

Aplicación multiplataforma Flutter para contratar servicios y adaptarlos a TFGs

Francisco Chillida Bernaudo

Resumen— Este trabajo final de grado se ha centrado en el desarrollo de una aplicación multiplataforma realizada con Flutter. La aplicación tiene como objetivo ayudar a estudiantes a encontrar proyectos para realizarlos como trabajos finales de grado, y a raíz de eso ayudar a una persona o entidad publicadora de la propuesta, la cual no tenga los medios suficientes para poder contratar a un profesional, pueda realizar su proyecto. La aplicación y el trabajo en global se ha desarrollado siguiendo un método de Scrum adaptado a una persona. Para finalizar todo el proceso se ha realizado una encuesta a una serie de estudiantes para comprobar el alcance que podría tener entre ellos y así estudiar la posibilidad de en un futuro, profundir en el desarrollo de la aplicación.

Palabras clave— Aplicación multiplataforma, Flutter, Dart, Firebase.

Abstract— This final degree project has focused on the development of a multi-platform application made with Flutter. The application goal is to help students find projects to carry out as final degree projects, and as a result of that to help a person or entity that publishes the proposal, which does not have enough requirements to be able to hire a professional, can carry out her project. The global application and work has been developed following a Scrum method adapted to a individual person. To finalize the whole process, a study was carried out on a series of students to verify the scope it could have among them and get the possibility to study how to deepen in the development of the application.

Keywords— Multiplatform application, Flutter, Dart, Firebase.

1 INTRODUCCIÓN

LA idea del siguiente trabajo de final de grado (en adelante, TFG), surgió debido a las dificultades e incertidumbres que padecen algunos estudiantes a la hora de escoger un tema para iniciar su propio TFG. No es fácil moldear una idea creativa, funcional e innovadora en todos los casos, y esto puede generar una frustración al estudiante que le lleve a realizar un estudio que no se adapte a sus preferencias.

Es por eso por lo que, con este TFG se propone realizar una aplicación (App) multiplataforma, para así poder llegar a un máximo número de usuarios, donde podamos encontrar dos tipos de partes interesadas. Por un lado, como se ha mencionado, los estudiantes, dispuestos a encontrar un proyecto que le resulte atractivo para hacer de él su TFG. Por otro lado, un conjunto de entidades como, empresas,

autónomos, universidades, o cualquier persona que quiera gestionar un proyecto, pero por los motivos que sean, no cuentan con los medios necesarios para llevarlo a cabo. Ya sea por motivos económicos, de falta de personal, tiempo, etc. De esta manera, la aplicación podría solventar el problema de todos los interesados, de forma que, conectando a ambas partes, los estudiantes puedan acabar realizando un proyecto para la otra entidad. Esto será siempre de manera no profesional, lo que significa que no habría una remuneración, pero si, una cesión de licencia del proyecto, para que se pueda presentar y defender como Trabajo de Fin de Grado.

La estructura del artículo constará de la explicación de los objetivos del proyecto, su estado del arte, la metodología y planificación que se siguió. También se especificarán los requisitos de la aplicación, que tecnologías se utilizaron, cómo se implementó y la exposición de los resultados finales.

2 OBJETIVOS

El objetivo principal, el cual es solventar el problema mencionado con anterioridad, será atajado creando una aplicación multiplataforma completa y funcional.

- E-mail de contacto: francisco.chillida@e-campus.uab.cat
- Mención realizada: Tecnologías de la Información
- Trabajo tutorizado por: Marta Prim Sabrià (Departament de Microelectrònica i Sistemes Electrònics)
- Curso 2020/21

Para alcanzar este objetivo, en el desarrollo de la aplicación se utilizará Flutter. Flutter es un kit de herramientas de User Interface (a partir de ahora, UI) de Google, para realizar aplicaciones compiladas nativamente, ya sean de móvil, web o escritorio desde una única base de código. Como lenguaje de programación, Flutter utiliza Dart. Originalmente llamado Dash, es un lenguaje de código abierto, también desarrollado por Google. El motivo de utilizar Flutter se basa en poder llegar a un máximo número de personas que se puedan beneficiar de este proyecto, si utilizamos esta tecnología no importará si el usuario dispone de un dispositivo Android, IOS o web, todos podrán acceder de la misma manera a la App.

El entorno de desarrollo integrado (IDE) donde se va a desarrollar el proyecto será Android Studio. Un IDE que da soporte a Flutter mediante la instalación de plug-ins. Este cuenta con un emulador Android, un dispositivo web y la posibilidad de acoplar un simulador IOS.

Por lo tanto, lo que se busca alcanzar al finalizar este TFG es, una aplicación intuitiva, donde un usuario sin experiencia pueda ser capaz de:

- Registrarse
- Iniciar sesión
- Acceder a su perfil
- Encontrar todas las ofertas de proyectos actuales o poder publicarlas
- Poder comunicarse con la otra parte interesada mediante un chat
- Cerrar el ciclo de gestión

3 ESTADO DEL ARTE

En este apartado veremos cuales son las aplicaciones que los estudiantes, en este caso de la UAB, tienen a su disposición para poder escoger un TFG. Por otro lado, también veremos cuales son los frameworks más utilizados para desarrollar aplicaciones multiplataforma, y que ofrecen cada uno de ellos.

3.1 Herramienta de Inscripción al TFG

Al inicio de todo el proceso que conlleva realizar un TFG, el primer paso que un estudiante debe hacer, es escoger un proyecto que se adecúe a la carrera y la mención que está realizando. Este paso lo puede hacer de tres maneras, a través de una empresa en la cual esté trabajando o realizando prácticas, de manera individual, o escogiendo un tema propuesto por un profesor. En caso de elegir la segunda opción, el estudiante puede proponer cualquier tema que se le ocurra, mientras como ya se ha mencionado, cumpla con los requisitos de su carrera y mención. Si decide escoger un tema propuesto por un profesor, el estudiante puede acceder a un repositorio donde los profesores del grado proponen distintos temas a tratar. Esta herramienta la encontramos en la página web <http://sia.uab.cat/> que ofrece servicios de trámites en línea para todos los estudiantes de la universidad. Si se accede al apartado de Inscripción al treball de Fi d'Estudis/Pràctiques, es ahí donde se pueden

encontrar todas las propuestas.

Esta herramienta cuenta con varios apartados, en los cuales los principales son:

- Los títulos de los TFGs
- Una descripción
- El profesor o profesora que publica la propuesta

Si en algún momento algún estudiante está interesado en cualquier trabajo encontrado dentro de la aplicación, este puede contactar con el docente para consultar cualquier duda o reservar la plaza a ese proyecto.

