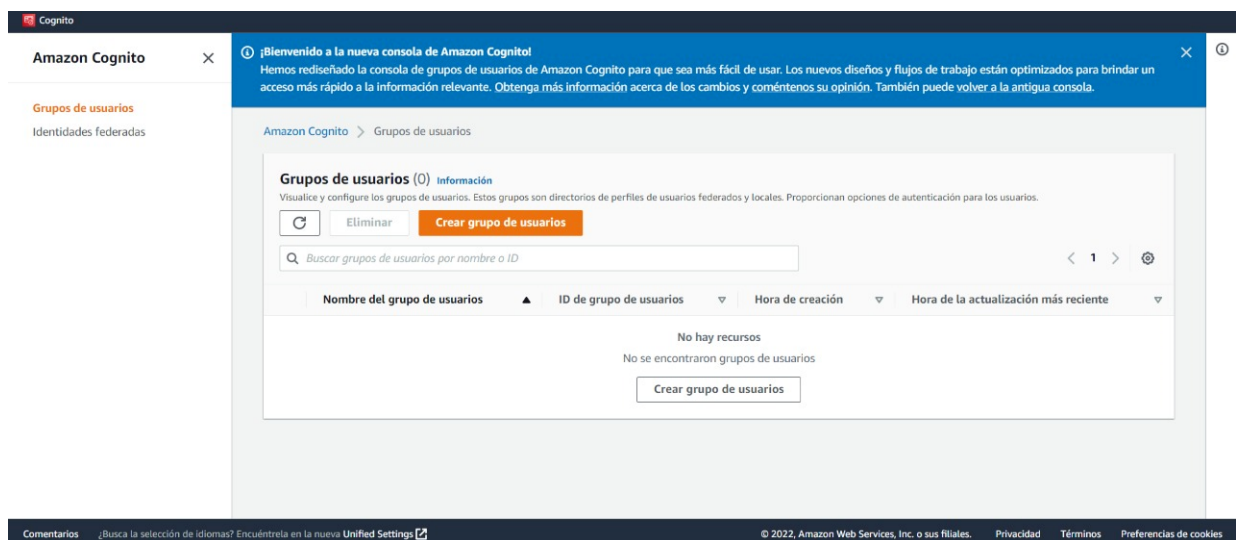


Figura 30: Amazon Cognito.

Fuente: [20]

Esto permitió tener un registro rápido de todos los potenciales usuarios de la aplicación móvil, sin embargo Amazon Cognito tiene varias limitaciones en cuanto a la escalabilidad de una aplicación ya que pierde su potencial cuando se trata de un gran volumen de usuarios debido a que su estructura no se encuentra realizada en una base de datos robusta como lo son las bases de datos de tipo SQL y NoSQL.

Otro de los servicios que fueron utilizados en primera instancia en la versión inicial de la aplicación y que también hace parte de los servicios que se encuentran disponibles dentro de Amazon AWS es Amazon S3, un sistema de almacenamiento rápido que permite subir a la nube todos los datos que puedan llegar a ser generados por un usuario, para nuestro caso de estudio, se hizo uso de este servicio para almacenar de una forma rápida y sencilla, sin la necesidad de crear una estructura

en alguna base de datos relacional o no relacional; para almacenar toda la información de posicionamiento en tiempo real del usuario, rutas guardadas y toda la información personal que el usuario proporcionaba a través de la aplicación para su identificación dentro de la aplicación móvil. La conexión entre el *framework* de *Flutter* y el servicio de Amazon S3 se hizo, de igual forma, haciendo uso de un plugin de código abierto mantenido por la comunidad, facilitando el desarrollo de código para la conexión entre el *frontend* y el *backend*. La plataforma Web del servicio de Amazon S3 se aprecia en la figura 31.

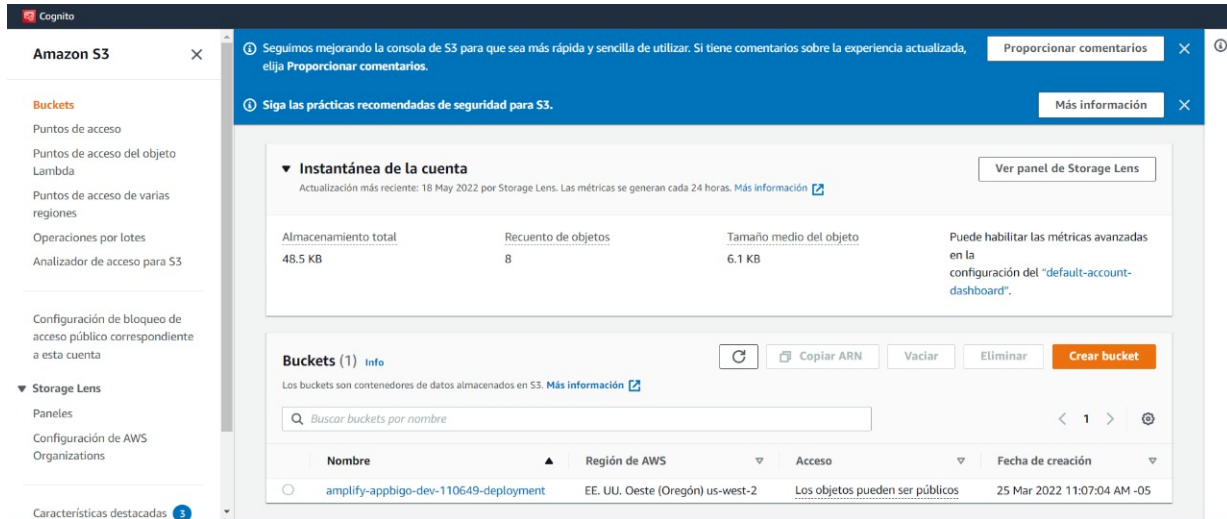


Figura 31: Amazon S3.
Fuente: [21]

4.2. Mejoras tecnológicas

Con el objetivo de realizar mejoras tecnológicas bajo la dirección de la compañía Cinnov S.A.S. Se diseña, planifica e implementa una actualización del firmware en relación al control del dispositivo desde una aplicación móvil. Lo anterior, para incrementar la versatilidad del sistema ofreciendo al usuario la capacidad de emitir los pulsos al señalizador luminoso sin la necesidad de contar con el sensor de movimiento requerido en versiones anteriores. De esta manera, se libera un nuevo método para el equipamiento y uso de dicho dispositivo. Por otro lado, resulta conveniente especificar la realización de un diseño de NAVI como complemento y a su vez, controlador alternativo a la aplicación móvil y sensor de movimiento anteriormente mencionado con el fin de expandir la comercialización de este producto, ampliando las opciones de control que pueda adquirir el usuario. Finalmente, se asistió en el proceso de diseño y fabricación de componentes modulares a los dispositivos BiGo/NAVI con el objetivo de extender las próximas operaciones o acciones de control que deberán ejecutar estos dispositivos en escenarios futuros.

Por otro lado, con el fin de realizar una transición del TRL8 al TRL9 se propone la realización de unos moldes de inyección fabricados en PLA. En estos moldes va inyectado la tecnología, para este caso de estudio se trata del dispositivo Bigo, y de esta forma se logra cumplir con la condición de TRL9 de tener procesos de producción masiva para pasar a un estatus industrial a nivel tecnológico.

4.3. Actualización de paquetes y librerías

Esta fue una de las primeras tareas que me fueron asignadas como practicante de ingeniería dentro de la empresa, a pesar de ser una tarea sencilla en cuanto complejidad técnica, fue necesario tener unas bases tanto en el lenguaje Dart como en su *framework* para desarrollo de aplicaciones multiplataforma; *Flutter*. Inicialmente, se debió hacer una actualización de la versión de *Dart* en que se encontraba desarrollada la aplicación hasta ese momento, la cual era la versión 2.5 y fue necesario actualizar hasta la versión 2.9 del lenguaje. Esto implica también una necesidad de realizar una actualización del *framework Flutter*, que para la versión 2.9 de Dart es necesario usar la versión 1.20.0 del SDK de *flutter*.

Una vez actualizado tanto el lenguaje como el SDK del *framework*, varios de los paquetes y librerías que hasta ese momento se estaban utilizando dentro de la aplicación móvil quedaron obsoletos, pues ya existían versiones actualizadas disponibles para su actualización. Los paquetes que debieron actualizarse para tener compatibilidad con las nuevas versiones fueron los utilizados para realizar la implementación de la API de *Google Maps*, otro paquete importante y que debió ser actualizado fue el que permite establecer la conexión bluetooth con el dispositivo Bigo y posteriormente con Navi también. Por otro lado, el cambio en las versiones de Dart y el SDK de Flutter significó algunos cambios en la sintaxis, por tanto, se debió corregir en todos los archivos de la aplicación móvil, de esta manera se logró realizar la compilación de la aplicación de forma correcta.

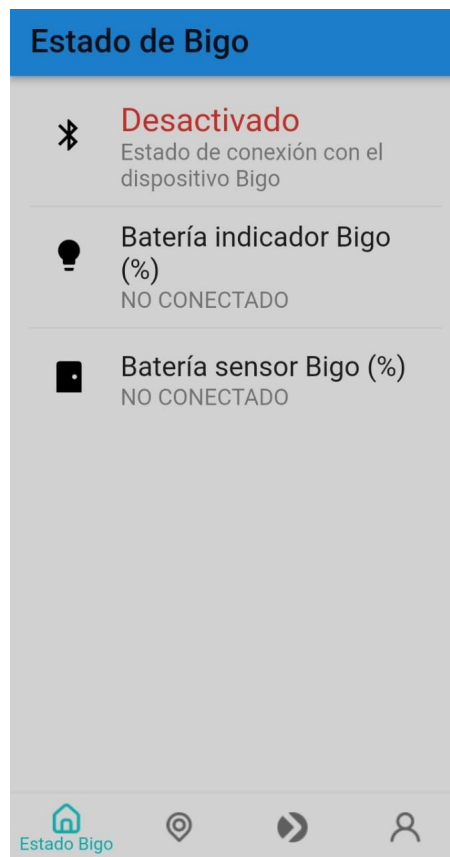


Figura 32: Vista de Dispositivos

Fuente: [22]

