Aplicación ios Para El Seguimiento Del Tratamiento Médico

Trabajo Fin de Máster
Curso 2018-2019

Autor
Andrés Aguirre Juárez

DIRECTOR
Adrián Riesco Rodríguez

Máster en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Aplicación ios Para El Seguimiento Del Tratamiento Médico

Trabajo de Fin de Máster en Ingeniería Informática
Departamento de Sistemas Informáticos y Computación

Autor
Andrés Aguirre Juárez

DIRECTOR
Adrián Riesco Rodríguez

Convocatoria: Septiembre 2019
Calificación: Notable 8

Máster en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

6 de SEPTIEMBRE de 2019

DEDICATORIA

A todos los que han confiado en mí.

También a mis padres, mis hermanos y a
José Javier, Iván y José Miguel.

AGRADECIMIENTOS

A mi familia y a mi novia por apoyarme cada día. A Pablo y María por todo el tiempo de preparación que hemos dedicado. A Juan Manuel por la paciencia que ha tenido.

RESUMEN

Aplicación móvil en el sector farmacéutico

Esta aplicación pretende sentar las bases para aprovechar la tecnología móvil en el ámbito de la farmacología aplicada a los pacientes. Se trata de una aplicación con la que podemos escanear un código QR de una caja de medicamentos, para obtener información acerca de esos medicamentos. Esto supone tener acceso al prospecto desde un dispositivo móvil para facilitar su lectura, abstenerse de imprimir un prospecto para cada caja, y tenerlo siempre a nuestra disposición. Además con esta aplicación se puede llevar el control de los medicamentos que conforman nuestro botiquín particular. Puesto que es una aplicación dedicada al control médico, dispone de una sección en la que el usuario puede llevar a cabo un control de su estado de salud. Esto podría servir al médico para obtener otro tipo de información más precisa, por ejemplo, saber cuánto tiempo desaparece el malestar con la cantidad de medicamentos ordenada por el anteriormente. Por otra parte, el usuario tendrá acceso a un mapa en el que se mostrarán las farmacias abiertas, así como la opción de tener acceso a la ruta desde la posición del usuario hasta la farmacia escogida. En último lugar, el usuario tendrá acceso a una sección de noticias relacionadas con la farmacología, la medicina y los productos sanitarios.

Palabras clave

Aplicaciones móviles, salud, control médico, medicinas, lector códigos QR…

ABSTRACT

Mobile application in the pharmaceutical sector

This application aims to lay the foundations for taking advantage of mobile technology in the field of pharmacology applied to patients. It is an application with which we can scan a QR code from a box of medicines to obtain information about those medicines. This means having access to the leaflet from a mobile device to make it easier to read, abstaining from printing a leaflet for each box, and always having it at our disposal. In addition, with this application you can keep track of the medicines that make up our particular first-aid kit. Since it is an application dedicated to medical control, it has a section in which the user can carry out a control of his state of health. This could help the doctor to obtain other more precise information, for example, to know how long the discomfort disappears with the amount of medicines ordered by the previous one. On the other hand, the user will have access to a map showing the open pharmacies, as well as the option of having access to the route from the user's position to the chosen pharmacy. Finally, the user will have access to a news section related to pharmacology, medicine and health products.

Keywords

Mobile applications, health, medical control, medicines, QR code reader

Índice de contenidos

Índice de figuras

Capítulo 1 - Introducción

El presente trabajo tiene como objetivo reunir en una aplicación móvil una serie de funcionalidades que permitan a un usuario o paciente, recabar toda la información posible acerca del sector farmacéutico y tener acceso a esa información de manera rápida y eficaz.

En la época en la que vivimos, rodeados de aparatos digitales, llama la atención que productos que tienen tanto uso como son los medicamentos, apenas dispongan de una aplicación móvil o una página web en la que poder realizar consultas y guardar esa información de manera sencilla. También llama la atención que se monitorice constantemente nuestra ubicación, nuestra actividad física, nuestros movimientos, pero ningún servicio tenga ninguna inquietud acerca de nuestro estado de emocional

Por último, llama la atención que una industria con tantos recursos como es la farmacéutica, no se preocupe por digitalizar la información de sus productos o los horarios de las farmacias. Es por esta razón por la que nos parece una buena idea que sirva como punto de entrada a la digitalización de esta industria.

1.1 Motivación

La idea principal en la que se ha basado este proyecto consiste en utilizar la tecnología móvil actual para facilitar al usuario tareas relacionadas con el sector farmacéutico. Estas tareas que se pretenden mejorar consisten actualmente en la búsqueda de información relativa a los medicamentos en distintos buscadores de Internet. La finalidad de esta aplicación es poder realizar una serie de tareas de información de manera rápida y sencilla.

1.2 Objetivos

Desarrollar una aplicación móvil con los siguientes requisitos:

1. Un usuario puede registrarse en la aplicación.
2. Un usuario puede hacer login en la aplicación. A partir de este momento se le considera "El usuario".
3. El usuario puede escanear un código QR y obtener información de un medicamento.
    a. El código QR debe corresponder al de la caja del medicamento.
    b. En caso de no poder realizarse, se creará uno que cumpla la función.
4. El usuario puede añadir ese medicamento a su botiquín. (El botiquín no es más que su lista de medicamentos.
5. El usuario puede consultar su botiquín en cualquier momento siempre que tenga conectividad a Internet.
6. El usuario puede realizar un seguimiento de su estado de ánimo.
7. El usuario puede encontrar farmacias abiertas.
8. El usuario puede conocer el camino a la farmacia que escoja.

Conocer qué datos aporta un código QR de una caja de medicamentos y conocer cuál es el método de encriptación de esos datos.

1.3 Plan de trabajo

Aquí se describe el plan de trabajo a seguir para la consecución de los objetivos descritos en el apartado anterior.

