



Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y Tecnología

Grado en Ingeniería Informática

Desarrollo de una aplicación iOS

para la gestión de alertas sanitaria

Trabajo fin de estudio presentado por:	Daniel Fornons Ortega
Director/a:	Victoriano Gómez Delgado
Fecha:	05-06-2025
Repositorio del código fuente:	https://github.com/DaniFornons/iACUT

Resumen

En el ámbito de la atención sanitaria urgente, la comunicación eficaz entre el personal sanitario y administrativo es fundamental para garantizar una respuesta rápida y coordinada. Actualmente, los equipos de “*Atenció Continuada i Urgent en el Territori*” (ACUT) de la “*Gerència d’Atenció Primària i a la Comunitat de l’Alt Pirineu i Aran*” enfrentan desafíos en la gestión de alertas sanitarias, como errores en los registros de datos, falta de trazabilidad y errores en la coordinación, debido al uso de métodos tradicionales como llamadas telefónicas y WhatsApp.

Este trabajo propone el desarrollo de iACUT, una aplicación nativa para iOS diseñada para optimizar la comunicación y el registro de intervenciones sanitarias en tiempo real. La aplicación sustituye los métodos obsoletos por un sistema seguro, eficiente y adaptado a las necesidades de los profesionales sanitarios y administrativos, garantizando la protección de datos sensibles y el cumpliendo de las normativas vigentes.

Los resultados demuestran una mejora significativa en la coordinación entre los equipos de ACUT, la trazabilidad de las alertas sanitarias y la experiencia del usuario. Además, se identifican líneas futuras de desarrollo como la expansión a Android, integración con sistemas hospitalarios y la implementación de inteligencia artificial para optimizar aún más el servicio.

Palabras clave: atención sanitaria urgente, aplicación móvil, comunicación en tiempo real, seguridad de datos, iOS

Abstract

In the field of urgent healthcare, effective communication between healthcare and administrative staff is essential to ensure a rapid and coordinated response. Currently, the “Atenció Continuada i Urgent en el Territori” (ACUT) teams of the “Gerència d’Atenció Primària i a la Comunitat de l’Alt Pirineu i Aran” face challenges in managing health alerts, such as data entry errors, lack of traceability, and coordination issues, due to the use of traditional methods such as phone calls and WhatsApp.

This project proposes the development of iACUT, a native iOS application designed to optimize communication and the real-time recording of healthcare interventions. The application replaces outdated methods with a secure, efficient system tailored to the needs of both healthcare and administrative professionals, ensuring the protection of sensitive data and compliance with current regulations.

The results show a significant improvement in coordination among ACUT teams, traceability of health alerts, and user experience. In addition, future development lines are identified, such as expansion to Android, integration with hospital systems, and the implementation of artificial intelligence to further optimize the service.

Keywords: urgent healthcare, mobile application, real-time communication, data security, iOS.

Índice de contenidos

1.	Introducción.....	1
1.1.	Motivación.....	1
1.2.	Planteamiento del trabajo	5
1.3.	Estructura del trabajo	6
2.	Contexto y Estado del Arte	7
2.1.	Contexto	7
2.2.	Estado del Arte	9
2.3.	Conclusiones.....	11
3.	Objetivos y metodología de trabajo.....	12
3.1.	Objetivo general	12
3.2.	Objetivos específicos.....	12
3.3.	Metodología de trabajo	13
3.3.1.	Equipo de trabajo.....	13
3.3.2.	Fases del proyecto	14
3.3.3.	Diagrama de Gannt	15
3.3.4.	Método Kanban	16
3.3.5.	Matriz de prioridad	17
4.	Desarrollo del proyecto	18
4.1.	Análisis de requisitos	19
4.1.1.	Toma de requisitos.....	19
4.1.2.	Requisitos funcionales	20
4.1.3.	Requisitos no funcionales	20
4.1.4.	Requisitos Técnicos	21
4.2.	Tecnologías utilizadas	22
4.2.1.	Xcode	22
4.2.2.	Swift	23
4.2.3.	SwiftUI.....	23
4.2.4.	Firebase.....	24
4.2.5.	Arquitectura Modelo-Vista-Vista de Modelo (MVVM)	25

4.3.	Cifrado de datos.....	26
4.4.	Diagrama de flujo de la aplicación.....	28
4.5.	Wireframes	29
4.5.1.	Vista autenticación y vista de restablecimiento de contraseña.....	30
4.5.2.	Vista principal para sanitario y administrativo.....	31
4.5.3.	Vista de usuarios y nuevo usuario para administrativos.....	32
4.5.4.	Vista de alertas para sanitario y administrativo.....	33
4.5.5.	Vista de nueva alerta para administrativo.	34
4.5.6.	Vista detalle de alerta para sanitario y administrativo.	35
4.5.7.	Vista de códigos de diagnóstico y vista de opciones.....	36
4.5.8.	Vista opciones y cambiar contraseña.....	37
4.6.	Firebase	38
4.6.1.	Firebase Authentication.....	38
4.6.2.	Colecciones Cloud Firestore.....	39
4.6.3.	Firebase Storage	40
4.6.4.	Reglas	41
4.7.	Casos de uso	42
4.7.1.	Acceso a la aplicación.....	42
4.7.2.	Recuperación de contraseña.....	43
4.7.3.	Mostrar alertas	44
4.7.4.	Mostrar detalles de una alerta.....	45
4.7.5.	Crear una nueva alerta.....	46
4.7.6.	Mostrar usuarios.....	47
4.7.7.	Mostrar detalles de un usuario	48
4.7.8.	Crear un nuevo usuario.....	49
4.7.9.	Buscar código de diagnóstico.....	50
4.7.10.	Mostrar opciones de usuario	51
4.7.11.	Cambiar contraseña de usuario	52
4.7.12.	Cerrar sesión de usuario	53
4.8.	Ejemplo práctico de Uso.....	54
4.8.1.	Recepción de la alerta.....	54
4.8.2.	Activación del personal sanitario	55
4.8.3.	Intervención del personal sanitario	56
4.8.4.	Cierre de la alerta.....	57
4.9.	Prueba piloto	58

5. Conclusiones y trabajo futuro.....	59
5.1. Conclusiones del trabajo	59
5.2. Líneas de trabajo futuro	60
Referencias bibliográficas	62
Anexo A. Leyenda del diagrama de actividad	64
Anexo B. Reglas de Firebase Firestore.....	65
Anexo C. Presentación reunión inicial	66
Anexo D. Presentación reunión de seguimiento	68
Índice de acrónimos.....	73

Índice de figuras

Figura 1. Distribución de los equipos ACUT al GAPiC Alt Pirineu	2
Figura 2. Proceso de comunicación entre SEM y ACUT.....	3
Figura 3. Ejemplo de error de registro	3
Figura 4. Diagrama de Gann para el desarrollo de la aplicación	15
Figura 5. Tablero Kanban para el desarrollo de la aplicación.....	16
Figura 6. Vista del entorno de programación Xcode	22
Figura 7. Arquitectura MVVM	25
Figura 8. Datos antes de la encriptación	27
Figura 9. Datos después de la encriptación.....	27
Figura 10. Diagrama de flujo de la aplicación	28
Figura 11. Vista autentificación.....	30
Figura 12. Vista reset de contraseña	30
Figura 13. Vista principal sanitario	31
Figura 14. Vista principal administrativo.....	31
Figura 15. Vista usuario	32
Figura 16. Vista nuevo usuario	32
Figura 17. Vista alertas sanitario	33
Figura 18. Vista alertas administrativo.....	33
Figura 19. Vista nueva alerta administrativo.....	34
Figura 20. Vista con mensaje error al crear una alerta	34
Figura 21. Vista detalle alerta administrativo	35
Figura 22. Vista detalle alerta sanitario.....	35
Figura 23. Vista código de diagnóstico	36
Figura 24. Vista opciones	37
Figura 25. Vista cambiar contraseña	37
Figura 26. colecciones de la aplicación iACUT.....	39
Figura 27. Aplicación del SEM	54
Figura 28. Creación de una alerta	54
Figura 29. Consulta de la alerta.....	55
Figura 30. Registro hora movilización del SEM	55

Figura 31. Navegación hacia el lugar de la alerta	55
Figura 32. Registro hora de entrada al domicilio	56
Figura 33. Registro hora de asistencia en la aplicación del SEM	56
Figura 34. Registro de la hora de salida y del diagnóstico	57
Figura 35. Registro finalización en la aplicación del SEM	57
Figura 36. Mapa de ubicación de las alertas	58
Figura 37. Reglas Firebase Firestore.....	65

Índice de tablas

Tabla 1. Cumplimiento del registro de datos de la aplicación del SEM durante el año 2004 ...	4
Tabla 2. Comparativa de aplicaciones sanitarias existentes	10
Tabla 3. Perfiles del grupo de trabajo y tareas asignadas	13
Tabla 4. Matriz de prioridad de las tareas del proyecto.....	17
Tabla 5. Distintas vistas previas de diseño de wireframes con Xcode	29
Tabla 6. Resultados de la prueba piloto	58

1. Introducción

En el presente trabajo de Final de Grado se pretende solucionar los problemas de comunicación y registro de datos durante las alertas sanitarias de “*Atenció Continuada i Urgent en el Territori*” (ACUT) de la “*Gerència d’Atenció Primària i a la Comunitat de l’Alt Pirineu i Aran*” (GAPiC Alt Pirineu).

Para ello, se ha desarrollado una aplicación móvil para iOS, llamada iACUT, que permite el intercambio de información entre los administrativos y los equipos sanitarios que forman el ACUT. Esta solución tecnológica pretende optimizar la coordinación y mejorar la eficiencia en la respuesta a emergencias sanitarias en una región caracterizada por su compleja geografía y dispersión poblacional.

1.1. Motivación

La Salud Pública en Catalunya se coordina políticamente desde el “*Departament de Salut*”, y económicaamente desde el “*Servei Català de la Salut*” (CatSalut), el cual se encarga de la contratación de los proveedores de salud que ofrecen los servicios a la ciudadanía.

Uno de los mayores proveedores de salud pública en Cataluña es el “*Institut Català de la Salut*” (ICS), que cuenta tanto con hospitales como con “*Centres d’Atenció Primària*” (CAP), donde las personas son atendidas por su médico y/o enfermero de familia.

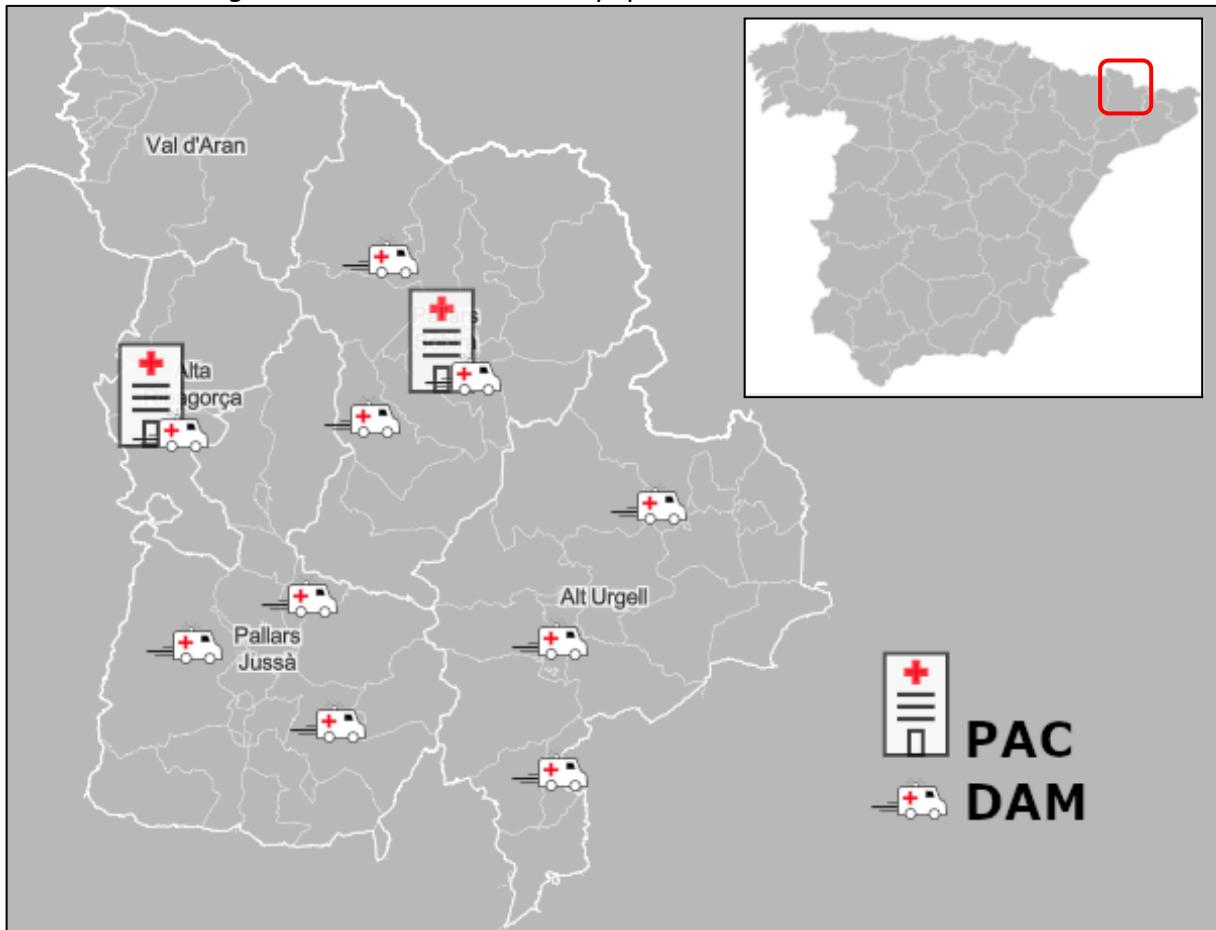
El servicio de salud está garantizado las 24 horas, ya sea de forma presencial o localizada, gracias a los equipos ACUT que aseguran la atención médica urgente y continuada en el territorio fuera del horario habitual de los CAP.

La GAPiC Alt Pirineu, donde actualmente trabaja como Ingeniero Técnico Informático, tiene dos centros ACUT llamados “*Punts d’Atenció Continuada*” (PAC) y 10 equipos ACUT de guardia, llamados “*Dispositius Aïllats de Muntanya*” (DAM), distribuidos en el área del Alt Pirineu¹, Figura 1, una región situada en el noroeste de Catalunya que se caracteriza por su amplia extensión territorial (5.776 km²), y baja densidad de población (13,3 habitantes/km²).

¹ Página web del GAPiC Alt Pirineu <https://www.icsgssleidapirineu.cat/>

² Fuente: <https://www.idescat.cat/indicadors/?id=aec&n=15227&lang=es&tema=xifpo&t=202400>

Figura 1. Distribución de los equipos ACUT al GAPIC Alt Pirineu

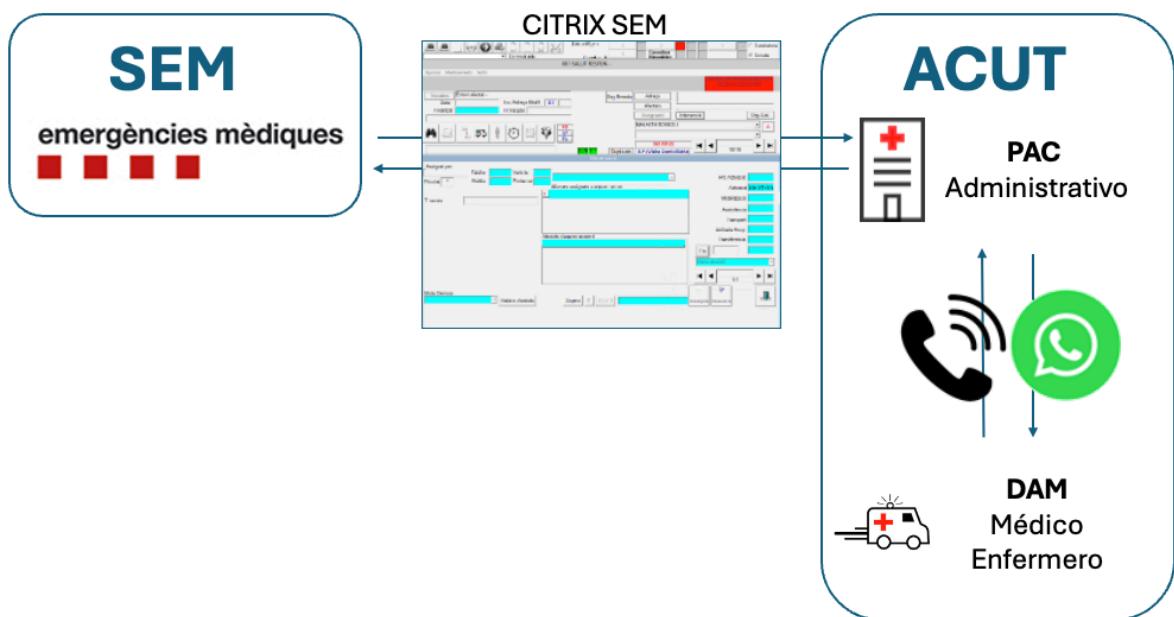


Las alertas sanitarias del ACUT provienen del “*Servei d’Emergència Mèdiques*” (SEM), quien alerta a los administrativos del ACUT ubicados en los CAP mediante una aplicación informática.

Los administrativos ACUT son los encargados de la activación del profesional sanitario de guardia mediante una llamada telefónica en la cual se le transmite toda la información relativa a la asistencia sanitaria.

Además, los administrativos deben registrar en la aplicación del SEM tanto la hora de salida del profesional sanitario como la hora de entrada y salida del domicilio del paciente en tiempo real. Estas comunicaciones se producen mediante WhatsApp. La Figura 2 muestra cómo se realiza la comunicación entre el SEM y el ACUT.

Figura 2. Proceso de comunicación entre SEM y ACUT



El cumplimiento del registro de datos en tiempo real durante las alertas sanitarias es uno de los indicadores utilizados por el SEM para medir la calidad del servicio. Aquellos datos que no se introducen o se introducen una vez finalizada la intervención quedan marcados con una X, tal como muestra la Figura 3.

Figura 3. Ejemplo de error de registro

Pre Activació	
Activació	07:45:53
Mobilització	08:05:27 X
Assistència	08:57:49
Transport	
Arribada Hosp.	
Transfèrencia	
PIRIN	09:01:34

El SEM establece como objetivo un umbral mínimo mensual del 85% para cada uno de los PACs. Sin embargo, los datos del año 2024 revelan incumplimientos recurrentes en algunas sedes, como se observa en la Tabla 1 en los valores mostrados en color rojo.

Tabla 1. Cumplimiento del registro de datos de la aplicación del SEM durante el año 2004

PAC	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO
Sort	100,00%	91,30%	90,48%	93,75%	94,44%	88,00%
Pont de Suert	94,74%	100,00%	100,00%	100,00%	100,00%	100,00%

PAC	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
Sort	88,89%	82,35%	100,00%	83,33%	100,00%	93,33%
Pont de Suert	78,57%	100,00%	100,00%	100,00%	100,00%	92,31%

Además, el uso de la llamada telefónica y el WhatsApp para la gestión y registro de las alertas sanitarias presenta importantes deficiencias que afectan a su eficiencia y precisión debido a los siguientes factores:

- **Riesgo de pérdida de información.**

La información transmitida verbalmente puede malinterpretarse, olvidarse o no quedar reflejada correctamente.

- **Dependencia de la disponibilidad.**

Para que la comunicación ocurra, tanto el emisor como el receptor deben estar disponibles al mismo tiempo.

- **Falta de registro automático.**

Las llamadas no generan un registro automático de la información compartida, lo que dificulta el seguimiento y la trazabilidad.

- **Falta de seguridad.**

Las llamadas telefónicas no suelen contar con mecanismos de encriptación o protección de datos sensibles, lo que representa un riesgo en un entorno sanitario donde se maneja información confidencial de pacientes.

Para cambiar esta situación y mejorar los indicadores de registro de control horario, la Dirección del ACUT del GAPiC Alt Pirineu propuso a la Unidad de Informática el desarrollo de una aplicación móvil que mejorase la comunicación, garantizase la seguridad de los datos y permitiese un registro horario más eficiente y fiable.

1.2. Planteamiento del trabajo

En este trabajo se propone el desarrollo de iACUT, una aplicación móvil diseñada para optimizar la comunicación y el registro de las intervenciones de los equipos de ACUT. Esta herramienta permitirá gestionar alertas sanitarias en tiempo real, registrar información relevante de cada caso y facilitar el acceso a los datos desde cualquier dispositivo autorizado, mejorando así la eficiencia y fiabilidad del servicio. Además, la aplicación deberá cumplir con las normativas relacionadas con la protección de datos personales de la Unión Europea y España, como el Reglamento General de Protección de Datos (RGPD) y la Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD).

Para la toma de requisitos, el seguimiento del proyecto y las pruebas se creará un grupo de trabajo formado por personal sanitario, administrativo, directivo y técnico del GAPiC Alt Pirineu.

