



EpilepsyAlarm: desarrollo de una aplicación móvil para la gestión y envío de alertas previo a un
ataque de epilepsia

David Santiago Rey Beltrán

11161827680

Universidad Antonio Nariño

Facultad de Ingeniería de Sistemas

Programa de Ingeniería de Sistemas y Computación

Bogotá

2023

EpilepsyAlarm: desarrollo de una aplicación móvil para la gestión y envío de alertas previo a un
ataque de epilepsia

David Santiago Rey Beltrán

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

Ingenieros de Sistemas y Computación

Director (a):

PhD. María del Pilar Salamanca Azula

Universidad Antonio Nariño

Facultad de Ingeniería de Sistemas

Programa de Ingeniería de Sistemas y Computación

Bogotá

2023

NOTA DE ACEPTACIÓN

El trabajo de grado titulado

_____, Cumple con

los requisitos para optar

Al título de _____.

Firma del Tutor

Firma Jurado

Firma Jurado

Bogotá, 11 de julio de 2023

Contenido

	Pág.
Resumen	12
Introducción	13
1.Planteamiento del problema	15
1.1. Descripción del problema	15
1.2. Formulación del problema	18
1.3. Justificación	19
1.4. Objetivos.....	20
1.4.1. Objetivo general.....	20
1.4.2. <i>Objetivos específicos</i>	20
1.5. Alcance y limitaciones del proyecto	21
1.5.1. <i>Alcance</i>	21
1.5.2. <i>Limitaciones</i>	21
2.Marco de referencia	22
2.1. Marco teórico	22
2.1.1. <i>Botón de pánico</i>	22
2.1.2. <i>Padecimiento médico</i>	22
2.1.3. <i>Epilepsia</i>	22
2.1.4. <i>Convulsiones</i>	23
2.1.5. <i>Aplicación móvil</i>	24
2.1.6. <i>Mensajería instantánea</i>	25

2.1.7. Kotlin.....	25
2.2. Antecedentes o estado del arte.....	25
2.3. Marco legal	28
3.Aspectos Metodológicos.....	29
3.1. Descripción de la metodología.....	29
3.1.1. Ritos	29
3.1.2. Artefactos	30
3.1.3. Roles.....	31
3.2. Aplicación de la metodología	32
3.3 Aplicación de la metodología	33
3.2.1. <i>Sprint 1 – Configuración de herramientas y plan de trabajo</i>	33
3.2.2. <i>Sprint 2 – Análisis de mensajería instantánea</i>	34
3.2.3. <i>Sprint 3 – Envío de mensajes con datos precargados y codificación de la</i>	
<i>integración</i>	34
3.2.4. <i>Sprint 4 – Muestra datos en pantalla</i>	35
4.Desarrollo del proyecto.....	36
4.1 Descripción de la aplicación	36
4.2 Fase de iniciación	36
4.3 Planeación y estimación	37
4.3.1 <i>Roles en la aplicación</i>	37
4.3.2 <i>Arquitectura y herramientas</i>	37
4.4 Sprint 1	43
4.4.1 <i>Historias de usuario</i>	43

4.4.2 <i>Diseño de Mockups</i>	45
4.5 Sprint 2	51
4.5.1 <i>Formulario</i>	52
4.5.2 Selección del medio de mensajería instantánea	52
4.5.2.1 <i>Enlace de mensajería con la aplicación</i>	53
4.5.3 <i>Interfaz de la aplicación</i>	54
4.6 Sprint 3	61
4.6.1 <i>Guardado de información</i>	61
4.6.2 <i>Envío de mensajes de alerta</i>	62
4.6.3 <i>Funcionalidad de botón de pánico</i>	63
4.6.4 <i>Conexión a un dispositivo físico de Botón de Pánico vía Bluetooth</i>	65
4.7 Sprint 4	69
4.7.1 <i>Generación de alarma</i>	70
4.7.2 <i>Configuración de la alerta</i>	70
4.7.3 <i>Mostrar instrucciones en pantalla</i>	71
4.7.4 <i>Activación de ubicación</i>	72
5. Análisis y resultados	79
5.1 Selección de medio para enviar mensaje de alerta.....	79
5.2 Codificar funcionalidad de envío de alerta	79
5.3 Mostrar datos en pantalla	80
5.4 Clase para conectar un dispositivo Bluetooth	81
5.5 Funcionalidades.....	81
5.5.1 <i>Enviar mensaje por mensajería instantánea</i>	81

5.5.2 Muestra de datos en pantalla	83
5.5.3 Habilidad de botón de bloqueo para envío de alerta.....	83
5.2.4 Enviar mensaje de texto por SMS.....	84
6.Conclusiones.....	86
Referencias bibliográficas	90

Lista de ilustraciones

	Pág.
Ilustración 1. Diagrama de casos de uso.....	38
Ilustración 2 Diagrama de secuencia	39
Ilustración 3. Diagrama de secuencia parte 1	40
Ilustración 4. Diagrama de secuencia parte 2	41
Ilustración 5. Complemento diagrama secuencia parte 2	42
Ilustración 6. Diagrama de secuencia - localización parte 3.....	43
Ilustración 7. Complemento Diagrama secuencia localización parte 3	43
Ilustración 8. Mockup inicio de la aplicación.....	46
Ilustración 9. Mockup registro del usuario	47
Ilustración 10. Mockup configuración de alarma	48
Ilustración 11. Mockup activar ubicación.....	49
Ilustración 12. Mockup Dashboard.....	50
Ilustración 13. Perfil del usuario.....	51
Ilustración 14. Librería Room.....	52
Ilustración 15. Interfaz para agregar mensaje	53
Ilustración 16. Código de autorización con Token	53
Ilustración 17. Enlace con la API	54
Ilustración 18. Registro de usuario	56
Ilustración 19. Registrar ubicación	58
Ilustración 20. Menú de usuario.....	59
Ilustración 21. Configurar sonido alarma	60

Ilustración 22. Librería Room.....	61
Ilustración 23. Guardado de información	62
Ilustración 24. Código de envío de mensaje	62
Ilustración 25. Código fuente configurar botón	64
Ilustración 26. Fragmento de código para conexión.....	67
Ilustración 27. Código fuente para envío de mensaje	69
Ilustración 28. Código fuente configuración alarma.....	70
Ilustración 29. Configuración alerta	71
Ilustración 30. Instrucciones en pantalla.....	72
Ilustración 31. Configurar mensaje de alerta	74
Ilustración 32. Permisos de ubicación	75
Ilustración 33. Código fuente de GoogleMaps	75
Ilustración 34. Código fuente para mostrar mensaje en pantalla	76
Ilustración 35. Pantalla de bloqueo con instrucciones	78
Ilustración 36. Fragmento de código para activación de función de envío de alerta.....	80
Ilustración 38. Mensaje recibido con ubicación	82

Lista de Tablas

	Pág.
Tabla 1. Tabla de productos y aplicaciones en el mercado.....	27
Tabla 2. Product Backlog.....	44
Tabla 3. Historia de usuario “Registrar usuario con epilepsia”	54
Tabla 4. Historia de usuario “Registrar ubicación en tiempo real”	57
Tabla 5. Historia de usuario “Editar información”	58
Tabla 6. Historia de usuario “Configuración de sonido de alarma”	60
Tabla 7. Historia de usuario “Configurar alerta”	63
Tabla 8. Historia de usuario “Enviar mensaje de alerta por mensajería”	64
Tabla 9. Historia de usuario "Clase para conectar un dispositivo Bluetooth"	67
Tabla 10. Historia de usuario “Configurar mensaje de alerta”	73
Tabla 11. Historia de usuario “Enviar ubicación en tiempo real”	74
Tabla 12. Historia de usuario “Mostrar instrucciones en pantalla para el manejo del paciente durante el ataque”	76

Dedicatoria

*Este trabajo de grado va dedicado primeramente a Dios
porque sin él, hoy no estaría donde estoy, cumpliendo
mis objetivos y anhelos del corazón.*

*En segundo lugar, mis padres que han hecho esfuerzos
muy grandes para que yo alcance mis metas en esta profesión,
siendo un apoyo incondicional en los momentos difíciles de todo
este camino recorrido.*

*Y, por último, a los docentes de la universidad Antonio
Nariño que me dieron el impulso para arriesgarme a nuevos
retos, tanto personales como profesionales.*

*“La imaginación es más importante que el conocimiento.
El conocimiento es limitado, mientras que la imaginación no
tiene límites”.*

Albert Einstein

Resumen

La epilepsia es un problema de salud bastante complejo de sobrellevar, que afecta tanto a hombres como a mujeres de todas las edades, provocando una alteración en la actividad de las células nerviosas y ocasionando indefensión ante situaciones críticas que se presenten debido a su condición médica. Esta enfermedad provoca convulsiones involuntarias en la persona que la padece, y al momento de sucederle un episodio de estos, la víctima queda vulnerable. No obstante, algunas personas que padecen de dicho problema son capaces de anticipar que un episodio crítico va a ocurrir. En este trabajo de grado se realizó una aplicación móvil llamada EpilpsyAlarm que permite gestionar y enviar una alerta mediante la simulación de un botón de pánico, para que estas personas puedan solicitar ayudas segundos antes de un ataque. La alerta enviada incluye la ubicación, el tipo de sangre y el nombre de la víctima, y se configura dentro de la aplicación para que el destinatario sea una persona cercana o un familiar.

Introducción

Un padecimiento médico es una afección relacionada con alguna enfermedad, y en muchas ocasiones aquellos que sufren de esta afección médica pueden tornarse vulnerables cuando la afección se agudiza. Tal es el caso de un trastorno en la actividad de las células nerviosas del cerebro, denominado epilepsia, el cual genera convulsiones en su víctima impidiéndole realizar movimientos voluntarios, que pueden ocurrir cuando la persona se encuentra sola, sin poder ser socorrida por alguien cercano que actúe de manera oportuna frente a esta situación.

La aplicación móvil EpilepsyAlarm puede ser un apoyo para las personas que sufren de aquellos trastornos, ofreciéndoles un medio para enviar una alerta de ayuda a alguna persona de confianza del paciente, permitiéndole conocer la ubicación de la persona y en dicho caso, acudir para atenderla de una manera correcta frente a esta situación. Esta aplicación se activa por medio de uno de los botones del dispositivo móvil, siendo accionado por el paciente momentos antes del ataque, enviando una señal de alerta a una persona de su entera confianza que se encuentre registrada dentro de la aplicación. Adicionalmente la aplicación envía la información y ubicación de la persona que está padeciendo el ataque, dándole la oportunidad a la persona de confianza de acudir en su ayuda. Cuando se envíe la alerta se activará una alarma y se mostrará en pantalla un conjunto de instrucciones sobre cómo manejar a la persona que está padeciendo el ataque.

Este documento se encuentra estructurado de la siguiente manera: en el capítulo 1 se describe el planteamiento del problema, donde se detalla el contexto del problema junto con los motivos que llevaron a la realización de este proyecto, además del planteamiento de los objetivos; en el capítulo 2 se describe el marco de referencia, se definen los conceptos importantes dentro del proyecto, se realiza una comparación con otras aplicaciones del mercado que tienen una función similar, explicando el funcionamiento de esta aplicación; el capítulo 3 describe la metodología que

se utilizó para realizar este proyecto, presentando la estrategia utilizada en el proceso de elaboración del mismo y sus componentes en detalle; el capítulo 4 presenta el proceso de desarrollo detalladamente del proyecto, con todas las herramientas que se utilizaron; el capítulo 5 describe el análisis de los resultados obtenidos del proyecto; y el capítulo 6, describe las conclusiones.

1. Planteamiento del problema

1.1. Descripción del problema

La epilepsia puede afectar a cualquier persona, de cualquier raza, género, grupo étnico o edad. Es un trastorno en el cual se interrumpe la actividad de las células nerviosas del cerebro, lo que provoca convulsiones (MayoClinic, 2021). Al momento de sobrevenir un ataque de epilepsia, la persona es vulnerable ante cualquier amenaza, porque esta interrupción cambia temporalmente la forma en la que funciona el cuerpo. La forma en la que se presenta una convulsión puede variar entre las personas que sufren de epilepsia, ya que depende del tipo de convulsión que pueda experimentar y la crisis en particular de epilepsia.

Para entender un poco acerca de estas convulsiones que puede suceder, es importante saber que dichas convulsiones comprenden cuatro fases, las cuales son:

- **Prodrómica:** esta fase puede ocurrir varias horas o días incluso antes de una convulsión real. Este término se trata del periodo desde el momento en el que comienzan los síntomas más obvios. En estos casos los síntomas más comunes incluyen confusión, ansiedad, irritabilidad, dolor de cabeza, temblor e ira u otros síntomas relacionados con el estado de ánimo.
- **Ictal temprana:** en esta fase, la mayoría de personas que sufren de ataques de epilepsia, pueden sentir una primera advertencia en su cuerpo antes de la actividad convulsiva, este sentir se le conoce como **aura**. Esta aura se considera como un signo de que viene una convulsión. Un 65% de las personas con epilepsia experimentan esta aura, no todas pueden sentirlo, pero sí la mayoría. En aquellos que lo sienten, los síntomas de esta aura varían según el tipo de convulsión. Algunos de estos síntomas comunes son:
 - Sabores amargos

- Mareos
- Visión parpadeante
- Alucinaciones
- Dolor de cabeza
- Hormigueo
- Pérdida de la visión
- Náuseas

Esta aura es parte importante en este trastorno, ya que aquellas personas que la sienten tienen un poco más de ventaja momentos antes de que vayan a tener un ataque epiléptico.

- **Ictal:** esta es la fase en la que ocurren las convulsiones. Se pueden manifestar de varias maneras por lo que los síntomas que siente la persona durante el ataque son diferentes, e incluyen, entre otros:

- Lapsos de memoria
- Salivación
- Confusión
- Movimientos de contracciones de los ojos
- Corazón acelerado
- Temblores
- Transpiración
- Rigidez en partes del cuerpo

- **Post-Ictal:** esta última fase va después de que ocurrieron las convulsiones. Algunas personas, dependiendo de la gravedad de las convulsiones, se recuperan al instante, mientras que muchas otras requieren de un tiempo más prolongado para que vuelvan a sentir que se encuentran en su estado normal (Epilepsy, 2023).

