



**Desarrollo de una aplicación Android para anunciar la presencia de sobrevivientes
mediante un botón de pánico**

Andrés Felipe Cifuentes Mendieta

11161717424

Sergio Giovanni Silva Arias

11161713516

Universidad Antonio Nariño

Programa Ingeniería de Sistemas y Computación

Facultad de Ingeniería de Sistemas

Bogotá, Colombia

2021

**Desarrollo de una aplicación Android para anunciar la presencia de sobrevivientes
mediante un botón de pánico**

Andrés Felipe Cifuentes Mendieta

Sergio Giovanni Silva Arias

Proyecto de grado presentado como requisito parcial para optar al título de:

Ingeniero de sistemas y computación

Director (a):

Doctora en Ingeniería, María del Pilar Salamanca

Asesor (a) metodológico (a):

Rosalba Cruz Cepeda

Línea de Investigación:

Redes y seguridad

Grupo de Investigación:

LACSER

Universidad Antonio Nariño

Programa Ingeniería de Sistemas y Computación

Facultad de Ingeniería de Sistemas

Bogotá, Colombia

2021

ACEPTACIÓN

El trabajo de grado titulado
_____, cumple con
los requisitos para optar
Al título de _____.

Firma del Tutor

Firma Jurado

Firma Jurado

Ciudad, Día Mes Año.

Contenido

	<u>Pág.</u>
Agradecimientos	9
Resumen	10
Abstract	11
Introducción.....	12
1. Planteamiento del problema	14
1.1 Descripción del problema	14
1.2 Formulación del problema	16
1.3 Justificación	17
1.4 Objetivos	18
<i>1.4.1 Objetivo general</i>	<i>18</i>
<i>1.4.2 Objetivos específicos</i>	<i>18</i>
1.5 Alcance y limitaciones del proyecto	19
<i>1.5.1 Alcance</i>	<i>19</i>
<i>1.5.2 Limitaciones</i>	<i>21</i>
2. Marco de referencia	22
2.1 Marco teórico	22
2.1.1 Desastres naturales	22
2.1.2 Beacon.....	23
2.1.3 Bluetooth.....	23
2.1.4 Descubrimiento de servicios	24
2.1.5 WIFI Direct.....	24
2.1.6 Android Studio.....	24
2.1.7 Kotlin	25
2.1.8 Firebase.....	25
2.1.9 Bases de datos no relacionales	25
2.1.10 Scrum.....	26
2.2 Antecedentes	28
2.2.1 Kaifa.....	28
2.2.2 AlertaMovil	28
2.2.3 Botón de pánico SOS	29
2.2.4 Comparaciones.....	29

2.3	Marco legal.....	30
2.3.1	Ley de derechos de autor	30
2.3.2	Ley de protección de datos.....	31
3.	Aspectos metodológicos	32
3.1	<i>Sprint 1</i>	32
3.2	<i>Sprint 2</i>	34
3.3	<i>Sprint 3</i>	36
4.	Desarrollo del proyecto.....	38
4.1	Descripción de la aplicación.	38
4.2	Fase de iniciación.....	38
4.3	Planeación y estimación	39
4.3.1	Roles del sistema	39
4.3.2	Arquitectura y herramientas.....	39
4.3.3	Historias de usuario.....	40
4.4	Implementación y desarrollo.....	40
4.4.1	<i>Sprint 1: módulo de registro de usuarios</i>	41
4.4.1.1	Pruebas.....	51
4.4.2	<i>Sprint 2: módulo de envío de alertas por WIFI Direct</i>	54
4.4.2.1	Pruebas.....	65
4.4.3	<i>Sprint 3: módulo de envío de alertas por Bluetooth e integración final</i>	67
4.4.3.1	Pruebas.....	74
5.	Análisis y resultados.....	77
5.4.1	Prueba de registro	82
5.4.2	Solicitar ayuda estando registrado en la aplicación.....	85
5.5	Pruebas de compatibilidad	93
6.	Conclusiones	95
	Referencias Bibliográficas	99

Tabla de figuras

	<u>Pág.</u>
Figura 1 <i>Tabla base de datos Firebase</i>	39
Figura 2 <i>Diagrama caso de uso: Iniciar sesión</i>	44
Figura 3 <i>Mockup pantalla de inicio de sesión</i>	45
Figura 4 <i>Mockup pantalla de registro</i>	45
Figura 5 <i>Pantalla de inicio de sesión</i>	47
Figura 6 <i>Pantalla de registro (1)</i>	47
Figura 7 <i>Pantalla de registro (2)</i>	48
Figura 8 <i>Pantalla de registro (fecha de nacimiento)</i>	48
Figura 9 <i>Diagrama de clases</i>	49
Figura 10 <i>Base de datos en la nube Firebase</i>	51
Figura 11 <i>Diagrama caso de uso: Cerrar sesión</i>	57
Figura 12 <i>Diagrama caso de uso: Envío de alertas por WIFI Direct</i>	58
Figura 13 <i>Diagrama caso de uso Recargar pantalla WIFI Direct</i>	58
Figura 14 <i>Diagrama de caso de uso Encender alarma sonora</i>	59
Figura 15 <i>Mockup pantalla principal</i>	60
Figura 16 <i>Mockup de dispositivos encontrados</i>	60
Figura 17 <i>Mockup pantalla de solicitar ayuda</i>	61
Figura 18 <i>Pantalla principal</i>	62
Figura 19 <i>Pantalla dispositivos encontrados</i>	62
Figura 20 <i>Diagrama de caso de uso Enviar alertas por Bluetooth</i>	70
Figura 21 <i>Diagrama de caso de uso Permanecer en la tecnología</i>	71
Figura 22 <i>Diagrama de caso de uso Cambiar de tecnología</i>	71
Figura 23 <i>Pantalla de dispositivos encontrados por Bluetooth</i>	72
Figura 24 <i>Pantalla de dispositivos encontrados por WIFI con los dos nuevos botones</i>	72
Figura 25 <i>Fragmento de código del módulo de WIFI Direct</i>	80
Figura 26 <i>Fragmento del código desarrollado para la interfaz de WIFI Direct</i>	80
Figura 27 <i>Fragmento de código del módulo de Bluetooth</i>	81
Figura 28 <i>Fragmento de código desarrollado para la interfaz Bluetooth</i>	81
Figura 29 <i>Pantalla de registro con datos faltantes</i>	83
Figura 30 <i>Mensaje que devuelve la aplicación si el usuario ya se ha registrado anteriormente.</i>	83
Figura 31 <i>Mensaje de registro exitoso</i>	84
Figura 32 <i>Pantalla de ingreso a la aplicación</i>	84
Figura 33 <i>Base de datos con la información de los usuarios registrados</i>	85
Figura 34 <i>Pantalla de dispositivos encontrados por WIFI Direct (Xiaomi Mi A3)</i>	86
Figura 35 <i>Información del dispositivo (Xiaomi Mi A3)</i>	86
Figura 36 <i>Resultados de dispositivos encontrados por WIFI Direct (Xiaomi Note 8)</i>	87

Figura 37 Información del dispositivo (Xiaomi Note 8)	87
Figura 38 Mapa de la ubicación de las pruebas.....	88
Figura 39 Diagrama de distancia entre los dispositivos en el parque Tunal	88
Figura 40 Pantalla de dispositivos encontrados por Bluetooth (Xiaomi Mi A3).....	89
Figura 41 Pantalla de dispositivos encontrados por Bluetooth (Xiaomi Note 8).....	89
Figura 42 Distancia de los dispositivos uno en un edificio y otro a campo abierto.....	90
Figura 43 Resultados de dispositivos encontrados por WIFI Direct (Samsung J5 Prime) dispositivo en el edificio.....	91
Figura 44 Resultados de dispositivos encontrados por WIFI Direct (Xiaomi Note 8) dispositivo en la calle	91
Figura 45 Resultados de dispositivos encontrados por Bluetooth (Samsung J5 prime) dispositivo en el edificio.....	92
Figura 46 Resultados de dispositivos encontrados por Bluetooth (Xiaomi Note 8) dispositivo en la calle.....	92

Tabla de tablas

	<u>Pág.</u>
Tabla 1 <i>Clasificación de desastres</i>	22
Tabla 2 <i>Comparaciones de aplicaciones similares</i>	29
Tabla 3 <i>Historia de usuario 1: Registrar usuario</i>	42
Tabla 4 <i>Historia de usuario 2: Inicio de sesión</i>	43
Tabla 5 <i>Caso de prueba Registrar usuario</i>	52
Tabla 6 <i>Caso de prueba Iniciar Sesión</i>	53
Tabla 7 <i>Historia de usuario 3: Enviar Alertas por WIFI Direct</i>	54
Tabla 8 <i>Historia de usuario 4: Recargar pantalla WIFI Direct</i>	55
Tabla 9 <i>Historia de usuario 5: Cerrar Sesión</i>	56
Tabla 10 <i>Historia de usuario 6: Encender Alarma Sonora</i>	57
Tabla 11 <i>Caso de prueba Envío de alerta por WIFI Direct</i>	65
Tabla 12 <i>Caso de prueba Cerrar sesión</i>	66
Tabla 13 <i>Caso de prueba Recargar pantalla WIFI Direct</i>	67
Tabla 14 <i>Historia de usuario 7: Enviar alertas por Bluetooth</i>	68
Tabla 15 <i>Historia de usuario 8: Permanecer en la tecnología Bluetooth o WIFI Direct</i>	69
Tabla 16 <i>Historia de usuario 9: Cambiar de tecnología</i>	70
Tabla 17 <i>Caso de prueba funcional (Envío de alerta por Bluetooth)</i>	75
Tabla 18 <i>Caso de prueba funcional (Cambio de tecnología)</i>	76

Agradecimientos

En este pequeño párrafo queremos dar nuestro profundo agradecimiento a la profesora María del Pilar Salamanca porque fue nuestra directora de tesis, nos guió, orientó, ayudó para la investigación, aportándonos conocimiento, y opiniones propias y de profesionales, teniendo en cuenta que ella llevó la gestión de las reuniones con aquellas personas expertas en situaciones de emergencia.

Por otra parte, queremos agradecer a la profesora Rosalba Cruz por ayudarnos y orientarnos con los temas de redacción y presentación de documentos del proyecto. Al profesor Jhonatan Rico por acompañarnos en los inicios de este proyecto junto con su experiencia y dedicación. Al profesor Wilson Forero por asesorarnos con temas de desarrollo en Android lo cual nos facilitó terminar la aplicación.

Por último, queremos agradecer a todos los profesores y directivos que hicieron parte de este crecimiento profesional, sin su dedicación y esfuerzo no habríamos llegado a este punto de nuestra carrera.

Resumen

Los desastres naturales se han ido incrementando al pasar de los años alrededor de todo el mundo. El medio ambiente se ha visto afectado por las acciones del hombre, que ha explotado los recursos naturales y deteriorado la naturaleza sin control. Cuando ocurre un evento catastrófico, y dependiendo del grado de preparación de la población en donde el hecho sucede, es posible que el evento tenga consecuencias nefastas para una parte de los habitantes de la región afectada.

Con el fin de ayudar a las personas y a los miembros de los equipos de rescate cuando ocurre un desastre, se propuso el desarrollo de una aplicación móvil que apoye el proceso de localización de las personas que se encuentren en zonas de difícil acceso para los socorristas. Para la realización de este proyecto se usó Scrum que es una metodología ágil, lo cual ayudó en el desarrollo de la aplicación teniendo en cuenta todos los factores necesarios para desplegarla a tiempo.

Palabras clave: Desastre natural, aplicación, botón de pánico, Bluetooth, WIFI Direct.

Abstract

Natural disasters have been increasing over the years around the world. The environment has been affected by the actions of man, who has exploited natural resources and deteriorated nature without control. When a catastrophic event occurs, and depending on the degree of preparedness of the population where the event occurs, it is possible that the event has dire consequences for a part of the inhabitants of the affected region.

In order to help people and members of rescue teams when a disaster occurs, it was proposed to develop a mobile application that supports the process of locating people who are in areas that are difficult to access for rescuers. Scrum was used to carry out this project, which is an agile methodology which helped in the development of the application, taking into account all the factors that were needed to deploy it on time.

Keywords: Natural disaster, application, panic button, Bluetooth, WIFI Direct.

Introducción

Los desastres naturales causan daños y perjuicios a todos los seres vivos. Algunos ejemplos de estos desastres son: terremotos, maremotos, erupciones volcánicas, entre otros. Últimamente se ha notado el aumento de estas catástrofes alrededor del mundo, por lo cual muchas más poblaciones están siendo afectadas (Capacci & Mangano, 2015).

En el caso particular de los sismos, en aquellos países en donde no hay normas estrictas que exigen sismo resistencia en las construcciones, el impacto de un terremoto puede ser muy grave en la población, ocasionando el desplome de las estructuras y trayendo como consecuencia que las personas queden atrapadas entre los escombros. En esa situación, los sobrevivientes intentan pedir auxilio o comunicarse con las líneas de emergencia o con sus familiares por medio de internet. Muchas veces este proceso es imposible ya que por causa de la catástrofe, la infraestructura de las redes de comunicaciones puede colapsar o congestionarse debido a la alta demanda de servicios de la red por parte de los usuarios. Esta situación se observa no solo en el caso de los terremotos, sino en cualquier tipo de catástrofe que afecte a una región.

Con el fin de aumentar la probabilidad de encontrar sobrevivientes después de un desastre se propuso diseñar una aplicación, que, sin usar las redes de comunicación móvil, permite enviar una señal de emergencia que notifique a los socorristas u otros sobrevivientes que tengan activa la aplicación en su celular, acerca de la presencia de una persona que necesita ayuda. Para que los dispositivos anuncien su presencia por medio de esta aplicación, se utilizaron los adaptadores Bluetooth y WIFI que vienen embebidos en el teléfono inteligente.

Se optó por usar Scrum para el desarrollo de este proyecto teniendo en cuenta que es una metodología ágil, que facilita la adaptación de cambios, la comunicación entre todo el equipo de trabajo y a tener un control de tiempo en la realización de la aplicación.

Este documento está estructurado de la siguiente manera: en el capítulo 1 se encuentra una descripción específica acerca del problema principal, enfatizando la dificultad que se presenta al momento de localizar supervivientes después de un desastre natural; además de esto, presenta los objetivos del trabajo de grado. En el capítulo 2 se describe el marco en el cual se desarrolló este proyecto, estipulando las leyes que se tuvieron en cuenta y un estado del arte. El capítulo 3 explica la metodología con la cual se desarrolló este proyecto, describiendo las características importantes, y las actividades por etapa que se realizaron. En el capítulo 4, se describe la etapa de desarrollo de acuerdo con los *sprints* planeados, los casos de uso, la programación del código y la documentación de pruebas. En el capítulo 5 se hicieron todas las pruebas de campo de la versión final de la aplicación. En el capítulo 6 se presentan las conclusiones y recomendaciones en el desarrollo de este proyecto.

1. Planteamiento del problema

1.1 Descripción del problema

Los desastres naturales son todos los eventos causados por la naturaleza o procesos naturales de la tierra, como por ejemplo: terremotos, tsunamis, maremotos, tornados, inundaciones, entre otros. Indudablemente, el impacto más grave de este tipo de eventos catastróficos es la pérdida de vidas. Por ejemplo, en el terremoto de Turquía y Grecia, ocurrido el 30 de octubre de 2020, fallecieron aproximadamente 69 personas y más de 900 quedaron heridas (RTVE.es, 2020). Otro ejemplo de este desastre es el terremoto en Haití el 19 de agosto del 2021, donde la magnitud fue de 7.2 en la escala de Richter, dejando 2000 muertos aproximadamente y más de 9900 heridos (Naciones Unidas, 2021). Otro tipo de desastre son las inundaciones. Recientemente ocurrió una inundación en Colombia en donde se vieron afectadas alrededor de 600 personas incluyendo niños y campesinos (Reliefweb, 2021).

La acción de los equipos de rescate después de un evento catastrófico es fundamental para rescatar sobrevivientes. Sin embargo, el éxito de la intervención de estos equipos depende de la rapidez de las acciones de rescate. Es ampliamente conocido que la probabilidad de rescatar personas con vida se reduce significativamente después de las primeras 72 horas, conocidas como “72 horas doradas” (El País, 2010), y es por ello que cualquier esfuerzo encaminado en agilizar las labores de rescate, puede significar un incremento en el número de vidas salvadas.

Cuando ocurre un desastre, la información es un elemento muy valioso para ofrecer una respuesta rápida y adecuada (Arroyo et al., 2009). Sin embargo, después de la ocurrencia de un desastre, es posible que la infraestructura de telecomunicaciones resulte afectada y se imposibilite la comunicación de los sobrevivientes con las autoridades y rescatistas o con sus seres queridos.

Adicionalmente, la infraestructura de los edificios o de las viviendas también puede verse afectada provocando que personas queden entre los escombros. En estas circunstancias, las personas al encontrarse atrapadas en un espacio reducido o en medio de un área inundada, por ejemplo, no tienen forma de pedir ayuda de una manera ágil y sencilla, lo que dificulta su rescate. Inclusive, si tuvieran un teléfono inteligente a su alcance y con suficiente energía en la batería, no se podría utilizar debido a las fallas en la infraestructura de telecomunicaciones.

Teniendo en consideración el potencial que ofrecen los teléfonos inteligentes para enviar y recibir información, incluso cuando no hay infraestructura de comunicaciones disponible, se propuso el desarrollo de una aplicación móvil que los sobrevivientes puedan activar después de la catástrofe, de tal manera que notifique a los dispositivos cercanos su presencia y sus datos personales. La persona podrá ingresar a la aplicación, accionar el botón de pánico para enviar la alarma y el teléfono emitirá señales que pueden ser captadas por otros usuarios cuyos teléfonos tengan activada la misma aplicación. Las tecnologías usadas para enviar y recibir la señal son Bluetooth y WIFI Direct, y tienen un rango aproximado de cien metros en campo abierto. Debido a que es un prototipo para un proyecto de investigación, el propósito es hacer uso de estas dos tecnologías para saber cuál es mejor para desempeñar esta función. Si el usuario lo desea, la aplicación también puede emitir una alarma sonora que sirva como un medio adicional para facilitar su localización.

El desarrollo se realizó para el sistema operativo Android ya que cuenta con la mayor participación en el mercado (74% de los usuarios) (Shum, 2020), y también debido a limitaciones de presupuesto en el proyecto de investigación al cual está adscrito este trabajo de grado.

La aplicación se diseñó para que pueda ser utilizada por cualquier persona, y puede ser de gran utilidad, especialmente para los cuerpos de rescate y autoridades que trabajan en escenarios de búsqueda y rescate de personas. Es importante señalar que, si bien esta aplicación se diseñó para ser utilizada en un escenario posterior a una catástrofe, también tiene usos potenciales a futuro en otros contextos, como el de seguridad ciudadana.

1.2 Formulación del problema

Como consecuencia de los desastres naturales, es posible que los sobrevivientes queden atrapados entre los escombros o en zonas aisladas y se les dificulta salir de allí; si los equipos de rescate se tardan en encontrarlos, numerosas vidas se pueden perder.

Dado que la comunicación es una herramienta fundamental para agilizar los procesos de búsqueda y rescate, se propuso el desarrollo de una aplicación móvil que ayude a las víctimas para que puedan anunciar su presencia en caso de estar atrapadas o necesitar ayuda, facilitando a los equipos de rescate su labor de encontrar a los sobrevivientes oportunamente. La aplicación está planteada para que funcione a partir de que la persona la active y estará en operación hasta que la batería del móvil se termine o la persona la desactive. Los socorristas y primeros respondientes, a su vez, podrán utilizar esta herramienta buscando las señales emitidas por los teléfonos inteligentes de los sobrevivientes.