3.2 Frameworks Multiplataforma

Un framework es una estructura de trabajo tecnológica con una asistencia definida, el cual ayuda al programador en el desarrollo del software. Suelen incluir un lenguaje de programación, librerías y soporte del programa entre otras cosas. Los tres frameworks multiplataforma más relevantes hoy en día son los siguientes:

- Flutter: Flutter[2] es un framework multiplataforma que apareció por primera vez en el año 2015. Creado por Google y de código abierto, Flutter utiliza Dart como lenguaje de programación. Este framework nos permite desarrollar aplicaciones para Android, iOS, web y de escritorio, estas últimas están en fase beta actualmente.
- React Native: React Native[10] es un framework cross-platform creado por Facebook en el año 2015, que utiliza JavaScript como lenguaje de programación. Permite a los desarrolladores implementar aplicaciones nativas que soporta los sistemas operativos de Android e iOS.
- Ionic: Ionic[11] fue creado en el año 2013 por Drifty Co, y al igual que las otras dos tecnologías, Ionic, nos permite desarrollar aplicaciones híbridas para iOS, Android i web. Ionic utiliza tecnologías web para el desarrollo de sus aplicaciones, tales como HTML5, CSS y JavaScript.

4 METODOLOGÍA Y PLANIFICACIÓN

La metodología escogida para asegurar un correcto seguimiento del trabajo ha sido Scrum Of One[5]. Esta es una adaptación del conocido modelo Scrum. Se trata de una metodología Agil que mantiene los conceptos y principios básicos que utiliza Scrum pero adapta ellos a un solo individuo.

De la misma manera que en Scrum, hay una división de las tareas a realizar. A estas tareas se le añade un valor según el esfuerzo y el tiempo estimado que se va a necesitar para cada una. Por otro lado, están los Sprints, un período de tiempo de quince días aproximadamente donde se marcan las tareas que se van a realizar. Al final de cada Sprint, se organiza un Sprint retrospective y un Sprint Review. Este proceso se ejecuta iterativamente hasta llegar a la finalización del proyecto, y, por ende, conseguir el producto final.

El motivo que me ha llevado a escoger esta metodología viene dado por ya estar familiarizado con el método tradicional de Scrum, y simplificarlo a un solo individuo se cree que puede ser de gran ayuda para gestionar las tareas de este TFG.

4.1 Sprints

Como se acaba de puntualizar, los Sprints son una de las principales características del método Scrum. La finalidad de los Sprints es poder dividir el proyecto en una carga de trabajo equitativa para cada tramo de tiempo. Al puntuar las tareas con diferentes grados de dificultad, podemos organizar Sprints donde a lo mejor el número de tareas es muy diferente, pero la carga de trabajo muy similar, con lo cual asegurarnos poder completar el proyecto en el plazo de tiempo planificado.

Dentro de esta categoría encontramos el Sprint Retrospective, donde en vez de distribuirnos nuevas tareas, se valora cómo de bien ha ido el Sprint ya finalizado y qué puntos se pueden mejorar. Seguidamente, se genera un Sprint Review, en este caso, se toma en cuenta cuáles serán las tareas para realizar en el siguiente Sprint.

La siguiente planificación viene dividida en nueve puntos totales. Estos son los nueve Sprints que se han realizado durante el desarrollo del TFG. Cada Sprint estará dividido en tareas más sencillas para gestionar de una manera más adecuada el avance de cada sesión.

Se toma como punto de partida la cuarta semana de febrero, desde el 22/02/21 hasta el día de la entrega final de la memoria, el 28/06/21. Al utilizar una metodología ágil, puede llegar a haber cambios en la planificación según se gestionen los Sprints, dándonos cuenta de que algunas tareas necesitan algo menos de elaboración, y otras, por lo contrario, se les ha de aumentar el tiempo dedicado

1. Formación en Flutter, Dart y Firebase (22/02–07/03)

En este apartado hemos dedicado los días en completar una serie de cursos[1] para familiarizarnos con el lenguaje de programación Dart, y aprender a cómo utilizar la herramienta Firebase y las ventajas que nos proporciona.

2. Diseño de las vistas principales (08/03– 21/03)

En estas dos semanas, diseñamos las vistas principales que tienen nuestra aplicación. Se realizaron una serie de mockups, y una vez obtuvimos el diseño completo se realizó el desarrollo de la versión móvil.

3. Diseño y construcción de la base de datos (22/03 – 04/04)

Llegados a este punto, nos dedicamos a diseñar y llevar a cabo la base de datos NoSQL que ofrece Firebase. Además de implementarla en nuestra aplicación.

4. Implementación de la arquitectura backend (05/04 – 18/04)

En estas semanas se programó las funcionalidades backend de la aplicación. Se añadieron al código las llamadas al servidor y se implementaron todos los servicios necesarios para garantizar la correcta funcionalidad de la App.

5. Desarrollo de test y actualizaciones de los apartados anteriores (19/04 – 02/05)

Durante este sprint, se

implementaron test para corroborar el adecuado funcionamiento de los apartados finalizados hasta ahora. Se corrigieron aquellos fallos que tuvo el software y se completaron los procesos que quedaron inacabados.

6. Desarrollo del chat de interacción de los usuarios (03/05 – 16/05)

En este punto, se desarrolló la principal forma de comunicación entre los usuarios, el chat interno de la aplicación.

7. Gestión de la seguridad de la aplicación (17/05 – 30/05)

Este sprint fue destinado a garantizar y gestionar la robustez del software agregando las pautas de seguridad también mencionadas en los objetivos.

8. Completar una versión estable de la App y realización de encuestas (31/05 – 13/06)

Quedando unas pocas semanas antes de la entrega final, se revisó un adecuado funcionamiento de toda la aplicación. Certificando la total funcionalidad de la aplicación para la cual se ha planteado y finalizando de esta manera la mayoría de los objetivos planteados.

9. Finalización de la memoria final (14/06 – 28/06)

Durante el último sprint se realizó una encuesta para recopilar información de otros estudiantes acerca de este TFG y su aplicación. Además se completó la memoria escrita para su entrega final.

A lo largo de todos los Sprints, se ha ido documentando en formato digital todo el proceso desarrollado. Describiendo el avance de la aplicación y las funcionalidades acabadas.

5 REQUISITOS

En este apartado se describirán los diferentes actores que utilizará la aplicación, y cuáles son los requisitos que debe cumplir. Tanto los funcionales como los no funcionales.

