



Family Care

El asistente en el cuidado de tu familia



Alumno: Alejandro Cimadevilla Estradera

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Consultor/a: Roger Montserrat Ribes

Profesor/a responsable de la asignatura: Carles Garrigues Olivella

Fecha de Entrega: 3 de Enero de 2018

A Alma y a Pablo, prisma multicolor por el que asomarse a la vida.

A MacPich, brother in arms in this war.



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

B) GNU Free Documentation License (GNU FDL)

Copyright © AÑO TU-NOMBRE.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Family Care: El asistente en el cuidado de tu familia.</i>
Nombre del autor:	<i>Alejandro Cimadevilla Estradera</i>
Nombre del consultor/a:	<i>Roger Montserrat Ribes</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	01/2018
Titulación:	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>IONIC, APP, FamilyCare</i>

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

Family Care surge como respuesta a la necesidad de llevar un control remoto y compartido de la enfermedad de un familiar. El objetivo es crear una herramienta que permita el seguimiento detallado, y bien organizado de cualquier proceso médico por el que pasa una persona, a la vez que permite la compartición entre usuarios de dicho seguimiento.

Family Care se centra en un contexto de familia con niños menores, con adultos dependientes, o incluso enfermos con enfermedades crónicas.

La forma de abordar este proyecto es haciendo uso de métodos de ingeniería de software, siguiendo un método en cascada, según las fases establecidas por el propio TFM (Planificación, Diseño, Implementación, Presentación).

Como resultado, se tiene una APP multiplataforma, con un interfaz intuitivo y atractivo, que será el asistente remoto para el cuidado de cualquier familiar. El producto final será la propia app, su documentación, la apk, así como diferentes manuales de usuario.

Si bien la metodología utilizada no es la recomendada para estos proyectos, las exigencias del TFM conducen a una metodología que ha podido influir en que los objetivos principales hayan sido alcanzados, aunque algunos secundarios queden como futuras ampliaciones de la APP.

Abstract (in English, 250 words or less):

Family Care arises in response to the need to take a remote and shared control of a family member's illness. The objective is to create a tool that allows the detailed and well-organized follow-up of any medical process that a person goes through, while allowing the sharing between users of that monitoring.

Family Care focuses on a family context with younger children, with dependent adults, or even patients with chronic illnesses.

The way to approach this project is by using software engineering methods, following a cascade method, according to the phases established by the TFM itself (Planning, Design, Implementation, Presentation).

As a result, we will have a multiplatform APP, with an intuitive and attractive interface, which will be the remote assistant for the care of any family member. The final product will be the app itself, its documentation, the apk, as well as different user manuals.

Although the methodology used is not the one recommended for these projects, the requirements of the TFM have guided a methodology that has allowed the main objectives to be achieved, although some secondary ones remain as future extensions of the APP.

Índice

1. Introducción	1
1.1. Contexto y justificación del Trabajo.....	1
1.2. Objetivos del Trabajo	3
1.3. Enfoque y método seguido.....	4
1.4. Planificación del Trabajo	5
1.5. Breve sumario de productos obtenidos	8
1.6. Breve descripción de los otros capítulos de la memoria	8
2. Diseño	9
2.1. Diseño Centrado en el Usuario	9
2.1.1. Análisis y diseño conceptual.....	9
2.1.2. Prototipado.....	13
2.1.3. Evaluación del diseño.	19
2.2. Diseño técnico.....	21
2.2.1. Casos de Uso.	21
2.2.2. Diseño de la arquitectura.	27
3. Implementación	33
3.1. Framework de desarrollo: IONIC.....	33
3.1.1. Preparación del entorno de desarrollo.....	33
3.2. Firebase.	35
3.2.1. Base de datos.	35
3.2.1. Diseño físico.	35
3.2.2. Realtime Data Base.	39
3.3. Implementación de Family Care.....	39
3.3.1. Revisión de modelado.	39
3.3.2. Revisión del diseño de Clases.....	39
3.3.3. Revisión del diseño de Vistas.	40
3.4. Pruebas.	42
3.5. Mantenimiento del Servicio Firebase.	45
3.6. Revisión de la planificación	45
4. Conclusiones	47
5. Glosario	48
6. Bibliografía	50
7. ANEXOS.	51
7.1. Anexo I. Manual de instalación.	51
7.2. Anexo II. Manual de usuario.....	55

Lista de figuras

Ilustración 1. Cifras del INE.....	1
Ilustración 2. Herramientas existentes en el mercado.	2
Ilustración 3. Funcionalidades a alto nivel de Family Care.	3
Ilustración 4. Ciclo de vida del Desarrollo Ágil.	4
Ilustración 5. Diagrama de Gantt para PEC2 Diseño.....	5
Ilustración 6. Diagrama de Gantt para PEC3 Implementación.....	6
Ilustración 7. Diagrama de Gantt para Entrega Final.	6
Ilustración 8. Diagrama de Gantt (Planificación del TFM).	7
Ilustración 9. Árbol de Navegación de Family Care	14
Ilustración 10. Prototipo Bienvenida.....	15
Ilustración 11. Prototipo acceso usuario	15
Ilustración 12. Prototipo registro de usuario.....	15
Ilustración 13. Prototipo Lista de pacientes.....	16
Ilustración 14. Prototipo de Área de paciente.	16
Ilustración 15. Prototipo pantalla de mediciones.	17
Ilustración 16. Prototipo del Historial.....	17
Ilustración 17. Prototipo de Sala de visitas.	18
Ilustración 18. Prototipo pantalla Doctor.	18
Ilustración 19. Alternativa a Área de Paciente	20
Ilustración 20. Diagrama de Casos de Uso.....	21
Ilustración 21. Diagrama UML de Base de Datos	30
Ilustración 22. Diagrama UML de Clases.....	31
Ilustración 23. Diagrama MVC.	32
Ilustración 24. Base de datos en Firebase de Family Care.....	36
Ilustración 25. Tabla Usuario en Firebase (Family Care).....	36
Ilustración 26. Tabla Expediente en Firebase (FamilyCare).	37
Ilustración 27. Tabla Rol de Firebase (Family Care).....	37
Ilustración 28. Tabla Paciente de Firebase (Family Care).	38
Ilustración 29. Tabla Medicion en Firebase (Family Care).....	38
Ilustración 30. Pantalla medición de tensión (Family Care).	41
Ilustración 31. Cierre de Expediente.	41

Ilustración 34- Planificación inicial PEC3.	45
Ilustración 33. Dedicación real en la PEC4.	46
Ilustración 34. Opciones de Seguridad, Orígenes desconocidos en Android. ...	51
Ilustración 35. Ejecución de Family Care en navegador modo Lab.	52
Ilustración 36. Emulador iPhone 8 Xcode, lanzando Family Care.	53

1. Introducción

1.1. Contexto y justificación del Trabajo

El que tiene hijos pequeños, sabrá que en los periodos en los que pasan por alguna enfermedad puntual como gripe, faringitis, varicela, virus estomacales, etc., así como cualquier proceso que el niño requiere de una supervisión periódica, el control de la medicación, como conocer la evolución de nuestros hijos supone una preocupación para los padres. Esto se puede extender al cuidado de cualquier persona que de alguna manera es dependiente, y es necesario llevar un control sobre su proceso médico.

Además, como factor influyente, hay que tener en cuenta que, en nuestra sociedad actual, es bastante habitual que no sea siempre la misma persona la que esté al cuidado del paciente, puesto que el entorno laboral tiende a ser cada vez más absorbente, por lo que, habitualmente, el cuidado de nuestro familiar recae en manos de otras personas como pueden ser abuelos, tíos, cuidadores especializados, etc.

Todo esto hace que el seguimiento del periodo de enfermedad del paciente, se pueda volver confuso en ciertas situaciones, y crear una inquietud añadida. Además, está el inconveniente de tener que dedicar tiempo a recabar la información (fiebre, tos, evolución de ronchas, o el indicador que sea) tomada por diferentes personas, haciendo uso de medios como papel, o en el mejor de los casos, mensajería instantánea para llevar la información de la evolución del proceso médico, que hace que esa información se lleve de forma poco organizada. Todo esto puede suponer un tiempo, y una sensación de descontrol, en situaciones donde son varios los responsables de cuidar al paciente a lo largo del día.

Si nos fijamos en algunas cifras del Instituto Nacional de Estadística [1] [2], se revela que hay un público muy significativo que dan valor a estas necesidades, y que convierten estas mismas en oportunidades de negocio.

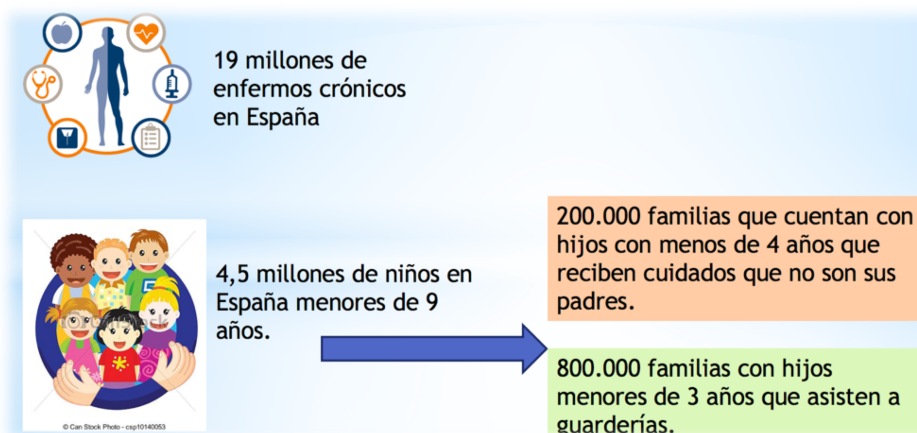


Ilustración 1. Cifras del INE.

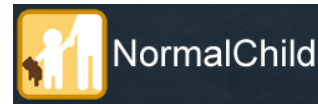
¿Qué medios se utilizan hasta ahora para cubrir estas necesidades? Lo más habitual es seguir haciendo uso de anotar en papel los indicadores médicos, o hacer uso de herramientas de mensajería para compartir con familiares control de evolución de un paciente.

Por otro lado, existen apps en el mercado que se mueven en el mismo terreno, pero sin llegar a dar solución a la necesidad expuesta. Ejemplo de ello es Pain Care o Normal Child.



Pain Care : Se centra en el registro de episodios de dolor. No es exactamente lo que hará nuestra app. Permite indicar, intensidad, tipo duración y localización del dolor. Se centra en un registro minucioso del episodio de dolor. Aparte, también se pueden incluir los medicamentos, si no aparece el tuyo puedes añadirlo a mano, indicar las dosis, y además permite realizar informes que se te envían a tu correo electrónico.

NormalChild Se centra en el seguimiento del crecimiento y desarrollo del bebé, independientemente de que esté pasando o no por un proceso médico. Permite a los padres llevar un registro detallado de la evolución de sus hijos. Desde el peso, la altura, perímetro craneal, fotos, tallas, enfermedades, vacunas, alergias e hitos en el desarrollo.



Otras apps que se mueven en un área similar: MeCuido, Doctor Mole, o MedPageToday.



Ilustración 2. Herramientas existentes en el mercado.

Además, hay que tener en cuenta otras herramientas del mercado, que tienen un uso más genérico, pero también se emplean para dar solución al registro y compartición de información, como son Evernote, Whatsapp y Google Docs.

De la necesidad expuesta anteriormente, y en base a las herramientas existentes en el mercado, nace la idea de Family Care, nuestra APP. Una aplicación que pretende ser una herramienta de seguimiento del proceso médico de un paciente. Pero no se queda ahí, porque, además, pretende ser un punto de encuentro donde las personas que queramos, puedan compartir y/o visualizar información sobre el proceso médico de nuestros hijos, nuestros mayores, o incluso nosotros mismos. Un panel donde podemos visualizar de forma sencilla y ordenada, los indicadores tomados al paciente (desde fiebre, tensión arterial, pulso), hasta fotografías para comprobar la evolución (ronchas en varicela, manchas, lunares, heridas). Todo ello presentado de forma intuitiva en una línea temporal. Incluyendo un sistema de notificaciones, para avisarnos de las actualizaciones de la

información del proceso médico de nuestro hijo. Así los familiares cercanos, podrán estar informados en tiempo real de cómo evoluciona el paciente. Aunque está abierto a todo tipo de público, en un principio está pensado para usuario que deseen llevar el registro y seguimiento médico de niños, personas mayores y enfermos dolencias crónicas [3].

1.2. Objetivos del Trabajo

El objetivo general es crear una herramienta sencilla e intuitiva que permita a familiares hacer el seguimiento médico de un paciente de forma remota y compartida.

El corazón de la aplicación será el Área de Paciente, que es la pantalla de la app que muestra la información principal del estado del paciente, y las últimas mediciones de indicadores médicos, y a la vez se puede acceder a todas las funcionalidades que va a ofrecer Family Care.

La idea es enfocar la experiencia del usuario como un registro de pacientes, de manera que, para cada paciente, se tenga la posibilidad de registrar un proceso médico (gripe, faringitis, hipertensión, diabetes, etc).

Para alcanzar el objetivo general, debemos dotar a nuestra app de una serie de funcionalidades:

- Registro de usuarios, y pacientes.
- Iniciar y finalizar procesos médicos (nueva dolencia o enfermedad).
- Registrar indicadores médicos concretos (temperatura, medicación, fotografías, notas genéricas)
- Compartir expedientes de pacientes, con otros usuarios.
- Informes médicos ordenados en línea temporal.
- El acceso compartido de la información, requiere un almacenamiento en la nube, lo que implica una independencia del terminal físico.

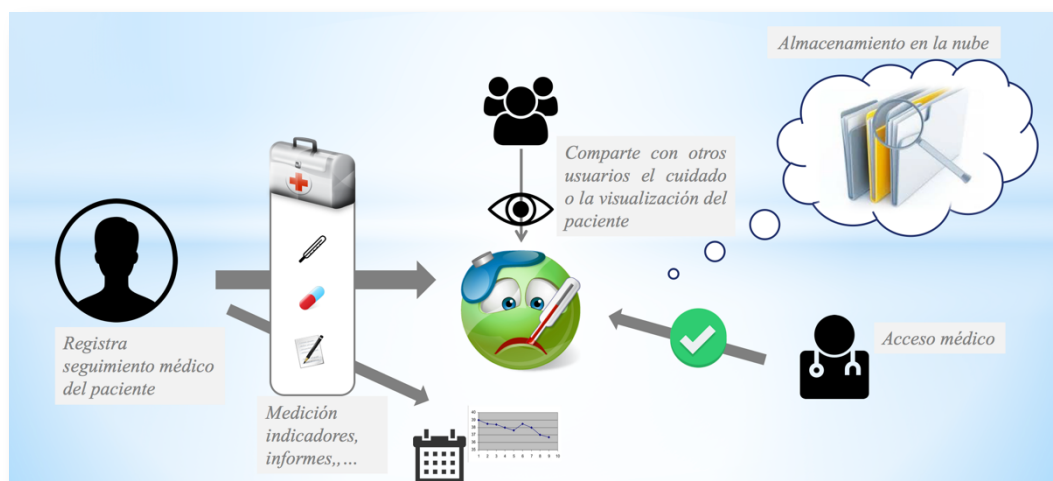


Ilustración 3. Funcionalidades a alto nivel de Family Care.

Una de estas funcionalidades, la compartición de expedientes con otros usuarios, abre la puerta a permitir acceso a un médico, y que la app se convierta en punto de consulta remota.

Otro de los objetivos más sólidos, es que Family Care, se convierta en el historial médico de una familia, de manera que vaya ganando valor a lo largo del tiempo, ya que irá guardando, poco a poco, todo el historial de procesos médicos por los que pasa un individuo.

1.3. Enfoque y método seguido

Como se ha expuesto en el punto 1.1., ya existen medios e incluso herramientas que de forma parcial pueden llegar a parte de los objetivos que planteamos. Es por ello que nuestra estrategia va a ser desarrollar un producto nuevo, inspirado en algunas herramientas que existen en el mercado, pero con otros propósitos.

En un desarrollo con el objetivo de sacar el producto al mercado, sería más idóneo usar el método llamado Desarrollo Ágil de Software [4], basado en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo, según las necesidades del

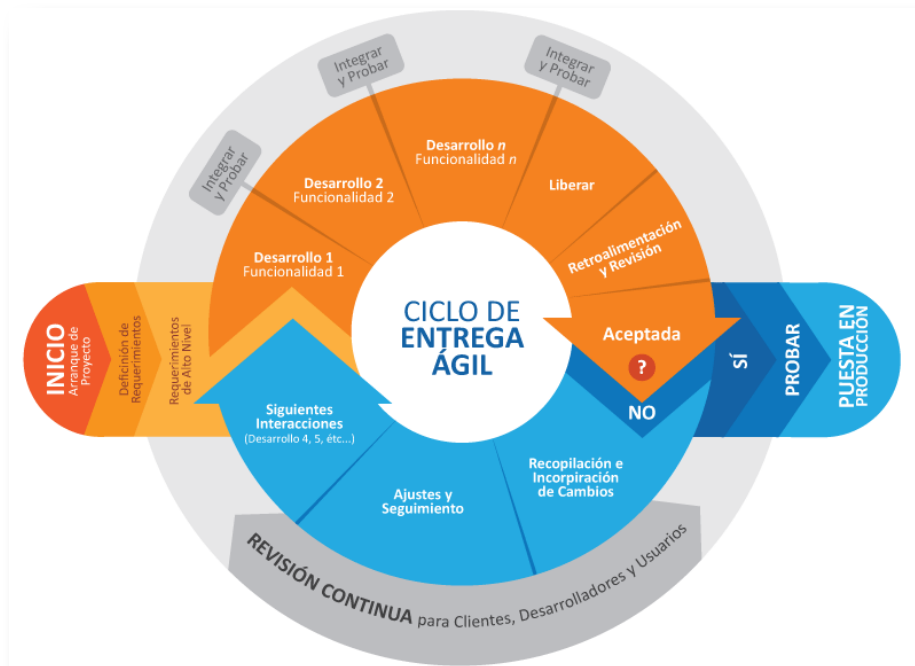


Ilustración 4. Ciclo de vida del Desarrollo Ágil.

proyecto.

No obstante, las propias PECs de este TFM marcan el método a seguir, que encaja con un modelo en Cascada.

1.4. Planificación del Trabajo

Comencemos enumerando los recursos humanos y tecnológicos que vamos a necesitar, y con los que contamos. Por un lado, como alumno, adoptaré el rol de diseñador y desarrollador. Por otro lado, incluso antes de entrar en detalles de diseño técnico, ya prevemos que necesitaremos un servidor, que debido al ámbito educativo en el que se realiza este proyecto, podré utilizar mi propio ordenador para simular las peticiones web services.

La planificación gira en torno a las actividades marcadas por este TFM, divididas en los tres próximos entregables (PECs). Cada una de las cuales contarán con sus propias subtareas.

En la PEC2, existe una tarea de evaluación que hace que el diseño conceptual y el prototipado sean tareas iterativas, estas deben incluir una tarea, cada una, que refleje las revisiones y cambios de ambos. Se estiman un total de 60 horas para esta PEC. El esquema de tareas sería el siguiente:

PEC2 - Diseño

- DCU (Diseño Centrado en el Usuario)
 - Análisis Usuarios y Contexto de usos.
 - Diseño conceptual.
 - Escenarios de uso.
 - Revisión Diseño.
 - Prototipado.
 - Prototipo inicial.
 - Revisión prototipado.
 - Evaluación del Diseño.
- Diseño Técnico.
 - Casos de uso.
 - Diseño arquitectura.

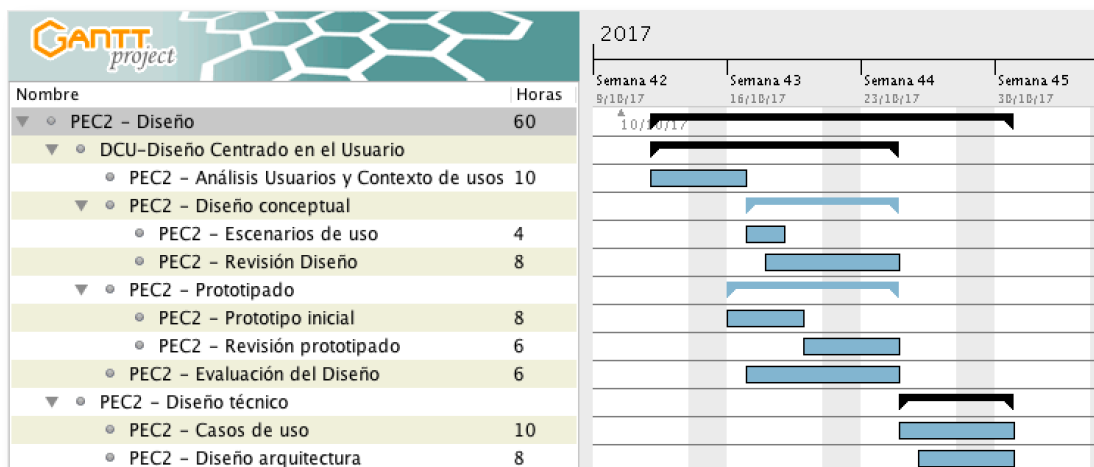


Ilustración 5. Diagrama de Gantt para PEC2 Diseño.

