

## **ANEXO V: DOCUMENTACIÓN TÉCNICA**

# *Habitr - Aplicación para fomentar los hábitos saludables.*

Trabajo de Fin de Grado

**Ingeniería Informática**



# **VNiVERSIDAD D SALAMANCA**

**Junio 2023**

**Autor:**

*Jorge Cruz García*

**Tutores:**

*André Filipe Sales Mendes*

*Gabriel Villarubia González*

# 1 TABLA DE CONTENIDO

---

Índice de figuras .....	3
Índice de tablas.....	4
2 Especificación de la documentación.....	5
3 Manual del programador .....	6
4 Referencias.....	7

## ÍNDICE DE FIGURAS

---

Figura 1: Página principal de la documentación.....	5
Figura 2: Ejemplo de documentación para la clase FriendsRepository .....	6
Figura 3: Menú Flutter .....	7

## **ÍNDICE DE TABLAS**

---

## 2 ESPECIFICACIÓN DE LA DOCUMENTACIÓN

Para generar la documentación, se ha utilizado la herramienta que incluye Dart en su kit de desarrollo, llamada *dartdoc*.

Dartdoc permite generar documentación de archivos de Dart a través de un sistema de anotaciones, que permite separar los ficheros por categorías, comentar métodos y clases.

Para acceder a la documentación, se encuentra en la carpeta “doc/api” que se incluye con el código fuente. Al acceder al archivo *index.html*, se tendrá una vista global de la aplicación, que se puede visualizar a través de un navegador.

Se han separado las clases en los mismos paquetes que se establecieron en el *Anexo IV*, es decir, tanto por su funcionalidad (gestión social, gestión de usuarios...) cómo por su lugar en la arquitectura (vista, bloc...)

Para realizar una mejor lectura de la documentación, se hacen las siguientes recomendaciones:

- Seguir una vista arquitectónica para evitar mezclar documentación de vistas con controladores. Es decir, visualizar la documentación de los BLoC, repositorios y datos en vez de los paquetes de “GestiónX”
- Cada vez que se quiere leer la documentación de una clase, hay que hacer click varias veces para llegar a la vista de detalle que incluye la documentación de los métodos.
- Las clases de vista no han sido documentadas en detalle, puesto que sus nombres son explicativos de lo que representan y no tienen lógica compleja ni métodos.

En la *figura 1* se muestra una vista previa de cómo es la página principal de la documentación (*index.html*).

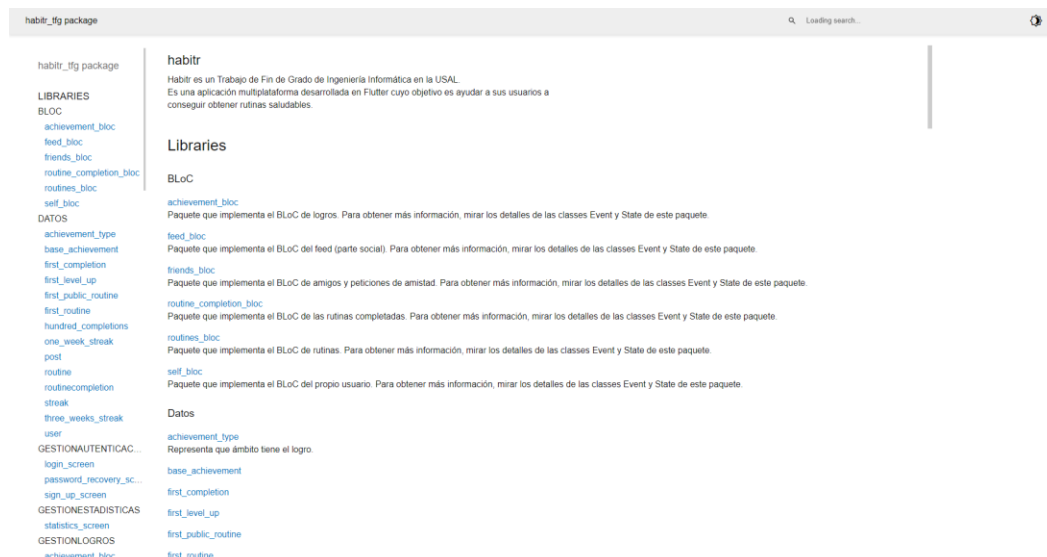


Figura 1: Página principal de la documentación

## Anexo V: Documentación Técnica

En la *figura 2*, se puede ver la documentación de la clase `FriendsRepository`, mostrándose la documentación de los métodos y sus parámetros.