5.1 Actores

Usuario no Autenticado Figura 13. Es un actor que corresponde a aquel usuario que todavía no se ha identificado en la aplicación.

Usuario Estudiante Figura 14. Este actor corresponde a un usuario ya identificado en la aplicación y que está cursando algún estudio universitario. Este actor no tiene los permisos para publicar un proyecto, pero si para verlos y guardar como favoritos aquellos que él quiera destacar.

Usuario no Estudiante Figura 15. Es un actor el cual corresponde a un usuario ya identificado en la aplicación, pero no se ha registrado como estudiante. El actor tendrá permisos para ver y publicar proyectos. Se quiere aclarar que a pesar de que el usuario se ha identificado como no estudiante, en la realidad este usuario si puede ser un estudiante, pero a efectos prácticos de la aplicación, se considera no estudiante todo aquel que se registre para realizar publicaciones de proyectos y no en busca de ellos.

5.2 Requisitos Funcionales

Requisitos de Usuario no Autentificado

Como usuario no autentificado se va a poder registrar en la aplicación mediante un usuario y contraseña o accediendo a partir de cuentas verificadas de Google, Facebook o Twitter.

Si el usuario ya se había registrado en la aplicación podrá acceder a ella verificando sus credenciales.

Requisitos de cualquier Usuario Autentificado

Como usuario autentificado se podrá tener acceso a visualizar todos los proyectos que se hayan publicado por parte del resto de usuarios. El usuario estudiante podrá ver una detallada información de cada proyecto, el título, una descripción, el nombre de la persona que lo publica y un listado de tags de filtrado.

El usuario ha de poder acceder a la información de su perfil. De la misma manera, si se ha de modificar algún dato, el usuario ha de poder hacer las modificaciones pertinentes sin dificultad y poder guardar la información enviada.

Se ha de tener acceso a todos los chats iniciados con otros usuarios. Así mismo poder interactuar mediante este de manera instantánea. Es el usuario estudiante el que ha de poder iniciar una comunicación vía chat, pero ha de ser cualquier usuario el que pueda eliminar la comunicación en el momento que desee.

Requisitos de Usuario Estudiante

El usuario estudiante ha de poder marcar aquellos proyectos por los cuales tiene interés como favoritos. Una vez marcados, podrá acceder a ellos a partir de la herramienta “Favoritos”. Si en algún momento algún proyecto deja de ser de su interés deberá poder borrarlo de esta lista.

Requisitos de Usuario no Estudiante Publicaciones

El usuario no estudiante ha de poder publicar tantos proyectos como él desee. Para ello puede usar el apartado “Proyectos”. El usuario tendrá que indicar el título de su propuesta, una descripción detallada sobre las tareas a realizar y si es necesario, añadirle los tags necesarios para poder facilitar la búsqueda de su proyecto en el área Home.

5.3 Requisitos No Funcionales

5.3.1 Eficiencia

El sistema ha de ser capaz de poder realizar las transacciones en pocos segundos. En nuestro caso tenemos incorporado un chat en línea que ha de poder enviar y recibir los mensajes instantáneamente, sin que el usuario pueda apreciar ningún retraso. Así mismo las publicaciones de proyectos realizadas tienen que efectuarse de manera inmediata una vez el usuario haya procedido.

5.3.2 Usabilidad

A pesar de que la aplicación está destinada a usuarios de una edad ya adulta, esta debe tener un sistema de acciones simples para poder garantizar el correcto uso y aprendizaje del

sistema en pocas horas. Se quiere minimizar al máximo la tasa de errores cometidos por los usuarios como por ejemplo podría ser una publicación errónea o no encontrar alguna funcionalidad que ofrece la App.

5.3.3 Seguridad

El sistema debe garantizar la privacidad de todos los usuarios, tanto la visualización de sus datos como el acceso a ellos. Para ello la información más sensible de cada usuario será encriptada con la ayuda de Encrypt5.0, un paquete que Flutter nos ofrece. Únicamente podrán acceder a la App aquellas personas que dispongan de un usuario y contraseña por haberse registrado con anterioridad.

5.3.4 Apariencia

La apariencia de la aplicación ha de reflejar un estado sencillo y minimalista el cual sea agradable a la vista de todos los usuarios. No ha de estar cargada de texto o imágenes y tendrá que cumplir el estándar Material Design, un estándar creado por Google que Flutter implementa en el código a través de un widget.

5.3.5 Hardware

Será necesario la propiedad de un dispositivo móvil cuyo sistema operativo sea Android o iOS. Así mismo también servirá un sistema con el navegador web Google Chrome.

5.3.6 Servidor

Aun que no sea necesario tener posesión de unos servidores físicos para alojar todos los datos de los usuarios, si se necesitará el acceso a un servidor en la nube lo suficientemente grande como para almacenar toda esa información.

6 TECNOLOGÍAS UTILIZADAS

A continuación, describiremos todas aquellas tecnologías o herramientas que hemos utilizado para poder llevar a cabo el proyecto.

6.1 Flutter

Flutter es un software de código abierto creado por Google. Es utilizado para crear aplicaciones nativas móviles, ya sean Android o iOS, web y de escritorio, tanto en Windows como MacOS o Linux. La primera versión que salió a la luz de Flutter fue en 2015, desde entonces Google ha ido trabajando en el software hasta llegar a donde está hoy en día, donde Flutter ha lanzado su versión 2.0 la cual destaca por pasar su versión web del estado beta a su canal estable. El gran atributo del cual dispone Flutter es la ventaja de poder crear aplicaciones nativas para diferentes plataformas con un único código nativo, o apenas añadiendo algunos detalles.

La forma en la que Flutter crea estas aplicaciones es mediante widgets, creados utilizando un framework inspirado en React. La idea principal es que desde la función main() de la aplicación, la primera que se ejecuta, se llame a un primer widget y dentro de este se vayan construyendo otros

sucesivamente, de esta forma acabar creando un árbol de widgets que forman toda la App.

6.2 Dart

El motivo por el cual se ha escogido Dart[7] como lenguaje de programación en este proyecto es debido a que es el lenguaje por defecto que Flutter utiliza para crear las aplicaciones. Este lenguaje fue creado al igual que Flutter por Google y de código abierto.

Con una sintaxis muy similar a la del lenguaje de programación C, Dart, está orientado a objetos. Esto nos facilita la creación de clases y su uso dentro del código. Una particularidad que Dart proporciona es la recarga asíncrona de una única parte de la interfaz de usuario que se esté programando, eso permite al programador ver instantáneamente los cambios que se están ejecutando en el código en la interfaz de su dispositivo.