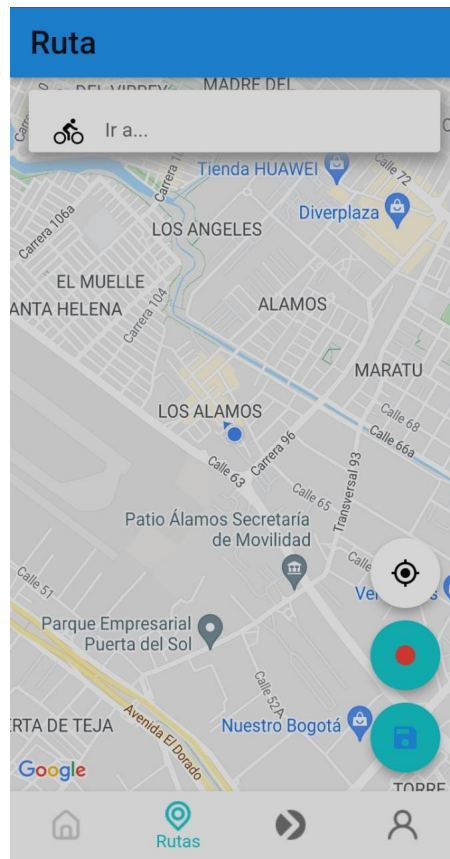


Figura 33: Vista de Mapa
Fuente: [22]

En la figura 32 y la figura 33 se aprecian dos vistas principales de la aplicación móvil. En la figura 32 se aprecia la vista que permite establecer la conexión por bluetooth con el dispositivo Bigo, donde la información que se recibe por parte de este es el estado de la batería en tiempo real, en la figura 33 se aprecia la vista de mapa donde se generan rutas para el usuario con el uso de la API de *Google Maps*, dentro de la ventana se encuentra la opción de buscar rutas, así como dos botones flotantes que permiten grabar la ruta del usuario en movimiento, de esta forma se registran todas las coordenadas en latitud y longitud por donde transita el usuario, el segundo botón permite guardar de forma local y en la nube todos los datos de la ruta grabada por el usuario, de esta forma la tiene disponible como rutas rápidas cada vez que así lo requiera el usuario.

Una vez finalizado la actualización tanto de la versión del lenguaje *Dart* como del SDK de *Flutter* y de todos los paquetes y librerías utilizados en la versión que se tenía de la aplicación hasta la fecha, se da cumplimiento de forma completa a una de las primeras tareas asignadas por parte de la empresa como ingeniero de prácticas. Hecho esto, se procedió con la realización de diferentes pruebas por parte del equipo de desarrollo, así como por parte de clientes usuarios de Bigo que tenían acceso a la versión de prueba, previa al lanzamiento de la aplicación móvil en la tienda de aplicaciones del sistema Android.

4.4. Actualización visual de la aplicación móvil

Para mejorar la experiencia de usuario con el uso de la aplicación, la empresa mandó a realizar un rediseño completo de la parte visual de la aplicación móvil con un diseñador gráfico independiente a la empresa y pensando en la escalabilidad de la aplicación, con intenciones de crear comunidad alrededor de todos los usuarios como sucede en aplicaciones como Waze, se diseñaron también nuevas ventanas y funcionalidades, que permiten a los usuarios tener una experiencia que va más allá de una aplicación GPS para generar rutas. Estas nuevas rutas están destinadas a tener disponibles a los usuarios un listado de eventos para ciclistas, establecimientos relacionados con productos para diferentes actores viales como usuarios de bicicleta, motocicleta, patineta, etc; talleres de reparación, principales números de emergencia, colectivos de actores viales a los que puede unirse y poder crear grandes comunidades que permitan el desarrollo de actividades como el cicloturismo.



Figura 34: Slider inicial
Fuente: [22]

En la figura 34 se aprecia uno de los *slider* del nuevo diseño de la aplicación móvil, estos se muestran al usuario al abrir la aplicación por primera vez, de esta forma no se vuelven a visualizar una vez completado todo el registro y la información se almacena tanto de forma local en el dispositivo del usuario como en los servidores montados en la nube de la empresa. En la figura 35 también vemos otro *slider* mostrado a los usuarios.



Figura 35: Slider inicial
Fuente: [22]

Para esta tarea fue necesario realizar una introducción a la arquitectura de diseño BLoC que permite separar la lógica de negocio de una aplicación y gestionar los diferentes estados que pueda llegar a tener una aplicación móvil. Esta arquitectura es un poco más compleja a la MVVP utilizada en la primera versión de la aplicación móvil, sin embargo permite un mejor desempeño conforme se tenga una mayor cantidad de usuarios pues es capaz de gestionar los recursos de físicos gracias a la reutilización de código y al nivel de abstracción que tiene. Inicialmente, la aplicación móvil tenía Provider como gestor de estados combinado con la arquitectura MVVP, sin embargo el código se

convirtió en lo que se conoce como código spaghetti, esta fue la razón por la cual la empresa decidió realizar el cambio al patrón BLoC.

De esta forma, toda la lógica de negocio de las vistas queda en su respectivo fichero BLoC, donde se gestionan todos los eventos y estados que se puedan tener, conexión con API, bases de datos locales y remotas. La abstracción de la arquitectura BLoC le da una escalabilidad a futuro, gracias a la facilidad de modificación de la lógica de cada una de las clases abstractas. Para poder usar las clases de los ficheros BLoC se utilizó el principio de inyección de dependencias.

4.5. NAVI

Las últimas estadísticas disponibles por la Secretaría de Movilidad de Bogotá, que son del año 2019, indican que diariamente en el distrito se realizan un aproximado de 1.2 millones de viajes por parte de biciusuarios con un tiempo promedio de viaje de 38 minutos. De ese total de viajes diarios, se distribuye de tal forma que el 74% de los biciusuarios diarios son hombres y el 26 % restante son mujeres y las localidades donde se presentan la mayor cantidad de viajes es en las localidades de: Bosa, Kennedy, Suba y Engativá. Dada la gran cantidad de biciusuarios se hace común la modalidad de hurtos de este sistema de transporte alternativo; las estadísticas disponibles por parte de la Secretaría de Movilidad y del Distrito de Bogotá indican que para el año 2020 se presentaron un total de 10.856 hurtos de bicicletas en la ciudad, mientras que para el año 2021 se registraron un total de 9.623 robos; lo que representa una disminución de un 11.35 % en los robos registrados en la ciudad. Sin embargo estas cifras no muestran completamente la realidad de los hurtos que se presentan en el Distrito pues se construyen a partir de las denuncias realizadas por parte de los biciusuarios que se han visto afectados por esta modalidad de hurto, por tanto se hace difícil poder calcular la cifra real de bicicletas hurtadas al año en la ciudad, donde se incluirían los casos denunciados y los casos no denunciados. [23]



Figura 36: Convocatoria: RETO CANDADO BICI

Fuente: [24]

El 5 de noviembre de 2021 nace la convocatoria llamada "Bici-candados para todos y todas", que se aprecia en la figura 36, la cual buscaba facilitar el acceso a todos los usuarios de un mecanismo

de seguridad de bajo coste, debido a los altos precios del mercado para los candados de seguridad, que fuese elaborado en la ciudad de Bogotá donde las mejores soluciones recibirían fondos por parte del Fondo de Innovación Tecnología e Industrias Creativas de Bogotá-FITIC. [25]

Dentro de las aclaraciones de la convocatoria se encuentra:

Necesidades de la organización: Desarrollar un mecanismo de seguridad bici (candado, tecnología u otros) de producción local y bajo costo, que sea asequible para todos los ciclistas urbanos de Bogotá. Requisitos obligatorios: Obligatorio cumplimiento de criterio “bajo costo de producción”

* Obligatorio cumplimiento de criterio de producción local. Requisitos NO deseables: Propuestas que no den respuesta a la solución planteada en el reto.

* Propuestas que por temas legales, de legislación o infraestructura, no puedan ser implementadas en Colombia.

* Soluciones existentes en el mercado de producción internacional.

Área: Ciudades Verdes e Inteligentes, Industrias Creativas, Culturales y de Contenidos. Sub áreas: Diseño sostenible, Transporte inteligente. Tecnologías: Micro y nano electrónica, *Blockchain*, *Big Data*, Materiales avanzados, Impresión 3D. [24]

Una vez claras todas las especificaciones y necesidades de la convocatoria, se procede en primera instancia a realizar una serie de bocetos iniciales por parte del departamento de diseño y paralelamente empiezo el desarrollo de la parte tecnológica que incorpora hasta este momento el dispositivo NAVI.

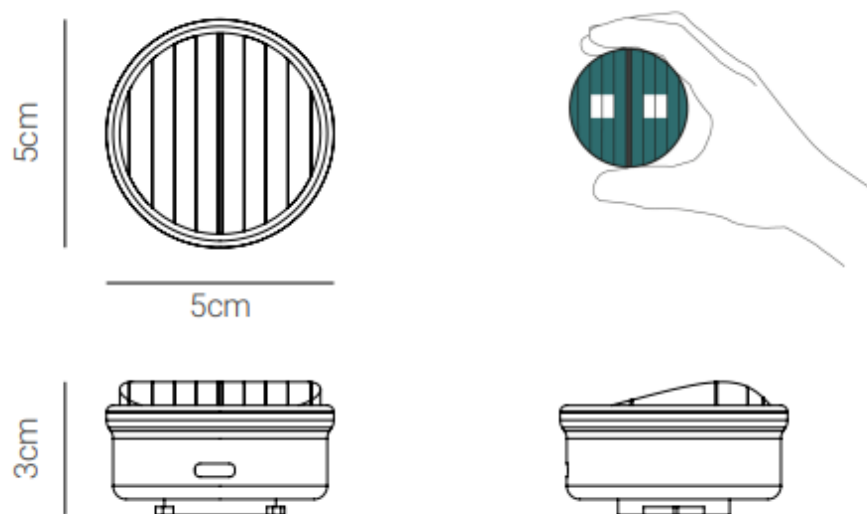


Figura 37: Dimensiones de Navi

Fuente: [26]

En la figura 37 se pueden apreciar las dimensiones finales de Navi después de diversas propuestas de diseño, donde inicialmente el dispositivo contaba con una forma rectangular, sin embargo para poder que el usuario pueda obtener una mejor ergonomía con Navi se prefirió optar, finalmente, por una geometría circular. En la figura 37 tenemos una vista superior donde se aprecia que la máxima dimensión del dispositivo donde se encuentra toda la electrónica es de tan solo 5 cm y en la vista

frontal y lateral se aprecia la altura de 3 cm del dispositivo.

