

TRABAJO DE GRADO
Opción Proyecto de Grado

Aplicación móvil AlzAlert

Corporación Universitaria Remington.

Facultad de ingenierías

Tecnología en desarrollo de software

Santiago Delgado Arango

Juan David Higueta Giraldo

Docente: Gloria Amparo Lora

Opción Proyecto de grado.

2025

Tabla de contenido	Pág.
Resumen.....	11
Palabras clave.....	12
Introducción	13
Marco teórico	15
Alzheimer	15
Tipos de Alzheimer	15
Etapas del alzhéimer.....	17
Dificultades para pacientes con Alzheimer y sus cuidadores	18
Tecnología en la salud.....	18
Aplicaciones móviles destinadas a pacientes con algún tipo de demencia	19
Comparación de aplicativos contra AlzAlert	20
Tecnologías usadas dentro del proyecto.....	22
Metodologías de desarrollo ágil	25
Requisitos de usuario	26
Requisitos funcionales.....	27
Requisitos no funcionales.....	27
Notificaciones en aplicaciones móviles	28
Diagrama de casos de uso	29
Planteamiento del Problema	30
Objetivos.....	32
Objetivo general	32

Objetivos específicos.....	32
Alcance	33
Actividades para resolver el objetivo específico número 1	34
Encuesta	35
Resultados de la encuesta.....	39
Identificación de usuarios claves.....	45
Requisitos de usuario	46
Requisitos funcionales (RF).....	46
Requisitos no funcionales (RNF).....	48
Escenarios de los casos de uso.....	48
Diagramas modelo de casos de uso.....	61
Diagrama de clases.....	64
Modelo entidad relación.....	65
Costos del Proyecto.....	69
Actividades para resolver el objetivo específico número 2	75
Diccionario de datos	75
Prototipo paso a paso	82
Conclusiones del prototipo.....	91
Actividades para resolver el objetivo específico número 3	92
Verificación de estado.....	92
Envío de SMS.....	94

Historial de ubicación.....	95
Conexión bluetooth	97
Recepción de la información.....	98
Código de Arduino IDE	99
Retos en la Fase de Diseño y Desarrollo	100
Actividades para resolver el objetivo específico número 4	102
Conclusión ejecución de pruebas	115
Conclusión general del desarrollo del proyecto	116
Referencias.....	118

Listado de tablas	Pág.
Tabla 1: Tipos del Alzheimer	16
Tabla 2: Etapas del Alzheimer	17
Tabla 3: Comparación de aplicativos contra AlzAlert.....	22
Tabla 4: Actores del sistema	46
Tabla 5: Requisitos de usuario	46
Tabla 6: Requisitos no funcionales	48
Tabla 7: Escenario caso de usos (Crear usuario)	50
Tabla 8: Escenario caso de uso (Modificar usuario).....	52
Tabla 9: Escenario caso de uso (Consultar usuario)	54
Tabla 10: Escenario caso de uso (Inhabilitar usuario).....	55
Tabla 11: Escenario caso de uso (Consulta frecuencia preguntas de verificación)	57
Tabla 12: Escenario caso de uso (Modificación frecuencia preguntas de verificación).....	58
Tabla 13: Escenario caso de uso (Inhabilitar frecuencia preguntas de verificación).....	59
Tabla 14: Escenario Caso de uso (Consulta historial de ubicación).....	61
Tabla 15: Herramientas necesarias para el desarrollo.....	69
Tabla 16: Costos del proyecto (Hardware)	71
Tabla 17: Costos del proyecto (Software)	72
Tabla 18: Costos del proyecto (Talento humano).....	73
Tabla 19: Costos estimados del proyecto.....	73
Tabla 20: Historias de usuario	74
Tabla 21: Diccionario de datos (tabla Perfil).....	75
Tabla 22: Diccionario de datos (tabla Usuario)	76
Tabla 23: Diccionario de datos (tabla Paciente)	77
Tabla 24: Diccionario de datos (tabla Cuidador).....	77

Tabla 25: Diccionario de datos (tabla contacto emergencia).....	78
Tabla 26: Diccionario de datos (tabla Configuración pregunta verificación).....	79
Tabla 27: Diccionario de datos (tabla Alerta).....	80
Tabla 28: Diccionario de datos (tabla Ubicación)	81
Tabla 29: Diccionario de datos (tabla Alerta contacto)	82
Tabla 30: Pruebas de usabilidad y funcionalidad	114

Listado de imágenes	Pág.
Ilustración 1: Preguntas encuesta 1 a 2	35
Ilustración 2: Preguntas encuesta 3 a 5	36
Ilustración 3: Preguntas encuesta 6 a 7	36
Ilustración 4: Preguntas encuesta 8 a 10	37
Ilustración 5: Preguntas encuesta 11 a 12	37
Ilustración 6: Preguntas encuesta 13 a 15	38
Ilustración 7: Preguntas encuesta 16 a 17	38
Ilustración 8: Preguntas encuesta 18 a 20	39
Ilustración 9: Resultado encuesta pregunta 1	40
Ilustración 10: Resultado encuesta pregunta 2	40
Ilustración 11: Resultado encuesta pregunta 3	41
Ilustración 12: Resultado encuesta pregunta 4	42
Ilustración 13: Resultado encuesta pregunta 8	43
Ilustración 14: Resultado encuesta pregunta 20	45
Ilustración 15: Diagrama general de casos de uso	61
Ilustración 16: Caso de uso extendido (Gestión de usuarios)	62
Ilustración 17: Caso de uso extendido (Gestión de Preguntas de Verificación)	62
Ilustración 18: Caso de uso extendido (Gestión de consulta)	63
Ilustración 19: Diagrama de clases	64
Ilustración 20: Modelo entidad relación	65
Ilustración 21: Prototipo (Bienvenida)	82
Ilustración 22: Prototipo (seleccionar perfil)	82
Ilustración 23: Prototipo (Pantalla inicial)	83
Ilustración 24: Prototipo (Crear cuenta validaciones)	83

Ilustración 25: Prototipo (Crear cuenta con datos)	83
Ilustración 26: Prototipo (Información personal validaciones)	83
Ilustración 27: Prototipo (Información personal con datos)	84
Ilustración 28: Prototipo (Agregar contacto validaciones)	84
Ilustración 29: Prototipo (Agregar contacto con datos).....	84
Ilustración 30: Prototipo (Contacto agregado).....	84
Ilustración 31: Prototipo (Iniciar sesión)	85
Ilustración 32: Prototipo (Iniciar sesión validaciones)	85
Ilustración 33: Prototipo (iniciar sesión con datos)	85
Ilustración 34: Prototipo (Inicio)	85
Ilustración 35: Prototipo (Consultar perfil).....	86
Ilustración 36: Prototipo (Editar info. Personal).....	86
Ilustración 37: Prototipo (Confirmación eliminar perfil).....	86
Ilustración 38: Prototipo (Perfil eliminado).....	86
Ilustración 39: Prototipo (Consultar contacto emergencia)	87
Ilustración 40: Prototipo (Editar contacto emergencia).....	87
Ilustración 41: Prototipo (Eliminar contacto emergencia).....	87
Ilustración 42: Prototipo (Contacto emergencia eliminado).....	87
Ilustración 43: Prototipo (Inicio config. alerta pág. 1).....	88
Ilustración 44: Prototipo (Inicio config. alerta Pág. 2)	88
Ilustración 45: Prototipo (Config. alerta intervalo 1).....	88
Ilustración 46: Prototipo (Config. alerta intervalo 2).....	88
Ilustración 47: Prototipo (Config. alerta guardar).....	89
Ilustración 48: Prototipo (Config. alerta inhabilitar)	89
Ilustración 49: Prototipo (Primera alerta verificación)	89

Ilustración 50: Prototipo (Segunda alerta verificación).....	89
Ilustración 51: Prototipo (Botón confirmación envió de alerta)	90
Ilustración 52: Prototipo (Alerta enviada)	90
Ilustración 53: prototipo (Historia de ubicaciones)	90
Ilustración 54: Modelo de actividades	91
Ilustración 55: Desarrollo (Preguntas de verificación)	93
Ilustración 56: Desarrollo (Envío de SMS)	94
Ilustración 57: Desarrollo (Lista de contactos de emergencia).....	95
Ilustración 58: Desarrollo (Historial de ubicaciones)	96
Ilustración 59: Desarrollo (Conexión a bluetooth)	97
Ilustración 60: Desarrollo (Captura de datos desde el bluetooth).....	98
Ilustración 61: Desarrollo (Comunicación del Arduino con el ESP32).....	99
Ilustración 62:Pruebas (Validación crear usuario).....	104
Ilustración 63: Pruebas (Crear usuario)	104
Ilustración 64: Pruebas (Validación campos vacíos 1)	104
Ilustración 65: Pruebas (Validación campos vacíos 2)	104
Ilustración 66: Pruebas (Validación agregar contacto).....	105
Ilustración 67: Pruebas (Agregar contacto)	105
Ilustración 68: Pruebas (Contacto guardado).....	106
Ilustración 69: Pruebas (Contacto modificado)	106
Ilustración 70: Pruebas (Nombre de usuario guardado)	107
Ilustración 71: Pruebas (Nombre de usuario modificado)	107
Ilustración 72: Pruebas (Info. Medica guardada).....	107
Ilustración 73: Pruebas (Info. Medica modificada)	107
Ilustración 74: Pruebas (Config. Alertas primer intervalo)	109

Ilustración 75: Pruebas (Config. Alertas segundo intervalo).....	109
Ilustración 76: Pruebas (Primera alerta)	110
Ilustración 77: Pruebas (Segunda alerta)	110
Ilustración 78: Pruebas (Alertas inactivas)	111
Ilustración 79: Pruebas (Alertas activadas).....	111
Ilustración 80: Pruebas (Confirmación de alertas activas)	112
Ilustración 81: Pruebas (Validación conexión bluetooth).....	113
Ilustración 82: Pruebas (Conexión exitosa)	113
Ilustración 83: Pruebas (Recepción de alerta exitosa)	114
Ilustración 84: Pruebas (Ejecución consola Arduino)	114

Resumen

AlzAlert es una aplicación móvil diseñada para monitorear el estado de pacientes con Alzheimer en etapas avanzadas, con el fin de prevenir situaciones de riesgo y facilitar una respuesta oportuna por parte de sus cuidadores, este se basa en la emisión periódica preguntas al paciente si se encuentra bien, en caso de respuesta negativa o ausencia de respuesta dentro de un intervalo predefinido se activa una alerta de emergencia.

Esta alerta incluye la ubicación geográfica exacta del paciente, obtenida mediante un módulo GPS externo (NEO-M8N) conectado vía bluetooth a un ESP32, el cual transmite las coordenadas al dispositivo móvil. La aplicación procesa esta información y envía un SMS con un enlace interactivo a Google Maps a los contactos de emergencia registrados. El sistema también permite configurar la frecuencia de las verificaciones, desactivarlas temporalmente y consultar un historial detallado de las ubicaciones registradas.

El desarrollo se llevó a cabo bajo la metodología ágil Scrum, utilizando Flutter y Dart para la interfaz móvil, y Arduino IDE para la programación del ESP32. La interfaz de usuario fue diseñada con criterios de accesibilidad: botones de gran tamaño, texto legible y flujos simplificados, pensados para usuarios con limitaciones cognitivas.

Las pruebas realizadas validaron tanto la funcionalidad del software como la integración con el hardware. Se comprobó la precisión en la obtención de coordenadas, la confiabilidad en el envío de alertas y la usabilidad general de la aplicación. Los resultados demuestran que AlzAlert cumple con su objetivo principal, convertirse en una herramienta práctica, eficiente y centrada en el usuario, capaz de mitigar riesgos reales y aportar un valor en el cuidado diario de personas con Alzheimer.

Palabras clave

Arduino: Es una plataforma de hardware y software de código abierto, diseñada para facilitar la creación de proyectos electrónicos interactivos, esto es posible ya que a partir de instrucciones dadas a un chip para que las ejecute permite crear múltiples proyectos tecnológicos.

Cíclico: Se refiere a algo que se repite con cierta regularidad o frecuencia, tiene un patrón recurrente que se vuelve a un punto de partida o a un estado anterior después de un cierto período o de completar una serie de eventos.

Latencia: se refiere al tiempo que transcurre entre que el usuario realiza una acción (como hacer clic, enviar una solicitud o abrir una pantalla) y la aplicación responde mostrando el resultado esperado.

Introducción

El alzhéimer representa uno de los mayores desafíos de salud pública del siglo XXI, ya que trata de una enfermedad neurodegenerativa progresiva que afecta las funciones cognitivas, la memoria y la capacidad de autonomía del paciente, generando una carga significativa no solo para quienes la padecen, sino también para sus cuidadores y el sistema de salud en general. En Colombia, su impacto se agrava por la presencia de la mutación genética PSEN1, conocida como “mutación paisa”, que acelera el inicio de la enfermedad y eleva las tasas de prevalencia en comparación con otros países.

Debido a lo anterior soluciones convencionales de cuidado resultan insuficientes y los sistemas de alerta tradicionales no están diseñados para responder a las particularidades cognitivas de los pacientes en etapas avanzadas. Es en este contexto que surge AlzAlert, una aplicación móvil desarrollada con el propósito de ofrecer un mecanismo de monitoreo y respuesta ante situaciones de riesgo.

El sistema funciona mediante notificaciones cíclicas que verifican el estado del paciente, si este no responde o indica encontrarse mal, se activa automáticamente un protocolo de emergencia, donde se obtiene su ubicación geográfica en tiempo real a través de un módulo GPS externo (ESP32 + NEO-M8N) y se envía un mensaje de texto con un enlace a Google Maps a los contactos de emergencia predefinidos. Además, la aplicación permite desactivar temporalmente el sistema cuando el paciente está bajo supervisión directa, ofreciendo flexibilidad y control al cuidador.

En este trabajo se documenta el proceso completo de concepción, diseño, desarrollo y validación de AlzAlert. Se fundamenta en una metodología ágil (Scrum), en el uso de tecnologías móviles modernas (Flutter, Dart, Firebase) y en la integración con hardware.

Marco teórico

Alzheimer

De acuerdo con Toro (2022) el alzhéimer es una forma agresiva de demencia que se manifiesta en déficits de memoria, lenguaje y comportamiento. Según la Organización Mundial de la Salud se pronostica que en las próximas décadas hasta el 2050 podría cuadruplicarse las cifras de personas con esta patología alcanzando 114 millones de personas. Adicional al deterioro en la calidad de vida del paciente, los principales cuidadores de esta población son los familiares, en quienes recae una mayor contribución monetaria y de tiempo, generando estrés emocional y depresión a los cuidadores.

En Colombia según un estudio de la Universidad de los Andes (2024) a pesar de no tener cifras exactas se estima que el rango de prevalencia de esta patología varía entre el 1,3% y 23,6% siendo este uno de los países donde más se podría albergar esta patología, ya que en la región y particularmente en este país de forma aún no comprendida totalmente por la ciencia existe una mutación genética llamada (PSEN1) asociada al Alzheimer de inicio temprano, la cual tiene tendencia a expandirse en cada generación venidera.

Tipos de Alzheimer

Según la Clínica Mayo (2025) el Alzheimer tiene tipos y etapas definidos que reflejan el avance del deterioro cognitivo y funcional y generalmente se clasifican en tres etapas.

Tipo de Alzheimer	Descripción	Características
Enfermedad de Alzheimer de inicio tardío	Es el tipo más común. Se presenta generalmente después de los 65 años. Su	Inicio progresivo y lento. Pérdida de memoria reciente.

	causa es multifactorial y suele involucrar factores genéticos y ambientales.	Afecta funciones ejecutivas y lenguaje.
Enfermedad de Alzheimer de inicio temprano	Representa menos del 10% de los casos. Aparece entre los 30 y 60 años. Se asocia comúnmente a mutaciones genéticas hereditarias.	Avance más rápido que el de inicio tardío. Mayor riesgo genético. Puede confundirse con otras condiciones.
Alzheimer Familiar (Genético)	Forma hereditaria causada por mutaciones específicas en genes como PSEN1, PSEN2 o APP. Se transmite de forma autosómica dominante.	Altamente hereditario. Edad de aparición temprana. Se ha documentado en poblaciones como Yarumal (Colombia).
Alzheimer atípico (variantes clínicas)	Variantes poco comunes que afectan áreas específicas del cerebro primero, como la visión o el lenguaje.	Por ejemplo: Afasia progresiva primaria, variante logopénica. Atrofia cortical posterior. Presentación no típica.
Alzheimer mixto	Ocurre cuando la enfermedad de Alzheimer coexiste con otros tipos de demencia, como demencia vascular o cuerpos de Lewy.	Síntomas variados. Mayor deterioro funcional. Diagnóstico complejo.

Tabla 1: Tipos del Alzheimer

Etapas del alzhéimer

Etapas	Descripción	Principales síntomas
Etapa leve (temprana)	El paciente empieza a mostrar olvidos leves, principalmente relacionados con hechos recientes. Puede continuar con su vida relativamente independiente.	Olvidos frecuentes (nombres, citas). Dificultades para encontrar palabras. Cambios de humor.
Etapa moderada	El deterioro es más evidente. Requiere ayuda para algunas actividades cotidianas. La memoria y el juicio se ven más afectados.	Confusión en lugares familiares. Dificultad para realizar tareas básicas. Cambios de conducta.
Etapa severa (avanzada)	Pérdida casi total de autonomía. El paciente necesita supervisión constante. Se deteriora el lenguaje, la movilidad y el reconocimiento de personas.	No reconoce familiares. Dificultad para tragar o caminar. Pérdida del control de esfínteres.

Tabla 2: Etapas del Alzheimer

Dificultades para pacientes con Alzheimer y sus cuidadores

De acuerdo con Naranjo (2021) los pacientes con alzhéimer presentan dificultades de comunicación progresivas, como confundir conceptos, perder vocabulario, no comprender lecturas o algo que escuchan, además de no perder la capacidad de relatar eventos, y a medida que avanza se incrementan las dificultades.

Las anteriores dificultades vienen acompañadas de cambios de conductas como irritabilidad, hostilidad o agitación lo que hace que se complique la relación con los cuidadores, para quienes estas situaciones generan sentimientos de inseguridad, impotencia, estrés, y desesperanza, quienes en caso de no cuidarse también podría verse afectado su bienestar.

Los cuidados en enfermería para este tipo de pacientes son desafiantes, ya que integran aspectos clínicos, epidemiológicos y humanistas, por ello es fundamental el apoyo de la familia y cuidadores principales, especialmente a lo relacionado a la atención primaria. Esto implica atender sus necesidades humanas afectadas por el estrés diario, promover la salud, prevenir enfermedades, y fomentar el autocuidado físico y mental. Un cuidado bien orientado hacia el cuidador facilita su adaptación y alivia la carga emocional y las demandas que experimenta.

Tecnología en la salud

Según García (2024) las herramientas tecnológicas son fundamentales para la vida diaria de cada persona, incluido en el ámbito de la salud, donde la telemedicina, inteligencia artificial, registros médicos electrónicos, aplicaciones, entre muchos más resultan ser de gran importancia para el diagnóstico, tratamiento y seguimiento de los pacientes, mejorando la eficiencia, calidad y oportunidad de acceso a los servicios de salud.

El internet de las cosas ha facilitado el monitoreo remoto de pacientes y la gestión de enfermedades crónicas. Los dispositivos médicos conectados, como glucómetros, tensiómetros y dispositivos de electrocardiograma portátiles permiten la recopilación continua de datos de salud fuera del entorno clínico tradicional.

Aplicaciones móviles destinadas a pacientes con algún tipo de demencia

Constante therapy: Diseñada por médicos y científicos de la Universidad de Boston para ayudar a personas con afasia, demencia y otros trastornos del habla y neurológicos. Las personas con demencia pueden practicar el habla, la memoria, la lectura, las matemáticas, la atención, la comprensión y otras habilidades cotidianas mediante ejercicios basados en la evidencia. Esta aplicación ofrece un programa personalizado, adaptado a sus objetivos, que puede añadir a la aplicación.

Esta aplicación también cuenta con una versión clínica, que permite a los profesionales médicos asignar tareas terapéuticas virtualmente mientras monitorean el progreso de sus pacientes. A través de la aplicación, los profesionales envían programas personalizados de terapia cognitiva directamente al teléfono o tableta del paciente, en lugar de entregar hojas de trabajo que se pueden extraviar fácilmente.

Nymbi: ofrece herramientas de entrenamiento del equilibrio, evaluaciones del riesgo de caídas, evaluaciones digitales del equilibrio y más para prevenir caídas y mejorar la independencia. Esta aplicación se desarrolló con base en 35 años de investigación clínica y solo se ofrece a través de aseguradoras médicas.

Alzheimer's Daily Companion: esta aplicación para cuidadores de personas con Alzheimer incluye más de 500 consejos y soluciones prácticas, directamente de especialistas

en memoria. Los cuidadores pueden usar esta app para acceder a materiales de capacitación e información sobre cómo afrontar conductas y situaciones difíciles relacionadas con la demencia. Incluso pueden compartir sus propios consejos en la aplicación. Además, ofrece asistencia telefónica o por correo electrónico las 24 horas para cualquier información que no esté incluida en la aplicación.

AmuseIT: es una aplicación intuitiva que estimula el cerebro y fomenta la conversación entre pacientes con demencia, sus cuidadores y sus seres queridos. Está diseñada para ayudar a los pacientes con demencia a interactuar mediante más de mil preguntas sencillas y componentes visuales. Las preguntas e imágenes sirven como estrategias de comunicación eficaces para conectar con recuerdos del pasado y generar conversaciones.

Memory Lane Games: ofrece una serie de juegos de memoria diseñados para ayudar a evocar recuerdos felices, fomentar la reminiscencia y promover la relajación. La misión de la empresa es mejorar la calidad de vida de las personas con demencia y sus cuidadores mediante una aplicación digital accesible. Los cuidadores sintieron que las actividades de la aplicación ayudaron a sus seres queridos con demencia a sentirse más relajados, según un ensayo clínico realizado por la empresa en 2022.

