

Grado en Ingeniería Informática



Medspace

Autora: M^a Queralt Sosa Mompel

Tutora: Beatriz Correas Suárez

Co-tutor: Agustín Trujillo Pino

Junio 2021

Resumen

En este Trabajo Fin de Grado (TFG) se implementa una aplicación para uso médico profesional, que pretende ser un espacio en el que mantener informados a médicos de las últimas noticias en su ámbito, resolver casos clínicos, publicar sus últimos artículos de investigación o abrir y participar en debates. El principal objetivo es compartir conocimiento y experiencias valiosas para el resto de la comunidad, interactuar y colaborar con otros compañeros de profesión.

Esta aplicación está destinada para dispositivos iOS, para su desarrollo se ha utilizado el entorno de programación XCode, mientras que los datos implicados en el sistema se almacenan en la plataforma en la nube Firebase de Google.

Palabras clave: Trabajo Fin de Grado, iOS, Firebase, Aplicaciones móviles.

Abstract

In this End of Title Thesis (TFT), it is implemented an application for professional medical use, which aims to be a space to keep doctors informed of the latest news in their field, to solve clinical cases, publish their latest research articles or open and participate in discussions. The main objective is to share knowledge and valuable experiences for the rest of the community, interact and collaborate with other colleagues in the profession.

This application is intended for iOS devices, for its development has been used XCode programming environment and the data involved in the system is stored on Google's Firebase cloud platform.

Keywords: Final Degree Project, iOS, Firebase, Mobile applications.

Tabla de contenidos

Capítulo 1. Introducción	8
Motivación	9
Objetivos	9
Capítulo 2. Estado del arte	10
Aplicaciones móviles para uso médico profesional	11
Medscape	11
DailyRounds - for Doctors	11
Read by QxMD	11
Capítulo 3. Competencias cubiertas y aportaciones sociales	12
Capítulo 4. Análisis de requisitos	13
Usuarios	14
Diagrama de clases	14
Diagramas de casos de uso	15
Especificación de los casos de uso	18
Capítulo 5. Diseño de la aplicación	34
Justificación del diseño de las vistas	34
Pantallas	35
Capítulo 6. Herramientas	40
Entorno XCode	40
Simulador	40
Firebase	40
Git	41
Capítulo 7. Desarrollo de la aplicación	42
Primeros pasos en XCode	42
Diagrama de tablas	44
Preparar la base de datos	46
Registro y autenticación de usuarios	46
Listado de una publicación	47
Creación de una publicación	49
Edición de una publicación	50
Borrado de una publicación	50
Notificación de publicaciones	52
Gestión de la cuenta de usuario	53
Ejecución de la aplicación en un simulador	54

Capítulo 8. Conclusiones	55
Valoración personal	55
Futuras ampliaciones o mejoras	55
Capítulo 9. Bibliografía	56
Anexo. Manual de usuario	58
Registro de una cuenta nueva	58
Inicio de sesión en el sistema	59
Listado de publicaciones	60
Ver publicación en detalle	61
Registro de una publicación	62
Edición de una publicación	64
Borrado de una publicación	65
Buscar una publicación	66
Gestionar comentarios	67
Gestionar palabras clave	68
Darse de baja	69
Actualización del perfil	70
Notificaciones de publicaciones	71

Tabla de figuras

Figura 1. Medscape	11
Figura 2. Daily Rounds - for Doctors	12
Figura 3. Read by QxMD	12
Figura 4. Significado de publicación para los usuarios	15
Figura 5. Diagrama de clases	16
Figura 6. Diagrama de caso de uso: Acciones básicas comunes I	17
Figura 7. Diagrama de caso de uso: Acciones básicas comunes II	17
Figura 8. Diagrama de caso de uso: Gestionar publicación	18
Figura 9. Diagrama de caso de uso: Buscar publicaciones	18
Figura 10. Diagrama de caso de uso: Gestionar comentarios	18
Figura 11. Diagrama de caso de uso: Gestionar notificaciones	19
Figura 12. Diseño de la aplicación: Pantalla de inicio de sesión con o sin errores	36
Figura 13. Diseño de la aplicación: Listado de publicaciones con y sin posts	37
Figura 14. Diseño de la aplicación: Creación de una publicación	38
Figura 15. Diseño de la aplicación: Edición de una publicación	39
Figura 16. Diseño de la aplicación: Actualizar la cuenta de un usuario	40
Figura 17. Desarrollo de la aplicación: Primeros pasos en Xcode I	43
Figura 18. Desarrollo de la aplicación: Primeros pasos en Xcode II	44
Figura 19. Desarrollo de la aplicación: Diagrama de tablas	45
Figura 20. Desarrollo de la aplicación: Registro y autenticación de usuarios	47
Figura 21. Desarrollo de la aplicación: Ejecución de la aplicación en un simulador	55
Figura 22. Anexo: Registro de una cuenta nueva	59
Figura 23. Anexo: Inicio de sesión en el sistema	60
Figura 24. Anexo: Listado de publicaciones	61
Figura 25. Anexo: Ver publicación en detalle	62
Figura 26. Anexo: Registro de una publicación I	63
Figura 27. Anexo: Registro de una publicación II	64
Figura 28. Anexo: Edición de una publicación	65
Figura 29. Anexo: Borrado de una publicación	66
Figura 30. Anexo: Buscar una publicación	67
Figura 31. Anexo: Gestionar comentarios	68
Figura 32. Anexo: Gestionar palabras clave	69
Figura 33. Anexo: Darse de baja	70
Figura 34. Anexo: Actualización del perfil	71
Figura 35. Anexo: Notificaciones de publicaciones I	72
Figura 36. Anexo: Notificaciones de publicaciones II	73

Capítulo 1. Introducción

Desde el inicio de la pandemia mundial a causa del COVID-19, la tecnología ha tenido un mayor papel en la actualidad, ha cambiado la forma en la que interactuamos con la tecnología.

En medio de esta crisis sanitaria, pasamos más tiempo con el móvil, de acuerdo con un informe de *App Annie*, el tiempo de uso de las apps incrementó un 20% interanual durante el primer trimestre de 2020 y el gasto de los consumidores incrementó un 15% en dispositivos iOS y un 5% para dispositivos Android.

Los sanitarios también han tenido que cambiar su forma de relacionarse con la tecnología para poder compaginar su formación con los largos turnos en el centro médico, es por ello que consideré que mi mayor aportación con este proyecto fin de grado, era crear un espacio para que los médicos encuentren una plataforma en la que seguir informados de los últimos avances y noticias en la Medicina, interactuar con otros compañeros de su profesión y compartir experiencias de forma sencilla y fácil teniendo a mano la app.

A lo largo de este documento, veremos el proceso de diseño e implementación de este sistema, donde se analizan requisitos, los problemas a los que aporta una solución, se describe en detalle su funcionalidad y diseño minimalista.

Motivación

Las razones que me han llevado a crear este espacio médico han sido mi pasión por la programación de aplicaciones móviles, más concretamente de los dispositivos iOS y por el mundo de la Medicina.

Google Play cuenta con aproximadamente 2.7 millones de apps, la App Store unos 1.8 millones, se publican más aplicaciones para los dispositivos Android, pero las apps para iOS ofrecen una mayor calidad y un mejor servicio, ya que garantizan unos mínimos de calidad, por ello, es una buena opción implementar una app para esta plataforma.

El día a día del personal sanitario actualmente es realmente dura, y aún así, siempre siguen formándose continuamente, los últimos estudios, herramientas, nuevas tecnologías. *Medspace* nace para formar una comunidad sólida “de médicos para médicos”, compartir y adquirir conocimiento, puntos de vista, experiencias y consejos.

Objetivos

El principal objetivo de este proyecto es construir un espacio para que doctores puedan interactuar, divulgar su conocimiento en las distintas áreas médicas y ofrecer ayuda a otros profesionales. Para desarrollar una aplicación que permita satisfacer esto, el sistema que se ha de desarrollar deberá alcanzar las siguientes tres principales metas:

- Autenticar, registrar y gestionar la cuenta de un usuario.
- Participación activa y colaboradora de los doctores.
- Gestión de las publicaciones de contenido médico y de sus respectivos comentarios.

Capítulo 2. Estado del arte

En este apartado se presenta un estudio sobre cómo se encuentra el mercado de las aplicaciones móviles para uso médico profesional presentando las apps con mayores descargas en la App Store.

Aplicaciones móviles para uso médico profesional

En el ámbito médico existen diversas aplicaciones para iOS, que brindan acceso a conocimiento médico y múltiples foros que facilitan el intercambio de comunicación entre profesionales. A continuación, veremos tres ejemplos de apps:

Medscape



Una aplicación con las mejores calificaciones en la categoría de Medicina en la App Store. Se conoce por ser la guía farmacológica más usada por médicos, ofrece las últimas noticias, comparte los últimos hallazgos clínicos, y la visión de cada profesional en los avances médico-tecnológicos.

Figura 1. Medscape

DailyRounds - for Doctors



Una aplicación basada en la publicación de casos clínicos para resolver, indicada para médicos y estudiantes de Medicina. Se convierte, por tanto, en una herramienta para aprender y compartir casos de distintas especialidades clínicas, ayudando a muchos estudiantes en la resolución de estos.

Figura 2. Daily Rounds - for Doctors

Read by QxMD



Esta aplicación tiene como principal propósito que los profesionales de la salud estén atentos de las últimas investigaciones, se fundamenta en la lectura de estos artículos y de revistas médicas, así como sus reseñas. Facilita que los doctores compartan su trabajo de investigación con el resto de usuarios.

Figura 3. Read by QxMD

Estas aplicaciones citadas tienen buenas calificaciones en la App Store y una estrecha relación con *Medspace*. Esta app supone una herramienta útil y práctica que proporciona ayuda y conocimiento, contiene novedades tecnológica-médicas, permite al doctor continuar su formación, mantener el contacto y ayudar a los demás profesionales.

Este proyecto es la mejor aplicación que puede tener un médico en su móvil, pues a diferencia del resto, recoge diversas funcionalidades en un único espacio, lo que supone al usuario no tener descargadas muchas apps similares, permite una mayor personalización del contenido, pues el usuario decide la temática de las publicaciones de su interés y de las que desea recibir notificaciones, aspecto que ninguna de las anteriores aplicaciones posee, tiene una interfaz más intuitiva que el resto, lo que aumenta la sencillez del acceso a los datos y aumenta la productividad y eficiencia de los médicos, pues de forma relativamente rápida, se puede consultar gran cantidad de datos.

Capítulo 3. Competencias cubiertas y aportaciones sociales

Las competencias específicas que incluye mi mención de Ingeniería del Software presentes en mi Trabajo Fin de Grado son las siguientes:

- **Competencia IS01.** Relacionada con la capacidad de desarrollar, mantener y evaluar software que satisfaga los requisitos del usuario, se comporten de forma fiable y cumpla normas de calidad.
- **Competencia IS02.** Relacionada con la capacidad de valorar y resolver las necesidades del cliente.
- **Competencia IS03.** Relacionada con la capacidad de dar solución a problemas de integración en función de las estrategias estándares y tecnologías disponibles.
- **Competencia IS04.** Relacionada con la capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software.
- **Competencia IS05.** Relacionada con la capacidad de identificar, evaluar y gestionar los riesgos potenciales asociados que pudieran presentarse.
- **Competencia IS06.** Relacionada con la capacidad de diseñar soluciones apropiadas en uno o más dominios de aplicación.
- **Competencia TFG01.** Relacionada con el ejercicio de realizar, presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática.

Medspace es un software implementado desde cero, es por ello que ha pasado todas las fases de desarrollo: análisis, diseño, implementación y testado. Así mismo la memoria presente recoge la documentación software necesaria para crear esta app, vemos como dispone de un fase de análisis de requisitos, las soluciones que ofrece, pautas de su diseño e implementación final, respecto a esto último, se han utilizado herramientas y entornos de programación oficiales y fiables para el desarrollo de aplicaciones iOS.

En este proyecto se debaten diversas cuestiones médicas sobre artículos, casos, noticias y se comparten críticas constructivas a través del *feedback* de la publicación, por lo que a raíz de esto, se construye una comunidad que fomenta la creación de nuevo conocimiento y finalmente esto contribuye a que los médicos se especialicen en la materia, lo que repercute en un mejor diagnóstico y tratamiento del paciente, aunque el hecho de compartir conocimiento a través de las publicaciones ya es en sí misma un gran aporte al mundo de la Medicina.

Capítulo 4. Análisis de requisitos

El análisis de requisitos es la fase más importante en el desarrollo de un software, puesto que durante esta, se realiza un estudio completo de las funcionalidades y características que debe incluir el sistema para satisfacer los problemas y necesidades de sus usuarios.

Usuarios

En este proyecto contamos con dos tipos de usuarios: doctores y un administrador. El administrador es único en el sistema, por lo que no puede darse de baja, mientras, los doctores podrán registrar o eliminar su cuenta del sistema.

Una publicación puede ser una noticia, un caso clínico, un debate o un artículo de investigación (véase la figura 4), todos los usuarios pueden acceder a cualquier tipo de publicación, realizar búsquedas, gestionar su contenido, publicar un comentario en ellas. Además, ambos roles pueden configurar los datos de su cuenta. Los administradores son exclusivamente, los que pueden crear noticias, los doctores podrán compartir el resto de publicaciones, esto es, casos, debates y artículos. Los creadores de estos contenidos respectivamente podrán modificarlo o eliminarlo permanentemente.



Figura 4. Significado de publicación para los usuarios

Diagrama de clases

Los diagramas de clase describen las relaciones entre las clases que forman parte del sistema. Una clase puede poseer atributos, por ejemplo “id” de tipo *String*, o una especialidad de tipo “*Speciality*”. En este sistema, las relaciones que predominan son la generalización y la composición.

Una relación de generalización representa a una superclase de la que derivan subclases, las cuales heredan sus atributos e implementación, pero poseen los suyos propios particulares, en nuestro caso, podemos observar como de la superclase “*User*”, derivan “*Administrator*” y “*Doctor*”, dos roles que participan de forma distinta en la aplicación. Una relación de composición es una relación en la que la existencia de una clase depende de la existencia de otra, si un usuario da de baja su cuenta, se eliminan las publicaciones que ha compartido. Por otra parte la clase “*Speciality*”, es de tipo enumeración, puesto que podemos tener 18 tipos diferentes de especialidades, esto es una forma de representar los datos, directamente por sus valores. Veamos cómo se comportan las clases de este sistema:

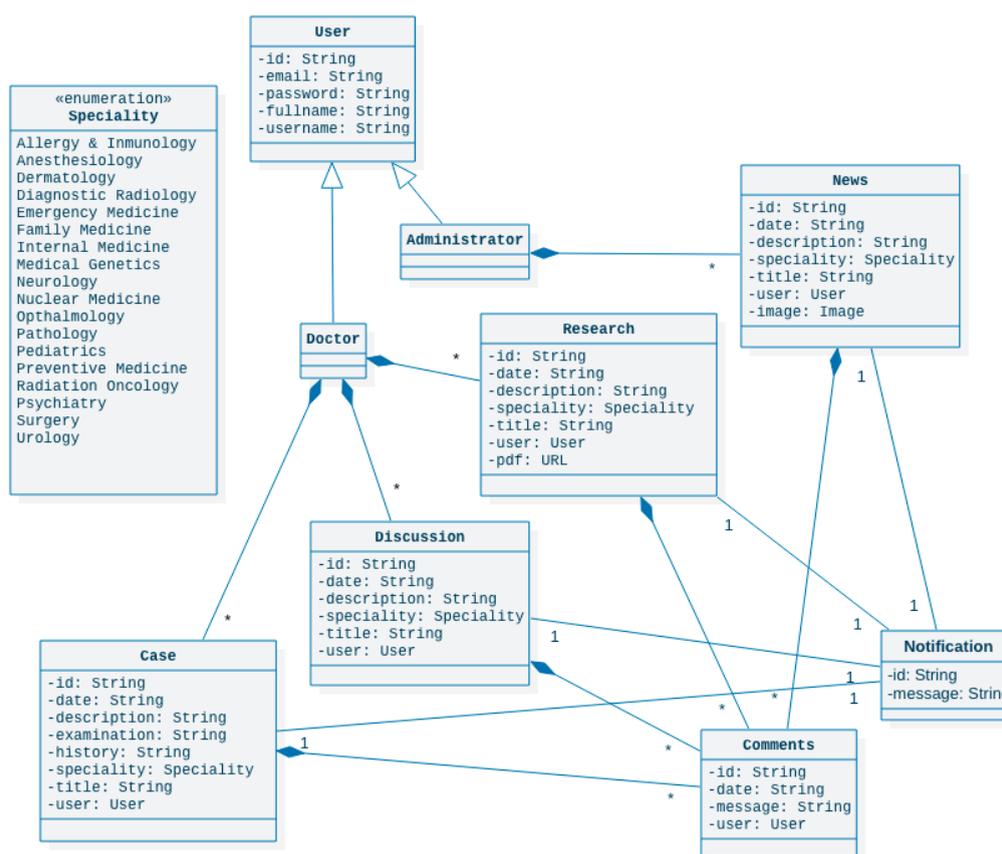


Figura 5. Diagrama de clases

Diagramas de casos de uso

Los casos de uso describen las posibles interacciones o relaciones de los usuarios en el sistema, que pueden ser de asociación, inclusión o extensión. La relación de asociación es una relación básica que se da entre dos casos de uso o entre un actor y un caso de uso, la relación de inclusión (include) es una interacción entre dos casos de uso, donde la invocación de una operación de uno depende del resultado del caso de uso incluido, por ejemplo, para ver una determinada publicación en detalle, tendremos primero que seleccionar una (caso incluido), por último, tenemos la relación de extensión (exclude) donde la invocación de un caso puede producir que se lleve a cabo otra acción, depende del usuario, por ejemplo, al visualizar la información de una publicación en detalle podemos si así lo deseamos, publicar un comentario en esta.