Con el fin de organizar eficientemente las tareas y asegurar una implementación alineada con las necesidades del grupo de trabajo se seguirá una metodología ágil basada en Kanban y una matriz de prioridad

Para garantizar una integración óptima con los dispositivos utilizados por los equipos ACUT, la aplicación se desarrollará exclusivamente para iOS, ya que actualmente estos equipos ya disponen de dispositivos Apple³. De esta forma, se aprovecharán las ventajas del ecosistema de Apple en términos de seguridad, estabilidad y rendimiento, asegurando una experiencia fluida para los usuarios.

En cuanto a las tecnologías empleadas, se utilizará el IDE Xcode⁴ y el lenguaje de programación Swift⁵. Se utilizará el framework SwiftUI⁶ para el diseño y el desarrollo de la aplicación, permitiendo crear una aplicación moderna, intuitiva y adaptada a las necesidades del personal sanitario. Además, se utilizará Firebase⁷ como backend para el almacenamiento y gestión de datos en la nube, garantizando accesibilidad y sincronización en tiempo real.

³ Página web de Apple <https://www.apple.com/>

⁴ Página web de xcode <https://developer.apple.com/xcode/>

⁵ Página web de Swift <https://www.swift.org/>

⁶ Página web de SwiftUI <https://developer.apple.com/xcode/swiftui/>

⁷ Página web de Firebase <https://firebase.google.com/>

1.3. Estructura del trabajo

El presente trabajo se estructura en los siguientes capítulos:

- **Capítulo 1: Introducción**

Contiene una introducción al funcionamiento del ACUT del GAPiC Alt Pirineu y al problema de comunicaciones y registro que enfrentan sus equipos durante las asistencias sanitarias.

- **Capítulo 2: Contexto y Estado del Arte.**

En este capítulo se analiza el impacto de las tecnologías móviles en la sanidad, destacando su papel en la gestión de datos y comunicación en tiempo real. También se identifica desafíos como la seguridad, trazabilidad y la necesidad de sustituir herramientas informales por aplicaciones adaptadas a las exigencias del entorno sanitario.

- **Capítulo 3: Objetivos y Metodología de Trabajo**

Aquí se definen los objetivos del proyecto, detallando las metas a alcanzar con la aplicación iACUT, y se describe la metodología utilizada para el desarrollo del proyecto, que abarca desde el análisis de requisitos hasta las pruebas y validación del sistema.

- **Capítulo 4: Desarrollo Práctico del Proyecto**

Este capítulo se centra en la descripción del proceso de desarrollo de la aplicación, incluyendo la implementación de la interfaz de usuario, la integración con Firebase para la gestión de datos en tiempo real, el sistema de notificaciones, y las medidas de seguridad implementadas.

- **Capítulo 5: Conclusiones y Trabajo Futuro**

Se exponen las conclusiones obtenidas a partir del desarrollo y evaluación de la aplicación iACUT, destacando los logros alcanzados y las posibles áreas de mejora. Además, se proponen líneas de trabajo futuro para ampliar y optimizar la aplicación.

2. Contexto y Estado del Arte

2.1. Contexto

El uso de aplicaciones móviles en el ámbito sanitario, conocido como mHealth, del inglés *mobile health*, ha experimentado una notable expansión en los últimos años, impulsado por la creciente necesidad de soluciones más eficientes y accesibles para la gestión de la salud. Este fenómeno global refleja los esfuerzos por integrar tecnologías móviles que mejoren la comunicación, el manejo de datos y la prestación de servicios en tiempo real, especialmente en regiones con dificultades de acceso geográfico.

El concepto de mHealth fue destacado por la Organización Mundial de la Salud (OMS) en su informe *mHealth: New horizons for health through mobile technologies* (eHealth, 2011) donde se subraya su potencial para transformar la prestación de servicios sanitarios, especialmente en zonas remotas. Desde entonces, el mercado de aplicaciones de salud móvil ha crecido exponencialmente, con una valoración de 63,10 mil millones de euros en 2023 y una proyección de alcanzar los 275,65 mil millones de euros para 2032, según datos de (Markets, 2025). Este crecimiento se debe a factores como el aumento en la adopción de smartphones, la mayor conciencia sobre salud preventiva, los avances tecnológicos y las iniciativas gubernamentales que fomentan la digitalización del sector sanitario.

Diversos estudios, como los realizados por (Boulos et al., 2014) han demostrado que las aplicaciones móviles mejoran significativamente el seguimiento de enfermedades crónicas, la accesibilidad a servicios médicos y la comunicación entre profesionales y pacientes. Sin embargo, también identifican desafíos críticos, como la falta de un marco regulatorio robusto que garantice la fiabilidad y seguridad de estas herramientas, especialmente cuando manejan datos sensibles.

En este sentido, (Martínez-Pérez et al., 2014) enfatizan la necesidad de implementar medidas de seguridad avanzadas, como el cifrado de datos y la autenticación multifactorial, para proteger la privacidad de los usuarios.

En el ámbito de la gestión sanitaria, investigaciones como las de (Mosa et al., 2012) y (Ventola, 2014) resaltan el impacto positivo de las aplicaciones móviles en la eficiencia de los equipos médicos, facilitando el acceso a información actualizada, mejorando la coordinación y agilizando la toma de decisiones.

En este marco, surge la aplicación iACUT, diseñada para abordar los problemas de comunicación y registro de alertas sanitarias en los equipos ACUT del GAPiC Alt Pirineu, una región montañosa de Catalunya con baja densidad de población, muy diseminada, y con significativas barreras logísticas. En este contexto, donde la coordinación eficiente es fundamental para garantizar una respuesta oportuna a emergencias, las aplicaciones móviles se presentan como herramientas clave para superar limitaciones geográficas y optimizar la atención sanitaria.

El desarrollo de iACUT no solo debe responder a necesidades operativas, sino también cumplir con estrictas normativas de protección de datos, como el RGPD (Reglamento General de Protección de Datos (RGPD) | EUR-Lex, s. f.) y la LOPDGDD (Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales, 2018), dado que maneja información sensible de pacientes, por lo que la seguridad y privacidad de los datos son pilares fundamentales en su diseño.

2.2. Estado del Arte

El análisis de las principales aplicaciones utilizadas en entornos sanitarios para comunicación segura revela un panorama diverso de soluciones, cada una con sus fortalezas y limitaciones, ver Tabla 2. Estas plataformas sirven como referentes para el desarrollo de iACUT, permitiendo identificar las mejores prácticas y oportunidades de innovación.

Siilo se ha establecido como el estándar en la mensajería médica segura en Europa. Desarrollada específicamente para profesionales de la salud, ofrece un entorno completamente regulado que cumple con los más estrictos estándares de protección de datos. Su sistema de validación de identidad profesional y la trazabilidad completa de todas las comunicaciones la hacen especialmente adecuada para el manejo de información clínica sensible. Sin embargo, su modelo de negocio orientado a instituciones privadas puede limitar su adopción en sistemas públicos de salud con presupuestos restringidos.

Hospify emerge como una alternativa interesante para organizaciones que buscan una solución balanceada entre seguridad y usabilidad. Con servidores ubicados en la UE y un diseño intuitivo similar a aplicaciones de mensajería convencionales, reduce la curva de aprendizaje para los usuarios. No obstante, carece de algunas funcionalidades avanzadas necesarias para la gestión integral de equipos médicos.

TigerConnect domina el mercado norteamericano con su potente integración con los principales sistemas de historia clínica electrónica. Su capacidad para priorizar alertas y gestionar flujos de trabajo complejos la hace ideal para entornos hospitalarios de alto volumen. Sin embargo, su enfoque en el mercado estadounidense y su estructura de costes la hacen menos accesible para instituciones europeas.

La Meva Salut representa el paradigma de plataforma integrada de salud pública. Desarrollada por la Generalitat de Catalunya, combina historia clínica electrónica con herramientas de comunicación segura, ofreciendo una solución completa para la gestión de pacientes. Su principal valor reside en la perfecta interoperabilidad con los sistemas sanitarios públicos catalanes, aunque su alcance geográfico está limitado al territorio catalán.

Tabla 2. Comparativa de aplicaciones sanitarias existentes

Aplicación	Fortalezas	Limitaciones
Siilo 	<ul style="list-style-type: none"> Entorno completamente regulado. Validación de identidad profesional. Trazabilidad completa. Comunicación en tiempo real cifrada. Registro seguro de mensajes e imágenes clínicas. 	<ul style="list-style-type: none"> Coste elevado. Complejidad de implementación.
Hospify 	<ul style="list-style-type: none"> Servidores en UE. Interfaz intuitiva. Cifrado de extremo a extremo. Mensajes auto-eliminables para mayor seguridad. 	<ul style="list-style-type: none"> Funcionalidades básicas. Poca adaptación a emergencias.
TigerConnect 	<ul style="list-style-type: none"> Integración con historias clínicas. Gestión avanzada de flujos de trabajo. Comunicación estructurada entre equipos. Alta seguridad con control de acceso y auditoría. 	<ul style="list-style-type: none"> Orientación al mercado norteamericano. Coste elevado.
La Meva Salut 	<ul style="list-style-type: none"> Perfecta integración con sistema sanitario catalán. Multifuncionalidad. Acceso seguro mediante autenticación fuerte. Permite registro y consulta de información médica personal. 	<ul style="list-style-type: none"> Baja personalización. No orientada a la comunicación en tiempo real entre profesionales.

2.3. Conclusiones

El desarrollo de la aplicación iACUT se enmarca en un contexto de creciente adopción de soluciones *mHealth*, impulsado por la necesidad de mejorar la eficiencia y accesibilidad de los servicios sanitarios. Esta herramienta no solo busca optimizar la comunicación y gestión de alertas entre los equipos ACUT, sino también garantizar el cumplimiento de las normativas de protección de datos, un aspecto crítico dado el manejo de información sensible.

El Estado del Arte evidencia que, si bien las aplicaciones móviles en salud han demostrado su potencial para transformar la atención médica, su éxito depende en gran medida de la implementación de medidas robustas de seguridad y privacidad. En este sentido, iACUT debe incorporar tecnologías avanzadas de cifrado, autenticación segura y protocolos de notificación de brechas, además de asegurar la transparencia en el tratamiento de datos y el consentimiento informado de los usuarios.

Asimismo, el análisis de aplicaciones similares revela la importancia de diseñar una solución que combine usabilidad, interoperabilidad y adaptabilidad a entornos de emergencia, sin descuidar los aspectos normativos. La integración de un desarrollo nativo en iOS con Firebase como backend parece una estrategia adecuada para alcanzar este equilibrio, proporcionando rendimiento, escalabilidad y seguridad.

En conclusión, iACUT representa una innovación alineada con las tendencias globales de digitalización en salud, abordando necesidades específicas del sistema sanitario catalán y ofreciendo una respuesta tecnológica a los desafíos de coordinación en zonas rurales con población dispersa. Su implementación exitosa requerirá no solo un diseño técnicamente sólido, sino también un estricto cumplimiento de las normativas de protección de datos, asegurando así su eficacia y confiabilidad en la práctica clínica.

3. Objetivos y metodología de trabajo

3.1. Objetivo general

El objetivo principal de este trabajo es el de desarrollar una aplicación móvil para iOS denominada iACUT para mejorar la eficiencia y coordinación de los equipos ACUT del GAPiC Alt Pirineu durante las alertas sanitarias.

3.2. Objetivos específicos

Los objetivos específicos del trabajo son los siguientes:

1. **Desarrollar una interfaz intuitiva** que facilite el acceso a la información relevante durante las alertas sanitarias, utilizando *SwiftUI* para crear una experiencia de usuario fluida y accesible.
2. **Implementar un sistema de alertas y notificaciones** para asegurar que los profesionales reciban información crítica de forma inmediata.
3. **Crear una base de datos en tiempo real** a través de *Firebase* que permita registrar incidencias, realizar seguimientos y mantener un historial de eventos durante las alertas sanitarias.
4. **Garantizar la seguridad y confidencialidad de los datos** mediante técnicas de autenticación y cifrado, respetando las normativas sanitarias y de protección de datos.
5. **Evaluar la usabilidad y efectividad de la aplicación** mediante pruebas prácticas con los potenciales usuarios de la aplicación.

3.3. Metodología de trabajo

En los siguientes apartados se detallan los aspectos clave de la metodología adoptada, incluida la estructura del grupo de trabajo, la organización de las tareas y el seguimiento realizado durante el desarrollo del proyecto.

3.3.1. Equipo de trabajo

Para la realización de este trabajo se creó un grupo de trabajo formado por personal sanitario ACUT, personal administrativo ACUT, personal directivo y personal técnico de la GAPiC Alt Pirineu. En la Tabla 3 podemos ver las tareas asignadas a cada uno de los perfiles.

Tabla 3. Perfiles del grupo de trabajo y tareas asignadas

Perfil	Tareas
 Personal sanitario ACUT	<ul style="list-style-type: none">Definición de requisitos funcionales.Pruebas de usabilidad.Validación de la aplicación.
 Personal administrativo ACUT	<ul style="list-style-type: none">Definición de requisitos funcionales.Pruebas de usabilidad.Validación de la aplicación.
 Personal directivo	<ul style="list-style-type: none">Supervisión del proyecto.Aprobación de requisitos.Evaluación de resultados.
 Personal técnico	<ul style="list-style-type: none">Desarrollo de la aplicación.Implementación de requisitos.Resolución de incidencias.

Durante el desarrollo del trabajo se planificaron tres reuniones con los miembros del grupo de trabajo:

- 1. Reunión inicial:** Donde se definieron los objetivos, requisitos y se estableció el cronograma. Se puede consultar la presentación que se realizó en la reunión en el Anexo C.
- 2. Reunión de seguimiento.** Antes de iniciar el programa piloto, para presentar la aplicación y establecer los equipos que realizaron la prueba. Se puede consultar la presentación que se realizó en la reunión en el Anexo D.
- 3. Reunión final.** Para evaluar los resultados de la prueba piloto.

3.3.2. Fases del proyecto

El desarrollo de la aplicación iACUT se estructura en cuatro fases principales, siguiendo una planificación temporal que abarca un total de diez semanas. Esta organización permite abordar de manera secuencial y coherente las distintas etapas del proyecto, desde la conceptualización hasta su despliegue final.

Fase 1: Análisis, planificación y diseño UX/UI (2 semanas)

- Reunión Inicial.
- Definir requisitos funcionales y no funcionales.
- Diseñar la interfaz de usuario (Vistas).
- Diseñar la arquitectura de la aplicación (Modelo y Modelo de Vista).

Fase 2: Desarrollo (4 semanas)

- Implementar autenticación con Firebase.
- Crear la base de datos en Firebase.
- Pruebas iniciales de usabilidad.
- Desarrollar el MVVM para lista de enfermedades.
- Desarrollar el MVVM para la gestión de usuarios y permisos.
- Desarrollar el MVVM para las alertas sanitarias.
- Integrar notificaciones emergentes.

Fase 3: Pruebas (2 semanas)

- Reunión de seguimiento.
- Pruebas con usuarios clave y ajustes.
- Documentación técnica y manual de usuario.
- Revisión de la seguridad de Firebase.

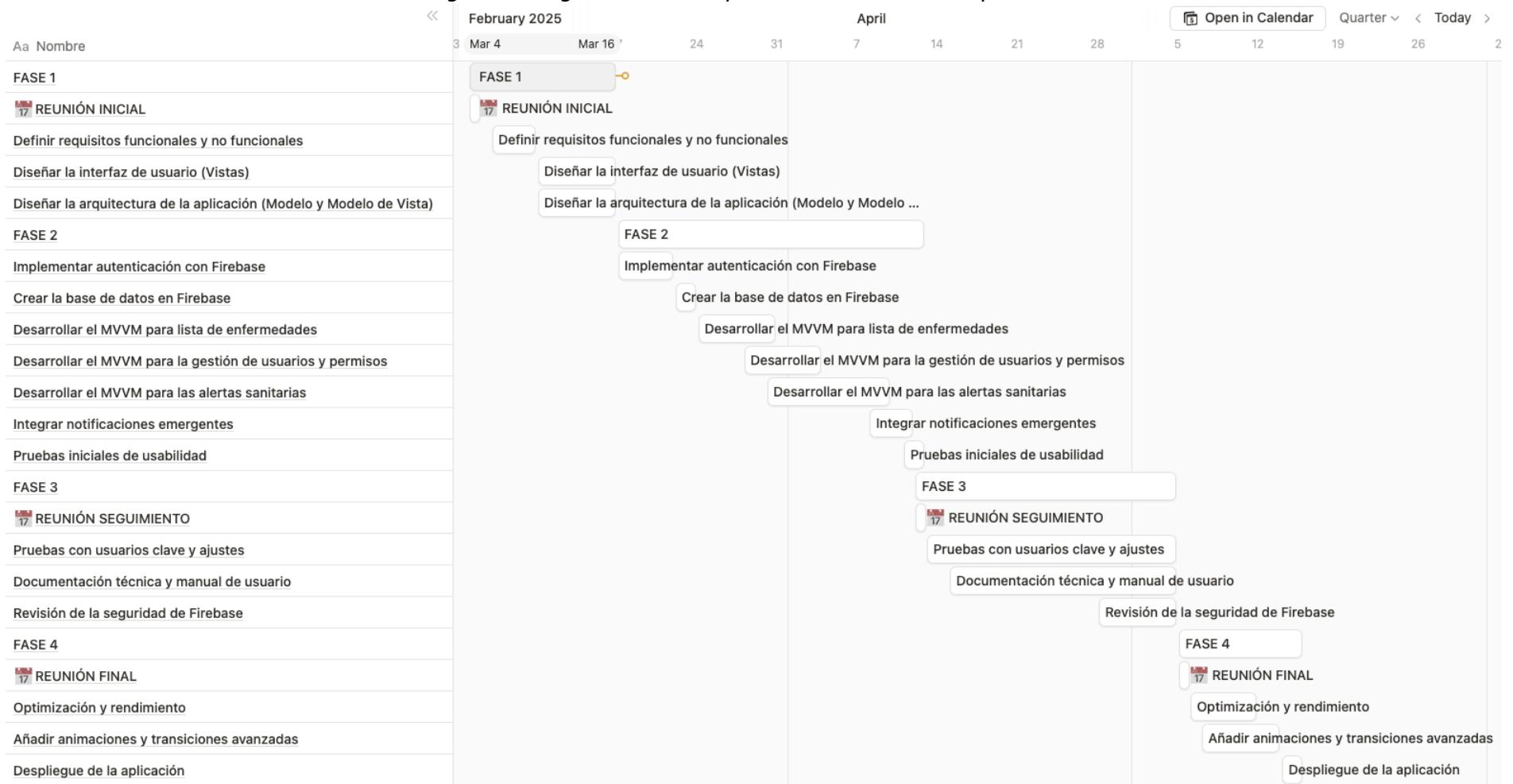
Fase 4: Formación y lanzamiento (2 semanas):

- Reunión Final.
- Optimización y rendimiento.
- Añadir animaciones y transiciones avanzadas en SwiftUI.
- Despliegue de la aplicación.

3.3.3. Diagrama de Gannt

Se utilizó un diagrama de Gantt, Figura 4, para representar visualmente la planificación del desarrollo de la aplicación iACUT. Esta herramienta facilita el seguimiento del progreso, permitiendo identificar hitos clave, tareas solapadas y la duración estimada de cada etapa del proyecto.