Comparación de aplicativos contra AlzAlert

Nombre de la aplicación	Ventajas	Desventajas	Diferenciador de AlzAlert
--------------------------------	-----------------	--------------------	----------------------------------

<p>Constant Therapy</p>	<p>Personalización de terapias cognitivas.</p> <p>Uso clínico y en casa.</p> <p>Monitoreo por parte de profesionales.</p>	<p>Enfocada en estimulación cognitiva más que en emergencias.</p> <p>Puede requerir supervisión.</p>	<p>AlzAlert se enfoca en prevención de riesgo con alertas y ubicación, no en terapias cognoscitivas.</p>
<p>Nymbi</p>	<p>Entrenamiento del equilibrio.</p> <p>Evaluaciones clínicas digitales.</p> <p>Prevención de caídas.</p>	<p>Disponible sólo a través de aseguradoras.</p> <p>Limitado acceso para usuarios generales.</p>	<p>AlzAlert tiene acceso directo desde cualquier dispositivo Android y enfoca su prevención en desorientación y ubicación, no equilibrio.</p>
<p>Alzheimer's Daily Companion</p>	<p>Consejos útiles para cuidadores.</p> <p>Acceso 24/7 a soporte.</p> <p>Basado en expertos en memoria.</p>	<p>No interactúa directamente con el paciente.</p> <p>No incluye geolocalización.</p>	<p>AlzAlert se comunica directamente con el paciente y genera respuestas automáticas ante inactividad.</p>
<p>AmuseIT</p>	<p>Estimula la conversación.</p> <p>Fomenta recuerdos y vínculos familiares.</p> <p>Interfaz sencilla.</p>	<p>Requiere participación del paciente.</p> <p>No tiene sistema de alertas.</p>	<p>AlzAlert actúa sin participación, útil en Alzheimer avanzado con interacción limitada.</p>
<p>Memory Lane Games</p>	<p>Juegos con fines terapéuticos.</p> <p>Promueve la relajación y el recuerdo.</p> <p>Validado clínicamente.</p>	<p>No está diseñada para seguimiento o emergencias.</p> <p>Se basa en momentos recreativos.</p>	<p>AlzAlert actúa en tiempo real para gestionar emergencias, sin depender momentos activos del paciente.</p>

Tabla 3: Comparación de aplicativos contra AlzAlert

Tecnologías usadas dentro del proyecto

Bluetooth: Según Rentería (2024) es una tecnología inalámbrica de corto alcance que permite la comunicación y transferencia de datos sin necesidad de cable, este utiliza ondas de radio para establecer conexiones entre distintos dispositivos, como teléfonos, audífonos, altavoces y otros, facilitando el intercambio de información.

Dart: Lenguaje de programación de código abierto, desarrollado por Google diseñado para construir aplicaciones rápidas y eficientes tanto en web, como para dispositivos móviles y de escritorio, conocido por su versatilidad, productividad y su papel clave en el framework de desarrollo móvil Flutter.

Flutter: Es un framework de código abierto creado por Google para desarrollar interfaces de usuario de aplicaciones multiplataforma con una sola base de código, este permite a los desarrolladores crear aplicaciones que se ejecutan de forma nativa en diferentes plataformas, como iOS, Android, web, Windows, macOS y Linux, utilizando un único lenguaje de programación, Dart.

Firebase: Plataforma en la nube de Google diseñada para facilitar el desarrollo de aplicaciones móviles y web, la cual ofrece una variedad de herramientas y servicios que permiten a los desarrolladores crear, mejorar y escalar sus aplicaciones, desde la gestión de bases de datos y autenticación hasta el análisis de datos y la monetización.

Figma: Herramienta de diseño vectorial basada en la nube que permite a equipos de diseño colaborar en tiempo real en proyectos de interfaces de usuario, prototipos y diseño web, entre otros, este es popular por su accesibilidad desde cualquier navegador y su enfoque en la colaboración, lo que facilita el trabajo conjunto de diseñadores, desarrolladores y otros miembros del equipo.

GitHub: Es una plataforma basada en la nube, ampliamente utilizada para el desarrollo de software, que proporciona herramientas para el control de versiones y la colaboración, este principalmente actúa como un repositorio centralizado donde los desarrolladores pueden almacenar su código, realizar un seguimiento de los cambios, colaborar en proyectos y compartir su trabajo con otros.

Visual Studio Code (VS Code): Es un editor de código fuente potente, desarrollado por Microsoft, que se ejecuta en múltiples plataformas como Windows, macOS y Linux, es conocido por su versatilidad, personalización a través de extensiones y soporte para una amplia gama de lenguajes de programación.

Android Studio: Es el entorno de desarrollo integrado oficial para crear aplicaciones Android, el cual es una herramienta poderosa y versátil que facilita la creación, depuración y prueba de aplicaciones para dispositivos Android.

Arduino IDE: Es un software que permite a los usuarios escribir, compilar y cargar código en las placas Arduino, este es esencial para la programación de estos microcontroladores y funciona como un puente entre el código que escribes y la placa física.

ESP32: Es un microcontrolador de bajo costo y consumo de energía, el cual es utilizado ampliamente para proyectos de internet de las cosas debido a su versatilidad y capacidades inalámbricas. Tiene las siguientes características:

- Doble núcleo de CPU que permite ejecutar múltiples tareas simultáneamente.
- Conectividad a WI-FI y Bluetooth, facilitando la conexión a múltiples dispositivos.
- Bajo consumo de energía, ideal para dispositivos que se alimentan por medio de baterías, como celulares y tabletas.
- Fácil de programar, compatible con plataformas como Arduino IDE, que facilita el desarrollo de aplicaciones.
- Amplias opciones de conexión a periféricos, gracias a sus componentes puede conectarse a sensores y actuadores.

GPS: Nombrado así por las siglas en inglés (Global Positioning System o Sistema de Posicionamiento Global), es un sistema de navegación por satélite que permite a un receptor de GPS determinar la posición en la que se encuentra en cualquier lugar de la tierra las 24 horas del día. Esta tecnología fue desarrollada por el Departamento de Defensa de Estados Unidos y ahora es de uso civil en todo el mundo de forma gratuita.

Módulo GPS NEO-M8N: Es un receptor de sistema de navegación global de alta precisión fabricado por u-blox, una empresa suiza líder en módulos inalámbricos y chips de posicionamiento. Es ampliamente utilizado en proyectos de electrónica y robótica debido a su excelente rendimiento y capacidad para rastrear múltiples constelaciones de satélites, permitiendo obtener la ubicación por medio de la utilización de GPS. Tiene características como:

- Bajo consumo de energía, lo que lo hace ideal para ser usado con dispositivos alimentados por batería.
- Alta sensibilidad, lo cual le permite obtener una posición bastante precisa.

Metodologías de desarrollo ágil

Según Narváez (2024) las metodologías de desarrollo ágil representan un paradigma de desarrollo y gestión de proyectos que prioriza la flexibilidad y la entrega de valor continua sobre la rigidez de la planificación tradicional., a diferencia de los enfoques lineales, donde las fases se completan secuencialmente, las metodologías ágiles operan mediante ciclos iterativos e incrementales, permitiendo una adaptación constante a los requisitos cambiantes del cliente y del mercado.

Básicamente el desarrollo por medio de una metodología ágil se lleva a cabo mediante la descomposición de un proyecto en segmentos cortos y gestionables llamados iteraciones, y cada uno de estos ciclos culmina en la entrega de un incremento de producto funcional y potencialmente desplegable. Las metodologías más usadas son:

Scrum: Marco de trabajo ágil para conseguir una cooperación efectiva en el equipo en diferentes proyectos, además utiliza reglas, artefactos y permite definir roles que crean la estructura indispensable para obtener la funcionalidad correcta. Scrum adopta un enfoque incremental basado en la teoría de gestión de procesos. Dicha teoría se basa en la transparencia, verificación y adaptación; la transparencia asegura la claridad durante el proceso de cosas que pueden influir en el resultado; la verificación permite descubrir desviaciones no deseadas del proceso; y la adaptación, efectúa arreglos apropiados para minimizar su impacto.

KANBAN: Método visual para la gestión de proyectos que se ejecuta mediante la utilización de tablero Kanban. Estos tableros permiten a los equipos observar de forma gráfica su flujo y carga de trabajo, además se encuentran organizados en columnas que indican distintas fases que conlleva el proceso laboral, tales como: “pendiente”, “en progreso” y “completado”. Las tareas específicas se representan mediante tarjetas que progresan a lo largo de las columnas hasta su finalización. Este enfoque se fundamenta en la transparencia y representación visual del proceso de trabajo, permitiendo a los equipos detectar cuellos de botella y mejorar continuamente el proceso

Extreme Programming (XP): Pone énfasis en la relevancia de perfeccionar las interacciones entre personas como un elemento fundamental para lograr el éxito a la hora de desarrollar software. XP promueve la colaboración, enseñanza y un entorno laboral positivo. Fundamenta su enfoque en la retroalimentación entre desarrolladores y cliente, la comunicación flexible entre los involucrados, la implementación de soluciones de manera sencilla y la disposición para realizar cambios. Es particularmente apropiado para proyectos con requerimientos indefinidos, altamente variables y con un riesgo técnico considerable.

Diagrama de clases

según Miro (s.f) los diagramas de clase son una de las diversas representaciones que forman parte de los diagramas de estructura UML, los cuales se enfocan en mostrar la estructura estática de un sistema, a diferencia de otros diagramas que presentan cómo cambian los objetos con el tiempo. Los diagramas de clase, específicamente, se encargan de visualizar las clases de un sistema y sus relaciones.

Requisitos de usuario

Según la Facultad de Ciencias Básicas de Ingeniería (2015) el Requisito de Usuario (RU), se relaciona directamente con la necesidad del cliente en relación con la automatización del grupo de datos, donde dicha necesidad es traducida al requisito del sistema. El RU describe las diferentes acciones que el usuario puede ejecutar sobre el sistema. Como se puede observar cada herramienta que se aplica, sigue evolucionando y clarificando la necesidad del cliente, de manera que pase hacia otros procesos el requisito concreto. Ver tabla 8, donde se ilustra dichos requisitos.

Requisitos funcionales

Según la Facultad de Ciencias Básicas de Ingeniería (2015) describe claramente los servicios que el sistema debe proporcionar al usuario final, ya que describe claramente cada una de las acciones que el actor ejecuta en el sistema, es así como se visualiza el rol que cada uno de ellos ejercerá cuando el software se encuentre implementado en la máquina del cliente o usuario.

Requisitos no funcionales

Según la Facultad de Ciencias Básicas de Ingeniería (2015) el requisito no funcional, está relacionado con todo aquello que no solicita directamente el cliente, en relación con lo que debe llevar o cumplir el sistema. Los RNF, tienen que ver con características externas de calidad como por ejemplo seguridad, facilidad de uso, interfaz práctica, velocidad en procesamiento, amigabilidad, es todo aquello que el sistema debe llevar, buscando con ello la satisfacción de dicho cliente o usuario. En las siguientes tablas se pueden observar algunos RNF.

Notificaciones en aplicaciones móviles

De acuerdo con Ortiz (2021) las notificaciones en aplicaciones móviles son mensajes a alertas que informan a los usuarios sobre eventos importantes, acciones o actualizaciones que requieren algún tipo de atención, se activan incluso cuando la aplicación no está siendo usada, siendo esenciales en la interacción del usuario con la máquina. Existen diferentes tipos de notificaciones, se clasifican según cómo se originan las entregas, las cuales son:

Notificaciones push: Se envían desde el servidor de las aplicaciones al dispositivo, requieren conexión a internet para ser recibidas, muy usadas en redes sociales para mostrar notificaciones, actualizaciones de estado, ofertas o alertas de actividad sospechosa en alguna cuenta asociada al usuario.

Notificaciones locales: Son programadas directamente desde el dispositivo del usuario por la propia aplicación, no requieren necesariamente que el dispositivo tenga conexión a internet ni un servidor remoto. Usado comúnmente para alertas como alarmas, cumpleaños y mensajes propios del sistema.

Notificaciones In-App: Son mensajes que aparecen mientras el usuario está activo dentro de la aplicación, no se muestran en la barra de estado del dispositivo ni pantalla de bloqueo, hacen parte de la interfaz de usuario de la aplicación. Usadas comúnmente para tutoriales propios del sistema, anuncios informativos.

Notificaciones externas: Estas no ocurren dentro de la interfaz de la aplicación, un sistema de software las envía utilizando canales de comunicaciones donde el usuario no interactúa directamente con la aplicación en el momento, ni necesariamente tenga la aplicación instalada, los mensajes se distribuyen por plataformas o servicios de terceros como correos, llamadas, mensajes de texto, WhatsApp entre otras.

Modelo entidad relación

De acuerdo con Lucidchart (s.f) un diagrama entidad relación, también conocido como modelo entidad relación o ERD, es un tipo de diagrama de flujo que ilustra cómo las 'entidades', como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Los diagramas ER se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, sistemas de información empresarial, educación e investigación. También conocidos como los ERD o modelos ER, emplean un conjunto definido de símbolos, tales como rectángulos, diamantes, óvalos y líneas de conexión para representar la interconexión de entidades, relaciones y sus atributos. Son un reflejo de la estructura gramatical y emplean entidades como sustantivos y relaciones como verbos.

Diagrama de casos de uso

De acuerdo con Patiño (2022) el diagrama de casos de uso es un diagrama UML (lenguaje de modelado unificado) que, a través de una representación gráfica del sistema, permite modelar los requerimientos funcionales de una aplicación, mostrándolos desde el punto de vista del usuario. Estos diagramas tienen dos funciones importantes: la primera es capturar los requisitos funcionales del sistema y segundo simplificar la construcción de los modelos de objetos. Estos diagramas son útiles también para que los usuarios realicen retroalimentación en la especificación de requerimientos, son insumos para posteriores etapas del ciclo de vida e incluso se utilizan como herramienta para la reingeniería de procesos.

Planteamiento del Problema

El Alzheimer es una enfermedad neurodegenerativa que deteriora la memoria, el pensamiento y la capacidad para realizar actividades cotidianas, está en aumento para la población que envejece en todo el mundo, la cual constituye una de las formas más comunes de demencia. Según la Organización Mundial de la Salud (OMS, 2021), en todo el mundo hay más de 57 millones de personas con esta patología, y el 60% de estos se encuentran en países de ingresos medianos y bajos. Se estima que cada año hay casi 10 millones de casos nuevos.

Según la Organización Panamericana de la Salud (OPS, 2023) en la región de las américas esta enfermedad va creciendo de manera rápida, más de 10 millones de personas padecen de demencia. Se estima que cada 20 años se duplicará esta cifra, siendo América Latina y el Caribe las zonas más afectadas. Para el año 2019 fue la tercera causa de muerte, siendo las mujeres las más afectadas en un 66% con respecto a los hombres.

De acuerdo con la Universidad de los Andes (2024) en Colombia no hay cifras exactas, sin embargo, se estima que la prevalencia de ya mencionada patología varía entre el 1,3% y 23,6% según diferentes estudios, siendo uno de los países que alberga mayor población afectada ya que existe una mutación genética llamada (PSEN1) asociada al Alzheimer de inicio temprano.

Tal como indica Anturi (2024), debido a las características de la patología los pacientes deben estar constantemente monitoreados, tanto en las actividades diarias como en lo relacionado a las atenciones médicas, lo cual resulta ser un proceso complejo y agotador para los cuidadores, un proceso complejo y agotador, que induce a estrés psicológico severo en

los pacientes y en los cuidadores, con consecuencias en su calidad de vida y estado de salud, además de grandes gastos económicos y sociales

Las personas con Alzheimer avanzado presentan un riesgo elevado de sufrir accidentes, desorientarse o experimentar complicaciones médicas súbitas que requieren atención inmediata. Sin embargo, debido a su condición, muchas veces no pueden comunicar efectivamente sus necesidades o solicitar ayuda, especialmente cuando se encuentran solos. Los sistemas de monitoreo tradicionales resultan insuficientes, ya que no contemplan las particularidades cognitivas de estos pacientes ni ofrecen mecanismos de verificación periódica de su bienestar con protocolos claros de respuesta ante emergencias. Esta situación genera una brecha significativa en la atención y seguridad de una población vulnerable, cuyas necesidades específicas no están siendo adecuadamente cubiertas por las soluciones tecnológicas existentes.

Se indagó en diversos sitios web y tiendas aplicaciones acerca de aplicaciones que prestarán servicios a la población con Alzheimer o integrarán un monitoreo para estos pacientes, se encontró diversidad de aplicaciones que permiten gestionar medicamentos, recordatorios y recomendaciones, sin embargo, ninguna de ellas ofrece un monitoreo constante. Debido a lo anterior surge la pregunta ¿cómo desarrollar una aplicación móvil que ayude a mejorar el bienestar y la calidad de vida de las personas que viven con Alzheimer, ofreciendo un sistema de seguimiento y alertas que facilite una atención oportuna ante cualquier eventualidad y que brinde tranquilidad tanto a los pacientes como a sus cuidadores.

Objetivos

Objetivo general

Desarrollar una aplicación móvil que ayude en el cuidado de las personas que padecen de Alzheimer, ofreciendo un sistema de seguimiento y alertas que facilite una atención oportuna ante cualquier eventualidad, brindando tranquilidad tanto a los pacientes como a sus cuidadores.

Objetivos específicos

Investigar acerca de la enfermedad del Alzheimer, desafíos que enfrentan tanto los pacientes como los cuidadores y las soluciones tecnológicas existentes, para formular adecuadamente los requerimientos de desarrollo de la aplicación.

Diseñar un prototipo funcional que incluya herramientas de seguimiento, envío de alertas y notificaciones automáticas, donde el manejo de la interfaz no requiere conocimientos técnicos por parte de los distintos actores.

Construir la aplicación de acuerdo con metodologías de desarrollo ágiles, incluyendo tecnologías móviles actuales, asegurando la operatividad entre el dispositivo y hardware añadidos.

Ejecutar pruebas de validación de operabilidad entre el dispositivo y hardware, además de verificar la usabilidad de la aplicación.

Alcance

El objetivo de este proyecto es desarrollar una herramienta tecnológica que ayude a seguir el estado de personas con Alzheimer en etapas avanzadas por medio de una aplicación móvil. Se verificará el estado del paciente con preguntas sencillas para comprobar su bienestar. Si la respuesta es negativa, se activará automáticamente una alerta para los contactos de emergencia registrados, incluyendo la ubicación en tiempo real del paciente, en caso de no recibir respuesta, el sistema generará una segunda notificación tras un tiempo determinado, y si aún no hay respuesta nuevamente se enviará la alerta automáticamente.

El sistema contará con funciones que permitirán verificar el estado del paciente de manera cíclica, lo que facilitará una atención rápida ante cualquier riesgo potencial. Además, se podrá ajustar la frecuencia de las notificaciones, gestionar los contactos de emergencia y revisar el historial de alertas. La aplicación también contará con un botón de pánico, el cual le permitirá al usuario en caso de tener algún tipo de emergencia alertar con inmediatez.

El enfoque del proyecto está diseñado para ser una herramienta de apoyo para los cuidadores, con el fin de disminuir el riesgo de desorientación y aumentar la seguridad del paciente. Es importante destacar que esta aplicación no sustituye la supervisión directa ni el acompañamiento físico.

Actividades para resolver el objetivo específico número 1

Para desarrollar el objetivo específico número 1: "Investigar acerca de la enfermedad del Alzheimer, desafíos que enfrentan tanto los pacientes como los cuidadores y las soluciones tecnológicas existentes, para formular adecuadamente los requerimientos de desarrollo de la aplicación" se tomó en consideración la relevancia de la fase de análisis de requisitos dentro del ciclo de vida del desarrollo de software.

Partiendo por lo indicado por Pressman (2010) en su libro “Ingeniería de Software: Un Enfoque Práctico”, quien indica que en las fases iniciales del desarrollo de software se debe tener en cuenta el ciclo de vida del software, específicamente la ingeniería de requisitos, ya que en esta se define qué es lo que el sistema debe hacer, las funcionalidades, rendimiento e interfaces, a partir de la comprensión del problema que se busca solucionar y necesidades de los usuarios finales. Para el levantamiento de requisitos ha sido fundamental iniciar con una encuesta, permitido recopilar de primera mano sus experiencias, los desafíos cotidianos que enfrentan y las funcionalidades que consideran más valiosas en una solución tecnológica.

Encuesta

A continuación, encontrará las preguntas que se formularon para recopilar la información, estas también las puede visualizar en:

Enlace de encuesta: <https://forms.office.com/r/ABg552bATZ>

When you submit this form, it will not automatically collect your details like name and email address unless you provide it yourself.

