



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



TFG del Grado en Ingeniería  
Informática

Frontend aplicación móvil ayuda  
Esclerosis Múltiple



Presentado por Javier Martín Castro  
en Universidad de Burgos — 8 de marzo de 2024  
Tutor: Pedro Renedo Fernández







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Javier Martín Castro, con DNI 24392279C, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 8 de marzo de 2024

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Pedro Renedo Fernández

D.





## Resumen

Dar solución a personas con **Esclerósis Múltiple** creando la interfaz visual.

## Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto  
Ej: aplicación móvil, android, ayuda, Esclerósis múltiple . . .

## **Abstract**

A **brief** presentation of the topic addressed in the project.

## **Keywords**

keywords separated by commas.



---

# Índice general

---

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
1. Introducción	1
2. Objetivos del proyecto	3
3. Conceptos teóricos	5
3.1. Secciones . . . . .	5
3.2. Referencias . . . . .	5
3.3. Imágenes . . . . .	5
3.4. Listas de items . . . . .	6
3.5. Tablas . . . . .	6
4. Técnicas y herramientas	9
4.1. Metodologías . . . . .	9
4.2. Patrón de diseño . . . . .	9
4.3. Repositorio . . . . .	10
4.4. Control de versiones . . . . .	10
4.5. Gestión del proyecto . . . . .	10
4.6. Entorno de desarrollo integrado (IDE) . . . . .	11
4.7. Integración y entregas continuas (CI/CD) . . . . .	11
4.8. Herramientas construcción del proyecto . . . . .	12
5. Aspectos relevantes del desarrollo del proyecto	13
5.1. Inicio del proyecto . . . . .	13
5.2. Metodologías . . . . .	13
5.3. Formación . . . . .	14
5.4. Desarrollo . . . . .	14
5.5. Publicación . . . . .	14
6. Trabajos relacionados	15

<b>7. Conclusiones y Líneas de trabajo futuras</b>	<b>17</b>
<b>Bibliografía</b>	<b>19</b>

---

# Índice de figuras

---

3.1. Autómata para una expresión vacía . . . . .	6
4.1. Arquitectura limpia . . . . .	10

---

# Índice de tablas

---

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto . . . . .	7
---	---

---

# 1. Introducción

---

El proyecto nace para dar solución y mejorar la calidad de vida de personas que tengan **Esclerosis Múltiple** [15, 11] (**EM**). Para ello se va a desarrollar el Frontend inicialmente para dispositivo móvil Android.

En un primer momento aunque hemos tenido antecedentes familiares con este tipo de enfermedad buscamos más información, resumiendo podemos decir que la EM es una enfermedad neurológica crónica cuya naturaleza es inflamatoria y autoinmunitaria. Se caracteriza por la afectar el cerebro y la médula espinal y hace más lentos o bloquea la conexión entre el cerebro y el cuerpo. Algunos síntomas de EM pueden ser:

- Alteraciones en la visión.
- Debilidad a nivel muscular.
- Problemas con la coordinación y el equilibrio.
- Aparición de lesiones cerebrales detectadas en la resonancia magnética.

Podemos clasificar esta enfermedad entre:

- **Recurrente-remitente (EMRR)** caracterizada por aparición periódica de ataques o brotes.
- **Secundaria progresiva (EMSP)** se caracteriza por el empeoramiento gradual y progresiva de la discapacidad.
- **Primaria progresiva (EMPP)** este tipo comienza de manera inofensiva y empeora con el tiempo.
- **Progresiva-recurrente (EMPR)** este es el más agresivo ya que las apariciones de los síntomas son severas y periódicas.

Para la creación de la aplicación se tendrá en cuenta aquellos aspectos que pueden aportar mejoras en la calidad de vida del usuario como puede ser:

- Recordatorios de la medicación o el tratamiento
- Recordatorios de las citas



---

## 2. Objetivos del proyecto

---

Crear una aplicación móvil para ayudar a personas con Esclerosis Múltiple, donde se realizarán las siguientes acciones:

- Crear pantalla para acceder a la aplicación el usuario.
- Crear pantalla para realizar el registro de un nuevo usuario.
- Crear una pantalla del detalle del usuario.
- Crear pantalla home

En el transcurso de la realización del proyecto se pretende aportar:

- Dar soporte a personas con EM.
- Facilitar el día a día de los usuarios que usen la aplicación.