Figura 4. Diagrama de Gantt para el desarrollo de la aplicación



3.3.4. Método Kanban

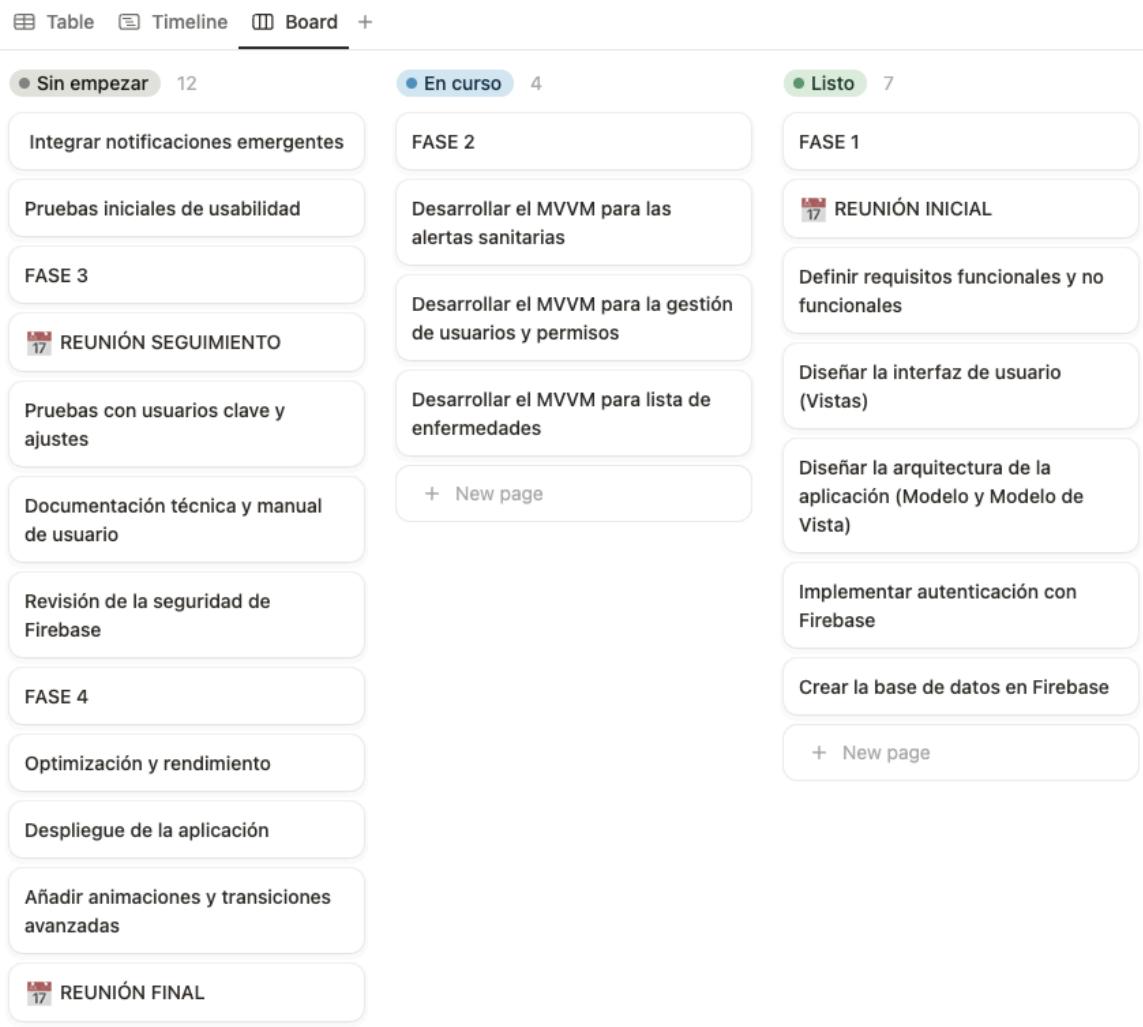
El método Kanban es un método de trabajo visual utilizado en la gestión de proyectos, basado en tarjetas visuales para representar el flujo de trabajo. Su capacidad para adaptarse a cambios y reducir tiempos de entrega lo convierte en una opción excelente para el diseño de aplicaciones y proyectos ágiles.

Para el desarrollo de la aplicación iACUT se utilizó el programa Notion⁸, dónde se creó un tablero Kanban, Figura 5, de tres dimensiones:

- **Sin empezar:** Las tareas o elementos que aún no se han iniciado.
- **En curso** Tareas que están siendo trabajadas.
- **Listo:** Tareas que han sido completadas.

Figura 5. Tablero Kanban para el desarrollo de la aplicación

Tareas



⁸ Página web de Notion <https://www.notion.com/desktop>

3.3.5. Matriz de prioridad

Para priorizar las tareas se catalogaron las mismas con un matriz de prioridad, ver Tabla 4, según su nivel de urgencia e importancia. Se clasifican las tareas en cuatro categorías:

- **Urgente e importante:** Prioridad alta. Hacer de inmediato.
- **Importante pero no urgente:** Planificar para más adelante.
- **Urgente pero no importante:** Delegar si es posible.
- **Ni urgente ni importante:** Evaluar si es necesario hacerlo.

Tabla 4. Matriz de prioridad de las tareas del proyecto

	Urgente	No urgente
Importante	Reunión Inicial. Definir requisitos funcionales y no funcionales.	Diseñar la interfaz de usuario (Vistas). Diseñar la arquitectura de la aplicación (Modelo y Modelo de Vista). Implementar autenticación con Firebase. Crear la base de datos en Firebase. Desarrollar el MVVM para lista de enfermedades. Desarrollar el MVVM para la gestión de usuarios y permisos. Desarrollar el MVVM para las alertas sanitarias Reunión de seguimiento. Pruebas con usuarios clave y ajustes. Revisión de la seguridad de Firebase. Documentación técnica y manual de usuario. Reunión Final. Despliegue de la aplicación.
No importante	Pruebas iniciales de usabilidad.	Integrar notificaciones emergentes. Optimización y rendimiento. Añadir animaciones y transiciones avanzadas.

4. Desarrollo del proyecto

El desarrollo del proyecto iACUT se llevó a cabo mediante una metodología ágil, centrada en la colaboración multidisciplinar, la planificación por fases y el seguimiento continuo de las tareas. La combinación del método Kanban y la matriz de prioridad permitió una gestión eficiente del tiempo y los recursos disponibles.

Se conformó un grupo de trabajo integrado por representantes de los perfiles clave involucrados en el entorno ACUT. Cada perfil asumió funciones específicas, como se detalla en la Tabla 3, lo que permitió una clara distribución de responsabilidades.

Durante el desarrollo del proyecto se organizaron tres reuniones clave: una reunión inicial para establecer los objetivos, los requisitos y el cronograma; una reunión de seguimiento previa al piloto, para asignar los equipos de prueba y presentar la aplicación; y una reunión final, destinada a evaluar los resultados y validar la utilidad de la solución propuesta.

El proyecto se estructuró en cuatro fases principales:

- **Análisis, planificación y diseño UX/UI:** En esta etapa se celebró la reunión inicial con el grupo de trabajo, donde se definieron los requisitos funcionales y no funcionales. Posteriormente, se diseñaron las vistas de la interfaz de usuario y la arquitectura técnica de la aplicación.
- **Desarrollo:** Durante esta fase se implementaron funcionalidades esenciales como la autenticación y la conexión a la base de datos. Se comenzaron las pruebas iniciales de usabilidad y se desarrollaron los principales módulos de la aplicación: gestión de enfermedades, usuarios y alertas sanitarias.
- **Pruebas:** En esta fase tuvo lugar la reunión de seguimiento, y se llevaron a cabo pruebas con usuarios clave. Se ajustaron funcionalidades a partir del feedback recibido y se elaboró la documentación técnica y el manual de usuario. Asimismo, se revisó la seguridad de la aplicación, especialmente en lo relativo a Firebase.
- **Formación y lanzamiento:** Finalmente, se realizó la reunión final con el grupo de trabajo para validar los resultados del piloto. Se llevaron a cabo tareas de optimización del rendimiento y se incorporaron animaciones y transiciones avanzadas con SwiftUI. La aplicación se desplegó y se dejó lista para su uso operativo.

4.1. Análisis de requisitos

4.1.1. Toma de requisitos

Tras la propuesta inicial de creación de la aplicación iACUT, se diseñó un primer prototipo mediante wireframes en Xcode, basado en las funcionalidades y necesidades estimadas por el equipo directivo del GAPiC Alt Pirineu.

Este prototipo sirvió como base para la reunión inicial, ver presentación en Anexo C, con los miembros del grupo de trabajo. Durante la reunión se contrastaron las hipótesis iniciales y se definieron conjuntamente los requisitos necesarios para la aplicación, dando lugar a la siguiente lista de funcionalidades:

- Desarrollo de una aplicación para dispositivos iOS.
- Acceso mediante el correo corporativo y una contraseña.
- Definición de roles: administrativo y sanitario.
- Permitir la creación de usuarios, tanto administrativos como sanitarios.
- Permitir la gestión de alertas: creación y eliminación.
- Permitir la gestión de contraseñas de los usuarios.
- Registro de horarios: inicio, entrada y salida del domicilio del paciente.
- Envío del código de diagnóstico de la enfermedad.
- Establecimiento de un código de colores que coincide con la aplicación del SEM.
- Enviar la ubicación del domicilio al navegador del móvil.
- Posibilidad de adjuntar una imagen con información adicional.
- Establecer la prioridad de la emergencia.
- Especificación de la información que deben enviar los administrativos al crear alertas.
- Especificación de la información que deben enviar los sanitarios.

Estas funcionalidades se agruparon posteriormente en los siguientes requisitos funcionales, no funcionales y tecnológicos.

4.1.2. Requisitos funcionales

Los requisitos funcionales describen las funcionalidades específicas que debe ofrecer la aplicación iACUT para cumplir con los objetivos del proyecto.

1. Gestión de usuarios

- Autenticación de usuarios mediante Firebase Authentication.
- Diferenciación de roles (sanitario, administrativo).
- Recuperación de contraseña y gestión del perfil de usuario.

2. Comunicación y alertas

- Notificaciones push en tiempo real sobre alertas sanitarias.
- Creación de alertas con detalles como fecha, ubicación y nivel de prioridad.
- Registro en tiempo real de las actividades del personal sanitario.

3. Gestión de equipos y recursos

- Asignación de equipos ACUT a diferentes incidencias.

4.1.3. Requisitos no funcionales

Los requisitos no funcionales definen las características de calidad y restricciones del sistema.

1. Usabilidad

- Interfaz intuitiva y accesible basada en SwiftUI.
- Adaptabilidad para usuarios con diferentes niveles de experiencia tecnológica.

2. Rendimiento

- Respuesta rápida a la interacción del usuario.
- Optimización del consumo de batería en dispositivos móviles.

3. Seguridad

- Protección de datos mediante autenticación segura y cifrado en Firebase.
- Restricción de accesos según los roles de usuario.

4. Disponibilidad

- Soporte para modo offline con sincronización posterior.

5. Compatibilidad

- Desarrollo para dispositivos iOS con versiones mínimas establecidas (iOS 15+).
- Optimización para diferentes tamaños de pantalla (iPhone y iPad).

4.1.4. Requisitos Técnicos

Los requisitos técnicos establecen las herramientas y tecnologías necesarias para el desarrollo e implementación de iACUT.

1. Lenguaje y frameworks

- Desarrollo en Swift utilizando SwiftUI para la interfaz gráfica.
- Uso de Firebase (Authentication, Firestore).

2. Infraestructura y almacenamiento

- Base de datos Firestore para almacenamiento de registros y usuarios.
- Almacenamiento de archivos en Firebase Storage.

3. Servicios y APIs

- Implementación de Firebase Cloud Functions para lógica en el servidor.
- Uso de Apple Push Notification Service (APNs) para notificaciones.

4. Control de versiones y despliegue

- Uso de GitHub para control de versiones.
- Distribución mediante TestFlight⁹ y publicación en la App Store¹⁰.

5. Seguridad y cumplimiento

- Encriptación de datos sensibles en Firestore.
- Cumplimiento con normativas de protección de datos como el GDPR y la LOPDGDD.

⁹ Página web de TestFlight <https://developer.apple.com/testflight/>

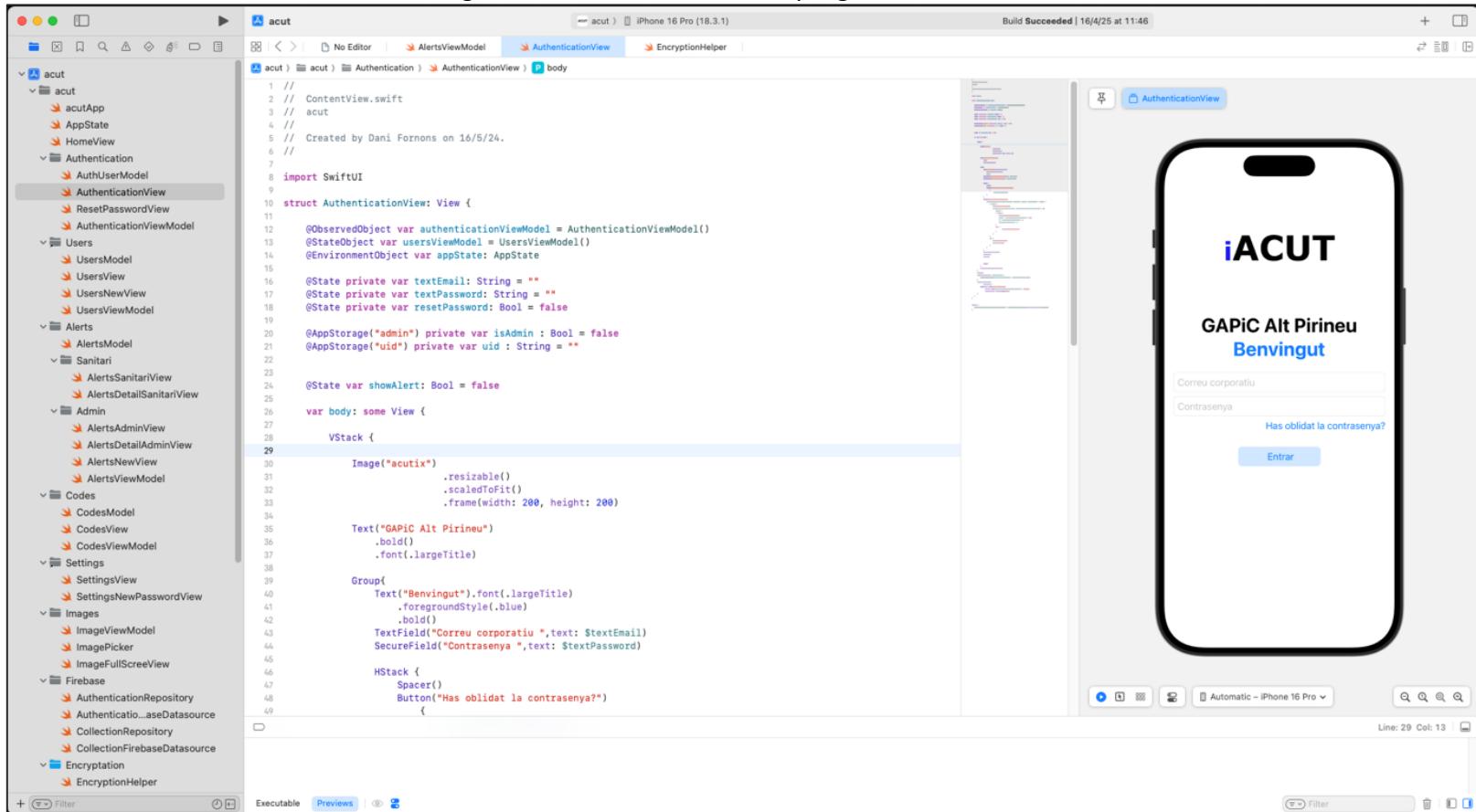
¹⁰ Página web de AppStore <https://www.apple.com/es/app-store/>

4.2.Tecnologías utilizadas

4.2.1. Xcode

Xcode, es el entorno de desarrollo integrado (IDE) de Apple, Figura 6, utilizado para el desarrollo de la aplicación iACUT. Este entorno proporciona herramientas avanzadas para la escritura, depuración y prueba del código, además de permitir la vista previa en tiempo real.

Figura 6. Vista del entorno de programación Xcode



4.2.2. Swift

Swift es el lenguaje de programación utilizado para desarrollar la aplicación iACUT. Diseñado por Apple, Swift es un lenguaje moderno, seguro y eficiente que facilita el desarrollo de aplicaciones con un alto rendimiento y una sintaxis clara y concisa.

4.2.3. SwiftUI

Se ha elegido SwiftUI para la construcción de la interfaz de usuario de iACUT. SwiftUI es un framework declarativo introducido por Apple que revoluciona la forma de construir interfaces de usuario para aplicaciones iOS.

Sus principales ventajas incluyen:

- **Desarrollo más rápido y eficiente:** Permite crear interfaces complejas con menos código, acelerando el proceso de desarrollo.
- **Diseño reactivo:** Las vistas se actualizan automáticamente cuando cambian los datos subyacentes, ideal para mostrar información en tiempo real como las alertas sanitarias.
- **Vista previa en tiempo real:** La integración con Xcode permite visualizar cambios inmediatamente sin necesidad de ejecutar la aplicación completa.
- **Consistencia entre plataformas:** Facilita la adaptación de la interfaz a diferentes dispositivos Apple (iPhone, iPad) manteniendo una experiencia coherente.

SwiftUI simplifica la creación de interfaces interactivas y adaptativas con menos código, lo que acelera el desarrollo y mejora la mantenibilidad de la aplicación.

Además, su integración con Xcode permite obtener una vista previa en tiempo real de las interfaces, optimizando así el proceso de diseño y desarrollo.

4.2.4. Firebase

Firebase se utiliza como backend para garantizar el almacenamiento seguro y la sincronización de datos en tiempo real.

Firebase es una plataforma de desarrollo de Google que ofrece diversos servicios en la nube, como bases de datos en tiempo real, autenticación de usuarios, almacenamiento y análisis. En particular, Firestore, su base de datos NoSQL en tiempo real, es ideal para sincronizar datos entre dispositivos, lo que resulta esencial en una aplicación que requiere actualizaciones constantes y comunicación fluida.

Firebase proporciona una infraestructura backend completa que simplifica significativamente el desarrollo de aplicaciones móviles. Las características más relevantes son:

- **Autenticación segura:** Firebase Authentication ofrece múltiples métodos de autenticación y cumple con estándares de seguridad rigurosos, esencial para una aplicación que maneja datos sanitarios sensibles.
- **Base de datos en tiempo real:** Firestore permite la sincronización instantánea de datos entre dispositivos, crucial para la coordinación entre administrativos y personal sanitario durante las alertas sanitarias.
- **Almacenamiento en la nube:** Firebase Storage facilita el almacenamiento de imágenes y documentos relacionados con las alertas sanitarias.
- **Análisis y monitoreo:** Firebase Analytics y Crashlytics proporcionan información valiosa sobre el uso de la aplicación y posibles problemas, permitiendo mejoras continuas.

La combinación de SwiftUI y Firebase proporciona una solución completa para el desarrollo ágil de la aplicación iACUT, permitiendo a los equipos centrarse en ofrecer una experiencia de usuario fluida y eficiente. Mientras tanto, Firebase gestiona el backend y las interacciones de datos en tiempo real, mejorando la productividad y facilitando la iteración sobre nuevas funcionalidades.

4.2.5. Arquitectura Modelo-Vista-Vista de Modelo (MVVM)

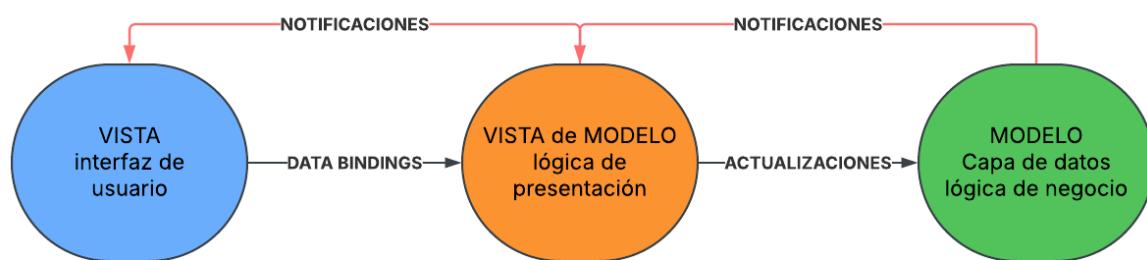
Para el desarrollo de la aplicación iACUT, se ha optado por el patrón de arquitectura Modelo-Vista-Vista de Modelo (MVVM), ampliamente utilizado en aplicaciones móviles y de escritorio por su capacidad de estructurar el código de manera eficiente.

El MVVM, Figura 7, está compuesto por:

- **Modelo:** Representa la capa de datos y la lógica de negocio de la aplicación. Se encarga de gestionar la información utilizada en la aplicación, incluyendo la interacción con bases de datos o servicios de red.
- **Vista:** Es la interfaz gráfica con la que interactúa el usuario. Está vinculada a un ViewModel mediante bindings, permitiendo la actualización automática de los datos sin necesidad de modificar directamente la interfaz.
- **Vista de Modelo:** Actúa como intermediario entre el modelo y la vista, preparando los datos del modelo para que la vista los muestre correctamente. Además, maneja la lógica de presentación y la manipulación de datos sin tener conocimiento directo de la interfaz de usuario.

El uso de MVVM en iACUT permite una mayor separación de responsabilidades, facilitando la prueba de la lógica de la aplicación y mejorando su mantenibilidad y escalabilidad.

Figura 7. Arquitectura MVVM



4.3. Cifrado de datos

En el contexto de la atención sanitaria urgente, el tratamiento de datos sensibles, como diagnósticos, ubicaciones o información personal del paciente, requiere un nivel de seguridad elevado, en conformidad con el RGPD y la LOPDGDD.

Aunque Firebase proporciona cifrado automático tanto en tránsito como en reposo, este no garantiza un cifrado de extremo a extremo, donde solo el emisor y el receptor puedan acceder al contenido. Por ello, se ha considerado oportuno incorporar una capa adicional de cifrado en la aplicación iACUT, implementando un mecanismo de cifrado local antes del envío de los datos a Firebase.

En el desarrollo de aplicaciones móviles existen principalmente dos enfoques de cifrado:

- **Cifrado simétrico (AES):** usa una única clave para cifrar y descifrar. Es rápido y eficaz, ideal para cifrar campos de texto.
- **Cifrado asimétrico (RSA, ECC):** usa un par de claves pública/privada. Más seguro para el intercambio de claves, pero menos eficiente para datos voluminosos.
- **Cifrado híbrido:** combina ambos. Se cifra el mensaje con una clave simétrica y esta se protege con cifrado asimétrico.