* Required

1. **Relación con el Alzheimer (marque una opción) ***

- Persona diagnosticada con Alzheimer
- Familiar/Cuidador de persona con Alzheimer
- Sin relación directa
- Profesional de la salud

2. **¿Qué características considera más importantes en una aplicación para localizar personas con Alzheimer? ***
(Múltiple respuesta)

- Facilidad de uso
- Alertas en tiempo real
- Precisión de la ubicación

Ilustración 1: Preguntas encuesta 1 a 2

3. **¿Con qué frecuencia utiliza aplicaciones móviles en su vida diaria? ***

- Frecuentemente (varias veces al día)
- Ocasionalmente (3-5 veces por semana)
- Constantemente (parte integral de mi rutina diaria)
- Raramente (1-2 veces por semana)
- Nunca

4. **En una escala del 1 al 5, ¿Qué tan útil considera una aplicación que monitoree periódicamente el estado de una persona con Alzheimer? ***

1	2	3	4	5
---	---	---	---	---

Nada útil Extremadamente útil

5. **¿Cree que una persona con Alzheimer avanzado podría responder de manera confiable a una pregunta simple de Sí/No en una aplicación móvil? ***

- Sí
- No
- Tal vez

Ilustración 2: Preguntas encuesta 3 a 5

6. **Si usted fuera cuidador o familiar, ¿estaría dispuesto a recibir alertas de emergencia a través de una aplicación móvil? ***

- Sí
- No
- Tal vez

7. **¿Qué tipo de interfaz sería más adecuada para que una persona con Alzheimer responda si está bien? ***

(Múltiple respuesta)

- Botones grandes con texto simple (Sí/No)
- Interfaz con imágenes/iconos (cara feliz/triste)
- Combinación de texto e imágenes

Ilustración 3: Preguntas encuesta 6 a 7

8. **¿Estaría dispuesto(a) a utilizar una aplicación que envía alertas automáticas si una persona no responde? ***

- Sí
- No
- Tal vez

9. **¿Cree que el seguimiento por GPS puede ser útil para personas con Alzheimer avanzado? ***
(Múltiple respuesta)

- Sí
- No
- Tal vez

10. **¿Le parecería útil que una app pueda notificar a varios contactos de emergencia? ***

- Sí
- No
- Tal vez

Ilustración 4: Preguntas encuesta 8 a 10

11. **¿Considera que la enfermedad de Alzheimer afecta a personas de distintos grupos etarios y no se limita únicamente a la población de edad avanzada? ***

(Múltiple respuesta)

- Sí
- No
- Tal vez

12. **¿Qué tan tranquilo(a) se sentiría sabiendo que puede localizar a su familiar en tiempo real? ***

1	2	3	4	5
---	---	---	---	---

Nada útil

Extremadamente
útil

Ilustración 5: Preguntas encuesta 11 a 12

13. **¿Quién debería tener acceso a las alertas y ubicación?** *

(Múltiple respuesta)

- Familiares
- Cuidadores
- Personal médico
- Vecinos de confianza

14. **¿Le preocuparía que su familiar con Alzheimer se desoriente fuera de casa?** *

(Múltiple respuesta)

- Sí
- No
- Tal vez

15. **¿Confía en el uso de la tecnología para mejorar el cuidado de personas con Alzheimer?** *

- Sí
- No
- Tal vez

Ilustración 6: Preguntas encuesta 13 a 15

16. **¿Qué tan de acuerdo está con que una app puede reducir el estrés del cuidador?** *

1	2	3	4	5
---	---	---	---	---

Nada útil

Extremadamente
útil

17. **¿Qué tan cómodo se siente utilizando tecnología para monitorear la salud o ubicación?** *

1	2	3	4	5
---	---	---	---	---

Nada útil

Extremadamente
útil

Ilustración 7: Preguntas encuesta 16 a 17

18. **¿En qué situaciones considera más útil esta aplicación? ***

(Múltiple respuesta)

- Caminatas o paseos fuera de casa
- Cuando la persona está sola
- Durante el traslado a centros de salud
- En el hogar, como respaldo preventivo

19. **¿Qué obstáculos cree que podrían dificultar el uso de esta aplicación? ***

(Múltiple respuesta)

- Falta de acceso a un smartphone
- Dificultades para entender la tecnología
- Desconfianza en la tecnología
- Costos del dispositivo o app

20. **¿Cree que una app con alertas puede prevenir emergencias en personas con Alzheimer? ***

(Múltiple respuesta)

- Sí
- No
- Tal vez

Ilustración 8: Preguntas encuesta 18 a20

Resultados de la encuesta

La encuesta aplicada a 20 participantes revela una diversidad en la relación con el Alzheimer, donde se observa que un 40% se identificó sin tener una relación directa, mientras que el 35% indicó desempeñarse como familiar o cuidador; además, el 15% afirmó ser una persona diagnosticada con Alzheimer y el 10% se catalogó como profesional de la salud. Esta distribución sugiere que la percepción sobre el uso de aplicaciones tecnológicas para el seguimiento y cuidado de la enfermedad varía según la experiencia y el vínculo personal con el Alzheimer.

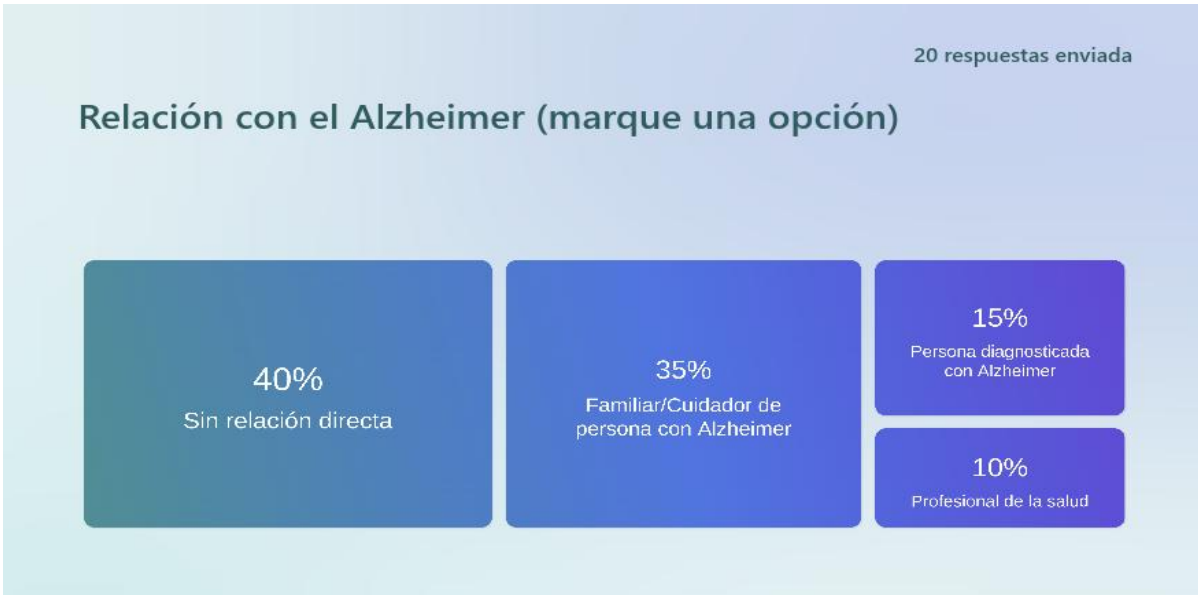


Ilustración 9: Resultado encuesta pregunta 1

Por otra parte, al analizar las características más importantes para una aplicación, se destaca que la “facilidad de uso” es prioritaria para el 43% de los encuestados, seguida de las “alertas en tiempo real” mencionadas por el 33% y la “precisión de la ubicación” señalada por el 23%. En consecuencia, se deduce que los usuarios buscan una interfaz intuitiva y confiable que respalde el monitoreo constante de la localización.



Ilustración 10: Resultado encuesta pregunta 2

En cuanto a la frecuencia de uso de aplicaciones móviles, el 60% de los participantes afirmó usarlas varias veces al día, mientras que el 20% indicó que integran estas aplicaciones de manera constante en su rutina diaria y otro 20% las utiliza de forma ocasional (3-5 veces por semana). De este modo, se evidencia una amplia familiaridad y comodidad con la tecnología móvil por parte de la mayoría de los encuestados.

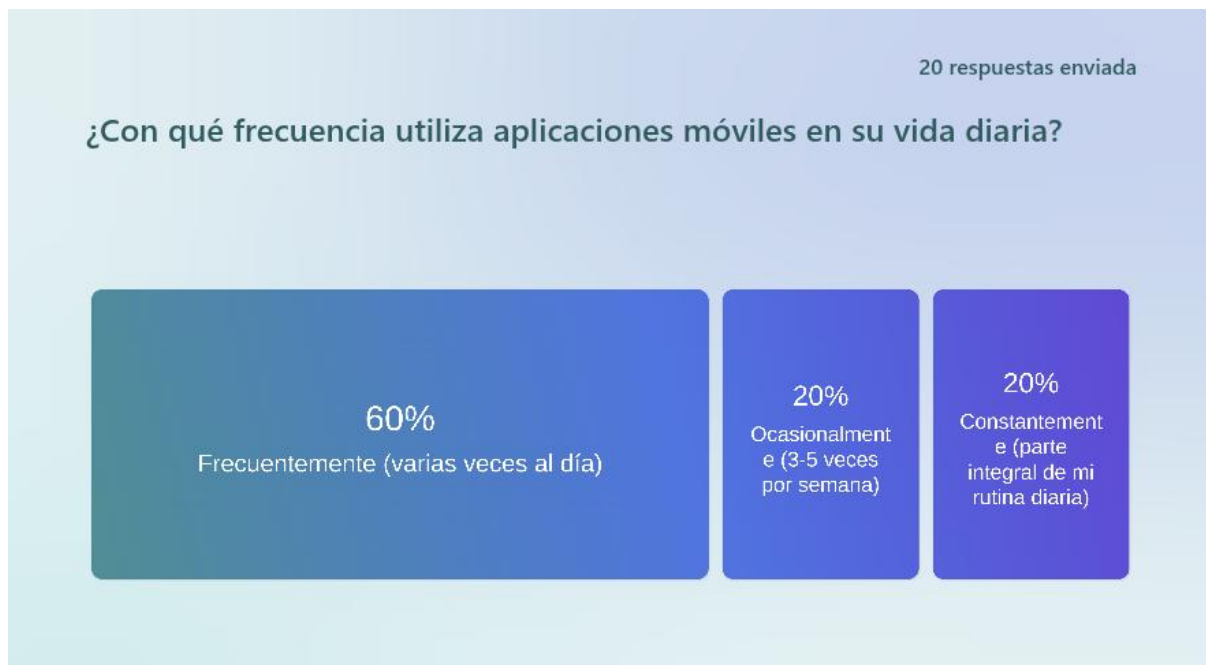


Ilustración 11: Resultado encuesta pregunta 3

Respecto a la utilidad de una aplicación que monitoree el estado del Alzheimer, se observa una alta valoración, ya que el 55% la calificó con el máximo puntaje (5), el 30% otorgó un 4 y el 15% optó por un 3. Esta tendencia se complementa con la percepción sobre la capacidad de una persona con Alzheimer avanzado para responder de manera confiable a preguntas simples; aquí, el 55% respondió afirmativamente, el 30% se mostró indeciso y el 15% opinó negativamente, evidenciando una confianza mayoritaria, aunque con cierto grado de incertidumbre.

20 respuestas enviadas

En una escala del 1 al 5, ¿Qué tan útil considera una aplicación que monitoree periódicamente el estado de una persona con Alzheimer?

4.40

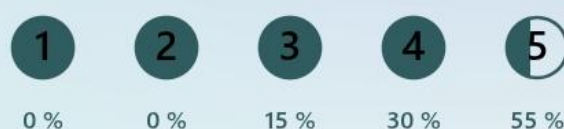


Ilustración 12: Resultado encuesta pregunta 4

Asimismo, la disposición a recibir alertas de emergencia a través de una aplicación fue altamente positiva, alcanzando un 85% de aceptación, mientras que apenas un 10% se mostró indeciso y un 5% rechazó la idea. En este sentido, los resultados también indican que la mayoría considera adecuada una interfaz visual basada en imágenes o íconos, opción elegida por el 70% de los participantes, complementada por el uso de botones grandes con texto simple (45%) y una combinación de texto e imágenes (35%). Estos hallazgos refuerzan la necesidad de contar con un diseño accesible y claro.

Por otro lado, el seguimiento por GPS se percibió como muy útil por el 85% de los encuestados, lo que se relaciona estrechamente con la valoración del sistema de notificación a varios contactos de emergencia, el cual fue aceptado por el 85% de los participantes, dejando solo un 15% con dudas o rechazo. En conjunto, estos aspectos demuestran que contar con tecnologías que permitan la localización y comunicación efectiva en situaciones críticas es fundamental para los usuarios.

Al analizar la percepción sobre la afectación etaria del Alzheimer, el 75% está de acuerdo en que la enfermedad afecta a distintos grupos, no limitándose únicamente a las personas de edad avanzada, mientras que un 20% se opuso y un 5% se mostró indeciso. Esta diversidad en opiniones se refleja también en la sensación de seguridad que genera la posibilidad de localizar a un familiar en tiempo real; en este caso, casi la mitad (45%) otorgó la máxima calificación (5) en una escala de 1 a 5, seguida por un 35% con valor 4, y menores calificaciones en el resto de la muestra.

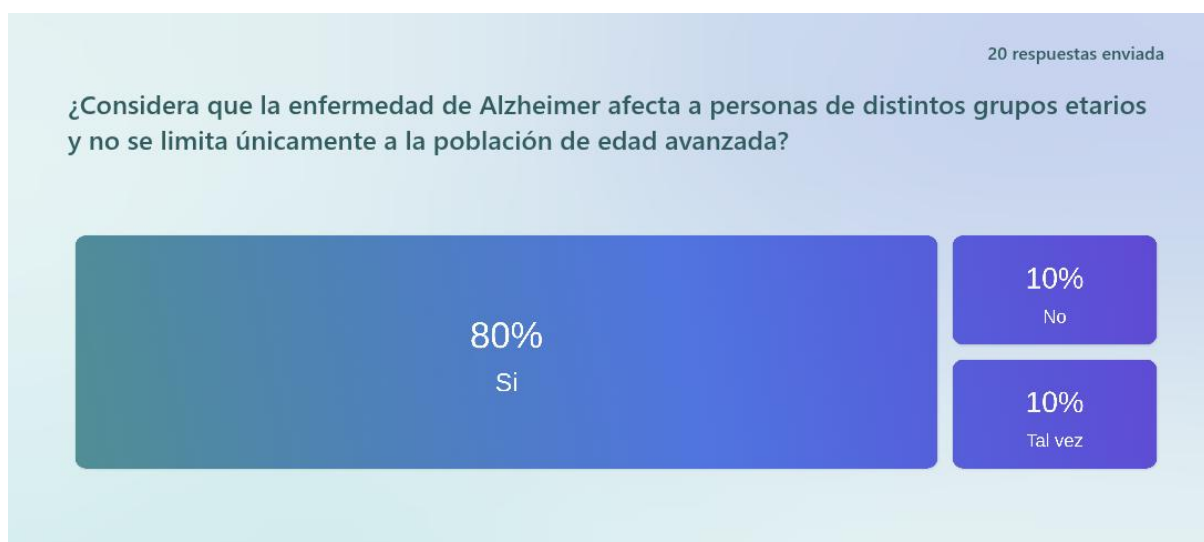


Ilustración 13: Resultado encuesta pregunta 8

En relación con el acceso a la información generada por alertas y la localización, los participantes coincidieron en que debería estar disponible principalmente para familiares (95%) y cuidadores (80%), complementado en menor medida con vecinos de confianza (25%) y personal médico (20%). Así mismo, surge una marcada preocupación, ya que el 85% indicó inquietud por la posibilidad de que su familiar se desoriente fuera del hogar, reflejando la necesidad de implementar medidas de seguridad efectivas.

Por otra parte, la confianza en el uso de la tecnología para mejorar el cuidado de personas con Alzheimer fue casi unánime, con un 90% de aprobación, mientras que solo un

5% dudó y otro 5% se mostró en desacuerdo. En paralelo, la percepción de que una aplicación puede ayudar a reducir el estrés del cuidador se manifestó con predominio de respuestas en el rango de 4 y 5, representando un 55% y un 35% respectivamente, lo que evidencia la expectativa de que estas herramientas pueden aliviar la carga emocional del cuidado.

La comodidad en el uso de tecnologías para monitorear la salud o ubicación también fue alta, con un 55% valorándose con un 4 y un 40% con un 5, mientras que apenas un 5% manifestó sentirse moderadamente cómodo, reafirmando que la mayoría posee confianza en este tipo de soluciones digitales.

Además, al considerar las situaciones en que la aplicación podría resultar más útil, destaca que el 80% la ve como esencial durante caminatas o paseos fuera de casa, y el 75% cuando la persona está sola, mientras que situaciones complementarias como su uso en el hogar como respaldo preventivo (30%) o durante el traslado a centros de salud (10%) tuvieron menor relevancia, lo que subraya la importancia de intervenciones en escenarios de mayor vulnerabilidad.

En cuanto a los obstáculos que podrían dificultar el uso de la aplicación, se identificaron principalmente las dificultades para entender la tecnología (mencionadas por el 75% de los participantes) y la falta de acceso a un smartphone (55%), si bien otros factores como los costos del dispositivo o la aplicación (15%) y la desconfianza en la tecnología (20%) también fueron señalados, lo que indica que para una implementación exitosa es crucial facilitar la accesibilidad y la usabilidad.

La percepción sobre la capacidad de una aplicación con alertas para prevenir emergencias es notablemente positiva, ya que el 95% considera que puede desempeñar un rol preventivo esencial, mientras que solo un 5% manifestó cierta incertidumbre.

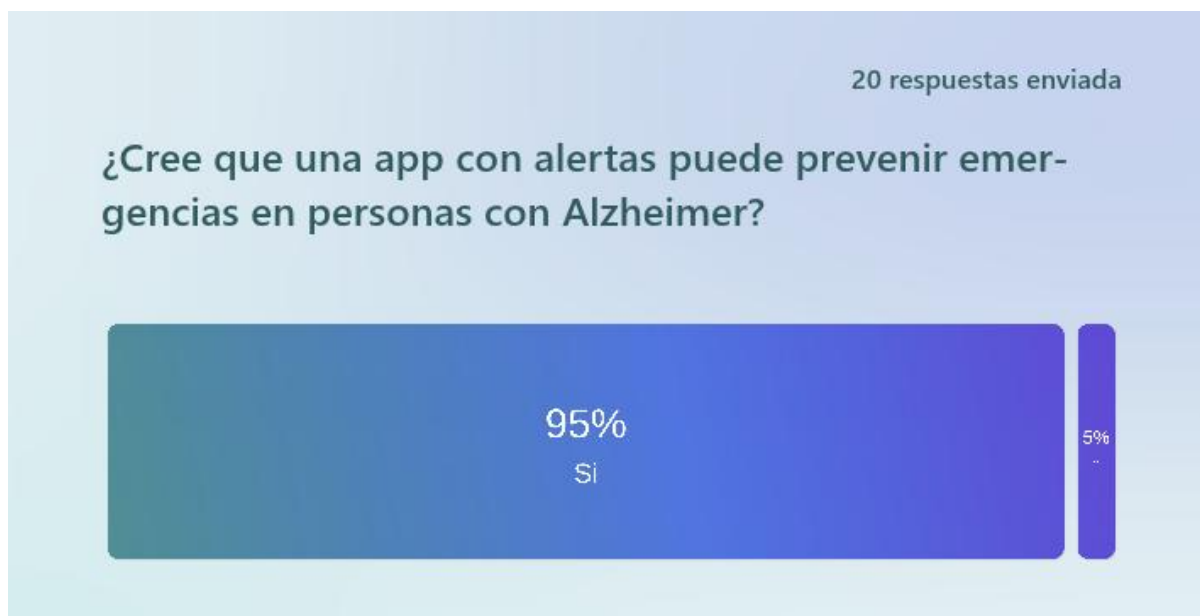


Ilustración 14: Resultado encuesta pregunta 20

Identificación de usuarios claves

En este proyecto de monitoreo para personas con alzhéimer se identificaron tres grupos clave que representan un ecosistema complejo de necesidades, capacidades y expectativas, cada uno con un papel fundamental en el cuidado y apoyo de pacientes con esta enfermedad neurodegenerativa.

Actor	Descripción
Paciente con Alzheimer	Persona con alzhéimer avanzado, quien es el usuario central del sistema. La aplicación está diseñada con una interfaz simple para que pueda responder preguntas periódicas sobre su estado. Si no responde o indica malestar, se activa un sistema de alertas.
Cuidador	Familiar o cuidador, quien recibe las notificaciones de alerta ante situaciones de riesgo que fuesen generadas al momento del monitoreo del paciente

Administrador	Encargado de la gestión técnica de la aplicación. Se ocupa de configurar la plataforma, asegurar su correcto funcionamiento y mantener la integridad de los datos del sistema.
---------------	--

Tabla 4: Actores del sistema

Requisitos de usuario

ID	Requisito	Descripción
RU-001	Gestión de usuarios	La aplicación debe permitir el registro de usuarios, utilizando correo electrónico y contraseña como credenciales, también permitir la recuperación de la contraseña a través del correo electrónico. Los roles y permisos se asignarán según el tipo de usuario (actor).
RU-002	Gestión de preguntas de verificación y envío de alertas	El sistema debe permitir gestionar la periodicidad en que se realizan las preguntas de verificación de estado del paciente (con respuestas de 'sí' o 'no'). En caso de una respuesta negativa o de no recibir respuesta se debe enviar una alerta a los contactos de emergencia por mensaje de texto.
RU-003	Gestión de consultas, historial de ubicaciones	El sistema debe permitir consultar el historial de ubicaciones generado por el paciente, este permitirá filtrar por fechas.