El cuidado de estas convulsiones requiere una especial atención, pues una reacción tardía puede terminar en fatalidad o discapacidad grave. Sin embargo, esta afección permite a las personas prever que tendrán una crisis epiléptica y, en caso de que esto ocurra, se podría enviar una alerta para recibir una asistencia pronta.

Una manera eficiente de enviar una alerta puede ser mediante el uso de una aplicación móvil, aprovechando la gran popularidad de los teléfonos inteligentes. Así mismo, se puede mostrar en la pantalla la información para que otras personas del entorno puedan colaborar durante el ataque, asistiéndola de manera adecuada y notificando a su contacto de emergencia, ya que la información de dicho contacto se encuentra visible en el dispositivo del paciente, permitiendo que la aplicación sea de apoyo tanto para la persona que se encuentra en problemas, como para las personas que están brindando su ayuda.

Los sistemas físicos y aplicaciones móviles que se evidenciaron funcionan de una manera diferente, ya que el propósito de estas aplicaciones es enviar una alerta en el momento en el que está ocurriendo el ataque de epilepsia, a diferencia de EpilepsyAlarm que lo hace justo antes de que ocurra un ataque, dando un poco más de tiempo a la persona para reaccionar ante la situación.

Los sistemas físicos de alertas tales como botones de pánico y las aplicaciones móviles de envío de alertas, que se encuentran actualmente en el mercado, ofrecen funcionalidades similares, pero con limitaciones tales como, el alcance del GPS, el modo en el que se activa la alerta, entre

otras, ya que ninguna aplicación o sistema reúne todas las funcionalidades que ofrece EpilepsyAlarm.

Por ello se desarrolló una aplicación móvil que al oprimir alguno de los botones del dispositivo envíe por medio de un sistema de mensajería instantánea datos como: nombre, ubicación en tiempo real, tipo de sangre. Adicionalmente dichos datos se muestran en la pantalla, incluso cuando el celular está bloqueado; esto junto con instrucciones sencillas de cómo asistir a una persona que está sufriendo un ataque de epilepsia. Todo lo anterior sentará las bases para que eventualmente más adelante, se pueda diseñar un dispositivo físico portátil que conectado al celular mediante Bluetooth sea el disparador de esta alerta. De esa manera, se podría reducir el riesgo que, en ciertos ámbitos, implica sacar el teléfono o cuando la situación médica no dé tiempo para tomar el teléfono y activar la aplicación.

1.2. Formulación del problema

La epilepsia es una de las afecciones de salud que permite a aquellas personas que pueden prever un posible ataque momentos antes de que ocurran las convulsiones. Actualmente no existe un medio diseñado para que las personas puedan alertar antes del ataque.

La rapidez en la atención de una persona que padezca de este trastorno puede ser determinante, ya que, si la persona se encuentra sola al momento de la crisis, cuando la ayuda llegue puede ser tarde para atender el ataque. Por ello la reacción inmediata de alguna persona cercana y de su contacto de confianza, puede representar la diferencia para la víctima.

Por lo anterior, la aplicación móvil EpilepsyAlarm puede contribuir en reducir la incertidumbre para aquellos pacientes que pueden sentir los síntomas antes mencionados, momentos antes de tener un ataque de epilepsia, al permitir alertar a un contacto de confianza justo antes de la ocurrencia del ataque, y al mostrarle a las personas que se encuentren en las

proximidades del enfermo, instrucciones básicas sobre el manejo de la crisis, así como algunos datos del paciente.

1.3. Justificación

El presente proyecto se enfoca en la implementación de una aplicación de botón de pánico, que le permita tener la oportunidad de brindar una atención más oportuna a aquellas personas que padecen del trastorno de epilepsia, ya que esta enfermedad al no ser atendida a tiempo puede generar lesiones futuras en la persona. La aplicación, llamada EpilepsyAlarm, beneficia a las personas que padecen dicha enfermedad, porque les da la oportunidad de enviar una señal de alerta por medio del toque del botón de desbloqueo del dispositivo móvil en el cual se encuentra la aplicación instalada, a una persona de confianza. Adicionalmente la aplicación mostrará en pantalla datos importantes del paciente, e indicará cómo actuar durante el ataque, lo cual será útil para que las personas que se encuentren cerca del paciente puedan colaborarle mientras el contacto de confianza llega al lugar donde este se encuentra.

Además de ser un beneficio para aquellas personas que padecen dicha enfermedad, es de gran alivio para sus familiares, ya que pueden estar al tanto del momento y el lugar en el que su familiar está teniendo un ataque de epilepsia.

Este proyecto es realizado sin ánimo de lucro, ni con objeto de ganancias, porque el objetivo es brindar un apoyo a aquellas personas que padecen de este trastorno, siendo un gran beneficio en el campo de la salud y de la tecnología. Se espera que en un futuro esta aplicación se pueda integrar con un dispositivo externo de hardware que dispare la alerta y que se comunique con el teléfono inteligente mediante Bluetooth. El dispositivo sería un pequeño botón que se llevaría en el cuerpo, sería de fácil acceso y así evitaría que se tenga que sacar el teléfono para activar la aplicación. El desarrollo de la aplicación EpilepsyAlarm es una gran oportunidad de

adquirir experiencia en el campo del desarrollo móvil, porque se trata del uso de patrones y modelos de arquitectura, además de metodologías que permiten hacer que un proyecto sea escalable y funcional.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar una aplicación móvil de botón de pánico para personas con epilepsia que les permita enviar una alerta con sus datos personales por medio de un sistema de mensajería instantánea, mostrando en la pantalla dichos datos personales y un conjunto de instrucciones sencillas de cómo asistir a la persona en el momento del ataque.

1.4.2. Objetivos específicos

1. Elegir el medio objetivo por el cual se enviarán los mensajes de emergencia con base en el análisis de los sistemas de mensajería instantánea actualmente existentes, identificando aquel con el que sea más factible interactuar programáticamente.
2. Codificar la funcionalidad de envío de alerta desde la aplicación móvil por medio de un sistema de mensajería instantánea, incluyendo los datos definidos previamente por el usuario, para notificar a un familiar o persona cercana sobre la ocurrencia de un ataque.
3. Permitir que en la pantalla de bloqueo se muestren los datos personales definidos anteriormente por el usuario, para que las personas que se encuentren en las proximidades conozcan los datos de la persona afectada por el ataque, y para que reciban algunas indicaciones sobre cómo asistirle.
4. Implementar funciones, procedimientos o código necesario para una futura integración con un hardware del botón enlazado mediante conexión Bluetooth.

1.5. Alcance y limitaciones del proyecto

1.5.1. Alcance

Las funcionalidades con las que contará la aplicación son:

- **Funcionalidad 1:** enviar por medio de un sistema de mensajería instantánea: nombre, ubicación en tiempo real, tipo de sangre.
- **Funcionalidad 2:** mostrar en la pantalla, aun cuando esté bloqueado el celular, nombre, tipo de sangre, nombre de contacto de emergencia y número de contacto de emergencia.
- **Funcionalidad 3:** habilitar uno de los botones del celular, para que al ser oprimido se active la funcionalidad de enviar alerta y mostrar datos.
- **Funcionalidad 4:** enviar un mensaje SMS cuando se active la funcionalidad de envío de alerta.

1.5.2. Limitaciones

- La alerta no se va a activar de manera autónoma, sin antes oprimir el botón.
- La aplicación se entregará en APK y no autorizada en la Play Store.
- Aunque eventualmente esta aplicación se podrá integrar a un dispositivo físico, el presente proyecto no contempla la realización de dicho hardware.
- La aplicación funcionará únicamente sobre el sistema operativo Android.

2. Marco de referencia

En este capítulo se presenta inicialmente el marco teórico, cuyo propósito es definir los términos que deben conocerse para entender este proyecto y su contexto. Luego se describe el estado del arte en cuanto a aplicaciones móviles y botones físicos que realizan funciones semejantes a las de EpilepsyAlarm. Por último, se explica el marco legal en el cual se desarrolla este proyecto. Marco teórico

2.1.1. Botón de pánico

Es un dispositivo físico o una aplicación de software que produce una señal de alerta, que permite notificar a un centro de control o a una persona cercana, en caso de una emergencia por un padecimiento médico o un problema, informando sobre alguna irregularidad ocurrida, en respuesta a alguna persona que decidió activar este medio (Securitas, 2022).

2.1.2. Padecimiento médico

Un padecimiento médico se relaciona con sufrir alguna afección o enfermedad, que muchas veces impide a la persona realizar sus funciones de manera normal. La epilepsia es un padecimiento neurológico que afecta la salud de la persona por unos instantes y que puede tener consecuencias a largo plazo si no se controla a tiempo, en este caso se trata de un trastorno neuronal, que altera la actividad cerebral de la persona (Pérez Porto, 2010).

2.1.3. Epilepsia

Esta enfermedad catalogada como un síndrome cerebral crónico, caracterizado por crisis recurrentes y autolimitadas (se detienen solas), en el que se interrumpe la actividad de las células nerviosas en el cerebro, provocando convulsiones en la persona que está teniendo este episodio, impidiéndole el control de su cuerpo durante el ataque, que a largo plazo se puede controlar con

medicamentos o en ocasiones con cirugía, realizando tratamientos que permitan a la persona disminuir la cantidad de ataques que le sobrevengan (Medical Education, 2022).

2.1.4. Convulsiones

Las convulsiones son síntomas de problemas en el cerebro, debido a una actividad anormal en el mismo. Pueden causar cambios en el comportamiento o movimientos y en los niveles de conciencia, estas mismas tienen como consecuencia la pérdida temporal de la memoria o del conocimiento. Se puede determinar la gravedad de las convulsiones por el tiempo de duración, ya que si una convulsión dura más de 5 minutos es considerada como una emergencia médica (MedlinePlus, 2022). Estas convulsiones que ocurren debido a la crisis epiléptica se pueden catalogar de dos tipos de convulsiones que son:

- **Convulsiones focales (parciales), afectan solo parte del cerebro:** estas convulsiones pueden durar unos segundos o incluso varios minutos, de los cuales algunos síntomas son:
 - Tener fuertes emociones como temor, pánico, tristeza
 - Dificultad para hablar
 - Movimientos extraños en el cuerpo
 - No estar consciente del entorno

En este tipo de convulsiones, se pueden desarrollar tres tipos de crisis:

- **Crisis parcial simple:** ocurre cuando se presentan una alteración de la memoria, el movimiento y las sensaciones, provocando que la persona esté alerta.

- **Crisis parcial compleja:** es más frecuente, en donde la persona pierde el conocimiento, con movimientos convulsivos.
- **Crisis parcial que desemboca en generalizada al extenderse al resto del cerebro**
- **Convulsiones generalizadas, que afectan todo el cerebro:** estas convulsiones pueden ocurrir de manera imprevista. Existen varios tipos de esta crisis:
 - **Crisis tónico-clónicas:** en la primera etapa, la persona pierde el conocimiento y se desmaya. Los músculos se contraen de manera tensa, lo que provoca que el cuerpo se sacuda. Después de un tiempo la persona recupera el conocimiento.
 - **Crisis de ausencia:** la persona pierde el conocimiento, sintiéndose perdido en el entorno.
 - **Crisis tónicas:** los músculos de la persona se ponen rígidos, si la persona se encuentra de pie, puede caer de espaldas.
 - **Crisis atónicas:** al contrario que la crisis anterior, los músculos se relajan, haciendo que el cuerpo se ponga laxo.
 - **Crisis mioclónicas:** los músculos se sacuden de manera descontrolada, afectando todo o solo una parte del cuerpo. (BupaSalud, s.f.)

2.1.5. Aplicación móvil

Es una aplicación que se ejecuta en los dispositivos móviles, tales como celulares o tabletas. Algunas aplicaciones se utilizan como un medio de mensajería para comunicarse con personas alrededor del mundo (Herazo, s.f.).

2.1.6. Mensajería instantánea

Es un sistema por el cual se puede intercambiar mensajes con otras personas, ya sea que se encuentren cerca o lejos, esto permite tener un mayor alcance en caso de intercambiar información importante. Para programar estas aplicaciones en los dispositivos móviles, se utilizan ciertos lenguajes de programación, tales como Kotlin o Java (Andújar, 2017).

2.1.7. Kotlin

Es un lenguaje de programación de tipado estático, que funciona sobre una máquina virtual de Java y es utilizado para la creación de aplicaciones móviles (Heiss, 2013).

2.2. Antecedentes o estado del arte

En el mercado ya se encuentran botones de alerta, tanto de manera física como en aplicaciones móviles, que permiten realizar un seguimiento durante o después de un ataque de epilepsia. Inicialmente se analizarán las aplicaciones que se enfocan en esta enfermedad, posteriormente se presentarán los dispositivos físicos que cumplen una función similar pero que están más enfocados en alertar a los familiares, que en llevar un control de la situación.

Aplicaciones móviles similares a EpilepsyAlarm:

- **SeizAlarm:** Es una aplicación móvil para el sistema operativo iOS, que se encarga de avisar a una persona cercana que el usuario está teniendo un ataque, dado que la aplicación se activa con el movimiento, no avisa antes del ataque sino durante la ocurrencia del mismo mediante un botón digital que envía la información y ubicación de esta persona. Para que funcione la aplicación es necesario tener el dispositivo pegado al cuerpo todo el tiempo. (Álvarez, 2015).

- **Control Diario de Epilepsia:** Esta aplicación permite llevar un seguimiento de la enfermedad, registrando la fecha y hora del ataque y una alarma para tomar los medicamentos (Asociación Coruseña de Epilepsia, 2021).
- **EpiWatch:** Por medio del reloj inteligente de Apple, esta aplicación lleva un seguimiento y realiza una serie de preguntas durante el ataque a la persona, para saber qué tan consciente se encuentra en el momento. Además, registra los temblores de su cuerpo (Asociación Coruseña de Epilepsia, 2021).