Para la aplicación de iACUT se opta por el cifrado simétrico con AES-GCM, utilizando la biblioteca nativa de Apple, CryptoKit¹¹.

En la Figura 8 y 9 se observa cómo quedan los datos registrados en Firebase antes y después de usar la encriptación

¹¹ Página web de CryptoKit <https://developer.apple.com/documentation/cryptokit/>

Figura 8. Datos antes de la encriptación

```
▼ adreça
  0 "Pobla de Segur"
  1 "Av. Catalunya"
  2 ""

codiDiagnostic: ""
codiSem: "1234"
comentari: "Persona que ha caigut dins de casa"
horaEntrada: ""
horaInici: ""
horaSortida: ""
imatge: ""
nom: "Médico ACUT"
nomDiagnostic: ""
prioritat: "P1"
uid: "OLw4vKUXZag32y7Za3PXHzbhycI2"
```

Figura 9. Datos después de la encriptación

```
adreça: "8nv8Nr0/cjWTIQGzfqTiRORgXvMZvsakI5fH1cc2zzQyJDw="
codiDiagnostic: "NUEj0G51Ecwpj2TF8Y44HgxQnipYaRMLp+JcOg=="
codiSem: "KC3ZxktT+X0gUFKjAloa3Ob1xFgE9xB0lvkyR1l7uuJ"
comentari: "rGLiftNMiBzPo/YQxdS11Lgw2RLKgMda6Z3L4g=="
horaEntrada: "TS4fhmMGiz2Bvm+ySG4EtYBZ2jFrKgfC6Ar/Ew=="
horaInici: "6MBhkimRUAEZ/gmX+mzDfLcRkyoP3mBntfcV0g=="
horaSortida: "LvTpYbgMC7nS2ptDD8qjLqa0jY6UvVrJpcbxFA=="
imatge: "QG1ph0BDyu/0GpDG92QIIHK00Q6GsrnVs14tBw=="
nom: "tHg6Z8xom+73TwokAGVFq0i9oyhUV31GYNSAexi1HBRjAi3SpXhGew=="
nomDiagnostic: "RvvZ+p+qZ06R4Y9l8IMMqlvWkbGDnx2iVHAN/A=="
prioritat: "dIDAaJWqSiXFasQnBl41XB/OkGXRYtWjzu/XcsI"
uid: "AGYAEsut750hl5i6jnTVGY7x8Ky3ilU0FBGK0vTopTMdFDGPQVMgb+00mffd
```

4.4. Diagrama de flujo de la aplicación

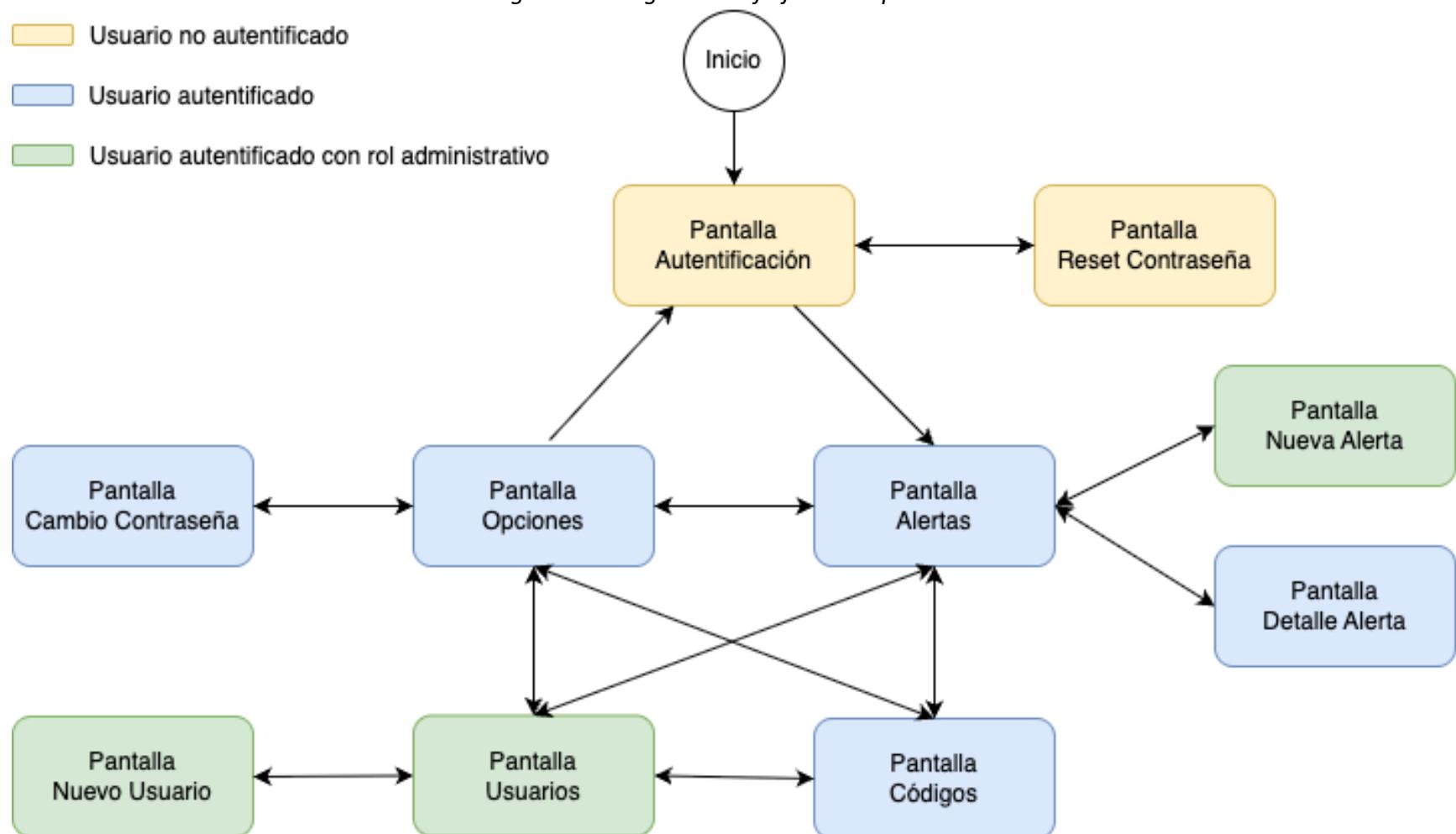
En el desarrollo de la aplicación iACUT, se ha optado por el patrón de arquitectura MVVM, en lugar de un enfoque tradicional basado en la Programación Orientada a Objetos. Debido a esta decisión, la representación visual más adecuada para comprender el funcionamiento de la aplicación es un diagrama de flujo, Figura 10.

Figura 10. Diagrama de flujo de la aplicación

■ Usuario no autentificado

■ Usuario autentificado

■ Usuario autentificado con rol administrativo



4.5. Wireframes

En el desarrollo de aplicaciones, los wireframes se usan para definir la estructura y la experiencia de usuario antes de la implementación.

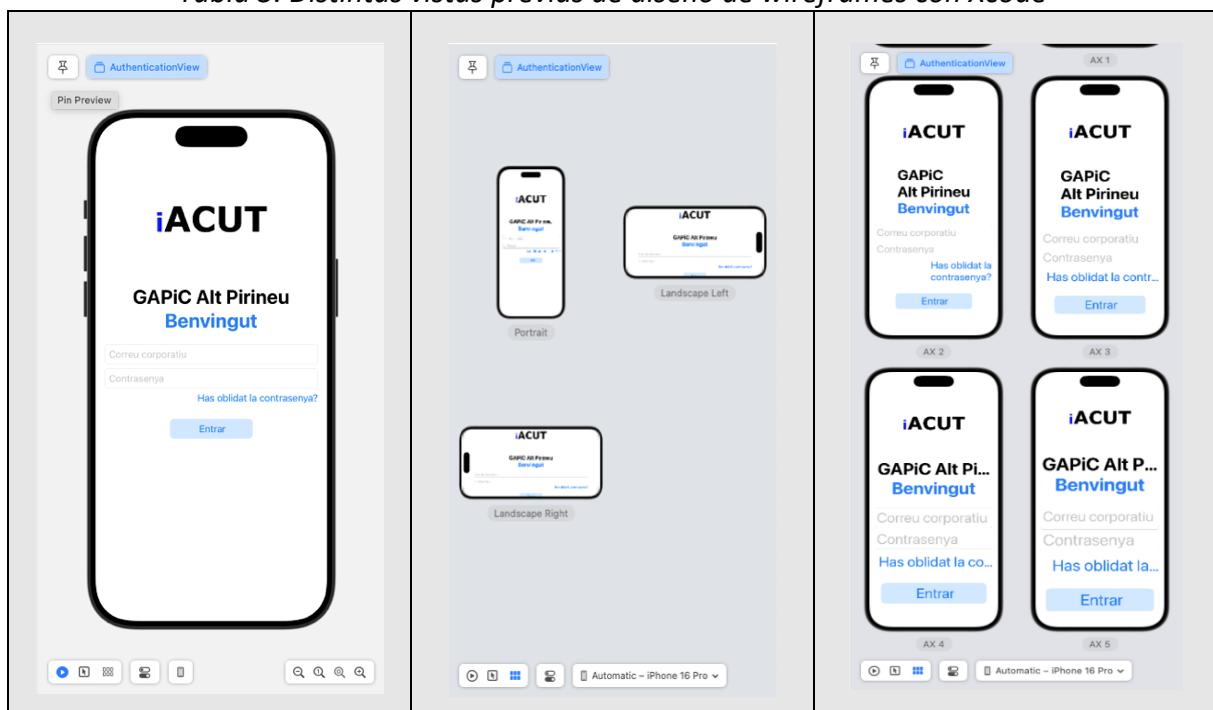
Para la generación de los wireframes se ha optado por usar directamente las vistas en Xcode.

Esta decisión permite tener:

- **Coherencia con la tecnología final:** Dado que la aplicación está desarrollada en SwiftUI, usar Xcode permite crear wireframes directamente en el entorno de trabajo.
- **Interactividad desde el inicio:** A diferencia de wireframes estáticos, en Xcode se pueden generar prototipos funcionales con navegación básica, permitiendo validar la experiencia de usuario en un dispositivo real.
- **Mayor fidelidad al diseño real:** Al trabajar directamente en código, los wireframes están más alineados con la estructura de la aplicación, evitando discrepancias entre el diseño inicial y la implementación.
- **Aprovechamiento de previews en SwiftUI:** La funcionalidad de vista previa en Xcode permite iterar rápidamente sobre los wireframes sin necesidad de ejecutar la aplicación en el simulador.

La Tabla 5 muestra las distintas vistas previas disponibles en Xcode.

Tabla 5. Distintas vistas previas de diseño de wireframes con Xcode



4.5.1. Vista autentificación y vista de restablecimiento de contraseña

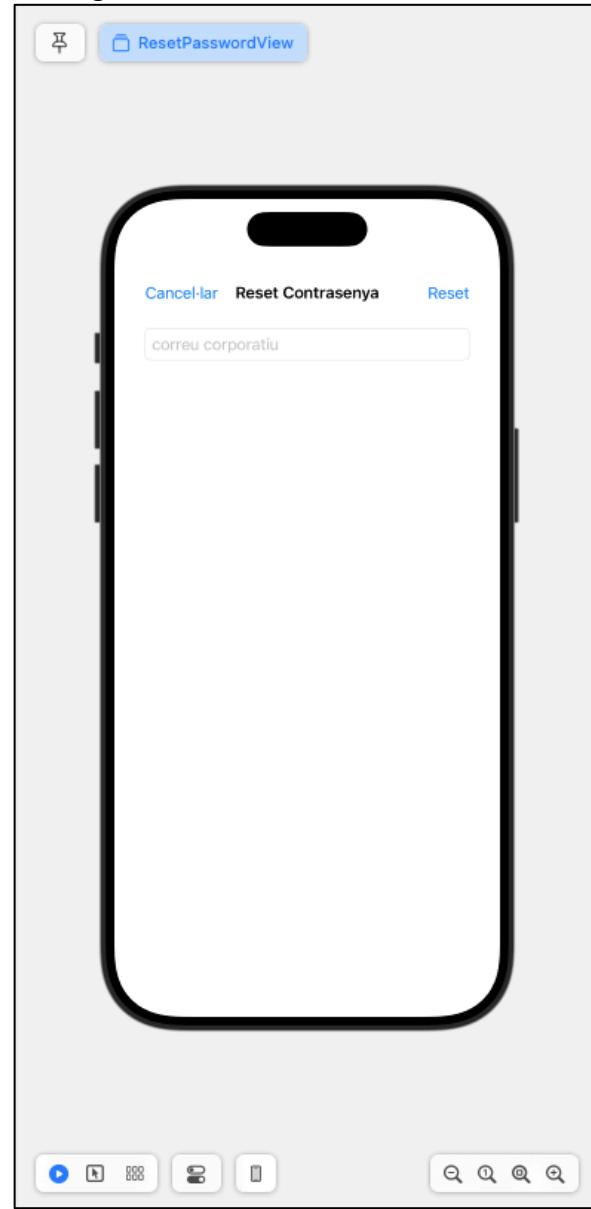
Por un lado, la vista AuthenticationView recoge el correo corporativo y la contraseña del usuario para entrar a la aplicación, Figura 11.

Por otro lado, la vista ResetPasswordView recoge el correo corporativo del usuario para poder restablecer las credenciales, Figura 12.

Figura 11. Vista autentificación



Figura 12. Vista reset de contraseña



4.5.2. Vista principal para sanitario y administrativo

La Figura 13, vista HomeView, muestra las opciones del menú “alertes”, “codis y “opcions” para los usuarios con el rol sanitario.

La Figura 14, vista HomeView, muestra las opciones del menú “alertes”, “usuaris”, “codis y “opcions” para los usuarios con el rol administrativo.

Figura 13. Vista principal sanitario

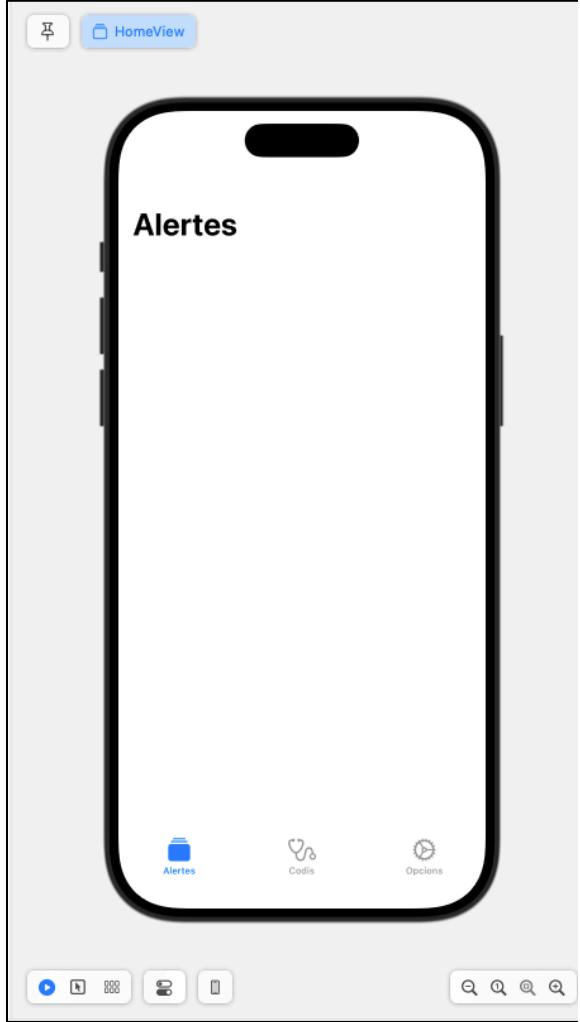
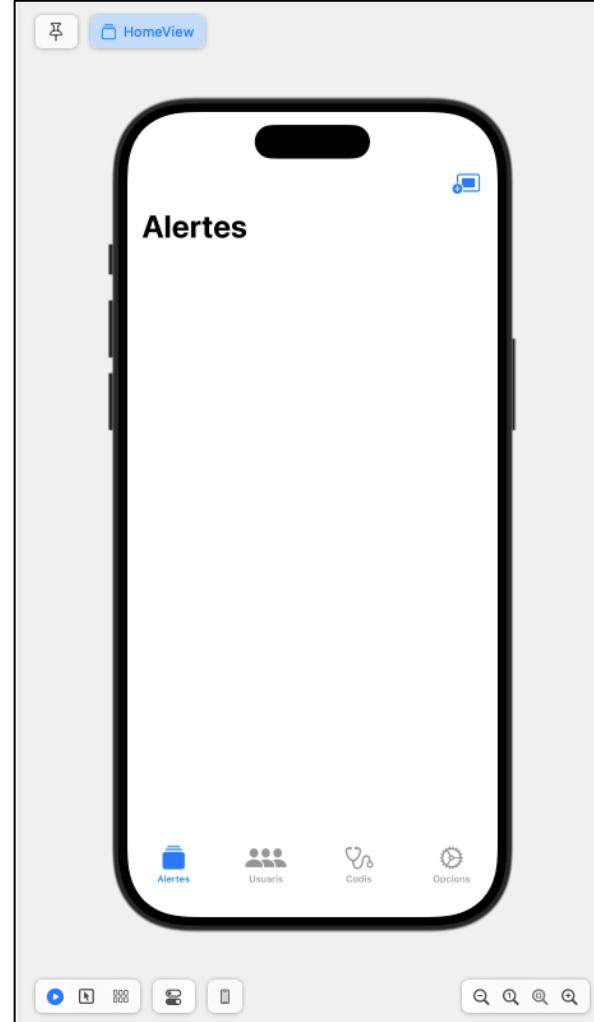


Figura 14. Vista principal administrativo



4.5.3. Vista de usuarios y nuevo usuario para administrativos

La vista UsersView y UsersNewView sólo está disponible para los usuarios con rol administrativo.

La vista UserView muestra todos los usuarios del sistema, identificando su rol mediante ACUT, para sanitarios, y UAAU, para administrativos, Figura 15, y permite acceder a crear un nuevo usuario.

La vista UsersNewView permite dar de alta a nuevos usuarios, tanto administrativos como sanitarios, Figura 16.

Figura 15. Vista usuario

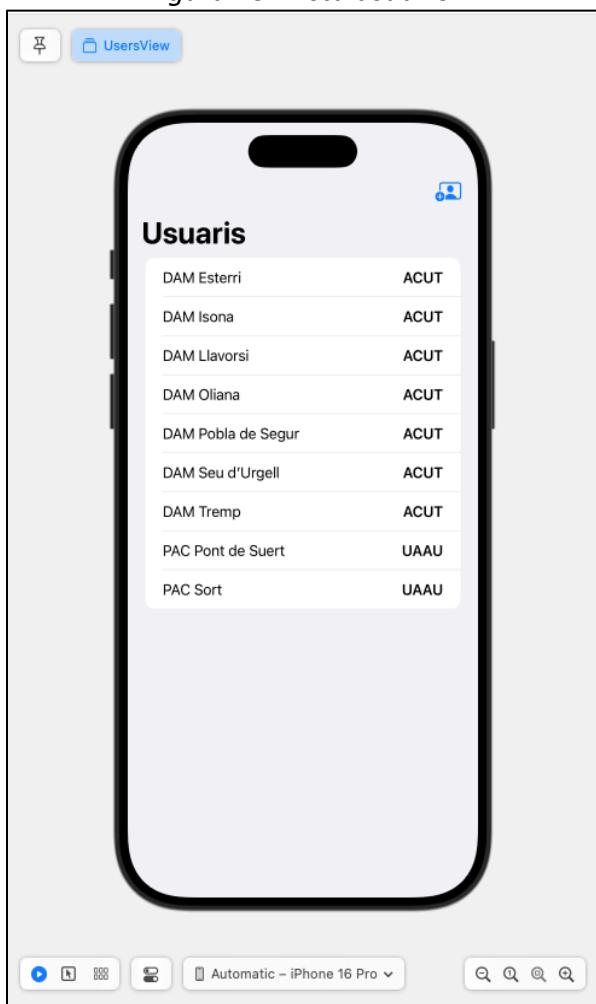
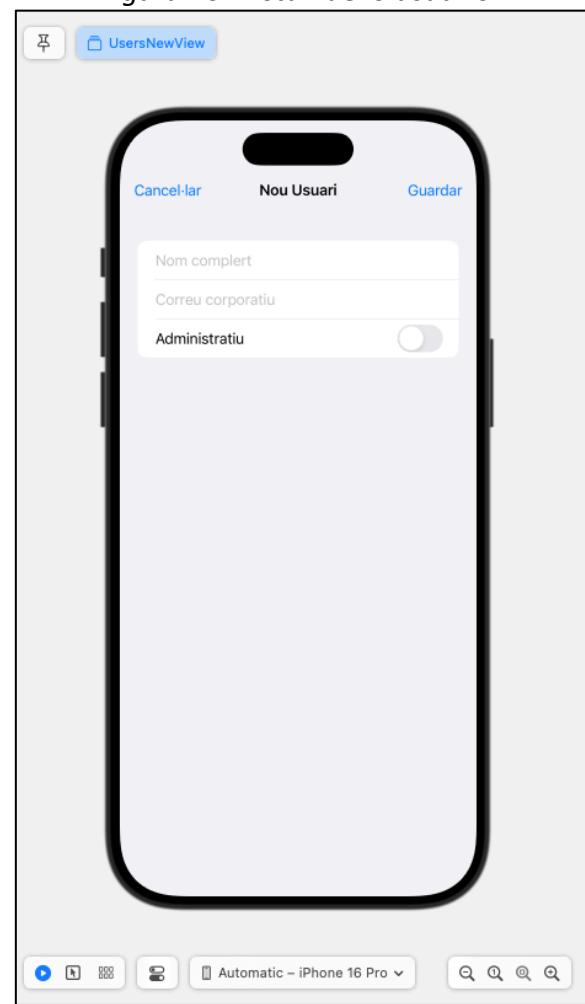


Figura 16. Vista nuevo usuario



4.5.4. Vista de alertas para sanitario y administrativo

Por un lado, la vista AlertsSanitariView muestra todas las alertas activas para un sanitario, Figura 17.