Los actores son los tipos de rol que toma el usuario de la aplicación, quien satisface sus necesidades mediante las funcionalidades que presenta el sistema. Tenemos dos tipos de actores: “Administrador” y “Doctor”, los cuales heredan de un tercer actor denominado “Usuario”. Decimos que esto es una relación de generalización porque tanto el administrador como el doctor heredan los casos de uso del actor general (usuario) o lo que es lo mismo, llevan a cabo acciones comunes, como por ejemplo iniciar sesión, mientras que cada rol especializado (administrador y doctores) tendrá sus propios casos de uso o acciones que sólo éste pueda realizar, es decir, un administrador puede publicar una noticia, pero un doctor no.

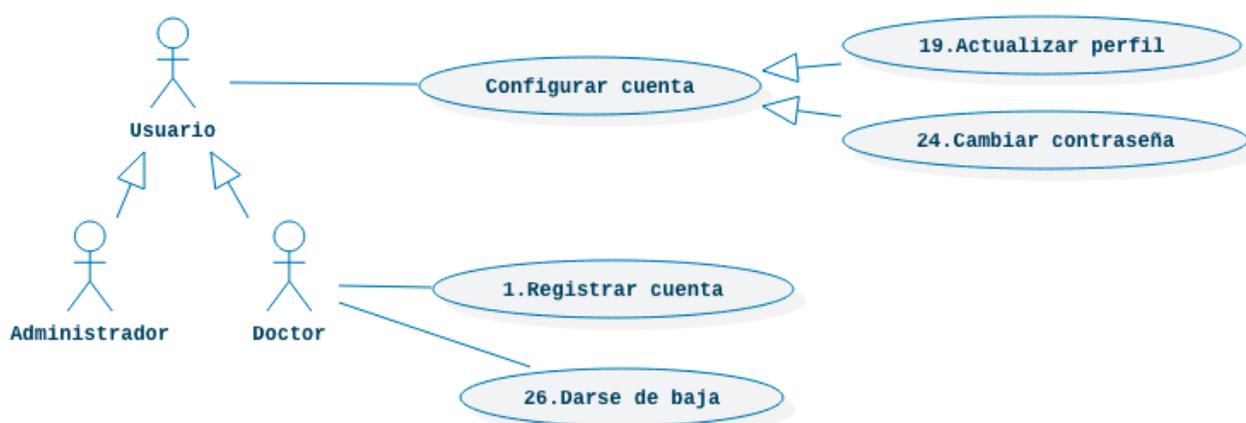


Figura 6. Diagrama de caso de uso: Acciones básicas comunes I



Figura 7. Diagrama de caso de uso: Acciones básicas comunes II

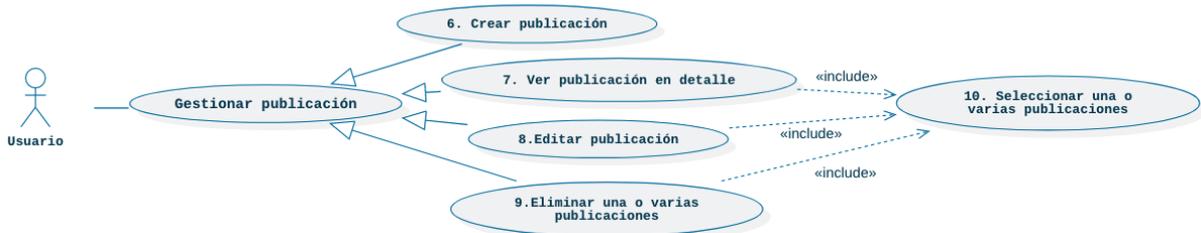


Figura 8. Diagrama de caso de uso: Gestionar publicación

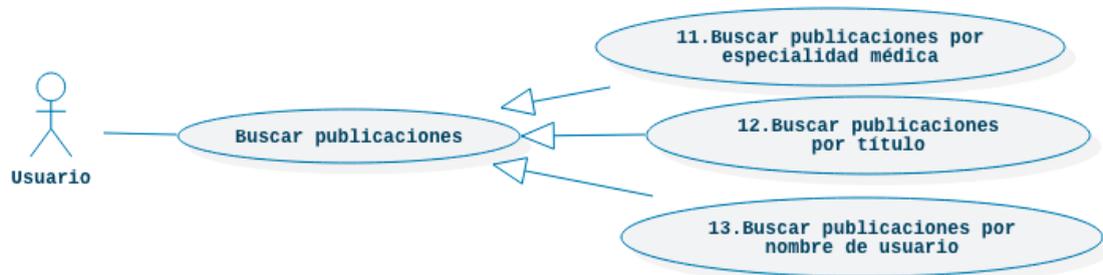


Figura 9. Diagrama de caso de uso: Buscar publicaciones

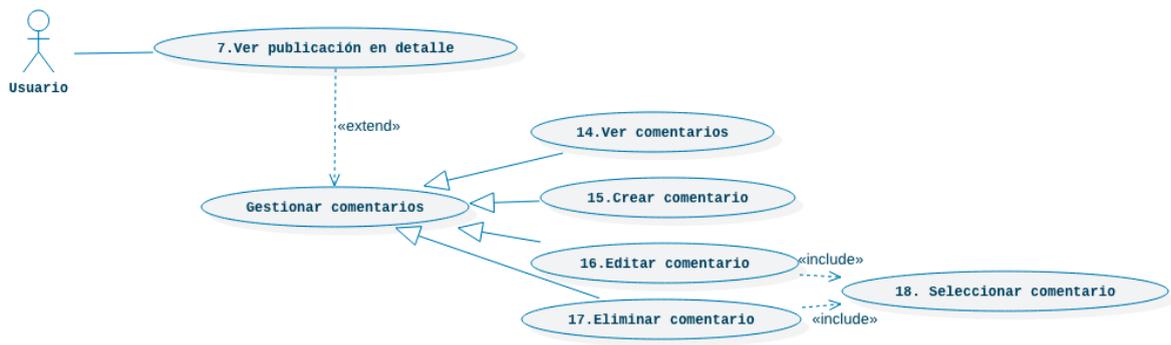


Figura 10. Diagrama de caso de uso: Gestionar comentarios

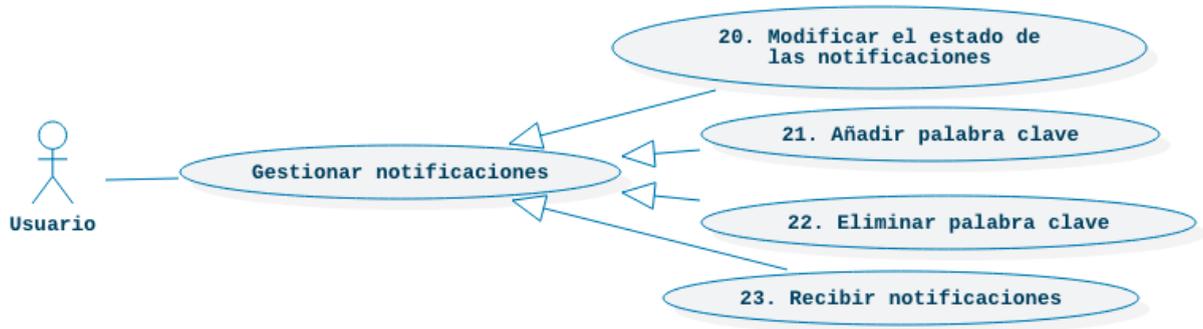


Figura 11. Diagrama de caso de uso: Gestionar notificaciones

Especificación de los casos de uso

Las especificaciones de casos de uso facilita la comprensión del funcionamiento del sistema y expresa la intención con la que el usuario interactúa con la aplicación. En este apartado se incluirán las especificaciones de los casos de uso definidos anteriormente:

ID: 1
Nombre: Registrar una cuenta
Descripción: Permite a un usuario crear una cuenta nueva en el sistema
Actores: Doctor
Precondiciones: El usuario debe encontrarse en el formulario de registro de cuentas
Flujo normal: <ol style="list-style-type: none">1. El usuario introduce los datos requeridos y pulsa “<i>Sign Up</i>”.2. El sistema comprueba la validez de los datos introducidos.3. El sistema registra un nuevo usuario.4. El sistema notifica al usuario si desea iniciar sesión.5. El usuario pulsa “<i>Yes</i>”.6. El sistema presenta la página de inicio.
Flujo alternativo: <ol style="list-style-type: none">1.1 El usuario introduce un nombre de usuario incorrecto.1.2 El usuario introduce un nombre de usuario ya existente.1.3 El usuario introduce un email incorrecto.1.4 El usuario introduce un email ya existente.1.5 El usuario introduce contraseñas incorrectas.1.6 El usuario introduce contraseñas distintas.2.1-2.6 El sistema notifica al usuario un mensaje de error.5.1 El usuario pulsa “<i>No</i>”.5.2 El sistema elimina la notificación de la pantalla.
Postcondiciones: El sistema registra un nuevo usuario en el sistema

ID: 2
Nombre: Iniciar sesión
Descripción: Permite a un usuario loguearse en el sistema
Actores: Usuario
Precondiciones: El usuario debe tener registrada una cuenta y encontrarse en el formulario de inicio de sesión
Flujo normal: <ol style="list-style-type: none"> 1. El usuario introduce los datos requeridos y pulsa “<i>Sign In</i>”. 2. El sistema comprueba la validez de los datos introducidos. 3. El sistema redirige al usuario a la página de inicio.
Flujo alternativo: <ol style="list-style-type: none"> 1.1 El usuario introduce un email no registrado. 1.2 El usuario introduce una contraseña incorrecta. 2.1-2.2 El sistema notifica al usuario un mensaje de error.
Postcondiciones: El sistema logea al usuario en la aplicación

ID: 3
Nombre: Cerrar sesión
Descripción: Permite a un usuario desconectarse del sistema
Actores: Usuario
Precondiciones: El usuario debe estar logueado en el sistema
Flujo normal: <ol style="list-style-type: none"> 1. El usuario despliega el menú lateral y selecciona “<i>Log Out</i>”. 2. El sistema elimina las variables de sesión del usuario y lo redirige al formulario de inicio de sesión.
Flujo alternativo: -
Postcondiciones: El sistema cierra la sesión del usuario

ID: 4
Nombre: Ver todas las publicaciones
Descripción: Permite al usuario visualizar el listado de todas las publicaciones
Actores: Usuario
Precondiciones: El usuario debe estar logueado en el sistema
Flujo normal: <ol style="list-style-type: none"> 1. El usuario despliega el menú lateral y pulsa “Home”. 2. El sistema realiza una lectura de todos los registros de las publicaciones. 3. El usuario visualiza un listado con los datos de las publicaciones.
Flujo alternativo: <ol style="list-style-type: none"> 2.1 El doctor despliega el menú lateral, pulsa “Posts” y selecciona el tipo de publicación: “Clinical cases”, “Discussions” o “Researches”. 3.1 El sistema notifica al usuario de que no existe ninguna publicación.
Postcondiciones: El sistema presenta el listado de publicaciones compartidas

ID: 5
Nombre: Ver mis publicaciones
Descripción: Permite al usuario visualizar el listado de todas sus publicaciones
Actores: Usuario
Precondiciones: El usuario debe estar logueado en el sistema
Flujo normal: <ol style="list-style-type: none"> 1. El usuario despliega el menú lateral, pulsa “My posts” (en caso de que el usuario sea un doctor, selecciona además el tipo de publicación: “Clinical cases”, “Discussions” o “Researches”). 2. El sistema realiza una lectura de todos los registros de las publicaciones pertenecientes al usuario logueado. 3. El usuario visualiza un listado con los datos de todas las publicaciones que ha compartido.
Flujo alternativo: <ol style="list-style-type: none"> 3.1 El sistema notifica al usuario de que no ha compartido ninguna publicación
Postcondiciones: El sistema presenta al usuario el listado de todas las publicaciones que ha compartido.

ID: 6
Nombre: Crear publicación
Descripción: Permite a un usuario registrar una nueva publicación
Actores: Usuario
Precondiciones: El usuario debe estar logueado en el sistema
Flujo normal: <ol style="list-style-type: none"> 1. El usuario despliega el menú lateral y pulsa “Add post” (en caso de que el usuario sea un doctor, selecciona además el tipo de publicación: “Clinical cases”, “Discussions” o “Researches”). 2. El sistema redirige al usuario al formulario de registro de la publicación. 3. El usuario completa los campos requeridos y pulsa “Save”. 4. El sistema notifica un mensaje de confirmación de la publicación. 5. El usuario confirma la notificación. 6. El sistema registra un nuevo post. 7. El sistema redirecciona al usuario al listado de todas sus publicaciones.
Flujo alternativo: <ol style="list-style-type: none"> 5.1 El usuario rechaza la notificación 6.1 El sistema cierra la notificación
Postcondiciones: El sistema registra una nueva publicación

ID: 7
Nombre: Ver publicación en detalle
Descripción: Permite al usuario visualizar la información completa de una publicación
Actores: Usuario
Precondiciones: El usuario debe estar logueado en el sistema y encontrarse ante el listado de publicaciones
Flujo normal: <ol style="list-style-type: none"> 1. El usuario selecciona una publicación. 2. El sistema realiza una lectura del registro de la publicación y redirige al usuario a la vista en detalle de la publicación. 3. El sistema visualiza los datos asociados a la publicación.
Flujo alternativo: -
Postcondiciones: El sistema presenta al usuario toda la información asociada a una publicación.

ID: 8
Nombre: Editar publicación
Descripción: Permite al usuario editar una publicación que haya seleccionado
Actores: Usuario
Precondiciones: El usuario debe encontrarse en la vista en detalle de la publicación seleccionada y ser su autor
Flujo normal: <ol style="list-style-type: none"> 1. El usuario pulsa el icono de editar del menú superior. 2. El sistema redirige al usuario al formulario de edición de la publicación. 3. El usuario modifica su información y pulsa “Done”. 4. El sistema muestra un mensaje de confirmación de la modificación. 5. El usuario confirma la notificación. 6. El sistema redirige al usuario a la vista de la publicación en detalle actualizada.
Flujo alternativo: <ol style="list-style-type: none"> 3.1 El usuario no modifica ninguno de los campos de la publicación. 3.2 El usuario deja algún campo vacío. 4.1 El sistema notifica al usuario un mensaje de error. 4.2 El sistema notifica al usuario un mensaje de error. 5.1 El usuario rechaza la notificación. 6.1 El sistema cierra la notificación.
Postcondiciones: El sistema actualiza los datos de una de sus publicaciones

ID: 9
Nombre: Eliminar una o varias publicaciones
Descripción: Permite al usuario eliminar publicaciones
Actores: Usuario
Precondiciones: El usuario debe de haber seleccionado una o varias publicaciones del listado de sus publicaciones
Flujo normal: <ol style="list-style-type: none"> 1. El usuario se encuentra en la vista en detalle de una publicación previamente seleccionada. 2. El usuario pulsa “Delete” del menú superior. 3. El sistema muestra un mensaje de confirmación del borrado. 4. El usuario pulsa “Yes”. 5. El sistema elimina la publicación. 6. El sistema redirige al usuario a la vista anterior (el listado de todas o sus publicaciones).

<p>Flujo alternativo:</p> <p>2.1 El usuario se encuentra en el listado de todas sus publicaciones, pulsa “<i>Select</i>” y selecciona varias de estas.</p> <p>2.2 El sistema muestra un menú auxiliar superior con la opción “<i>Cancel</i>” y otro inferior con las opciones “<i>All</i>” y “<i>Done</i>”.</p> <p>2.3 El usuario pulsa “<i>Done</i>”.</p> <p>2.4 El usuario pulsa “<i>All</i>”.</p> <p>2.5 El usuario pulsa “<i>Cancel</i>”.</p> <p>3.3 El sistema elimina las publicaciones seleccionadas.</p> <p>3.4 El sistema elimina todas las publicaciones seleccionadas.</p> <p>3.5 El sistema elimina el mensaje de confirmación de la vista.</p>
<p>Postcondiciones: El sistema elimina las publicaciones que decida el usuario</p>

<p>ID: 10</p>
<p>Nombre: Seleccionar una o varias publicaciones</p>
<p>Descripción: Permite al usuario seleccionar una o varias publicaciones en función de lo que desea realizar a posteriori</p>
<p>Actores: Usuario</p>
<p>Precondiciones: El usuario debe encontrarse ante el listado de todas o sus publicaciones</p>
<p>Flujo normal:</p> <ol style="list-style-type: none"> 1. El usuario pulsa una publicación del listado. 2. El sistema redirige al usuario a la vista en detalle de la publicación.
<p>Flujo alternativo:</p> <p>2.1 El usuario se encuentra en la vista del listado de sus publicaciones y pulsa “<i>Select</i>”.</p> <p>2.2 El sistema muestra un menú auxiliar superior con la opción “<i>Cancel</i>” y uno inferior con las opciones “<i>All</i>” y “<i>Done</i>”.</p>
<p>Postcondiciones: El sistema permite al usuario seleccionar una o varias publicaciones</p>

ID: 11
Nombre: Buscar publicaciones por especialidad médica
Descripción: Permite a un usuario encontrar publicaciones en función de su especialidad
Actores: Usuario
Precondiciones: El usuario debe estar en la vista del listado de publicaciones
Flujo normal: <ol style="list-style-type: none"> 1. El usuario pulsa el botón de la lupa situada en el menú superior. 2. El sistema despliega un formulario de búsqueda que contiene una caja de texto y un selector. 3. El usuario selecciona una opción del selector de especialidades y pulsa el botón de buscar (icono de lupa). 4. El sistema realiza una búsqueda con los filtros aplicados y muestra los resultados en la vista.
Flujo alternativo: <ol style="list-style-type: none"> 3.1 El usuario pulsa “Cancel” en el menú superior. 3.2 El sistema pliega de nuevo el formulario de búsqueda. 4.1 El sistema muestra un mensaje de que no existen publicaciones con los filtros aplicados.
Postcondiciones: El sistema presenta al usuario un listado con las publicaciones de una determinada especialidad.