Tabla 5: Requisitos de usuario

Requisitos funcionales (RF)

Id requisito	Nombre del requisito	Descripción del requisito	Actor	Id requisito usuario
RF-001	Crear un usuario	El sistema debe permitir crear los usuarios, se solicitará información para el registro de acuerdo con el rol escogido (paciente,	Paciente, cuidador y administrador	RU-001

		cuidador, administrador), utilizando el correo electrónico y una contraseña como credenciales.		
RF-002	Consultar usuario	El sistema debe permitir consultar la información de los usuarios.	Paciente, cuidador y administrador	RU-001
RF-003	Modificar usuario	El sistema debe permitir modificar la información de los usuarios ya previamente registrados.	Paciente, cuidador y administrador	RU-001
RF-004	Inhabilitar un usuario	El sistema debe permitir inhabilitar un usuario.	Cuidador y administrador	RU-001
RF-005	Consultar la frecuencia de las preguntas de verificación	El sistema debe permitir consultar los intervalos de tiempo en que se encuentran las preguntas de verificación de estado que se les realizan a los usuarios.	Paciente, cuidador y administrador	RU-002
RF-006	Modificar la frecuencia de preguntas de verificación	El sistema debe permitir que un usuario con el rol cuidador configure la periodicidad de las preguntas de verificación de estado para cada uno de los usuarios con el rol de pacientes que estén a su cargo, las opciones de respuesta serán si o no.	Cuidador y administrador	RU-002
RF-007	Inhabilitar preguntas de verificación	El sistema debe permitir que se inhabilite las preguntas de verificación en caso de que el paciente tenga supervisión directa y no sea necesario su uso.	Cuidador y administrador	RU-002
RF-008	Consultar el historial	El sistema deberá permitir consultar el	Paciente, cuidador y administrador	RU-003

	ubicaciones enviadas	historial de ubicaciones que han sido enviadas a los contactos de emergencia cuando se genera la alerta, este permitirá filtrar por fechas facilitando la consulta.		
--	----------------------	---	--	--

Requisitos no funcionales (RNF)

ID	Descripción del requisito	Tipo
RNF-001	El sistema debe estar disponible las 24 horas del día, los 7 días de la semana, los 365 días del año	Confiabilidad
RNF-002	La aplicación debe tener un tiempo de respuesta rápido al cargar pantallas, con una latencia máxima de 3 segundos en condiciones de red óptimas.	Rendimiento
RNF-003	La aplicación debe funcionar en al menos el 90 % de dispositivos Android desde la versión 8.0 en adelante, solo para dispositivos móviles.	Compatibilidad

Tabla 6: Requisitos no funcionales

Escenarios de los casos de uso

Nombre	Crear usuario	
Descripción	Permite registrar la información del usuario como nombres, fecha de nacimiento, dirección, correo electrónico, información médica (diagnóstico y medicamentos que ingiere), contactos de emergencia	
Actor	Paciente/Cuidador/Administrador	
Precondiciones	El usuario debe tener un correo electrónico	
Flujo básico		
Pasos	Actor	Sistema

	<p>1. El usuario ingresara a la opción registrarse.</p>	<p>2. El sistema abrirá la interfaz con las casillas de correo electrónico, contraseña y confirmar contraseña habilitados para llenar la información.</p>
	<p>3. El usuario da clic en botón Registrarse.</p>	<p>4. El sistema muestra la siguiente pantalla del formulario con las casillas de nombre completo, fecha de nacimiento y dirección habilitadas que deberá ser llenada para poder continuar</p>
	<p>5. El usuario da clic en Continuar</p>	<p>6. El sistema muestra la siguiente pantalla del formulario con las casillas Diagnóstico como campo obligatorio, Alergias, Nombre del médico y Teléfono del médico que deberán ser llenadas si aplica.</p>
	<p>7. El usuario da clic en Continuar</p>	<p>8. El sistema muestra la siguiente pantalla del formulario donde solicitará el ingreso de al menos un contacto de emergencia.</p>

	9. El usuario da clic en el botón más (+) para agregar contacto.	10. El sistema abrirá un recuadro con las casillas nombre teléfono, lista de opciones del parentesco y una casilla de verificación si es contacto primario o no.
	11. El usuario da clic en continuar	12. El sistema redirige a la pantalla de inicio de sesión para permitir el ingreso con las credenciales recién creadas.
	13. Clic en salir y/o volver	
Flujo alternativo		
Pasos	Actor	Sistema
		1. El sistema detectará que los campos obligatorios no están vacíos.
	2. El usuario corrige la información y continúa.	3. El sistema verifica los datos nuevamente.
		4. El sistema almacena los datos de forma exitosa.
	5. Clic en Salir	
Postcondiciones	Se creó un nuevo usuario	
Requisito Funcional	RF-002	

Tabla 7: Escenario caso de usos (Crear usuario)

Nombre	Modificar usuario	
Descripción	Permite Modificar la información del usuario como nombres, fecha de nacimiento, dirección, correo electrónico, información médica, contactos de emergencia	
Actor	Paciente/Cuidador	
Precondiciones	El usuario debe iniciar sesión.	
Flujo básico		
Pasos	Actor	Sistema
	1. El usuario ingresara a la opción perfil.	2. El sistema mostrará la interfaz del perfil con la foto de usuario nombre edad y dirección con las opciones: Datos personales, Información médica, Contactos de emergencia y Acerca de la aplicación.
	3. El usuario da clic en la opción que desea modificar.	4. El sistema abrirá la interfaz correspondiente con los campos habilitados para su edición.
	5. El usuario modifica los datos y presiona guardar	6. El sistema validará y guardará la información y volverá a la interfaz del perfil.
	7. Clic en salir y/o volver	

Flujo alternativo		
Pasos	Actor	Sistema
		1. El sistema detectará que los campos obligatorios no están vacíos.
	2. El usuario corrige la información y continúa.	3. El sistema verifica los datos nuevamente.
		4. El sistema almacena los datos de forma exitosa.
	5. clic en salir y/o volver	
Postcondiciones	Se modificó un usuario	
Requisito Funcional	RF-003	

Tabla 8: Escenario caso de uso (Modificar usuario)

Nombre	Consultar usuario	
Descripción	Permite consultar la información del usuario como nombres, fecha de nacimiento, dirección, correo electrónico, información médica, contactos de emergencia, historial de ubicaciones.	
Actor	Paciente/Cuidador	
Precondiciones	El usuario debe iniciar sesión.	
Flujo Básico		
Pasos	Actor	Sistema

	1. El usuario ingresara a la opción perfil.	2. El sistema mostrará la interfaz del perfil con la foto de usuario nombre edad y dirección con las opciones: Datos personales, Información médica, Contactos de emergencia y Acerca de la aplicación.
	3. El usuario da clic en la opción que desea consultar.	4. El sistema abrirá la interfaz correspondiente con los campos llenos con la información actual
	5. El usuario Presiona volver	6. El sistema volverá a la interfaz del perfil.
	7. Clic en salir y/o volver	

Flujo alternativo

Pasos	Actor	Sistema
		1. El sistema detectará que los campos obligatorios no están vacíos.
	2. El usuario corrige la información y continúa.	3. El sistema verifica los datos nuevamente.
		4. El sistema almacena los datos de forma exitosa.

	5. Clic en salir y/o volver	
Postcondiciones	Se consultó un Paciente	
Requisito Funcional	RF-002	

Tabla 9: Escenario caso de uso (Consultar usuario)

Nombre	Inhabilitar usuario	
Descripción	Permite eliminar de forma permanente el perfil de un usuario, y toda la información asociada.	
Actor	Administrador	
Precondiciones	El usuario debe estar logueado	
Flujo básico		
Pasos	Actor	Sistema
	1. El usuario ingresara a la opción perfil administrador.	2. El sistema mostrará la interfaz del administrador con la opción de usuarios habilitada
	3. el usuario dará clic en la opción usuario.	4. El sistema abrirá una interfaz con el listado de usuarios registrados.

	5. El usuario selecciona la opción de eliminar del usuario deseado.	6. El sistema mostrará un mensaje de confirmación.
	7. El usuario selecciona en aceptar	8. El sistema mostrará un mensaje de eliminación exitosa y volverá a la interfaz del administrador
	9. Clic en salir y/o volver	
Flujo alternativo		
Pasos	Actor	Sistema
		1. Si hay error de conexión o fallo de base de datos, el sistema muestra un mensaje de error y no realiza la acción.
	2. Si el usuario cancela en el paso 7, el sistema interrumpe el proceso sin eliminar los datos.	3. El sistema regresará a la interfaz de administrador
	4. Clic en salir y/o volver	
Postcondiciones	Se Eliminó un usuario	
Requisito Funcional	RF-004	

Tabla 10: Escenario caso de uso (Inhabilitar usuario)

Nombre	Consulta frecuencia preguntas de verificación	
Descripción	Permite consultar los intervalos de tiempo en que están configuradas las preguntas de verificación que es le hacen al paciente para verificar su estado	
Actor	Paciente/Administrador/Cuidador	
Precondiciones	El usuario debe estar logueado	
Flujo básico		
Pasos	Actor	Sistema
	1. El usuario deberá ingresar a la opción de configuración de alertas	2. El sistema abrirá la interfaz con la configuración de las alertas, con dos listas desplegables, una con intervalos de una hora para la alerta principal y otra con intervalos de 5 minutos para la alerta secundaria que se activará en caso de no tener respuesta en la alerta principal.
	3. Clic en salir y/o volver	
Flujo alternativo		
Pasos	1. El usuario cambia de interfaz sin presionar guardar configuración.	2. el sistema navegará a la interfaz seleccionada por el usuario.

Postcondiciones		
Requisito	RF-005	
Funcional		

Tabla 11: Escenario caso de uso (Consulta frecuencia preguntas de verificación)

Nombre	Modificación frecuencia preguntas de verificación	
Descripción	Permite modificar los intervalos de tiempo en que están configuradas las preguntas de verificación que es le hacen al paciente para verificar su estado	
Actor	Paciente/Administrador/Cuidador	
Precondiciones	El usuario debe estar logueado	
Flujo básico		
Pasos	Actor	Sistema
	1.El usuario deberá ingresar a la opción de configuración de alertas	2. El sistema abrirá la interfaz con la configuración de las alertas, con dos listas desplegables, una con intervalos de una hora para la alerta principal y otra con intervalos de 5 minutos para la alerta secundaria que se activará en caso de no tener respuesta en la alerta principal.
	3. El usuario establecerá el intervalo de tiempo deseado tanto en la	4. El sistema toma la elección que elija el usuario tanto para la

	validación principal como en la secundaria.	validación principal como la secundaria.
	5. El usuario presionara el botón de guardar configuración para establecer los intervalos deseados.	6.El sistema guardara la configuración seleccionada por el usuario.
	7.Clic en salir y/o volver	
Flujo alternativo		
Pasos	8. El usuario cambia de interfaz sin presionar guardar configuración.	9. el sistema navegará a la interfaz seleccionada por el usuario.
Postcondiciones		
Requisito Funcional	RF-006	

Tabla 12: Escenario caso de uso (Modificación frecuencia preguntas de verificación)

Nombre	Inhabilitar frecuencia preguntas de verificación
Descripción	Permite inactivar la aparición de las preguntas de verificación de estado
Actor	Administrador/Cuidador
Precondiciones	El usuario debe estar logueado
Flujo básico	

Pasos	Actor	Sistema
	1. El usuario deberá ingresar a la opción de inicio	2. El sistema abrirá la interfaz de inicio donde mostrará un botón de activar o desactivar sistema, el nombre del usuario, el estado actual del sistema y el botón de pánico.
	3. El usuario presiona el botón de inactivar.	4. El sistema mostrará inhabilitaba el botón de pánico y suspenderá las notificaciones de bienestar, mostrará un mensaje de alerta de sistema desactivado.
	3. Clic en salir y/o volver	
Flujo alternativo		
Pasos	1. El usuario cambia de interfaz sin presionar el botón de inactivar sistema.	2. El sistema navegará a la interfaz seleccionada por el usuario, el sistema seguirá activo por defecto.
Postcondiciones	Se inactivó el envío de alertas.	
Requisito Funcional	RF-007	

Tabla 13: Escenario caso de uso (Inhabilitar frecuencia preguntas de verificación)

Nombre	Consulta historial de ubicaciones	
Descripción	Permite al usuario realizar consultas de historial de ubicaciones enviados como alertas, se guardarán datos como latitud, longitud fecha y hora, además se podrá usar filtros por fecha.	
Actor	Administrador/Cuidador/Paciente	
Precondiciones	El usuario debe estar logueado	
Flujo básico		
Pasos	Actor	Sistema
	1. El usuario deberá ingresar a la opción de ubicaciones	2. El sistema abrirá la interfaz del historial de ubicaciones donde mostrará una lista de todas las ubicaciones por defecto con la dirección, ciudad, país, fecha y hora. Tendrá una lista desplegable con todas las fechas de las alertas generadas que permitirá filtrar.
	3. El usuario selecciona la lista desplegable y elige una fecha.	4. El sistema mostrará la lista de ubicaciones generadas en la fecha seleccionada.
	5. Clic en salir y/o volver	

Postcondiciones	Se consultó el historial de ubicaciones.	
Requisito Funcional	RF-008	

Tabla 14: Escenario Caso de uso (Consulta historial de ubicación)

Diagramas modelo de casos de uso

Un diagrama de casos de uso puede resumir los detalles de los usuarios de su sistema (también conocidos como actores) y sus interacciones con el sistema. Para crearlo, utilizará un conjunto de símbolos y conectores especializados (Lucidchart, s.f.).

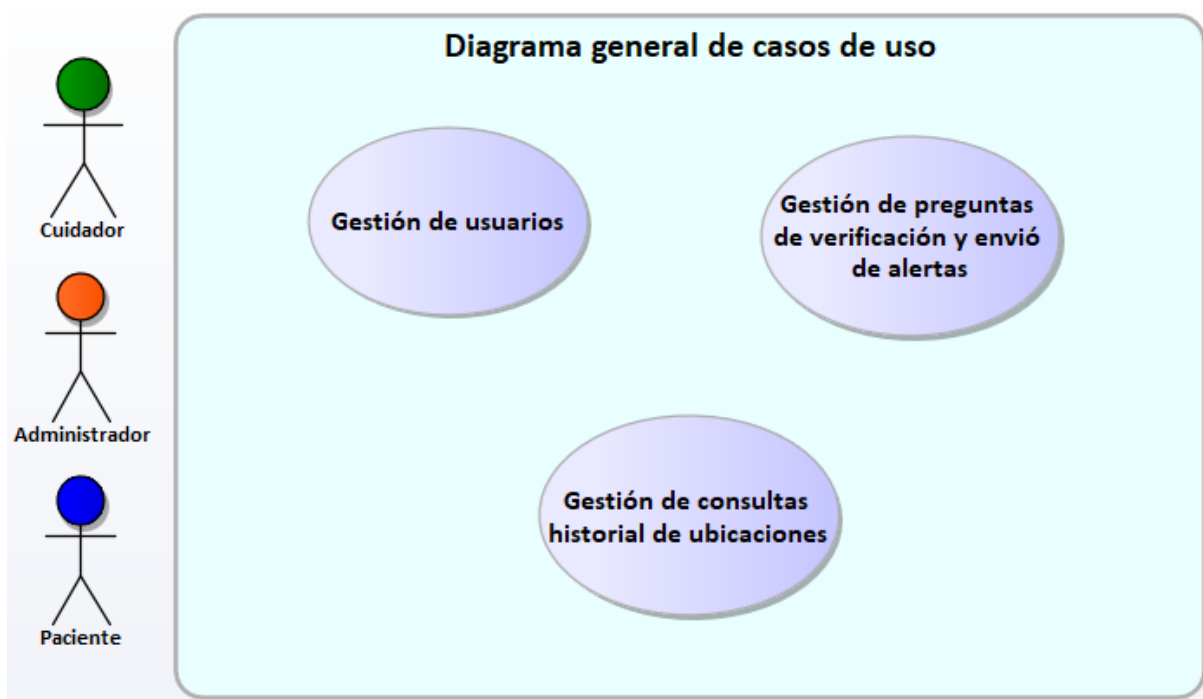


Ilustración 15: Diagrama general de casos de uso

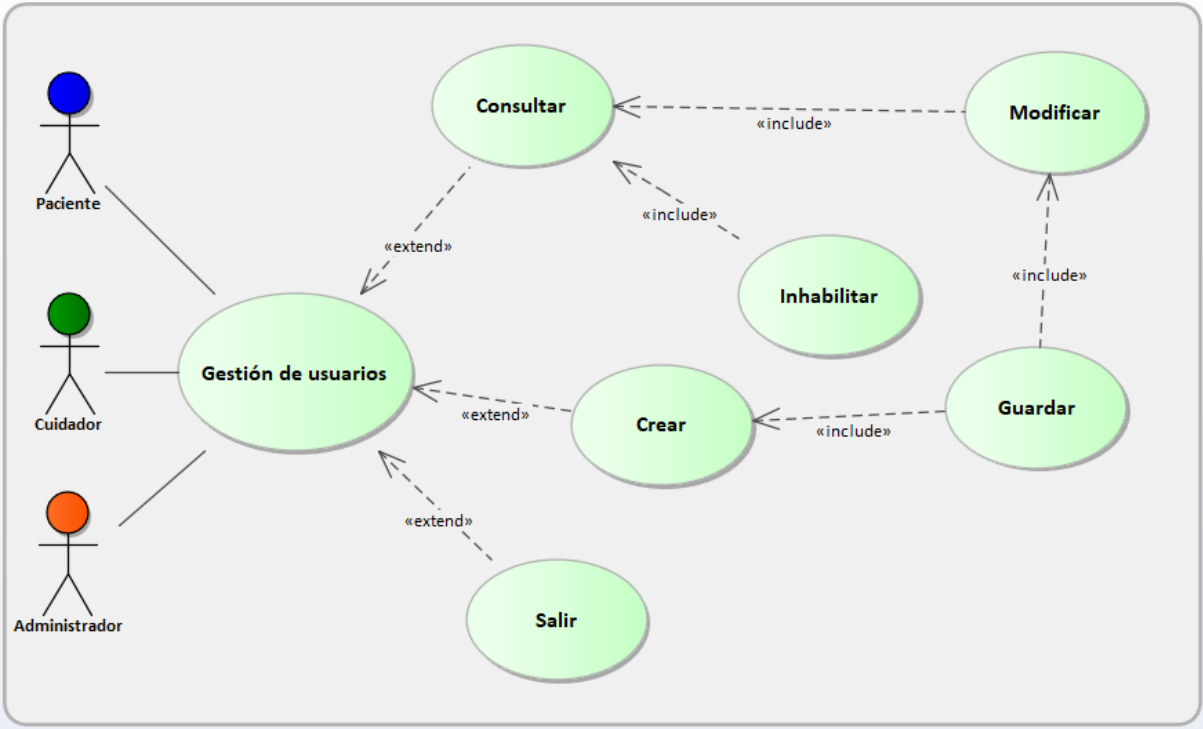


Ilustración 16: Caso de uso extendido (Gestión de usuarios)

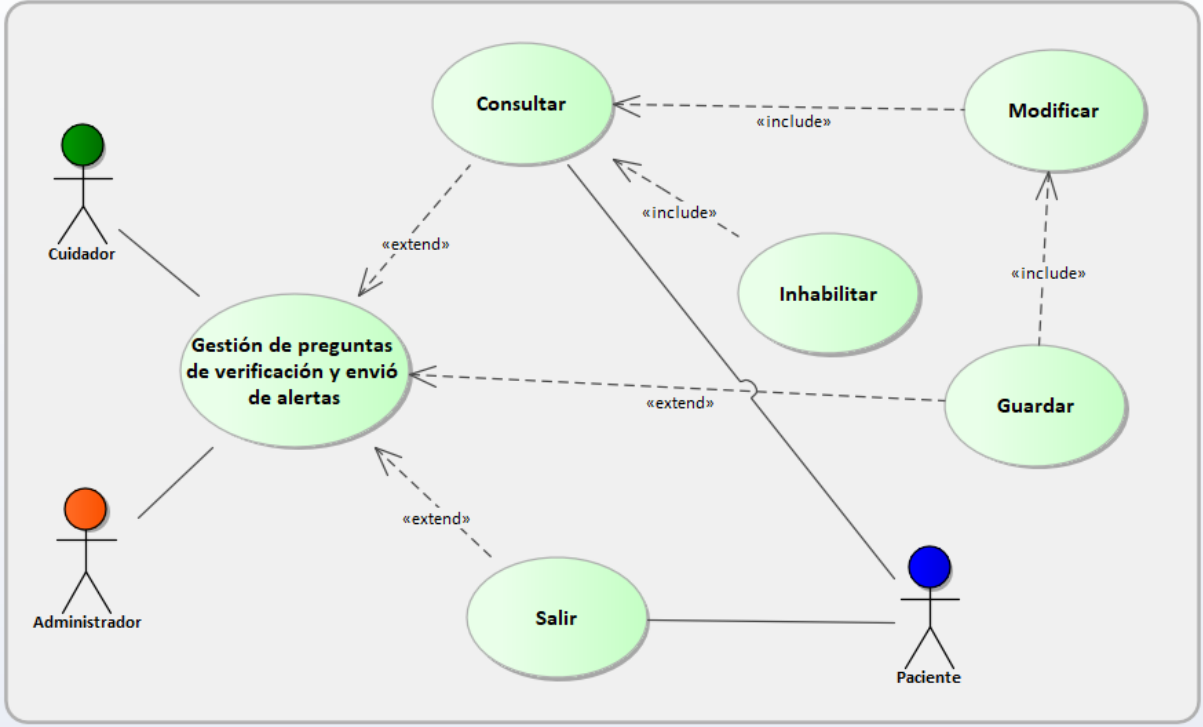


Ilustración 17: Caso de uso extendido (Gestión de Preguntas de Verificación)

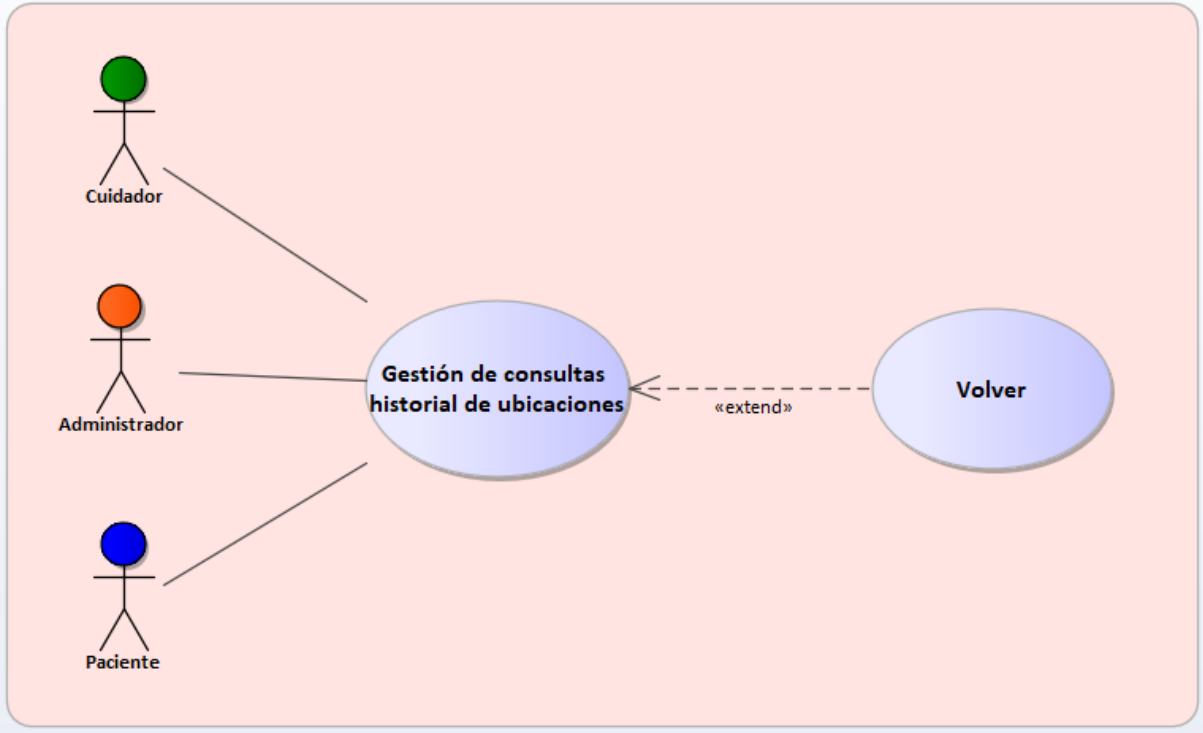


Ilustración 18: Caso de uso extendido (Gestión de consulta)