Por otro lado, la vista AlertsAdminView muestra todas las alertas activas, Figura 18, y permite acceder a crear una nueva alerta.

Las alertas se muestran en un color dependiendo del estado de esta, siendo:

- Gris, para las alertas que aún no se ha introducido la hora de inicio.
- Verde, para las alertas que sólo tiene la hora de inicio.
- Amarillo, para las alertas que tienen hora de inicio y hora de entrada.
- Azul, para las alertas que ya han finalizado y tienen el registro horario completo y el código de diagnóstico introducido.

Figura 17. Vista alertas sanitario

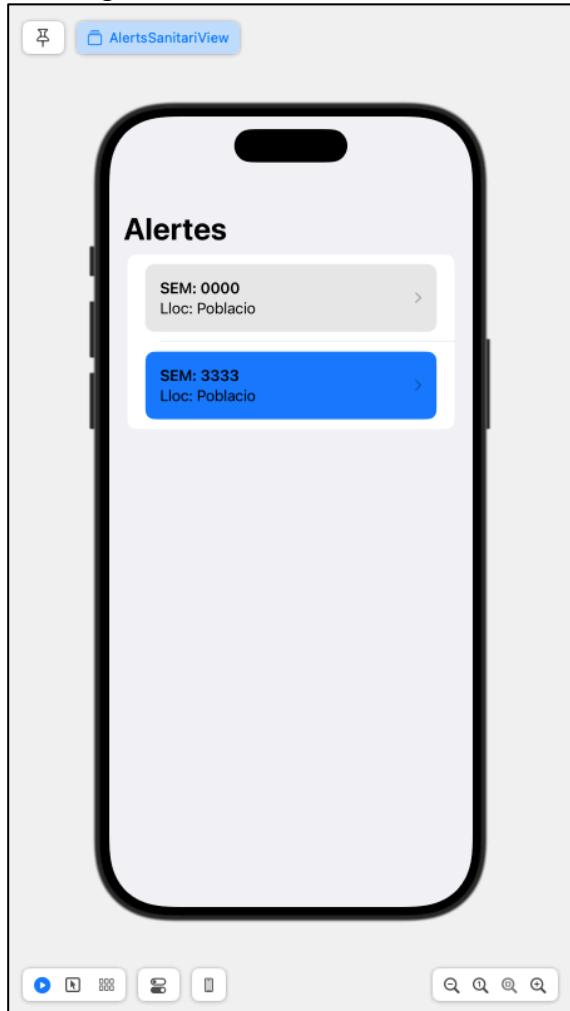
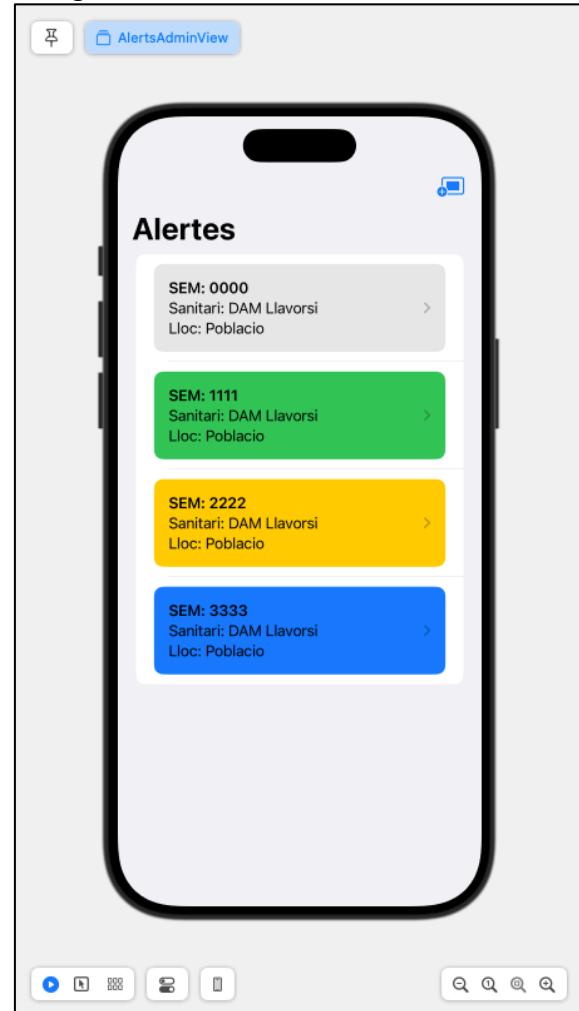


Figura 18. Vista alertas administrativo



4.5.5. Vista de nueva alerta para administrativo.

La vista AlertsNewView permite introducir los campos necesarios para crear una nueva alerta y seleccionar una imagen, Figura 19. Los campos “codi SEM”, “Professional” y “Població” son obligatorios, Figura 20

Figura 19. Vista nueva alerta administrativo

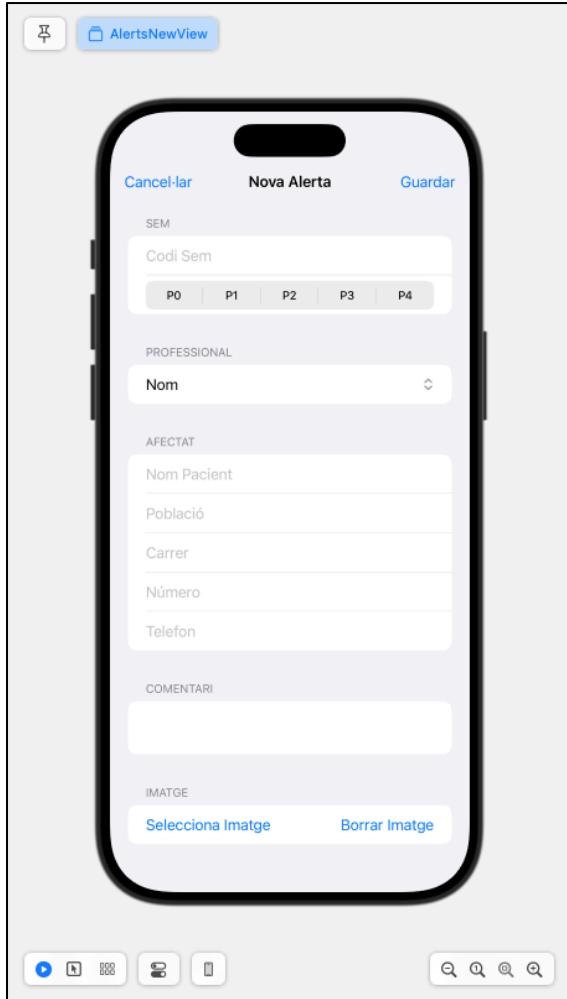
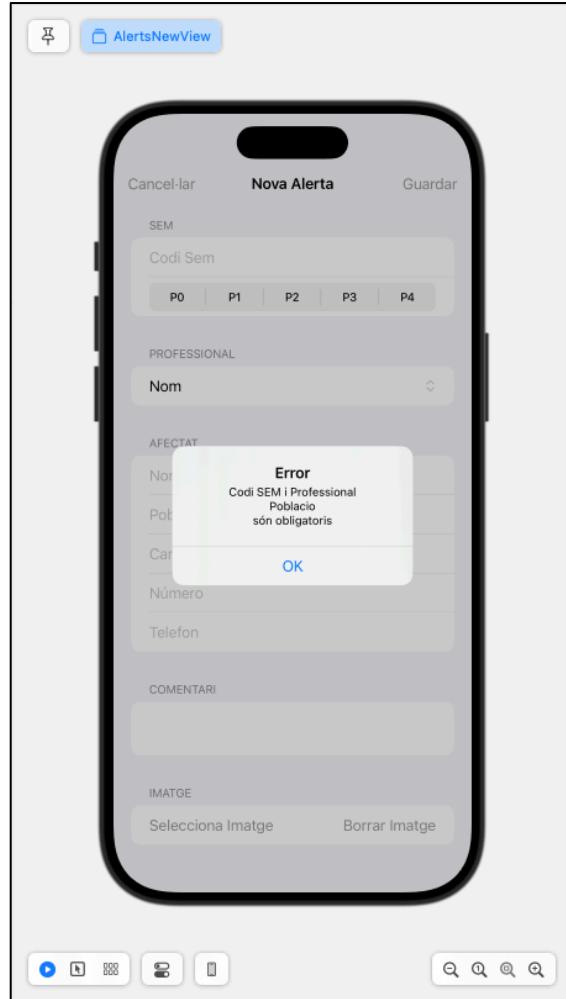


Figura 20. Vista con mensaje error al crear una alerta



4.5.6. Vista detalle de alerta para sanitario y administrativo.

Por un lado, la vista AlertsDetailAdminView permite, a los administrativos, consultar los campos de la alerta, Figura 21.

Por otro lado, la vista AlertsDetailSanitariView permite, a los sanitarios, consultar los campos de la alerta e introducir el registro horario y el código de diagnóstico. También permite llamar directamente al teléfono y abrir el navegador con la dirección de la alerta, Figura 22.

Figura 21. Vista detalle alerta administrativo

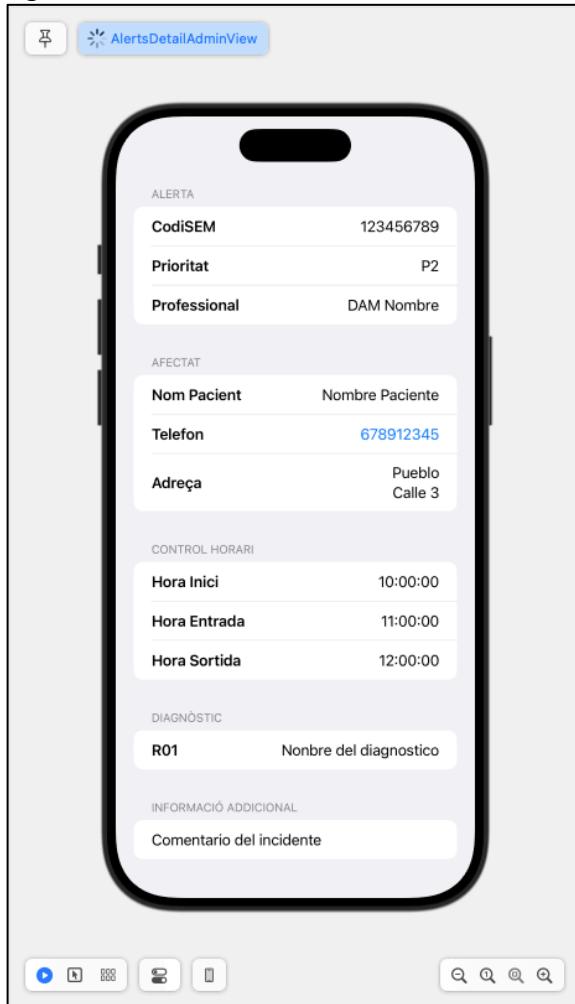
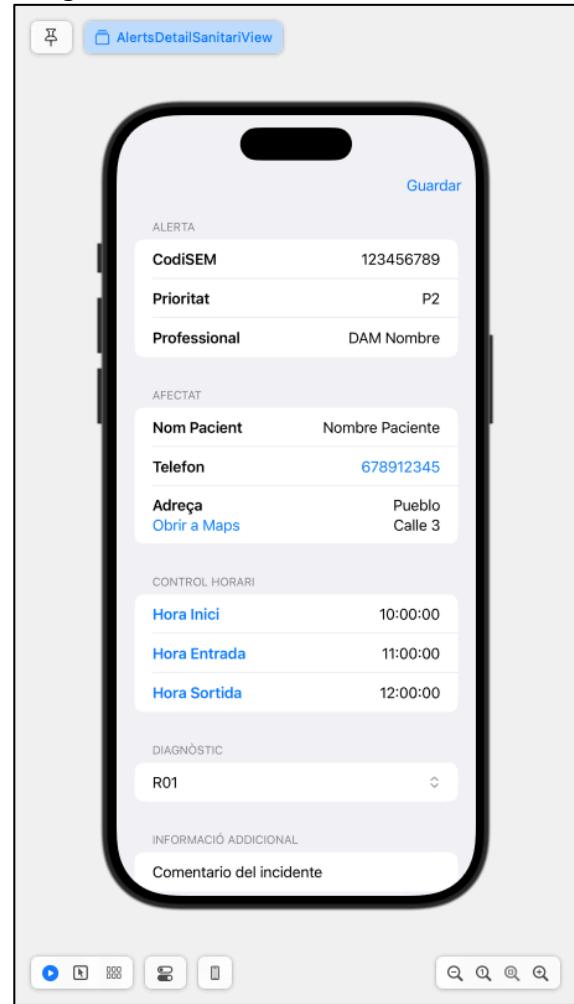


Figura 22. Vista detalle alerta sanitario



4.5.7. Vista de códigos de diagnóstico y vista de opciones.

La figura 23, vista CodesView, muestra todos los códigos de diagnóstico informados y permite buscarlos por texto.

Figura 23. Vista código de diagnóstico



4.5.8. Vista opciones y cambiar contraseña

La vista SettingsView permite acceder al cambio de contraseña y salir de la aplicación, Figura 24.

La vista SettingsNewPasswordView permite cambiar la contraseña, Figura 25.

Figura 24. Vista opciones

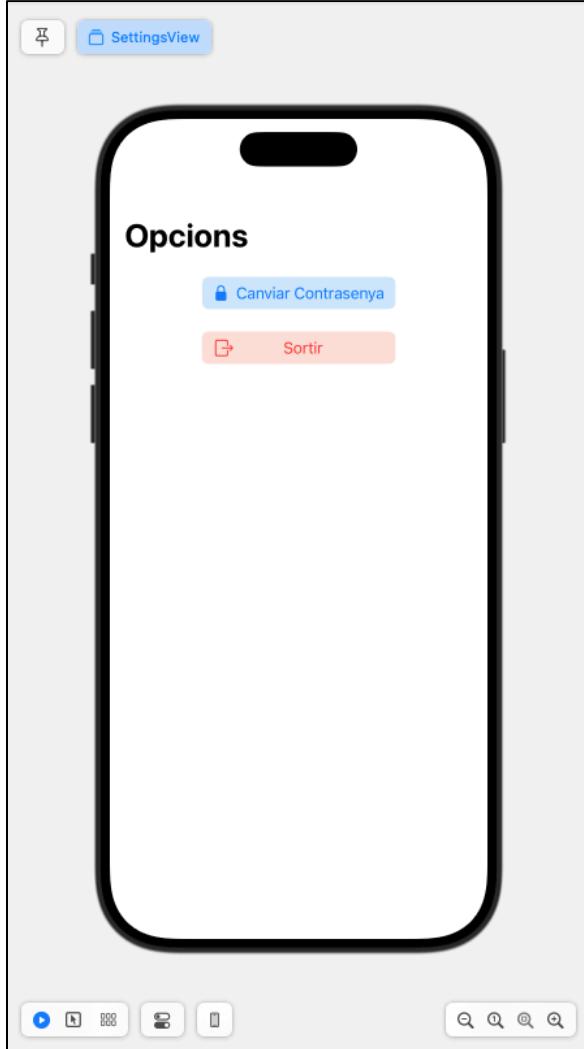
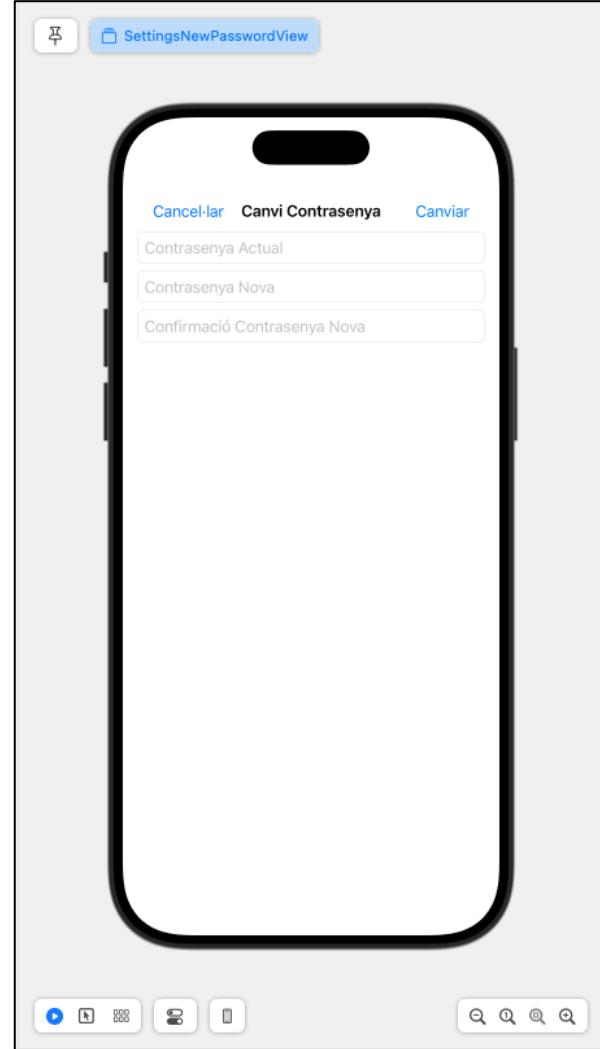


Figura 25. Vista cambiar contraseña



4.6. Firebase

Firebase es la plataforma seleccionada para el desarrollo del backend de la aplicación iACUT, debido a su integración con Xcode, su compatibilidad con iOS y sus funcionalidades en tiempo real. En esta sección se describen los principales componentes de Firebase utilizados en el proyecto: Authentication, Cloud Firestore, Storage y Reglas de Seguridad.

4.6.1. Firebase Authentication

El módulo de Firebase Authentication permite gestionar la identidad de los usuarios que acceden a la aplicación. En el caso de iACUT, se ha implementado la autenticación mediante correo electrónico y contraseña, lo que proporciona una solución segura y fácil de usar para los equipos ACUT.

Este sistema permite:

- Verificar que solo personal autorizado pueda acceder a la aplicación.
- Asociar cada acción o dato registrado con un usuario específico.
- Controlar el acceso a funciones sensibles de la aplicación según el tipo de usuario (administrador, profesional sanitario, etc.).
- Integrar métodos de autenticación más seguros, como autenticación biométrica o mediante cuenta corporativa.

4.6.2. Colecciones Cloud Firestore

Cloud Firestore es la base de datos NoSQL en tiempo real utilizada por la aplicación para almacenar y sincronizar datos entre los dispositivos y la nube.