La primera tarea que se realizará consistirá en realizar una investigación acerca de los medicamentos. Qué necesita conocer un usuario acerca de los medicamentos, cómo realiza un usuario la tarea de investigación acerca de un medicamento, qué problemas tiene el usuario con los prospectos, para qué sirve el código QR de las cajas de medicamentos, aportan información útil para el usuario. Estas son algunas de las preguntas que han de responderse para conocer más acerca de la farmacología.

Después de realizar la tarea de investigación, debe realizarse una tarea de análisis de las herramientas. Esta tarea de análisis tiene como finalidad buscar una

solución informática, en este caso en forma de aplicación móvil, para aportar al usuario una herramienta con la que pueda realizar un conjunto de acciones, descritas en el apartado de objetivos. Para llegar a esta solución, se debe realizar un debate con el director del TFM.

Una vez finalizada la tarea de análisis, se procederá a la tarea de desarrollo de la aplicación. Para esta tarea, se realizarán entregas iterables como forma de autoevaluación. Estas entregas consistirán en realizar una revisión del estado actual de la aplicación.

Para finalizar se realizará una entrega tanto del código desarrollado para la aplicación, como la presente memoria.

Puesto que la tarea de investigación no se completó a tiempo, ha surgido un cambio en el plan de trabajo que ha hecho que las distintas entregas sufran cambios no previstos entre ellas. Este hecho provocó que los códigos QR de las cajas de medicamentos no puedan leerse con la aplicación. En su defecto, la aplicación lee códigos QR diseñados previamente para la aplicación, y después de realizar su lectura, muestra al usuario los datos asociados a ese medicamento.

1.4 Descripción de la memoria

El resto de la memoria se estructura de la siguiente manera. En el capítulo 2 se explica cómo ha sido el proceso de investigación, esto incluye la búsqueda de información relativa a la información que presentan los medicamentos, qué tipo de información muestran los códigos QR de las cajas de medicamentos y también, qué tipo de funcionalidades han podido implementarse en la aplicación y por otro lado, cuáles no.

Más adelante, en el capítulo 3, se explica la solución tecnológica en la que se ha basado la aplicación. Este capítulo incluye desde las herramientas que se han utilizado, hasta el patrón de diseño.

Por otro lado, en el capítulo 4, se encuentra lado estructura de la aplicación. Esta estructura se divide en subsecciones entre las que se encuentran las librerías externas utilizadas, y las pantallas que conforman la aplicación.

Para finalizar, en el capítulo 5, pueden encontrarse las conclusiones del trabajo, así como los trabajos futuros que pueden implementarse en la aplicación.

Capítulo 2 - Estado de la cuestión

En esta sección se explicará cómo ha sido del proceso de investigación, partiendo desde la primera reunión entre el alumno y el director. En esta primera reunión con el director hicimos una sesión de lluvia de ideas en la que aportamos un listado de ideas para comenzar a investigar. Hay una gran diferencia entre ese listado inicial de ideas que podría contener la aplicación y la aplicación desarrollada. Esto ha sido debido a las múltiples circunstancias que nos hemos encontrado a la hora de investigar y desarrollar la aplicación.

En un principio, este fue el listado de ideas:

- Log in.
- BBDD con medicinas/enfermedades/tiempos en el nube.
- Usar cámara para escanear medicinas.
- Usar GPS para solicitar ayuda.
- Comprobar contra-indicaciones / alergias.
- Contador de cantidades -> alarma comprar.
- Recomendación genéricos.
- Comprar -> recetas/avisar médico.
- ¿Ajustar tamaño letra?
- Integrar para invidentes
- Roles médico/usuario/familiar.
- Alarma para avisar.
- Integración con Siri.
- Notificación a familiares.

La tarea de investigación comenzó en octubre de 2018. Es en esa fecha cuando se comenzó a observar de manera más frecuente la aparición de códigos QR en las cajas de los medicamentos. Después de estar conversando con mi director acerca de una aplicación de medicamentos, se me ocurrió que ya habría alguna aplicación ofreciendo el servicio de lectura de códigos QR de los medicamentos.

La inquietud me llevó a intentar leer el código QR con la cámara de mi teléfono móvil, puesto que por defecto, la cámara de los iPhone lee los códigos QR. La sorpresa llegó cuando la cámara no me devolvía ningún resultado. Esa misma tarde comencé a investigar.

En diciembre de 2018 se me ocurrió consultar a la Agencia Española de Medicamentos y Productos Sanitarios (AEMPD). Contacté con AEMPD por medio del teléfono de Información general [1].

En la propia llamada telefónica reconocieron que no sabían explicarme qué información aportaba el código QR, por lo que me remitieron a su página web, en la que podría encontrar un formulario para atender mi pregunta. Rellené el cuestionario con mis datos personales con el fin de conocer qué era lo que escondía el código QR. No he obtenido respuesta aún. Prosiguiendo con mi investigación, consulté con un familiar que regenta una residencia para personas de la tercera edad. Por suerte, en unos días vendría una farmacéutica a explicarles para qué sirve el código QR. Fue en esta reunión donde conocí los usos del código QR en las cajas de medicamentos.

El 9 de febrero de 2019 entró en vigor el nuevo reglamento de la UE que pretende luchar contra la falsificación de medicamentos [2]. En este reglamento se establece la obligatoriedad de incluir la trazabilidad de los medicamentos, esto es, saber quién produce el medicamento, en qué fecha lo hizo, a quién se lo entregó, quién lo almacenó y quién lo vendió [3].

Partiendo de esta información, y teniendo en cuenta la estructura actual de las cajas de medicamentos, que además de incluir los propios medicamentos, incluyen un prospecto con la letra muy pequeña, y repetido en cada medicamento, surge la idea de utilizar un código QR propio en el que incluir toda la información de los medicamentos. Además, se propuso añadir a la aplicación un enlace directo con la

AEMPS en el que se puede notificar de la sospecha de contraindicaciones de los medicamentos. Puesto que esta es una web externa, se debería redirigir a otra aplicación, como por ejemplo un navegador [4].