ID: 12
Nombre: Buscar publicaciones por título
Descripción: Permite a un usuario encontrar publicaciones en función de palabras contenidas en su título
Actores: Usuario
Precondiciones: El usuario debe estar en la vista del listado de publicaciones
Flujo normal: <ol style="list-style-type: none"> 1. El usuario pulsa el botón de la lupa situada en el menú superior. 2. El sistema despliega un formulario de búsqueda que contiene una caja de texto y un selector. 3. El usuario introduce un texto en el campo de texto y pulsa el botón de buscar (icono de lupa). 4. El sistema realiza una búsqueda con los filtros aplicados y muestra los resultados en la vista.

Flujo alternativo:

- 3.1 El usuario pulsa “Cancel” en el menú superior.
- 3.2 El sistema pliega de nuevo el formulario de búsqueda.
- 4.1 El sistema muestra un mensaje de que no existen publicaciones con los filtros aplicados.

Postcondiciones: El sistema presenta al usuario un listado con las publicaciones cuyo título coincida con las palabras introducidas.

ID: 13

Nombre: Buscar publicaciones por nombre de usuario

Descripción: Permite a un usuario encontrar publicaciones en función de su autor

Actores: Usuario

Precondiciones: El usuario debe estar en la vista del listado de publicaciones

Flujo normal:

1. El usuario pulsa el botón de la lupa situada en el menú superior
2. El sistema despliega un formulario de búsqueda que contiene una caja de texto y un selector.
3. El usuario introduce un texto en el campo de texto y pulsa el botón de buscar (icono de lupa).
4. El sistema realiza una búsqueda con los filtros aplicados y muestra los resultados en la vista.

Flujo alternativo:

- 3.1 El usuario pulsa “Cancel” en el menú superior.
- 3.2 El sistema pliega de nuevo el formulario de búsqueda.
- 4.1 El sistema muestra un mensaje de que no existen publicaciones con los filtros aplicados.

Postcondiciones: El sistema presenta al usuario un listado con las publicaciones cuyo nombre de usuario del autor coincida con las palabras introducidas.

ID: 14
Nombre: Ver comentarios
Descripción: Permite a un usuario visualizar los comentarios de una publicación
Actores: Usuario
Precondiciones: El usuario debe estar en la vista en detalle de una publicación
Flujo normal: <ol style="list-style-type: none"> 1. El usuario pulsa el botón asociado de los comentarios en el menú superior derecho. 2. El sistema realiza una lectura de los comentarios de la publicación y redirige al usuario a una nueva vista. 3. El usuario visualiza un listado con los datos de los comentarios de la publicación, sus fechas y autores.
Flujo alternativo: <ol style="list-style-type: none"> 3.1 El usuario visualiza un mensaje de que no se ha publicado ningún comentario por el momento.
Postcondiciones: El sistema presenta al usuario los comentarios realizados en una publicación.

ID: 15
Nombre: Crear comentarios
Descripción: Permite a un usuario comentar una publicación
Actores: Usuario
Precondiciones: El usuario debe estar en la vista de los comentarios de una publicación
Flujo normal: <ol style="list-style-type: none"> 1. El usuario pulsa el botón de más situado en el menú superior derecho. 2. El sistema redirige al usuario al formulario de publicación de comentarios. 3. El usuario completa el campo de mensaje y pulsa "Done". 4. El sistema muestra un mensaje de confirmación. 5. El usuario pulsa "Yes". 6. El sistema registra un nuevo comentario y redirecciona al usuario a la vista actualizada de todos los comentarios asociados de la publicación.
Flujo alternativo: <ol style="list-style-type: none"> 3.1 El usuario deja el campo del mensaje vacío y pulsa "Done". 3.2 El sistema notifica un mensaje de error. 5.1 El usuario pulsa "Cancel". 5.1 El sistema elimina de la vista el mensaje de confirmación.
Postcondiciones: El sistema registra un nuevo comentario en una publicación

ID: 16
Nombre: Editar comentario
Descripción: Permite a un usuario editar un comentario de una publicación
Actores: Usuario
Precondiciones: El usuario debe estar en la vista del listado de comentarios y ser el autor de alguno de ellos
Flujo normal: <ol style="list-style-type: none"> 1. El usuario desliza la pantalla hacia la izquierda 2. El sistema presenta al usuario un panel auxiliar en el lado derecho con las opciones <i>"Edit"</i> y <i>"Delete"</i> 3. El usuario pulsa <i>"Edit"</i> 4. El sistema redirecciona al usuario al formulario de edición de comentarios 5. El usuario modifica el mensaje del comentario y pulsa <i>"Save"</i> 6. El sistema muestra un mensaje de confirmación de la modificación 7. El usuario pulsa <i>"Ok"</i> 8. El sistema registra la modificación y redirecciona de nuevo al usuario a la vista de comentarios actualizada
Flujo alternativo: <ol style="list-style-type: none"> 5.1 El usuario no modifica el campo del mensaje y pulsa <i>"Save"</i> 6.1 El sistema muestra un mensaje de error al usuario 7.1 El usuario pulsa <i>"Cancel"</i> 8.1 El sistema deja de presentar la confirmación de la actualización del comentario
Postcondiciones: El sistema modifica el comentario en una publicación

ID: 17
Nombre: Eliminar comentario
Descripción: Permite a un usuario eliminar un comentario de una publicación
Actores: Usuario
Precondiciones: El usuario debe estar en la vista de los comentarios de una publicación y ser el autor de alguno de ellos
Flujo normal: <ol style="list-style-type: none"> 1. El usuario desliza la pantalla hacia la izquierda. 2. El sistema presenta al usuario un panel auxiliar en el lado derecho con las opciones <i>"Edit"</i> y <i>"Delete"</i>. 3. El usuario pulsa <i>"Delete"</i>.

<ol style="list-style-type: none"> 4. El sistema muestra un mensaje de confirmación del borrado 5. El usuario pulsa "Ok" 6. El sistema elimina el comentario y actualiza la pantalla
Flujo alternativo: <ol style="list-style-type: none"> 5.1 El usuario pulsa "Cancel" 6.1 El sistema deja de presentar la confirmación del borrado
Postcondiciones: El sistema elimina un comentario de una publicación

ID: 18
Nombre: Seleccionar comentario
Descripción: Permite a un usuario seleccionar un comentario de una publicación
Actores: Usuario
Precondiciones: El usuario debe estar en la vista de comentarios
Flujo normal: <ol style="list-style-type: none"> 1. El usuario desliza la pantalla hacia la izquierda 2. El sistema presenta al usuario un panel auxiliar en el lado derecho con las opciones "Edit" y "Delete"
Flujo alternativo: <ol style="list-style-type: none"> 1.1 El usuario no puede deslizar la pantalla hacia la izquierda dado que no es autor del comentario seleccionado
Postcondiciones: El sistema permite al usuario seleccionar un comentario para realizar a posteriori operaciones de edición o borrado de este

ID: 19
Nombre: Actualizar perfil
Descripción: Permite a un usuario modificar la información de su cuenta
Actores: Usuario
Precondiciones: El usuario debe estar logueado en el sistema
Flujo normal: <ol style="list-style-type: none"> 1. El usuario despliega el menú lateral izquierdo y pulsa "Settings" 2. El sistema redirecciona al usuario a la vista de configuración 3. El usuario pulsa "Edit profile"

<ol style="list-style-type: none"> 4. El sistema redirecciona al usuario a la vista de actualización de los datos de su cuenta 5. El usuario modifica los campos de nombre de usuario, su nombre completo y pulsa "Save" 6. El sistema muestra la confirmación de la solicitud de modificación 7. El usuario pulsa "Yes" 8. El sistema modifica los datos de la cuenta del usuario y muestra un mensaje de éxito 9. El usuario pulsa "Ok" 10. El sistema redirecciona al usuario de nuevo a la vista de configuración
<p>Flujo alternativo:</p> <ol style="list-style-type: none"> 5.1 El usuario no modifica ningún dato o no completa todos los campos requeridos y pulsa "Save" 6.1 El sistema muestra un mensaje de error 7.1 El usuario pulsa "Ok" 8.1 El sistema elimina el mensaje de error y mantiene al usuario en la vista
<p>Postcondiciones: El sistema modifica los datos de la cuenta del usuario</p>

<p>ID: 20</p>
<p>Nombre: Modificar el estado de las notificaciones</p>
<p>Descripción: Permite a un usuario activar o desactivar las notificaciones</p>
<p>Actores: Usuario</p>
<p>Precondiciones: El usuario debe estar en la vista de configuración</p>
<p>Flujo normal:</p> <ol style="list-style-type: none"> 1. El usuario pulsa "Notifications" 2. El sistema presenta al usuario un deslizador (switch) 3. El usuario desliza hacia la izquierda el switch para desactivar las notificaciones
<p>Flujo alternativo:</p> <ol style="list-style-type: none"> 3.1 El usuario desliza hacia la derecha el switch para activar las notificaciones
<p>Postcondiciones: El sistema permite al usuario configurar si desea o no recibir notificaciones</p>

ID: 21
Nombre: Añadir palabra clave
Descripción: Permite a un usuario registrar una palabra clave para notificar el registro de una publicación relacionada con esta
Actores: Usuario
Precondiciones: El usuario debe estar en la vista de configuración de notificaciones
Flujo normal: <ol style="list-style-type: none"> 1. El usuario pulsa el botón “más” situado en el menú superior derecho 2. El sistema presenta al usuario un formulario con un campo de texto 3. El usuario completa el campo de texto y pulsa “Done” 4. El sistema muestra la confirmación del registro de la palabra clave
Flujo alternativo: <ol style="list-style-type: none"> 3.1 El usuario deja el campo vacío y pulsa “Done” 3.2 El usuario pulsa “Cancel” 4.1 El sistema muestra un mensaje de error 4.2 El sistema elimina de la vista el formulario
Postcondiciones: El sistema permite al usuario introducir las palabras claves que contengan las publicaciones para la posterior notificación de estas

ID: 22
Nombre: Eliminar palabra clave
Descripción: Permite a un usuario eliminar una palabra clave registrada
Actores: Usuario
Precondiciones: El usuario debe estar en la vista de configuración de notificaciones
Flujo normal: <ol style="list-style-type: none"> 1. El usuario selecciona la palabra clave, desliza la pantalla hacia la izquierda y pulsa “Delete” 2. El sistema elimina la palabra clave y actualiza la vista
Flujo alternativo: -
Postcondiciones: El sistema permite al usuario eliminar las palabras claves que contengan las publicaciones para la posterior notificación de estas

ID: 23
Nombre: Recibir notificaciones
Descripción: Permite a un usuario recibir notificaciones de la publicación de posts que contengan determinadas palabras clave
Actores: Usuario
Precondiciones: El usuario debe estar registrado
Flujo normal: <ol style="list-style-type: none"> 1. El sistema realiza una lectura en segundo plano y observa una nueva publicación que contiene una palabra clave registrada por el usuario, si el usuario tiene las notificaciones activadas, entonces notifica su existencia 2. El usuario visualiza la notificación
Flujo alternativo: <ol style="list-style-type: none"> 1.1 El sistema realiza una lectura en segundo plano y observa una nueva publicación que contiene una palabra clave registrada por el usuario, si el usuario no tiene las notificaciones activadas, no se notifica su existencia 1.2 El sistema realiza una lectura en segundo plano y observa que no existen publicaciones que contengan palabras claves registrada por el usuario
Postcondiciones: El sistema notifica al usuario publicaciones relacionadas con algún tema de su interés

ID: 24
Nombre: Cambiar contraseña
Descripción: Permite a un usuario modificar la contraseña de su cuenta
Actores: Usuario
Precondiciones: El usuario debe estar en la vista de configuración
Flujo normal: <ol style="list-style-type: none"> 1. El usuario pulsa "<i>Change password</i>" 2. El sistema muestra un formulario con un campo de texto 3. El usuario introduce su nueva contraseña y pulsa "<i>Done</i>" 4. El sistema modifica la contraseña del usuario y elimina el formulario de la vista
Flujo alternativo: <ol style="list-style-type: none"> 3.1 El usuario deja el campo de texto vacío y pulsa "<i>Done</i>" 4.1 El sistema muestra un mensaje de error 4.2 El usuario pulsa "<i>Cancel</i>" 4.3 El sistema elimina el formulario de la vista
Postcondiciones: El sistema modifica la contraseña actual del usuario

ID: 25
Nombre: Recordar contraseña
Descripción: Permite a un usuario recuperar su contraseña
Actores: Usuario
Precondiciones: El usuario debe encontrarse en la vista de inicio de sesión
Flujo normal: <ol style="list-style-type: none"> 1. El usuario pulsa <i>"Forgot password?"</i> 2. El sistema muestra al usuario un formulario con un campo de texto para que este introduzca su email 3. El usuario introduce su email y pulsa <i>"Send"</i> 4. El sistema verifica que el email introducido está registrado y le envía un correo electrónico para que el usuario recupere su contraseña (gestionado por Firebase) 5. El sistema muestra un mensaje de éxito
Flujo alternativo: <ol style="list-style-type: none"> 3.1 El usuario pulsa <i>"Cancel"</i> 4.1 El sistema elimina el formulario de la vista de usuario 4.2 El sistema verifica que el email introducido no está registrado y muestra un mensaje de error
Postcondiciones: El sistema envía un email al usuario para introducir una nueva contraseña de inicio de sesión

ID: 26
Nombre: Darse de baja
Descripción: Permite a un usuario eliminar su cuenta del sistema
Actores: Doctor
Precondiciones: El usuario debe estar en la vista de configuración
Flujo normal: <ol style="list-style-type: none"> 6. El usuario despliega el menú lateral izquierdo y pulsa <i>"Settings"</i> 7. El sistema redirecciona al usuario a la vista de configuración 8. El usuario pulsa <i>"Delete account"</i> 9. El sistema muestra una confirmación del borrado de la cuenta 10. El usuario pulsa <i>"Ok"</i> 11. El sistema elimina la cuenta de usuario y muestra un mensaje de despedida 12. El usuario pulsa <i>"Ok"</i> 13. El sistema redirecciona al usuario al formulario de inicio de sesión

Flujo alternativo:

5.1 El usuario pulsa "Cancel"

6.1 El sistema elimina el mensaje de confirmación y mantiene al usuario en la vista actual

6.2 El sistema muestra un mensaje de error cuando se produce un error en el borrado de la cuenta del usuario

Postcondiciones: El sistema elimina la cuenta del usuario

Capítulo 5. Diseño de la aplicación

Justificación del diseño de las vistas

Entre los objetivos de este proyecto, se encuentra el de alcanzar la mayor simplicidad en los formularios para la gestión de las publicaciones, por ello, en el diseño de la aplicación destaca el minimalismo y priman las tonalidades pastel, blancos y negros. Por otro lado, las publicaciones se rigen por un mismo estilo estético, aunque las noticias cuentan además con una imagen de portada, y los artículos de investigación con un visor de documentos.

Las pantallas tienen un fondo de color blanco, los tipos de letra elegidos en la aplicación son “Copperplate Bold 20.0” en negrita para los títulos, para el autor y fecha de publicación “footnote” y “body” para los demás campos de texto. En el siguiente apartado veremos casos más concretos de cómo se visualizan las vistas con datos reales.