---

## 3. Conceptos teóricos

---

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de L<sup>A</sup>T<sub>E</sub>X <sup>1</sup>.

### 3.1. Secciones

Las secciones se incluyen con el comando `section`.

#### Subsecciones

Además de secciones tenemos subsecciones.

#### Subsubsecciones

Y subsecciones.

### 3.2. Referencias

Las referencias se incluyen en el texto usando `cite` [14]. Para citar webs, artículos o libros [10], si se desean citar más de uno en el mismo lugar [4, 10].

### 3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de L<sup>A</sup>T<sub>E</sub>X, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:

---

<sup>1</sup>Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz



Figura 3.1: Autómata para una expresión vacía

### 3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

**Primer item** más información sobre el primer item.

**Segundo item** más información sobre el segundo item.

▪

### 3.5. Tablas

Igualmente se pueden usar los comandos específicos de  $\text{\LaTeX}$  o bien usar alguno de los comandos de la plantilla.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto



---

## 4. Técnicas y herramientas

---

### 4.1. Metodologías

#### Scrum

Scrum [19] es un marco que se emplea para la gestión de proyectos de metodología ágil y que facilita a los equipos la tarea de gestionar y estructurar el trabajo. Para ello, a través de iteraciones (sprints) se realiza una serie de tareas en un periodo de tiempo y se van incorporando al software.

#### Gitflow

Gitflow [7] se define como un flujo de trabajo para la creación de ramas en Git y llevar un control de versiones. En este flujo existen dos ramas principales, master y develop, y otras ramas que pueden ayudarnos en nuestro desarrollo como pueden ser: feature, release o hotfix.

### 4.2. Patrón de diseño

#### Model-View-ViewModel MVVM

Es un patrón MVVM [9], también conocido como Model View ViewModel que se centra en separar la interfaz del usuario (View) de la parte lógica (Model). La interacción entre la parte lógica y la interfaz del usuario a través del recurso ViewModel.

Algunas de las ventajas al usar este patrón son:

- Fácil desarrollo ya que al poder separar la vista de la lógica varios equipos pueden trabajar simultáneamente en varios componentes.
- Fácil testeo ya que no es necesario utilizar la vista para crear tests para el model o el viewmodel.
- Fácil mantenimiento ya que al realizar la separación de los componentes se crea un código simple y limpio.

Vamos a describir a continuación cada uno de los componentes que forman el MVVM:

- **Model:** es el componente donde se encapsulan los datos de nuestra aplicación. En ella se pueden encontrar validación y lógica de negocio.
- **View:** nos muestra el diseño y la apariencia de nuestra aplicación. En ella se verán los datos pero sin contener nada de la lógica de negocio.
- **ViewModel:** es el componente que enlaza los datos o cambios de estado que puede tener nuestra aplicación.

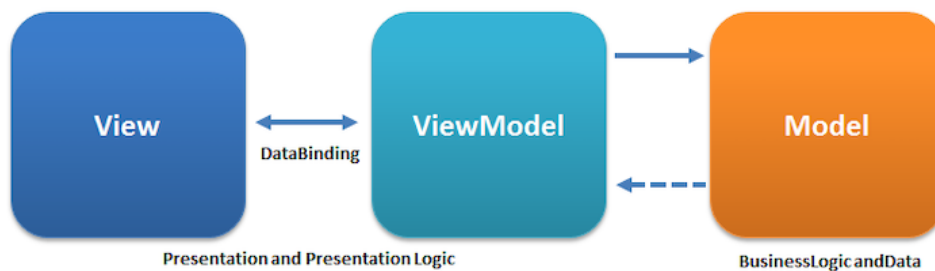


Figura 4.1: Arquitectura limpia

### 4.3. Repositorio

Entre las herramientas consideradas GitHub [2], Bitbucket [3], GitLab [18] se decide utilizar GitHub, porque nos permite alojar proyectos gratuitamente. Además podemos crear documentación a través de wikis, crear tareas, sprints. Al crear el repositorio en GitHub podemos usar el gestor de proyecto que tiene a través de sus paneles y saber el estado en el que está cada issue (tarea). Otra de las ventajas que encontramos con GitHub es que está integrado en múltiples servicios de integración continua.