Por otro lado, se realizó una búsqueda en la tienda de aplicaciones de Apple (App Store) tratando de encontrar una aplicación de la AEMPS. Esta agencia posee una aplicación actualmente. En ella se puede encontrar la información de cada medicamento. En mi investigación llegué a tratar con la aplicación y observé que las llamadas a los servicios que utiliza la aplicación no están cifradas. Esto produce que podamos realizar llamadas a ese servicio para tener la misma funcionalidad. Descartamos esta propuesta porque era ilegal.

Después de todas las experiencias, llegamos a las siguientes conclusiones:

1. La aplicación debe servir como buscador de medicamentos (al estilo de la App de la AEMPS).
2. La aplicación debe poder leer códigos QR pero con información de los medicamentos.
3. El usuario puede guardar la información de los medicamentos en su dispositivo a modo de botiquín.
4. La aplicación debe incluir un enlace externo para poder informar de las contraindicaciones de un medicamento.
5. Un usuario debe poder almacenar en su dispositivo los distintos estados de ánimo a lo largo del día.

Además de las tareas de investigación en el sector farmacéutico, también hubo una labor de investigación en el entorno del desarrollo de aplicaciones iOS. En un primer momento, se pensó en esta aplicación como una aplicación que podría ayudar a personas dependientes a mostrar imágenes nítidas de los medicamentos, solicitar ayuda a la persona a cargo del paciente, o incluso poder establecer algún tipo de relación entre el paciente y el médico pudiendo llegar a enviar mensajes. También se pensó en integrar *Siri* en la aplicación para realizar comando tan simples como: "Oye *Siri*, cuál es el prospecto del ibuprofeno", o también "Oye *Siri*, necesito ayuda, avisa a mi responsable".

Apple permite a los desarrolladores realizar una gran multitud de funcionalidades, pero se reserva algunos derechos para los desarrolladores que disponen de una cuenta de *iOS Developer Program* o *iOS Developer Enterprise Program*. Dado que la universidad no dispone de ninguno de los dos roles mencionados anteriormente, y que el alumno solo dispone de una *Account Developer Website,* algunas de estas funcionalidades no han podido desarrollarse.

Para entrar más en detalle, se explicará por qué se han abandonado distintas funcionalidades de la aplicación.

Integración de comandos con *Siri*: Esta funcionalidad fue la primera que surgió en la primera conversación con el director. En la carrera profesional del alumno ya se ha desarrollado algún comando con *Siri*, por lo que era a priori una tarea sencilla, y aportaba un gran valor a la aplicación. Desafortunadamente, el desarrollo de aplicaciones utilizando *SiriKit*, está restringido.

Envío de notificaciones *push*: Esta funcionalidad pretendía utilizarse de dos formas bien diferenciadas. La primera podría haber sido al enviar mensajes entre un médico y un paciente. Al enviar un mensaje, se enviaría una notificación para avisar al otro usuario para informarle de que debe mirar el chat. La otra funcionalidad que podría haberse desarrollado consistiría en el envío de notificaciones de un usuario dependiente de otro. Por ejemplo, una persona dependiente, en caso de necesitar ayuda, podría enviar una notificación a la persona responsable de esta. La funcionalidad de solicitar ayuda se descartó puesto que es imposible el envío de notificaciones *push* sin disponer de cuenta de desarrollador. No aportaría valor a la aplicación el tener a un usuario responsable del otro, constantemente en la aplicación para saber cuando éste necesita ayuda. Es por esta razón por la que se desestimó realizar la funcionalidad de la mensajería.

Por estas razones también se desechó hacer la aplicación únicamente para personas dependientes.

Capítulo 3 - Solución tecnológica

La aplicación desarrollada se ha realizado utilizando la tecnología que proporciona Apple a sus desarrolladores. El entorno de desarrollo utilizado ha sido XCode que está integrado el propio sistema operativo de Mac. Para el desarrollo del presente trabajo, se ha utilizado tanto el ordenador personal, para realizar las tareas de investigación y desarrollo, como el dispositivo móvil personal, para las pruebas correspondientes. Por otra parte, el lenguaje en el que se ha desarrollado la aplicación es Swift.

Para la aplicación se pensó en un primer lugar en una arquitectura MVC. Esta solución es óptima en aplicaciones en las que trabaja un único desarrollador, teniendo en cuenta que la extensión de la aplicación no es muy grande.

Se pensó en la utilidad de la aplicación y en que podría servir para trabajos futuros, por lo que podría requerir de escalabilidad. Esto se opone con el patrón MVC por lo que se decidió utilizar la arquitectura de VIPER[5].

VIPER es una arquitectura cuyo nombre proviene del acrónimo de *View, Interactor, Presenter, Entity y Router.* Está basada en el principio de responsabilidad única, en el que cada módulo o clase debe tener responsabilidad sobre una única parte de la funcionalidad.

A continuación se procede a explicar en qué consiste cada clase. Ha de saberse que a la hora de desarrollar una aplicación en *SWIFT*, para facilitar el desarrollo, la aplicación se divide en lo que se denomina *View Controllers*. Un *View Controller* gestiona un conjunto de vistas y ayuda a crear la interfaz de usuario. Fuera del ámbito del desarrollo de aplicaciones móviles, un *View Controller* es una pantalla.

Cada *View Controller* está formado por un conjunto de clases que forman la estructura de VIPER, esto quiere decir que cada *View Controller* tiene sus propias clases de *View, Interactor, Presenter, Entity y Router*. A continuación se explicará en qué consiste cada una de ellas.