Sistemas físicos similares:

- **Daytech:** Es un sistema de alarma de pánico, que trae un botón de llamada de emergencia para cuidadores, alarma de seguridad para ancianos. Este dispositivo funciona con un botón de llamada el cual al ser presionado por el paciente o la persona que necesite ayuda, llama a un cuidador que es aquel que tiene un receptor con un alcance de 300 pies (Daytech, 2020).
- **Botón S.O.S con GPS:** Este botón utilizado en adultos mayores, cuenta con un botón de llamada SOS a tres números fijos o celulares elegidos por el usuario, que permite mediante el botón hablar con la persona que se contactó anteriormente al oprimirlo, además cuenta con un sistema de rastreo GPS, que muestra la ubicación de la persona y la batería del dispositivo (Asistronic, s.f.).
- **Pulsera D-Atom:** Este botón permite hacer una petición al personal de ayuda, ya que genera una alarma y envía un aviso a la unidad Trex que es el dispositivo que lleva el personal capacitado para atender estas situaciones en un centro geriátrico (Miray Consulting, s.f.).

En la Tabla 1 se comparan los sistemas anteriormente explicados. Para ello los criterios de comparación que se tomaron en cuenta fueron los siguientes.

- Alarma
- GPS
- Alcance de más de 100 metros
- Mensaje de alerta
- Es una aplicación móvil
- Funciona en el sistema operativo Android
- Muestra instrucciones en pantalla para el manejo del ataque
- Se activa con un botón del celular

Tabla 1.

Tabla de productos y aplicaciones en el mercado

Producto/ Característica	Alarma	Envío de coordenadas GPS al contacto	Alcance de más de 100 metros	Mensaje de alerta	Es Móvil	Funciona en Android	Manual de ayuda durante el ataque	Funciona botones dispositivo móvil
EpilepsyAlarm	X	X	X	X	X	X	X	X
Daytech	X	X						
Botón S.O.S con GPS	X	X	X					
Pulsera D-Atom	X			X				
SeizAlarm	X	X	X	X	X			
Control Diario de Epilepsia					X	X		
EpiWatch	X							

Nota. Datos tomados de diferentes sitios de la Web.

Se puede concluir que de los botones de pánico que se encuentran en el mercado ninguno cumple con todas las características que posee la aplicación EpilepsyAlarm, principalmente el

rango de alcance de los botones físicos ofrecidos en el mercado es muy limitado o el GPS de la aplicación solo muestra la ubicación al dispositivo del usuario.

2.3. Marco legal

La siguiente lista muestra las normas asociadas a la información que se deben cumplir en el marco del desarrollo del presente proyecto:

Ley 23 de 1982 Sobre derechos de autor: Esta norma reglamenta que los autores de obras literarias, científicas, y artísticas gozarán de protección para sus obras en la forma prescrita en esta Ley. También protege a los intérpretes o ejecutantes, en sus derechos conexos a los del autor (Congreso de la República, 1982).

Ley 1581 de 2012 Sobre protección de datos personales: Esta norma tiene como objetivo desarrollar el derecho constitucional que tienen todas las personas de conocer, actualizar y rectificar las informaciones que se haya recogido sobre ellas en una base de datos o archivos, y los demás derechos, libertades y garantías que se refiere el artículo 15 de la constitución política. Además, el artículo 5 de la ley 1581 de 2012, trata sobre el manejo de datos sensibles, aquellos que afectan la intimidad del titular o cuyo uso indebido puede generar su discriminación de alguna forma (Congreso de la República, 2012).

3. Aspectos Metodológicos

3.1. Descripción de la metodología

La metodología que se utilizó para este proyecto es Scrum. Esta es considerada una metodología ágil, ya que permite regular un conjunto de buenas prácticas para trabajar colaborativamente, y obtener el mejor resultado en el proyecto. Esto permite trabajar en equipo de manera altamente productiva. La ventaja de utilizar una metodología como Scrum es que permite al equipo dividir el proyecto en distintas tareas a lo largo del tiempo, por lo que ayuda a optimizar el tiempo de entrega de cada tarea.

Esta metodología está basada en entregas parciales y regulares del producto con base en el valor que ofrecen a los clientes. Además, permite al equipo organizarse y aprender a partir de las experiencias que se obtienen durante el proceso. Esto se lleva a cabo, con un equipo de trabajo conformado por: Product Owner, Scrum Master y Scrum Team.

3.1.1. Ritos

- **Sprint**

Es un periodo de tiempo que se le da a las tareas para ser resueltas una vez se planificaron en el sprint backlog, ya que esto supone entregas parciales para observar el avance del producto (Requena, 2018).

- **Sprint Review**

El objetivo de esta reunión es la revisión de los resultados obtenidos y estudiar las posibles modificaciones que podrían realizarse en el futuro. El equipo de Scrum junto con las partes

interesadas, revisan lo que se logró en el sprint y lo que se va a continuar realizando (Apoorva Srivastava, 2017).

- **Sprint Planning**

Esta es la reunión de planeación de las actividades que se desarrollarán durante el Sprint. En esta parte se discuten los elementos más importantes considerados por el Product Owner para llegar al objetivo del producto. La planeación del sprint está sujeta a los cambios que se crean pertinentes en el Product Backlog (Hurtado, 2021).

- **Scrum Daily**

El Scrum Daily tiene una duración de 15 minutos, y en él participa todo el equipo de desarrollo. Esto permite tener una buena comunicación y reducir la complejidad sobre las tareas que cada desarrollador y miembro del equipo está realizando (Hurtado, 2021).

El objetivo de la reunión diaria es inspeccionar lo que se ha ido realizando con base en el progreso diario hacia el objetivo del Sprint y así ir ajustando el Sprint Backlog según sea necesario para el próximo trabajo planificado.

Los desarrolladores para cumplir estos objetivos pueden utilizar las técnicas que deseen, siempre y cuando el progreso sea encaminado al objetivo del Sprint.

3.1.2. Artefactos

- **Product Backlog**

Se trata de un archivo genérico que recoge toda la información sobre las tareas, requerimientos y funcionalidades requeridas para el proyecto. Este documento puede ser

modificado por cualquier miembro del equipo, pero solo el Product Owner puede dar prioridades dentro del documento ya que es el encargado del mismo (Hurtado, 2021).

- **Sprint Backlog**

Es un documento en el que se recopilan todas las tareas que se van a realizar durante el sprint y los responsables de dichas tareas. Es importante tener en cuenta el tiempo que puede tomar resolver cada tarea asignada al iniciar el proyecto, porque durante la ejecución de las mismas pueden salir errores o situaciones que alteren el cronograma en que se estipularon las fechas de entrega inicialmente (Hurtado, 2021).

3.1.3. Roles

- **Product Owner**

Es la voz del cliente en todo el proceso del proyecto. El Product Owner se encarga de obtener el máximo valor con el menor costo posible, representando a las partes interesadas del proyecto, coordinando todos los entregables para lograr el objetivo propuesto, siendo claro en todas las ideas e indicaciones para el equipo (Hurtado, 2021).

- **Scrum Master**

El Scrum Master apoya al equipo en el entendimiento del objetivo planteado para el proyecto. Es el responsable de la efectividad del equipo, asegurándose de ir cumpliendo con los objetivos propuestos desde un inicio, realizando reuniones importantes para despejar dudas con respecto al proyecto o las tareas que están realizando cada uno de los integrantes. También hace

la tarea de entrenador, porque apoya y motiva al equipo, ayudando a que los tiempos de entrega se cumplan dentro del marco establecido (Hurtado, 2021).

- **Scrum Team**

Es el equipo encargado del desarrollo y entrega del producto. Cada uno es capaz de auto gestionarse y organizarse de manera estructural teniendo presente cada una de sus funciones dentro del proyecto que se está llevando a cabo (Hurtado, 2021).

3.2. Aplicación de la metodología

- **Roles**

- **Product Owner:** Este rol es desempeñado por la profesora María del Pilar Salamanca, ya que es la persona que tiene el rol del cliente del proyecto.
- **Scrum Master:** Este rol es desempeñado por la profesora María del Pilar Salamanca, apoyando al equipo de desarrollo en el proceso de entregas del proyecto, resolviendo las dudas acerca de la solución de las tareas propuestas.
- **Development Team:** Este rol lo desempeña el estudiante David Santiago Rey Beltrán, siendo el responsable del desarrollo de todo el proyecto, evaluando cada tarea asignada y realizando las entregas en cada uno de los Sprints.

- **Artefactos**

- **Product Backlog:** El backlog es realizado por el equipo de desarrollo que en este caso es el estudiante David Santiago Rey Beltrán, pero dirigido por el Product Owner que es la profesora María del Pilar Salamanca, quien indica las tareas y requerimientos, además de las historias de usuario que debe realizar el estudiante para la solución del proyecto.

- **Sprint Backlog:** Para el sprint Backlog el equipo de desarrollo junto con el Product Owner, define las historias de usuario que se desarrollarán en el sprint y los tiempos para realizar cada una de ellas, según lo planeado en el Product Backlog.
- **Ritos**
 - **Sprint Planning:** La planeación del Sprint la realiza el equipo de desarrollo conformado por el estudiante David Santiago Rey Beltrán, en el que se hace la revisión de las actividades que se realizaron en el review.
 - **Sprint Daily:** En este proyecto, los ritos de la metodología se adaptan, por limitaciones de tiempo, a una reunión semanal. En estos Dailys se reunieron el Product Owner, el Scrum Master y el equipo de desarrollo en este caso una vez a la semana para ver el avance que se tuvo en las tareas asignadas.

3.3 Aplicación de la metodología

3.2.1.Sprint 1 – Configuración de herramientas y plan de trabajo

En esta fase se realizaron las siguientes tareas:

- Definir las funcionalidades que va a tener la aplicación
- Configuración de herramientas de trabajo
- Requerimientos del proyecto
- Definir el propósito de la aplicación

En esta fase se obtuvieron los siguientes productos:

- Configuración de herramientas de desarrollo
- Requerimientos y funcionalidades del proyecto

3.2.2.Sprint 2 – Análisis de mensajería instantánea

En esta fase se realizaron las siguientes tareas:

- Selección del medio de mensajería instantánea
- Registrar usuario con epilepsia
- Configurar sonido de alarma
- Registrar ubicación del dispositivo
- Editar información de registro
- Enlace de mensajería instantánea con la aplicación
- Desarrollar interfaz de la aplicación
- Configurar base de datos

En esta fase se obtuvieron los siguientes productos:

- Enlace de aplicación de mensajería instantánea
- Interfaz

3.2.3.Sprint 3 – Envío de mensajes con datos precargados y codificación de la integración

En esta fase se realizaron las siguientes tareas:

- Guardado de información para envío de mensaje
- Prueba de envío con mensaje precargado
- Configurar alerta
- Funcionalidades del botón en la aplicación
- Implementación botón del celular con la aplicación
- Desarrollo de la función para futura conexión con dispositivo Bluetooth

En esta fase se obtuvieron los siguientes productos:

- Envío de mensajes precargados

- Implementación del botón con la aplicación

3.2.4.Sprint 4 – Muestra datos en pantalla

En esta fase se realizaron las siguientes tareas:

- Generación de alarma al oprimir el botón
- Configuración de la alarma
- Mostrar datos en pantalla del celular emisor
- Activación ubicación en tiempo real en el celular receptor

En esta fase se obtuvieron los siguientes productos:

- Muestra de datos en pantalla
- Envío de datos de alerta

4. Desarrollo del proyecto

En este capítulo se describirá la aplicación de la metodología Scrum para el desarrollo del proyecto.

4.1 Descripción de la aplicación

EpilepsyAlarm es una aplicación móvil de botón de pánico, que permite a los usuarios que sufren ataques de epilepsia, registrarse y configurar un contacto de emergencia, para posteriormente, cuando sientan que les va a sobrevenir un ataque de epilepsia, pedir ayuda mediante la aplicación. La ayuda se solicita al oprimir el botón de bloqueo del celular en el que se encuentra instalada la aplicación, lo que activará el envío de la ubicación del paciente por medio de un mensaje, a la persona que registraron anteriormente, de manera que esta persona podrá ver la ubicación del usuario, indicándole que está padeciendo un ataque; además, la aplicación activa una alarma en el celular del usuario, lo que permite alertar a las personas que se encuentran alrededor de la víctima, para que la puedan asistir de manera oportuna, y si observan en el dispositivo del usuario, encontrarán instrucciones para atender a la persona durante el ataque de epilepsia.

4.2 Fase de iniciación

Para esta fase se siguieron los lineamientos que establece la metodología Scrum para comenzar el proyecto. En este orden de ideas, se determinaron los roles y tareas que se llevarían a cabo a lo largo de todo el proceso de realización del proyecto, lo cual se acordó en una reunión con la directora del proyecto.

4.3 Planeación y estimación

4.3.1 Roles en la aplicación

- **Paciente de epilepsia:** es el rol encargado de registrarse en la aplicación, con su información personal como: nombre completo, tipo de sangre, número de documento de identidad, nombre de contacto de emergencia y el número de teléfono del contacto de emergencia. Una vez ingrese los datos solicitados podrá configurar la alarma y la ubicación actual. Cuando se completa todo el proceso inicial, ya no es necesario volver a registrar los datos en la aplicación, e ingresará de manera normal al dashboard; la aplicación permite editar y configurar el mensaje que se enviará al contacto registrado, la ubicación, el tono de alarma y los datos que se ingresaron inicialmente.
- **Persona que se encuentra cerca del paciente:** es quien observará en la pantalla del dispositivo del paciente las instrucciones para el manejo del ataque de epilepsia.
- **Contacto de emergencia:** esta persona es quien recibirá la notificación del paciente, con la ubicación y el mensaje de alerta.