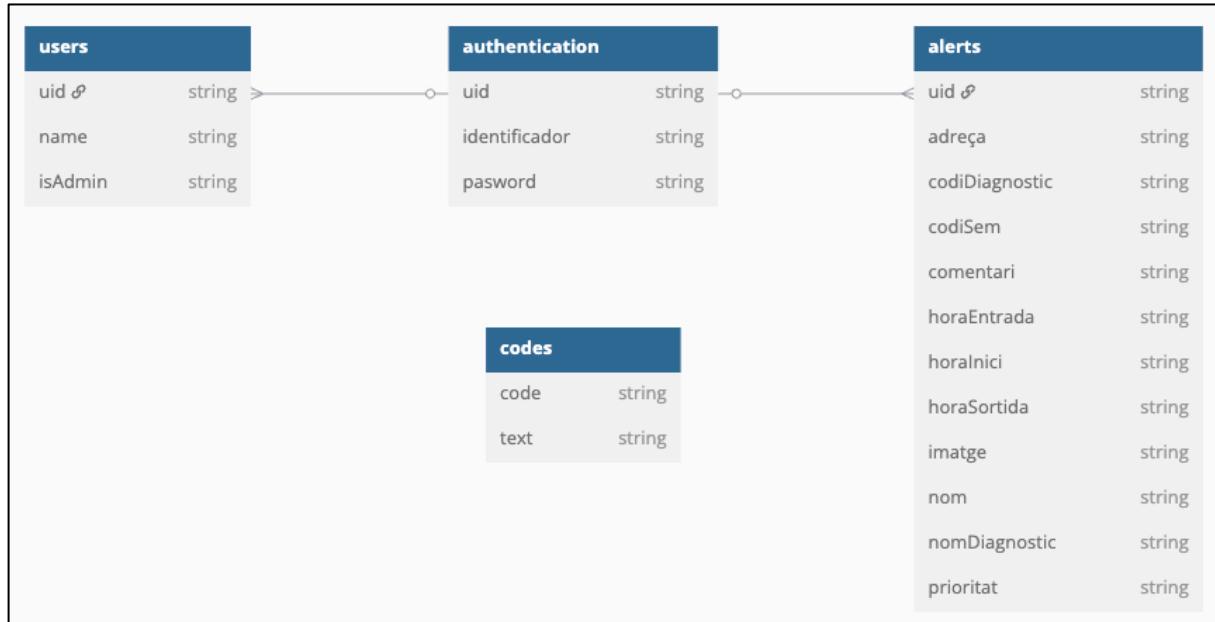
La estructura de Firestore, basada en documentos y colecciones, permite escalar fácilmente y trabajar con datos estructurados, lo que es ideal para el contexto de uso en dispositivos móviles.

En iACUT, se han creado tres colecciones clave:

- **alerts**: almacena las alertas sanitarias registradas por los usuarios.
- **users**: contiene información básica sobre los usuarios registrados en el sistema.
- **codes**: guarda códigos o Tablas de referencia que pueden ser consultadas por los profesionales (por ejemplo, tipos de incidentes, niveles de gravedad, etc.).

La Figura 26 muestra los campos de las colecciones creadas y sus relaciones.

Figura 26. colecciones de la aplicación iACUT



4.6.3. Firebase Storage

Firebase Storage permite almacenar archivos no estructurados como imágenes, documentos PDF o informes. En la aplicación iACUT, se contempla su uso para guardar archivos relacionados con alertas o documentación complementaria generada por los equipos de atención.

El uso de Firebase Storage permite:

- Adjuntar una imagen de un informe clínico.
- Guardar un archivo con instrucciones específicas ante un tipo de alerta.
- Controlar quién puede subir o descargar archivos, reforzando la seguridad y el control del acceso a la información.

4.6.4. Reglas

En el desarrollo de la aplicación iACUT, se ha implementado un conjunto de reglas de seguridad en Firebase Firestore, ver Anexo B, para garantizar que solo los usuarios autorizados puedan acceder a los datos y realizar operaciones sobre ellos. Estas reglas son esenciales para proteger la privacidad y la integridad de la información almacenada, controlando el acceso según el tipo de usuario y sus privilegios.

Las reglas establecen permisos diferenciados para tres colecciones clave dentro de la base de datos: alerts, users y codes.

1. **Colección alerts:** En esta colección, los usuarios que estén autenticados pueden realizar tanto operaciones de lectura como de escritura. Esto significa que cualquier usuario que haya iniciado sesión en la aplicación puede interactuar con los datos relacionados con las alertas sanitarias, ya sea para consultarlos o registrarlos.
2. **Colección users:** Para esta colección, las reglas restringen la escritura a los administradores del sistema, asegurando que solo personas con los privilegios adecuados puedan modificar los datos. Los usuarios pueden leer sus propios datos o los de otros usuarios únicamente si tienen permisos de administrador, lo que proporciona un control más estricto sobre la información sensible.
3. **Colección codes:** En esta colección, las reglas permiten que los usuarios autenticados solo puedan leer los datos, pero no pueden escribir ni modificar la información. Esto establece un acceso de solo lectura, garantizando que la integridad de los datos almacenados en esta colección no sea alterada.

Estas reglas también incluyen funciones auxiliares que verifican si un usuario está autenticado o si tiene permisos de administrador, simplificando la implementación de controles de acceso sin necesidad de repetir la lógica en cada regla. En conjunto, estas medidas aseguran un control preciso y seguro sobre el acceso a los datos en la aplicación.

4.7. Casos de uso

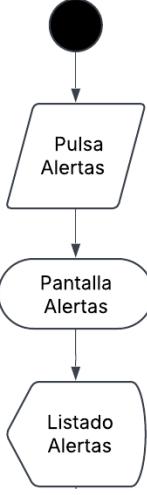
4.7.1. Acceso a la aplicación

Actor	Usuario
Descripción	Un usuario debe iniciar sesión para acceder a la aplicación.
Flujo principal	<ol style="list-style-type: none"> El usuario abre la aplicación. El sistema verifica si el usuario ya está identificado: <ul style="list-style-type: none"> Si sí, lo redirige al menú principal. Si no, lo lleva a la pantalla de autentificación. El usuario introduce su email y contraseña. El usuario pulsa el botón “Entrar” El sistema valida las credenciales. Si son correctas, el usuario accede al menú principal.
Flujo alternativo	<ol style="list-style-type: none"> Si las credenciales son incorrectas se muestra un mensaje de error. El usuario puede reintentar el inicio de sesión.
Precondición	El usuario debe estar registrado en la base de datos.
Postcondición	<p>Si el inicio de sesión es exitoso, el usuario accede al menú principal.</p> <p>Si el inicio de sesión falla, el usuario permanece en la pantalla de login con un mensaje de error.</p>
Diagrama de actividad (leyenda en Anexo A)	<pre> graph TD Start(()) --> Intro[Introducir Usuario y Contraseña] Intro --> Pulse[Pulsa Entrar] Pulse --> Validacion{Usuario Válido} Validacion -- Sí --> PantallaAlertes([Pantalla Alertes]) Validacion -- No --> MensajeError([Mensaje Error]) MensajeError --> Intro </pre>

4.7.2. Recuperación de contraseña

Actor	Usuario
Descripción	Un usuario registrado quiere recuperar la contraseña.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario pulsa en “Has oblidat la contraseña”. 2. El usuario introduce su correo corporativo. 3. El usuario pulsa “Reset”.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario pulsa “Cancelar”.
Precondición	<p>El usuario debe tener la sesión abierta. El usuario debe estar en la pantalla de autentificación.</p>
Postcondición	El usuario recibe un correo para cambiar su contraseña.
Diagrama de actividad (leyenda en Anexo A)	<pre> graph TD Start(()) --> Pulsar[Has oblidat la contraseña?] Pulsar --> Introducir[Introducir correo corporativo] Introducir --> PulsarReset[Pulsa Reset] Introducir --> PulsarCancelar[Pulsa Cancelar] PulsarReset --> Envio[Se envia correo para recuperar contraseña] Envio --> PantallaAlertas([Pantalla Alertas]) PulsarCancelar --> PantallaAlertas </pre>

4.7.3. Mostrar alertas

Actor	Usuario
Descripción	Un usuario registrado quiere ver el listado de alerta.
Flujo principal	1. El usuario pulsa en “Alertas”.
Precondición	El usuario debe estar registrado.
Postcondición	Se muestra un listado con todas las alertas del usuario.
Diagrama de actividad (leyenda en Anexo A)	 <pre>graph TD; Start(()) --> Pulse[Pulsa Alertas]; Pulse --> Pantalla{Pantalla Alertas}; Pantalla --> Listado[Listado Alertas]</pre> The diagram illustrates the flow of the activity. It begins with a solid black circle representing the initial state. An arrow points down to a rounded rectangle labeled "Pulsa Alertas". From there, another arrow points down to an oval labeled "Pantalla Alertas". Finally, an arrow points down to a hexagon labeled "Listado Alertas".

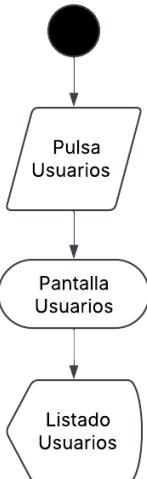
4.7.4. Mostrar detalles de una alerta

Actor	Usuario
Descripción	Un usuario registrado quiere ver el detalle de una alerta.
Flujo principal	<ol style="list-style-type: none">1. El usuario pulsa en “Alertas”.2. El usuario pulsa encima de una alerta.
Precondición	El usuario debe tener la sesión abierta.
Postcondición	Se muestra los detalles de una alerta.
Diagrama de actividad (leyenda en Anexo A)	<pre>graph TD; Start(()) --> Pulsar[\"Pulsar Alertas\"]; Pulsar --> PantallaAlertas((\"Pantalla Alertas\")); PantallaAlertas --> ListadoAlertas{\"Listado Alertas\"}; ListadoAlertas --> PulsarAlerta[\"Pulsar Alerta\"]; PulsarAlerta --> PantallaDetalleAlerta((\"Pantalla Detalle Alerta\")); PantallaDetalleAlerta --> CamposAlerta{\"Campos Alerta\"};</pre>

4.7.5. Crear una nueva alerta

Actor	Usuario administrativo
Descripción	Un usuario administrativo quiere crear una nueva alerta
Flujo principal	<ol style="list-style-type: none"> 1. El usuario pulsa en “Alertas”. 2. El usuario pulsa en el ícono de crear nueva alerta. 3. El usuario rellena los campos de datos. 4. El usuario pulsa en “Guardar”.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario pulsa “Cancelar”.
Precondición	<p>El usuario debe tener la sesión abierta. El usuario debe tener el rol administrativo.</p>
Postcondición	Se crea una nueva alerta.
Diagrama de actividad (leyenda en Anexo A)	<pre> graph TD Start(()) --> PulsaAlertas[Pulsa Alertas] PulsaAlertas --> PantallaAlertas{Pantalla Alertas} PantallaAlertas --> ListadoAlertas[Listado Alertas] ListadoAlertas --> PulsaIcono[Pulsa icono nueva alerta] PulsaIcono --> PantallaNuevaAlerta{Pantalla Nueva Alerta} PantallaNuevaAlerta --> RellenarCampos[Rellenar Campo de datos] RellenarCampos --> PulsarGuardar[Pulsar Guardar] PantallaAlertas --> PulsarCancelar[Pulsar Cancelar] </pre>

4.7.6. Mostrar usuarios

Actor	Usuario administrativo
Descripción	Un usuario administrativo quiere ver todos los usuarios.
Flujo principal	1. El usuario pulsa en “Usuarios”.
Precondición	El usuario debe estar registrado. El usuario debe tener el rol administrativo.
Postcondición	Se muestra un listado con todos los usuarios.
Diagrama de actividad (leyenda en Anexo A)	 <pre>graph TD; Start(()) --> Click[\"Pulsa Usuarios\"]; Click --> Screen([\"Pantalla Usuarios\"]); Screen --> List([\"Listado Usuarios\"]);</pre>

4.7.7. Mostrar detalles de un usuario

Actor	Usuario administrativo
Descripción	Un usuario administrativo quiere ver el detalle de un usuario.
Flujo principal	<ol style="list-style-type: none">1. El usuario pulsa en “Usuario”.2. El usuario pulsa encima de un usuario.
Precondición	El usuario debe tener la sesión abierta. El usuario debe tener el rol administrativo.
Postcondición	Se muestra los detalles de un usuario.
Diagrama de actividad (leyenda en Anexo A)	

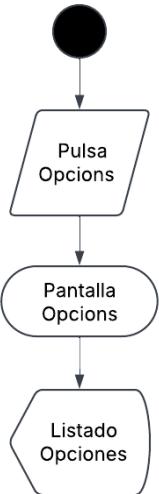
4.7.8. Crear un nuevo usuario

Actor	Usuario administrativo
Descripción	Un usuario administrativo quiere crear un nuevo usuario.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario pulsa en “Usuarios”. 2. El usuario pulsa en el ícono de crear nuevo usuario. 3. El usuario rellena los campos de datos. 4. El usuario pulsa en “Guardar”.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario pulsa “Cancelar”.
Precondición	<p>El usuario debe tener la sesión abierta. El usuario debe tener el rol administrativo.</p>
Postcondición	Se crea una nueva alerta.
Diagrama de actividad (leyenda en Anexo A)	<pre> graph TD Start(()) --> PulsaUsuarios[Pulsa Usuarios] PulsaUsuarios --> PantallaUsuarios([Pantalla Usuarios]) PantallaUsuarios --> ListadoUsuarios{Listado Usuarios} ListadoUsuarios --> PulsaIcono[Pulsa icono nuevo usuario] PulsaIcono --> PantallaNuevoUsuario([Pantalla Nuevo Usuario]) PantallaNuevoUsuario --> Rellenar[Rellenar Campo de datos] Rellenar --> PulsarGuardar[Pulsar Guardar] PulsarGuardar --> PulsaUsuarios PantallaNuevoUsuario --> PulsarCancelar[Pulsar Cancelar] </pre>

4.7.9. Buscar código de diagnóstico

Actor	Usuario
Descripción	Un usuario registrado quiere ver el listado de códigos.
Flujo principal	<ol style="list-style-type: none">1. El usuario pulsa en “Codis”.2. El usuario escribe parte del nombre del código.
Precondición	El usuario debe tener la sesión abierta.
Postcondición	Se muestra un listado con todos los códigos disponibles.
Diagrama de actividad (leyenda en Anexo A)	<p>El diagrama de actividad muestra el siguiente flujo:</p> <ul style="list-style-type: none">Un punto de inicio (attività) conduce a un cuadro rectangular rotulado "Pulsa Codis".Este cuadro conduce a un óvalo rotulado "Pantalla Codis".Este óvalo conduce a un hexágono rotulado "Listado Códigos".Este hexágono conduce a un cuadro rectangular rotulado "Escribe Código".Existe una retroalimentación (bucle) que vuelve desde el final de "Escribe Código" de vuelta al principio de "Pulsa Codis".

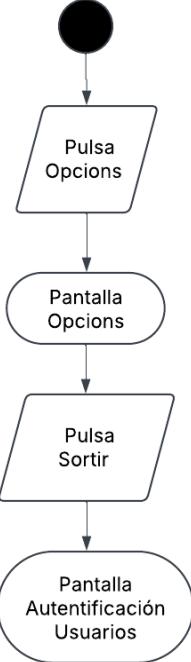
4.7.10. Mostrar opciones de usuario

Actor	Usuario
Descripción	Un usuario registrado quiere ver las opciones de usuario.
Flujo principal	1. El usuario pulsa en “Opcions”.
Precondición	El usuario debe tener la sesión abierta.
Postcondición	Se muestran las opciones de usuario.
Diagrama de actividad (leyenda en Anexo A)	 <pre>graph TD; Start(()) --> Pulse[Pulsa Opcions]; Pulse --> Pantalla((Pantalla Opcions)); Pantalla --> Listado[Listado Opciones]</pre>

4.7.11. Cambiar contraseña de usuario

Actor	Usuario.
Descripción	Un usuario registrado quiere ver las opciones de usuario.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario pulsa en “Opcions”. 2. El usuario pulsa “Canvi de contrasenya”. 3. El usuario escribe la contraseña actual. 4. El usuario escribe la nova contraseña. 5. El usuario escribe la confirmación de la contraseña. 6. El usuario pulsa “Canviar”.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario pulsa “Cancel·lar”.
Precondición	El usuario debe tener la sesión abierta.
Postcondición	Se cambia la contraseña del usuario.
Diagrama de actividad (leyenda en Anexo A)	<pre> graph TD Start(()) --> PulsarOpciones[Pulsa Opcions] PulsarOpciones --> PantallaOpciones{Pantalla Opcions} PantallaOpciones --> PulsarCanvi[\"Pulsa Canvi de contrasenya\"] PulsarCanvi --> PantallaCanvi{Pantalla Canvi Contraseña} PantallaCanvi --> Introducir[\"Introduce Contraseña Actual Contraseña Nueva Confirmación Contraseña\"] Introducir --> PulsarCanviar[Pulsa Canviar] </pre> <p>Este diagrama de actividad ilustra el flujo principal para cambiar la contraseña. Comienza con un inicio (nodo final). Luego, se pulsa "Opcions" (nodo rectangular). Se muestra la "Pantalla Opcions" (nodo oval). Luego, se pulsa "Canvi de contrasenya" (nodo rectangular). Se muestra la "Pantalla Canvi Contraseña" (nodo oval). Finalmente, se introduce la contraseña actual, nueva contraseña y confirmación (nodo rectangular). Por último, se pulsa "Canviar" (nodo rectangular).</p>

4.7.12. Cerrar sesión de usuario

Actor	Usuario.
Descripción	Un usuario registrado quiere ver las opciones de usuario.
Flujo principal	<ol style="list-style-type: none">1. El usuario pulsa en “Opcions”.2. El usuario pulsa “Sortir”.
Precondición	El usuario debe tener la sesión abierta.
Postcondición	Se cierra la sesión de usuario.
Diagrama de actividad (leyenda en Anexo A)	 <pre>graph TD; Start(()) --> PulsarOpciones[Pulsa Opcions]; PulsarOpciones --> PantallaOpciones((Pantalla Opcions)); PantallaOpciones --> PulsarSortir[Pulsa Sortir]; PulsarSortir --> PantallaAutentificacion((Pantalla Autentificación Usuarios));</pre> <p>El diagrama de actividad muestra el flujo de acciones para cerrar la sesión de usuario. Comienza con un nodo inicial (nodo vacío). Luego, se ejecuta la acción “Pulsa Opcions” (nodo rombo). Esto lleva a la “Pantalla Opcions” (nodo óvalo). Finalmente, se realiza la acción “Pulsa Sortir” (nodo rombo), lo que resulta en la “Pantalla Autentificación Usuarios” (nodo óvalo).</p>

4.8. Ejemplo práctico de Uso

El siguiente ejemplo ilustra el flujo de trabajo de la aplicación iACUT durante una intervención real de los equipos ACUT del GAPiC Alt Pirineu. Se simula una alerta sanitaria recibida desde el SEM para demostrar la eficiencia del sistema en condiciones operativas reales.

4.8.1. Recepción de la alerta

Actor:

Usuario administrativo.

Acciones:

- El administrativo recibe un aviso del SEM a través del sistema institucional, Figura 27.
- El administrativo crea una alerta en iACUT desde su dispositivo iOS corporativo, Figura 28.

Figura 27. Aplicación del SEM

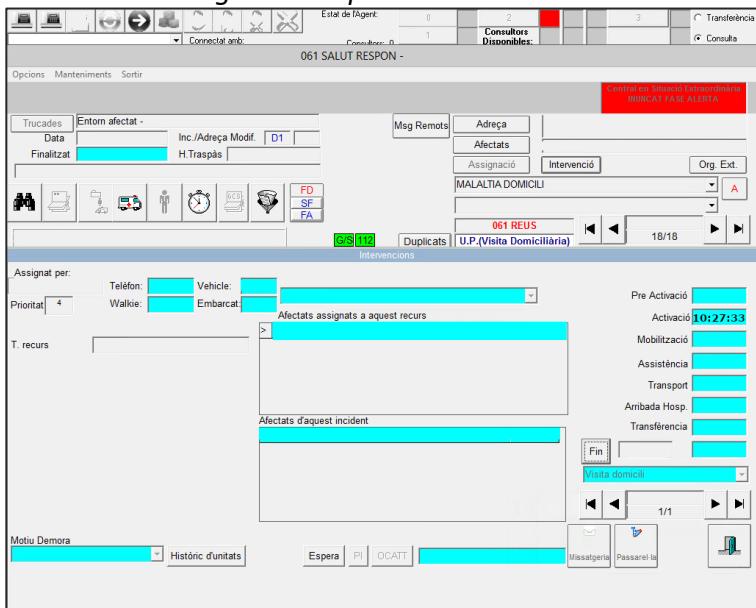
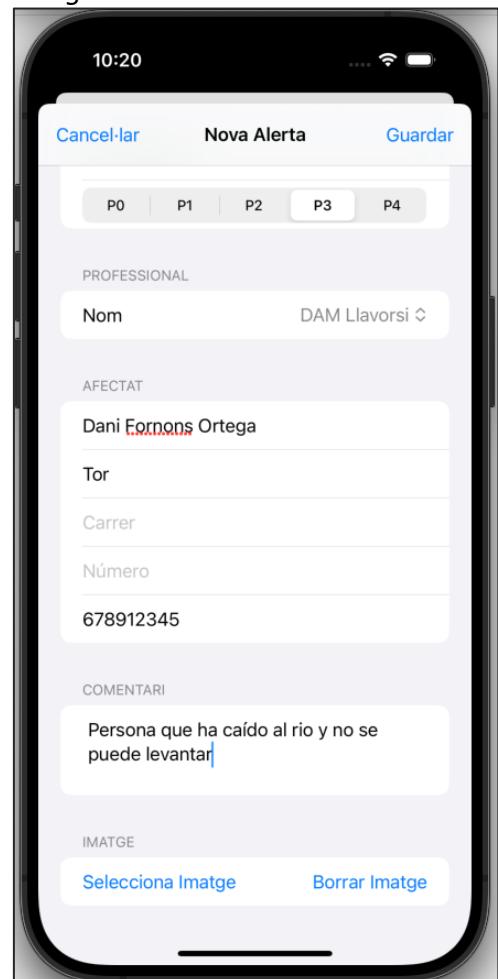


Figura 28. Creación de una alerta



4.8.2. Activación del personal sanitario

Actor:

- Usuario sanitario.
- Usuario administrativo.