Al crear un *View Controller* con las clases correspondientes, se crea una clase adicional, la clase *Assembly*, que proporciona las conexiones necesarias entre el conjunto de clases, asociadas al mismo *View Controller,* utilizando referencias a las otras clases del mismo conjunto. La estructura que se crea es la siguiente:

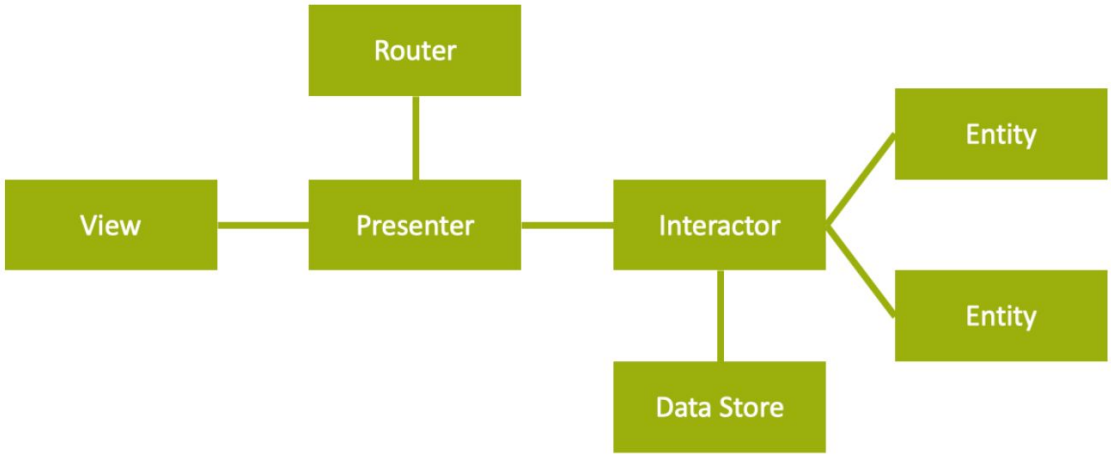


Ilustración 1. Arquitectura VIPER

Entity: representan la clases y/o estructuras que forman los modelos que se van a utilizar en el conjunto de clases correspondiente a cada *View Controller*. En nuestro caso, todos los *View Controller* con sus correspondientes conjuntos de clases, están contenidos en un mismo módulo. Esto hace que las *Entity* que se crean en un módulo VIPER, no tengan que ser replicadas para otro *View Controller* que quiera utilizarlas, porque por defecto, un objeto tiene un acceso *Internal* que hace que sea conocido por todo el módulo.

View: Contiene las referencias de todos los objetos del *ViewController*. En esta clase no se realiza ninguna lógica. Aquí se implementan los métodos delegados de las tablas de datos. La lógica menor se realiza en el *Presenter*, del que además tiene una referencia la View.

Presenter: Realiza la lógica interna del *View Controller*. El *Presenter* tiene acceso al *Interactor*, al *Router* y a la *View*, por lo que si un botón tiene que proporcionar datos a esa vista, el botón llamará a un método de View, este al *Presenter*, y el *Presenter* al

Interactor. La respuesta la devolverá en sentido inverso, hasta que el *Presenter* haga cambios en la vista. En caso de tener que navegar a otra pantalla, el *Presenter* llamará al *Router* y este invocará al siguiente controlador, o volverá al anterior.

Interactor: Este objeto se encarga de la capa de datos del *View Controller*. Cuando se invoca al *Interactor* desde el *Presenter*, es porque la aplicación requiere de datos. En un primer momento, al crearse el *View Controller,* éste carece de datos, por lo que el *Interactor* debe realizar las gestiones necesarias para proveer al *View Controller* de esos datos. Estos datos pueden obtenerse a través de algún servicio web, o pueden estar guardados en el dispositivo de manera local. En cualquier caso, el *Interactor* debe guardar los datos que ha consultado en un primer momento, pero solo debe proveer a la aplicación de los datos que le han sido requeridos. Para que se entienda de con un ejemplo, pongámonos en el caso en el que el *Presenter* necesita los últimos cinco medicamentos que ha consumido el usuario, puesto que son los que aparecen en una vista paginada. El servicio web que está desarrollado para esta funcionalidad devuelve los últimos 30 medicamentos. El usuario podría realizar hasta cinco paginaciones más sin que el *Interactor* tenga que recuperar más datos del servicio ya que los correspondientes a las seis primeras páginas los devolvió el servicio con la primera respuesta.

Router: Se encarga de realizar la navegación de un *View Controller* a otro. Esto lo hace utilizando el componente nativo llamado *Navigation Controller,* que es el que se encarga de la navegación. Es una pila de objetos de tipo *View Controller*. Cuando se instancia un nuevo *View Controller,* se incluye en la pila, en cambio, cuando se destruye un *View Controller,* se desapila. El *View Controller* que se muestra en la pantalla hace referencia al último elemento añadido a la pila. Además de apilar o desapilar un *View Controller,* se encarga de la inyección de dependencias.

A continuación se describe un ejemplo para comprender qué es la inyección de dependencias.

Desde un *View Controller* que llamaremos *Login* se realiza una navegación al *View Controller* que llamaremos *Home*. Cuando hacemos *login* en la aplicación, la Home necesita el nombre de usuario que devuelve el servicio al hacer *login*. La

manera de pasar datos de un *View Controller* a otro es por medio de la ya citada inyección de dependencias. Para ello, desde la clase *Router* correspondiente al *View Controller* de *Login* se inyecta un objeto con la información que queremos inyectar, en caso el nombre del usuario.

Capítulo 4 - Estructura de la aplicación

En este capítulo se pretende explicar el conjunto de dependencias y pantallas de la aplicación, así como su funcionamiento.

4.1 Gestión de dependencias externas

En un principio, para la instalación de dependencias se pensó en utilizar *Carthage* [6]. *Carthage* es un gestor de dependencias que descarga las librerías o *frameworks* ya compilados. Al intentar realizar la instalación de dependencias de *Firebase* con *Carthage* se descubrió un problema sin solución aparente. Después de una tarde de investigación se decidió migrar a otro gestor de dependencias. Es por esto que para la instalación de dependencias en el proyecto se ha utilizado *CocoaPods* [7]. *CocoaPods* es un gestor de dependencias para *Objective-C y Swift* que descarga las librerías o *frameworks* sin compilar, y es en la tarea de compilación del proyecto cuando se compilan en un primer paso las dependencias, y en un segundo paso el proyecto. Es una tarea un poco más tediosa con *CocoaPods*, pero a la vez más segura.

