

APLICACIÓN/RED SOCIAL PARA INTERCAMBIAR AYUDA EN ASIGNATURAS DE LA CARRERA

APPLICATION/SOCIAL NETWORK TO EXCHANGE HELP IN UNIVERSITY SUBJECTS

Carlos Luengo Heras

Javier Vaquero Fernández

Joaquín Asensio Manzano

GRADO EN INGENIERÍA DE SOFTWARE
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID



TRABAJO DE FIN DE GRADO
CURSO 2021-2022

DIRECTOR
Antonio Sarasa Cabezuelo

RESUMEN

En este trabajo se comprenden aspectos tales como la especificación, el diseño y el desarrollo de Uni4All la aplicación multiplataforma de la cual trata este trabajo de fin de grado.

La aplicación tiene como objetivo ser una red social para estudiantes que ayude a los estudiantes en su vida universitaria y en el ámbito del estudio principalmente.

Sus principales funciones son la gestión y organización de grupos de trabajo para las diferentes asignaturas, crear e interactuar en foros con temáticas universitarias y el intercambio de conocimiento mediante reuniones online o presenciales.

PALABRAS CLAVE

Flutter, Dart, Firebase, XCode, Android Studio, GitHub, Visual Studio Code

ABSTRACT

This work includes aspects such as the specification, design and development of Uni4All, the multiplatform application that this final degree project deals with.

The application aims to be a social network for students that helps students in their university life and mainly in the field of study.

Its main functions are the management and organization of work groups for the different subjects, creating and interacting in forums with university themes and the exchange of knowledge through online or face-to-face meetings.

KEY WORDS

Flutter, Dart, Firebase, XCode, Android Studio, GitHub, Visual Studio Code

ÍNDICE

ÍNDICE	4
ÍNDICE DE FIGURAS	6
ÍNDICE DE TABLAS	9
CAPÍTULO 1. INTRODUCCIÓN.....	11
1.1. MOTIVACIÓN	11
1.2. OBJETIVOS	12
1.3. ESTRUCTURA DE LA MEMORIA.....	12
1.4. PLANIFICACIÓN DEL TRABAJO DE FIN DE GRADO	13
CHAPTER 1. INTRODUCTION	15
1.1. MOTIVATION.....	15
1.2. OBJECTIVES.....	15
1.3. MEMORY STRUCTURE.....	16
1.4. PLANNING OF THE PROJECT	17
CAPÍTULO 2. ESTADO DEL ARTE	19
2.1. ¿TIENES SAL?:	19
2.2. WUOLAH:.....	21
2.3. MEETUP:.....	22
2.4. TEAYUDO:.....	25
CAPÍTULO 3. TECNOLOGÍA EMPLEADA.....	27
3.1. HERRAMIENTAS PARTE CLIENTE	27
3.1.1. <i>Android Studio</i>	27
3.1.2. <i>XCode</i>	28
3.1.3. <i>Visual Studio Code</i>	29
3.2. HERRAMIENTAS DEL LADO DEL SERVIDOR	30
3.2.1. <i>Firebase Database</i>	30
3.2.2. <i>Flutter</i>	30
3.3. OTRAS HERRAMIENTAS	31
3.3.1. <i>GitHub</i>	31
3.3.2. <i>Dart</i>	32
3.3.3. <i>Google Drive</i>	33
3.3.4. <i>Diagrams.net</i>	33
CAPÍTULO 4. CASOS DE USO	35
4.1. ACTORES DEL SISTEMA.....	35
4.2. DESCRIPCIÓN DE LOS CASOS DE USO	35
4.2.1. <i>Módulo estudiante</i>	36
4.2.2. <i>Módulo administrador</i>	88
CAPÍTULO 5: ARQUITECTURA.....	109
5.1. INTRODUCCIÓN ARQUITECTURA EN FLUTTER	109

5.2.	ARQUITECTURA PRINCIPAL DE LA APLICACIÓN	110
5.2.1.	<i>Introducción</i>	110
5.2.2.	<i>Patrones arquitectura</i>	110
5.3.	PATRONES DE DISEÑO	112
5.3.1.	<i>Inyección de dependencias</i>	112
5.3.2.	<i>Singleton</i>	113
5.3.3.	<i>Observer</i>	113
5.3.4.	<i>Data Transfer Object</i>	114
CAPÍTULO 6. MODELO DE DATOS.....		115
6.1.	MODELO E-R	115
6.2.	IMPLEMENTACIÓN DE LA BASE DE DATOS.....	116
6.3.	COLECCIONES DE DOCUMENTOS	116
CAPÍTULO 7: DISEÑO		119
7.1.	COLORES Y DISEÑO	119
7.2.	FUNCIONALIDAD PRINCIPAL	120
7.2.1.	<i>User y login</i>	120
7.2.2.	<i>Perfil</i>	123
7.2.3.	<i>Foros</i>	126
7.2.4.	<i>Grupos</i>	128
7.2.5.	<i>Servicios</i>	130
CAPÍTULO 8: EVALUACIÓN.....		133
8.1.	HERRAMIENTAS Y PLANTEAMIENTO	133
8.2.	RESULTADO DE LA EVALUACIÓN:.....	134
8.2.1.	<i>General</i>	134
8.2.2.	<i>Registro/Inicio de sesión</i>	137
8.2.3.	<i>Grupos</i>	138
8.2.4.	<i>Perfil</i>	139
8.2.5.	<i>Foros</i>	140
8.2.6.	<i>Conclusiones finales y sugerencias:</i>	141
CAPÍTULO 9. CONCLUSIONES Y TRABAJO FUTURO.		143
9.1.	CONCLUSIONES.....	143
9.2.	TRABAJO FUTURO.....	144
CHAPTER 9: CONCLUSIONS AND FUTURE WORK.....		151
9.1.	CONCLUSIONS	151
9.2.	FUTURE WORK.....	152
CAPÍTULO 10: APORTACIONES INDIVIDUALES.....		157
CARLOS LUENGO HERAS:		157
JAVIER VAQUERO FERNÁNDEZ:.....		158
JOAQUÍN ASENSIO MANZANO:.....		159
BIBLIOGRAFÍA.....		160
ANEXO I. GUÍA DEL USUARIO.....		161

ÍNDICE DE FIGURAS

Figura 1 - Diagrama de Gantt de tareas en las que se desglosa el proyecto.....	14
Figura 2 - Gantt diagram of the tasks' roadmap made during the TFG development.....	18
Figura 3 - Introducción a ¿Tienes Sal?.....	19
Figura 4 - Sección publicaciones de ¿Tienes Sal?	19
Figura 5 - Sección mercadillo de ¿Tienes Sal?	20
Figura 6 - Sección eventos de ¿Tienes Sal?	20
Figura 7 - Sección grupos de ¿Tienes Sal?	20
Figura 8 - Sección mensajes de ¿Tienes Sal?.....	21
Figura 9 - Interfaz de Wuolah de un estudiante de la FDI de 3º	21
Figura 10 - Interfaz de la sección de preguntas y respuestas.	22
Figura 11 - Resumen de Meetup	22
Figura 12 - Selección de intereses en Meetup.....	23
Figura 13 - Bienvenida UI Meetup	23
Figura 14 - Eventos en Meetup	24
Figura 15 - Grupos en Meetup	24
Figura 16 - Página inicio TeAyudo.....	25
Figura 17 - Temáticas de ayuda entre vecinos	25
Figura 18 - Preguntas y respuestas de TeAyudo	26
Figura 19 - Pantalla desarrollo Android Studio con emulador.....	27
Figura 20 - IDE Xcode Con emulador iPhone	28
Figura 21 - Imagen Interfaz de edición de código de Visual Studio Code	29
Figura 22 - pantalla Data de Firebase Realtime Database	30
Figura 23 - Flutter mostrando la multiplataformidad.....	31
Figura 24 - vista general proyecto GitHub.....	32
Figura 25 - Icono Dart.....	32
Figura 26 - vista página principal Google Drive.....	33
Figura 27 - vista principal diagrams.net.....	34
Figura 28 - Interfaz diferenciadora del usuario administrador.....	35
Figura 29 - Diagrama caso de uso módulo usuario estudiante.....	36
Figura 30 - Diagrama caso de uso módulo usuario estudiante funcionalidad perfil	36
Figura 31 - Diagrama caso de uso módulo usuario estudiante funcionalidad servicios	46
Figura 32 - Diagrama caso de uso módulo usuario estudiante funcionalidad grupo	60
Figura 33 - Diagrama caso de uso módulo usuario estudiante funcionalidad foros	77
Figura 34 - Diagrama de casos de uso de administrador.....	88
Figura 35 - marco de capas en Flutter	109
Figura 36 - Flutter, Arquitectura de Aplicaciones.....	109
Figura 37 - Esquema app desarrollada	110
Figura 38 - Estructura patrón MVVM	111
Figura 39 - Diagrama del patrón de service layer y repository intercambiando datos..	112
Figura 40 - Dependencia provider.....	112
Figura 41 - Singleton ejemplo getInstance().....	113
Figura 42 - Patrón observer.....	113

Figura 43 - Patrón DTO	114
Figura 44 - Modelo E-R de la aplicación Uni4All	115
Figura 45 - Implementación de la bbdd	116
Figura 46 - Diagrama colecciones documentos	117
Figura 47 - tfg_theme.dart del proyecto Uni4All.....	119
Figura 48 - Almacenamiento clave login	120
Figura 49 - Código del registro	120
Figura 50 - Código del login	121
Figura 51 - Servicios, grupos y posts.	121
Figura 52 - Dashboard usuario	122
Figura 53 - Barra inferior con navegado a los módulos.....	122
Figura 54 - Modificado de datos Figura 55 - Servicio actualización usuario	123
Figura 56 – Código del cambio de foto.....	124
Figura 57 - Código de la llamada a la API para actualización de usuario	124
Figura 58 - Formulario autenticación actualización credenciales.....	124
Figura 59 - Modulo administración.....	125
Figura 60 - eliminar y salir de la cuenta.....	125
Figura 61 - Vista principal foros Figura 62 – Vista posts de subforo	126
Figura 63 - Servicio de foro	126
Figura 64 - SearchDelegateForums de Uni4All.....	127
Figura 65 - Llamada al servicio de creación de una respuesta.....	127
Figura 66 - Visualización hilo y respuestas	128
Figura 67 - Servicio de unirse a un grupo	129
Figura 68 - Figura dashboard grupos	129
Figura 69 - Mostrado de grupo.....	129
Figura 70 – Selección de grupo.....	129
Figura 71 - Código MessageGroup de Uni4All	130
Figura 72- Editado y borrado de grupo (owner)	130
Figura 73 - Búsqueda de servicios	131
Figura 74 - Detalle de servicio propio.....	131
Figura 75 - Creación de servicios.....	131
Figura 76 - Servicio de unirse a un servicio.....	131
Figura 77 - Código de envío de mensajes en servicios	132
Figura 78 - Editado y borrado de un grupo propio.....	132
Figura 79 - Gráfico pregunta: ¿Qué edad tienes?.....	134
Figura 80 - Gráfico pregunta: ¿Qué valoración general darías a la app?.....	134
Figura 81 - Gráfico pregunta: ¿Habrías utilizado esta herramienta en la universidad? 135	
Figura 82 - Grafico pregunta: Si has respondido que sí, comenta los motivos.....	135
Figura 83 - Gráfico pregunta: ¿Te parece atractiva la interfaz visual de la app?.....	136
Figura 84 - Pregunta: Si has contestado no, comenta los motivos y la modificación... 136	
Figura 85 - Gráfico pregunta: ¿Encuentras la sección de servicios intuitiva?	137
Figura 86 - Pregunta: ¿Cambiarías algo o has encontrado alguna dificultad?	137
Figura 87 - Gráfico pregunta: ¿Encuentras la sección de grupos intuitiva?.....	138

Figura 88 - Pregunta: ¿Cambiarías o añadirías algo en los grupos?	138
Figura 89 - Gráfico pregunta: ¿Encuentras la sección de perfil intuitiva?	139
Figura 90 - Pregunta: ¿Cambiarías o añadirías algo en el perfil?	139
Figura 91 - Gráfico pregunta: ¿Encuentras la sección de foros intuitiva?	140
Figura 92 - Pregunta: ¿Cambiarías o añadirías algo en foros?.....	140
Figura 93 - Gráfico pregunta: ¿Cuál te parece la sección más útil?.....	141
Figura 94 - Gráfico pregunta: ¿Y la que menos útil?.....	141
Figura 95 - Pregunta: ¿Tienes alguna sugerencia?.....	142
Figura 96 - Coste de los servicios	145
Figura 97 - Situación actual de búsqueda.....	146
Figura 98 - Post sin responder.....	147
Figura 99 - Post respondido	147
Figura 100 - Posible ubicación del botón de citado en respuestas	149
Figura 101 - Situación actual del editado y ubicación para editado de respuesta.....	150
Figura 102 - Cost of services	153
Figura 103 - Current search situation.....	154
Figura 104 - Answered post.....	155
Figura 105 - Unanswered post	155
Figura 106 - Possible location of the quote button in answers	156
Figura 107 - Current status of post editing and location for response editing	156
Figura 108 - Pantalla login.....	161
Figura 109 - Pantalla registro	161
Figura 110 - Pantalla inicio	162
Figura 111 - Dashboard usuario registrado	162
Figura 112 - Pantalla perfil usuario.....	163
Figura 113 - Modificado de datos	163
Figura 114 - Panel usuario	164
Figura 115 - Panel foros.....	164
Figura 116 - Panel grupos	165
Figura 117 - Panel servicios.....	165
Figura 118 - Eliminar cuenta y Log out.....	166
Figura 119 - Vista principal foros	166
Figura 120 - Posts de un subforo.....	167
Figura 121 - Visualización hilo y respuestas	168
Figura 122 - Mostrado de grupo.....	169
Figura 123 - Dashboard grupos.....	169
Figura 124 - Selección de grupos.....	170
Figura 125 - Búsqueda de servicios	171
Figura 126 - Creación de servicio	171
Figura 127 - Detalle de servicio propio.....	172

ÍNDICE DE TABLAS

Tabla 1 - Registro.....	37
Tabla 2 - Inicio de sesión con login manual.....	38
Tabla 3 - Autenticado automático.....	39
Tabla 4 - Mostrar perfil de usuario desde barra de app.....	40
Tabla 5 - Modificar perfil de usuario	41
Tabla 6 - Modificar datos login.....	43
Tabla 7 - Cerrar sesión	44
Tabla 8 - Eliminar cuenta	45
Tabla 9 - Alta publicación servicio	47
Tabla 10 - Baja publicación servicio.....	48
Tabla 11 - Mostrar servicio	49
Tabla 12 - Modificar publicación servicio	51
Tabla 13 - Listar todos los servicios propios	51
Tabla 14 - Listar todos los servicios publicados	52
Tabla 15 - Buscar servicios por código de asignatura del servicio	53
Tabla 16 - Unión a servicio	54
Tabla 17 - Envío de mensaje en servicio.....	56
Tabla 18 - Acceso a videoconferencias.....	58
Tabla 19 - Mostrar perfil de miembro de servicio	59
Tabla 20 - Alta grupo	61
Tabla 21 - Baja grupo.....	62
Tabla 22 - Mostrar grupo	63
Tabla 23 - Modificar grupo	65
Tabla 24 - Listar grupos propios	65
Tabla 25 - Listar todos los grupos.....	66
Tabla 26 - Buscar grupos por código de asignatura	67
Tabla 27- Unión a grupo	69
Tabla 28 - Salida de grupo	69
Tabla 29 - Envío de mensaje grupal.....	71
Tabla 30 - Acceso a repositorio de Github.....	72
Tabla 31 - Acceso a almacenamiento de Drive.....	74
Tabla 32 - Listar miembros de un grupo	75
Tabla 33 - Mostrar perfil de miembro del grupo.....	76
Tabla 34 - Alta post.....	78
Tabla 35 - Baja Post	79
Tabla 36 - Mostrar post.....	80
Tabla 37 - Modificación Post.....	82
Tabla 38 - Listar todos los posts de un subforo	82
Tabla 39 - Buscar posts por título	83
Tabla 40 - Alta respuesta.....	84

Tabla 41 - Mostrar perfil de autor de post.....	85
Tabla 42 - Mostrar perfil de autor de respuesta	86
Tabla 43 - Listar todos los subforos	87
Tabla 44 - Listar todos los usuarios	89
Tabla 45 - Modificar usuario.....	91
Tabla 46 - Eliminar usuario.....	93
Tabla 47 - Listar todos los posts de los foros	94
Tabla 48 - Editar post foro	96
Tabla 49 - Eliminar post foro	98
Tabla 50 - Listar todos los grupos.....	99
Tabla 51 - Editar grupo	101
Tabla 52 - Eliminar grupo	103
Tabla 53 - Listar todos los servicios	104
Tabla 54 - Editar servicio	106
Tabla 55 - Eliminar servicio.....	108

Capítulo 1. Introducción

En este capítulo se describirán la motivación para realizar este trabajo de fin de grado, los objetivos planteados, la organización de la memoria, así como la planificación del mismo.

1.1. Motivación

Existen varias herramientas disponibles para alumnos tales como el campus virtual o grupos temáticos y otras aplicaciones de terceros, pero no son muy populares entre los estudiantes, son poco intuitivas, cuentan con pocas funcionalidades o no están todas integradas en una misma aplicación y muchas no ofrecen un acceso multiplataforma, sino que están pensadas o bien para móvil o bien para ordenador lo cual dificulta su uso.

Como respuesta a estos problemas en este TFG se ha desarrollado una plataforma con una interfaz intuitiva y familiar, accesible tanto a través del móvil como web que proporciona varios servicios fundamentales hoy en día para una experiencia más dinámica, sencilla y eficaz del ámbito universitario.

Empleando elementos visuales familiares para el usuario y con unos casos de uso sencillos de replicar se consigue una interfaz limpia e intuitiva que permite que el usuario se sienta cómodo desde el primer uso de la aplicación, y decida utilizarla de manera regular para gestionar aspectos académicos.

Habilitando una versión web y móvil de calidad permite que la aplicación se adapte al usuario y que no sea al contrario. Esto permite que el estudiante utilice una versión u otra en función de sus necesidades o sus preferencias

Los servicios que comprende Uni4All son: Intercambio de servicios entre estudiantes, foros y ayuda para la formación de grupos.

A través de estos tres pilares se simplifica en gran medida la vida de los estudiantes. En primer lugar, pueden mantenerse informados y compartir ellos mismos información a través de los foros, en segundo lugar, pueden fácilmente buscar un grupo de trabajo que se adapte a sus necesidades y exigencias y adherirse al mismo obteniendo probablemente mejores resultados de lo normal y por último ayudar o ser ayudado con el fin de obtener un rendimiento académico superior.

1.2. Objetivos

El objetivo principal es desarrollar una aplicación multiplataforma que permita a los estudiantes compartir información e informarse sobre aspectos universitarios, crear, gestionar y unirse a grupos de trabajo para las asignaturas, practicas, o trabajos de fin de grado y por último concertar reuniones para ayudarse y compartir conocimiento.

Este objetivo se particulariza en los siguientes objetivos más específicos:

- Desarrollar una sección de foros que permite a los estudiantes tanto postear como leer artículos dentro de los foros de diferentes temáticas (establecidas por el administrador) y realizar búsquedas y filtrados.
- Desarrollar una sección de intercambio de ayuda para que los estudiantes impartan entre ellos clases de las diferentes asignaturas tanto de manera telemática como presencial.
- Desarrollar una sección para la formación de grupos tanto de trabajo (para asignaturas de la carrera) como para el TFG que permita a los estudiantes filtrar en función de la asignatura y solicitar la unión a un grupo y a su vez permita también la creación de grupos en los que se podrá detallar el tamaño, la asignatura y una descripción en la cual definir el objetivo y necesidades del grupo.
- Crear una base de datos para la recopilación y organización de los datos que permita tener la información actualizada y a disposición de los alumnos.

1.3. Estructura de la memoria

En este apartado se detalla la estructura que se seguirá en la memoria:

- Capítulo 1: Introducción. En este capítulo se describe la motivación del trabajo, los objetivos, la estructura de la memoria y la planificación del trabajo de fin de grado.
- Capítulo 2: Estado del arte. En este capítulo se estudian herramientas existentes en el mercado que ofrecen funcionalidad similar a la realizada en el trabajo.
- Capítulo 3: Tecnología empleada. En este capítulo se describe la tecnología utilizada para implementar el proyecto.

- Capítulo 4: Casos de uso. En este capítulo se definen los actores y casos de uso que se explicarán mediante tablas junto a sus requisitos.
- Capítulo 5: Arquitectura. En este capítulo se explicará la arquitectura de la aplicación con los patrones utilizados.
- Capítulo 6: Modelo de datos. En este capítulo se explicará el diagrama entidad relación en el que se ha planeado el proyecto para su implementación, detallando los documentos y entidades.
- Capítulo 7: Diseño. En este capítulo se realizará un estudio sobre el diseño e implementación de los distintos casos de uso.
- Capítulo 8: Evaluación. En este capítulo se expondrá las opiniones aportadas por usuarios especializados a los que se ha suministrado la aplicación y que han probado durante unas horas.
- Capítulo 9: Conclusiones y trabajo futuro. En este capítulo se condensa una reflexión sobre la utilidad de la aplicación y se listan casos de usos que se harán en un futuro explicándolos brevemente.
- Capítulo 10: Aportaciones individuales. En este capítulo se expone el trabajo realizado por cada uno de los autores.

1.4. Planificación del trabajo de fin de grado

El desarrollo del trabajo se ha estructurado en 8 fases:

- SRS: en esta fase ha dividido la aplicación en los principales módulos sobre los que trata la misma.
- Casos de uso: en esta fase se ha especificado para cada módulo y actor los casos de uso.
- Entidad – Relación: en esta fase se ha dado forma a las distintas entidades y como se relacionan entre ellas para satisfacer los casos de uso especificados.
- Modelado en Base de datos: en esta fase se ha creado y organizado la base de datos utilizada en la aplicación.
- Desarrollo de front: en esta fase se ha codificado las vistas de la aplicación.
- Desarrollo de back: en esta fase se ha desarrollado la lógica de la aplicación.
- Corrección de errores: en esta fase se ha llevado a cabo una exhaustiva corrección de errores y validación de la aplicación.
- Documentación – Memoria: en esta fase se ha realizado el documento correspondiente con la memoria del TFG.

En la figura 1 se muestra el diagrama de Gantt donde se muestra la distribución temporal de las fases

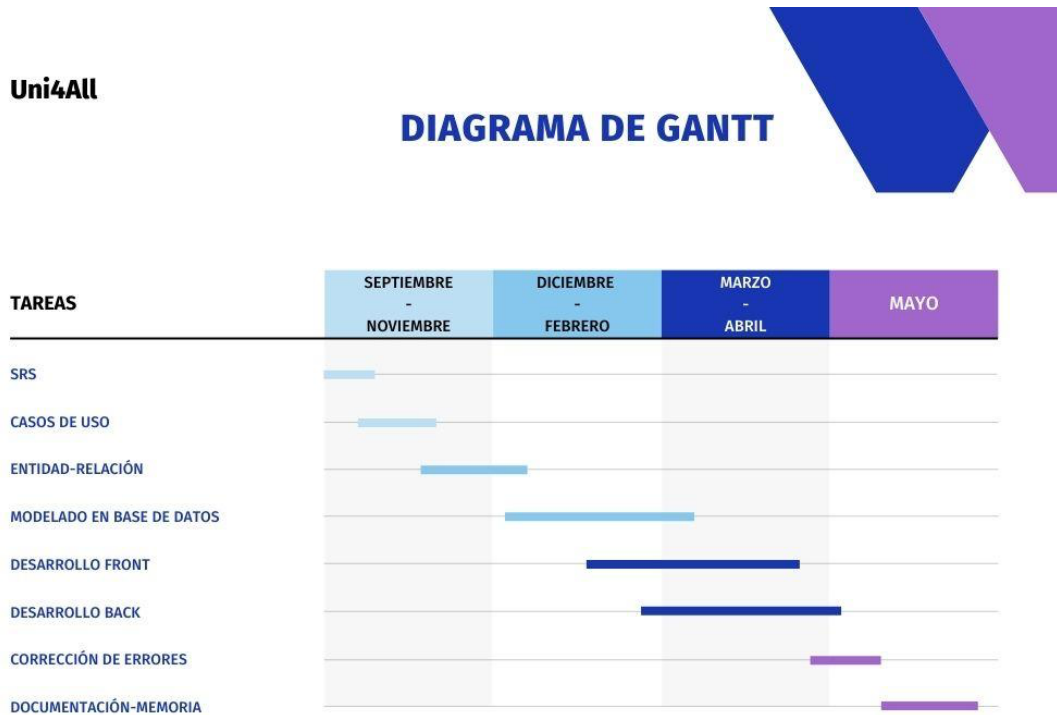


Figura 1 - Diagrama de Gantt de tareas en las que se desglosa el proyecto.

Chapter 1. Introduction

This chapter will describe the motivation to carry out this end-of-degree project, the objectives set, the organization of the memory, as well as its planning.

1.1. Motivation

There are several tools available for students such as the virtual campus or thematic groups and other third-party applications, but they are not very popular among students, they are not very intuitive, they have few functionalities or they are not all integrated in the same application and many do not offer a multiplatform access, but they are designed either for mobile phones or for computers, which makes their use difficult.

In response to these problems in this TFG, a platform has been developed with an intuitive and familiar interface, accessible both through mobile and web that provides several fundamental services today for a more dynamic, simple and effective experience in the university environment.

Using familiar visual elements for the user and with simple to replicate use cases, a clean and intuitive interface is achieved that allows the user to feel comfortable from the first use of the application and decide to use it on a regular basis to manage academic aspects.

Enabling a quality web and mobile version allows the application to adapt to the user and not the other way around. This allows the student to use one version or another depending on their needs or preferences.

The services included in Uni4All are: Exchange of services between students, forums and help for group formation.

Through these three pillars, the lives of students are greatly simplified. Firstly, they can stay informed and share information themselves through the forums, secondly, they can easily find a working group that suits their needs and demands and join it, probably obtaining better results than usual and therefore last help or be helped in order to obtain superior academic performance.

1.2. Objectives

The main objective is to develop a multiplatform application that allows students to share information and find out about university aspects, create, manage and

join work groups for subjects, internships, or final degree projects and finally arrange meetings to help each other and share knowledge.

This objective is specified in the following more specific objectives:

- Develop a forum section that allows students to both post and read articles within the forums on different topics (established by the administrator) and perform searches and filters.
- Develop a help exchange section for students to teach classes on different subjects among themselves, both online and in person.
- Develop a section for the formation of both work groups (for career subjects) and for the TFG that allows students to filter according to the subject and request joining a group and, in turn, also allows the creation of groups in which can detail the size, the subject and a description in which to define the objective and needs of the group.
- Create a database for the compilation and organization of the data that allows the information to be up-to-date and available to the students.

1.3. Memory structure

This section details the structure that the memory will follow:

- Chapter 1: Introduction. This chapter describes the motivation of the work, the objectives, the structure of the memory and the planning of the final degree project.
- Chapter 2: State of the Art. This chapter studies existing tools on the market that offer functionalities.
- Chapter 3: Technology used. This chapter describes the technology used to implement the project.
- Chapter 4: Use Cases. This chapter defines the actors and use cases that will be explained through tables along with their requirements.
- Chapter 5: Architecture. This chapter will explain the architecture of the application with the patterns used.

- Chapter 6: Data Model. This chapter will explain the entity relationship diagram in which the project has been planned to be implemented in the database.
- Chapter 7: Design. In this chapter, a study will be carried out on the design and implementation of the different use cases.
- Chapter 8: Evaluation. This chapter will present the opinions provided by specialized users to whom the application has been supplied and who have tested for a few hours.
- Chapter 9: Conclusions and future work. This chapter condenses a reflection on the usefulness of the application and a list of use cases that will be developed in the future with a brief explanation.
- Chapter 10: Individual Contributions. This chapter presents the work done by each of the authors.

1.4. Planning of the project

The development of the work has been structured in 8 phases:

- SRS: In this phase, the application has been divided into the main modules that it deals with.
- Use cases: In this phase, the use cases have been specified for each module and actor.
- Entity – Relationship: In this phase, the different entities have been shaped and how they relate to each other to satisfy the specified use cases.
- Modeling in Database: In this phase, the database used in the application has been created and organized.
- Front development: In this phase, the views of the application have been coded.
- Back development: In this phase the logic of the application has been developed.
- Error correction: In this phase, an exhaustive error correction and validation of the application has been carried out.
- Documentation – Memory: In this phase, the corresponding document with the memory of the TFG has been made.

Figure 1 shows the Gantt diagram where the temporal distribution of the phases is shown.

Uni4All

GANTT CHART

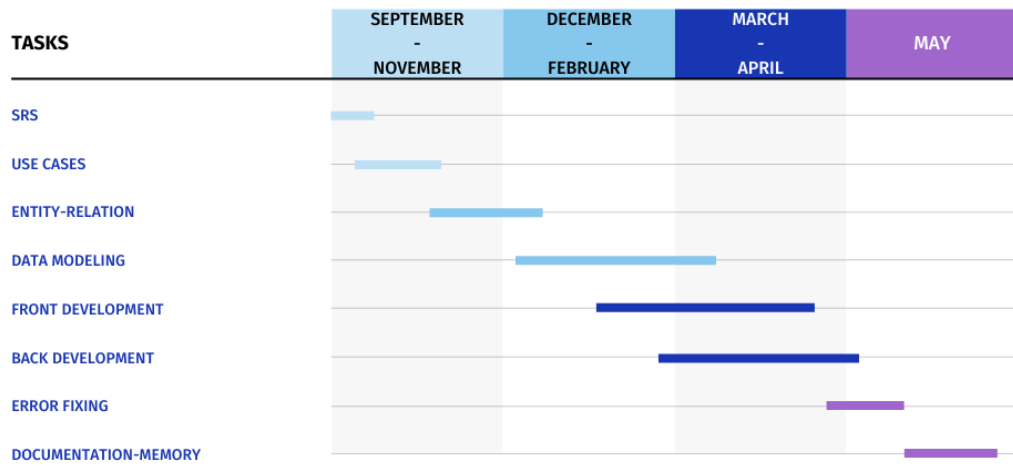


Figura 2 - Gantt diagram of the taks' roadmap made during the TFG development.