Diagrama de clases

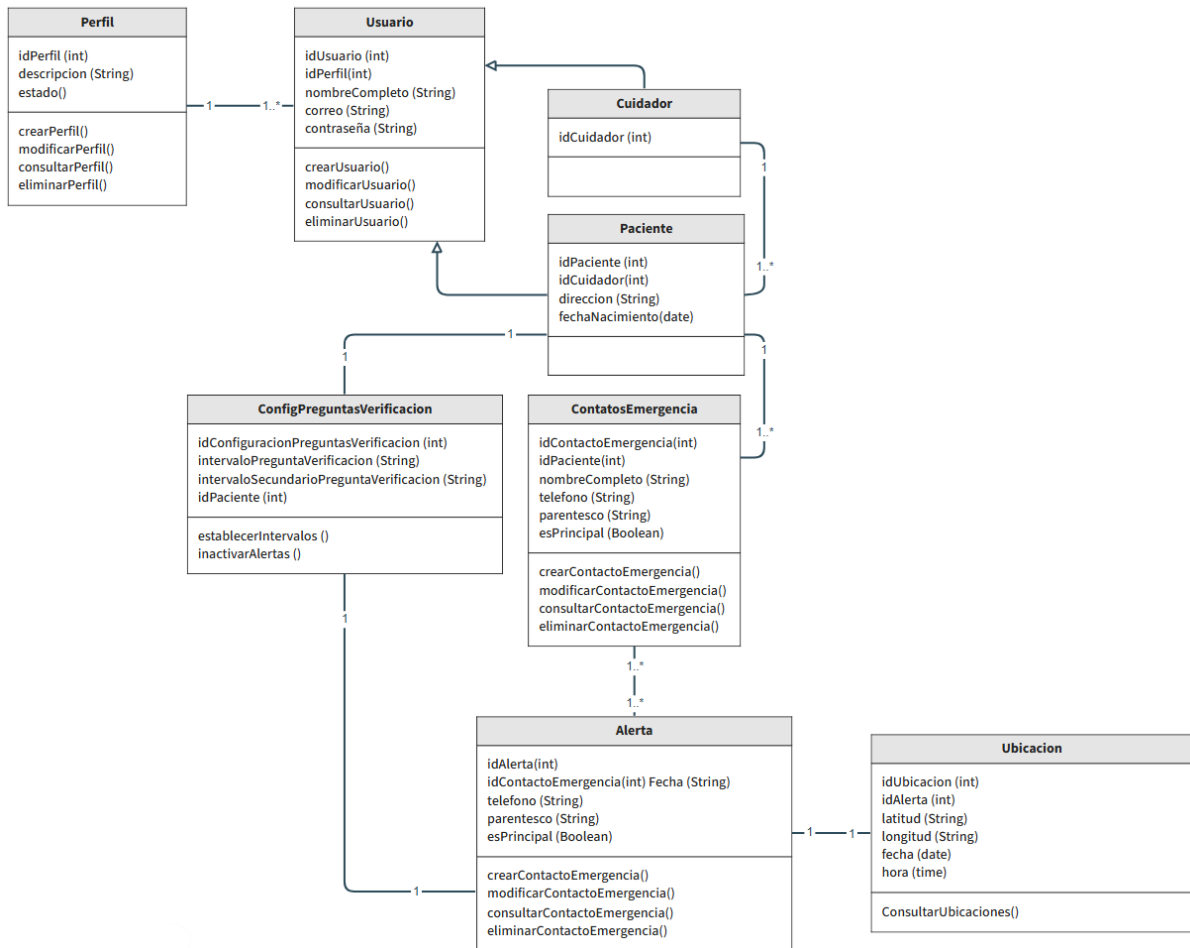


Ilustración 19: Diagrama de clases

Modelo entidad relación

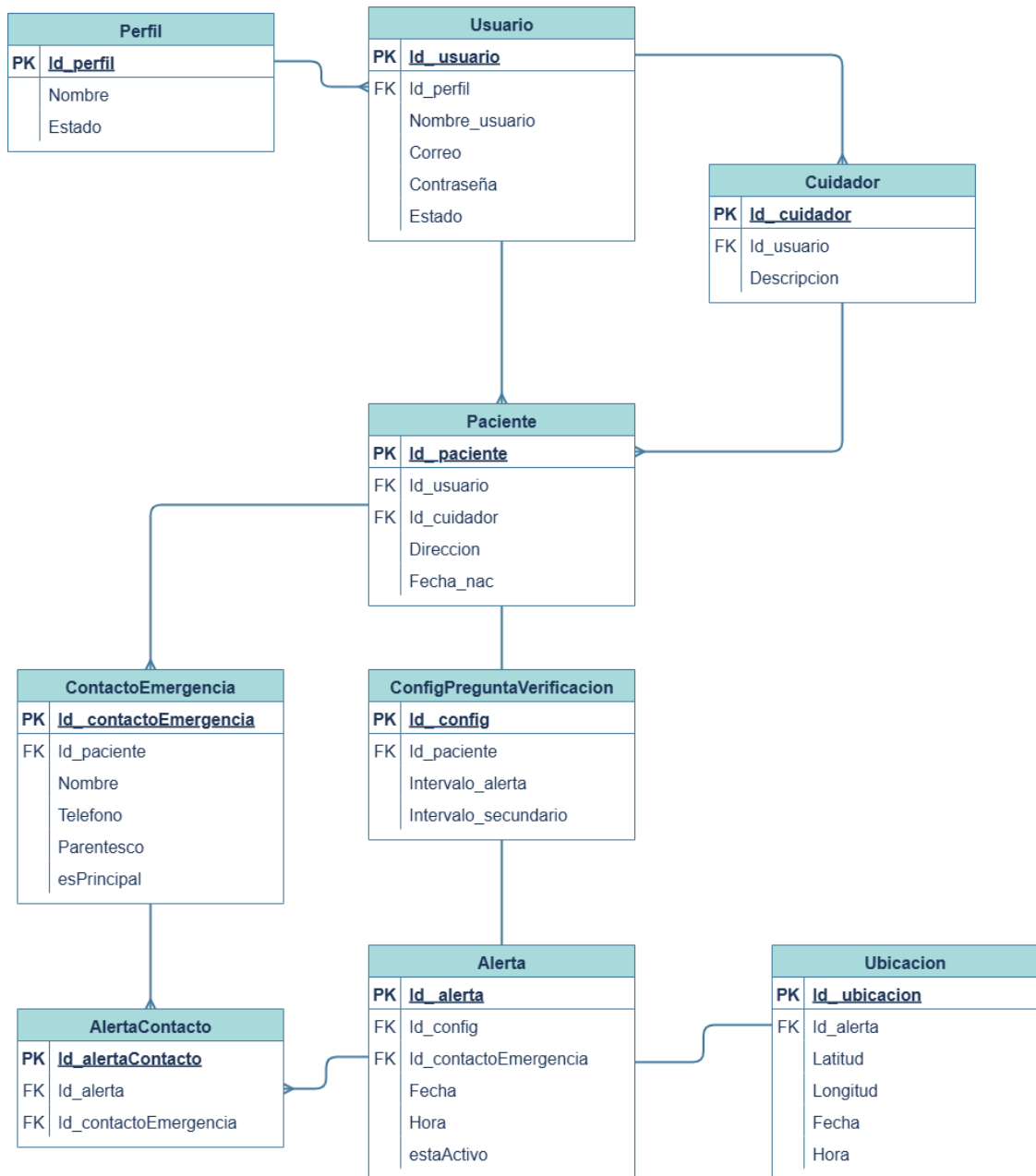


Ilustración 20: Modelo entidad relación

Metodología de trabajo ágil

Según Narváz (2024) las metodologías de desarrollo ágil representan un paradigma de desarrollo y gestión de proyectos que prioriza la flexibilidad y la entrega de valor continua sobre la rigidez de la planificación tradicional., a diferencia de los enfoques lineales, donde las fases se completan secuencialmente, las metodologías ágiles operan mediante ciclos iterativos e incrementales, permitiendo una adaptación constante a los requisitos cambiantes del cliente y del mercado.

Para el desarrollo de la aplicación se concluya con éxito se eligió la metodología ágil llamada Scrum, ya que a lo largo de la fase de construcción del prototipo se pueden encontrar oportunidades de mejora, adiciones o modificaciones que no se contemplen inicialmente, y esta al no ser rígida permite que se pueda ajustar a medida que se avanza.

Scrum es una forma de trabajar en equipo organizada y flexible, permite construir la aplicación por partes, en ciclos cortos de tiempo para ir ajustando en el camino. Al final de cada ciclo llamado técnicamente como sprint se cuenta con una versión del prototipo funcional, que permite ser mostrado, probar y generar retroalimentación para el siguiente sprint.

Roles en el proyecto

Aunque solo dos personas intervienen en el desarrollo del proyecto completo, la metodología Scrum indica que es importante asumir roles predefinidos por la misma para mantener el orden durante todo el ciclo de vida del proyecto, para ello los autores se distribuyeron de la siguiente manera:

Product Owner: Ambos autores son encargados de decidir qué es lo más importante para la aplicación, para esto, se realizó durante la una encuesta y revisión bibliográfica para

identificar las necesidades que conllevan a la definición de requisitos, este rol tiene la responsabilidad de asegurar de que se construya lo que realmente se necesita.

Scrum Master: Ambos autores son encargados de que el proceso funcione sin tropiezos, resolviendo problemas técnicos y administrativos que puedan impedir la continuidad del proyecto.

Equipo de desarrollo: Ambos autores son encargados de desarrollar todo lo diseñando, es decir construir la aplicación con las indicaciones y requerimientos.

Los Sprint

Para este proyecto cada Sprint durará dos semanas. Al inicio de cada Sprint, se realiza un plan para decidir qué funcionalidades de la lista principal se van a implementar en este rango de tiempo. De acuerdo con la metodología Scrum cada día se reúne todo el equipo para responder a las preguntas de ¿qué hice ayer?, ¿qué voy a hacer hoy? y ¿tengo algún problema que me impida avanzar?, teniendo en cuenta la naturaleza del proyecto y la cantidad de personas que participan en la creación se definió que es suficiente con reunirse 2 veces por semana.

Cuando terminen las dos semanas, se harán las siguientes actividades:

Sprint Review: Se hará revisión de lo desarrollado, se comprobará su funcionalidad e integralidad con lo que ya fue ejecutado previamente, es decir, anteriores desarrollos, bases de datos y demás implicaciones, verificando que si se cumplió con lo requerido.

GitHub: Para el código, se usará esta plataforma, ayudando a llevar control de todos los cambios que haga y a tener una copia de seguridad segura de todo el proyecto.

Despliegue: Posterior a que los desarrollos realizados durante cada sprint sean probados y aprobados se pasara a desplegar en el ambiente de producción, permitiendo el uso en el prototipo.

En resumen, esta forma de trabajar me permitirá desarrollar la aplicación de una manera ordenada y, sobre todo, inteligente. Al construirla poco a poco y revisando constantemente, me aseguro de que el resultado final sea una aplicación de alta calidad que realmente cumpla con el objetivo de ayudar a los familiares de personas con Alzheimer.

Herramientas de desarrollo necesarias

Hardware				
Recurso	Descripción	Cantidad	Obtenido	
			Si	No
Smartphones/Tabletas (Android 8.0+)	Para pruebas y uso de la aplicación.	2	X	
Módulo GPS externo con conectividad Bluetooth	Dispositivo para mejorar la precisión de la ubicación.	1	X	
Computadoras para desarrollo	Con entornos adecuados para programación en Flutter	2	X	
Software				
Flutter	Framework para el desarrollo de la aplicación móvil.	2	X	

Dart	Lenguaje de programación para Flutter.	2	X	
Firestore	Para almacenamiento de datos, autenticación y notificaciones en tiempo real.	1	X	
Figma	Herramientas de diseño UI/UX	1	X	
GitHub	Para control de versiones y colaboración en el desarrollo.	2	X	
IDE para desarrollo (VS Code/Android Studio/Arduino IDE)	Para programar y probar la aplicación	2	X	

Tabla 15: Herramientas necesarias para el desarrollo

Costos del Proyecto

Hardware

Recursos	Cantidad	Costo Unitario	Costo total
ASUS Vivobook Go 15 OLED AMD Ryzen 5 7520U RAM 16 GB 512 GB SSD	2	\$2.299.954	\$4.599.908
Sistema Operativo Windows Generación del procesador Serie 7000			

<p>Pantalla 15.6 Pulgadas</p>  <p>16" WUXGA RAM 16GB 512GB SSD PCIE ASUS RYZEN 9 5900HS MILITARY GRADE MIL-STD-883C</p>			
<p>Modulo GPS Neo-8m Incluye conexión Wifi y Bluetooth</p> 	1	\$ 250.000	\$ 250.000
<p>Celular HONOR X5b Plus 256GB Negro 256 GB Rom / 8 GB Ram Cámara de 50 MP, Pantalla nítida de 6,56 pulgadas</p> 	2	\$399.000	\$698.000



<p align="center">Servidor Dell T150</p> <p>Procesador Intel Xeon E-2336G, 16GB de memoria RAM Almacenamiento de 4TB</p> 	1	\$7.499.976	\$7.499.976
Subtotal	\$ 13.047.884		

Tabla 16: Costos del proyecto (Hardware)

Software

Recursos	Cantidad	Costo Unitario	Costo total
<p>Licencia Vitalicia de Windows Server 2022 Standard</p> <p>Licencia Vitalicia de Windows Server 2022 Standard, características necesarias para una gestión integral de servidores.</p> 	1	\$5.599.921	\$5.599.921



<p>Seguridad Total para tu Empresa con Kaspersky Small Office 8</p> <p>Kaspersky Small Office Security 8 protege computadoras, servidores y dispositivos móviles contra virus, ransomware y phishing</p> 	1	\$249.410	\$249.410
<p>Edición Estándar de SQL Server 2022</p> <p>capacidad de consultar cualquier base de datos como Oracle, MongoDB, Cassandra y Hadoop de manera altamente segura y eficiente.</p> 	1	\$1.188.600	\$1.188.600
Subtotal	\$ 7.037.931		

Tabla 17: Costos del proyecto (Software)

Talento Humano

Cargo	Cantidad	Detalle	Costo estimado
Desarrolladores	2	2 x 160h x \$35.000/h	\$11.200.000
Diseñador UX/UI	1	80h x \$38.000/h	\$3.040.000

Tester / QA	1	60h x \$30.000/h	\$1.800.000
Subtotal	\$16.040.000		

Tabla 18: Costos del proyecto (Talento humano)

Costo estimado del desarrollo

Descripción	Valor
Hardware	\$13.047.884
Software	\$7.037.931
Talento humano	\$16.040.000
Total	\$36.125.815

Tabla 19: Costos estimados del proyecto

Historias de usuario

Numero de historia	Requisito funcional relacionado	Descripción
1	RF-001	Se requiere crear el usuario, registrando toda la información de necesaria para poder monitorearlo.
2	RF-002	Se requiere contar con la opción de consultar el usuario y obtener su información, opción que siempre debería estar disponible.
3	RF-003	Se requiere poder modificar la información ya registrada de un usuario, para actualizarla cuando sea necesario.
4	RF-004	Se requiere poder inhabilitar a un usuario del sistema.

5	RF-006	Se requiere que el sistema permita la modificación la periodicidad de la generación de preguntas de verificación de estado hacia el paciente.
6	RF-007	Se requiere que el sistema tenga la opción de inhabilitar las preguntas de verificación que se le hacen al paciente
7	RF-008	El sistema debe permitir consultar el historial de ubicaciones que han sido enviadas a los contactos de emergencia.

Tabla 20: Historias de usuario

Conclusión para el objetivo específico número 1

La fase de análisis permitió establecer una base sólida para el diseño del aplicativo, a través del levantamiento de información mediante encuestas e información recolectada previamente en el marco teórico, se identificaron las necesidades más relevantes, destacando la facilidad de uso, la implementación de alertas en tiempo real y la precisión en la localización como características prioritarias. También se evidenció la importancia de un diseño accesible y simple, que minimice las barreras tecnológicas. Como resultado, se definieron los requisitos funcionales y no funcionales que guiarán la construcción del sistema, asegurando que responda a las expectativas y necesidades identificadas en esta etapa.

Actividades para resolver el objetivo específico número 2

Para desarrollar el objetivo específico número 2: “Diseñar un prototipo funcional que incluya herramientas de seguimiento, envío de alertas y notificaciones automáticas, donde el manejo de la interfaz no requiere conocimientos técnicos por parte de los distintos actores”, se tomó en consideración la relevancia de la información recopilada en la fase de análisis para establecer el diseño de la aplicación dando continuidad a la creación del producto siguiendo el ciclo de vida del software.

Diccionario de datos

Tabla: Perfil							
Descripción:	En esta tabla se guardarán los perfiles.						
Campo	Tipo de Dato	Tamaño	Nulo	Auto incrementable	Tipo de Campo	Tabla Relacionada	Descripción
Id_perfil	Int	4	No	Si	Clave Primaria		En este campo se guardará el código de cada perfil.
Nombre	Varchar	50	No				Aquí se guarda el nombre del perfil.

Tabla 21: Diccionario de datos (tabla Perfil)

Tabla: Usuario							
Descripción:	En esta tabla se guardará toda la información de los usuarios.						
Campo	Tipo de Dato	Tamaño	Nulo	Auto incrementable	Tipo de Campo	Tabla Relacionada	Descripción
Id_usuario	Int	4	No		Clave Primaria		En este campo se guardará el código de cada usuario.

Nombre_usu	Varchar	50	No				Aquí se guarda el nombre del Paciente.
E_mail	Varchar	50	No				Aquí guardamos el correo del usuario.
Contraseña	Int	10	No				Aquí se guardará la contraseña del usuario.
Id_perfil	Int	4	No	Si	Clave Foránea	Perfil	En este campo se guardará el código de cada perfil.

Tabla 22: Diccionario de datos (tabla Usuario)

Tabla: Paciente							
Descripción:	En esta tabla se guardará toda la información de los pacientes.						
Campo	Tipo de Dato	Tamaño	Nulo	Auto incrementable	Tipo de Campo	Tabla Relacionada	Descripción
Id_paciente	Int	4	No		Clave Primaria		En este campo se guardará el código de cada paciente.
Direccion	Varchar	50	No				Aquí guardamos la dirección del Paciente.
Fecha_nacimiento	Date	10	No				Aquí guardamos la fecha de nacimiento del Paciente.

Id_usuario	Int	4	No		Clave Foránea	Usuario	En este campo se guardará el código de cada usuario.
Id_cuidador	Int	4	No		Clave Foránea	Cuidador	En este campo se guardará el código de cada cuidador.

Tabla 23: Diccionario de datos (tabla Paciente)

Tabla: Cuidador							
Descripción:	En esta tabla se guardará toda la información de los cuidadores.						
Campo	Tipo de Dato	Tamaño	Nulo	Auto incrementable	Tipo de Campo	Tabla Relacionada	Descripción
Id_cuidador	Int	4	No		Clave Primaria		En este campo se guardará el código de cada cuidador.
Descripcion	Varchar	50	No				Aquí se guarda alguna descripción del cuidador.
Id_usuario	Int	4	No		Clave Foránea	Usuario	En este campo se guardará el código de cada usuario.

Tabla 24: Diccionario de datos (tabla Cuidador)

Tabla: ContactoEmergencia							
Descripción:	En esta tabla se guardará toda la información de los contactos de emergencia de los Pacientes.						
Campo	Tipo de Dato	Tamaño	Nulo	Auto incrementable	Tipo de Campo	Tabla Relacionada	Descripción
Id_contacto	Int	4	No		Clave Primaria		En este campo se guardará el

							código de cada contacto de emergencia del Paciente.
Nombre	Varchar	50	No				Aquí se guarda el nombre del contacto.
Telefono	Int	10	No				Aquí guardamos el teléfono del Paciente.
Parentesco	Varchar	40	No				En este campo se guarda el parentesco del contacto con el paciente.
esPrincipal	Bool		No				Este campo guardara si es un contacto principal de emergencia.
Id_paciente	Int	4	No		Calve Foránea	Paciente	En este campo se guardará el código de cada paciente.

Tabla 25: Diccionario de datos (tabla contacto emergencia)

Tabla: ConfigPreguntaVerificacion							
Descripción:	En esta tabla se guardará toda la información de las alertas personalizadas.						
Campo	Tipo de Dato	Tamaño	Nulo	Auto incrementable	Tipo de Campo	Tabla Relacionada	Descripción
Id_config	Int	4	No		Clave Primaria		En este campo se guardará el

							código de cada configuración.
Intervalo_alerta	Int	6	No				Aquí se guarda el tiempo que transcurre para cada alerta.
Intervalo_secundario	Int	6	No				Aquí se guarda el tiempo que transcurre para la segunda alerta en caso de no respuesta de la primera.
Id_paciente	Int	4	No		Clave Foránea	Paciente	En este campo se guardará el código de cada paciente.

Tabla 26: Diccionario de datos (tabla Configuración pregunta verificación)

Tabla: Alerta							
Descripción:	En esta tabla se guardará toda la información de los cuidadores.						
Campo	Tipo de Dato	Tamaño	Nulo	Auto incrementable	Tipo de Campo	Tabla Relacionada	Descripción
Id_alerta	Int	4	No		Clave Primaria		En este campo se guardará el código de cada alerta.
Fecha	Date		No				Aquí se guarda la fecha de la alerta
Hora	Time		No				En este campo se guardará la hora de la alerta generada.

estaActivo	Bool		No				Este campo guarda si la alerta se encuentra activa
Id_config	Int	4	No		Clave Foránea	configPreguntaVerificacion	En este campo se guardará el código de configuración de alerta.
Id_contactoEmergencia	Int	4	No		Clave Foránea	AlertaContacto	En este campo se guardará el id de la tabla intermedia entre alerta y contacto de emergencia.