4.1.1 Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y móviles, propiedad de Google [8]. Es una plataforma *cloud* que proporciona a los desarrolladores una plataforma sencilla y escalable. Utiliza herramientas multiplataforma y conforma un entorno en el que no es necesario ningún servidor adicional a parte del de Google.

Para añadir *Firebase* al proyecto, primero se debe descargar las dependencias de CocoaPods [9]. Después en la plataforma de Firebase se debe añadir el bundle de la aplicación y acto seguido, descargar el archivo "GoogleService-Info.plist" para añadirlo a la raíz del proyecto.

Firebase provee al desarrollador de un SDK para trabajar. En el caso de la aplicación Control Médico, se han utilizado dos módulos de *Firebase*. El primero de los

dos módulos que se ha instalado es el módulo de *Authentication*. Este módulo gestiona la autenticación de los usuarios. Para el registro de un usuario solo es necesario un correo y una contraseña. El segundo módulo que se utiliza en la aplicación es el módulo de *Database*. Dentro del módulo de base de datos de *Firebase* existen dos tipos de bases de datos diferenciadas. La primera base de datos es Cloud *Firestone*, la segunda, y la que se utiliza es Real Time *Database*. La base de datos está estructurada en forma de árbol. De este árbol surgen tres ramas.

1. Medicinas. Se guarda la información relativa a cada medicamento. Se ha almacenado la información de los medicamentos puesto que no se ha encontrado ninguna base de datos con la información de los medicamentos y con una API documentada. Además esta aplicación trata de ser una prueba de lo que se puede realizar con una aplicación.
2. Farmacias. De la misma manera que se buscó una base de datos con una API documentada para los medicamentos, trató de buscarse otra con las farmacias presentes en toda España, con la información de apertura y cierre de cada farmacia cada día.
3. Medicinas del usuario. Se permite al usuario guardar los medicamentos como si estuvieran presentes en su botiquín. Con esto, se pretende que el usuario guarde las medicinas que tiene disponibles, para tener un control, así como tener disponible en todo momento su información.

4.1.2 SwiftLint

SwiftLint es una herramienta para cumplir con un código de buenas prácticas. Se puede agregar a *XCode* de manera sencilla y establecer reglas propias de buenas prácticas [10]. En este caso, por ejemplo, se ha añadido una regla que genera un error en el proyecto cuando se crea una variable con menos de tres caracteres de longitud, impidiendo que el proyecto compile correctamente. Con esta regla se trata de proporcionar un nombre descriptivo a cada variable. Para la instalación de *SwiftLint* se han seguido las instrucciones descritas en el repositorio de *SwiftLint*, y se ha realizado la instalación por medio de *CocoaPods*.

4.1.3 Charts

Charts es una librería para realizar gráficas [11]. En esta aplicación, se realiza una gráfica para representar el estado emocional del usuario. Para la instalación de esta librería, también se ha hecho uso de *CocoaPods*.

4.2 Pantallas de la aplicación

A continuación se explicará en detalle en qué consiste cada una de las pantallas de la aplicación. La pantalla corresponde a la primera pantalla con la que el usuario puede interactuar. Cuando el usuario hace *Login*, el usuario dispone de cuatro distintas pantallas en las que podrá interactuar por medio de un *TabBarController*.

4.2.1 Splash Screen

Esta pantalla aparece siempre que se carga la aplicación por primera vez. No permite al usuario interactuar con ella.

Ilustración 2. Splash Screen

4.2.2 Pantalla de Login y Registro

Esta pantalla es la primera con la que el usuario puede interactuar. Esta pantalla está conectada con el servicio de *Authentication* de *Firebase*. Desde esta pantalla el usuario puede registrarse en el servicio para acceder, o directamente acceder.

En el caso en el que el usuario seleccione el botón "Regístrate", se mostrará una ventana de alerta en la que el usuario debe rellenar con su correo y su contraseña elegida. En el caso en el que el usuario introduzca un correo que Firebase admita como válido, pasados unos segundos, se realizará la navegación dentro de la aplicación, y navegará a la pantalla de Noticias.