Capítulo 2. Estado del arte

En este capítulo se describirán un conjunto de aplicaciones que disponen de funciones similares a las implementadas en este trabajo.

2.1. ¿Tienes sal?: Una aplicación muy utilizada que ofrece una plataforma, similar a una red social, pensada para que los vecinos de un mismo barrio puedan ponerse en contacto entre ellos de forma rápida y sencilla para intercambiar ayuda, intereses, ofertar actividades que hacer. En la figura 3 se explica en detalle toda la funcionalidad de ¿Tienes Sal? desde la sección descubre de su web^[1].



Figura 3 - Introducción a ¿Tienes Sal?

La aplicación cuenta con 5 módulos principales, publicaciones, mercadillo, eventos, grupos y finalmente mensajes. A continuación, se presenta un pequeño resumen de cada uno de ellos:

- En la figura 4 se recoge el módulo publicaciones, donde se comparten posts libres sobre ideas, opiniones, necesidades, comentarios libres...

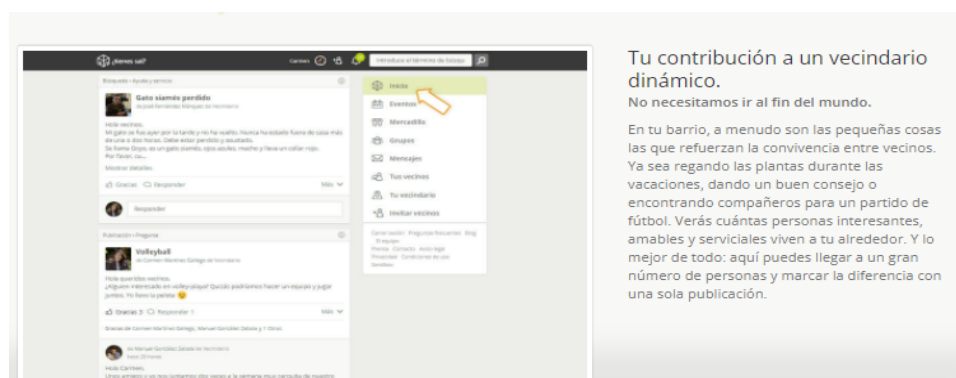


Figura 4 - Sección publicaciones de ¿Tienes Sal?

- En la figura 5 se muestra mercadillo, donde se venden cosas que no se usan, se buscan servicios, se piden productos...



Figura 5 - Sección mercadillo de ¿Tienes Sal?

- En la figura 6 se recogen los eventos, donde se organizan planes, se dan a conocer sitios, se coordinan salidas...

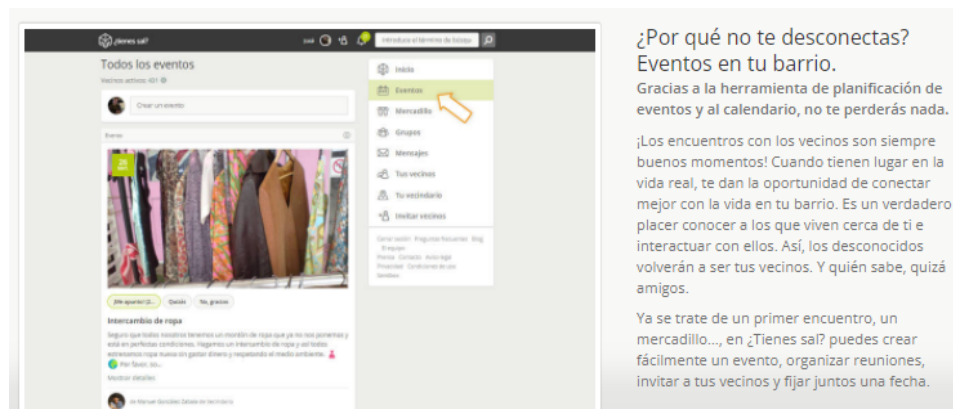


Figura 6 - Sección eventos de ¿Tienes Sal?

- En la figura 7 se muestran los grupos, donde se permite al usuario crear grupos para suplir sus necesidades, unirse a grupos ya creados...



Figura 7 - Sección grupos de ¿Tienes Sal?

- En la figura 8 se muestra mensajes, para conectar vecinos, ya sean privados o comunicaciones abiertas...



Figura 8 - Sección mensajes de ¿Tienes Sal?

2.2. Wuolah: Es una aplicación que ha ganado mucho éxito en los últimos años. La aplicación ofrece una red social donde los estudiantes al darse de alta seleccionan su facultad entre todas las universidades de España, donde entran en contacto con el resto de los estudiantes de esa universidad e incluso de los mismos grados, pero de otras universidades. Allí pueden subir sus apuntes para ayudar a otros estudiantes, compartir dudas, opiniones y otros comentarios^[2].

En la figura 9 se muestra la interfaz de usuario que ofrece Wuolah al entrar siendo un estudiante de 3º curso del grado Ingeniería del Software de la Universidad Complutense de Madrid^[3].

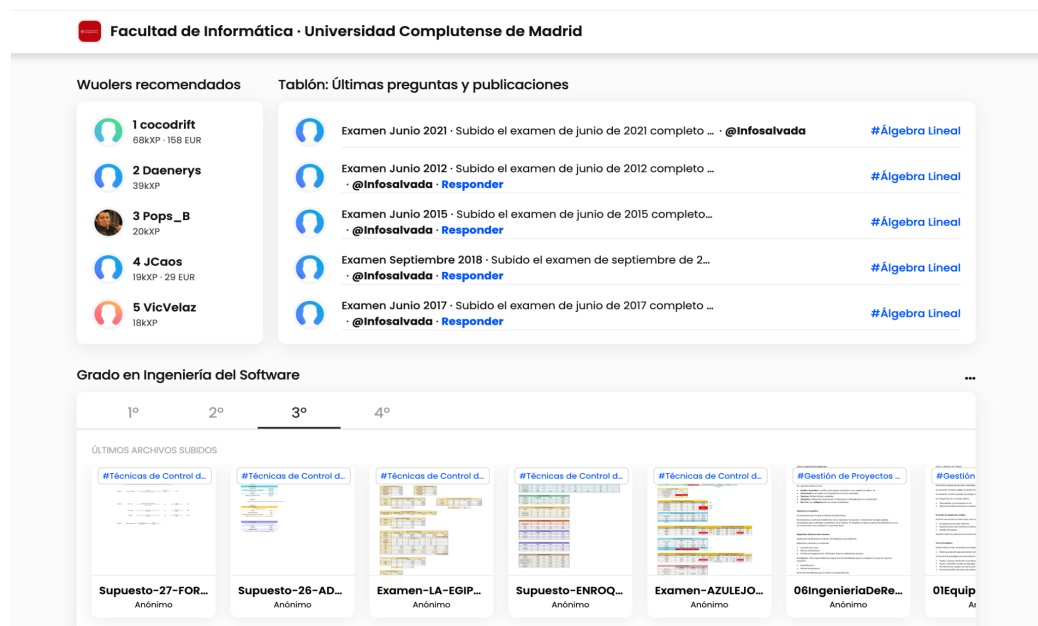


Figura 9 - Interfaz de Wuolah de un estudiante de la FDI de 3º

A diferencia de Wuolah, la aplicación que se ha desarrollado en este TFG no tiene ninguna funcionalidad para facilitar la subida de apuntes, sin embargo, la ayuda de un estudiante a otro sí que existe, además de publicaciones y comentarios libres reflejados en la figura 10, a modo tablón, para la publicación de información.

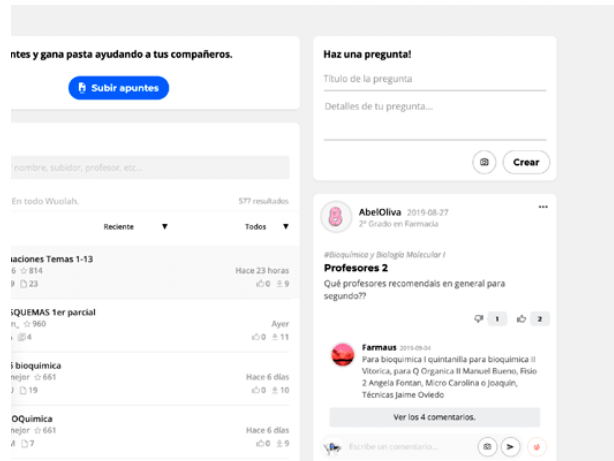


Figura 10 - Interfaz de la sección de preguntas y respuestas.

En la aplicación que se ha desarrollado tampoco se ha incluido la opción de monetizar la ayuda, aunque se ha planteado en el capítulo 9 de trabajo futuro.

2.3. Meetup: Aplicación que permite a los usuarios compartir eventos, formar grupos, conocer a gente nueva, hacer amigos, buscar apoyo... e incluso desarrollar negocios. Se puede encontrar esta información desarrollada en la sección *Acerca de*^[4]. La figura 11 resume brevemente el potencial de Meetup.



Figura 11 - Resumen de Meetup

Al registrarte la aplicación pide introducir la ubicación y acto seguido ofrece una serie de temas de interés entre los que elegir, que se podrán editar y ampliar (o reducir) después, como se puede ver en la figura 12.

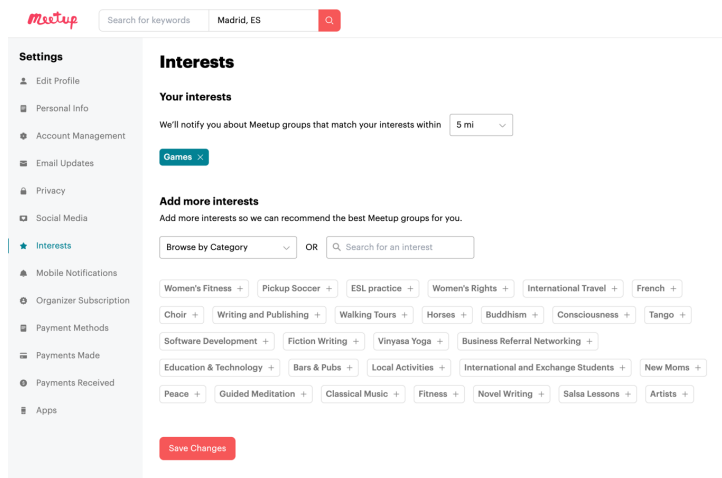


Figura 12 - Selección de intereses en Meetup

Una vez unido, la aplicación ofrece al usuario una pequeña introducción sobre la formación de grupos y la unión a eventos.

En la figura 13 se muestra la interfaz de un usuario recién registrado, donde puede navegar entre los futuros eventos de su zona según los intereses seleccionados previamente.

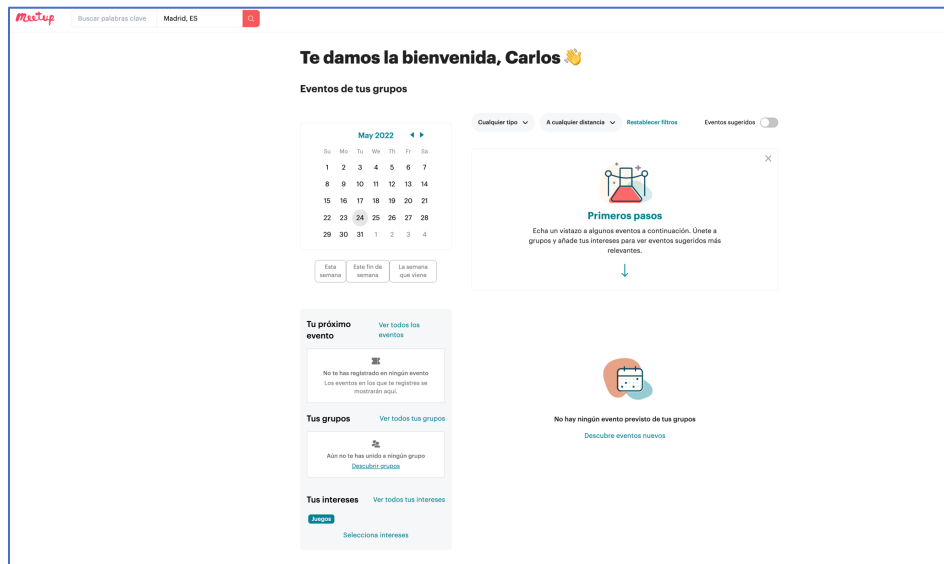


Figura 13 - Bienvenida UI Meetup

Desde la web de Meetup se pueden gestionar y navegar los grupos y eventos disponibles y ya unidos. En las figuras 14 (eventos) y 15 (grupos) se puede

apreciar la interfaz que muestra al usuario un conjunto de estos para elegir, utilizando ubicación e intereses.

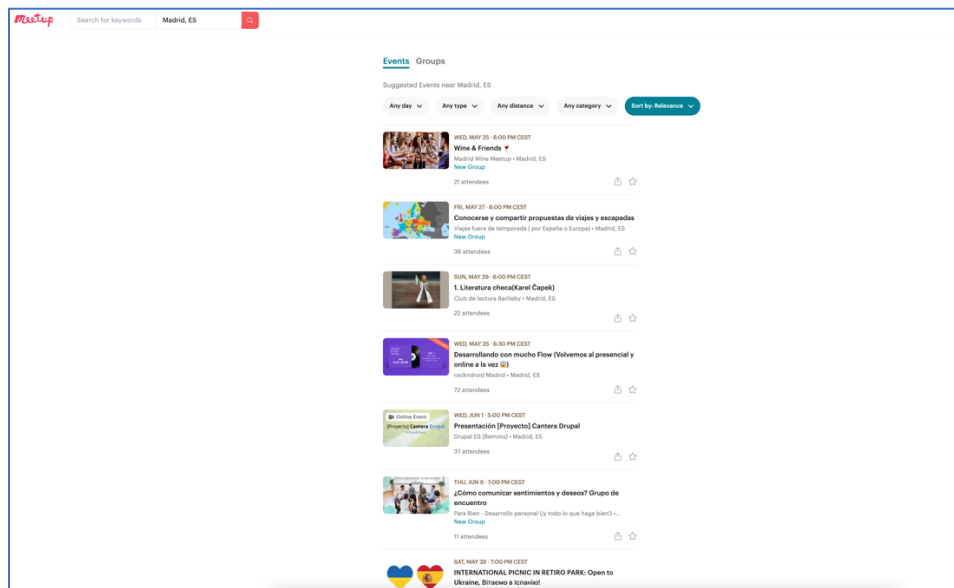


Figura 14 - Eventos en Meetup

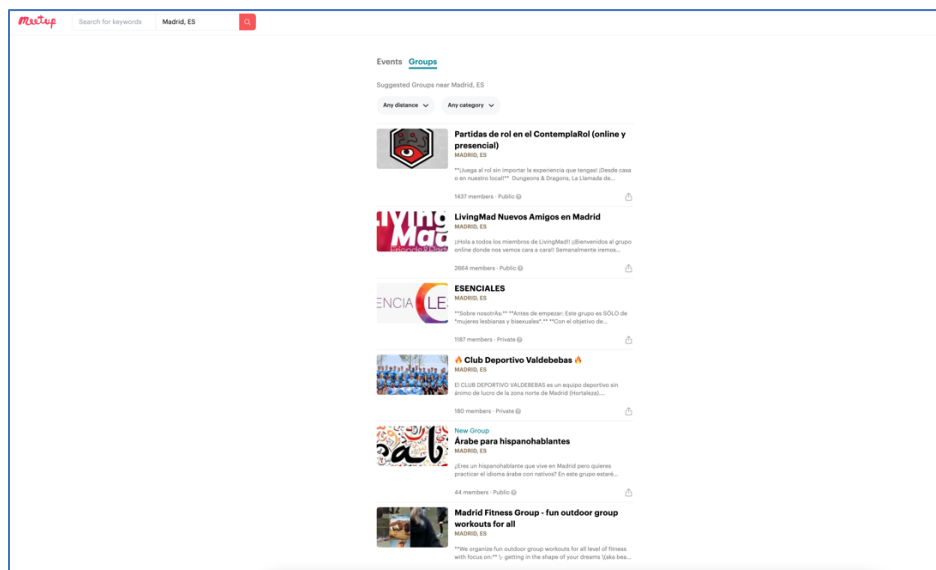


Figura 15 - Grupos en Meetup

2.4. TeAyudo: Aplicación a modo red social que permite intercambiar ayuda entre vecinos de barrio^[5]. Un usuario crea una petición de ayuda, o la ofrece, y otro usuario responde a la petición. Los mensajes se organizan a modo hilo. Además, cuando el usuario recibe la ayuda puede poner valoración a otros usuarios.

La figura 16 muestra la página de inicio de la aplicación.



Figura 16 - Página inicio TeAyudo

A diferencia de la aplicación realizada en el trabajo de fin de grado, TeAyudo no está enfocada a los estudios ni a los estudiantes, es más general. En la figura 17 se pueden observar distintos ejemplos de uso y los temas de las ayudas que se ofrecen y se piden.

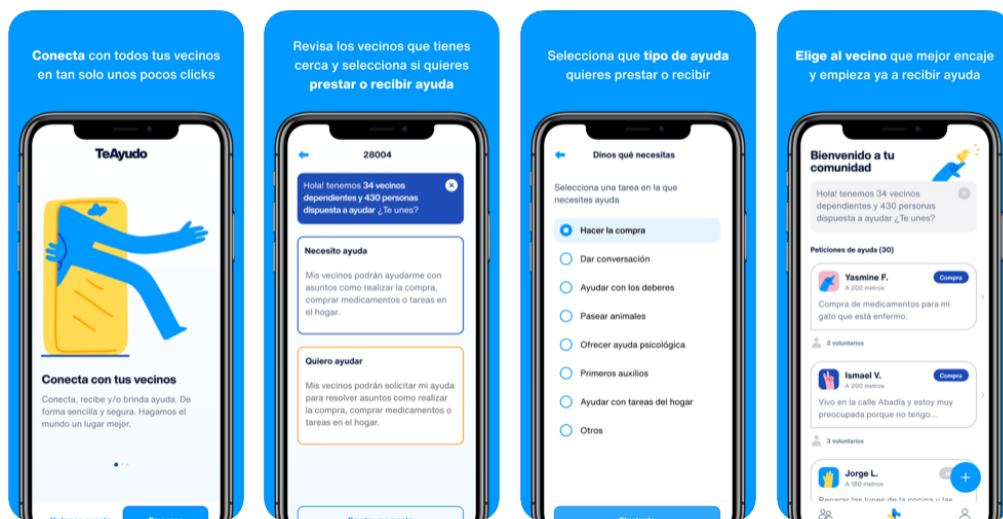


Figura 17 - Temáticas de ayuda entre vecinos

No se ha tenido en cuenta la posibilidad de utilizar la aplicación del TFG con mayores temáticas que las de estudios, pero podría ser una opción para trabajo futuro.

En la sección de preguntas y respuestas^[6] explica su funcionamiento y ciertas cuestiones sanitarias, pues nació durante el confinamiento en el año 2020. En la figura 18 se puede ver el estilo de esta página.

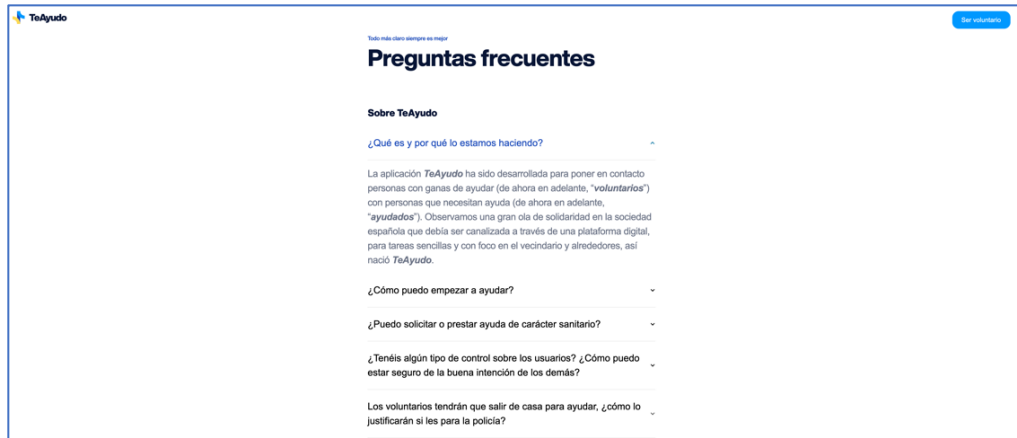


Figura 18 - Preguntas y respuestas de TeAyudo

Capítulo 3. Tecnología empleada

En este capítulo se van a describir las tecnologías utilizadas para el desarrollo del proyecto.

3.1. Herramientas parte cliente

3.1.1. Android Studio

Android Studio ^[7] es un IDE gratuito a través de la Licencia Apache 2.0 destinado al desarrollo Android. Admite lenguajes de programación tales como Java, C++, Go y Kotlin. Esta herramienta se encuentra disponible para Linux, macOS, Chrome OS y Windows.

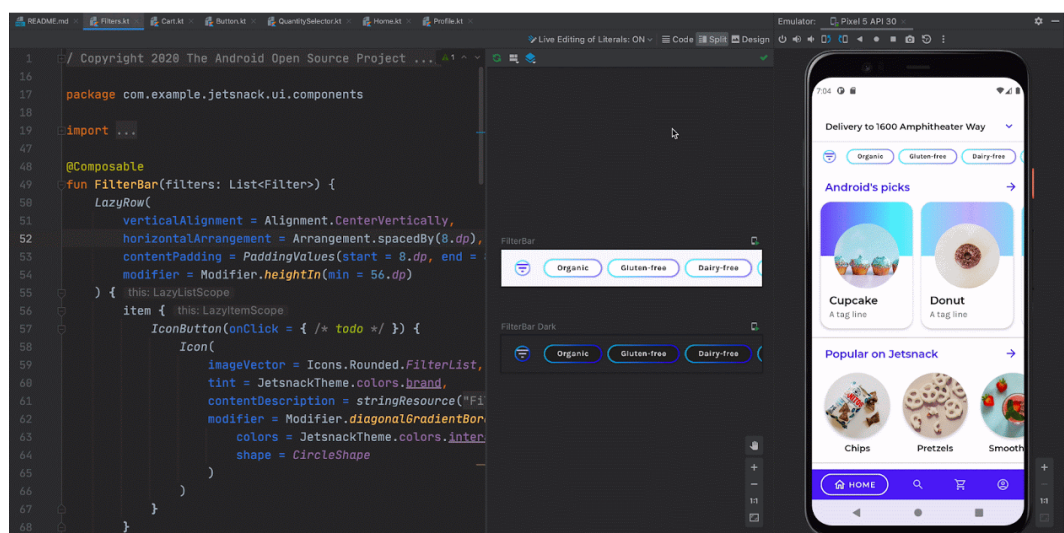


Figura 19 - Pantalla desarrollo Android Studio con emulador

En la figura 19 se muestra una ejecución de código virtualizado directamente en un simulador de Android.

Entre la multitud de herramientas de las que dispone, en este proyecto ha sido utilizada para genera el emulador del terminal Android, que han permitido la ejecución del código y sus respectivas pruebas. En concreto para la mayoría de las pruebas el terminal utilizado ha sido un Nexus 6.

3.1.2. XCode

XCode^[8] es un entorno de desarrollo integrado para macOS con un conjunto de herramientas destinadas al desarrollo de software para productos Apple.

Es una herramienta totalmente gratuita, en el App Store de Apple, solo tiene soporte para macOS, aunque desde hace años la comunidad de desarrolladores lleva intentando que Apple lo porte al iPad Pro, que ya cuenta con mismo hardware (e incluso superior) que muchísimos ordenadores Mac.

Dentro de los lenguajes que soporta se encuentran C, C++, Swift, Java, AppleScript, Objective-C y Objective-C++.

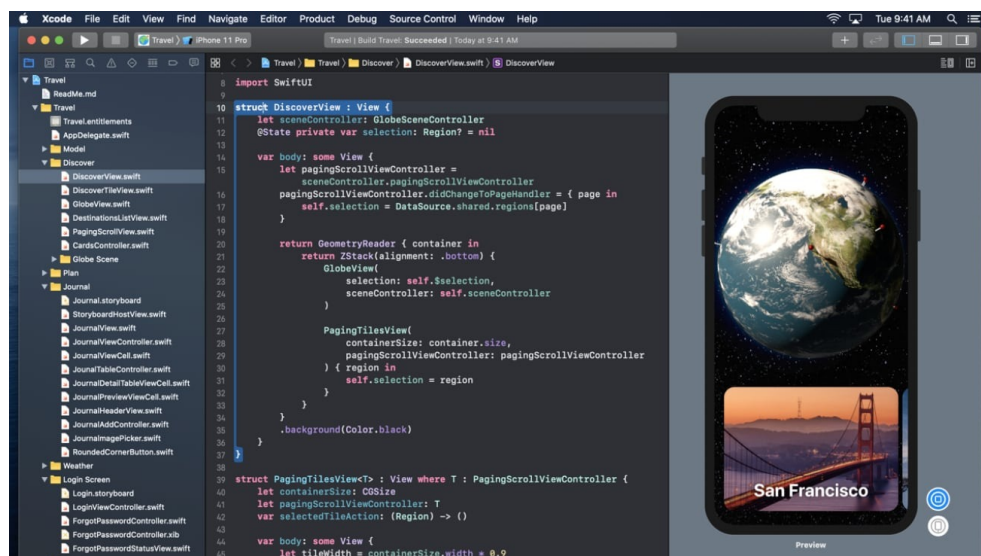


Figura 20 - IDE Xcode Con emulador iPhone

Al igual que Android Studio ha permitido ejecutar la aplicación en un terminal de Apple, en concreto se ha empleado un iPhone 12 Pro Max con iOS 15.2. En la figura 20 se puede apreciar la compilación de una aplicación en un iPhone 11 Pro.

3.1.3. Visual Studio Code

Visual Studio Code^[9] es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web.

Es una herramienta totalmente gratuita que permite desarrollar en una gran cantidad de distintos lenguajes de programación. Resulta un programa sumamente interesante y útil dado a su alto grado de personalización a través de extensiones, instalables a través de su Marketplace.

En este proyecto han sido empleadas varias extensiones referentes al desarrollo de Dart y Flutter, Live Share que se emplea para desarrollar de manera simultánea una misma versión de código y por último para la gestión de repositorios, Git Graph y Git Lens.

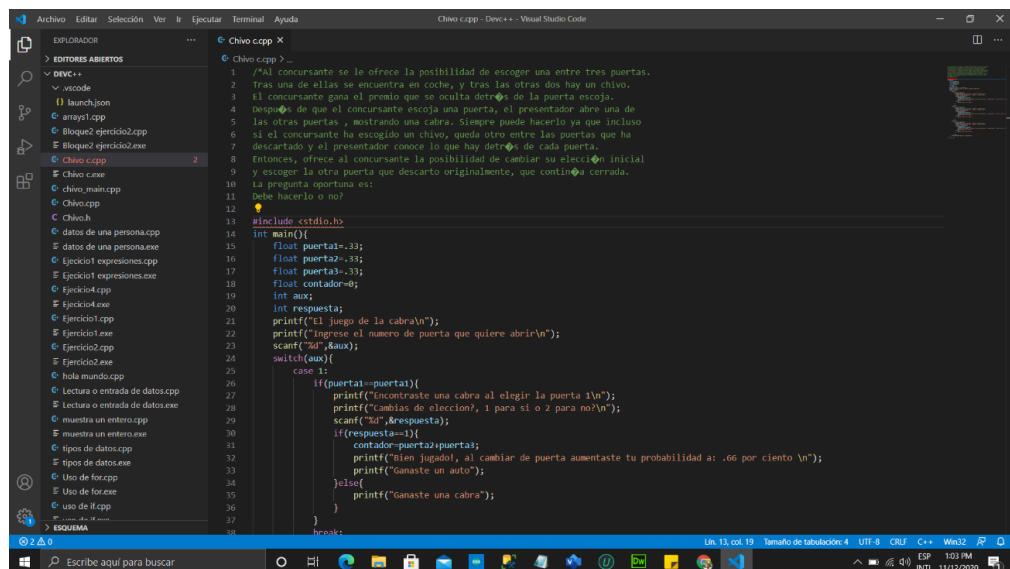


Figura 21 - Imagen Interfaz de edición de código de Visual Studio Code

Ha sido empleado en el desarrollo del código dado que es una herramienta muy eficiente, debido a la gran cantidad de extensiones y personalizaciones que permite.

En la figura 21 se muestra un ejemplo de .cpp corriendo dentro de Visual Studio Code, con la misma interfaz que ha tenido el proyecto (a excepción del emulador que siempre se ha tenido corriendo en un lado de la pantalla).

3.2. Herramientas del lado del servidor

3.2.1. Firebase Database

Firebase ^[10] es una plataforma perteneciente a Google, ubicada en la nube, cuya labor fundamental es el desarrollo de aplicaciones tanto web como móvil. Esta se encuentra integrada con Google Cloud Platform que cuenta con un conjunto de herramientas cuya finalidad es la creación y la sincronización de proyectos.

En concreto dentro de estas herramientas, se ha utilizado Realtime Database. Se trata de una base de datos, en tiempo real. Esta API da la posibilidad de que los datos sean sincronizados y almacenados en la nube de Firebase.

Esta herramienta ha permitido organizar y gestionar los datos de la base de datos de la aplicación desarrollada en la nube.

En la figura 22 se muestra la interfaz de Firebase, donde hemos controlado la base de datos sobre todo al hacer las pruebas para hacer trazas en el código.

