



UNIVERSIDAD DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA GRADUADO EN INGENIERÍA DEL SOFTWARE

SCOUT CONNECTIONS

Realizado por
Iván Hijano Pérez

Tutorizado por
José Luis Pastrana Brincones

Departamento
Lenguajes y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO 2022

Resumen

El desarrollo de este proyecto está enfocado en la creación y el desarrollo de una aplicación móvil para la Asociación “Grupo Scout 415 Los Olivos”. Se trata de una organización civil de carácter educativo para jóvenes y niños/as, siendo socio de Scouts de Andalucía y que se adhiere a la Constitución Mundial Scout. Actualmente cuenta con 180 socios, y el fin del Grupo es la formación integral a través del método Escultismo, potenciando la responsabilidad, la libertad, la concienciación en temas sociales y el servicio a los demás sin ningún tipo de discriminación.

“Scout Connections” es una aplicación de carácter social para dispositivos móviles, la cual permite la comunicación e intercambio de información entre los miembros de la comunidad de la Asociación “Grupo Scout 415 Los Olivos”. Cada miembro puede registrarse o iniciar/cerrar sesión en ella pudiendo adicionalmente configurar su perfil, el cual también será visible para los demás miembros. El *software* dispone de publicaciones de post al estilo red social con “me gusta” y comentarios. Incluye también la posibilidad de mantener conversaciones entre usuarios de la organización, además de la creación de grupos para compartir información de manera más sencilla. Asimismo, permite señalar futuros eventos de la asociación en un calendario que estará a disposición de todos los integrantes.

Para su desarrollo, se ha utilizado Android Studio para el *framework*, Kotlin como lenguaje y Firebase para la base de datos. La metodología usada durante el proyecto es SCRUM, una metodología ágil utilizada profesionalmente basada en un ciclo de vida iterativo e incremental, con lo que se consiguen revisiones periódicas y posibles evoluciones de los requisitos y funcionalidades.

Palabras clave

Aplicación móvil, Android, Social, Kotlin, Scouts.

Abstract

The development of this project is focused on the creation and development of a mobile application for the "Scout Group 415 Los Olivos " Association. It is a civil organization of an educational nature for young people and children, being a member of Scouts de Andalucía and that belongs to the World Scout Constitution. It currently has 180 members, and the goal of the Group is comprehensive training through the Scouting method, promoting responsibility, freedom, awareness of social issues and service to others without any discrimination.

"Scout Connections" is a social application for mobile devices, which allows communication and exchange of information between members of the community of the "Scout Group 415 Los Olivos" Association. Each member can register or log in/log out in it and can additionally configure her profile, which will also be visible to other members. The software has social network-style post publications with “likes” and comments. It also includes the possibility of holding conversations between users of the organization, in addition to the creation of groups to share information more easily. Likewise, it allows pointing out future events of the association in a calendar that will be available to all members.

For its development, Android Studio has been used for the framework, Kotlin as the language and Firebase for the database. The methodology used during the project is SCRUM, an agile methodology used professionally based on an iterative and incremental life cycle, with which periodic reviews and possible evolutions of the requirements and functionalities are achieved.

Keywords

Mobile Application, Android, Social, Kotlin, Scouts.

Índice

Resumen.....	1
Palabras clave	2
Abstract	3
Keywords	4
Índice	5
Introducción.....	7
1.1 Motivación.....	7
1.2 Objetivos	7
1.3 Estructura de la memoria	9
Tecnologías utilizadas.....	13
2.1 Sistema operativo	13
2.2 Lenguaje de programación	17
2.3 Entorno de desarrollo	19
2.4 Sistema de base de datos	23
Metodología.....	26
Fases del desarrollo	31
3.1 Planificación	31
3.2 Análisis de requisitos	34
3.3 Diseño y arquitectura	39
3.4 Programación e implementación.....	46

3.4.1	Usuario.....	50
3.4.2	Publicaciones.....	55
3.4.3	Perfil	58
3.4.4	Social.....	59
3.4.5	Eventos	61
3.5	Pruebas y revisión	63
3.6	Instalación y despliegue	64
3.7	Documentación	65
Conclusiones y Líneas Futuras		66
Bibliografía.....		68
Apéndices		72
	Manual de Usuario.....	72

Introducción

1.1 Motivación

Desde que los teléfonos inteligentes o *smartphones* salieron al mercado a fines de la década de los 2000, su uso y avances no hacen más que aumentar. Vivimos en una época en la que cualquier acción cotidiana puede realizarse desde un *smartphone*, tal como escuchar música, comunicarse a través de redes sociales y consultar noticias, el tiempo o el tráfico, entre muchas otras. Actualmente, cualquier empresa y/o negocio dispone de una aplicación o una página web, y es por ello por lo que la Asociación “Grupo Scout 415 Los Olivos” necesita un soporte para una comunidad de más de 180 miembros.

Este proyecto pretende proveer una solución a esta organización de la mejor manera posible, la cual es a través de una aplicación móvil completa con tantas características y funcionalidades como las que se podría encontrar en una aplicación ya existente en el mercado. El *software* permitirá una comunicación completa para todos los usuarios, a través de las tecnologías modernas, contando con interfaces amenas, sencillas y familiares, las cuales permitirán un fácil aprendizaje y uso.

1.2 Objetivos

Este trabajo de fin de grado (TFG) tiene varios objetivos fundamentales, los cuales podrían dividirse en dos grupos: curriculares y de desarrollo. Entre los objetivos

curriculares que se quieren cubrir durante el desarrollo del proyecto se encuentran las competencias que todo ingeniero informático debería poseer:

- **Capacidad para elaborar y gestionar proyectos:** Todo ingeniero debe planificar y organizar bien el proyecto que lleve a cabo (establecer objetivos y prioridades, organizar el tiempo de realización, etc.). Para llevar esta competencia a cabo se ha hecho uso de la metodología SCRUM, la cual será explicada posteriormente.
- **Competencias de búsqueda de información y documentación:** Realizar un análisis y síntesis cuando sea necesario valorar que acción debe tomarse y ser capaz de seleccionar los conocimientos, recursos y estrategias para encontrar una solución válida forman parte de un razonamiento crítico y una buena toma de decisiones.
- **Desarrollo de *software* a través de las herramientas y recursos de la Sociedad del Conocimiento:** Para ello también será necesario una gestión de la calidad y la innovación. Muchas de las competencias necesarias para llevar esto a cabo han sido adquiridas durante el transcurso del Grado de Ingeniería Informática.

Por otro lado, el proyecto desarrollado presenta también objetivos de desarrollo, los cuales dan respuesta al problema a resolver expuesto anteriormente. Los principales son:

- **Desarrollo de una aplicación móvil para el sistema operativo Android:**

Para que se lleve a cabo, es necesario la creación de una base de datos para archivar toda la información necesaria para su correcto funcionamiento, la cual se ha realizado en Firebase. La aplicación permitirá un uso social para los miembros de la Asociación “Grupo Scout 415 Los Olivos”. Las funcionalidades más importantes de la aplicación, a alto nivel y sin entrar en detalles son:

- Cada miembro puede registrarse o iniciar/cerrar sesión en ella pudiendo adicionalmente configurar su perfil, el cual también será visible para los demás miembros.
- El *software* dispone de publicaciones de post al estilo red social con “me gusta” y comentarios.
- Mantener conversaciones entre usuarios de la organización, además de la creación de grupos para compartir información de manera más sencilla.
- Señalar futuros eventos de la asociación en un calendario que estará a disposición de todos los integrantes.

1.3 Estructura de la memoria

La estructura de la memoria que se va a seguir es la siguiente:

- **Resumen/*Abstract* y Palabras clave/*Keywords*:** Resumen con una idea general del TFG para poder entender la problemática y la solución que se presentan, y una lista de 5 palabras clave utilizadas para la catalogación bibliográfica, ambos en castellano e inglés.

- **Introducción:** Se trata de un capítulo dedicado a información inicial necesaria para conocer el significado del proyecto desarrollado antes de profundizarse sobre él.
 - **1.1 Motivación:** Se describen las razones que han dado lugar a que se elaborase este proyecto.

 - **1.2 Objetivos:** Metas u objetivos que se pretenden alcanzar durante el desarrollo del proyecto.

 - **1.3 Estructura de la memoria:** Organización de la memoria de una manera más sencilla y general.

- **Tecnologías utilizadas:** Medios que se han utilizado para el desarrollo de la solución propuesta.
 - **2.1 Sistema operativo:** Descripción breve sobre el sistema operativo sobre el que se va a desarrollar el proyecto y las razones de su elección.

 - **2.2 Lenguaje de programación:** Presentación del lenguaje de programación escogido para realizar el trabajo y su importancia en la actualidad.

- **2.3 Entorno de desarrollo:** Exposición del entorno de desarrollo en el que se va a construir el proyecto y cómo se estructura.
- **2.4 Sistema de base de datos:** Definición del sistema de base de datos utilizado en el trabajo y sus ventajas.
- **Metodología:** Tal como su nombre indica, la metodología que se ha llevado a cabo para que el proyecto se haya realizado en tiempo y de una manera organizada.
- **Fases del desarrollo:** Son las etapas en las que se ha dividido la solución entregada, desde que nace la idea hasta que se entrega.
 - **3.1 Planificación:** Explicación sobre la planificación del proyecto y todo lo que implica: determinación del ámbito del proyecto, planificación temporal, asignación de recursos...
 - **3.2 Análisis de requisitos:** Definición sobre la fase de análisis de requisitos y lo que involucra: extracción de requisitos y funcionalidades del proyecto...
 - **3.3 Diseño y arquitectura:** Exposición sobre el diseño y arquitectura de la solución: estudio de las posibles opciones de implementación y decisión de una estructura general para el proyecto...

- **3.4 Programación e implementación:** Descripción sobre la programación e implementación del proyecto: desarrollo de la solución planteada y la base de datos asociada, tanto en *backend* como en *frontend*...
 - **3.5 Pruebas y revisión:** Explicación sobre la fase de pruebas y revisión: los problemas detectados y sus correcciones para que se cumplan los requisitos iniciales...
 - **3.6 Instalación y despliegue:** Definición de la instalación y despliegue de la solución desarrollada: puesta del *software* en funcionamiento, despliegue en dispositivos móviles...
 - **3.7 Documentación:** Exposición del apartado dedicado a la documentación y entregables del trabajo.
- **Conclusiones y Líneas Futuras:** Se trata de las conclusiones que se han alcanzado tras la elaboración del proyecto, además de las posibles ampliaciones que se podrían realizar en trabajos futuros.
 - **Bibliografía:** Referencias de información que han sido utilizadas en la realización de este TFG.
 - **Apéndices:** Información complementaria que no se encuentre en el cuerpo del TFG, como por ejemplo el manual de usuario.

Tecnologías utilizadas

2.1 Sistema operativo

El sistema operativo que se ha decidido utilizar para el proyecto es Android, ya que actualmente es el sistema operativo más utilizado del mundo, con una cuota de mercado superior al 90% en el año 2018, muy por encima de iOS.

Android es un sistema operativo basado en el núcleo Linux y otros *softwares* de código abierto. Fue adquirido por Google en 2005 y presentado en 2007 bajo licencia de Apache con el propósito de avanzar en los estándares abiertos de los dispositivos móviles. Actualmente está diseñado para su uso en dispositivos móviles con pantalla táctil, tal como lo son los teléfonos inteligentes o tabletas, aunque su desarrollo también puede encontrarse en relojes (Android Wear), televisiones (Android TV) y automóviles (Android Auto).

Además, Android posee una gran comunidad de desarrolladores, que actualizan diariamente la tienda online de aplicaciones Google Play, llegando a contar actualmente con más de 400.000 aplicaciones. La arquitectura general de una aplicación de Android está formada por las siguientes capas:

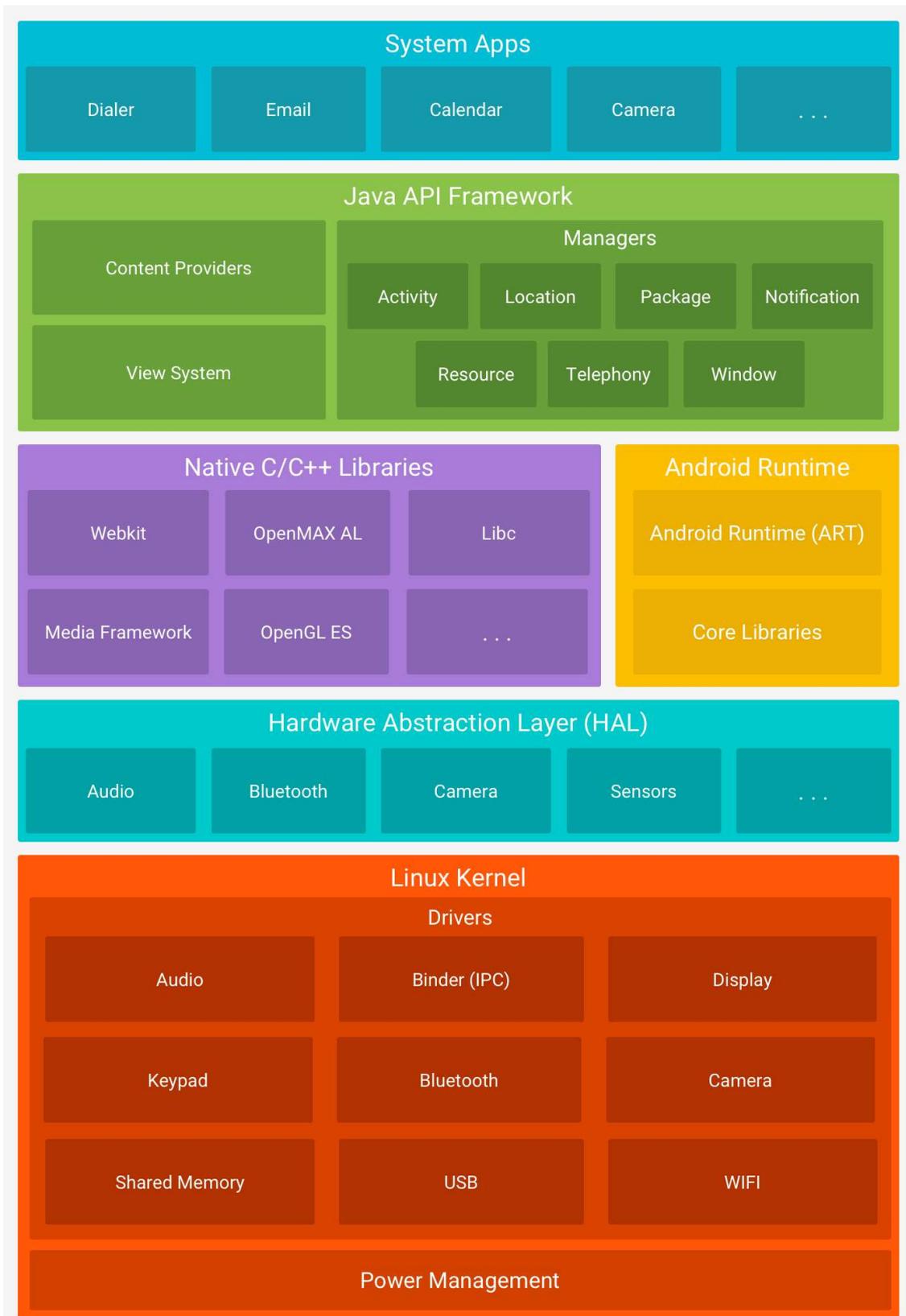


Figura 1.1: *Arquitectura de la plataforma Android.* [Imagen]. Recuperado el 19 de abril de 2022, de <https://developer.android.com/guide/platform?hl=es-419>.

- **Kernel de Linux:** La base de la plataforma Android es el *kernel* de Linux. Por ejemplo, funcionalidades como la generación de subprocesos y la administración de memoria de bajo nivel por parte del tiempo de ejecución de Android (ART) se basa en el *kernel* de Linux. Su uso permite que Android obtenga ventajas respecto a la seguridad y a los fabricantes desarrollar controladores de *hardware* para un *kernel* existente.
- **Capa de abstracción de *hardware* (HAL):** La capa de abstracción de *hardware* (HAL) proporciona interfaces respecto a las capacidades de *hardware* del dispositivo para la capa superior del marco de trabajo de la API de Java. HAL consiste en módulos de librerías con interfaces para los distintos componentes hardware, tales como pueden ser el módulo de Bluetooth o la cámara. El sistema carga el módulo de librería para cada componente cuando se realizan llamadas desde el marco de trabajo.
- **Tiempo de ejecución de Android/Android *Runtime* (ART):** En los dispositivos con Android 5.0 (API 21) o superior, cada aplicación ejecuta en su propio proceso y con su propia instancia del Android *Runtime* (ART). El ART está diseñado para correr múltiples máquinas virtuales en dispositivos con baja memoria ejecutando archivos DEX, los cuales están optimizados para ocupar un espacio de memoria mínimo en Android.

Algunas de las principales funcionalidades del ART son:

- Compilación antes del tiempo de ejecución (AOT) y compilación en tiempo de ejecución (JIT).
- Recolector de archivos no utilizados optimizado (GC).

- Mejora en la compatibilidad con la depuración, excepciones de diagnóstico detalladas y la capacidad de mantener puntos de control para supervisar campos específicos.

Dalvik era el Android Runtime en las versiones de Android anteriores a la 5.0. Si una aplicación se ejecuta de manera correcta en el ART, también debería hacerlo en Dalvik, aunque puede que no suceda a la inversa.

- **Bibliotecas C/C++ nativas:** Muchos de los servicios de Android como el ART o HAL está construido en código nativo el cual requiere de librerías nativas escritas en C y C++. Android proporciona el *framework* de APIs en Java para que estas librerías puedan ser utilizadas en las aplicaciones. Si se desea desarrollar una aplicación con C o C++, se puede utilizar el NDK de Android para acceder a estas librerías nativas directamente desde el código.
- **Marco de trabajo de la API de Java/Java API *framework*:** Todas las funciones de Android están disponibles en APIs escritas en Java, las cuales son los cimientos para crear aplicaciones en Android ya que simplifican la reutilización de componentes del sistema y servicios del sistema como:
 - Un sistema de vistas variado y extensible utilizado para construir la interfaz gráfica de las aplicaciones. Algunos ejemplos de estas vistas pueden ser botones, listas, cuadros de texto, etc.
 - Un administrador de recursos, el cual proporciona acceso a recursos que no pertenezcan al código como cadenas de texto o archivos de diseño.

- Un administrador de notificaciones que permite a cada aplicación mostrar alertas personalizadas en la barra de estado.
- Un administrador de actividad encargado de manejar el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- Proveedores de contenido que permiten a las aplicaciones acceder a datos de otras aplicaciones o compartir sus datos con otras.
- **Aplicaciones del sistema:** Android dispone de un conjunto de aplicaciones internas para correo electrónico, SMS, contactos, calendarios, navegación por internet, etc. Estas aplicaciones no tienen más preferencia que las instaladas por el usuario por lo que una aplicación externa puede convertirse en una que se utilice por defecto.