Acciones:

- El personal sanitario consulta los datos de la alerta y registra la hora de inicio en la aplicación iACUT, Figura 29.
- El personal administrativo consulta la hora en iACUT y la registra en la aplicación del SEM, Figura 30.
- El personal sanitario usa el navegador del móvil para ir al lugar de la alerta, Figura 31.

Figura 29. Consulta de la alerta

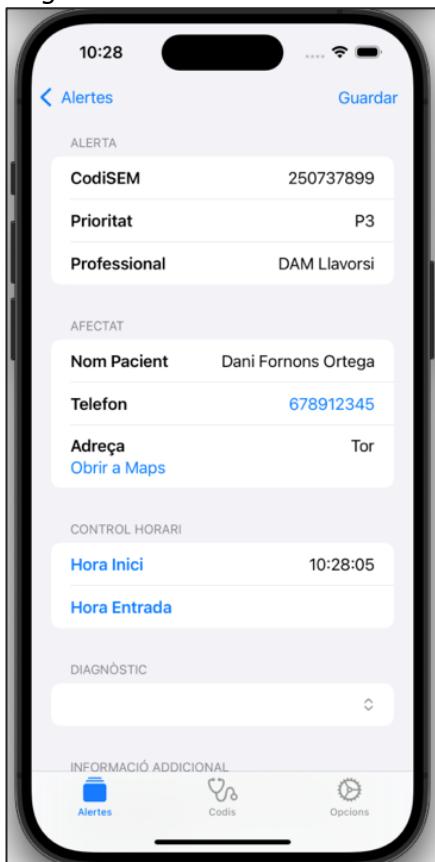


Figura 30. Registro hora movilización del SEM

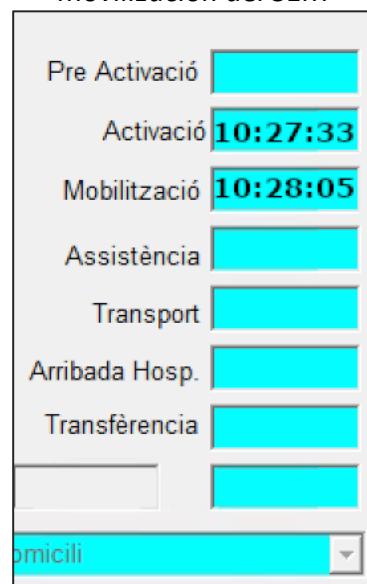


Figura 31. Navegación hacia el lugar de la alerta



4.8.3. Intervención del personal sanitario

Actor:

- Usuario sanitario.
- Usuario administrativo.

Acciones:

- El personal sanitario llega al lugar de la alerta y registra la hora en el iACUT, Figura 32.
- El personal administrativo registra la hora de llegada en la aplicación del SEM, Figura 33.

Figura 32. Registro hora de entrada al domicilio

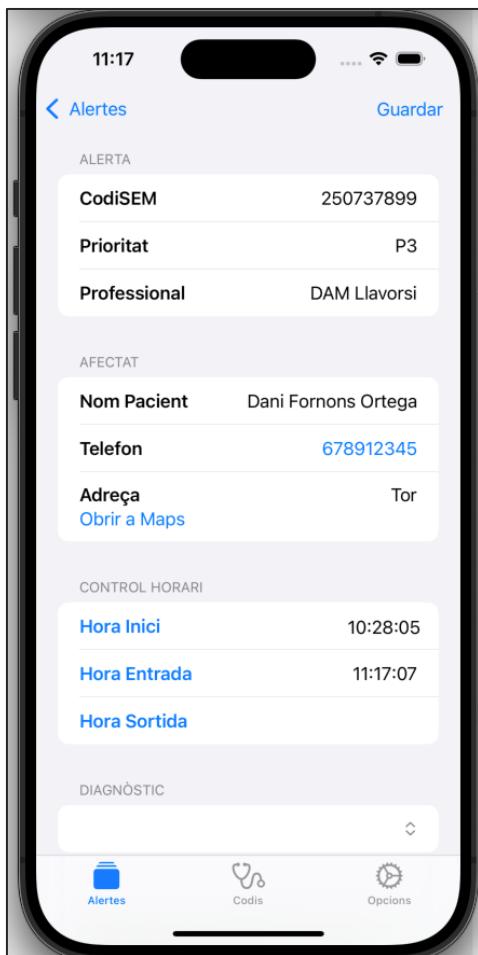
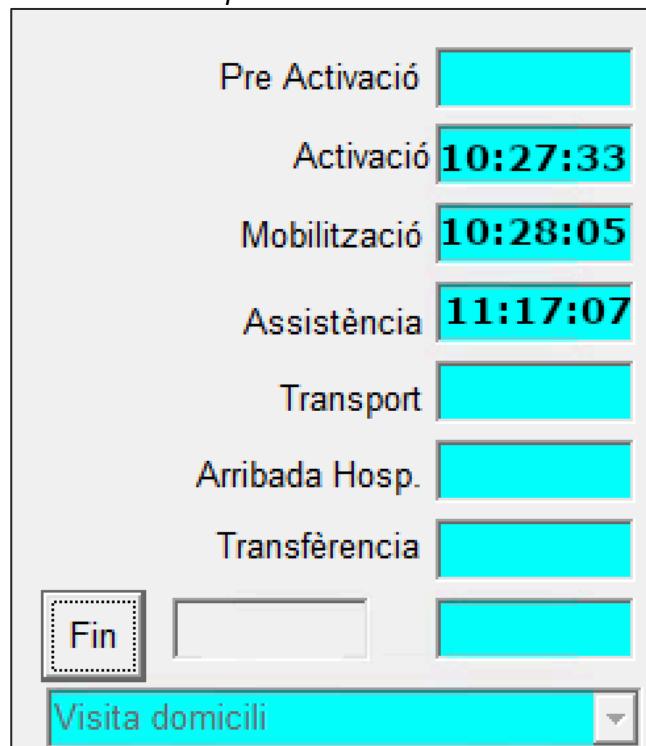


Figura 33. Registro hora de asistencia en la aplicación del SEM



4.8.4. Cierre de la alerta

Actor:

- Usuario sanitario.
- Usuario administrativo.

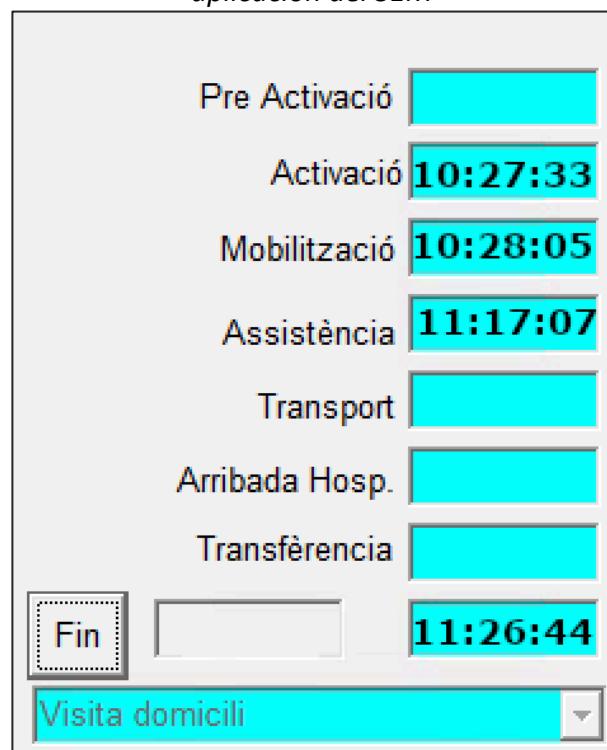
Acciones:

- El personal sanitario realiza la intervención sanitaria y al finalizar registra la hora de salida y el código de la patología en la aplicación del SEM, Figura 34.
- El personal administrativo registra la hora de finalización en la aplicación del SEM, Figura 35.

Figura 34. Registro de la hora de salida y del diagnóstico



Figura 35. Registro finalización en la aplicación del SEM



4.9. Prueba piloto

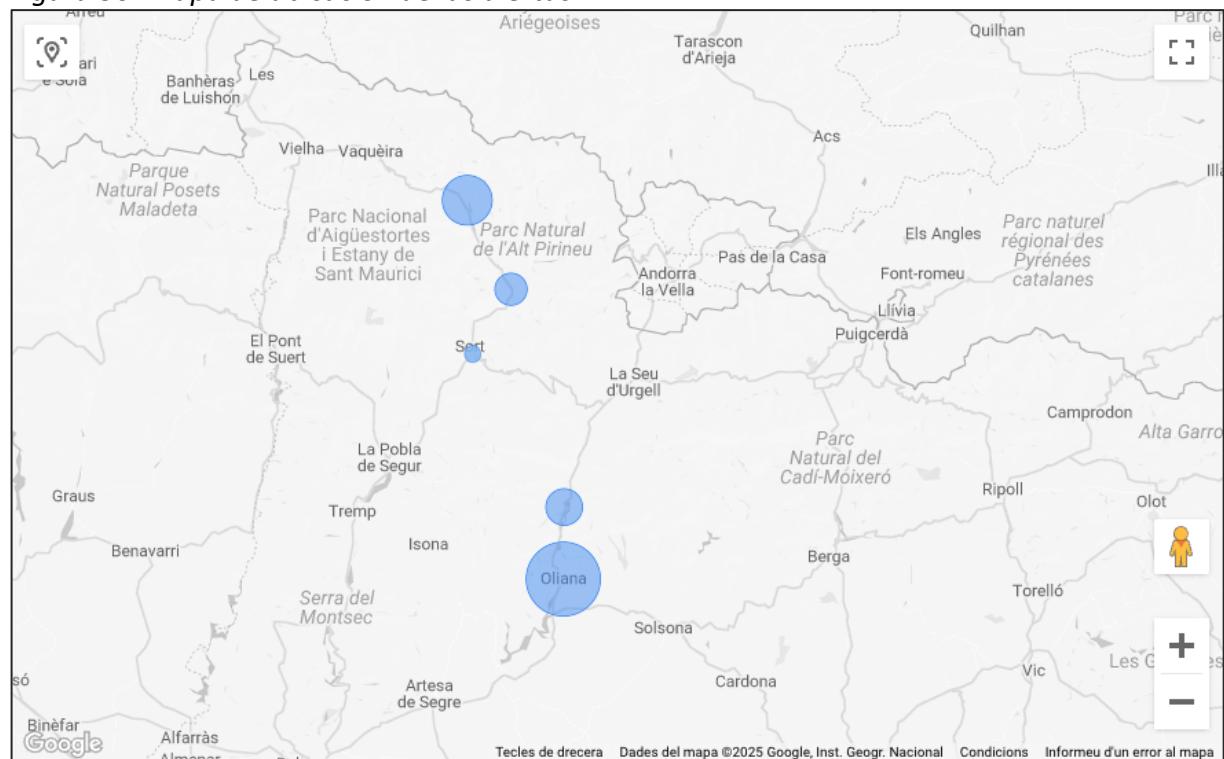
Durante tres semanas se realizó una prueba piloto con los equipos DAM del PAC de Sort. En la prueba intervinieron 5 equipos, Figura 36, y se realizaron 68 actuaciones, ver Tabla 6.

No se detectaron incidencias con la aplicación iACUT y se obtuvo un porcentaje de registro del 100%.

Tabla 6. Resultados de la prueba piloto

Equipo	Número actuaciones	Registro completado
DAM Esterri d'Aneu	17	100%
DAM Llavorsí	9	100%
DAM Oliana	28	100%
DAM Organyà/Coll de Nargó	11	100%
PAC Sort	3	100%
TOTAL	68	100%

Figura 36. Mapa de ubicación de las alertas



5. Conclusiones y trabajo futuro

Este trabajo ha demostrado cómo las tecnologías móviles pueden tener un impacto significativo en la mejora de los procesos de atención sanitaria, especialmente en entornos críticos como las alertas sanitarias. La aplicación iACUT constituye un paso importante hacia la modernización de los sistemas de gestión de emergencias en el GAPiC Alt Pirineu y su implementación podría ser un modelo que seguir en otros territorios.

5.1. Conclusiones del trabajo

El desarrollo de la aplicación iACUT ha logrado los objetivos propuestos, proporcionando una herramienta eficaz para los profesionales del ACUT durante situaciones de alerta sanitaria.

Entre los logros más destacados se incluyen:

1. Optimización en el registro de incidencias:

El registro de incidencias ha sido simplificado y centralizado, lo que permite un seguimiento más preciso de las intervenciones.

2. Seguridad y privacidad:

Se han implementado medidas de seguridad robustas para proteger los datos sensibles de los pacientes y garantizar el cumplimiento de las normativas de protección de datos.

3. Facilidad de uso:

La interfaz ha sido bien recibida por los usuarios de la prueba piloto, quienes han destacado su eficacia en situaciones de alta presión. Además, los participantes expresaron mayor satisfacción al utilizar una aplicación corporativa en un dispositivo proporcionado por la empresa, en lugar de tener que emplear sus equipos personales. Esto refuerza la comodidad y seguridad en el entorno laboral.

4. Mejora en la coordinación y comunicación:

La aplicación ha mejorado la comunicación en tiempo real entre los miembros del equipo llegando a obtener un 100% de datos correctamente registrados durante la prueba piloto.

5.2. Líneas de trabajo futuro

El desarrollo de iACUT representa un primer paso significativo en la mejora de la comunicación y gestión de alertas sanitarias en el GAPIC Alt Pirineu. Sin embargo, existen numerosas oportunidades para expandir y mejorar la aplicación en futuras iteraciones:

1. Expansión a plataforma Android

Aunque actualmente los equipos ACUT utilizan dispositivos Apple, una versión para Android ampliaría la accesibilidad de la aplicación. Esto podría implementarse utilizando frameworks multiplataforma como Flutter o React Native, o mediante desarrollo nativo en Kotlin, evaluando cuidadosamente el equilibrio entre rendimiento, experiencia de usuario y esfuerzo de desarrollo.

2. Integración con sistemas de información hospitalaria

La conexión directa con los sistemas de Historia Clínica Electrónica y otros sistemas hospitalarios permitiría:

- Acceso inmediato a información relevante del paciente durante emergencias.
- Registro automático de intervenciones en la historia clínica.
- Reducción de la duplicación de datos y posibles errores de transcripción.

3. Implementación de inteligencia artificial (IA)

La incorporación de algoritmos de IA podría mejorar significativamente la aplicación mediante:

- Sistemas de selección automatizado para priorizar alertas.
- Predicción de recursos necesarios basada en patrones históricos.
- Asistentes virtuales para guiar a los profesionales en protocolos de actuación.
- Análisis predictivo de zonas con mayor incidencia de emergencias.

4. Funcionalidades avanzadas de geolocalización

Mejorar las capacidades de geolocalización permitiría:

- Seguimiento en tiempo real de los equipos sanitarios.
- Optimización de rutas basada en condiciones de tráfico y climatología.
- Identificación del profesional más cercano a una emergencia.
- Mapeo de recursos sanitarios disponibles en la zona.

5. Mejoras en la experiencia de usuario

Basándose en la retroalimentación de los usuarios, se podrían implementar:

- Personalización de la interfaz según preferencias del usuario.
- Modos de visualización adaptados a diferentes condiciones (día/noche).
- Comandos por voz para uso manos libres durante emergencias.
- Mejoras de accesibilidad para usuarios con diversidad funcional.

6. Incorporación de informes y estadísticas

Se propone desarrollar un módulo de generación de informes y estadísticas personalizadas a partir de las intervenciones registradas en la aplicación. Esto permitiría realizar análisis de las alertas facilitando la evaluación del servicio, la planificación de recursos y la toma de decisiones estratégicas en el ámbito sanitario.

7. Ampliación a otros servicios sanitarios

El modelo de iACUT podría adaptarse a otros servicios como:

- Equipos de atención domiciliaria.
- Servicios de transporte sanitario.
- Coordinación entre diferentes niveles asistenciales.

La implementación de estas líneas de trabajo futuro requerirá una evaluación continua de las necesidades de los usuarios, los avances tecnológicos disponibles y las prioridades estratégicas del GAPiC Alt Pirineu. El enfoque ágil adoptado durante el desarrollo inicial de iACUT proporciona un marco adecuado para incorporar estas mejoras de manera iterativa, asegurando que la aplicación evolucione para satisfacer las necesidades cambiantes del sistema sanitario.

Referencias bibliográficas

- Boulos, M. N. K., Brewer, A. C., Karimkhani, C., Buller, D. B., & Dellavalle, R. P. (2014). Mobile medical and health apps: State of the art, concerns, regulatory control and certification. *Online Journal of Public Health Informatics*, 5(3), e61275.
<https://doi.org/10.5210/ojphi.v5i3.4814>
- eHealth, W. G. O. for. (2011). *mHealth: New horizons for health through mobile technologies: second global survey on eHealth*. World Health Organization.
<https://iris.who.int/handle/10665/44607>
- Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales, Pub. L. No. Ley Orgánica 3/2018, BOE-A-2018-16673 119788 (2018). <https://www.boe.es/eli/es/lo/2018/12/05/3>
- Madaminov, U. A., & Allaberganova, M. R. (2023). Firebase Database Usage and Application Technology in Modern Mobile Applications. *2023 IEEE XVI International Scientific and Technical Conference Actual Problems of Electronic Instrument Engineering (APEIE)*, 1690-1694. <https://doi.org/10.1109/APEIE59731.2023.10347828>
- Markets, R. and. (2025, enero 14). *MHealth Apps Market Research and Forecasts 2024-2032: Expansion into Emerging Markets, Focus on Chronic Disease Management, Emphasis on Preventive Healthcare*. GlobeNewswire News Room.
<https://www.globenewswire.com/news-release/2025/01/14/3009044/28124/en/MHealth-Apps-Market-Research-and-Forecasts-2024-2032-Expansion-into-Emerging-Markets-Focus-on-Chronic-Disease-Management-Emphasis-on-Preventive-Healthcare.html>

Martínez-Pérez, B., de la Torre-Díez, I., & López-Coronado, M. (2014). Privacy and Security in Mobile Health Apps: A Review and Recommendations. *Journal of Medical Systems*, 39(1), 181. <https://doi.org/10.1007/s10916-014-0181-3>

Mosa, A. S. M., Yoo, I., & Sheets, L. (2012). A Systematic Review of Healthcare Applications for Smartphones. *BMC Medical Informatics and Decision Making*, 12(1), 67.
<https://doi.org/10.1186/1472-6947-12-67>

(PDF) Native or Web or Hybrid which is better for Mobile Application. (2024). *ResearchGate*.
<https://doi.org/10.16949/TURKBILMAT.56034>

Real-time Database Synchronization Techniques in Firebase for Mobile App Development – IJSREM. (s. f.). Recuperado 1 de abril de 2025, de <https://ijsrem.com/download/real-time-database-synchronization-techniques-in-firebase-for-mobile-app-development/>

Reglamento general de protección de datos (RGPD) / EUR-Lex. (s. f.). Recuperado 10 de abril de 2025, de https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=legissum:310401_2

Ventola, C. L. (2014). Mobile Devices and Apps for Health Care Professionals: Uses and Benefits. *Pharmacy and Therapeutics*, 39(5), 356-364.

Anexo A. Leyenda del diagrama de actividad



Inicio

Se utiliza para indicar el inicio del diagrama. Sólo puede salir una línea de flujo.

Final

Se utiliza para indicar el final del diagrama. Sólo puede entrar una línea de flujo.

Acción / Proceso General

Indica una acción o instrucción general que debe realizar la aplicación.

Entrada por teclado

Instrucción de entrada de datos por teclado.

Indica que la aplicación debe esperar a que el usuario introduzca un dato que se guardará en una variable.

Decisión

Indica la comparación de dos datos y dependiendo del resultado lógico (falso o verdadero) se toma la decisión de seguir un camino u otro.

Salida de Pantalla

Instrucción de presentación de mensajes o resultados por pantalla.

Anexo B. Reglas de Firebase Firestore

La Figura 37 muestra el conjunto de reglas de acceso utilizadas durante la prueba piloto.