6.3 Firebase

Firebase[4] es una plataforma basada en la nube y creada por Google nuevamente. Con Firebase podemos usar el conjunto de herramientas que nos ofrece para poder sincronizar nuestro proyecto y generar así la integración de nuestro backend, de esta manera poder crear usuarios para la App, agregar archivos, leer los datos subidos anteriormente, etc. Además de esto utilizaremos algunas tecnologías que nos ofrece Firebase, entre las que se encuentran Firebase Realtime Database y Firebase Authentication.

Firebase Realtime Database almacena y sincroniza datos con nuestra base de datos NoSQL alojada en la nube, en un formato JSON. Los datos se sincronizan con todos los clientes en tiempo real y se mantienen disponibles cuando la App no tiene conexión.

Firebase Authentication se integra estrechamente con otros servicios de Firebase y aprovecha estándares de la industria como OAuth 2.0 y OpenID Connect, por lo que se puede integrar con facilidad en tu backend personalizado.

6.4 Android Studio

Android Studio[3] es un programa de entorno de desarrollo integrado, basado en otro famoso IDE llamado IntelliJ IDEA, de JetBrains. Incorpora las características necesarias para interconectar el sistema donde se está ejecutando el código con el dispositivo móvil Android, o un emulador que viene integrado en el software.

Esta será la herramienta principal para desarrollar todo el código fuente de nuestra App, ya que Android Studio proporciona la opción de instalar unos plugins para la creación de proyectos Flutter con el soporte necesario que estos requieren. Y de la misma manera que disponemos del emulador Android, nos permite vincular un simulador iOS y web para poder testear nuestra aplicación en todas las plataformas posibles.

6.5 Xcode

Xcode[6], al igual que Android Studio, es un entorno de desarrollo integrado y en este caso específico, para la plata-

forma de iOS.

La necesidad de tener instalado este programa viene por el emulador de iPhone que el programa nos proporciona y por los controladores necesarios para que la aplicación se pueda ejecutar correctamente en cualquier dispositivo iOS, ya sea un móvil, tablet o una computadora.

7 IMPLEMENTACIÓN

En este punto se hará una descripción de cómo se ha llevado a cabo la implementación de las diferentes vistas y la gestión de los datos durante el desarrollo del proyecto.

7.1 Vistas

Todas las vistas siguen la misma estructura de árbol jerárquico de widgets.

Los Widgets son el componente principal que forman cada una de las vistas. Toda la estructura que forma el layout de la aplicación, ya sea desde texto, su posicionamiento, iconos, imágenes, campos de texto, etc. A todo lo describimos como un widget independiente, y es la superposición de cada uno de ellos lo que acaba formando una estructura de árbol la cual forma aquella vista que tenemos en la pantalla en ese momento concreto.

En la siguiente imagen, se encuentra el diagrama de flujo de la aplicación. En esta, se puede observar todas las posibles acciones que pueden realizar los usuarios. Figura 1.

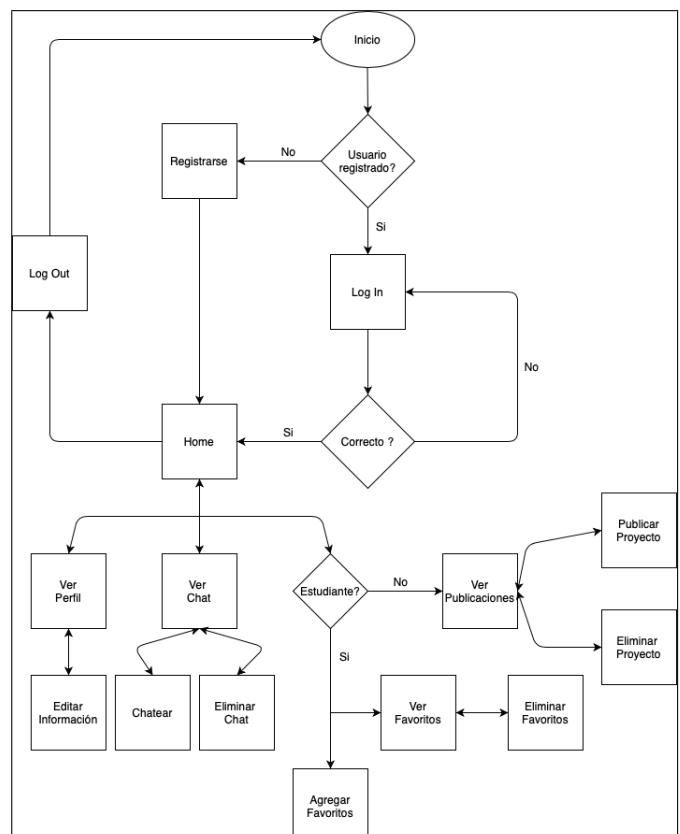


Fig. 1: Diagrama de flujo de la aplicación

7.1.1 Registro y Log In

Para las dos primeras pantallas con las que el usuario se puede topar se ha implementado un diseño sencillo. Ambas vistas se caracterizan por tener una estructura en forma de columna centrada, donde todos los elementos van uno detrás de otro siguiendo la misma alineación. La única diferencia que se puede encontrar es la adición de una última fila extra en la pantalla de registro, en el cual el usuario podrá decidir si quiere registrarse con alguna cuenta social oficial.

7.1.2 Home

En la pantalla Home, donde encontramos el Menú, sigue una estructura muy similar a las vistas anteriormente, con la diferencia que el numero de filas y columnas intercaladas es superior. Además, se ha agregado un nuevo widget llamado “drawer”, el cual nos permite implementar una barra lateral para poder así navegar a través de todas las funcionalidades de la aplicación. La barra lateral está compuesta por el nombre y el correo del usuario que ha ingresado, seguido de los campos a los que podemos acceder, en los cuales encontramos un cambio dependiendo de si el usuario es un estudiante o no.

7.1.3 Perfil

Para implementar el perfil del usuario, seguimos trabajando sobre el widget “Scaffold”. También, añadimos un nuevo widget sobre nuestra “AppBar” para que cualquier usuario, sea estudiante o no, pueda modificar su información personal en el momento que desee.

7.1.4 Proyectos Favoritos y Publicaciones

Para la implementación de estas dos vistas, debemos tener en cuenta principalmente si el usuario se trata de un estudiante o no. En caso de ser un estudiante quien esté utilizando la App, tendrá como opción ver cuales son los proyectos que ha guardado como favoritos. La vista sigue la misma línea que la vista principal del Menú, reciclando de esta manera los widgets utilizados anteriormente.