La PEC3, tendrá el mayor número de horas de dedicación con un total de 114 horas. El esquema de tareas sería el siguiente:

PEC3 – Implementación.

- Desarrollo.
 - Entorno y herramientas.
 - Servidor Web y SGBD.
 - Programación APP.
- Pruebas.

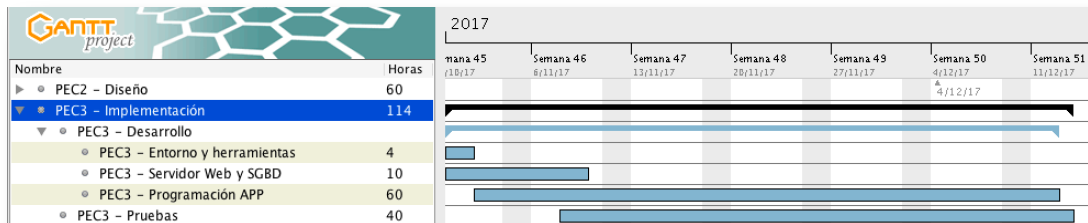


Ilustración 6. Diagrama de Gantt para PEC3 Implementación.

Tres tareas componen la última PEC, para las cuales se estiman un total de 40 horas:

EF - Entrega Final.

- Empaquetar código APP.
- Manual usuario y fin memoria.
- Presentación.

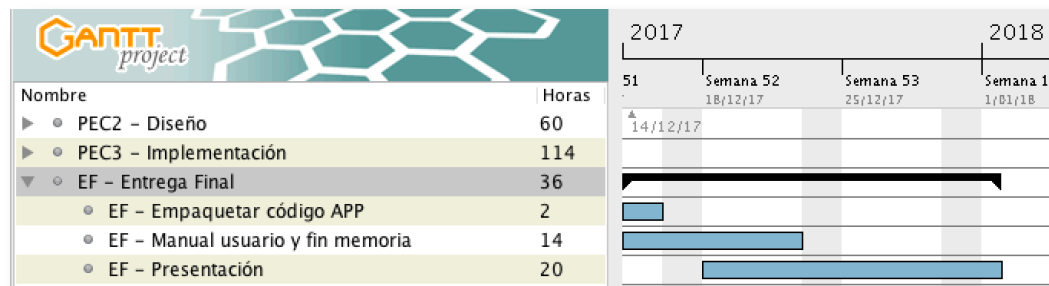


Ilustración 7. Diagrama de Gantt para Entrega Final.

Se ha realizado una estimación total de 200 horas totales para estas actividades, Este reparto se refleja en el diagrama de Gantt:

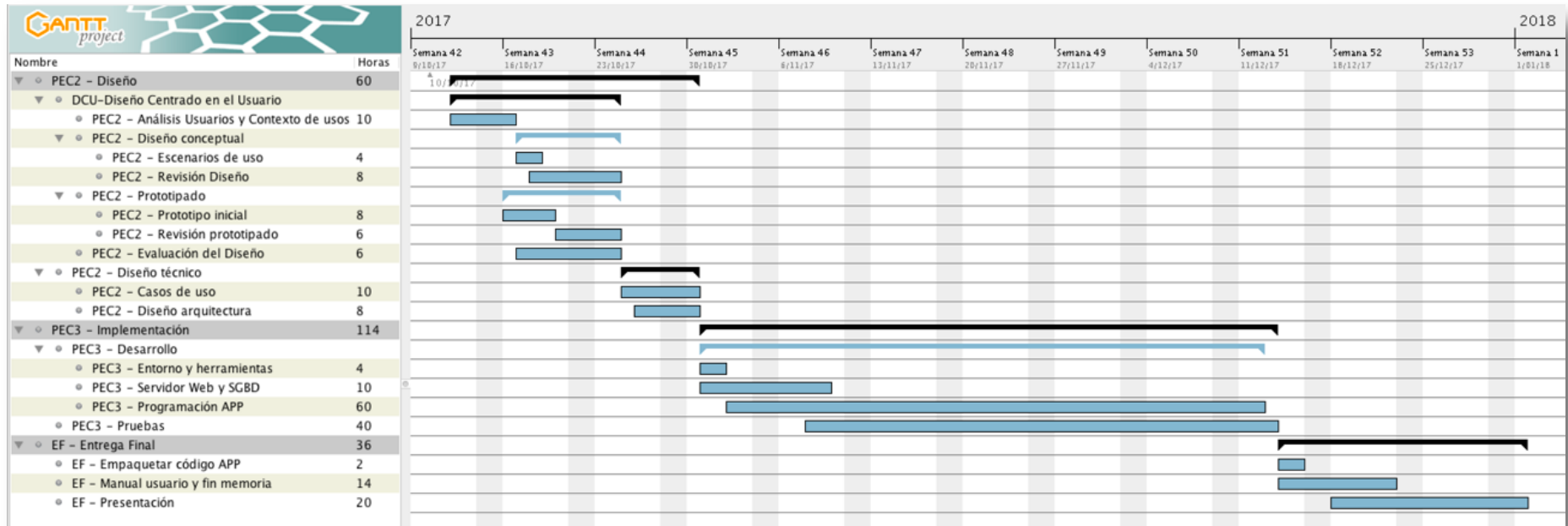


Ilustración 8. Diagrama de Gantt (Planificación del TFM).

En base a la disponibilidad del alumno, se ha calculado la dedicación de horas diaria al TFM:

- La dedicación media en días laborable es de 3 horas al día.
- La dedicación media en días festivos es de 2 horas al día.

1.5. Breve resumen de productos obtenidos

Los productos obtenidos, una vez finalizado este TFM, serán:

- App Family Care para Android: Archivo apk para poder ser instalado en cualquier dispositivo móvil.
- App Family Care para emulador iOS. Archivo para ser instalado en un emulador iOS de XCODE.
- Código fuente de la app: Proyecto IONIC, para compilar y lanzar en emuladores y navegadores. Cabe destacar que el proyecto se encuentra cargado en un repositorio GIT, en modo privado.
- Documentación: Memoria del TFM.
 - Manual Usuario.
 - Manual compilación/despliegue.
- Material multimedia con la exposición y presentación del TFM.

1.6. Breve descripción de los otros capítulos de la memoria

2. Diseño. Este capítulo estará formado por las cuatro partes que siguen las fases del Diseño Centrado en el Usuario (DCU):
 - Usuarios y contexto de uso [Análisis]
 - Diseño conceptual [Diseño]
 - Prototipado [Diseño]
 - Evaluación [Evaluación].
3. Implementación. En este capítulo se muestra el proceso de adquisición de las herramientas de desarrollo, su configuración, así como los pasos para la codificación de la solución de Family Care. También incluye el plan de pruebas, y una reflexión y análisis sobre la revisión del diseño.
4. Conclusiones. Se incluye una reflexión final sobre la elaboración del TFM, sobre los objetivos alcanzados y los no alcanzados. Y las conclusiones sacadas tras la elaboración del TFM.
5. Glosario. Definición de los términos y acrónimos más relevantes utilizados dentro de la Memoria.
6. Bibliografía. Listado de los recursos bibliográficos y web utilizados durante la elaboración del TFM.

2. Diseño

2.1. Diseño Centrado en el Usuario

El diseño de nuestra app, tanto funcional como gráfico, debe responder a las características definidas en el proceso de análisis. El primer paso para la elaboración del diseño, es elaborar el modelado del usuario. Proseguirá con el diseño visual, elaborando un prototipo del interfaz de la app. Y finalizará con una etapa de evaluación, donde se revisen los diseños de forma iterativa. En esta memoria dedicaremos ese apartado para indicar las modificaciones realizadas, antes las diferentes alternativas de diseño. Comenzamos, pues, con la primera etapa de modelado del usuario [5].

2.1.1. Análisis y diseño conceptual

En la primera fase de análisis, buscamos dar respuesta a cuestiones como:

- ¿A qué tipo de usuario va dirigida? ¿Cuáles son sus necesidades?
- ¿Cuál es el contenido de la aplicación?
- ¿Cuáles son los recursos humanos disponibles?
- ¿Cuál es el tiempo de vida del producto?

En el Contexto y Justificación del trabajo (apartado 1.1. de esta memoria), ya se explica esta idea previa, y se da respuesta a estas preguntas. El segmento de clientes en el que nos vamos a centrar es, padres con hijos pequeños, adultos con personas dependientes bajo su cuidado, y personas con enfermedades crónicas.

Los problemas o necesidades que se pretenden atender son los siguientes:

- Inquietud y dificultad en el seguimiento remoto de su hijo enfermo.
- Organización tediosa de los datos médicos (mediciones, tomas, ...)
- Dificultad para compartir información con otros familiares.
- Pérdida de tiempo, ordenando datos para enseñárselo al médico.

Ahora vamos a profundizar algo más en las personas (público principal) y las necesidades que nuestra app puede cubrir.

Para el modelado del usuario se reúne la información sobre los usuarios potenciales y se definen los perfiles de usuario teniendo en cuenta atributos comunes como las necesidades de información, la experiencia y conocimientos o las condiciones de acceso a Family Care.

Vamos a usar la técnica de las fichas de persona. Para ello vamos a elaborar las fichas de persona, de los arquetipos de público objetivo. Asociado a cada ficha de persona, elaboraremos un escenario de uso de la aplicación. Los escenarios de uso nos servirán para poder contextualizar el proceso de interacción entre la persona y nuestra aplicación, Family Care. Éstos describen casos concretos de utilización, teniendo en cuenta las tareas que el sistema debe llevar a cabo y el contexto en el que la persona va a utilizar Family Care [6].

A continuación, se muestran diferentes fichas de personas y sus escenarios de uso asociados.

Ficha:

Nombre: Iñaki

Edad: 38 años

Profesión: Consultor de Sistemas de Información.



Descripción de la persona:

Iñaki está casado y tiene dos hijos, de 2 y 5 años, una va a la guardería y el mayor al colegio. Vive en un pueblo periférico cerca de Madrid. De lunes a viernes, se desplaza al centro de Madrid para trabajar una jornada de nueve horas. Su trabajo le deja poco tiempo libre. Tiene poca flexibilidad para poder faltar al trabajo para cuidar a sus hijos en caso de enfermedad. Su mujer, también trabaja en condiciones similares a las suyas.

Cuando alguno de sus hijos se ha puesto malo, necesita tirar de una cuidadora, y sólo en caso de que no pueda la cuidadora, echan mano de su suegra. Iñaki es una persona acostumbrado al uso de las tecnologías, internet, Smartphone, Tablet incluso Smartwatch.

Escenario:

Es el segundo día que la hija pequeña de Iñaki, Julia tiene que quedarse en casa por un proceso gripal (fiebre, tos, etc.) A mitad de mañana, Iñaki accede a Family Care, y selecciona a Julia en la lista de pacientes. A Iñaki le aparecen las últimas mediciones introducidas por la cuidadora de Julia, y descubre que a Julia le ha bajado la fiebre y lleva toda la mañana sin toser.

Iñaki, marca los indicadores, como que los ha revisado.

Ficha:

Nombre: Alejandro

Edad: 35 años

Profesión: Profesor de Formación Profesional.



Descripción de la persona:

Alejandro, vive en Córdoba y tiene un hijo de algo menos de 3 años, el cual va a la guardería hasta las tres de la tarde. La pareja de Alejandro, y madre del pequeño, también tiene turno de mañana, lo que impide que en casos de enfermedad, pueda quedarse con el hijo, excepto que sea el mismo día que asiste al pediatra.

Cuando el pequeño se ha puesto malo, piden ayuda a la abuela materna, que se desplaza desde Cádiz para echar una mano. Cuando no puede, Alma y Alejandro, tienen a Mayca, una cuidadora a la que pagan por horas. En casos normales de enfermedades pasajeras, los padres son informados al llegar a casa. En momentos puntuales, sí se comunican por teléfono. Todos cuentan con smartphones.

Escenario:

Hoy es el tercer día en que el pequeño Pablo está malo por un virus infantil llamado comúnmente “virus boca-mano-pie” que le obliga a quedarse en casa, mientras Alma y Alejandro, se ven obligados a pedir ayuda a Mayca, la cuidadora, ya que a los abuelos no les da tiempo a llegar a Córdoba.

Alma y Alejandro, están informados de cómo ha ido el día (indicadores médicos, toma de medicamentos, etc) gracias a Family Care.

Alejandro quiere informar a su madre, y su hermana que es enfermera, para informarles cómo evoluciona el pequeño. Mientras Alma informa a sus padres por su lado.

Tanto Alejandro como Alma, no necesitan trasladar la información de evolución de su hijo a su familia directamente, simplemente Alejandro entra en el área de visitas, donde añade a su hermana y a su madre, y a sus suegros como visitantes del proceso médico de su hijo, por lo que ahora todos ellos, pueden ver la información sobre la evolución de Pablo.

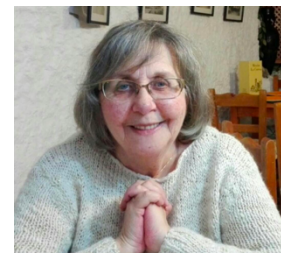
Ficha:

Nombre: Cristina

Edad: 69 años

Profesión: Maestra jubilada.

Descripción de la persona:



Cristina tiene cuatro hijos y vive sola en un pueblo de Jaén. Sus cuatro hijos viven en diferentes provincias andaluzas. Cristina, cada mes se trae a su madre, Pepa de 91 años de edad a su casa, la cual se turna con la hermana de Cristina, para cuidar de ella. La hija de Cristina, Carolina, es enfermera, y siempre es un referente ante consultas sobre el estado de Pepa.

Escenario:

Hoy ha sido un día algo más complicado para Pepa, la madre de Cristina. Y como cada día, le miden las tensiones cada 2 horas, para llevarle un control muy periódico, que permita adelantarse a una bajada excesiva de tensión, que le lleve a un desvanecimiento o incluso una parada del corazón, o a una subida excesiva que pueda provocar pequeños ictus, los

cuales ya le dieron en el pasado. Cristina cada vez que toma la tensión, abre Family Care, y accede al expediente de Pepa. Toca para introducir una nueva medición de tensión, que queda registrada por defecto a fecha y hora actual.

A los pocos minutos, el Smartphone de Cristina suena con una notificación de Family Care, indicando que hay una nueva nota de Carolina, que le indica a Cristina que la tensión está algo baja, que le dé frutos secos a la abuela.

Ficha:

Nombre: Ana

Edad: 41 años

Profesión: Médico.



Descripción de la persona:

Elena, vive en Málaga y tiene tres hijas de 2, 4 y 5 años. Su marido Greg, vive en otra ciudad fuera de su provincia. En caso de que las niñas se pongan malas, Elena deja a las niñas con los abuelos maternos, y en los casos en los que no pueden, en los paternos.

Los abuelos informan a Ana siempre directamente de la evolución de las niñas. Greg por su parte, para saber cómo evolucionan las niñas, suele llamar a Elena, o a veces a los abuelos.

Los fines de semana, cuando Ana y Greg están juntos, si la situación lo requiere se turnan por las noches para hacer ronda nocturna y ver a la que esté mala, o darle la medicación si es necesario.

Escenario:

A la pequeña hoy le han salido unas ronchas un poco extrañas en los brazos, acompañada de fiebre. Tras ir al médico, todo apunta a que puede ser varicela.

En el segundo día, siguen saliendo más ronchas, y Ana desde su trabajo, se pregunta cómo irá evolucionando la pequeña.

Accede a Family Care, y comprueba en un par de foros, tomadas entre las últimas mediciones por la abuela de la pequeña, que la cantidad de ronchas es similar a la que tenía esta mañana. Y algunas ya están secándose.

Según otro indicador, tomado hace una hora, la fiebre no baja, y también comprueba que la abuela le ha dado supositorio 150 mg, febrectal. Esta información será útil para calcular el tiempo para poder darle otra dosis.

Ficha:

Nombre: Violeta

Edad: 35 años

Profesión: Ingeniera de Montes.

Descripción de la persona:



Violeta, vive en Córdoba y tiene un hijo de algo menos de 2 años. Violeta hace uso habitual de las tecnologías, y es una persona con grandes inquietudes tecnológicas, y le gusta consultar todo en internet sobre todo en lo que se embarca, tomando siempre referencia de científicos o expertos, no de cualquier página web.

Violeta es una persona muy organizada, tanto en casa como en el trabajo, tiene todas sus cosas catalogadas y bien organizadas. En la medida de lo posible, tiende a usar la informática para llevar buen orden de sus cosas, y organizado el acceso a su información personal y profesional.

Escenario:

Desde hace un tiempo, Violeta se lleva observando unas decoloraciones en los nudillos de las manos, como unas manchas. Desde que se lo ha detectado, Violeta, se creó un expediente médico, y se hace periódicamente fotos en las manos, mediante FamilyCare, quedando registradas en su expediente. Llegado el momento de la cita con el dermatólogo, hace uso de Family Care, para acceder al listado de fotos por fecha, de su expediente, y mostrarle la evolución.

Empieza un tratamiento, que según le informan va a tener una muy larga duración, y ella como persona organizada y metódica que es, le resulta útil llevar un registro de toda la evolución para poder ir comparando, pasado un periodo de tiempo considerable.

El doctor, que tiene cuenta de Family Care, le pide que le agregue como Doctor, para poder seguir su evolución a través de la aplicación.

2.1.2. Prototipado

Para la elaboración de un prototipo de alto nivel, debemos partir por establecer el flujo de navegación de Family Care, para lo cual haremos uso de un árbol de navegación. En él, cada pantalla estará representada por un rectángulo, y las flechas son las acciones de interacción con la app.

En la siguiente página, se muestra la figura con el árbol de navegación al completo. En ella se muestra como en el centro de la aplicación, se sitúa el corazón de la app, que es el Área de Paciente, el punto desde el cual se puede acceder al resto de funcionalidades y pantallas.

Árbol de navegación:

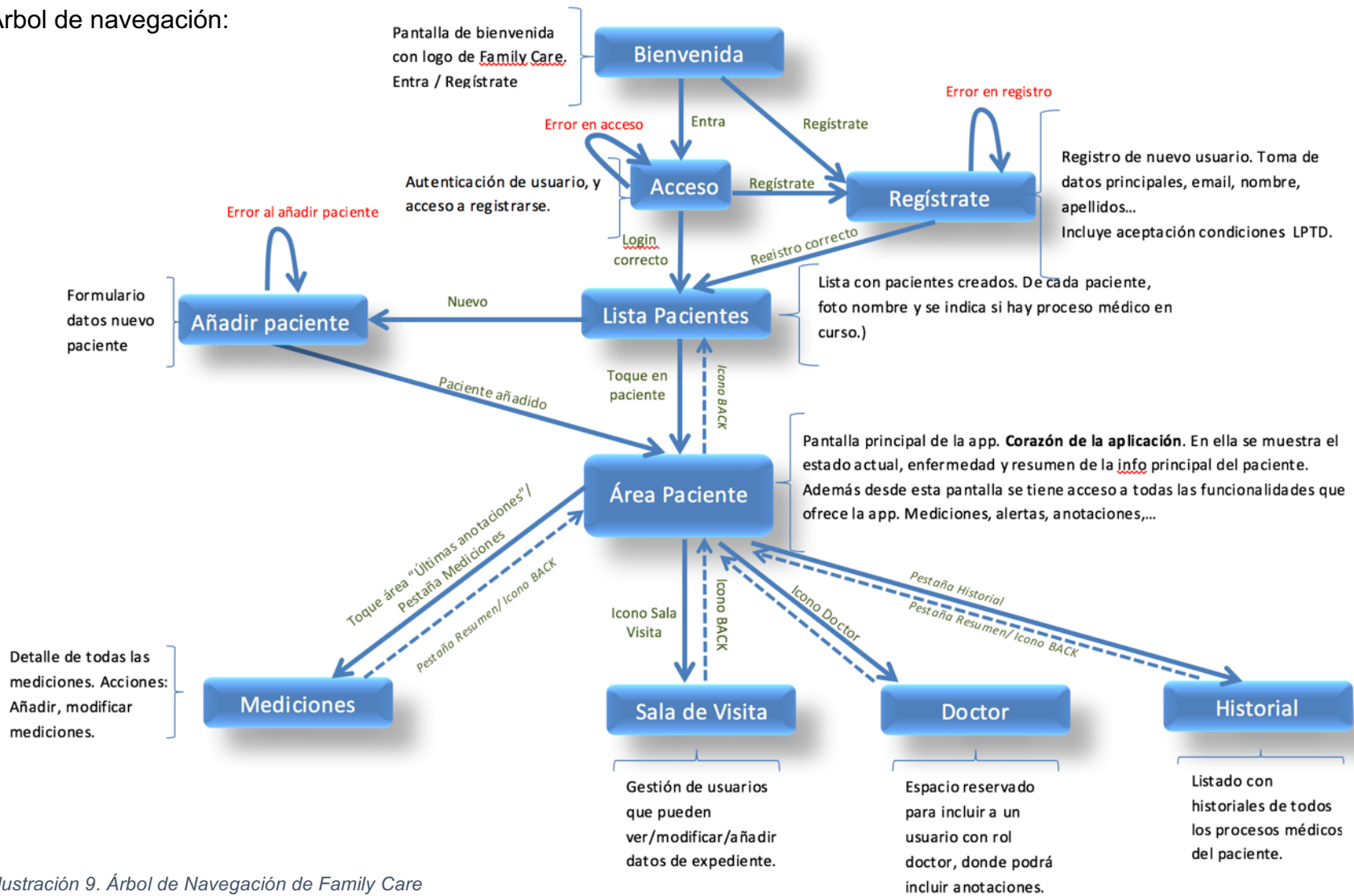


Ilustración 9. Árbol de Navegación de Family Care

Una vez tenemos una primera aproximación del árbol de navegación, comenzamos a trabajar en cada una de las pantallas que van a formar el prototipo de Family Care.