4.3.2 Arquitectura y herramientas

Para la arquitectura de esta aplicación se utilizó el patrón de arquitectura Modelo-Vista-Modelo (MVVM), este patrón permite desacoplar al máximo toda la interfaz del usuario en cuanto a la lógica de la aplicación, evitando que la parte visual y la capa de datos tengan un solo comportamiento, creando así una capa visual, una capa de datos y una tercera capa en donde se realiza la comunicación entre la vista y la capa de datos (Stonis, 2022).

Esta arquitectura está representada en 3 elementos:

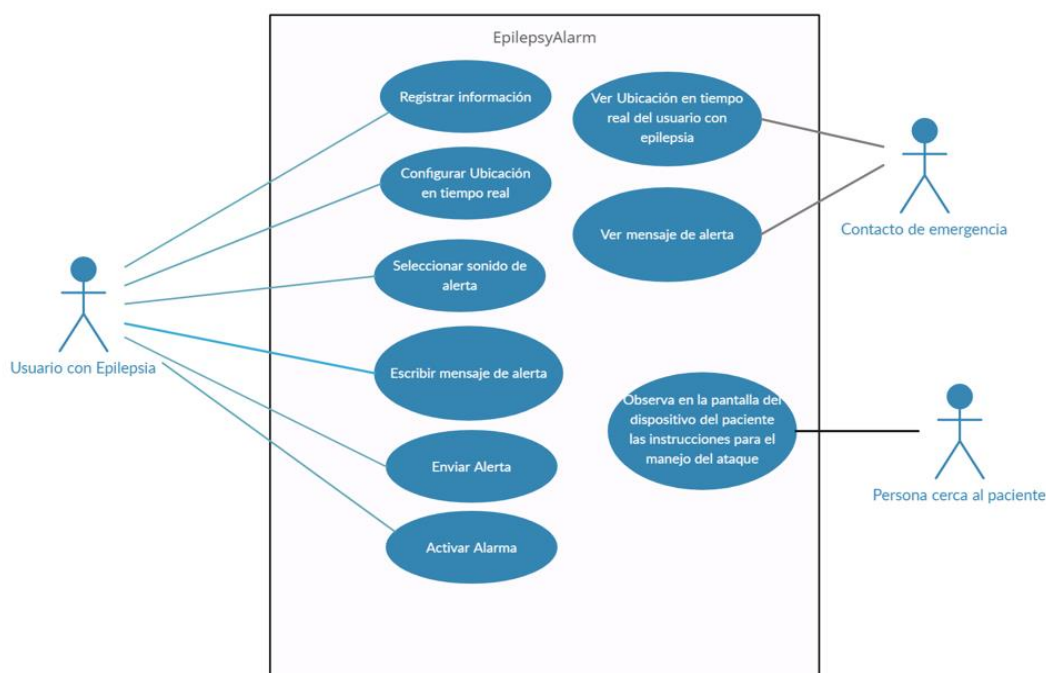
- **Modelo:** representa la información y la lógica de negocio de la aplicación
- **Vista:** es la interfaz de usuario que se presenta al usuario final
- **VistaModelo:** es el intermediario entre la vista y el modelo, que se encarga de manejar la lógica de presentación de la vista y la lógica de negocio del modelo.

Para el almacenamiento de información se utilizó SQLite, guardando los datos del usuario de manera local, y para ello se empleó la librería Room, que permite acceder a la información de manera sencilla y sin problemas.

A continuación, se presenta el diagrama de casos de uso que se utilizó en la arquitectura de la aplicación.

Ilustración 1

Diagrama de casos de uso

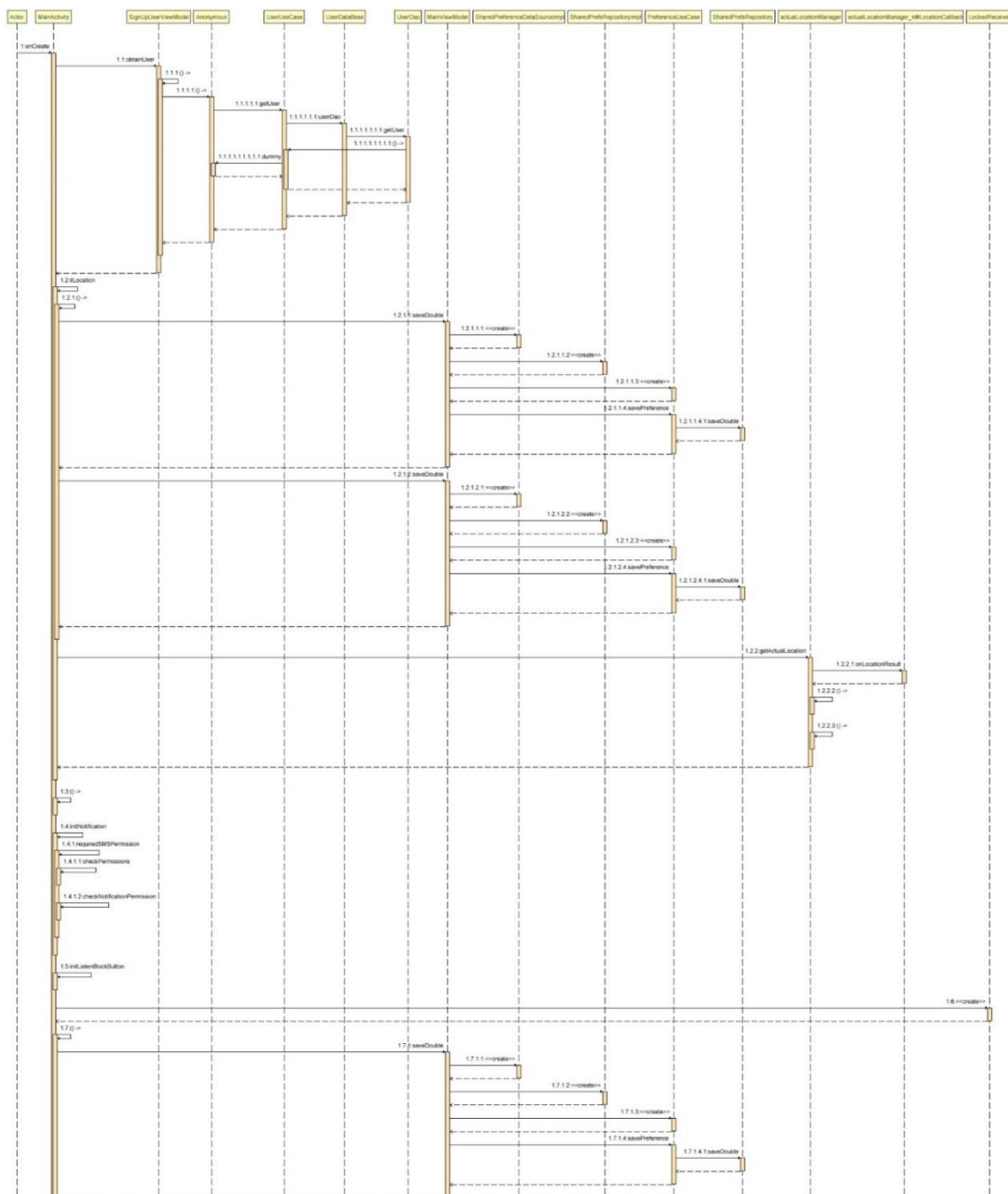


Fuente. elaboración propia

En el siguiente diagrama de secuencia se puede observar la interacción de las clases de la actividad principal de la aplicación (ilustración 2). Para una mejor comprensión se va a dividir el diagrama y su funcionamiento será explicado más detalladamente.

Ilustración 2

Diagrama de secuencia

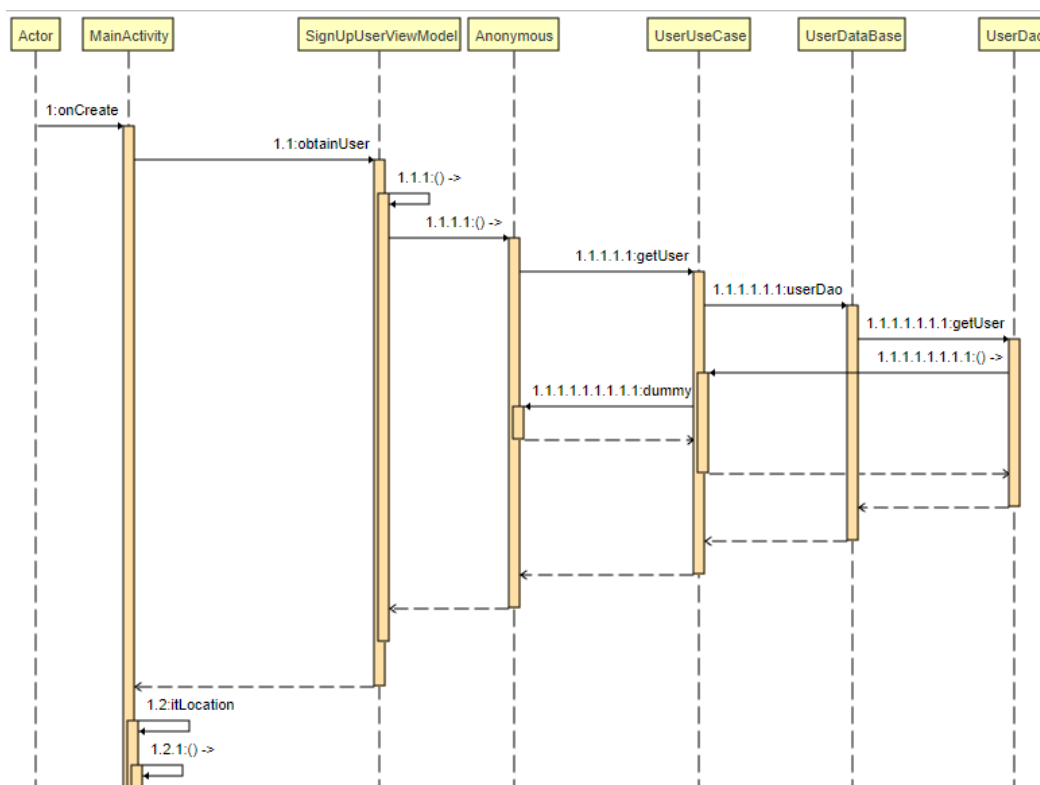


Fuente. elaboración propia

En la primera parte del diagrama de secuencia se puede observar la actividad principal “MainActivity”, en donde se maneja la parte fundamental de todo el código, seguidamente de la clase “SignUpUserViewModel” en la cual se guarda y se obtiene la información que el usuario registró al configurar por primera vez la aplicación. En la clase “UserUseCase” se tiene el método “getUser” el cual obtiene el usuario creado de la base de datos “UserDataBase” la cual fue generada inicialmente, para guardar el usuario y posteriormente para traer su información.

Ilustración 3

Diagrama de secuencia parte 1

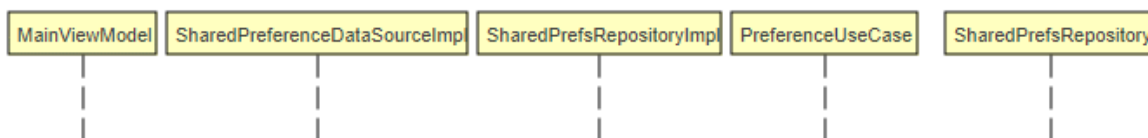


Fuente. elaboración propia

La segunda parte de este diagrama de secuencia se puede observar en la ilustración 4 y 5. La clase “MainViewModel”, que contiene el método “saveDouble” se encarga de guardar el sonido de la alarma y la información de la ubicación del dispositivo. La clase “SharedPreferenceDataSourceImpl” proporciona métodos para guardar y obtener datos de tipo booleano, double y entero, utilizando claves únicas. La clase “SharedPrefsRepositoryImpl” es una implementación de la interfaz “SharedPrefsRepository”, actuando como intermediario entre la lógica de la aplicación y la clase mencionada inicialmente que proporciona método para guardar y obtener datos.

Ilustración 4

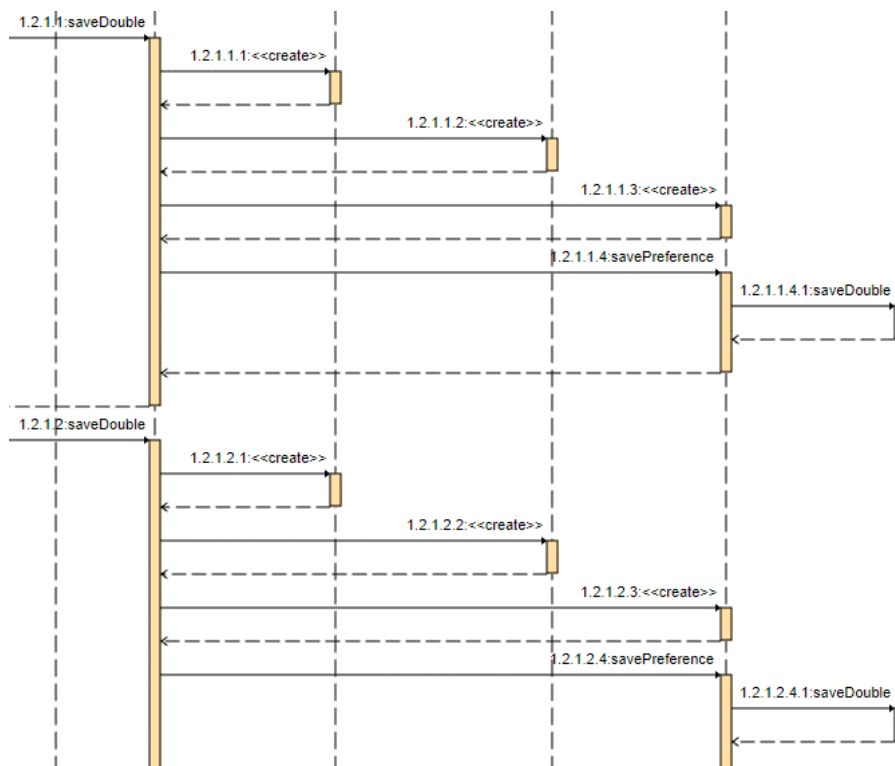
Diagrama de secuencia parte 2



Fuente. elaboración propia

Ilustración 5

Complemento diagrama secuencia parte 2

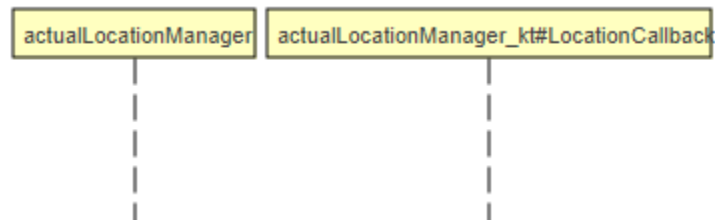


Fuente. elaboración propia

Para la tercera parte de este diagrama, mostrada en las ilustraciones 6 y 7, contamos con dos clases “actualLocationManager” y “actualLocationManager Location Callback”, que permiten obtener la información de la ubicación actual en la que se encuentra el dispositivo. Para ello utiliza el método “getActualLocation” para obtener la información actual de ubicación, y el método “onLocationResult” se usa para obtener las coordenadas de dicha localización.