2.2 Lenguaje de programación

El desarrollo de aplicaciones para Android puede realizarse en distintos lenguajes de programación, entre ellos los más importantes son Kotlin y Java. El lenguaje finalmente seleccionado para realizar el proyecto ha sido Kotlin, y esto se debe a:

- Kotlin es un lenguaje de programación “pragmático” de tipo estático, gratuito, *open source* y de propósito general creado por JetBrains, la compañía fundadora de IntelliJ IDEA. Es ejecutado sobre la máquina virtual de Java y también puede ser compilado a código fuente de JavaScript, y combina características de

programación funcional y orientada a objetos. Kotlin principalmente se centra en la seguridad, interoperabilidad, claridad y soporte de herramientas.

- Respecto a Java, en Kotlin existen un gran número de ventajas, entre las cuales se encuentran:
 - **Conciso:** Kotlin es un lenguaje más conciso que Java. Se calcula que con Kotlin se reduce un 40% de líneas de código en comparación con Java, lo cual puede equivaler a menos *bugs*.
 - **Interoperabilidad:** Kotlin posee un nivel absoluto de interoperabilidad con Java. No existirá ningún problema en usar Kotlin en un proyecto Java y viceversa, todo compilando a la perfección.
 - **Soporte:** Kotlin proporciona una serie de funciones avanzadas propias. Además, Android Studio está basado en IntelliJ IDEA por lo que existe un excelente soporte para Kotlin, incluyendo *plugins* y configuraciones especiales.
 - **Sencillo aprendizaje:** Kotlin apunta a ser una mejora para Java, más que una reescritura completa, por lo que las habilidades adquiridas y la experiencia en Java pueden ser aplicables en Kotlin.
 - **Menos errores:** Kotlin es un lenguaje de tipo estático, el cual permite detectar errores en tiempo de compilación. Además, los NPE (*Null Pointer Exception*) son eliminados, incluso permitiendo el uso de funciones anónimas y de orden superior.

2.3 Entorno de desarrollo

Actualmente existen varios entornos de desarrollo importantes para crear aplicaciones para dispositivos móviles en Android, tales como Xamarin, React, Eclipse o Android Studio. Para realizar este proyecto, se ha escogido Android Studio, siendo uno de los entornos más conocidos y usados, y esto es debido a:

- Android Studio es el entorno de desarrollo integrado (IDE) oficial de Google y la más directa para el desarrollo de aplicaciones para Android y está basado en IntelliJ IDEA. Todas las herramientas de desarrollo avanzan a la vez que las evoluciones de Android, teniendo las mejores cualidades en cuanto a rendimiento, compatibilidad y acceso a funciones del sistema operativo.
- Se trata de un sistema de compilación flexible basado en *Gradle*, donde posee un emulador Android rápido y cargado de funciones pudiendo así probar y ejecutar aplicaciones desarrolladas.
- Android Studio es un entorno unificado creado para el desarrollo en todos los dispositivos Android, incluyendo soporte para Android Wear, GitHub y Google Cloud Platform a través de Firebase Cloud Messaging y Google App Engine.
- Posee compatibilidad con aplicaciones creadas con Java y Kotlin, incluso llegando a permitir el desarrollo en C o C++.

- En Android Studio existe una gran variedad de marcos de trabajo, plantillas y herramientas de prueba, además de una consola de desarrollador con la que poder conseguir consejos de optimización, estadísticas o traducciones.

Un proyecto de Android Studio contiene todo lo que define el lugar de trabajo de una aplicación: desde el código fuente y recursos hasta código de prueba y compilación. Un proyecto se encuentra formado por:

- **Módulos:** Un módulo se trata de un conjunto de archivos de origen y opciones de compilación que permiten dividir un proyecto en distintas unidades de funcionalidad, Un proyecto puede estar formado por varios módulos, los cuales pueden ser compilados, probados y depurados por separado. Hay distintos tipos de módulos:
 - **Módulo de aplicación para Android:** Proporciona un contenedor para el código fuente de la aplicación, archivos de recursos y las configuraciones de niveles de aplicación. Android Studio ofrece módulos de teléfono y Tablet, Wear OS, Android TV y Glass, cada uno con plantillas y archivos esenciales.
 - **Módulo de función:** Es una función modularizada de la aplicación que puede aprovechar Play Feature Delivery, como Google Play Instant.
 - **Módulo de biblioteca:** Proporciona un contenedor para código reutilizable que puede usarse como dependencia en otros módulos o importarse a otros proyectos. Android ofrece una biblioteca de Android

con todos los tipos de archivos admitidos en un proyecto de Android, y una biblioteca Java con archivos de origen Java.

- **Módulo de Google Cloud:** Provee un contenedor para el código de *backend* de Google Cloud. Utiliza HTTP simple, Cloud Endpoints y Cloud Messaging para poder conectarse.

- **Archivos de proyecto:** De forma predeterminada, en Android Studio se muestran los archivos organizados en módulos y tipos de archivos para simplificar la navegación, ocultándose así directorios o archivos que no son usados con frecuencia.

Son mostrados los archivos de configuración relacionados con la compilación del proyecto en “Gradle Script”, los archivos de manifiesto para cada módulo y todos los archivos de recursos en un grupo, en lugar de carpetas separadas por calificador de recursos.

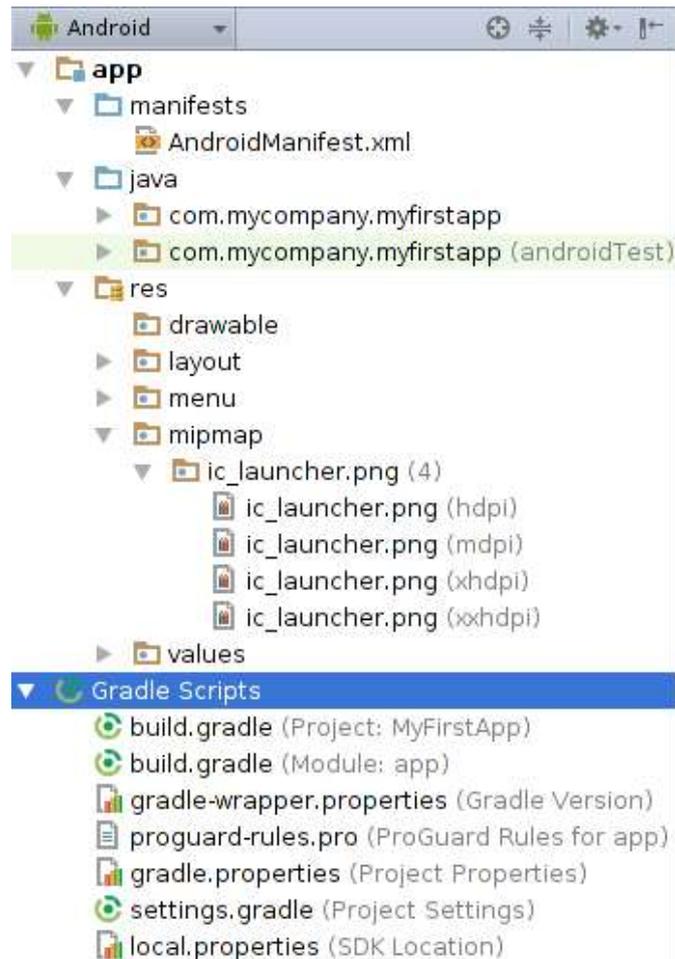


Figura 2.2: Descripción general de proyectos Android. (2022). [Imagen]. Recuperado el 20 de mayo de 2022, de <https://developer.android.com/studio/projects?hl=es-419>.

- **Configuración de la estructura de proyecto:** Las diferentes opciones de configuración de un proyecto en Android Studio son:
 - **Ubicación SDK:** Ubicación del JDK, SDK y NDK de Android del proyecto.

- **Proyecto:** En este apartado se encuentra la versión del *Gradle*, el complemento de Android para *Gradle* y el nombre de la ubicación del repositorio.
- **Módulos:** Permite editar opciones de configuración de compilación del módulo, dependencias de biblioteca o la firma de la aplicación.

2.4 Sistema de base de datos

Prácticamente cualquier aplicación de móvil actual necesita una base de datos asociada en la que almacenar la información necesaria para su funcionamiento. En Android, existen multitud de posibilidades, como por ejemplo LevelDb, MySQL, SQLite o Firebase. Finalmente, para el desarrollo del proyecto se ha utilizado Firebase, y esto se debe porque:

- Firebase es una plataforma para el desarrollo de aplicaciones web y móviles lanzada en 2011 y adquirida en 2014 por Google. Se encuentra ubicada en la nube, y se encuentra integrada con Google Cloud Platform. Contiene diversas funciones para que cualquier desarrollador sea capaz de combinar y adaptar la plataforma en función de sus necesidades.
- Firebase permite la creación de aplicaciones mejores, minimizando el tiempo de optimización y desarrollo. Cuenta con detección de errores y de testeo, y poder tener almacenamiento en la nube permiten dar un salto en la calidad de una aplicación

- Usa la infraestructura de Google y puede escalarse para cualquier tipo de aplicación. Los proyectos pueden ser creados sin la necesidad de un servidor y los datos se sincronizan fácilmente sin tener que administrar conexiones.
- Contiene Cloud Storage, por lo que permite contar con una base de datos con almacenamiento de imágenes y ficheros, además de Cloud Functions, permitiendo ahorrar infraestructura de *backend*.
- Firebase permite gestionar de manera sencilla los usuarios registrados en la aplicación, además de facilitarlo a través de invitaciones y notificaciones. Cuenta con Secure Sockets Layer (SSL) y con un panel muy intuitivo, simple y de fácil acceso.

Todos los datos que se encuentran almacenados en Firebase Realtime Database se encuentran en JSON. La base de datos puede describirse como un árbol JSON alojado en la nube. Al tratarse de una base de datos no relacional, a diferencia de una base de datos SQL, no hay tablas ni registros.

```
{
  "users": {
    "alovelace": {
      "name": "Ada Lovelace",
      "contacts": { "ghopper": true },
    },
    "ghopper": { ... },
    "eclarke": { ... }
  }
}
```

Figura 2.3: Estructura de la base de datos Firebase. [Imagen].

Recuperado el 15 de mayo de 2022, de <https://firebase.google.com/docs/database/web/structure-data?hl=es#:~:text=C%C3%B3mo%20se%20estructuran%20los%20datos%3A%20un%20%C3%A1rbol%20JSON,no%20hay%20tablas%20ni%20registros>.

Metodología

Para el desarrollo de este proyecto se ha adaptado la metodología SCRUM, aunque al tratarse de un trabajo individual y la metodología esté orientada hacia proyectos en equipo, ha tenido que ser adaptada a la situación.

SCRUM es una metodología ágil introducida en los años 90. Adopta un enfoque que parte de la premisa de que los problemas no pueden ser totalmente definidos de antemano y que es necesario centrarse en poder responder al cambio. El *framework* se basó en la idea de la mejora continua a través de un enfoque orientado al proceso. Los problemas serán resueltos de una manera más eficiente si son divididos en piezas más pequeñas.

Todas las tareas para realizar en el proyecto se organizan en el *Product Backlog*, para que estén visibles para todo el equipo de desarrollo y así se pueda tener una vista panorámica de todo lo que hay que realizar. Las prioridades de las actividades se asignarán en el *Product Backlog* en función de las necesidades del cliente y la complejidad que le corresponden. SCRUM se organiza en *sprints*, los cuales son iteraciones cortas y de duración fija, normalmente de dos semanas, aunque puede variar. En cada iteración es necesario proporcionar un resultado completo, pudiendo ser entregado con el mínimo esfuerzo al cliente cuando sea solicitado.

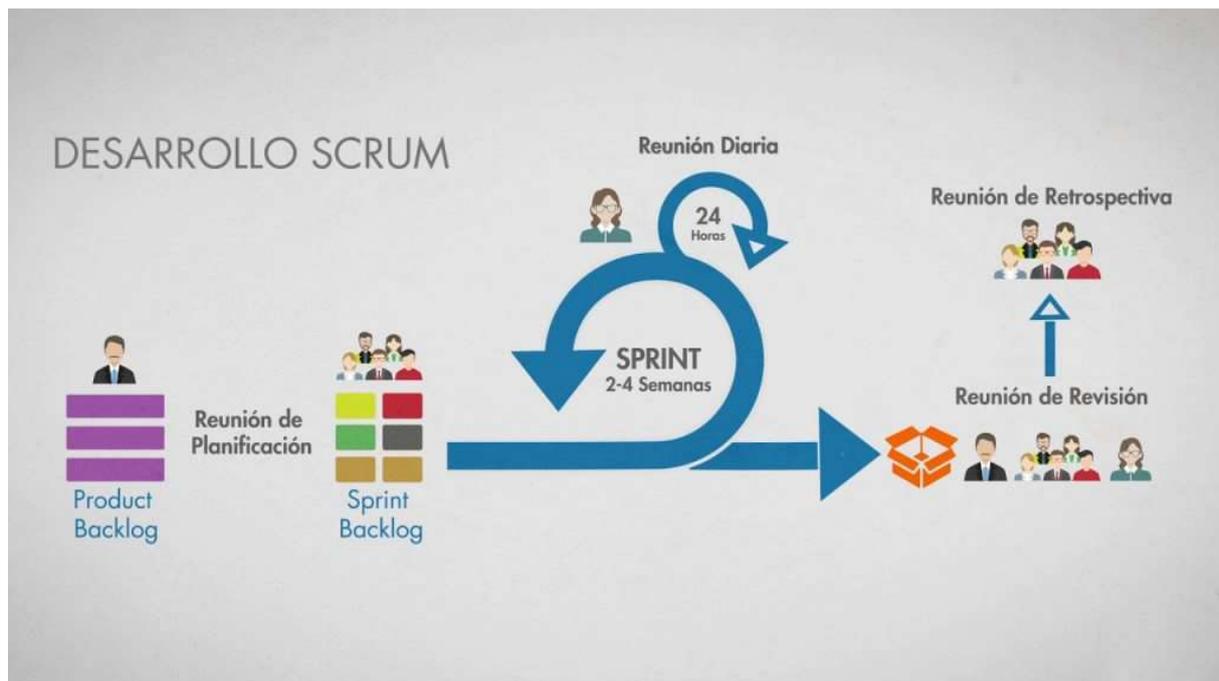


Figura 2.5: Metodología Scrum, una herramienta útil para agilizar tus proyectos. [Imagen]. Recuperado el 15 de mayo de 2022, de <https://hondurasdigitalchallenge.com/2020/05/21/metodologia-scrum-una-herramienta-util-para-agilizar-tus-proyectos/>.

SCRUM cuenta principalmente con tres roles:

- **Propietario del producto/Product Owner:** Encargado de optimizar y maximizar el valor del producto, además de ser un gestor de requisitos o un cliente que se encarga de gestionar el *Product Backlog*, estructurándolo, detallándolo y priorizando las tareas. Se ocupa de marcar las metas de los diferentes *sprints* de manera clara y acordada con el equipo de desarrollo. El Propietario del producto debe convertirse en el altavoz del cliente, transmisor de las demandas y del *feedback* otorgado por los mismos.

En este proyecto, el rol de Propietario del producto lo lleva a cabo el tutor del proyecto que delega habitualmente en el realizador de este.

- **Scrum Master:** Es el responsable de que las técnicas SCRUM sean comprendidas y bien aplicadas en la organización, intentando que se facilite la ejecución del proceso. Además, se le asigna la función de eliminar impedimentos que puedan afectar a la entrega del producto.

En este proyecto, el rol de Scrum Master lo lleva a cabo el tutor del proyecto.

- **Equipo de desarrollo:** Es el encargado de realizar las tareas priorizadas por el Propietario del producto, además de estimar las tareas del *Product Backlog*, sin dependencias externas.

En este proyecto, el rol de Equipo de desarrollo es llevado a cabo por el realizador del proyecto.

El proceso de SCRUM se divide en 5 fases:

- **Inicio:** Estudio y análisis del proyecto con el fin de identificar las necesidades básicas del *sprint*. También es necesario estimar que se va a entregar como resultado para el siguiente *sprint* y como se va a realizar el trabajo.
- **Planificación y estimación:** Durante esta fase se identifican y estiman las tareas, además de crear el *Sprint Backlog* (iteración de tareas), lo cual es necesario para establecer metas fijas y cumplir con los plazos.

- **Implementación:** Producción del entregable a partir de las tareas de la fase anterior. Los objetivos de esta fase son: crear entregables a partir del *Sprint Backlog*, realizar un *Daily Standup* (revisión diaria sobre el avance del equipo de desarrollo) y un refinamiento del *Product Backlog*.
- **Revisión y retrospectiva:** Se trata de una revisión del entregable y medir y mejorar la eficiencia del equipo. El equipo de desarrollo muestra al propietario del producto el incremento de valor generado y se realiza una revisión del *sprint* utilizando unos criterios de aceptación previamente creados, por lo que también se buscarán soluciones para mejorar los próximos *sprints*.
- **Lanzamiento:** Este proceso de SCRUM tiene como objetivo entregar el resultado final al cliente cuando se cumplen los requisitos que se piden. Además, otra acción que se da es la retrospectiva del proyecto con la que se realiza una inspección a todo lo ocurrido durante el proyecto.

La metodología escogida para el proyecto es SCRUM, y esto es debido a que:

- La dimensión de los proyectos puede dimensionarse de una manera más efectiva. Tratar de abarcar un proyecto entero dividiéndolo en pequeños bloques a través de *sprints* crea una mejor gestión e identificación de objetivos y posibles incidencias o contratiempos.
- A través de SCRUM se consigue un fomento en la motivación y el compromiso del equipo ya que se encuentran en un continuo desarrollo en el que se mejoran las capacidades personales y profesionales, y sobre todo una mayor

productividad. Una de las grandes diferencias también se dará en la calidad del *software* al utilizarse este método de trabajo y revisiones continuas.

- Las metodologías ágiles implican a todas las partes que forman un proyecto, tanto el equipo de desarrollo como el cliente o los proveedores. Esta situación genera una mayor auto-organización, autonomía y responsabilidad en el equipo. Con el desarrollo del proyecto, los márgenes de error en las entregas van minimizándose y ajustándose, con lo cual el equipo aprende rápidamente, *sprint* a *sprint*, aprendiendo y corrigiendo en lo que sea conveniente.

Fases del desarrollo

Tal como se mencionó en el anteproyecto de este trabajo, las fases en las que se ha dividido el proyecto, coincidiendo con las de un desarrollo de *software*, son: Planificación, Análisis de requisitos, Diseño y arquitectura, Programación e implementación, Pruebas y revisión, Instalación y despliegue, y Documentación.

3.1 Planificación

Antes de comenzar con el desarrollo del proyecto, es necesario hacer una serie de tareas para que sea realice de forma exitosa. Algunas de estas tareas son:

- **Determinación del ámbito del proyecto y estudio de viabilidad:** Para la realización del trabajo, lo primordial es conocer la idea principal que se va a desarrollar antes de tomar decisiones, es decir, entender el problema a resolver y ver la viabilidad de proporcionar una solución. Para este proyecto, la problemática planteada es:

La Asociación “Grupo Scout 415 Los Olivos” es una organización civil de carácter educativo para jóvenes y niños/as, socio de Scouts de Andalucía y que se adhiere a la Constitución Mundial Scout. Está domiciliado al Colegio “Los Olivos” y el fin del Grupo es la formación integral a través del método Escultismo, potenciando la responsabilidad, la libertad, la concienciación en temas sociales y el servicio a los demás sin ningún tipo de discriminación.