Tabla 27: Diccionario de datos (tabla Alerta)

Tabla: Ubicación							
Descripción:	En esta tabla se guardará toda la información de los cuidadores.						
Campo	Tipo de Dato	Tamaño	Nulo	Auto incrementable	Tipo de Campo	Tabla Relacionada	Descripción
Id_ubicacion	Int	4	No		Clave Primaria		En este campo se guardará el código de cada ubicación generada.
Latitud	Decimal	9, 6	No				En este campo se guardará la latitud de la

							ubicación generada.
Longitud	Decimal	9, 6	No				En este campo se guardará la longitud de la ubicación generada.
Fecha	Date		No				En este campo se guardará la fecha de la alerta generada.
Hora	Time		No				En este campo se guardará la hora de la alerta generada
Id_alerta	Int	4	No		Clave Foránea	Alerta	En este campo se guardará el código de cada alerta.

Tabla 28: Diccionario de datos (tabla Ubicación)

Tabla: AlertaContacto							
Descripción:	En esta tabla se guardará toda la información de las alertas y los contactos de emergencia asociadas.						
Campo	Tipo de Dato	Tamaño	Nulo	Auto incrementable	Tipo de Campo	Tabla Relacionada	Descripción
Id_alertaCont acto	Int	4	No		Clave Primaria		En este campo se guardará el id de alertaContacto
Id_alerta	Int	4	No		Clave Foránea	Alerta	Este campo guardara el id de cada alerta generado

Id_contactoEmergencia	Int	4	No		Clave Foránea	ContactoEmergencia	En este campo se guardará el id de los contactos de emergencia a quienes les llegará la alerta
-----------------------	-----	---	----	--	---------------	--------------------	--

Tabla 29: Diccionario de datos (tabla Alerta contacto)

Prototipo paso a paso

Bienvenida



Ilustración 21: Prototipo (Bienvenida)

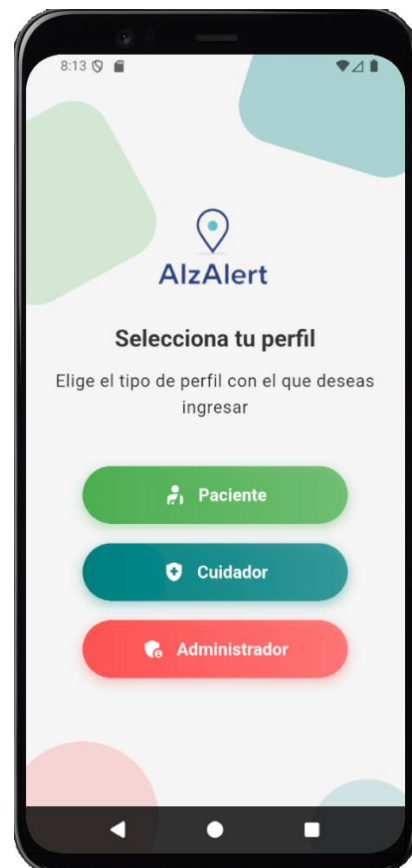


Ilustración 22: Prototipo (seleccionar perfil)

Crear nuevo usuario

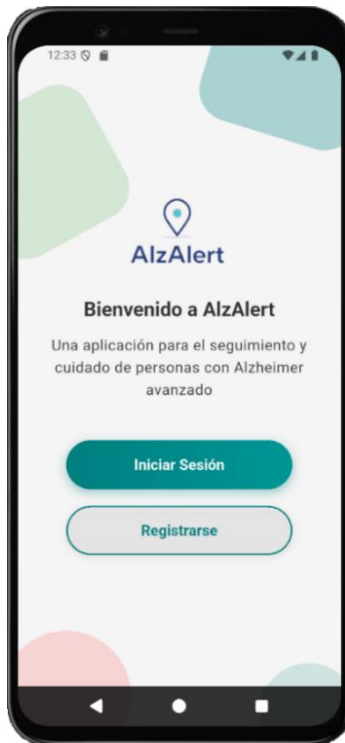


Ilustración 23: Prototipo (Pantalla inicial)

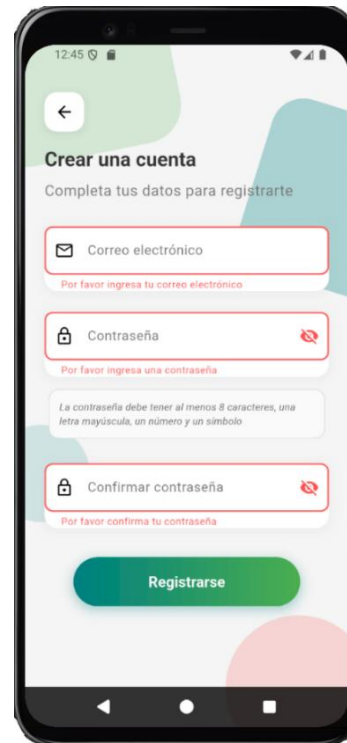


Ilustración 24: Prototipo (Crear cuenta validaciones)

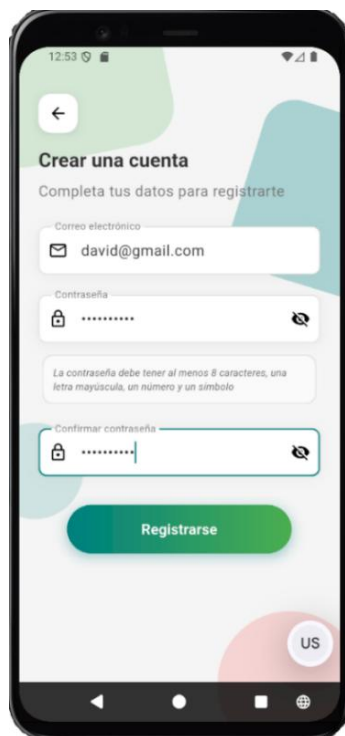


Ilustración 25: Prototipo (Crear cuenta con datos)



Ilustración 26: Prototipo (Información personal validaciones)

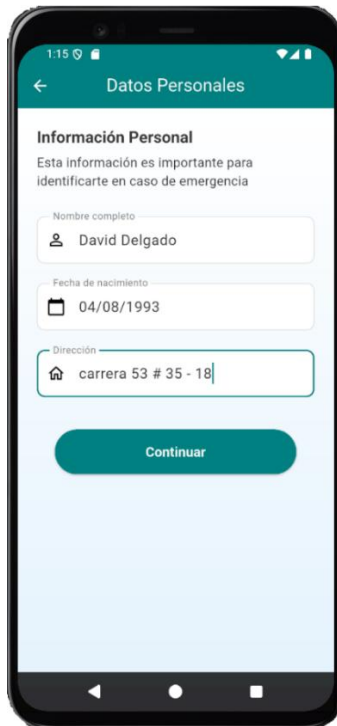


Ilustración 27: Prototipo (Información personal con datos)



Ilustración 28: Prototipo (Agregar contacto validaciones)



Ilustración 29: Prototipo (Agregar contacto con datos)

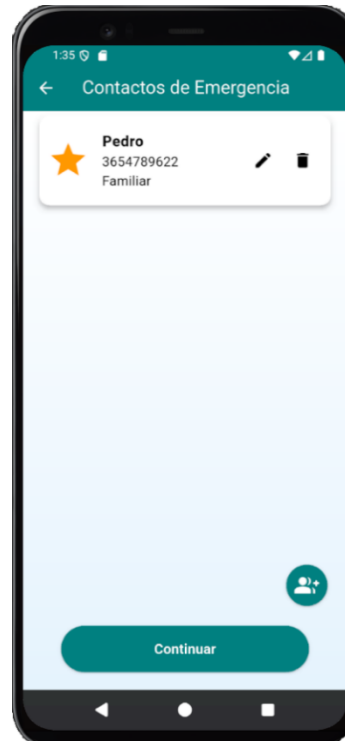


Ilustración 30: Prototipo (Contacto agregado)

Iniciar sesión

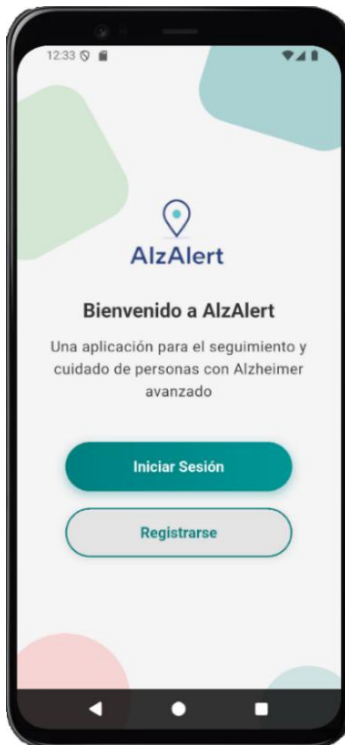


Ilustración 31: Prototipo (Iniciar sesión)

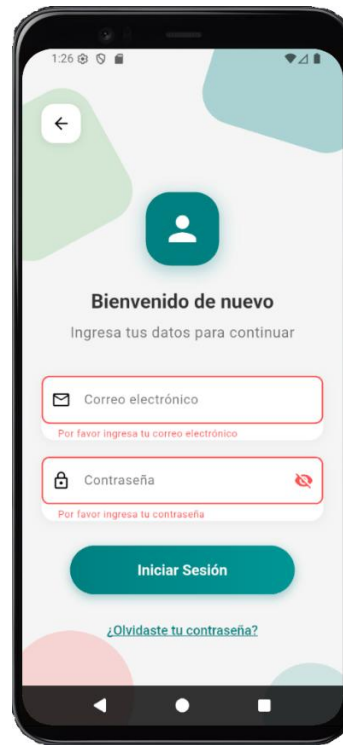


Ilustración 32: Prototipo (Iniciar sesión validaciones)

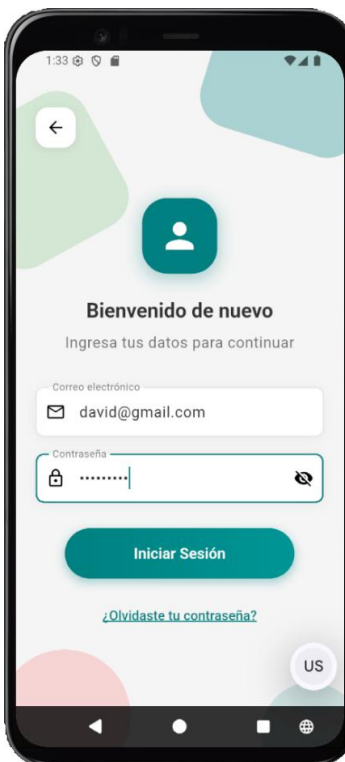


Ilustración 33: Prototipo (iniciar sesión con datos)



Ilustración 34: Prototipo (Inicio)

Consultar, Editar, Eliminar usuario



Ilustración 35: Prototipo (Consultar perfil)

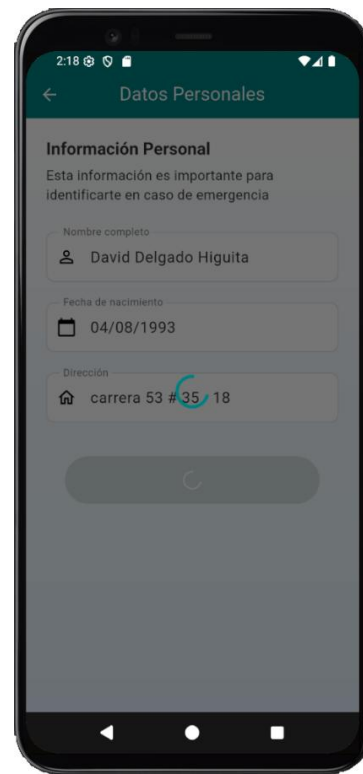


Ilustración 36: Prototipo (Editar info. Personal)



Ilustración 37: Prototipo (Confirmación eliminar perfil)

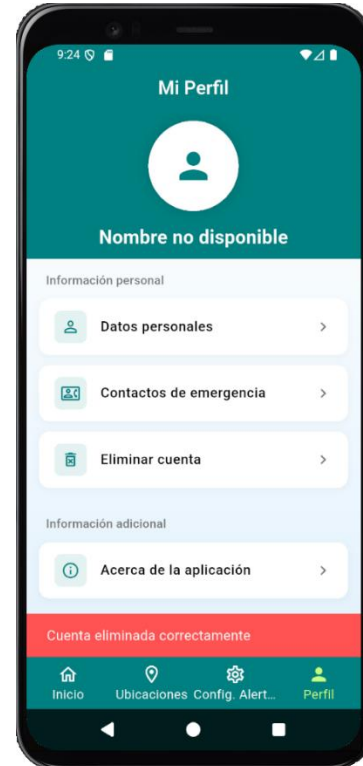


Ilustración 38: Prototipo (Perfil eliminado)

Consultar, Editar, Eliminar contacto de emergencia

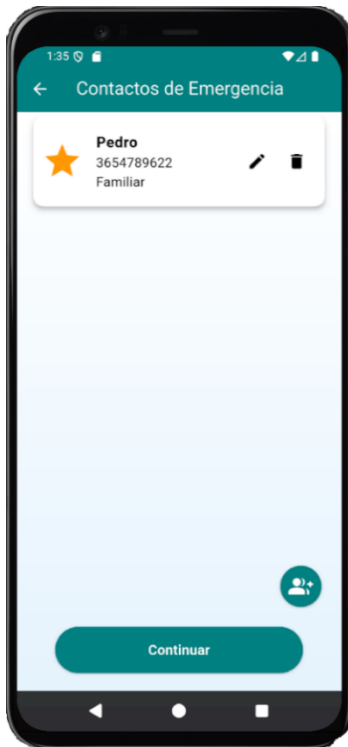


Ilustración 39: Prototipo (Consultar contacto emergencia)



Ilustración 40: Prototipo (Editar contacto emergencia)



Ilustración 41: Prototipo (Eliminar contacto emergencia)

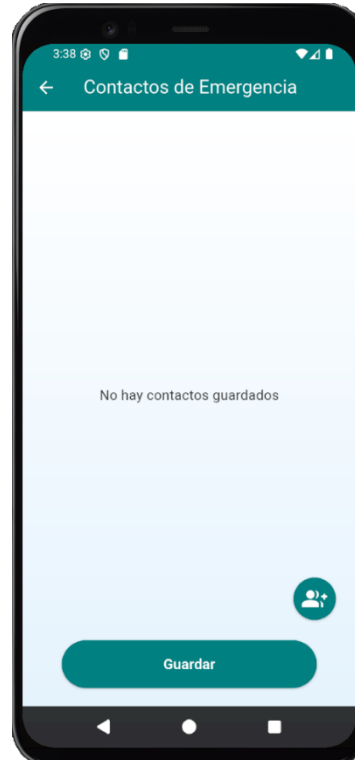


Ilustración 42: Prototipo (Contacto emergencia eliminado)

Configuración de alertas



Ilustración 43: Prototipo (Inicio config. alerta pág. 1)



Ilustración 44: Prototipo (Inicio config. alerta Pág. 2)

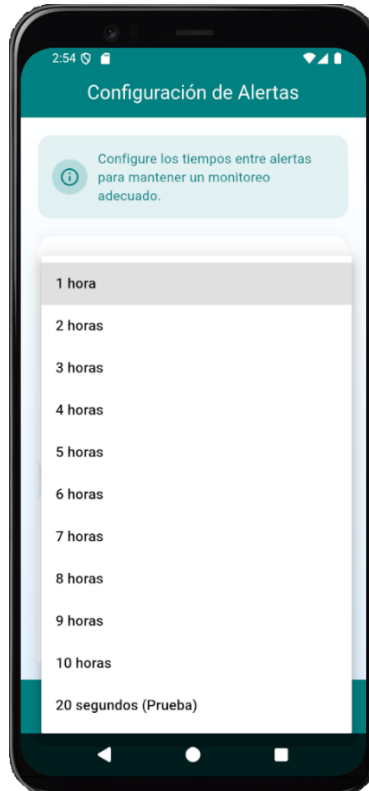


Ilustración 45: Prototipo (Config. alerta intervalo 1)

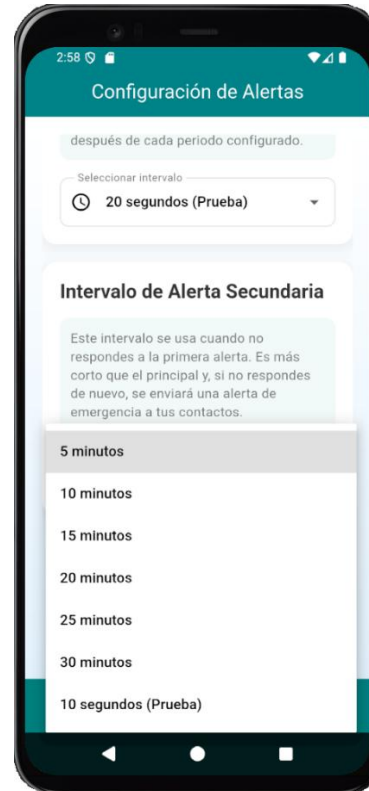


Ilustración 46: Prototipo (Config. alerta intervalo 2)



Ilustración 47: Prototipo (Config. alerta guardar)



Ilustración 48: Prototipo (Config. alerta inhabilitar)

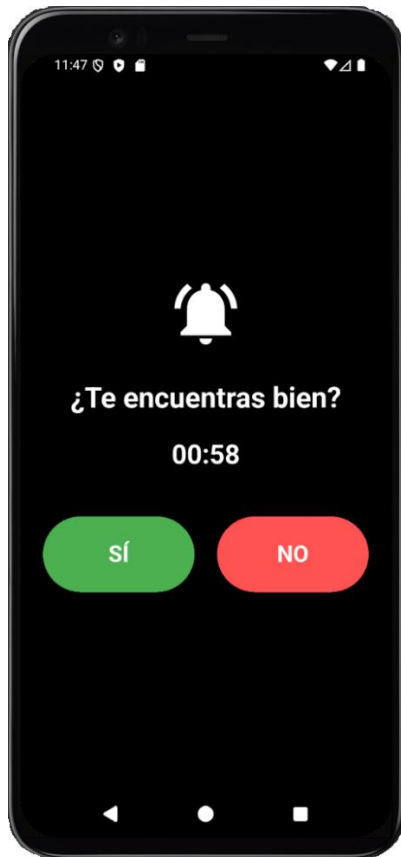


Ilustración 49: Prototipo (Primera alerta verificación)

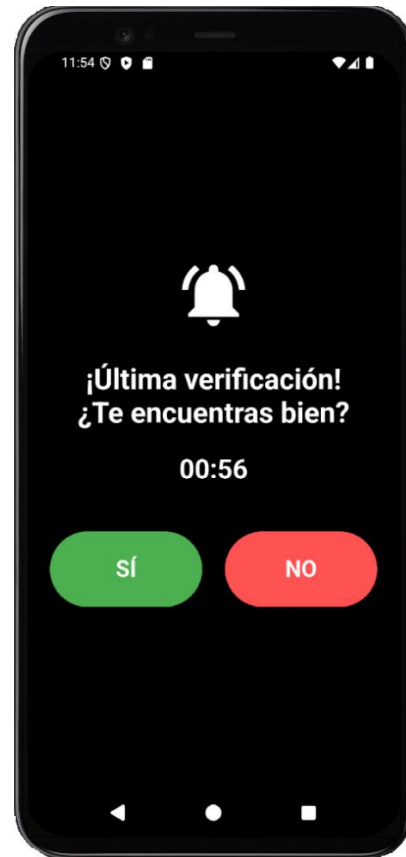


Ilustración 50: Prototipo (Segunda alerta verificación)



Ilustración 51: Prototipo (Botón confirmación envío de alerta)

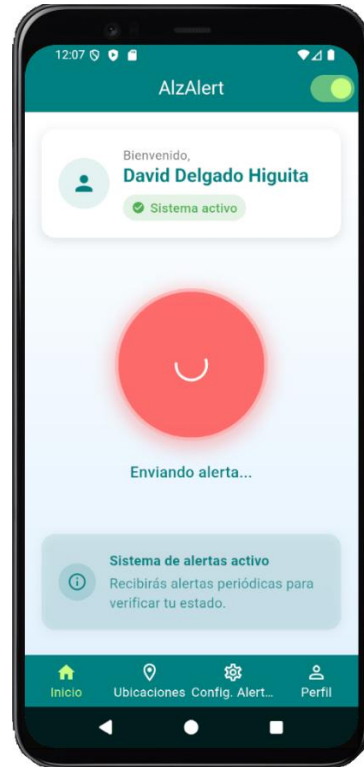


Ilustración 52: Prototipo (Alerta enviada)

Historial de ubicaciones

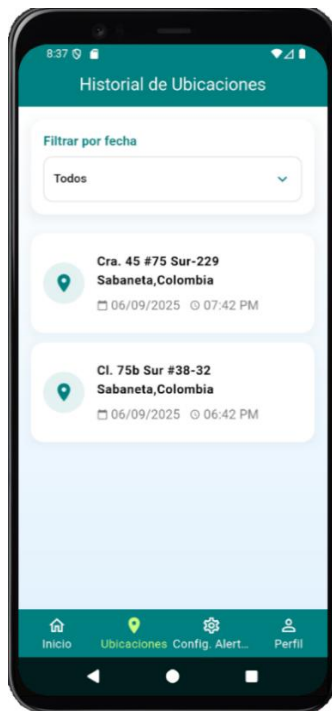


Ilustración 53: prototipo (Historia de ubicaciones)

Integración de componentes: aplicación, ESP 32 y modulo GPS NEO 8M

A continuación, se presenta un diagrama de actividades, esta muestra cómo funciona la integración entre el sistema y los módulos ES32 Y GPS NEO 8M, especialmente al momento de realizar las preguntas de confirmación de estado al usuario, posterior en la generación, envío y guardado de la alerta a los contactos de emergencia del paciente.