Por el otro lado, si el usuario no es un estudiante, la opción que encontramos es la de publicar un proyecto. Para esta vista nos hemos ayudado del widget “Container”, que modificando sus atributos a nuestro gusto obtenemos una vista en la cual la persona que publicita su proyecto puede añadir un título, una descripción y añadir una serie de tags para facilitar el filtrado de la publicación.

7.2 Base de datos

El segundo componente para el desarrollo de la App es el de gestionar un Backend como servicio (BaaS, por sus siglas en inglés). Un modelo en la nube que ofrece una buena escalabilidad en función de la necesidad de nuestra aplicación. La plataforma BaaS que se utilizará en este TFG será Firebase, de la popular compañía Google. Se ha decidido utilizar Firebase ya que es un servicio que da soporte a Flutter. Así mismo, Firebase va a facilitarnos diferentes tareas

que debemos realizar, como la creación del chat o la autenticación para mejorar la seguridad de nuestra aplicación. Para almacenar todos los datos que se moverán dentro de la aplicación y gestionarlos, esta contará con una base de datos NoSQL que sigue el protocolo JSON (Fig.16).

7.2.1 Firestore Database

Para que todas las vistas mencionadas anteriormente estén sincronizadas entre ellas, toda la información se almacena y se procesa desde Firestore Database de Firebase. Desde el momento que un usuario se registra, de él se guarda un ID único que utilizaremos como referencia para actualizar la información de ese perfil y que esta se vea reflejado en la aplicación. Para lograr esto, se ha seguido la documentación que Firebase nos proporciona, y moldeándola a Flutter logramos conectar la base de datos con toda la aplicación. La implementación varía según si se quiere crear un nuevo usuario, si se quiere añadir nuevos datos a un usuario ya creado, o si deseamos acceder a los datos de un usuario existente para reflejar estos en alguna pantalla.

Como se menciona anteriormente, para hacer un adecuado registro de usuario o log in, lo gestionamos a través de Firebase Authentication. Para llevar a cabo esto a través del código, utilizamos la librería de Dart `firebase_auth.dart`. De esta manera utilizando las correctas funciones de la librería, el usuario puede acceder a su cuenta mediante a un usuario y contraseña.

Por otro lado, para acceder a la información guardada o añadir nuevos datos, utilizamos la librería `cloud_firestore.dart`. En este caso por lo contrario al anterior, no encontramos funciones ya predefinidas para gestionar todos estos datos, con lo que, a nivel de programación, somos nosotros mismos quienes implementamos las funciones para acceder, agregar y modificar todos los datos necesarios.

7.3 Seguridad

Una parte muy importante dentro de las aplicaciones es la seguridad de los datos que circulan por ella, y poder garantizar a todos los usuarios que sus datos están seguros cuando están haciendo uso de ella. Es por eso por lo que tomamos este punto con importancia, de manera que queremos implementar una serie de configuraciones dentro de la App que garantice que será robusta

7.3.1 OAuth 2.0

Hasta ahora hemos podido contemplar que los usuarios necesitan de un correo electrónico y una contraseña para poder acceder a la aplicación. Se puede decir que esta es la forma más básica de controlar el acceso de terceros dentro de la aplicación, y se ha conseguido a través de la implementación de Firebase Authentication. Además de esta forma de autenticación, se quiere implementar una Autenticación OAuth 2.0.[9] que es un estándar abierto para la autorización de APIs, permitiendo compartir la información entre sitios, pero sin compartir la identidad. Estos son mecanismos utilizados por grandes compañías como Google, Facebook, Microsoft, etc.

OAuth 2.0. surge como media principal para palear la necesidad de envío continuo de credenciales entre cliente y servidor, de manera que reduce significativamente la posibilidad de que un ciberdelincuente puede interceptar información sensible. Así mismo damos la posibilidad al usuario a no tener la necesidad de almacenar un usuario y una contraseña como método de autenticación.

Como caso de uso en nuestra aplicación, si algún usuario previamente dispone de una cuenta de Google, Facebook o Twitter, podrá autenticarse con cualquiera de ellas.

7.3.2 Encriptación Algoritmo AES

Otra implementación que hemos añadido al proyecto es el encriptado de los datos de cada usuario, de manera que toda la información que se guarde en la base de datos quedará encriptada, evitando así que cualquier persona que acceda a ella pueda obtener datos sensibles de alguno de los usuarios. El algoritmo escogido para llevar a cabo la encriptación es AES-256[8]. AES ejecuta un cifrado simétrico por bloques de 128 bits cada uno, para lograr esto, utiliza una clave criptográfica específica que puede ser de 128, 192 o 256 bits, este último será nuestro caso.

Para lograr lo mencionado anteriormente, se utilizará una librería pública llamada Encrypt 5.0, que cuenta con una gran popularidad en la comunidad de Flutter y está altamente valorada por las funcionalidades que proporciona.

8 RESULTADO FINAL

En esta sección veremos cual ha sido el resultado final de la aplicación, relacionando imágenes de cada vista con la explicación previa realizada en apartado 7, Implementación. Añadir que plasmaremos los resultados de una encuesta que se realizó para estudiar la viabilidad de la aplicación entre los estudiantes.

8.1 Aplicación

Haciendo un recorrido a lo largo de la aplicación, las dos primeras vistas que se encuentra el usuario son, la autentificarse o la de registrarse como se observa en la Figura 2.

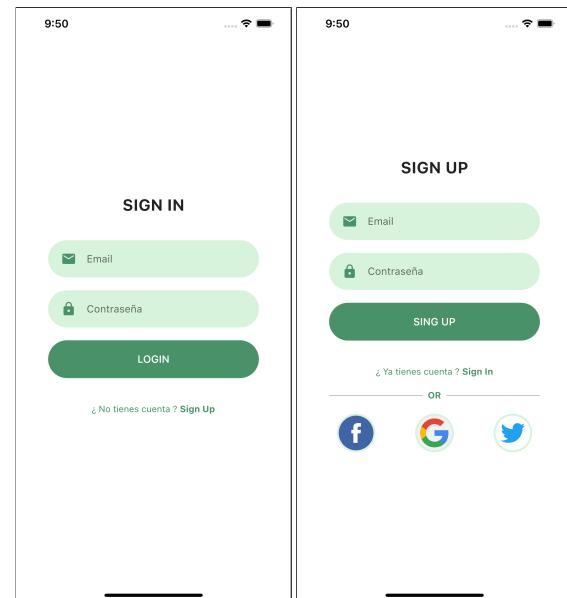


Fig. 2: Pantallas de SignIn y SignUp

En caso de entrar por primera vez y registrarnos, acabaremos de completar el registro en la siguiente pantalla. Figura 3.