1.3 Justificación

Con esta aplicación se ven beneficiados dos grupos: las personas afectadas por las catástrofes, ya que es una herramienta que les ayuda a anunciarse frente a los equipos de rescate; por otro lado, los equipos de rescate, porque pueden hacer su labor con mayor eficacia y eficiencia.

El impacto tecnológico se evidencia en el enfoque humanitario de la aplicación, que permite al teléfono inteligente emitir señales de emergencia utilizando las tecnologías Bluetooth y WIFI Direct para localizar a personas que necesiten ayuda. WIFI Direct tiene un rango de 30 a 200 metros (Yubal, 2021). La velocidad de transferencia de archivos es un punto a favor del WIFI Direct ya que cuenta con una velocidad de hasta 250 Mb/s (Gutierrez, 2016). Las bandas de frecuencia que utiliza esta tecnología son las de 2.4 Ghz y de 5,0 Ghz (Gutierrez, 2016). Por otra parte, Bluetooth presenta un rango de 200 metros para su versión 5.0 y esta tecnología trabaja en la banda de 2.4 Ghz (Kaisar, 2021). Adicionalmente tiene una velocidad de transferencia de hasta 50 Mb/s (Yubal, 2021).

En el ámbito profesional, los conocimientos adquiridos a través de los diferentes semestres se pusieron en práctica en este proyecto de grado, lo cual fue un gran reto, pues se utilizan dos tecnologías: Bluetooth y WIFI Direct, en las que este equipo de trabajo no tenía experiencia previa. Adicionalmente, se desarrolló esta aplicación con un enfoque humanitario, contribuyendo al cumplimiento de los objetivos del proyecto de investigación *Plataforma para apoyar el proceso de rescate de sobrevivientes de un desastre a través de teléfonos móviles*, al cual se encuentra adscrito.

1.4 Objetivos

1.4.1 *Objetivo general*

Desarrollar una aplicación para sistema operativo Android que apoye la localización y el rescate de personas afectadas por catástrofes naturales, a través de un botón de pánico.

1.4.2 *Objetivos específicos*

1. Determinar los requerimientos de la aplicación Android que facilita la localización de personas, a partir de las necesidades manifestadas por el proyecto de investigación.
2. Diseñar la estructura de la base de datos, historias de usuario y casos de uso mediante lenguaje UML, para tener un bosquejo general de la aplicación al momento de programarla.
3. Programar todos los módulos de registro, WIFI Direct y Bluetooth que conforman la aplicación mediante Android Studio con lenguaje Kotlin y Java e implementar la base de datos con Firebase.
4. Probar la efectividad de los módulos de la aplicación mediante pruebas de campo para asegurar que se cumplan los requerimientos establecidos.

1.5 Alcance y limitaciones del proyecto

1.5.1 Alcance

De acuerdo con el alcance establecido en este proyecto y las diferentes funcionalidades, se definió cada uno de los módulos con los que va a contar la aplicación. A continuación, se realiza una descripción de cada uno de ellos:

- **Módulo registro de datos:** el registro de datos es opcional cuando el usuario utiliza por primera vez la aplicación. Para ello, los siguientes datos son obligatorios:
 - ✓ Tipo y número de documento.
 - ✓ Nombres y apellidos completos.
 - ✓ Género.
 - ✓ Grupo sanguíneo.
 - ✓ Contraseña.
 - ✓ Fecha de nacimiento.

El registro debe realizarse cuando el usuario tenga conexión a internet y sus datos quedarán almacenados en una base de datos en la nube, lo que evitará la duplicidad de registros. Después de hacer el registro, el usuario podrá utilizar la aplicación sin necesidad de conectarse a internet.

- **Módulo para envío de alertas mediante Bluetooth:** en este módulo se utiliza el adaptador Bluetooth del teléfono inteligente para enviar balizas (beacons) que puedan ser captadas por teléfonos cercanos que tengan activa la misma aplicación. Las balizas

transmiten la distancia a la que se encuentra el dispositivo que envía respecto al dispositivo que recibe. Además de emitir balizas, el módulo también permite recibirlas.

- **Módulo para envío de alertas mediante WIFI Direct:** este módulo opera de manera semejante al módulo de Bluetooth, variando la tecnología empleada y en este módulo se puede enviar y recibir los datos registrados por el usuario. En el caso de WIFI Direct se utilizó la opción denominada *descubrimiento de servicios*, que permite detectar los dispositivos cercanos sin que se requiera configurar una red.
- **Alarma sonora:** esta funcionalidad se encuentra tanto en el módulo de alertas mediante Bluetooth, como en el módulo de WIFI Direct. Esta señal será opcional si la persona la quiere activar para facilitar el rescate. Al ser esta funcionalidad opcional, si la persona quisiera apagar la alarma, lo podrá realizar de la misma forma en que se enciende. Al activarse la opción, sonarán los parlantes del móvil con un sonido de emergencia.
- **Tecnologías:**
 - ✓ Android Studio: es el IDE que se utilizó para hacer la programación del software mediante el lenguaje de Kotlin. En este IDE se desarrolló tanto el Frontend como el Backend.
 - ✓ Firebase: es la base de datos que se usó para hacer la autenticación en el software y almacenar los datos de los usuarios.

1.5.2 Limitaciones

- La aplicación no muestra la ubicación exacta del individuo en algún mapa. La aplicación calcula la distancia que separa al individuo de sus vecinos a partir de las coordenadas GPS que estos le transmiten.
- La aplicación no usa “modo avión” para hacer el ahorro de batería. El uso de esta opción, deshabilita el uso de las tecnologías para poder establecer la conexión.
- La aplicación no requiere que el teléfono esté conectado a redes móviles y red WIFI, ya que la idea de este proyecto es utilizar las dos tecnologías mencionadas para comunicar los celulares cercanos.
- Los datos emitidos por los dispositivos móviles no están encriptados. Teniendo en cuenta que es un prototipo, por ahora no es necesario incluir la encriptación en este proyecto.
- El proyecto se realizó solamente para el sistema operativo Android debido a que posee la mayor participación en el mercado y a limitaciones en el presupuesto del proyecto.

2. Marco de referencia

2.1 Marco teórico

2.1.1 Desastres naturales

Los desastres naturales se pueden definir como cualquier fenómeno natural, pero con una variante que son los daños que perjudican a los seres humanos sin previo aviso (Yirda, 2021). Los desastres naturales se pueden clasificar de la manera mostrada en la Tabla 1.

Tabla 1

Clasificación de desastres.

CLASIFICACIÓN DE DESASTRES	TIPOS DE DESASTRES
Desastres tectónicos y geológicos	Terremoto Tsunami Erupción volcánica
Desastres topográficos y geotécnicos	Deslizamiento Derrumbe Alud Aluvión Corrimiento de tierra
Desastres generados por fenómenos meteorológicos	Inundaciones Sequías Heladas Tormentas Granizada Tornados Huracanes
Desastres espaciales	Tormenta solar Impacto de asteroide o meteorito Lluvia de meteoritos

Fuente: Adaptado de (Instituto de Geografía (IGUNNE) Facultad de Humanidades, n.d.).

Por causa de estos desastres naturales se ha aumentado la cantidad de muertes, las cuales se han quintuplicado en los últimos 50 años alrededor del mundo (Capacci y Mangano, 2015).

2.1.2 Beacon

Un beacon, de manera genérica, se refiere a una trama de tamaño pequeño que un dispositivo envía para anunciar su presencia (Gerasenko et al., 2001).

2.1.3 Bluetooth

Es una tecnología de red inalámbrica que propusieron Jaap Haartsen y Mattison Sven. Actualmente, esta tecnología sigue siendo desarrollada por Bluetooth Special Interest Group. Bluetooth permite hacer transferencias de datos y se clasifica en cuatro clases. La clase uno tiene un alcance de 100 metros. La clase dos tiene alcance de 20 metros. La clase 3 tiene 1 metro. Por último, la clase 4 ofrece un alcance de 0.5 metros (Mundo Altavoces, n.d.) .

En la aplicación de Botón de Pánico, mediante el uso de los *beacons*, se hace el descubrimiento de los dispositivos y el cálculo de la distancia sin necesidad de hacer emparejamiento, a diferencia del uso tradicional de Bluetooth en el que deben autorizarse mutuamente los dispositivos para intercambiar información.

Cuando los dispositivos se emparejan por Bluetooth es necesario crear una *piconet* que es un grupo para poder intercambiar información el cual está compuesto por un líder de grupo y los integrantes. Una *piconet* tiene un máximo de 7 integrantes. Además de eso, las *piconets* pueden interconectarse entre sí y a esto se le denomina *scatternet* (Kaisar,2021).

2.1.4 Descubrimiento de servicios

El descubrimiento de servicios o *SSDP* (*simple service discovery protocol*), es un protocolo para anunciar la presencia de un dispositivo mediante el cual se puede compartir información o mensajes cuando se escuchan mutuamente, además se puede anunciar para todos los dispositivos que estén haciendo uso de este protocolo (Revista Transformación Digital, n.d.).

2.1.5 WIFI Direct

Es un protocolo que permite la comunicación e intercambio de archivos entre dispositivos de manera inalámbrica. La diferencia entre WIFI Direct y Bluetooth, es que WIFI Direct permite el intercambio de archivos a una mayor velocidad que la de Bluetooth; además, WIFI Direct tiene un mayor alcance (Fernández, 2020).

El uso de WIFI Direct en la aplicación se realiza mediante el descubrimiento de servicios con el fin de recibir información de otros dispositivos sin necesidad de formar un grupo y, por lo tanto, de tener un *group owner* (es el dispositivo que administra el grupo creado en WIFI Direct). Cuando se tiene un *group owner*, se pueden tener conexiones uno a uno o uno a muchos (Kaisar,2021).

2.1.6 Android Studio

Es un entorno de desarrollo para aplicaciones en sistema operativo Android. Además de ser un editor de código, ofrece más herramientas que pueden aumentar la productividad al momento del desarrollo de aplicaciones (Android Studio, n.d.). En Android Studio se puede usar

lenguajes de programación, los cuales son lenguajes formales, que se usan por medio de una serie de instrucciones y órdenes para poder crear programas que controlen el comportamiento físico y lógico de los dispositivos. Este es el medio por el cual el programador le ordena a la máquina de una manera muy precisa aspectos como cuándo debe operar el software, de qué forma lo debe hacer, almacenamiento, transmisión de datos, entre otros (Rockcontent, n.d.).

2.1.7 Kotlin

Es un lenguaje de programación para el sistema operativo Android, que permite hacer programación orientada a objetos. Este es un lenguaje administrado y creado por JetBrains y Google (Android Studio, n.d.).

2.1.8 Firebase

Es una plataforma en la nube para la ayuda del desarrollo de aplicaciones web y móviles. Fue creada en el año 2014 y es compatible para distintas plataformas (IOS, Android y web). Una de las herramientas que nos ofrece Firebase es una base de datos en tiempo real, esta se aloja en la nube y es una base de datos no relacional (López, 2020).

2.1.9 Bases de datos no relacionales

Tienen como característica principal que no utilizan tablas para la organización de los datos. Este modelo de almacenamiento está optimizado por requisitos específicos en los datos que se almacenan en estructuras de tipo Json (Tejada, 2018). Además de esto, las bases de datos no relacionales están diseñadas para el uso, cuando se necesite un gran volumen de datos. Las ventajas de utilizar este tipo de bases, es que son: flexibles porque permiten el desarrollo y realizar consultas

más rápida e iterativamente; de alto rendimiento ya que permiten hacer la búsqueda más rápida de la información que almacena y altamente funcionales ya que se adaptan a la necesidad de la aplicación (Amazon, n.d.).

2.1.10 Scrum

Scrum es una metodología ágil que facilita el trabajo entre equipos. Este modelo presenta dos características importantes: puede generar soluciones adaptables para cualquier tipo de problema y organizar el desarrollo mediante los llamados *sprints* (Schwaber y Sutherland, 2015). Un *sprint* es cada uno de los ciclos o iteraciones que se tienen dentro de la metodología con una duración que varía entre dos semanas y un máximo de dos meses (Requena, 2018).

Las buenas prácticas de Scrum se reflejan en:

- **Reunión de planificación de *sprint*:** consiste en pautar los objetivos para las historias de usuario que se van a trabajar en ese *sprint*. También se establece el procedimiento para el desarrollo de estos objetivos. Para un *sprint* de dos semanas, las reuniones normalmente tienen una duración de 4 horas una vez al inicio de cada *sprint* (Bara, 2015).
- **Reunión diaria:** esta reunión consiste en sincronizar las actividades que se han trabajado previamente para revisarlas; de igual manera, se facilita la comunicación con todo el equipo de trabajo para mejorar la toma de decisiones, el planteamiento de las actividades a desarrollar y evaluar el progreso del *sprint*. Estas reuniones tienen una duración de 15 minutos (Bara, 2015).

- **Trabajo de desarrollo durante el *sprint*:** cuando los *sprints* están en progreso, hay que asegurar que no se van a cambiar los objetivos estipulados en la reunión de planificación de *sprint* (Bara, 2015).
- **Revisión del *sprint*:** estas reuniones se realizan al final de cada *sprint* entre el equipo de trabajo y los *stakeholders* (son todos los interesados en el proyecto incluyendo a los clientes) con el fin de contrastar el *Product Backlog* (lista de las actividades iniciales del producto organizadas por prioridad) y los entregables. El objetivo de esta revisión es lograr una retroalimentación por parte del interesado e identificar lo que se ha hecho del backlog. Por otro lado, el equipo de trabajo manifiesta las complicaciones que pudieron tener y cómo las resolvieron (Bara, 2015).
- **Retroalimentación:** en esta reunión lo que se busca es analizar el rendimiento del equipo de trabajo y buscar las mejoras de cómo el equipo debe hacer su trabajo (Bara, 2015).

Para poder llevar a cabo esta metodología es necesario tener en cuenta los siguientes roles:

- ***Scrum master*:** es el líder del equipo de trabajo y aparte de tener comunicación directa con el *Product owner*, guía el equipo para cumplir las etapas o las buenas prácticas de la metodología (Softeng, n.d.).
- ***Product owner*:** es el responsable de establecer el *Product Backlog* de acuerdo con las historias de usuario que determina el equipo de trabajo, revisando las prioridades de

acuerdo a como va avanzando el proyecto. Por otro lado, es el representante de los stakeholders (Softeng, n.d.).

- **Equipo de trabajo:** son las personas que se encargan del desarrollo de software, de acuerdo con los entregables de cada *sprint* (Softeng, n.d.).

2.2 Antecedentes

Buscando información de aplicaciones similares a la que se planteó, se encontraron las que se explican a continuación.

2.2.1 Kaifa

Esta aplicación envía señales de emergencia en un rango de 250 metros. Cuando la persona activa el reporte de ayuda, las personas que se encuentren dentro del rango especificado, tendrán la posibilidad de aceptar o rechazar la alerta. La aplicación funciona bajo conexión a internet, usa GPS y temporizador para cancelar la activación de emergencia si se llega a activar por equivocación (Mody et al., 2020).

2.2.2 AlertaMovil

Esta aplicación consiste en enviar mensajes de texto y la localización a seis contactos, de preferencia, al momento de activar la emergencia. La señal se puede emitir mediante un botón en la pantalla principal, o presionando 5 veces el botón de encendido. Esta aplicación hace uso de GPS, Bluetooth e Internet. Sin embargo, se puede enviar la señal de ayuda sin necesidad de

conexión a internet, pero solo la recibirán las personas que estén dentro de la aplicación en un rango determinado (Acosta et al., 2017).

2.2.3 Botón de pánico SOS

Esta aplicación usa una lista de contactos de emergencia a quienes envía la ubicación del móvil mediante un mensaje de texto cuando se agita el celular. De igual manera, al presionar un botón que está en la pantalla de la aplicación se realizará una llamada de emergencia a un número predefinido. Además de esto, la aplicación solo permite un uso limitado de alertas, después de las cuales es necesario pasarse a la versión premium para seguir usándola (Kisara Labs, 2021).

2.2.4 Comparaciones

En la siguiente tabla se muestra una comparación entre las aplicaciones antes mencionadas y la aplicación que se propuso en este proyecto llamada *botón de pánico*.

Tabla 2
Comparaciones de aplicaciones similares.

	Kafya	AlertaMovil	Botón de pánico SOS	Botón de pánico
Bluetooth	✗	✓	✗	✓
WIFI Direct	✗	✗	✗	✓
Ubicación	✓	✓	✓	✓
Activación rápida de alarma	✓	✓	✓	✓
Alarma sonora	✗	✗	✗	✓

Uso de contactos				
Llamada de emergencia				
Internet	Siempre es necesario el uso de internet.	No siempre es necesario el uso de internet.	Siempre es necesario el uso de internet.	Solo es necesario para hacer el registro.
Gratis				

Fuente: elaboración propia.

En conclusión, hay varias aplicaciones similares a la que se ha planteado en el proyecto, las cuales tienen un enfoque en seguridad ciudadana en tanto que la aplicación propuesta se enfoca en situaciones de desastres naturales con el fin de lograr rescatar y salvar vidas. Otra diferencia importante es que se incluye un módulo de conexión mediante WIFI Direct, tecnología que las otras aplicaciones no poseen. Ya como se mencionó en el capítulo anterior, esta tecnología tiene el mismo alcance que Bluetooth pero mayor velocidad de transferencia de datos por lo que es adecuada para el enfoque de esta aplicación.

2.3 Marco legal

2.3.1 Ley de derechos de autor

La ley 23 de 1982 sobre derechos de autor, busca proteger las obras científicas, literarias, artísticas, para que no sean divulgadas o reproducidas sin reconocer el patrimonio del autor (Congreso de Colombia, 1982). En este caso, la dueña de los derechos de autor será la Universidad Antonio Nariño ya que será la encargada de acoger esta normativa.

2.3.2 Ley de protección de datos

La ley 1581 de 2012 es la que reconoce y protege los datos personales donde tienen el derecho de conocer, actualizar y verificar la información que se tenga sobre ella y que sean susceptibles de tratamiento por cualquier entidad pública o privada. También acoge la protección de los datos por parte de las entidades que recopilen este tipo de información sensible (Congreso de Colombia, 2012). En el marco del presente proyecto se conoce y se aplicará esta ley teniendo en cuenta que el proyecto es un desarrollo que solicita información personal para el registro de un usuario.

3. Aspectos metodológicos

En este capítulo se describe la manera como se aplicó la metodología *Scrum* para el desarrollo de este trabajo de grado.

Para este proyecto, los *sprints* tuvieron una duración de 4 semanas cada uno, para un total de 3 *sprints*, teniendo en cuenta que se desarrolló un módulo de la aplicación por cada *sprint*. Con el fin de adaptar esta metodología a este proyecto académico, se programaron reuniones semanales para tener un control de lo que se iba realizando y quedaba pendiente en cada uno de los *sprints*.

Se implementó el *Product Backlog* a fin de tener presentes los requerimientos del cliente para el *sprint* que se estaba cursando. También al finalizar cada *sprint*, se realizó una reunión con el equipo de trabajo para analizar las falencias y el buen rendimiento del equipo.

Los roles que se trabajaron en este proyecto enmarcados en la metodología Scrum son:

- *Scrum master*: Sergio Giovanni Silva Arias, Andrés Felipe Cifuentes Mendieta. Se turnaron los roles en cada *sprint*.
- *Product owner*: María del Pilar Salamanca.
- Equipo de trabajo: Sergio Giovanni Silva Arias, Andrés Felipe Cifuentes Mendieta.