Actualmente cuenta con 180 socios, entre los cuales se encuentran 30 monitores y unos 150 educandos entre la edad de 6 y 21 años.

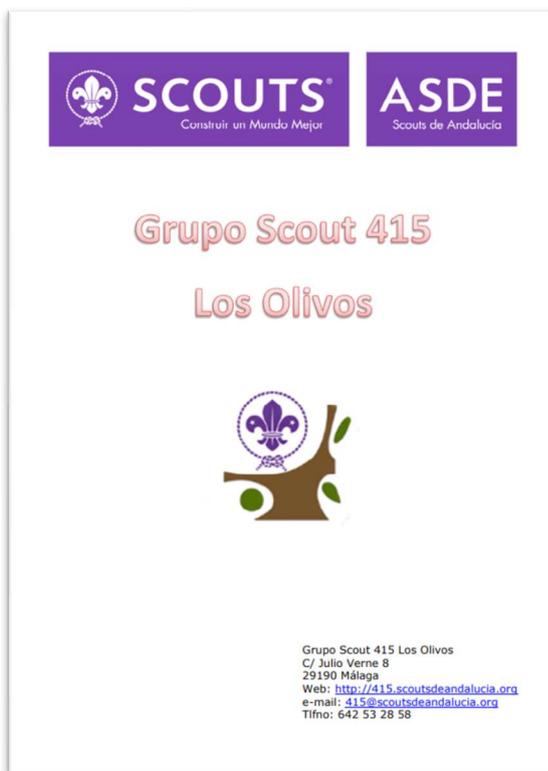


Figura 3.1: Dossier de "Grupo Scout 415 Los Olivos"

La solución que se presenta inicialmente es:

“Scout Connections” es una aplicación para dispositivos móviles que permite la comunicación, compartir e intercambio de información entre miembros de la comunidad de la Asociación “Grupo Scout 425 Los Olivos”, como por ejemplo a través de *posts*, mensajería privada individual y grupal, calendario, etc.

- **Análisis de riesgos asociados y estimación del coste:** Una vez se conoce la idea que se propone para el proyecto, es necesario hacer un estudio sobre los

riesgos asociados que tiene realizar el proyecto. Uno de los mayores problemas iniciales fue el hecho de realizar una aplicación de móvil para Android ya que durante el Grado de Ingeniería del Software no se hace mucho hincapié en ello, por lo que el uso de nuevas tecnologías como Android Studio dificulta la velocidad de desarrollo. También es importante el lenguaje de programación en el que está realizada la solución, ya que Kotlin es un lenguaje reciente y no se había realizado ningún otro proyecto con anterioridad. Otra cuestión es el tiempo asignado para la realización del proyecto, ya que el proyecto ha ido floreciendo al mismo tiempo que se han realizado prácticas externas y asignaturas del Grado. Un aporte importante en un proyecto es la estimación del coste, en este caso no siendo monetario, pero siendo necesario para la obtención del título de Ingeniería.

- **Planificación temporal y asignación de recursos:** La planificación temporal normalmente se realiza a través de técnicas como red de tareas, diagrama de Gantt, PERT, CPM, etc. Para este trabajo, se ha hecho uso de un diagrama de Gantt para la distribución del tiempo y las tareas:

Fases del desarrollo	14 mar – 27 mar	28 mar – 10 abr	11 abr – 24 abr	25 abr – 8 may	9 may – 22 may	23 may – 5 jun	6 jun – 19 jun
Planificación	10 h						
Análisis de requisitos	16 h						
Diseño y	20 h						

arquitectura		
Programación e implementación		(3 (Inicio) + 7 (Planificación y estimación) + 20 (Implementación) + 9 (Revisión y retrospectiva) + 3 (Lanzamiento)) x 5 <i>sprints</i> = 210 h
Pruebas y revisión		20 h
Instalación y despliegue		10 h
Documentación		10 h

El plazo empleado para el proyecto ha sido desde marzo hasta junio de 2022, y a su vez dividido en plazos/*sprints* de dos semanas. Los recursos han sido asignados en función del tiempo restante y la importancia que ha sido necesario dedicar.

3.2 Análisis de requisitos

La fase de análisis de requisitos es importante ya que es imprescindible averiguar qué es lo que tiene que hacer el *software* que se va a desarrollar. No es necesario entrar en grandes aspectos técnicos y debe existir una buena comunicación entre el equipo de desarrollo y el cliente. Es una etapa en la que es necesario invertir el tiempo que sea necesario y la finalidad es generar un documento de requisitos funcionales (funcionalidad del *software*, servicios que prestará el sistema) y no funcionales

(propiedades del sistema, no referidas directamente a funciones específicas del *software*).

Para este proyecto, la lista de requisitos es:

ID	Nombre	Descripción
Requisitos funcionales de usuario		
RF 1.1	Registro	Se debe poder registrarse con un correo electrónico y una contraseña
RF 1.2	Inicio de sesión	Se debe poder iniciar sesión con un correo electrónico y una contraseña
RF 1.3	Inicio de sesión con Google	Se debe poder iniciar sesión con una cuenta de Google existente
RF 1.4	Recuperar contraseña	Se debe poder recuperar la contraseña con un correo de una cuenta ya creada en la aplicación
RF 1.5	Cerrar sesión	Se debe poder cerrar sesión
Requisitos funcionales de publicaciones		
RF 2.1	Crear publicaciones	Se debe poder crear publicación en caso de poseer rol de monitor. Para ello será necesario un título y descripción, y una imagen opcionalmente
RF 2.2	Ver publicaciones	Se debe poder ver las publicaciones creadas
RF 2.3	Dar “me gusta” a una publicación	Se debe poder dar “me gusta” a una publicación creada
RF 2.4	Comentar una publicación	Se debe poder comentar una publicación creada
RF 2.5	Eliminar un	El creador de un comentario podrá eliminarlo

	comentario	
RF 2.6	Compartir una publicación	Se debe poder compartir una publicación creada
RF 2.7	Editar publicación	El creador de una publicación podrá editarla, pudiendo cambiar título, descripción e imagen
RF 2.8	Eliminar publicación	El creador de una publicación podrá eliminarla
RF 2.9	Ver creador de una publicación	Se debe poder ver el perfil del creador de una publicación
RF 2.10	Ver personas a las que les gusta una publicación	Se debe poder ver las personas a las que les gusta una publicación
RF 2.11	Ver comentarios de una publicación	Se debe poder ver los comentarios de una publicación
Requisitos funcionales de perfil		
RF 3.1	Ver perfil	Un usuario que haya iniciado sesión debe poder ver su perfil
RF 3.2	Ver perfil externo	Se debe poder ver el perfil de un usuario diferente al que se ha iniciado sesión
RF 3.3	Información de perfil	En el perfil del usuario se debe poder ver el nombre, correo electrónico, teléfono y rol asociados
RF 3.4	Información de perfil	En el perfil del usuario se debe poder ver el nombre, imagen, fondo, correo electrónico, teléfono y rol asociados, además de las publicaciones creadas
RF 3.5	Editar perfil	Se debe poder editar el perfil del usuario con el

		que se ha iniciado sesión
RF 3.5.1	Editar nombre	Se debe poder editar el nombre del usuario con el que se ha iniciado sesión
RF 3.5.2	Editar teléfono	Se debe poder editar el teléfono del usuario con el que se ha iniciado sesión
RF 3.5.3	Editar imagen	Se debe poder editar la imagen del usuario con el que se ha iniciado sesión
RF 3.5.4	Editar fondo	Se debe poder editar el fondo del usuario con el que se ha iniciado sesión
RF 3.6	Cambiar contraseña	Se debe poder cambiar la contraseña del usuario con el que se ha iniciado sesión
RF 3.7	Solicitar rol de monitor	Un usuario con rol de scout podrá solicitar el rol de monitor
Requisitos funcionales de social		
RF 4.1	Ver conversaciones	Se debe poder ver una lista con las conversaciones individuales con usuarios de la aplicación
RF 4.2	Ver grupos	Se debe poder ver una lista con las conversaciones grupales a las que el usuario con el que se ha iniciado sesión pertenece
RF 4.3	Filtrar conversaciones y grupos	Se debe poder filtrar las conversaciones y grupos a través de una búsqueda
RF 4.4	Ver usuarios de la aplicación	Se debe poder ver una lista con los usuarios registrados en la aplicación, permitiendo filtrarlos y conversar con ellos
RF 4.5	Crear grupo	Un usuario debe poder crear un grupo. Para ello

		deberá añadir un título, y una descripción e imagen opcionalmente
RF 4.6	Enviar mensaje	Se debe poder enviar mensajes, tanto en conversaciones individuales como grupales
RF 4.7	Enviar imagen	Se debe poder enviar imágenes, tanto en conversaciones individuales como grupales
RF 4.8	Ver usuario de una conversación	Se debe poder ver el perfil del usuario con el que se está conversando
RF 4.9	Ver información de un grupo	Se debe poder la información de un grupo al que se pertenece, entre la que está el título, descripción, imagen y miembros del grupo
RF 4.10	Editar grupo	El creador debe poder editar el grupo. Puede cambiar el título, descripción e imagen
RF 4.11	Añadir participante	El creador de un grupo o un administrador deben poder añadir un nuevo participante
RF 4.12	Cambiar rol de grupo	El creador de un grupo o un administrador deben poder cambiar el rol de grupo de un miembro
RF 4.13	Eliminar participante	El creador de un grupo debe poder eliminar un miembro
RF 4.14	Salir de un grupo	Un miembro de un grupo podrá salir de este
RF 4.15	Eliminar grupo	El creador de un grupo debe poder eliminar el grupo
Requisitos funcionales de eventos		
RF 5.1	Ver siguientes eventos	Se debe poder ver los siguientes eventos ordenados por proximidad
RF 5.2	Filtrar eventos	Se debe poder filtrar los eventos por fecha

RF 5.3	Crear evento	Un usuario con rol de monitor debe poder crear eventos. Es necesario aportar un título, descripción y fecha, y opcionalmente una imagen
RF 5.4	Editar evento	El creador de un evento debe poder editar su título, descripción, imagen y fecha asociados
RF 5.5	Eliminar evento	El creador de un evento debe poder eliminarlo
Requisitos no funcionales		
RNF 1	Plataforma	La aplicación debe desarrollarse para Android
RNF 2	Interfaz de usuario	La interfaz de usuario de la aplicación debe ser intuitiva, clara, coherente y atractiva
RNF 3	Idiomas	La aplicación debe estar preparada para los idiomas español e inglés
RNF 4	Temas	La aplicación estar adaptada a los temas de colores de Android claro y oscuro
RNF 5	Documentación	Entregar un manual de usuario de la aplicación

Los requisitos funcionales están agrupados por funcionalidad general, la cual será asignada para cada *sprint*/plazo de la fase de “Programación e implementación”.

3.3 Diseño y arquitectura

En esta fase son estudiadas las posibles opciones para la implementación del *software* a construir, además de la estructura general del mismo. Suele ocurrir que la solución inicial no sea la más adecuada, lo que hará que vaya refinándose a lo largo del proyecto. Para poder tener una idea de cómo es el comportamiento de la solución, es

necesario tener un modelo visual de las interacciones que puede tener el usuario en la aplicación y el *frontend* del *software*. Para ello, se ha hecho uso de un lenguaje de modelo de flujo de interacción llamado IFML, el cual es un lenguaje de modelado estandarizado en la Ingeniería de Software.

Para la solución que se propone, el modelo que se ha generado (para ver con mayor claridad, acceder al archivo suministrado referente al modelo IFML), dividido según las distintas funcionalidades, es:

- **Funcionalidades de usuario:**

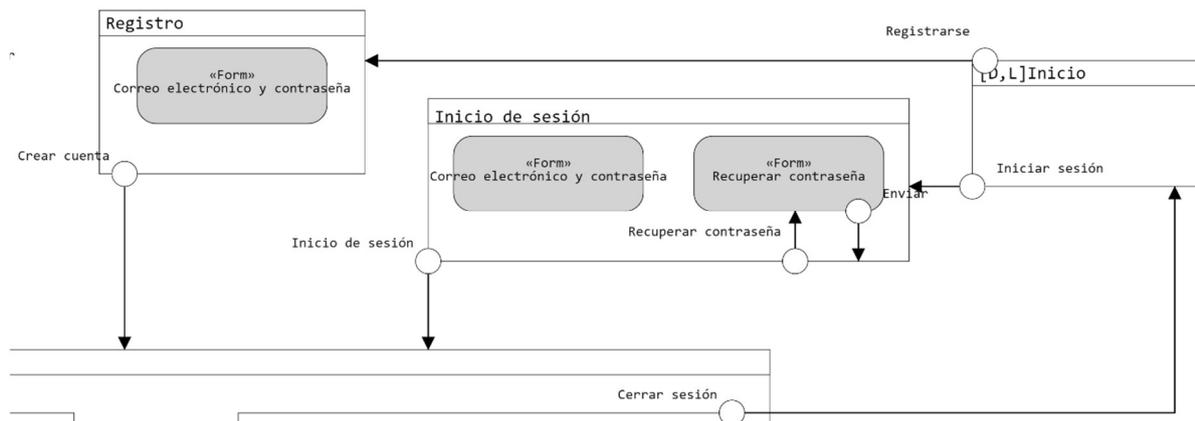


Figura 3.2: IFML de las funcionalidades de usuario

- **Funcionalidades de publicaciones:**

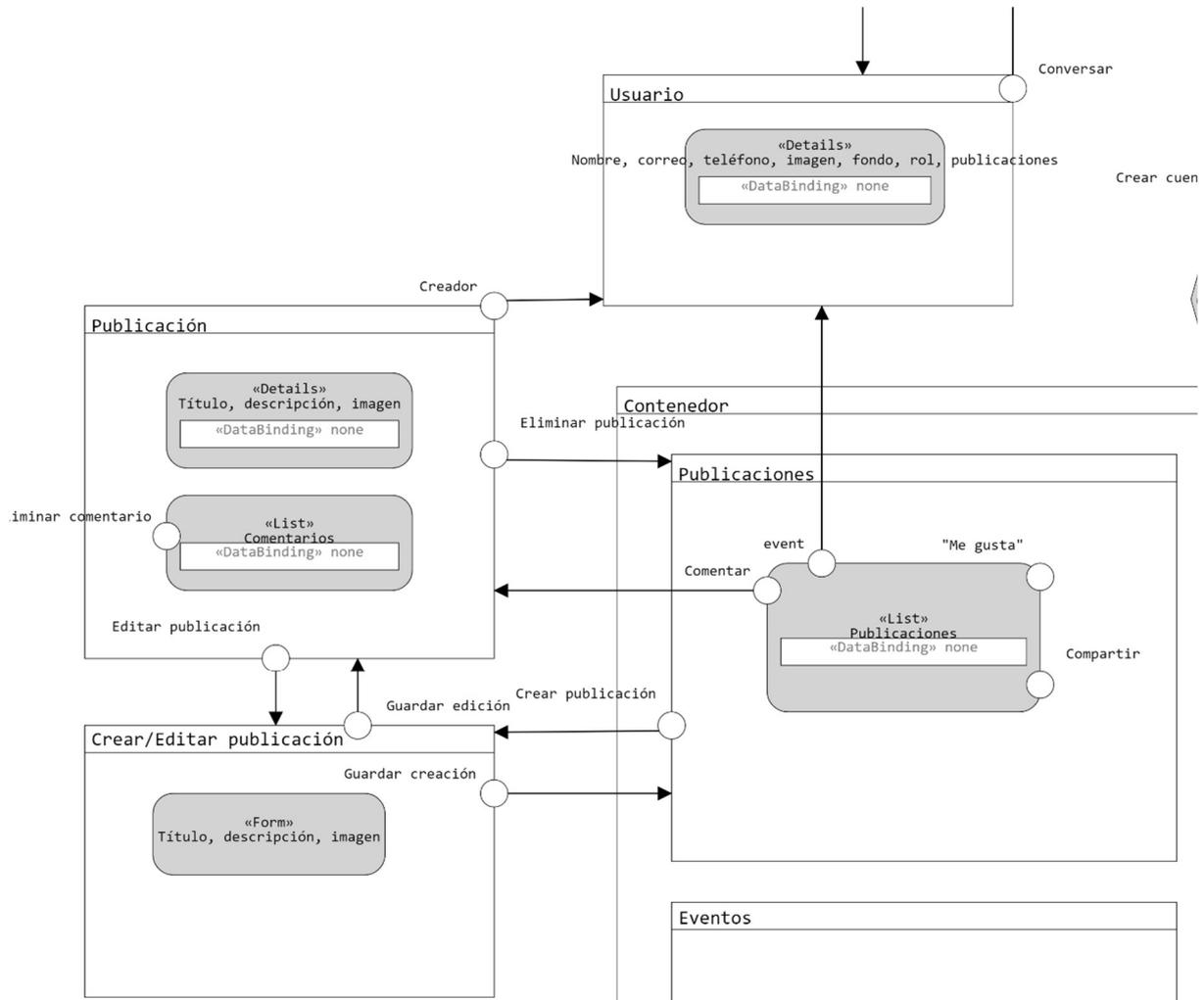


Figura 3.3: IFML de las funcionalidades de publicaciones

- **Funcionalidades de perfil:**

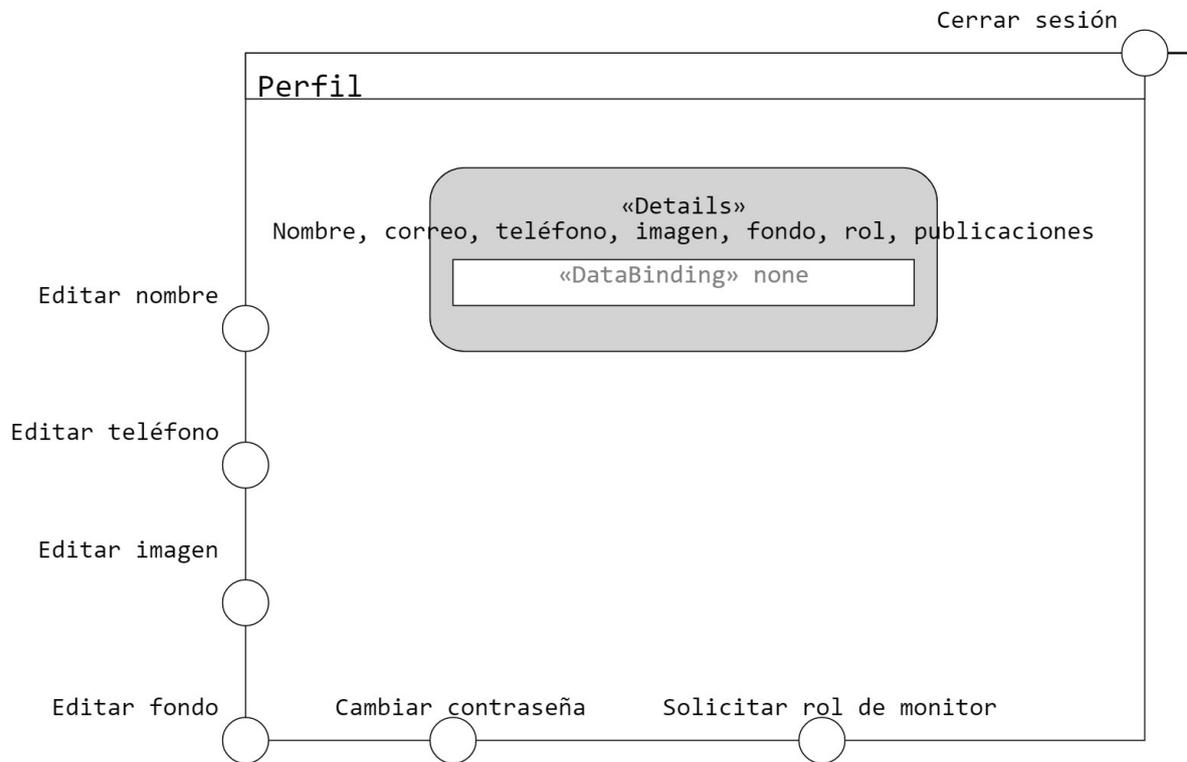


Figura 3.4: IFML de las funcionalidades de perfil

- Funcionalidades de social:

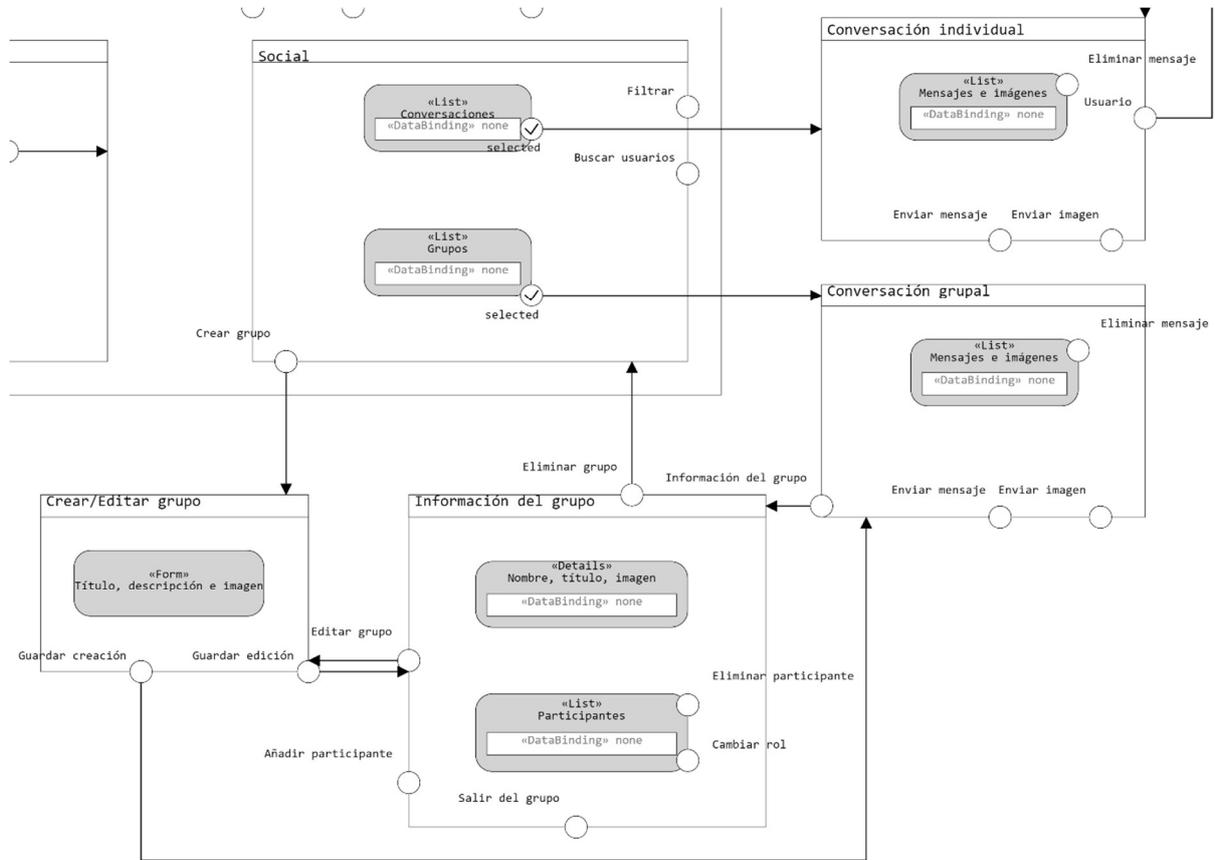


Figura 3.5: IFML de las funcionalidades de social

- Funcionalidades de eventos:

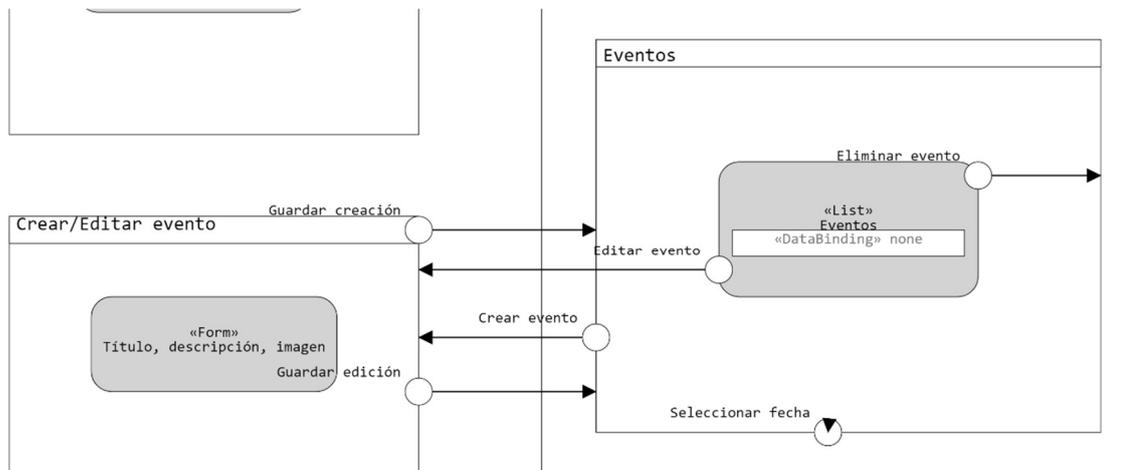


Figura 3.6: IFML de las funcionalidades de eventos

Durante esta fase del desarrollo, es interesante también crear diseños y maquetas del *software* que se va a realizar, comunmente llamado como mockups. Estos mockups ayudarán a que se entienda mejor la idea de la solución, la revisión del proyecto es más ávida, además que durante el desarrollo de un proyecto grande, el ahorro de dinero y trabajo es considerable.

Para el desarrollo de este proyecto se han creado una serie de mockups a través de la aplicación Balsamiq Wireframes, la cual la permite diseñar interfaces de usuario a través de bocetos, sin necesidad de escribir código y de una manera visible y fácil de entender.

Estos son algunos de los mockups que se han realizado para la implementación de la aplicación (el archivo original con los mockups se encuentra suministrado):

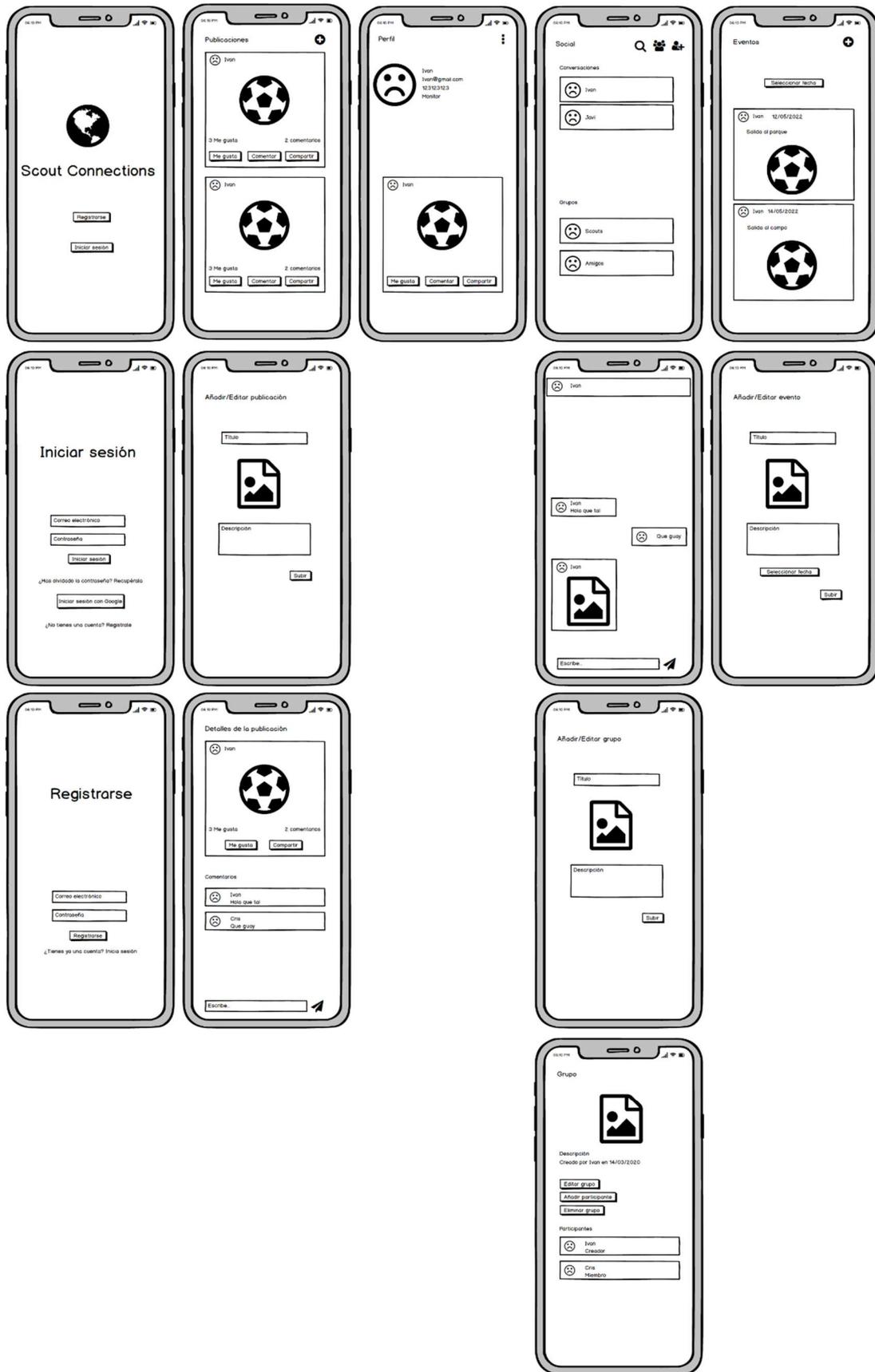
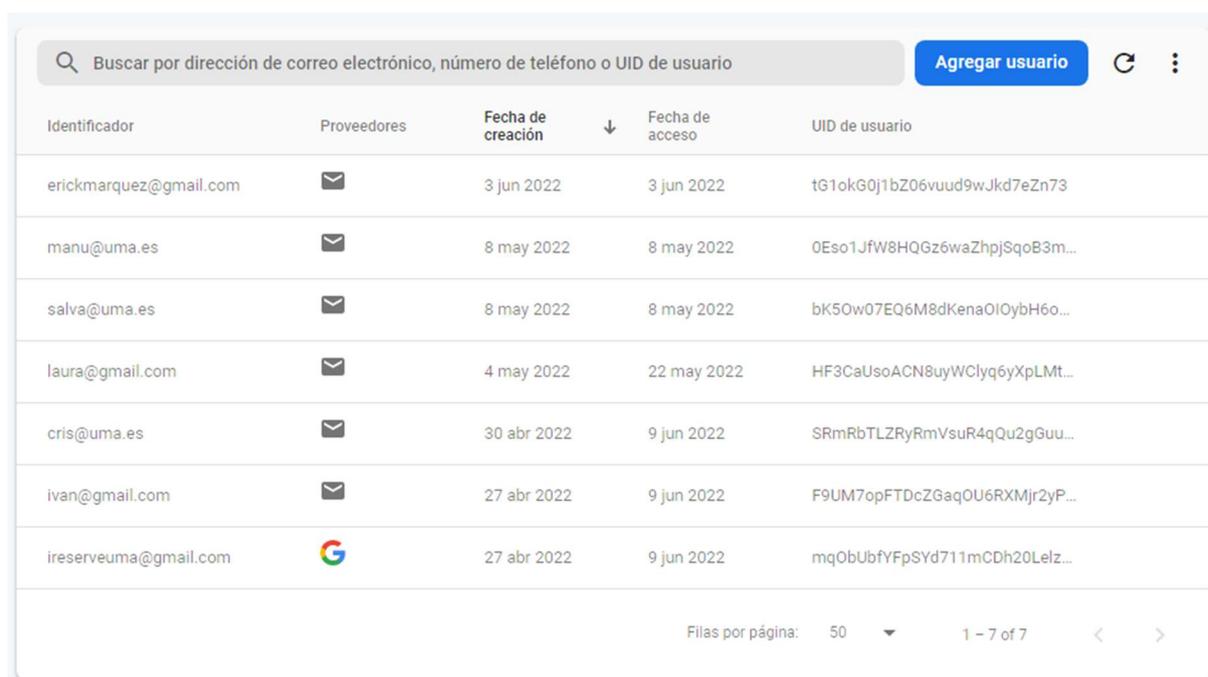


Figura 3.7: Mockups de la aplicación

3.4 Programación e implementación

Tal como se ha mostrado en el apartado anterior de metodología, SCRUM está formado por *sprints* o fases, y en concreto, la programación e implementación de este proyecto se ha dividido en 5 *sprints* de 2 semanas cada uno, ya que las funcionalidades principales de la aplicación fueron asociadas en 5 principales grupos: usuario, publicaciones, perfil, social y eventos. Aunque antes de comenzar con las funcionalidades es necesario primero crear y configurar el proyecto y la base de datos, e iniciar la conexión entre ambos.

La base de datos de Firebase utilizada para la aplicación desarrollada tiene la siguiente estructura general:



Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
erickmarquez@gmail.com		3 jun 2022	3 jun 2022	tG1okG0j1bZ06vuud9wJkd7eZn73
manu@uma.es		8 may 2022	8 may 2022	0Eso1JfW8HQGz6waZhpjSqoB3m...
salva@uma.es		8 may 2022	8 may 2022	bK50w07EQ6M8dKena0IOybH6o...
laura@gmail.com		4 may 2022	22 may 2022	HF3CaUsoACN8uyWClyq6yXpLMT...
cris@uma.es		30 abr 2022	9 jun 2022	SRmRbTLZRyRmVsuR4qQu2gGuu...
ivan@gmail.com		27 abr 2022	9 jun 2022	F9UM7opFTDcZGaqOU6RXMjr2yP...
ireserveuma@gmail.com		27 abr 2022	9 jun 2022	mQObUbfYFpSYd711mCDh20Lelz...

Figura 3.8: Autenticación de la base de datos de Firebase

```

ChatList
├── F9UM7opFTDcZGaqOU6RXMjr2yPj2
│   ├── HF3CaUsoACNBuyWClYq6yXpLMH2
│   │   └── uid: "HF3CaUsoACNBuyWChy6yXpLMH2"
│   ├── SRmRbTLZrYRmVsUr4qQu2yGuu8o1
│   ├── mq0ubfYFpSYd711mCDh20Le1zW2
│   ├── tG1okG8j1bZ86vuu9wJk07eZn73
│   ├── HF3CaUsoACNBuyWClYq6yXpLMH2
│   ├── SRmRbTLZrYRmVsUr4qQu2yGuu8o1
│   ├── mq0ubfYFpSYd711mCDh20Le1zW2
│   └── tG1okG8j1bZ86vuu9wJk07eZn73
├── Chats
│   ├── -N2cyaJ3-q-zopuJ9xjR
│   │   ├── message: "Hola que tal?"
│   │   ├── receiver: "mq0ubfYFpSYd711mCDh20Le1zW2"
│   │   ├── seen: true
│   │   ├── sender: "F9UM7opFTDcZGaqOU6RXMjr2yPj2"
│   │   ├── time: "1653176165636"
│   │   └── type: "text"
│   ├── -N2cyut64snj0DLAD-
│   ├── -N2d-K1ZUR6IaZ6jyY
│   ├── -N2fRbYOKY0yJ-j-yf0t
│   ├── -N3QNZLZLLe2cNXYq2J
│   ├── -N3cpv_CULhNysAX2XkH
│   ├── -N3ca3ER2-k80vcv0yM0
│   ├── -N3cuz28cG2JlgXue1ba
│   ├── -N3cu4kAmw0k1k1JL
│   └── -N3cv4TWjtmZ1n0D9e
├── Events
│   ├── 1653229947493
│   │   ├── creator: "F9UM7opFTDcZGaqOU6RXMjr2yPj2"
│   │   ├── description: "Se realizará en la ONG Manos Unidas de Testinos"
│   │   ├── eid: "1653229947493"
│   │   ├── image: "https://firebasestorage.googleapis.com/v0/b/scout-connections.appspot.com/o/Events%2FEvent_1653229947493?alt=media&token=11c75fc-164a-48c"
│   │   ├── tEvent: "1658505600000"
│   │   ├── time: "1653229947493"
│   │   └── title: "Recogida de alimentos"
│   └── 1653230131543
├── Groups
│   ├── 1653176348892
│   │   ├── Messages
│   │   │   ├── 1653216570096
│   │   │   │   ├── message: "https://firebasestorage.googleapis.com/v0/b/scout-connections.appspot.com/o/GroupChats%2F1653216570096?alt=media&token=6fdd1b"
│   │   │   │   ├── sender: "SRmRbTLZrYRmVsUr4qQu2yGuu8o1"
│   │   │   │   ├── time: "1653216570096"
│   │   │   │   └── type: "image"
│   │   ├── Participants
│   │   │   ├── creator: "SRmRbTLZrYRmVsUr4qQu2yGuu8o1"
│   │   │   ├── description: "Este es el grupo que se utilizará para los scouts de tarde"
│   │   │   ├── gid: "1653176348892"
│   │   │   ├── image: "https://firebasestorage.googleapis.com/v0/b/scout-connections.appspot.com/o/Groups%2Fimage_1653176348892?alt=media&token=f8da042d-3486-4"
│   │   │   ├── time: "1653176348892"
│   │   │   └── title: "Scouts Grupo Tarde"
│   │   ├── 1653176683791
│   │   └── 1653213918602
│   └── Posts
│       ├── 1653161503838
│       │   ├── Likes
│       │   │   ├── F9UM7opFTDcZGaqOU6RXMjr2yPj2: "Liked"
│       │   │   └── creator: "F9UM7opFTDcZGaqOU6RXMjr2yPj2"
│       │   ├── description: "A partir de la semana que viene empezaremos a trabajar con UNICEF en un nuevo proyecto. Habrán nuevas noticias en el futuro..."
│       │   ├── image: "https://firebasestorage.googleapis.com/v0/b/scout-connections.appspot.com/o/Posts%2Fpost_1653161503838?alt=media&token=202d918e-1d69-494"
│       │   ├── nComments: 0
│       │   ├── nLikes: 1
│       │   ├── pid: "1653161503838"
│       │   ├── time: "1653161503838"
│       │   └── title: "¡Trabajamos con UNICEF!"
│       └── 1653161653389
├── Tokens
│   ├── 0Eso1JfWBHQz6waZhpjSooB3mz1
│   │   ├── token: "cVlqQ3_7So6As5OemNvD--APA91bfPISbrknwFop19Jh9vHqUR6Ze4CbpXUVRIB9dUpZORCTXjwosBVzp7b99RyQuXw7MuSeFVROowZ1p1dQ272db11L"
│   ├── F9UM7opFTDcZGaqOU6RXMjr2yPj2
│   ├── HF3CaUsoACNBuyWClYq6yXpLMH2
│   ├── SRmRbTLZrYRmVsUr4qQu2yGuu8o1
│   ├── bK50w0TEGm8dfenaCI0ybH6oPR2
│   ├── mq0ubfYFpSYd711mCDh20Le1zW2
│   └── tG1okG8j1bZ86vuu9wJk07eZn73
└── Users
    ├── 0Eso1JfWBHQz6waZhpjSooB3mz1
    │   ├── cover: ""
    │   ├── email: "mmano@uma.es"
    │   ├── image: ""
    │   ├── monitor: false
    │   ├── name: "Manuel"
    │   ├── phone: "123456789"
    │   ├── status: "1652404762971"
    │   ├── typingTo: "noOne"
    │   └── uid: "0Eso1JfWBHQz6waZhpjSooB3mz1"
    ├── F9UM7opFTDcZGaqOU6RXMjr2yPj2
    └── HF3CaUsoACNBuyWClYq6yXpLMH2