Ilustración 6

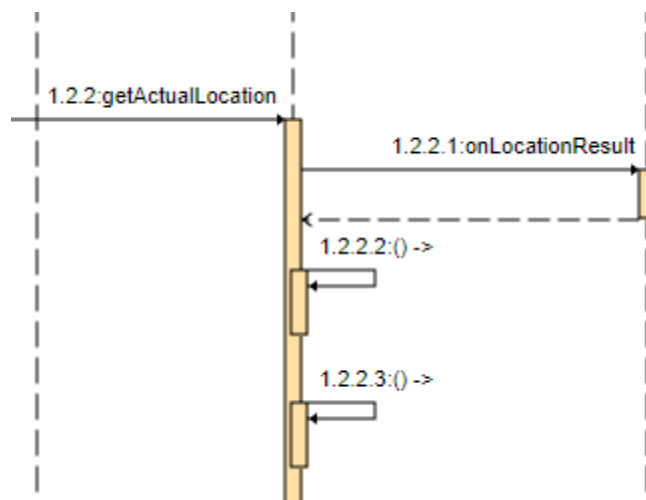
Diagrama de secuencia - localización parte 3



Fuente. elaboración propia

Ilustración 7

Complemento Diagrama secuencia localización parte 3



Fuente. elaboración propia

4.4 Sprint 1

En la reunión inicial con el Product Owner, se acordaron las tareas iniciales que se iban a entregar en la siguiente reunión. Estas tareas iniciales se enfocaron en preparar el entorno de desarrollo y definir los requerimientos del usuario realizando las historias de usuario.

4.4.1 Historias de usuario

Para la fase inicial de este proyecto y cumpliendo con las tareas de este primer sprint, se realizaron las historias de usuario, lo que permite darle un enfoque más claro a las funcionalidades que tiene esta aplicación. Para ello se definieron las historias de usuario que se muestran en el Sprint backlog, presentado en la siguiente tabla.

Tabla 2.

Product Backlog

HISTORIAS DE USUARIO		
NOMBRE	CÓDIGO	SPRINT EN EL QUE SE DESARROLLA
Registro usuario con epilepsia	001	2
Configuración alerta	002	3
Registrar ubicación del dispositivo	003	2
Editar información de registro	004	2
Configurar mensaje de alerta	005	4
Enviar ubicación en tiempo real	006	4
Enviar mensaje de alerta por mensajería instantánea	007	3
Mostrar instrucciones en pantalla para manejo del paciente durante el ataque	008	4
Creación de clase para futura integración con dispositivo Bluetooth	009	3
Configuración de sonido de alarma	010	2

Nota. Elaboración propia

4.4.2 Diseño de Mockups

Para el diseño visual de cada historia de usuario se diseñaron unos mockups, que tienen como finalidad observar de qué forma se verá la aplicación. Estos mockups se realizaron con un diseño sencillo, para que el usuario se familiarice fácilmente con el funcionamiento de la aplicación. Para el diseño de cada mockup se utilizó la herramienta Balsamiq, ya que es una herramienta fácil de usar y ofrece opciones para el manejo visual de la aplicación.

En la ilustración 3 se observa el diseño de la pantalla de inicio de la aplicación. Esta pantalla se despliega la primera vez que el usuario va a utilizar la aplicación. Se puede ver un espacio en el cual se muestran las instrucciones para el manejo del paciente durante el ataque de epilepsia, y un botón para iniciar el registro.

Ilustración 8

Mockup inicio de la aplicación

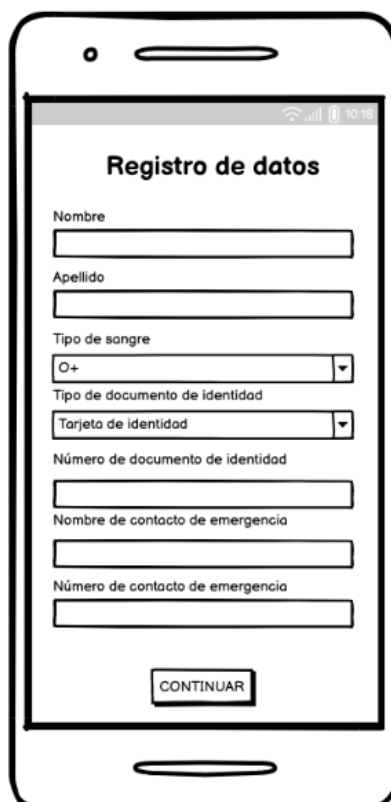


Fuente. elaboración propia

En la ilustración 9 se puede observar el formulario de registro con los campos que deberán ser llenados con los datos del usuario de la aplicación y del contacto de emergencia a quien se le enviará el mensaje de alerta cuando se active la aplicación.

Ilustración 9

Mockup registro del usuario



Registro de datos

Nombre

Apellido

Tipo de sangre

Tipo de documento de identidad

Número de documento de identidad

Nombre de contacto de emergencia

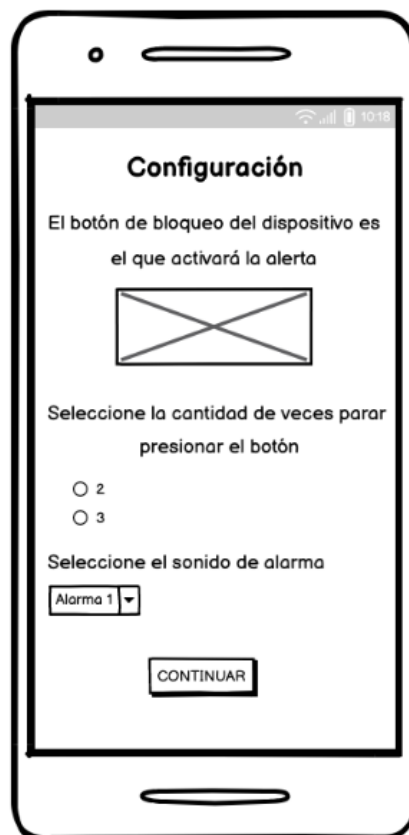
Número de contacto de emergencia

Fuente. elaboración propia

En la ilustración 10 se muestra la pantalla en donde se va a configurar la cantidad de veces que el usuario desea presionar el botón de bloqueo para enviar la alerta, y el sonido que desea activar cuando se oprima el botón.

Ilustración 10

Mockup configuración de alarma



Fuente. elaboración propia

En la Ilustración 11 se presenta la pantalla de configuración de la ubicación. Al instalar la aplicación, se le solicita al usuario activar la ubicación y, en caso de haber rechazado la activación en ese momento, aquí tendrá una nueva oportunidad para ponerla en funcionamiento. Si en esta pantalla la rechaza, se le mostrará un mensaje informándole que la aplicación no podrá enviar la ubicación cuando active la alerta. Adicionalmente, si acepta usar la localización, se mostrará en el mapa la ubicación actual del usuario.

Ilustración 11

Mockup activar ubicación

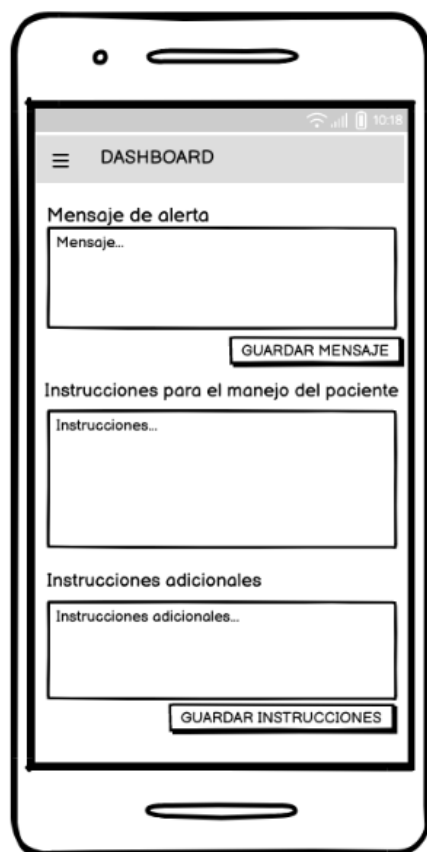


Fuente. elaboración propia

En la ilustración 12, se presenta la pantalla principal de la aplicación donde se puede observar un apartado para configurar el mensaje de alerta, otro apartado donde se pueden ver las instrucciones para el manejo del paciente durante el ataque de epilepsia y uno más para seleccionar el sonido que desea el usuario suene cuando se active la alerta.

Ilustración 12

Mockup Dashboard

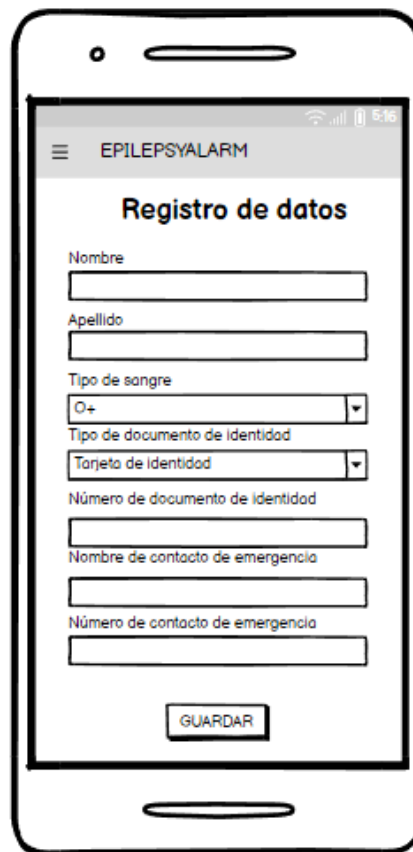


Fuente. elaboración propia

En la ilustración 13, se muestra la pantalla en donde se pueden editar los datos del usuario de la aplicación, en caso de que el contacto de confianza no se encuentre en la ciudad y desee enviar el mensaje de alerta a un nuevo contacto.

Ilustración 13

Perfil del usuario



The image shows a mobile application interface for 'EPILEPSYALARM'. The screen displays a 'Registro de datos' (Data Registration) form. The form includes the following fields and controls:

- Nombre**: A text input field.
- Apellido**: A text input field.
- Tipo de sangre**: A dropdown menu with 'O+' selected.
- Tipo de documento de identidad**: A dropdown menu with 'Tarjeta de Identidad' selected.
- Número de documento de identidad**: A text input field.
- Nombre de contacto de emergencia**: A text input field.
- Número de contacto de emergencia**: A text input field.

At the bottom of the form is a button labeled 'GUARDAR' (Save).

Fuente. elaboración propia

Al finalizar este sprint, se obtuvieron el product backlog, las historias de usuario, el diseño de los mockups de la aplicación, además, la configuración de todo el entorno de desarrollo y el análisis de las funcionalidades de la aplicación, dando así cumplimiento a las actividades previstas para el primer sprint.

4.5 Sprint 2

Durante esta segunda etapa del desarrollo de la aplicación, se trabajó en la creación de la base de datos con el formulario y el medio de mensajería por el que se enviaría el mensaje de alerta, además de la interfaz visual de la aplicación, entre otros.

4.5.1 Formulario

Cuando se abre por primera vez la aplicación aparece una pantalla de inicio. Posteriormente se muestra el formulario en donde el usuario va a almacenar sus datos y los datos del contacto de emergencia. Para guardar esta información se utilizó la librería Room, que ofrece el lenguaje Kotlin, para el almacenamiento de datos de manera local, sin la necesidad de un servidor (Ilustración 14).

Ilustración 14

Librería Room

```
def room_version = "2.5.0"
implementation("androidx.room:room-runtime:$room_version")
```

Fuente. elaboración propia

4.5.2 Selección del medio de mensajería instantánea

Para la selección del medio de mensajería instantánea adecuado para el envío del mensaje de alerta, se investigó cuáles son las aplicaciones más populares para el envío de mensajes y que además tuvieran una API a la cual poder acceder. Se seleccionó la red social WhatsApp por ser la aplicación más utilizada en el mundo para el envío de mensajes, además cuenta con una API que permite enviar mensajes sin necesidad de que el usuario lo haga directamente.

4.5.2.1 Enlace de mensajería con la aplicación

Para el enlace de EpilepsyAlarm con una API de mensajería instantánea, en este caso WhatsApp, fue necesario utilizar la API de WhatsApp Business (Ilustración 15), para ello fue importante tener en cuenta el token que brinda la API (Ilustración 16), y el siguiente código que permite enlazar la API con la aplicación (Ilustración 17).