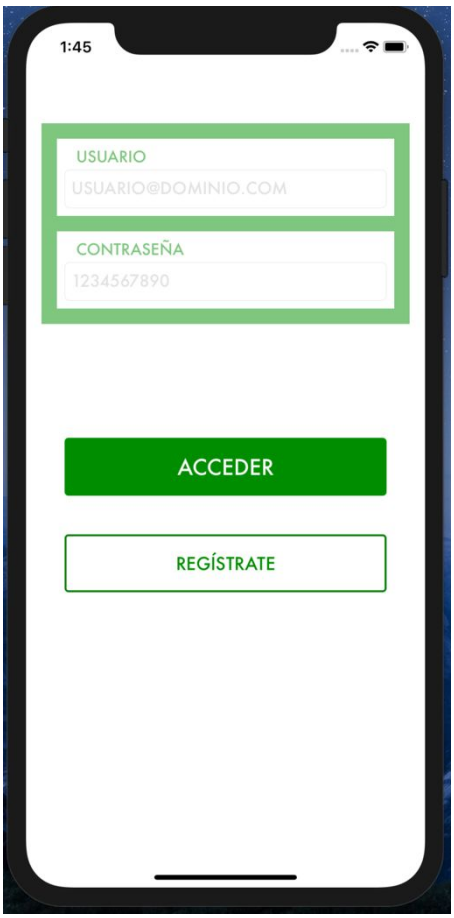


Ilustración 3. Login normal

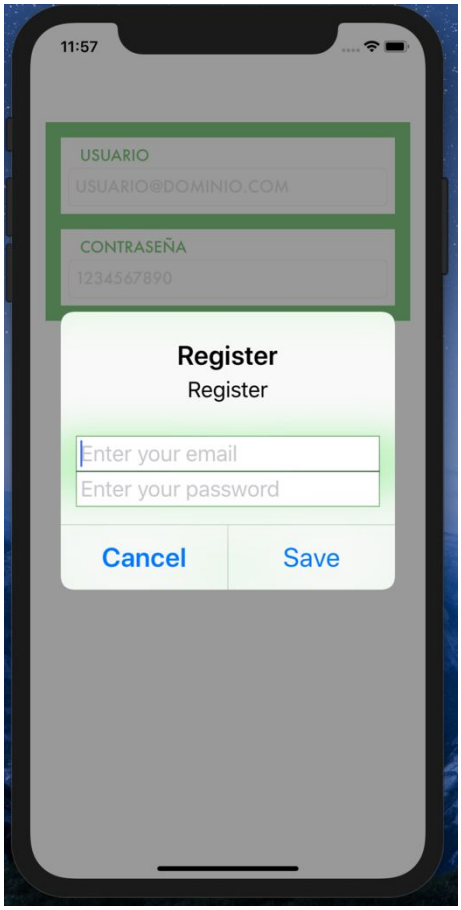


Ilustración 4. Login con ventana de registro

En el caso en el que el usuario decida entrar a la aplicación con su usuario y contraseña, debe rellenar las cajas de texto con su usuario y contraseña y pulsar el botón "Acceder".

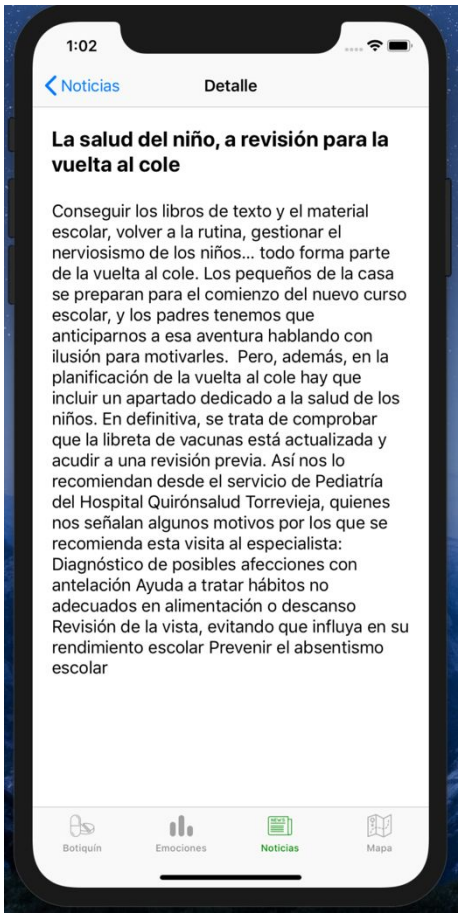
4.2.3 Pantalla de Noticias

La pantalla de noticias es la primera que el usuario puede encontrar una vez que entra en la aplicación. Aquí se le muestra un conjunto de noticias proporcionadas por un *feed* de noticias. Esta es una de las cuatro pantallas que forman el *TabBarController*.

Ilustración 5.Noticias

4.2.4 Pantalla de detalle de Noticias

En esta pantalla, el usuario puede leer las noticias en su versión completa.



4.2.5 Pantalla de Estado emocional

Desde esta pantalla el usuario puede ver un histórico de su estado emocional. Además puede seleccionar su estado de ánimo actual para que quede constancia de cómo se encuentra. Esta pantalla está pensada para que además, una tercera persona pueda ver la evolución del estado emocional del usuario, como puede ser un médico, o una persona que actúe como responsable. Los iconos utilizados en la aplicación han sido tomados de la página de Flat Icon [12].

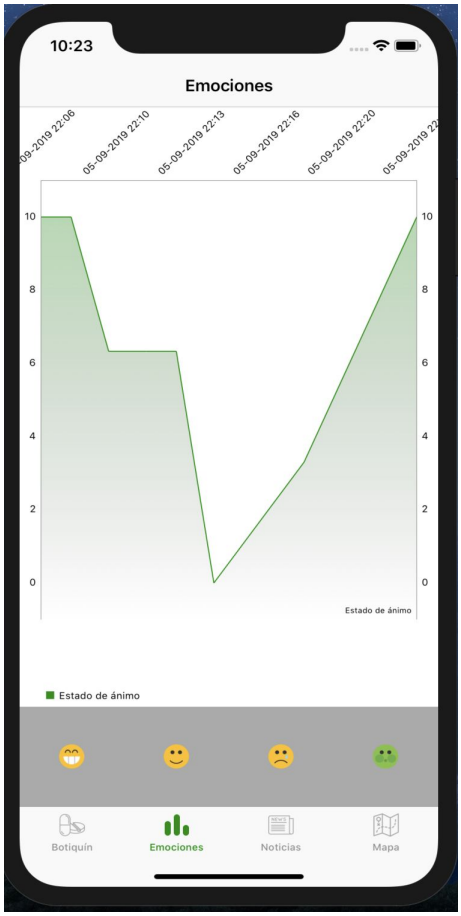


Ilustración 6. Estado emocional

4.2.6 Pantalla de Listado de medicinas

En esta pantalla el usuario tiene un listado de medicinas. Desde esta pantalla, el usuario puede acceder al detalle de cada una de las medicinas. Además, el usuario puede abrir su cámara para escanear un código QR. Después de escanearlo, el usuario navegará a la pantalla de detalle de medicina.

Ilustración 7. Listado de medicinas

4.2.7 Pantalla de Detalle de medicinas

Aquí es donde el usuario puede encontrar la información relativa a cada medicamento. Además, se puede añadir el medicamento desde esta pantalla al listado de medicinas del usuario. Para utilizar la cámara y leer códigos QR se ha realizado un tutorial [13].