Haremos uso de la herramienta llamada Balsamiq [], para dibujar nuestro prototipo, en base al árbol de navegación construido.

Las primeras pantallas que se muestran, son las habituales en toda app. Pantalla de bienvenida (Ilustración 10), la de acceso/identificación del usuario (Ilustración 11), y la de registro de un nuevo usuario (Ilustración 12).

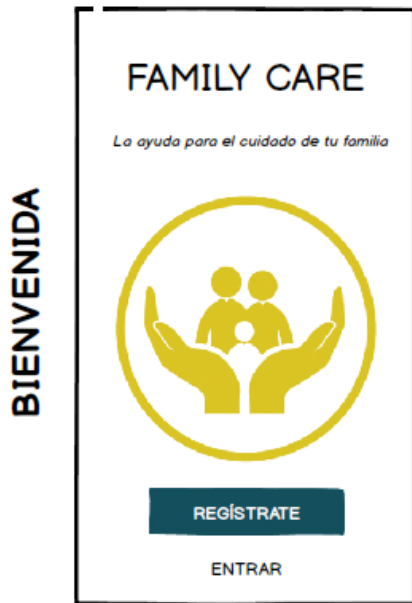


Ilustración 10. Prototipo Bienvenida



Ilustración 11. Prototipo acceso usuario



Ilustración 12. Prototipo registro de usuario

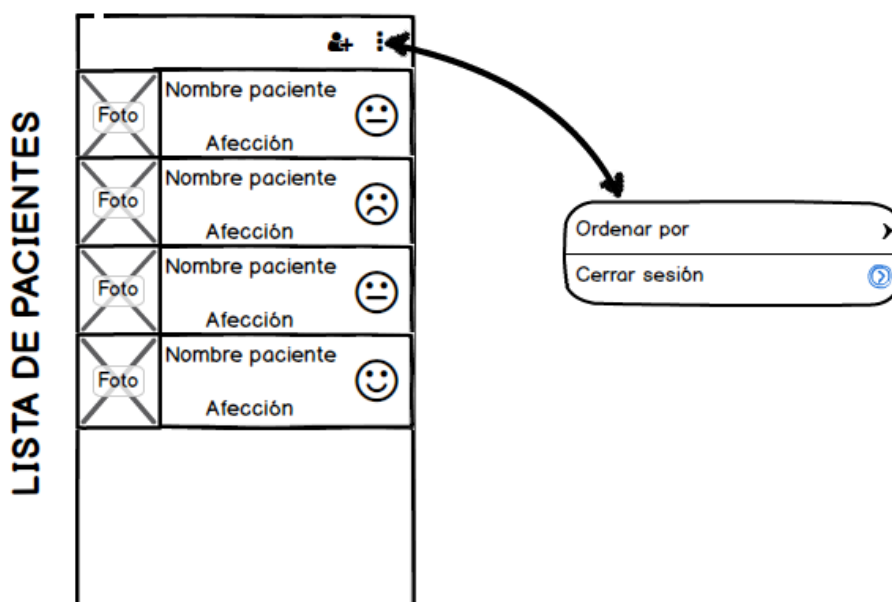


Ilustración 13. Prototipo Lista de pacientes.

Listado de los pacientes (Ilustración 13), donde cada entrada, muestre foto del paciente, nombre, la dolencia en curso, y un icono indicativo si el proceso está en curso o cerrado.

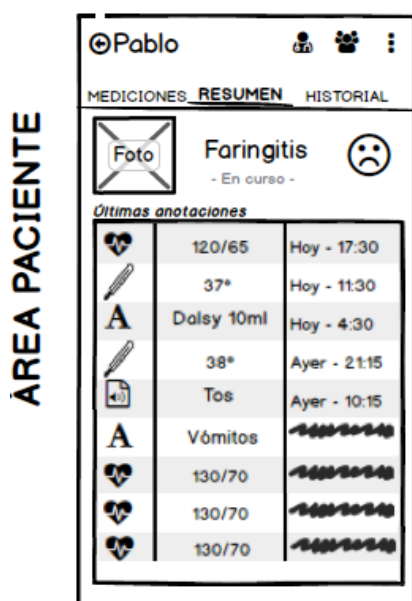


Ilustración 14. Prototipo de Área de paciente.

El área de paciente es el corazón de Family Care (Ilustración 14). Desde esta pantalla, se accede a toda la funcionalidad de la app. Además, es la pantalla que pretende mostrar la información clave de cada paciente: identifica al paciente, foto, afección, y últimas anotaciones.

Además, tiene acceso directo mediante pestañas, a la pantalla mediciones, y al historial de un paciente.

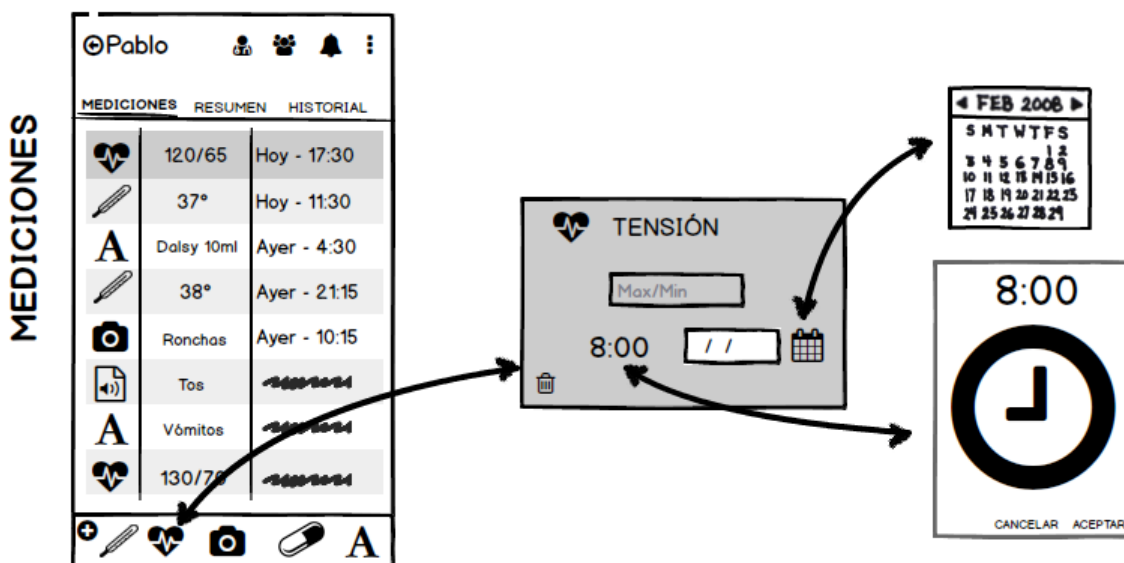


Ilustración 15. Prototipo pantalla de mediciones.

Esta pantalla es donde se insertan todas las mediciones de nuestro paciente (Ilustración 15). En la zona inferior, habrá unos iconos, que al tocar en ellos nos habilitarán una pop-up del indicador seleccionado, para introducir el dato de medición.



Ilustración 16. Prototipo del Historial.

En esta sección (Ilustración 16), se listarán las enfermedades/expedientes por los que ha pasado o está pasando un paciente.

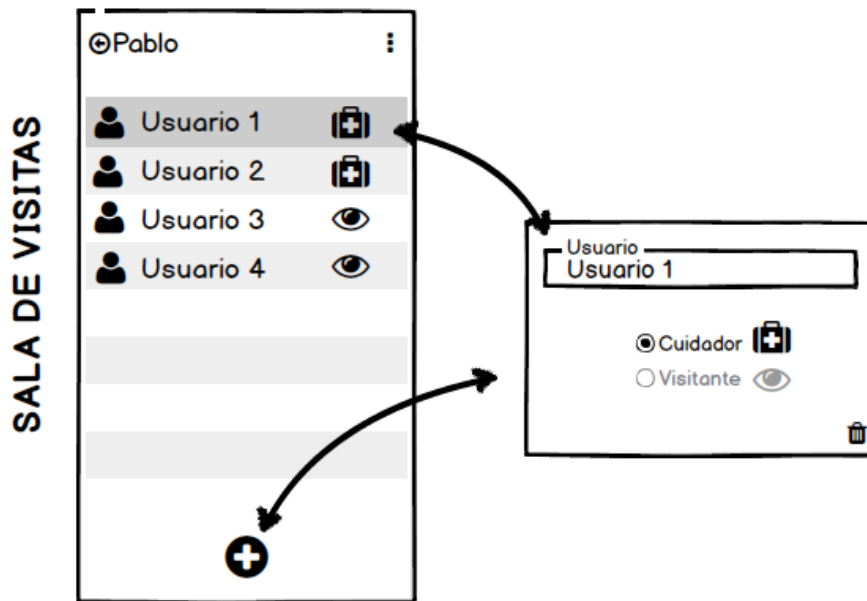


Ilustración 17. Prototipo de Sala de visitas.

La sala de visitas (Ilustración 17) es la pantalla desde la cual, invitamos a otros usuarios a ser cuidadores, que pueden hacer las mismas acciones que el cuidador principal, o visitantes, que sólo pueden consultar la información introducida por los cuidadores.

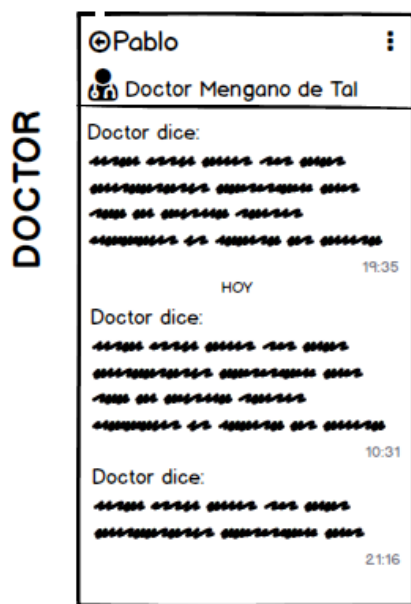


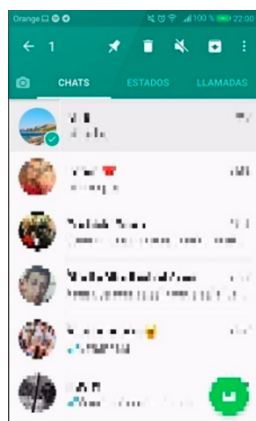
Ilustración 18. Prototipo pantalla Doctor.

Esta pantalla permitirá a un usuario que esté como doctor, incluir notas informativas (Ilustración 18).

2.1.3. Evaluación del diseño.

La evaluación del diseño realizado, se basa en métodos heurísticos. Algunos de los factores que se analizan son:

1. Aspectos globales del proyecto:
 - Declaración de los objetivos del proyecto y su relación con el diseño y los contenidos ofrecidos en Family Care.
 - Coherencia de la estructura, elementos de navegación y diseño visual a través de la app. Estableciendo el corazón de la app, como zona principal(rea de paciente), donde estén los accesos a todas la funcionalidades de la app.
 - Existencia de diferentes vías de contacto y su visibilidad dentro de la app.
2. Estructura y elementos de navegación:
 - Existencia de elementos que le indiquen al usuario en qué parte del sitio se encuentra. Evitamos, accesos anidados que lleven al usuario a un laberinto de pantallas. Por eso Family Care, cuenta con una pantalla central, desde la cual se accede al resto de pantallas.
 - Uso de iconos representativos de la funcionalidad relacionada. Es importante, para que la app se convierta en una herramienta fluida, que la interacción vaya unida a iconos para facilitar al usuario su acceso y asociación a funcionalidades concretas.
 - Esquema de navegación similar a otras apps populares del mercado. Usamos el esquema de organización de whatsapp, para la pantalla principal:



- i. Listado de anotaciones en la parte central.
- ii. Tres pestañas, para acceder a las funcionalidades más significativas: Mediciones, Área de Paciente, e Historial.
- iii. Tres iconos en la parte superior, para funcionalidades más esporádicas.

3. Diseño visual de la interfaz:

- Capacidad para demostrar una jerarquía clara.
- Uso correcto de colores y tipografías. Escogiendo un estilo asociado a Family Care. Usaremos la misma gama de colores para todas las pantallas.
- Claridad y limpieza del diseño. Evitando sobrecargar de información y simbología nuestras pantallas.

Entre las revisiones realizadas, se han planteado diversas alternativas en el proceso de diseño.

A continuación se listan los elementos alternativos de cada fase, y el porqué de su descarte.

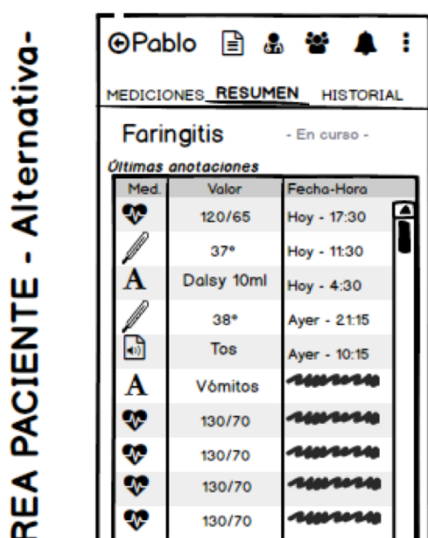


Ilustración 19. Alternativa a Área de Paciente

Para la pantalla principal, Área de Paciente, se plantearon dos diseños. El diseño descartado (Ilustración 19), suponía sacrificar el espacio dedicado para identificar al paciente, con su foto y nombre, para dar más espacio a las últimas anotaciones.

Tras evaluar los diseños, con diferentes usuarios potenciales, se llega a la conclusión de que favorece el entendimiento, tener identificado al usuario mediante una foto, y además liberar un poco de carga de información a esta pantalla, mostrando menos mediciones, entre las últimas. De hecho, siempre se podrá acceder mediante un clic, al listado de todas las mediciones.

También se han descartado algunas funcionalidades extra, como sería un área de alertas, que podría plantearse como una ampliación actualización futura.

2.2. Diseño técnico.

2.2.1. Casos de Uso.

Para la elaboración del diagrama UML de casos de uso (Ilustración 20), primeros debemos identificar los actores, y acciones que compondrán nuestra aplicación.

- Actores:
 - Cuidador. Usuario que puede realizar cualquier acción sobre el expediente.
 - Visitante: Sólo puede consultar la información existente.
 - Doctor. Puede consultar información, y añadir notas médicas.
- Acciones:
 - Registro.
 - Acceso(Login).
 - Añadir/Modificar Paciente.
 - Añadir/modificar proceso médico.
 - Añadir modificar medición.
 - Añadir modificar usuarios en Sala Visita.

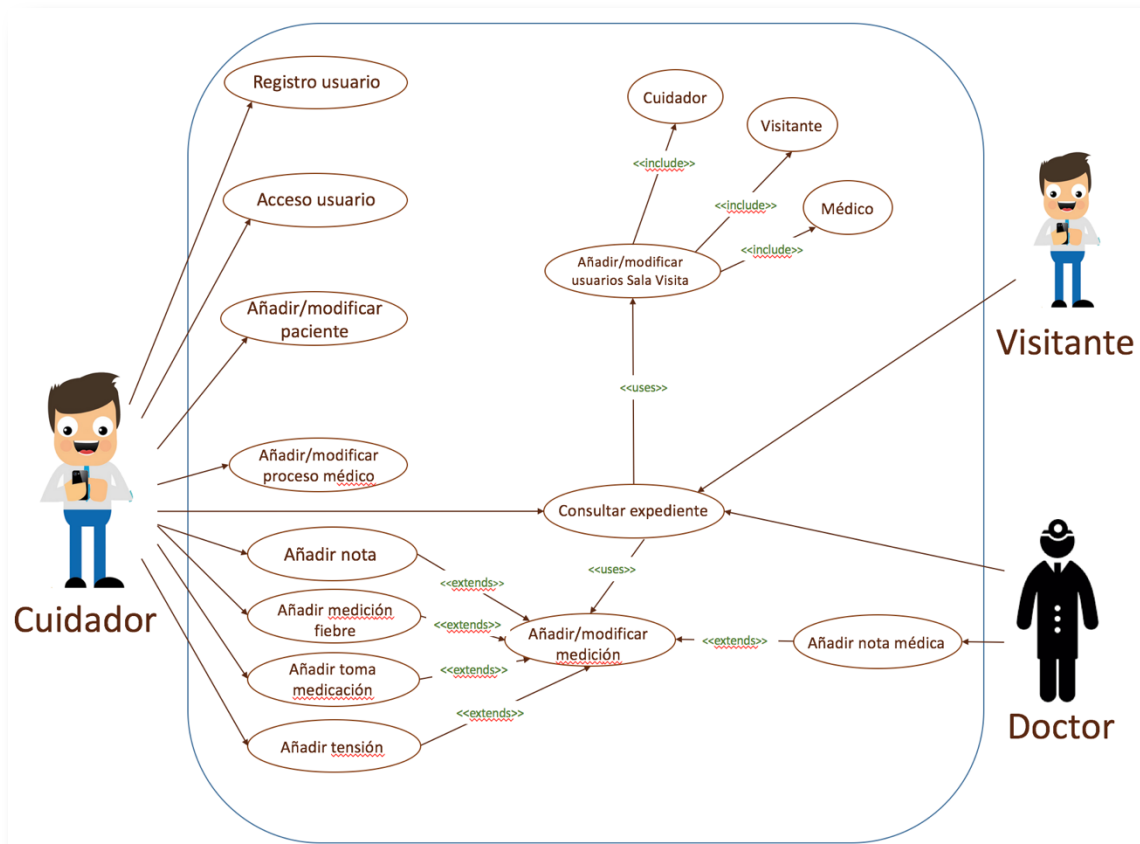


Ilustración 20. Diagrama de Casos de Uso

A continuación, se expone el listado de los diferentes casos de uso generados.

Nombre	C.U.1: Registrarse en Family Care
Actores	Cuidador, Visitante o Doctor (Usuario)
Objetivo	Crear una cuenta de usuario en Family Care.
Precondiciones	No existe el usuario en el sistema.
Postcondiciones	Usuario nuevo registrado.
Flujo de acciones	<ol style="list-style-type: none"> 1. Usuario accede a Registro de Family Care. 2. Usuario introduce datos para completar formulario de registro. 3. El sistema solicita confirmación de los datos introducidos. 4. El usuario confirma los datos.
Flujo Alternativo	<ol style="list-style-type: none"> 1. Usuario accede a Registro de Family Care. 2. Usuario introduce datos para completar formulario de registro. 3. El sistema solicita confirmación de los datos introducidos. 4. El usuario cancela registro.

Nombre	C.U.2: Acceso a Family Care (Login)
Actores	Cuidador, Visitante o Doctor (Usuario)
Objetivo	Identificarse como usuario, y acceder a las funcionalidades de la app.
Precondiciones	Tener una cuenta creada en Family Care. No estar logado con la misma u otra cuenta de usuario.
Postcondiciones	<ul style="list-style-type: none"> • Redirección a pantalla listado de pacientes, si los datos de acceso son válidos. • Mostrará mensaje de error en caso contrario.
Flujo de acciones	<ol style="list-style-type: none"> 1. Usuario abre Family Care. 2. Se le solicita usuario y contraseña. 3. El usuario introduce usuario y contraseña.

	<ol style="list-style-type: none"> 4. El sistema valida usuario y contraseña. 5. Family Care se sitúa en la pantalla Listado de Pacientes
Flujo alterantivo	<ol style="list-style-type: none"> 1. Usuario abre Family Care. 2. Se le solicita usuario y contraseña. 3. El usuario introduce usuario y contraseña. 4. El sistema comprueba que el usuario y/o contraseña, no son válidos. 5. Family Care muestra un mensaje indicando que los datos de acceso no son correctos.