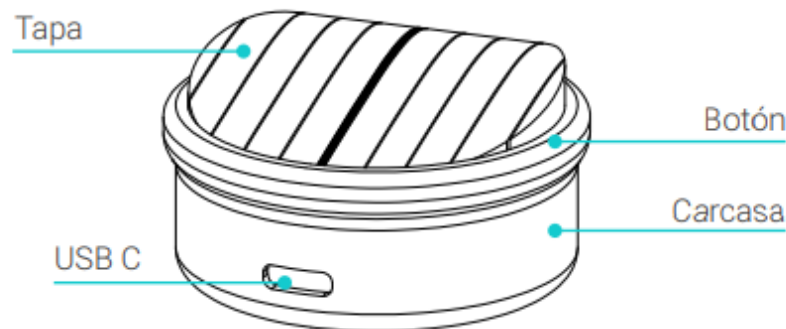


Figura 38: Partes Navi

Fuente: [26]

En la figura 38 se puede apreciar todas las partes principales que componen a Navi, la cúpula que tiene una geometría especial que permite obtener que la potencia de los led's de tipo NeoPixel no se pierda y cumplan con su función a cualquier hora del día y bajo cualquier condición climática. Esta cúpula se encuentra sobre el pulsador, por lo tanto es posible ejercer presión sobre ella para poder activarlo y realizar todos los controles disponibles y por último se señala la carcasa, la cual recubre toda la electrónica que se encuentra dentro y tiene un pequeño orificio de tamaño específico para insertar un cargador de tipo C en el módulo de carga del dispositivo.

Todas las partes, a excepción de la cúpula, se encuentran realizadas por una impresora 3D en material PLA. La cúpula se encuentra fabricada en resina de plástico que permite que la luz de los leds de NeoPixel atraviesen el material con la potencia suficiente para que el usuario que porta el Navi sea visible a cualquier hora del día. Para poder apreciarlo de una mejor forma, vamos hasta las figuras 39 y 40 donde se puede observar el diseño final de Navi impreso en 3D.



Figura 39: Vista lateral de Navi

Fuente: [26]



Figura 40: Vista posterior de Navi

Fuente: [26]

Se puede apreciar el tamaño compacto del diseño final, con un diámetro de 45 [mm], donde se encuentra toda la electrónica. La cúpula cuenta con un diseño que permite la refracción de la luz hasta un punto que la potencia de salida es suficiente para que sea visible de día, cumpliendo con el propósito de servir como dispositivo de seguridad vial para hacer más visible a los usuarios: ciclistas, motociclistas, etc. En la figura 40 específicamente, se aprecia el modelo de agarre diseñado para ser sujeto a los manillares, este se acopla a una segunda parte impresa en 3D que se instala sobre los diferentes manillares.

Para cumplir con el objetivo de la convocatoria, Navi integra tres funcionalidades novedosas en un dispositivo de seguridad vial que permite al usuario hacer reportes accionando un pushbutton ubicado en la capa top de la PCB y el cual es accionado haciendo presión sobre la cúpula de la parte superior del Navi. De esta forma, el usuario puede reportar si ha presenciado un siniestro vial, ha sido víctima de robo o ha presenciado uno, así como reportar un acoso, dadas las altas tasas de acoso a las ciclistas en la ciudad de Bogotá. Para poder hacer el reporte de robo es necesario presionar una vez el pushbutton, para el segundo reporte es necesario presionar dos veces y en caso del reporte de acoso es necesario mantener la pulsación durante un mínimo de cuatro segundos y un máximo de seis.

Cada uno de los reportes es enviado a la aplicación móvil mediante la conexión bluetooth que se establece entre ambos, desde la aplicación se hace el envío de alertas a los contactos de emergencia registrados por parte del usuario, así como la realización automática de una llamada al número de emergencias nacional 123 o al número de acoso a las mujeres de la ciudad de Bogotá. También se almacena en la nube de AWS la posición del usuario al momento de realizar el reporte, así como toda su información personal. Para servir como complemento de Bigo, ya que este solo puede ser utilizado en la espalda del usuario, Navi también se encuentra diseñado y programado para recibir las indicaciones de GPS generadas por la API de *Google Maps* enviadas a través de la aplicación móvil. De esta forma el usuario se encuentra visible tanto en la parte frontal, como en la parte posterior de su cuerpo, minimizando la posibilidad de sufrir un siniestro vial por no ser visible en las principales carreteras del país. En caso de no ser utilizado con la aplicación móvil para generar las indicaciones de giro del usuario, este también se conecta al mismo sensor de movimiento implementado con Bigo, haciendo que el usuario sea libre en elegir su modo de uso.

Otro complemento extra, que no hace parte de las necesidades a resolver con la convocatoria, pero que sí aportan un punto de utilidad a Navi, es la posibilidad de usarlo como notificador mediante el

enlace establecido con la aplicación móvil. Este es capaz de avisar al usuario cuando está recibiendo una llamada telefónica o ha recibido un mensaje de texto, sin embargo se limita a solo servir como notificación.

Todo el *firmware* de control que se instaló en el microcontrolador del ESP 32 instalado en la PCB del Navi, fue programado en el editor de texto VS Code ya que permite el uso de diferentes lenguajes de programación, para este caso, el uso de C++; requerido para la programación del ESP 32. Gracias a la extensión PlataformIO de VS Code se tiene acceso a una gran variedad de microprocesadores y microcontroladores, así como el control y uso de librerías necesarias para el desarrollo. De esta forma se facilitan los tiempos requeridos por un proyecto para la realización del desarrollo, en este caso del dispositivo Navi.

4.6. Manual de Usuario

En este apartado se muestra el manual de usuario final generado para el correcto uso de Navi. La figura 41 hace referencia a la portada del mismo.



Figura 41: Manual de Usuario

Fuente: [26]

Controles



Usuario en movimiento



Las direccionales se indicaran con color azul



Presionar dos veces en caso de tener o ser testigo de un siniestro



Figura 42: Controles de Navi

Fuente: [26]

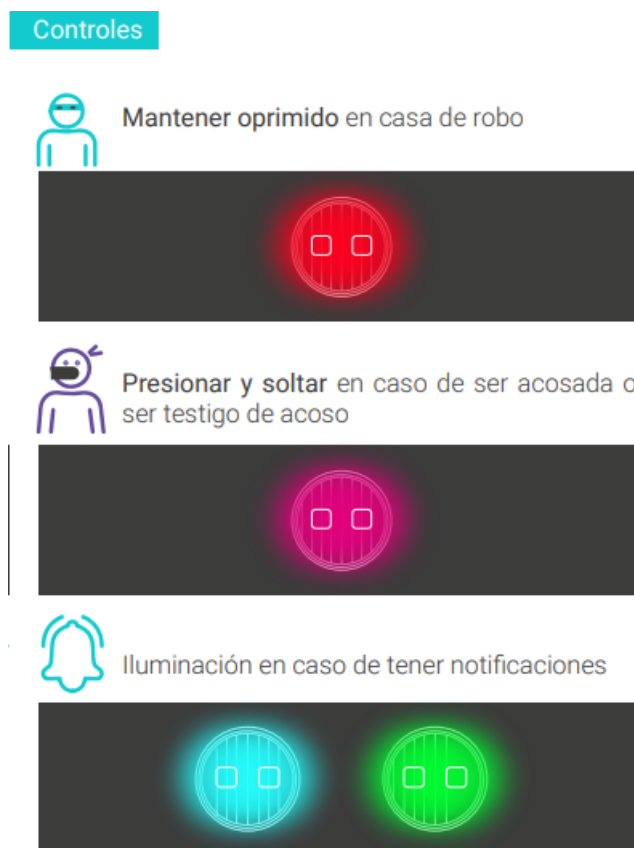


Figura 43: Controles de Navi

Fuente: [26]

La figura 42 y figura 43 indican los diferentes modo de uso que se tienen con el dispositivo, en la figura 42 específicamente se le indica al usuario los tipos de direccionales que pueden generarse, el modo fijo cuando el usuario se encuentra en movimiento recto, así como una de las opciones de reporte que tiene, específicamente para siniestros viales. La figura 43 especifica los otros dos tipos de reportes que puede realizar el usuario del dispositivo además de la opción de notificación de Navi.

Inicio del dispositivo

1. Mantenga oprimido el botón durante unos segundos.
2. Elace el dispositivo con su telefono vía Bluetooth
3. Mantenga pulsado el botón durante unos segundos hasta que se encienda las luces blancas.
4. Desde su teléfono celular encienda su Bluetooth y abra la aplicación Bi Go!. Si no tiene instalada la aplicación descarguela aquí: <https://bit.ly/37a4ujX>
5. Enlace su dispositivo NAVI con su teléfono vía Bluetooth en la lista de dispositivos que le muestra la aplicación.
6. Puede activar las funciones de reporte oprimiendo y soltando el botón de NAVI y colocando la ruta para llegar a su destino, disfrute un viaje seguro con NAVI.



Figura 44: Indicaciones Iniciales

Fuente: [26]

La figura 44 es una lista de indicaciones iniciales para que el usuario pueda enlazar el dispositivo con la aplicación Móvil y la figura 45 es una advertencia de seguridad para el buen funcionamiento del dispositivo.

Seguridad

Lea toda la información de seguridad antes de utilizar el dispositivo a fin de asegurar el uso correcto y la seguridad de nuestro usuarios



Advertencia

No usar el botón cuando **no** sea un caso de emergencia

No exponga el dispositivo a impactos o daños físicos

Evite el contacto directo con el agua

No permita que los niños o animales mastiquen o laman el dispositivo.