```

Figura 3.9: Estructura de la base de datos en tiempo real de Firebase

Nombre	Tamaño	Tipo	Modificación más reciente
Events/	–	Carpeta	–
GroupChats/	–	Carpeta	–
Groups/	–	Carpeta	–
Posts/	–	Carpeta	–
Users/	–	Carpeta	–

Figura 3.10: Estructura del almacenamiento de la base de datos de Firebase

Nombre	Tamaño	Tipo	Modificación más reciente
event_1653229947493	1.05 MB	image/png	22 may 2022
event_1653230131543	31.77 KB	image/jpeg	22 may 2022

event_1653230131543 ×



Nombre
[event_1653230131543](#)

Tamaño
32,533 bytes

Tipo
image/jpeg

Creado
22 may 2022 16:35:33

Actualizado
22 may 2022 16:35:33

Ubicación del archivo ▾

Otros metadatos ▾

Figura 3.11: Contenido de una carpeta perteneciente al almacenamiento de la base de datos de Firebase

Antes de comenzar con las funcionalidades también será necesario será crear los archivos necesarios para los temas claro y oscuro de la aplicación, y los distintos idiomas que se van a añadir. Todo esto se realizará en el directorio “values”:

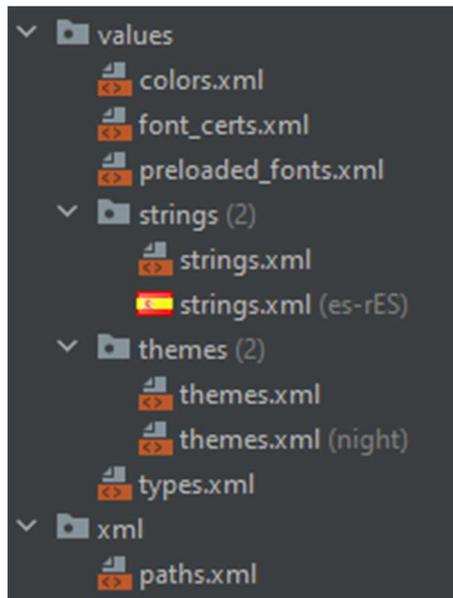


Figura 3.12: Directorio "values" del proyecto

Los idiomas serán definidos a partir de traducciones creadas en los archivos “strings.xml” en el directorio

```
1 <resources>
2 <string name="app_name">Scout Connections</string>
3 <string name="registering_user">Registering user...</string>
4 <string name="invalid_email">Invalid email</string>
5 <string name="password_length">Password length must be at least 6 characters</string>
6 <string name="user_registered">The user has been registered</string>
7 <string name="authentication_error">There was an authentication error</string>
8 <string name="email">Email</string>
9 <string name="register">Register</string>
10 <string name="have_account">Already have an account? Log in</string>
11 <string name="login">Login</string>
12 <string name="logging_in">Logging in...</string>
13 <string name="recover_password">Recover password</string>
14 <string name="recover">Recover</string>
15 <string name="existing_email">Please fill in the field with an existing email</string>
16 <string name="cancel">Cancel</string>
17 <string name="sending_email">Sending email...</string>
18 <string name="email_sent">Email sent</string>
```

Figura 3.13: Parte del código de “strings.xml”

Los temas correspondientes para el modo claro y oscuro de Android serán creados a través de Material Design. Material Design es una normativa de diseño enfocada en la visualización del sistema operativo Android, en la web y en cualquier plataforma, la cual está desarrollada por Google. Para poder aplicarlo, se realizará a través de los archivos “themes.xml” en el proyecto, los cuales serán importados a partir del resultado generado por la web de Material Design en función de los parámetros que se han definido previamente:

```
1 <resources>
2
3   <style name="ScoutConnections" parent="Theme.Material3.Light">
4     <item name="colorPrimary">@color/md_theme_light_primary</item>
5     <item name="colorOnPrimary">@color/md_theme_light_onPrimary</item>
6     <item name="colorPrimaryContainer">@color/md_theme_light_primaryContainer</item>
7     <item name="colorOnPrimaryContainer">@color/md_theme_light_onPrimaryContainer</item>
8     <item name="colorSecondary">@color/md_theme_light_secondary</item>
9     <item name="colorOnSecondary">@color/md_theme_light_onSecondary</item>
10    <item name="colorSecondaryContainer">@color/md_theme_light_secondaryContainer</item>
11    <item name="colorOnSecondaryContainer">@color/md_theme_light_onSecondaryContainer</item>
12    <item name="colorTertiary">@color/md_theme_light_tertiary</item>
13    <item name="colorOnTertiary">@color/md_theme_light_onTertiary</item>
14    <item name="colorTertiaryContainer">@color/md_theme_light_tertiaryContainer</item>
15    <item name="colorOnTertiaryContainer">@color/md_theme_light_onTertiaryContainer</item>
16    <item name="colorError">@color/md_theme_light_error</item>
```

Figura 3.14: Parte del código de “themes.xml”

3.4.1 Usuario

Las primeras funcionalidades que son necesarias añadir a la aplicación debido a su importancia son: añadir una página de inicio, permitir registrarse e iniciar/cerrar sesión, además de poder recuperar la contraseña en caso de su olvido. Pueden verse estas funciones de manera más visual en el manual de usuario adjunto.

- **Página de inicio:** La página de inicio será lo que se muestre al abrir la aplicación en caso de que no se haya iniciado sesión en ella, por lo que debe permitir registrarse o iniciar sesión. Si un usuario ha iniciado sesión con anterioridad, será redirigido a la página principal de la aplicación relacionada con las publicaciones. La clase dedicada para esta funcionalidad se encuentra en “MainActivity.kt”, a la cual se le asigna una vista “activity_main.xml” que se encuentra en el directorio res/layout:

```
9 class MainActivity : AppCompatActivity() {
10
11     private lateinit var mAuth: FirebaseAuth
12
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15         setContentView(R.layout.activity_main)
16
17         mAuth = FirebaseAuth.getInstance()
18
19         supportActionBar?.hide()
20
21         val registerBtn = findViewById<Button>(R.id.register_btn)
22         registerBtn.setOnClickListener { it: View!
23             startActivity(Intent(packageContext: this, RegisterActivity::class.java))
24         }
25
26         val loginBtn = findViewById<Button>(R.id.login_btn)
27         loginBtn.setOnClickListener { it: View!
28             startActivity(Intent(packageContext: this, LoginActivity::class.java))
29         }
30
31     }
```

Figura 3.15: Parte del código de “MainActivity.kt”

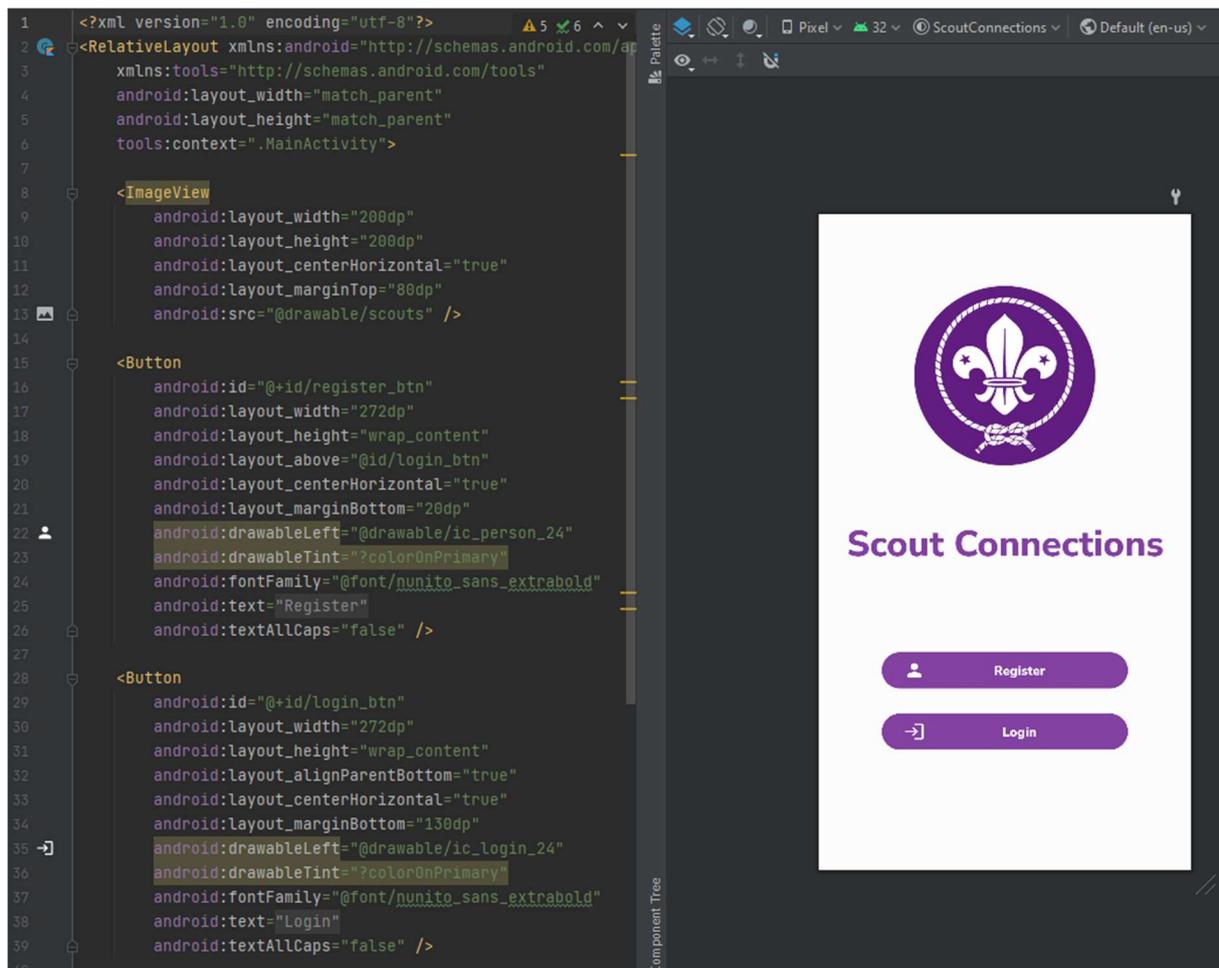


Figura 3.16: Parte del código de “activity_main.xml”

El código fuente de las siguientes clases no será mostrado por razones de extensibilidad, exceptuando funciones que sean necesarias para la comprensión del código.

La función “checkUserStatus” es constantemente utilizada en todas las clases de la aplicación, ya que en caso de que el usuario tenga iniciada la sesión, al abrir la aplicación será redireccionado a la página principal, y en caso de que este en otra parte de la aplicación y haya cerrado sesión, será redireccionado a la página de inicio. Estos son los casos descritos:

```

private fun checkUserStatus() {
    val user = mAuth.currentUser
    if (user != null) {
        startActivity(Intent(packageContext: this, DashboardActivity::class.java))
        finish()
    } else {
    }
}
}

```

Figura 3.17: Función “checkUserStatus” en “MainActivity.kt”

```

private fun checkUserStatus() {
    val user = mAuth.currentUser
    if (user == null) {
        startActivity(Intent(packageContext: this, MainActivity::class.java))
        finish()
    } else {
    }
}
}

```

Figura 3.18: Función “checkUserStatus” ubicada en las clases de la aplicación exceptuando “MainActivity.kt”

- **Registro:** Para realizar el registro se hace uso de la clase “RegisterActivity.kt” y su *layout* “activity_register.xml”. Será necesario aportar un correo electrónico y una contraseña segura (longitud mayor a 6). Una vez realizado el registro, se producirá un redireccionamiento a la página principal de publicaciones.

Cabe destacar la función “registerUser” que se encargará de crear la autorización para la cuenta y la añada a la base de datos:

```

 mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener { task ->
    if (task.isSuccessful) {
        progressDialog.dismiss()

        val user = mAuth.currentUser

        val email = user?.email
        val uid = user?.uid
        val pos = email?.indexOf( string: "@" )
        val name = email?.substring(0, Integer.parseInt(pos.toString()))

        var hashMap = HashMap<String, Any>()
        if (email != null) {
            hashMap["email"] = email
        }
        if (uid != null) {
            hashMap["uid"] = uid
        }
        if (name != null) {
            hashMap["name"] = name
        }
        hashMap["phone"] = "123456789"
        hashMap["image"] = ""
        hashMap["status"] = "online"
        hashMap["typingTo"] = "noOne"
        hashMap["cover"] = ""
        hashMap["monitor"] = false

        val db = FirebaseDatabase.getInstance( url: "https://scout-connections-default-rtdb.europe-west1.firebaseio.com" )
        val reference = db.getReference( path: "Users" )
        if (uid != null) {
            reference.child(uid).setValue(hashMap)
        }
    }
}

```

Figura 3.18: Parte de la función “registerUser” en “RegisterActivity.kt”

- **Inicio/cierre de sesión:** El inicio de sesión puede realizarse usando un correo electrónico y una contraseña de una cuenta registrada previamente o a través de una cuenta de Google. Se hace uso de la clase “LoginActivity.kt” y su *layout* “activity_login.xml”. El cierre de sesión se realizará desde el apartado de usuarios, que será explicado posteriormente. Una vez se haya iniciado sesión, se producirá un redireccionamiento a la página principal de publicaciones.

```

 googleLoginBtn.setOnClickListener { it: View!
    val signInIntent = googleSignInClient.signInIntent
    startActivityForResult(signInIntent, RC_SIGN_IN)
}

```

Figura 3.19: Acción realizada al pulsar el botón de inicio de sesión de Google en “LoginActivity.kt”

3.4.2 Publicaciones

Antes de comenzar con la explicación del apartado de publicaciones, es necesario mencionar la estructura general de la aplicación. Los diferentes apartados importantes del proyecto (publicaciones, perfil, social, eventos) podrán ser accedidos desde un menú de navegación, el cual se encuentra en la actividad de Android “DashboardActivity.kt”, y desde donde se puede navegar a través de los diferentes Fragmentos de Android “HomeFragment.kt”, “ProfileFragment.kt”, “SocialFragment.kt” y “EventsFragment.kt”, además de sus respectivos archivos xml con las interfaces, dedicados a los apartados antes mencionados.

```
navigationMenu = findViewById(R.id.navigation_menu)
navigationMenu.setOnItemClickListener { menuItem ->
    when (menuItem.itemId) {
        R.id.nav_home -> {
            actionBar?.title = "Home"
            val fragment = HomeFragment( dashboardActivity: this)
            val ft = supportFragmentManager.beginTransaction()
            ft.replace(R.id.content, fragment)
            ft.commit()
            true ^setOnNavigationItemSelectedListener
        }
        R.id.nav_perfil -> {
            actionBar?.title = "Profile"
            val fragment = ProfileFragment( dashboardActivity: this)
            val ft = supportFragmentManager.beginTransaction()
            ft.replace(R.id.content, fragment)
            ft.commit()
            true ^setOnNavigationItemSelectedListener
        }
        R.id.nav_chats -> {
            actionBar?.title = "Social"
            val fragment = SocialFragment( dashboardActivity: this)
            val ft = supportFragmentManager.beginTransaction()
            ft.replace(R.id.content, fragment)
            ft.commit()
            true ^setOnNavigationItemSelectedListener
        }
        R.id.nav_events -> {
            actionBar?.title = "Events"
            val fragment = EventsFragment( dashboardActivity: this)
            val ft = supportFragmentManager.beginTransaction()
            ft.replace(R.id.content, fragment)
            ft.commit()
            true ^setOnNavigationItemSelectedListener
        }
    }
}
```

Figura 3.20: Navegación de las funcionalidades de la aplicación en “DashboardActivity.kt”

La página de publicaciones será la página principal de la aplicación, una vez se ha iniciado sesión. Las funcionalidades destacadas para esta fase son: ver/crear/editar/eliminar publicaciones (CRUD (funciones básicas en una base de datos o en un *software*)), además de las acciones dar “me gusta”, comentar y compartir publicaciones. Pueden verse estas funciones de manera más visual en el manual de usuario adjunto.