Nombre	C.U.3. Añadir paciente.
Actores	Cuidador.
Objetivo	Crear una nueva entrada en la lista de pacientes.
Precondiciones	Estar logado con un usuario válido, y situado en la pantalla de listado de pacientes.
Postcondiciones	<p>Un nuevo paciente es creado, con un proceso médico abierto.</p> <p>El usuario queda asociado como cuidador principal.</p>
Flujo de acciones	<ol style="list-style-type: none"> 1. Toque/clic en icono añadir paciente. 2. Se abre una nueva pantalla con formulario de nuevo paciente. 3. El usuario completa los datos Nombre, fecha de nacimiento, y proceso médico, y se añade foto. 4. El usuario da toque en botón Nuevo paciente. 5. El paciente queda registrado, y la app regresa a la lista de pacientes, donde se muestra el recién registrado paciente.

Nombre	C.U.4: Modificar paciente.
Actores	Cuidador
Objetivo	Modificar algún dato general sobre el paciente.
Precondiciones	El paciente debe existir y aparece en la lista de pacientes.

Postcondiciones	Los datos del paciente pueden haber sido modificados.
Flujo de acciones	<ol style="list-style-type: none"> 1. El usuario hace clic/toque mantenido en la fila del paciente. 2. La app muestra pop-up pidiendo confirmación para modificar el paciente. 3. Se muestran los campos editables del paciente: nombre, fecha de nacimiento, y proceso médico. 4. El usuario modifica los datos concretos. 5. El usuario hace clic en el botón modificar paciente. 6. La app regresa a la pantalla lista de pacientes.

Nombre	C.U.5: Añadir medición
Actores	Cuidador
Objetivo	Incluir una medición sobre un indicador concreto (fiebre, tensión, toma de medicamento, foto, nota)
Precondiciones	El paciente debe existir y tener un proceso médico abierto.
Postcondiciones	Una nueva medición queda registrada y mostrada en primer lugar en la lista que se muestra en el área de paciente.
Flujo de acciones	<ol style="list-style-type: none"> 1. El usuario accede al paciente desde la lista de pacientes. 2. El usuario accede a la pestaña mediciones. 3. El usuario hace clic en el icono del indicador concreto a introducir (fiebre, tensión, medicamento, foto o nota). 4. La app muestra una pop-up, donde introducir la medición concreta. Aparece un campo fecha y hora, editable, con valor por defecto la fecha y hora actual. 5. El usuario hace clic en botón OK, para insertar la medición. 6. La app muestra la medición en el primer lugar de la lista de mediciones.

Nombre	Consultar expediente
--------	----------------------

Actores	Cuidador, Visitante o Doctor (Usuario)
Objetivo	Consultar expediente de un paciente.
Precondiciones	El paciente debe existir (proceso médico abierto o cerrado)
Postcondiciones	Sin modificaciones de estado.
Flujo de acciones	<ol style="list-style-type: none"> 1. Usuario accede desde lista de pacientes, o desde historial, haciendo clic en el paciente concreto.. 2. La app abre la pantalla área de paciente, donde se muestran las últimas mediciones.

Nombre	C.U.6: Añadir proceso médico
Actores	Cuidador
Objetivo	Añadir un proceso médico a un paciente ya existente.
Precondiciones	El paciente ya debe existir.
Postcondiciones	Se genera una nueva entrada en el listado de pacientes en tratamiento.
Flujo de acciones	<ol style="list-style-type: none"> 1. El usuario accede desde listado de paciente (clic mantenido sobre el paciente). 2. Se muestra una pop-up para insertar el nombre del proceso médico. 3. El usuario, introduce el nombre del nuevo proceso médico. 4. El usuario hace clic en ok. 5. Se muestra el nuevo proceso médico en el listado.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario accede desde el historial (clic sobre botón +). 2. Se muestra una pop-up para insertar el nombre del proceso médico. 3. El usuario, introduce el nombre del nuevo proceso médico. 4. El usuario hace clic en ok. 5. Se muestra el nuevo proceso médico en el listado.

Nombre	C.U.7Añadir usuario a Sala de Visita
Actores	Cuidador, Visitante o Doctor (Usuario)
Objetivo	Compartir expediente médico con otro usuario.
Precondiciones	Debe existir el proceso médico. La persona con la que queremos compartir tiene que tener una cuenta de usuario de Family Care.
Postcondiciones	El usuario añadido a la sala de visita, quedará con el rol asignado: <ul style="list-style-type: none"> • Cuidador: modificar y consultar. • Visitante: sólo consultar la información del expediente. • Doctor: consultar, y añadir notas especiales.
Flujo de acciones	<ol style="list-style-type: none"> 1. Usuario cuidador accede a Sala de visita con icono superior de la app (grupo usuarios). 2. La app muestra una pantalla con el actual listado de usuarios que hay en la sala de visita, y el rol de cada uno. 3. Usuario cuidador hace clic en botón +. 4. App muestra pop-up, con un campo usuario, y un botón de selección, cuidador, visitante, médico. 5. Usuario introduce un usuario y selecciona su rol, y hace clic en OK. 6. App pide confirmación. 7. Usuario confirma. 8. Nuevo usuario aparece en la Sala de visita con su rol correspondiente.

Nombre	C.U.7: Añadir nota médica
Actores	Doctor (Usuario)
Objetivo	Añadir información médica a un expediente concreto.
Precondiciones	El expediente médico debe estar abierto. Usuario Doctor exista en la sala de visita.
Postcondiciones	Lanzará una notificación a los usuarios de la sala de visita.

Flujo de acciones	<ol style="list-style-type: none"> 1. Usuario Doctor accede al expediente correspondiente. 2. Hace clic en el botón de doctor, en la barra superior. 3. La app muestra una pop-up con un campo editable para insertar un texto variable. 4. El doctor escribe su nota, y hace clic en OK. 5. Se envía una notificación push a los usuarios de la Sala de Visita. 6. La nueva nota es mostrada en la pantalla de notas del doctor.
-------------------	---

Nombre	C.U.8: Consultar nota médica
Actores	Cuidador, Visitante, Doctor (Usuario)
Objetivo	Acceder a las notas añadidas por el doctor.
Precondiciones	Debe existir un doctor en la sala de visitas.
Postcondiciones	Sin modificaciones de estado.
Flujo de acciones	<ol style="list-style-type: none"> 1. El usuario (cualquiera) accede al área de paciente, y hace clic en el icono de doctor. 2. La app muestra la pantalla con las últimas notas introducidas por el doctor.

2.2.2. Diseño de la arquitectura.

Para definir la arquitectura de Family Care, haremos uso de diferentes técnicas, que muestren detalles de la construcción conceptual del Modelo, de la Vista y del Controlador, para ello usaremos:

- Diagrama UML de Base de Datos (Modelo).
- Diagrama UML de Clases (parte del controlador).
- Esquema MVC, que muestre la relación de todo lo anterior con la Vista (pantallas prototipo).

Diagrama UML del diseño de Base de Datos

La persistencia de la información es clave en Family Care, por lo que será necesario tener la información estructurada y organizada de forma óptima para un acceso eficiente y seguro.

Empecemos por estudiar los requisitos de nuestro modelo conceptual de Base de Datos.

Usuarios
Queremos llevar un registro de usuarios, caracterizados por: <ul style="list-style-type: none">• Email, como identificador único.• Clave de acceso.• Nombre.• Primer apellido• Segundo apellido.

Los usuarios, podrán añadir pacientes nuevos, creando para estos un nuevo expediente médico. Como una de las funcionalidades clave, es la de compartir expedientes, un usuario podrá crear diferentes expedientes, y su vez, un expediente, puede pertenecer a varios usuarios (cardinalidad mucho a muchos). De aquí surgirán dos entidades, Paciente y Expediente Médico.

Paciente
Del paciente necesitamos tener: <ul style="list-style-type: none">• Identificador Paciente• Nombre.• Primer apellido.• Segundo apellido.• URI foto, enlace a la foto.

Expediente médico
Del expediente necesitamos tener: <ul style="list-style-type: none">• Identificador del expediente.• Nombre de la enfermedad.• Fecha Inicio de la enfermedad.• Fecha fin de la enfermedad.

Cada usuario que comparte un expediente, tiene un rol asociado al expediente. Por lo que cada expediente podrá pertenecer a, al menos, un

usuario, o varios usuarios. A su vez, un usuario podrá tener varios expedientes para consultar.

Con esto, tendremos una relación que se convertirá en tabla, Acceso, que tenga el rol asociado al usuario con cada expediente.

Acceso médico
Esta relación permitirá guardar el rol del usuario en un expediente concreto.

Por último, queremos llevar diferentes mediciones, las cuales sólo existirán para un mismo expediente médico (relación uno a muchos, entre expediente y medición). Consideramos pues, otra entidad, llamada Medición.

Medición
De las mediciones queremos guardar: <ul style="list-style-type: none">• Identificador de la medición.• Tipo (Fiebre, Tensión, Medicamento, Foto, Nota).• Fecha de medición.• Hora de medición.• Detalle de la medición.• iD del usuario que la ha tomado.

El diagrama resultante, que muestra la relación entre entidades se muestra en la Ilustración 20.

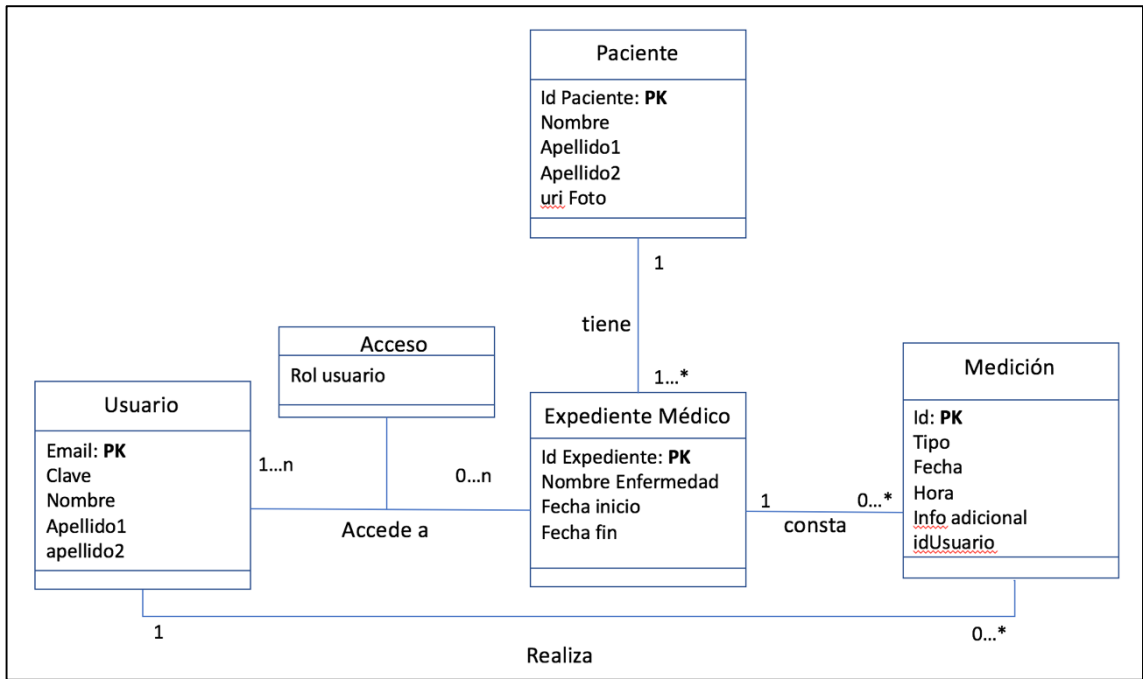


Ilustración 21. Diagrama UML de Base de Datos

Diagrama UML de Clases

El diagrama de clases, nos va a permitir mostrar la relación entre las entidades de nuestra app, y que conforman parte del controlador de nuestra arquitectura. Sabiendo la base de datos que necesitamos, ahora construiremos las clases desde la perspectiva del controlador, para dar respuesta, a través de las vistas, a los requerimientos extraídos en el diseño conceptual.

Clases:

- Usuario. Representa al usuario y sus posibles acciones.
- Paciente. Atributos y acciones sobre el paciente.
- Expediente Médico. Es un proceso médico concreto.
- Medición. Cada una de las mediciones que compone el expediente médico.
- Cuidador. Que representa los datos del usuario, y de la relación que hay entre el usuario y el expediente médico concreto.

El diagrama de clases se muestra en la Ilustración 21.

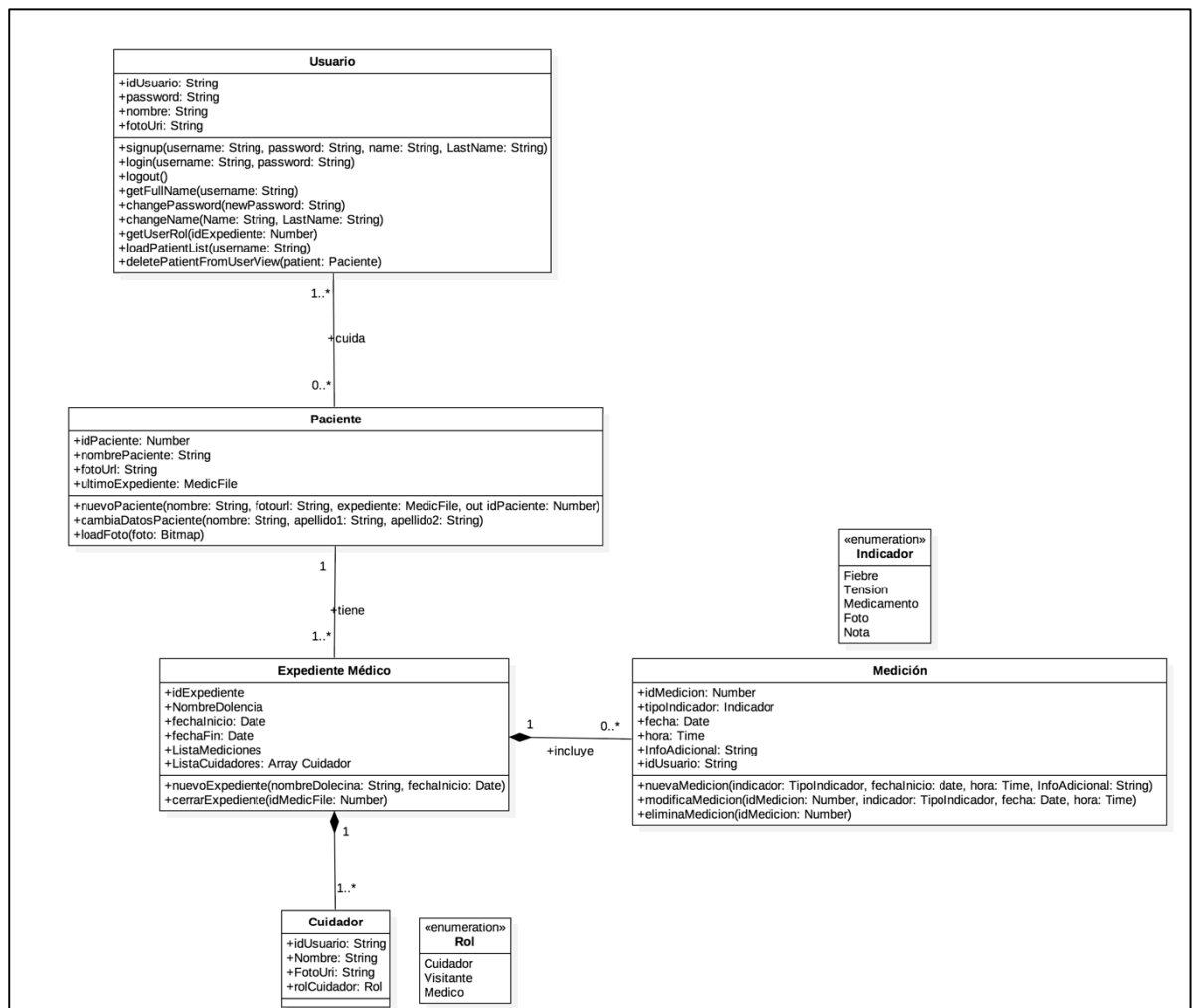


Ilustración 22. Diagrama UML de Clases.

Arquitectura del sistema. Diagrama MVC.

Esta app pretende funcionar para el mayor número de dispositivos posibles. Es por ello que se ha elegido una estrategia de aplicación cross-platform, también conocidas como aplicaciones híbridas. Se usará un framework llamado IONIC, basado en el paradigma MVC. En la ilustración 23, se muestra un diagrama explicativo, formado por:

- El **modelo**, que es la representación de la información, estará en un servidor externo, de Google (Firebase). Firebase Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado.

Firestore nos proporcionará las siguientes funcionalidades:

- Base de datos no relacional en tiempo real.
- Autenticación de usuarios. Incluyendo validación de correo electrónico, y servicio de recuperación de contraseña.
- Firebase storage: almacenamiento en la nube para las fotos que generen los usuarios.
- La **vista**, que estaría compuesto por las pantallas, de las cuales ya tenemos su prototipo.
- El **controlador**, que serán las instancias de las clases, para que a través de su métodos, y lógica de control añadida, respondiendo a los eventos o acciones del usuarios, e invocando peticiones al modelo.

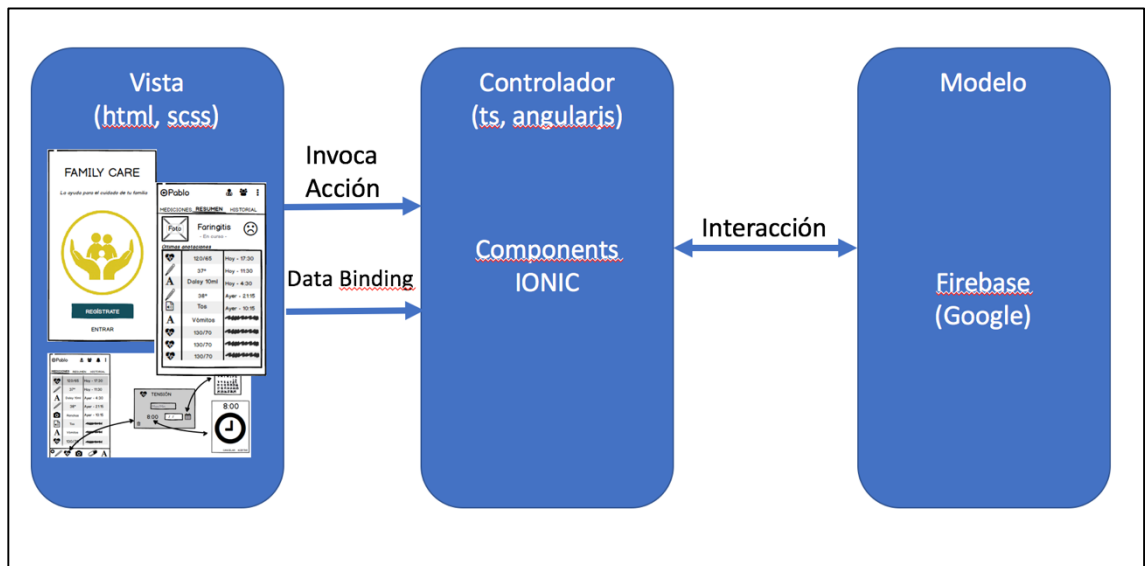


Ilustración 23. Diagrama MVC.

3. Implementación

3.1. Framework de desarrollo: IONIC.

IONIC es un framework open source, que permite construir, desplegar, probar y monitorizar apps híbridas.

La decisión de utilizar este framework se sustenta en varios motivos:

- a. Posibilidad de mostrar su uso en las dos principales plataformas para dispositivos móviles, iOS y Android [7].
- b. Aplicar los conocimientos adquiridos en dos de las asignaturas cursadas, Desarrollo web, y Desarrollo web avanzado para dispositivos móviles.

3.1.1. Preparación del entorno de desarrollo.

A continuación vamos a hacer un análisis de las herramientas que necesitamos para la implementación de la app.

Webstorm. Editor de código.

- Problema: Cuando se edita, código, y sobretodo al empezar, se cometen errores, hace falta escribirlo, etc., además de que en proyectos modernos es imprescindible usar muchos archivos, muchos lenguajes de programación y frameworks, hace falta compilar código, etc. y esto supone un problema en el rendimiento si no se dispone de algún tipo de ayuda
- Solución: Webstorm es un IDE optimizado para el uso de JS y que tiene herramientas de ayuda de todo tipo.
- Instalación desde su web: <https://www.jetbrains.com/webstorm/>

GIT. Control de Versiones (Gitlab)

- Problema: en el proceso de desarrollo trabajamos con código alojado en varias partes. Tener que copiar y pegar archivos supone una pérdida de productividad y puede hacer que se pierda código. Incluso puede ocurrir que necesitemos volver a una versión previa del código porque nos hayamos equivocado.
- Solución: herramientas de control de versiones. Consisten en sistemas que guardan versiones del estado del proyecto en un momento determinado. Esas versiones se pueden luego recuperar e instalar automáticamente en los sitios donde queramos. Con ello se hacen copias sin errores, solo escribiendo un simple comando. También permiten volver a versiones anteriores. Para ello nosotros usaremos Git, que es el control de versiones más usado.
- Instalación desde su web.