Pantallas

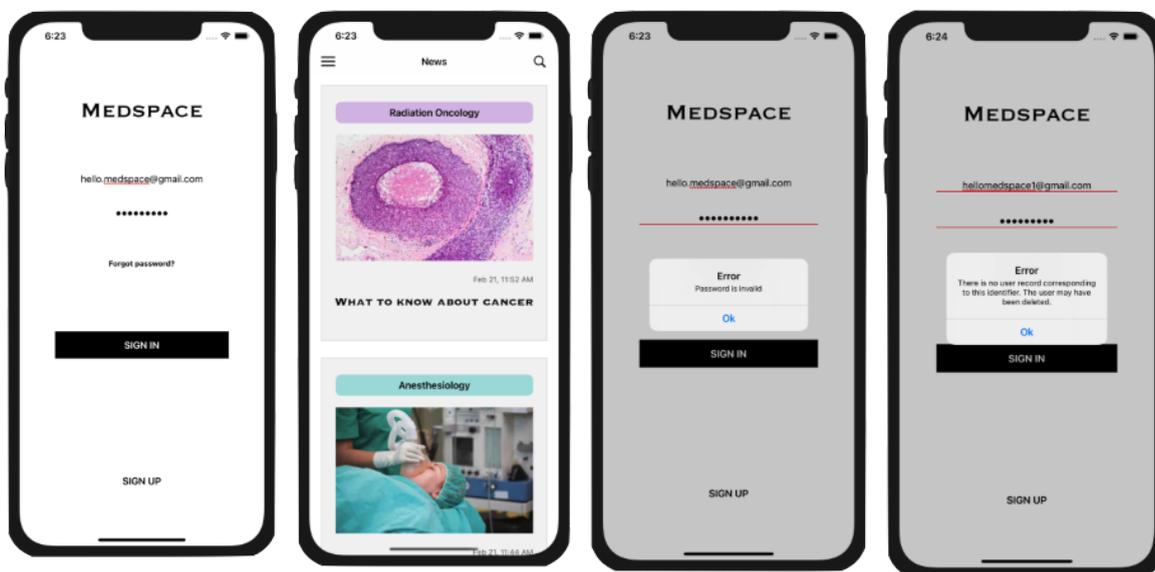


Figura 12. Diseño de la aplicación: Pantalla de inicio de sesión con o sin errores

Todos los formularios contienen el mismo tipo de campos de texto como el del inicio de sesión, es decir, no contienen ningún tipo de estilo definido: no poseen color de fondo o de letra, líneas de color. Cuando no existen errores en el logueo se continúa con la segunda vista, las noticias, pero cuando sí hay, se muestra una alerta en el dispositivo iOS con el mensaje del error que se ha producido, tampoco se han definido estilos específicos para ello.

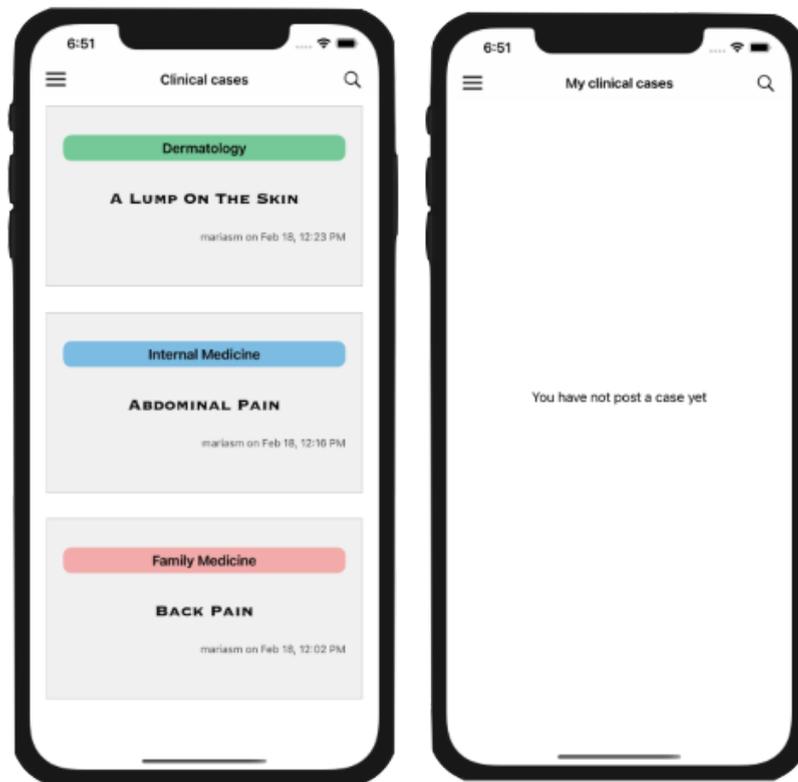


Figura 13. Diseño de la aplicación: Listado de publicaciones con y sin posts

Los listados de publicaciones tienen el mismo patrón de diseño, la especialidad médica, el título y en último lugar autor y fecha de publicación juntos. En el sistema, Las especialidad médicas de las publicaciones tienen asociada un color de fondo en la aplicación:

```
let specialities = [
    Speciality(name: "Allergy & Inmunology", color: UIColor.init(hexString: "#EC7063")),
    Speciality(name: "Anesthesiology", color: UIColor.init(hexString: "#A1DBDC")),
    ...
    Speciality(name: "Pediatrics", color: UIColor.init(hexString: "#f2b8c6")),
    Speciality(name: "Preventive Medicine", color: UIColor.init(hexString: "#99cc99")),
    ...
]
```

Las especialidades son los valores de la clave “name”, por ejemplo “Allergy and Immunology” o “Pediatrics”, el color del fondo que hemos comentado es el valor de la clave “color”, por ejemplo, para los casos anteriores “EC7063” y “f2b8c6” respectivamente. Los títulos como mencionamos, van en negrita y el autor y la fecha de la publicación en gris. Así mismo, las celdas de la tabla aportan un tono beige a la vista.

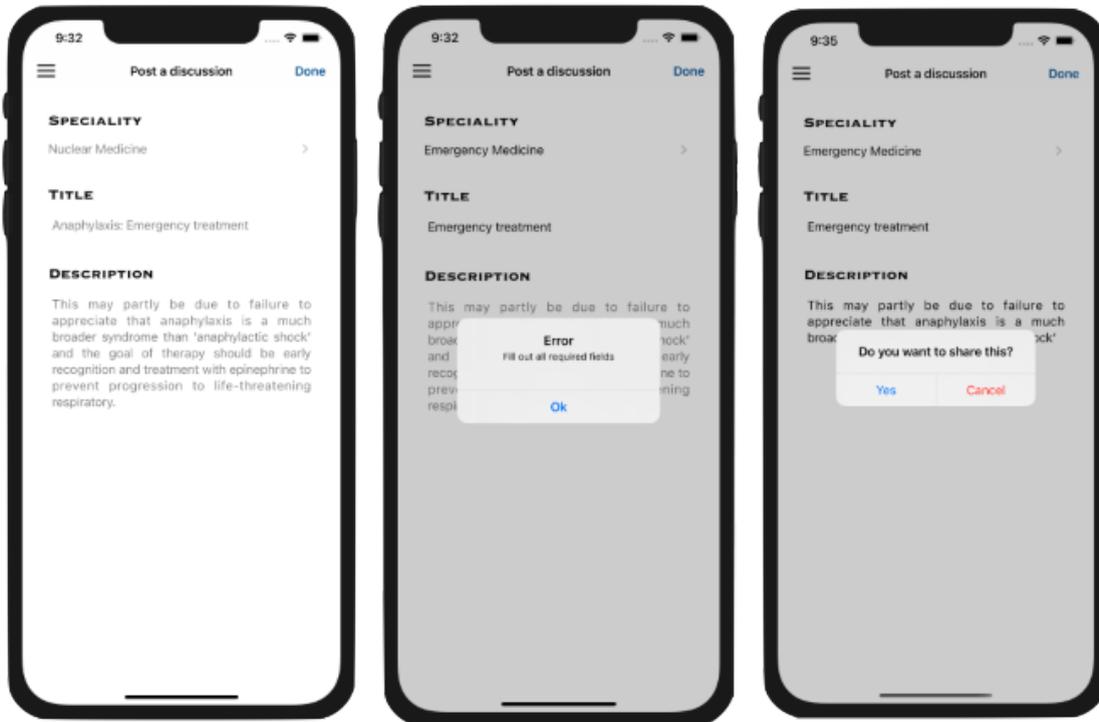


Figura 14. Diseño de la aplicación: Creación de una publicación

Para crear o editar una publicación la especialidad se selecciona con un selector que tiene un estilo propio predefinido por el entorno. Para crear una, los campos de texto presentan un texto por defecto que actúa como un “placeholder” en HTML, todos ellos modificables y obligatorios, cuando al menos uno de ellos no contiene texto y se envía el formulario, se presenta una alerta indicando que se deben completar todos los campos requeridos.

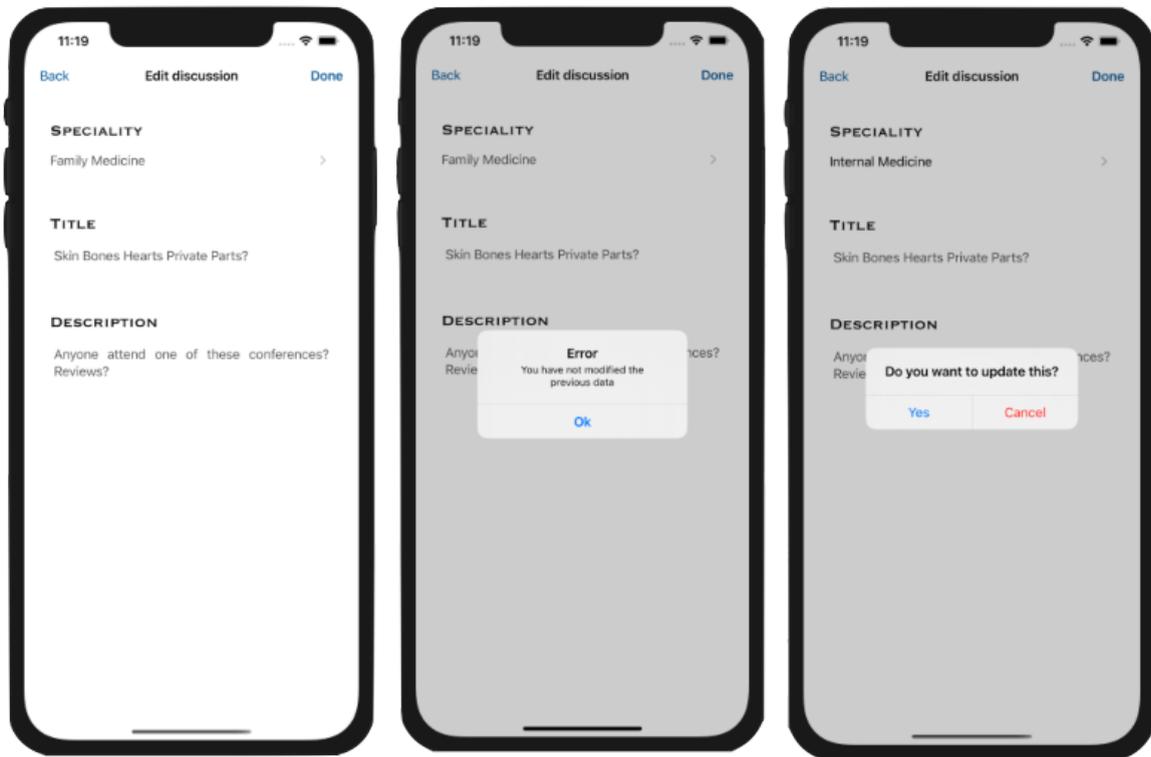


Figura 15. Diseño de la aplicación: Edición de una publicación

La única diferencia entre crear o editar una publicación es que los campos de texto en la edición como vemos en la Figura 15, se encuentran completados con la información del post, mientras que en la creación hay un texto por defecto. En cualquier caso los campos pueden ser modificados por el usuario. Cuando un campo de texto en la edición no se modifica se indica con una alerta de error, definida por defecto.

Tanto para los formularios de creación como de edición o borrado se confirma la operación, esto es un mensaje que contiene de qué se trata la acción a realizar y dos opciones, seguir con el procedimiento o cancelar, para ello se utiliza un estilo predeterminado por el dispositivo iOS para las alertas de confirmación.

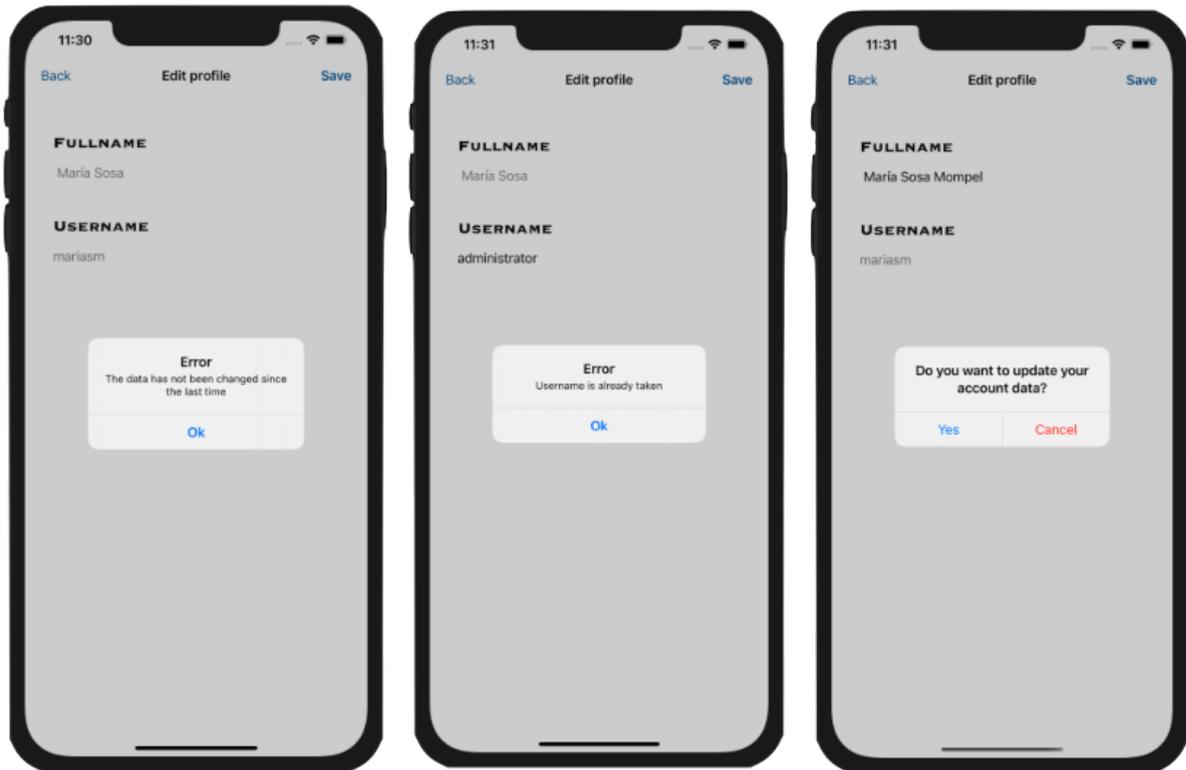


Figura 16. Diseño de la aplicación: Actualizar la cuenta de un usuario

Para actualizar la información de la cuenta tenemos dos campos de texto modificables con el contenido de la información de la cuenta del usuario, los títulos de estos se encuentran en negritas, en el mismo formato que para los títulos de las publicaciones. Cuando no modificamos ningún campo, completamos los campos con la misma información que tenía ya el usuario, o queremos un nombre de usuario que ya está siendo usado en la aplicación, salta una alerta de error, como las que hemos visto anteriormente.

Como vemos en estos casos concretos y en líneas generales, el estilo empleado en el sistema es homogéneo, los formularios y listados siguen el mismo patrón (a excepción de las noticias que incluyen una imagen), los tonos, los tipos de letra, la indicación de que existe un error en el sistema. Predomina un estilo minimalista en las vistas, que lo hace intuitivo de primeras y una app fácil de usar.

Capítulo 6. Herramientas

Entorno XCode

Entorno de desarrollo diseñado por Apple Inc., está integrado en todos los sistemas operativos MacOS y nos permite publicar aplicaciones en la App Store. Para esto, se debe disponer de una cuenta como desarrollador, registrándose en [Apple Developer Program](#) y disponer de un mac para compilar un ejecutable del software que hemos implementado. En el desarrollo de este proyecto se ha empleado la versión 12.1 (12A7403) y Swift, como lenguaje de programación.

Swift salió a la luz en 2014 en la Conferencia Anual de Desarrolladores que ofrece Apple. Entre sus principales características, destaca por garantizar un desarrollo más rápido del código, puesto que dispone de unos gráficos en los que objetos, clases, librerías se identifican de forma sencilla y los errores son detectables antes de poner en ejecución del código, así mismo incluye gran cantidad de simuladores, iPhone, iPad, etc.

Simulador

Los simuladores nacen con el fin de poder ejecutar la aplicación sin necesidad de disponer de un dispositivo iOS, o para comprobar que su diseño y funcionalidad es correcta también en otros modelos.

Esta aplicación está preparada para diferentes dispositivos iPhone o iPad, sin embargo, las capturas de las capturas que aparecen en esta memoria se han realizado ejecutando el programa en el simulador de XCode en su versión 12.1 (940.16), iPhone 12 Pro Max (14.1).

Firestore

Plataforma en la nube desarrollada por Google para desarrollar aplicaciones, tanto móviles, como web. Firestore ofrece gran cantidad de servicios para el desarrollo de aplicaciones, las veremos a continuación.

- **Firestore** permite almacenar los datos de la aplicación en la nube de manera sencilla y rápida.
- **Firebase Auth** permite autenticar a los usuarios mediante un código generado automáticamente, un correo y una contraseña encriptada.
- **Firebase Analytics** proporciona una visión del uso de la aplicación, cómo se comportan los usuarios en ella y su grado de participación, o recoger datos estadísticos, un muestreo con el tipo de dispositivo que emplean más los usuarios, su edad, género, etc.
- **Firebase Cloud Messaging** es una plataforma que permite el envío de notificaciones y mensajes en tiempo real, a través de las plataformas Android, IOS y aplicaciones web.
- **Realtime Database** proporciona una base de datos a tiempo real que almacena la información de nuestros usuarios, publicaciones y comentarios en formato JSON.
- **Firebase Storage** proporciona un espacio para cargar y descargar ficheros, por ejemplo, en nuestro caso almacenamos las imágenes de las noticias y los documentos de los artículos de investigación.