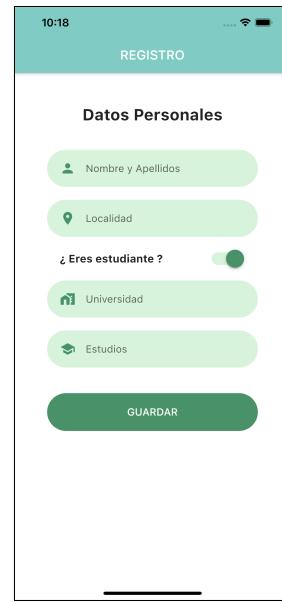


Fig. 3: Pantalla de Registro

Al completar el registro o en caso de ya haber estado registrados, la aplicación nos re conducirá a la página principal, donde se pueden encontrar todas las ofertas publicadas y desde ahí navegar a las diferentes opciones que la aplicación ofrece. Figura 4.

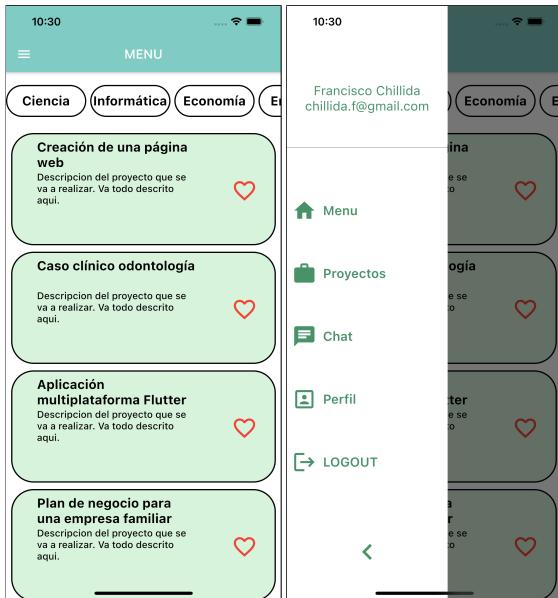


Fig. 4: Vistas principales



Fig. 6: Pantalla de favoritos

Entre las opciones que la aplicación nos permite, podemos encontrar una sección que varía según seamos estudiantes o no. Aquí como estudiantes podremos ver cuales de aquellos proyectos tenemos guardado en favoritos. En caso de no ser estudiantes, nos dará la opción de publicar algún nuevo proyecto y clasificarlo según su ámbito. Figura 5 y Figura 6.

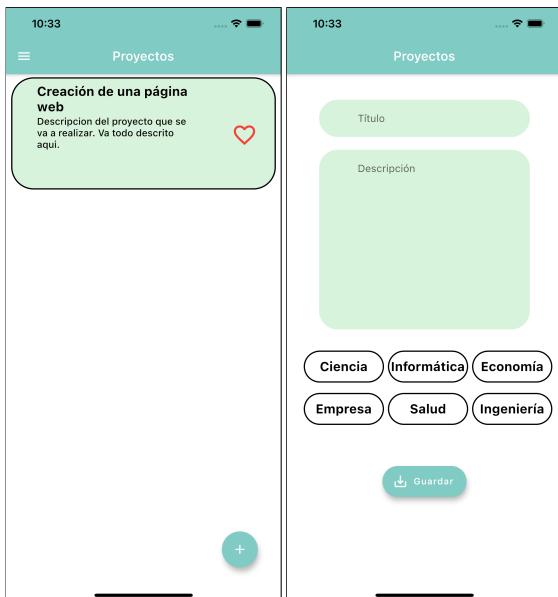


Fig. 5: Pantalla para añadir proyectos

El usuario también podrá acceder a su perfil, donde se podrá ver y modificar la información personal de uno mismo. Figura 7.

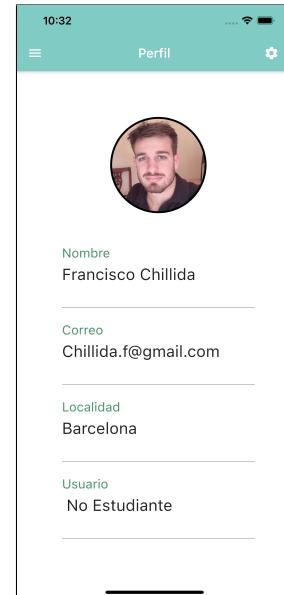


Fig. 7: Perfil

Para acabar encontramos el chat, por donde una vez el estudiante haya encontrado un proyecto que se adecúe a sus necesidades, pueda ponerse en contacto con la persona o entidad que esté al mando de esa publicación. De manera que puedan gestionar o resolver cualquier duda que pueda surgir del proyecto.

8.2 Encuestas

Para realizar el estudio final de viabilidad de la aplicación, se ha realizado una encuesta a un conjunto de estudiantes de diferentes facultades y universidades. En total, se ha obtenido una participación de 67 personas.

Con este estudio, se quería saber que tan útil puede ser la herramienta desarrollada, y a que porcentaje de alumnos podría haber ayudado a la hora de empezar su propio TFG. Es por ello que las preguntas realizadas son las siguientes:

- Del 1 al 5. ¿Cuánto te costo escoger tema para tu TFG? Siendo 1 muy poco y 5 mucho.**

Esta primera pregunta, recoge un poco el núcleo del problema, se quiere saber que porcentaje de gente ha estado en un punto en el cual, no se encontraban cómodos con ningún tema que se le planteaba o que ellos mismos pensaban. Como se puede observar en el siguiente gráfico, al rededor de un 50% de los encuestados tuvieron serias dificultades para encontrar un tema con el que se veían cómodos, y otro 20% tuvieron algunas dificultades, apenas un tercio de los encuestados encontró de manera sencilla un proyecto el cual realizar como TFG. Figura 8.