SourceTree

- Problema: el uso de git a través de línea de comandos puede ser poco intuitivo
- Solución: utilizar una herramienta visual de control de versiones. Estas herramientas lo que hacen es ofrecer una interfaz visual que permite hacer mediante botones lo que de otro modo se haría con la línea de comandos. Además, permite comparar cambios entre versiones, lo que facilita descubrir problemas en caso de conflictos. La herramienta que hemos decidido usar es SourceTree.

NodeJS

- Problema: muchas tareas son difíciles de hacer, y algunas incluso imposibles, cuando trabajamos con ciertas tecnologías. Además, en muchos proyectos utilizaremos librerías y frameworks que dependen de otros, y cada uno de ellos es compatible con solo ciertas versiones de los demás, así que tener actualizados todas las librerías para que funcionen sin dar problemas entre sí, puede ser casi imposible.
También puede ser muy compleja la instalación de algunas herramientas, frameworks y librerías
- Solución: NodeJs, que funciona como un ejecutor de tareas de todo tipo, como descargar e instalar paquetes, comprobar dependencias, actualizar respetando estas dependencias, etc.

IONIC Framework

- Problema: Elaborar una app que sea independiente de la plataforma donde se lanza. Además, el problema de tener herramientas que permitan probar, emular, lanzar y empaquetar la app.
- Solución: Ionic facilita una serie de funciones que permiten, generar providers(servicios-controlador), pages(páginas-Vista), así como acciones para emular, testear, y lanzar la app.
- Su instalación es sencilla, aunque primero es necesario contar con el nodejs:

```
npm install -g cordova ionic
```

- Los plugins de IONIC. Para llevar a cabo diversas funcionalidades en nuestra app, IONIC nos presenta multitud de plugins. Un ejemplo de ello es *cordova-plugin-camera*, este plugin incluye unas librerías para controlar la cámara del dispositivo, y capturar fotos.

3.2. Firebase.

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles.

Como se ha comentado anteriormente vamos a hacer uso de sus servicios para:

- La autenticación de usuarios, incluyendo recuperación de contraseñas, y validación de estas.
- Base de datos en tiempo real.
- Storage on cloud.

Este servicio es gratuito con ciertas limitaciones de uso. Basta con registrarse con una cuenta de GMAIL.

Una vez registrado, es necesario recopilar los datos que emplearemos en nuestro código, en concreto en el archivo app.module.ts:

```
apiKey: "AIzaSyA-qlCa6yyS7-mHraVTY6Lxg_ilv8kyqfM",
authDomain: "familycare-2440a.firebaseio.com",
databaseURL: "https://familycare-2440a.firebaseio.com",
projectId: "familycare-2440a",
storageBucket: "familycare-2440a.appspot.com",
messagingSenderId: "775404482998"
```

Estos datos lo utiliza las librerías de firebase para conectar con el servicio de Firebase.

3.2.1. Base de datos.

3.2.1. Diseño físico.

Las bases de datos de Firebase son no-relacionales, por lo que debemos cambiar el enfoque de trabajo acostumbrado en un esquema relacional basado en diagramas E-R.

No obstante, nuestra forma de trabajar está intrínsecamente basada en relacionar cada una de las tablas no relacionales. Esto quiere decir, que aunque la base de datos cree ramas (tablas) sin relaciones entre ellas, nosotros seguiremos relacionando unas ramas con otra según el modelo de base de datos diseñado.

En la siguiente figura (Ilustración 24), se muestra el esquema de la base de datos de Family Care en Firebase. Al igual que en nuestro modelo, consta de cuatro tablas (Usuarios, Rol del usuario, Pacientes, Expedientes médicos y Mediciones).

familycare-2440a



Ilustración 24. Base de datos en Firebase de Family Care.

La información se almacena en formato JSON, que utiliza un esquema jerárquico. En la raíz está la base de datos de FamilyCare, y los hijos directos son las tablas. Cada registro se guarda en cada tabla, mediante una clave primaria que identifica de forma única al registro

Esa clave, será la que aparezca en otras tablas, como un atributo más, para así relacionar unas tablas con otras.

Si nos basamos en el modelo de Base de Datos de la fase de diseño, tenemos las siguientes tablas:

Tabla de Usuario. Usuarios registrados (Ilustración 25):

- Id: identificador único (no visible para el usuario).
- Email: id de login de usuario.
- Nombre: Nombre de pila.
- Apellidos: Apellidos del usuario.
- UriPhoto: URL de la foto de perfil, almacenada en el storage de Firebase.



Ilustración 25. Tabla Usuario en Firebase (Family Care).

Tabla de Expediente. Expedientes médicos (Ilustración 26):

- Id: identificador único (no visible para el usuario).
- enfermedad: Nombre de la afección en el proceso médico.

- fechaini: Fecha de inicio de la enfermedad.
- Fechafin: Fecha finalización de la enfermedad.

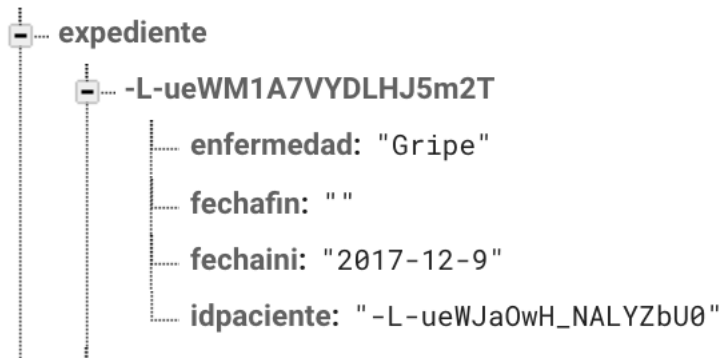


Ilustración 26. Tabla Expediente en Firebase (FamilyCare).

Tabla de Rol. Rol del usuario en un expediente (Ilustración 27). Puede ser Creador, Cuidador o Visitante. Así si varios usuarios comparten expediente, habrá un registro por cada usuario, con el mismo ID expediente:

- Id: identificador único (no visible para el usuario).IDexpediente+IDUsuario.
- IDUsuario.
- IDExpediente: Nombre de pila.
- Sysdate: fecha del sistema, para saber cuándo se le concedió el rol.

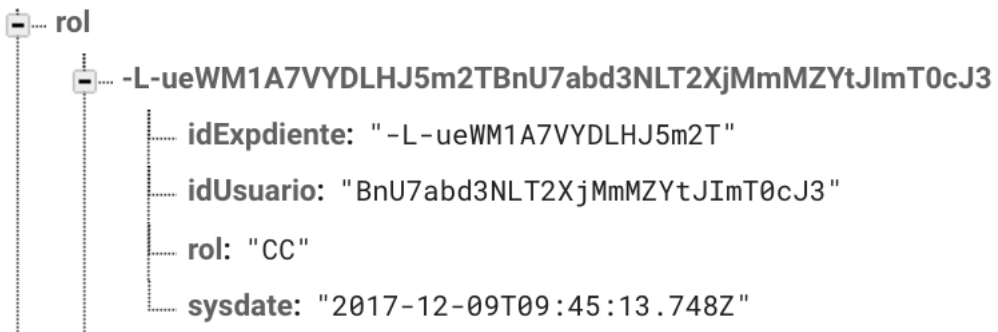


Ilustración 27. Tabla Rol de Firebase (Family Care).

Tabla de Paciente. (Ilustración 27).

- Id: identificador único (no visible para el usuario).

- Nombre: Nombre completo del paciente.
- Fechanacimiento: Fecha de nacimiento, para futuras funcionalidades operando con edades.
- UrlPhoto: URL de la foto de perfil, almacenada en el storage de Firebase
- Enfermedad: última enfermedad registrada.

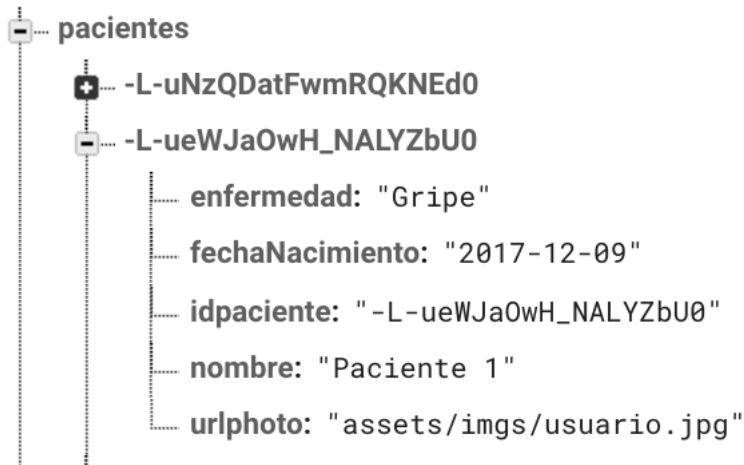


Ilustración 28. Tabla Paciente de Firebase (Family Care).

Tabla de Mediciones:

- ID: identificador único (no visible para el usuario).
- IDUsuario.
- IDEXpediente: Nombre de pila.
- Fecha: Fecha y hora.
- tipo: F (Fiebre), T (Tensión), P(Medicación), C (Foto), N(Nota)
- datos: Datos medidos.
- Datoextra: Almacena información extra para algún tipo, como para la foto, urlphoto almacenada en storage.



Ilustración 29. Tabla Medicion en Firebase (Family Care).

3.2.2. Realtime Data Base.

Firestore ofrece una base de datos en tiempo real, dando la opción a que cualquier cambio sea notificado a los controladores en tiempo real y mostrado en las vistas de forma inmediata. Esta funcionalidad es clave para nuestra app, en la cual hay diferentes usuarios compartiendo datos de un mismo paciente.

En concreto nos beneficiamos de esta funcionalidad expresamente, en el listado de pacientes, listado de mediciones, y listado de usuarios en la Sala de Visitas.

3.3. Implementación de Family Care.

3.3.1. Revisión de modelado.

Cambios en el diseño de base de datos, y su justificación.

Tabla de Usuario:

- Quitamos el campo **clave**: Firestore lleva la autenticación en un Servicio a parte, por lo que no necesitamos almacenar la contraseña en nuestro modelo de datos.
- Agrupamos apellidos en un campo.
- Añadimos campo urlphoto: Almacenamos la url del storage de firestore donde se guarda la foto de perfil del usuario.

Tabla de Mediciones:

- Unimos en un campo fecha y hora: No es necesario dedicar un campo a cada, de esta manera simplificamos.

Salvo estos cambios, se mantiene el modelo.

3.3.2. Revisión del diseño de Clases.

Durante el proceso de implementación, surgen algunas variaciones con respecto al modelo UML de clases. Surgen necesidades relacionadas con la visualización en las vistas que hace que tengamos que incluir algunos atributos adicionales a las clases ideadas en el diseño.

Clases:

- ElementoListaPaciente. Datos que guardamos en cada item de la lista de pacientes para poder acceder a otras pantallas.
 - Idpaciente.
 - Idexpediente.
 - Nombre.
 - fechaNacimiento.
 - fechaFinEnfermedad.
 - Estado. En curso o finalizada con fecha.
 - Enfermedad.

- urlphoto.
- rol. Icono y rol CC, C, o V.
- ElementoListaPaciente. Datos que guardamos en cada item de la lista de pacientes para poder acceder a otras pantallas.
 - idmedicion
 - idexpediente
 - idpaciente
 - idusuario:
 - tipo.
 - icono. (icono de medición).
 - fechaFormateada: Fecha legible por el usuario.
 - datos
 - datosExtra.

3.3.3. Revisión del diseño de Vistas.

A la hora de implementar Family Care, partimos del diseño de prototipos realizados. Durante el proceso de desarrollo de vistas han surgidos diferentes variaciones que se van a listar y a justificar a continuación. Empecemos, primero por listar las diez vistas y controladores de las que consta Family Care.

- Welcome: Vista de bienvenida, al iniciar la app, con logo de Family Care, y accesos a Login y Registro.
- Registro. Registro nuevo usuario.
- Login. Control de autenticación
- RememberPsw: Recuperación de contraseña.
- Nuevopaciente: Formulario para nuevo paciente.
- Editapaciente: vista con formulario para modificar datos de paciente.
- Listapacientes: Listado de pacientes, aunque en realidad se muestran los expedientes abiertos, con la posibilidad de filtrar por expedientes abiertos y cerrados.
- Areapaciente: Dividida en dos pestañas, por un lado muestra y gestiona listado de mediciones sobre el expediente médico seleccionado, mientras que en la otra pestaña muestra el historial del paciente.
- Mediciones-modal: Vista modal, donde se introducen/muestran las mediciones tomadas por los usuarios. Es la misma vista para los 5 tipos de indicadores. Se mostrarán diferentes controles para cada indicador, cuya lógica de control cambia en base al tipo de indicador. Por ejemplo, para la toma de tensiones se muestra un barra deslizable (Range), como se muestra en la Ilustración 30:



Ilustración 30. Pantalla medición de tensión (Family Care).

- Sala de visitas. Gestiona la inclusión, salida y asignación de roles a los usuarios en la Sala de Visitas.
- Para dar más visibilidad al cierre de expediente, se decide añadir un icono junto a sala de visitas.

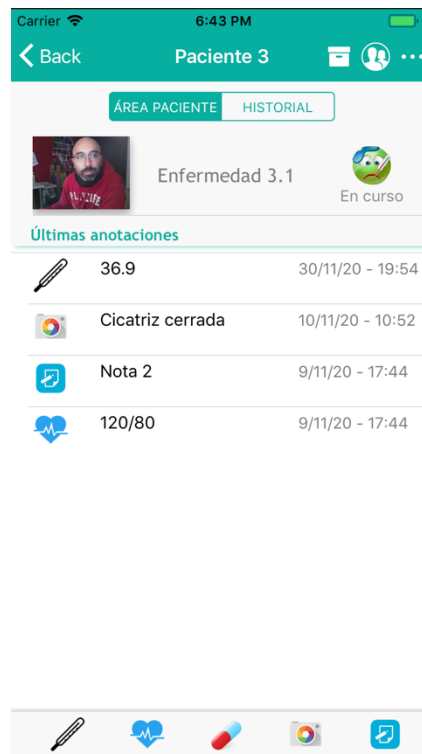


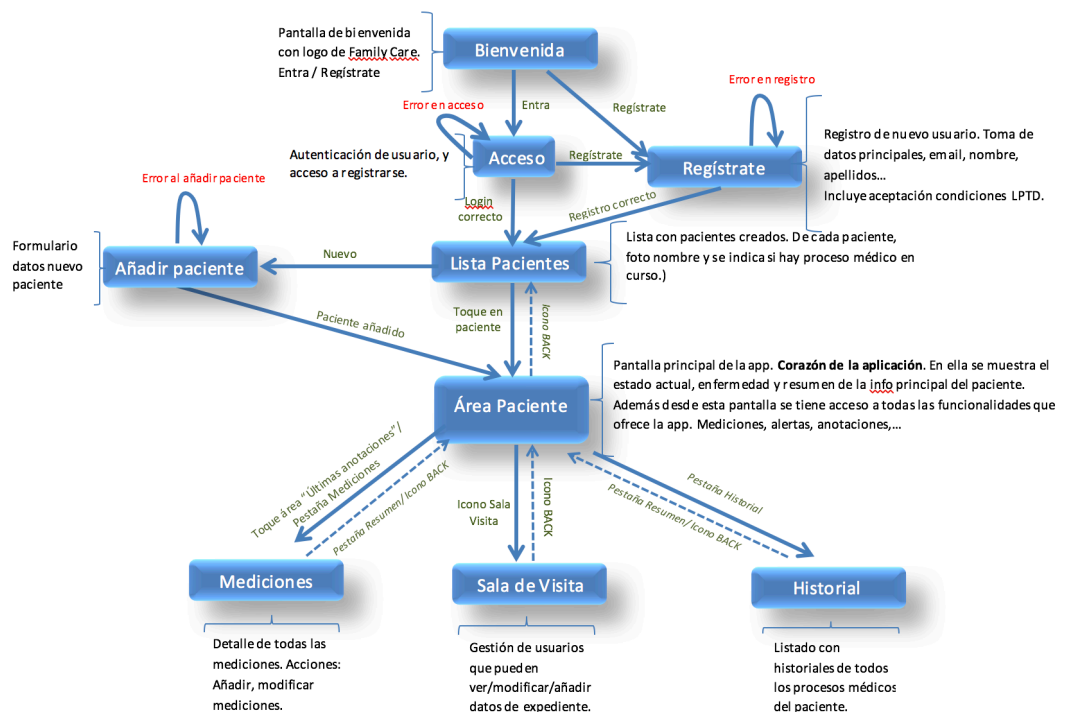
Ilustración 31. Cierre de Expediente.

Cambios en el diseño inicial

Con respecto a lo diseñado inicialmente, han surgido los siguientes cambios.

- La pantalla de Área de Paciente, en un inicio se ideó para que tuviera tres pestañas. No obstante, durante la implementación se ha considerado, que la pestaña de la izquierda, “Mediciones”, no termina de adquirir sentido por sí misma, puesto que pretendía mostrar todas las mediciones, mientras que, en la pestaña central, “Área de Paciente”, sólo mostraría las últimas 5. Pero finalmente, para evitar un acceso adicional al usuario, se ha fusionado las dos pestañas en una, de manera que, en Área de Paciente, aparecen todas las mediciones, y se puede deslizar el dedo para ver mediciones antiguas.
- Funcionalidad excluida, comunicación con el Doctor. Hay dos motivos principales, el primero porque me ha faltado tiempo para poder embarcarme en la funcionalidad de las notificaciones PUSH de firebase, y que habría permitido mostrar notificaciones cada vez que el doctor indicase algo. Partiendo de este punto, hay otro motivo y es que sin notificaciones PUSH, se puede ofrecer el mismo servicio al usuario, incluyendo al doctor como usuario Cuidador, y este incluyendo mediciones de tipo Nota.

Debido a los cambios comentados, el árbol de navegación final, quedaría de la siguiente manera:



3.4. Pruebas.

A continuación, se explica, la estrategia adoptada para el plan de pruebas. Este plan se clasifica en diferentes categorías de pruebas.

Partimos de la idea, que estamos en un modelo de desarrollo que permite la visualización en emuladores desde el comienzo. IONIC permite lanzar la app en modo depuración, mostrando la app en el navegador, lo que permite ver en la consola los mensajes de logs y errores que programemos.

Podemos considerar dos grupos de pruebas en la fase de testing de Family Care, pruebas unitarias y las pruebas integradas.

Las pruebas unitarias han sido constantes desde el inicio de la implementación. Para la elaboración de estas, nos hemos basado en técnicas de ingeniería del software como son las pruebas en los límites, esto es probar en aquellas casuísticas donde las circunstancias de los datos cambian.

Además, dentro de las **pruebas unitarias**, se pueden distinguir las pruebas de visualización, y las de funcionamiento.

Ejemplos de visualización:

- Formularios.
 - Comprobar tipo de campos.
 - Habilitar/Deshabilitar botones.
 - Funciones de botón cámara.
- Cambio de aspecto en diferentes dispositivos (diferentes resoluciones). Controlado con Media queries en SCSS.
- Etc.

Dentro de las pruebas de funcionamiento, se situarían todas las que se consigue la funcionalidad esperada, por ejemplo:

- Registro de Nuevos usuarios guardado en Firebase.
- Control de mensajes de error cuando usuario introduce un usuario inexistente.
- Inserción de datos en una tabla.
- Lectura de un registro de la base de datos.

Las **pruebas integradas**, se empezaron a realizar en un punto más avanzado de la implementación. Comprobando funcionamiento entre pantallas. Resolviendo los principales problemas de esta app, la compartición de pacientes entre usuarios.

Otras de las pruebas realizadas, son **pruebas de red**. Family Care depende de la conexión a internet para poder funcionar de forma completa. Uno de los problemas que se detectó con estas pruebas, es que Family Care no puede funcionar al completo sin él, ya que aunque Firebase guarda en local los datos, para sincronizarlos una vez tenga red, las funcionalidades de Family Care se basan en la visualización de información que se encuentra en Firebase, por lo que si no hay internet, no se podrá consultar los datos de un paciente. Eso, sí ante pérdida momentáneas de conexión, los datos se sincronizan sin problema.

Pruebas de permisos. Centradas en las acciones que puede realizar cada uno de los roles:

- Creador, todas.
- Cuidador, todas menos eliminar un expediente.
- Visitante, solo visualizar y salir de sala de visitas.