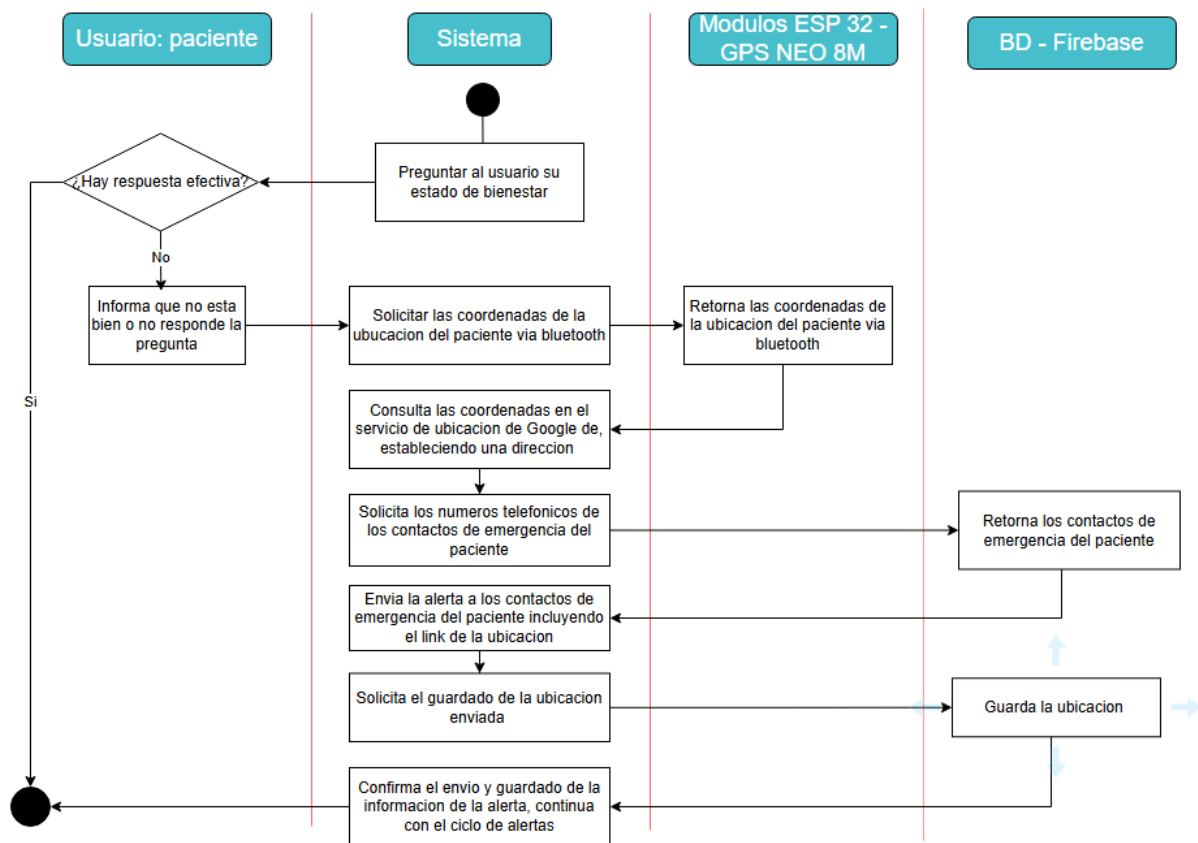


Ilustración 54: Modelo de actividades

Conclusiones del prototipo

Las imágenes del prototipo tienen un papel muy importante en el desarrollo de la aplicación, ya que permiten planificar cómo interactuarán los usuarios con el sistema antes de llevar a cabo la implementación final. Esta actividad no solo verifica la usabilidad y la claridad del flujo de la aplicación, sino que también conecta los requisitos funcionales con el diseño de la base de datos, tales como:

- Las pantallas de registro y login se relacionan directamente con las tablas Usuario, Perfil y Paciente/Cuidador.
- Las pantallas de configuración de alertas se vinculan con la tabla ConfigPreguntaVerificacion y con las entidades Alerta y Ubicación.
- Las pantallas de contactos de emergencia se asocian con la tabla ContactoEmergencia y con la tabla intermedia AlertaContacto.

Para finalizar el uso de prototipos permite comprobar que el flujo definido sea claro para el usuario y que las acciones tengan claridad en la estructura de datos diseñada, asegurando coherencia entre interfaz, lógica y modelo relacional.

Actividades para resolver el objetivo específico número 3

Para realizar el objetivo específico número 3: “Construir la aplicación de acuerdo con metodologías de desarrollo ágiles, incluyendo tecnologías móviles actuales, asegurando la operatividad entre el dispositivo y hardware añadidos”, se tomó en consideración la relevancia de la información recopilada en la fase de análisis y el diseño propuesto para así iniciar el desarrollo de la aplicación siguiendo el ciclo de vida del software.

La aplicación se construyó en el lenguaje de programación Dart usando el framework Flutter, a continuación, se mostrará parte del flujo principal de la aplicación a nivel de código, incluyendo desde donde al paciente se le hacen las preguntas de verificación de estado hasta el envío de alerta a los contactos de emergencia.

Verificación de estado

En esta parte se define la creación de una interfaz de la aplicación, en la cual se le hace la pregunta ¿Te encuentras bien?, donde el usuario va a tener dos opciones de respuesta, si o no, a continuación, se muestran detalles del código fuente.

1. `@override Widget build (BuildContext context)`: Este método es el encargado de construir la interfaz gráfica de la pantalla. Es como el plano que describe cómo se verá la pantalla en la aplicación.
2. `WillPopScope`: Este componente evita que el usuario regrese a la pantalla anterior al presionar el botón "atrás" del dispositivo, asegura que no se pueda salir de esta pantalla accidentalmente si da una respuesta.
3. `Scaffold`: Es la estructura básica de la pantalla. Define el diseño general, como el fondo y el contenido. Aquí, se establece un fondo negro semitransparente.
4. `Text`: Muestra un mensaje en la pantalla haciendo la pregunta ¿Te encuentras bien?

```
@override
Widget build(BuildContext context) {
  return WillPopScope(
    //Configuración de mensaje
    onWillPop: () async => false,
    child: Scaffold(
      backgroundColor: Colors.black.withOpacity(
        0.9,
      ),
      body: SafeArea(
        child: Center(
          child: Padding(
            padding: const EdgeInsets.all(24.0),
            child: Column(
              mainAxisAlignment:
                MainAxisAlignment
                  .center,
              const Icon(
                Icons.notifications_active,
                color:
                  AppTheme
                    .primaryWhite,
                size: 80,
              ),
              const SizedBox(height: 32),
              // Texto mostrado en el mensaje de verificación
              Text(
                widget.isRetry
                  ? '¡Última verificación!\n¿Te encuentras bien?'
                  : '¿Te encuentras bien?',
                style: Theme.of(context).textTheme.displayMedium?.copyWith(
                  color: AppTheme.primaryWhite,
                  fontWeight: FontWeight.bold,
                ),
                textAlign: TextAlign.center,
              ),
            ),
          ),
        ),
      ),
    ),
  ),
);
```

Ilustración 55: Desarrollo (Preguntas de verificación)

Envió de SMS

Esta parte del código define una función que envía un SMS de emergencia, pero primero verifica si el sistema de alertas está activo.

1. **Future<void> _sendEmergencySMS() async**: Esta es una función llamada `_sendEmergencySMS` que se encarga de enviar un mensaje de texto (SMS) en caso de emergencia.
2. **Provider.of<AlertSystemProvider>(context, listen: false)**: Usa el paquete **Provider** para acceder al estado del sistema de alertas (`AlertSystemProvider`). Este estado indica si el sistema de alertas está activado o no.
3. **if (!alertSystemProvider.isAlertSystemActive)**: Comprueba si el sistema de alertas está desactivado. La condición `!isAlertSystemActive` significa "si el sistema no está activo".
4. **ScaffoldMessenger.of(context).showSnackBar(...)**: Si el sistema de alertas está desactivado, se muestra un mensaje emergente en la pantalla. Este mensaje dice: "Sistema de alertas desactivado. No se puede enviar la alerta." y tiene un fondo rojo (`AppTheme.secondaryRed`) para indicar un problema.
5. **return;**: Si el sistema está desactivado, la función se detiene aquí y no envía el SMS.

```
Future<void> _sendEmergencySMS() async {  
  // Verificar si el sistema está activo antes de continuar  
  final alertSystemProvider = Provider.of<AlertSystemProvider>(context, listen: false);  
  if (!alertSystemProvider.isAlertSystemActive) {  
    ScaffoldMessenger.of(context).showSnackBar(  
      const SnackBar(  
        content: Text('Sistema de alertas desactivado. No se puede enviar la alerta.'),  
        backgroundColor: AppTheme.secondaryRed,  
      ),  
    );  
    return;  
  }  
}
```

Ilustración 56: Desarrollo (Envió de SMS)

6. Mensaje de alerta: Se combina un mensaje de alerta con la ubicación en un enlace de Google Maps, se toma del dispositivo ESP32 el cual está conectado al módulo de GPS.

7. Listado de contactos List <String> números: Se toman en un listado los números de los contactos de emergencia del paciente, que serán usados en el envío del SMS.

```
// Mensaje de emergencia
String mensaje = 'EMERGENCIA! ${userName.isNotEmpty ? userName : 'Usuario'} necesita ayuda urgente';

// Agregar ubicación si disponible, en formato simple
if (_currentLocationString.isNotEmpty && !_currentLocationString.startsWith('Error')) {
  mensaje += '\nUbicación: https://www.google.com/maps/search/?api=1&query=\$\_currentLocationString';
}

// Se crea la lista vacía para almacenar los números de los contactos de emergencia
List<String> numeros = [];

if (contactos.isNotEmpty) {
  // Se agregan los números de los contactos de emergencia
  numeros = contactos.map((c) => c.phone).toList();
  debugPrint('Sending emergency SMS to configured contacts.');
```

Ilustración 57: Desarrollo (Lista de contactos de emergencia)

Historial de ubicación

En la siguiente parte del código se implementa la lógica para registrar y mostrar las ubicaciones del paciente. Cada vez que se genera una alerta, la posición (latitud, longitud, fecha y hora) queda almacenada.

1. **Función `_LocationHistoryScreenState`**: permite a cuidadores y administradores consultar un historial filtrable.
2. **`DropDownButton<String>`**: Crea una lista de ubicaciones y usa consultas a la base de datos para traer registros por fecha. Se renderiza en la interfaz de Flutter como un `ListView`, lo que facilita la visualización de las ubicaciones guardadas.
3. Conecta directamente con la tabla Ubicación en la base de datos, asegurando trazabilidad de eventos y proporcionando claridad en el monitoreo.

```

class _LocationHistoryScreenState extends State<LocationHistoryScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Historial de Ubicaciones'),
      ),
      body: Consumer<LocationHistoryProvider>(
        builder: (context, locationHistoryProvider, child) {
          if (locationHistoryProvider.isLoading) {
            return const Center(child: CircularProgressIndicator());
          }

          final allLocations = locationHistoryProvider.locationHistory;
          final filteredLocations = _getFilteredLocations(allLocations);
          final uniqueDates = _getUniqueDates(allLocations);

          // Ajusta el valor del filtro si el filtro actual ya no existe en las fechas disponibles
          if (!uniqueDates.contains(_selectedFilter) && _selectedFilter != 'Todos') {
            _selectedFilter = 'Todos';
          }
          if (uniqueDates.length == 1 && uniqueDates.first != 'Todos') {
            _selectedFilter = uniqueDates.first;
          }

          return Column(
            children: [
              Padding(
                padding: const EdgeInsets.all(16.0),
                child: Row(
                  children: [
                    const Text('Filtrar por: '),
                    const SizedBox(width: 8),
                    DropdownButton<String>(
                      value: _selectedFilter,
                      items: uniqueDates.map<DropdownMenuItem<String>>((String value) {
                        return DropdownMenuItem<String>(
                          value: value,
                          child: Text(value),
                        );
                      }).toList(),
                      onChanged: (value) {
                        setState(() {
                          _selectedFilter = value!;
                        });
                      },
                    ),
                  ],
                ),
              ),
            ],
          );
        },
      ),
    );
  }
}

```

Ilustración 58: Desarrollo (Historial de ubicaciones)

Esto permite una lectura más detallada de los eventos generados por el paciente durante una crisis, permitiendo ver la frecuencia de ocurrencia, teniendo como resultado un mejor control de los patrones de riesgo y facilitando la toma de decisiones clínicas y preventivas por parte de cuidadores y familiares.

Conexión bluetooth

Aquí se programa la comunicación con el dispositivo ESP32 que contiene el módulo GPS.

1. **Clase BluetoothService:** establece, mantiene y finaliza la conexión entre el smartphone y el hardware externo. Esta usa funciones de la librería flutter_blue para escanear dispositivos cercanos, emparejarse con el ESP32 y recibir datos. También maneja eventos de conexión/desconexión y errores.
2. **Función _connectToDevice:** garantiza que la aplicación esté conectada con el dispositivo móvil y lo muestre en pantalla, asegurando que reciba datos en tiempo real desde el GPS, lo cual es esencial para las alertas.

```
class BluetoothService {  
  
  Future<void> _scanDevices() async {  
    if (_connectedDevice != null) return;  
  
    try {  
      _updateStatus("Buscando dispositivo...", Colors.blue);  
  
      await fbp.FlutterBluePlus.stopScan();  
  
      final subscription = fbp.FlutterBluePlus.scanResults.listen((results) {  
        for (var result in results) {  
          if (result.device.name == "ESP32_BLUETOOTH" &&  
              _connectedDevice == null) {  
            _connectToDevice(result.device);  
            break;  
          }  
        }  
      });  
  
      await fbp.FlutterBluePlus.startScan(  
        timeout: const Duration(seconds: 10),  
        androidUsesFineLocation: false,  
      );  
  
      await Future.delayed(const Duration(seconds: 12));  
  
      await fbp.FlutterBluePlus.stopScan();  
      await subscription.cancel();  
  
      if (_connectedDevice == null) {  
        _updateStatus("Dispositivo no encontrado", Colors.red);  
      }  
    } catch (e) {  
      _updateStatus("Error en escaneo", Colors.red);  
      debugPrint("Error _scanDevices: $e");  
    }  
  }  
  
  Future<void> _connectToDevice(fbp.BluetoothDevice device) async {  
    try {  
      _updateStatus("Conectando...", Colors.blue);  
  
      await _connectionSubscription?.cancel();  
  
      await device.connect(  
        autoConnect: false,  
        timeout: const Duration(seconds: 15),  
      );  
    }  
  }  
}
```

Ilustración 59: Desarrollo (Conexión a bluetooth)

Recepción de la información

En la parte del código que recibe datos del ESP32 por Bluetooth, la lógica se centra en leer un mensaje en forma de texto, separarlo en piezas y convertirlo en coordenadas GPS que el sistema pueda entender.

1. **Función onDataReceived():** se activa cada vez que llega un dato nuevo por Bluetooth. Piensa en ella como la persona que revisa el buzón de correo: si hay un mensaje, lo abre. El dato llega como una cadena de texto (ejemplo: "6.25184,-75.56359").
2. El código usa operaciones de **parsing** (separar por la coma) para extraer primero la latitud y luego la longitud.
3. Estas coordenadas se convierten en números (double) y se almacenan en la base de datos en la tabla Ubicación, junto con la fecha y hora actual.
4. Finalmente, otra parte del código las envía al historial visual de la app (ListView en Flutter), para que el cuidador pueda ver todos los registros.

```
Future<void> _discoverServices(fbp.BluetoothDevice device) async {
  try {
    List<fbp.BluetoothService> services = await device.discoverServices();
    debugPrint("Servicios descubiertos: ${services.Length}");
    for (var service in services) {
      String serviceUuid = service.uuid.toString().toLowerCase();
      debugPrint("Servicio UUID: $serviceUuid");
      for (var characteristic in service.characteristics) {
        String charUuid = characteristic.uuid.toString().toLowerCase();
        debugPrint(" Característica UUID: $charUuid");
        debugPrint(
          " Propiedades: notify=${characteristic.properties.notify}, write=${characteristic.properties.write}",
        );
      }
      if (serviceUuid == "00001234-0000-1000-8000-00805f9b34fb" ||
        serviceUuid == "1234") {
        debugPrint("Servicio encontrado: $serviceUuid");
        for (var characteristic in service.characteristics) {
          String charUuid = characteristic.uuid.toString().toLowerCase();
          if (charUuid == "00005678-0000-1000-8000-00805f9b34fb" ||
            charUuid == "5678") {
            debugPrint("Característica encontrada: $charUuid");
            _targetCharacteristic = characteristic;
            if (characteristic.properties.notify) {
              await characteristic.setNotifyValue(true);
              characteristic.value.listen((value) {
                if (value.isNotEmpty) {
                  // Recepcion de Los datos correspondientes a La Latitud y Longitud
                  String receivedData = String.fromCharCode(value);
                  debugPrint("Dato recibido: $receivedData");
                }
              });
            }
          }
        }
      }
    }
  }
}
```

Ilustración 60: Desarrollo (Captura de datos desde el bluetooth)

Código de Arduino IDE

El ESP32 es el encargado de recolectar la ubicación real del paciente usando un módulo GPS y enviarla por Bluetooth al celular.

1. **Librería TinyGPS++:** actúa como traductor entre el satélite GPS y el ESP32. Recibe las señales simples del satélite y las convierte en valores de latitud y longitud.
2. **Librería BluetoothSerial:** permite que el ESP32 se comunique con el celular por Bluetooth como si fuera un chat en tiempo real.
3. **Función loop():** es como un vigilante. Dentro de este ciclo, el ESP32 lee constantemente si hay datos nuevos del GPS.
4. **Función gps.encode():** procesa cada fragmento de información que llega del satélite y lo convierte en coordenadas válidas.
5. **Condición if (gps.location.isUpdated()):** se asegura de que los datos sean frescos (no repetidos). Si son nuevos, se guardan en las variables lat y lng.
6. Luego se arma un mensaje de texto (por ejemplo "6.25184,-75.56359") y se envía por Bluetooth con la instrucción SerialBT.println().

```
#include <BluetoothSerial.h>
#include <TinyGPS++.h>
#include <HardwareSerial.h>

// Definimos los pines y la velocidad de comunicación para el módulo GPS.
// EL NEO-M8N suele usar 9600 baudios por defecto.
static const int RXPin = 16, TXPin = 17;
static const uint32_t GPSBaud = 9600;

// Objetos para la comunicación
BluetoothSerial SerialBT;
HardwareSerial mySerial(2); // Usamos el Serial 2 del ESP32
TinyGPSPlus gps;

void setup() {
  // Inicializamos la comunicación con el monitor serial para depuración
  Serial.begin(115200);

  // Inicializamos el módulo Bluetooth con un nombre
  SerialBT.begin("ESP32_GPS_Tracker");
  Serial.println("Bluetooth ESP32 iniciado.");

  // Inicializamos la comunicación con el módulo GPS
  mySerial.begin(GPSBaud, SERIAL_8N1, RXPin, TXPin);
  Serial.println("Serial para GPS iniciado.");
}

void loop() {
  // Mientras haya datos disponibles del GPS, los procesamos.
  while (mySerial.available() > 0) {
    if (gps.encode(mySerial.read())) {
      if (gps.location.isValid()) {
        double lat = gps.location.lat();
        double lon = gps.location.lng();

        // Creamos la cadena de datos que enviaremos por Bluetooth
        String locationData = "LAT:" + String(lat, 6) + ",LON:" + String(lon, 6);

        // Enviamos la cadena a través de Bluetooth
        SerialBT.println(locationData);
        Serial.println("Datos enviados por Bluetooth: " + locationData);
      } else {
        Serial.println("Ubicación no válida.");
      }
    }
  }
}
```

Ilustración 61: Desarrollo (Comunicación del Arduino con el ESP32)

Retos en la Fase de Diseño y Desarrollo

Durante la fase de diseño se identificaron varios factores que pusieron a prueba nuestra capacidad de adaptación y respuesta, algunos de estos fueron:

- ✓ **Integración de hardware y software:** La conexión estable entre el módulo GPS y la aplicación vía Bluetooth presentó inconsistencias iniciales, tales como condiciones climáticas, entornos con baja señal o muchos obstáculos, lo que nos ocasiono pérdidas de datos en la transmisión. Esto lo resolvimos implementando un protocolo de reconexión automática y optimizando los intervalos de lectura.
- ✓ **Usabilidad de la interfaz:** Implementar una aplicación para pacientes con Alzheimer avanzado implicó un reto adicional que fue simplificar al máximo la interacción sin perder funcionalidad. Lo que no llevo a adoptar un enfoque más iterativo en el prototipo, ajustando botones, colores y tamaños de texto tras revisiones sucesivas y teniendo como base historias de usuario y datos recolectados en la encuesta.
- ✓ **Seguridad y privacidad de datos:** La protección de la información sensible fue prioridad. Al inicio nos surgieron dudas sobre el nivel de cifrado más adecuado y seguro, pero finalmente implementamos AES-256 por ser el más robusto utilizando claves de 256 bits, junto con transmisión segura vía TLS/HTTPS siendo este último un protocolo de transferencia de hipertexto para cifrar la conexión entre el navegador y un servidor web, cumpliendo así con estándares reconocidos en protección de datos.
- ✓ **Gestión de incidencias:** Durante el desarrollo nos surgieron errores relacionados con la visualización del historial de ubicaciones, estas no se estaban mostrando de forma correcta. Estas se fueron registrando en un tablero de incidencias en GitHub, lo que permitió asignar a responsables para así dar seguimiento y poder llegar a la solución más adecuada.

- ✓ **Control de versiones:** Para poder mantener el orden y trazabilidad en el código, utilizamos un flujo de trabajo en GitHub con ramas específicas para cada funcionalidad. De esta forma evitamos conflictos entre versiones y se garantizó que cada cambio quedara registrado y se pudiera revisar a detalle el cambio implementado, con la posibilidad de volver atrás en caso de fallos o inconvenientes presentados.

Actividades para resolver el objetivo específico número 4

Para realizar el objetivo específico número 3: “Construir la aplicación de acuerdo con metodologías de desarrollo ágiles, incluyendo tecnologías móviles actuales, asegurando la operatividad entre el dispositivo y hardware añadidos”, se tomó en consideración la relevancia de la información recopilada en la fase de análisis y el diseño propuesto para así iniciar el desarrollo de la aplicación siguiendo el ciclo de vida del software.