Ilustración 15

Interfaz para agregar mensaje

```
import retrofit2.Response
import retrofit2.http.Body
import retrofit2.http.Headers
import retrofit2.http.POST

interface MessageApi {

    @Headers("Content-Type: application/json")
    @POST("messages")
    suspend fun postMessage(@Body message: MessageEntity) : Response<MessageData>

}
```

Fuente. elaboración propia

Ilustración 16

Código de autorización con Token

```
private fun getRetrofit(): Retrofit {
    // Crear un objeto OkHttpClient con un Interceptor que agrega el encabezado Authorization
    val client = OkHttpClient.Builder()
        .addInterceptor { chain ->
            val original = chain.request()

            val request = original.newBuilder()
                .header("Authorization", value: "Bearer EAAkt4SA1Vy0BAHU082BrQw1ixwvXaAN56h0bwSPa19zZA06xy91HyDyp4ZA7F4onX5LAbsagEkE4UKMexHDKUqJT2A")
                .method(original.method(), original.body())
                .build()

            chain.proceed(request)
        }
        .build()
}
```

Fuente. elaboración propia

Ilustración 17

Enlace con la API

```
return Retrofit.Builder() Retrofit.Builder
    .baseUrl("https://graph.facebook.com/v16.0/108029642271423/") Retrofit.Builder
    .client(client)
    .addConverterFactory(GsonConverterFactory.create())
    .build()
```

Fuente. elaboración propia

4.5.3 Interfaz de la aplicación

Se replicaron los mockups diseñados en el sprint anterior, pero con un diseño más interesante para el usuario, en donde se puede apreciar detalles que no se observan en los mockups, además de una usabilidad intuitiva para el usuario.

Esta primera etapa de la aplicación, se enfoca en las historias de usuario 001, 003, 004 y 010, explicadas en las tablas 2, 3, 4 y 5.

Tabla 3.

Historia de usuario “Registrar usuario con epilepsia”

HISTORIA DE USUARIO	
NOMBRE HISTORIA DE USUARIO: Registrar usuario con epilepsia	
CÓDIGO: 001	FECHA: 20 de febrero
DESCRIPCIÓN: EpilepsyAlarm permite al usuario que padece epilepsia, registrarse con los siguientes datos. - Nombre - Apellido - Tipo de sangre	PRIORIDAD: Alta
	PROGRAMADORES ENCARGADOS: Santiago Rey
	ROL: Usuario con epilepsia

<ul style="list-style-type: none"> - Documento de identidad - Nombre de contacto de emergencia - Número de contacto de emergencia 	
<p>CRITERIOS DE ACEPTACIÓN:</p> <ul style="list-style-type: none"> - Solo se puede continuar con el registro si se llenaron los campos solicitados - La información se guarda en la base de datos localmente - Se continua luego de validar cada campo con la información del usuario 	
<p>MOCKUP: Ilustración 9</p>	

Fuente. elaboración propia

Como se observa en la ilustración 9, se diseñó el mockup para el formulario el cual el usuario deberá llenar, ya que la información que se guarde, será la que aparecerá en pantalla al momento de padecer de un ataque de epilepsia, permitiendo ser socorrido de manera oportuna y facilitándole los datos del paciente a las personas que se encuentren cerca y deseen ayudar. En la ilustración 18, se observa la pantalla del registro de usuario.

Ilustración 18

Registro de usuario

The image shows a mobile application interface for 'EPILEPSYALARM'. The screen displays a registration form titled 'Registro de datos'. The form includes the following fields and values:

- Nombre: Santiago
- Apellido: Rey
- Tipo de sangre: O+ (dropdown menu)
- Tipo de documento: Cedula de ciudadanía (dropdown menu)
- Número de documento de identidad: 1032569234
- Nombre Contacto de Emergencia: Andrés González
- Número contacto de emergencia: 3152369523

At the bottom of the form is a blue button labeled 'CONTINUAR'.

Fuente. elaboración propia

Para el registro de la ubicación en tiempo real fue necesario incluir unos permisos en el código que son los que permiten implementar los métodos para pedir activación de ubicación en tiempo real y observar al mismo tiempo el sitio exacto donde se encuentra el dispositivo móvil. Estos permisos se piden luego de llenar el formulario de registro. Ilustración 18. En la ilustración 19 se puede observar el resultado del prototipo para la activación de la ubicación.

Tabla 4.**Historia de usuario “Registrar ubicación en tiempo real”**

HISTORIA DE USUARIO	
NOMBRE HISTORIA DE USUARIO: Registrar ubicación en tiempo real	
CÓDIGO: 003	FECHA: 23 de febrero
DESCRIPCIÓN: Para terminar el registro y configuración en EpilepsyAlarm, se debe tener la ubicación activada y seleccionar en el mapa el lugar en tiempo real del dispositivo	PRIORIDAD: Alta
	PROGRAMADORES ENCARGADOS: Santiago Rey
	ROL: Usuario con epilepsia
CRITERIOS DE ACEPTACIÓN: - Se debe tener la ubicación del dispositivo activada - La aplicación generará un mensaje si no se ha activado la ubicación	
MOCKUP: Ilustración 11	

Fuente. elaboración propia

Ilustración 19

Registrar ubicación



Fuente. elaboración propia

Tabla 5.

Historia de usuario “Editar información”

HISTORIA DE USUARIO	
NOMBRE HISTORIA DE USUARIO: Editar información	
CÓDIGO: 004	FECHA: 14 de febrero
DESCRIPCIÓN: Después de configurar el registro de usuario, en el dashboard podrá editar la información ingresada inicialmente	PRIORIDAD: Media
	PROGRAMADORES ENCARGADOS: Santiago Rey
	ROL: Usuario con epilepsia
CRITERIOS DE ACEPTACIÓN: - No podrá ingresar al dashboard de la aplicación sin completar el proceso de registro	

MOCKUP: Ilustración 9

Fuente. elaboración propia

Cuando el registro esté completo, el usuario ingresará al dashboard de la aplicación, donde verá un menú que le permite cambiar la configuración de la alarma y los datos del usuario (ilustración 20)

Ilustración 20

Menú de usuario



Fuente. elaboración propia

Para la configuración del sonido que el usuario elija al momento de ser activada la alerta durante el ataque, se presentan varias opciones para elección del usuario, ilustración 21, de una manera más explícita como se puede observar en la tabla 6.

Tabla 6.

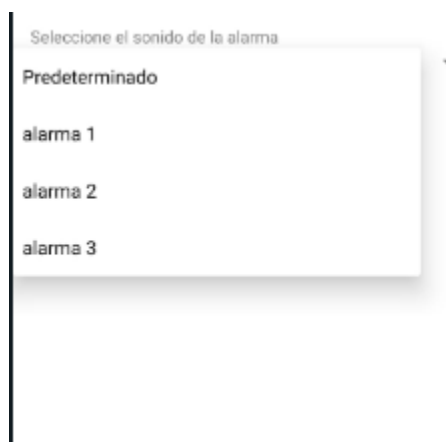
Historia de usuario “Configuración de sonido de alarma”

HISTORIA DE USUARIO	
NOMBRE HISTORIA DE USUARIO: Configuración de sonido de alarma	
CÓDIGO: 010	FECHA: 4 de abril
DESCRIPCIÓN: Se seleccionará un sonido para que cuando el usuario active la alerta, suene durante el ataque.	PRIORIDAD: Alta
	PROGRAMADORES ENCARGADOS: Santiago Rey
	ROL: Usuario con epilepsia
CRITERIOS DE ACEPTACIÓN: - La alarma sonará con un sonido predeterminado por la aplicación si el usuario no seleccionó ninguna de las opciones de alarma	
MOCKUP: Ilustración 10	

Fuente. elaboración propia

Ilustración 21

Configurar sonido alarma



Fuente. elaboración propia

4.6 Sprint 3

La tercera etapa del desarrollo de la aplicación se enfoca en el guardado de la información, el envío de mensajes de alerta y la implementación de la funcionalidad del botón.

4.6.1 Guardado de información

Una vez se llenen todos los campos, la información se almacena en una base de datos local. Para ello se utilizó la librería Room (ilustración 22), ofrecida por Android, junto con la base de datos local SQLite (ilustración 23).

Ilustración 22

Librería Room

```
import androidx.room.Database
import androidx.room.RoomDatabase
import com.sr.configuration.data.dao.UserDao
import com.sr.configuration.data.User

@Database(
    entities = [User::class],
    version = 1
)
abstract class UserDataBase: RoomDatabase() {
    abstract fun userDao(): UserDao
}
```

Fuente. elaboración propia

Ilustración 23

Guardado de información

```

private val _user = MutableLiveData<UserModel>()
val user = _user as LiveData<UserModel>
val userUseCase = UserUseCase()
fun saveUser(
    etName: String,
    etLastName: String,
    etRh: String,
    etNumId: String,
    etNameContact: String,
    etPhoneContact: String
) {
    _user.value = UserModel( id: 1, etName, etLastName, etRh, etNumId, etNameContact, etPhoneContact)
    viewModelScope.launch { this: CoroutineScope
        withContext(Dispatchers.IO){ this: CoroutineScope
            _user.value?.let { userUseCase.createUser(it) }
        }
    }
}

```

Fuente. elaboración propia

4.6.2 Envío de mensajes de alerta

Una de las partes más importantes de EpilepsyAlarm es el envío de mensajes de alerta, para ello se implementó el siguiente código (ilustración 24) que permite enviar el mensaje al número de celular que haya registrado el usuario como su contacto de confianza.

Ilustración 24

Código de envío de mensaje

```

private fun startFunction(context: Context) {
    val result = goAsync()
    val coroutineScope = CoroutineScope(Dispatchers.Default)

    coroutineScope.launch { this: CoroutineScope
        val messageRepository = MessageRepository()
        messageRepository.sendMessage(context)
        messageRepository.sendSMS(context)
        result.finish()
    }
    showMessage(context)
    NotificationManager.showNotification(context)
}

```

Fuente. elaboración propia

4.6.3 Funcionalidad de botón de pánico

Para realizar esta segunda etapa de la aplicación, se enfoca el desarrollo en las historias de usuario explicadas en las tablas 7, 8.

Tabla 7.

Historia de usuario “Configurar alerta”

HISTORIA DE USUARIO	
NOMBRE HISTORIA DE USUARIO: Configurar alerta	
CÓDIGO: 002	FECHA: 25 de marzo
DESCRIPCIÓN: EpilepsyAlarm le pedirá al usuario configurar la cantidad de veces que quiere oprimir el botón para activar la alarma y el sonido que quiere que suene al activarla	PRIORIDAD: Media
	PROGRAMADORES ENCARGADOS: Santiago Rey
	ROL: Usuario con epilepsia
CRITERIOS DE ACEPTACIÓN: - Se debe seleccionar un número de veces a oprimir el botón para activar la alerta - Se debe seleccionar un sonido para la alerta	
MOCKUP: Ilustración 10	

Fuente. elaboración propia

Para la configuración de la alerta, en donde el usuario puede establecer la cantidad de veces que desea oprimir el botón de bloqueo del dispositivo, se realizó el siguiente código que permitió obtener esta funcionalidad (ilustración 25).

Ilustración 25

Código fuente configurar botón

```

override fun onReceive(context: Context, intent: Intent?) {
    if (intent?.action == Intent.ACTION_SCREEN_ON) {
        val clickTime = System.currentTimeMillis()
        println("tiempo respuesta: ${clickTime - lastClickTime}")

        if ((clickTime - lastClickTime) < 5000 ) {
            if (numClicks < pulseCount) {
                numClicks++
            } else if (numClicks == pulseCount) {
                startFunction(context)
                numClicks = 0
            }
            //Menos de 2 segundos
            numClicks ++
        }else{
            numClicks = 1 //Reinicia el contador
        }
        lastClickTime = clickTime

        if(numClicks == 3){
            startFunction(context)
            numClicks = 0
        }
    }else{
        NotificationManager.clearNotification(context)
    }
}

```

Fuente. elaboración propia

Para que el mensaje de alerta sea enviado con satisfacción es necesario que se haya configurado previamente la aplicación, como se puede observar en la historia de usuario 7, de la tabla 8.

Tabla 8.

Historia de usuario “Enviar mensaje de alerta por mensajería”

HISTORIA DE USUARIO	
NOMBRE HISTORIA DE USUARIO: Enviar mensaje de alerta por mensajería instantánea	
CÓDIGO: 007	FECHA: 14 de marzo

DESCRIPCIÓN: Se podrá enviar el mensaje de alerta que se guardó después de registrarse el usuario en la aplicación, con el botón de bloqueo oprimido las veces que el usuario seleccionó al descargar la aplicación	PRIORIDAD: Alta
	PROGRAMADORES ENCARGADOS: Santiago Rey
	ROL: Usuario con epilepsia
CRITERIOS DE ACEPTACIÓN: - El mensaje solo se enviará si se oprimió el botón de bloqueo la cantidad de veces que se configuró al iniciar la aplicación - El mensaje se enviará con la última ubicación solo sí, la localización del dispositivo está activa - El mensaje se enviará si se guardó correctamente	
MOCKUP: Ilustración 12	

Fuente. elaboración propia

4.6.4 Conexión a un dispositivo físico de Botón de Pánico vía Bluetooth

Se desarrolla una clase para realizar una futura integración con un dispositivo físico de botón de pánico mediante Bluetooth. Para ello se realizó el siguiente código, mostrado en la Ilustración 26, lo que permite mediante el objeto `pairedDevices` obtener la lista de los dispositivos que se encuentran emparejados, para posteriormente, en la condición `if`, agregar los dispositivos emparejados a un arreglo. De una manera más detallada el código funciona de la siguiente manera:

- Crea una lista vacía **list** para almacenar los dispositivos Bluetooth emparejados.
- Verifica si hay dispositivos emparejados en **pairedDevices**. Si la lista no está vacía, itera sobre cada dispositivo en **pairedDevices**.
- Dentro del bucle, agrega cada dispositivo a la lista **list** utilizando el método **add(device)**.