La última fase de pruebas, las componen, lo que podemos llamar **pruebas de usuario**. Se ha facilitado la app bajo plataforma Android, a varios usuarios del público objetivo para que “jueguen” con ella.

De ahí surgieron algunas modificaciones, algunas de visualización, de fechas y formatos de algunos datos, así como comprobamos algunas restricciones que se debe realizar a algunos usuarios.

Acciones restringidas:

- Cerrar expediente. Cuidador.
- Eliminar expediente. Creador.
- Eliminarse uno mismo de la Sala de Visitas, a pesar de ser Visitante. Permitir a un usuario, salir por su propia voluntad.

La última tanda de pruebas, se ha realizado al empaquetar, y recompilar el paquete en otro equipo, comprobando que los emuladores funcionan, y que el acceso a la cámara también.

3.5. Mantenimiento del Servicio Firebase.

A continuación se muestran los datos de la cuenta asociada a la app Family Care:

Usuario:
Contraseña:

Desde ahí se puede gestionar autenticación de usuarios (incluyendo plantillas de correo de activación de cuentas, y recuperación de contraseñas), Realtime Data Base, y el Storage de Firebase.

3.6. Revisión de la planificación

Revisando la planificación inicial contábamos con la siguiente estimación:

- Desarrollo.
 - Entorno y herramientas.
 - (Firebase) Servidor Web y SGBD.
 - Programación APP.
- Pruebas.

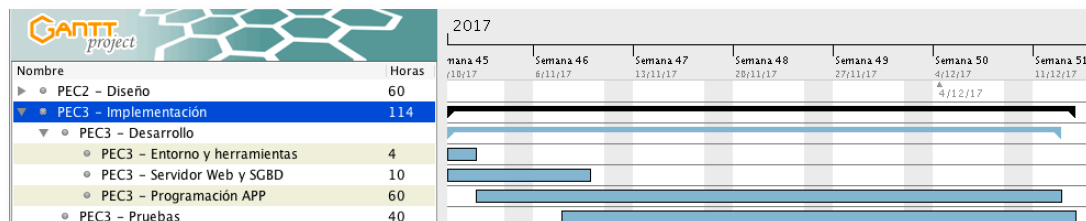


Ilustración 32- Planificación inicial PEC3.

En un inicio, se estimó 10 horas para montar el servicio Web y el Sistema Gestor de Base de Datos. En cambio, montar Firebase para que sea funcional, ha sido más rápido, y en 3 horas se realizó toda la investigación y configuración inicial.

No obstante, han surgido complicaciones con las que no se contaban, y son toda la parte de consulta a la Base de datos *realtime*. Llegar a entender correctamente, e implementar todas las casuísticas de trabajo colaborativo, ha implicado más programación de lo previsto. Esto ha hecho que haya tenido que aumentar las horas de dedicación tanto a pruebas como a programación de la APP. De hecho, he tenido que dedicar un sobreesfuerzo, para poder cubrir de forma apropiada todas las funcionalidades. Pasando la dedicación media diaria de 2, a algo más de 3 horas diarias de lunes a domingo.

La dedicación real ha sido la siguiente:

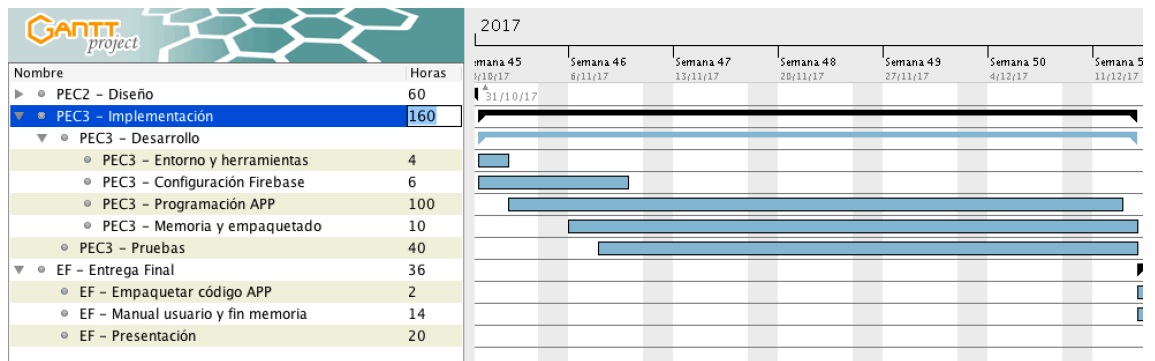


Ilustración 33. Dedicación real en la PEC4.

Cabe destacar que se ha decidido descartar una de las funcionalidades, como ya se ha comentado anteriormente. Hablamos del botón del doctor, que permite acceder a mensajes enviados por un usuario con rol de doctor. Los motivos ya han sido justificados anteriormente en el punto 3.3.2.

4. Conclusiones

Comencemos reflexionando sobre el producto generado. El trabajo realizado en esta asignatura, ha permitido generar una app válida y funcional para cumplir con el objetivo planteado desde el inicio. Una app de atención remota y colaborativa a las enfermedades de un paciente.

El resultado, a nivel personal es bastante satisfactorio, a pesar de haber dejado de lado alguna funcionalidad extra.

Aquellos objetivos secundarios que no hemos logrado, en concreto la funcionalidad del doctor, es como consecuencia de dos factores determinantes, el primero, es el ajustado tiempo restante para entregar la APP. Por otro lado, en base al primero, fue reevaluar la utilidad de esa funcionalidad. Y es que, de alguna manera, ya se ofrece una alternativa a esta funcionalidad, al permitir que un doctor pueda hacer anotaciones en el Área de Paciente. Si bien, lo que sí queda sin realizar es la implementación de notificaciones PUSH, y que queda como una futura mejora.

Si bien, podemos achacar parte de esta falta de previsión, a la metodología empleada, que sin duda no es la más adecuada. Esto se debe a que, como se ha visto en otras asignaturas de este Máster [4], la metodología recomendada en estos proyectos, es la del Desarrollo Ágil, de esta manera haber comenzado antes con la implementación de una versión simple, y esto nos permite ser consciente antes de las mejoras del producto, una vez publicado. Esto se deriva, en parte, del feedback recibido por los usuarios.

En cambio, la planificación de la asignatura, ha obligado a llevar otro método de desarrollo, más parecido al tradicional método en cascada.

No obstante, como todo proyecto, este permite diversas ampliaciones de funcionalidades. A continuación, se muestran una serie de sugerencias:

- Área de doctor con notificaciones PUSH.
- Área de alertas programables. Se podría colocar en el Área de Pacientes, indicando las alertas próximas. Esto implica añadir un botón junto al de Sala de Visitas, para añadir una alerta nueva. Evidentemente, esta alerta haría uso de notificaciones al usuario.
- Habilitar de nuevo la pestaña mediciones, puesto que, en Área de paciente, sólo se mostrarían las últimas 5 mediciones. Parte del espacio de la pantalla se utilizaría para las alertas.
- Una propuesta, por parte de los usuarios, es añadir un campo de pulsaciones dentro de la medición de la tensión.
- Otra propuesta de los usuarios es la posibilidad de exportar a pdf las mediciones de un paciente.

Por último, expresar la satisfacción personal de elaborar un proyecto como este, y obtener un producto útil, real y práctico, y por qué no, posible producto comercial, y que de alguna manera me ha permitido demostrar parte de lo aprendido durante este Máster.

5. Glosario

A

Android Studio: entorno de desarrollo integrado oficial para la plataforma Android autenticación. 46, 47

B

Balsamiq: Balsamiq es una herramienta de prototipado. Reproduce la experiencia de dibujar en una pizarra, pero usando una computadora. 15

D

DCU: Referente a Diseño Centrado en el Usuario.

Diseño Centrado en el Usuario: una forma de diseñar productos “basada en los intereses y las necesidades del usuario, que pone el énfasis en fabricar productos usables y fáciles de entender” 5, 8

E

Evernote es una aplicación informática cuyo objetivo es la organización de información personal mediante el archivo de notas. 2

F

Febrectal: marca de medicamento, cuyo compuesto es el paracetamol, en formato de supositorio. 12

Firebase: es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles. 32, 35, ..

framework: entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

32, 33, 34

I

Interfaz: conjunto de elementos de la pantalla que permiten al usuario interactuar con el sistema informático. 9, 20, 34

L

log: grabación secuencial en un archivo o en una base de datos de todos los acontecimientos (eventos o acciones) que afectan a un proceso particular (aplicación, actividad de una red informática, etc.) 42

M

media queries: Una media query consiste en un tipo de medio y al menos una consulta que limita las hojas de estilo utilizando características del medio, como ancho, alto y color. 42

MVC: referente a Modelo-Vista-Controlador.

Modelo-vista-controlador (MVC): es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. 27, 32

P

PEC: Proceso de evaluación continua en la UOC. 5, 6, 48

PUSH, notificaciones: son mensajes enviados a una aplicación móvil (smartphone o tablet) y que solo se les pueden enviar a los usuarios que hayan instalado la aplicación y hayan aceptado recibir notificaciones push. 27, 41, 49

R

Realtime Database: es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. 32, 39

S

servidor web: es un programa que utiliza el protocolo de transferencia de hiper texto, HTTP (Hypertext Transfer Protocol), para servir los archivos que forman páginas Web a los usuarios, en respuesta a sus solicitudes, que son reenviados por los clientes HTTP de sus computadoras. 6, 48

SGBD: referente a Sistema Gestor de Base de Datos. 6, 48

Sistema Gestor de Base de Datos (SGBD): es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos 6, 48

T

TFM: Trabajo Fin de Máster.

U

UML: referente a Unified Modeling Language

Unified Modeling Language: El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el

lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG). 21, 27, 30

X

XCODE: es un entorno de desarrollo integrado (IDE, en sus siglas en inglés) para macOS que contiene un conjunto de herramientas creadas por Apple destinadas al desarrollo de software para macOS, iOS, watchOS y tvOS. 44, 46

W

Web service: es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. 5

6. Bibliografía

[1] Estadísticas: Niños (hasta 12 años) que reciben cuidados o asisten a centros. Revisado el 4 de Octubre de 2017.

- http://www.ine.es/ss/Satellite?c=INESeccion_C¶m3=1259926137287&p=1254735110672&pagename=ProductosYServicios%2FPYSLayout&cid=1259935025774&L=0

[2] INE: Cantidad de niños en España. Revisado el 4 de Octubre de 2017.

- <http://www.ine.es/prensa/np980.pdf>
- <http://www.ine.es/revistas/cifraine/0309.pdf>

[3] Personas con enfermedades crónicas. Revisado el 4 de Octubre de 2017.

- <http://www.esronicos.com/images/las-cifras/las-cifras.pdf>

[4] Ramírez Vique, R. Métodos para el desarrollo de aplicaciones móviles. Material didáctico. Máster Universitario Desarrollo de Aplicaciones para Dispositivos Móviles. UOC.

[5] Flamarich Zampalo, J. Diseño de interfaces para dispositivos móviles. Máster Universitario Desarrollo de Aplicaciones para Dispositivos Móviles. UOC.

[6] Gil, P.; de Lera, E.; Tatjer, E. Monjo Palau, A. Usuarios y sistemas interactivos. Máster Universitario Desarrollo de Aplicaciones para Dispositivos Móviles. UOC.

[7] Morillo Pozo, JD. Entornos de programación móviles. Material didáctico. Máster Universitario Desarrollo de Aplicaciones para Dispositivos Móviles. UOC.

[8] Firebase. Documentación. Revisado el 10 de Diciembre de 2017.

- <https://firebase.google.com/docs/?hl=es-419>

[9] IONIC. Documentación. Revisado el 11 de Diciembre de 2017.

- <https://ionicframework.com/docs/>

7. ANEXOS.

7.1. Anexo I. Manual de instalación.

La mejor manera de probar Family Care como producto final, es cargar el fichero *FamilyCare.apk* en un dispositivo Android, e instalarlo. Para ello hay que habilitar la opción de *Orígenes desconocidos*, dentro de la configuración de Android, tal y como aparece en la Ilustración 31:

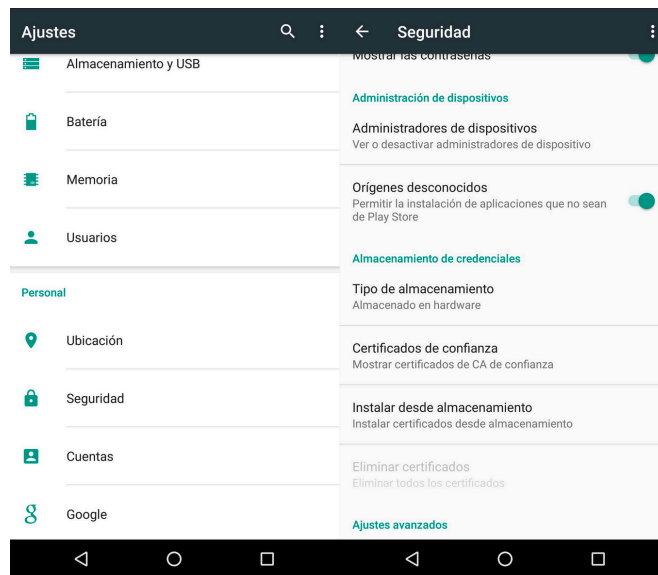


Ilustración 34. Opciones de Seguridad, Orígenes desconocidos en Android.

De esta manera, se podrá hacer uso de la app, utilizando la cámara sin restricciones. Los emuladores tienen el problema de que, o bien hay que configurar la cámara, en el caso del navegador, o que directamente no te permite usarla, como es el caso de XCODE.

Recompilar, y lanzar en el ordenador.

El paquete que se entrega con el código fuente del proyecto, permite recompilar y lanzar la aplicación en local, para probarla bien en el navegador o en un emulador.

Para ello primero hay que instalar nodeJS, de la página oficial. Después habrá que instalar IONIC tal y como se mencionaba en el punto.

```
npm install -g cordova ionic
```

Probablemente pida instalarlo con permisos de administrador (sudo)

Tras esto, descomprimir el proyecto FamilyCare.rar, que generará una carpeta. Desde esa carpeta, en línea de comandos, se deberá lanzar:

```
npm install
```

Esto instalará todos los plugins necesarios para la ejecución de la app. Ahora estamos listos para lanzarla, tenemos varias opciones:

Emular desde el navegador:

```
ionic serve --lab
```

Como se muestra en la Ilustración 32, ese comando lanza la app en el navegador por defecto, mostrando las plataformas configuradas en ionic.

Desde la esquina superior derecha, en Platforms, se puede controlar si queremos mostrar la plataforma Android, iOS, o ambas.



Ilustración 35. Ejecución de Family Care en navegador modo Lab.

Uso de la cámara.

Los emuladores, tienen el inconveniente de que no pueden hacer uso de la webcam, en cambio sí el navegador, pero para ello hay que habilitar el browser como plataforma. Eliminamos el configurado inicialmente para crear uno nuevo, y después lanzarlo:

```
ionic cordova platform rm browser  
ionic cordova platform add browser
```

```
ionic cordova run browser
```

a. Emular desde XCODE o Android Studio:

XCODE.

Se mostrará en el emulador de XCODE, tal y como aparece en la Ilustración 33. Para lanzarlo sobre el dispositivo por defecto basta con lanzar el siguiente comando

```
ionic cordova run ios
```

Si se desea escoger dispositivo, se puede listar los existentes con el comando siguiente:

```
ionic cordova emulate ios --list
```

Y de la lista escoger el dispositivo lanzando el comando anterior de la siguiente manera:

```
ionic cordova run ios --target iPhone-8
```

Otra alternativa, es coger directamente el archivo Family Care, dentro de la carpeta FamilyCare_iOS, que se adjunta en esta PEC, y arrastarlo encima del emulador. Esto instalará la app, y permitirá lanzarla en él.



Ilustración 36. Emulador iPhone 8 Xcode, lanzando Family Care.

ANDROID STUDIO.

Para ello es necesario tener instalado correctamente Android Studio y tener creados algún Virtual Device desde el AVD Manager.

Partiendo de esta situación, basta con lanzar

```
ionic cordova run Android
```



7.2. Anexo II. Manual de usuario.

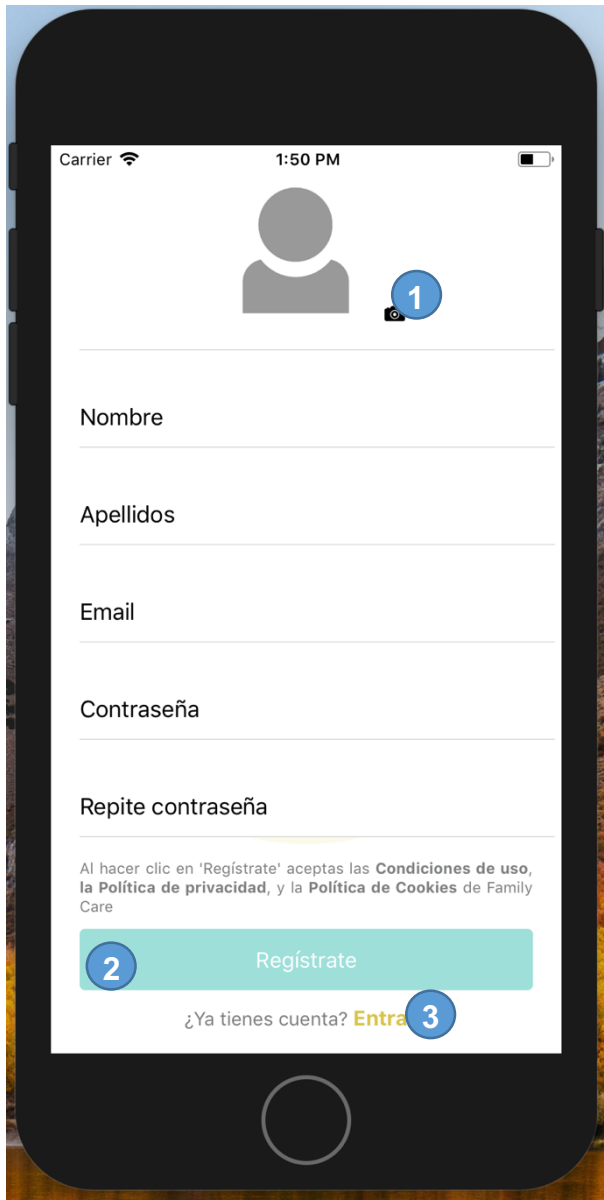
7.2.1. Inicio	55
7.2.2. Registro	56
7.2.3. Recuperar contraseña	58
7.2.4. Añadir un paciente y un proceso medico (Expediente)	59
7.2.5. Modificar un paciente	60
7.2.6. Añadir medición de fiebre	61
7.2.1. Añadir medición de tensión	62
7.2.2. Añadir medición de medicación	63
7.2.3. Añadir foto como medición	64
7.2.4. Añadir usuario a Sala de Visitas	65
7.2.5. Modificar/Eliminar usuario de Sala de Visitas	66
7.2.6. Consultar historial y añadir expediente	67
7.2.7. Cerrar de expedientes	68
7.2.8. Filtrar lista pacientes	69
7.2.9. Cerrar sesión	70

7.2.1. Inicio.

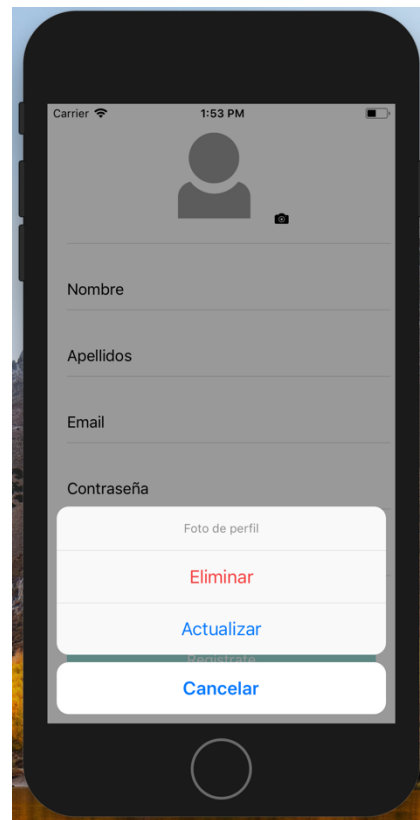


- 1 Toque: va a Registro.
- 2 Toque: va a Autenticación.

7.2.2. Registro



- 1 Toque: Opciones de foto.
 - Actualizar la existente con la cámara.
 - Eliminar la que haya.