3.1 *Sprint 1*

En este *sprint* se programó el módulo de registro de datos, por lo que se realizaron las siguientes actividades:

- Etapa de levantamiento de requerimientos:
 - ✓ Reunión con el cliente.
 - ✓ Plantear requerimientos del módulo.
 - ✓ Generar historias de usuario para el módulo.
 - ✓ Aprobación de historias de usuario.

Entregable: historias de usuario aprobadas por el cliente.

- Etapa de diseño de la estructura del software:
 - ✓ Definir datos a implementar en la base de datos.
 - ✓ Diagrama de clases para el módulo.
 - ✓ Diseño de *Mockup* para el módulo.

Entregable: diagrama de clases y *Mockup*.

- Etapa de programación:
 - ✓ Integración de Firebase con Android Studio.
 - ✓ Programación de interfaz gráfica del módulo.
 - ✓ Programación del módulo.

Entregable: módulo programado.

- Etapa de pruebas:
 - ✓ Pruebas de sistema.
 - ✓ Pruebas de aceptación.

Entregable: informe de las pruebas realizadas.

- Etapa de documentación:
 - ✓ Manual de usuario
 - ✓ Manual técnico
 - ✓ Monografía

Entregable: avances de los documentos mencionados anteriormente en cuanto al módulo de registro.

3.2 *Sprint 2*

En este *sprint* se desarrolló el módulo de WIFI Direct para lo que se realizaron las siguientes actividades:

- Etapa de levantamiento de requerimientos:
 - ✓ Reunión con el cliente.
 - ✓ Plantear requerimientos del módulo.
 - ✓ Generar historias de usuario para el módulo.

- ✓ Aprobación de historias de usuario.

Entregable: historias del módulo aprobadas por el cliente.

- Etapa de diseño de la estructura del software:

- ✓ Diseño de *Mockup* para el módulo

- ✓ Diagrama de clases para el módulo

Entregable: diagrama de clases y *Mockup*.

- Etapa de programación:

- ✓ Programación de interfaz gráfica del módulo

- ✓ Programación del módulo

Entregable: módulo programado.

- Etapa de pruebas:

- ✓ Pruebas de integración

- ✓ Pruebas de sistema

- ✓ Pruebas de aceptación

Entregable: informe de las pruebas realizadas.

- Etapa de documentación:

- ✓ Manual de usuario
- ✓ Manual técnico
- ✓ Monografía

Entregable: avances de los documentos mencionados anteriormente en cuanto al módulo de WIFI Direct.

3.3 *Sprint 3*

En este *sprint* se programó el módulo de Bluetooth para lo que se realizaron las siguientes actividades:

- Etapa de levantamiento de requerimientos:

- ✓ Reunión con el cliente.
- ✓ Plantear requerimientos del módulo.
- ✓ Generar historias de usuario para el módulo.
- ✓ Aprobación de historias de usuario.

- Etapa de diseño de la estructura del software:

- ✓ Diseño de *Mockup* para el módulo
- ✓ Diagrama de clases para el módulo

Entregable: diagrama de clases y *Mockup*.

- Etapa de programación
 - ✓ Programación de interfaz gráfica del módulo
 - ✓ Programación del módulo

Entregable: módulo programado.

- Etapa de pruebas:
 - ✓ Pruebas de integración
 - ✓ Pruebas de sistema
 - ✓ Pruebas de aceptación:

Entregable: informe de las pruebas realizadas.

- Etapa de documentación:
 - ✓ Manual de usuario
 - ✓ Manual técnico
 - ✓ Monografía

Entregable: documentos finalizados.

4. Desarrollo del proyecto

En el proceso de desarrollo de esta aplicación móvil, se optó por usar la metodología scrum, con la intención de poder agilizar el desarrollo de cada uno de los módulos propuestos en el capítulo anterior. A continuación, se realiza una descripción de como se ejecutó el proyecto.

4.1 Descripción de la aplicación.

Botón de pánico es una aplicación móvil que apoya a los rescatistas en la búsqueda de personas que hayan sido afectadas por alguna catástrofe natural y necesiten ser rescatadas. Para cumplir con este objetivo se usaron las tecnologías WIFI Direct y Bluetooth, que permiten enviar datos de la persona almacenados en el dispositivo, calculando la distancia para poderlos encontrar. Se usaron estas dos tecnologías alternativas, ya que se sabe que en situaciones de catástrofes naturales, las redes WIFI y redes móviles pueden colapsar.

4.2 Fase de iniciación.

En esta fase se determinó cuales *sprints* se llevarían a cabo en el proyecto, definiendo lo que se iba a desarrollar en cada uno de estos. Siguiendo con las recomendaciones de la metodología, se determinaron las personas responsables de cada rol como se estipuló en el capítulo anterior y diseñaron los diagramas y casos de uso.

4.3 Planeación y estimación

4.3.1 Roles del sistema

- **Usuario:** Rol único de la aplicación; es el encargado de crear su propia cuenta para poder registrarse en la aplicación, iniciar sesión con las credenciales que usó para crear la cuenta, abrir la aplicación en caso de necesitar ayuda, encender la alarma sonora y, posteriormente, en caso de no necesitar más ayuda, cerrar la aplicación.

4.3.2 Arquitectura y herramientas

Para la base de datos, se optó por utilizar un sistema de almacenamiento no relacional para guardar el registro de los usuarios. Firebase se escogió como la herramienta para almacenar la información del usuario, ya que facilita la creación de una estructura de datos en la cual no es necesario enfocarse en las relaciones con otras tablas y proporciona la autenticación de usuarios para evitar duplicidad de registros. Firebase utiliza un solo tipo de dato que es una cadena de texto. La persistencia y guardado de los datos se puede observar en la figura 1

Figura 1

Tabla base de datos Firebase

User
apellido
contraseña
documento
fecha_nacimiento
genero
nombre
rh
tipo_documento

Fuente: elaboración propia

El tipo de dato para los campos que se deben diligenciar en el registro son los siguientes:

- Apellido: tipo texto
- Contraseña: tipo texto
- Documento: tipo numérico
- Fecha de nacimiento: tipo texto
- Género: tipo texto
- Nombre: tipo texto
- Rh: tipo texto
- Tipo de documento: tipo texto

4.3.3 Historias de usuario

Según el levantamiento de requerimientos, se establecieron unos requisitos con los cuales tenía que contar la aplicación, se elaboraron unas historias de usuario en donde se describen cada uno de los requerimientos, asignándoles una persona encargada de de realizar esas actividades y unos requisitos para que fuera aprobada.

4.4 Implementación y desarrollo

En esta fase se pusieron en desarrollo las actividades propuestas en cada *sprint*, dándole soluciones a los requerimientos e historias de usuario previamente planteadas, agregando el desarrollo de los *Mockups* haciendo el bosquejo de como se conforma la aplicación y el funcionamiento de cada módulo. Este proyecto de grado se dividió en 3 *sprints*.

4.4.1 *Sprint 1*: módulo de registro de usuarios

Con base en la reunión inicial con el *Product owner* se establecieron las historias de usuario que se trabajaron en el primer módulo de la aplicación. Estas historias son *registro de usuario* e *inicio de sesión*, cuyos detalles se pueden observar en las tablas **3** y **4**.

Tabla 3*Historia de usuario 1: Registrar usuario*

Indicador:HU01	Indispensable	Prioridad: alta
Nombre de Caso de Uso:	Registrar usuario	
Autor:	Andrés Felipe Cifuentes	
Fecha:	02 de agosto de 2021	
Categoría (Visible):	Actores Involucrados: Usuario	
Resumen:	El usuario seleccionará la opción Registrarse, después digita los datos solicitados (contraseña de tipo texto, número de documento de tipo texto, tipo de documento de tipo texto, nombres de tipo texto, apellidos de tipo texto, RH de tipo texto, fecha de nacimiento de tipo texto) y por último debe seleccionar la opción Registrarme para culminar el proceso.	
Curso Básico Eventos	<ol style="list-style-type: none"> 1. El usuario selecciona la opción Registrarse. 2. El sistema lo direcciona a la pantalla para registrar los datos. 3. El usuario debe ingresar la información en los campos contraseña, número de documento, tipo de documento, nombres, apellidos de tipo texto, RH de tipo texto, fecha de nacimiento de tipo texto y selecciona Registrarme para finalizar. 4. El sistema realiza la validación de los datos y los guarda en Firebase. Luego le aparece un mensaje de confirmación del registro y por último lo devuelve a la pantalla para iniciar sesión. 	
Caminos Alternativos	N/A	
Caminos de Excepción	<ol style="list-style-type: none"> 1. Los datos ingresados no están completos por lo que no guarda nada y se solicita que diligencie todo. 2. El documento de identidad ya está registrado por lo que no guarda nada y se le informa esta situación. 3. No tiene conexión a internet por lo que no guarda nada y se le informa que debe revisar la conexión a internet. 	
Puntos de Extensión	HU02	
Pre - Condiciones	Debe estar conectado a internet para hacer el registro. Si no está conectado no podrá registrarse en la aplicación.	
Post - Condiciones	Después de registrar la cuenta, la aplicación lo direcciona a la pantalla para iniciar sesión.	
Criterios de Aceptación	Debe mostrar los campos para diligenciar la información de la persona, cuando el usuario selecciona la opción Registrarme debe realizar la validación de los campos. Dependiendo de esa validación deberá guardar o no los datos en Firebase.	

Fuente: elaboración propia.

Tabla 4*Historia de usuario 2: Inicio de sesión*

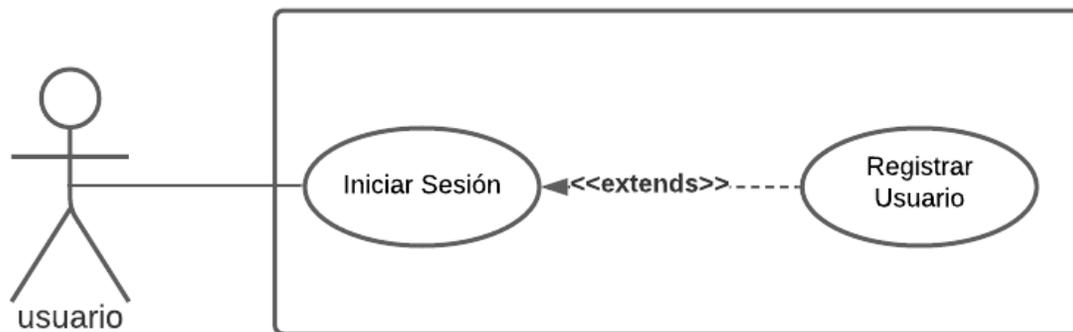
Indicador:HU02	Indispensable	Prioridad: alta
Nombre de Caso de Uso:	Inicio de sesión	
Autor:	Sergio Silva	
Fecha:	02 de agosto de 2021	
Categoría (Visible):	Actores Involucrados: Usuario	
Resumen:	El usuario podrá iniciar sesión en la aplicación con las credenciales que escribió en el registro (este inicio de sesión será solo la primera vez que use la aplicación).	
Curso Básico Eventos	<ol style="list-style-type: none"> 1. El usuario ingresa las credenciales que son el tipo y número de documento y la contraseña y selecciona iniciar sesión. 2. El sistema valida que las credenciales sean correctas y lo direcciona a la interfaz principal de la aplicación. 	
Camino Alternativos	<ol style="list-style-type: none"> 1. El usuario selecciona la opción para Ingresar sin usuario. 2. El sistema lo direcciona a la interfaz principal. 	
Camino de Excepción	<ol style="list-style-type: none"> 1. Si al momento de seleccionar la opción de iniciar sesión, las credenciales son incorrectas, le advertirá que debe validar la información ingresada. 2. Si el sistema detecta campos vacíos al momento de tocar el botón Iniciar sesión, le saldrá un mensaje que le informa los campos vacíos. 3. Si selecciona la opción Iniciar Sesión y no está conectado a internet, el sistema mostrará un mensaje informándolo. 	
Puntos de Extensión	N/A	
Pre - Condiciones	<p>Se necesita haber creado una cuenta para poder iniciar sesión.</p> <p>Debe estar conectado a internet para hacer la validación.</p>	
Post - Condiciones	Se debe poder iniciar sesión tanto con credenciales como sin ellas.	
Criterios de Aceptación	Debe cargar la interfaz gráfica donde aparezca los campos para introducir las credenciales, una opción para iniciar la sesión validando los datos de los campos, y una opción para iniciar sesión sin los datos anteriores. Adicionalmente, debe aparecer la opción para registrarse.	

Fuente: elaboración propia.

El siguiente paso fue la elaboración del diagrama de casos de uso para el módulo, que es útil para diseñar la estructura del código al momento de programar. Este diagrama se muestra en la figura 2.

Figura 2

Diagrama caso de uso: Iniciar sesión



Fuente: elaboración propia

Por otra parte, se hizo el desarrollo de los *Mockups* del módulo, los cuales se muestran en las figuras 3 y 4. Este modelo de la aplicación se hizo mediante la herramienta online de Marvel, que posteriormente se expuso en una reunión al *Product owner*. En esta reunión fue aprobado este diseño para empezar con el desarrollo del módulo.

Figura 3
Mockup pantalla de inicio de sesión

Botón de pánico

Tipo de documento

Número de documento

Contraseña

INICIAR SESIÓN

REGISTRARME

INGRESAR SIN CUENTA

Fuente: elaboración propia

Figura 4
Mockup pantalla de registro

Botón de pánico

Tipo de documento

Número de documento
Ingrese su número de documento

Nombres
Ingrese sus nombres

Apellidos
Ingrese sus apellidos

Género

RH

0 +

Fecha de nacimiento
16/12/1999

Contraseña
Ingrese su contraseña

REGISTRARME

Fuente: elaboración propia

En seguida se inició la programación del módulo de registro. Para este proceso fue necesario crear un proyecto en el entorno de desarrollo de Android Studio. Seguido de esto, se creó un nuevo proyecto en la nube de Firebase para poder vincularlos mediante el *package* (paquete del proyecto). Al tener enlazados los dos entornos, Firebase solicitó agregar un archivo de tipo Json en el IDE Android Studio que él mismo genera para poder acceder a las librerías.

Posteriormente se hizo la interfaz gráfica para la pantalla de inicio de sesión que se muestra en la figura **5**, así como la pantalla para registrar al usuario que se observa en las figuras **6 a 8**.

Figura 5
Pantalla de inicio de sesión

BOTÓN DE PÁNICO

Tipo de documento
Seleccione

Número de documento

Contraseña

INICIAR SESIÓN

REGISTRAME

INGRESAR SIN USUARIO

Fuente: elaboración propia

Figura 6
Pantalla de registro (1)

REGISTRO

Tipo de documento
Seleccione

Número de documento

Nombres

Apellidos

Género
Seleccione

RH
Seleccione +

Fecha de nacimiento

Contraseña

Fuente: elaboración propia

Figura 7
Pantalla de registro (2)

Fuente: elaboración propia

Figura 8
Pantalla de registro (fecha de nacimiento)

Fuente: elaboración propia

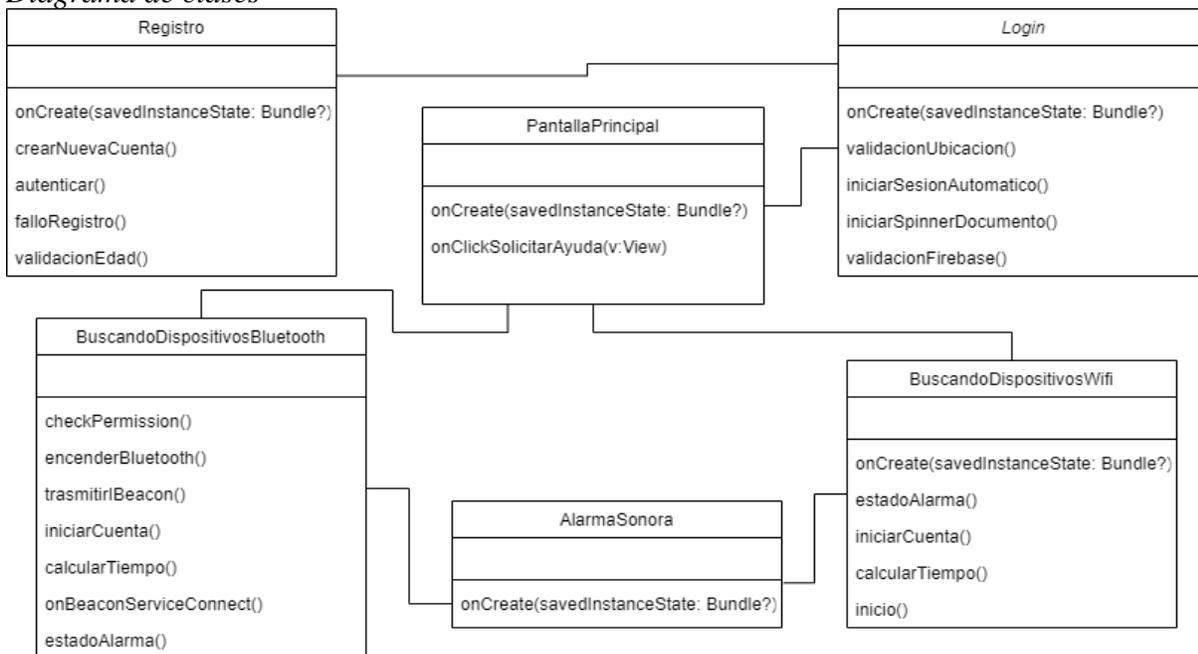
En la parte superior del inicio de sesión (figura 5), se encuentra el nombre de la aplicación (Botón de Pánico) seguido del logo que se diseñó; debajo se encuentra un menú desplegable donde se solicita el tipo de documento y luego dos campos donde se ingresa el número de documento y la contraseña. En la parte inferior se ubican tres botones, el primero es para iniciar sesión, el segundo es para hacer el registro y el tercero es para ingresar sin necesidad de estar registrado. Para cada una de las acciones de esta pantalla, se desarrollaron varias funciones que ayudan al

buen funcionamiento de la aplicación. La primera función que se implementó, fue *onCreate*, allí se carga la vista y se llaman los siguientes métodos: *validacionUbicacion*, *iniciarSesionAutomatico* e *inicializarSpinnerDocumento*.

En el método *validacionUbicacion*, se verifica que el teléfono tenga habilitado el uso de la ubicación; si el usuario no la activa, posteriormente se le va a solicitar que lo haga. Cuando una persona ingresa las credenciales e inicia sesión, la aplicación guardará los datos localmente y cuando desee volver a ingresar, se cargará el método *iniciarSesionAutomatico* y no tendrá que volver a escribir las credenciales, a no ser que la persona cierre la sesión. Por último, *inicializarSpinnerDocumento* inicializa el menú desplegable del tipo de documento.

Las clases y los métodos que se implementaron en toda la aplicación se muestran en la Figura 9.

Figura 9
Diagrama de clases



Fuente: elaboración propia

Cada botón del módulo tiene una función asociada para realizar su acción correspondiente. El botón *Iniciar sesión* valida en Firebase que las credenciales coincidan y, de ser así, almacenará la información registrada localmente y direccionará la aplicación hacia la pantalla principal. Para ir a la pantalla del registro, el usuario debe seleccionar la opción *Registrarme*, pero si no desea hacerlo, tendrá la posibilidad de ingresar a la pantalla principal de la aplicación mediante el botón *Ingresar sin usuario*.