- Si la lista de dispositivos emparejados está vacía, se registra un mensaje de información en el registro ("No se encontraron dispositivos emparejados").
- Crea un adaptador de ArrayAdapter utilizando *android.R.layout.simple_list_item_1* como diseño de elemento de lista y la lista *list* como origen de datos.
- Establece el adaptador en un *ListView* llamado *listView* para mostrar los dispositivos emparejados.
- Configura un *OnItemClickListener* en el *listView* para manejar la selección de elementos de la lista. Cuando se selecciona un elemento, se obtiene el dispositivo Bluetooth correspondiente desde la lista *list* utilizando la posición seleccionada.
- A continuación, se obtiene la dirección Bluetooth del dispositivo seleccionado utilizando *device.address*.
- Se crea un *Intent* para iniciar una nueva actividad llamada *ControlActivity* y se le agrega la dirección Bluetooth como un extra utilizando *intent.putExtra(EXTRA_ADDRESS, address)*.
- Finalmente, se inicia la actividad *ControlActivity* utilizando *startActivity(intent)*.

De esta manera se da cumplimiento a la historia de usuario 009, que se muestra la tabla 9.

Ilustración 26

Fragmento de código para conexión

```

Santiago Rey *
private fun showPairedDevices() {
    // pairedDevices = bluetoothAdapter.bondedDevices
    val list: ArrayList<BluetoothDevice> = ArrayList()

    if (pairedDevices.isNotEmpty()) {
        for (device: BluetoothDevice in pairedDevices) {
            list.add(device)
            // Log.i("Device", "${device.name} - ${device.address}")
        }
    } else {
        Log.i(tag: "Device", msg: "No paired devices found.")
    }

    val adapter = ArrayAdapter(requireContext(), android.R.layout.simple_list_item_1, list)
    listView.adapter = adapter

    listView.setOnItemClickListener { _, _, position, _ ->
        val device: BluetoothDevice = list[position]
        val address: String = device.address

        val intent = Intent(requireContext(), ControlActivity::class.java)
        intent.putExtra(EXTRA_ADDRESS, address)
        startActivity(intent)
    }
}

```

Fuente. elaboración propia

Tabla 9.

Historia de usuario "Clase para conectar un dispositivo Bluetooth"

HISTORIA DE USUARIO	
NOMBRE HISTORIA DE USUARIO: Creación de clase para futura integración con dispositivo Bluetooth	
CÓDIGO: 009	FECHA: 29 de abril

DESCRIPCIÓN: Clase que permite implementar un dispositivo de hardware por conexión Bluetooth, permitiendo ofrecer una funcionalidad única al botón de pánico.	PRIORIDAD: Alta
	PROGRAMADORES ENCARGADOS: Santiago Rey
	ROL: Usuario con epilepsia
CRITERIOS DE ACEPTACIÓN: - Para integrar un dispositivo por medio de Bluetooth es necesario que el dispositivo pueda conectarse por Bluetooth	
MOCKUP:	

Fuente. elaboración propia

Para el envío del mensaje por medio del botón de bloqueo del dispositivo, se utilizó el siguiente código (ilustración 27), que permite registrar la acción realizada y enviar la información que se tiene almacenada en la base de datos local.

Ilustración 27

Código fuente para envío de mensaje

```

    registerReceiver(lockedReceiver, filter)
}

override fun onDestroy() {
    super.onDestroy()
    unregisterReceiver(lockedReceiver)
}

override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
    val notificationBuilder = NotificationCompat.Builder(context, channelId)
        .setSmallIcon(R.drawable.ic_launcher_background)
        .setContentTitle("Epilepsy Alarm")
        .setContentText("Tu mensaje aqui")
        .setPriority(NotificationCompat.PRIORITY_HIGH)

    val channel = NotificationChannel(
        channelId,
        channelName,
        NotificationManager.IMPORTANCE_HIGH
    )
    val notificationManager =
        getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
    notificationManager.createNotificationChannel(channel)

    startForeground(id, notificationBuilder.build())
}

```

Fuente. elaboración propia

4.7 Sprint 4

Esta última etapa de desarrollo de la aplicación se enfocó en la generación del sonido de la alarma cuando se oprima el botón del dispositivo, además de mostrar los datos cuando se active la alarma y el envío de la ubicación en tiempo real del dispositivo.

4.7.1 Generación de alarma

Para la generación de la alarma se debe oprimir el botón de bloqueo la cantidad de veces que el usuario configuró. Para la selección de está alarma se realizó el siguiente código (ilustración 28).

Ilustración 28

Código fuente configuración alarma

```
private fun setAlarm(){
    val selectMusicSpinner = binding.alarmSpinner
    val mediaPlayer = MediaPlayer()
    //UpMaxVolume()
    //flashOn()

    val audioResources = mapOf(
        "alarma 1" to R.raw.alarm_one,
        "alarma 2" to R.raw.alarm_two,
        "alarma 3" to R.raw.alarm_three
    )

    val adapter = ArrayAdapter(
        requireContext(),
        android.R.layout.simple_spinner_item,
        objects: listOf("Predeterminado")+audioResources.map { it.key }
    )
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
    selectMusicSpinner.adapter = adapter
    selectMusicSpinner.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
        override fun onItemSelected(
            parent: AdapterView<*>,
            view: View?,
            position: Int,
            id: Long
        ) {
            if (position != 0) {
```

Fuente. elaboración propia

4.7.2 Configuración de la alerta

Para la configuración de la alerta en el registro inicial del usuario, se puede apreciar una pantalla en la que configura el sonido de la alarma y la cantidad de veces que el usuario desea oprimir el botón de bloqueo para activar la alerta (ilustración 29).

Ilustración 29

Configuración alerta



Fuente. elaboración propia

4.7.3 Mostrar instrucciones en pantalla

El usuario cuando inicia la aplicación puede observar un cuadro de texto en el que se pueden presentar las instrucciones para el manejo de la persona durante el ataque de epilepsia. Estas instrucciones se obtuvieron de un documento publicado por el Ministerio de Salud, donde se indica cómo debe actuar la persona que se encuentra cerca en el momento de ocurrir un ataque de epilepsia (Cataño, 2016).

Ilustración 30

Instrucciones en pantalla



Fuente. elaboración propia

4.7.4 Activación de ubicación

Cuando se abre la aplicación por primera vez, se le pide al usuario activar la ubicación del dispositivo, si este decide rechazar dicha activación, al finalizar el registro, nuevamente se le muestra la opción para activar la ubicación. Si nuevamente el usuario decide rechazarla, no podrá continuar con el proceso de configuración de la aplicación.

El desarrollo de esta última etapa de la aplicación se enfoca en las historias de usuario explicadas en las tablas 10 y 11.

Tabla 10.

Historia de usuario “Configurar mensaje de alerta”

HISTORIA DE USUARIO	
NOMBRE HISTORIA DE USUARIO: Configurar mensaje de alerta	
CÓDIGO: 005	FECHA: 2 de marzo
DESCRIPCIÓN: Después de generar el registro y configuración, el usuario podrá escribir el mensaje que quiere, sea enviado al contacto de emergencia	PRIORIDAD: Alta
	PROGRAMADORES ENCARGADOS: Santiago Rey
	ROL: Usuario con epilepsia
CRITERIOS DE ACEPTACIÓN: - El mensaje solo se guarda si se oprimió el botón de guardar - El mensaje no puede tener caracteres especiales	
MOCKUP: ilustración 12	

Fuente. elaboración propia

Para la configuración del mensaje que se desea enviar al contacto de emergencia, en el dashboard hay un editText (ilustración 31) que permite al usuario escribir el mensaje que desee y guardarlo para posteriormente enviarlo con el botón de bloqueo.

Ilustración 31.*Configurar mensaje de alerta*
*Fuente. elaboración propia***Tabla 11. Historia de usuario “Enviar ubicación en tiempo real”**

HISTORIA DE USUARIO	
NOMBRE HISTORIA DE USUARIO: Enviar ubicación en tiempo real	
CÓDIGO: 006	FECHA: 25 de abril
DESCRIPCIÓN: EpilepsyAlarm permite al usuario enviar su ubicación en tiempo real después de oprimir un número limitado de veces el botón de bloqueo del celular, que el usuario configuró inicialmente	PRIORIDAD: Alta
	PROGRAMADORES ENCARGADOS: Santiago Rey
	ROL: Usuario con epilepsia
CRITERIOS DE ACEPTACIÓN:	
<ul style="list-style-type: none"> - La ubicación a WhatsApp solo se enviará si se está conectado a internet, o por SMS si solamente hay conexión a la red móvil, sin datos. - La ubicación se enviará si no se ha desactivado la ubicación del dispositivo - La ubicación será enviada si el mensaje se ha configurado 	
MOCKUP: ilustración 19	

Fuente. elaboración propia

En el proceso de activación y envío de la ubicación en tiempo real, se utilizó la API de GoogleMaps que permite integrar el mapa a la aplicación y configurar la ubicación que se desea aparezca en el mapa. Para ello se realizó el siguiente código (ilustración 33), además de activar los permisos de ubicación exacta y parcial (ilustración 32).

Ilustración 32

Permisos de ubicación

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Fuente. elaboración propia

Ilustración 33

Código fuente de GoogleMaps

```
private fun getRealLocation(){
    if (ActivityCompat.checkSelfPermission(
        requireContext(),
        Manifest.permission.ACCESS_FINE_LOCATION
    ) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(
        requireContext(),
        Manifest.permission.ACCESS_COARSE_LOCATION
    ) != PackageManager.PERMISSION_GRANTED
    ){
        return
    }
}
```

Fuente. elaboración propia

Para mostrar las instrucciones en pantalla cuando se accione el botón de bloqueo del dispositivo, se realizó una investigación en la que se buscaba cómo mostrar en pantalla, incluso si

la aplicación no se encontraba abierta, las instrucciones para el manejo de una persona que está sufriendo de un ataque de epilepsia. Se agregó el siguiente código que permitió cumplir con la tarea (ilustración 34), la propiedad **showWhenLocked** permite mostrar una actividad, en este caso el mensaje de instrucciones, incluso cuando la pantalla esté bloqueada y la propiedad **theme** permite mostrar esta actividad en la pantalla completa del dispositivo.

Ilustración 34

Código fuente para mostrar mensaje en pantalla

```
android:showWhenLocked="true"
android:theme="@style/FullScreen" />
```

Fuente. elaboración propia

Tabla 12.

Historia de usuario “Mostrar instrucciones en pantalla para el manejo del paciente durante el ataque”

HISTORIA DE USUARIO	
NOMBRE HISTORIA DE USUARIO: Mostrar instrucciones en pantalla para manejo del paciente durante el ataque	
CÓDIGO: 008	FECHA: 2 de mayo
DESCRIPCIÓN: Cuando el usuario active la alarma, se mostrará en la pantalla del dispositivo un recuadro con un conjunto de instrucciones donde se indica cómo se maneja a la persona que	PRIORIDAD: Alta
	PROGRAMADORES ENCARGADOS: Santiago Rey
	ROL: Usuario con epilepsia

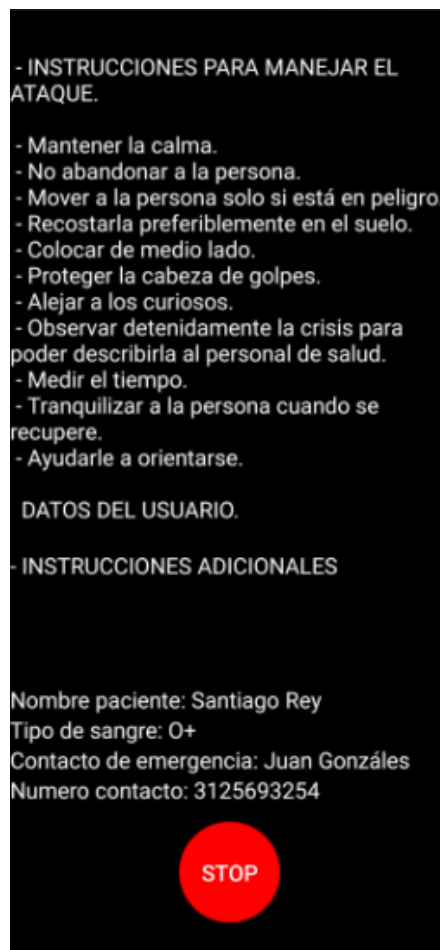
está padeciendo el ataque, durante las convulsiones, además de sus datos personales.	
CRITERIOS DE ACEPTACIÓN: - Las instrucciones se mostrarán en pantalla, después de que se active la alerta	
MOCKUP:	

Fuente. elaboración propia

Una vez se guarda el mensaje de alerta en la aplicación, ya se podrá accionar el botón de bloqueo del dispositivo para activar el proceso, adicionalmente se mostrará en la pantalla del dispositivo, un conjunto de instrucciones para el manejo del paciente durante el ataque y los datos de este para poder contactar a la persona de confianza que el usuario guardó en la configuración de la aplicación (ilustración 35).

Ilustración 35

Pantalla de bloqueo con instrucciones



Fuente. elaboración propia

5. Análisis y resultados

En este capítulo se evidencia el cumplimiento de cada objetivo planteado para este proyecto, así como la implementación de las distintas funcionalidades de la aplicación.

5.1 Selección de medio para enviar mensaje de alerta

El primer objetivo específico hace referencia a la búsqueda y selección de un medio de mensajería instantánea, que permita a la aplicación enviar un mensaje al oprimir el botón de bloqueo del dispositivo. Este objetivo se cumplió ya que se utilizó como medio la API de WhatsApp Business, siendo la mejor opción entre otras API de mensajería como Telegram o Messenger, porque WhatsApp es la aplicación que más se utiliza como medio de comunicación en todo el mundo por su facilidad de manejo. La API de WhatsApp permitió mediante un token que ofrece la página y un id, ingresarlos y poder configurar el número del contacto que recibirá el mensaje. En el fragmento de código en la Ilustración 15 se puede observar la respectiva configuración de la aplicación con la API para su funcionamiento.

5.2 Codificar funcionalidad de envío de alerta

El segundo objetivo hace referencia a la funcionalidad que se empleó para poder realizar la acción de enviar la alerta al contacto que previamente el usuario almacenó. Inicialmente se hizo una prueba con un botón hecho desde la aplicación para verificar si efectivamente se podía enviar un mensaje, seguido de implementar métodos para enviar el mensaje sin que la aplicación estuviera abierta, luego de eso se procedió a implementar dicho código para el botón de bloqueo del celular, utilizando permisos de Android.