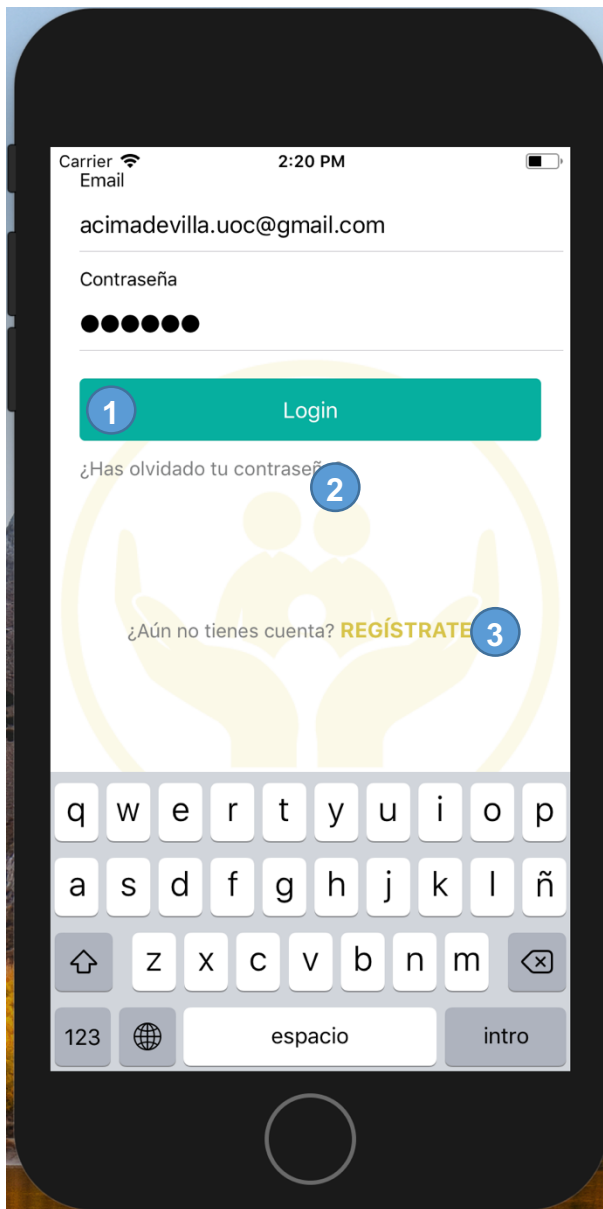


- 2 Toque: realiza registro. Se envía correo de validación de cuenta.

- 3 Toque: va a Autenticación.



Autenticación



- 1 Toque: Si el usuario es válido, accede a lista de pacientes.
- 2 Toque: Va a recuperación de contraseña.
- 3 Toque: va a Registro.

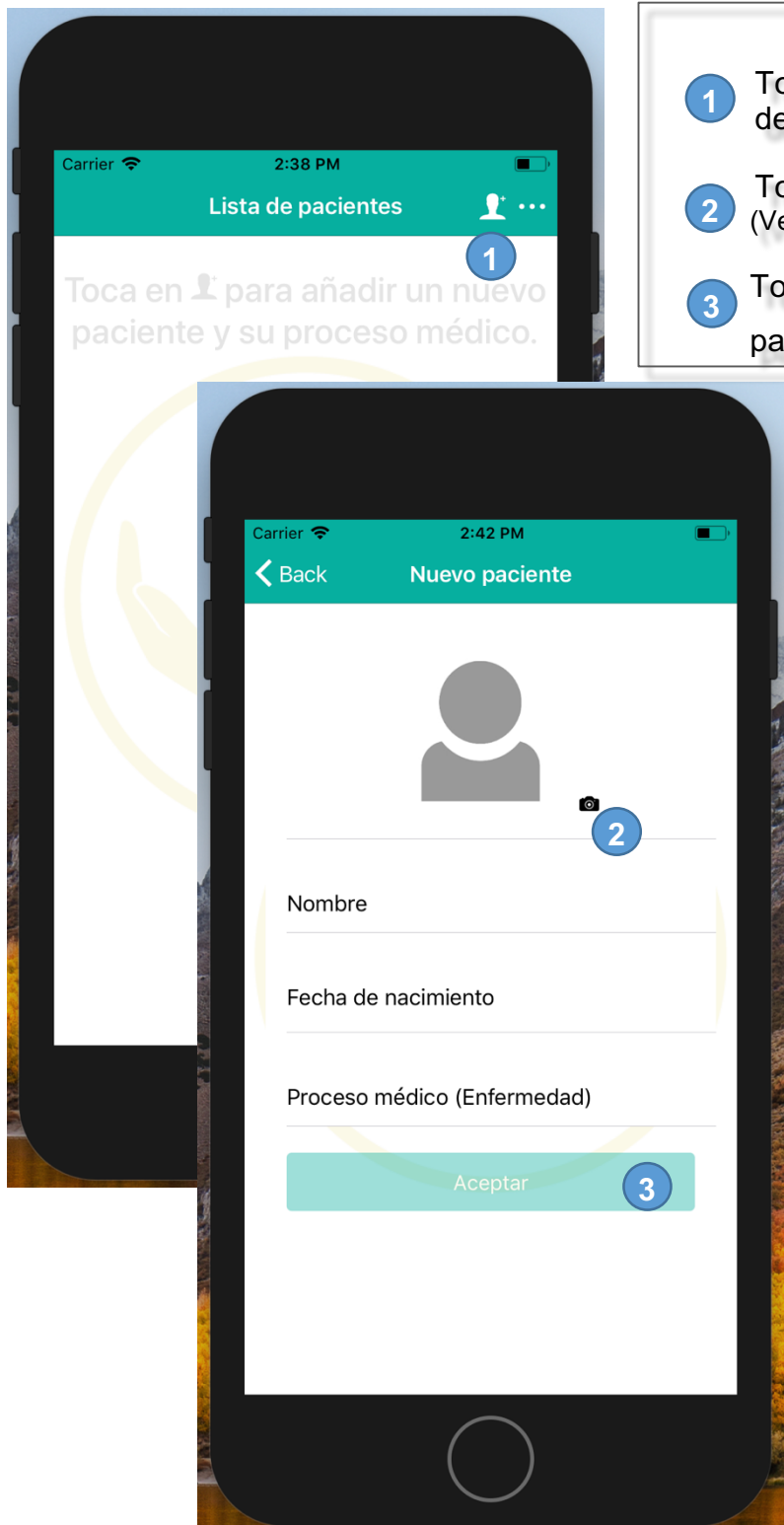
7.2.3. Recuperar contraseña



- 1 Toque: solicita recuperar contraseña, por email.
- 2 Toque: Va a autenticación.
- 3 Toque: va a Registro.

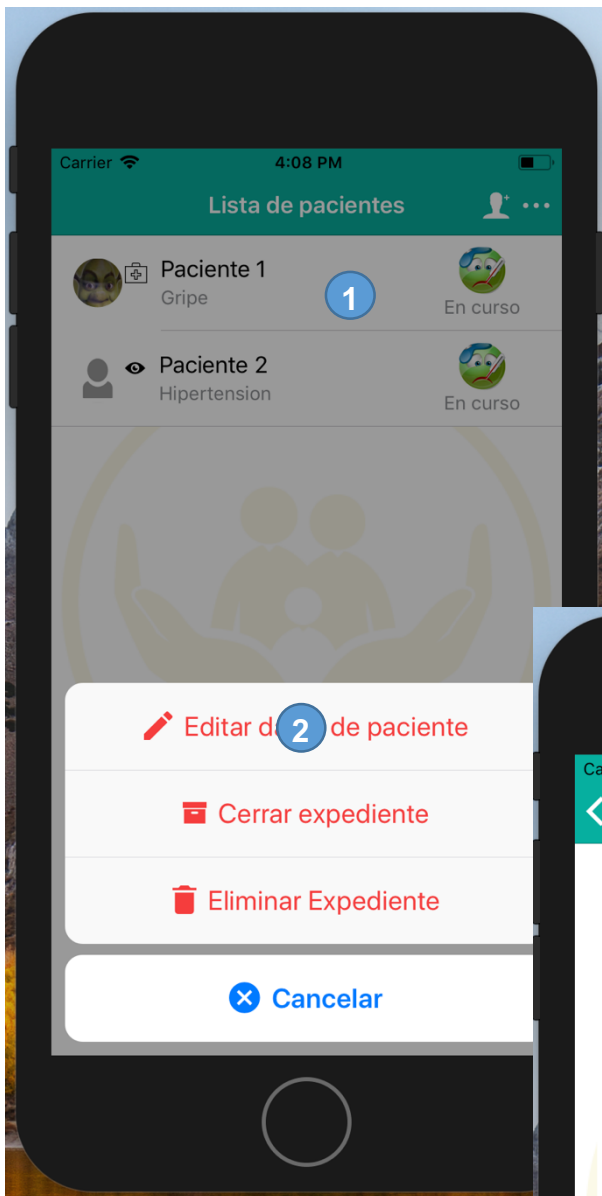


7.2.4. Añadir un paciente y un proceso medico (Expediente).

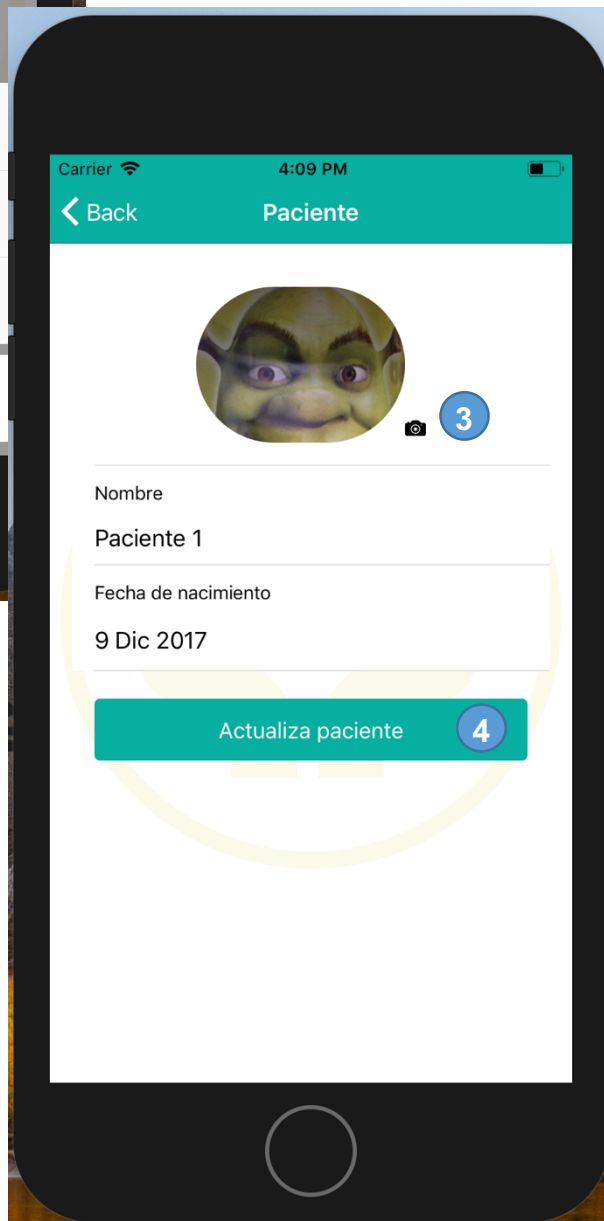


- 1 Toque: Accede a pantalla de nuevo paciente.
- 2 Toque: Opciones de foto. (Ver 7.2.2)
- 3 Toque: Crea nuevo paciente.

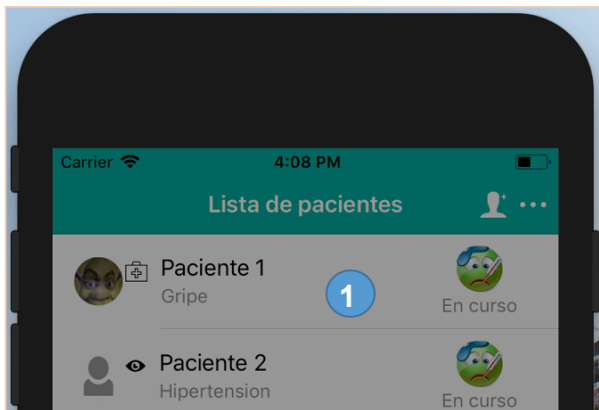
7.2.5. Modificar un paciente.



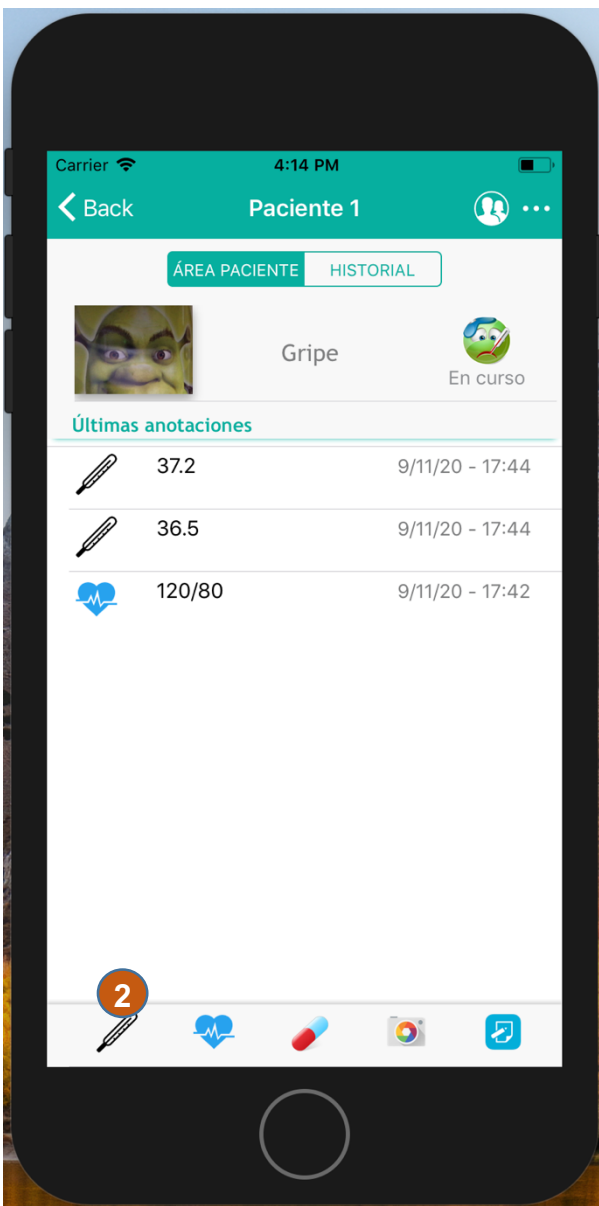
- 1 Toque *mantenido*: Accede a menú.
- 2 Toque: Editar paciente.
- 3 Toque: Opciones de foto. (Ver 7.2.2).
- 4 Toque: Actualizar datos paciente.



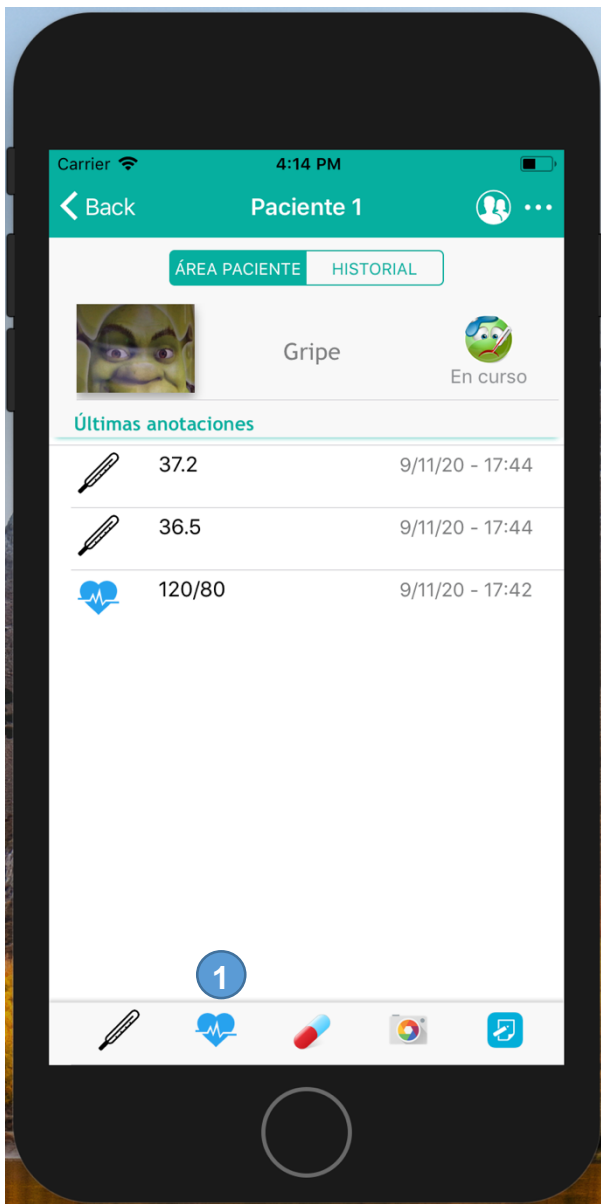
7.2.6. Añadir medición de fiebre.



- 1 Toque: Accede a Área de Paciente.
- 2 Toque: Abre ventana de medición.
- 3 Toque: Aumentar o disminuir temperatura.
- 4 Toque: Fecha y hora.
- 5 Toque: Hacer efectiva la medición.



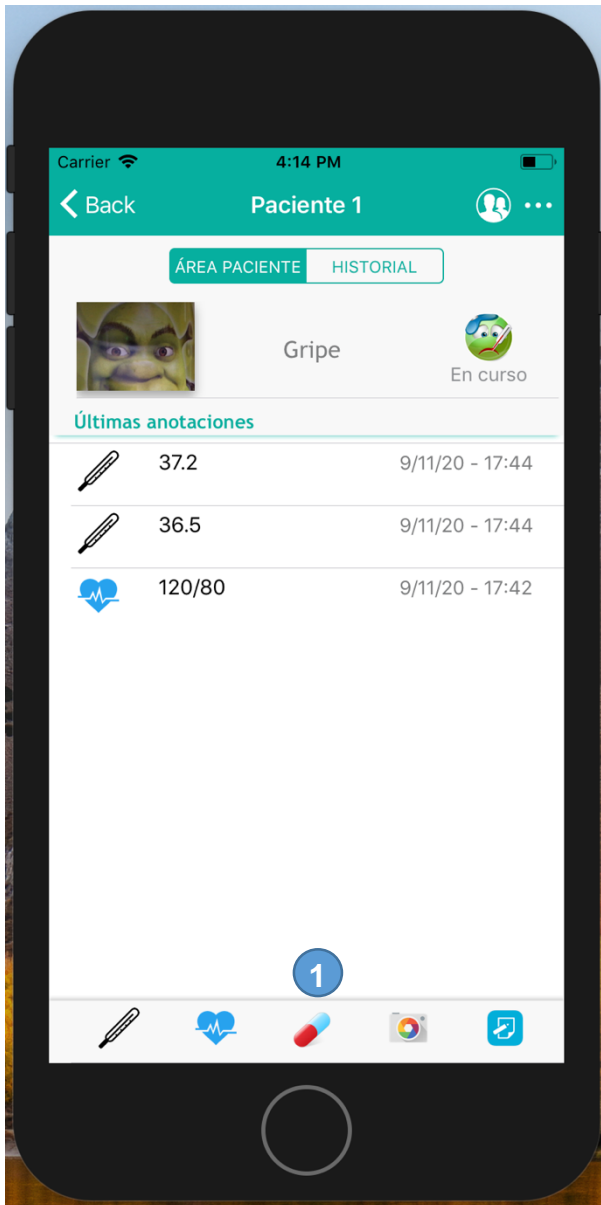
7.2.1. Añadir medición de tensión.



- 1 Toque: Abre ventana de medición.
- 2 *Deslizar*: Aumentar o disminuir tensión.
- 3 Toque: Fecha y hora.
- 4 Toque: Hacer efectiva la medición.