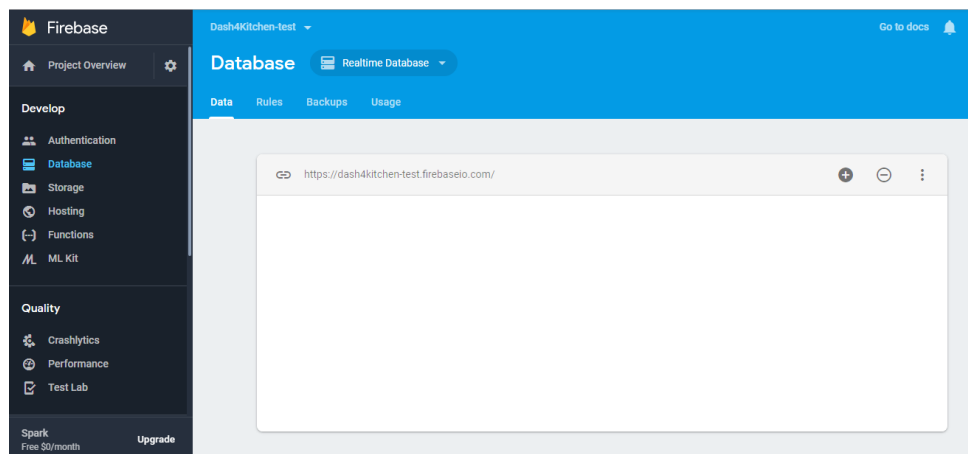


Figura 22 - pantalla Data de Firebase Realtime Database

3.2.2. Flutter

Flutter ^[11] es un SDK desarrollado por Google que permite crear aplicaciones multiplataforma.

Inicialmente fue pensado para un uso privado dentro de la compañía, pero finalmente dado el potencial que tenía dicha framework decidieron publicarlo.



Figura 23 - Flutter mostrando la multiplataformidad

Entre sus principales características además de su desarrollo tanto web como Android e iOS destaca su compilación en nativo, haciendo que pese a que sea un desarrollo multiplataforma no pierda fluidez ni eficiencia la aplicación.

Por otro lado, posee una opción llamada ‘Hot reload’, que permite refrescar los cambios realizados en app de una manera muy rápida, facilitando así el desarrollo de la misma.

Por estos motivos, pero sobre todo con la idea en mente de que mediante 1 sola aplicación codificada correctamente podríamos tener 3 (versión web – Android e iOS, como en la figura 23) ha sido el SDK elegido para este TFG.

3.3. Otras herramientas

3.3.1. GitHub

GitHub^[12] es un repositorio online gratuito que permite gestionar proyectos y controlar versiones de código.

Está pensado para ser una especie de red social para desarrolladores y es uno de los repositorios más usados de todo el mundo, contando con una gran popularidad debido a su facilidad para compartir y gestionar el código y a que es totalmente gratuita.

En la figura 24 se muestra el repositorio real de nuestro proyecto, con nosotros 3 como participantes.

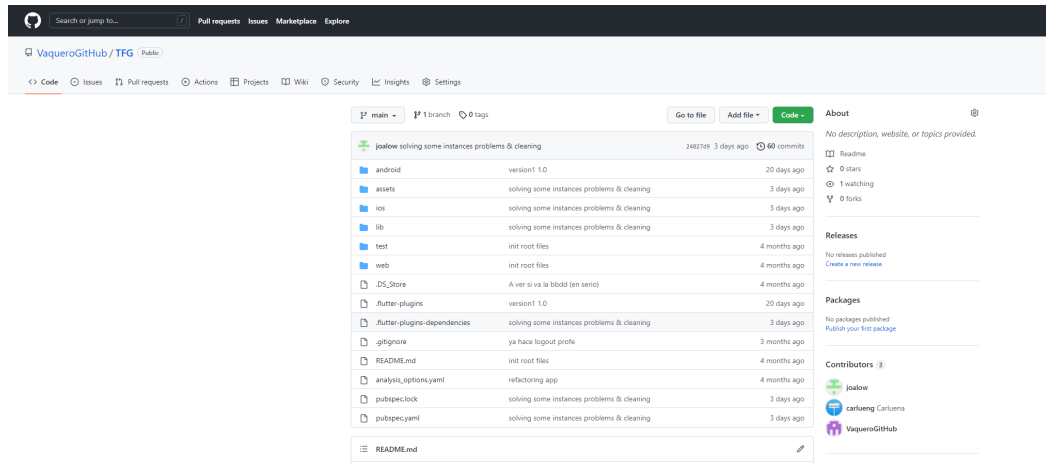


Figura 24 - vista general proyecto GitHub

Esta herramienta ha permitido organizar y mantener actualizado el código de los participantes y tener un control de las versiones.

3.3.2. Dart

Dart^[13] es un lenguaje creado por Google. Este fue ideado con la finalidad de ciertos errores y ofrecer mejores resultados que JavaScript. Fundamentalmente se emplea en proyectos pequeños y ofrece una gran seguridad.

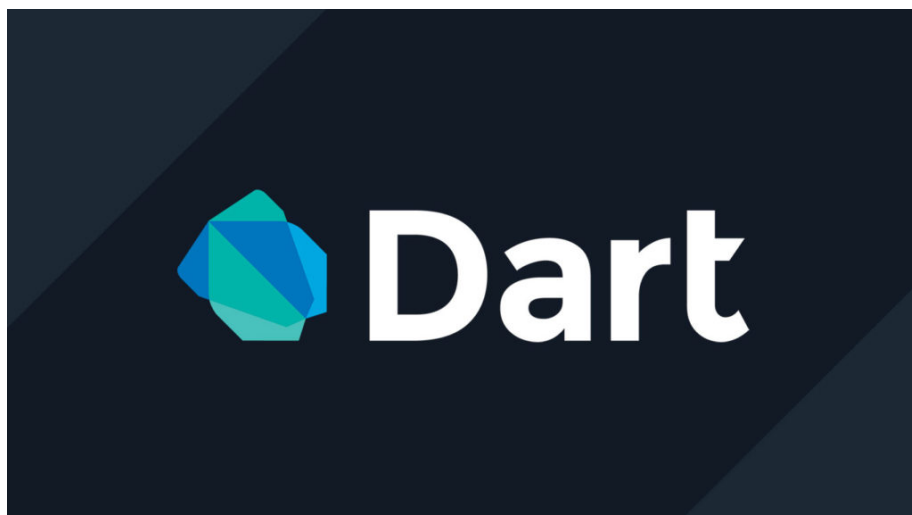


Figura 25 - Icono Dart

Este lenguaje es el utilizado para el desarrollo de aplicaciones en Flutter por lo tanto ha sido el empleado en Uni4All.

3.3.3. Google Drive

Google Drive [14] es un servicio de almacenamiento en la nube perteneciente a la empresa Google.

Es una herramienta disponible para la gran mayoría de plataformas, en la que cada usuario dispone de 15GB de espacio de almacenamiento. Permite almacenar archivos y organizarlos en carpetas. Estas carpetas a su vez pueden ser privadas o compartidas con la finalidad de que varias personas sean contribuyentes y se intercambien información entre ellos.

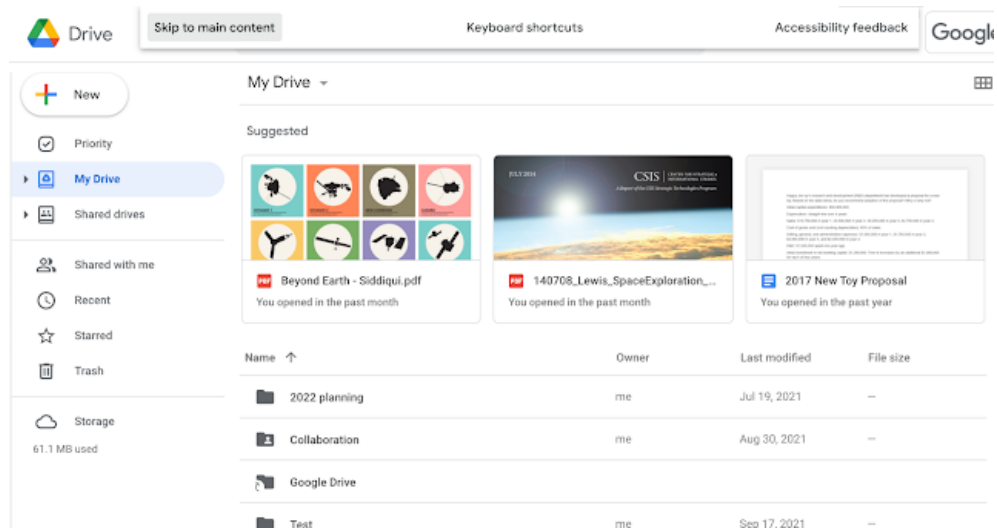


Figura 26 - vista página principal Google Drive

Este servicio ha permitido el intercambio de archivos entre los integrantes del grupo y la edición de estos de manera simultánea.

Nos ha permitido, además de almacenar documentos que compartíamos que no eran código (habrían ido a GitHub), compartir documentos con el tutor del trabajo de fin de grado.

3.3.4. Diagrams.net

Diagrams.net [15] es un software de dibujo de gráficos.

Permite crear diferentes formas y diseños o por el contrario utilizar una de las muchas formas prediseñadas de las que dispone.

Estas últimas se encuentran organizadas por categorías en función de su objetivo principal y son ideales para crear gráficos o diagramas. Cuenta con integración con servicios Cloud tales como Dropbox OneDrive, GitHub o Google Drive el cual ha sido empleado en este proyecto.

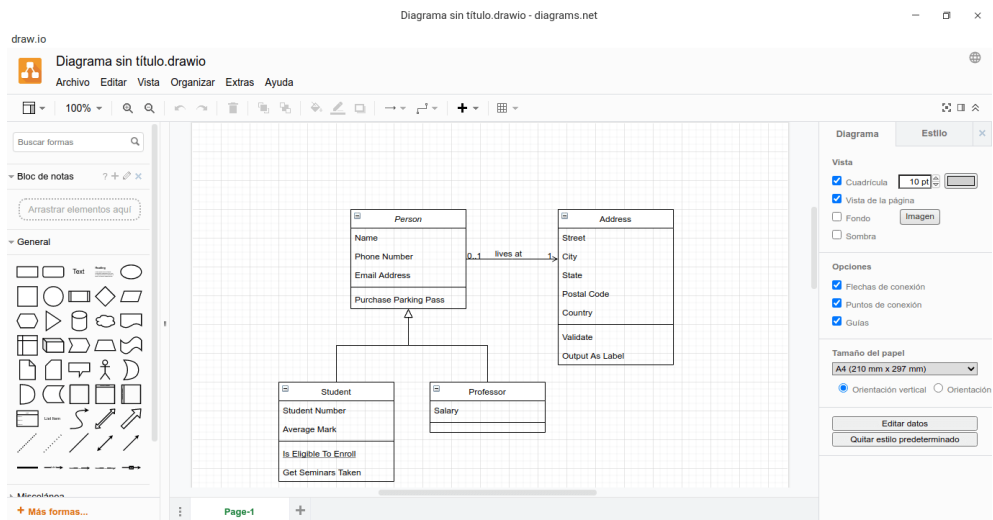


Figura 27 - vista principal diagrams.net

Ha sido utilizado tanto para diseñar casos de uso y modelos entidad relación, como para realizar esquemas y borradores con explicaciones entre los miembros del equipo.

En la figura 27 se muestra un ejemplo de diagrama hecho en la herramienta.

Capítulo 4. Casos de uso

En este capítulo, se describen los actores y los casos de uso definidos para la aplicación desarrollada.

4.1. Actores del sistema

Se han definido dos tipos de actores.

- **Usuario estudiante:** Representan a los usuarios registrados, los cuales podrán acceder a todas las funciones disponibles para los estudiantes. Las principales son: del módulo usuario (la creación de cuenta, inicio de sesión y gestión de su perfil), del módulo servicios (crear publicaciones de servicio, listarlas y aceptarlas), del módulo grupos (crear grupos, unirse a grupos ya creados, listarlos, acceder a repositorios) y por último del módulo foros (crear posts, responder a posts, listar subforos y posts)
- **Usuario administrador:** Representa al encargado de la aplicación y tendrá distintas funciones relacionadas con la gestión siendo las más importantes las de gestión de usuario (editar cualquier perfil), gestión de servicios (editado y eliminado de servicios), gestión de grupos (editado, listado y eliminado) y gestión de posts (editado y eliminado).
En la figura 19 se muestra el botón especial que solo tiene un actor usuario administrador para acceder a estas funcionalidades.

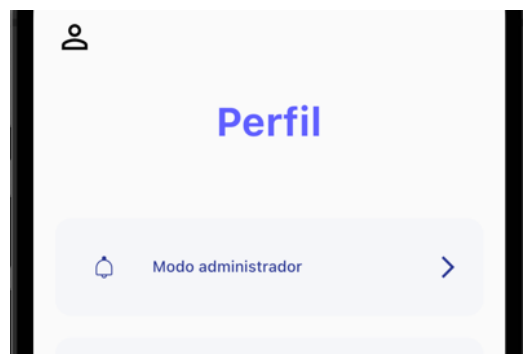


Figura 28 - Interfaz diferenciadora del usuario administrador

4.2. Descripción de los casos de uso.

En esta sección se van a describir los casos de uso definidos. Para ello se han agrupado en módulos de acuerdo con el tipo de actor.

4.2.1. Módulo estudiante

Al haber tanta funcionalidad en la figura 16, hemos subagrupado los casos de uso del estudiante en 4 funcionalidades principales, la gestión de perfil, la gestión de servicios, la gestión de grupos y la gestión de foros.

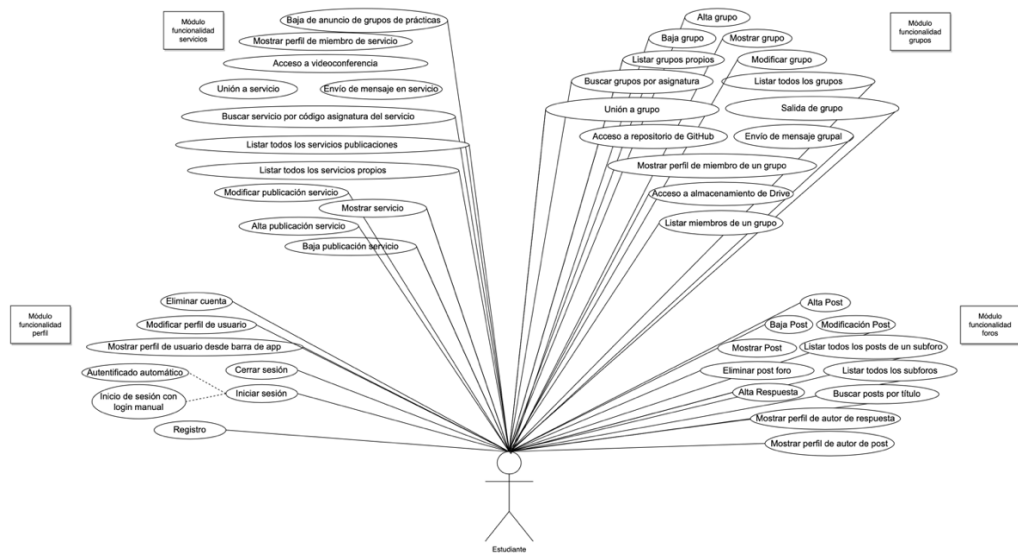


Figura 29 - Diagrama caso de uso módulo usuario estudiante

- **Módulo funcionalidad perfil**

En este módulo se recoge todo lo referido a la gestión del perfil de usuario.

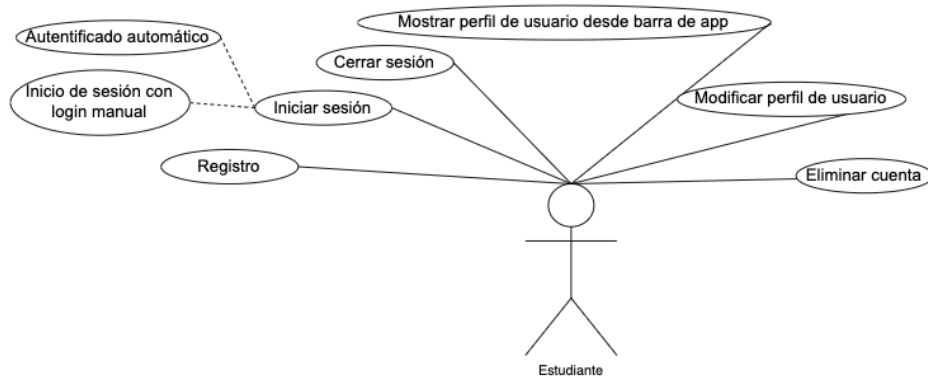


Figura 30 - Diagrama caso de uso módulo usuario estudiante funcionalidad perfil

E-01 Registro	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Registra a un nuevo usuario en la aplicación.
Actor	Estudiante.
Entrada	Nombre completo, alias, correo universitario, contraseña (y verificación de contraseña), y biografía.
Salida	Redirección a pantalla inicio o mensaje de error.
Precondición	El usuario no existe en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El sistema muestra una pantalla inicial con un botón “inicio de sesión” y un botón “registrarse”. 3. El usuario pulsa el botón “registrarse”. 4. El sistema redirige a la pantalla de registro. 5. El usuario introduce nombre completo, alias, correo universitario, contraseña (se pide dos veces para verificar), y biografía. 6. El usuario presiona el botón “registrarse”.
Postcondición si éxito	Se registra al usuario en la aplicación y se establece su rol como usuario no administrador. El sistema le redirige a la pantalla inicial.
Postcondición si fallo	El nuevo usuario no se registra y se le notifica el tipo de error.
Excepciones	<ol style="list-style-type: none"> 5.1. Las contraseñas no coinciden. 6.1. El usuario ya existe y se muestra un mensaje de error. 6.2. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	El usuario estudiante dado de alta no tiene rol de administrador. La contraseña y la verificación de la contraseña se ocultan con dots.

Tabla 1 - Registro

E-02 Inicio de sesión con login manual	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Media.
Descripción	Autentifica a un usuario ya registrado en la aplicación.
Actor	Estudiante.
Entrada	Correo universitario y contraseña.
Salida	Inicio de sesión correcto y redirección a perfil o mensaje de error.
Precondición	El usuario había sido registrado previamente en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa el botón inicio de sesión. 2. El sistema redirige a la pantalla de login. 3. El usuario introduce correo y contraseña. 4. El usuario presiona el botón de “Iniciar sesión”.
Postcondición si éxito	Se loggea correctamente al usuario en la aplicación y se le dirige a la página principal de su perfil.
Postcondición si fallo	El usuario no se loggea y no accede a la aplicación.
Excepciones	<ol style="list-style-type: none"> 4.1. El usuario no existe y se muestra un mensaje de error. 4.2. El usuario no ha introducido bien la contraseña y se muestra un mensaje de error. 3.2. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	<p>Si el usuario accede por primera vez a Inicio de sesión sin haberse registrado tendrá un enlace “¿Aún sin cuenta? Regístrate” para acceder a registro.</p> <p>La contraseña se oculta con dots.</p>

Tabla 2 - Inicio de sesión con login manual

E-03 Autenticado automático	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Media.
Descripción	Autentifica automáticamente a un usuario ya registrado en la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Redirección a perfil o redirección a pantalla de inicio.
Precondición	El usuario había sido registrado previamente en la base de datos y había hecho login al menos una vez en la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El sistema redirige a la pantalla de perfil del usuario.
Postcondición si éxito	Se loggea correctamente al usuario en la aplicación y se le dirige a la página principal de su perfil.
Postcondición si fallo	El usuario no se loggea y no accede a la aplicación directamente.
Excepciones	2.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Cuando el usuario inicia sesión por primera vez se guarda un token de sesión. Mientras no pulse “cerrar sesión” se mantiene activa y no tiene que volver a iniciarla.

Tabla 3 - Autenticado automático

E-05 Modificar perfil de usuario	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Modificar los datos del perfil de un usuario de la aplicación.
Actor	Estudiante.
Entrada	Datos nuevos (foto de perfil, nombre completo, nick y/o biografía).
Salida	Datos modificados correctamente y redirección correcta al módulo de gestión de usuario o mensaje de error.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa el botón home superior izquierdo. 2. El sistema redirige al módulo de gestión de usuario. 3. El usuario presiona el botón de “Modificar datos” y sobrescribe los campos que considera. 4. El usuario pulsa el botón “Guardar cambios” 5. El sistema redirige al usuario al módulo de gestión de usuarios.
Postcondición si éxito	Se modifican los datos del usuario en la aplicación y en la base de datos.
Postcondición si fallo	Los datos del usuario no se actualizan y se le notifica del error.
Excepciones	<ol style="list-style-type: none"> 4.1. El usuario borra y no rellena alguno de los campos y se muestra un mensaje de error. 4.2. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	Los campos input text cargan los datos actuales y reales del usuario almacenados en la base de datos, por lo que si no quiere cambiarlos basta con no tocarlos. Si se borran y se dejan vacío dará error puesto que los está intentando actualizar con datos vacíos.

Tabla 5 - Modificar perfil de usuario

E-06 Modificar datos login	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Modificar los datos de inicio de sesión de un usuario de la aplicación.
Actor	Estudiante.
Entrada	Datos nuevos (email, contraseña y verificación de contraseña).
Salida	Datos modificados correctamente y redirección correcta al módulo de gestión de usuario o mensaje de error.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario dentro de la pestaña de modificar datos del caso de uso E-05 pulsa el botón “Cambiar email/password”. 2. El sistema abre un modal (un pop-up) al usuario donde le permite introducir nuevos datos y sobrescribe los campos que considera. 3. El usuario pulsa el botón “Confirmar” y “Guardar cambios”. 4. El sistema redirige al usuario al módulo de gestión de usuarios.
Postcondición si éxito	Se modifican los datos del usuario en la aplicación y en la base de datos.
Postcondición si fallo	Los datos del usuario no se actualizan y se le notifica del error.
Excepciones	<ol style="list-style-type: none"> 3.1. El usuario borra y no rellena alguno de los campos y se muestra un mensaje de error. 3.2. El usuario no escribe correctamente la contraseña y verificación de contraseña, dando un mensaje de error al no coincidir. 3.3. Ocurre un error en el servidor y se muestra un mensaje de error.

Comentarios	Al igual que en el caso de uso de modificación de datos del perfil, los campos input text cargan el email y contraseña actual del usuario almacenados en la base de datos, por lo que si no quiere cambiarlos basta con no tocarlos. Si se borran y se dejan vacío dará error puesto que los está intentando actualizar con datos vacíos.
--------------------	---

Tabla 6 - Modificar datos login

E-07 Cerrar sesión	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Cierra la sesión del usuario actual de la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Cierre de sesión correcto y redirección a la pantalla de inicio de la aplicación.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa el botón home superior izquierdo. 2. El sistema redirige al módulo de gestión de usuario. 3. El usuario presiona el botón de “Log Out”. 4. El sistema redirige al usuario a la pantalla de inicio de la aplicación.
Postcondición si éxito	Se liberan los datos de sesión de la aplicación y se redirige al usuario a la pantalla inicial donde puede iniciar sesión o registrarse.
Postcondición si fallo	El usuario se mantiene logueado en la aplicación.
Excepciones	3.1. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	Después de hacer log out el usuario deja de acceder automáticamente a la aplicación.

Tabla 7 - Cerrar sesión

E-08 Eliminar cuenta	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Da de baja a un usuario de la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Cuenta eliminada correctamente y redirección a la pantalla de inicio de la aplicación o mensaje de error.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa el botón home superior izquierdo. 2. El sistema redirige al módulo de gestión de usuario. 3. El usuario presiona el botón de “Eliminar cuenta”. 4. El sistema muestra un modal (un pop-up) al usuario preguntando si está seguro de que desea borrar todos los datos. 5. El usuario pulsa confirmar. 6. El sistema redirige al usuario a la pantalla de inicio de la aplicación.
Postcondición si éxito	Se da de baja el usuario de la aplicación y se elimina de la base de datos. Se liberan los datos de sesión de la aplicación y se redirige al usuario a la pantalla inicial donde puede iniciar sesión o registrarse.
Postcondición si fallo	No se da al usuario de baja y se le notifica el error.
Excepciones	<p>3.1. Ocurre un error en servidor y se muestra un mensaje de error</p> <p>5.1. Ocurre un error en servidor y se muestra un mensaje de error</p>
Comentarios	

Tabla 8 - Eliminar cuenta

- **Módulo funcionalidad servicios**

En este módulo se recoge todo lo referido a la gestión de servicios del usuario.

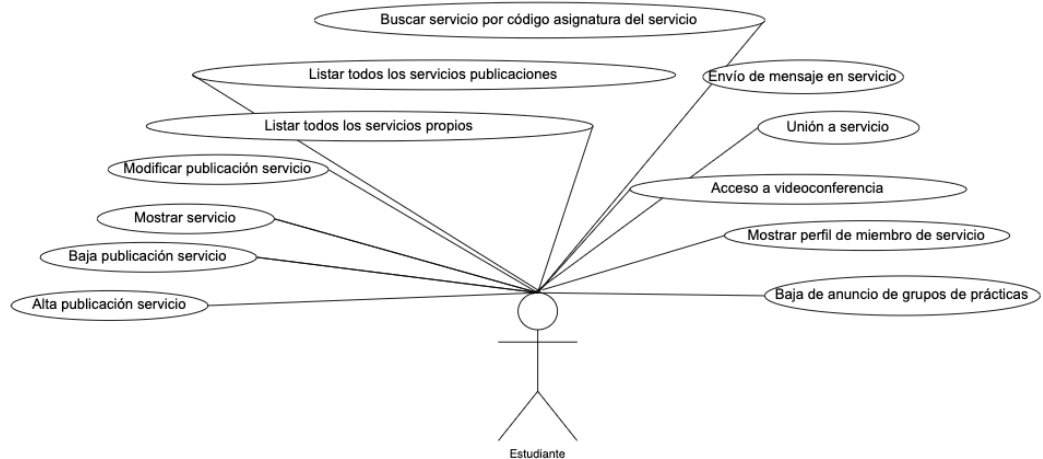


Figura 31 - Diagrama caso de uso módulo usuario estudiante funcionalidad servicios

S-01 Alta publicación servicio	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Da de alta una nueva publicación (oferta o petición) de servicio en la aplicación.
Actor	Estudiante.
Entrada	Código de la asignatura del servicio, descripción detallada del servicio a prestar, url de la conferencia online (a través de zoom).
Salida	Servicio creado correctamente y redirección a la pantalla de servicios o mensaje de error.
Precondición	El usuario está autenticado, y no existe otra publicación del mismo usuario con la misma descripción.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón servicios de la barra de navegación. 2. El sistema le redirige a la pantalla de servicios.

	<ol style="list-style-type: none"> 3. Presionar el botón de “Nueva publicación de servicio”. 4. El sistema le redirige a la pantalla nueva publicación de servicio. 5. El usuario rellena los campos de entrada (código de la asignatura, descripción detallada y url de la conferencia online). 6. El usuario pulsa sobre el botón “confirmar”. 7. El sistema da de alta la publicación y redirige al usuario a sus servicios (creados y participados) activos.
Postcondición si éxito	Se registra la publicación del servicio en la aplicación y se almacena en la base de datos.
Postcondición si fallo	La nueva publicación de servicio no se registra y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 6.1. La publicación de servicio ya existe y se muestra un mensaje de error 6.2. Ocurre un error en servidor y se muestra un mensaje de error 7.1. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	El servicio que se da de alta puede ser de tipo “ofrecer” o de tipo “pedir”, por eso se pide una descripción detallada.

Tabla 9 - Alta publicación servicio

S-02 Baja publicación servicio	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Da de baja una publicación de servicio en la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Eliminado del servicio en la base de datos y redirección del sistema a la lista de servicios propios (creados y unidos), o mensaje de error.
Precondición	El usuario está autenticado y existe una publicación de servicio del usuario activa para dar de baja.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón servicios de la barra de navegación. 2. El sistema le redirige a la pantalla de servicios propios. 3. El usuario accede al servicio creado por él mismo y pulsa el botón superior derecho de eliminado. 4. Se elimina el servicio y el sistema redirige al usuario a la página de todos sus servicios.
Postcondición si éxito	Se elimina de la base de datos el servicio y el usuario es redirigido a la página de sus servicios.
Postcondición si fallo	La publicación de servicio no se da de baja y se le notifica el error.
Excepciones	3.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Si el usuario da de baja un servicio activo (con otra persona prestando o recibiendo dicho servicio) también se da de baja para el otro estudiante.