Para el proceso de registro será necesario diligenciar un formulario que solicita los datos: tipo de documento, número de documento, nombres, apellidos, género, grupo sanguíneo, fecha de nacimiento y una contraseña que se usará para el inicio de sesión, como se muestra en las figuras **6 a 8**. Teniendo conformada la parte gráfica de la aplicación, se implementó e integró cada uno de los campos para que el usuario al momento de diligenciarlos y registrarse se pudieran guardar de forma exitosa en la base de datos almacenada en la nube de Firebase, como se observa en la figura **10**. Para completar el registro, se debe presionar el botón *Registrarme*.

Figura 10

Base de datos en la nube Firebase



Fuente: elaboración propia

4.4.1.1 Pruebas

Para este primer *sprint* se hicieron las pruebas correspondientes, evaluando los resultados obtenidos al momento de registrar un usuario y de iniciar sesión, como se muestra en las tablas **5** y **6**. Para la realización de estas pruebas se utilizaron tres dispositivos de marcas diferentes: un Samsung J7 Prime con Android 8.1, un Xiaomi Redmi Note 8 con Android 11 y un Samsung J5 Prime con Android 8.0.

Tabla 5*Caso de prueba Registrar usuario*

Nombre del proyecto: BOTÓN DE PÁNICO		Caso No: 1	
Nombre de caso de la prueba: Registrar Usuario			
Estado de la prueba: finalizada		Pantalla de registro	
Escrito por: Sergio Silva		Ejecutado por: Andrés Cifuentes	
Descripción del caso de prueba: se valida que el usuario pueda registrarse correctamente.			
Configuración de la prueba: se requiere de un dispositivo móvil con versión de Android superior a la 5.0, con la aplicación instalada para poder diligenciar los campos			
Flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Abrir la aplicación en el dispositivo	Se muestra una pantalla que contiene los botones: iniciar sesión, registrarme e ingresar sin usuario.	Exitoso
2	Seleccionar la opción Registrarse	Se muestra una pantalla que tiene todos los campos necesarios para diligenciar	Exitoso
3	Seleccionar la opción Registrarme	Lo devuelve a la pantalla de inicio de sesión de la aplicación y se almacena correctamente la información en Firebase.	Exitoso
Excepciones			
1	Ingresar un documento ya registrado anteriormente y seleccionar Registrarme	Se muestra un mensaje que dice: "El usuario ya está registrado" y no guarda nada.	Exitoso
2	Seleccionar "Registrarme" dejando uno o más campos en blanco	Se muestra un aviso debajo del campo o campos vacíos y no guarda nada.	Exitoso
3	Ingresar una contraseña de menos de 6 caracteres	Se muestra un aviso debajo del campo de la contraseña indicando que debe ser 6 o más caracteres y no guarda nada.	Exitoso
4	Ingresar dos contraseñas distintas en el campo de contraseña y confirmar contraseña	Se muestra un mensaje que indica que las contraseñas no son iguales y no guarda nada.	Exitoso
5	Seleccionar Registrarme con el dispositivo sin conexión a internet	Se muestra un mensaje que le pide que verifique la conexión a internet.	Exitoso

Fuente: elaboración propia.

Tabla 6*Caso de prueba Iniciar Sesión*

Nombre del proyecto: BOTÓN DE PÁNICO		Caso No: 2	
Nombre de caso de la prueba: Iniciar Sesión			
Estado de la prueba: finalizada		Pantalla de inicio de sesión	
Escrito por: Andrés Cifuentes		Ejecutado por: Sergio Silva	
Descripción del caso de prueba: se valida que el usuario pueda iniciar sesión correctamente.			
Configuración de la prueba: se requiere de un dispositivo móvil con versión de Android superior a la 5.0, la aplicación en ejecución y haberse registrado previamente.			
Flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Abrir la aplicación en el dispositivo	Se muestra una pantalla que contiene los botones: Iniciar sesión, Registrarme e Ingresar sin usuario.	Exitoso
2	Diligenciar los campos de: tipo de documento, número de documento y contraseña	Se muestra una pantalla que tiene todos los campos para poder iniciar sesión	Exitoso
3	Seleccionar la opción de iniciar sesión	Lo dirige a la pantalla donde se encuentra el botón de pánico para pedir ayuda.	Exitoso
Excepciones			
1	Ingresar un tipo y número de documento no registrados en la aplicación previamente	Se muestra un mensaje indicando que las credenciales son incorrectas	Exitoso
2	Ingresar los datos de tipo y número de documento y contraseña y que alguno de estos no coincida con lo registrado en Firebase	Se muestra un mensaje indicando que las credenciales son incorrectas	Exitoso
3	Ingresar los datos de tipo y número de documento y contraseña y que el dispositivo no esté conectado a internet	Se muestra un mensaje solicitando que verifique la conexión a internet	Exitoso

Fuente: elaboración propia.

4.4.2 *Sprint 2*: módulo de envío de alertas por WIFI Direct

Las historias de usuario que se desarrollaron en este módulo fueron 4: *Enviar alertas por WIFI Direct, Recargar pantalla WIFI Direct, Prender alarma sonora y Cerrar sesión*. Las tablas 7 a 10 muestran los detalles de cada una de ellas.

Tabla 7

Historia de usuario 3: Enviar Alertas por WIFI Direct

Indicador:HU03	Indispensable	Prioridad: alta
Nombre de Caso de Uso:	Enviar alertas por WIFI Direct	
Autor:	Sergio Silva	
Fecha:	01 de septiembre de 2021	
Categoría (Visible):	Actores Involucrados: Usuario	
Resumen:	El usuario toca el botón Solicitar Ayuda, el sistema lo direcciona a la pantalla para solicitar ayuda por WIFI Direct y el teléfono empezará a emitir señales mediante esta tecnología.	
Curso Básico Eventos	<ol style="list-style-type: none"> 1. El usuario toca el botón Solicitar Ayuda. 2. El sistema lo direcciona a la pantalla donde se le indica que está emitiendo las señales y muestra a los dispositivos cercanos. 	
Caminos Alternativos	Si el teléfono tiene apagado el adaptador WIFI, la aplicación le solicita prenderlo.	
Caminos de Excepción	No se podrá emitir señales si no acepta los permisos al inicio de la aplicación para el uso de los módulos.	
Puntos de Extensión		
Pre - Condiciones	HU02	
Post - Condiciones	Después de activar la opción, el sistema lo direcciona a la interfaz donde la señal está activa y se observan los dispositivos cercanos disponibles	
Criterios de Aceptación	Debe cargar la interfaz gráfica donde aparezca el botón Solicitar Ayuda. Al presionar el botón, debe validar los permisos y si son aceptados, debe empezar a emitir información mediante WIFI Direct.	

Fuente: elaboración propia.

Tabla 8*Historia de usuario 4: Recargar pantalla WIFI Direct*

Indicador:HU04	Indispensable	Prioridad: alta
Nombre de Caso de Uso:	Recargar pantalla WIFI Direct	
Autor:	Sergio Silva	
Fecha:	01 de septiembre de 2021	
Categoría (Visible):	Actores Involucrados: Usuario	
Resumen:	El usuario seleccionará la opción Recargar y el sistema actualizará la lista de dispositivos encontrados	
Curso Básico Eventos	1-El usuario selecciona la opción Recargar. 2-El sistema actualizará la lista de dispositivos cercanos que encuentre por WIFI Direct y los dispositivos ya detectados, si los hay.	
Caminos Alternativos	N/A	
Caminos de Excepción	N/A	
Puntos de Extensión	N/A	
Pre - Condiciones	HU03	
Post - Condiciones	Después de recargar la pantalla, se debe continuar emitiendo señales, así como mostrar los dispositivos que se vayan encontrando.	
Criterios de Aceptación	Debe estar el botón Recargar y, al momento de tocarlo, se debe reiniciar todo el proceso de envío y recepción de señales.	

Fuente: elaboración propia.

Tabla 9*Historia de usuario 5: Cerrar Sesión*

Indicador:HU05	Indispensable	Prioridad: alta
Nombre de Caso de Uso:	Cerrar sesión	
Autor:	Andrés Cifuentes	
Fecha:	01 de septiembre de 2021	
Categoría (Visible):	Actores Involucrados: Usuario	
Resumen:	El usuario podrá cerrar sesión cuando lo desee desde la pantalla donde aparece el botón Solicitar Ayuda, haciendo uso del botón que dice Cerrar Sesión.	
Curso Básico Eventos	1-El usuario debe estar en la pantalla principal y seleccionar la opción de cerrar sesión 2-El sistema lo direccionará a la pantalla de inicio de sesión.	
Caminos Alternativos		
Caminos de Excepción		
Puntos de Extensión	HU03	
Pre - Condiciones	Haber ingresado a la aplicación	
Post - Condiciones	Se debe poder cerrar sesión con o sin acceso a internet	
Criterios de Aceptación	Debe cargar la interfaz gráfica de la pantalla principal y el botón de Cerrar Sesión debe mostrarse. Una vez toque el botón, la aplicación debe dirigirlo a la pantalla de inicio de sesión	

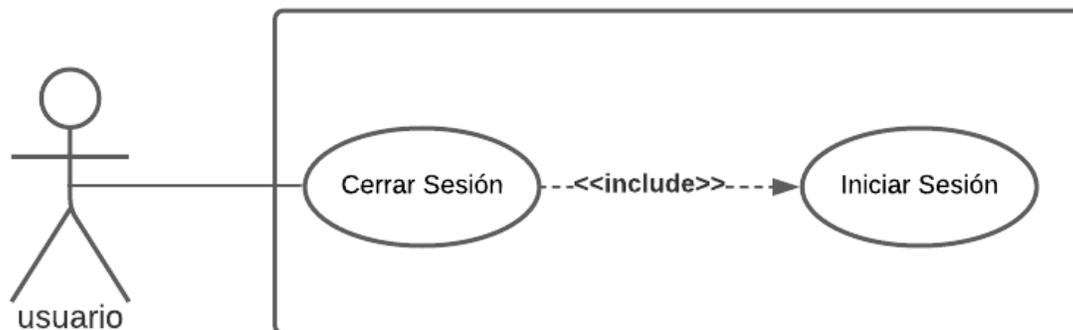
Fuente: elaboración propia.

Tabla 10*Historia de usuario 6: Encender Alarma Sonora*

Indicador:HU06	Indispensable	Prioridad: alta
Nombre de Caso de Uso:	Encender alarma sonora	
Autor:	Andrés Felipe Cifuentes Mendieta	
Fecha:		
Categoría (Visible):	Actores Involucrados: Usuario	
Resumen:	El usuario presionará el botón Alarma Sonora y el dispositivo empezará a emitir sonidos de emergencia.	
Curso Básico Eventos	1-El usuario seleccionará la opción de alarma sonora 2-El dispositivo empezará a emitir sonidos de emergencia.	
Caminos Alternativos	N/A	
Caminos de Excepción	N/A	
Puntos de Extensión		
Pre - Condiciones	HU03, HU07	
Post - Condiciones	Después de seleccionar la alarma, el dispositivo debe sonar	
Criterios de Aceptación	Debe cargar la interfaz gráfica donde aparezca la opción de la alarma sonora y poder activarla.	

Fuente: elaboración propia.

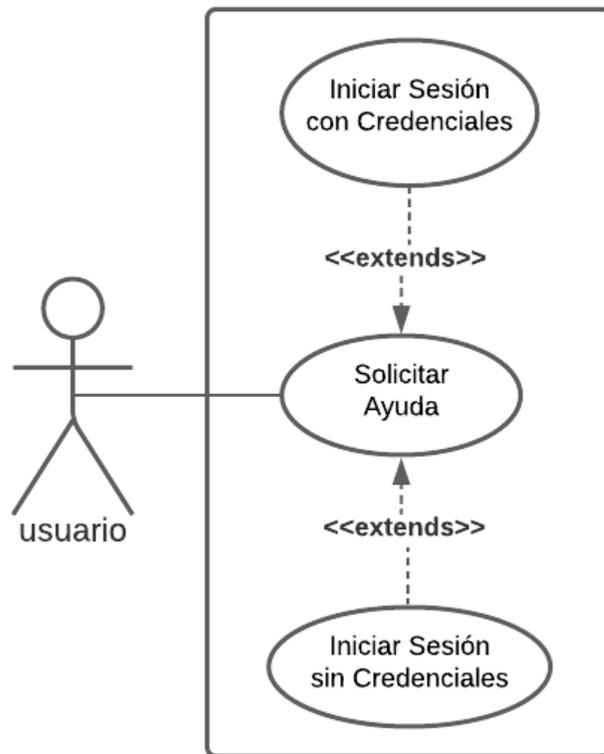
El siguiente paso fue elaborar los diagramas de casos de uso que se presentan en las figuras 11 a 14.

Figura 11*Diagrama caso de uso: Cerrar sesión*

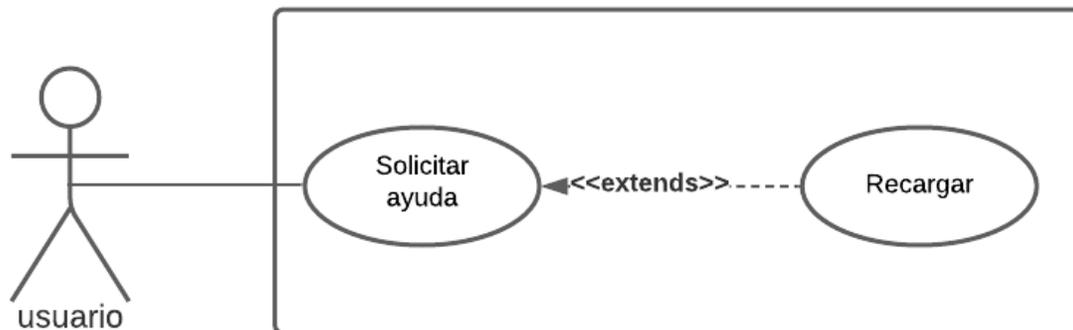
Fuente: elaboración propia.

Figura 12

Diagrama caso de uso: Envío de alertas por WIFI Direct



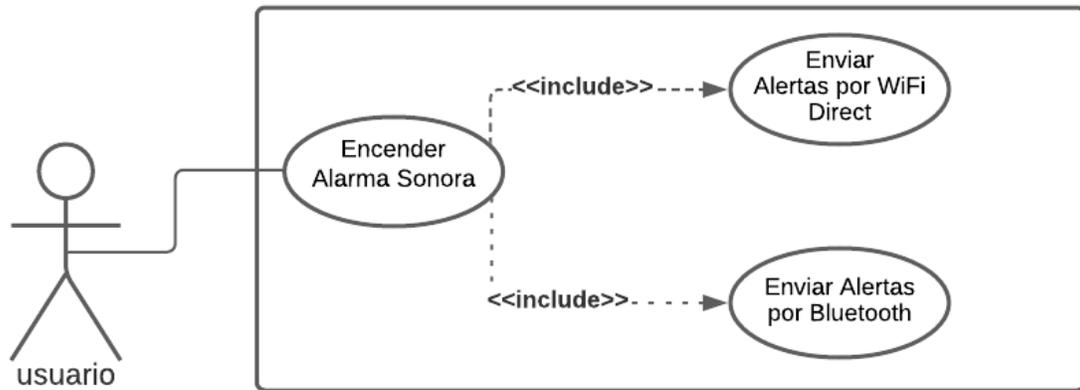
Fuente: elaboración propia

Figura 13 Diagrama caso de uso Recargar pantalla WIFI Direct

Fuente: elaboración propia.

Figura 14

Diagrama de caso de uso Encender alarma sonora



Fuente: elaboración propia.

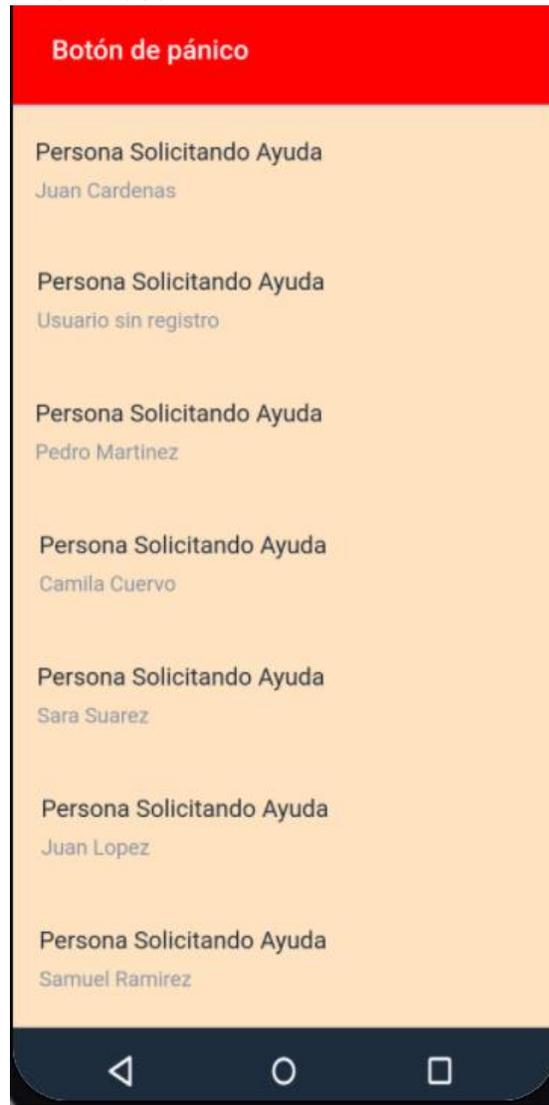
Posteriormente se elaboraron los *Mockups* correspondientes al módulo de WIFI Direct que se pueden observar en las figuras **15** a **17**, esto para planear cómo se iba a mostrar la aplicación al usuario en el momento de pedir ayuda.

Figura 15
Mockup pantalla principal



Fuente: elaboración propia

Figura 16 *Mockup de dispositivos encontrados*



Fuente: elaboración propia

Figura 17*Mockup pantalla de solicitar ayuda*

Fuente: elaboración propia

Una vez el diseño se completó y fué aprobado por el *Product owner*, se procedió a programar el módulo en Android Studio. Fue necesario agregar los permisos en el manifiesto de la aplicación para poder utilizar la tecnología WIFI Direct en los dispositivos. Posteriormente se realizó la interfaz gráfica de la pantalla principal y de la pantalla donde se muestran los dispositivos encontrados, como se observa en las figuras **18** y **19**. Entre los *Mockups* y las interfaces gráficas

de la aplicación, existen algunas diferencias ya que en el proceso de desarrollo, se llegó a la conclusión de que es más práctico incluir el recuadro blanco que muestra la ubicación del usuario y de los teléfonos cercanos, para tener alguna noción de la localización de los nodos vecinos. El listado de los teléfonos cercanos se sigue mostrando pero en la parte inferior de la pantalla. También se decidió que lo más conveniente era colocar el botón Activar Alarma en cada uno de los módulos, para que el usuario lo pueda activar y desactivar cuando lo considere conveniente.

Figura 18
Pantalla principal



Fuente: elaboración propia

Figura 19
Pantalla dispositivos encontrados



Fuente: elaboración propia

En la pantalla principal se encuentran dos botones, uno para cerrar sesión, y otro mediante el cual se solicita ayuda. Al momento de cargar la pantalla, en el método *onCreate* ya descrito en el *sprint* anterior, se carga la vista del xml y posteriormente ejecuta la función *estadoAlarma* que crea una preferencia con la llave “Alarma” y el valor “Inactiva”. Para que el usuario pueda empezar a pedir ayuda debe presionar el botón Solicitar Ayuda, en ese momento la aplicación lo direccionará a la pantalla donde se emite la señal por WIFI Direct y se muestran los dispositivos encontrados. Si el usuario desea cerrar sesión, lo podrá realizar presionando la opción que lleva el mismo nombre desde la pantalla principal y que lo direccionará a la pantalla de inicio de sesión.