Figura 37. Reglas Firebase Firestore

```
1  rules_version = '2';
2  service cloud.firestore {
3      match /databases/{database}/documents {
4
5          // Reglas para la colección 'alerts'
6          match /alerts/{alertId} {
7              allow read, write: if isAuthenticated();
8          }
9
10         // Reglas para la colección 'users'
11         match /users/{userId} {
12             allow read: if isAdmin() || request.auth.uid == userId;
13             allow write: if isAdmin();
14         }
15
16         // Reglas para la colección 'codes'
17         match /codes/{codeId} {
18             allow read: if isAuthenticated();
19             allow write: if false; // Solo lectura para todos los usuarios
20         }
21
22         // Función auxiliar: verifica si el usuario está autenticado
23         function isAuthenticated() {
24             return request.auth != null;
25         }
26
27         // Función auxiliar: verifica si el usuario es administrativo
28         function isAdmin() {
29             return isAuthenticated() &&
30                 get(/databases/$(database)/documents/users/${request.auth.uid}).data.isAdmin == true;
31         }
32     }
33 }
34 }
```

Anexo C. Presentación reunión inicial



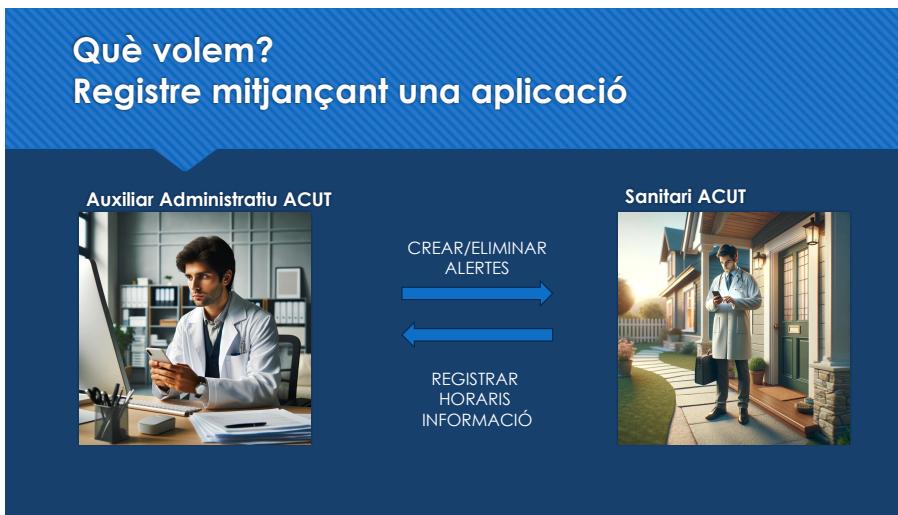
Què fem actualment? Registre mitjançant el Whatsapp

Auxiliar Administratiu ACUT

- Reben Alerta
- Truquen al Sanitari per notificar alerta
- Passen codi alerta al Whatsapp
- Registren horaris
- Registren codi de malaltia

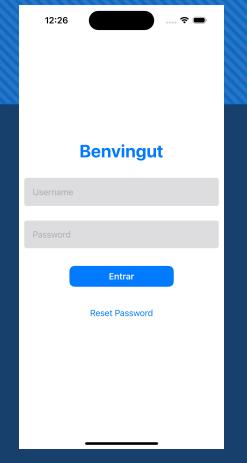
Sanitari ACUT

- Passen hora d'inici al Whatsapp
- Passen hora d'arribada al domicili al Whatsapp
- Passen hora de sortida del domicili al Whatsapp
- Passen codi de malaltia al Whatsapp



Què volem? Aplicació ACUTiX

- Una aplicació multiplataforma, Android e iOS
- Accés mitjançant usuari i password
- Usuaris amb rol administratiu i rol sanitari



Què volem? Rol administratiu

- Permet gestionat usuaris, tan administratiu com sanitaris
- Permet gestionar alertes, crear i eliminar-ne
- Permet gestionar el seu propi usuari

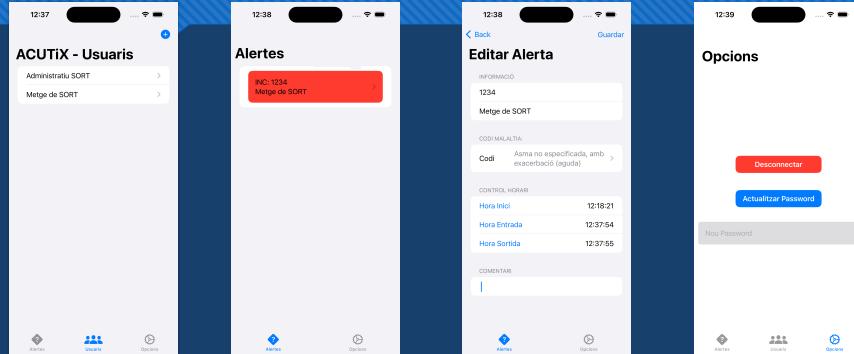


Què volem? Aplicació Sanitari

- Permet registrar horaris: inici, entrada i sortida del domicili
- Permet enviar codi de malaltia i comentari
- Permet gestionar el propi usuari



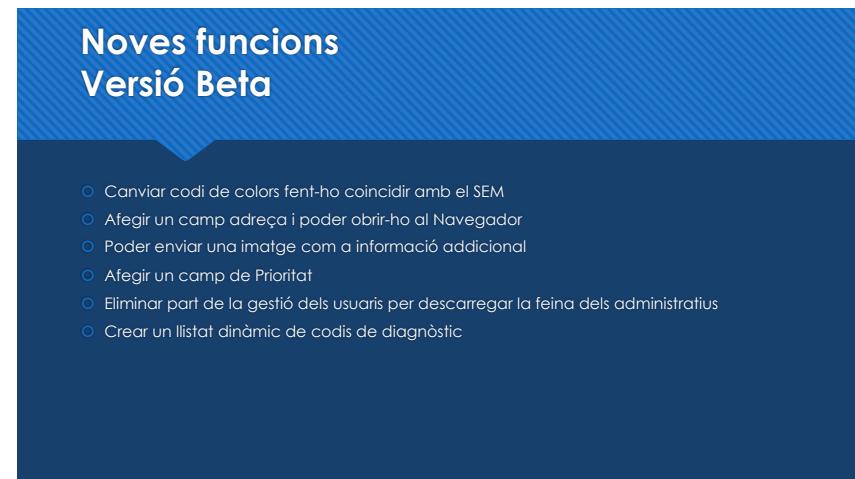
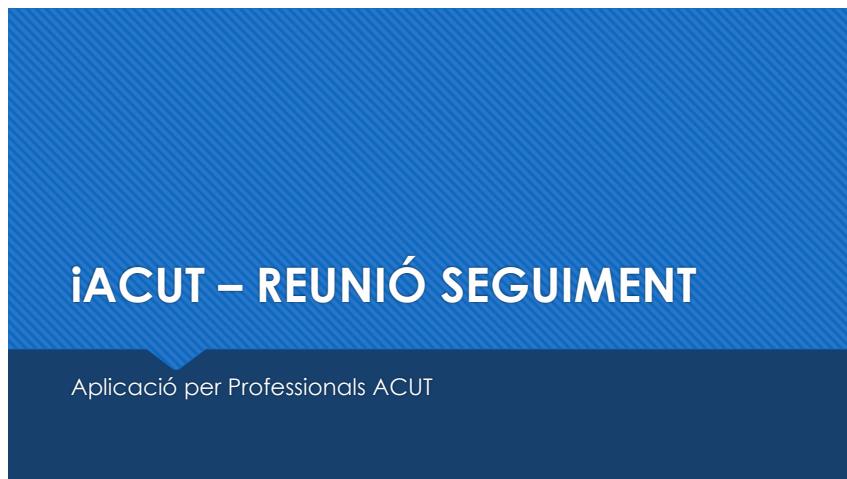
Què volem? Aplicació iACUT

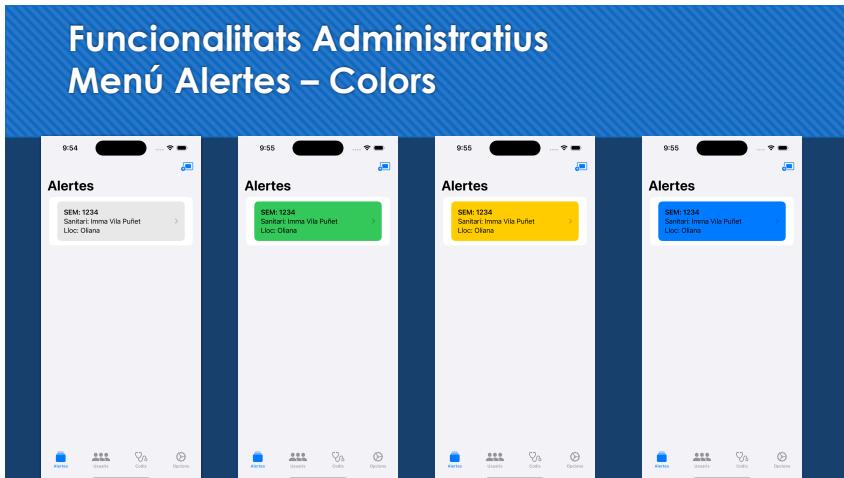
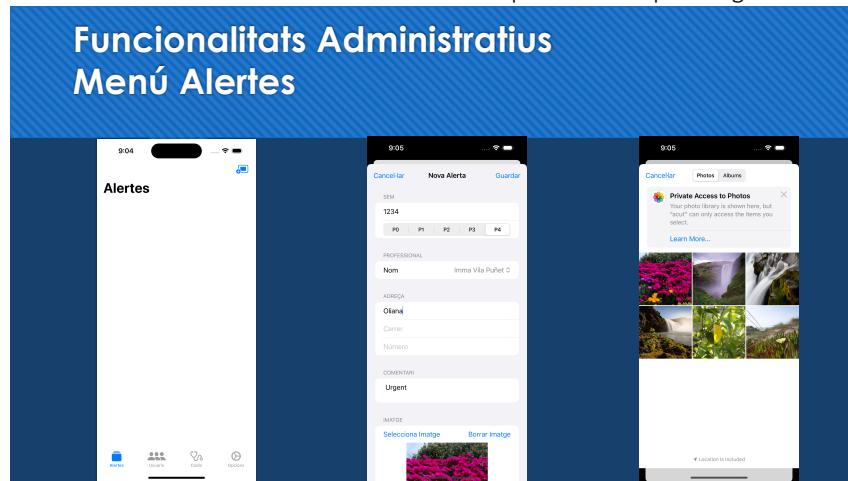
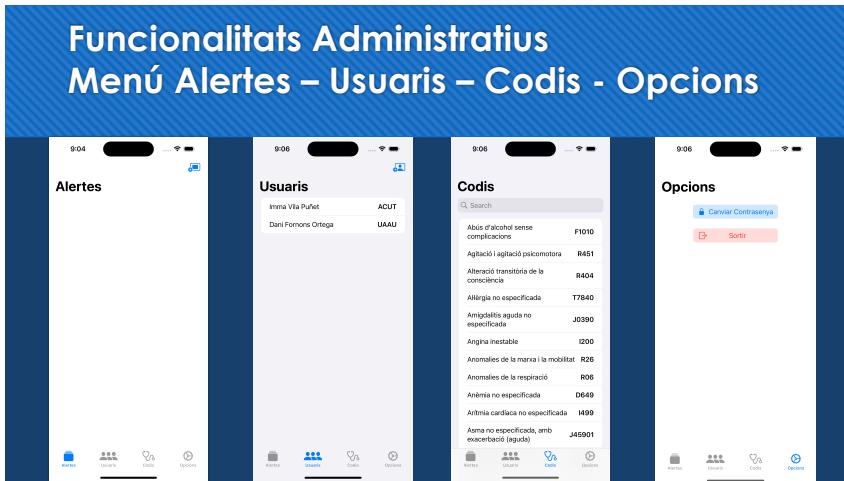


Què falta?

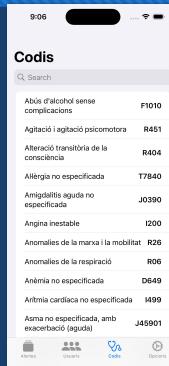
- Cal decidir si aquesta proposta s'ajusta al que necessitem
- Cal decidir quines funcionalitats volem afegir o treure
- Cal crear les aplicació

Anexo D. Presentación reunión de seguimiento

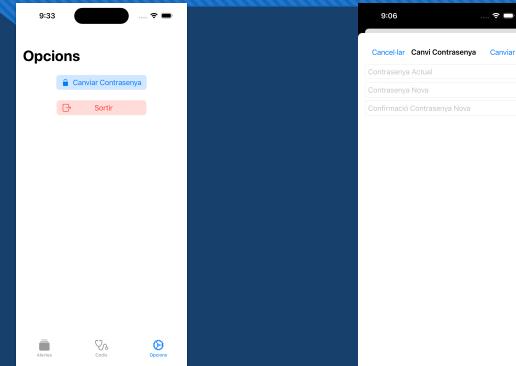




Funcionalitats Administratius Menú Codis

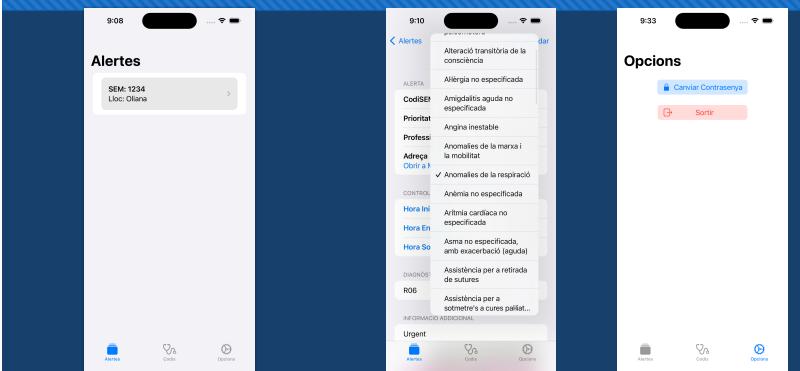


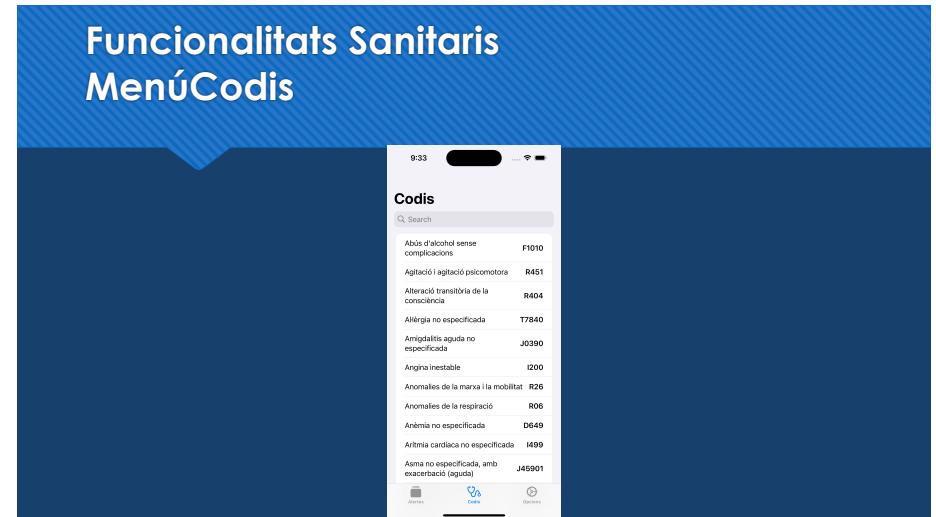
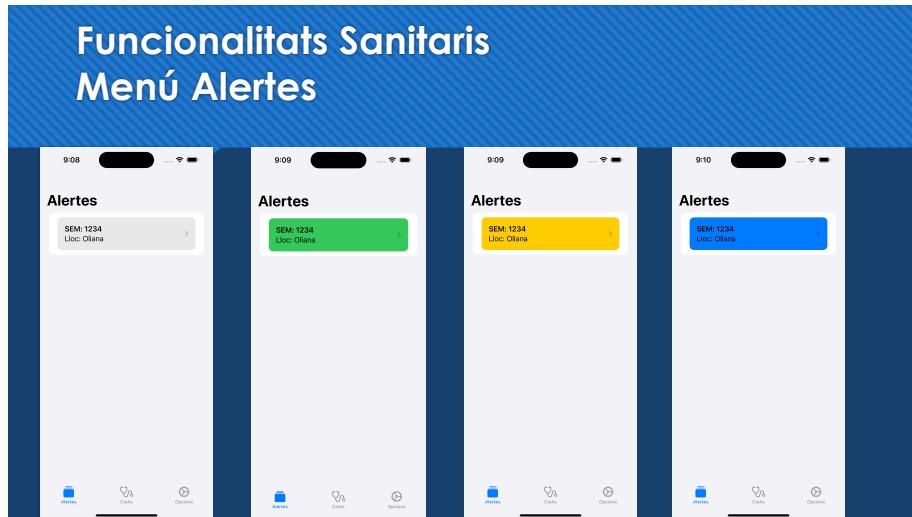
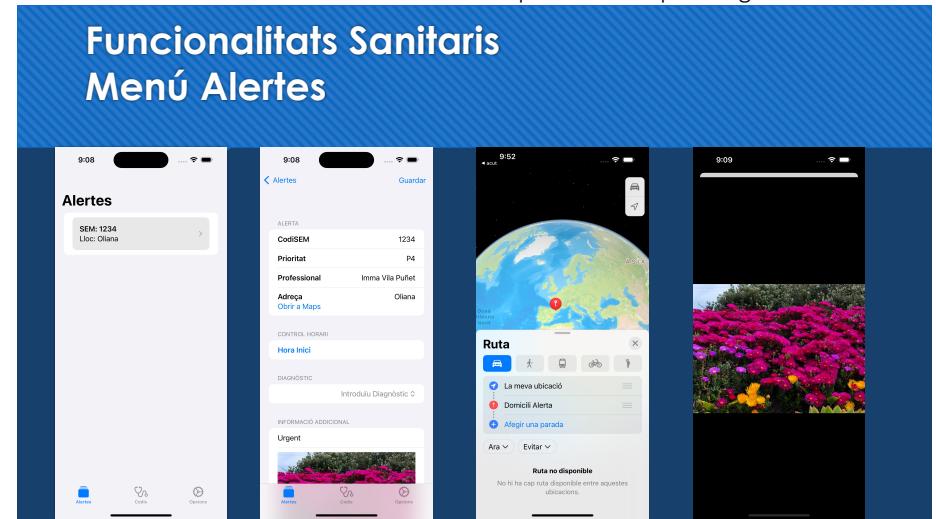
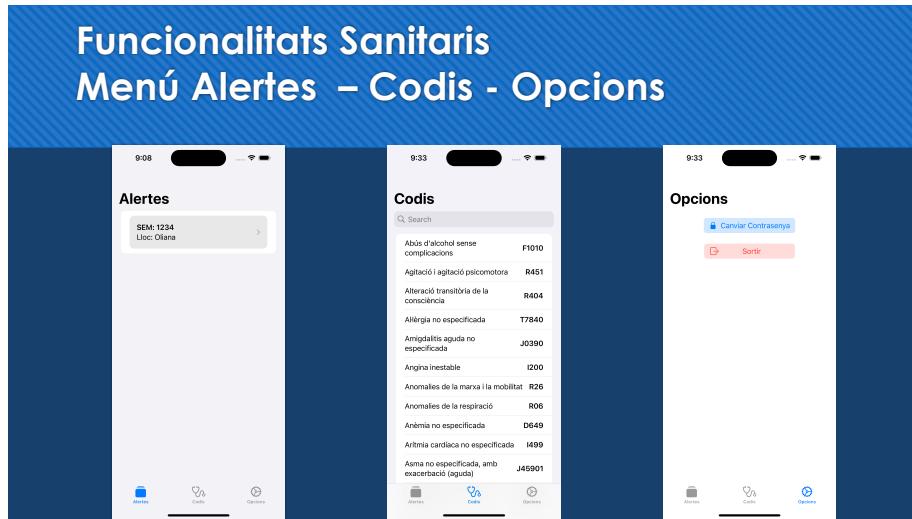
Funcionalitats Administratius Menú Opcions - Sortir i Canvi Contrassenya



Versió Beta Vista Sanitaris

Funcionalitats Sanitaris Menú Alertes – Usuaris - Opcions







Què falta? Versió Beta

- Cal modificar l'aplicació per a iOS
 - Traduir els missatges d'error
 - Afegir animacions
- Cal modificar les regles de seguretat de la base de dades per a producció
- Cal crear la documentació de l'aplicació
- Cal decidir com distribuir les apps als usuaris

Índice de acrónimos

ACUT	Atenció Continuada i Urgent en el Territori. Atención Continuada y Urgente en el Territorio.
CAP	Centre d'Atenció Primària . Centro de Atención Primaria.
Catsalut	Servei Català de la Salut. Servicio Catalán de la Salud.
DAM	Dispositiu Aïllat de Muntanya. Dispositivo Aislado de Montaña.
GAPIC Alt Pirineu	Gerència d'Atenció Primària i a la Comunitat de l'Alt Pirineu i Aran. Gerencia de Atención Primaria y a la Comunidad del Alto Pirineo y Arán.
ICS	Institut Català de la Salut. Instituto Catalán de la Salud.
IDE	Integrated Development Environment. Entorno de Desarrollo Integrado .
LOPDGDD	Ley de Protección de Datos y Garantía de Derechos Digitales.
PAC	Punt Atenció Continuada. Punto de Atención Continuada.
RPGD	Reglamento General de Protección de Datos .
SEM	Servei d'Emergències Mèdiques. Servicio de Emergencias Médicas.
UAAU	Unitat d'Atenció Als Usuaris. Unidad de Atención A los Usuarios.