Git

Esta herramienta se utiliza para el control de versiones del proyecto. Durante el desarrollo del mismo, esta plataforma nos ayuda a guardar en la nube y disponer de forma relativamente sencilla, rápida y eficazmente el estado del código del proyecto en cada incremento, de esa manera podemos acceder al historial del desarrollo de la aplicación.

En GitHub un proyecto se almacena en un repositorio o directorio en el que se encuentran todos los archivos de código, imágenes y todo lo que necesite nuestro proyecto. A lo largo del desarrollo de *Medspace*, la aplicación se ha mantenido en un repositorio privado, para salvaguardar la confidencia de esta. Puede consultarse el repositorio en el siguiente enlace <https://github.com/QueraltSM/Medspace>.

Capítulo 7. Desarrollo de la aplicación

En este apartado se va a mostrar cómo se ha implementado la aplicación desde cero. Así mismo, podemos visitar el código fuente al completo su correspondiente repositorio en GitHub: <https://github.com/QueraltSM/Medspace>

Primeros pasos en XCode

Para comenzar a desarrollar una app en XCode, tendremos que tener instalado este entorno de programación, en los equipos MacOS normalmente viene instalado por defecto. Una vez lanzado el entorno tendremos que visualizar un listado de las apps que estamos desarrollando o en su defecto una lista vacía. En mi caso como vemos, se encuentra *Medspace* y algunos otros proyectos en los que estoy trabajando:



Figura 17. Desarrollo de la aplicación: Primeros pasos en Xcode I

Para crear la app solo tenemos que hacer click en “Create a new Xcode project” que es lo que tuve que hacer para empezar a desarrollar. En la implementación de aplicaciones móviles, tenemos por un lado la vista principal en la que se encuentran las pantallas denominada “Main.storyboard” (pantalla derecha de la figura 13) y los ficheros que contienen la lógica detrás de estas (pantalla izquierda de la figura 13):

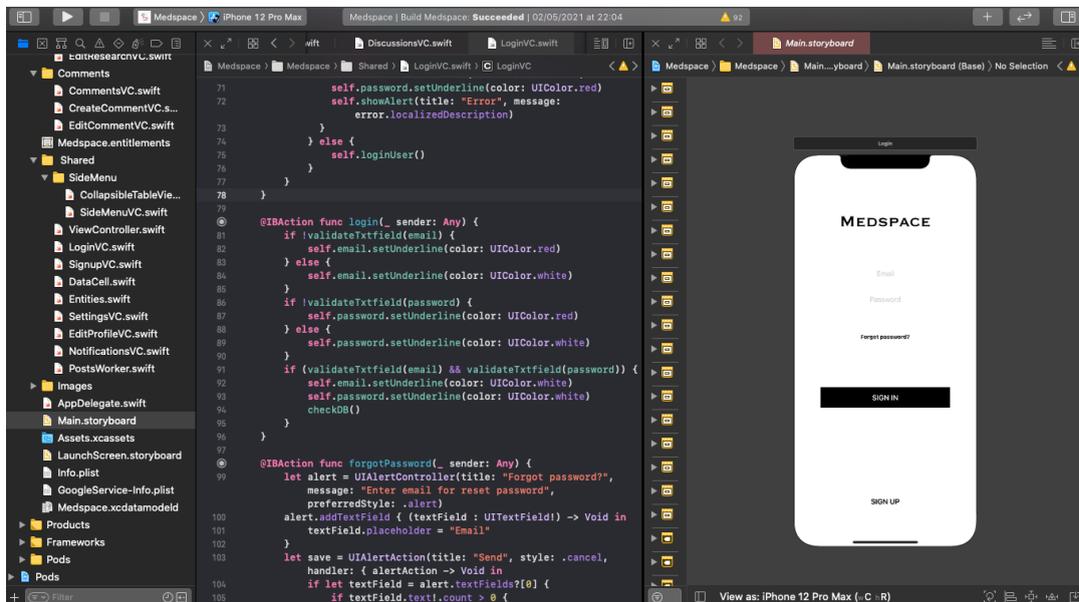


Figura 18. Desarrollo de la aplicación: Primeros pasos en Xcode II

Las pantallas o vistas es el medio por el que el usuario interactúa con la app. Para crear una tenemos que registrar un nuevo fichero denominado “Cocoa Touch Class” de tipo “ViewController”. Una vez hecho esto, tendremos en “Main.storyboard” una vista más y un nuevo fichero de extensión .swift, donde se va a escribir las líneas de código funcional pertinentes.

Diagrama de tablas

La aplicación está formada por usuarios que publican noticias, casos clínicos, debates y artículos de investigación que contienen comentarios. Así mismo los usuarios deciden sobre qué tipos de publicaciones quieren obtener notificaciones. Los datos se almacenan y gestionan en *Google Firebase*, veamos entonces un diagrama con las tablas que participan:

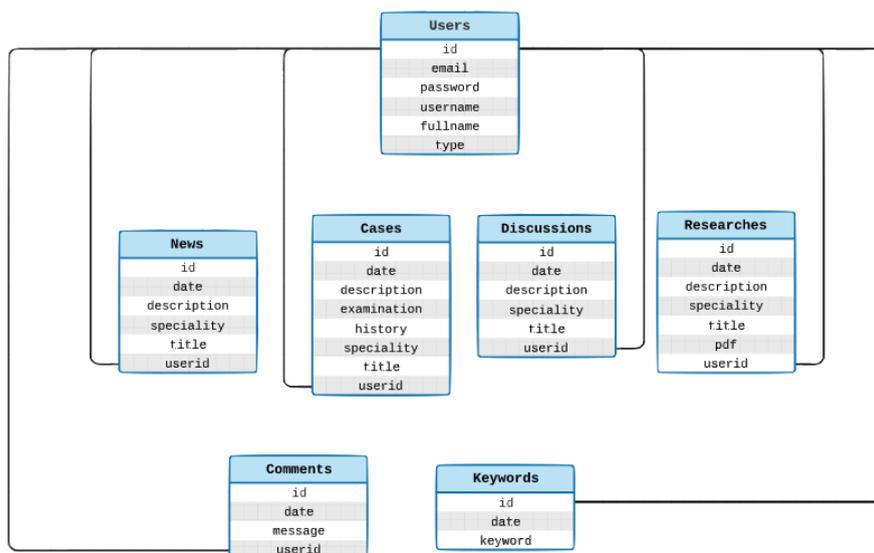


Figura 19. Desarrollo de la aplicación: Diagrama de tablas

Los usuarios se identifican con un id que Firebase genera automáticamente, un email, contraseña, nombre completo, nombre de usuario único en la aplicación, y tipo de usuario del que se trata, doctor o administrador.

Las publicaciones se organizan como árboles, tienen un padre cuyo id coincide con el id del usuario que publica el post y los hijos se identifican por la fecha de su publicación. Este mecanismo se debe a que Firebase no proporciona ninguna forma de generar ids auto genéricos si no se trata de cuentas de usuario. Si identificamos un post por la fecha únicamente, puede darse el caso de que dos usuarios publiquen a la vez, esto provoca un error pues sólo se registraría uno de los dos posts. Si identificamos un post por el id del usuario seguido de la fecha de su publicación obtendremos más filas, por ejemplo:

[xWNH4ZN8m2X7iwlZPScgJbakUj2-2021-02-18 12:02:27 +0000](#)

[xWNH4ZN8m2X7iwlZPScgJbakUj2- 2021-02-18 12:16:03 +0000](#)

En cambio un usuario solo puede publicar un post a la vez, resulta más limpio de esta manera y más arbóreo como busca conseguir Firebase:

```
"xWNH4ZN8m2X7iwlZPScgxJbakUj2" : {  
  "2021-02-18 12:02:27 +0000" : {  
    "date" : "2021-02-18 12:02:27 +0000",  
    "description" : "A 27-years-old woman is admitted to the emergency department  
complaining of pain across her back.",  
    "",  
    "speciality" : "Family Medicine",  
    "title" : "Back Pain",  
    "user" : "xWNH4ZN8m2X7iwlZPScgxJbakUj2"  
  },  
  "2021-02-18 12:16:03 +0000" : {  
    "date" : "2021-02-18 12:16:03 +0000",  
    "description" : "A 25-year-old woman presents to her general practitioner (GP)  
complaining of pain in her right loin radiating in to the right iliac fossa. ",  
    "",  
    "speciality" : "Internal Medicine",  
    "title" : "Abdominal Pain",  
    "user" : "xWNH4ZN8m2X7iwlZPScgxJbakUj2"  
  }  
}
```

Los cuatro tipos de publicaciones, noticias, casos clínicos, debates y artículos de investigación tienen en común varios campos, el título, la descripción, especialidad médica, fecha y autor del post. Las noticias contienen imágenes y los artículos de investigación un documento digital, donde se encuentra el estudio llevado a cabo por el usuario. Los casos clínicos se diferencian del resto en que contienen un campo denominado “*examination*” e “*history*”, el primero se refiere a las pruebas relevantes realizadas por el doctor, el segundo al historial clínico del paciente.

Los comentarios se identifican por un identificador, el id del usuario del post seguido de la fecha de la publicación del post, el campo mensaje que el usuario haya postado y su fecha. Por último las palabras clave también funcionan como árboles, el id padre es el del usuario que las registra y sus hijos se identifican con la fecha del registro y contendrán la palabra.

Preparar la base de datos

Firebase es una plataforma que facilita numerosos tutoriales para su configuración, para realizar la conexión de la base de datos con nuestro proyecto, y algunas operaciones lógicas, [Agrega Firebase a tu proyecto de iOS](#) es una buena referencia para empezar en Firebase si la aplicación está destinada para dispositivos iOS. He seguido este para crear un nuevo proyecto en este entorno y conectarlo a mi app creada en el paso anterior, se deben seguir sus instrucciones, la descarga del fichero *GoogleService-Info.plist* y la importación de la librería *Firebase* o *FirebaseCore* entre otros, en función de lo que necesitemos implementar.

Registro y autenticación de usuarios

Para que un usuario pueda registrarse en la app primero comprobaremos que no esté ya registrado, haremos uso de la función que nos proporciona Firebase Auth denominada *createUser* como vemos en el [tutorial oficial de Firebase, apartado registra usuarios nuevos](#). Si se trata de un nuevo usuario, dicha función creará un nuevo registro en la pestaña *Authentication* de Firebase como vemos a continuación:



Figura 20. Desarrollo de la aplicación: Registro y autenticación de usuarios

En este caso, además, tendremos que almacenar los datos del usuario en la tabla *Users* haciendo uso de la función *setValue* como indica el tutorial [lee y escribe datos en iOS](#) al que indicaremos por parámetros todo lo que queremos incorporar a la tabla, en nuestro caso, el email, contraseña, nombre completo, nombre y tipo de usuario.

Cuando un usuario se encuentra registrado tendrá que iniciar sesión, se comprobará que tanto el email como la contraseña sean válidos y haremos uso de la función que nos proporciona Firebase Auth denominada *signIn* como vemos en el [tutorial oficial de Firebase, apartado acceso de usuarios existentes](#), si resulta un logueo exitoso, la aplicación lanza la página de inicio donde se encuentran las noticias publicadas.

Listado de una publicación

Para visualizar el listado de publicaciones debemos de crear una vista que contenga un objeto de tipo *UITableView*, el cual representa la tabla que contendrán los datos. Por ejemplo, vamos a obtener los casos clínicos publicados:

```
1. let ref = Database.database().reference()
2. ref.child("Cases").observeSingleEvent(of: .value, with: { snapshot in
3.   if (snapshot.children.allObjects.count == 0) {
4.     self.cases_timeline.setEmptyView(title: "No case has been posted yet")
5.   } else {
6.     self.cases_timeline.restore()
7.     self.loopCases(ref: ref, snapshot: snapshot)
8.   }
9. })
10. ...
11. for child in snapshot.children.allObjects as! [DataSnapshot] {
12.   let dict = child.value as? [String : AnyObject] ?? [:]
13.   for childDict in dict {
14.     let data = childDict.value as? [String : AnyObject] ?? [:]
15.     let title = data["title"]! as! String
16.     let description = data["description"]! as! String
17.     ...
18.   }
19.   ...
20.   self.cases.append(Case(id: childDict.key, title: title, description: description, history:
    history, examination: examination, date: date, speciality: Speciality(name: speciality,
    color: color), user: User(id: userid, fullname: fullname, username: username)))
21.   let sortedCases = self.cases.sorted {
22.     $0.date > $1.date
23.   }
24.   self.cases=sortedCases
```

Primero iteramos en un bucle cada registro que nos devuelve la función *observeSingleEvent* (línea 2). Cuando no encontramos ninguno incorporamos a la vista un mensaje de que no existen publicaciones (línea 4), si no, iteramos cada uno de los hijos (línea 11) y obtenemos los valores de sus campos (líneas 14-16, etc), por último almacenamos los valores obtenidos en una lista (línea 20) y los ordenados por el más reciente primero (línea 21-24). Cuando la publicación se trata de una noticia o un artículo tenemos que consultar a Firebase Storage para obtener la imagen y documento digital respectivamente.

1. `let dataChild = childDict.value as? [String : AnyObject] ?? [:]`
2. `let storageRef = Storage.storage().reference().child("News/\(child.key)/\(childDict.key)")`
3. `storageRef.getData(maxSize: 1 * 1024 * 1024) { (data, error) -> Void in`
4. `let title = dataChild["title"]! as! String`
5. ...
6. `let pic = UIImage(data: data!)`
7. ...

Una imagen la obtenemos creando la referencia en la que se encuentra (línea 2), la obtenemos con la función *getData* (línea 3) y la asignamos a la imagen de tipo *UIImage* que tenemos declarada en la vista (línea 6).

1. `let storageRef =`
2. `Storage.storage().reference().child("Researches/\(child.key)/\(childDict.key)")`
3. `storageRef.downloadURL { (url, error) in`
4. `if error == nil {`
5. `self.researches.append(Research(id: childDict.key, pdf: url!, date: date, title: title, speciality: Speciality(name: speciality, color: color), description: description, user: User(id: userid, fullname: fullname, username: username)))`
6. ...

Una artículo lo obtenemos mediante la declaración de la ruta de su registro (líneas 1-2), la descarga de su URL (línea 3) y la posterior asignación a un nuevo objeto que carga como campo *pdf* el fichero que se encuentra en la nube (línea 5).

Creación de una publicación

Para registrar una publicación creamos una vista que contenga un formulario con los campos de texto y botones necesarios. Cuando todos los campos se completan la función que inserta los datos en la base de datos será *setValue*, cuyos parámetros son los datos introducidos por el usuario. Por ejemplo, para registrar un nuevo debate hacemos:

1. `let ref = Database.database().reference()`
2. `ref.child("\path/title").setValue(title)`
3. `ref.child("\path/description").setValue(description)`
4. `ref.child("\path/speciality").setValue(speciality)`
5. `ref.child("\path/user").setValue(user)`
6. `ref.child("\path/date").setValue(date)`

Utilizamos como referencia la base de datos que hemos inicializado en Firebase (línea 1), declarada en el fichero *GoogleService-Info.plist*, y al tratarse en este caso, de un registro de un debate, las claves serán “*title*”, “*description*”, “*speciality*”, “*user*” y “*date*”, los valores los campos introducidos por el usuario (líneas 2-6).

Cuando el administrador publica una noticia tendremos que crear un objeto de tipo *StorageMetadata* para indicar que vamos a registrar una imagen de la siguiente manera:

1. `let metaDataConfig = StorageMetadata()`
2. `metaDataConfig.contentType = "image/jpg"`
3. `let storageRef = Storage.storage().reference(withPath: path)`
4. `storageRef.putData(imageData, metadata: metaDataConfig) { (metaData, error) in`
5. `...`
6. `}`

Cuando el doctor publica un artículo de investigación tendremos que subir el documento a Firebase Storage con la URL del fichero que hemos cargado:

1. `Storage.storage().reference().child(path).putFile(from: self.documentURL!, metadata:`
2. `nil) { metaData, error in`
3. `...`
4. `}`

Edición de una publicación

Para editar una publicación creamos una vista que contenga un formulario con los correspondientes campos que necesitemos para mostrar la información en detalle del post, serán campos modificables. Cuando se modifica alguno de los campos la función que actualiza los datos en la base de datos será de nuevo *setValue*, cuyos parámetros son los datos que han sido modificados y los que no, de tal manera que se actualizan aquellos campos en la tabla que sí han sido modificados, pasamos por parámetro los no modificados también, porque si no tendría que comprobarse con anterioridad si han sido actualizados, nos ahorramos un tiempo de comprobación. Por ejemplo, para editar un caso clínico hacemos:

1. `let path = "Cases/^(uid!)/^(self.clinical_case!.id)"`
2. `self.postCase(path: path, title: self.case_title.text!, description: self.case_description.text!,`
3. `history: self.case_history.text!, examination: self.case_examination.text!,`
4. `speciality:self.speciality_textfield.text!, user: uid!, date: self.clinical_case!.date)`

En este caso, en la línea 1 indicamos donde tienen que ser almacenados los datos actualizados, al tratarse de un caso clínico será en la tabla “Cases”, en el registro que tenga como identificador el id (o uid) del usuario logueado, es decir, el autor del post, y el identificador del caso, que como hemos comentado, es la fecha de la publicación. Por último, llamamos a la función *postCase* el cual se encuentra implementado en el fichero *Shared/ViewController.swift*, su lógica se basa en la función *setValue*.