Cuando la aplicación carga la pantalla de dispositivos encontrados, se ejecuta la función *inicio* y lo que hace es verificar que los permisos para utilizar la ubicación hayan sido aceptados. Adicionalmente, si el teléfono no tiene activa la ubicación o el adaptador WIFI, la aplicación le solicita prenderlos para continuar.

Una vez se han activado estos dos elementos, la aplicación empieza a emitir señales; si la persona ingresó con credenciales, los paquetes emitidos incluirán su información personal tal como la digitó en el momento del registro, más la ubicación, la fecha y hora en la que se enviaron los paquetes. Si accede por la opción de ingresar sin usuario, en vez de enviar la información personal, la aplicación emitirá el nombre del dispositivo. Simultáneamente, la aplicación estará recibiendo paquetes de otros dispositivos para poder mostrar la información de estos por pantalla. Si la aplicación detecta dispositivos, la información que se muestre de cada uno de ellos dependerá del tipo de ingreso que el usuario haya efectuado, es decir, si ingresó con un usuario registrado o si lo hizo sin registrarse previamente.

El tiempo que la aplicación permanece en el módulo de WIFI Direct se controla mediante la función *iniciarCuenta*, que tiene un temporizador. El tiempo que permanece enviando paquetes se obtiene mediante la función *calcularTiempo*, que genera un número aleatorio entero entre -2 y 2 con distribución normal estándar, por medio de la función *nextGaussian*, obtenida a través de la clase *Random* de Java. Posteriormente, al número aleatorio se le suma 5 y por último se multiplica por 10000; todo este procedimiento da como resultado un número con media 50000, que es un valor en milisegundos, dado que la función *iniciarCuenta* lo requiere en esas unidades. Todo lo anterior implica que el temporizador tendrá una duración de 30, 40, 50 60 70 u 80 segundos. La aplicación cambia de tecnología cada vez que el temporizador llega a cero, es decir que si la aplicación se encuentra en el módulo de WIFI Direct y el temporizador calculó que debe permanecer 50 segundos en ese estado, cambiará a enviar alertas por Bluetooth cuando transcurra ese tiempo; nuevamente el temporizador calculará el tiempo de espera y se devolverá a WIFI Direct y esto lo realizará infinitamente hasta que se cierre la aplicación. El propósito de utilizar este temporizador es asegurar que dos celulares que tengan la aplicación activa, puedan encontrarse cuando ambos estén usando la misma tecnología (WIFI Direct o Bluetooth), en algún momento.

Adicionalmente, la aplicación cuenta con un botón llamado Recargar, cuyo objetivo es actualizar la pantalla de dispositivos encontrados cuando el usuario lo desee, sin necesidad de esperar a que el temporizador llegue a cero.

En esta pantalla también se encuentra un botón que se llama Activar Alarma. Al momento de presionarlo, el dispositivo empieza a emitir un sonido de emergencia y el botón cambiará su

nombre a Desactivar Alarma; al presionarse, dejará de sonar, pero continuará emitiendo la señal via WIFI Direct, ya que un proceso es completamente independiente del otro.

4.4.2.1 Pruebas

Para este segundo *sprint* se hicieron las pruebas correspondientes y los resultados se muestran en las tablas **11** a **13**. Para la realización de estas pruebas se utilizaron, nuevamente, tres dispositivos diferentes: un Samsung J7 Prime con Android 8.1, un Xiaomi Redmi Note 8 con Android 11 y Samsung J5 Prime con Android 8.0.

Tabla 11

Caso de prueba Envío de alerta por WIFI Direct

Nombre del proyecto: BOTÓN DE PÁNICO		Caso No: 3	
Nombre de caso de la prueba: Envío de Alertas por WIFI Direct			
Estado de la prueba: Finalizada		Pantalla de envío de señales por WIFI Direct	
Escrito por: Sergio Silva		Ejecutado por: Andrés Cifuentes	
Descripción del caso de prueba: se valida que el usuario pueda enviar y recibir señales de alerta por medio de WIFI Direct.			
Configuración de la prueba: se requiere de un dispositivo móvil con versión de Android superior a la 5.0, la aplicación instalada y en ejecución.			
Flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Presionar el botón de solicitar ayuda	En la pantalla de botón de pánico, se va a encontrar la opción de solicitar ayuda y lo dirige a la pantalla donde se empieza a buscar los dispositivos por WIFI Direct.	Exitoso
2	El dispositivo empieza a enviar y recibir información	Si hay dispositivos cercanos emitiendo señales, los muestra en la pantalla. Dependiendo del tipo de registro que haya hecho el usuario, la información presentada puede variar.	Exitoso
3	Encender y apagar alarma sonora	El dispositivo emitirá un sonido de alarma y solo se apagará cuando el usuario la desactive.	Exitoso
Excepciones			
1	N/A	N/A	N/A

Fuente: elaboración propia.

Tabla 12*Caso de prueba Cerrar sesión*

Nombre del proyecto: BOTÓN DE PÁNICO		Caso No: 4	
Nombre de caso de la prueba: Cerrar sesión			
Estado de la prueba: Finalizada		módulo de Usuarios	
Escrito por: Andrés Cifuentes		Ejecutado por: Sergio Silva	
Descripción del caso de prueba: se valida que el usuario pueda cerrar sesión.			
Configuración de la prueba: se requiere de un dispositivo móvil con versión de Android superior a la 5.0, la aplicación instalada y en ejecución.			
Flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Ingresar a la aplicación	El usuario ingresa a la aplicación con o sin registro	Exitoso
2	Presionar el botón Cerrar sesión	En la pantalla principal, se va a encontrar el botón de cerrar sesión, al presionarlo debe cerrar la sesión y dirigirlo a la pantalla de iniciar sesión.	Exitoso
Excepciones			
1	N/A	N/A	N/A

Fuente: elaboración propia.

Tabla 13*Caso de prueba Recargar pantalla WIFI Direct*

Nombre del proyecto: BOTÓN DE PÁNICO		Caso No: 5	
Nombre de caso de la prueba: Recargar pantalla WIFI Direct			
Estado de la prueba: Finalizada		Pantalla de envío de señales por WIFI Direct	
Escrito por: Sergio Silva		Ejecutado por: Andrés Cifuentes	
Descripción del caso de prueba: se valida que el rol de usuario pueda recargar la pantalla.			
Configuración de la prueba: se requiere de un dispositivo móvil con versión de Android superior a la 5.0, la aplicación instalada y en ejecución.			
Flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Presionar el botón Recargar	La pantalla se recargará reiniciando el envío de alertas y actualizando la lista de los dispositivos encontrados	Exitoso
Excepciones			
1	N/A	N/A	N/A

Fuente: elaboración propia.

4.4.3 *Sprint 3: módulo de envío de alertas por Bluetooth e integración final*

En este *sprint* se desarrolló el módulo de envío de alertas por Bluetooth. Se hizo la investigación para implementar esta tecnología en la aplicación pero la información encontrada estaba escrita en el lenguaje de programación de Java. Con el fin de pasarla al lenguaje Kotlin, se realizó una adaptación de la información encontrada, sin embargo en unas de las pruebas se observó que a pesar de que el envío de beacons funcionaba, no sucedía lo mismo con la detección. Es por esta limitación que se decidió desarrollar este módulo en Java, aunque el resto de funcionalidades sí se implementaron en Kotlin.

Un aspecto importante para el proyecto de investigación y que se adicionó en este *sprint* fue la posibilidad de alternar el envío de alertas entre una tecnología y la otra. Una vez el usuario

ingrese a la aplicación con o sin credenciales, podrá usar el módulo de WIFI Direct o el de Bluetooth según lo desee.

Teniendo en cuenta lo anterior, las historias de usuario que conforman el tercer *sprint* son *Enviar alertas por Bluetooth, Permanecer en la tecnología Bluetooth o WIFI Direct y Cambiar de tecnología*. Los detalles de cada historia se pueden observar en las tablas **14 a 16**.

Tabla 14

Historia de usuario 7: Enviar alertas por Bluetooth

Indicador:HU07	Indispensable	Prioridad: alta
Nombre de Caso de Uso:	Enviar alertas por Bluetooth	
Autor:	Andrés Cifuentes	
Fecha:	02 de octubre de 2021	
Categoría (Visible):	Actores Involucrados: Usuario	
Resumen:	Al finalizar el temporizador que controla el envío de señales por WIFI Direct, se pasa a la pantalla que activa la emisión de balizas desde Bluetooth.	
Curso Básico Eventos	<ol style="list-style-type: none"> 1. El usuario selecciona la opción Solicitar Ayuda. 2. El sistema lo direcciona a la pantalla de envío de señales vía WIFI Direct. 3. Cuando termina el temporizador de esa pantalla, pasa a la pantalla de Bluetooth. 	
Caminos Alternativos	Si el teléfono tiene apagado el adaptador Bluetooth, la aplicación le solicita prenderlo.	
Caminos de Excepción	No se podrá emitir señales si no aceptan los permisos al inicio de la aplicación para el uso de los módulos.	
Puntos de Extensión		
Pre - Condiciones	HU03	
Post - Condiciones	Después de activar la opción, el sistema lo direcciona a la interfaz donde será visible que la señal está activa y los dispositivos están disponibles	
Criterios de Aceptación	Debe cargar la interfaz gráfica donde empieza a emitir información vía Bluetooth y, si hay dispositivos cercanos que están enviando balizas, estos deben aparecer en pantalla, así como la distancia de separación con cada uno de ellos.	

Fuente: elaboración propia.

Tabla 15*Historia de usuario 8: Permanecer en la tecnología Bluetooth o WIFI Direct*

Indicador:HU08	Indispensable	Prioridad: alta
Nombre de Caso de Uso:	Permanecer en la tecnología Bluetooth o WIFI Direct	
Autor:	Sergio Silva	
Fecha:	02 de octubre de 2021	
Categoría (Visible):	Actores Involucrados: Usuario	
Resumen:	Al cargarse la pantalla de envío de alertas en cualquiera de las dos tecnologías, el botón Permanecer aparece etiquetado con la palabra NO. Si el usuario presiona el botón, el sistema lo cambiará a SI para que se detenga la alternancia entre una y otra tecnología.	
Curso Básico Eventos	<ol style="list-style-type: none"> 1. El usuario presiona el botón cuando se encuentra en NO. 2. El sistema lo cambiará a SI y no alternará las tecnologías. 	
Caminos Alternativos	N/A	
Caminos de Excepción	El Botón volverá a cambiarse a NO cuando se recargue la pantalla de la tecnología donde se esté.	
Puntos de Extensión		
Pre - Condiciones	HU02	
Post - Condiciones	Después de activar la opción, el sistema no debe alternar las tecnologías a menos que se vuelva a cambiar el botón a NO.	
Criterios de Aceptación	Debe cargar la interfaz gráfica donde aparezca el botón Permanecer y poder dar click en él para que cambie su estado a SI y se detenga la operación alterna entre una tecnología y la otra.	

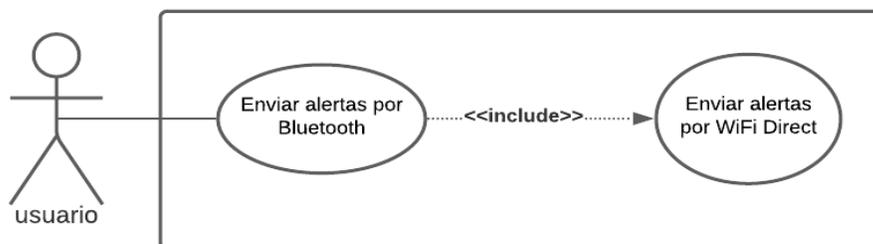
Fuente: elaboración propia.

Tabla 16*Historia de usuario 9: Cambiar de tecnología*

Indicador:HU09	Indispensable	Prioridad: alta
Nombre de Caso de Uso:	Cambiar de tecnología	
Autor:	Andrés Cifuentes	
Fecha:	02 de octubre de 2021	
Categoría (Visible):	Actores Involucrados: Usuario	
Resumen:	El usuario podrá cambiar la tecnología de búsqueda cuando lo prefiera: si está en Bluetooth, pasa a WIFI Direct, y viceversa.	
Curso Básico Eventos	<ol style="list-style-type: none"> 1. El usuario presiona el botón que tiene el logo de la tecnología a la cual se quiere cambiar. 2. El sistema lo direccionará a la tecnología seleccionada para enviar y recibir alertas. 	
Caminos Alternativos	N/A	
Caminos de Excepción	N/A	
Puntos de Extensión	N/A	
Pre - Condiciones	HU02	
Post - Condiciones	Después de usar la opción, el sistema debe cambiar de tecnología las veces que sean requeridas por el usuario	
Criterios de Aceptación	Debe cargar la interfaz gráfica de cualquiera de las dos tecnologías utilizadas, allí debe aparecer el botón con el logo de la otra tecnología disponible, y podrá ir de una tecnología a otra las veces que sean requeridas.	

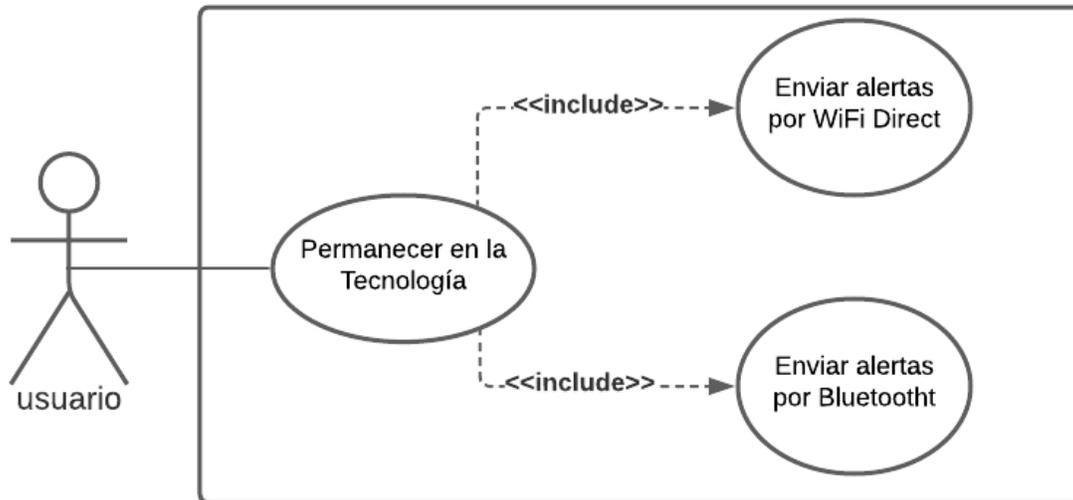
Fuente: elaboración propia.

El siguiente paso fue elaborar los diagramas de casos de uso que se presentan en las figuras 20 a 22.

Figura 20*Diagrama de caso de uso Enviar alertas por Bluetooth*

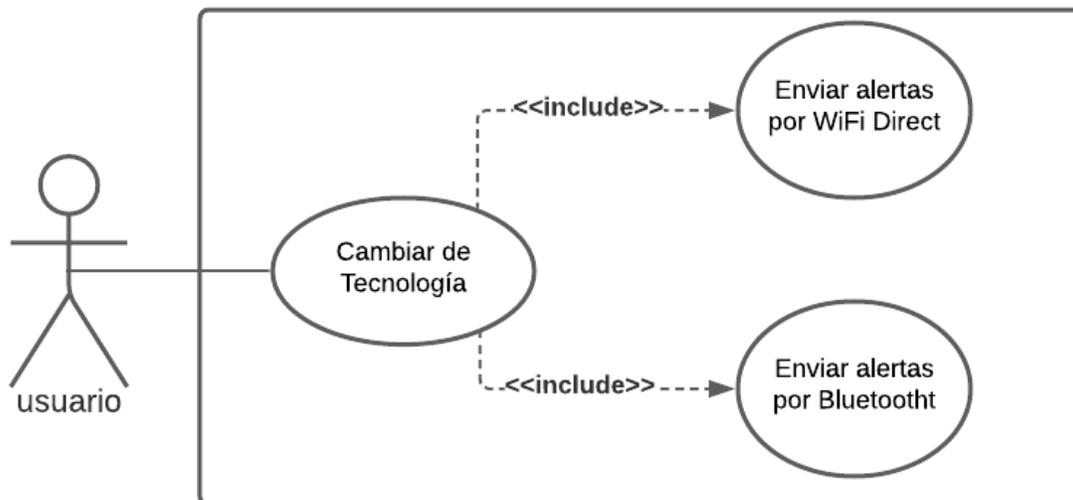
Fuente: elaboración propia

Figura 21
Diagrama de caso de uso Permanecer en la tecnología



Fuente: elaboración propia

Figura 22
Diagrama de caso de uso Cambiar de tecnología



Fuente: elaboración propia

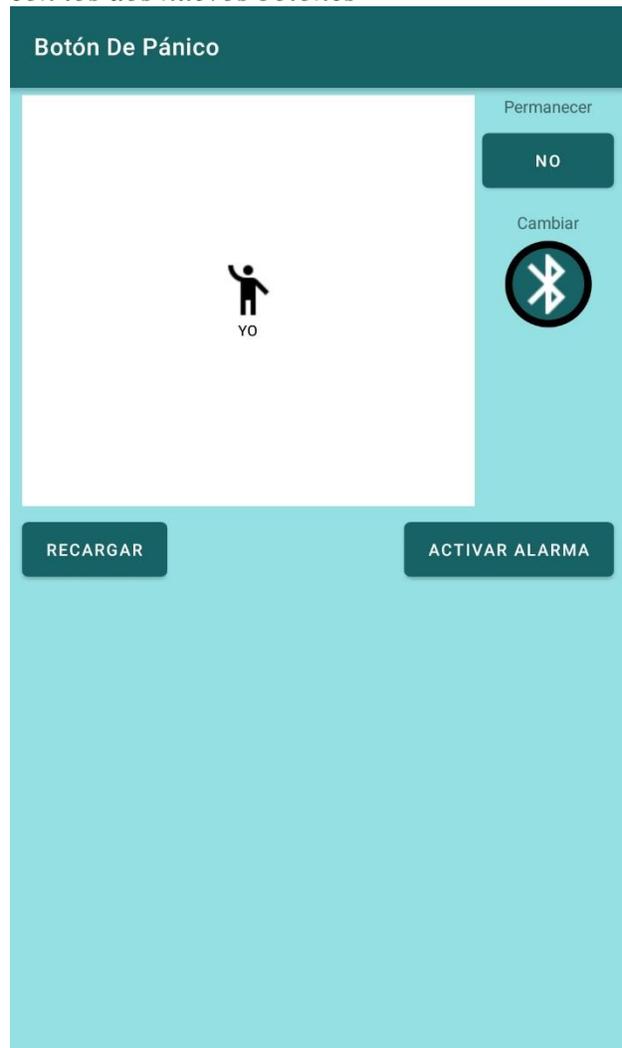
Para el desarrollo se comenzó construyendo la interfaz gráfica del módulo, como se puede observar en la figura 23. Una vez el diseño se completó, se procedió a programar el módulo en Android Studio. Fue necesario agregar los permisos en el manifiesto de la aplicación para poder utilizar la tecnología.