La función que se observa en la (ilustración 36) se encarga de ejecutar varias tareas, una de ellas es la de enviar el mensaje y obtener la información del usuario. Los métodos sendMessage y sendSMS se usan para el envío del mensaje cuando empieza la función.

Ilustración 36

Fragmento de código para activación de función de envío de alerta

```

Santiago Rey *
private fun startFuncion(context: Context) {
    val result = goAsync()
    val coroutineScope = CoroutineScope(Dispatchers.Default)

    coroutineScope.launch { this: CoroutineScope
        val messageRepository = MessageRepository()
        messageRepository.sendMessage(context)
        messageRepository.sendSMS(context)
        val user = messageRepository.getUser()
        val activityIntent = Intent(context, BlockMessageActivity::class.java)
        activityIntent.putExtra(name: "user_emergency", user)
        activityIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK)
        // Iniciar la actividad
        context!!.startActivity(activityIntent)
        result.finish()
    }
    // showMessage(context)
    NotificationManager.showNotification(context)
    startAlarm(context)
}

```

Fuente. elaboración propia

5.3 Mostrar datos en pantalla

En este tercer objetivo se hace referencia a la presentación en pantalla luego de activar la alerta, de un conjunto de instrucciones para el manejo del paciente durante el ataque, además de

sus datos como nombre, tipo de sangre, nombre del contacto de confianza y número del contacto de confianza. Estos datos se ingresaron en el registro cuando se instaló la aplicación por primera vez, como se puede observar en la (ilustración 9).

5.4 Clase para conectar un dispositivo Bluetooth

En el cuarto objetivo específico se crea una clase no implementada en la que se codifican métodos para conectar un dispositivo externo, que permite mostrar una lista de dispositivos que se encuentran disponibles para conexión bluetooth, se puede observar un método **showPairedDeives()** que permite mostrar los dispositivos Bluetooth emparejados y permite seleccionar uno de ellos para conectarse, creando inicialmente una lista vacía ‘list’ para almacenar los dispositivos emparejados, luego verifica los dispositivos vinculados utilizando ‘pairedDevices’ como se observa en (ilustración 26).

5.5 Funcionalidades

5.5.1 Enviar mensaje por mensajería instantánea

Para cumplir con esta funcionalidad se implementó la API de WhatsApp que permite el envío de mensajes, para esto se utilizó el siguiente fragmento de código que envía el mensaje de ayuda que el usuario guardó junto con la ubicación en tiempo real (ilustración 11).

En este método “sendMessage”, ilustración 37, se obtiene el objeto que contiene el número de teléfono guardado por el usuario, el mensaje que desea enviar y la ubicación en tiempo real.

Ilustración 37

Fragmento de código que envía el mensaje y ubicación

```

Santiago Rey
suspend fun sendMessage(context: Context) {
    getRetrofit().create(MessageApi::class.java).postMessage(
        MessageEntity( messagingProduct: "whatsapp",
            recipientType: "individual",
            to: "57${getPhone()}",
            type: "text",
            Text( previewUrl: true, body: getMessage() + "\n mi ubicación es esta : " + getLocation(context))
        )
    )
}

```

Fuente. elaboración propia

Esto se puede mostrar en la ilustración 38 como evidencia de cumplimiento del objetivo específico.

Ilustración 37

Mensaje recibido con ubicación



Fuente. elaboración propia

5.5.2 Muestra de datos en pantalla

Para el cumplimiento de esta funcionalidad se investigó bastante la documentación en la red y la forma para que se pudieran mostrar los datos en pantalla cuando está se encuentre bloqueada, fue bastante complejo ya que no se encuentran muchas soluciones funcionales para esto, ya que la información suministrada por el usuario es muy relevante al momento de que una persona actúe en su ayuda, hay un adicional para mostrar instrucciones que el usuario desee que se conozcan sobre sí mismo al momento del socorro y se obtuvo el siguiente resultado (ilustración 35).

Como se observa en la (ilustración 35), es el resultado luego de oprimir el botón de bloqueo, se muestra en la pantalla bloqueada un conjunto de instrucciones para el manejo del paciente durante el ataque y los datos para que el personal de salud tenga conocimiento si se presenta.

5.5.3 Habilitación de botón de bloqueo para envío de alerta

Para esta funcionalidad se implementó el siguiente código (ilustración 39), creando un servicio para el botón de bloqueo que permite el envío de la alerta y mostrar los datos en la pantalla, para ellos el método “startFunction” permite obtener la información del usuario registrado en la base de datos local y les pasa esa información a los métodos “sendMessage” y “sendSMS”, que se encargan de enviar la ubicación y el mensaje a su destino. Cómo resultado se puede observar la información en (ilustración 35), cumpliendo con la funcionalidad de enviar información con el botón de bloqueo del dispositivo.

Ilustración 39

Fragmento de código que activa la función del botón de bloqueo

```

private fun startFunction(context: Context) {
    val result = goAsync()
    val coroutineScope = CoroutineScope(Dispatchers.Default)

    coroutineScope.launch { this: CoroutineScope
        val messageRepository = MessageRepository()
        messageRepository.sendMessage(context)
        messageRepository.sendSMS(context)
        val user = messageRepository.getUser()
        val activityIntent = Intent(context, BlockMessageActivity::class.java)
        activityIntent.putExtra( name: "user_emergency", user)
        activityIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK)
        // Iniciar la actividad
        context!!.startActivity(activityIntent)
        result.finish()
    }
    // showMessage(context)
    NotificationManager.showNotification(context)
}

```

Fuente. elaboración propia

5.2.4 Enviar mensaje de texto por SMS

Este requerimiento siendo muy similar al de enviar un mensaje al WhatsApp del contacto de emergencia, utiliza la mensajería convencional para también avisar a la persona de confianza que está padeciendo un ataque de epilepsia, para este requerimiento se empleó el siguiente código (ilustración 40).

Ilustración 40

Fragmento de código para enviar SMS

```
suspend fun sendSMS(context: Context) {
    val smsManager = SmsManager.getDefault()
    smsManager.sendTextMessage(getPhone(), scAddress: null, text: getMessage() + "\n Mi ubicación es esta: " + ge
    Toast.makeText(context, text: "SMS enviado", Toast.LENGTH_SHORT).show()
}
```

Este método permite conectarse con la librería de SMS y obtener los permisos para enviar el mensaje junto con la ubicación en tiempo real (ilustración 41), así dando cumplimiento al envío del mensaje por medio del sistema SMS.

Ilustración 41

Mensaje enviado solicitando ayuda y ubicación en tiempo real



Fuente. elaboración propia

6. Conclusiones

Con base en lo expuesto y principalmente en el capítulo 5, se puede concluir que se obtuvieron los resultados esperados al cumplir con los objetivos propuestos, dando como resultado una aplicación útil para aquellas personas que necesiten sentir un apoyo en el momento de padecer un ataque de epilepsia.

El desarrollo de EpilepsyAlarm surge de la necesidad de brindar a aquellas personas que padecen de epilepsia, una aplicación que les permita pedir ayuda y hacerse notar con las funcionalidades que tiene EpilepsyAlarm, tales como el botón de pánico que le enviará indicaciones claras a sus contactos, haciendo la diferencia en cuanto a botones de pánico se trata. Teniendo en cuenta todo lo anterior se concluye lo siguiente:

- La metodología Scrum como el punto central para el manejo de todo el proceso de desarrollo del proyecto fue de manera positiva un acierto, en cuanto a organización se trata.
- Android con el tiempo se ha vuelto más restrictivo con los permisos que se implementan en las aplicaciones ya que siempre se está pensando en la privacidad de los usuarios, por lo que el hecho de manejar permisos del mismo software del dispositivo implica una responsabilidad con el usuario bastante importante. Al desarrollar EpilepsyAlarm fue importante y fundamental tener presente cada permiso que se necesitaba para desarrollar una nueva funcionalidad, además los métodos empleados deben ser entendibles para la siguiente persona que decida hacer ajustes a la aplicación.
- Todas las funcionalidades recaen principalmente en el funcionamiento del botón del dispositivo móvil, por lo que esa implementación era la más importante, pero, así como

es de importante, es de compleja ya que como se menciona anteriormente, Android se ha vuelto más restrictivo en la libertad del manejo del software de los dispositivos.

- La implementación del API de WhatsApp fue de gran ayuda ya que es una aplicación que a diario utilizan las personas, siendo este el punto clave para estar conectados con el mundo exterior, junto con el envío de mensaje SMS, para que sea enviado de una forma u otra.
- La API de WhatsApp Business es difícil de comprender ya que su interacción con el usuario no es sencilla, adicionalmente para poder emplearla de manera adecuada es necesario tener una cuenta de Facebook de negocio y cumplir con unas normas de tener verificada dicha empresa.
- La API de GoogleMaps fue de gran utilidad para que el usuario pudiera enviar su ubicación en tiempo real, pero para ello fue necesario un token que permitió el uso de sus funciones dentro del código.
- Con el pasar de los años las posibilidades de seguir utilizando las mismas APIS de manera gratuita ha disminuido considerablemente, ya que las empresas han visto que es más rentable cobrar por su uso, por lo que resulta necesario contar con un rubro que se descuenta cuando se ingresa un método de pago. Tal es el caso de la API de Google Maps pues hace un par de años se podía utilizar de manera gratuita, pero al día de hoy es necesario contar con una tarjeta de crédito para poder hacer uso de ella.
- Se obtuvo una aplicación que puede funcionar para aquellas personas que sufren de ataques de epilepsia, pero tambien en un futuro se puede transformar en un botón de pánico de seguridad, para alertar cuando una persona sea víctima de un ataque o robo.

También queda pendiente para el grupo de investigación de la Facultad de Ingeniería de Sistemas, la validación y mejora de esta aplicación con personal médico y pacientes.

Esta estructura se podría utilizar como base de un botón de pánico en el tema de la seguridad ya que la función que cumpliría, sería la de alertar a las personas que se encuentran alrededor que algo está pasando, dando la oportunidad de actuar en el momento justo.

Recomendaciones para el futuro de la aplicación:

- En desarrollos futuros es importante llevar un registro más detallado de la información médica del paciente, para ello cuando se implemente un hardware como el botón físico la información almacenada ya no debería ir a una base de datos local, porque la información almacenada no va a persistir debido a que se enfoca solo en el espacio de la aplicación.
- Se recomienda al continuar utilizando las APIS de Google y WhatsApp utilizar métodos de pago, ya que pueden ofrecer una mejor interacción con el usuario al momento de enviar los mensajes o la ubicación actual.
- Es recomendable hacer una base de datos para almacenar la información de los ataques ocurridos y enviar dicha información a la nube para llevar un control al momento de asistir con un médico.
- Se recomienda implementación de código que permita recibir información de la frecuencia cardiaca del paciente, así cuando se implemente un hardware de botón pueda también enviarse la señal sin ser oprimido el botón. Para muchas aplicaciones de ejercicio se utilizan los dispositivos SmartWatch que al ser utilizados en la muñeca

pueden medir la frecuencia cardiaca. Esto se podría implementar para aumentar la información enviada a la aplicación.

Referencias bibliográficas

4 aplicaciones que ayudan a controlar la epilepsia. (2021, abril 1). Asociación Coruñesa de Epilepsia. <https://www.asociacionepilepsiacoruna.org/4-aplicaciones-que-ayudan-a-controlar-la-epilepsia/>

Álvarez, R. (2015, abril 26). *SeizAlarm, la aplicación que promete ayudar en caso de algún trastorno convulsivo.* Applesfera.com; Applesfera. <https://www.applesfera.com/aplicaciones-ios-1/seizalarm-la-aplicacion-que-promete-ayudar-en-caso-de-algun-trastorno-convulsivo>

Amazon.com: Daytech Sistema de alarma de pánico con botón de llamada de emergencia para cuidadores, alarma de seguridad para ancianos, pacientes mayores, 2 receptores+2 botones de pánico: Productos de Oficina. (s/f). Recuperado el 28 de octubre de 2022, a partir de <https://www.amazon.com/-/es/emergencia-cuidadores-seguridad-pacientes-receptores/dp/B084GKJZT8?th=1>

Arboleda, A. (2020, marzo 13). *Conoce las funciones de un Product Owner y su importancia en los proyectos ágiles.* Rock Content - ES; Rock Content. <https://rockcontent.com/es/blog/product-owner/>

Botón de pánico. (2017, enero 31). Miray Consulting. <https://www.mirayconsulting.com/boton-panico-centros-asistenciales/>

Convulsiones. (s/f). Recuperado el 2 de noviembre de 2022, a partir de

<https://medlineplus.gov/spanish/seizures.html>

Definición de padecimiento - Qué es, Significado y Concepto. (s/f). Recuperado el 2 de

noviembre de 2022, a partir de <https://definicion.de/padecimiento/>

Epilepsia - Síntomas y causas - Mayo Clinic. (s/f). Recuperado el 2 de noviembre de 2022, a

partir de <https://www.mayoclinic.org/es-es/diseases-conditions/epilepsy/symptoms-causes/syc-20350093>

Hurtado, J. S. (2021). Cómo funciona la Metodología Scrum: Qué es y cómo utilizarla.

Thinking for Innovation. <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>

Ley 1581 de 2012 - Gestor Normativo - Función Pública. (s/f). Recuperado el 28 de octubre de

2022, a partir de

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>

¿Qué es el botón del pánico y cómo me puede ayudar? – Cajas fuertes - Cajas Fuertes y

Armeros | Tienda Online De Cajas Fuertes. (s/f). Recuperado el 2 de noviembre de 2022, a partir de <https://www.tiendacajasfuertes.com/EntradaBlog-Que-es-el-boton-del-panico-y-como-me-puede-ayudar-105.html>

Rehkopf, D. M. (s/f). *Todo lo que necesitas saber sobre los sprints de scrum.* Atlassian. Recuperado el

12 de noviembre de 2022, de <https://www.atlassian.com/es/agile/scrum/sprints>