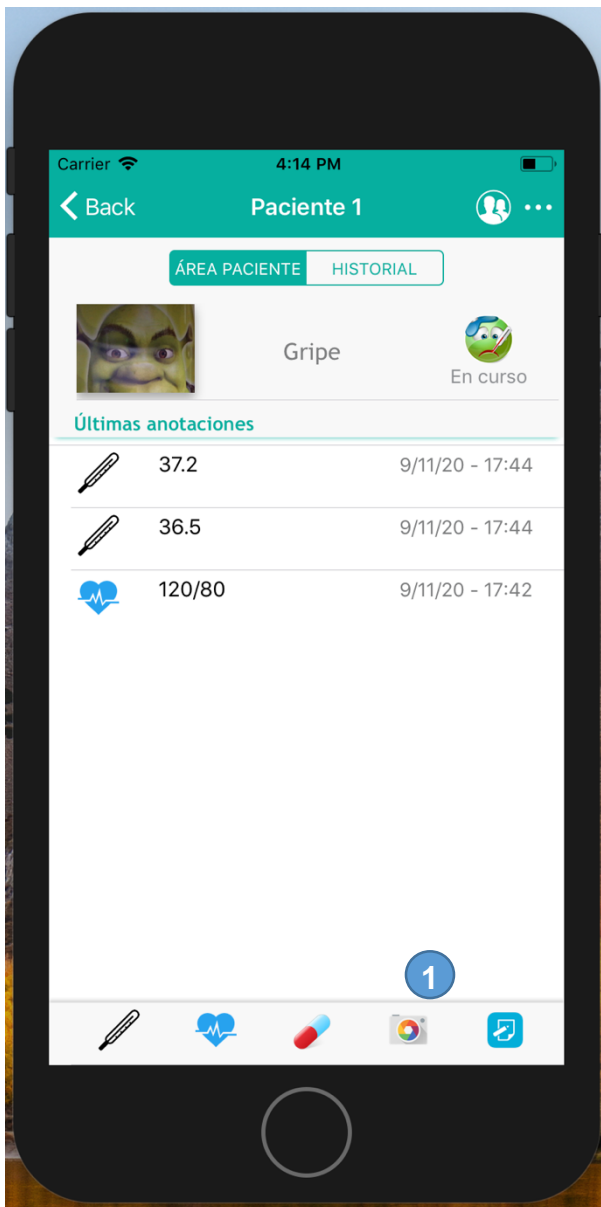
7.2.2. Añadir medición de medicación.



- 1 Toque: Abre ventana de medición.
- 2 *Escribir*: Texto.
- 3 Toque: Fecha y hora.
- 4 Toque: Hacer efectiva la medición.



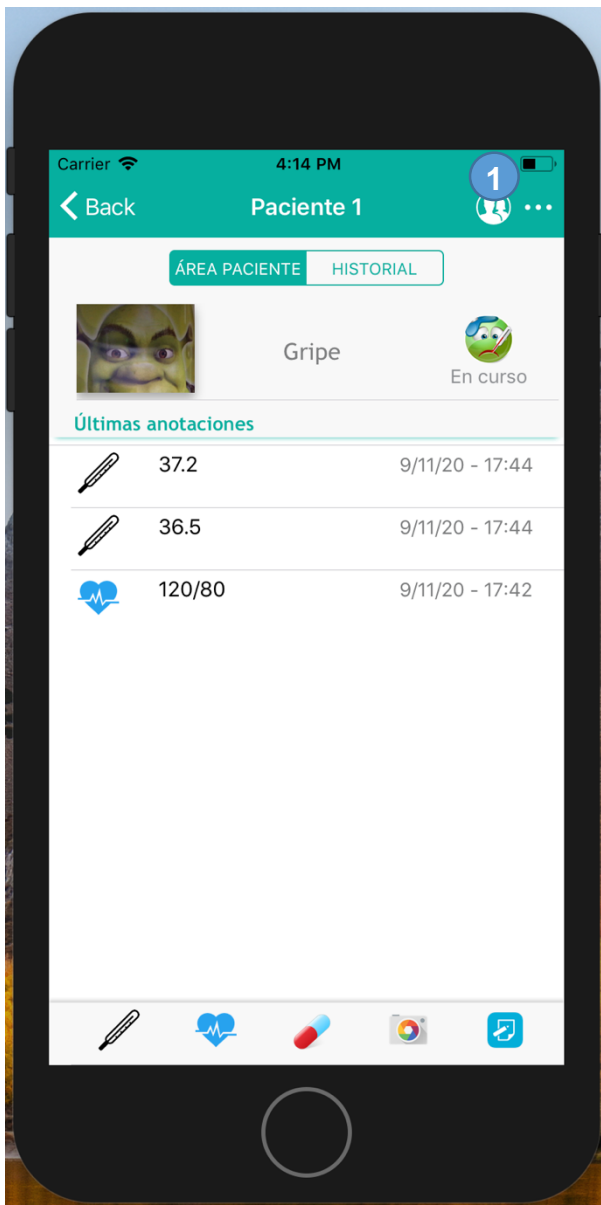
7.2.3. Añadir foto como medición.



- 1 Toque: Abre ventana de medición.
- 2 Toque: Opciones de foto. (Ver 7.2.2).
- 3 Toque: Fecha y hora.
- 4 Toque: Hacer efectiva la medición.



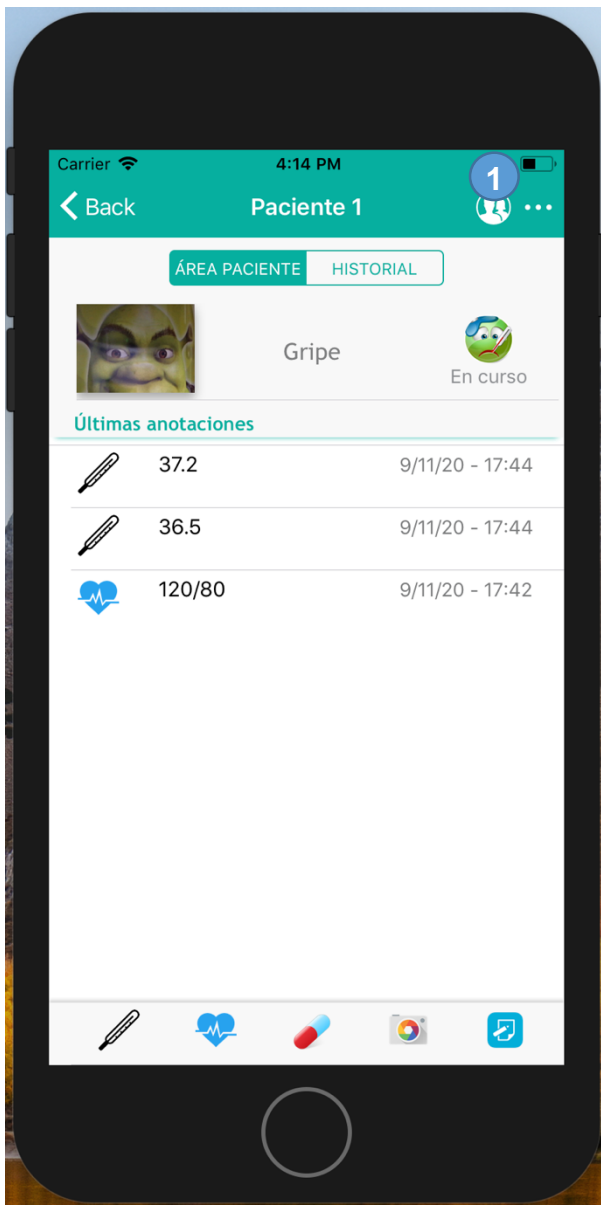
7.2.4. Añadir usuario a Sala de Visitas.



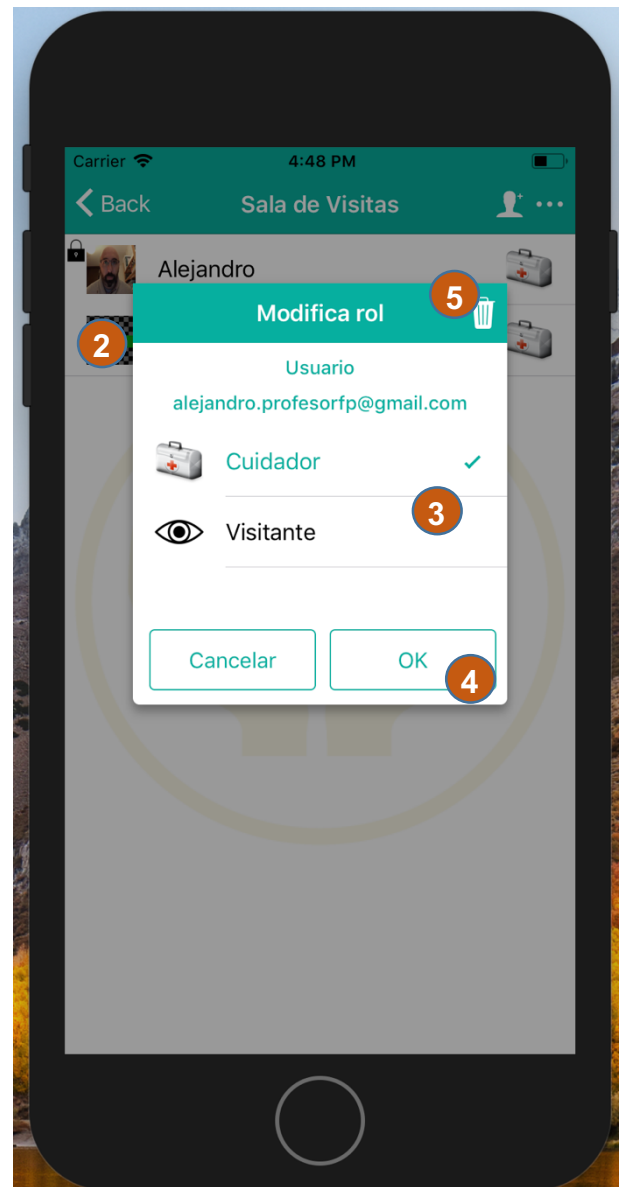
- 1 Toque: Abre ventana de Sala de Visitas.
- 2 Toque: Fecha y hora.
- 3 Toque: Elige rol del usuario:
 - Cuidador: puede hacer cualquier acción.
 - Visitante: solo visualiza información.
- 4 Toque: Hacer efectiva la inserción.



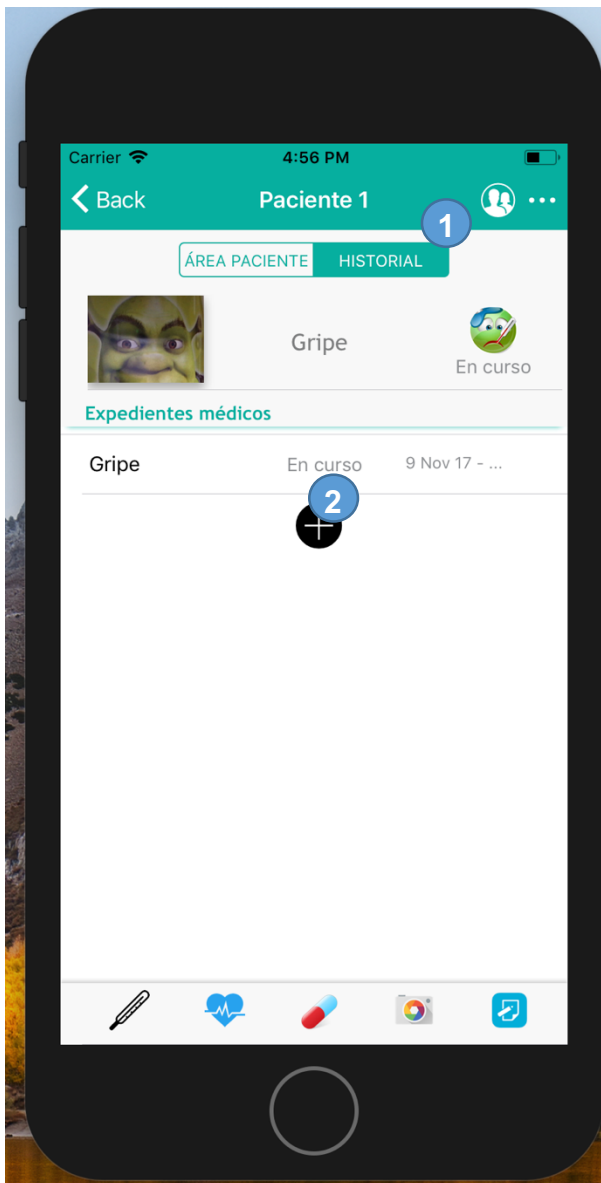
7.2.5. Modificar/Eliminar usuario de Sala de Visitas.



- 1 Toque: Abre ventana de Sala de Visitas.
- 2 Toque en usuario a modificar/sacar de sala de visitas.
- 3 Toque para cambiar rol.
- 4 Toque: Hacer efectivos los cambios.
- 5 Toque: Eliminar usuario de sala de visitas.



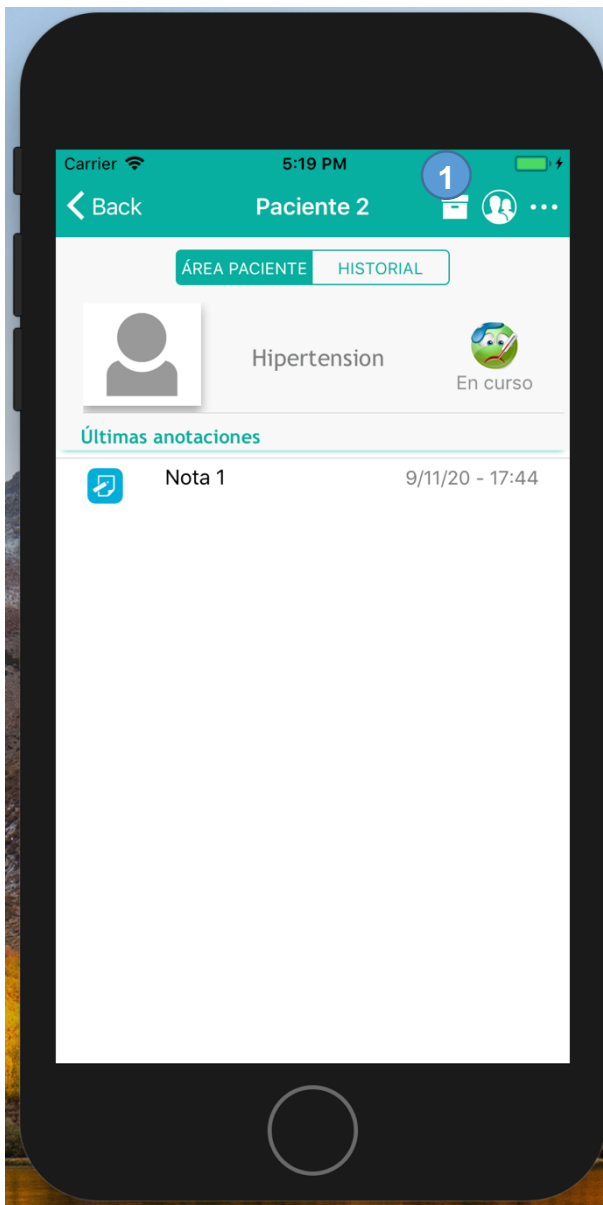
7.2.6. Consultar historial y añadir expediente.



- 1 Toque: Abre ventana de Sala de Visitas.
- 2 Toque: Abre ventana de Nuevo expediente.
- 3 Toque: Tras completar nombre de enfermedad, hace efectiva la creación del nuevo expediente.



7.2.7. Cerrar de expedientes.



Desde Área de paciente.

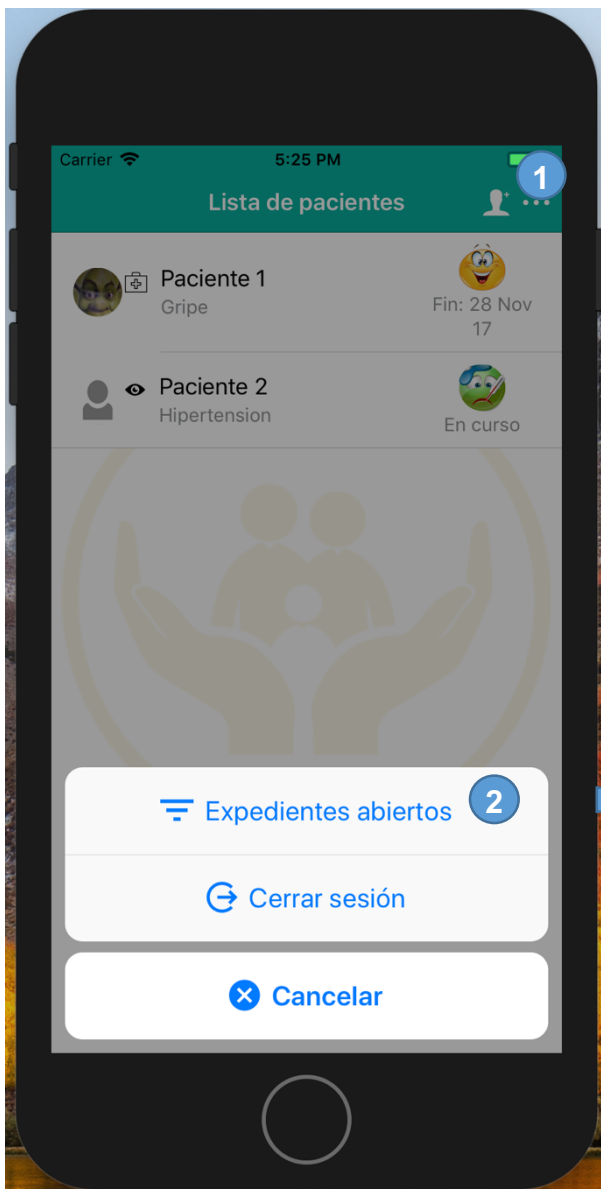
- 1 Toque en icono de archivar:
Cierre de expediente con confirmación.

Desde Lista de pacientes.

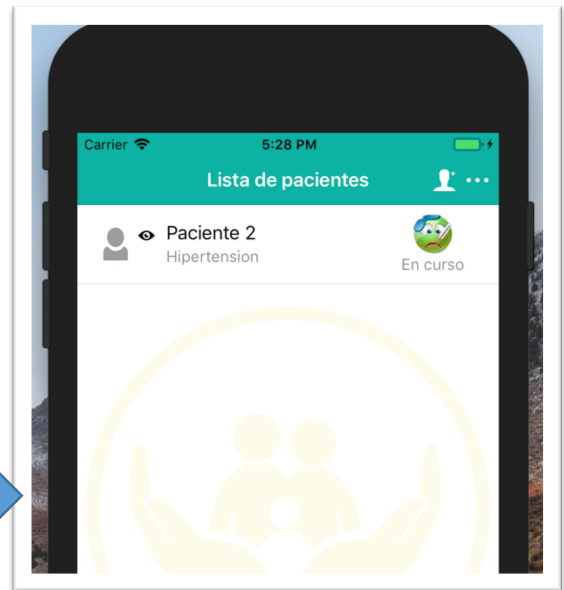
- 1 Toque ***mantenido*** en paciente-expediente.
- 2 Toque ***Cerrar*** en ***expediente***.



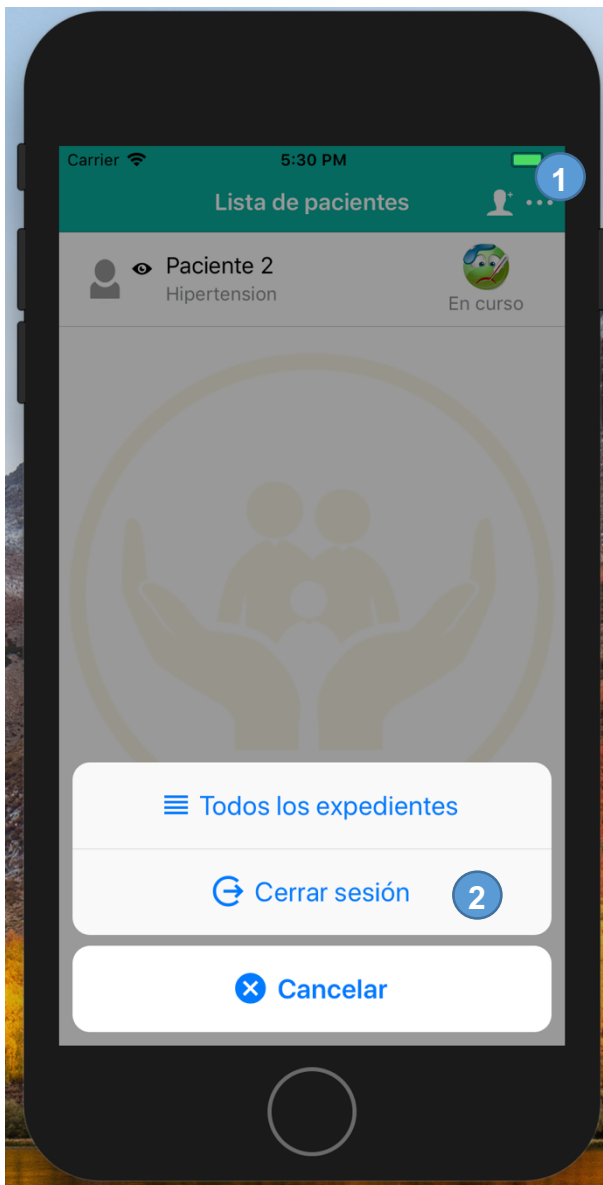
7.2.8. Filtrar lista pacientes.



- 1 Toque en icono de menu.
- 2 Toque para mostrar solo expedientes abiertos.



7.2.9. Cerrar sesión.



Desde cualquier ventana:

- 1 Toque en icono de menu.
- 2 Cerrar sesión