Tabla 10 - Baja publicación servicio

S-04 Modificar publicación servicio	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Media.
Descripción	Modificar los datos del servicio publicado en la aplicación.
Actor	Estudiante.
Entrada	Datos nuevos (código del servicio, descripción detallada, dirección de la conferencia del servicio).
Salida	Redirección correcta al módulo de servicios propios o mensaje de error.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón servicios de la barra de navegación. 2. El sistema le redirige a la pantalla de servicios propios. 3. El usuario accede al servicio creado por él mismo y pulsa el botón superior derecho de editado y sobrescribe los campos que considera. 6. El usuario pulsa el botón "Editar servicio" 7. El sistema redirige al usuario al módulo de servicios propios.
Postcondición si éxito	Se modifican los datos del servicio en la aplicación y en la base de datos.
Postcondición si fallo	Los datos del servicio no se actualizan y se le notifica del error.
Excepciones	<ol style="list-style-type: none"> 6.1. El usuario borra y no rellena alguno de los campos y se muestra un mensaje de error. 6.2. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	Los campos input text cargan los datos actuales y reales del servicio almacenados en la base de datos, por lo que si no quiere cambiarlos basta con no tocarlos. Si se borran y se dejan vacío dará error puesto

	que los está intentando actualizar con datos vacíos.
--	--

Tabla 12 - Modificar publicación servicio

S-05 Listar todos los servicios propios	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Lista todas las publicaciones de servicio propias, tanto publicaciones de servicio propias como ofertas y peticiones de servicio obtenidas, del propio usuario.
Actor	Estudiante.
Entrada	-
Salida	Lista de servicios propios pedidos y ofertados.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón servicios de la barra de navegación. 2. El sistema le redirige a la pantalla de servicios. 3. El sistema muestra una lista de servicios propios pedidos y ofertados.
Postcondición si éxito	Se muestra una lista con servicios propios pedidos u ofertados.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 3.1. No existe ningún servicio que cumpla el criterio de búsqueda. 3.2. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 13 - Listar todos los servicios propios

S-06 Listar todos los servicios publicados	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Lista todas las publicaciones de servicio, tanto peticiones de servicio como ofertas de servicio, de cualquier usuario de la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Lista de servicios pedidos u ofertados.
Precondición	El usuario está autenticado y hay al menos 1 servicio publicado en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón servicios de la barra de navegación. 2. El sistema le redirige a la pantalla de servicios. 3. El usuario presiona el botón superior derecho de “Listado de todos los servicios”. 4. El sistema muestra una lista de servicios pedidos y ofertados.
Postcondición si éxito	Se muestra una lista con servicios pedidos u ofertados.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 3.1. No existe ningún servicio que cumpla el criterio de búsqueda. 3.2. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 14 - Listar todos los servicios publicados

S-08 Unión a servicio	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Confirma un servicio ofrecido o pedido por otro estudiante.
Actor	Estudiante.
Entrada	-
Salida	Unión al servicio o mensaje de error.
Precondición	El usuario está autenticado y al menos otro usuario ha dado de alta una publicación de servicio en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón servicios de la barra de navegación. 2. El sistema le redirige a la pantalla de servicios. 3. El usuario pulsa un servicio. 4. El sistema redirige al usuario a una pantalla que muestra todos los datos de un servicio 5. El usuario pulsa “Obtener”. 6. El sistema hace visible al usuario la conferencia del servicio y todos los datos de este servicio aceptado.
Postcondición si éxito	Se une el estudiante al servicio y se registra en la base de datos.
Postcondición si fallo	El usuario no se une al servicio y se le notifica con un mensaje de error.
Excepciones	<ol style="list-style-type: none"> 3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 16 - Unión a servicio

S-09 Envío de mensaje en servicio	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Contacta con el propietario de un servicio (o con el otro miembro en caso de ser propietario).
Actor	Estudiante.
Entrada	Mensaje de texto.
Salida	Mensaje registrado en la pantalla o mensaje de error.
Precondición	El usuario está autenticado y unido (o propietario) en un servicio donde registrar el mensaje.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón servicios de la barra de navegación. 2. El sistema le redirige a la pantalla de servicios. 3. El usuario pulsa un servicio. 4. El sistema redirige al usuario a una pantalla que muestra todos los datos de un servicio 5. El usuario pulsa “Chat”. 6. El sistema despliega todo el historial de chats de la aplicación. 7. El usuario pulsa “Nuevo mensaje”. 8. El sistema redirige al usuario a una pantalla donde rellenar el mensaje. 9. El usuario escribe el mensaje y pulsa “Añadir” para enviarlo. 10. El sistema devuelve al usuario a la pantalla de mensajes para ver todos los enviados (incluido el nuevo).
Postcondición si éxito	Se registra el mensaje y se almacena en la base de datos.
Postcondición si fallo	El mensaje no se envía y se muestra un error.

Excepciones	<p>3.1. Ocurre un error en el servidor y se muestra un mensaje de error.</p> <p>5.1. Ocurre un error en el servidor y se muestra un mensaje de error.</p> <p>7.1. Ocurre un error en servidor y se muestra un mensaje de error</p> <p>9.1. El usuario no rellena el mensaje y sale un mensaje de error para que rellene el campo.</p> <p>9.2. Ocurre un error en servidor y se muestra un mensaje de error</p>
Comentarios	

Tabla 17 - Envío de mensaje en servicio

S-10 Acceso a videoconferencia	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Accede al enlace de videoconferencia de un servicio en la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Redirección del sistema al usuario a la web de zoom donde tendrá lugar la conferencia, dentro del navegador integrado en la app.
Precondición	El usuario está autenticado y está unido, o es el publicador, a ese servicio.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón servicios de la barra de navegación. 2. El sistema le redirige a la pantalla de servicios. 3. El usuario pulsa un servicio. 4. El sistema redirige al usuario a una pantalla que muestra todos los datos de un servicio. 5. El usuario pulsa “Conference url”. 6. El sistema redirige al usuario a la web donde tiene lugar la conferencia.
Postcondición si éxito	Se redirige al usuario a la web donde tiene lugar la conferencia.
Postcondición si fallo	La web no carga y da error al usuario.
Excepciones	<ol style="list-style-type: none"> 3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Desde la versión iOS el navegador integrado en la aplicación permite tener la reunión sin salir de esta, aunque da la opción para abrir en el navegador web y en caso de tener la aplicación zoom instalada se abre el enlace ahí. Al ser una aplicación multiplataforma, al usarla desde un navegador web en un ordenador

	se podría tener la reunión desde el propio navegador también.
--	---

Tabla 18 - Acceso a videoconferencias

- **Módulo funcionalidad grupos**

En este módulo se recoge todo lo referido a la gestión de grupos del usuario.

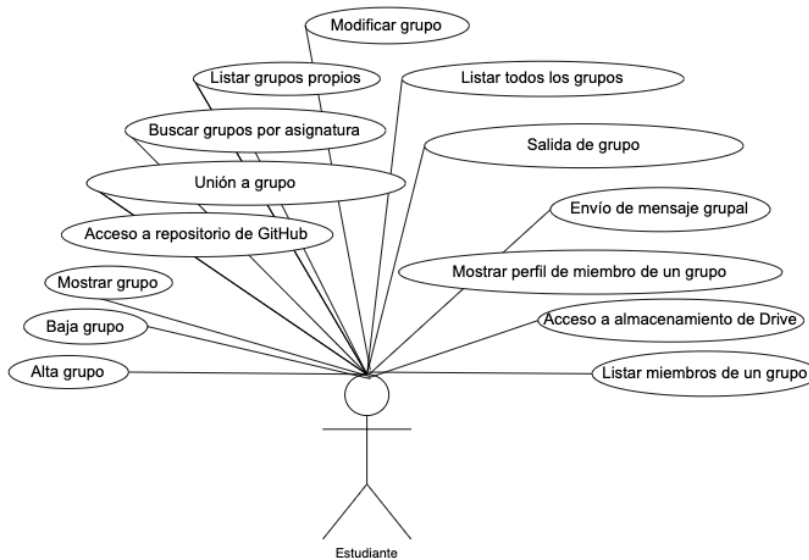


Figura 32 - Diagrama caso de uso módulo usuario estudiante funcionalidad grupo

G-01	Alta grupo
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Da de alta un nuevo grupo en la aplicación.
Actor	Estudiante.
Entrada	Código de la asignatura del grupo, tamaño del grupo, curso del grupo de la asignatura para la que se necesita un grupo, url del repositorio de GitHub donde se alojará el código, url del repositorio de Drive donde se alojarán otros documentos, descripción detallada de lo que se espera de ese grupo.
Salida	Grupo creado correctamente y redirección a la pantalla de servicios o mensaje de error.

Precondición	El usuario está autenticado, y no existe otro grupo del mismo usuario con la misma descripción en esa asignatura.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos propios. 3. Presionar el botón de “Nuevo grupo”. 4. El sistema le redirige a la pantalla nueva publicación de grupo. 5. El usuario rellena los campos de entrada (código de la asignatura, tamaño, curso de la asignatura, url de GitHub, url de Drive y descripción). 6. El usuario pulsa sobre el botón “Crear grupo”. 7. El sistema da de alta la publicación y redirige al usuario a sus servicios (creados y participados) activos.
Postcondición si éxito	Se registra la publicación del grupo en la aplicación y se almacena en la base de datos.
Postcondición si fallo	La nueva publicación del grupo no se registra y se notifica el error.
Excepciones	<ol style="list-style-type: none"> 6.1. La publicación de grupo ya existe y se muestra un mensaje de error. 6.2. No se rellenan todos los campos o se rellenan incorrectamente y se muestra mensaje de error. 6.2. Ocurre un error en servidor y se muestra un mensaje de error 7.1. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	El grupo que se da de alta puede ser de otras cosas que no sean de asignaturas, aunque están pensados para asignaturas y prácticas.

Tabla 20 - Alta grupo

G-02 Baja grupo	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Da de baja un grupo ya creado en la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Eliminado del grupo en la base de datos y redirección del sistema a la lista de grupos propios (creados y unidos), o mensaje de error.
Precondición	El usuario está autenticado y existe un grupo activo creado por el usuario para dar de baja.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos propios. 3. El usuario accede al grupo creado por él mismo y pulsa el botón superior derecho de eliminado. 4. Se elimina el grupo y el sistema redirige al usuario a la página de todos sus grupos.
Postcondición si éxito	Se elimina de la base de datos el grupo y el usuario es redirigido a la página de sus grupos.
Postcondición si fallo	El grupo no se da de baja y se le notifica el error.
Excepciones	3.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Si el usuario da de baja un grupo activo (con más miembros) también se da de baja para el resto de miembros, no solo para el creador.

Tabla 21 - Baja grupo

G-03 Mostrar grupo	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Muestra el grupo creado por un usuario de la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Redirección del sistema al usuario a la pantalla de descripción detallada de un grupo, o mensaje de error.
Precondición	El usuario está autenticado y hay al menos un grupo creado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupo de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El usuario pulsa un grupo. 4. El sistema redirige al usuario a una pantalla que muestra todos los datos de un grupo.
Postcondición si éxito	Se muestran los datos del grupo publicado en la aplicación almacenados en la base de datos, redirigiendo al usuario a una pantalla detallada del grupo.
Postcondición si fallo	Los datos del grupo no se muestran y se notifica al usuario del error.
Excepciones	4.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Desde esta pantalla además de ver datos como la descripción del grupo, la asignatura - código del grupo, se pueden realizar acciones como editar, eliminar, chatear, acceder al repositorio y al Drive, y ver una lista de todos los miembros actuales del grupo y sus perfiles.

Tabla 22 - Mostrar grupo

G-04 Modificar grupo	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Media.
Descripción	Modificar los datos del grupo publicado en la aplicación.
Actor	Estudiante.
Entrada	Datos nuevos (código de la asignatura del grupo, tamaño del grupo, curso del grupo de la asignatura para la que se necesita un grupo, url del repositorio de GitHub donde se alojará el código, url del repositorio de Drive donde se alojarán otros documentos, descripción detallada de lo que se espera de ese grupo).
Salida	Redirección correcta al módulo de grupos propios o mensaje de error.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos propios. 3. El usuario accede al grupo creado por él mismo y pulsa el botón superior derecho de editado y sobrescribe los campos que considera. 8. El usuario pulsa el botón “Editar grupo” 9. El sistema redirige al usuario al módulo de grupos propios.
Postcondición si éxito	Se modifican los datos del grupo en la aplicación y en la base de datos.
Postcondición si fallo	Los datos del grupo no se actualizan y se le notifica del error.
Excepciones	<ol style="list-style-type: none"> 6.1. El usuario borra y no rellena alguno de los campos y se muestra un mensaje de error. 6.2. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	Los campos input text cargan los datos actuales y reales del grupo almacenados en la base de datos, por lo que si no quiere cambiarlos

	basta con no tocarlos. Si se borran y se dejan vacío dará error puesto que los está intentando actualizar con datos vacíos.
--	---

Tabla 23 - Modificar grupo

G-05 Listar grupos propios	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Lista todos los grupos propios, tanto creados por el propio usuario como unidos.
Actor	Estudiante.
Entrada	-
Salida	Lista de grupos propios (grupos unidos y grupos creados).
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El sistema muestra una lista de grupos propios (grupos unidos y grupos creados).
Postcondición si éxito	Se muestra una lista con grupos propios pedidos u ofertados.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 3.1. No existe ningún grupo que cumpla el criterio de búsqueda. 3.2. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 24 - Listar grupos propios

G-06 Listar todos los grupos	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Lista todos los grupos, tanto los propios (creados y unidos), como los ajenos y sin unir.
Actor	Estudiante.
Entrada	-
Salida	Lista de todos los grupos.
Precondición	El usuario está autenticado y hay al menos 1 grupo publicado en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El usuario presiona el botón superior derecho de “Listado de todos los grupos”. 4. El sistema muestra una lista de todos los grupos.
Postcondición si éxito	Se muestra una lista con todos los grupos.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 3.1. No existe ningún grupo que cumpla el criterio de búsqueda. 3.2. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 25 - Listar todos los grupos

G-07 Buscar grupos por código de asignatura	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Busca por código de asignatura grupos creados.
Actor	Estudiante.
Entrada	Código del grupo que se busca.
Salida	Lista de grupos que cumplen con la condición de búsqueda (el código)..
Precondición	El usuario está autenticado y hay al menos 1 grupo dado de alta en la base de datos de la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El usuario presiona el botón superior derecho de “Listado de todos los grupos”. 4. El sistema le redirige a la pantalla de todos los grupos ofertados. 5. El usuario presiona el botón de “Búsqueda de grupos” e introduce los datos de búsqueda. 6. El sistema muestra una lista de grupos filtrada por el código de asignatura introducido.
Postcondición si éxito	Se muestra una lista con grupos que cumplen los criterios de búsqueda.
Postcondición si fallo	No se carga la página de búsqueda y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 4.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. No existe ningún grupo que cumpla el criterio de búsqueda. 5.2. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 26 - Buscar grupos por código de asignatura

G-08 Unión a grupo	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	El usuario se une a un grupo (que no ha creado él).
Actor	Estudiante.
Entrada	-
Salida	Unión al grupo o mensaje de error.
Precondición	El usuario está autenticado y al menos otro usuario ha dado de alta un grupo en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El usuario presiona el botón superior derecho de “Listado de todos los grupos”. 4. El sistema le redirige a la pantalla de todos los grupos ofertados. 5. El usuario pulsa un grupo. 6. El sistema redirige al usuario a una pantalla que muestra todos los datos de un grupo. 7. El usuario pulsa “Únete”. 8. El sistema hace visible al usuario los enlaces de GitHub y Drive, el chat del grupo y un botón “Salir del grupo”.
Postcondición si éxito	Se une el estudiante al grupo y se registra en la base de datos.
Postcondición si fallo	El usuario no se une al grupo y se le notifica con un mensaje de error.
Excepciones	<ol style="list-style-type: none"> 3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error. 7.1. Ocurre un error en el servidor y se muestra un mensaje de error.

Comentarios	
--------------------	--

Tabla 27- Unión a grupo

G-09 Salida de grupo	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	El usuario sale de un grupo (al que previamente se había unido y del que no es el dueño).
Actor	Estudiante.
Entrada	-
Salida	Salida del grupo o mensaje de error.
Precondición	El usuario está autenticado y al menos otro usuario ha dado de alta un grupo en la base de datos al cual está unido.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El usuario pulsa un grupo y pulsa “Salir del grupo”. 4. El sistema redirige al usuario a la pantalla de grupos (con un grupo menos).
Postcondición si éxito	Se une el estudiante al grupo y se registra en la base de datos.
Postcondición si fallo	El usuario no se une al grupo y se le notifica con un mensaje de error.
Excepciones	3.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 28 - Salida de grupo

G-10 Envío de mensaje grupal	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Contacta con el propietario de un grupo (o con el otro miembro en caso de ser propietario).
Actor	Estudiante.
Entrada	Mensaje de texto.
Salida	Mensaje registrado en la pantalla o mensaje de error.
Precondición	El usuario está autenticado y unido (o propietario) en un grupo donde registrar el mensaje.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El usuario pulsa un grupo. 4. El sistema redirige al usuario a una pantalla que muestra todos los datos de un grupo. 5. El usuario pulsa “Chat de grupo”. 6. El sistema despliega todo el historial de chats de la aplicación. 7. El usuario pulsa “Nuevo mensaje”. 8. El sistema redirige al usuario a una pantalla donde rellenar el mensaje. 9. El usuario escribe el mensaje y pulsa “Añadir” para enviarlo. 10. El sistema devuelve al usuario a la pantalla de mensajes para ver todos los enviados en el grupo (incluido el nuevo).
Postcondición si éxito	Se registra el mensaje y se almacena en la base de datos.
Postcondición si fallo	El mensaje no se envía y se muestra un error.

<p>Excepciones</p>	<p>3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error. 7.1. Ocurre un error en servidor y se muestra un mensaje de error 9.1. El usuario no rellena el mensaje y sale un mensaje de error para que rellene el campo. 9.2. Ocurre un error en servidor y se muestra un mensaje de error</p>
<p>Comentarios</p>	

Tabla 29 - Envío de mensaje grupal

G-11 Acceso a repositorio de Github	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Accede al enlace de GitHub de un grupo en la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Redirección del sistema al usuario a la web de GitHub donde tendrá lugar el repositorio, dentro del navegador integrado en la app.
Precondición	El usuario está autenticado y está unido o es el creador de un grupo.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El usuario pulsa un grupo. 4. El sistema redirige al usuario a una pantalla que muestra todos los datos de un grupo. 5. El usuario pulsa “GitHub repo”. 6. El sistema redirige al usuario a la web donde tiene lugar el repositorio.
Postcondición si éxito	Se redirige al usuario a la web donde tiene lugar el repositorio.
Postcondición si fallo	La web no carga y da error al usuario.
Excepciones	<ol style="list-style-type: none"> 3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Desde la versión iOS el navegador integrado en la aplicación permite tener acceso al repositorio sin salir de esta, aunque da la opción para abrir en el navegador web y en caso de tener la aplicación de GitHub instalada se abre el enlace ahí.

Tabla 30 - Acceso a repositorio de Github

G-12 Acceso a almacenamiento de Drive	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Accede al enlace de Drive de un grupo en la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Redirección del sistema al usuario a la web de Drive donde tendrá lugar el repositorio de archivos variados, dentro del navegador integrado en la app.
Precondición	El usuario está autenticado y está unido o es el creador de un grupo.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El usuario pulsa un grupo. 4. El sistema redirige al usuario a una pantalla que muestra todos los datos de un grupo. 5. El usuario pulsa "Drive storage". 6. El sistema redirige al usuario a la web donde tiene lugar el almacenamiento de Drive.
Postcondición si éxito	Se redirige al usuario a la web donde tiene lugar el repositorio.
Postcondición si fallo	La web no carga y da error al usuario.
Excepciones	<ol style="list-style-type: none"> 3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error.

Comentarios	Desde la versión iOS el navegador integrado en la aplicación permite tener acceso al repositorio de Drive sin salir de esta, aunque da la opción para abrir en el navegador web y en caso de tener la aplicación de Drive instalada se abre el enlace ahí, que permite hacer más acciones que la versión web.
--------------------	---

Tabla 31 - Acceso a almacenamiento de Drive

G-13 Listar miembros de un grupo	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Lista todos los miembros de un grupo, los ya unidos.
Actor	Estudiante.
Entrada	-
Salida	Lista de todos los miembros del grupo.
Precondición	El usuario está autenticado y hay al menos 1 grupo publicado en la base de datos del que es miembro.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El usuario pulsa un grupo. 4. El sistema redirige al usuario a una pantalla que muestra todos los datos de un grupo. 5. El usuario pulsa sobre el número de miembros del grupo. 6. El sistema redirige al usuario a la pantalla de lista de todos los miembros del grupo.
Postcondición si éxito	Se muestra una lista con todos los miembros del grupo.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.

Excepciones	<p>3.1. No existe ningún grupo que cumpla el criterio de búsqueda.</p> <p>5.1 Ocurre un error en el servidor y se muestra un mensaje de error.</p>
Comentarios	En realidad, en grupos propios recién creados también funciona, la lista muestra al creador solamente (al usuario que lo ha creado).

Tabla 32 - Listar miembros de un grupo

G-14 Mostrar perfil de miembro del grupo	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Muestra los datos del perfil de un miembro del grupo (propietario o unido) de la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Mostrado de datos correcto y redirección del sistema al usuario a la pantalla de perfil o mensaje de error.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón grupos de la barra de navegación. 2. El sistema le redirige a la pantalla de grupos. 3. El usuario pulsa un grupo. 4. El sistema redirige al usuario a una pantalla que muestra todos los datos de un grupo. 5. El usuario pulsa sobre el número de miembros del grupo. 6. El sistema redirige al usuario a la pantalla de lista de todos los miembros del grupo. 7. El usuario pulsa sobre el nombre de uno de ellos. 8. El sistema redirige al usuario al perfil de ese usuario pulsado.

Postcondición si éxito	El sistema muestra al usuario nombre, biografía y correo además de su participación en la app.
Postcondición si fallo	Los datos del usuario no se muestran y se le notifica del error.
Excepciones	<p>3.1. Ocurre un error en el servidor y se muestra un mensaje de error.</p> <p>5.1. Ocurre un error en el servidor y se muestra un mensaje de error.</p> <p>7.1. Ocurre un error en el servidor y se muestra un mensaje de error.</p>
Comentarios	Con esto un usuario puede ver con quien está a punto de formar grupo.

Tabla 33 - Mostrar perfil de miembro del grupo

- **Módulo funcionalidad foros**

En este módulo se recoge todo lo referido a la gestión de foros del usuario.

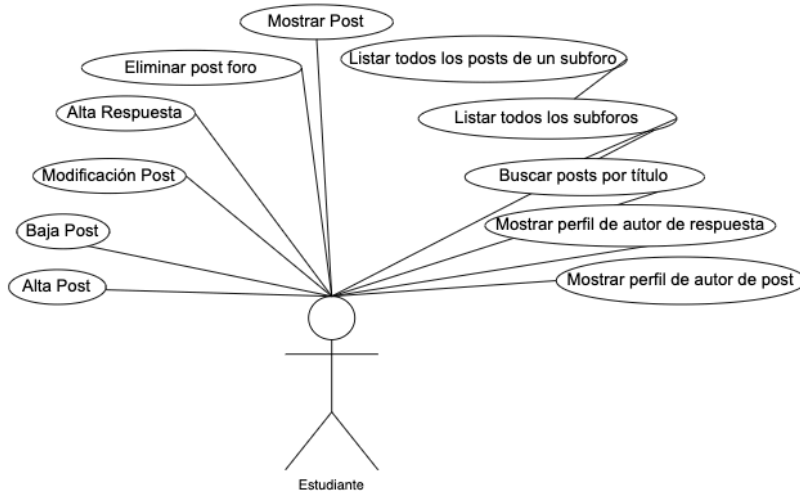


Figura 33 - Diagrama caso de uso módulo usuario estudiante funcionalidad foros

F-01	Alta post
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Da de alta una nueva publicación (post) dentro de un subforo.
Actor	Estudiante.
Entrada	Título y cuerpo del post.
Salida	Post creado correctamente y redirección a la pantalla de posts del mismo subforo, o mensaje de error.
Precondición	El usuario está autenticado, y no existe otra publicación (post) del mismo usuario con el mismo título.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón foros de la barra de navegación. 2. El sistema le redirige a la pantalla de foros. 3. El usuario selecciona el subforo que desea. 4. El sistema le redirige a la pantalla de todos los posts

	<p>de este subforo</p> <ol style="list-style-type: none"> 5. El usuario pulsa el botón de nueva publicación. 6. El sistema le redirige a la pantalla nueva publicación. 7. El usuario rellena los campos de entrada (título del post y cuerpo (texto) del post). 8. El usuario pulsa sobre el botón “Crear post”. 9. El sistema da de alta la publicación y redirige al usuario los posts del subforo.
Postcondición si éxito	Se registra la publicación del post en la aplicación y se almacena en la base de datos, en los subforos.
Postcondición si fallo	La nueva publicación de post no se registra y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 7.1. La publicación de post ya existe y se muestra un mensaje de error. 7.2. Ocurre un error en el servidor y se muestra un mensaje de error. 8.1. El usuario no rellena todos los datos y da error. 8.2. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 34 - Alta post

Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Da de baja un post de la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Eliminado de la publicación en la base de datos y redirección del sistema a la lista de posts del subforo, o mensaje de error.
Precondición	El usuario está autenticado y existe una publicación del usuario activa para dar de baja.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón foros de la barra de navegación. 2. El sistema le redirige a la pantalla de foros. 3. El usuario selecciona el subforo que desea. 4. El sistema le redirige a la pantalla de todos los posts de este subforo. 5. El usuario accede al post creado por él mismo y pulsa el botón superior derecho de eliminado. 6. Se elimina el post y el sistema redirige al usuario a la página de todos los posts del subforo.
Postcondición si éxito	Se elimina de la base de datos el post y el usuario es redirigido a la página de otros posts de este subforo.
Postcondición si fallo	La publicación no se da de baja y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 35 - Baja Post

Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Muestra la publicación de un usuario de la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Redirección del sistema al usuario a la pantalla de descripción detallada de un hilo (o post), o mensaje de error.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón foros de la barra de navegación. 2. El sistema le redirige a la pantalla de foros. 3. El usuario selecciona el subforo que desea. 4. El sistema le redirige a la pantalla de todos los posts de este subforo. 5. El usuario accede al post. 6. El sistema redirige al usuario a una pantalla que muestra todos los datos de un post.
Postcondición si éxito	Se muestran los datos del post publicado en la aplicación almacenados en la base de datos, redirigiendo al usuario a una pantalla detallada del post.
Postcondición si fallo	Los datos del post no se muestran y se notifica al usuario del error.
Excepciones	5.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Desde esta pantalla además de ver datos como si el post tiene respuestas, fecha de las respuestas, autores e incluso los perfiles de esos autores.

Tabla 36 - Mostrar post

F-04 Modificación Post	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Media.
Descripción	Modificar los datos del post publicado en la aplicación.
Actor	Estudiante.
Entrada	Datos nuevos (título nuevo, cuerpo del post nuevo).
Salida	Redirección correcta al listado de posts del subforo o mensaje de error.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón foros de la barra de navegación. 2. El sistema le redirige a la pantalla de foros. 3. El usuario selecciona el subforo que desea. 4. El sistema le redirige a la pantalla de todos los posts de este subforo. 5. El usuario accede al post y pulsa el botón superior derecho de editado y sobrescribe los campos que considera. 6. El usuario pulsa el botón “Editar post” 7. El sistema redirige al usuario a una pantalla que muestra todos los posts del mismo subforo, con este editado.
Postcondición si éxito	Se modifican los datos del post en la aplicación y en la base de datos.
Postcondición si fallo	Los datos del post no se actualizan y se le notifica del error.
Excepciones	<ol style="list-style-type: none"> 5.1. El usuario borra y no rellena alguno de los campos y se muestra un mensaje de error. 6.1. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	Los campos input text cargan los datos actuales y reales del post almacenados en la base de datos, por lo que si no quiere cambiarlos basta con no tocarlos. Si se borran y se dejan vacío dará error puesto

	que los está intentando actualizar con datos vacíos.
--	--

Tabla 37 - Modificación Post

F-05 Listar todos los posts de un subforo	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Lista todas las publicaciones de un subforo determinado.
Actor	Estudiante.
Entrada	-
Salida	Lista de posts de un subforo..
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón foros de la barra de navegación. 2. El sistema le redirige a la pantalla de foros. 3. El usuario selecciona el subforo que desea. 4. El sistema le redirige a la pantalla de todos los posts de este subforo.
Postcondición si éxito	Se muestra una lista con posts.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 3.1. No existe ningún post en este subforo. 3.2. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 38 - Listar todos los posts de un subforo

F-06 Buscar posts por título	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Busca por título todos los posts (da igual el subforo).
Actor	Estudiante.
Entrada	Título del post que se busca.
Salida	Lista de posts que cumplen con la condición de búsqueda (el código)..
Precondición	El usuario está autenticado y hay al menos 1 post dado de alta en la base de datos de la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón foros de la barra de navegación. 2. El sistema le redirige a la pantalla de foros. 3. El usuario selecciona el subforo que desea. 4. El sistema le redirige a la pantalla de todos los posts de este subforo. 5. El usuario presiona el botón de “Búsqueda de posts” e introduce los datos de búsqueda. 6. El sistema muestra una lista de posts filtrada por el título del post.
Postcondición si éxito	Se muestra una lista con posts que cumplen los criterios de búsqueda.
Postcondición si fallo	No se carga la página de búsqueda y se le notifica el error.
Excepciones	<p>3.1. Ocurre un error en el servidor y se muestra un mensaje de error.</p> <p>5.1. No existe ningún post que cumpla el criterio de búsqueda.</p> <p>5.2. Ocurre un error en el servidor y se muestra un mensaje de error.</p>
Comentarios	Se podría sacar fuera de los subforos el botón de búsqueda, puesto la búsqueda es común a todos los subforos.

Tabla 39 - Buscar posts por título

F-07		Alta respuesta	
Versión	1.0.0		
Prioridad	Alta.		
Estabilidad	Alta.		
Descripción	Da de alta una nueva respuesta a un post.		
Actor	Estudiante.		
Entrada	Texto de la respuesta.		
Salida	Respuesta al post creada correctamente y redirección a la pantalla del post, o mensaje de error.		
Precondición	El usuario está autenticado, y existe el post sobre el que se quiere crear una respuesta.		
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón foros de la barra de navegación. 2. El sistema le redirige a la pantalla de foros. 3. El usuario selecciona el subforo que desea. 4. El sistema le redirige a la pantalla de todos los posts de este subforo. 5. El usuario pulsa el post al que desea contestar y pulsa “Responder” 6. El sistema le redirige a la pantalla de nueva respuesta. 7. El usuario rellena el texto de la nueva respuesta. 8. El usuario pulsa sobre el botón “Responder”. 9. El sistema da de alta la respuesta y redirige al usuario al post. 		
Postcondición si éxito	Se registra la respuesta del post en la aplicación y se almacena en la base de datos.		
Postcondición si fallo	La nueva publicación de respuesta del post no se registra y se le notifica el error.		
Excepciones	<ol style="list-style-type: none"> 7.1. El usuario no rellena todos los datos y da error. 7.2. Ocurre un error en el servidor y se muestra un mensaje de error. 		
Comentarios			

Tabla 40 - Alta respuesta

F-10 Listar todos los subforos	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Lista todos los subforos del foro de la aplicación.
Actor	Estudiante.
Entrada	-
Salida	Lista de subforos.
Precondición	El usuario está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y pulsa sobre el botón foros de la barra de navegación. 2. El sistema le redirige a la pantalla de foros y le muestra una lista interactiva de todos los subforos.
Postcondición si éxito	Se muestra una lista con subforos.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 1.1. Ocurre un error en el servidor y se muestra un mensaje de error. 2.1. No existe ningún subforo. 2.2. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	

Tabla 43 - Listar todos los subforos

4.2.2. Módulo administrador

En esta sección se muestran los casos de uso asociados al usuario administrador. En la figura 21 se puede observar el diagrama de casos de uso del usuario.