Figura 23
Pantalla de dispositivos encontrados por Bluetooth



Fuente: elaboración propia

Figura 24
Pantalla de dispositivos encontrados por WIFI con los dos nuevos botones



Fuente: elaboración propia

Al iniciar la actividad para descubrir dispositivos por medio de Bluetooth, lo primero que hace el código es ingresar al método *onCreate*, ahí se carga la vista del xml que se diseñó previamente, y luego se ejecuta el método *checkPermission*, el cual valida que los permisos para usar el adaptador Bluetooth hayan sido aceptados por el usuario cuando fueron solicitados; de lo contrario, le pedirá nuevamente que acepte los permisos. Siendo exitosa la validación, enciende el adaptador Bluetooth mediante el método *encenderBluetooth*.

Posteriormente es necesario hacer la configuración del beacon mediante el método *setupBeacon*, después se ejecuta el método *transmitBeacon* el cual empezará a emitir balizas, para que los teléfonos cercanos que hagan uso de esta opción puedan ver que hay una persona solicitando ayuda.

Luego se inicia el descubrimiento de los dispositivos Bluetooth, se empieza configurando la estructura del beacon que se va a recibir y, posteriormente, mediante el método *beaconManager.bind* junto con *onBeaconServiceConnect*; en este método se captura la información del beacon y se llaman dos funciones, *getBluetoothAddress*, que se encarga de obtener la dirección MAC para ese dispositivo detectado y *getDistance*, que teniendo en cuenta el RSSI *received signal strength indicator* (indicador de intensidad de la señal recibida), que es el valor de la señal medida en decibeles y la potencia de la transmisión, realiza el cálculo de la distancia aproximada entre el dispositivo que envía el beacon y el dispositivo que lo recibe. Si por algún motivo la función no detecta la dirección MAC del dispositivo, se llama el método *macAleatoria*, el cuál retorna un número de 12 dígitos obtenido aleatoriamente con distribución normal. Adicionalmente, cuenta con las funciones de alarma sonora y el temporizador en donde se hace uso de los métodos mencionados en el anterior *sprint*.

El dispositivo seguirá haciendo el descubrimiento hasta que la aplicación alterne de tecnología o cuando el usuario force el cambio a WIFI Direct.

El módulo de Bluetooth cuenta con el botón de recargar cuya función es volver a cargar la pantalla de dispositivos encontrados en este módulo y así poder encontrar los nuevos dispositivos que estén emitiendo señales. Adicionalmente también tiene el botón de activar alarma el cual emite un sonido de emergencia y solo dejará de sonar cuando el usuario la desactive.

Con el módulo de WIFI Direct y Bluetooth ya desarrollados, se implementaron funcionalidades adicionales para las dos tecnologías. Teniendo el temporizador programado en el módulo de WIFI Direct, se reutilizó para que se pudiera alternar el envío de alertas por WIFI Direct y por Bluetooth de manera automática. También se diseñaron dos botones que se pueden usar en ambas funcionalidades, como se observa en las figuras **23** y **24**. El primero es para poder permanecer en la tecnología que se está utilizando en el momento, este botón siempre traerá por defecto la etiqueta NO y cada vez que se recargue la pantalla volverá a NO, si se ha utilizado esta función. El segundo botón permite cambiar de tecnología de manera manual sin necesidad de que el temporizador llegue a cero para que alterne de manera automática.

4.4.3.1 Pruebas

Para este tercer *sprint* se hicieron las pruebas correspondientes, mostrando los resultados obtenidos al momento de pedir ayuda y emitir señales por Bluetooth, como se muestra en las tablas **17** y **18**. Para la realización de estas pruebas, una vez más, se utilizaron tres dispositivos: un Xiaomi MI A3 con Android 11, un Xiaomi Redmi Note 8 con Android 11 y Samsung J5 Prime con Android 8.0.

Tabla 17*Caso de prueba Envío de alerta por Bluetooth*

Nombre del proyecto: BOTÓN DE PÁNICO		Caso No: 6	
Nombre de caso de la prueba: enviar alertas por Bluetooth			
Estado de la prueba: Finalizada		Pantalla de dispositivos encontrados por Bluetooth	
Escrito por: Andrés Cifuentes		Ejecutado por: Sergio Silva	
Descripción del caso de prueba: se valida que el usuario pueda enviar y recibir alertas por Bluetooth			
Configuración de la prueba: se requiere de un dispositivo móvil con versión de Android superior a la 5.0, la aplicación instalada y en ejecución.			
Flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Presionar el botón de solicitar ayuda	En la pantalla principal, tocar el botón Solicitar ayuda. La aplicación se dirige a la pantalla de WIFI Direct, esperar a que el temporizador llegue a cero y que cambie a la tecnología de Bluetooth.	Exitoso
2	El dispositivo empieza a enviar y recibir información	Envía balizas que pueden capturar los dispositivos cercanos. Si hay dispositivos emitiendo señales, los muestra en la pantalla con el cálculo de la distancia, la dirección MAC y la hora del envío de la señal. Si no es posible obtener la dirección MAC, se crea un número aleatorio de 12 dígitos que posteriormente se envía y queda como identificador de ese dispositivo	Exitoso
Excepciones			
1	N/A	N/A	N/A

Fuente: elaboración propia.

Tabla 18*Caso de prueba Cambio de tecnología*

Nombre del proyecto: BOTÓN DE PÁNICO		Caso No: 7	
Nombre de caso de la prueba: Cambio de tecnología			
Estado de la prueba: Finalizada		Pantalla de dispositivos encontrados por Bluetooth	
Escrito por: Sergio Silva		Ejecutado por: Andrés Cifuentes	
Descripción del caso de prueba: se valida que el usuario pueda cambiar de tecnología cuando lo requiera			
Configuración de la prueba: se requiere de un dispositivo móvil con versión de Android superior a la 5.0, la aplicación instalada y en ejecución.			
Flujo de eventos:			
Paso	Acción	Resultados esperados	Exitoso/Fallido
1	Presionar el botón cambiar de tecnología  (Bluetooth)	En la pantalla de dispositivos encontrados de WIFI Direct se encuentra el botón cambiar de tecnología y debe cambiar a la tecnología de Bluetooth	Exitoso
2	Presionar el botón cambiar de tecnología  (WIFI)	En la pantalla de dispositivos encontrados de Bluetooth se encuentra el botón cambiar de tecnología y debe cambiar a la tecnología de WIFI Direct	Exitoso
Excepciones			
1	N/A	N/A	N/A

Fuente: elaboración propia.

5. Análisis y resultados

En este capítulo se evidencia el cumplimiento de cada uno de los objetivos planteados para el trabajo de grado.

5.1 Requerimientos de la aplicación Botón de Pánico

El primer objetivo específico se refiere al levantamiento de los requerimientos de la aplicación. Los requerimientos fueron manifestados por la directora del proyecto al equipo de desarrollo al inicio del proyecto y en el capítulo anterior se describen las actividades que se realizaron para su cumplimiento.

Concretamente, los requerimientos se listan a continuación:

- La aplicación debe operar en celulares con sistema operativo Android.
- La aplicación debe emitir señales mediante las tecnologías WIFI Direct y Bluetooth, utilizando las interfaces que el teléfono trae embebidas.
- La aplicación debe tener un módulo de registro de usuario que permita obtener los datos básicos de la persona que la va a utilizar y que se van a difundir a los usuarios cercanos en caso de una emergencia.
- Si es posible, mediante estas señales debe transmitir los datos capturados durante el registro del usuario y debe mostrar un dato aproximado de la distancia que separa los dos móviles.

- La aplicación debe mostrar los teléfonos que estén corriendo la misma aplicación y que se encuentran dentro del radio de cobertura del mismo, que es aproximadamente de 100 m según datos de la directora del proyecto.
- La aplicación debe ser de fácil uso.
- Si es posible, y es parte de los aspectos que deben determinarse durante el desarrollo, debe trabajar alternadamente la búsqueda de dispositivos por WIFI Direct y por Bluetooth. Si eso no es posible, entonces debe implementarse un módulo aparte para WIFI Direct y otro para Bluetooth, y el usuario escogerá cuál usar según lo que desee.

5.2 Diseño de la base de datos, diagramas UML e historias de usuario.

En el segundo objetivo se hace referencia a diseñar la base de datos y casos de uso, así como las historias de usuario. Se puede observar en el capítulo 4 que para el primer *sprint* se diseñó la base de datos en Firebase, específicamente en la figura 1 se muestra la tabla en donde se almacenan los datos que se le piden al usuario para hacer el registro en la aplicación. También se elaboró el diagrama de clases resultado del diseño del programa, como se observa en la figura 9 del capítulo anterior. Además de esto, se realizaron los diagramas de casos de uso específicos para cada uno de los módulos de la aplicación, así como para la integración de los mismos.

En cuanto a las historias de usuario, estas se detallan en el capítulo 4, indicando cuáles se desarrollaron en cada uno de los *sprints*. Cada historia de usuario describe a qué funcionalidad hace referencia, presenta un breve resumen, el curso básico de eventos que se debe ejecutar para observar la funcionalidad y sus criterios de aceptación.

Este objetivo se cumple en su totalidad teniendo todo el diseño y explicación ya mencionado.

5.3 Programación de todos los módulos de la aplicación

El tercer objetivo hace referencia a la programación de todos los módulos mediante los lenguajes Kotlin y Java. En el capítulo anterior se explicó cómo se llevó a cabo cada uno de los *sprints* del proyecto, las funcionalidades que se implementaron en cada uno de ellos y como quedó conformado cada módulo, así como los métodos y funciones que se usaron para el funcionamiento correcto de cada tecnología. En las figuras **25** a **28**, se pueden observar fragmentos del código que se implementó para los módulos WIFI Direct y Bluetooth, incluyendo las funcionalidades adicionales como la alarma sonora, permanecer en una tecnología, recargar la información de los dispositivos descubiertos y poder alternar entre los dos módulos. Este objetivo se cumplió en su totalidad.

Figura 25
Fragmento de código del módulo de WIFI Direct

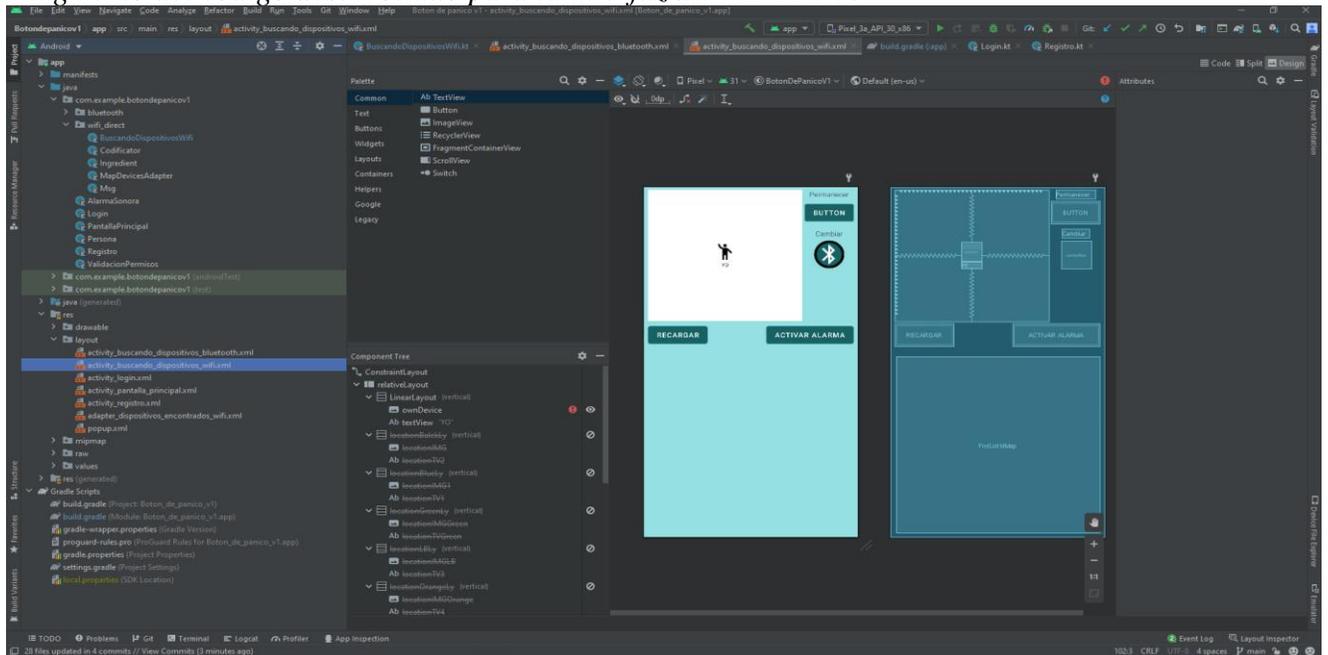
```

91 //Alarma
92 private lateinit var playPauser : Button
93 private lateinit var mp : MediaPlayer
94 private val alarma = AlarmaSonora()
95
96 //Nombre Dispositivo
97 private lateinit var nombreDispositivo: String
98
99 @RequiresApi(Build.VERSION_CODES.JELLY_BEAN)
100 override fun onCreate(savedInstanceState: Bundle?) {
101     super.onCreate(savedInstanceState)
102     setContentView(R.layout.activity_buscando_dispositivos_wifi)
103     sharedPreferences = getSharedPreferences(PREFERENCES_DATE, AppCompatActivity.MODE_PRIVATE)
104     bm = getSystemService(AppCompatActivity.BATTERY_SERVICE) as BatteryManager
105
106     val ifilter = IntentFilter(Intent.ACTION_BATTERY_CHANGED)
107     val batteryStatus = this.registerReceiver(receiver, null, ifilter)
108     val level = batteryStatus!!.getExtra(BatteryManager.EXTRA_LEVEL, 0)
109     val scale = batteryStatus.getExtra(BatteryManager.EXTRA_SCALE, 0)
110     val nivel = (level * 100 / scale).toInt()
111
112     playPauser = findViewById(R.id.)
113     mp = MediaPlayer.create(context, this.R.raw.alarma_sonora)
114
115     val prefs = PreferenceManager.getDefaultSharedPreferences(context)
116     Log.d(tag, "Sergio", msg, "INICIO ACTIVIDAD" + prefs.getString(keyAlarma, defValue) + ".toString()")
117
118     nombreDispositivo = "Dispositivo: ${Build.MANUFACTURER.toUpperCase(Locale.ROOT)} ${Build.MODEL}"
119
120     alarma.estadoPreferencia(playPauser, context, this)
121     inicio()
122     iniciarCuenta()
123     Log.d(tag, "Sergio", msg, "fin")
124 }
125
126 override fun onBackPressed() {
127     terminarActividad()
128     alarma.abajarFinActividad(playPauser, context, this)
129     Log.d(tag, "Sergio", msg, "terminarActividad = $terminarActividad")
130     finish()
131 }

```

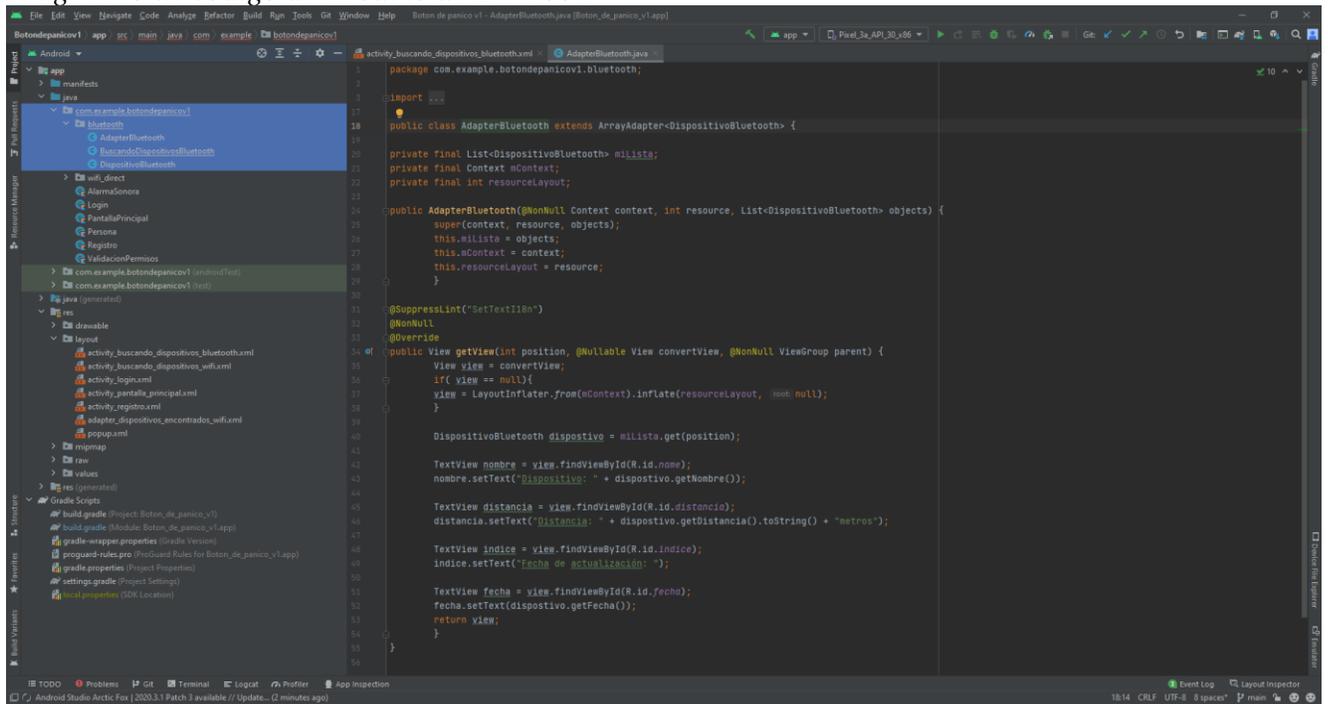
Fuente: elaboración propia.

Figura 26
Fragmento del código desarrollado para la interfaz de WIFI Direct



Fuente: elaboración propia.