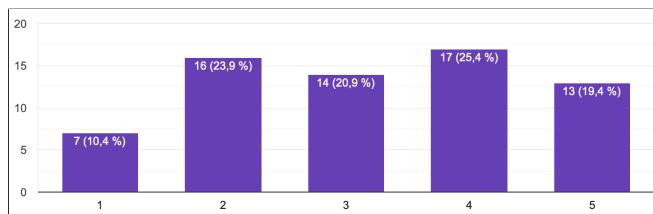


Fig. 8: Respuestas a la primera pregunta de la encuesta

- ¿El tema lo escogiste tú?**

Con esta pregunta se quería lograr a ver cuanta gente al final lograba sacar una idea por si misma y que porcentaje recurría a otras vías, ya sea un profesor, una empresa, un familiar, etc. De los resultados observamos que uno de cada tres estudiantes acaban sacando ideas de otras fuentes. Figura 9.

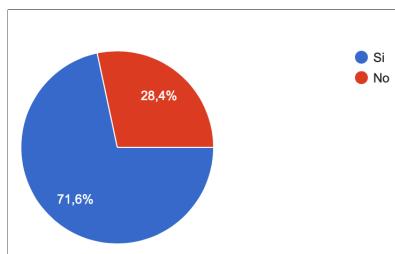


Fig. 9: Respuestas a la segunda pregunta de la encuesta

- Si es que no. ¿Quién te propuso el proyecto?**

En su mayoría, escogen una propuesta publicada por algún docente de la universidad. Algunos otros realizan el TFG con una empresa, esto suele pasar cuando el estudiante está realizando prácticas y es la propia empresa quien le ofrece realizar el TFG. Figura 10.

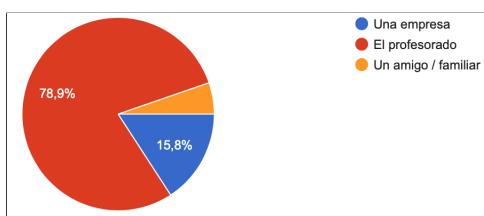


Fig. 10: Respuestas a la tercera pregunta de la encuesta

- ¿En algún momento utilizaste alguna aplicación para buscar ideas?**

Casi un 40% de los estudiantes recurren a alguna aplicación en busca de ideas. Debido a que la cantidad de aplicaciones que hay relacionadas con este tema es muy limitado, el otro 60% de los encuestados, en ningún momento han buscado información en ninguna aplicación. Figura 11.

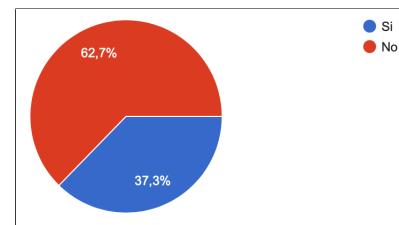


Fig. 11: Respuestas a la cuarta pregunta de la encuesta

- ¿Qué tan satisfecho estabas con el tema escogido? Siendo 1 muy poco y 5 mucho.**

A pesar de ser un proceso complicado y a veces largo, 57% está altamente satisfecho con el proyecto finalmente escogido. No hay que dejar de lado, que a pesar de ser un porcentaje bajo, prácticamente 1 de cada 3 estudiantes acaba realizando un TFG que no le gusta o le gusta muy poco. Figura 12.

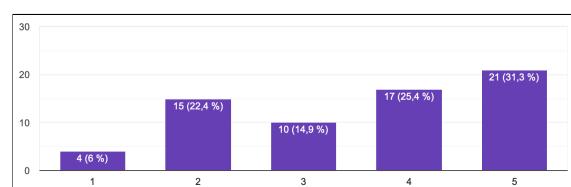


Fig. 12: Respuestas a la quinta pregunta de la encuesta

- Del 1 al 5. ¿Estarías dispuesto a realizar un proyecto para un tercero sin obtener una remuneración económica, pero pudiendo presentar este proyecto como un TFG propio? Siendo 1 en ningún caso y 5 altamente probable.**

Seguramente esta sea la pregunta más relevante de toda la encuesta, ya que con ella podemos ver el porcentaje de éxito que tendría la aplicación entre todos los estudiantes. Según la encuesta realizada, algo más de la mitad de los encuestados realizarían un proyecto para una persona, empresa, universidad, etc. sin recibir una remuneración económica a cambio. Y un 12% probablemente lo haría. Entre los encuestados 1 de cada 5 jamás se plantearía esa opción, y creemos que esto se debe, a que en carreras tecnológicas como ingeniería informática, prácticamente todas las prácticas en una empresa son remuneradas. Esto puede provocar un rechazo al alumno a realizar un proyecto para alguien de manera gratuita. Figura 13.

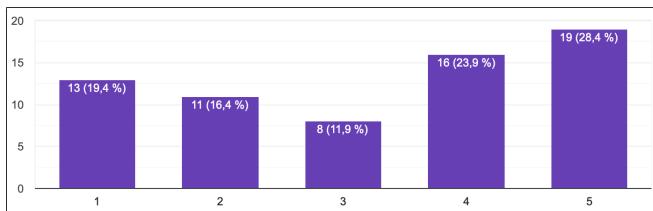


Fig. 13: Respuestas a la sexta pregunta de la encuesta

9 CONCLUSIONES

El objetivo de este TFG era proporcionar una nueva herramienta a todos los estudiantes universitarios para facilitarles a encontrar un proyecto y emplearlo como su propio trabajo final de grado. Por otro lado, también se beneficiarían aquellas personas, instituciones, empresas, etc. que por falta de medios no les fuera posible contratar a un trabajador y la utilicen. Gracias a la aplicación esta segunda parte involucrada podrá acceder a tener esta colaboración que no hubiera conseguido de otra forma.

Para ello hemos podido finalizar una aplicación multiplataforma, que se ve funcional, estética correcta e intuitiva. El hecho de haber utilizado Flutter, ha permitido que en este trabajo de seis meses, podamos llegar a un máximo número de personas que podrían utilizarla, ya que no todas ellas comparten el mismo sistema operativo en sus dispositivos móviles.

Con los resultados obtenidos de las encuestas, podemos asegurar que una parte de los estudiantes llegarían a probar la aplicación en un futuro. De esta manera hacer del proyecto no solo un prototipo teórico sino también una aplicación útil en la sociedad para resolver un problema que afecta a una porción de ella.

Personalmente estoy muy contento con el trabajo realizado, gracias a ello he aprendido a utilizar nuevas tecnologías las cuales no conocía, me ha ayudado a mejorar a auto planificar mis tareas y a realizar proyectos de una magnitud más elevada a los que me había enfrentado hasta el momento. En líneas futuras, preveo que este proyecto se puede seguir desarrollando de manera que pueda mejorar el diseño de la aplicación para que llegue a ser más atractiva al usuario.