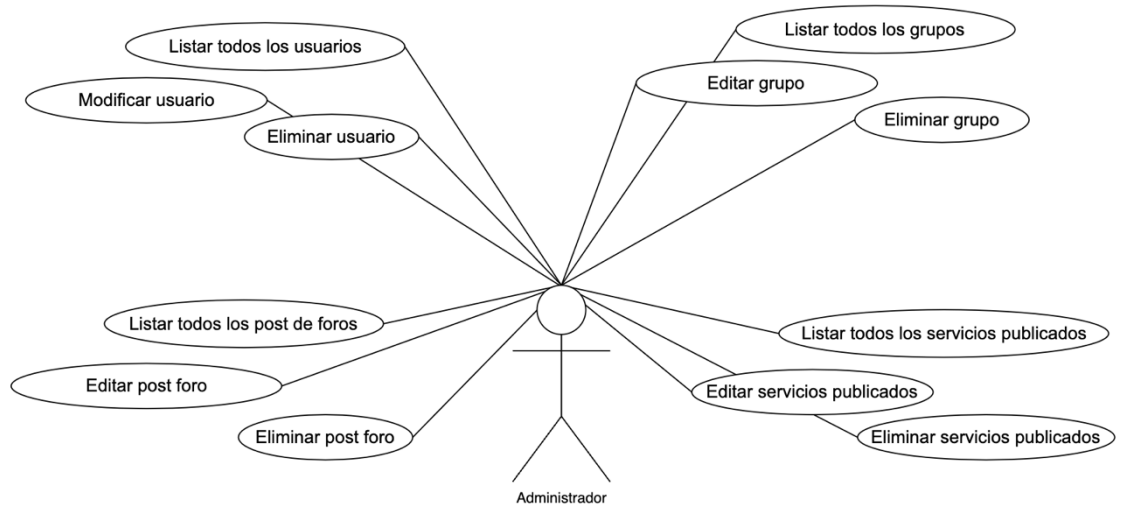


Figura 34 - Diagrama de casos de uso de administrador

A-01 Listar todos los usuarios	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Lista todos los usuarios de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Administrador.
Entrada	-
Salida	Todos los usuarios listados.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuario (perfil). 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo usuarios. 6. El sistema redirige al usuario a la pantalla de lista de todos los usuarios de la aplicación.
Postcondición si éxito	Se muestra una lista con todos los usuarios activos en la aplicación.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Sería interesante notificar de cambios a los usuarios una vez hechos.

Tabla 44 - Listar todos los usuarios

A-02 Modificar usuario	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Media.
Descripción	Modificar los datos del perfil de un usuario de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Estudiante.
Entrada	Datos nuevos (foto de perfil, nombre completo, nick y/o biografía).
Salida	Datos modificados correctamente y redirección a la lista de todos los usuarios para administrar o mensaje de error.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuario (perfil). 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo usuarios. 6. El sistema redirige al usuario a la pantalla de lista de todos los usuarios de la aplicación. 7. El usuario presiona el botón de “Modificar datos”, sobrescribiendo los campos que considera. 8. El usuario pulsa el botón “Guardar cambios” 9. El sistema redirige al usuario a la pantalla de lista de todos los usuarios de la aplicación.
Postcondición si éxito	Se modifican los datos del usuario en la aplicación y en la base de datos.
Postcondición si fallo	Los datos del usuario no se actualizan y se le notifica del error.

Excepciones	<p>10.1. El administrador borra y no rellena alguno de los campos y se muestra un mensaje de error.</p> <p>10.2. Ocurre un error en servidor y se muestra un mensaje de error</p>
Comentarios	<p>Los campos input text cargan los datos actuales y reales del usuario almacenados en la base de datos, por lo que si no quiere cambiarlos basta con no tocarlos. Si se borran y se dejan vacío dará error puesto que los está intentando actualizar con datos vacíos.</p>

Tabla 45 - Modificar usuario

A-03 Eliminar usuario	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Da de baja a un usuario de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Estudiante.
Entrada	-
Salida	Cuenta eliminada correctamente y redirección a la lista de todos los usuarios para administrar o mensaje de error.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuario (perfil). 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo usuarios. 6. El sistema redirige al usuario a la pantalla de lista de todos los usuarios de la aplicación. 7. El usuario presiona el botón de “Eliminar cuenta” en un usuario de la lista. 8. El sistema muestra un modal (un pop-up) al usuario preguntando si está seguro de que desea borrar todos los datos. 9. El usuario pulsa confirmar. 10. El sistema redirige al usuario a la pantalla de lista de todos los usuarios de la aplicación.
Postcondición si éxito	Se da de baja el usuario de la aplicación y se elimina de la base de datos.
Postcondición si fallo	No se da al usuario de baja y se le notifica el error.

Excepciones	5.1. El administrador borra y no rellena alguno de los campos y se muestra un mensaje de error. 9.1. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	

Tabla 46 - Eliminar usuario

A-04 Listar todos los posts de los foros	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Lista todos los posts de foros de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Administrador.
Entrada	-
Salida	Todos los posts listados.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuarios (perfil). 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo posts (foros). 6. El sistema redirige al usuario a la pantalla de lista de todos los posts de la aplicación.
Postcondición si éxito	Se muestra una lista con todos los posts activos en la aplicación.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Sería interesante notificar de cambios a los usuarios una vez hechos.

Tabla 47 - Listar todos los posts de los foros

A-05 Editar post foro	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Media.
Descripción	Modificar los datos de un post (de un foro) de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Estudiante.
Entrada	Datos nuevos (título y cuerpo).
Salida	Datos modificados correctamente y redirección a la lista de todos los posts para administrar o mensaje de error.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuario (perfil). 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo foros. 6. El sistema redirige al usuario a la pantalla de lista de todos los posts de la aplicación. 7. El usuario presiona el botón de “Modificar datos”, sobrescribiendo los campos que considera. 8. El usuario pulsa el botón “Guardar cambios” 9. El sistema redirige al usuario a la pantalla de lista de todos los posts (foros) de la aplicación.
Postcondición si éxito	Se modifican los datos del post en la aplicación y en la base de datos.
Postcondición si fallo	Los datos del post no se actualizan y se le notifica del error.

Excepciones	<p>10.1. El administrador borra y no rellena alguno de los campos y se muestra un mensaje de error.</p> <p>10.2. Ocurre un error en servidor y se muestra un mensaje de error</p>
Comentarios	<p>Los campos input text cargan los datos actuales y reales del post almacenados en la base de datos, por lo que si no quiere cambiarlos basta con no tocarlos. Si se borran y se dejan vacío dará error puesto que los está intentando actualizar con datos vacíos.</p>

Tabla 48 - Editar post foro

A-06 Eliminar post foro	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Da de baja un post de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Estudiante.
Entrada	-
Salida	Cuenta eliminada correctamente y redirección a la lista de todos los posts para administrar o mensaje de error.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuario (perfil). 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo posts. 6. El sistema redirige al usuario a la pantalla de lista de todos los posts de la aplicación. 7. El usuario presiona el botón de “Eliminar post” en un post de la lista. 8. El sistema muestra un modal (un pop-up) al usuario preguntando si está seguro de que desea borrar todos los datos. 9. El usuario pulsa confirmar. 11. El sistema redirige al usuario a la pantalla de lista de todos los posts de la aplicación.
Postcondición si éxito	Se da de baja el post de la aplicación y se elimina de la base de datos.
Postcondición si fallo	No se da al post de baja y se le notifica el error.

Excepciones	5.1. El administrador borra y no rellena alguno de los campos y se muestra un mensaje de error. 9.1. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	

Tabla 49 - Eliminar post foro

A-07 Listar todos los grupos	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Lista todos los grupos creados de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Administrador.
Entrada	-
Salida	Todos los grupos listados.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuarios. 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo grupos. 6. El sistema redirige al usuario a la pantalla de lista de todos los grupos de la aplicación.
Postcondición si éxito	Se muestra una lista con todos los grupos activos en la aplicación.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Sería interesante notificar de cambios a los usuarios una vez hechos cambios en los grupos a los que pertenecen.

Tabla 50 - Listar todos los grupos

A-08 Editar grupo	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Media.
Descripción	Modificar los datos de un grupo de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Estudiante.
Entrada	Datos nuevos (código de la asignatura del grupo, tamaño del grupo, curso del grupo de la asignatura para la que se necesita un grupo, url del repositorio de GitHub donde se alojará el código, url del repositorio de Drive donde se alojarán otros documentos, descripción detallada de lo que se espera de ese grupo).
Salida	Datos modificados correctamente y redirección a la lista de todos los grupos para administrar o mensaje de error.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuario (perfil). 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo foros. 6. El sistema redirige al usuario a la pantalla de lista de todos los grupos de la aplicación. 7. El usuario presiona el botón de “Modificar datos”, sobrescribiendo los campos que considera. 8. El usuario pulsa el botón “Guardar cambios” 9. El sistema redirige al usuario a la pantalla de lista de todos los grupos de la aplicación.
Postcondición si éxito	Se modifican los datos del grupo en la aplicación y en la base de datos.
Postcondición si fallo	Los datos del grupo no se actualizan y se le notifica del error.

Excepciones	<p>10.1. El administrador borra y no rellena alguno de los campos y se muestra un mensaje de error.</p> <p>10.2. Ocurre un error en servidor y se muestra un mensaje de error</p>
Comentarios	<p>Los campos input text cargan los datos actuales y reales del grupo almacenados en la base de datos, por lo que si no quiere cambiarlos basta con no tocarlos. Si se borran y se dejan vacío dará error puesto que los está intentando actualizar con datos vacíos.</p>

Tabla 51 - Editar grupo

A-09 Eliminar grupo	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Da de baja un grupo de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Estudiante.
Entrada	-
Salida	Cuenta eliminada correctamente y redirección a la lista de todos los grupos para administrar o mensaje de error.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuario (perfil). 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo grupos. 6. El sistema redirige al usuario a la pantalla de lista de todos los grupos de la aplicación. 7. El usuario presiona el botón de “Eliminar grupo” en un grupo de la lista. 8. El sistema muestra un modal (un pop-up) al usuario preguntando si está seguro de que desea borrar todos los datos. 9. El usuario pulsa confirmar. 12. El sistema redirige al usuario a la pantalla de lista de todos los grupos de la aplicación.
Postcondición si éxito	Se da de baja el grupo de la aplicación y se elimina de la base de datos.
Postcondición si fallo	No se da al grupo de baja y se le notifica el error.

Excepciones	5.1. El administrador borra y no rellena alguno de los campos y se muestra un mensaje de error. 9.1. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	

Tabla 52 - Eliminar grupo

A-10 Listar todos los servicios	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Lista todos los servicios creados de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Administrador.
Entrada	-
Salida	Todos los servicios listados.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuarios. 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo servicios. 6. El sistema redirige al usuario a la pantalla de lista de todos los servicios de la aplicación.
Postcondición si éxito	Se muestra una lista con todos los servicios activos en la aplicación.
Postcondición si fallo	No se carga la página de listado y se le notifica el error.
Excepciones	<ol style="list-style-type: none"> 3.1. Ocurre un error en el servidor y se muestra un mensaje de error. 5.1. Ocurre un error en el servidor y se muestra un mensaje de error.
Comentarios	Sería interesante notificar de cambios a los usuarios una vez hechos cambios en los servicios a los que pertenecen.

Tabla 53 - Listar todos los servicios

A-11 Editar servicio	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Media.
Descripción	Modificar los datos de un post (de un foro) de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Estudiante.
Entrada	Datos nuevos (código del servicio, descripción detallada, dirección de la conferencia del servicio).
Salida	Datos modificados correctamente y redirección a la lista de todos los servicios para administrar o mensaje de error.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuario (perfil). 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo foros. 6. El sistema redirige al usuario a la pantalla de lista de todos los servicios de la aplicación. 7. El usuario presiona el botón de “Modificar datos”, sobrescribiendo los campos que considera. 8. El usuario pulsa el botón “Guardar cambios” 9. El sistema redirige al usuario a la pantalla de lista de todos los servicios de la aplicación.
Postcondición si éxito	Se modifican los datos del servicio en la aplicación y en la base de datos.
Postcondición si fallo	Los datos del servicio no se actualizan y se le notifica del error.

Excepciones	<p>10.1. El administrador borra y no rellena alguno de los campos y se muestra un mensaje de error.</p> <p>10.2. Ocurre un error en servidor y se muestra un mensaje de error</p>
Comentarios	<p>Los campos input text cargan los datos actuales y reales del servicio almacenados en la base de datos, por lo que si no quiere cambiarlos basta con no tocarlos. Si se borran y se dejan vacío dará error puesto que los está intentando actualizar con datos vacíos.</p>

Tabla 54 - Editar servicio

A-12 Eliminar servicio	
Versión	1.0.0
Prioridad	Alta.
Estabilidad	Baja.
Descripción	Da de baja un servicio de la aplicación. El botón encargado de esto solo está disponible para los usuarios con rol de administrador.
Actor	Estudiante.
Entrada	-
Salida	Cuenta eliminada correctamente y redirección a la lista de todos los servicios para administrar o mensaje de error.
Precondición	El usuario administrador está autenticado.
Secuencia normal	<ol style="list-style-type: none"> 1. El administrador accede a la aplicación y pulsa sobre el home situado arriba a la izquierda. 2. El sistema le redirige a la pantalla de gestión de usuario (perfil). 3. El administrador pulsa “Modo administrador”. 4. El sistema redirige al usuario a una pantalla que muestra los cuatro módulos para administrar (usuarios, foros, grupos y servicios). 5. El usuario pulsa sobre el módulo servicios. 6. El sistema redirige al usuario a la pantalla de lista de todos los servicios de la aplicación. 7. El usuario presiona el botón de “Eliminar servicio” en un servicio de la lista. 8. El sistema muestra un modal (un pop-up) al usuario preguntando si está seguro de que desea borrar todos los datos. 9. El usuario pulsa confirmar. 13. El sistema redirige al usuario a la pantalla de lista de todos los servicios de la aplicación.
Postcondición si éxito	Se da de baja el servicio de la aplicación y se elimina de la base de datos.
Postcondición si fallo	No se da al servicio de baja y se le notifica el error.

Excepciones	5.1. El administrador borra y no rellena alguno de los campos y se muestra un mensaje de error. 9.1. Ocurre un error en servidor y se muestra un mensaje de error
Comentarios	

Tabla 55 - Eliminar servicio

Capítulo 5: Arquitectura

En este capítulo se explicará la arquitectura de la aplicación con los patrones utilizados.

5.1. Introducción arquitectura en Flutter

La aplicación de arquitectura Flutter^[16] consiste principalmente en:

1. Widgets: son el componente principal de cualquier aplicación Flutter, actuando como la UI.
2. Gestos: cualquier forma física de interacción con una aplicación Flutter se realiza a través de gestos predefinidos, utilizando los GestureDetector.
3. Concepto de Estado: Los estados son objetos de datos, utilizando StatefulWidget, el re-renderizado de los widgets específicos del estado se produce cada vez que el estado cambia.
4. Capas: jerarquía basada en el nivel decreciente de complejidades.

Las figuras 29 y 30 muestran la arquitectura de una aplicación en Flutter y la jerarquía de las capas nombradas. Se explicarán en detalle a continuación, ahora mismo solo ilustran la idea de estructura de cualquier aplicación en Flutter que sigue el convenio establecido, como lo ha sido la desarrollada en el TFG.

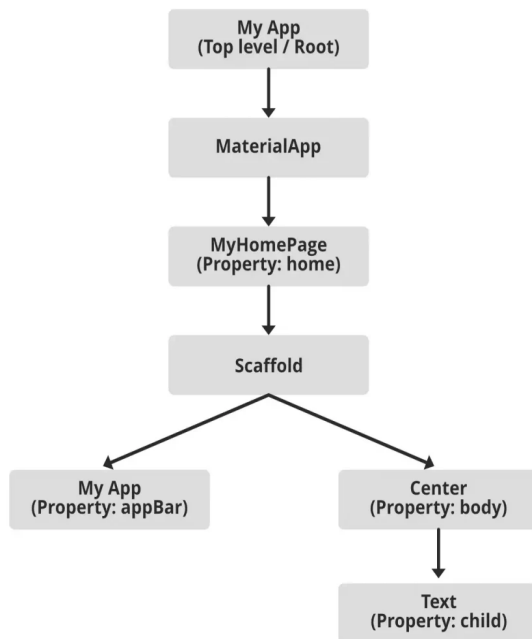


Figura 36 - Flutter, Arquitectura de Aplicaciones

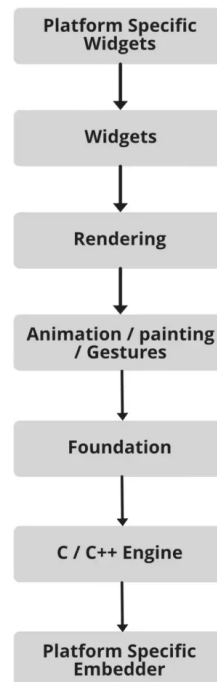


Figura 35 - marco de capas en Flutter

5.2. Arquitectura principal de la aplicación

5.2.1. Introducción

Para la arquitectura específica de la aplicación hemos pensado una lo más simple posible, por lo que será mucho más comprensible e intuitiva de seguir, siendo más escalable, organizada, con menor índice de errores y evitando una ‘sobreengeiería’^[17] innecesaria. La arquitectura pensada se ha representado en la figura 31.

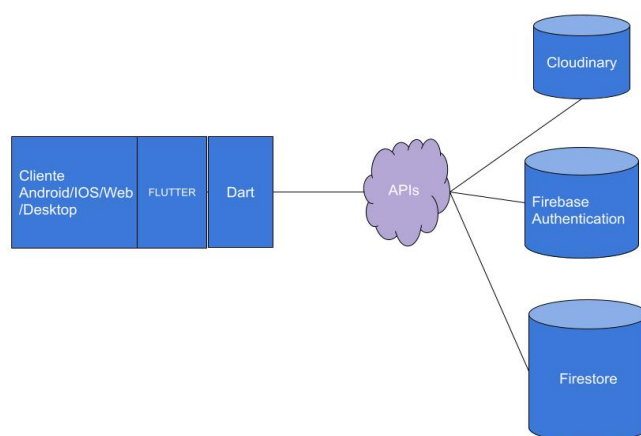


Figura 37 - Esquema app desarrollada

5.2.2. Patrones arquitectura

5.2.2.1. MVVM

El patrón MVVM^[18], Modelo-Vista-VistaModelo es un patrón que ayuda a desacoplar al máximo la interfaz de usuario de la lógica de la aplicación, facilitando el mantenimiento, las pruebas y la escalabilidad del proyecto. Sus tres componentes principales definen el nombre del patrón:

- **Model:** representa la capa de los datos. Contiene la información, pero no las acciones o servicios que lo manipulan.
- **View:** representa la interfaz de usuario. Contiene la parte visual y las interacciones con el usuario a través de animaciones, eventos, etc.
- **ViewModel:** será el intermediario entre los datos y la interfaz. Contiene la lógica de presentación y las operaciones con los datos del modelo.

Existen dos reglas fundamentales de este patrón:

1. Cada vista vendrá siempre acompañada por su modelo de vista.
2. La vista sólo conoce a su modelo de vista y este al modelo, la vista no conoce al modelo.

Los modelos de vista tienen que ser reactivos para proporcionar los datos en tiempo real y notificar los cambios que se vayan produciendo. Nos apoyaremos de la librería *Provider* para que la vista reaccione a los cambios que nos envíe el modelo de vista, la cual añadiremos a continuación. En la figura 32 se incluye una pequeña representación de este patrón.

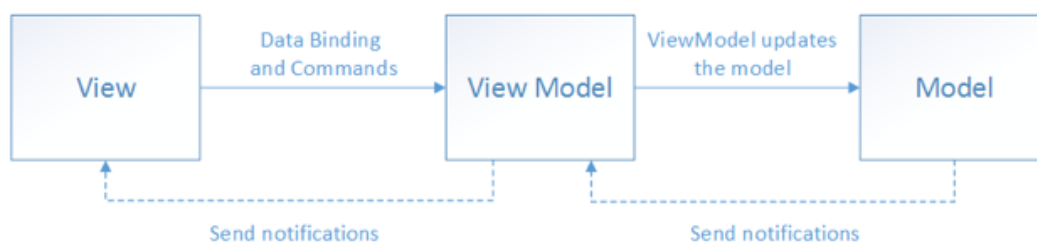


Figura 38 - Estructura patrón MVVM

5.2.2.2. Service Layer

Una capa de servicio "Service Layer"^[19] define el conjunto de operaciones disponibles desde la perspectiva de las capas de clientes que se interconectan. Encapsula la lógica de negocio de la aplicación, controlando las transacciones y coordinando las respuestas en la implementación de sus operaciones. (Figura 33)

5.2.2.3. Repository Layer

El patrón Repository^[20] es un patrón de diseño de software que desacopla la lógica de acceso a los datos de la lógica de negocio, mediante la introducción de un componente centralizado llamado repositorio. Proporciona registros de datos según un formato específico de la capa de datos.

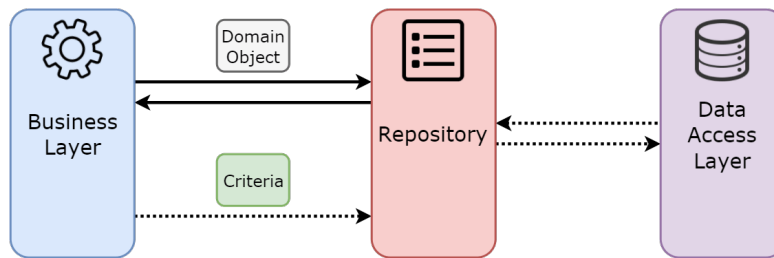


Figura 39 - Diagrama del patrón de service layer y repository intercambiando datos

5.3. Patrones de diseño

Aquí algunos patrones de diseño utilizados en la aplicación:

5.3.1. Inyección de dependencias

Consiste en crear una instancia del objeto requerido por la clase actual a través del contenedor, sin la necesidad de que la propia clase cree una instancia del objeto. El propósito es desacoplar la clase. En proyectos pequeños, es posible que el valor de la inyección de dependencia no se refleje, pero en proyectos de colaboración de varias personas a gran escala, la inyección de dependencia^[21] puede hacer que todo el proyecto sea más sólido y fácil de mantener. En este caso nos hemos decidido por la dependencia *Provider* ya que nos ofrece esta funcionalidad entre otras (figura 34).

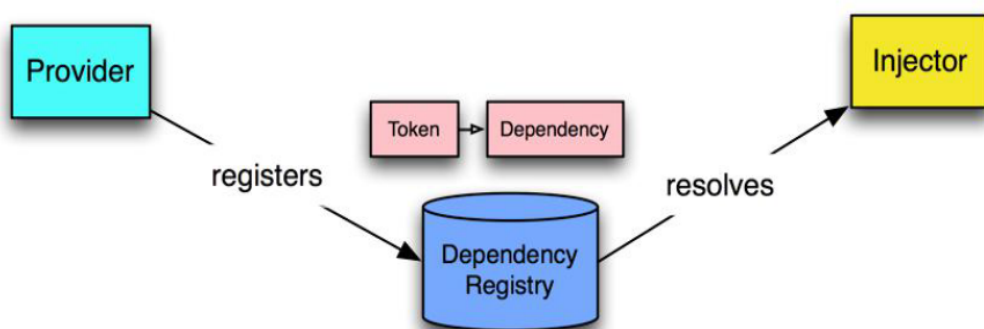


Figura 40 - Dependencia provider

5.3.2. Singleton

De acuerdo con impedir la creación de varios objetos de una clase, el patrón Singleton se utiliza para controlar la creación y el uso de objetos de una clase de forma dinámica en tiempo de ejecución. En el caso de este proyecto para controlar que solo haya un único objeto de las clases repositorio de acceso a datos y de servicios para acceder a la lógica de aplicación, mediante el uso y la ayuda de la dependencias `get_it` y `Provider`.

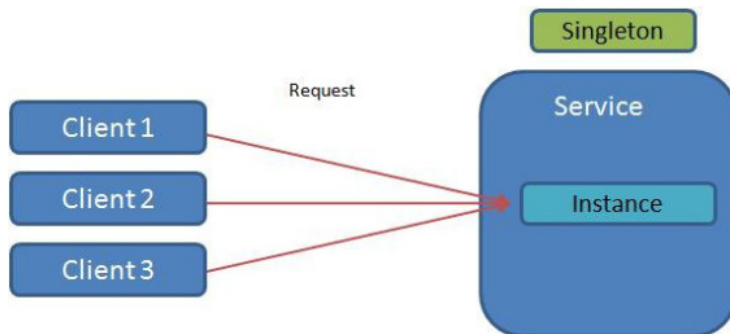


Figura 41 - Singleton ejemplo `getInstance()`

5.3.3. Observer

Es uno de los patrones de diseño de software más usados, ofreciendo la posibilidad de definir una dependencia entre dos o más objetos transmitiendo los cambios de un objeto al resto. Se registra en un objeto cualquier otro objeto, que funcionará como observador.

Este patrón es muy importante en la aplicación para mantener los elementos/Widgets actualizados en las pantallas ante cualquier cambio, se utilizan Widgets con estado (`StatefulWidgets`) y/o también nos lo resuelve la dependencia `Provider`.

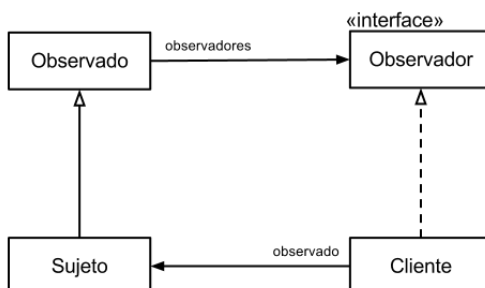


Figura 42 - Patrón observer

5.3.4. Data Transfer Object

Es un objeto que transporta datos entre procesos. La motivación para su uso es que la comunicación entre procesos suele realizarse recurriendo a interfaces remotas (ya sea por ejemplo con Firebase o Cloudinary).

La diferencia entre los objetos de transferencia de datos y los objetos de negocio o los objetos de acceso a datos es que un DTO^[22] no tiene ningún comportamiento, excepto el almacenamiento, la recuperación, la serialización y la deserialización de sus propios datos. Para la realización de estos nos hemos

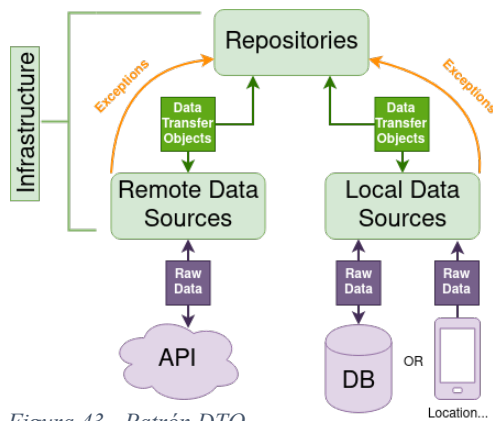


Figura 43 - Patrón DTO

Con este diagrama queda explicado de un vistazo como el sistema tiene 4 entidades principales: servicios, grupos, foros y usuarios. Obtenemos visualmente información de los atributos de cada tabla y como se relacionan entre ellas para identificar al dueño de un servicio, al usuario que envía un mensaje, etc.

6.2. Implementación de la base de datos

El modelo E-R de la figura 26 se ha implementado mediante una base de datos NoSQL de tipo FireBase, y para ello se han definido las siguientes colecciones de documentos del apartado 5.3.

Cada entidad cuenta un identificador único, generado aleatoriamente por FireBase. Este identificador se ha obviado del modelo E-R de la figura 26, pero aparece descrito en las tablas detalladas que aparecen en el siguiente punto de implementación de la base de datos. En la figura 27 se puede ver el identificador de cada subforo.

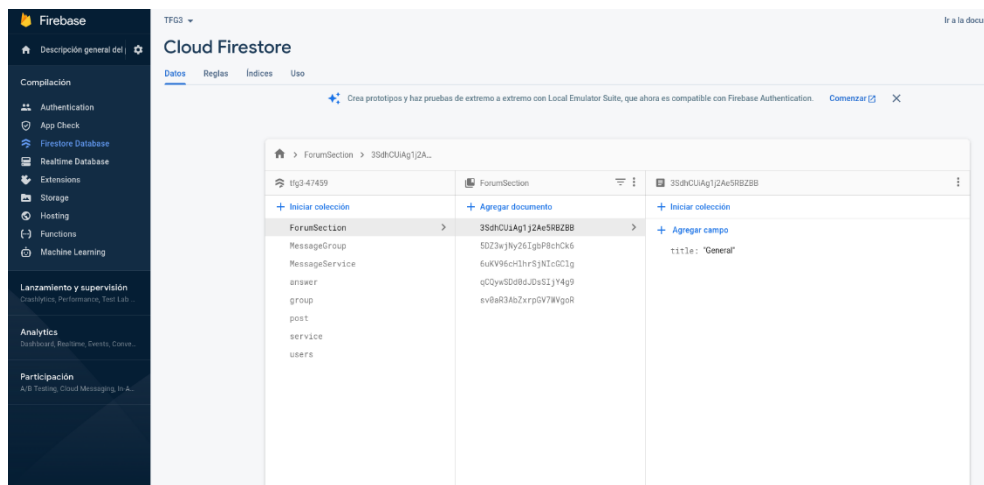


Figura 45 - Implementación de la bbdd

6.3. Colecciones de documentos

En esta sección se describen todas las colecciones de documentos definidas en el modelo de datos de la aplicación Uni4All que se ha desarrollado en este trabajo de fin de grado. En la figura 28 se presenta un diagrama con todas las colecciones definidas y como se relacionan entre sí, previo al desarrollo de los datos de cada colección.