Figura 45: Advertencia de Uso

Fuente: [26]

5. Modelado funcional

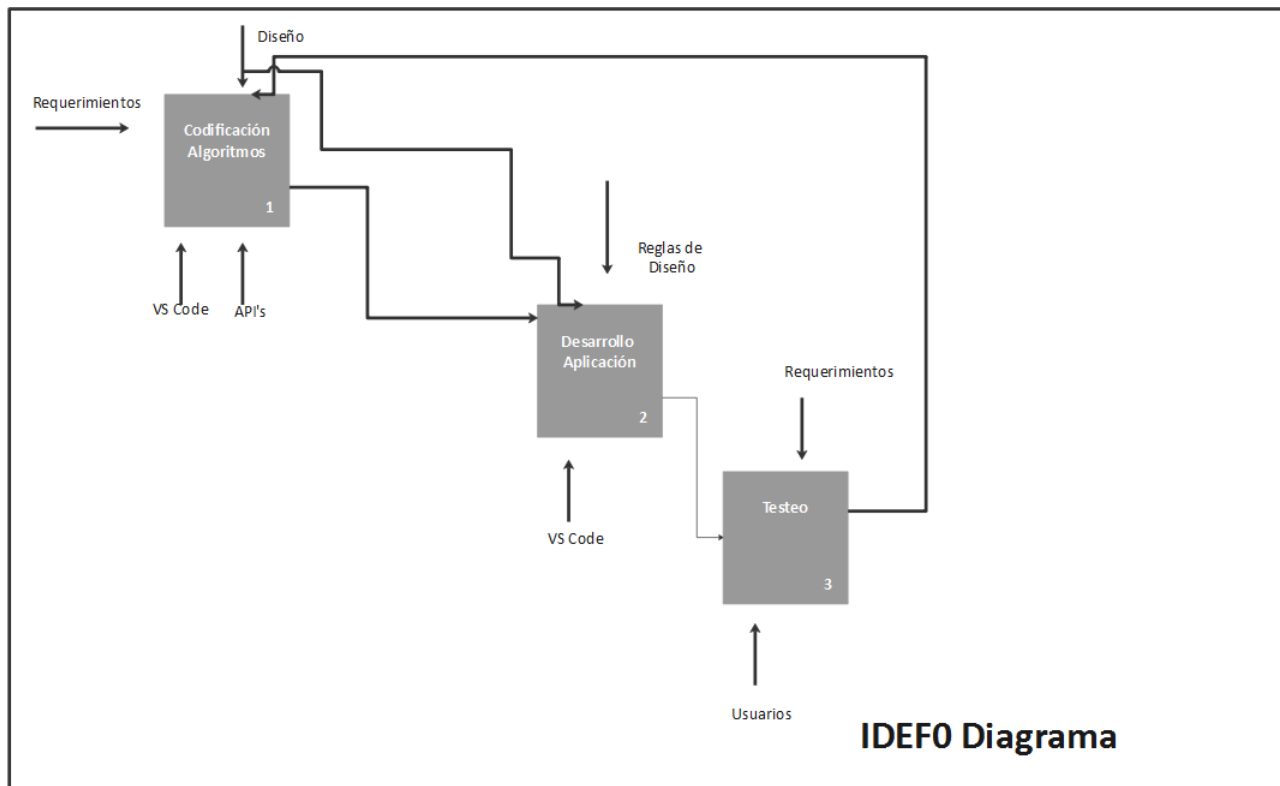


Figura 46: Diagrama IDEF0 Aplicación
Fuente: Autor

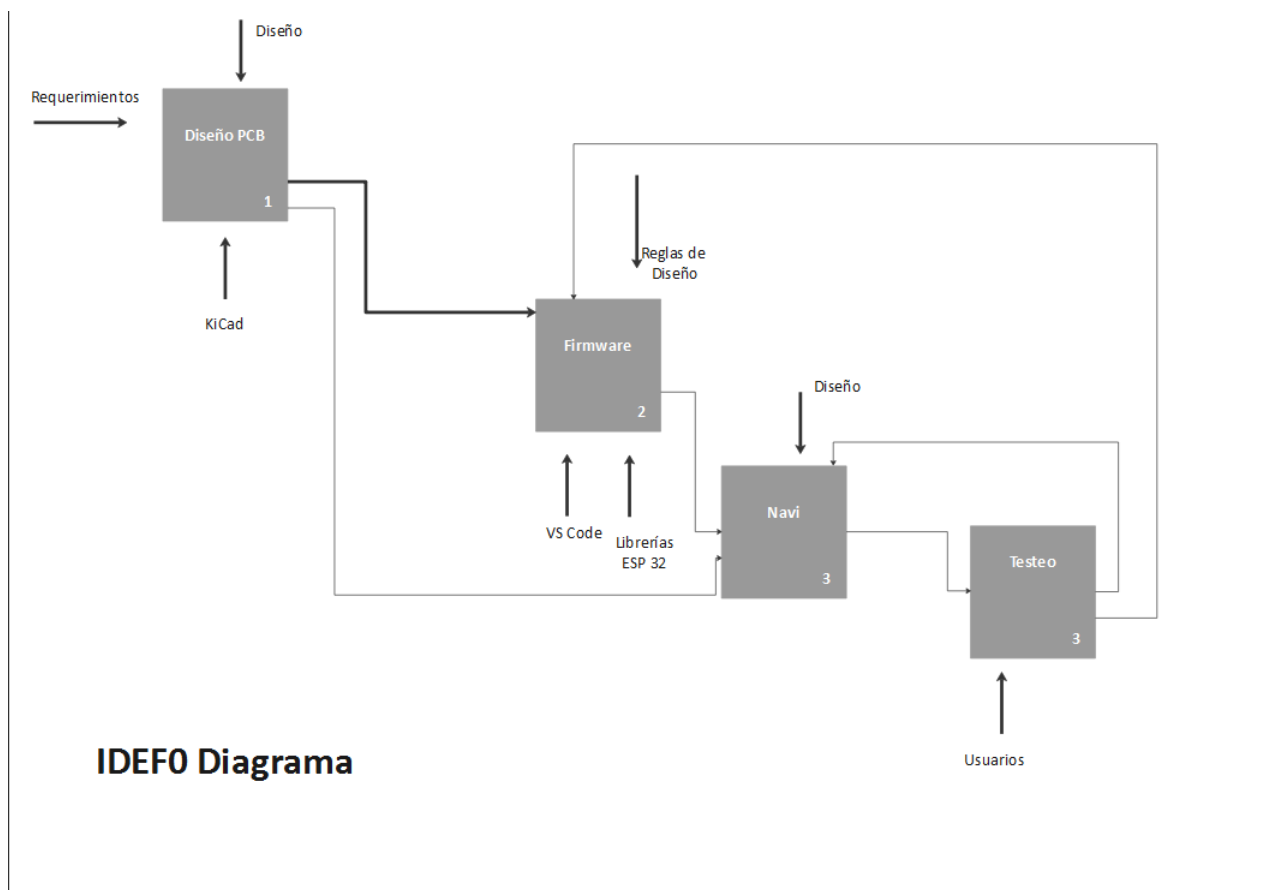


Figura 47: Diagrama IDEF0 Navi
Fuente: Autor

6. Resultados

En este apartado del libro se recopilan los resultados más importantes y documentables, durante todo el proceso de prácticas dentro de la empresa CiNNoV como ingeniero junior en desarrollo de software y hardware.

6.1. Requerimientos de Diseño

Los principales requerimientos de diseño obtenidos con el fin de aplicar las mejoras tecnológicas al dispositivo Bigo y cumplir con los objetivos son los siguientes:

- Control de Bigo por medio de un modo automático sin el uso del sensor.
- Complementar la tecnología Bigo con otro dispositivo que permita al usuario ser visible en la parte delantera.
- Desarrollar un Manual de Usuario para el nuevo dispositivo.
- Actualización del Firmware de Bigo para ser controlado por medio del nuevo dispositivo desarrollado.
- Reducir los costos de producción de Bigo.
- Funcionamiento de la aplicación móvil en segundo plano.

Los requerimientos de diseño para la actualización de la aplicación móvil se pueden ver en la figure 48

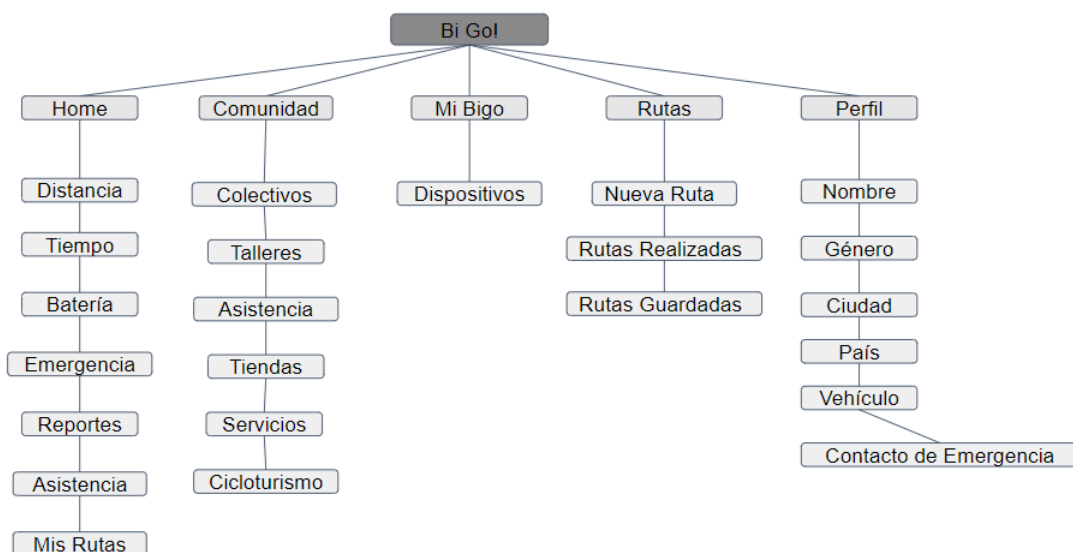


Figura 48: Requerimientos de Diseño de la Aplicación Móvil

Fuente: Autor

6.2. Mejoras Tecnológicas

En este apartado del libro se enlistan las mejoras tecnológicas más relevantes que se pueden documentar dando cumplimiento a los objetivos planteados para el desarrollo de las prácticas. Estas mejoras son el resultado de los requerimientos de diseño obtenidos.

- Transición del TRL8 al TRL9 mediante la producción masiva de moldes de inyección fabricados en material PLA para obtener el estado de nivel industrial.
- Actualización del firmware de control de Bigo que permita el uso en modo automático con la aplicación móvil. Actualización del firmware de control de Bigo para su conexión con el dispositivo Navi para la generación de indicaciones de giro.
- Finalizar el desarrollo de la aplicación móvil con funcionalidades de control inalámbrico y automático en las tecnologías Bigo y Navi.
- Desarrollo de Navi como complemento y mejora tecnológica del dispositivo Bigo.

Todas estas diversas mejoras tecnológicas fueron validadas por parte de usuarios actuales de la tecnología Bigo y todos los resultados obtenidos se encuentran protegidos también por secretos empresariales, por tanto no es posible realizar su documentación escrita.

6.3. Aplicación Móvil

La figura 49 muestra el diseño inicial de la vista de registro de usuario dentro de la aplicación móvil.