Figura 27
Fragmento de código del módulo de Bluetooth



```

package com.example.botondepanicov1.bluetooth;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import java.util.List;

public class AdapterBluetooth extends ArrayAdapter<DispositivoBluetooth> {

    private final List<DispositivoBluetooth> miLista;
    private final Context mContext;
    private final int resourceLayout;

    public AdapterBluetooth(@NonNull Context context, int resource, List<DispositivoBluetooth> objects) {
        super(context, resource, objects);
        this.miLista = objects;
        this.mContext = context;
        this.resourceLayout = resource;
    }

    @SuppressWarnings("SetTextI18n")
    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
        View view = convertView;
        if (view == null) {
            view = LayoutInflater.from(mContext).inflate(resourceLayout, null);
        }

        DispositivoBluetooth dispositivo = miLista.get(position);

        TextView nombre = view.findViewById(R.id.nombre);
        nombre.setText("@Dispositivo: " + dispositivo.getNombre());

        TextView distancia = view.findViewById(R.id.distancia);
        distancia.setText("@Distancia: " + dispositivo.getDistancia().toString() + " metros");

        TextView indice = view.findViewById(R.id.indice);
        indice.setText("Fecha de actualización: ");

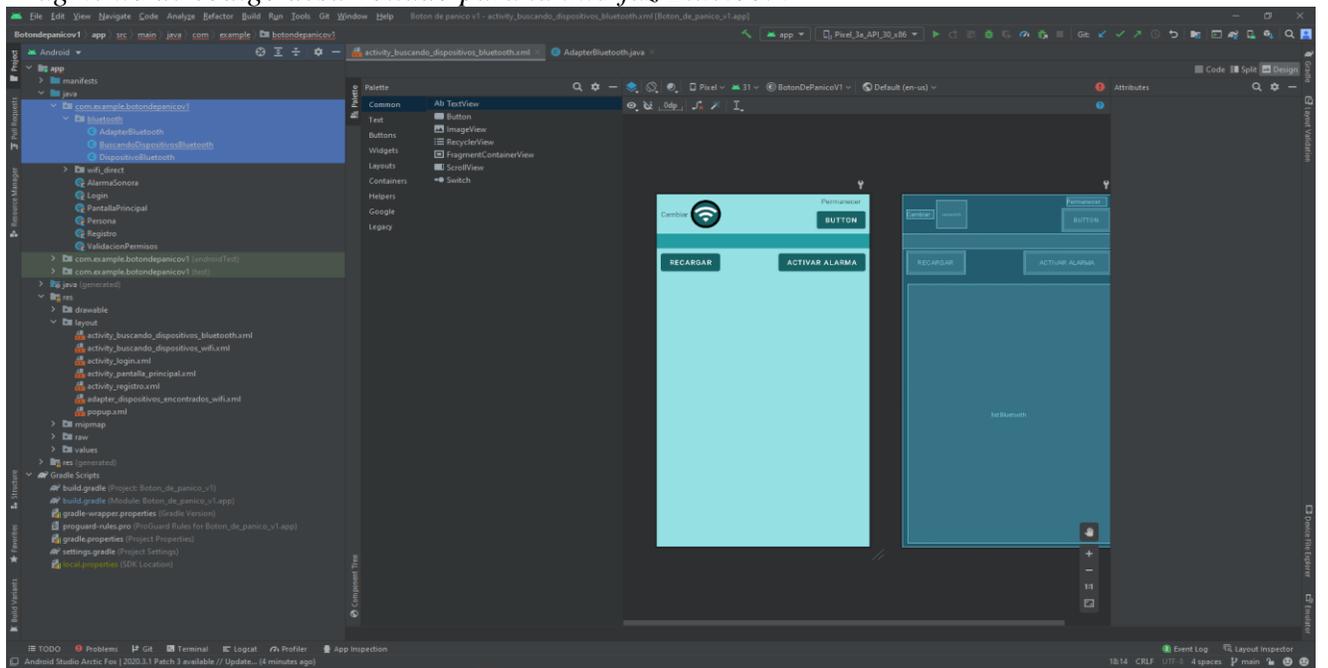
        TextView fecha = view.findViewById(R.id.fecha);
        fecha.setText(dispositivo.getFecha());

        return view;
    }
}

```

Fuente: elaboración propia.

Figura 28
Fragmento de código desarrollado para la interfaz Bluetooth



Fuente: elaboración propia.

5.4 Pruebas de la aplicación en campo

El cuarto y último objetivo hace referencia a los resultados de las pruebas de campo de la aplicación. Las dos tecnologías y sus funcionalidades fueron probadas en campo abierto y campo cerrado como se describe a continuación.

5.4.1 Prueba de registro

La aplicación botón de pánico cuenta con almacenamiento en Firebase, para poder hacer el registro y la autenticación. Se realizaron múltiples registros para comprobar que se hicieran efectivamente y que persistiera la información de los usuarios en la base de datos. También se validó que al momento de finalizar el registro informara si existe algún inconveniente o dato faltante.

Del proceso realizado, se obtuvieron los siguientes resultados. En las figuras **29** a **30** se puede observar que el módulo de registro informa si hay datos faltantes, o si un documento ya se ha registrado anteriormente.

Figura 29
Pantalla de registro con datos faltantes

The screenshot shows a registration form titled "REGISTRO" with the following fields and error messages:

- Número de documento:** "Ingrese su número de documento" (red error icon)
- Nombres:** "Ingrese sus nombres" (red error icon)
- Apellidos:** "Ingrese sus apellidos" (red error icon)
- Género:** "Seleccione" (red error icon)
- Tipo de sangre:** "Seleccione" (red error icon)
- Factor:** "Seleccione una opción" (red error icon)
- Fecha de nacimiento:** "Ingrese su fecha de nacimiento" (red error icon)
- Contraseña:** "Ingrese la contraseña" (red error icon)
- Confirmar contraseña:** "Ingrese la contraseña" (red error icon)

A "REGISTRAME" button is located at the bottom of the form.

Fuente: elaboración propia.

Figura 30
Mensaje que devuelve la aplicación si el usuario ya se ha registrado anteriormente.

The screenshot shows the same registration form as in Figure 29, but with an error message overlay:

Error
Ya se ha registrado anteriormente con este documento

An "ACEPTAR" button is visible on the right side of the error message. The "REGISTRAME" button is still visible at the bottom.

Fuente: elaboración propia.

Si el registro es exitoso, también se evidencia el mensaje que confirma que ese proceso se llevó a cabo correctamente y que el usuario podrá posteriormente, iniciar sesión.

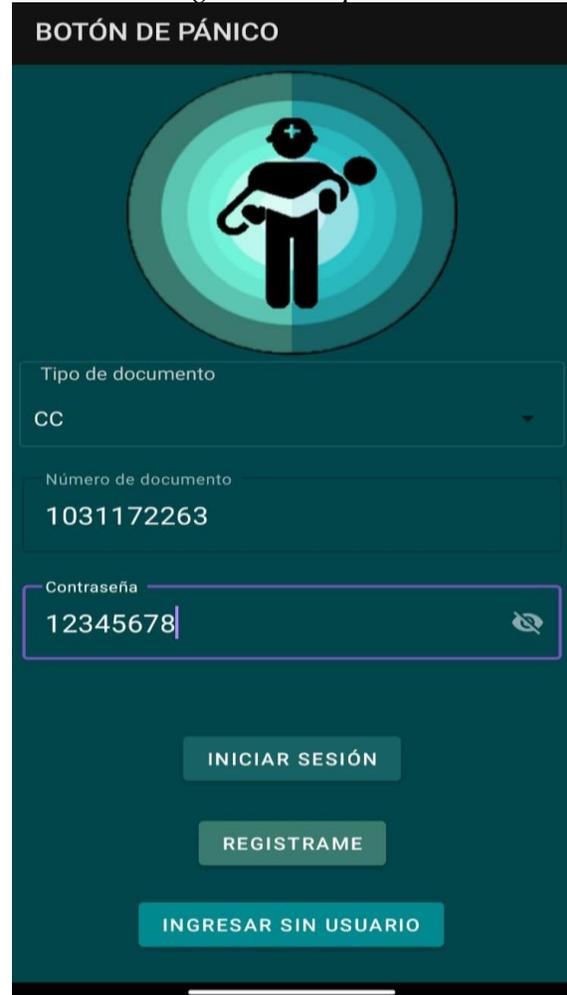
Figura 31
Mensaje de registro exitoso



The screenshot shows a registration form titled "REGISTRO" with the following fields: "Nombres" (Andrés felipe), "Apellidos" (Cifuentes Mensñdiet), "Género" (M), "Fecha de nacimiento" (26 / 10 / 1997), "Contraseña" (masked with dots), and "Confirmar contraseña" (masked with dots). A "REGISTRAME" button is at the bottom. A modal dialog box is displayed in the center with the text "Registro Exitoso" and "Gracias por confiar en nosotros", with an "ACEPTAR" button.

Fuente: elaboración propia.

Figura 32
Pantalla de ingreso a la aplicación



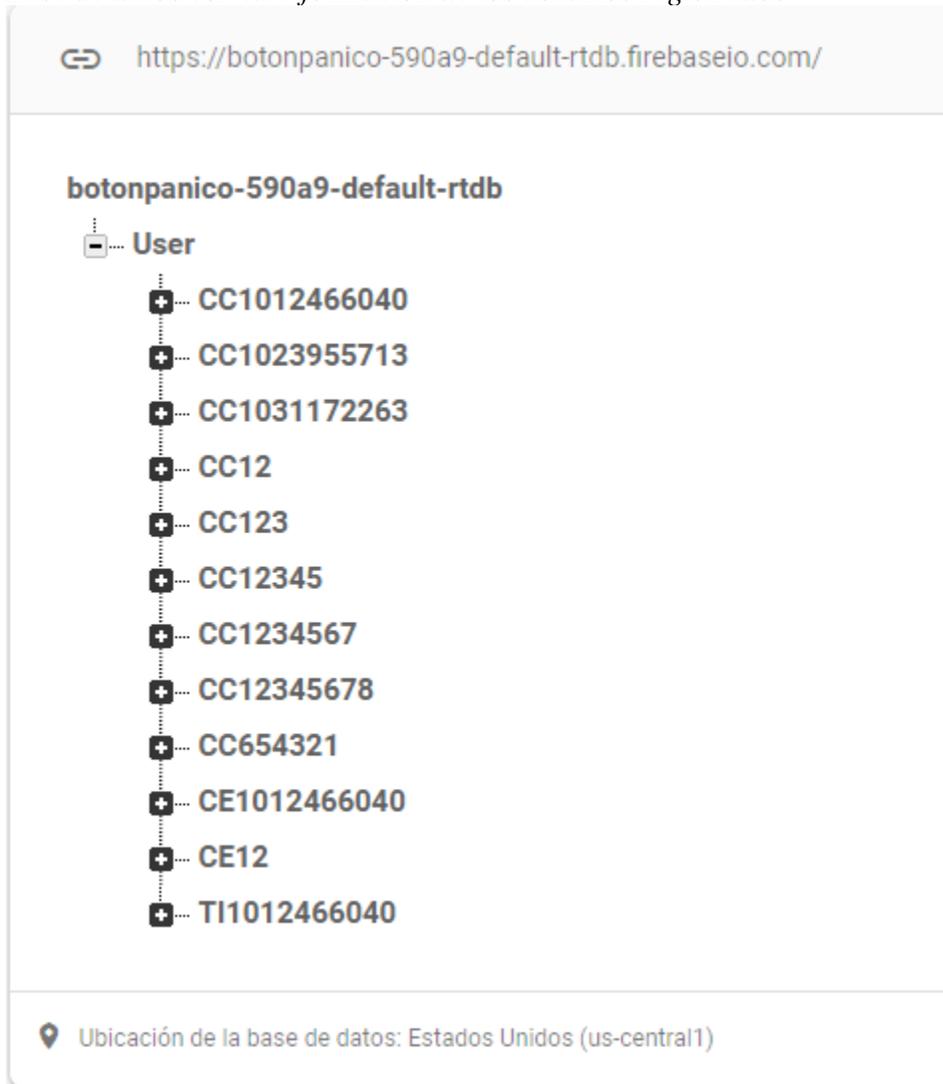
The screenshot shows a login screen titled "BOTÓN DE PÁNICO" with a central icon of a person holding a baby. Below the icon are three input fields: "Tipo de documento" (CC), "Número de documento" (1031172263), and "Contraseña" (12345678). At the bottom, there are three buttons: "INICIAR SESIÓN", "REGISTRAME", and "INGRESAR SIN USUARIO".

Fuente: elaboración propia.

En la base de datos en Firebase se puede comprobar que los datos del registro fueron almacenados correctamente.

Figura 33

Base de datos con la información de los usuarios registrados



Fuente: elaboración propia.

5.4.2 Solicitar ayuda estando registrado en la aplicación.

Se hicieron las pruebas para solicitar ayuda mediante WIFI Direct y Bluetooth, como se observa en las figuras 34 a 37, teniendo una sesión iniciada en la aplicación. La aplicación funciona alternadamente con estas dos tecnologías. El tiempo que permanece en cada una de ellas es aleatorio, con el fin de evitar que los dispositivos no se detecten cuando simultáneamente operan

en tecnologías diferentes. Los teléfonos utilizados en las pruebas fueron un Xiaomi Mi A3 y un Xiaomi Note 8.

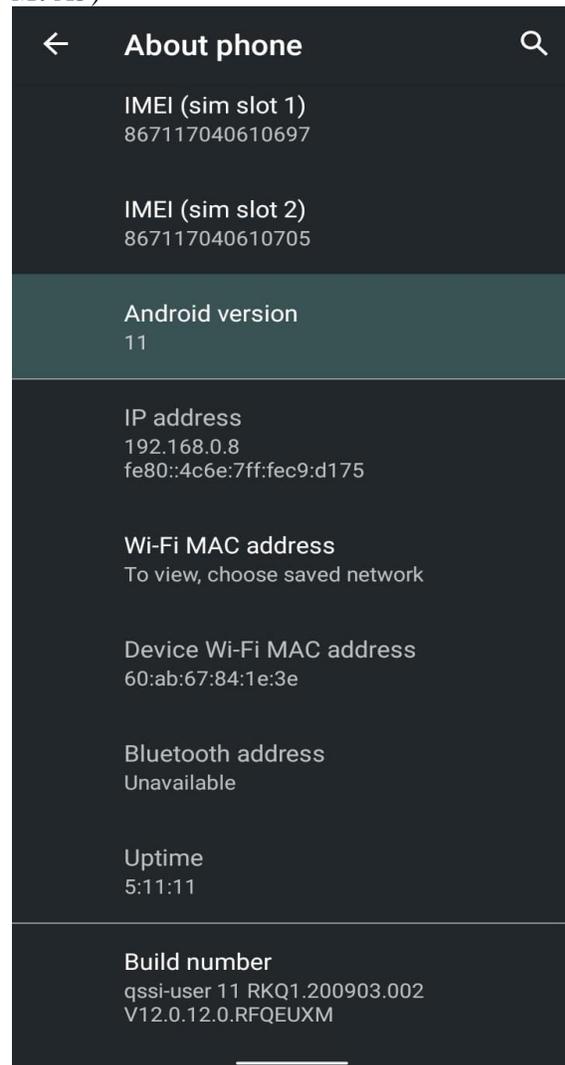
En la figura 34 se presenta el resultado de la detección con el dispositivo Xiaomi Mi A3 con sistema operativo Android 11 (figura 35).

Figura 34 Pantalla de dispositivos encontrados por WIFI Direct (Xiaomi Mi A3)



Fuente: elaboración propia.

Figura 35 Información del dispositivo (Xiaomi Mi A3)



Fuente: elaboración propia.

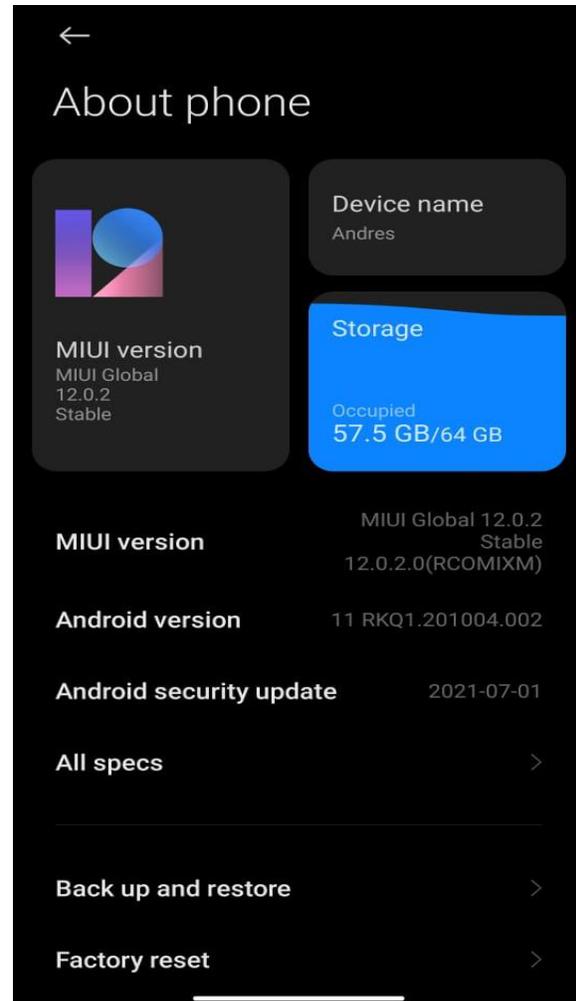
La figura 36 muestra el resultado de la detección con el dispositivo Xiaomi Note 8, que también tiene sistema operativo Android 11.

Figura 36
Resultados de dispositivos encontrados por WIFI Direct (Xiaomi Note 8)



Fuente: elaboración propia.

Figura 37
Información del dispositivo (Xiaomi Note 8)

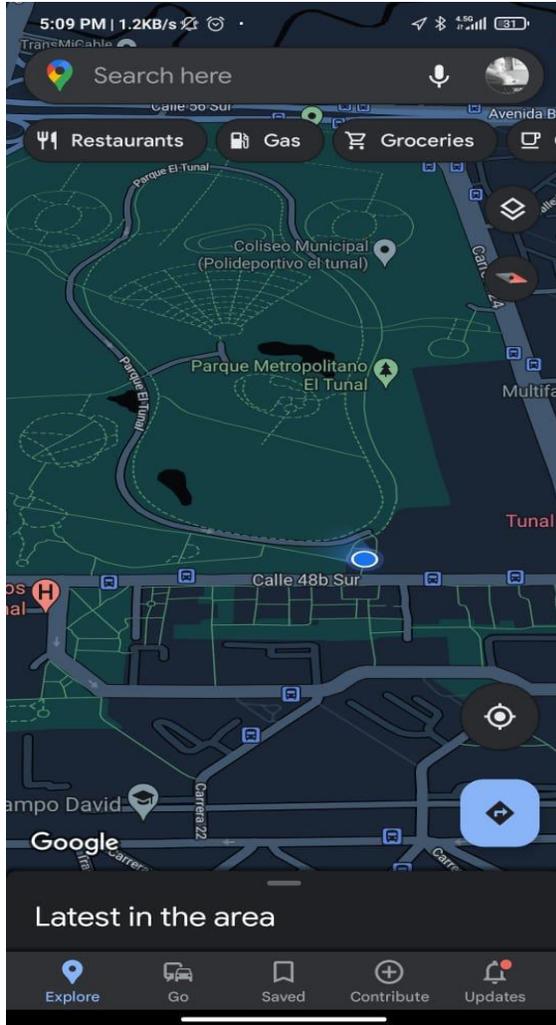


Fuente: elaboración propia.

Estos resultados se obtuvieron en una prueba realizada en el parque el Tunal de la ciudad de Bogotá, cuya localización en la figura 38 Los dispositivos estaban a una distancia de 15 metros (figura 39), donde la distancia obtenida con el módulo WIFI Direct por el dispositivo Xiaomi Mi A3 fue de 13 metros, , en tanto que la estimada por el Xiaomi Note 8 fué de 10 metros. En las diferentes pruebas realizadas los resultados variaban aproximadamente en 5 metros, lo cual para este ejemplo se ve reflejado.

Figura 38

Mapa de la ubicación de las pruebas



Fuente: elaboración propia.

Figura 39

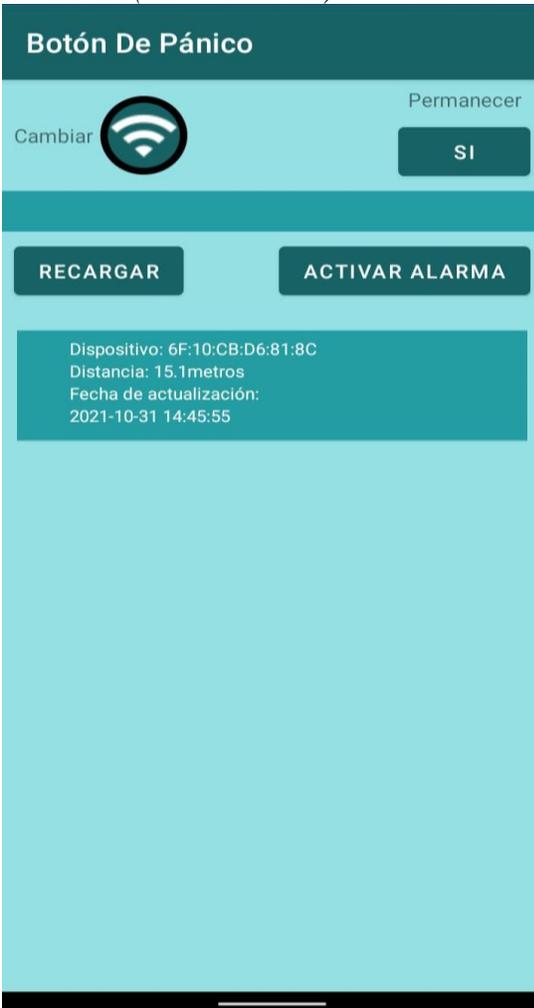
Diagrama de distancia entre los dispositivos en el parque Tunal



Fuente: elaboración propia.