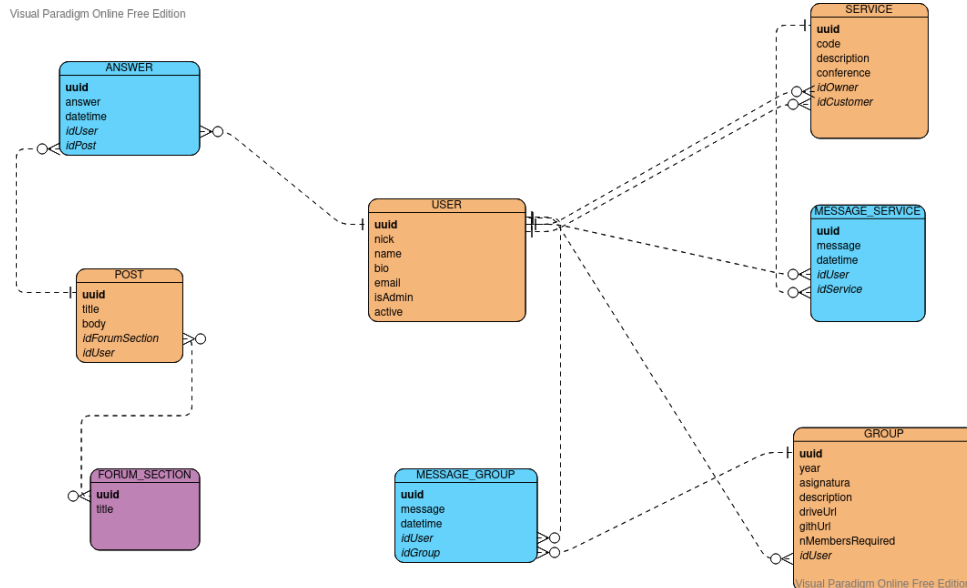


Figura 46 - Diagrama colecciones documentos

- Colección de documentos de tipo User: contiene información sobre los usuarios y está constituida por los siguientes campos:
 1. Email: correo electrónico del usuario.
 2. Active: determina si se trata de un usuario activo o no.
 3. Bio: biografía sobre el usuario.
 4. isAdmin: determina si se trata de un usuario administrador o estándar.
 5. Name: nombre del usuario.
 6. Nick: alias del usuario.
 7. Password: contraseña del usuario.
 8. Id: id único del registro

- Colección de documentos de tipo Answer: en ella se encuentra la información de los diferentes Post del foro, sus campos son los siguientes:
 1. DateTime: fecha en la que se publicó la respuesta.
 2. Answer: contenido de la respuesta.
 3. Id: id único del registro.

- Colección de documentos de tipo Post: guarda la información perteneciente a las respuestas de cada uno de los Post del foro, está formada por los siguientes campos:
 1. Title: título de la publicación del foro.
 2. Body: cuerpo de la publicación del foro.
 3. Id: id único del registro.

- Colección de documentos de tipo Forum Section: alberga la información de cada uno de los temas del foro, sus campos son los siguientes:
 1. Title: título de la sección del foro.
 2. Id: id único del registro.

- Colección de documentos de tipo Service: contiene la información referente al servicio.
 1. Code: código de la asignatura.
 2. Conference: enlace a vídeo conferencia.
 3. Description: texto explicativo del servicio.
 4. Id: id único del registro.

- Colección de documentos de tipo Group: contiene la información de los grupos de trabajo.
 1. nMembersRequired: número de miembros que admite el grupo.
 2. GithubUrl: enlace al GitHub en el que se encuentran el código del grupo.
 3. DriveUrl: enlace al Google Drive donde el equipo almacena sus documentos.
 4. Description: descripción del grupo.
 5. Asignatura: código único de la asignatura.
 6. Year: curso al que pertenece la asignatura.
 7. Id: id único del registro.

- Colección de documentos de tipo Message Service: en ella se encuentran los datos de los mensajes enviados entre los usuarios que comparten un servicio.
 1. DateTime: fecha de publicación del mensaje.
 2. Message: cuerpo del mensaje.
 3. Id: id único del registro.

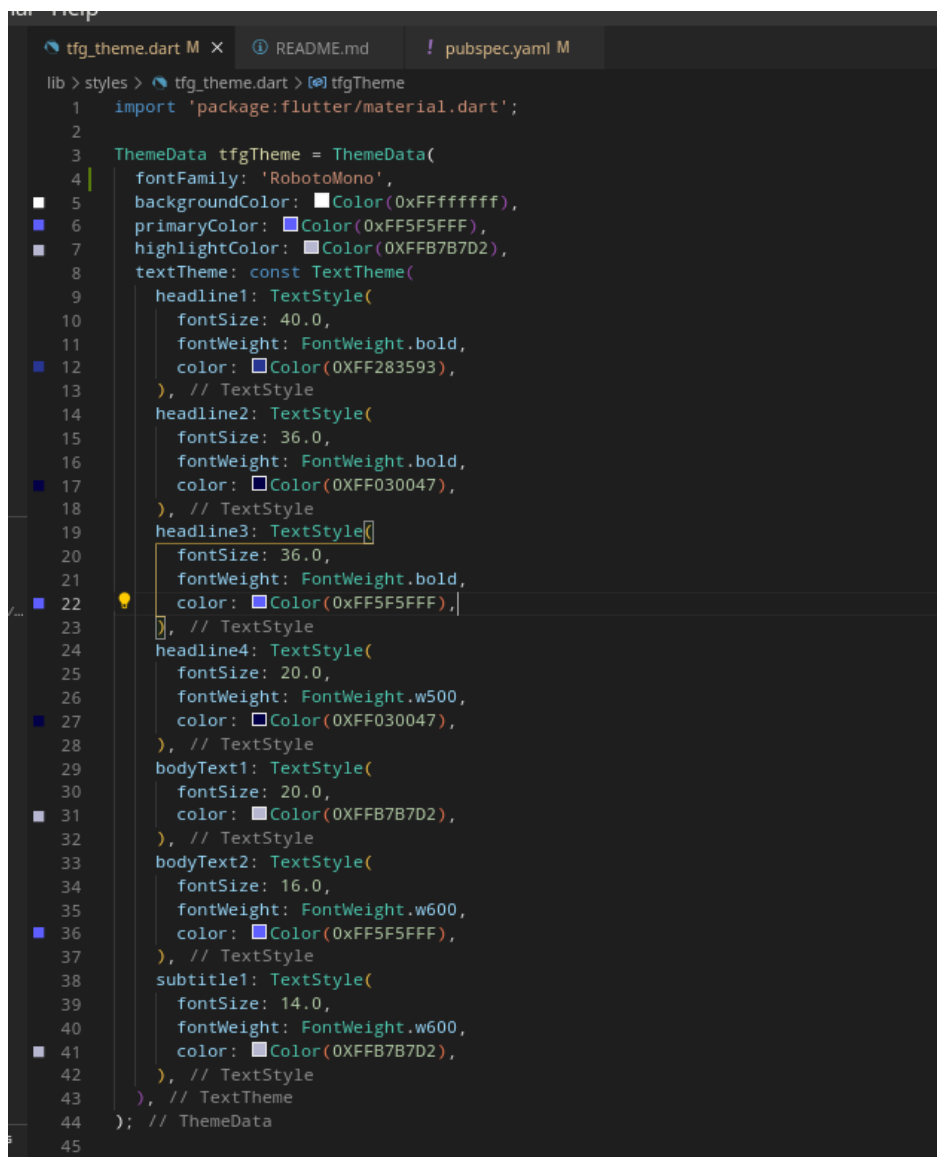
- Colección de documentos de tipo Message Group: en ella se encuentran los datos de los mensajes enviados entre usuarios del mismo grupo.
 1. Message: cuerpo del mensaje.
 2. DateTime: fecha de publicación del mensaje.
 3. Id: id único del registro.

Capítulo 7: Diseño

En este capítulo, se van a describir los principales aspectos acerca del diseño realizado para el desarrollo del proyecto y las principales funcionalidades.

7.1. Colores y Diseño

En la elección de colores se buscó una paleta de colores muy sencilla para la aplicación, de forma que no sobrecargue visualmente al usuario y sean colores lo más suaves posibles. Está formada el blanco como color base principal, el negro y varios colores entre azul y morado para dar contraste. De cara al diseño, se optó por un ‘material-design’, debido a su sencillez y minimalismo, en el que el usuario ya está familiarizado, conjunto a la tipografía característica de ‘Roboto’. En la figura X se muestran cómo se compone el tema utilizado para la aplicación.



```
lib > styles > tfg_theme.dart > tfgTheme
1  import 'package:flutter/material.dart';
2
3  ThemeData tfgTheme = ThemeData(
4
5    fontFamily: 'RobotoMono',
6    backgroundColor: Color(0xFFffffff),
7    primaryColor: Color(0xFF5F5FFF),
8    highlightColor: Color(0xFFB7B7D2),
9    textTheme: const TextTheme(
10     headline1: TextStyle(
11       fontSize: 40.0,
12       fontWeight: FontWeight.bold,
13       color: Color(0xFF283593),
14     ), // TextStyle
15     headline2: TextStyle(
16       fontSize: 36.0,
17       fontWeight: FontWeight.bold,
18       color: Color(0xFF030047),
19     ), // TextStyle
20     headline3: TextStyle(
21       fontSize: 36.0,
22       fontWeight: FontWeight.bold,
23       color: Color(0xFF5F5FFF),
24     ), // TextStyle
25     headline4: TextStyle(
26       fontSize: 20.0,
27       fontWeight: FontWeight.w500,
28       color: Color(0xFF030047),
29     ), // TextStyle
30     bodyText1: TextStyle(
31       fontSize: 20.0,
32       color: Color(0xFFB7B7D2),
33     ), // TextStyle
34     bodyText2: TextStyle(
35       fontSize: 16.0,
36       fontWeight: FontWeight.w600,
37       color: Color(0xFF5F5FFF),
38     ), // TextStyle
39     subtitle1: TextStyle(
40       fontSize: 14.0,
41       fontWeight: FontWeight.w600,
42       color: Color(0xFFB7B7D2),
43     ), // TextStyle
44   ), // TextTheme
45 ); // ThemeData
```

Figura 47 - tfg_theme.dart del proyecto Uni4All

7.2. Funcionalidad principal

7.2.1. User y login

Para acceder a la aplicación es necesario estar registrado como usuario, por lo que el usuario tendrá que registrarse y posteriormente hacer iniciar sesión para obtener el token e UUID[23] de usuario que guardaremos en el Keystore[24] de android o el Keychain[25] en IOS, contenedores especiales seguros donde guardar claves criptográficas[26]. Este es utilizado en la aplicación mediante la dependencia FlutterSecureStorage[27], instanciada como se ve en la figura 39.

```
class AuthService {  
  final storage = new FlutterSecureStorage();  
}
```

Figura 48 - Almacenamiento clave login

Al registrarse el usuario se da de alta en el servicio de FirebaseAuthentication y al mismo tiempo el documento relativo a su información es guardado en la base de datos Firestore, como se puede ver en la figura 40, donde primero se llama al repositorio de autenticación y posteriormente al servicio de Usuario. Como se puede observar son instanciados mediante la dependencia GetIt[28] la cual implementa el patrón Singleton.

```
7 | import 'package:flutter_application_core/services/user_service.dart';  
8 | import 'package:get_it/get_it.dart';  
9 | import 'package:flutter_secure_storage/flutter_secure_storage.dart';  
10 |  
11 | class AuthService {  
12 |   final storage = FlutterSecureStorage();  
13 |  
14 |   Future<String?> signUpUser(User user) async {  
15 |     final resp = await GetIt.I<AuthRepository>().signUpUser(user);  
16 |  
17 |     final Map<String, dynamic> decodedResp = jsonDecode(resp.body);  
18 |  
19 |     if (decodedResp.containsKey('idToken')) {  
20 |       //Registro realizado con exito  
21 |       //Guardar informacion usuario con su uid auto generado  
22 |       GetIt.I<UserService>().updateUserData(user, true);  
23 |       return null;  
24 |     } else {  
25 |       if (decodedResp.containsKey('error')) {  
26 |         if (decodedResp['error']['message'] == 'EMAIL_EXISTS') {  
27 |           return 'Email ya existente.';  
28 |         } else {  
29 |           return 'Error de conexion';  
30 |         }  
31 |       }  
32 |       return 'Error de conexion';  
33 |     }  
34 |   }  
35 | }
```

Figura 49 - Código del registro

Si el usuario ya está registrado en la aplicación puede hacer login manualmente (figura 41), o si recientemente ha iniciado sesión y el dispositivo continúa almacenando el token de sesión del usuario utilizará este token entrando automáticamente a la aplicación.

```
Future<String?> signInUser(
    String email, String password, BuildContext context) async {
    final Map<String, dynamic> decodedResp =
        await GetIt.instance<AuthRepository>().signInUserInfo(email, password);

    if (decodedResp.containsKey('idToken')) {
        var userInfo =
            await GetIt.I<UserService>().getUserData(decodedResp['localId']);
        if (userInfo == null) return 'Información del Usuario no encontrada';

        //Token e info relativa a usuario hay que guardarlo en lugar seguro para el uso de la app
        await storage.write(key: 'token', value: decodedResp['idToken']);
        await storage.write(key: 'userId', value: decodedResp['localId']);

        return null;
    } else {
        if (decodedResp.containsKey('error')) {
            if (decodedResp['error']['message'] == 'EMAIL_NOT_FOUND') {
                return 'Usuario no registrado';
            } else if (decodedResp['error']['message'] == 'INVALID_PASSWORD') {
                return 'Contraseña incorrecta';
            } else {
                return 'Error de conexión';
            }
        }
        return 'Error de conexión';
    }
}
```

Figura 50 - Código del login

Al acceder el usuario a la aplicación, aparece en la página principal de la misma, mostrándose en pantalla su perfil de usuario, con su información personal y unas estadísticas de: los servicios que ha solicitado y ofertado, los grupos a los que pertenece y los posts en los que ha participado. En la figura 42 se puede apreciar.



Figura 51 - Servicios, grupos y posts.

Todo esto a través de unas queries proporcionadas por FireBase, igual que la información de usuario obtenida. En la figura 43 se puede ver una captura de que ve al registrarse o iniciar sesión.



Figura 52 - Dashboard usuario

Para salir de esta pantalla se pulsará el botón que hay en la esquina superior izquierda, apareciendo como primera pantalla la sección de Perfil. Desde esta pantalla, y cualquiera del resto de la aplicación, se puede navegar mediante la barra de navegación inferior a las distintas pantallas de la aplicación, entre ellas, las 3 principales funcionalidades: Servicios, Foros, Grupos y Perfil, esta última aparecerá por defecto seleccionada. Para volver a la pantalla inicial se podrá pulsar de nuevo el botón de la esquina superior izquierda (Figura 44).

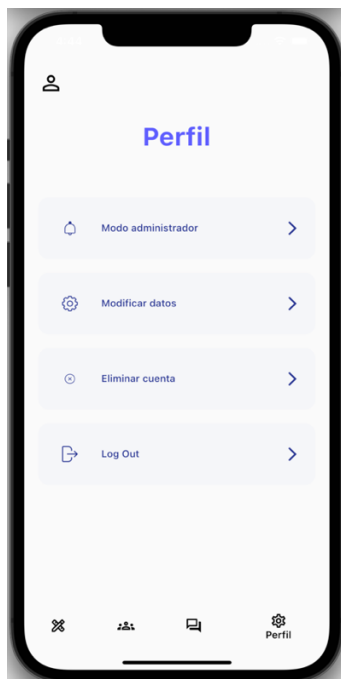


Figura 53 - Barra inferior con navegado a los módulos

7.2.2. Perfil

Este apartado de Perfil es el primero al que accederemos por defecto al entrar a la aplicación. Aquí se aparecerán varias opciones: ‘Modificar datos’, ‘Eliminar cuenta’, ‘Log out’ y ‘Modo administrador’, este último será solo visible para aquel usuario que tiene rol administrador, el cual solo se podrá especificar via Firebase. (Figura 44)

La opción de ‘Log Out’ y ‘Eliminar cuenta’ son claras, si pulsas ‘Log Out’ saldrás de la aplicación y se tendrá que iniciar sesión de nuevo para volver a entrar. Y ‘Eliminar cuenta’ se borrarán todos tus datos, avisando previamente de esto al usuario antes de confirmar la operación.

En la opción de ‘Modificar datos’ el usuario podrá actualizar su información personal y su foto de perfil, todo esto representado visualmente para el usuario en la figura 45, y ejecutado en la figura 48.

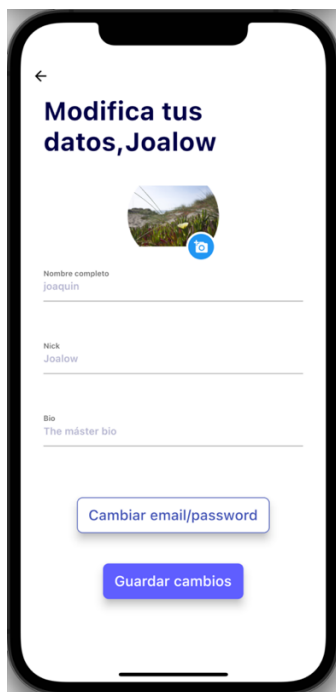


Figura 54 - Modificado de datos

```
Future updateUserData(User user, bool active) async {
  if (user.isAdmin) {
    return await GetIt.I<UserRepository>().updateUser({
      'name': user.fullName,
      'email': user.email,
      'password': user.password,
      'nick': user.nick,
      'isAdmin': true,
      "bio": user.bio,
      "url": user.url,
      "active": active
    }, user.id!);
  } else {
    return await GetIt.I<UserRepository>().updateUser({
      'name': user.fullName,
      'email': user.email,
      'password': user.password,
      'nick': user.nick,
      'isAdmin': false,
      "bio": user.bio,
      "url": user.url,
      "active": active
    }, user.id!);
  }
}
```

Figura 55 - Servicio actualización usuario

En la opción de ‘Modificar datos’ el usuario podrá actualizar su información personal y su foto de perfil. La foto es subida mediante la galería o cámara del dispositivo, utilizando una API a la cual se envía la imagen al servicio cloud[29] ‘Cloudinary’ almacenando esta de forma segura y pudiendo acceder a ella cuando se quiera y con múltiples opciones. La imagen primero será mostrada en pantallas, y si el usuario confirma los cambios a guardar será guardada esta nueva imagen para el usuario, si no los confirmase no se guardaría. (Figuras 45 y 46).

```

final picker = ImagePicker();
final XFile? pickedFile = await picker.pickImage(
  source: ImageSource.gallery,
  imageQuality: 100);

if (pickedFile == null) {
  return;
}
editUserProvider.newPictureFile =
  File.fromUri(Uri(path: pickedFile.path));
String? newPictureLocal = await editUserProvider.uploadImage();

setState(() {
  localImage = newPictureLocal;
});

```

Figura 56 – Código del cambio de foto

```

if (!editUserProvider.isValidForm()) return;
if (localImage != null) {
  args.user.url = localImage;
  if (args.userSession) {
    setState(() {
      userSessionProvider.user.url = localImage;
    });
  }
}
GetIt.I<UserService>().updateUserData(args.user, true);
if (args.userSession) {
  Navigator.pop(context); // pop current page
} else {
  Navigator.pushNamedAndRemoveUntil(context,
    'adminHomePage', (Route<dynamic> route) => false,
    arguments: 0);
}

```

Figura 57 - Código de la llamada a la API para actualización de usuario

También en esta pantalla se podrá modificar los datos de sesión (email y password), con un formulario específico el cual aparecerá en un diálogo como se muestra en la figura X vista a continuación, ya que el servicio utilizado para la autenticación es diferente al de base de datos.

```

if (!emailPassFormProvider.isValidForm()) return;

// Close the dialog
Navigator.of(context).pop();

String? resp = await GetIt.I<AuthService>()
  .changeEmailPassword(
    emailPassFormProvider, userSessionProvider);

print(resp);
if (resp != null) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text("ERROR: $resp")));
} else {
  ScaffoldMessenger.of(context).showSnackBar(SnackBar(
    content: Text(
      "EXITO: Volver hacer login para aplicar cambios"))); // Text // SnackBar
}

```

Figura 58 - Formulario autenticación actualización credenciales

Por último, el ‘Modo administrador’ dirige a una pantalla en la cual tendremos cuatro pestañas como en la figura X vista a continuación con las entidades principales: Usuarios, Foros, Grupos y Servicios, recogidos en la Figura 50.

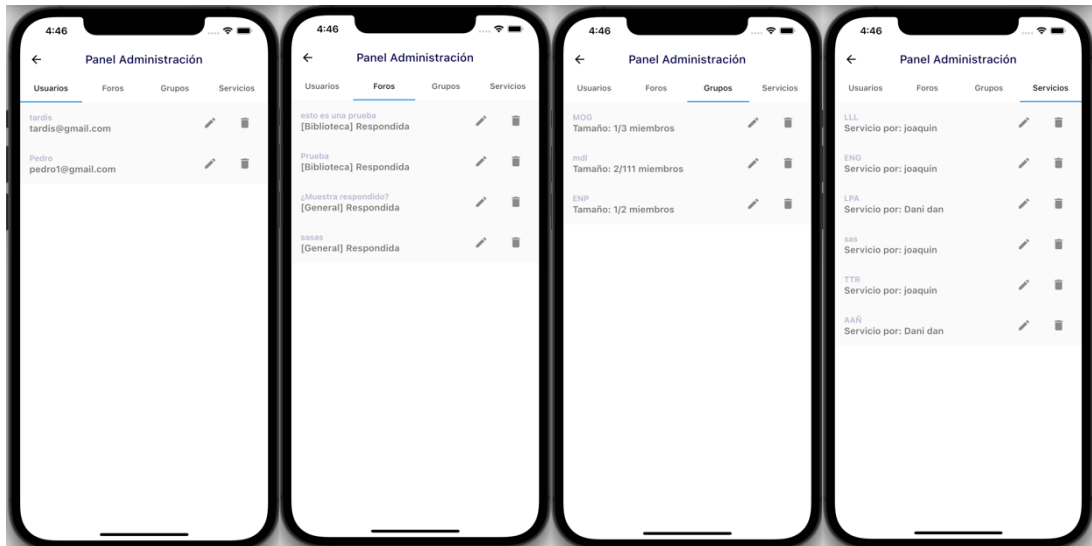


Figura 59 - Modulo administración

Por último, el usuario podría eliminar su cuenta, borrando todos los datos y credenciales del usuario sin posibilidad de volver a entrar en la aplicación, y hacer log out de la aplicación. Estos botones están referenciados por fechas en la figura 51.

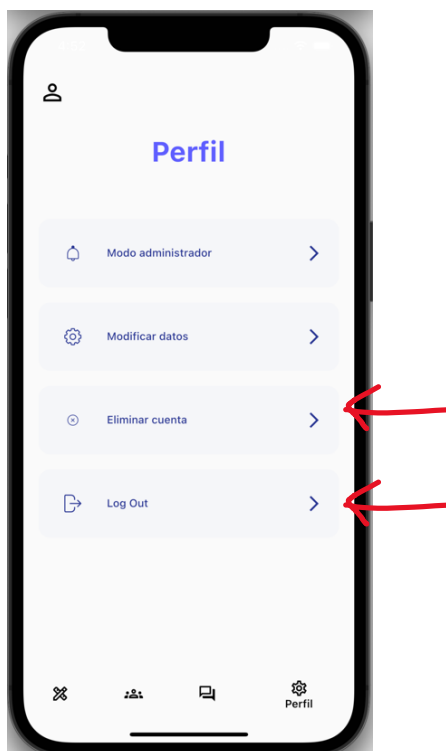


Figura 60 - eliminar y salir de la cuenta

7.2.3. Foros

En el apartado de foros aparecen las secciones almacenadas en la base de datos actualmente: General, Biblioteca, Asignaturas, Erasmus, Cafetería. Con la posibilidad de que el administrador dé de alta más manualmente bajo petición. Estas son obtenidas via el documento almacenado 'ForumSection'.

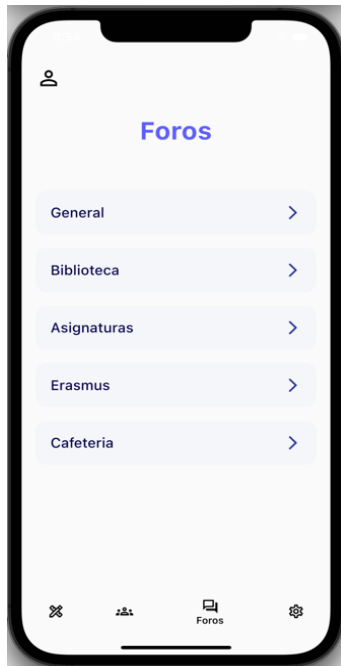


Figura 61 - Vista principal foros



Figura 62 – Vista posts de subforo

En la figura 52 se muestra el servicio de foro, que hace una llamada al repositorio de foros y retorna una lista de 'ForumSection'.

```
import 'package:get_it/get_it.dart';

class ForumService {
  Future<List<ForumSection>> getForumSections() async {
    final QuerySnapshot forumCollection =
      await GetIt.I<ForumRepository>().getForumSections();

    List<ForumSection> listForumSections = [];
    for (var element in forumCollection.docs) {
      Map<dynamic, dynamic> forumSectionMap = element.data() as Map;
      ForumSection forumSection = ForumSection(
        title: forumSectionMap["title"],
        id: element.id,
      );

      listForumSections.add(forumSection);
    }

    return listForumSections;
  }
}
```

Figura 63 - Servicio de foro

Si le damos a una de estas secciones nos llevará a una lista de todos los posts que almacena esta sección, con la posibilidad de añadir uno nuevo y buscar por título,

mediante un buscador en la barra superior de la pantalla implementada con el SearchDelegate de Flutter (figura 53).

```
class MySearchDelegateForums extends SearchDelegate {
  @override
  String get searchFieldLabel => 'Buscar post por titulo';

  @override
  List<Widget>? buildActions(BuildContext context) => [
    IconButton(
      onPressed: () => query = '',
      icon: Icon(Icons.clear),
    ) // IconButton
  ];

  @override
  Widget? buildLeading(BuildContext context) => IconButton(
    onPressed: () => close(context, null),
    icon: Icon(Icons.arrow_back),
  ); // IconButton

  @override
  Widget buildResults(BuildContext context) {
    return buildSuggestions(context);
  }
}
```

Figura 64 - SearchDelegateForums de Uni4All

Si le damos a un post, nos llevará a la pantalla de detalle del post, en la cual aparecerá la información relativa a este: Título, Cuerpo del post, Sección a la que pertenece, si el post ha sido respondido o aun no y el usuario que lo creó.

En la parte de abajo, aparecerá una lista de todas las respuestas que ha recibido este post, con la posibilidad de participar añadiendo una nueva (figura 57). En la respuesta se visualiza la persona que ha escrito la respuesta, el mensaje de respuesta, y la fecha en la que se envió. Los mensajes aparecerán en orden de más reciente a más antiguo. Esta respuesta está vinculada a un Post y a un Usuario en la entidad 'Answer'.

```
if (!answerFormProvider.isValidForm()) return;
answerFormProvider.idPost = args.post!.id!;
answerFormProvider.idUser =
  userSessionProvider.user.id!;
await GetIt.I<AnswerService>()
  .updateAnswer(answerFormProvider.answer(), null);
final postMainProvider =
  Provider.of<PostMainProvider>(context,
    listen: false);
await postMainProvider.loadPost(args.post!.id!);
postMainProvider.notifyListeners();
Navigator.pop(context, args);
```

Figura 65 - Llamada al servicio de creación de una respuesta.



Figura 66 - Visualización hilo y respuestas

También en esta misma pantalla (seguimos en la figura 37) se da la opción de poder editar y eliminar el post con los botones que aparecerán en la esquina superior derecha, únicamente si el usuario es el creador del mismo.

7.2.4. Grupos

En el apartado de Grupos aparecen en la pantalla principal los grupos a los que tienes acceso, los que has creado y en los que participas (figura 59). También aparecen las opciones de crear un nuevo grupo y de acceder a una nueva pantalla con la lista de todos los grupos existentes.

Si vamos a esta pantalla de todos los grupos existentes, aparecerán aquellos en los que participemos y no, con la posibilidad de unirnos a los que no. También en la barra superior tenemos la opción de buscar por código, implementada con el SearchDelegate de flutter.

Al acceder a un grupo nos llevará a la pantalla de información del mismo: el número de usuarios que participan y máximo, con la posibilidad de acceder a ellos, el código de asignatura del grupo, el usuario que creó el grupo, la descripción del grupo, las urls si existiesen de github y drive (las cuales podríamos hacer click y que nos redirija a estas páginas con el navegador por defecto del terminal), y por

último un apartado de chat en el que se podrán intercambiar mensajes entre usuarios, tal y como aparece en la situación de la figura 61.

```
Future joinGroup(Group group, String idUser) async {
  if (!group.idMembers.contains(idUser)) group.idMembers.add(idUser);

  return await GetIt.I<GroupRepository>().joinGroup({
    "asignatura": group.asignatura,
    "year": group.year,
    "description": group.description,
    "githubUrl": group.githubUrl,
    "driveUrl": group.driveUrl,
    "nMembersRequired": group.nMembersRequired,
    "idMembers": group.idMembers,
    "idUser": group.idUser,
  }, group.idUser);
}
```

Figura 67 - Servicio de unirse a un grupo

Esta será reducida si el usuario no pertenece a este y tendrá la posibilidad de unirse al mismo si el grupo no ha llegado al número de usuarios máximos permitidos, como se muestra en la figura 60.