### 4.4. Control de versiones

Al haber escogido como repositorio GitHub para alojar nuestro proyecto, utilizamos como control de versiones Git [17] que es un sistema distribuido que nos permite realizar nuestro trabajo sin conexión, es ligero y rápido. El crear ramas y hacer los merges es muy rápido y pocas veces se obtienen conflictos. El historial es muy detallado.

### 4.5. Gestión del proyecto

Como gestor de proyectos utilizamos el que viene integrado en GitHub. Con esta herramienta podemos gestionar nuestro proyecto y ver en que estado está cada tarea, incluso podemos darle prioridad.

## 4.6. Entorno de desarrollo integrado (IDE)

### Flutter

Flutter [1] es un SDK (Kit de Desarrollo de Software) desarrollado por Google que nos permite crear aplicaciones para varios dispositivos. Hemos elegido esta herramienta porque nos permite crear aplicaciones multiplataforma. Entre alguna de las ventajas destacamos:

- **Compila en nativo** tanto para Android como iOS.
- Mediante el uso de widgets, se pueden **crear interfaces gráficas flexibles**.
- **Rapidez** en el desarrollo.

### Dart

Dart [8] es un **lenguaje de programación moderno**, desarrollado por Google, donde se mezcla la programación orientada a objetos POO con los lenguajes de programación basados en scripts.

### FlutterFlow

FlutterFlow [5] es una herramienta que nos permite crear aplicaciones insertando los componentes que tiene la herramienta, arrastrando y soltando. Es posible añadir lógica durante la creación de las páginas.

### LaTeX

Entre las distintas herramientas de  $\text{\LaTeX}$  que existen hemos decidido utilizar TeXmaker [20, 12] que es un editor de  $\text{\LaTeX}$  gratuito y multiplataforma. Además, incluye soporte unicode, corrección ortográfica, autocompletado y un visor de PDF integrado.

### Documentación

Para la documentación del trabajo de fin de grado hemos empleado  $\text{\LaTeX}$  [14] que es un sistema que permite componer textos con una alta calidad tipográfica. En el caso de la documentación del repositorio hemos empleado Markdown [16] que es un lenguaje de marcado ligero.

## 4.7. Integración y entregas continuas (CI/CD)

### Codemagic

Codemagic [6] es producto basado en la nube que permite realizar integración y entregas continuas. Te permite crear workflow versátiles y activar la compilación, análisis de código y pruebas de manera automática.

## **4.8. Herramientas construcción del proyecto**

### **Librerías**



---

## 5. Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más relevantes del desarrollo del proyecto, pasando por la creación del proyecto, que decisiones se tomaron y la resolución de problemas encontrados y como se solucionaron.

### 5.1. Inicio del proyecto

El proyecto surgió a partir de la necesidad de ayudar a personas diagnosticadas con EM (Esclerosis Múltiple) y poder hacer su día a día más fácil a través de consejos, atención, ayuda.

Para poder hacer posible esta solución, se crea una aplicación inicialmente para el móvil, donde se incluirán secciones que pueden ayudar a los usuarios que hagan uso de ellas. Tras la asignación del proyecto se inicia un estudio de las posibilidades tecnológicas disponibles para resolver esta necesidad.

Desde el proyecto realizaremos llamadas a una API del cliente que nos proveerá de los datos necesarios para crear una aplicación amigable para los usuarios.

### 5.2. Metodologías

Desde el comienzo se realiza una arquitectura limpia para reutilizar los componentes y que sean independientes sin que haya ningún impacto si se realizan cambios en alguno de ellos. Para la gestión del proyecto, se utiliza el panel que nos proporciona GitHub, usando la metodología ágil Scrum. El equipo al ser un proyecto creado para la realización del trabajo de fin de carrera, está formado por una persona. Por otro lado, al tener que hacer uso de una API externa, realizada por un compañero de la universidad, se mantienen reuniones para recabar requisitos técnicos para desarrollar la aplicación.

Para el desarrollo de la metodología ágil se siguieron los siguientes puntos:

- crear iteraciones incrementales (sprints) y revisiones
- una vez terminado el sprint se añade a la rama main que será nuestro producto final.

- los sprints por regla general los hemos hecho de duración semanal.,
- para cada sprint se le asocian las tareas más importantes.

### 5.3. Formación

Para abordar la realización y creación del proyecto se necesitaban adquirir una serie de conocimientos que inicialmente eran inexistentes. Entre los conocimientos adquiridos se destacan:

- Formación sobre Flutter [1, 13].
- Formación sobre Dart [8].

### 5.4. Desarrollo

Durante el desarrollo de la aplicación nos aparecieron diversos retos:

- Conexión entre los diferentes componentes.
- Internacionalización de la aplicación.
- Contactar con el cliente ante desconexión de la dirección de la API facilitada inicialmente.
- Contactar con el cliente para resolución de dudas sobre la API.

### 5.5. Publicación

Una vez hubo una versión lista para desplegar en producción, se creó una cuenta en Google Play y se desplegó.

Para próximas versiones, se tratará de crear un despliegue para otros dispositivos, como es iOS.

---

## **6. Trabajos relacionados**

---

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.



---

## **7. Conclusiones y Líneas de trabajo futuras**

---

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.



---

# Bibliografía

---

- [1] aures tic. ¿qué es flutter? <https://aurestic.es/que-es-flutter/>.
- [2] Gustavo B. Qué es github y cómo empezar a usarlo. <https://www.hostinger.es/tutoriales/que-es-github>, 2023.
- [3] Atlassian Bitbucket. Breve presentación de bitbucket. <https://bitbucket.org/product/es/guides/getting-started/overview#a-brief-overview-of-bitbucket>, 2024.
- [4] Zachary J Bortolot and Randolph H Wynne. Estimating forest biomass using small footprint lidar data: An individual tree-based approach that incorporates training data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(6):342–360, 2005.
- [5] Mateus Castelo. Flutterflow | la revolución en la creación de apps nocode. <https://nocodestartup.io/es/flutterflow/#:~:text=Conclusi%C3%B3n%20sobre%20FlutterFlow-,%C2%BFQu%C3%A9%20es%20FlutterFlow%3F,e%20inserta%20l%C3%B3gica%20en%20ella>.
- [6] Codemagic. Welcome to codemagic. <https://docs.codemagic.io/getting-started/about-codemagic/>.
- [7] Aula de Software Libre. Flujo de trabajo con git (git flow). <https://aulasoftwarelibre.github.io/taller-de-git/gitflow/>.
- [8] Juan Gabriel Gomila. ¿qué es dart? historia, características y ventajas de aprenderlo. <https://cursos.frogamesformacion.com/pages/blog/que-es-dart>, 2023.
- [9] Flutter Guru. Flutter mvvm architecture. <https://blog.devgenius.io/flutter-mvvm-architecture-with-provider-a81164ef6da6>, 2022.
- [10] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [11] MedlinePlus. <https://medlineplus.gov/spanish/multiplesclerosis.html>.
- [12] Texmaker. <https://www.xmlmath.net/texmaker/>.

- [13] Udemy. Flutter desde cero - guía completa con arquitectura limpia. <https://www.udemy.com/course/flutter-desde-cero-darwin-morocho/?couponCode=2021PM20>.
- [14] Wikipedia. Latex — wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=LaTeX&oldid=84209252>, 2015. [Internet; descargado 30-septiembre-2015].
- [15] Wikipedia. [https://es.wikipedia.org/wiki/Esclerosis\\_m%C3%BAltiple](https://es.wikipedia.org/wiki/Esclerosis_m%C3%BAltiple), 2024.
- [16] Wikipedia. <https://es.wikipedia.org/wiki/Markdown>, 2024.
- [17] Wikipedia. Git. <https://es.wikipedia.org/wiki/Git>, 2024.
- [18] Wikipedia. Gitlab. <https://es.wikipedia.org/wiki/GitLab>, 2024.
- [19] Wikipedia. Scrum (desarrollo de software). [https://es.wikipedia.org/wiki/Scrum\\_\(desarrollo\\_de\\_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)), 2024.
- [20] Wikipedia. Texmaker. <https://es.wikipedia.org/wiki/Texmaker>, 2024.