AGRADECIMIENTOS

Primero de todo quería agradecer a todos mis amigos y compañeros, tanto de la universidad como los de fuera de ella, que me han estado apoyando a lo largo de toda la carrera. También a mi familia, que ha dedicado sus recursos y tiempo para que yo pueda formarme durante estos últimos años. Por último, me gustaría agradecer a mi tutora Marta Prim, la cual me ha guiado y ayudado a lo largo de el desarrollo de este TFG.

REFERÈNCIES

- [1] A. Yu, “The complete 2021 Flutter Development Bootcamp with Dart”, Appbrewery.co, 2021. [Online]. Available: <https://www.appbrewery.co/courses/>. [Accedido: 05- Feb-2021].
- [2] Flutter Team, “Flutter Docs”, Flutter.dev, 2021. [Online]. Available: <https://flutter.dev/docs/get-started/install>. [Accedido: 05-Feb-2021].
- [3] Android Studio Team, “Developers”, Android Studio, 2021. [Online]. Available: <https://developer.android.com/studio/intro?hl=es>. [Accedido: 05-Feb-2021].
- [4] Google Team, “Firebase”, Firebase, 2021. [Online]. Available: <https://firebase.google.com/>. [Accedido: 05-Feb-2021].
- [5] A.Andrew, “Scrum Of One: How to Bring Scrum into your One-Person Operation”, raywenderlich.com, 2021. [Online]. Available: <https://www.raywenderlich.com/585-scrum-of-one-how-to-bring-scrum-into-your-one-person-operation>. [Accedido: 09-Feb-2021].
- [6] Apple Team, “Xcode”, Xcode, 2021. [Online]. Available: <https://developer.apple.com/xcode/>. [Accedido: 12-Feb-2021].
- [7] Dart Team, “Dart”, Dart.dev, 2021. [Online]. Available: <https://dart.dev> [Accedido: 15-Abr-2021].
- [8] lc@leocavalcante.com, “Encrypt”, 2021. [Online]. Available: <https://github.com/leocavalcante/encrypt>. [Accedido: 10-May-2021].
- [9] A.Parecki, “Oauth 2.0”, Oauth.net, 2021. [Online]. Available: <https://oauth.net/2/grant-types/authorization-code/>. [Accedido: 20-May-2021].
- [10] Facebook Inc, ”React Navite”, 2021. [Online]. Available: <https://reactnative.dev/>. [Accedido: 10-Jun-2021].
- [11] Ionic Service, ’Ionic’, 2021. [Online]. Available: <https://ionicframework.com/>. [Accedido: 10-Jun-2021].

ANEXOS

A.1 Casos de Uso

Usuario no autentificado

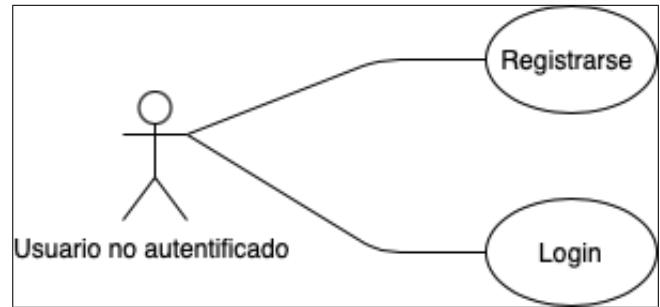


Fig. 14: Caso de uso, usuario no autentificado

Usuario estudiante

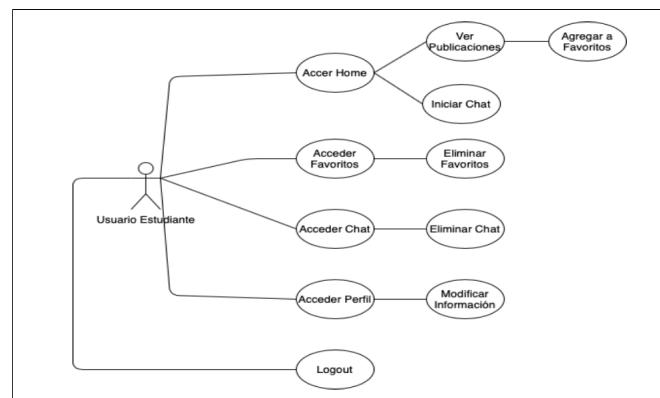


Fig. 15: Caso de uso, usuario estudiante

Usuario no estudiante

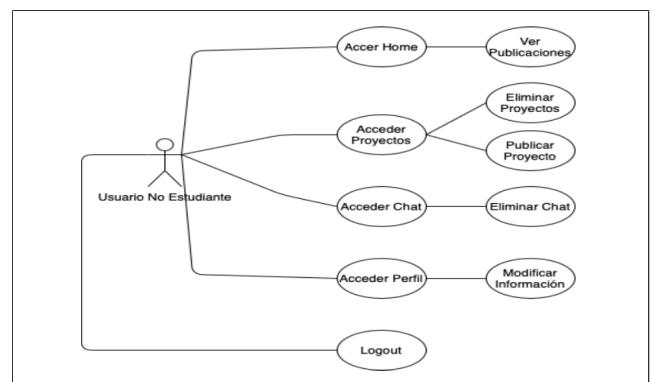


Fig. 16: Caso de uso, usuario no estudiante

A.2 Base de datos

```
{
  "Usuarios": {
    "idUsuario": "<Int>{
      "nombre": "<String>",
      "mail": "<String>",
      "contraseña": "<String>",
      "localidad": "<String>",
      "tipoUsuario": "<Int>{
        "universidad": "<String>",
        "estudios": "<String>",
        "dashboard": {
          "idDashboard": "<Int>{
            "titulo": "<String>",
            "descripcion": "List <String>",
            "tags": "List <string>",
            "creador": "<String>",
            "favorito": "<Bool>"
          }
        }
      }
    }
    "favoritos": {
      "idFavorito": "<Int>{
        "titulo": "<String>",
        "descripcion": "List <String>",
        "tags": "List <string>",
        "creador": "<String>",
        "favorito": "<Bool>",
        "valoracion": "<Int>"
      }
    }
    "publicaciones": {
      "idPublicacion": "<Int>{
        "titulo": "<String>",
        "descripcion": "List <String>",
        "tags": "List <string>",
        "creador": "<String>",
        "fecha": "<dateTime>"
      }
    }
    "chat": {
      "idChat": "<Int>{
        "nombreChat": "<String>",
        "nombreProyecto": "<String>"
      }
    }
  }
}
```

Fig. 17: Diseño de la base de datos noSQL