The screenshot shows the Dart documentation for the `FriendsRepository` class. The page is titled "FriendsRepository class" and includes a search bar at the top right. On the left, there is a navigation menu with "friends\_repository library" and "CLASSES FriendsRepository". The main content area is divided into sections: "Constructors" (listing `FriendsRepository()`), "Properties" (listing `hashCode` and `runtimeType`), and "Methods" (listing `getFriendsAndRequests()`, `noSuchMethod()`, `replyToFriendRequest()`, and `sendFriendRequest()`). A right-hand sidebar contains a table of contents with sections for "CONSTRUCTORS", "PROPERTIES", "METHODS", and "OPERATORS". The footer of the page indicates the version "habitr\_tfg 1.0.0+1".

Figura 2: Ejemplo de documentación para la clase `FriendsRepository`

### 3 MANUAL DEL PROGRAMADOR

En este apartado, se explicará cómo realizar la compilación del proyecto para poder generar los ficheros ejecutables.

Lo primero de todo, es necesario tener instalado el SDK de Dart y de Flutter, junto a Unity. Ambos tutoriales de instalación se encuentran en la sección de referencias de este documento.

Una vez tenemos Flutter instalado, hemos de abrir una consola en la carpeta raíz del proyecto, y usar el comando `flutter pub get` para instalar las bibliotecas que se usaron en el desarrollo de la aplicación.

Cuando ya se tienen los pre-requisitos instalados, se puede proceder a la compilación en sí:

- Se abre la escena que se encuentra en la carpeta `unity/Habitr` en Unity, y se comprueba que ha cargado la extensión de Flutter. Debería de aparecer el menú que se ve en la *figura 3* en la parte superior.

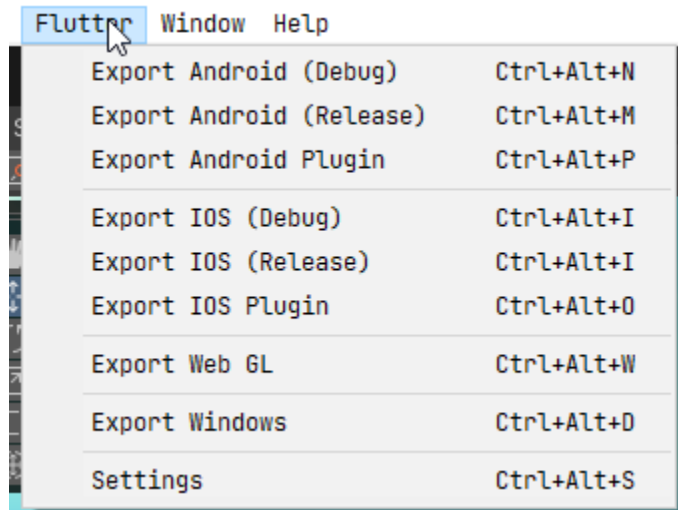


Figura 3: Menú Flutter

- Se selecciona la opción que se quiera compilar; por ejemplo, Export Android (Release). Una vez se haga esto, la extensión se encargará de compilar los archivos necesarios y de modificar los archivos de *Gradle* que definen la compilación del proyecto de Flutter
- Se abre una consola en la carpeta raíz del proyecto y se usa el comando *flutter build apk* para obtener un fichero binario de Android.
- Para construir una aplicación de iOS, es necesario realizar este mismo proceso desde una máquina Macintosh, usando “Export IOS” en Unity y el comando *flutter build ipa*.

## 4 REFERENCIAS

---

- dartdoc (accedido el 2-7-2023)
  - <https://pub.dev/packages/dartdoc>
- Guía de instalación de Flutter (accedido el 2-7-2023)
  - <https://docs.flutter.dev/get-started/install/windows>
- Guía de instalación de Unity (accedido el 2-7-2023)
  - <https://learn.unity.com/tutorial/install-the-unity-hub-and-editor>