- **Ver/crear/editar/eliminar publicaciones (CRUD):** Todas las entidades de la aplicación que aparezcan listadas (publicaciones, usuarios, conversaciones, eventos...) necesitan tres archivos necesarios cada una: un adaptador (ajusta la entidad en la lista con funcionalidades), una vista (aspecto de la entidad en la lista) y un modelo (estructura de los datos de la entidad utilizada al recoger los datos de la base de datos).

```
class PostModel {
    var pid: String? = null
    var title: String? = null
    var description: String? = null
    var image: String? = null
    var time: String? = null
    var creator: String? = null
    var nLikes: Int? = null
    var nComments: Int? = null

    constructor() {}
    constructor(
        pid: String?,
        title: String?,
        description: String?,
        image: String?,
        time: String?,
        creator: String?,
        nLikes: Int?,
        nComments: Int?
    ) {
        this.pid = pid
        this.title = title
        this.description = description
    }
}
```

Figura 3.21: Modelo de publicación en “PostModel.kt”

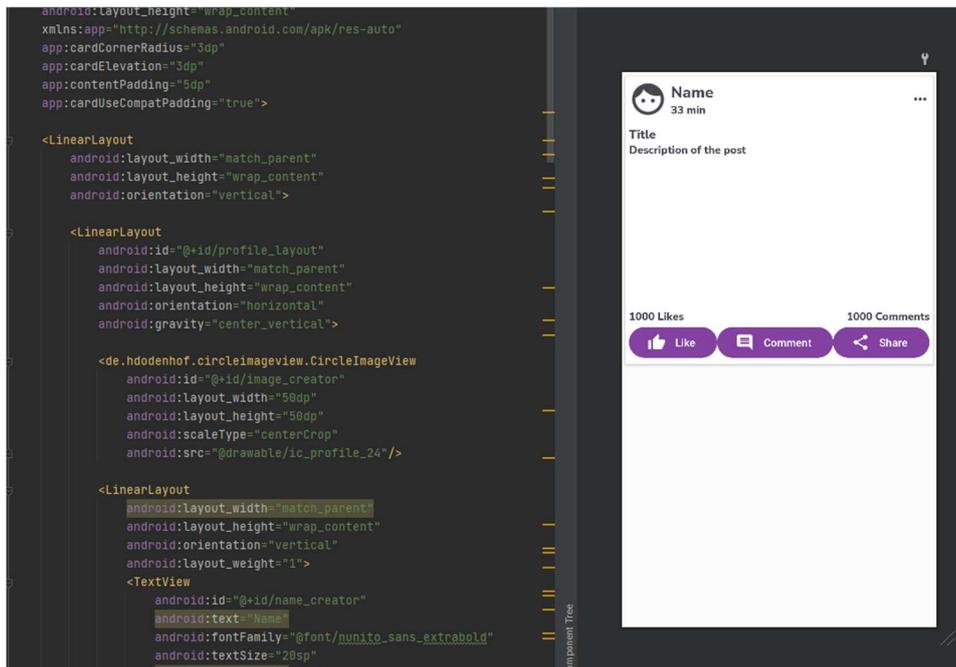


Figura 3.22: Parte del código de la vista de publicaciones en “row_posts.xml”

- **Dar “me gusta”, comentar y compartir publicaciones:** Para estas acciones, se han usado las mismas clases que se han utilizado en el CRUD anterior, entre ellas “PostAddActivity.kt”, “PostDetailActivity.kt”, “PostLikedByActivity.kt”, y sus respectivos archivos de vista xml.

```
private fun shareText(pTitle: String?, pDescription: String?) {
    val body = pTitle + "\n" + pDescription
    val sIntent = Intent(Intent.ACTION_SEND)
    sIntent.type = "text/plain"
    sIntent.putExtra(Intent.EXTRA_SUBJECT, "Subject here")
    sIntent.putExtra(Intent.EXTRA_TEXT, body)
    context.startActivity(Intent.createChooser(sIntent, "Share via"))
}
```

Figura 3.23: Función de compartir publicación en “PostAdapter.kt”

3.4.3 Perfil

El apartado de perfil se encuentra en el archivo “ProfileFragment.kt” y su respectiva vista “fragment_profile.xml”. Deberá permitir ver el perfil del usuario con el que se ha iniciado sesión, además de poder editarlo, cambiar la contraseña o solicitar rol de monitor en caso de ser scout. Pueden verse éstas funciones de manera más visual en el manual de usuario adjunto.

- **Ver perfil:** En esta página puede verse la información personal del usuario con el que se ha iniciado sesión (nombre, correo electrónico, teléfono, rol, imagen y fondo), y sus publicaciones creadas, en caso de que tenga.
- **Editar perfil, cambiar contraseña y solicitar rol:** Constantemente en la aplicación se hace uso de diálogos para acceder a distintas acciones, ya que permiten que no se haga uso de tantos archivos y clases, por lo que se consigue una mayor brevedad en el código.

También, se hace uso de la cámara y del almacenamiento del dispositivo por lo que es necesario añadir estos servicios al proyecto y controlarlos de manera que el usuario debe permitirlo también.

```

private fun showPhotoDialog() {
    val optionsPhoto = arrayOf("Take photo", "Select from gallery", "Delete current photo")
    val constructorPhoto = AlertDialog.Builder(activity)
    constructorPhoto.setTitle("Select an image")
    constructorPhoto.setItems(optionsPhoto) { _, pos ->
        when (pos) {
            0 -> {
                if (!checkCameraPermission()) {
                    requestCameraPermission()
                } else {
                    selectFromCamera()
                }
            }
            1 -> {
                if (!checkStoragePermission()) {
                    requestStoragePermission()
                } else {
                    selectFromGallery()
                }
            }
            2 -> {
                deleteCurrentPhoto()
            }
        }
    }
    constructorPhoto.create().show()
}

```

Figura 3.24: Función de muestra de las opciones de cambiar una imagen en “ProfileFragment.kt”

3.4.4 Social

Este apartado es uno de los más grandes de la aplicación, ya que esta formado por las conversaciones, grupos y filtros, además de todo lo que conlleva para su funcionamiento. Es por esto por lo que son necesarias muchas clases, entre ellas las más importantes son “SocialFragment.kt” y “ChatActivity.kt”, y sus archivos de vistas xml. Pueden verse las funciones de manera más visual en el manual de usuario adjunto.

- **Listar conversaciones individuales y grupales y su funcionamiento:** El listado de ambas conversaciones se realiza a través de adaptadores, vistas y modelos, tal como se mencionaba anteriormente, donde aparece el usuario y

último mensaje escrito en cada fila. El funcionamiento permitirá el envío de texto e imágenes, permitiendo ser borrados por su creador, contando también con el estado del último mensaje y notificaciones en la conversación individual.

```
private fun sendNotification(userId: String, name: String?, message: String) {
    val tokens = FirebaseDatabase.getInstance( url: "https://scout-connections-default-rtdb.europe-west1.firebaseio.com")
        .getReference( path: "Tokens")
    val query = tokens.orderByKey().equalTo(userId)
    query.addValueEventListener(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            for (ds: DataSnapshot in snapshot.children) {
                val token = ds.getValue(Token::class.java)
                val data = Data(user!!.uid, message, name, userId, R.drawable.ic_chat_24)
                val sender = Sender(data, token!!.token)

                try {
                    val senderJsonObj = JSONObject(Gson().toJson(sender))

                    val jsonObjectRequest: JsonObjectRequest = object : JsonObjectRequest(
                        Method.POST,
                        url: "https://fcm.googleapis.com/fcm/send",
                        senderJsonObj,
                        Response.Listener { response ->
                            Log.d(
                                tag: "JSON_RESPONSE",
                                msg: "onResponse:$response"
                            )
                        },
                        Response.ErrorListener { error ->
                            Log.d(
                                tag: "JSON_RESPONSE",
                                msg: "onResponse:$error"
                            )
                        }
                    ) {
                        @Throws(AuthFailureError::class)
                        override fun getHeaders(): Map<String, String> {
                            val headers = HashMap<String, String>()
                            //headers.put("Content-Type", "application/json")
                        }
                    }
                } catch (e: Exception) {
                    // Log.e("sendNotification", e.message)
                }
            }
        }
    })
}
```

Figura 3.25: Parte de la función “sendNotification” en “ChatActivity.kt”

- **Ver/crear/editar/eliminar grupos (CRUD):** El CRUD de grupos tiene cierto parecido al de las publicaciones, pero tiene una gran diferencia en la edición, ya que además de la edición de título, descripción o imagen, se debe poder salir de un grupo, añadir/eliminar participantes y cambiar sus roles.

```

LeaveGroupInfo.setOnClickListener { it: View!
    var title = ""
    var description = ""
    var positiveButton = ""
    if (role == "creator") {
        title = "Delete group"
        description = "Are you sure you want to delete this group?"
        positiveButton = "Delete"
    } else {
        title = "Leave group"
        description = "Are you sure you want to leave this group?"
        positiveButton = "Leave"
    }
}
val builder = AlertDialog.Builder(context: this@GroupInfoActivity)
builder.setTitle(title).setMessage(description)
builder.setPositiveButton(
    positiveButton,
    DialogInterface.OnClickListener { _, _ ->
        if (role == "creator") {
            deleteGroup()
        } else {
            leaveGroup()
        }
    })
builder.setNegativeButton(
    "Cancel",
    DialogInterface.OnClickListener { dialogInterface, _ ->
        dialogInterface.dismiss()
    })
builder.create().show()
}

```

Figura 3.26: Funcionamiento del botón Eliminar/salir de un grupo en “GroupInfoActivity.kt”

- **Buscar/filtrar usuarios y conversaciones:** La aplicación también permite la búsqueda y filtro de los usuarios y conversaciones en el apartado de social.

3.4.5 Eventos

Esta se trata de la última funcionalidad añadida realizada en el último *sprint*, la cual permite listar/crear/editar/borrar eventos, además de filtrarlos por fecha. Pueden verse estas funciones de manera más visual en el manual de usuario adjunto.

- **Ver/crear/editar/eliminar eventos (CRUD):** Se trata de un CRUD básico sobre los eventos del proyecto.
- **Filtrar eventos:** Esta es una de las características más importantes de los eventos, ya que la diferencia más importante que posee es la fecha. Al entrar al apartado aparecen los siguientes eventos, aunque el filtro hará que aparezcan los eventos que se realizan en la fecha exacta seleccionada.

```

private fun loadEventsDate(timeInMillis: Long) {
    val db = FirebaseDatabase.getInstance( url: "https://scout-connections-default-rtdb.firebaseio.com")
    val reference = db.getReference( path: "Events")
    val dateSelected = SimpleDateFormat( pattern: "dd/MM/yyyy").format(timeInMillis)
    timeEvent.visibility = View.VISIBLE
    timeEvent.setText(dateSelected)
    titleEvent.text = "Events" + " " + "on" + " " + dateSelected

    var listEvents: MutableList<EventModel> = ArrayList()

    reference.addValueEventListener(object : ValueEventListener {
        override fun onDataChange(dataSnapshot: DataSnapshot) {
            listEvents.clear()
            dataSnapshot.children.forEach { it: DataSnapshot!
                val eventModel = it.getValue(EventModel::class.java)
                val t = eventModel!!.timeEvent!!.toLong()
                val dateEM = SimpleDateFormat( pattern: "dd/MM/yyyy").format(t)

                if(dateSelected == dateEM){
                    listEvents.add(eventModel!!)
                }
            }
            listEvents.sortBy { it.timeEvent }

            val eventAdapters = EventAdapter(dashboardActivity, listEvents)
            recyclerView.adapter = eventAdapters
        }
    })

    override fun onCancelled(error: DatabaseError) {
        Toast.makeText(activity, error.message, Toast.LENGTH_SHORT).show()
    }
}
}
}

```

Figura 3.27: Función de muestra de las opciones de cambiar una imagen en “ProfileFragment.kt”

3.5 Pruebas y revisión

El proyecto desarrollado ha sido tratado a fondo a través de pruebas de funcionamiento real con varios dispositivos virtuales de Android Studio, en distintas versiones de Android, los cuales serán explicados a continuación. Android Studio posee *tests* internos los cuales son pasados correctamente, por lo que no existen errores al compilar y ejecutar el proyecto. Asimismo, las llamadas a la base de datos de Firebase son realizadas con éxito.

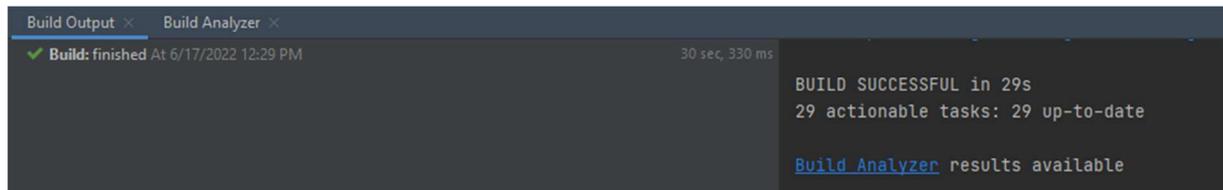


Figura 3.28: Resultado de la compilación del proyecto en Android Studio

Queda pendiente la creación y desarrollo de *tests* unitarios a través de Android Studio y Firebase, los cuales están planeados de ser realizados en un futuro y serán comentados en el apartado de “Conclusiones y Líneas Futuras”.

El proyecto ha sido revisado y aceptado por el Propietario del producto, por lo que se da por correcta y completada, ya que se cumplen las acciones que se esperan de la aplicación y todos los requisitos descritos previamente de la mejor manera posible.

3.6 Instalación y despliegue

Para el desarrollo y ejecución de la aplicación desarrollada se ha hecho uso de varios modelos de dispositivos virtuales emulados de Android con diferentes versiones de Android proveídos por Android Studio, aunque el más utilizado ha sido el Pixel 3a.



Figura 3.29: *Google Pixel 3a XL*. [Imagen]. Recuperado el 8 de junio de 2022, de <https://www.notebookcheck.org/Google-Pixel-3a-XL.421866.0.html>.

Características del dispositivo:

- Android 11
- Resolución 1080 x 2220; 440dpi
- 13 GB de almacenamiento en disco

- 1.5 GB de RAM
- 6 GB de almacenamiento interno
- 10 núcleos Multi-Core

Aunque el desarrollo y ejecución de la aplicación se ha realizado sobre un dispositivo virtual, también es posible ejecutarla en un dispositivo físico a través de la instalación en Android del archivo “app-debug.apk” el cual es generado automáticamente por Android Studio y se encuentra en el directorio “ScoutConnections\app\build\outputs\apk\debug”.

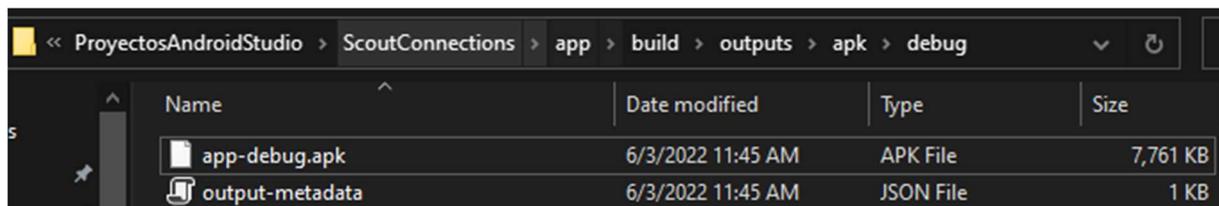


Figura 3.30: Ubicación del archivo “app-debug.apk”

3.7 Documentación

Durante la fase de documentación, se realiza la creación de los informes necesarios para la finalización del proyecto y la comprensión de la aplicación desarrollada. Los entregables del proyecto para su documentación son:

- **Manual de Usuario**
- **Memoria del Trabajo de Fin de Grado**

Conclusiones y Líneas Futuras

Los objetivos que se habían propuesto para el desarrollo de este proyecto han sido logrados de forma eficaz y eficiente, además de haber sido extendidos con nuevas ideas y funcionalidades. Esto ha sido gracias a la elaboración de “Scout Connections”, una aplicación para dispositivos móviles, desarrollada a partir del entorno Android Studio, con el lenguaje de programación Kotlin, con Firebase como base de datos y una serie de librerías necesarias para su correcto funcionamiento.

Este proyecto ha supuesto un gran reto ya que se ha hecho uso de tecnologías, lenguajes, librerías, herramientas y fuentes de datos desconocidos y nuevos, provocando una experiencia muy satisfactoria, aunque en ciertos momentos abrumadora. Abordar las tareas de la Ingeniería del Software en un proyecto complejo también ha sido un enorme desafío, sin embargo, este trabajo no es en vano ya que la experiencia recogida será de gran provecho en el mundo laboral.

Cualquier desarrollo de una aplicación móvil necesita de un continuo mantenimiento, actualización de librerías, versiones y tecnologías, ya que implicará ventajas respecto a rendimiento, gestión de memoria, batería, etc. Igualmente, puede resultar interesante la extensión de las funcionalidades del sistema, como, por ejemplo:

- **Migración de la aplicación móvil desarrollada en Android a iOS:** De esta manera el número de usuarios que se puede alcanzar es mucho mayor, y así quedaría disponible para prácticamente todo el mercado móvil.
- **Migración de la aplicación móvil desarrollada en Android a aplicación web:** Así se conseguiría que la aplicación se pudiese ejecutar desde navegadores

web en cualquier dispositivo móvil y no móvil sin requerir que se realice ninguna instalación.

- **Nuevos métodos de identificación:** Actualmente el inicio de sesión puede realizarse con una cuenta creada manualmente o con una cuenta de Google. Podría extenderse a otras plataformas tales como Facebook o Twitter, entre otras.
- **Creación de *tests* unitarios:** Siempre es recomendable crear *tests* específicos para las distintas funcionalidades de la aplicación en Android Studio y Firebase, ya que así se consigue una comprobación de las diferentes llamadas que se realizan, además del correcto funcionamiento.
- **Implementación completa de notificaciones:** Ahora mismo la aplicación provee de notificaciones para las conversaciones individuales, aunque es buena idea añadir notificaciones en grupos, creaciones de publicaciones y eventos, además de comentarios y “me gusta”.
- **Sección de ajustes:** Una buena incorporación podría ser un apartado de ajustes donde poder cambiar ajustes avanzados de la aplicación, tal como la descarga de imágenes, activar/desactivar notificaciones o bloquear/desbloquear usuarios, etc.

Bibliografía

Competencias profesionales (Ingeniería Informática) – Salidas Profesionales COIE. Blogs.uned.es. Recuperado el 13 de abril de 2022, de <http://blogs.uned.es/salidasprofesionales/grados/ingenieria-y-arquitectura/grado-en-ingenieria-informatica/competencias-profesionales-ingenieria-informatica/>.

¿Qué es Android?. Xataka. Recuperado el 13 de abril de 2022, de <https://www.xatakandroid.com/sistema-operativo/que-es-android>.

Android - Wikipedia, la enciclopedia libre. Es.wikipedia.org. Recuperado el 13 de abril de 2022, de <https://es.wikipedia.org/wiki/Android>.

Qué es Android - Definición, significado y ejemplos. Arimetrics. Recuperado el 25 de abril de 2022, de <https://www.arimetrics.com/glosario-digital/android>.

Platform Architecture | Android Developers. Android Developers. Recuperado el 29 de abril de 2022, de <https://developer.android.com/guide/platform>.

Qué es Kotlin y características. OpenWebinars.net. Recuperado el 31 de abril de 2022, de <https://openwebinars.net/blog/que-es-kotlin/>.

¿Qué es Kotlin? ¿Es mejor que Java? | KeepCoding Tech School. KeepCoding Tech School. Recuperado el 1 de mayo de 2022, de <https://keepcoding.io/blog/que-es-kotlin-es-mejor-que-java/#:~:text=Kotlin%20es%20un%20lenguaje%20de,funcional%20y%20orientada%20a%20objetos>.