Ilustración 8. Detalle de medicinas

4.2.8 Pantalla de Farmacias

El usuario puede encontrar las farmacias abiertas así como la ruta a cada una de ellas. Las farmacias que se muestran en esta pantalla no son reales, puesto que no se dispone de un listado de farmacias con el horario y las guardias de cada una.

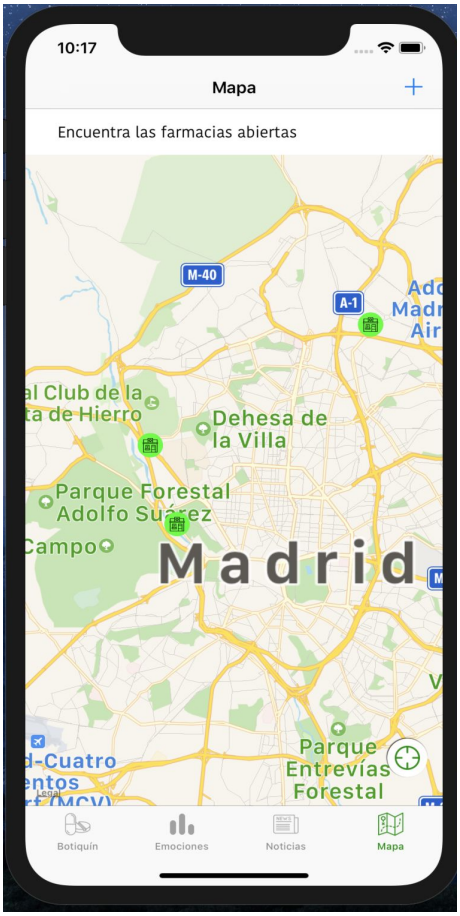


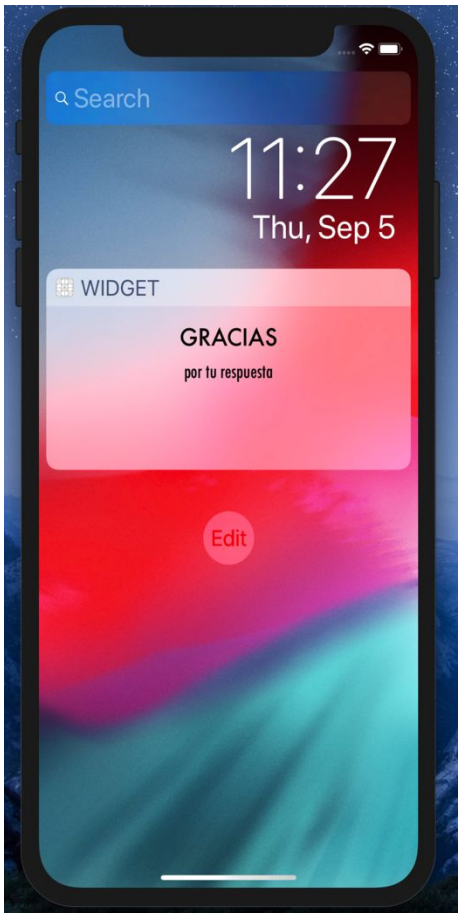
Ilustración 9. Mapa de las farmacias

Además, en esta pantalla, el usuario puede abrir la aplicación de Mapas seleccionando una farmacia y pulsando el botón "IR".

4.2.9 Widget de estado emocional

Desde la pantalla de bloqueo, el usuario puede seleccionar un emoticono para mostrar cuál es su estado emocional. La próxima vez que el usuario entre en la aplicación, todas las interacciones con el Widget quedarán registradas.

Cuando el usuario selecciona una opción, se le muestra un mensaje. No se deja al usuario meter continuos estado de ánimo para no saturar la base de datos.



Capítulo 5 - Conclusiones y trabajo futuro

5.1 Conclusiones

El código se encuentra en el repositorio del alumno [13]. Viendo el listado de ideas, se puede observar cómo el trabajo se ha centrado más en la investigación y en crear una demo, que en crear una aplicación para subir a la App Store, o por lo menos esté preparada para ello.

El código QR que aparece en las cajas de medicamentos sí que aporta información a los usuarios aunque estos no puedan obtenerla por sí solos. Esta información de la trazabilidad de sus medicamentos asegura que éstos no han sido manipulados por otros y que se conoce su procedencia. En mi opinión, creo que esta información es importante, pero puede utilizarse esta tecnología para ofrecer un mejor servicio a los clientes finales. En este trabajo se ha demostrado que es una labor sencilla aportar un código QR con la información del medicamento para los clientes finales.

Me gustaría destacar también, los cambios que han surgido en la planificación de la aplicación por no disponer de cuenta de desarrollador. Las distintas funcionalidades descritas en todo el trabajo son perfectamente realizables con los medios oportunos, es por esto que como trabajo futuro se proponen las siguientes ideas:

- Integración con *Siri*.

- Notificación a familiares.

5.2 Trabajos futuros

Además de las futuras funcionalidades ya mencionadas en el apartado anterior, la aplicación podría avanzar con distintas funcionalidades:

- Integración para invidentes.
- Recomendación de genéricos.
- Contador de cantidades.

- Alarma para tomar medicamentos.
- Desarrollo de una aplicación de escritorio para la redacción de noticias que se almacenen en la consola de Firebase, para mostrar las noticias en la aplicación.
- Agregador de noticias en la sección de noticias de la aplicación mediante el parseo de RSS.
- Alerta de producto que contenga alérgenos.
- Desarrollo de la aplicación para dispositivos Android.
- Desarrollo de un chat para resolver la interacción entre paciente y especialista.

Chapter - Introduction

The code is in the student repository [13]. The aim of this work is to bring together in a mobile application a series of functionalities that allow a user or patient to gather as much information as possible about the pharmaceutical sector and have access to that information quickly and efficiently.

In these times in which we live, surrounded by digital devices, it is striking that products that have as much use as medicines, as soon as they have a mobile application or a web page in which they can make queries and store this information in a simple way. It is also noticeable that our location, our physical activity, our movements are constantly monitored, but no service has any concerns about our emotional state.