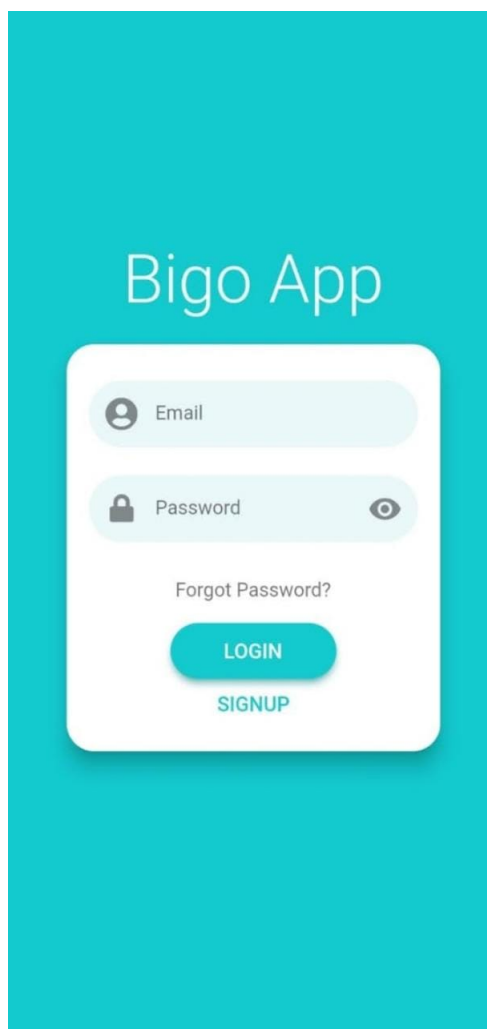


Figura 49: Vista Login
Fuente: [22]

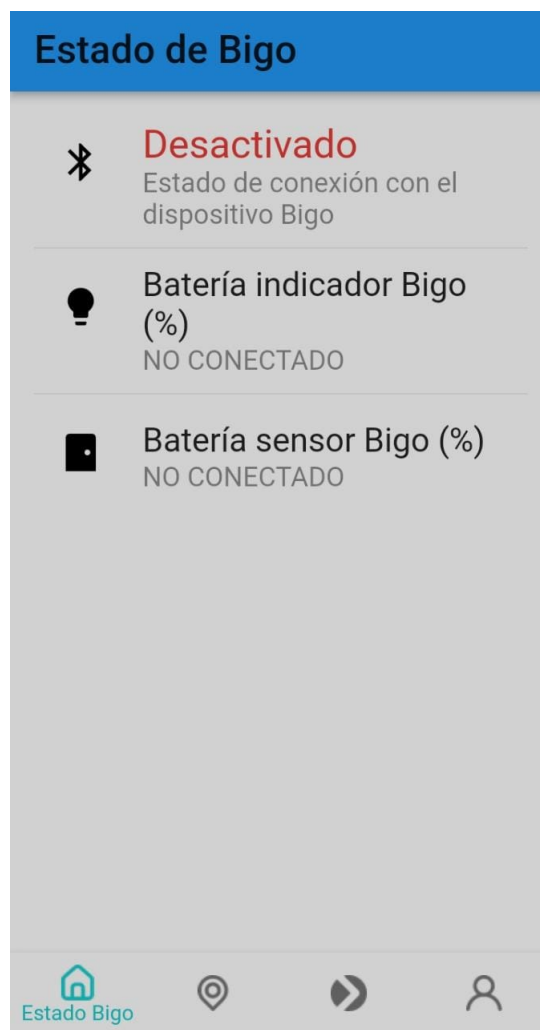


Figura 50: Vista Estado Dispositivos
Fuente: [22]

En la figura 50 se aprecia la vista que tiene el usuario para ver el estado de conexión entre la aplicación móvil y cualquiera de los dispositivos con los que es posible conectarse. Otra información que muestra esta vista es el estado de batería en que se encuentra el dispositivo Bigo o Navi, así como el nivel de batería del sensor de movimiento que indica las direccionales.

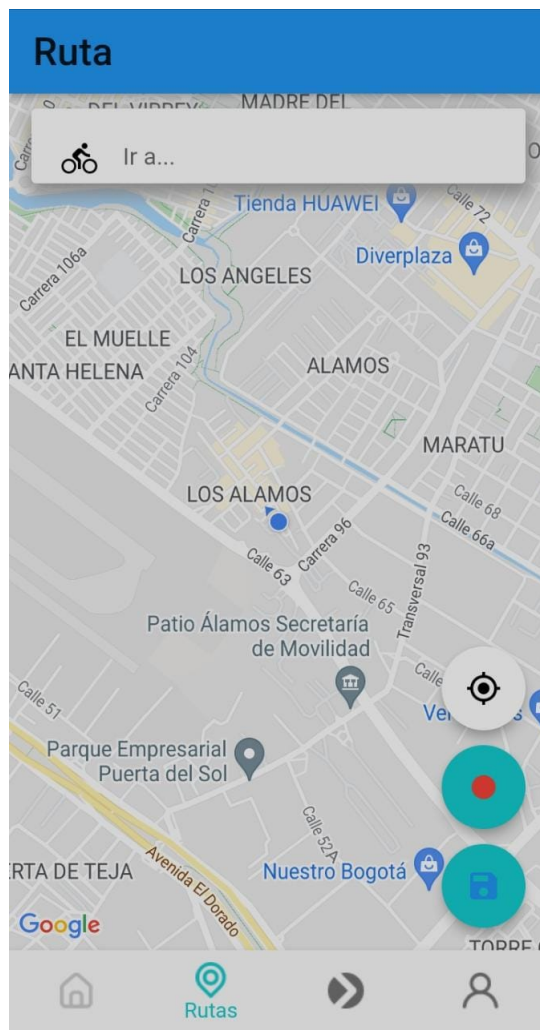


Figura 51: Vista Rutas

Fuente: [22]

En la figura 51 se aprecia el uso de la API de *Google Maps* dentro de la aplicación, dentro de esta vista se encuentra la barra de búsqueda de direcciones para la generación de rutas al usuario y los botones flotantes que permiten grabar las rutas por el usuario y guardarlas en su perfil para su uso recurrente.

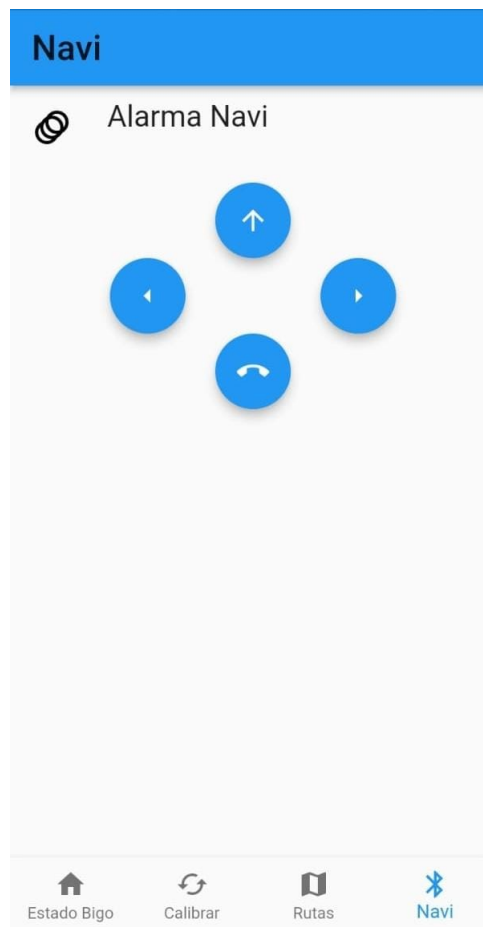


Figura 52: Vista Control Navi Manual

Fuente: [22]

La figura 52 muestra la vista de control manual diseñada para enviar las indicaciones de direccional a Navi, además cuenta con un apartado donde se le indica al usuario el tipo de alerta que ha sido reportado desde el dispositivo y un botón que permite realizar una llamada de emergencia desde la aplicación al número nacional de emergencia 123.

6.4. Actualización Visual Aplicación Móvil

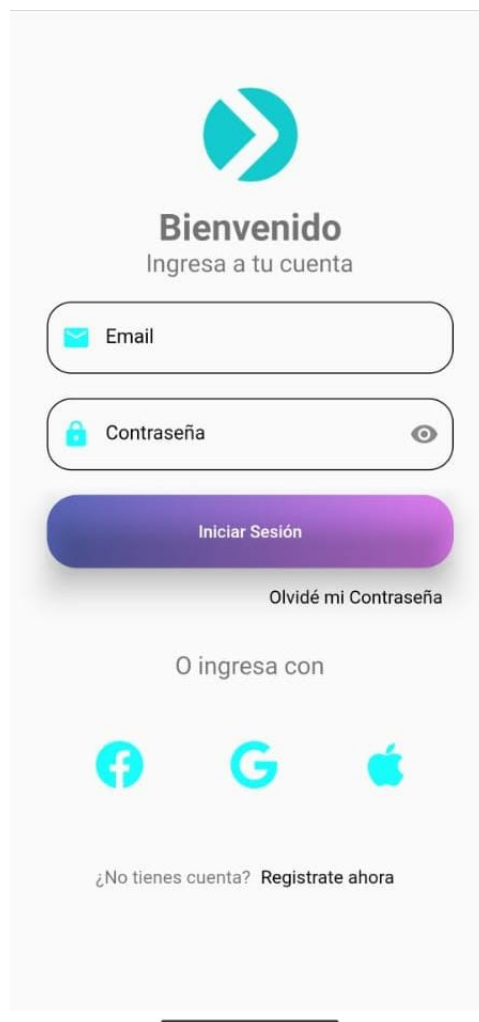


Figura 53: Vista Login Actualizada
Fuente: [22]

La figura 53 muestra el diseño final de la vista de registro de usuario dentro de la aplicación móvil donde además se cuenta con la posibilidad de realizar el registro por medio de las diferentes redes sociales que pueda tener el usuario.

SALTAR



Calibra tu dispositivo Bigo para usar

Mantente siempre seguro en la vía



Figura 54: Slider inicial

Fuente: [22]

SALTAR



**Encuentra talleres,
tiendas, eventos y más**

Una comunidad para ti



Siguiente

Figura 55: Slider inicial

Fuente: [22]

SALTAR



**Crea tus rutas, registra y
comparte tu actividad**

Encuentra la mejor ruta para tu viaje



Figura 56: Slider inicial

Fuente: [22]

Las figuras 54, 55 y 56 son las vistas que el usuario ve por primera y única vez al descargar la aplicación, estas vistas son previas al registro del usuario y hacen referencia a las funciones principales que se encontrarán dentro de la aplicación móvil.

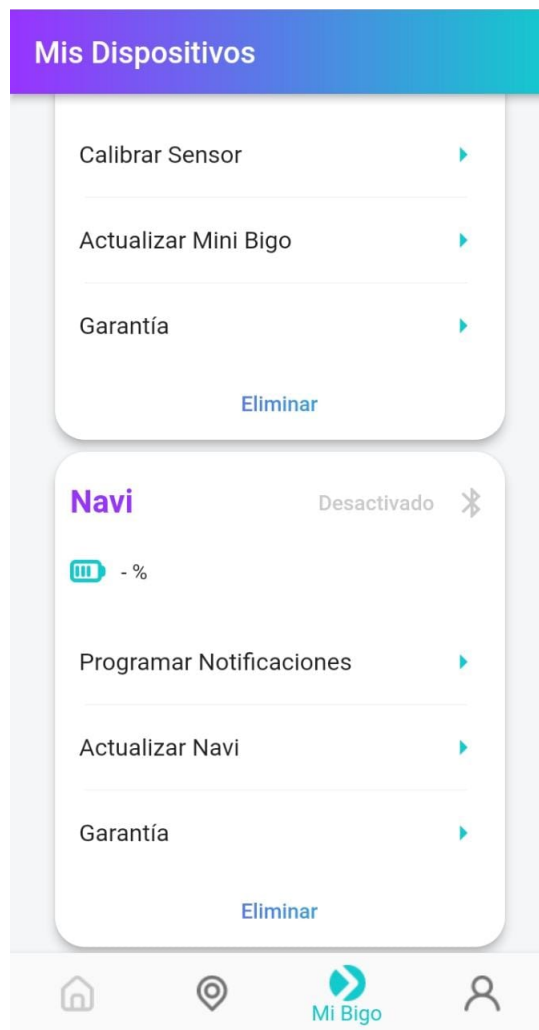


Figura 57: Vista Mis Dispositivos
Fuente: [22]

La vista que se aprecia en la figura 57 es el rediseño de la vista presentada en la figura 50 donde el usuario puede ver el estado de conexión de su dispositivo, además en este nuevo diseño el usuario es capaz de ver el estado de todos los dispositivos que ha registrado dentro de la aplicación, en la figura 57 se aprecia específicamente el estado de conexión del dispositivo Navi, además de la información del estado de conexión esta vista también brinda información sobre el estado de batería y un apartado de dirección a otras ventanas que permiten, por ejemplo, cambiar el color que Navi emite al recibir una notificación o toda la información sobre la garantía que tiene el usuario sobre el dispositivo.

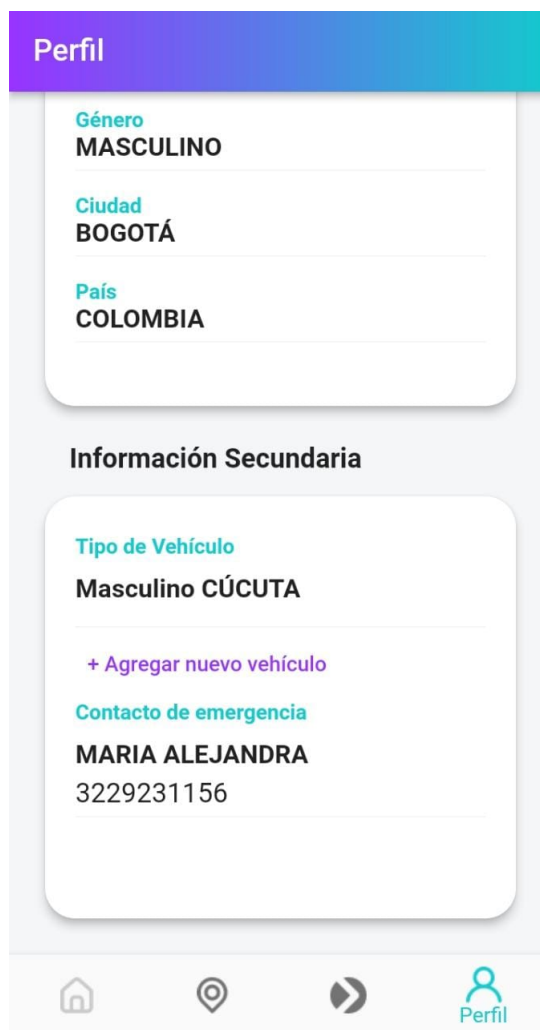


Figura 58: Vista Perfil

Fuente: [22]

La vista de perfil de la figura 58 muestra la información principal registrada por el usuario, como su ubicación y además una información secundaria que hace referencia a los vehículos registrados por el usuario o sus contactos principales de emergencia.

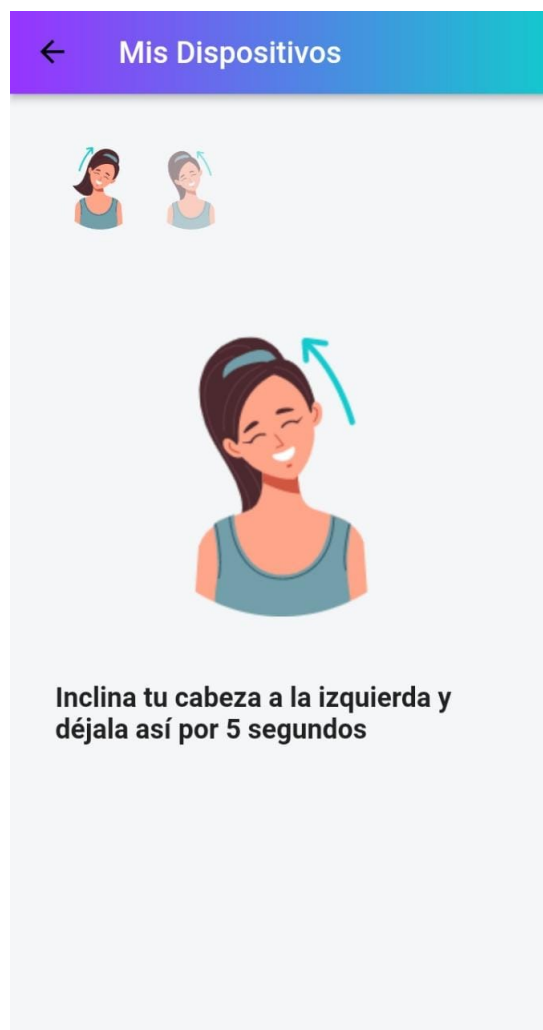


Figura 59: Vista Calibración Sensor
Fuente: [22]

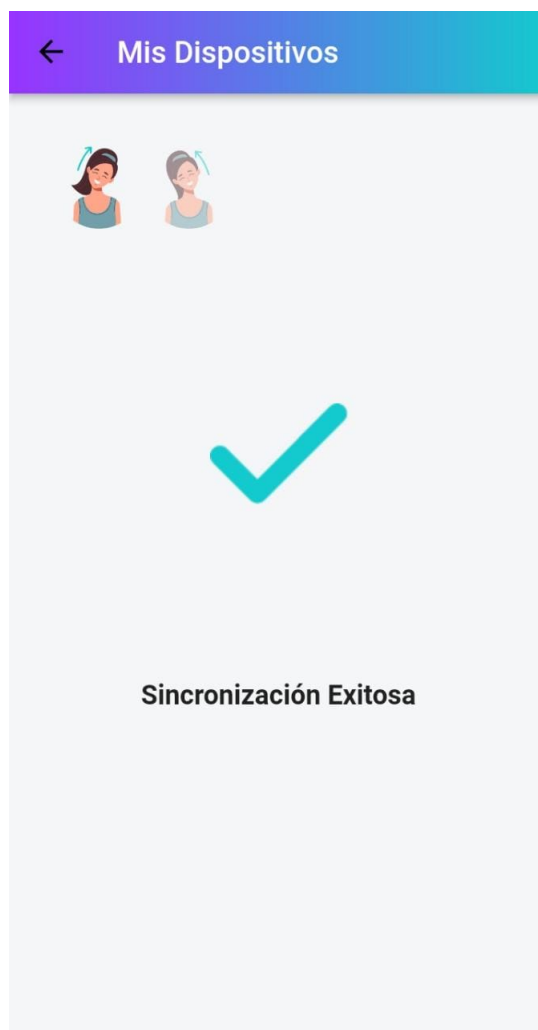


Figura 60: Vista Calibración Sensor
Fuente: [22]

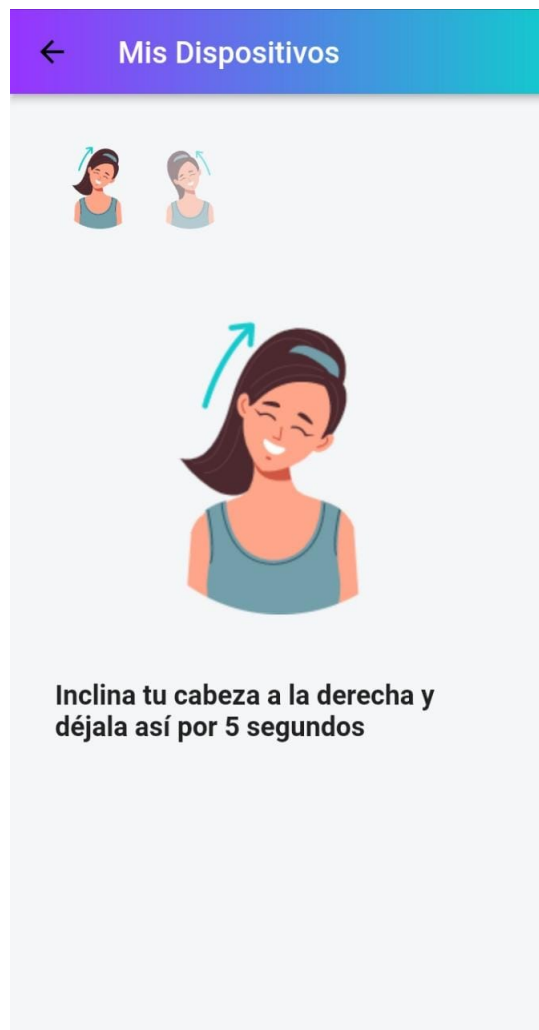


Figura 61: Vista Calibración Sensor
Fuente: [22]

Las figuras 59, 60 y 61 son las vistas disponibles para que el usuario pueda realizar la calibración del sensor, esto con el fin de darle la posibilidad al usuario de elegir el grado de inclinación que quiere realizar con el sensor para generar las indicaciones de giro tanto en el Navi como en Bigo.