Thornsby, J. *Java vs. Kotlin: ¿Deberías Usar Kotlin en Desarrollo Android?.* Code Envato Tuts+. Recuperado el 1 de mayo de 2022, de

<https://code.tutsplus.com/es/articles/java-vs-kotlin-should-you-be-using-kotlin-for-android-development--cms-27846>.

Kotlin vs Java, diferencias y ventajas. OpenWebinars.net. Recuperado el 7 de mayo de 2022, de <https://openwebinars.net/blog/kotlin-vs-java/>.

Introducción a Android Studio | Desarrolladores de Android | Android Developers. Android Developers. Recuperado el 17 de mayo de 2022, de <https://developer.android.com/studio/intro?hl=es-419>.

Hacer aplicaciones para Android: ¿Qué es un entorno de desarrollo y cuál es el mejor?. El Español. Recuperado el 27 de mayo de 2022, de https://www.elespanol.com/elandroidelibre/20200518/hacer-aplicaciones-android-entorno-desarrollo-mejor/490952339_0.html.

Android Studio - Wikipedia, la enciclopedia libre. Es.wikipedia.org. Recuperado el 27 de mayo de 2022, de https://es.wikipedia.org/wiki/Android_Studio.

Descripción general de proyectos | Desarrolladores de Android | Android Developers. Android Developers. Recuperado el 27 de mayo de 2022, de <https://developer.android.com/studio/projects?hl=es-419>.

Firebase - Wikipedia, la enciclopedia libre. Es.wikipedia.org. Recuperado el 30 de mayo de 2022, de <https://es.wikipedia.org/wiki/Firebase>.

Cardona, M. *Firebase, qué es y para qué sirve la plataforma de Google.* Thinking for Innovation. Recuperado el 30 de mayo de 2022, de <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>.

Qué es Firebase: Conoce la plataforma de Google. OpenWebinars.net. Recuperado el 1 de junio de 2022, de <https://openwebinars.net/blog/que-es-firebase-de-google/>.

Estructura tu base de datos / *Firestore Documentation*. Firebase. Recuperado el 1 de junio de 2022, de <https://firebase.google.com/docs/database/web/structure-data?hl=es#:~:text=C%C3%B3mo%20se%20estructuran%20los%20datos%3A%20un%20%C3%A1rbol%20JSON,no%20hay%20tablas%20ni%20registros>.

Eventos, E. *6 ventajas de la metodología Scrum (Agile) para tu empresa* – Edt Eventos. Edt.es. Recuperado el 8 de junio de 2022, de <https://www.edt.es/6-ventajas-de-la-metodologia-scrum-agile-para-tu-empresa/>.

Qué es SCRUM. Proyectos Ágiles. Recuperado el 8 de junio de 2022, de <https://proyectosagiles.org/que-es-scrum/>.

Arrarte, A. *Las 5 fases de la metodología Scrum para tus proyectos* . AlvaroArrarte.com. Recuperado el 8 de junio de 2022, de <https://alvaroarrarte.com/fases-de-la-metodologia-scrum/#Tercera fase de Scrum Implementacion>.

Scrum: roles y responsabilidades / *Deloitte España*. Deloitte Spain. Recuperado el 10 de junio de 2022, de <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>.

▷ *Fases de la Metodología Scrum* **【5 Etapas y 16 Procesos】**. Blog - ComparaSoftware. Recuperado el 10 de junio de 2022, de <https://blog.comparasoftware.com/fases-metodologia-scrum/>.

Metodología Scrum. Honduras Digital Challenge. Recuperado el 15 de junio de 2022, de <https://hondurasdigitalchallenge.com/2020/05/21/metodologia-scrum-una-herramienta-util-para-agilizar-tus-proyectos/>.

Los 10 beneficios de la metodología Scrum. Lãberit. Recuperado el 15 de junio de 2022, de <https://www.laberit.com/los-10-beneficios-la-metodologia-scrum/>.

Ciclo de vida del software: todo lo que necesitas saber. Intelequia. Recuperado el 18 de junio de 2022, de <https://intelequia.com/blog/post/2083/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>.

5 Fases del Proceso de Desarrollo de Software - Blog Solbyte. Blog Solbyte. Recuperado el 18 de junio de 2022, de <https://www.solbyte.com/blog/5-etapas-del-proceso-de-desarrollo-de-software/>.

Interaction Flow Modeling Language - Wikipedia. En.wikipedia.org. Recuperado el 18 de junio de 2022, de https://en.wikipedia.org/wiki/Interaction_Flow_Modeling_Language.

Apéndices

Manual de Usuario

MANUAL DE USUARIO

SCOUT CONNECTIONS

Índice

Inicio de la aplicación	76
Registro	77
Inicio de sesión	78
Publicaciones	79
Listado de publicaciones.....	79
Dar “Me gusta” a una publicación	80
Ver personas a las que les gusta una publicación	81
Crear/Ver comentarios de una publicación	82
Eliminar comentario de una publicación	83
Compartir una publicación.....	84
Ver creador de una publicación.....	85
Crear publicación.....	85
Editar publicación	86
Borrar publicación	87
Perfil	88
Ver mi perfil.....	88
Editar mi perfil.....	89
Editar imagen de perfil.....	90
Editar fondo de perfil	90
Cambiar nombre de perfil	91
Cambiar teléfono de perfil.....	92
Cambiar contraseña de perfil	93
Solicitar rol de monitor	94
Cerrar sesión.....	95
Ver perfil de otro usuario	96
Social	97
Conversaciones	97
Listado de conversaciones.....	98

Crear conversación	98
Ver conversación	99
Enviar mensaje.....	99
Enviar imagen	100
Eliminar mensaje/imagen.....	101
Ver perfil del usuario con el que se está conversando.....	101
Grupos.....	102
Listado de grupos	102
Crear grupo	102
Ver/Crear mensajes.....	103
Información del grupo	104
Editar grupo.....	105
Añadir participante al grupo.....	106
Ver participantes del grupo.....	106
Eliminar participante del grupo	107
Cambiar rol de un participante del grupo	107
Eliminar grupo	108
Salir del grupo.....	108
Buscar conversaciones/grupos	109
Buscar usuarios	110
Eventos.....	111
Sigüientes eventos.....	111
Filtrar por fecha	112
Crear evento	113
Editar evento.....	114
Eliminar evento	115

Inicio de la aplicación

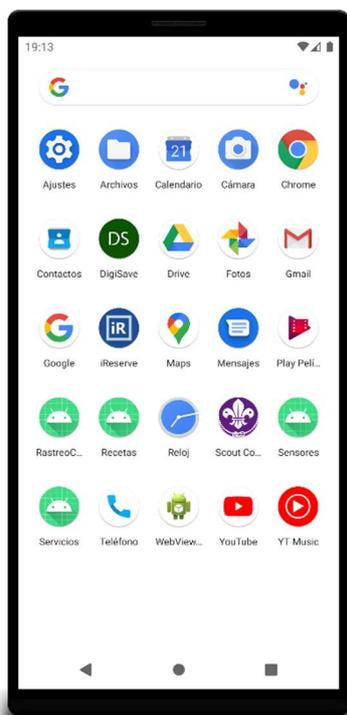


Figura 4.1.1: Aplicaciones del usuario



Figura 4.1.2: Inicio de la aplicación

Se da por hecho que la aplicación ya se encuentra instalada en el dispositivo y que su versión de Android es superior a 4.4, por lo que solo será necesario pulsar sobre su icono en el apartado de aplicaciones instaladas (ver figura 4.1.1). Se abrirá la aplicación junto con su inicio (ver figura 4.1.2).

Registro



Figura 4.1.3: Registro

Para poder registrarse es necesario acceder al apartado de registro, rellenar los campos con el correo electrónico y una contraseña con mínimo 6 dígitos, y pulsar en “Registrarse” (ver figura 4.1.3). En caso de ya tener una cuenta, pulsando el enlace inferior se accedería al inicio de sesión. Al crear la cuenta, se abre la página principal de la aplicación.

Inicio de sesión



Figura 4.1.4: Inicio de sesión



Figura 4.1.5: Recuperar contraseña

El inicio de sesión se podrá realizar mediante una cuenta creada previamente, rellenando los campos con el correo electrónico y contraseña, o a través de una cuenta de Google. En caso de no tener una cuenta, podrá crearse una nueva pulsando sobre el enlace inferior (ver figura 4.1.4). Si se ha olvidado la contraseña, podrá recuperarse pulsando el botón proporcionado, el cual abrirá un desplegable en el que se indicará el correo con el que la cuenta está creada y en el que se recibirá un email donde podrá cambiar a una nueva contraseña (ver figura 4.1.5).

Publicaciones



Figura 4.2.1: Publicaciones

Listado de publicaciones

El listado de publicaciones aparece en el apartado “Principal” de la aplicación (ver figura 4.2.1).

Dar “Me gusta” a una publicación



Figura 4.2.2: Dar "Me gusta" a una publicación

Para dar “Me gusta” a una publicación habrá que pulsar el botón “Me gusta” (ver figura 4.2.1). El texto del botón cambiará a “Te gusta”, produciéndose la acción de forma exitosa (ver figura 4.2.2).

Ver personas a las que les gusta una publicación

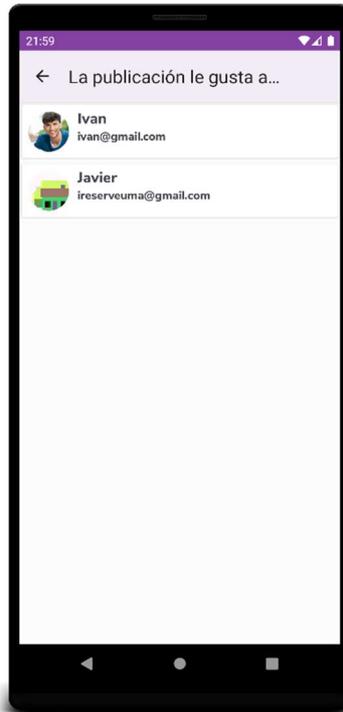


Figura 4.2.3: Personas a las que le gusta una publicación

Al pulsar el texto “X Me gusta” (ver figura 4.2.1) siendo X el número de “Me gusta” que tiene una publicación, se abrirá el listado de personas a las que les gusta tal publicación (ver figura 4.2.3). Si pulsamos sobre cada uno de ellos en la lista, accederemos a sus perfiles.

Crear/Ver comentarios de una publicación



Figura 4.2.4: Comentarios de una publicación



Figura 4.2.5: Escribir un comentario

Para poder acceder al apartado de comentarios de una publicación, podrá hacerse a través del botón “Comentar” y el texto “X Comentarios” siendo X el número de comentarios (ver figura 4.2.4). Para poder añadir un comentario es necesario rellenar el campo del comentario inferior (ver figura 4.2.4), y la información de la publicación con el nuevo comentario será actualizada (ver figura 4.2.5). En esta página también se podrá dar “Me gusta” y compartir la publicación.

Eliminar comentario de una publicación

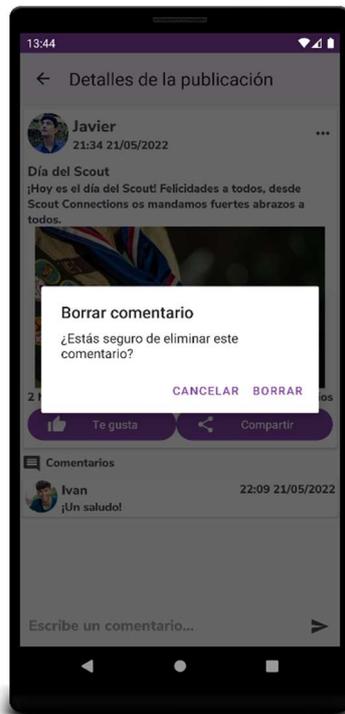


Figura 4.2.6: Borrar comentario

Un comentario podrá ser eliminado al ser pulsado sobre sí mismo por el usuario que lo creó (ver figura 4.2.6).

Compartir una publicación

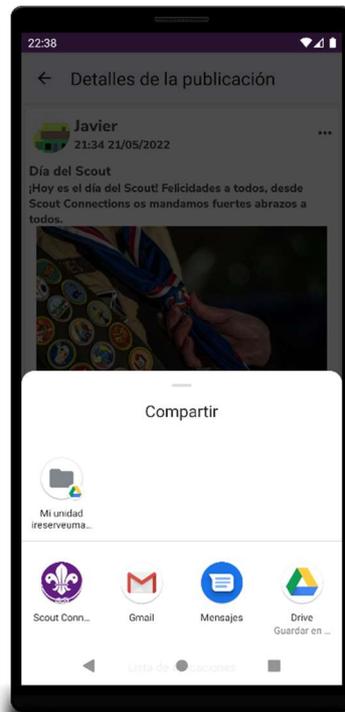


Figura 4.2.7: Compartir publicación

Pulsando el botón “Compartir” (ver figura 4.2.1) se podrá compartir la publicación en una aplicación externa a “Scout Connections” a través de las aplicaciones que lo permitan (ver figura 4.2.7).

Ver creador de una publicación

El perfil del creador de la publicación puede verse pulsando sobre el nombre del creador que aparece encima de la publicación (ver figura 4.2.1).

Crear publicación



Figura 4.2.8: Añadir publicación

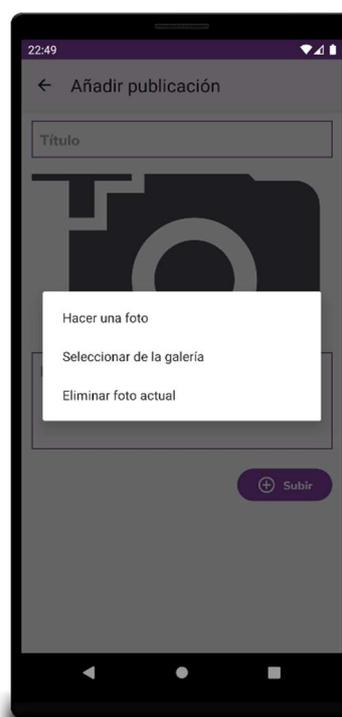


Figura 4.2.9: Selección de imagen

Para añadir una nueva publicación es necesario pulsar en el botón superior derecho (ver figura 4.2.1) y tener el rol Monitor, en caso de no tenerlo no aparecerá el botón. Para crear la publicación es necesario dar un título y descripción, aunque no es necesario añadir imagen (ver figura 4.2.8 y 4.2.9). Para crearlo, finalmente se pulsará en “Subir”.

Editar publicación



Figura 4.2.10: Ajustes de publicación



Figura 4.2.11: Editar publicación

Al pulsar el botón “...” se abrirá un panel con ajustes sobre la publicación de la que es creador (ver figura 4.2.10), pudiendo elegir si editar o borrar. Escogiendo “Editar” se abrirá el editor de la publicación, pudiendo editar título, imagen o descripción (ver figura 4.2.11).

Borrar publicación

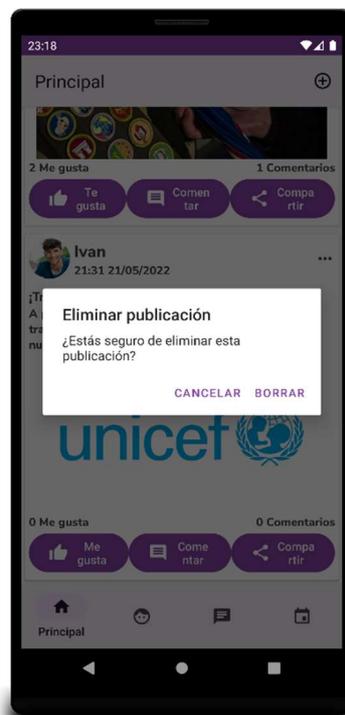


Figura 4.2.12: Borrar publicación

Se puede borrar una publicación ya creada escogiendo “Borrar” en las opciones (ver figura 4.2.10). Será necesario confirmar el borrado.

Perfil



Figura 4.3.1: Perfil

Para acceder al apartado de perfil es necesario pulsar en el botón “Perfil” en el menú de navegación inferior (ver figura 4.3.1).

Ver mi perfil

En el apartado de perfil, se puede visualizar información personal sobre el usuario con el que se ha iniciado sesión, además de sus publicaciones creadas (ver figura 4.3.1).

Editar mi perfil

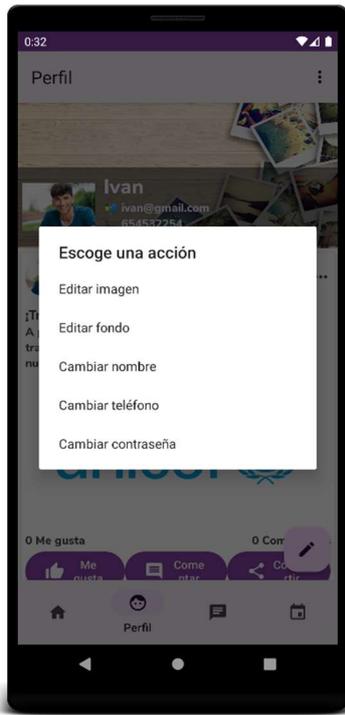


Figura 4.3.2: Editar perfil de Monitor



Figura 4.3.3: Editar perfil de Scout

El menú desplegable para editar el perfil (ver figura 4.3.2 y 4.3.3) aparecerá al pulsar el menú inferior derecho con el símbolo de un lápiz (ver figura 4.3.1) en función de si el usuario tiene rol de “Monitor” o “Scout”.

Editar imagen de perfil



Figura 4.3.4: Editar imagen de perfil

El menú de selección de imagen de perfil aparecerá al seleccionar “Editar imagen” (ver figura 4.3.2). Podrá ser actualizada mediante una foto de la galería o una tomada en el momento, o a su vez, ser eliminada (ver figura 4.3.4).

Editar fondo de perfil

El menú de selección de fondo de perfil aparecerá al seleccionar “Editar fondo” (ver figura 4.3.2). Podrá ser actualizada mediante una foto de la galería o una tomada en el momento, o a su vez, ser eliminada. Se trata de un menú similar al de editar imagen (ver figura 4.3.4).

Cambiar nombre de perfil

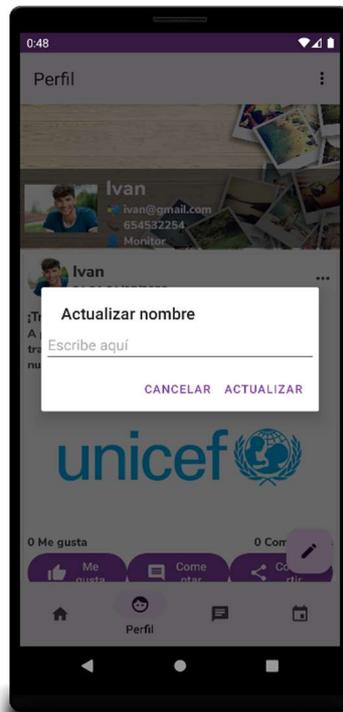


Figura 4.3.5: Cambiar nombre de perfil

Podrá editarse el nombre de perfil al seleccionar “Cambiar nombre” (ver figura 3.2). Será necesario rellenar el campo con un nombre no vacío y pulsar en “Actualizar” (ver figura 3.5).