Cuando el administrador actualiza la imagen de una noticia o el doctor el documento digital de uno de sus artículos de investigación sigue el mismo procedimiento que para la subida de imágenes y documentos mencionados en el anterior apartado.

Borrado de una publicación

Para eliminar una publicación se pueden dar dos circunstancias, que el usuario se encuentre en la lista de sus publicaciones, o que se encuentre en la vista en detalle de una publicación.

Cuando el usuario se encuentra en la lista de sus publicaciones, se encuentra con un botón en el menú superior para poder seleccionar posts, una vez hecho esto le sale un menú auxiliar, para poder borrarlos, cuando hace click se ejecuta un código como el siguiente, por ejemplo para eliminar un artículo de investigación:

1. if let selectedRows = researches_timeline.indexPathsForSelectedRows {
2. var items = [Research]()
3. for indexPath in selectedRows {
4. items.append(researches[indexPath.row])
5. }
6. for _ in items {
7. let index = researches.firstIndex(where: { (item) -> Bool in
8. item.id == item.id
9. })
10. let path = "Researches/(uid!)/(researches[index!.id])"
11. removeDataDB(path: path)
12. removeDataStorage(path: path)
13. removeDataDB(path: "Comments/(path)")
14. researches.remove(at: index!)
15. }
16. researches_timeline.beginUpdates()
17. researches_timeline.deleteRows(at: selectedRows, with: .automatic)
18. researches_timeline.endUpdates()
19. }

En la línea 1 comprobamos si se han seleccionado publicaciones, en ese caso, en la línea 7 obtenemos tras iterar cada una, las que deben de ser eliminadas del listado, para finalmente en las líneas 10-13 eliminarlas de la base de datos, estas funciones se explican con más detalle en el siguiente caso. Luego, tenemos que actualizar la tabla eliminando los registros que hemos borrado (línea 14) y refrescar la tabla dinámicamente, líneas 16-18.

Cuando el usuario se encuentra en la vista en detalle de una publicación y es el autor de la misma, en el menú superior puede seleccionar el botón de borrado, en ese caso, se ejecuta un código similar:

1. let path = "Researches/(uid!)/(self.research!.id)"
2. self.removeDataDB(path: path)
3. self.removeDataStorage(path: path)
4. self.removeDataDB(path: "Comments/(path)")

La ruta a seguir se encuentra en la tabla de los artículos de investigación “*Researches*”, más concretamente en los registros del autor (uid, usuario actual), cuyo identificador sea el del artículo seleccionado y visualizado en la vista. Tenemos que eliminar tanto el registro (línea 2), como el documento digital (línea 3), como los comentarios asociados al artículo (línea 4).

Notificación de publicaciones

Para que un usuario reciba notificaciones primero creamos una vista que contendrá un *switch* con el que activar o desactivar notificaciones y una tabla con un botón para añadir palabras claves.

Cuando pulsamos el botón “+” situado en el menú superior, creamos un formulario con un campo de texto que debe de ser completado por el usuario.

1. `if keywords.contains(where: {$0.keyword.compare(keyword, options: .caseInsensitive) == .orderedSame}){`
2. `self.showAlert(title: "Error", message: "Keyword already exists")`
3. `} else {`
4. `let now = Date().description`
5. `keywords.append(Keyword(id: now, keyword: keyword))`
6. `let path = "Keywords/\\(uid!)"`
7. `let ref = Database.database().reference()`
8. `ref.child("\\(path)/\\(now)").setValue(keyword)`
9. `allKeywords.reloadData()`
10. `}`

Para añadir una palabra clave tendremos que comprobar que esta no exista previamente (línea 1), en ese caso, la añadimos al listado de palabras clave del usuario (línea 5) y lo almacenamos en la base de datos en la tabla “*Keywords*”, en el registro destinado para el usuario actualmente logueado (uid) y la de la fecha de registro, es decir, el identificador de la palabra clave introducida en la tabla de palabras clave del usuario (líneas 6-8). Por último actualizamos la tabla de la vista (línea 9). Para notificar una publicación tenemos que conocer qué publicaciones nuevas le pueden interesar al usuario y si desea que se le notifiquen.

1. `let isUserLoggedIn = UserDefaults.standard.bool(forKey: "isUserLoggedIn")`
2. `let sendNotifications = UserDefaults.standard.object(forKey: "notificationsState") as? Bool`
3. `if (isUserLoggedIn && sendNotifications != nil && sendNotifications!) {`
4. `PostsWorker().start()`
5. `}`

La variable de sesión que nos indica si el usuario tiene activada las notificaciones es “*notificationsState*”, si está activada y el usuario está logueado llamamos a la clase *Shared/PostsWorker*, encargada de comprobar qué publicaciones le interesa.

En la clase *PostWorker* comprobamos si hay una nueva publicación que no hayamos notificado al usuario, para eso almacenamos en variables de sesión cuantas publicaciones hemos notificado y cuantas se han posteado, cuando hay más posteadas que notificadas, comprobamos si las que se han publicado, en su título o descripción contienen alguna de las palabras clave de la lista de las palabras clave del usuario, en ese caso con ayuda de *UNUserNotificationCenter.current()* en *AppDelegate.swift*, notificamos el post e incrementamos las variables de sesión contadoras que han sido mencionadas.

Gestión de la cuenta de usuario

Para que un usuario modifique sus datos, creamos una interfaz con dos campos de texto modificables, que contendrá su nombre completo y nombre de usuario, cuyos valores se obtienen de consultar la tabla “*Users*” de la base de datos.

1. `let uid = Auth.auth().currentUser!.uid`
2. `self.ref.child("Users/(uid)/username").setValue(self.username.txt.text)`
3. `self.ref.child("Users/(uid)/fullname").setValue(self.fullname.txt.text)`
4. ...
5. `self.setUserData(fullname: self.fullname.txt.text!, usertype: usertype, username: self.username.txt.text!, isUserLoggedIn: true)`
6. `}}`

Para almacenar los valores nuevos que introduzca el usuario en la base de datos usamos *setValue*, en el registro cuyo id coincida con el identificador del usuario logueado (líneas 1-3). Por último, debemos actualizar los valores del menú lateral, donde se encuentra el nombre del usuario, para ello la función *setUserData* modifica las variables de sesión de la siguiente manera:

1. `UserDefaults.standard.set(usertype, forKey: "usertype")`
2. `UserDefaults.standard.set(fullname, forKey: "fullname")`
3. ...

Para cambiar la contraseña tenemos que usar *updatePassword*, una función propia de Firebase que podemos consultar en el [tutorial de Firebase, apartado configura la contraseña de un usuario](#), para dar de baja empleamos la función que nos proporciona Firebase Auth en el [tutorial de Firebase, apartado borra a un usuario](#), y por último eliminar toda la información de las publicaciones del usuario, palabras claves registradas, comentarios. Para borrar cualquier registro usamos *removeValue*, pasándole por parámetro la dirección de este.

Ejecución de la aplicación en un simulador

Para ejecutar una aplicación en XCode tenemos que seleccionar un simulador y ejecutar “Run”. De esta manera se compila y ejecuta nuestra app:

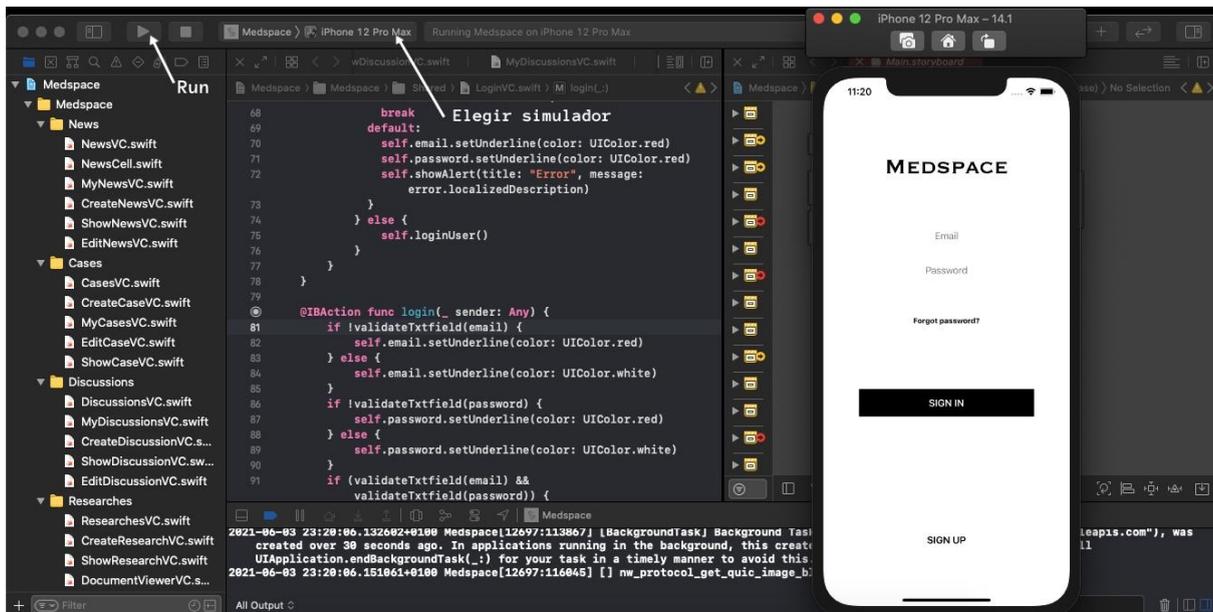


Figura 21. Desarrollo de la aplicación: Ejecución de la aplicación en un simulador

Capítulo 8. Conclusiones

Valoración personal

Desde que comencé la carrera tenía claro que mi Trabajo Fin de Grado estaría relacionado con el mundo de la Medicina, ciencia que me apasiona. Durante mis estudios en el grado, me introduje en la programación de aplicaciones nativas, Android e iOS, y pude completar mi formación en esto cuando realicé mis prácticas externas en Disoft Servicios Informáticos, donde tuve que implementar varias aplicaciones híbridas con Ionic y React.

Este proyecto me ha proporcionado un mayor conocimiento en el empleo de librerías y objetos que nunca antes había utilizado y he profundizado en la implementación de los diseños adaptativos, para distintas resoluciones y tamaños de pantalla. Por otro lado, decidí crear este espacio médico para dispositivos iOS en lugar de otras plataformas porque *Swift* es el primer lenguaje de programación de aplicaciones que aprendí, por lo que significa mucho para mí terminar la carrera de esta manera, y seguir aportando en este mundo en el que quiero seguir trabajando.

Futuras ampliaciones o mejoras

Este proyecto podría personalizar el contenido que desea el usuario, esto es, cuando un doctor registra una cuenta, seleccionar qué tipo de publicaciones quiere que se le notifique, por ejemplo, en función de sus preferencias en cuanto a especialidades médicas.

Por otro lado, sería interesante introducir reacciones al contenido y seleccionarlo como favorito, por lo que, la aplicación contaría con una sección de “*Favoritos*”, de esta manera el listado de publicaciones podrían ordenarse en función de sus “*Me gusta*”. La sección de comentarios mejoraría si adjuntamos fotos o documentos, para poder responder de forma más completa.

Por último, añadir una nueva tabla en la base de datos denominada *Specialities*, la cual contendrá un listado de las especialidades médicas, de esta manera no tendremos que modificar el código, compilarlo, ejecutarlo y publicar de nuevo nuestra aplicación para ver reflejados los cambios, simplemente añadiremos un registro más en la tabla.

Medspace es un proyecto que en sus inicios está ideado para médicos, lo suyo sería expandir su alcance y ofrecer un espacio para todo tipo de sanitarios, es decir, enfermeros, farmacéuticos, odontólogos, etc.

Capítulo 9. Bibliografía

<https://es.digitaltrends.com/celular/sistemas-ios-y-android/>

Android vs. iOS: comparamos a fondo ambos sistemas operativos

<https://marketing4ecommerce.net/el-coronavirus-ha-disparado-el-tiempo-de-uso-de-las-apps-en-todo-el-mundo/>

El coronavirus ha disparado el tiempo de uso de las apps en todo el mundo

<https://www.trecebits.com/2021/01/13/la-descarga-de-apps-moviles-crecio-un-7-en-2020/>

Las apps crecieron un 7%

<https://campussanofi.es/e-professionals/noticias/las-mejores-apps-para-medicos-son-gratuitas/>

Campus Sanofi, Mejores aplicaciones móviles para profesionales sanitarios

<https://espanol.medscape.com/public/acerca-de-medscape>

Información sobre la aplicación Medscape

<https://apps.apple.com/us/app/daily-rounds-for-doctors/id1003799400?l=es>

Información sobre la aplicación Daily Rounds - for Doctors

<http://josepalomar.com/wp-content/uploads/UML.pdf>

Tipos de relaciones de un diagrama de clases

<https://www.tokioschool.com/noticias/que-es-swift-lenguajes-de-programacion/>

Qué es Swift. Lenguajes de programación

<https://www.medicalnewstoday.com/articles/324694>

Datos de la noticia “*Alzheimer's disease: An eye test could provide early warning*”

<https://www.medicalnewstoday.com/articles/322375>

Datos de la noticia “*You may be partly conscious under general anesthesia*”

<https://www.medicalnewstoday.com/articles/323648#causes>

Datos de la noticia “*What to know about cancer*”

https://www.moscomm.org/pdf/100_Cases_in_Clinical_Medicine.pdf, Página 37

Datos del caso “*Back pain*”

https://www.moscomm.org/pdf/100_Cases_in_Clinical_Medicine.pdf, Página 121

Datos del caso “*Abdominal pain*”

https://www.moscomm.org/pdf/100_Cases_in_Clinical_Medicine.pdf, Página 189

Datos del caso “*A lump on the skin*”

<https://www.physicianassistantforum.com/topic/58774-is-it-ever-ok-to-be-paternalistic-play-god/>

Datos del debate “*Is it ever ok to be paternalistic (play God?)*”

<https://www.physicianassistantforum.com/topic/45710-pediatric-hourly-salary-without-benefits/>

Datos del debate “*Pediatric hourly salary without benefits?*”

<https://www.physicianassistantforum.com/topic/53635-pursuing-outpatient-psych-how-can-i-stay-up-to-date-on-general-medicine/>

Datos del debate y comentarios “*Pursuing outpatient psych - how can I stay up-to-date on general medicine?*”

<https://www.physicianassistantforum.com/topic/53586-skin-bones-hearts-private-parts>

Datos del debate “*Skin bones hearts private parts?*”

https://www.researchgate.net/publication/262778715_Clinical_neurology_Why_this_still_matters_in_the_21st_century

Datos del artículo de investigación “*Clinical neurology: Why this still matters in the 21st century*”

https://www.researchgate.net/publication/6856275_ABC_of_urology_-_Urological_evaluation

Datos del artículo de investigación “*ABC of urology - Urological evaluation*”

https://www.researchgate.net/publication/236264877_Allergy_and_Inflammation_An_Immunological_and_Therapeutic_Approach

Datos del artículo de investigación “*Allergy and Inflammation: An Immunological and Therapeutic Approach*”

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

Plataforma Firebase

<https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones>

Herramienta de control de versiones Git

Anexo. Manual de usuario

Registro de una cuenta nueva

Un usuario que no tenga ninguna cuenta en el sistema debe acceder a la vista de registro y completar los datos de su nombre completo, nombre de usuario que lo identifica en la aplicación, email, y dos contraseñas idénticas, para verificar que la contraseña que desea el usuario sea correcta. Cuando el usuario envía el formulario y todo es correcto, inicia sesión automáticamente si así lo desea.

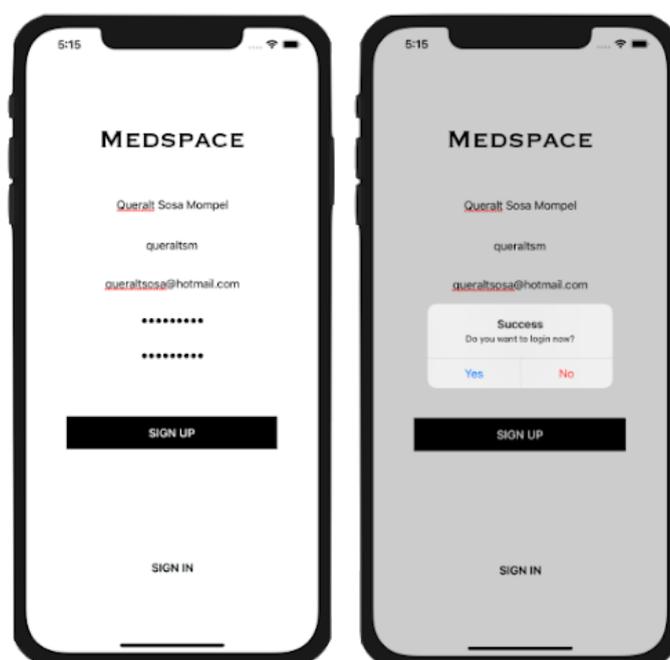


Figura 22. Anexo: Registro de una cuenta nueva

Inicio de sesión en el sistema

Para iniciar sesión el usuario completa los campos email y contraseña, a continuación se le dirige a la vista principal donde se encuentran todas las noticias publicadas.