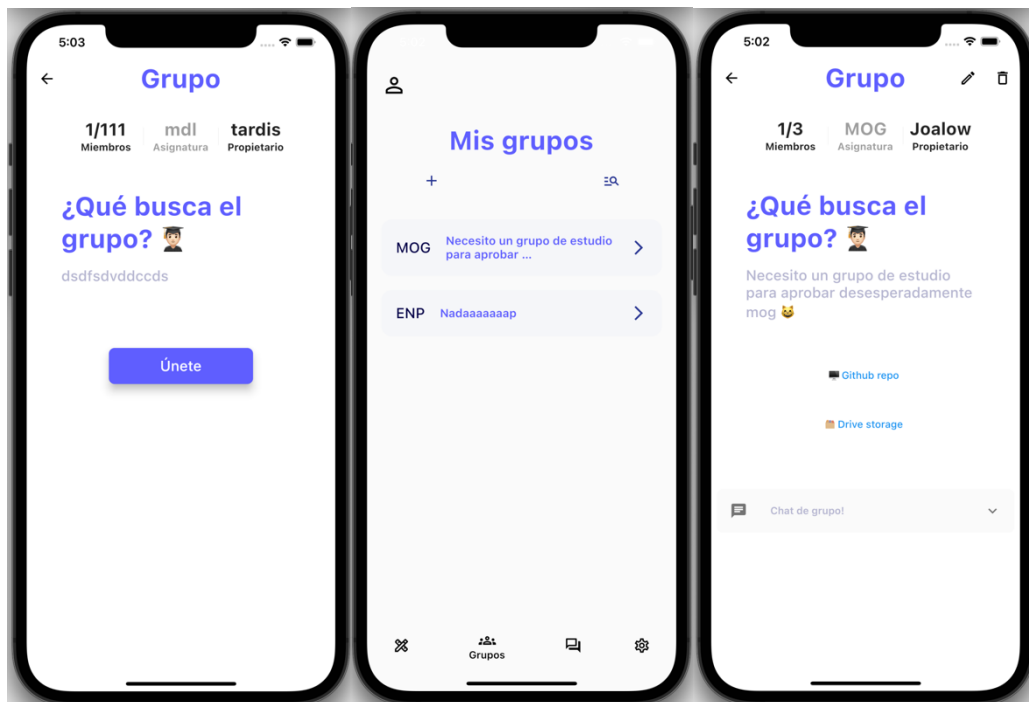


Figura 69 - Mostrado de grupo

Figura 68 - Figura dashboard grupos

Figura 70 – Selección de grupo

De igual forma, si el usuario ya pertenece al grupo, tendrá la opción de acceder a los mensajes y urls de interés. También tendrá la opción de dejar el grupo (figura 61).

Los mensajes intercambiados entre los miembros del grupo están ordenados por fecha, de más reciente a más antiguo. Estos son creados por la entidad 'MessageGroup' la cual vincula el Usuario y el Grupo con un mensaje.

```
unpressed: () async {
  if (messageFormProvider.message != '') {
    messageFormProvider.idGroup = args.group!.id!;
    messageFormProvider.idUser =
      userSessionProvider.user.id!;
    await GetIt.I<MessageGroupService>()
      .updateGroupMessage(
        messageFormProvider.groupMessage(), null);
    await Provider.of<GroupListProvider>(context,
      listen: false)
      .loadMessagesGroupList(args.group!.id!);
    Navigator.pop(context, args);
  } else {
    ScaffoldMessenger.of(context).showSnackBar(SnackBar(
      content: Text("RELLENA TODOS LOS CAMPOS")); // SnackBar
  }
}
```

Figura 71 - Código MessageGroup de Uni4All

Por último, si el usuario es el dueño del grupo, podrá editarlo y eliminarlo, con las acciones situadas en la esquina superior derecha.



Figura 72- Editado y borrado de grupo (owner)

7.2.5. Servicios

En el apartado de Servicios aparecen en la pantalla principal los servicios a los que tienes acceso, los que has creado y los que has adquirido. También aparecen las opciones de crear un nuevo servicio (figura 65) y de acceder a una nueva pantalla con la lista de todos los servicios existentes.

Si vamos a esta pantalla de todos los servicios existentes, aparecerán aquellos que hayamos ofertado, que hayamos adquirido y no, con la posibilidad de adquirir los que no.

También en la barra superior tenemos la opción de buscar por código, implementada con el SearchDelegate de flutter.

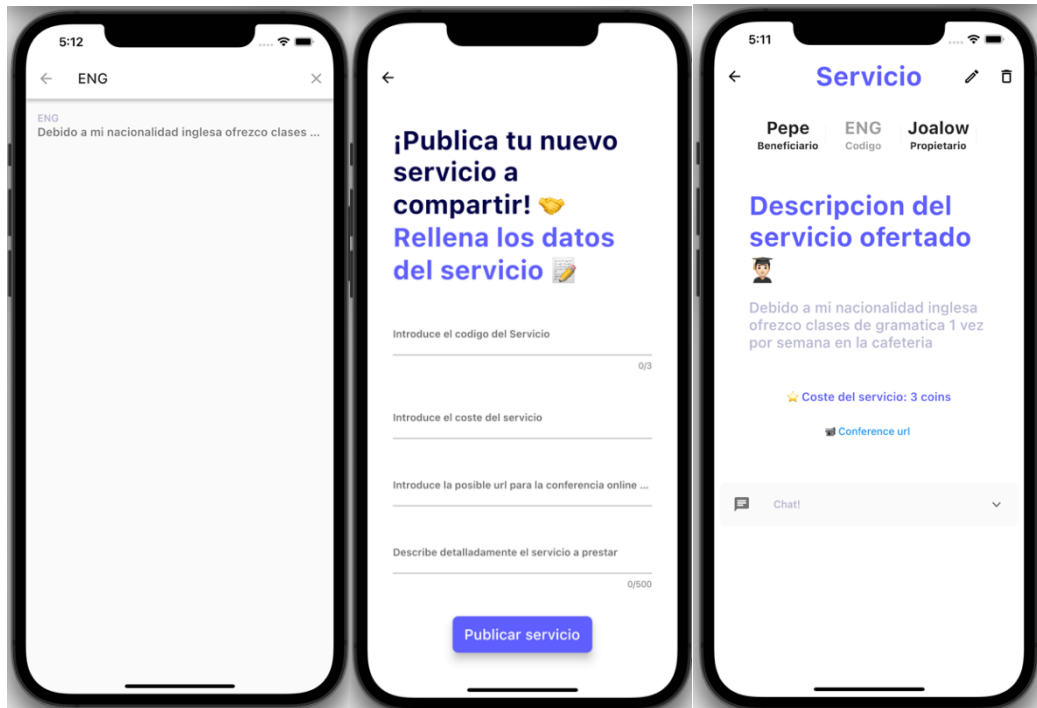


Figura 73 - Búsqueda de servicios Figura 75 - Creación de servicios Figura 74 - Detalle de servicio propio

Al acceder a un servicio, como en la figura 34, nos llevará a la pantalla de información del mismo: el usuario que ha adquirido el servicio, el código de del servicio, el usuario que ha creado el servicio, la descripción del servicio, una url con el link a una conferencia de Zoom (las cuales podríamos hacer click y que nos redirija a esta sesión en el navegador por defecto del terminal, si el servicio se realiza vía online), y por último un apartado de chat en el que se podrán intercambiar mensajes entre los usuarios participantes (que creador del servicio y el que lo consume).

```
Future obtainService(Service service, String idUser) async {
  if (service.idOwnerUser != idUser) service.idCustomerUser = idUser;

  return await GetIt.I<ServiceRepository>().updateService({
    "code": service.code,
    "conference": service.conference,
    "description": service.description,
    "idOwnerUser": service.idOwnerUser,
    "idCustomerUser": service.idCustomerUser,
  }, service.id);
}
```

Figura 76 - Servicio de unirse a un servicio

Los mensajes intercambiados entre los miembros del servicio están ordenados por fecha, de más reciente a más antiguo. Estos son creados por la entidad 'MessageService' la cual vincula el Usuario y el Servicio con un mensaje. El código se recoge en la figura 68.

```
onPressed: () async {
  if (messageFormProvider.message != '') {
    messageFormProvider.idService = args.service!.id!;
    messageFormProvider.idUser =
      userSessionProvider.user.id!;
    await GetIt.I<MessageServiceService>()
      .updateServiceMessage(
        messageFormProvider.serviceMessage(), null);
    await Provider.of<ServiceListProvider>(context,
      listen: false)
      .loadMessagesServiceList(args.service!.id!);
    Navigator.pop(context, args);
  } else {
    ScaffoldMessenger.of(context).showSnackBar(SnackBar(
      content: Text("RELLENA TODOS LOS CAMPOS"))); // SnackBar
  }
}
```

Figura 77 - Código de envío de mensajes en servicios

Por último, si el usuario es el dueño del servicio, podrá editarlo y eliminarlo, con las acciones situadas en la esquina superior derecha como siempre, como se refleja en la figura 69.



Figura 78 - Editado y borrado de un grupo propio

Capítulo 8: Evaluación

En este capítulo se realiza un análisis de los resultados que hemos obtenido en una encuesta realizada a 33 personas, en las que se valora principalmente la usabilidad de la aplicación que se ha desarrollado además de algunos aspectos más generales de la misma.

8.1. Herramientas y planteamiento

Para la evaluación se ha utilizado Google Forms. Esta ha sido realizada por treinta y tres personas de entre 20 y 30 años con diferentes titulaciones universitarias, pertenecientes a distintos ámbitos del conocimiento y con diversos grados de familiarización con la tecnología.

En la evaluación se distinguen diferentes secciones:

- **General:** en la que se tratan aspectos más generales como una valoración numérica de la propia aplicación, se pregunta a los usuarios si creen que harían uso de la aplicación en el caso de que su universidad se la proporcionará o si la interfaz de usuario resulta atractiva.
- **Registro / inicio de sesión:** se preguntan aspectos de la usabilidad de la sección de registro y del inicio de sesión.
- **Servicios:** se preguntan aspectos de la usabilidad de la sección de servicios.
- **Grupos:** se preguntan aspectos de la usabilidad de la sección de registro y del inicio de sesión.
- **Perfil:** se preguntan aspectos de la usabilidad de la sección del perfil del usuario.
- **Foros:** se preguntan aspectos de la usabilidad de la sección de foros.
- **Conclusiones y sugerencias:** en este apartado se pregunta al usuario sobre la utilidad de las diferentes secciones y si desean compartir con nosotros alguna sugerencia para mejorar la aplicación.

8.2. Resultado de la evaluación:

En esta sección se muestran las preguntas que se han realizado y una conclusión de cada uno de los resultados obtenidos en las mismas.

8.2.1. General

8.2.1.1. Edad

¿Qué edad tienes?
33 respuestas

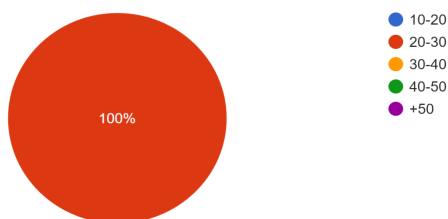


Figura 79 - Gráfico pregunta: ¿Qué edad tienes?

En primer lugar, como se comenta anteriormente se ha preguntado a los usuarios sobre su edad.

La totalidad de los usuarios encuestados se encontraba en la franja de edad de los 20 - 30 años. (Figura 79).

8.2.1.2. Valoración de la aplicación

¿Qué valoración general darías a la app?
33 respuestas

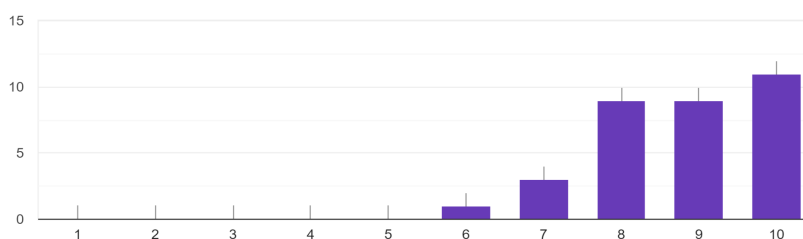


Figura 80 - Gráfico pregunta: ¿Qué valoración general darías a la app?

En esta pregunta de la figura 80 se solicita al usuario una valoración general de la aplicación. La distribución de las respuestas ha sido 6 (3%), 7 (9,1%), 8 (27,3%), 9 (27,3%), 10 (33,3%). El resto de las opciones (1 – 5) no han recibido ningún voto (0%).

8.2.1.3. Uso de la aplicación

¿Si tu centro universitario te proporcionara/hubiera proporcionado esta herramienta, ¿la utilizarías/hubieras utilizado?

33 respuestas

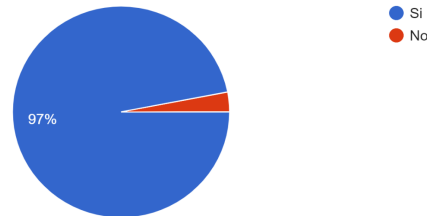


Figura 81 - Gráfico pregunta: ¿Habías utilizado esta herramienta en la universidad?

Se plantea a los usuarios la posibilidad de utilizar / haber utilizado en su periodo universitario Uni4All. La gran mayoría, el 97% han contestado que la utilizarían, y un 3% ha contestado que no la utilizarían. (Figura 81).

8.2.1.4. Motivos por los cuales usarían la aplicación

Si has respondido que si, comenta los motivos

23 respuestas

Si, me parece tremendamente útil para la universidad pues muchas veces existen complicaciones para buscar grupo, conocer gente, intercambiar conocimientos etc.

Permite establecer una comunicación más directa del temario dado en clase

Hubiera utilizado esta herramienta porque en algunas asignaturas he necesitado refuerzo y me hubiera sido útil tener acceso a una aplicación como esta, ya que todos salimos beneficiados al ayudarnos unos a otros.

Es más completa que otras que he utilizado anteriormente

Gran herramienta para conocer gente y ayudarnos unos a otros

Es una aplicación muy útil

me parece interesante el hecho de darnos clases unos a otros pues de este modo se saca lo mejor de cada uno y fomenta la sociabilidad.

Figura 82 - Gráfico pregunta: Si has respondido que sí, comenta los motivos

En esta sección (figura 82) se recogen algunas respuestas proporcionadas por los encuestados en las cuales mencionan los motivos por los que han respondido que sí anteriormente.

8.2.1.5. Interfaz visual

¿Te parece atractiva la interfaz visual de la app?

33 respuestas

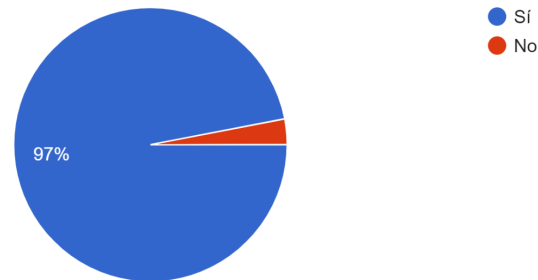


Figura 83 - Gráfico pregunta: ¿Te parece atractiva la interfaz visual de la app?

Los resultados que refleja el gráfico de la figura 83 son 97% de los usuarios ha contestado que si les resulta intuitiva y 3% no consideran que Uni4All sea intuitiva.

8.2.1.6. No intuitividad y posibles modificaciones.

Si has contestado no, comenta los motivos y la modificación

2 respuestas

Un poco minimalista pero esta bien, al final es intuitiva

Creo que necesitaría añadirse un modo noche, y algún elemento de accesibilidad.

Figura 84 - Pregunta: Si has contestado no, comenta los motivos y la modificación

Como continuación de la pregunta anterior se encuentra esta (figura 84), en la que los usuarios comentan los motivos por los cuales no encuentran intuitiva la aplicación y alguna posible modificación.

Entre las respuestas se encuentra la sugerencia de desarrollo de elementos de accesibilidad entre los que destaca el modo noche.

8.2.2. Registro/Inicio de sesión

8.2.2.1. ¿Es intuitivo el proceso Registro/Inicio de sesión?

¿Encuentras el registro / inicio de sesión intuitivo?

33 respuestas

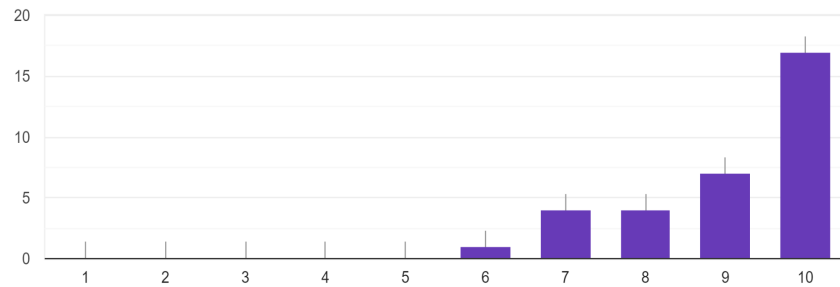


Figura 85 - Gráfico pregunta: ¿Encuentras la sección de servicios intuitiva?

Como se puede observar en el gráfico de la figura 85 la totalidad de los usuarios encuestados consideran que tanto el registro como el inicio de sesión de la aplicación es intuitiva.

Los resultados son 6 (3% de los usuarios), 7 (12,1% de los usuarios), 8 (12,1% de los usuarios), 9 (21,2% de los usuarios) y 10 (51,5% de los usuarios). Los valores por debajo del 6 no han recibido ningún voto por parte de los usuarios (0%).

8.2.2.2. Cambios y dificultades durante el proceso

¿Cambiarías algo o has encontrado alguna dificultad durante dichos procesos?

33 respuestas

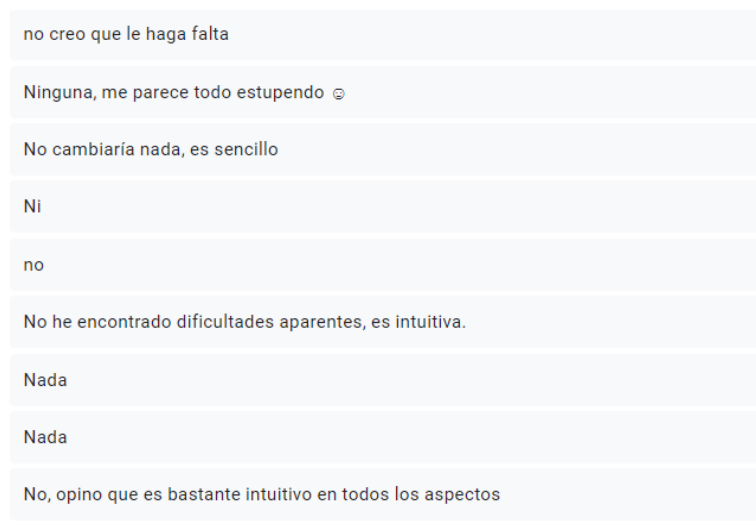


Figura 86 - Pregunta: ¿Cambiarías algo o has encontrado alguna dificultad?

En las respuestas de los usuarios de la figura 86, se observa que ninguno ha encontrado dificultades ni en el proceso de inicio de sesión ni de registro y no cambiarían nada al respecto.

8.2.3. Grupos

8.2.3.1. ¿Es intuitiva la sección de grupos?

¿Encuentras la sección de grupos intuitiva?
33 respuestas

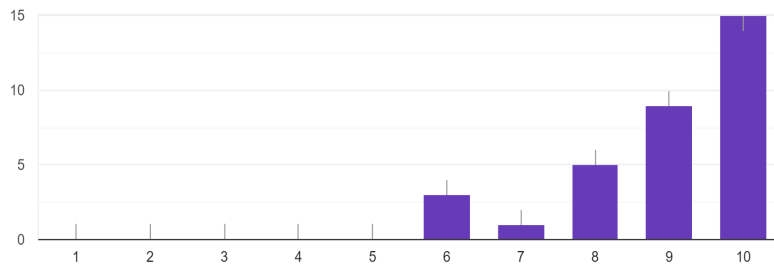


Figura 87 - Gráfico pregunta: ¿Encuentras la sección de grupos intuitiva?

En el gráfico de la figura 87 se observan los siguientes resultados 1-5 (0% de los votos), 6 (9,1% de los votos), 7 (3% de los votos), 8 (15,2% de los votos), 9 (27,3% de los votos), 10 (45,5% de los votos).

8.2.3.2. Cambios y modificaciones de la sección de grupos

¿Cambiarías o añadirías algo en los grupos?
33 respuestas



Figura 88 - Pregunta: ¿Cambiarías o añadirías algo en los grupos?

En esta pregunta de la figura 88 la mayoría de las respuestas coinciden en que no cambiarían nada.

En cambio, existe una en la que se puede leer que proponen la inclusión de un calendario. Por lo tanto, por lo general los usuarios están satisfechos con la sección y se valorará la propuesta del calendario.

8.2.4. Perfil

8.2.4.1. ¿Es intuitiva la sección de perfil?

¿Encuentras la sección de perfil intuitiva?
33 respuestas

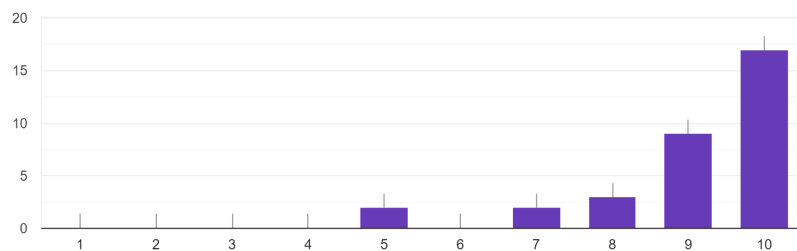


Figura 89 - Gráfico pregunta: ¿Encuentras la sección de perfil intuitiva?

Al observar el gráfico de la figura 89 se pueden extraer los siguientes datos. Ningún usuario ha votado los valores comprendidos entre el 1 y el 4 (0%), el 6,1% han votado 5, el 6,1% han votado 7, el 9,1% han votado 8, el 27,3 han votado 9 y el 51,5 han votado 10.

8.2.4.2. Cambios o modificaciones en la sección de perfil

¿Cambiarías o añadirías algo en el perfil?
33 respuestas

No
Nada
no
No cambiaría nada
No, tanto su ubicación como su uso me son familiares, se parecen a los de otras apps
mas información para verificar la cuenta
No, su ubicación es clara

Figura 90 - Pregunta: ¿Cambiarías o añadirías algo en el perfil?

Se muestra en la figura 90 la pregunta de si realizarían una modificación y en caso positivo una breve explicación sobre la

misma. Aparece la posibilidad de más información para verificar la cuenta, se analizará y valorará dicha posibilidad.

8.2.5. Foros

8.2.5.1. ¿Es intuitiva la sección de foros?

¿Encuentras la sección de foros intuitiva?
33 respuestas

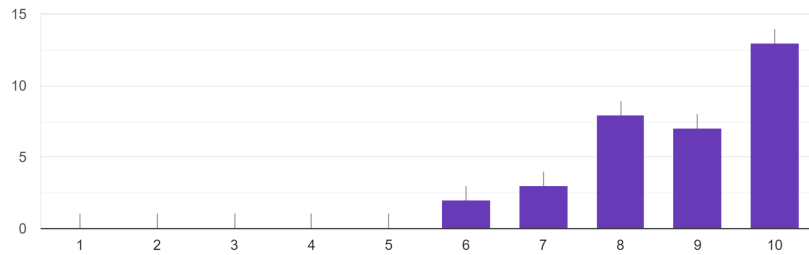


Figura 91 - Gráfico pregunta: ¿Encuentras la sección de foros intuitiva?

Los resultados del gráfico de la figura 91 muestran que ningún usuario ha valorado la pregunta con una nota inferior a 6, por lo que 1-5 (0% de los votos de los usuarios), el 6,1% han votado 6, el 9,1% han votado 7, el 24,2% han votado 8, el 21,2% han votado 9, el 39,4% han votado 10.

8.2.5.2. Cambios o modificaciones en la sección de foros

¿Cambiarías o añadirías algo en foros?
33 respuestas

que haya notificaciones
nada
Se podría añadir opción de menciones o hilos de mensajes
No
No, creo que es una sección que puede ayudar mucho a los usuarios.
no
Mejor diseño
No añadiría nada
NO

Figura 92 - Pregunta: ¿Cambiarías o añadirías algo en foros?

Entre estas respuestas de la figura 92 se pueden encontrar varias sugerencias, añadir notificaciones, implementar la opción de menciones o hilos de mensajes y mejorar el diseño. Aunque la mayoría de las respuestas consideran que no sería necesario realizar ningún cambio existen propuestas, las cuales se sopesará incluir en un futuro.

8.2.6. Conclusiones finales y sugerencias:

8.2.6.1. Sección más útil

¿Cuál te parece la sección mas útil?

33 respuestas

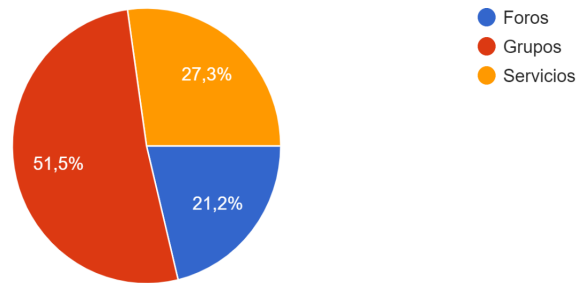


Figura 93 - Gráfico pregunta: ¿Cuál te parece la sección más útil?

Al observar el grafico de la figura 93, los resultados de la encuesta muestran que los usuarios que consideran más útil es la de grupos son un 51,5%, seguida de la de servicios 27,3%, y por último la sección de foros 21,2 % de los votos de los usuarios.

8.2.6.2. Sección menos útil

¿Y la que menos útil?

31 respuestas

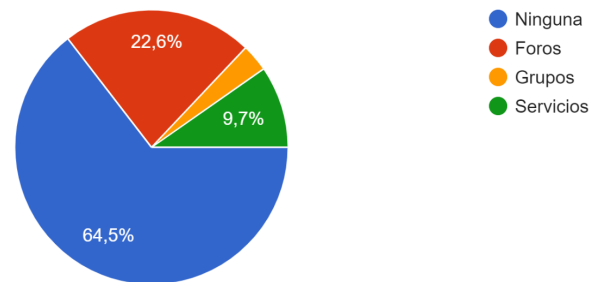


Figura 94 - Gráfico pregunta: ¿Y la que menos útil?

Se puede observar el gráfico de la figura 94, que los resultados de la encuesta muestran que los usuarios que consideran menos útil es la de foros con un 22,6%, seguida de la de servicios 9,7%, y por último la sección de grupos 3,2 % de los votos de los usuarios. El resto de los votos de los usuarios están destinados a la opción de Ninguna con un 64,5% de los votos.

8.2.6.3. Sugerencias

¿Tienes alguna sugerencia?

5 respuestas

No

Verificar los perfiles de alguna forma

Figura 95 - Pregunta: ¿Tienes alguna sugerencia?

Como última pregunta se ha planteado la posibilidad de comentar alguna sugerencia para la aplicación (figura 95), la mayoría de los usuarios no han propuesto nada nuevo. No obstante, se repite la opción de verificar los perfiles tal y como se comenta en el apartado de Perfil.

Capítulo 9. Conclusiones y trabajo futuro.

9.1. Conclusiones

En este proyecto se ha desarrollado una aplicación que permitirá a los estudiantes desarrollar su vida universitaria centralizando en 1 solo lugar la gestión grupal de todos sus proyectos y necesidades universitarias. **Uni4All** se ha creado con la intención de conectar estudiantes con otros estudiantes, intercambiando ayudas en distintas asignaturas, formando grupos y gestionando proyectos mediante mensajes y acceso a repositorios donde almacenar archivos. Finalmente, usando foros, resolver cualquier duda que pueda surgir, comentar temas de interés académico y muchísimo más.

Al comenzar a desarrollar el trabajo de fin de grado se plantearon una serie de objetivos sobre cómo sería aplicación que finalmente se ha desarrollado. Entre los objetivos estaba el intercambio de ayuda entre estudiantes, la formación de grupos (para prácticas u otras necesidades académicas) y el uso de foros para resolución de dudas y la discusión de otros temas de interés. Las ventajas que se pueden lograr gracias al uso de esta aplicación son varias:

- En vez de utilizar foros de temática universal para tratar sobre temas universitarios personales se ofrece la posibilidad de utilizar foros especializados en temas universitarios.
- Formación de grupos de trabajo de una manera fácil y sencilla, con posibilidad de gestionar esos trabajos grupales desde una sola aplicación, sin necesidad de estar investigando durante varios días para anexionarse a un grupo.
- Uso de 1 sola aplicación para la comunicación, reduciendo la necesidad de mensajear por distintas plataformas.
- Gestión de enlaces repositorios de código, de almacenamiento y de videoconferencias en 1 solo lugar frente al uso de desde distintas aplicaciones.
- Posibilidad de pedir y ofrecer ayuda en asignaturas y otros temas académicos utilizando un portal común, evitando así el colgado ofertas, o demandas, de clases y otros temas por tablones (de la calle) y otros medios.

La aplicación que se ha desarrollado en este trabajo de fin de grado cubre todos estos objetivos y ofrece grandes ventajas para su uso. El código de la aplicación puede obtenerse a través de este enlace:



<https://github.com/VaqueroGitHub/TFG>

9.2. Trabajo futuro

La funcionalidad de la aplicación podría mejorarse de la siguiente manera:

- Integración de monedas:
La aplicación inicialmente se pensó con un mayor contenido de ayuda mutua, en ambos sentidos, el usuario 1 ofrece ayuda en el tema 1 y el usuario 2 a cambio ofrece ayuda en el tema 2. Pero ¿y si el usuario 1 no necesita la ayuda en el tema 2? Al ofrecer un servicio se esperaba recibir otro a cambio, una ayuda mutua, pero se ha visto que hay casos en los que no se puede ofrecer lo que se pide a cambio. Sin embargo, sí puede haber un usuario 3 que ofrezca ese servicio pedido, pero no necesite lo que ofrece el usuario 1, entrando en juego el sistema de trueque de esa ayuda por un ítem no tangible (monedas).

Por lo tanto, para un trabajo futuro se implementaría un sistema de precio por servicio (moneda que se le quita al que recibe el servicio, el beneficiado, y que se le suma al que lo oferta), con comprobaciones de monedas disponibles antes de pedir un servicio.

En la figura 96 hemos incluido una idea de cómo se vería en la interfaz de la aplicación. En realidad, gran parte del desarrollo se haría en el back, no en el front.

También creemos que se debería incluir en el perfil un indicador del número de monedas, pero habría un debate ético sobre si debiese ser público para el resto de usuario que visitar un perfil.