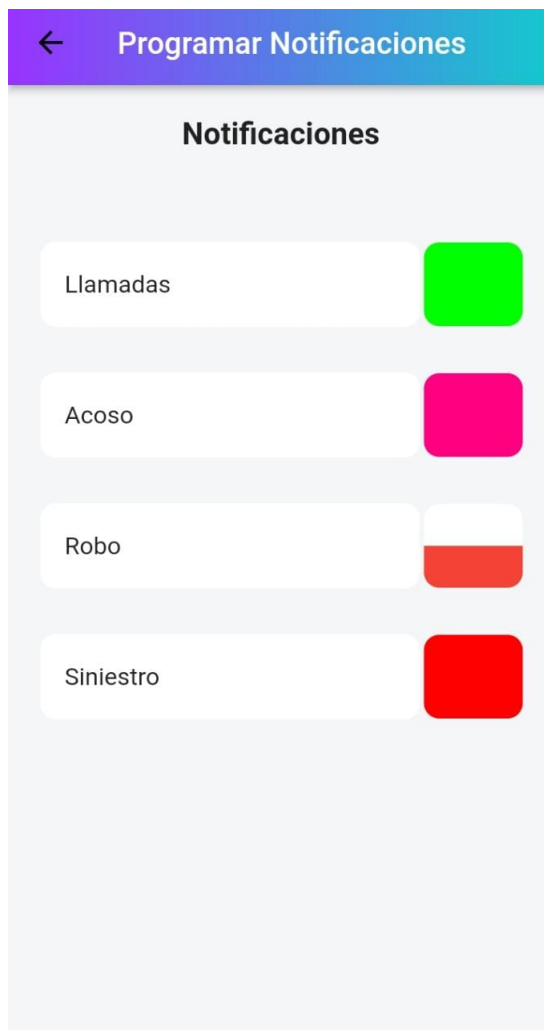


Figura 62: Vista Notificaciones Navi
Fuente: [22]

La figura 62 es la vista que tiene el usuario disponible para indicarle a Navi qué color debe emitir al recibir una notificación o al emitir alguno de los tres tipos de reportes que se pueden realizar con el dispositivo.

The image shows a mobile application screen titled 'Nuevo Dispositivo' (New Device) with a blue header bar containing a back arrow. The form is divided into three sections: 'Elige tipo de dispositivo' (Choose device type) with a dropdown menu labeled 'Nombre del dispositivo'; 'Ingresa serial' (Enter serial) with a text input field labeled 'Digita aquí el número de serie' and a character count '0/12'; and 'Fecha de compra' (Purchase date) with a date picker labeled 'DD/MM/AAAA' and a character count '0/10'. A large blue 'Registrar' (Register) button is at the bottom.

Figura 63: Vista Registrar Dispositivo
Fuente: [22]

En la figura 63 el usuario tiene disponible un apartado para realizar el registro de alguno de los dispositivos adquiridos; Navi, Bigo, Sensor. Una vez registrado el dispositivo este aparecerá en la vista de la figura 57.

← Perfil

Editar información personal

Sexo del usuario

Seleccione sexo ↓

Ingresa nombre y apellido *

EJ: Maria Paula

Ingresa ciudad *

EJ: Bogot...

Guardar

Figura 64: Vista Editar Perfil

Fuente: [22]

La vista de la figura 64 permite que el usuario modifique toda su información principal, la cual aparece en la vista de la figura 58.

← Registrar Vehículo

Tipo de vehículo *

Selecciona vehículo ↓

Ingresa serial y/o placas del vehículo *

EJ: TM5F34 0/12

Ingresa marca o modelo *

EJ: Giant Talon... 0/12

Fecha de compra *

DD/MM/AAAA 0/10

+ Adjuntar imagen

Registrar

Figura 65: Vista Registrar Vehículo
Fuente: [22]

Para que el usuario pueda registrar el vehículo, cuya información se despliega en la vista de la figura 58, se debe hacer en la vista que se aprecia en la figura 65 donde se debe introducir el tipo de vehículo; patineta, bicicleta, motocicleta, así como otra información específica que puede ser su número de matrícula o su número serial, marca, modelo y fecha de compra.

6.5. Navi

Como resultado de las mejoras realizadas al dispositivo Bigo se tiene Navi, que no solo es capaz de funcionar de manera independiente sino que también funciona como complemento para Bigo ya que también es posible que Navi establezca una conexión bluetooth con Bigo para generar las indicaciones de giro sin que este se encuentre conectado ni a la aplicación móvil ni al sensor.



Figura 66: Funciones Navi

Fuente: [27]

Las funciones principales de Navi se enmarcan en la figura 66 y el diseño final del dispositivo se puede apreciar mejor en las figuras 67, 68, 69 y 70



Figura 67: Vista Lateral Navi
Fuente: [26]



Figura 68: Carcasa Navi
Fuente: [26]



Figura 69: Vista Posterior Navi
Fuente: [26]



Figura 70: Vista Cúpula Navi
Fuente: [26]

6.6. Manual de Usuario

En cumplimiento de los objetivos, se realiza la creación de un manual de usuario para el dispositivo desarrollado; Navi. El diseño final del manual está compuesto de 7 partes:



Figura 71: Portada Manual de Usuario

Fuente: [26]

La figura 71 es la porta del manual de usuario donde se aprecia una vista lateral de Navi.

Contenido caja

- Dispositivo
- Cable USB C
- Adaptador de corriente
- Manual de usuario

Obtener más información

Para ver el el manual en español, ingrese a la pagina de Bigo <https://www.bigosafe.com/>
Selecciones inicio, productos, navi y descripciones



Tambien puedes escanear para agilizar el proceso

2

Figura 72: Contenido Caja

Fuente: [26]

La figura 72 es un resumen del contenido que el usuario podrá encontrar dentro de la caja del dispositivo.

Diseño del dispositivo

Navi es un asistente luminoso controlado por el celular para cualquier usuario que le guste rodar por la ciudad. Este asistente te guiará en la carretera prometiendo ser tu compañero, darte mayor visibilidad y con ellos mayor protección; Para que llegues seguro a casa juntos a las personas que amas

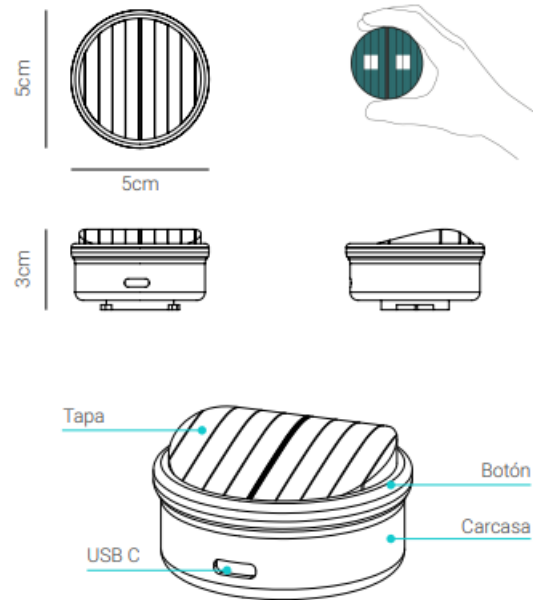


Figura 73: Diseño Dispositivo

Fuente: [26]

Las dimensiones del dispositivo, así como algunas de sus vistas principales como la vista superior y frontal se están registradas en la parte 3 del manual que se aprecia en la figura 73.

Inicio del dispositivo

1. Mantenga oprimido el botón durante unos segundos.
2. Elace el dispositivo con su telefono vía Bluetooth
3. Mantenga pulsado el botón durante unos segundos hasta que se encienda las luces blancas.
4. Desde su teléfono celular encienda su Bluetooth y abra la aplicación Bi Go!. Si no tiene instalada la aplicación descarguela aquí: <https://bit.ly/37a4ujX>
5. Enlace su dispositivo NAVI con su teléfono vía Bluetooth en la lista de dispositivos que le muestra la aplicación.
6. Puede activar las funciones de reporte oprimiendo y soltando el botón de NAVI y colocando la ruta para llegar a su destino, disfrute un viaje seguro con NAVI.



4

Figura 74: Primero Pasos

Fuente: [26]

Los pasos iniciales que debe seguir el usuario para realizar la conexión entre el dispositivo y la aplicación móvil se encuentran enlistados en la parte 4 del manual que hace referencia a la figura 74.



Figura 75: Controles de Navi
Fuente: [26]

En la figura 75, que compone la parte 5 del manual, el usuario tiene un resumen de los modos que Navi es capaz de emitir para las direccionales, así como uno de los tipos de reportes que pueden hacerse desde el dispositivo.

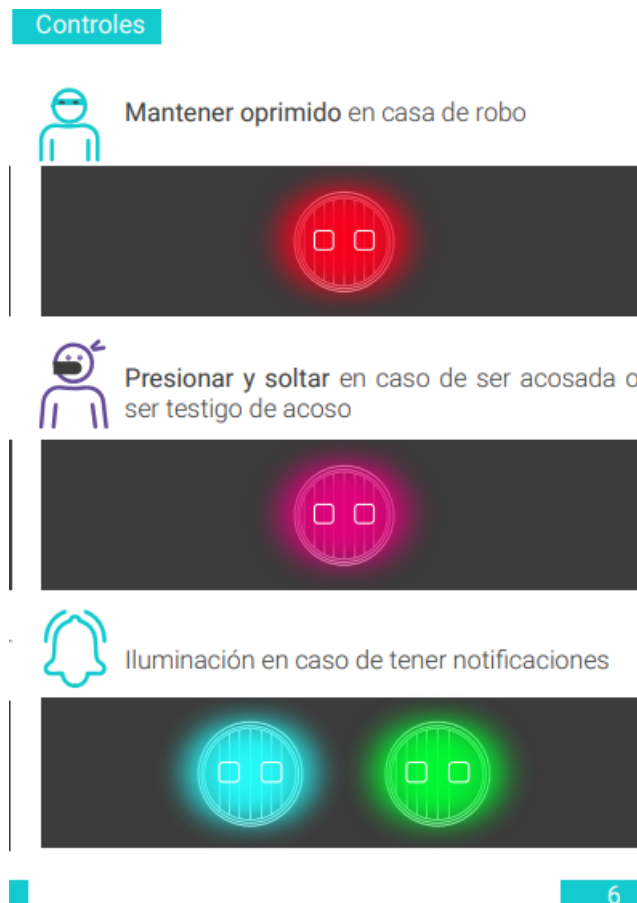


Figura 76: Controles de Navi
Fuente: [26]

La figura 76 muestra al usuario los demás tipos de reporte que son posible realizar desde el dispositivo en comunicación con la aplicación móvil así como los colores que emite Navi al recibir una notificación.