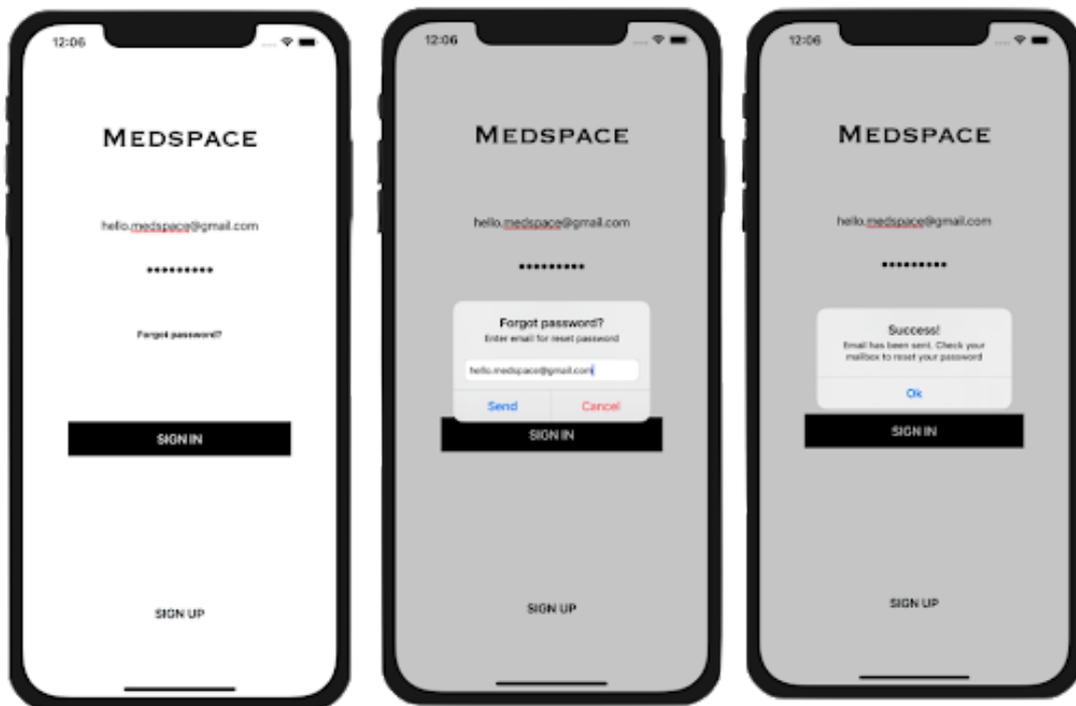


Figura 23. Anexo: Inicio de sesión en el sistema

Listado de publicaciones

El usuario pulsa el menú lateral y selecciona el tipo de publicación del que desea visualizar el listado de publicaciones, a continuación el sistema lo redirige a la correspondiente vista.

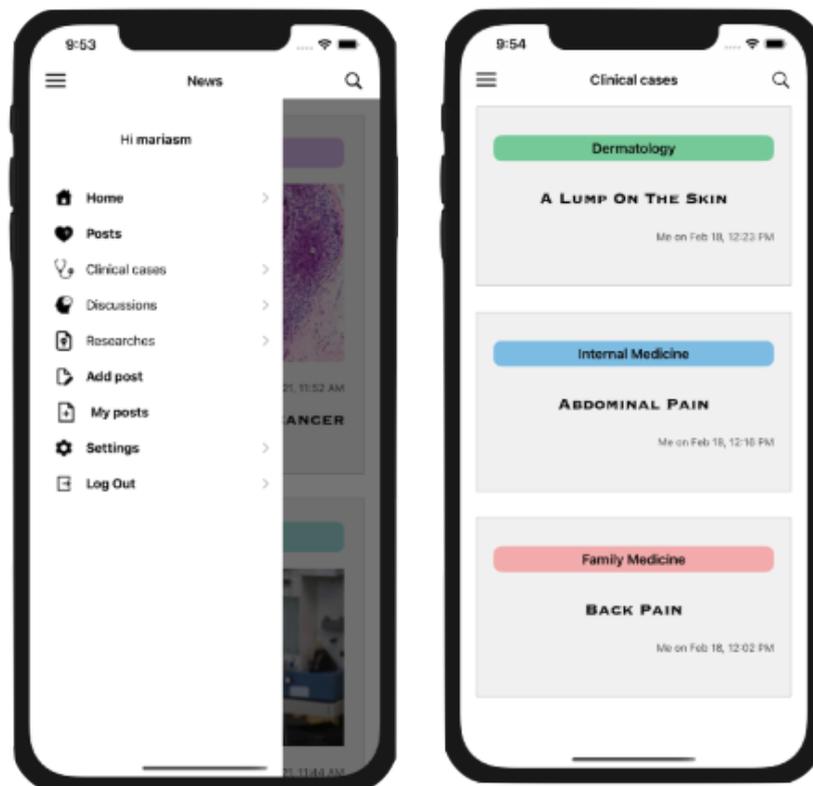


Figura 24. Anexo: Listado de publicaciones

Ver publicación en detalle

El usuario visualiza una vista con la información detallada del post, título, descripción, fecha, autor, especialidad médica, imagen si se trata de una noticia y un botón en la vista en detalle de un artículo de investigación para abrir su documento digital.

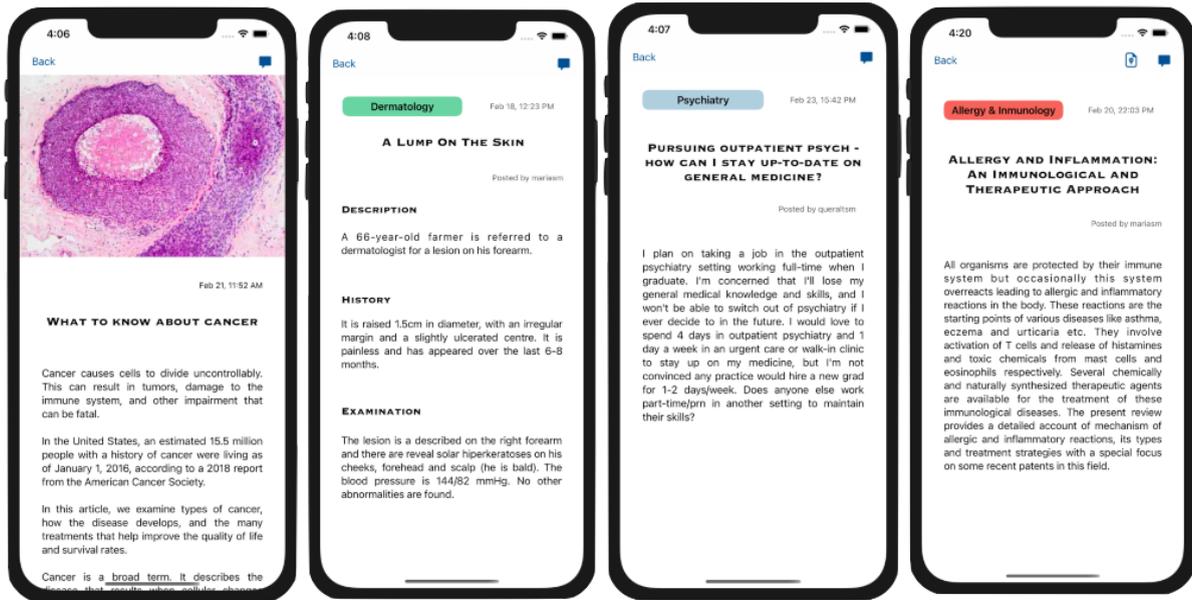


Figura 25. Anexo: Ver publicación en detalle

Registro de una publicación

El usuario completa los campos de texto del formulario de registro de publicaciones y envía el formulario. Cuando el administrador publica una noticia, además adjunta una imagen pulsando el icono de la cámara situada en el menú superior. Cuando el doctor publica un artículo de investigación tiene que subir un documento digital, para ello pulsa el icono de fichero situado al lado de “Document”.

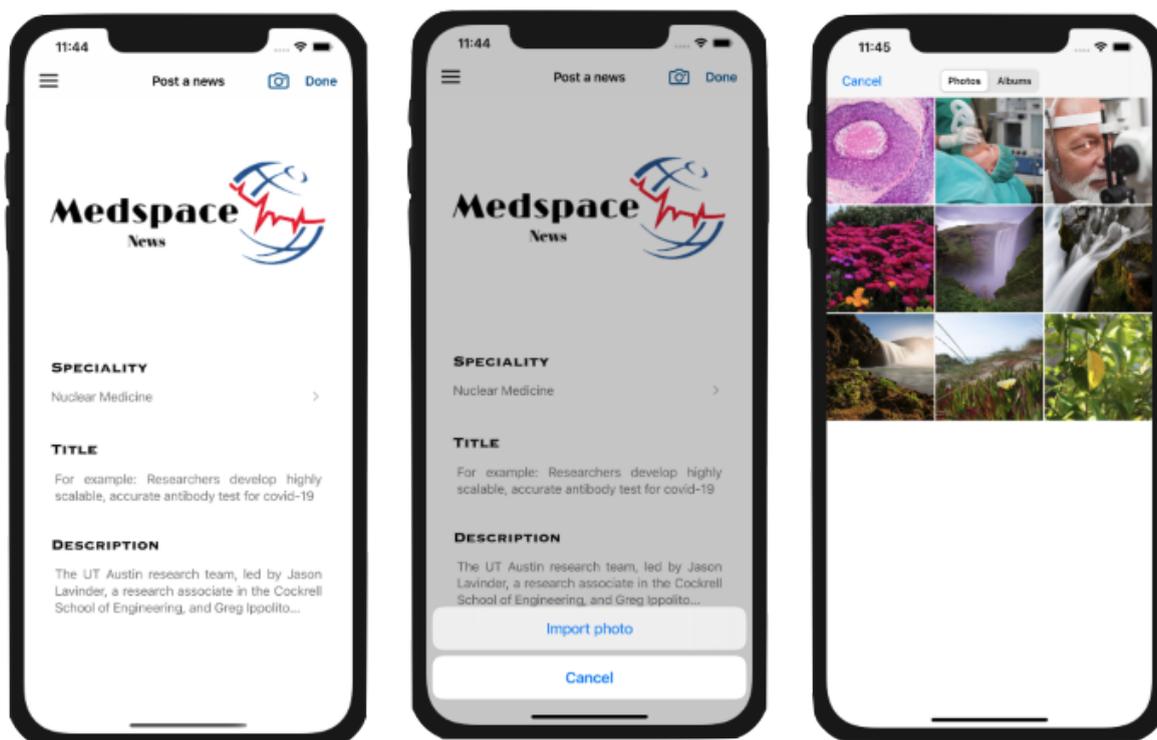


Figura 26. Anexo: Registro de una publicación I

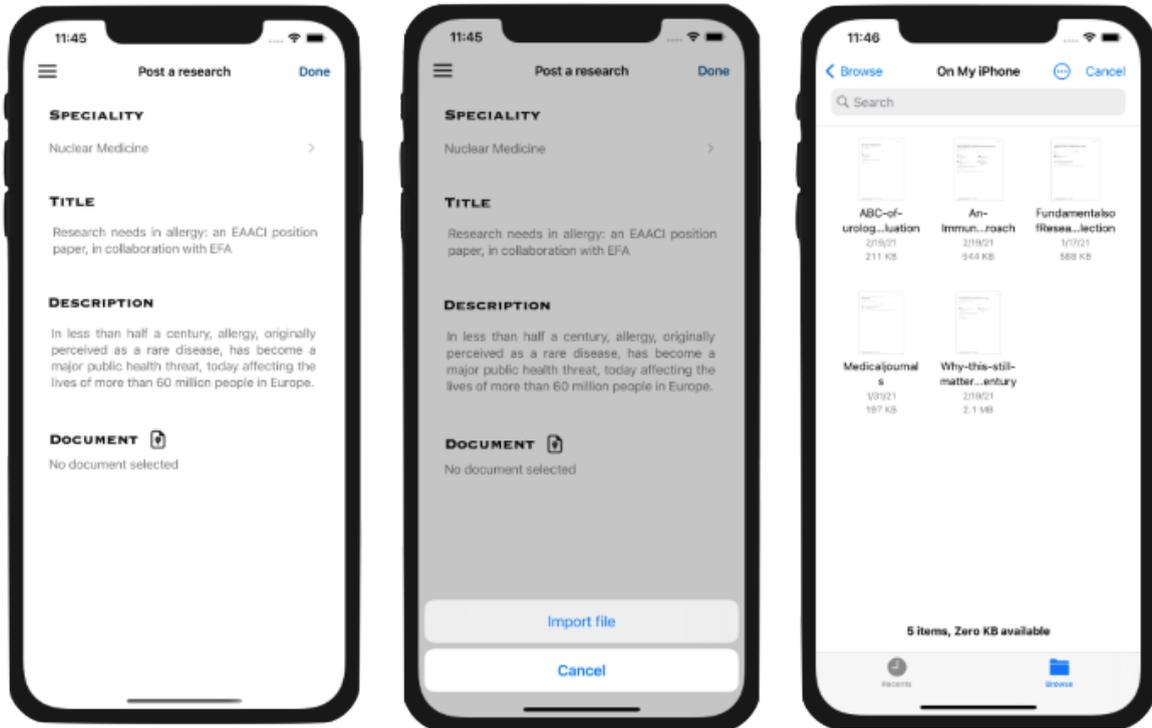


Figura 27. Anexo: Registro de una publicación II

Edición de una publicación

El usuario modifica pulsa el botón de edición situado en el menú superior. A continuación visualiza un formulario con los campos de texto modificables de la publicación, modifica el/los que desee y pulsa “Done” para efectuar los cambios.

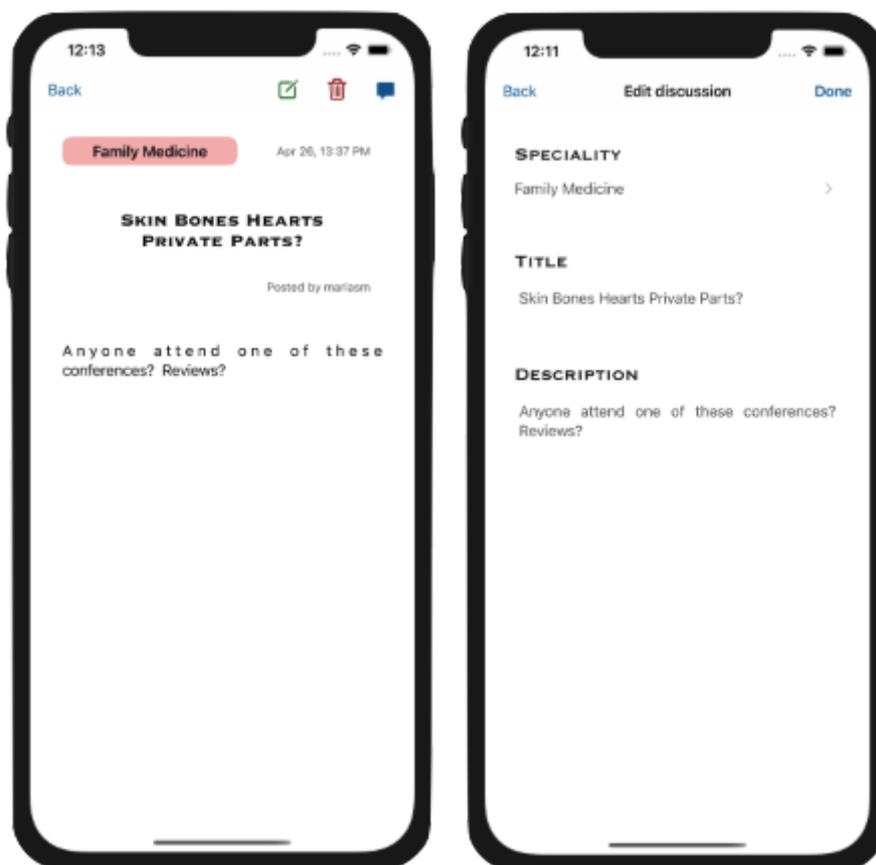


Figura 28. Anexo: Edición de una publicación

Borrado de una publicación

El usuario al haber seleccionado una publicación y encontrarse en la vista en detalle, pulsa el botón de borrar o papelera situado en el menú superior. Cuando selecciona alguna publicación en el listado de sus publicaciones, pulsa “All” para seleccionar todas o “Done” para borrar las seleccionadas.

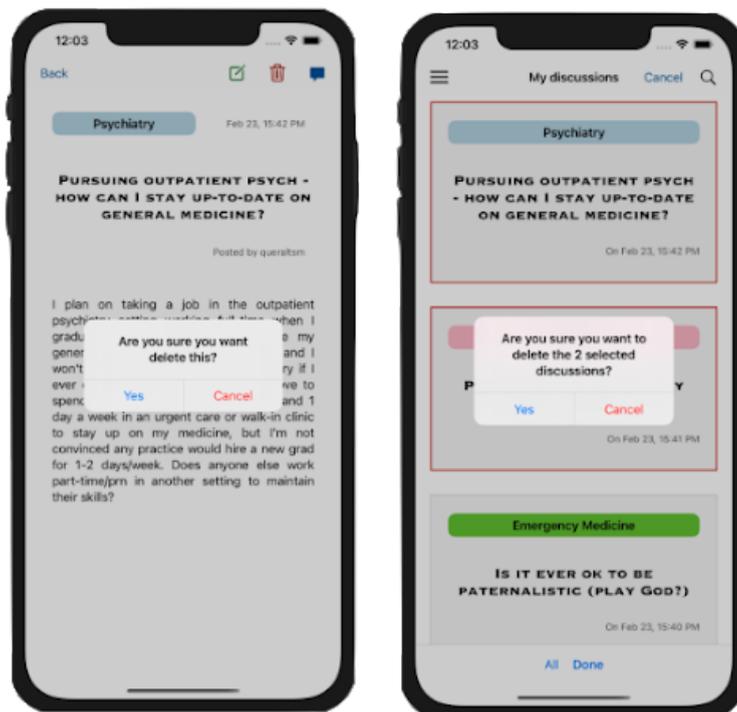


Figura 29. Anexo: Borrado de una publicación

Buscar una publicación

El usuario cuando se encuentra en la vista del listado de publicaciones pulsa el icono de búsqueda (lupa), debe completar el campo de texto introduciendo una palabra en concreto o una especialidad médica, verá los resultados obtenidos de la búsqueda dinámicamente.