La aplicación se construyó en el lenguaje de programación Dart usando el framework Flutter, a continuación, se mostrará parte del flujo principal de la aplicación a nivel de código, incluyendo desde donde al paciente se le hacen las preguntas de verificación de estado hasta el envío de alerta a los contactos de emergencia.

Para realizar el objetivo específico número 4: “Ejecutar pruebas de validación de operabilidad entre el dispositivo y hardware, además de verificar la usabilidad de la aplicación”, se realizaron pruebas desde el registro de un usuario hasta el envío de los mensajes de alerta por medio de SMS. Se elaboro la siguiente matriz donde se podrá visualizar detalladamente los casos probados.

Id requisito	Acción	Descripción
RF-001	Descripción del requisito	El sistema debe permitir crear los usuarios, se solicitará información para el registro de acuerdo con el rol escogido (paciente, cuidador, administrador), utilizando el correo electrónico y una contraseña como credenciales.
	Id caso prueba	001

	Descripción general caso de prueba	<p>Se realiza la prueba creando un usuario tipo paciente nuevo, ingresando un correo, contraseña y diligenciando los campos solicitados en el formulario, ingresando caracteres especiales, correos no válidos, contraseñas diferente al estándar solicitado (como mínimo una letra en mayúscula, números y caracteres)</p>
	Resultado esperado	<p>Se espera que la aplicación logre:</p> <ul style="list-style-type: none"> • Detectar correos y contraseñas no validas • Validar el ingreso de información en los campos obligatorios • El usuario mínimo debe tener un contacto de emergencia para completar el guardado • Guardar el usuario
	Resultado obtenido	<p>La aplicación logro detectar que tanto el correo como la contraseña no son válidos al no cumplir con estándares, posterior se ingresaron datos validos con los cuales si completo el registro</p>



Ilustración 62: Pruebas (Validación crear usuario)



Ilustración 63: Pruebas (Crear usuario)

La aplicación logro detectar cuando en los campos obligatorios no se llenaban



Ilustración 64: Pruebas (Validación campos vacíos 1)



Ilustración 65: Pruebas (Validación campos vacíos 2)

La aplicación logro detectar que al momento de intentar completar el guardado del usuario sin tener un contacto de emergencia no procedió con el guardado, posteriormente se registró un contacto que fue guardado

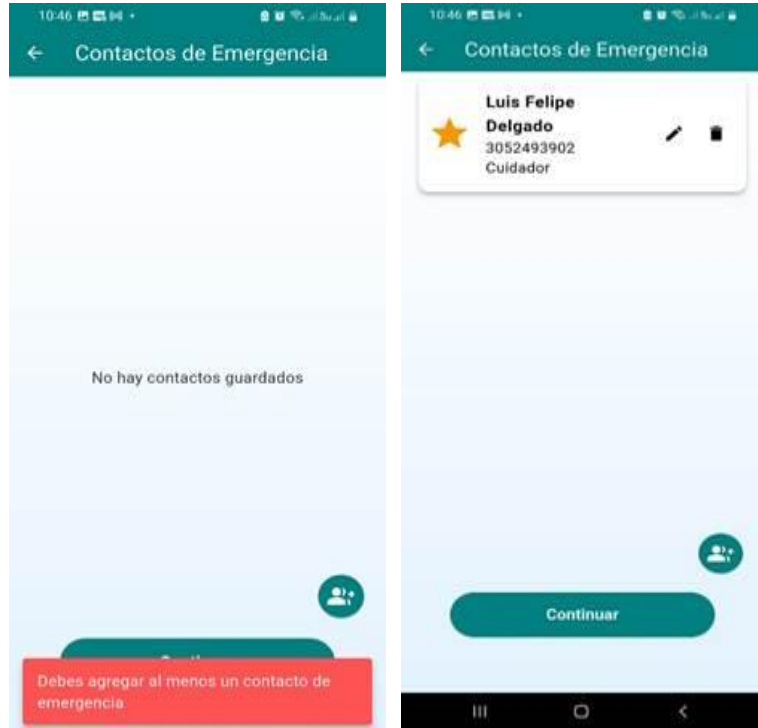


Ilustración 66: Pruebas (Validación agregar contacto)

Ilustración 67: Pruebas (Agregar contacto)

La aplicación logro guardar el usuario exitosamente.

RF-003	Descripción del requisito	El sistema debe permitir modificar la información de los usuarios ya previamente registrados.
	Id caso prueba	002
	Descripción general caso de prueba	Se realiza la prueba intentando modificar los datos personales de un usuario ya registrado en la base de datos.

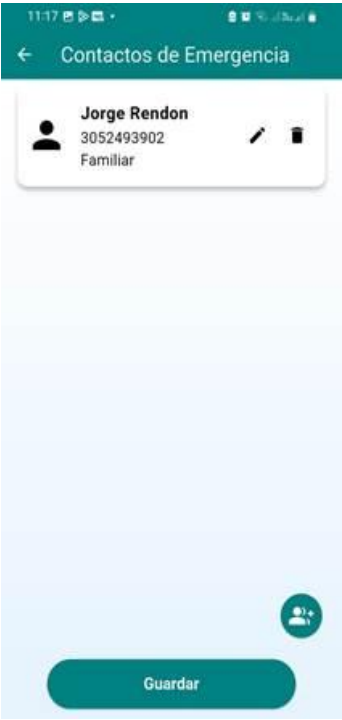

	Resultado esperado	Se espera que la información del usuario como el nombre, fecha de nacimiento, diagnóstico, dirección y contactos de emergencia sea posible modificarla.
	Resultado obtenido	<p>La aplicación logra modificar exitosamente los campos indicados, a continuación, se presentan imágenes, en la parte izquierda se mostrará como estaba previo a la modificación y en la parte derecha encontrará como cambio el dato posterior a la modificación.</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  <p>Ilustración 68: Pruebas (Contacto guardado)</p> </div> <div style="text-align: center;">  <p>Ilustración 69: Pruebas (Contacto modificado)</p> </div> </div>



Ilustración 70: Pruebas (Nombre de usuario guardado)



Ilustración 71: Pruebas (Nombre de usuario modificado)



Ilustración 72: Pruebas (Info. Médica guardada)

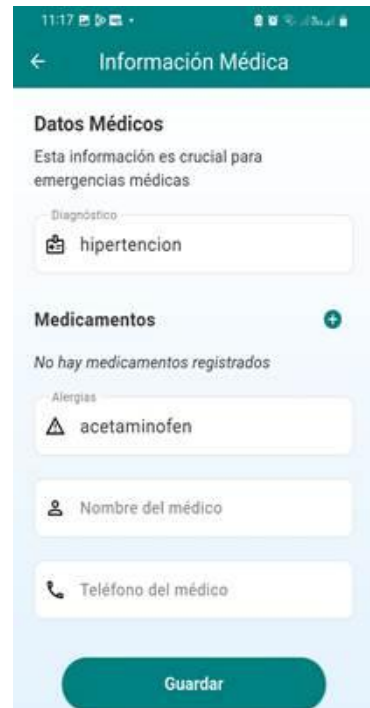


Ilustración 73: Pruebas (Info. Médica modificada)

RF-006	Descripción del requisito	El sistema debe permitir que un usuario con el rol cuidador configure la periodicidad de las preguntas de verificación de estado para cada uno de los usuarios con el rol de pacientes que estén a su cargo, las opciones de respuesta serán si o no.
	Id caso prueba	003
	Descripción general caso de prueba	Se ingresará a la opción funcional configuración de alertas, se modificarán los intervalos de tiempo en que el sistema hará tanto las preguntas de verificación de estado primario y secundaria.
	Resultado esperado	Se espera que las preguntas de validación de estado aparezcan dentro de los límites establecidos.

**Resultado
obtenido**



Ilustración 74: Pruebas (Config. Alertas primer intervalo)

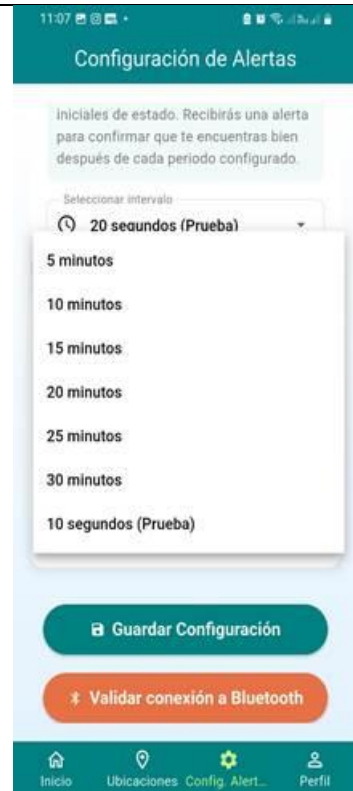







Ilustración 75: Pruebas (Config. Alertas segundo intervalo)

Se programo el intervalo principal en 20 segundos para efectos de la prueba, posteriormente se programó el intervalo secundario en 10 segundos, este último aparece en caso de que el usuario no haya alcanzado a responder la primera pregunta.

		<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Ilustración 76: Pruebas (Primera alerta)</p> </div> <div style="text-align: center;">  <p>Ilustración 77: Pruebas (Segunda alerta)</p> </div> </div> <p>Tras esperar 20 segundos la pregunta de validación de estado apareció de forma correcta, esta no se respondió, con el fin probar la pregunta secundaria, la cual apareció 10 segundos después de cerrarse la pregunta principal, cumpliendo con lo establecido.</p>
RF-007	Descripción del requisito	El sistema debe permitir que se inhabilite las preguntas de verificación en caso de que el paciente tenga supervisión directa y no sea necesario su uso.
	Id caso prueba	004
	Descripción general caso de prueba	El sistema cuenta con un botón que inhabilita las preguntas de verificación de estado, entonces al presionarlo ningún tipo de pregunta debería de aparecer.

	<p>Resultado esperado</p>	<p>Se espera que al inhabilitar el botón las preguntas de verificación de estado no aparezcan, y cuando se habiliten nuevamente las preguntas de verificación de estado aparezcan según los intervalos establecidos.</p>
	<p>Resultado obtenido</p>	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Ilustración 78: Pruebas (Alertas inactivas)</p> </div> <div style="text-align: center;">  <p>Ilustración 79: Pruebas (Alertas activadas)</p> </div> </div> <p>Se logro habilitar e inhabilitar las preguntas de verificación de estado exitosamente, al momento del botón estar habilitado las preguntas de validación de estado aparecían nuevamente según los intervalos predefinidos.</p>

		 <p style="text-align: center;">Ilustración 80: Pruebas (Confirmación de alertas activas)</p>
Prueba de integración componentes	Descripción del requisito	La aplicación móvil se conecta por medio de bluetooth al módulo ESP32, este a su vez se encuentra conectado al módulo GPS NEO 8M, el cual es el encargado de brindar las coordenadas al momento de ser solicitadas desde el sistema, las cuales serán enviadas posteriormente a los contactos de emergencia por medio de un SMS, este contiene un enlace que direcciona a Google Maps logrando así ver la ubicación del paciente en el mapa.
	Id caso prueba	005
	Descripción general caso de prueba	Se hará validación de referente a la integración entre los 3 componentes, aplicación, modulo ESP32 y modulo utilizando el bluetooth, posterior se validará el envío correcto del SMS a los contactos de emergencia del paciente junto con el enlace que mostrara su ubicación.

	<p>Resultado esperado</p>	<p>Se espera que los componentes se integren de forma correcta, que se logre la operabilidad mediante la obtención y envío de la ubicación del paciente.</p>
	<p>Resultado obtenido</p>	<p>En primera instancia se probó la conexión por medio de bluetooth entre la aplicación móvil y el módulo ESP32, la cual resulto exitosa.</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="619 633 997 1384"> </div> <div data-bbox="1013 633 1390 1384"> </div> </div> <p>Ilustración 81: Pruebas (Validación conexión bluetooth) Ilustración 82: Pruebas (Conexión exitosa)</p> <p>Posteriormente se probó el envío de la alerta de emergencia al responder a una de las preguntas de verificación de estado que el paciente no se encontraba bien, en la siguiente imagen se puede observar la salida en consola en la aplicación Arduino IDE, mostrando la conexión a bluetooth con el aplicativo y el envío de las coordenadas, las pruebas resultaron exitosas.</p>

		<div data-bbox="798 194 1198 1057" data-label="Image"> </div> <p data-bbox="726 1093 1270 1122">Ilustración 83: Pruebas (Recepción de alerta exitosa)</p> <div data-bbox="638 1146 1358 1615" data-label="Code-Block"> <pre> -> BLE activo. Nombre: ESP32_BLUETOOTH -> Servicio UUID: 00001234-0000-1000-8000-00805F9B34FB -> Característica UUID: 00005678-0000-1000-8000-00805F9B34FB -> Dispositivo conectado -> Latitud: 6.2784959 -> Longitud: -75.5506522 -> Ubicacion enviada LAT=6.2784959,LON=-75.5506522 -> Dispositivo desconectado -> Dispositivo conectado -> Latitud: 6.2785066 -> Longitud: -75.5506338 -> Ubicacion enviada LAT=6.2785066,LON=-75.5506338 -> Dispositivo desconectado </pre> </div> <p data-bbox="726 1650 1270 1680">Ilustración 84: Pruebas (Ejecución consola Arduino)</p>
--	--	--

Tabla 30: Pruebas de usabilidad y funcionalidad

Conclusión ejecución de pruebas

Esta fase cumple con ser una de las más importantes en el ciclo de vida del software, ya que no se trata solamente de validar el funcionamiento de la aplicación, sino también verificar si la solución tecnológica desarrollada si cumplía con lo diseñado inicialmente.

Las pruebas realizadas fueron exitosas, la aplicación respondió a lo solicitado en cada una de las opciones funcionales, asegurando que al momento de guardar y modificar información se realizara de forma correcta. Desactivar las alertas es fundamental, ya que cuando los cuidadores estén presentes con los pacientes no es necesario que la aplicación haga preguntas de validación de estado, algo útil que también funcione de forma correcta.

Lo más retador fue comprobar la integración entre el software y el hardware, ver cómo la aplicación se conectaba sin problemas al módulo ESP32, cómo este recibía las coordenadas GPS en tiempo real y cómo, finalmente, esos datos se convertían en un mensaje de alerta con un enlace de Google Maps enviado por SMS, las pruebas generaron múltiples situaciones de mejora, tanto en la integración como en el código fuente de la aplicación y en Arduino IDE. Por lo anterior se generaron acciones que permitieron lograr el funcionamiento operativo del sistema.

Conclusión general del desarrollo del proyecto

El principal aporte que nos dejó el proyecto fue que el desarrollo de una aplicación móvil orientada al monitoreo preventivo de personas con Alzheimer avanzado, integrando alertas periódicas, localización GPS y un flujo de comunicación simple entre pacientes y cuidadores. Constituye una herramienta innovadora si bien no reemplaza la supervisión humana, pero sí reduce la incertidumbre y el estrés en quienes tienen a su cargo el cuidado de pacientes.

Aprendimos que la comunicación constante y el uso de herramientas de gestión como control de versiones y flujos de incidencias en GitHub son esenciales para mantener la coherencia del desarrollo. Como equipo comprendimos la importancia de dividir tareas, documentar cada avance y mantener retroalimentación entre roles técnicos y no técnicos para alinear los objetivos del proyecto.

Si tuviéramos que repetir el proceso, se mejoraría la planificación inicial, definiendo con mayor detalle los requisitos antes de empezar el desarrollo. Además, se estableceríamos un cronograma más rígido con hitos de validación temprana, lo que reduciría los ajustes en etapas avanzadas. También contemplaríamos incluir desde el inicio a más usuarios en pruebas piloto para obtener retroalimentación temprana.

El resultado final que obtuvimos cumple de manera coherente con los objetivos iniciales, ofrecemos una solución tecnológica accesible que mejora la seguridad y bienestar de las personas con Alzheimer avanzado. La aplicación permite verificar el estado del paciente, enviar alertas en caso de riesgo y brindar tranquilidad a cuidadores y familiares, lo

que se alinea con la meta principal de aportar a la calidad de vida a través de tecnología preventiva.

Finalmente, el desarrollo de este proyecto nos demuestra que la tecnología puede ser un verdadero aliado en el cuidado de personas con Alzheimer avanzado y de cualquier otro tipo de enfermedad. Más allá del código, diagramas o bases de datos y de la arquitectura que implica el desarrollo del software, lo que se obtuvo fue una herramienta que busca dar tranquilidad a las familias, cuidadores y seguridad a los pacientes, apoyando en momentos donde la supervisión humana muchas veces no es suficiente. Cada parte desde el análisis de actores y requisitos, pasando por el diseño de la aplicación y su base de datos, hasta la implementación de módulos de hardware y software nos permitió comprobar que la integración entre hardware y una aplicación accesible puede traducirse en un sistema confiable, seguro y fácil de usar.

Esto refleja cómo se logró unir una idea y práctica en un mismo camino, priorizando siempre la usabilidad, la protección de datos y la claridad del flujo de información. Este proyecto no solo ofrece una solución tecnológica, sino también un aporte a nuestra sociedad, porque responde a una necesidad real que es acompañar y proteger a quienes viven con una enfermedad que puede ser muy grave como lo es el Alzheimer, mientras se disminuye la carga emocional de quienes los cuidan.

Referencias

1. Organización Mundial de la Salud. (2021). *Demencia*. <https://www.who.int/es/news-room/fact-sheets/detail/dementia>
2. Organización Panamericana de la Salud. (2023). *Demencia*. <https://www.paho.org/es/temas/demencia>
3. Universidad de los Andes. (2024). *Demencia: Un problema de salud pública invisible en Colombia*. Gobierno Uniandes. <https://gobierno.uniandes.edu.co/demencia-un-problema-de-salud-publica-invisible-en-colombia/>
4. Dificultades psicológicas, sociales y económicas de pacientes con enfermedad de Alzheimer (EA), en una población del sur de Colombia. (2024). *Revista de Salud Pública (Córdoba)*, 30(2), Article 42613. <https://doi.org/10.31052/1853.1180.v30.n2.42613>
5. Toro Paca, M. J., Parra Parra, D. R., Pacheco Mena, N. V., & Alvarado Galarza, A. G. (2022). Enfermedad de Alzheimer. *RECIMUNDO*, 6(4), 68–76. <https://doi.org/10.26820/recimundo.6.4.68>
6. García, J. (s. f.). *Tecnologías emergentes para la telemedicina y la atención médica remota: Una revisión de la literatura* [Manuscrito no publicado]. ResearchGate. https://www.researchgate.net/publication/382555055_Tecnologias_emergentes_para_la_telemedicina_y_la_atencion_medica_remota_Una_revision_de_la_literatura_Emerging_technologies_for_telemedicine_and_remote_healthcare_A_literature_review
7. Alzheimer's Association. (s. f.). *Alzheimer y demencia*. <https://www.alz.org/es-mx/alzheimer-demencia>
8. National Institute on Aging. (s. f.). *Enfermedad de Alzheimer*. U.S. Department of Health and Human Services. <https://www.nia.nih.gov/espanol/enfermedad-alzheimer>

9. Calle Bernal, J. J., & Lopera, F. (1998). Alzheimer familiar por mutación paisa: Revisión de las historias clínicas del siglo XX en el Hospital Mental de Antioquia. *Acta Neurológica Colombiana*, 14(3), 148–156.
10. Mayo Clinic. (s. f.). *Alzheimer's disease: Symptoms and causes*.
<https://www.mayoclinic.org/diseases-conditions/alzheimers-disease/symptoms-causes/syc-20350447>
11. Rentería, R., Ortiz, C., & Gaxiola, O. (2024). Análisis comparativo de tecnologías Bluetooth, Wi-Fi, Zigbee y LoRa con un enfoque en la implementación de una red de malla. *Revista Electro*, 46, 148–153.
https://itchihuahua.mx/revista_electro/2024/A26_148-153.pdf
12. Narváez, N. (2024). Comparación de metodologías ágiles para el desarrollo de software. *Journal Scientific*, 8, 5052–5074.
<https://www.investigarmqr.com/ojs/index.php/mqr/article/view/1159/4270>
13. Ortiz, M., & De Sola, J. (2021). La utilización de las notificaciones de las aplicaciones móviles en los medios de comunicación españoles. *Revista Latina de Comunicación Social*, 79, 283–302.
<https://nuevaepoca.revistalatinacs.org/index.php/revista/article/view/1498>
14. Naranjo, Y., & Pérez, L. (2021). Cuidado del paciente con enfermedad de Alzheimer. *Revista Archivo Médico de Camagüey*. http://scielo.sld.cu/scielo.php?pid=S1025-02552021000100014&script=sci_arttext
15. Vega, M. (2010). *Casos de uso UML* [Apuntes de clase]. Universidad de Granada.
<https://lsi2.ugr.es/~mvega/docis/casos%20de%20uso.pdf>
16. Pressman, R. S. (2010). *Ingeniería de software: Un enfoque práctico* (7a ed.). McGraw-Hill Interamericana.

17. Miro. (s. f.). *Diagrama de clases: Qué es, cómo hacerlo y ejemplos*.
<https://miro.com/es/diagrama/que-es-diagrama-clases-uml/>
18. Corporación Universitaria Remington. (2015). *Módulo: Ingeniería de software I*. Studocu. <https://www.studocu.com/co/document/corporacion-universitaria-remington/ingenieria-de-software/ingenieria-de-software-i-2016/33827784>
19. Gaitán, M. (2012, 15 de octubre). *Métricas orientadas a la función* [Blog]. WordPress. <https://marlonei.wordpress.com/modelos-de-desarrollo-de-la-gestion-del-software/metricas/metricas-orientadas-a-la-funcion/>
20. Lucidchart. (2025, 20 de junio). *¿Qué es un diagrama entidad-relación?*
<https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>
21. Patiño, F. (2022). *Guías prácticas: Elaboración de diagramas de casos de uso*. Universidad Cooperativa de Colombia.
<https://repository.ucc.edu.co/server/api/core/bitstreams/a7ea175f-de13-49fd-afba-c021baf7a6d0/content>
22. Lucidchart. (2025, 18 de junio). *UML use case diagram tutorial*.
<https://www.lucidchart.com/pages/uml-use-case-diagram>