En las figuras **40** y **41** se observan las pruebas que se hicieron en el módulo de Bluetooth en el parque el Tunal, en la misma disposición que se utilizó en las pruebas con WIFI Direct. Los dispositivos se encontraban a una distancia de 15 metros, donde la distancia calculada por el Xiaomi Mi A3 fue de 15.1 metros y la del Xiaomi Note 8 fue de 13.02 metros.

Figura 40
Pantalla de dispositivos encontrados por Bluetooth (Xiaomi Mi A3)



Fuente: elaboración propia.

Figura 41
Pantalla de dispositivos encontrados por Bluetooth (Xiaomi Note 8)



Fuente: elaboración propia.

Se puede determinar que la distancia calculada por Bluetooth es más exacta que la que se calculó mediante WIFI Direct en distancias menores a 20 metros. Aunque no están las imágenes de estas pruebas en distancias superiores a 20 metros, en los distintos experimentos realizados con ambos módulos, WIFI Direct parece tener un mayor rango de acción pues detecta dispositivos entre 0 y 45 metros de distancia.

Después de terminar este tipo de prueba, se hizo otra donde uno de los dispositivos se encontraba en la calle y otro en un tercer piso. El objetivo de esta prueba era probar la conexión, teniendo en cuenta los obstáculos e interferencia de otras señales y la distancia aproximada entre ellos era de 12 metros, como se observa en la figura 42. Los dispositivos se detectaron en ambas tecnologías.

Figura 42

Distancia de los dispositivos uno en un edificio y otro a campo abierto



Fuente: elaboración propia

Los resultados encontrados en las pruebas para el módulo de WIFI Direct se presentan en las figuras 43 y 44. Se observa que la distancia calculada por el dispositivo Samsung J5 Prime fue

de 16 metros, en tanto que en el Xiaomi Note 8 fue de 15 metros. Los dispositivos se detectan entre sí, y la distancia es una aproximación a la distancia real.

Figura 43

Resultados de dispositivos encontrados por WIFI Direct (Samsung J5 Prime) dispositivo en el edificio



Fuente: elaboración propia.

Figura 44

Resultados de dispositivos encontrados por WIFI Direct (Xiaomi Note 8) dispositivo en la calle



Fuente: elaboración propia.

En cuanto a los resultados encontrados en las pruebas para el módulo de Bluetooth, las figuras 45 y 46 muestran que el dispositivo Samsung J5 Prime estimó un valor de 10.28 metros,

mientras que el Xiaomi Note 8 calculó 10.84 metros. Se observa que este cálculo es semejante para ambos dispositivos, con una precisión posiblemente mayor que la mostrada por WIFI Direct.

Figura 45

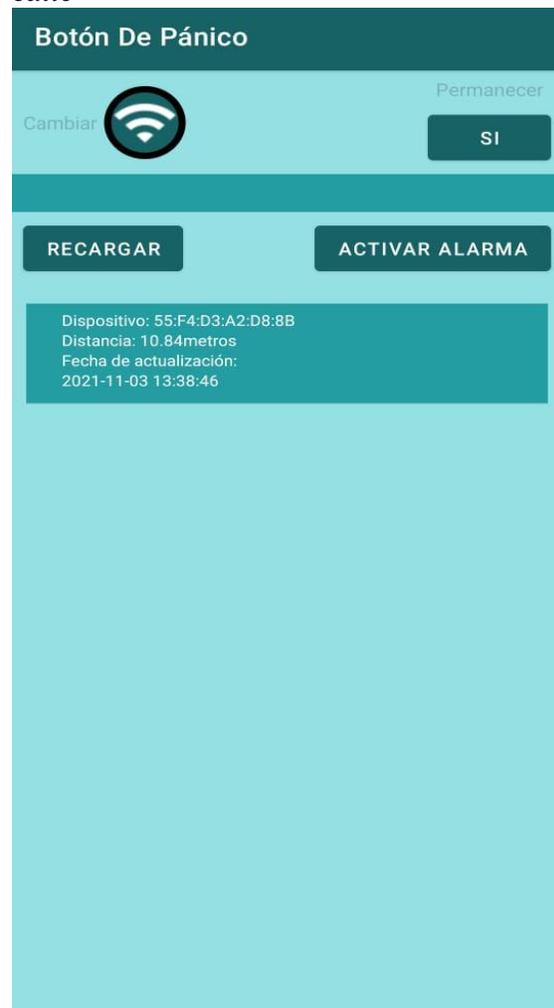
Resultados de dispositivos encontrados por Bluetooth (Samsung J5 prime) dispositivo en el edificio



Fuente: elaboración propia.

Figura 46

Resultados de dispositivos encontrados por Bluetooth (Xiaomi Note 8) dispositivo en la calle



Fuente: elaboración propia.

Se destaca que con ambas tecnologías, a pesar de los obstáculos, fue posible detectar al otro dispositivo.

5.5 Pruebas de compatibilidad

Las pruebas de compatibilidad, se realizaron con dispositivos que tenían diferentes versiones de sistema operativo Android. Para hacer las pruebas de compatibilidad, las características de los dispositivos usados son las siguientes :

- Xiaomi Redmi Note 8 con sistema operativo Android 11
- Xiaomi Redmi 9c con sistema operativo Android 10
- Samsung J5 Prime con sistema operativo Android 8
- Xiaomi Mi A3, con sistema operativo Android 11
- Huawei y5 2018 con sistema operativo Android 8.1
- Samsung J7 Prime con sistema operativo Android 8.1

Se observó con estos dispositivos que la aplicación tiene compatibilidad completa con sus módulos ya que todas las pruebas de campo salieron exitosas, sin embargo se encontró que en los dispositivos con Android 10 y 11, no se puede encender el WIFI automáticamente debido a que Google aumentó algunas restricciones de seguridad que no permiten manipular todos los componentes del dispositivo.

En el Samsung J7 Prime, se observó que con el módulo de Bluetooth no se puede hacer envío de beacons por que el dispositivo no es compatible con esa funcionalidad. Sin embargo, este módulo permite recibir la información enviada por otros dispositivos.

Se demuestra que se cumplió con el cuarto objetivo en su totalidad haciendo las pruebas necesarias a los módulos de la aplicación.

6. Conclusiones

El desarrollo del prototipo de la aplicación Botón de Pánico contribuye significativamente al proyecto de investigación *Plataforma para apoyar el proceso de rescate de sobrevivientes de un desastre a través de teléfonos móviles*. La idea de desarrollar esta aplicación surge de la iniciativa de la Cruz Roja Colombiana, Seccional Cundinamarca y Bogotá, ya que durante la formulación de la propuesta para el proyecto de investigación, los delegados de esta institución manifestaron la necesidad de tener una aplicación con estas características.

Teniendo en cuenta la experiencia vivida en el desarrollo de este proyecto, las conclusiones más importantes de este trabajo de grado se presentan a continuación:

- En primer lugar, como se demostró en el capítulo 5, se evidencia el cumplimiento de los objetivos propuestos. Como resultado de este trabajo de grado, se obtuvo una aplicación estable que permite anunciar la presencia de personas y, a su vez, recibir las señales provenientes de los teléfonos de otros usuarios mediante las tecnologías WIFI Direct y Bluetooth. En el proceso de desarrollo se levantaron los requerimientos, se diseñó la base de datos, se obtuvieron los diagramas UML y las historias de usuario, luego se aplicó la metodología scrum para desarrollar cada uno de los módulos y, finalmente, se llevaron a cabo las pruebas de campo de la aplicación.
- WIFI Direct permite enviar más información del usuario que Bluetooth, es por eso que mientras en WIFI Direct se pueden transmitir todos los datos personales del usuario, en Bluetooth únicamente es posible enviar el número que identifica al teléfono en la red. Por

otra parte, WIFI Direct ofrece un mayor alcance que Bluetooth y puede detectar más rápido los dispositivos; sin embargo, el punto débil en ambos casos son los cambios que ha hecho Google con las versiones de Android 10 en adelante, debido a las restricciones de uso para encender y apagar el adaptador por medio del código.

- Otra de las restricciones de Google que afectó el módulo de Bluetooth fue la de no poder obtener, desde el código, la dirección MAC del adaptador. Inicialmente, la aplicación enviaba este dato mediante balizas y servía para identificar al teléfono en la red, pero pronto fue evidente que esta dirección era aleatoria y que cambiaba cada vez que se intentaba obtenerla. Esto ocasionaba que un dispositivo apareciera múltiples veces en el módulo de Bluetooth como si fuera un teléfono diferente cada vez. La alternativa para resolver esto fue que cada teléfono generara un número aleatorio para identificarse en la red. El número se eligió dentro de un conjunto amplio, con el fin de reducir la probabilidad de duplicidad de identificadores en la red y, por lo tanto, de que dos dispositivos diferentes aparecieran como uno mismo. Este tema queda pendiente para que el proyecto de investigación busque otras opciones de identificación de los dispositivos cuando se utiliza Bluetooth.
- La integración de Firebase a proyectos de Android es una opción óptima para proyectos que no requieran una base de datos relacional, ya que facilita el desarrollo y la integración con la aplicación porque no es necesario hacer el modelo de entidad relación que sí requieren este tipo de bases ya que implicaría más tiempo de desarrollo para el proyecto.
- Teniendo en cuenta que el desarrollo de esta aplicación es un prototipo y no se cifraron los datos personales del usuario, se recomienda hacer una mejora del código para aplicar

seguridad a la información, como es el caso de la contraseña que en ningún momento se cifra para guardarse en la base de datos.

- En esta aplicación, se destaca que el desarrollo se hizo de tal manera que no se requiere previamente un emparejamiento de los dispositivos mediante Bluetooth, ni la creación de un grupo de WIFI Direct. El uso de balizas para Bluetooth y la funcionalidad de descubrimiento de servicios que ofrece WIFI Direct, ambas utilizadas en la aplicación, facilitan su operación.
- Los aspectos legales relacionados con el uso y difusión de información personal mediante esta aplicación cuando un usuario se registra en la misma, se validarán posteriormente en el proyecto de investigación con expertos en la materia. Esta aplicación es un primer prototipo para el proyecto, que se irá adaptando de acuerdo con las necesidades del mismo.

Como recomendaciones para el proyecto de investigación, se destacan:

- Se recomienda en un futuro desarrollo, evaluar la posibilidad de enviar la información que el usuario ingresó en el registro, a través de la tecnología de Bluetooth.
- Se recomienda incluir una representación gráfica para el módulo de Bluetooth, semejante a la que se ofrece en el módulo de WIFI Direct. En este prototipo no fue posible implementarla ya que implicaba más tiempo de investigación, lo cual retrasaba el cronograma del trabajo de grado.
- En el proceso de migración realizada al módulo de WIFI Direct de Java a Kotlin, se observó que se reduce la cantidad de líneas de código, lo cual hace más sencilla su

implementación y mantenimiento. Se recomienda hacer la misma migración para el módulo de Bluetooth.

- Se pudo implementar el envío de los datos registrados en la aplicación por el usuario mediante WIFI Direct, incluyendo las coordenadas del GPS. En el caso de Bluetooth, el cálculo de la distancia se hace con base en la intensidad de la señal. Esta forma de cálculo hace que difiera la distancia medida en cada módulo: mientras en WIFI Direct se obtiene la distancia horizontal que separa al emisor del receptor, en Bluetooth se infiere la distancia efectiva de separación entre los dos dispositivos.
- Con respecto a la incompatibilidad del Samsung J7 Prime con Bluetooth, se recomienda hacer pruebas en otros dispositivos diferentes a los usados en este proyecto, para saber si existe el mismo inconveniente.

Referencias Bibliográficas

- Acosta, F., Wister, M. A., Hernandez-Nolasco, J. A., & Pancardo, P. (2017). Moving smartphones to send emergency messages during assaults. *Advances in Intelligent Systems and Computing*, 612, 849–859. https://doi.org/10.1007/978-3-319-61542-4_87
- Amazon. (n.d.). Bases de datos no relacionales | Bases de datos de gráficos | AWS. Retrieved March 24, 2021, from <https://aws.amazon.com/es/nosql/>
- Android Studio. (n.d.-a). Descripción general de Kotlin | Desarrolladores de Android. Retrieved March 15, 2021, from <https://developer.android.com/kotlin/overview?hl=es>
- Android Studio. (n.d.-b). Introducción a Android Studio | Desarrolladores de Android. Retrieved March 13, 2021, from <https://developer.android.com/studio/intro?hl=es-419>
- Bara, M. (2015). Las 5 etapas en los “*Sprints*” de un desarrollo Scrum | OBS Business School. <https://www.obsbusiness.school/blog/las-5-etapas-en-los-sprints-de-un-desarrollo-scrum>
- Congreso de Colombia. (1982). CECOLDA - Centro Colombiano del Derecho de Autor - LEY 23 DE 1982 SOBRE DERECHO DE AUTOR. <http://www.cecolda.org.co/index.php/derecho-de-autor/normas-y-jurisprudencia/normas-nacionales/124-ley-23-de-1982-sobre-derecho-de-autor>
- Congreso de Colombia. (2012). Ley Estatutaria 1581 De 2012. *Journal of Chemical Information and Modeling*, 48(9), 1–11.
- El Comercio Perú. (2020). Terremoto en Turquía y Grecia EN VIVO: Sismo de magnitud 7,0 deja al menos 41 muertos y se teme un centenar de atrapados bajo los escombros | FOTOS Y VIDEOS | Esmirna | MUNDO | EL COMERCIO PERÚ. <https://elcomercio.pe/mundo/europa/turquia-fuerte-sismo-de-magnitud-7-sacude-el-oeste-turco-izmir-mar-eggeo-videos-nndc-noticia/>

Fernandez, Y. (2020). WIFI Direct: qué es, para qué sirve y cómo funciona.

<https://www.xataka.com/basics/WIFI-direct-que-sirve-como-funciona>

Instituto de Geografía (IGUNNE) Facultad de Humanidades. (n.d.). vulnerabilidad. Retrieved

March 23, 2021, from

<https://hum.unne.edu.ar/revistas/geoweb/Geo2/contenid/vulner3.htm>

Ionos. (2020). ¿Qué es Bluetooth? Todo lo que debes saber - IONOS.

<https://www.ionos.es/digitalguide/servidores/know-how/que-es-bluetooth/>

Kisara Labs. (2021). SOSMEX || Home. <https://sosmex-niunamenos.com/>

Kaisar, S. (2021). Emergency Response and Post-Disaster Recovery Using Smartphone-Based

Applications. In L. Oliveira, F. Tajariol, & L. Gonçalves (Eds.), Digital Services in

Crisis, Disaster, and Emergency Situations (pp. 31-49). IGI Global.

<http://doi:10.4018/978-1-7998-6705-0.ch002>

Lopez, S. (2020). Firebase: qué es, para qué sirve, funcionalidades y ventajas.

<https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/>

Mody, V., Mody, V., & Parekh, S. (2020). Distress-An Application for Emergency Response and

Disaster Management. Proceedings - International Conference on Smart Electronics and

Communication, ICOSEC 2020, Icosec, 830–836.

<https://doi.org/10.1109/ICOSEC49089.2020.9215288>

Requena, A. (2018). Qué es un *Sprint* de Scrum | OpenWebinars.

<https://openwebinars.net/blog/que-es-un-sprint-scrum/>

- Rockcontent. (n.d.). ¿Qué es un lenguaje de programación y qué tipos existen? 2019. Retrieved March 13, 2021, from <https://rockcontent.com/es/blog/que-es-un-lenguaje-de-programacion/>
- Schwaber, K., & Sutherland, J. (2015). Scrum Guide V7. November, 133–152.
- Softeng. (n.d.). Proceso y Roles de Scrum. Retrieved March 17, 2021, from <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>
- S. Gerasenko, A. Joshi, S. Rayaprolu, K. Ponnaivaikko and D. P. Agrawal, "Beacon signals: what, why, how, and where?," in *Computer*, vol. 34, no. 10, pp. 108-110, Oct. 2001, doi: 10.1109/2.955103.
- Tejada, Z. (2018). Datos no relacionales y NoSQL - Azure Architecture Center | Microsoft Docs. <https://docs.microsoft.com/es-es/azure/architecture/data-guide/big-data/non-relational-data>
- Yirda, A. (2021). ¿Qué es Fenómeno Natural? » Su Definición y Significado [2021]. <https://conceptodefinicion.de/fenomeno-natural/>
- Acrylic. (2021). *Beacon Bluetooth. Conoce más sobre esta tecnología y sus ventajas.* <https://www.acrylicwifi.com/blog/beacon-bluetooth-tecnologia-ventajas/>
- Arroyo Barrantes, Susana; Rodríguez, Martha; Pérez, R. (2009). *Gestión de la información y comunicación en emergencias y desastres.*
- Capacci, A., & Mangano, S. (2015). Las catástrofes naturales. *Cuadernos de Geografía: Revista Colombiana de Geografía*, 24(2), 35–51. <https://doi.org/10.15446/rcdg.v24n2.50206>
- Digital, R. T. (n.d.). *El protocolo Simple Service Discovery Protocol - Revista Transformación Digital.* Retrieved November 24, 2021, from

<https://www.revistatransformaciondigital.com/2021/11/18/el-protocolo-simple-service-discovery-protocol/>

Gutierrez, M. (2016). *Qué es WiFi Direct y en qué se diferencia del Bluetooth.*

https://www.elespanol.com/elandroidelibre/20161018/wifi-direct-diferencia-bluetooth/163984740_0.html

Mundo Altavoces. (n.d.). *Bluetooth clases y versiones, ¿en qué se diferencian?* | Mundo

Altavoces /. Retrieved August 27, 2021, from <https://mundoaltavoces.com/bluetooth-todas-las-clases-y-versiones-en-que-se-diferencian/>

País, E. (2010). *¿Cuánto se puede vivir bajo los escombros?* | Internacional | EL PAÍS.

https://elpais.com/internacional/2010/01/15/actualidad/1263510008_850215.html

Reliefweb. (2021). *Colombia: Inundaciones en Arauca (Agosto 2021) - Colombia* | ReliefWeb.

<https://reliefweb.int/report/colombia/colombia-inundaciones-en-arauca-agosto-2021>

RTVE.es. (2020). *Terremoto | Suben a 71 los muertos en Turquía y Grecia.*

<https://www.rtve.es/noticias/20201101/terremoto-turquia-grecia-53-muertos-desconocen-numero-desparecidos/2051540.shtml>

Unidas, N. (2021). *Haití: El terremoto deja un creciente número de muertos, heridos y grandes*

daños | Noticias ONU. <https://news.un.org/es/story/2021/08/1495672>

Yubal. (2021). *Bluetooth 5.2 vs Bluetooth 5.1 vs Bluetooth 5.0: cuáles son las diferencias.*

<https://www.xataka.com/basics/bluetooth-5-2-vs-bluetooth-5-1-vs-bluetooth-5-0-cuales-diferencias>