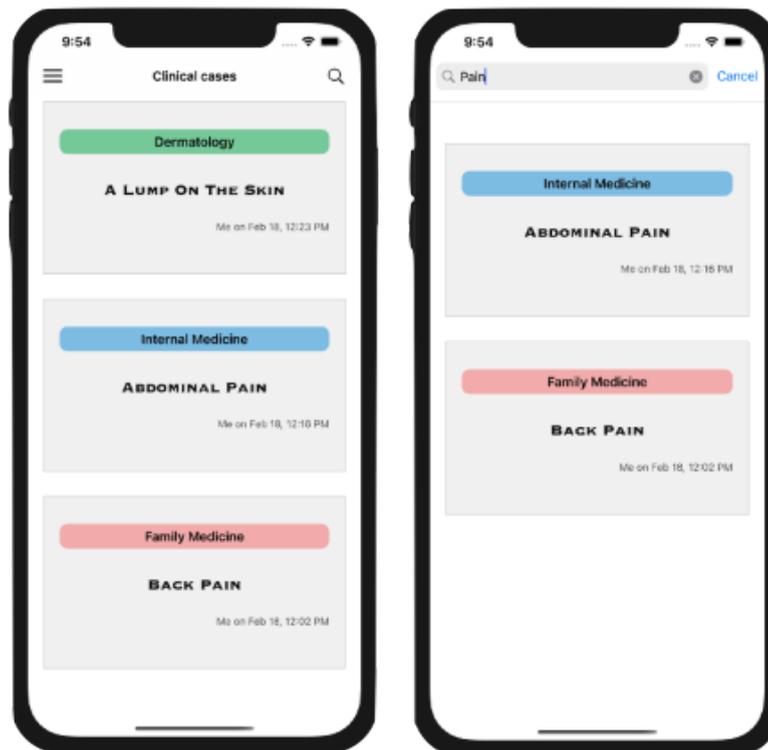


Figura 30. Anexo: Buscar una publicación

Gestionar comentarios

El usuario cuando se encuentra en la vista en detalle de una publicación, pulsa en el menú superior los comentarios y a continuación visualiza un listado de estos. Cuando pulsa “+”, visualiza un formulario con un campo de texto para introducir un mensaje, al enviarlo recibe un mensaje de confirmación y acto seguido encuentra su comentario en el listado.

Para empezar a editar o borrar un comentario desliza el mensaje, en el primer caso encuentra otro formulario con el mensaje que ha postado para ser modificado, en el segundo, debe confirmar un mensaje de alerta para eliminarlo.

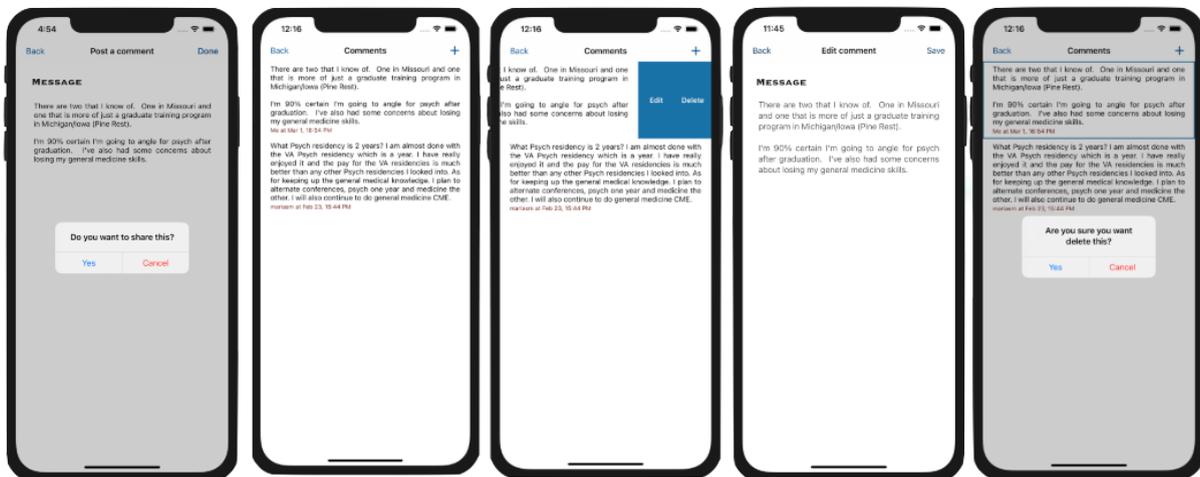


Figura 31. Anexo: Gestionar comentarios

Gestionar palabras clave

El usuario puede activar o desactivar las notificaciones deslizando un slider. Para registrar una nueva palabra clave pulsa el botón superior “+”, debe completar el cuadro de texto que le aparece a continuación y pulsar acabar, verá reflejado en la lista la nueva adición. Para eliminar una palabra clave del listado desliza la que desea borrar y pulsa aceptar.

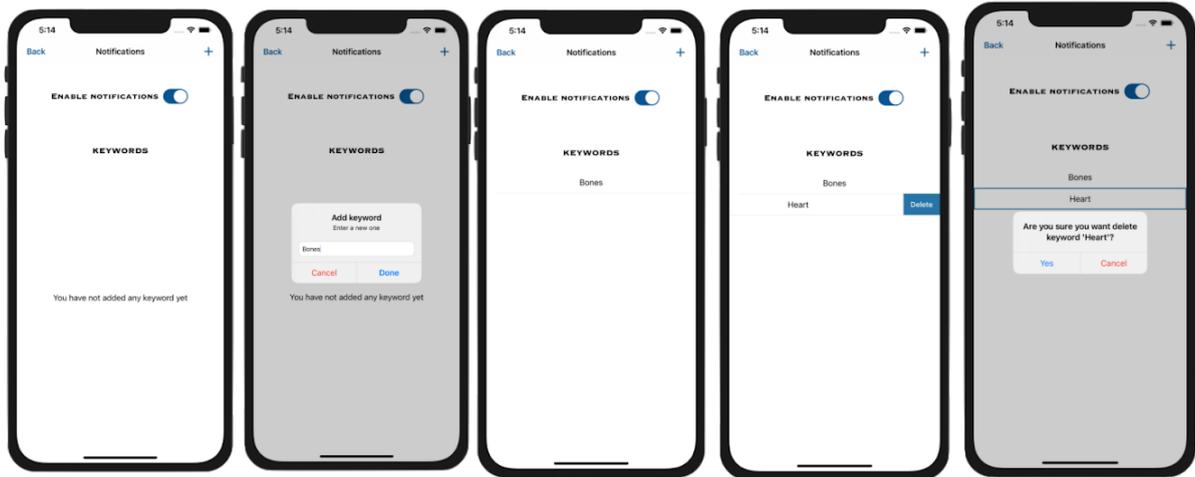


Figura 32. Anexo: Gestionar palabras clave

Darse de baja

El usuario se encuentra en la vista de configuración de su cuenta, pulsa “Delete account”, confirma la acción y se elimina su cuenta del sistema.

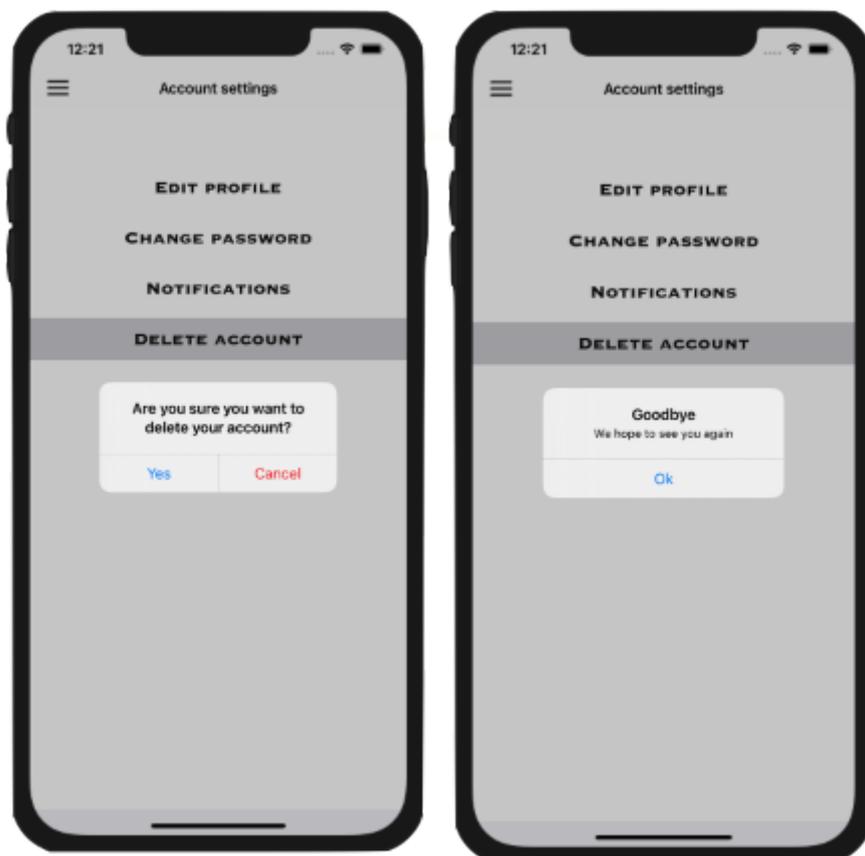


Figura 33. Anexo: Darse de baja

Actualización del perfil

El usuario en la vista de configuración de su cuenta pulsa “*Edit profile*”. A continuación se encuentra con dos campos de texto, que se refieren a su nombre completo y nombre de usuario en la app. Para actualizar la información de su cuenta modifica estos campos, si todo es correcto se procede. En caso contrario, muestra un mensaje de error.

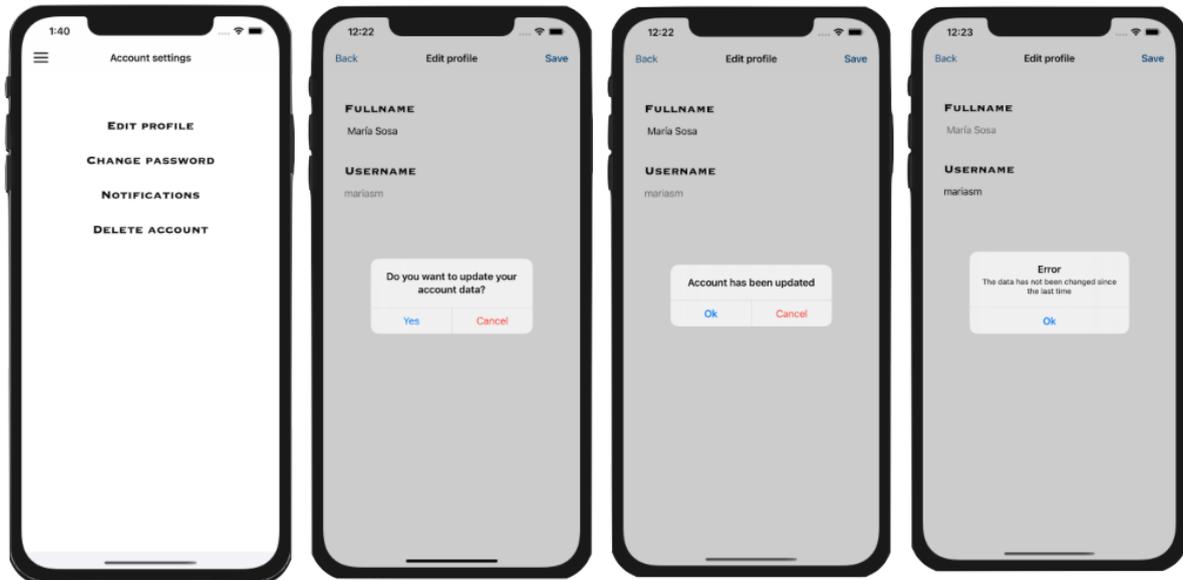


Figura 34. Anexo: Actualización del perfil

Notificaciones de publicaciones

El usuario en la vista de configuración de su cuenta pulsa “*Notifications*”. A continuación se encuentra con un switch para activar o desactivar las notificaciones, una tabla de palabras clave y un botón “+” para añadir una nueva. Cuando hace click en este aparece un formulario para completar.

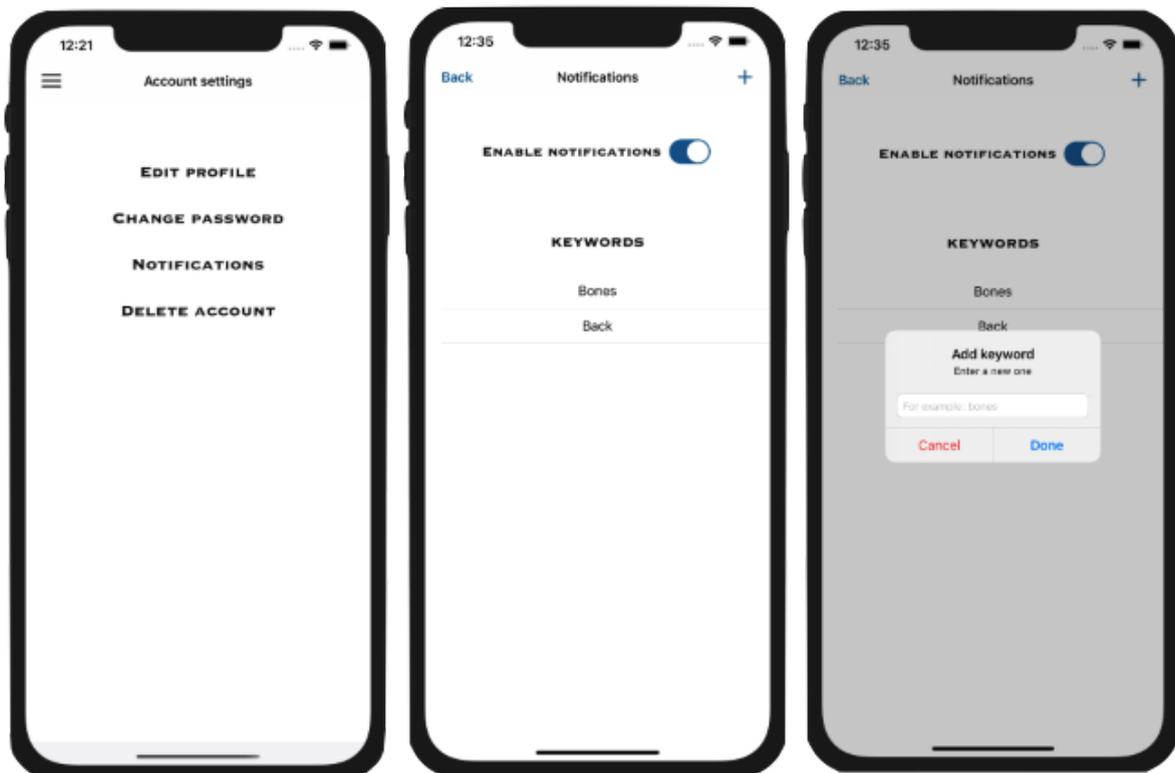


Figura 35. Anexo: Notificaciones de publicaciones I

Cuando el sistema notifica al usuario de una nueva publicación de su interés, aparece la notificación en su dispositivo.

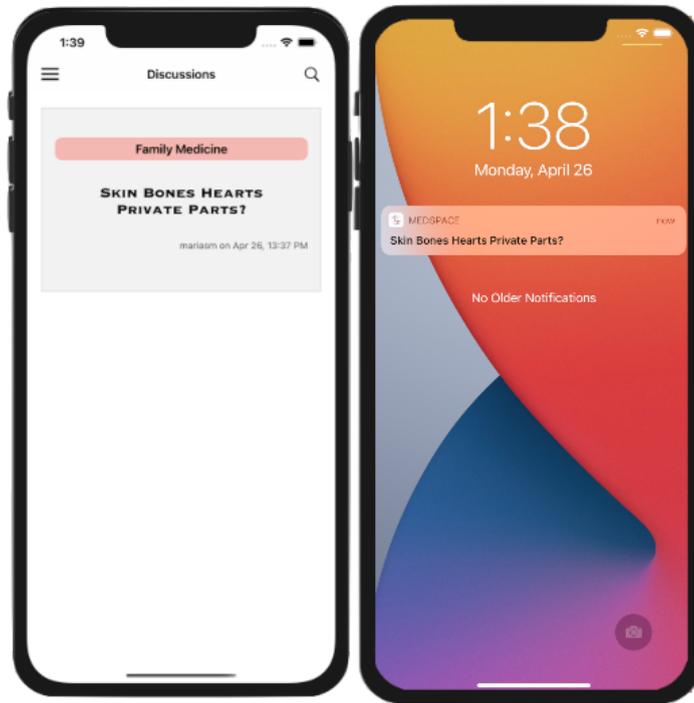


Figura 36. Anexo: Notificaciones de publicaciones II