Seguridad

Lea toda la información de seguridad antes de utilizar el dispositivo a fin de asegurar el uso correcto y la seguridad de nuestro usuarios



Advertencia

No usar el botón cuando **no** sea un caso de emergencia

No exponga el dispositivo a impactos o daños físicos

Evite el contacto directo con el agua

No permita que los niños o animales mastiquen o laman el dispositivo.

Figura 77: Mensaje de Seguridad

Fuente: [26]

La figura 77 es la parte final del manual y es un mensaje de advertencia para el usuario para mantener un buen uso con Navi.

7. Conclusiones

El objetivo primordial del presente trabajo de grado era poder llegar a desarrollar *software* o *hardware* que pudiese mejorar la experiencia que tenían los usuarios con el dispositivo Bigo, esto requirió de la aplicación de metodología ágiles como Scrum para el trabajo en equipo y el cumplimiento de las tareas y requerimientos asignados por la empresa. Para poder complementar la tecnología de Bigo y atendiendo a una convocatoria organizada por diferentes organismos, entre ellos la Alcaldía Mayor de Bogotá y la Secretaría de Desarrollo Económico, el practicante estuvo a cargo del desarrollo de Navi, dispositivo que ahora hace parte de la familia de CiNNNoV, que junto al Bigo cumplen con su propósito de mantener más visible a los diferentes actores viales en la ciudad de Bogotá y diferentes carreteras del país. Con todos estos desarrollos se permite que una mayor cantidad de usuarios tengan acceso a tecnología de menor costo que pueda ayudar a prevenir accidentes y pérdidas humanas, además favorece al avance del país en su competencia con otras ciudades inteligentes, que están a la vanguardia de la implementación de tecnologías de la industria 4.0. Además, el practicante logró demostrar ser capaz de cumplir con tiempos de entrega, importante para mantener el correcto funcionamiento de una empresa, así como poder estar completamente a cargo del desarrollo de proyectos tecnológicos.

- Tras los resultados obtenidos durante todo el proceso de prácticas por parte del practicante, quedó evidenciado la importancia de la realización de una etapa introductoria donde el aprendiz sea capaz de tener todos los conocimientos necesarios y las bases más importantes, no solo en los aspectos técnicos sino también en la estructura de la empresa, en el cumplimiento de su objetivo a través del tiempo y en el área tecnológica en la cual está enfocada. Así mismo saber que se cumple de una labor social por el campo de aplicación a la cual está destinada la tecnología desarrollada y lo importante que esto representa para el país, tanto en el desarrollo social como en el desarrollo tecnológico.
- Todo desarrollo tecnológico destinado a mantener más visibles a los diversos actores viales en las diferentes carreteras del país es importante, sin embargo Bigo representa una gran ventaja con respecto a la diferente competencia que existe en el mercado pues no se limita a una superficie en específico, que frecuentemente es la espalda del usuario; sino que este puede montarse sobre cualquier tipo de superficie como lo son también, maletas, maletines, bolsos, etc. Esto hace que Bigo haya tenido un gran impacto sobre los productos de seguridad vial al ser capaz de acoplarse sobre cualquier superficie, así como la posibilidad de incorporarse dentro de maletines con compartimientos plásticos que lo protegen del agua, eliminando esta limitante de los diferentes productos de la competencia.
- El desarrollo de una aplicación móvil para complementar el dispositivo Bigo nace de la necesidad de los usuarios de poder automatizar la generación de indicaciones sin tener que limitarse exclusivamente al uso de un sensor de movimiento que va acoplado sobre el casco del actor vial y que obliga al usuario que lo porta a realizar movimientos con la cabeza que pueden resultar inapropiados bajo ciertas condiciones. Otra de las necesidades es la de poder crear una comunidad alrededor de los usuarios Bigo de tal forma que se pueda compartir información en tiempo real con todos los usuarios como puede ser el uso de Navi para la generación de

alertas sobre siniestros, robos o abusos presentados o vividos por el usuario. Toda esta información puede ayudar a crear mapas de calor de la ciudad de Bogotá que permite mantener actualizados a los usuarios sobre la situación de seguridad por las zonas donde se encuentran transitando.

- El desarrollo de Navi demostró la importancia y la necesidad de acercar la tecnología a una mayor cantidad de actores viales, una tecnología que es más económica que Bigo, pero que también sirve como complemento para todos los usuarios pues Bigo tiene ciertas limitaciones que se implementaron en Navi, como la posibilidad de realizar denuncias en tiempo real gracias a su conexión con la aplicación móvil y los servicios de Cloud Computing. Así como el hacer más visible a todos los usuarios no solo en la parte posterior de este sino también en la parte frontal gracias al uso de leds de tipo NeoPixel, que tienen la suficiente potencia para hacerse visible a cualquier hora del día.
- Se pudo hacer una entrega parcial de la aplicación móvil en su rediseño y nuevas funciones implementadas para poder crear comunidad y su integración con las diferentes funciones que se desarrollaron dentro de Navi. Así como la entrega completa de Navi, que en su desarrollo final cumplió con las expectativas de la empresa, así como las condiciones de la convocatoria RETO CANDADO BICI, donde Navi fue seleccionado entre los ganadores como mejor desarrollo tecnológico para la seguridad de biciusuarios para la ciudad de Bogotá. Dentro de todos estos desarrollos se pudieron aplicar, no solo los conocimientos en desarrollo de algoritmos y diseño electrónico durante la carrera sino también todos las bases en desarrollo móvil en la ruta de aprendizaje inicial del practicante.

Referencias

- [1] Bogotá: aumentaron los ciclistas y su accidentalidad — El Nuevo Siglo. <https://www.elnuevosiglo.com.co/articulos/09-2020-el-aumento-en-los-accidentes-de-ciclistas-responde-al-aumento-en-el-numero-de>.
- [2] Radiografía a uso de bicicleta en Bogotá — El Nuevo Siglo. <https://www.elnuevosiglo.com.co/articulos/06-2018-radiografia-uso-de-bicicleta-en-bogota>.
- [3] What is the Internet of Things, and how does it work? <https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/>.
- [4] Home - IBM Developer. <https://developer.ibm.com/>.
- [5] Oswaldo Rojas Lazo, Luis Rojas: *Diseño asistido por computador*. 2006.
- [6] Prayoga, Regawa Rama, Alifia Syalsabila, Ghifari Munawar y Rahil Jumiyan: *Diseño Electrónico y su Evolución en el País*. Colombia: Ciencia y Tecnología. 4(1), 1986.
- [7] What is Frameworks? [Definition] Types of Frameworks. <https://hackr.io/blog/what-is-frameworks>.
- [8] Benbelkacem, Samir, Nadia Zenati-Henda, Djamel Aouam, Yousra Izountar y Samir Otmane: *MVC-3DC: Software architecture model for designing collaborative augmented reality and virtual reality systems*. Journal of King Saud University - Computer and Information Sciences, 32:433–446, Mayo 2020, ISSN 1319-1578.
- [9] León, Pedro J Ponce De: *UD 2 CONCEPTOS BÁSICOS de la programación orientada a objetos Versión 0.1 (Curso 11/12)*.
- [10] Lesionados, Fallecidos Y, Siniestros Viales, Serie Nacional, Luis Felipe Lota, Jasson Cruz, Villamil Director, Especializado Carlos y Alberto Hernández: *BOLETÍN ESTADÍSTICO COLOMBIA*. <https://ansv.gov.co/>.
- [11] ¿Qué son las ciudades inteligentes? — En Profundidad — teleSUR. <https://www.telesurtv.net/telesuragenda/Que-son-las-ciudades-inteligentes-20161027-0021.html>.
- [12] Sommerville, Ian: *INGENIERÍA DEL SOFTWARE*. Universidad de Alicante, 7:5–12, 2006.
- [13] Martínez, Raúl Noriega: *El proceso de Desarrollo de Software*. 2:5–25, 2017.
- [14] Delía, Lisandro Nahuel: *Desarrollo de Aplicaciones Móviles Multiplataforma*. B.S. thesis, 2017.
- [15] Medina, Leiner Jose Barrios: *APOYO A PROYECTOS DE INGENIERÍA TI REQUERIDOS POR LA ACADEMIA INGLÉS PARA TODOS-IPT*. B.S. thesis, 2022.
- [16] A, Cárdenas Rodríguez: *Lumiback, sistema de códigos visuales para ciclistas*. B.S. thesis, 2019.
- [17] Guamán Taipe, D. A., Soto Bermello K. B.: *Implementación de un sistema de navegación para ciclistas, que le permita llegar a una dirección*. B.S. thesis, 2021.
- [18] Qué es Github y qué es lo que le ofrece a los desarrolladores. <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>.

- [19] *Comandos de GitHub — Lista de comandos básicos y avanzados de GitHub.* <https://es.education-wiki.com/9938590-github-commands>.
- [20] *Capa gratuita de AWS — Cloud computing gratis — AWS.* https://aws.amazon.com/es/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Categories=categories%23storage&trk=cd137bfd-7e67-4be5-a7e9-6f0ca650691f&sc_channel=ps&sc_campaign=acquisition&sc_medium=ACQ-P|PS-BI|Brand|Desktop|SU|Storage|S3|LATAM|EN|Text&ef_id=edaac96a5d071f156d27abf51568bad9:G:s&s_kwcid=AL!4422!10!72018164920093!72018610017277&awsf.Free%20Tier%20Types=*all.
- [21] *AWS — Gestión de identidades y autenticación de usuario en la nube.* <https://aws.amazon.com/es/cognito/>.
- [22] CiNNoV S.A.S, Alexander Nieves: *BiGo!*, Noviembre 2021.
- [23] *Encuestas de Movilidad — Observatorio de Movilidad.* https://observatorio.movilidadbogota.gov.co/encuestas_de_movilidad.
- [24] *Reto — Bici-candados para todos y todas.* <https://bogotechabierta.com/post/4i8IeTi81dNwegSZZl1c>.
- [25] *Bici-candados para todos y todas.* <https://colombiafintech.co/lineaDeTiempo/articulo/bici-candados-para-todos-convocatoria>.
- [26] Nanyomi Parra, Alexander Nieves: *Navi*, Noviembre 2021.
- [27] *Navi – Bigo Safe.* <https://www.bigosafe.com/producto/navi/>.