Figura 96 - Coste de los servicios

- Sistema de recompensas por inicio de sesión continuado:
Actualmente, al no haber monedas, no existe una recompensa que dar, pero, en caso de introducirlas se podría dar monedas cada varios días de conexión continuada para que así las personas que usan activamente la aplicación puedan pedir más servicios, lo cual es beneficioso tanto para ellos como para que los reciben las monedas, pues podrán pedir ayuda también en otras asignaturas.
- Contraoferta de servicios:
En caso de introducir un sistema de pagos por servicio sería interesante la posibilidad de negociar las monedas. Puede que el usuario 2 no tenga suficientes monedas para que el usuario 1 le ayude a preparar un trabajo, o que el precio de la oferta de ayuda sea muy elevado para el usuario 2, y este ofrezca por recibirlo algo menos de monedas.
- Módulo de soporte: A pesar de haber creado una app muy intuitiva, si un usuario tiene alguna duda sobre la funcionalidad de la aplicación no existe ningún apartado de ayuda o preguntas frecuentes. Es cierto que podría abrir un post en el foro preguntando sobre algún apartado de la aplicación, pero para futuras versiones se debería crear un apartado de ayuda y la posibilidad de contactar al desarrollador para proporcionar soporte real.

- **Reporte de errores:**
Ahora mismo la aplicación no ofrece ningún sistema de reporte de errores a los desarrolladores, ni de ofrecer feedback (más allá de las encuestas que hemos hecho durante el último mes sobre la funcionalidad de la aplicación). Se podría desarrollar una funcionalidad que permita a cualquier usuario emitir incidencias en caso de que la funcionalidad de la aplicación o la consistencia de sus datos se vea comprometida.
- **Ampliación del sistema de administración:**
En este momento, cualquier cambio que haya que realizarse en la inclusión de administradores en la aplicación se debe de realizar directamente sobre la base de datos, en este sentido, se podría desarrollar en el módulo de administración de la app un sistema rápido y sencillo para aumentar los privilegios de usuarios estándar, por ejemplo, por un administrador ya existente.
- **Filtrado de posts por “No respondida”:**
Actualmente se pueden buscar posts por el título, pero no hay una forma de filtrar todos los posts por “No respondida”, así podría facilitarse la búsqueda de posts que aún no han recibido respuesta, con la intención de responderlos.



Figura 97 - Situación actual de búsqueda

- Selección de la respuesta útil en foros:
Actualmente cuando un usuario da de alta un post aparece como no respondido. Sin embargo, en cuanto recibe una respuesta (aunque no responda realmente a la pregunta o al tema) el post se considera automáticamente como respondido, y así lo refleja la interfaz.



Figura 98 - Post sin responder



Figura 99 - Post respondido

En una futura versión se podría seleccionar la respuesta “útil”, en caso de haberla, para que se muestre arriba del todo al navegar dentro del post, y así facilitar la solución de dudas a personas con el mismo problema.

También podría no considerarse un post como contestado si lo contesta el mismo autor, que es lo que se ha hecho. La razón de esto es que muchas veces uno mismo encuentra solución a una duda que ha publicado, y puede que más gente la tenga en el futuro, pero también puede añadir un mensaje sin ser una respuesta y que ese Post pase desapercibido como que ya tiene una respuesta y no requiera atención. Es un tema para tratar en una versión futura muy interesante.

- Citado de posts en foros:
Actualmente no se puede citar una respuesta para responder, dentro de un post, a otra persona que no sea el OP (posteador original). Se podría incluir en una versión posterior la posibilidad de citar respuestas de alguna manera para facilitar el hilo de la conversación a usuarios que leen todos los comentarios.

En la figura 100 se hace una referencia a donde podría incluirse el botón de citado de una respuesta.



Figura 100 - Posible ubicación del botón de citado en respuestas

- Editado de respuestas en posts: Actualmente se puede editar un post creado en un foro, pero no las respuestas de este. En una versión futura se podría editar también la respuesta, aunque también podría incluirse un sistema de etiquetas de edición para facilitar la indicación de que se ha editado recientemente y no es la respuesta original.

En la figura 101 se referencia donde se podría poner un lápiz (el mismo que el de editado del post al que se ha respondido).

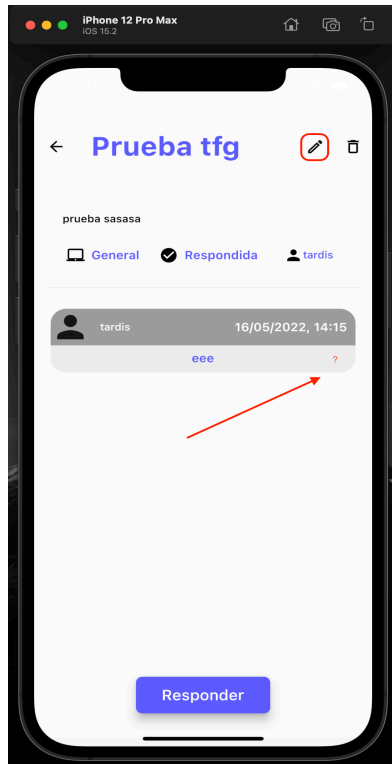


Figura 101 - Situación actual del editado y ubicación para editado de respuesta

Chapter 9: Conclusions and future work

9.1. Conclusions

In this project, an application has been developed that will allow students to develop their university life by centralizing in one place the group management of all their university projects and needs. Uni4All has been created with the intention of connecting students with other students, exchanging help in different subjects, forming groups and managing projects through messages and access to repositories to store files. Finally, using forums, answer any questions that may arise, comment on topics of academic interest and much more.

When beginning to develop the end-of-degree project, a series of objectives were set on what the application would be like, which has finally been developed. Among the objectives was the exchange of help between students, the formation of groups (for practices or other academic needs) and the use of forums to resolve doubts and discuss other topics of interest. The advantages that can be achieved thanks to the use of this application are several:

- Instead of using forums with a universal theme to deal with personal university issues, the possibility of using specialized forums on university issues is offered.
- Formation of work groups in an easy and simple way, with the possibility of managing these group works from a single application, without the need to investigate for several days to join a group.
- Use of a single application for communication, reducing the need to message on different platforms.
- Management of links to code repositories, storage and videoconferences in a single place compared to the use of from different applications.
- Possibility of requesting and offering help in subjects and other academic subjects using a common portal, thus avoiding posting offers, or requests, of classes and other subjects by boards (from the street) and other ways.

La aplicación que se ha desarrollado en este trabajo de fin de grado cubre todos estos objetivos y ofrece grandes ventajas para su uso. El código de la aplicación puede obtenerse a través de este enlace:



<https://github.com/VaqueroGitHub/TFG>

9.2. Future work

The functionality of the app could be improved as follows:

- Integration of coins:
The application was initially thought with a more sophisticated content of mutual help, in both directions, user 1 offers help on topic 1 and user 2 in return offers help on topic 2. But what happens when user 1 doesn't need help on topic 2? By offering a service, it is expected to receive another in return, mutual help, but it has been seen that there are situations where what is asked for in return cannot be offered. However, there may be a user 3 who offers that requested service, but does not need what user 1 offers, there's where the need of a system of trading would really start coming into mind.

Therefore, for future work, a service price system would be implemented (money, coins, that are taken from the buyer of the service, the beneficiary, and added to the one who offers it), with a money check given before requesting a service.



Figura 102 - Cost of services

- **System of rewards for continued login:**
Currently, as there are no coins, there is no reward to be given, but, if money is introduced, coins could be given every several days of continued connection so that people who actively use the application are rewarded. As a result, students could ask for more services, which is beneficial both for them and for those who receive the coins, since they can also ask for help in other subjects.
- **Counteroffer of services:**
In case of introducing a system of payment for services, the possibility of coin negotiating would be interesting. It may happen that user 2 does not have enough coins for user 1 to help him prepare a job, or that the price of the help offer is too high for user 2, and he offers to receive it a little less coin.
- **Support module:**
Despite having created a very intuitive app, if a user has any questions about the functionality of the application, there is no help section or frequently asked questions. It is true that the user could open a post in the forum asking about some topic or question about the application, but for a future version release a help section could be created with the possibility of contacting the developer to provide real support.

- **Bug Reporting:**
Right now, the app doesn't offer any bug reporting system to developers or offer any kind of feedback (beyond the surveys we've done over the last month about the app's functionality). A functionality could be developed that allowed any user to issue incidents in the possible situation where the functionality of the application or the consistency of its data is compromised.
- **Extension of the administration system:**
At this time, any change related to the inclusion of administrators in the application must be made directly on the database, in this sense, it could be developed over the administration module of the application a quick and easy system to increase the privileges of standard users, for example by an existing administrator.
- **Filtering posts by “Not answered”:**
Currently you can search by post’s titles, but there is no way to filter all posts by “Not answered”, so it could be easier to find posts that have not yet received a response, with the intention of answering them.



Figura 103 - Current search situation

- Selection of the useful answer in forums:
Currently, when a user registers a post, it appears as an unanswered post. However, as soon as a reply is received (even if it doesn't actually reply to the question or topic) the post is automatically considered answered, and the interface reflects this.



Figura 105 - Unanswered post



Figura 104 - Answered post

In a future version, the “useful” answer could be selected, if there is one, so that it is displayed at the top when browsing within the post, and thus facilitate the solution of doubts to users with the same problem.

- Quoted from posts in forums:
Currently you cannot quote an answer to reply, inside a post, to someone other than the OP (original poster). The possibility of quoting answers in some way could be included in a future version release, helping users who read all the comments follow the conversation thread.

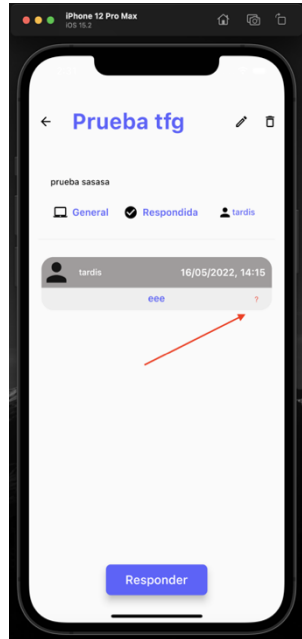


Figura 106 - Possible location of the quote button in answers

- Editing responses in posts:

Currently you can edit a post created in a forum, but not its responses. In a future version the answer could also be edited, although an edited tagging system could also be included to make it easier to indicate that it was recently edited and not the original answer.

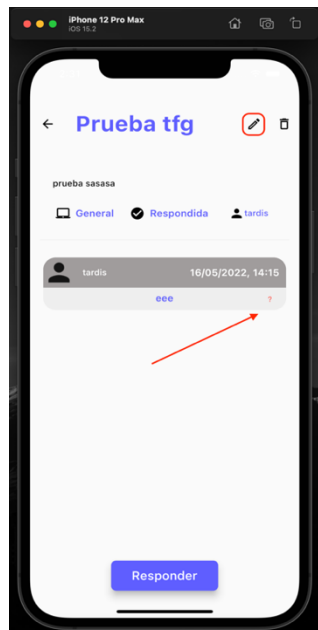


Figura 107 - Current status of post editing and location for response editing

Capítulo 10: Aportaciones individuales

En este apartado se detallan las actividades desarrolladas por cada uno de los miembros del equipo.

Carlos Luengo Heras:

- **Especificación de requisitos:**
Fue parte activa de tanto las reuniones diarias y semanales como de la toma de decisiones en referencia a las tecnologías implicadas en el desarrollo de Uni4All. También participó en las reuniones periódicas con el tutor del TFG en las que se trataban diferentes aspectos de diseño y de funcionalidad de la propia aplicación.
- **Diseño:**
A su vez también fue parte fundamental a la hora de diseñar la interfaz de la app, las entidades centrales y decidir las funciones básicas a desarrollar para obtener un resultado funcional y atractivo.
- **Base de datos:**
Fue partícipe de la elaboración de los diagramas entidad - relación sobre los cuales se fundamenta la aplicación. Al mismo tiempo también colaboró a la hora de crear los distintos modelos de datos.
- **Desarrollo:**
Colaboró tanto en el desarrollo del frontend (diseño y desarrollo de las vistas e interfaz en general) como en el backend (desarrollo del controller y aplicación de los patrones de diseño).
- **Memoria:**
La memoria fue elaborada de manera conjunta en su mayoría. Teniendo esto en cuenta, se focalizó en el apartado de historia del arte, casos de uso y el diseño de la aplicación.

Javier Vaquero Fernández:

- **Especificación de requisitos:**

Fue parte activa de tanto las reuniones diarias y semanales como de la toma de decisiones en referencia a las tecnologías implicadas en el desarrollo de Uni4All. También participó en las reuniones periódicas con el tutor del TFG en las que se trataban diferentes aspectos de diseño y de funcionalidad de la propia aplicación.

- **Diseño:**

A su vez también fue parte fundamental a la hora de diseñar la interfaz de la app, las entidades centrales y decidir las funciones básicas a desarrollar para obtener un resultado funcional y atractivo.

- **Base de datos:**

Fue partícipe de la elaboración de los diagramas entidad - relación sobre los cuales se fundamenta la aplicación. Al mismo tiempo también colaboró a la hora de crear los distintos modelos de datos.

- **Desarrollo:**

Colaboró tanto en el desarrollo del frontend (diseño y desarrollo de las vistas e interfaz en general) como en el backend (desarrollo del controller y aplicación de los patrones de diseño).

- **Memoria:**

Fue elaborada de manera conjunta en su mayoría. Teniendo esto en cuenta, se focalizó en recopilar información por parte de los usuarios y recabarla a través de preguntas formuladas mediante Google Forms y también de las herramientas empleadas

Joaquín Asensio Manzano:

- **Especificación de requisitos:**

Fue parte activa de tanto las reuniones diarias y semanales como de la toma de decisiones en referencia a las tecnologías implicadas en el desarrollo de Uni4All. También participó en las reuniones periódicas con el tutor del TFG en las que se trataban diferentes aspectos de diseño y de funcionalidad de la propia aplicación.
- **Diseño:**

A su vez también fue parte fundamental a la hora de diseñar la interfaz de la app, las entidades centrales y decidir las funciones básicas a desarrollar para obtener un resultado funcional y atractivo.
- **Base de datos:**

Fue partícipe de la elaboración de los diagramas entidad - relación sobre los cuales se fundamenta la aplicación. Al mismo tiempo también colaboró a la hora de crear los distintos modelos de datos.
- **Desarrollo:**

Colaboró tanto en el desarrollo del frontend (diseño y desarrollo de las vistas e interfaz en general) como en el backend (desarrollo del controller y aplicación de los patrones de diseño).
- **Memoria:**

Fue elaborada de manera conjunta en su mayoría. Teniendo esto en cuenta, se focalizó en desarrollar el apartado de los patrones empleados, del diseño y de la conclusión.

Bibliografía

- [1] ¿Tienes sal?: <https://tienes-sal.es/about>
- [2] Wuolah: <https://www.wuolah.com/>
- [3] Wuolah, Facultad informática UCM: <https://www.wuolah.com/es/universidad-complutense-de-madrid/ucm-facultad-de-informatica>
- [4] Meetup: <https://www.meetup.com/about/>
- [5] TeAyudo: <https://www.teayudo.io/>
- [6] TeAyudo, preguntas y respuestas: <https://www.teayudo.io/preguntas-frecuentes>
- [7] Android Studio: https://es.wikipedia.org/wiki/Android_Studio
- [8] Xcode: <https://es.wikipedia.org/wiki/Xcode>
- [9] Visual Studio Code: https://es.wikipedia.org/wiki/Visual_Studio_Code
- [10] Firebase: <https://es.wikipedia.org/wiki/Firebase>
- [11] Flutter: <https://aurestic.es/que-es-flutter/>
- [12] GitHub: <https://www.webempresa.com/hosting/que-es-github.html>
- [13] Dart: <https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart>
- [14] Google Drive: <https://www.caracteristicas.co/google-drive/>
- [15] diagrams.net: <https://en.wikipedia.org/wiki/Diagrams.net>
<https://bigcode.es/introduccion-a-la-arquitectura-de-aplicaciones-con-flutter/>
<https://medium.com/flutter-madrid/arquitecturas-y-patrones-de-dise%C3%B1o-en-flutter-cap%C3%ADtulo-1-mvvm-591ad3d63bb>
- [17] sobreingeniería: "¡Deja de hacer sobreingeniería! Archivado el 16 de marzo de 2016 en [Wayback Machine](https://www.waybackmachine.org/).", revista *Desarrollo de software*, Joshua Kerievsky, abril de 2002
- [18] MVVM: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>
- [19] Service Layer: https://en.wikipedia.org/wiki/Service_layer_pattern
- [20] Repository: <https://codingsight.com/entity-framework-antipattern-repository/>
- [21] Inyección dependencias: <https://rubenjromo.com/flutter-arquitectura-limpia-13-inyeccion-de-dependencia/>
- [22] DTO: https://en.wikipedia.org/wiki/Data_transfer_object
- [23] UUID: «uuid — UUID objects according to RFC 4122» [uuid — objetos UUID de acuerdo con RFC 4122] (html). *Python Software Foundation* (en inglés)
- [24] Keystore: <https://developer.android.com/training/articles/keystore.html>
- [25] keychain: https://developer.apple.com/documentation/security/keychain_services#//apple_ref/doc/uid/TP30000897-CH203-TP1
- [26] Clave criptográfica: https://es.wikipedia.org/wiki/Clave_%28criptograf%C3%ADa%29
- [27] Flutter Secure storage: https://pub.dev/packages/flutter_secure_storage
- [28] Get_it singleton: https://pub.dev/packages/get_it
- [29] Cloud: <https://www.cloudflare.com/learning/cloud/what-is-the-cloud>

ANEXO I. Guía del usuario

Una vez instalada la aplicación y ejecutada por primera vez obtendremos un menú de inicio con dos botones, registro y login. Las figuras 108, 109 y 110 muestran las 3 pantallas por las que debe pasar un usuario para registrarse e iniciar sesión por primera vez (después será automático el login).



Figura 108 - Pantalla login



Figura 109 - Pantalla registro

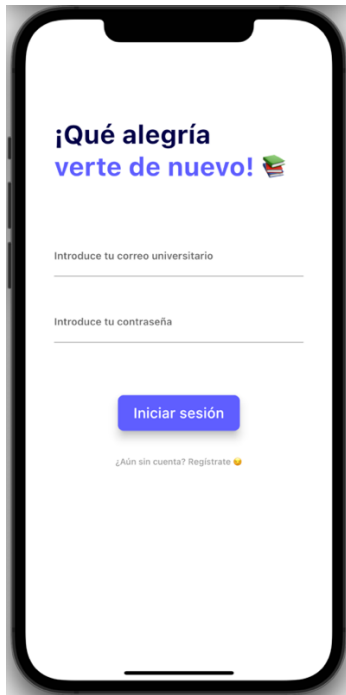


Figura 110 - Pantalla inicio

Una vez iniciado sesión se llegará a la pantalla principal de la aplicación (figura 111). Tras hacer login el usuario le aparece su perfil de usuario, con su información personal y unas estadísticas de: los servicios que ha solicitado y ofertado, los grupos a los que pertenece y los posts en los que ha participado.



Figura 111 - Dashboard usuario registrado

Desde esta pantalla se puede navegar mediante la barra de navegación inferior a las distintas pantallas de la aplicación, entre ellas, las 3 principales funcionalidades: Servicios, Foros, Grupos; y Perfil. También aparece un botón centrado en esta barra de navegación en el cual podremos volver a esta pantalla inicial, figura 112.

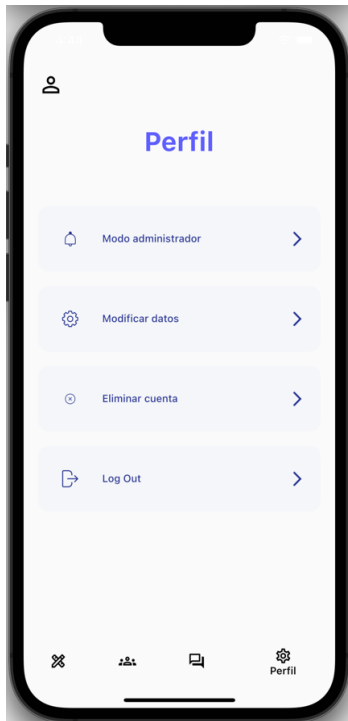


Figura 112 - Pantalla perfil usuario

En el apartado de perfil, el usuario puede modificar sus datos personales, ya sea su imagen de perfil, sus credenciales de sesión, o su información de usuario.

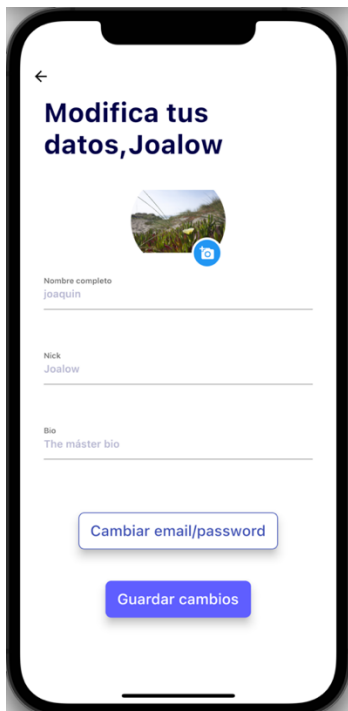


Figura 113 - Modificado de datos

También, el usuario tendrá acceso al apartado ‘Modo administrador’, en el cual podrá administrar las diferentes entidades: baja y modificación de todos los usuarios, posts, grupos y servicios.

En las figuras 114, 115, 116 y 117 se recogen los 4 módulos de administración:

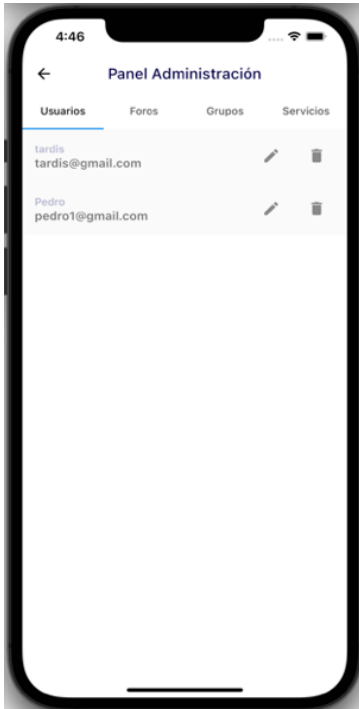


Figura 114 - Panel usuario

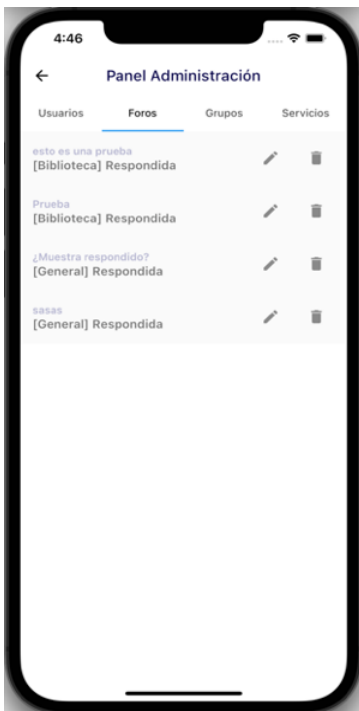


Figura 115 - Panel foros

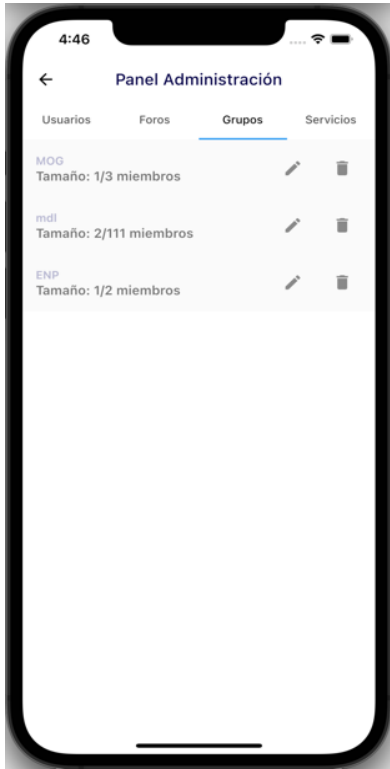


Figura 116 - Panel grupos

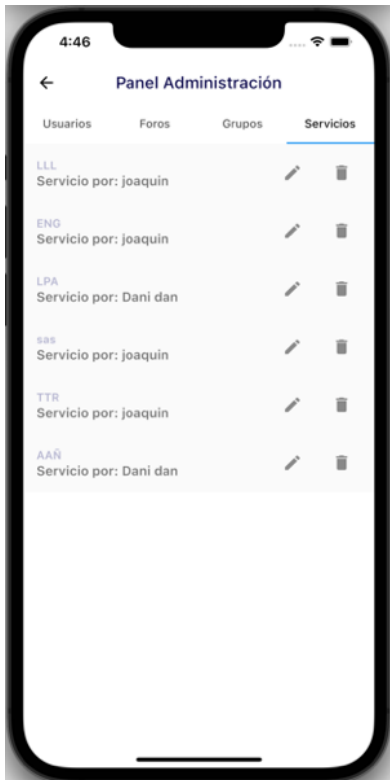


Figura 117 - Panel servicios

Por último, el usuario podría eliminar su cuenta, borrando todos los datos y credenciales del usuario sin posibilidad de volver a entrar en la aplicación, y hacer log out de la aplicación, en la figura 118.

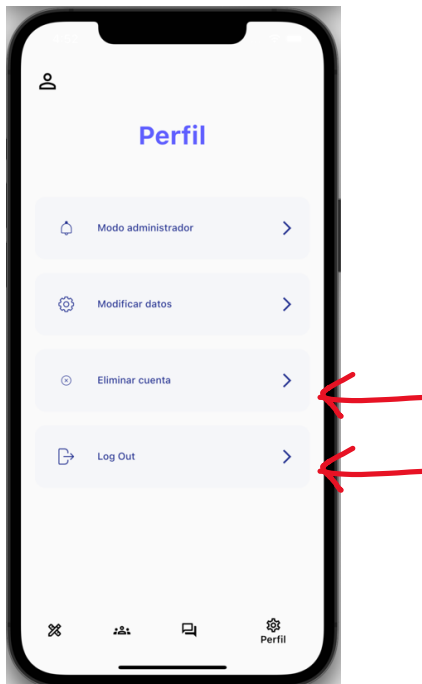


Figura 118 - Eliminar cuenta y Log out

En el apartado de foros aparecen las secciones almacenadas en la base de datos actualmente: General, Biblioteca, Asignaturas, Erasmus, Cafetería. Con la posibilidad de que el administrador dé de alta más manualmente bajo petición. (Figura 119).

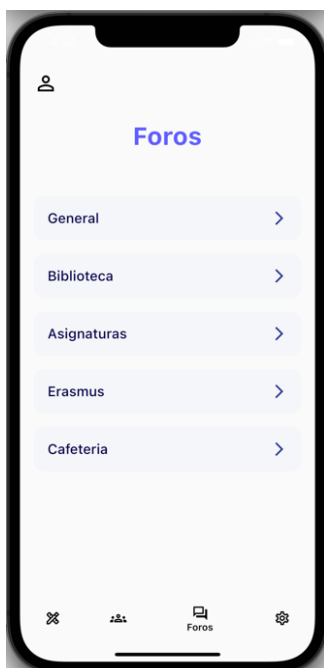


Figura 119 - Vista principal foros

Si le damos a una de estas secciones nos llevará a una lista de todos los posts que almacena esta sección, con la posibilidad de añadir uno nuevo y buscar por título, mediante un buscador en la barra superior de la pantalla.



Figura 120 - Posts de un subforo

Si le damos a un post, nos llevará a la pantalla principal de post, en la cual aparecerá la información relativa a este: Título, Cuerpo del post, Sección a la que pertenece, si tiene respuestas, el usuario que lo creó.

En la parte de abajo, aparecería una lista de todas las respuestas que ha recibido este post, con la posibilidad de añadir una nueva. En la respuesta se visualiza la persona que ha escrito la respuesta, el mensaje de respuesta, y la fecha en la que se envió. Los mensajes aparecerán en orden de más reciente. (Figura 121)



Figura 121 - Visualización hilo y respuestas

En el apartado de Grupos aparecen en la pantalla principal los grupos a los que tienes acceso, los que has creado y en los que participas. También aparecen las opciones de crear un nuevo grupo y de acceder a una nueva pantalla con la lista de todos los grupos existentes.

Si vamos a esta pantalla de todos los grupos existentes, aparecerán aquellos en los que participemos y no, con la posibilidad de unirnos a los que no. También en la barra superior tenemos la opción de buscar por código.

Si pulsamos un grupo, nos aparecería la información relativa a él: el número de usuarios que participan y máximo, con la posibilidad de acceder a ellos, el código de asignatura del grupo, el usuario que creó el grupo, la descripción del grupo, las urls si existiesen de Github y Drive (las cuales podríamos hacer click y que nos redirija a estas páginas con el navegador por defecto del terminal), y por último un apartado de chat en el que se podrán intercambiar mensajes entre usuarios.



Figura 122 - Mostrado de grupo



Figura 123 - Dashboard grupos



Figura 124 - Selección de grupos

También, el usuario creador del grupo tendrá la opción de editar y eliminar el grupo, y cualquier otro usuario podrá unirse y/o salirse del mismo.

Finalmente, la guía de usuario acaba tratando el módulo de servicios. Se accede, como en los anteriores, pulsando en la barra de navegación en servicios.

En el apartado de Servicios aparecen en la pantalla principal los servicios a los que tienes acceso, los que has creado y en los que participas. También aparecen las opciones de crear un nuevo servicio y de acceder a una nueva pantalla con la lista de todos los servicios existentes.

Si vamos a esta pantalla de todos los servicios existentes, aparecerán aquellos que hayamos ofertado, que hayamos adquirido y no, con la posibilidad de adquirir los que no. También en la barra superior tenemos la opción de buscar por código.

Si pulsamos un servicio, nos aparecería la información relativa a él: el usuario que ha adquirido el servicio, el código de del servicio, el usuario que ha creado el servicio, la descripción del servicio, una url con el link a una conferencia de Zoom (la cuales podríamos hacer click y que nos redirija a esta sesión en el navegador por defecto del terminal, si el servicio se realiza vía online), y por último un apartado de chat en el que se podrán intercambiar mensajes entre usuarios.

También, el usuario creador del servicio tendrá la opción de editar y eliminar el servicio, y cualquier otro usuario podrá unirse y/o salirse del mismo, si otro usuario no lo ha adquirido.



Figura 125 - Búsqueda de servicios



Figura 126 - Creación de servicio



Figura 127 - Detalle de servicio propio