Finally, it is striking that an industry with so many resources, such as pharmaceuticals, does not have to worry about digitizing the information of its products or the schedules of pharmacies. It is for this reason that we think it is a good idea to serve as an entry point to the digitization of this industry.

Motivation

The main idea behind this project is to use current mobile technology to make tasks related to the pharmaceutical sector easier for the user. These tasks, which are intended to be improved, currently consist of searching for information related to medicines in different Internet search engines. The aim of this application is to be able to carry out a series of information tasks quickly and easily.

Objetives

Develop a mobile application with the following requirements:

1. A user can register in the application.
2. A user can login to the application. From this moment he is considered "The User".
3. The user can scan a QR code and get information from a medication.

a. The QR code must correspond to the QR code on the medicine box.
b. If this cannot be done, a QR code will be created to fulfil the function.

4. The user can add that medicine to their medicine cabinet. (The medicine cabinet is just your list of medicines.
5. The user can consult his medicine cabinet at any time as long as he has Internet connectivity.
6. The user can keep track of his mood.
7. The user can find open pharmacies.
8. The user can know the way to the pharmacy of his choice.

To know what data a QR code from a medicine box provides and to know the method of encrypting that data.

Work plan

This describes the work plan to be followed in order to achieve the objectives described in the previous section.

The first task to be carried out will be to carry out research into medicines. What a user needs to know about medicines, how a user carries out the research task about a medicine, what problems the user has with the leaflets, what the QR code of the medicine boxes is used for, provide useful information for the user. These are some of the questions that need to be answered to learn more about pharmacology.

After carrying out the research task, an analysis of the tools must be carried out. The purpose of this analysis task is to find a computer solution, in this case in the form of a mobile application, in order to provide the user with a tool with which to carry out a set of actions, described in the objectives section. In order to reach this solution, a debate must be held with the director of the FMW.

Once the analysis task is completed, the application development task will proceed. For this task, iterable deliveries are made as a form of self-evaluation. These deliveries will consist of a review of the current state of the application.

Finally, there will be a delivery of both the code developed for the application and this report.

Since the research task was not completed on time, a change in the work plan has arisen that has caused the different deliveries to suffer unforeseen changes between them. This meant that the QR codes on the medicine boxes could not be read with the application. In its defect, the application reads QR codes previously designed for the application, and after reading them, it shows the user the data associated with that medicine.

Description of the report

The rest of the memory is structured as follows. Chapter 2 explains how the research process has been, including the search for information related to the information presented by the medicines, what type of information the QR codes on the medicines boxes show and also what type of functionalities could be implemented in the application and, on the other hand, what not.

Later, in chapter 3, the technological solution on which the application has been based is explained. This chapter includes everything from the tools that have been used to the design pattern.

On the other hand, in chapter 4, there is the structure side of the application. This structure is divided into subsections among which are the external libraries used, and the screens that make up the application.

Finally, in chapter 5, you can find the conclusions of the work, as well as the future work that can be implemented in the application.

Chapter - Conclusions and future work

Conclusions

Looking at the list of ideas, you can see how the work has focused more on research and creating a demo, than on creating an application to upload to the App Store, or at least be prepared for it.

The QR code that appears in the medicine boxes does provide information to users, even if they cannot obtain it on their own. This information on the traceability of your medicines ensures that they have not been manipulated by others and that their provenance is known. In my opinion, I think this information is important, but this technology can be used to provide a better service to end customers. This work has shown that it is a simple task to provide a QR code with the medicine information for end customers.

I would also like to highlight the changes that have arisen in the planning of the application because I do not have a developer account. The different functionalities described in all the work are perfectly achievable with the appropriate means, which is why as future work we propose the following ideas:

- Siri integration.

- Notification to relatives.

Future work

In addition to the future functionalities already mentioned in the previous section, the application could move forward with different functionalities:

- Integration for the blind.

- Generic recommendation.

- Quantity counter.

- Alarm to take medicines.

- Development of a desktop application for writing news to be stored in the Firebase console, to show the news in the application.

- News aggregator in the news section of the application using RSS parsing.

- Product alert containing allergens.

- Development of the application for Android devices.

- Development of a chat to resolve the interaction between patient and specialist.

BIBLIOGRAFÍA

[1] Página web con información de contacto de la AEMPS.

https://www.aemps.gob.es/informa/info-atencion-ciudadano/home.htm#atencion_Internet

[2] Documento que estipula la entrada en vigor el nuevo reglamento de la UE con respecto a la lucha contra la falsificación de los medicamentos. https://www.boe.es/doue/2016/032/L00001-00000.pdf

[3] Trazabilidad de los medicamentos con códigos QR.

https://www.aemps.gob.es/informa/info-atencion-ciudadano/home.htm#atencion_Internet

[4] Enlace para notificar contraindicaciones en los alimentos.

https://www.notificaram.es/Pages/CCAA.aspx#no-back-button

[5] Arquitectura VIPER

https://apiumhub.com/es/tech-blog-barcelona/arquitectura-viper/

[6] Carthage como gestor de dependencias.

https://github.com/Carthage/Carthage

[7] CocoaPods como gestor de dependencias.

https://cocoapods.org/

[8] ¿Qué es Firebase?

https://es.wikipedia.org/wiki/Firebase

[9] Cómo agregar Firebase a una aplicación móvil

https://firebase.google.com/docs/ios/setup

[10] SwiftLint, la herramienta para programar con buenos hábitos.

https://github.com/realm/SwiftLint

[11] Charts, la librería para realizar gráficas.

https://github.com/danielgindi/Charts

[12] Fuente de iconos de la aplicación.

https://appicon.co/

[13] Crear una aplicación con lector de códigos QR

https://www.appcoda.com/barcode-reader-swift/

[13] Enlace al código de la aplicación en Github.

https://github.com/andragui8/MedicControl