Cambiar teléfono de perfil

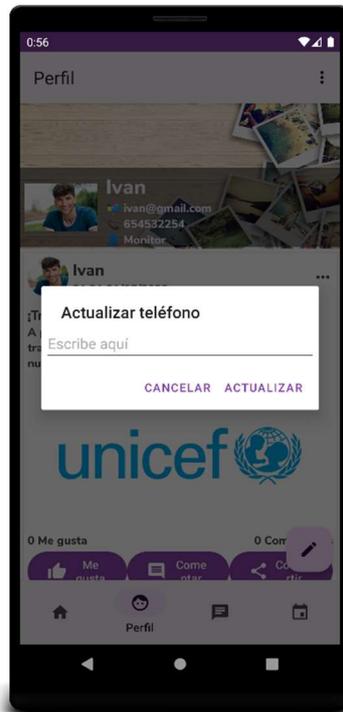


Figura 4.3.6: Cambiar teléfono de perfil

Podrá editarse el teléfono de perfil al seleccionar “Cambiar teléfono” (ver figura 4.3.2). Será necesario rellenar el campo con un teléfono no vacío y pulsar en “Actualizar” (ver figura 4.3.6).

Cambiar contraseña de perfil



Figura 4.3.7: Cambiar contraseña de perfil

Para poder actualizar la contraseña con la que se realiza el inicio de sesión habrá que pulsar en “Cambiar contraseña” (ver figura 4.3.2). Será necesario poner la antigua contraseña y una nueva con un mínimo de 6 dígitos. Para que se realicen los cambios habrá que pulsar en “Actualizar” (ver figura 4.3.7).

Solicitar rol de monitor

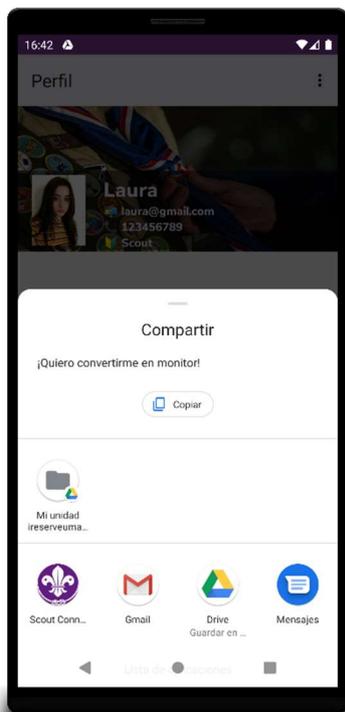


Figura 4.3.8: Solicitar rol de monitor



Figura 4.3.9: Enviar correo de solicitud de rol

Un usuario con rol de Scout puede solicitar rol de monitor a través de un correo electrónico a la cuenta scoutconnectionstfg@gmail.com, el cual se realizará pulsando “Solicitar rol de monitor” en el menú (ver figura 4.3.3), y a su vez a través de cualquier servicio de correo electrónico (ver figura 4.3.8 y 4.3.9).

Cerrar sesión



Figura 4.3.10: Cerrar sesión

Al desplegar el menú superior derecho (ver figura 4.3.1), al pulsar “Cerrar sesión” (ver figura 4.3.10), automáticamente se cerrará sesión del usuario con el que se encontraba ya iniciado.

Ver perfil de otro usuario



Figura 4.3.11: Perfil de otra persona

El perfil de una persona diferente al que se ha iniciado sesión también puede verse, aunque para ello, es necesario acceder desde distintos apartados de la aplicación como han sido o serán mencionados. Podrá verse su información personal y publicaciones creadas, además de tener la oportunidad de abrir el chat para empezar o continuar la conversación (ver figura 4.3.11).

Social



Figura 4.4.1: Social

El apartado social de la aplicación podrá accederse a través del menú de navegación inferior, pulsando “Social”, apareciendo su página principal (ver figura 4.4.1).

Conversaciones

La lista de conversaciones con usuarios aparece en la página principal de “Social” (ver figura 4.4.1) junto con el último mensaje enviado y su fecha correspondiente.

Listado de conversaciones

El listado, tal como se ha comentado, aparece en la página principal de “Social” (ver figura 4.4.1).

Crear conversación

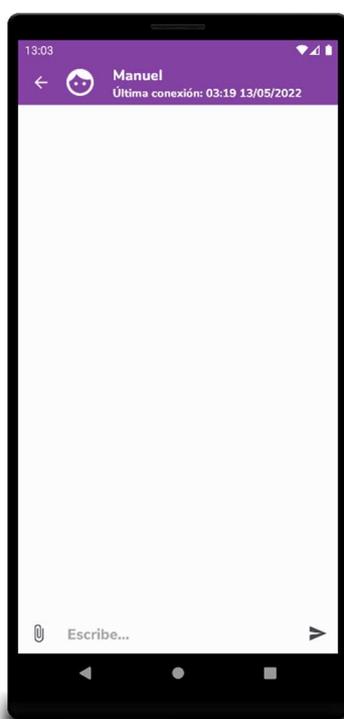


Figura 4.4.2: Crear conversación

Se podrá crear una conversación desde varios apartados de la aplicación, los cuales han sido o será comentados durante el manual. Al crear la conversación, la lista de mensajes entre los usuarios aparecerá vacía (ver figura 4.4.2).

Ver conversación



Figura 4.4.3: Conversación

Para acceder a una conversación, bastará con pulsar en cualquiera de las que ya existan (ver figura 4.4.1), abriéndose la lista de mensajes que existe entre los dos usuarios (ver figura 4.4.3)

Enviar mensaje

Para enviar un mensaje, solo habrá que rellenar el campo de texto inferior y darle al botón derecho (ver figura 4.4.2).

Enviar imagen



Figura 4.4.4: Enviar imagen en una conversación

Para enviar una imagen, solo habrá que pulsar en el clip izquierdo (ver figura 4.4.2). Se podrá seleccionar una imagen desde la galería o tomar una nueva con la cámara

Eliminar mensaje/imagen

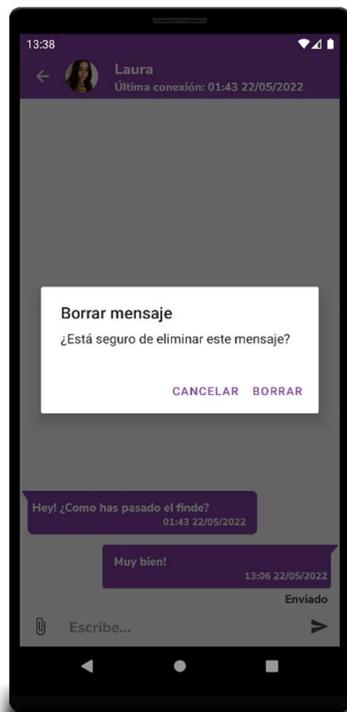


Figura 4.4.5: Eliminar mensaje

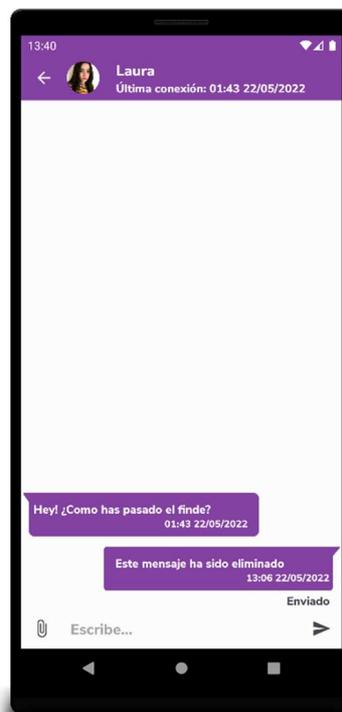


Figura 4.4.6: Mensaje eliminado

Un mensaje o imagen podrá ser eliminado. Para ello hay que pulsar en el que se quiere eliminar, mostrando un mensaje de confirmación de borrado (ver figura 4.4.5). Al confirmar, la información del mensaje/imagen ya no será visible (ver figura 4.4.6).

Ver perfil del usuario con el que se está conversando

Pulsando sobre el nombre del usuario con el que se está conversando se podrá visitar el perfil.

Grupos

Los grupos a los que pertenece el usuario aparecen en la página principal de “Social” (ver figura 4.4.1).

Listado de grupos

Como se ha comentado, el listado aparece en la página principal de “Social” (ver figura 4.4.1).

Crear grupo



Figura 4.4.7: Crear grupo

Para crear un grupo, es necesario pulsar el botón derecho que aparece en la barra superior (ver figura 4.4.1), apareciendo la página para crear el grupo, necesitando añadir un título y descripción, y una imagen opcionalmente (ver figura 4.4.7).

Ver/Crear mensajes



Figura 4.4.8: Conversación de grupo

Para ver y crear mensajes, se realizará de la misma manera que se realiza con las conversaciones individuales.

Información del grupo



Figura 4.4.9: Información del grupo

Puede accederse a la información de un grupo pulsando en su nombre dentro de la conversación (ver figura 4.4.8).

Editar grupo



Figura 4.4.10: Editar grupo

Para editar un grupo, será necesario ser el creador del grupo. Habrá que pulsar “Editar grupo” en la información del grupo (ver figura 4.4.9). Se abrirá una página para su edición (ver figura 4.4.10).

Añadir participante al grupo

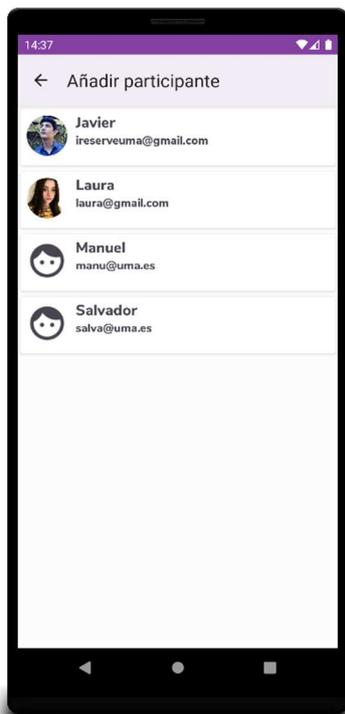


Figura 4.4.11: Añadir participante

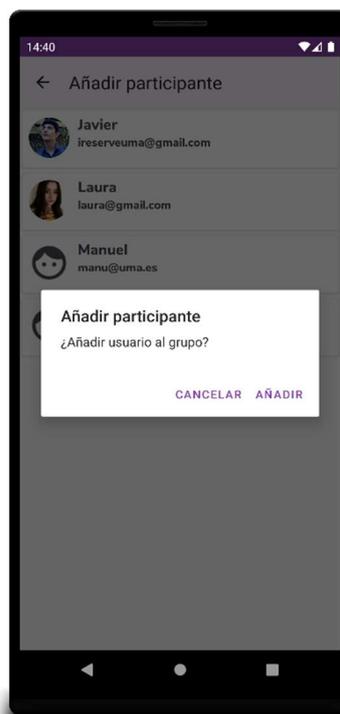


Figura 4.4.12: Confirmar adición de usuario

Para añadir un nuevo participante al grupo será necesario ser el creador o un administrador del grupo. Habrá que pulsar “Añadir participante” en el menú de información del grupo (ver figura 4.4.9). Aparecerá un listado con los usuarios que no pertenecen al grupo aún (ver figura 4.4.11). Para añadirlos habrá que pulsar sobre el usuario y confirmar la adición (ver figura 4.4.12).

Ver participantes del grupo

Los participantes de un grupo aparecen en un listado en la zona inferior de la página de información del grupo junto con sus roles asignados (ver figura 4.9).

Eliminar participante del grupo

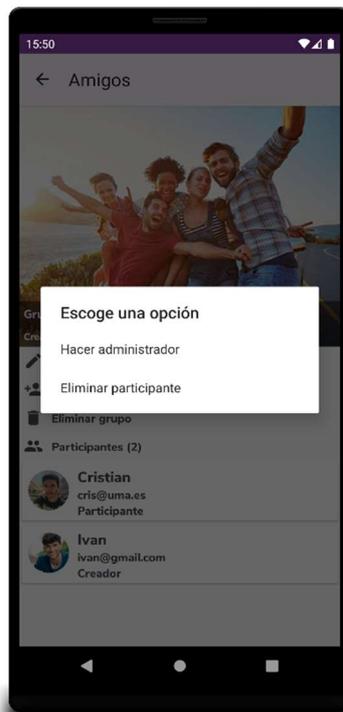


Figura 4.4.13: Administrar participante

Para eliminar un participante es necesario ser el creador o un administrador del grupo. Para ello habrá que pulsar sobre el usuario que se quiera eliminar de la lista suministrada en el apartado de información del grupo (ver figura 4.4.9) y después seleccionar “Eliminar participante” (ver figura 4.4.13).

Cambiar rol de un participante del grupo

El creador o un administrador del grupo puede cambiar el rol de los diferentes participantes, tanto ascendiendo como descendiendo (ver figura 4.4.13).

Eliminar grupo

Solo el creador del grupo tiene permitido eliminar el grupo. Para ello deberá pulsar “Eliminar grupo” en el panel de información del grupo (ver figura 4.4.9).

Salir del grupo

Cualquier usuario que pertenezca a un grupo tiene la libertad para salirse de este, excepto el creador. Para salirse del grupo tendrá que pulsar “Salir del grupo” en el panel de información del grupo (ver figura 4.4.9).

Buscar conversaciones/grupos

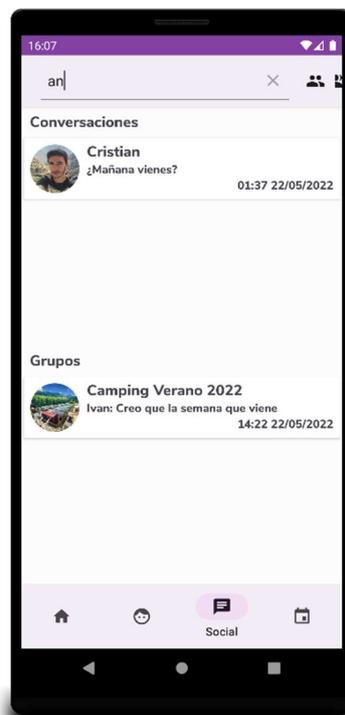


Figura 4.4.14: Buscar conversaciones y grupos

Pulsando la lupa perteneciente al menú superior de “Social” (ver figura 4.4.1), se podrán filtrar las conversaciones y grupos por nombre (ver figura 4.4.14).

Buscar usuarios

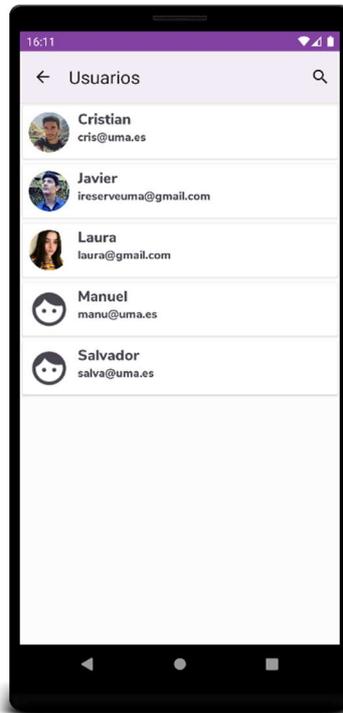


Figura 4.4.15: Usuarios

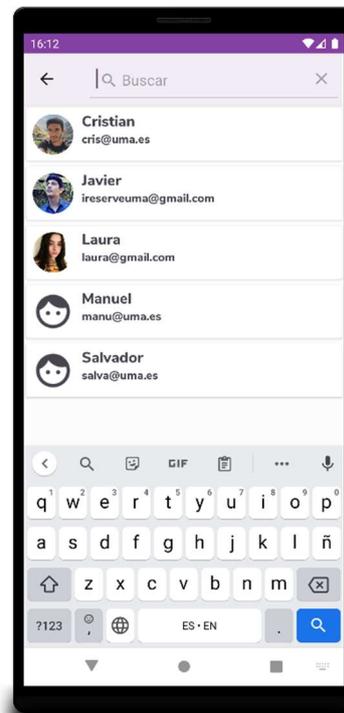


Figura 4.4.16: Buscar usuarios

Para ver el listado con todos los usuarios de la aplicación (ver figura 4.4.15) es necesario pulsar el botón central del menú superior de “Social” (ver figura 4.4.1). Además, también podrán filtrarse los usuarios por nombre o correo electrónico (ver figura 4.4.16).

Eventos



Figura 4.5.1: Eventos

El apartado de eventos es accesible a través del menú de navegación inferior, pulsando “Eventos” (ver figura 4.5.1).

Sigüientes eventos

Los próximos eventos aparecerán en la página principal de eventos, ordenados por fecha (ver figura 4.5.1).

Filtrar por fecha

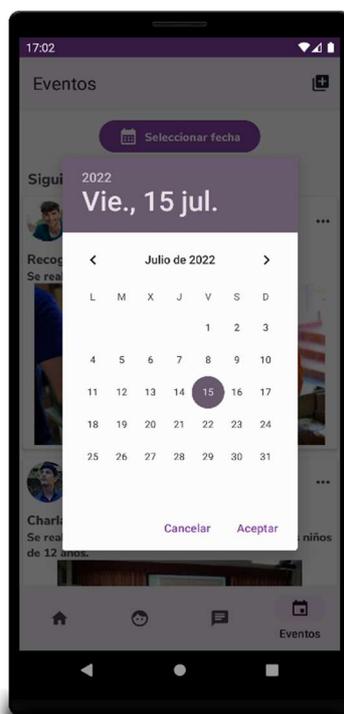


Figura 4.5.2: Selección de fecha para el filtro de eventos

Podrán filtrarse eventos por fecha concreta a través de un selector en la página principal, pulsando “Seleccionar fecha” (ver figura 4.5.1 y 5.2).

Crear evento



Figura 4.5.3: Crear evento

Solo los usuarios con rol de monitor podrán crear eventos. Para ello es necesario pulsar el botón del menú superior derecho (ver figura 4.5.1). Aparecerá un formulario donde será necesario rellenar título, descripción y fecha, siendo la imagen opcional (ver figura 4.5.3).

Editar evento



Figura 4.5.4: Acciones sobre un evento Figura 4.5.5: Editar evento

Un evento solo podrá ser editado por su creador. Para ello habrá que pulsar sobre “...” y posteriormente en “Editar” (ver figura 4.5.4). Aparecerá una página con un formulario para editar el evento ya creado. Para guardarlo habrá que pulsar en “Editar” (ver figura 4.5.5).

Eliminar evento



Figura 4.5.6: Borrar evento

Un evento solo podrá ser borrado por su creador. Para ello habrá que pulsar sobre “...” y posteriormente en “Borrar” (ver figura 4.5.4). Finalmente, será necesario confirmar la eliminación (ver figura 4.5.6).



UNIVERSIDAD
DE MÁLAGA

| uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA