



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

FRAB: Diseño e implementación de una APP en el ámbito de la fotografía

PROYECTO FINAL DE CARRERA
INGENIERÍA INFORMÁTICA

Autor: Daniel Andújar Lorenzo

Director: Sergio Sáez Barona

Co-Director: Vicente Gil Coscollá

Septiembre 2014

Resumen

Los dispositivos móviles se han convertido en una herramienta de comunicación entre las personas. Su avance y desarrollo ha permitido introducir sistemas operativos que son capaces de soportar y realizar tareas complejas como ya lo llevan haciendo los ordenadores personales. Permiten la instalación de mini-aplicaciones informáticas que están optimizadas para operar en estos terminales. Asimismo, los desarrolladores crean aplicaciones que cubren las necesidades latentes de los sistemas operativos que vienen preinstalados de fábrica.

Existe una gran diversidad de aplicaciones categorizadas en el ámbito de la fotografía pero ninguna de ellas tienen la funcionalidad de poder votar fotos ni que estas fotos pertenezcan a grupos privados accesibles únicamente mediante invitación.

En este documento se citarán, explicarán y comentarán las herramientas necesarias para la realización y desarrollo de la aplicación. Para ello, ha sido necesario el aprendizaje de tecnologías relacionadas con el desarrollo de aplicaciones móviles, aplicar conocimientos y protocolos de comunicación con el servidor, así como seguir unos patrones para almacenaje correcto de la información privada de los usuarios. Además este reporte documentado contiene imágenes que ayudarán a la comprensión del funcionamiento de la misma.

Palabras clave: app, subir fotos, dispositivo móvil, Android, foto, puntuación, grupo, scroll

Índice general

1. Introducción	4
1.1. Resumen	4
1.2. Contexto	4
1.3. Objetivos	5
1.4. Trabajo realizado	5
1.5. Estructura de la memoria	6
2. Especificación de requisitos	8
2.1. Introducción	8
2.1.1. Propósito	8
2.1.2. Alcance	8
2.1.3. Abreviaciones, definiciones y siglas	9
2.1.4. Visión global	13
2.2. Descripción general	13
2.2.1. Perspectiva de producto	13
2.2.2. Funciones del producto	15
2.2.3. Características de los usuarios	16
2.2.4. Restricciones	16
2.2.5. Supuestos y dependencias	17
2.2.6. Requisitos futuros	18
2.3. Requisitos Específicos	19
2.3.1. Requisitos de las interfaces externas	19
2.3.2. Requisitos funcionales	21
2.3.3. Requisitos no funcionales	24
3. Análisis	28
3.1. Introducción	28
3.2. Diagramas de casos de uso	28
3.2.1. Actores de la aplicación	29
3.2.2. Usuario no registrado	30
3.2.3. Usuario registrado	31

3.2.4.	Usuario administrador del grupo	55
3.2.5.	Usuario desactivado	55
3.3.	Diagrama de clases	58
3.3.1.	Usuarios de la aplicación	58
4.	Diseño	60
4.1.	Introducción	60
4.2.	Arquitectura del sistema	61
4.3.	Capa de persistencia	63
4.3.1.	SQLite	63
4.3.2.	Shared Preferences	66
4.3.3.	Disk LRU Cache	66
4.3.4.	Protocolo remoto	67
4.4.	Capa de presentación	68
4.5.	Capa de control	71
4.5.1.	Mapa de alcanzabilidad	71
4.5.2.	Manejador de Fragments para el Drawer	78
4.5.3.	Adaptador	79
4.5.4.	PagerAdapter	79
4.5.5.	FragmentManagerAdapter	79
5.	Detalles de implementación	81
5.1.	Introducción	81
5.2.	Tecnologías	81
5.2.1.	Java	81
5.2.2.	SDK Android	82
5.2.3.	Symfony	82
5.2.4.	XML	82
5.2.5.	API Google Cloud Messaging	82
5.2.6.	JSON	84
5.3.	Herramientas y entorno	84
5.3.1.	Base de datos	84
5.3.2.	Control de versiones	84
5.3.3.	Editor de imágenes	85
5.3.4.	Entorno de desarrollo	85
5.3.5.	Servidor Web	85
5.4.	Estructura de ficheros	85
5.4.1.	Directorio src	86
5.4.2.	Directorio gen	93
5.4.3.	Android X.X	93
5.4.4.	Android Private Libraries	93

5.4.5.	Directorio assets	93
5.4.6.	Directorio bin	93
5.4.7.	Directorio libs	94
5.4.8.	Directorio res	94
5.4.9.	AndroidManifest.xml	95
6.	Pruebas	96
6.1.	Pruebas de uso	96
6.1.1.	Registro	96
6.1.2.	Crear grupo	97
6.1.3.	Aceptar/cancelar invitaciones	98
6.1.4.	Visitar perfil del propio usuario	99
6.1.5.	Subir fotos	100
6.1.6.	Puntuar fotos	101
6.1.7.	Enviar comentario	102
6.1.8.	Cambiar contraseña	102
6.1.9.	Eliminar cuenta	103
6.1.10.	Modificar notificaciones	104
6.1.11.	Salir de un grupo	105
6.1.12.	Invitar un usuario a un grupo	106
6.1.13.	Eliminar usuarios de un grupo	106
6.1.14.	Modificar grupo	107
6.1.15.	Ver perfil usuario miembro de un grupo	108
6.2.	Resoluciones de pantalla	109
6.2.1.	Small screen	109
6.2.2.	Normal screen	110
7.	Conclusiones	111
7.1.	Técnicas	111
7.2.	Personales	112
7.3.	Agradecimientos	112
8.	Referencias bibliográficas	113

Capítulo 1

Introducción

1.1. Resumen

Este documento engloba toda la documentación necesaria del proyecto cuya finalidad ha sido desarrollar una aplicación móvil comprendida en el ámbito de la fotografía para terminales móviles con sistema operativo Android.

Para alcanzar este objetivo se ha llevado a cabo un análisis y un diseño de la aplicación a partir de una especificación de requisitos. La metodología empleada ha sido la iterativa y creciente [39] ya que el trabajo ha estado repartido y segmentado en pequeñas tareas las cuales han ido satisfaciéndose poco a poco, a medida que iban alcanzándose los objetivos para su posterior análisis e implantación en el sistema final. Si se daba el caso de encontrarse con algún error, se retrocedía al análisis de la porción de código en cuestión, se buscaba el problema, se arreglaba y se ejecutaba. Así sucesivamente hasta la versión final.

1.2. Contexto

El entorno en el que se desplegará la aplicación viene determinado por el crecimiento de las actuales "Redes Sociales" [50], definidas teóricamente como un grafo con nodos, en este caso personas, que se conectan mediante aristas estableciendo cierta propiedad entre ellos, en este caso, relaciones de amistad.

El objetivo principal de la aplicación será compartir fotos entre usuarios que pertenezcan a un mismo grupo con una temática que vendrá determinada por el creador. El acceso será mediante invitación y su contenido nunca será accesible por terceros, dotando así al sistema de privacidad.

Un usuario que acceda a un grupo desde una invitación, podrá invitar más gente, otorgando siempre la posibilidad al administrador¹ de echar a alguien fuera, si éste no cumple con las normas o si las fotos que sube no son las apropiadas. Un usuario podrá subir tantas fotos al grupo como quiera y serán puntuadas por el resto de miembros mediante un voto ^[49].

Según la interacción de los componentes, quedará reflejado aquéllos que son más activos y/o cuelguen mejor contenido. Los usuarios tendrán la capacidad de penalizar o favorecer la foto con su puntuación, elevando al propietario de la foto al podium o trasladándolo a la última posición del ránking.

1.3. Objetivos

El principal propósito de esta aplicación es llegar al mayor número de usuarios que se instalen la aplicación. Como segundo logro sería producir en los usuarios el interés de ser utilizada diariamente, ya sea accediendo a sus grupos para visualizar contenido y subiendo nuevas fotos. Como ya ocurren en otras aplicaciones, el usuario podrá valorar fotografías de otros usuarios para provocar en ellos la necesidad de ser usuarios bien valorados en vista a otros usuarios.

1.4. Trabajo realizado

El proyecto está formado por dos partes:

- **Front-End:** aplicación móvil, landing-page y panel de administración web².
- **Back-End:** servidor³.

El primer día de reunión con los empresarios se establecieron los objetivos, se dividió y segmentó el trabajo entre los dos miembros del equipo, así como se determinaron los objetivos a alcanzar a corto plazo y por último la fecha final de entrega del sistema.

El proyecto se ha desarrollado utilizando control de versiones para facilitar el avance en paralelo por los programadores del equipo, para evitar esperas en el avance y sobre todo para subsanar errores que pudiesen aparecer en el transcurso del desarrollo.

¹Creador del grupo.

²Parte delantera del proyecto que se encarga de interactuar y recoger los datos por parte del usuario

³Parte trasera que se ocupa del proceso y almacén de los datos que le llegan desde el Front-End

El código fuente se ha alojado en un repositorio privado con acceso restringido a terceras personas. Además el servidor se escogió y configuró a pocos días después de empezar a trabajar en el entorno que finalmente se encontrarían los usuarios reales.

La parte del proyecto que se describirá en detalle es la parte de **Front-End**, en concreto, la aplicación móvil. El usuario debería encontrarse con una interfaz gráfica visualmente atractiva e intuitiva, que tenga un manejo simple y sencillo y además que con una mirada a grosso modo pueda entender qué acciones puede realizar en todo momento.

Dada la importancia que recae sobre esta parte, se determinó que la mejor elección fuese contratar a un diseñador gráfico que diese un aspecto interesante y diferente a la app para diferenciarla de las ofertas ya existentes en el mercado.

En este proyecto se ha diseñado la estructura de la *app* y sus funcionalidades. En ciertas partes se ha reutilizado código, o se han hecho pequeñas modificaciones, para aprovechar el funcionamiento que se repetía en ciertas pantallas.

Se ha añadido una *api* de caché de imágenes que gestiona correctamente las imágenes descargadas desde el servidor y la *api* de notificaciones push ^[33] de Google que permite la comunicación asíncrona desde el servidor con cada cliente, para que cada usuario tenga la información actualizada al instante o deba conectarse para obtener datos nuevos.

1.5. Estructura de la memoria

La memoria está estructurada y dividida en siete capítulos organizados de la siguiente manera.

El primer capítulo contiene la introducción, en la cual se detalla el contenido del proyecto, el contexto sobre el cual se desenvolverá la *app*, los objetivos que se pretenden lograr cuando sea pública, el trabajo realizado durante el transcurso del proyecto y la estructura que se ha seguido en el desarrollo del mismo.

El segundo capítulo se basa en la especificación de requisitos software siguiendo el estándar IEEE Std 830-1998 ^[12]. Este estándar presenta el conjunto de características necesarias para la obtención de una buena especificación de requisitos que satisfagan y cumplan los objetivos y condiciones planteadas por el cliente.

El tercer capítulo se define el análisis de la aplicación móvil. En este apartado se

explicará el comportamiento de la *app* y las funcionalidades disponibles para un usuario. Se definirán los casos de uso mediante UML, el diagrama de clases y el diagrama de secuencia.

El cuarto capítulo contiene el diseño de la aplicación, detallando la arquitectura del sistema. Se apoya en la arquitectura por capas de Android, donde cada una de estas capas utiliza servicios que ofrecen las anteriores. Asimismo se mencionarán y detallarán las capas de persistencia, control y presentación como especificaciones más detalladas de implementación.

El quinto capítulo se precisan las tecnologías que han sido utilizadas, el entorno de desarrollo, sus herramientas y una breve explicación del sistema de ficheros de Android junto con una breve descripción de las clases principales de Java.

El sexto capítulo se desglosarán una selección de pruebas de uso con usuarios y grupos reales en el sistema. Además contendrá capturas de pantalla para dispositivos con tamaños y resoluciones de pantalla diferentes.

El séptimo capítulo se hará hincapié en las valoraciones personales y técnicas en la realización del proyecto y los agradecimientos personales.

Y para finalizar, en el último capítulo se listarán las referencias bibliográficas utilizadas tanto en el proyecto como en realización de esta memoria.

Capítulo 2

Especificación de requisitos

2.1. Introducción

2.1.1. Propósito

En la siguiente especificación de requisitos se establecerán las características y funcionalidades de la *app*. Todas ellas deberán cumplir con los requerimientos determinados por el usuario. La especificación de requisitos, como se ha enunciado anteriormente, se formalizará siguiendo el estándar IEEE Std 839-1998 ^[12].

En esta sección se determinarán las funcionalidades para los usuarios que participarán en el sistema, diferenciados como usuario logueado, usuario administrador de un grupo, usuario no registrado y usuario desactivado.

2.1.2. Alcance

La creación de esta aplicación emerge de la idea de tomar fotos cuyo contenido se categorice como "*peligroso*", con carácter nocivo, subido de todo o cuyo contenido sea comprometedor para terceras personas. El usuario que tome y suba una foto desde la propia aplicación se asegura que dicha foto no queda almacenada en su dispositivo y además, todas las imágenes que visualice desde cada uno de sus grupos no serán accesibles desde la aplicación Galería de Android. Después de esta primitiva idea floreció la funcionalidad que le otorgaría la característica diferenciadora con respecto al resto de aplicaciones existentes en el mercado, la posibilidad de puntuar dichas fotos para crear rankings, poder conocer qué foto ha tenido más impacto y diferenciar a usuarios que suben contenido atrayente para otros usuarios.

La *app* tiene un funcionamiento sencillo, el desplazamiento entre pantallas es fluido, tiene una estética diferente a lo visto actualmente y no está destinada a un público en

concreto, pudiendo ser utilizada por cualquier sector de personas sin deferencia de sexo o edad.

Antes del lanzamiento, previsto para Septiembre de este año en Google Play [17], la empresa ha registrado como marca **FrabApp** y ha contratado un dominio web (www.frabapp.com) cuyo servidor reside en los servidores de Amazon. Éste será el encargado de recibir las peticiones de conexión y obtención datos por parte de la aplicación móvil y de la landing page.

2.1.3. Abreviaciones, definiciones y siglas

En esta parte se engloban los términos que se utilizan en la especificación de requisitos:

1. **Administrador de un grupo** : Usuario que crea un grupo y automáticamente se le asigna el rol de administrador.
2. **Android** : Es un sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, relojes, televisores y automóviles [28].
3. **Api** (Application Programming Interface) : Conjunto de funciones y métodos que ofrece cierta biblioteca para ser utilizados por otro software como una capa de abstracción [29].
4. **App** : Programa o pieza de programa diseñado para cumplir un determinado objetivo. Generalmente se conoce este término como una aplicación que descargada un usuario en su dispositivo móvil [22].
5. **Autenticación** : Verificación del usuario con el sistema mediante dirección de correo electrónico o teléfono y contraseña [30].
6. **Base de datos** : Colección organizada de datos. Modelan aspectos de la realidad de tal manera que apoyan los procesos que requieren esta información [31].
7. **Caché** : memoria de acceso rápido de un ordenador, que guarda temporalmente las últimas informaciones procesadas. Se utiliza para reducir el tiempo de acceso a datos ubicados en la memoria principal que se utilizan con más frecuencia [32].
8. **Compartir** : Dar a conocer la aplicación utilizando terceras aplicaciones que permiten el paso de información sobre ellas.

9. **Contraseña** : Secuencia de dígitos alfanuméricos con cierta longitud y que cumple ciertas propiedades que permite el acceso al sistema mediante el proceso de autenticación.
10. **Correo electrónico** : Servicio de red que permite a los usuarios enviar y recibir mensajes mediante sistemas de comunicación electrónica. Identificador unívoco para el login.
11. **Drawer** : Panel que transiciona desde la parte izquierda o derecha de la aplicación y permite mostrar las principales opciones de navegación entre pantallas de la aplicación [8].
12. **ERS** : Especificación de requisitos software [16].
13. **Facebook** : Red social, <https://www.facebook.com/>.
14. **FAQ** (Frequently Asked Questions) : Preguntas frecuentes.
15. **Fotografía** : Técnica de obtener imágenes duraderas debidas a la acción de la luz [36].
16. **Frab** : Nombre de la aplicación.
17. **Google Play** : Tienda de aplicaciones móviles para dispositivos con sistema operativo Android.
18. **Grupo** : Conjunto privado de usuarios donde comparten fotos.
19. **Home** : Pantalla que muestra todos los grupos que un usuario es miembro junto con las acciones de compartir aplicación y crear un nuevo grupo.
20. **iOS** : Es un sistema operativo móvil de la empresa *Apple Inc* en el que su manipulación se realiza con gestos multitáctiles. Los elementos de control consisten en deslizadores, interruptores y botones. Actualmente ocupa la segunda posición en el ránking mundial en cuanto a cuota de mercado [38].
21. **Invitación** : Petición que se realiza a un usuario para que acceda a un grupo privado.
22. **Landing page** : Página de bienvenida a la que un usuario llega después de haber hecho clic en un enlace, un banner, un anuncio de texto, incluso en los propios resultados de búsqueda de cualquier buscador [25].
23. **Manifest** : Archivo necesario de las aplicaciones Android en el cual se presenta y enumera información esencial para el sistema operativo pueda ejecutarlas y otras aplicaciones puedan comunicarse entre ellas.

24. **Maquetación** : Componer gráficamente las pantallas de la aplicación, distribuyendo los elementos gráficos que forman parte de ella, así como los tipos de las letras.
25. **Login** : Proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario [41].
26. **Mi perfil** : Pantalla que muestra la información personal del usuario que está utilizando la aplicación. En ella aparecen dos pestañas. En la primera estarán todas las fotos y en la segunda las diez mejores fotos por orden de puntuación.
27. **Notificación** : Aviso que aparece en la parte superior de la pantalla cuando una aplicación tiene nuevos datos que obtener del servidor.
28. **Pestaña Ranking** : Idem que ^[29] pero ordenadas por puntuación media.
29. **Pestaña Todas** : Listado de todas las fotos que ha votado un usuario más todas sus fotos que hayan votado otros miembros.
30. **Php** : Es un lenguaje de programación de uso general de código del lado del servidor diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante [43].
31. **Puntuación foto** : Valoración media de una foto que ha sido puntuada y pertenece a un grupo.
32. **Puntuación usuario** : Valoración media de todas las fotos que un usuario posee y han sido puntuadas.
33. **Push** : Estilo de comunicaciones sobre Internet donde la petición de una transacción se origina en el servidor hacia los clientes [44].
34. **Ránking** : Lista ordenada de fotos por puntuación o lista ordenada de usuarios según la media de puntuación de todas sus fotos.
35. **Registro** : Inscribirse en el sistema como un usuario nuevo facilitando los datos necesarios.
36. **Root** : Cuenta de administrador. Es el nombre convencional de la cuenta de usuario que posee todos los derechos en todos los modos. El usuario root puede hacer muchas cosas que un usuario común no puede, tales como cambiar el dueño o los permisos de archivos.

37. **SDK** (Kit de desarrollo de software): Conjunto de herramientas de desarrollo de software que permiten al programador crear aplicaciones para un sistema concreto, paquetes software o sistemas operativos entre otros [47].
38. **Selfie** : Termino acuñado actualmente a la acción de tomar una persona una foto a sí mismo o a un pequeño grupo de personas utilizando una cámara fotográfica o teléfono móvil.
39. **SGDB** : Conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de añadir, borrar, modificar y analizar los datos [49].
40. **SmartPhone** : Teléfono con pantalla táctil que ofrece al usuario la conexión a internet, gestión de cuentas de correo electrónico, como la instalación de aplicaciones móviles como si fuese un pequeño ordenador portátil.
41. **SQL** : Lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas [52].
42. **Subir foto** : Copiar foto desde el móvil al servidor y registrarla en un grupo.
43. **Teléfono** : Sistema de comunicación que transmite la voz y el sonido a larga distancia por medios eléctricos o electromagnéticos. Identificador unívoco para el registro y para el login.
44. **Twitter** : Red social de microblogging, <https://twitter.com/>.
45. **Usuario administrador grupo** : Usuario común con la particularidad de administrar un grupo privado.
46. **Usuario logueado** : El usuario que utiliza los servicios de la aplicación para crear grupos privados, invitar amigos, subir y puntuar fotos.
47. **View**: Parte primordial en las aplicaciones Android ya que son elementos que confeccionan las interfaces de usuarios (botones, listas, imágenes, etc.).
48. **ViewPager** : Controlador de Android que permite al usuario desplazar a la izquierda y/o derecha la información que contenga la pantalla con un simple gesto.
49. **Voto** : Opinión de cada una de los usuarios para aprobar o rechazar una foto con un grado de aceptación que está comprendido de 0 a 5 con intervalos de 0.5 puntos.

2.1.4. Visión global

La especificación de requisitos se encuentra estructurada en dos apartados, descripción general y requisitos específicos.

En la descripción general se especifican las características de los tipos de usuarios existentes, las restricciones del sistema, las funciones de la aplicación y elementos que pueden influir a los requisitos.

En el apartado de requisitos específicos se detalla los requerimientos con un cierto nivel de detalle para que si se proporcionasen a los diseñadores, éstos pudiesen construir un sistema que cumpliera con estos requisitos y además que pudiesen construir un plan de pruebas para verificar la satisfacibilidad de éstos. Para alcanzar este objetivo es necesario definir los requisitos funcionales, los requisitos no funcionales, requisitos relacionados con la carga que sostiene el sistema, siguiendo las decisiones tomadas para el diseño de la *app* y las interfaces externas que actúan.

2.2. Descripción general

2.2.1. Perspectiva de producto

Análisis de mercado

En este subapartado se lleva a cabo una comparación entre las aplicaciones que más éxito y acogida han tenido dentro del ámbito de la fotografía.

Buscando por las tiendas de aplicaciones más importantes, se puede distinguir que existen dos grupos de competidores para esta aplicación. Exactamente no sería correcto definirlos como competidores en cuanto a uso, ya que no suponen una competencia directa en respecto a usabilidad, ni siquiera referido a las características del servicio que se quiere ofrecer, pero sí pueden facilitar un poco la tarea de diseño o búsqueda de usuarios. Revisando en profundidad estas aplicaciones, se puede llegar a patrocinadores cercanos a nuestro target y sobre todo estimar de antemano el usuario que podría estar interesado en este producto.

La clasificación de los dos grupos de competidores es amplia, pero está bastante clara: aplicaciones de ligue/contacto o aplicaciones referidas a la gestión de fotos y que pueden compartir esas fotos con otros usuarios.

La aplicación más conocida para conocer gente, sin lugar a duda es Badoo, <http://badoo.com/es/>, ofrece las mismas opciones que su portal web pero accesible desde

el smartphone. Permite la recepción directa de mensajes, contador actualizado del número total de visitas recibidas, notificaciones push de la actividad de tu perfil, intereses del resto de personas hacia el usuario (tales como atracciones mutuas o favoritos) y gente que se encuentra alrededor del usuario. El inconveniente es que cobra por saber al instante la opinión de otros usuarios respecto al usuario actual.

Otra *app* con gran índice de descargas es **Meetic** con más de 3.000.000 usuarios activos, <http://www.meetic.es/>, bastante similar a la mencionada anteriormente sólo que en esta el método de presentación es mediante "*flechazos*". Ofrece chat en directo con otros usuarios y otras opciones que sólo son accesibles mediante el servicio de pago.

Mencionando ahora las aplicaciones de fotografía, la líder indiscutible es **Instagram**, <http://instagram.com/>, con 200 millones de usuarios en todo el mundo, se caracteriza por ser una red social de fotografía la cual permite subir videos, fotos con o sin efectos, escribir comentarios a las fotos, añadir hastags, dar "*Me gusta*", etiquetar a amigos en fotos y seguir a usuarios famosos. Como desventajas se podría mencionar que su página web no permite la subida de contenido, los comentarios no se pueden gestionar y no se tiene control sobre quienes pueden ver qué fotos de un usuario.

Pinterest, <https://es.pinterest.com/>, es una red social basada prácticamente en imágenes cuyo acceso es mediante invitación de usuarios o desde el panel de su portada. Una vez dentro del sistema aparecen tableros de fotografías que están categorizados y usuarios a los que se está siguiendo. Para subir una foto hay que hacer un "*pin*" para colocarla en un tablón y como en las anteriores aplicaciones se podrá compartir el contenido en otras redes sociales.

Nuestro producto

El producto diseñado es una aplicación móvil categorizada como red social privada de fotografía.

Esta aplicación intenta distinguirse de la competencia añadiéndole pequeñas y nuevas funcionalidades tales como la privacidad de las fotografías que se suben al sistema, capacidad de valorar las fotos, interactuar con gente de todo el mundo en grupos públicos y acceder a grupos privados.

Se han recopilado conceptos de otras *app* tales como la entrada a los grupos privados debe ser mediante invitación de un miembro de ellos, almacenamiento en la nube de las fotografías, puntuación y recompensa por haber alcanzado el podium entre todas las imágenes del grupo.

La aplicación es totalmente gratuita, orientada a cualquier persona, sin distinción de sexo y para todas las edades.

2.2.2. Funciones del producto

Seguidamente se expone un listado de las funcionalidades de la aplicación.

- **Autenticación de usuario:** El usuario para poder acceder al sistema registrándose y autenticarse mediante su usuario¹ y contraseña.
- **Ayuda al usuario:** La aplicación dispone de un servicio de comentarios que son enviados directamente al administrador.
- **Compartir aplicación:** El usuario podrá compartir de la aplicación mediante las redes sociales y con aplicaciones que permitan el paso de información.
- **Gestionar un grupo:** El usuario podrá modificar el título y la imagen del grupo, invitar a nuevos usuarios, salir del grupo y el administrador tendrá la capacidad de echar a usuarios.
- **Gestionar datos de la aplicación:** Se podrá acceder página FAQ^[14], cancelar la cuenta y modificar las notificaciones.
- **Gestionar perfil de usuario:** El usuario podrá modificar su imagen de perfil, nombre, descripción y sexo, además de modificar su contraseña y de administrar las notificaciones .
- **Puntuar fotografías:** El usuario valorará fotografías de otros usuarios que estén en el mismo grupo.
- **Registrar un usuario:** El nuevo usuario podrá registrarse en el sistema introduciendo pocos datos en un único paso.
- **Seleccionar un grupo:** El usuario podrá visualizar las fotos nuevas que tenga por votar, así como el ranking de las fotos que ya hayan sido votadas.
- **Subir una foto:** El usuario podrá subir fotos sin límite a los grupos que desee, eligiendo la imagen desde la galería o tomando una foto al instante.
- **Visualizar fotografías:** El usuario tendrá acceso completo al contenido de los grupos públicos, acceso restringido al contenido de aquellos usuarios que no tenga grupos en común y acceso normal a los usuarios que pertenezcan a los mismos grupos.

¹Entendiendo usuario como número de teléfono o correo electrónico.

2.2.3. Características de los usuarios

En la aplicación existen diferentes tipos de usuarios dependiendo de la tarea que desempeñen:

- **Usuario administrador del sistema:** usuario especial que gestiona los problemas del sistema, administra la página web, responde a los comentarios, mantiene actualizada la página FAQ^[14], controla la base de datos y verifica el correcto funcionamiento del servidor. Debe poseer conocimientos informáticos altos, capacidad de resolución de problemas y dominar los aspectos técnicos de la aplicación móvil como del servidor.
- **Usuario no registrado:** usuario que intenta acceder a la aplicación sin estar registrado. Este usuario no puede llevar a cabo ninguna acción salvo registrarse en el sistema.
- **Usuario registrado:** usuario que tiene acceso al contenido privado propio de la aplicación tras el registro en el sistema. Este usuario puede administrar su perfil. Puede subir, ver y puntuar fotos. Puede crear e invitar a sus amigos a un grupo, salir de él y ponerse en contacto con el administrador del sistema. La aplicación no necesita conocimientos expertos debido a su sencillez.
- **Usuario administrador del grupo:** usuario que posee la misma funcionalidad que un usuario registrado con la funcionalidad adicional de poder eliminar a usuarios del grupo que haya creado él.
- **Usuario desactivado:** usuario registrado que tomó la decisión de salir del sistema.

Una persona que utilice la *app* puede ejercer varios roles en a la vez. El **usuario administrador del sistema** es único y no se explicará su funcionalidad en esta memoria.

2.2.4. Restricciones

La aplicación debe ser sencilla y de manejo simple.

Los usuarios que utilicen la aplicación deben poder acceder al sistema en cualquier momento de cualquier día.

Las limitaciones hardware vienen determinadas por los terminales que no tengan instalado el sistema operativo Android, intentando abarcar el mayor número de dispositivos móviles en el mercado. La mínima API^[3] permitida para ejecutar la aplicación es la 15, es decir, aquellos terminales que tengan instalado el sistema *IceCream Sandwich*

y posteriores.

El servidor debe poder atender sin errores a un número indeterminado de usuarios que solicitan servicio simultáneamente desde la aplicación.

Las imágenes que se visualicen dentro de la aplicación no deben ser accesibles desde un usuario sin un terminal con el root activado ^[36].

Los datos del usuario no deben ser visibles. Las contraseñas y los datos se envían encriptados al servidor con un protocolo de cifrado.

La aplicación debe cumplir con la política de privacidad y protección de datos para los datos personales de los usuarios registrados en el sistema, siguiendo la normativa española [18, 24].

El lenguaje del servidor debe ser PHP^[30], utilizando un SGBD^[39] con soporte para SQL^[41].

2.2.5. Supuestos y dependencias

La aplicación utiliza la tecnología push, concretamente el servicio de mensajería en la nube **Google Cloud Messaging** (ver sección 5.2.5), en el cual el servidor envía mensajes de aviso a los dispositivos para que se conecten a él o para enviarles directamente en el mensaje la información que necesitan, de tal manera que los clientes no tengan que estar continuamente consultando o comunicándose con el servidor para saber si dispone de nueva información.

De esta manera, los usuarios de un grupo pueden estar al corriente de la actividad del resto de usuarios, ya que para ciertas acciones que realice un usuario, la información que visualicen el resto puede no estar actualizada. Si **GCM** falla, la información común que es accesible por un grupo de usuarios no sería la misma que para el resto. Por tanto, la aplicación está ligada fuertemente con este servicio y depende mucho de su funcionamiento. Si éste falla, los usuarios no tendrían notificaciones y no estarían al corriente de las interacciones del resto de usuarios con el sistema.

La aplicación depende del servidor contratado en Amazon. Si éste tiene una baja respuesta, se vería reflejado en el aumento del tiempo de espera para la obtención de datos. Por otro lado, podría suceder que dejase de ofrecer servicio durante algún intervalo de tiempo, bloqueando el sistema e imposibilitando el acceso a los usuarios.

2.2.6. Requisitos futuros

Este subapartado contiene mejoras y características que se podrían implementar en las futuras versiones de la aplicación.

Sin lugar a duda, la mejora más importante sería conectar la aplicación a las redes sociales (Facebook, Twitter o Google+), de tal manera que un nuevo usuario pudiese registrarse y acceder al sistema desde su cuenta. También ofrecer la posibilidad de poder compartir sus fotos en sus perfiles propios.

Implementar la misma versión de la aplicación pero para dispositivos con sistema operativo iOS.

Crear grupos públicos cuyos usuarios puedan acceder al contenido sin previa invitación de un administrador.

Comentarios en las fotos para debatir su contenido o para explicar porque un usuario introdujo una puntuación concreta.

Etiquetar a usuarios en fotos, denunciar a un usuario por el contenido de su foto y ofrecer la posibilidad de eliminar fotografías.

Implementar envío directo de mensajes entre usuarios, sean o no amigos.

Permitir a un usuario ver fotos propias de un grupo en concreto, sin poder visualizar las del resto de usuarios del grupo.

Opción en la pantalla principal de búsqueda de grupos y de usuarios por su nombre.

Extender la funcionalidad también a la subida de vídeos de corta duración.

Crear una versión adaptada para web, cuya funcionalidad sea lo más semejante posible a la de la aplicación, sin cambiar de interfaz gráfica y específica para navegadores web.

Una posible forma de monetización sería permitir a las empresas, previo pago, crear grupos patrocinados en los cuales introducir productos para anunciarlos durante un período de tiempo o añadir publicidad en forma de banners.

2.3. Requisitos Específicos

Se especifican los requisitos que se necesitan para el desarrollo de la aplicación. Todos los requisitos que figuran, indican alguna demanda real del sistema y pueden modificarse.

2.3.1. Requisitos de las interfaces externas

Requisitos de usuario

La interfaz de usuario debe ser básica y cómoda, debe requerirse poco esfuerzo para entender su funcionamiento y debe ser original. Deberá ser sutil para que cualquier usuario sin conocimiento alguno pueda acceder a cualquier parte de la aplicación sin perderse por el camino.

Para obtener el requisito de originalidad, se ha recurrido a un diseñador gráfico que ha realizado la labor de dibujo, utilizando el editor gráfico rasterizado Adobe Photoshop. Dotando los ficheros en formato psd² para su posible modificación posterior.

El usuario debe obtener de la aplicación un funcionamiento fluido y rápido para no impacientarse cuando la use, porque puede salir de ella si el tiempo de respuesta del servidor es lento.

La página de inicio de la aplicación tiene que ser sencilla, con un golpe de vista el usuario sabe cuantos grupos y fotos nuevas tiene. Habrá una barra fija en la parte inferior que será accesible en todas las pantallas cuya función será de subir fotos, situarse en Home^[19] o en Mi perfil^[26].

Previa maquetación^[24], debe realizarse un estudio previo de los tamaños de pantalla de los terminales más utilizados.

²Formato estándar de Photoshop con soporte de capas.

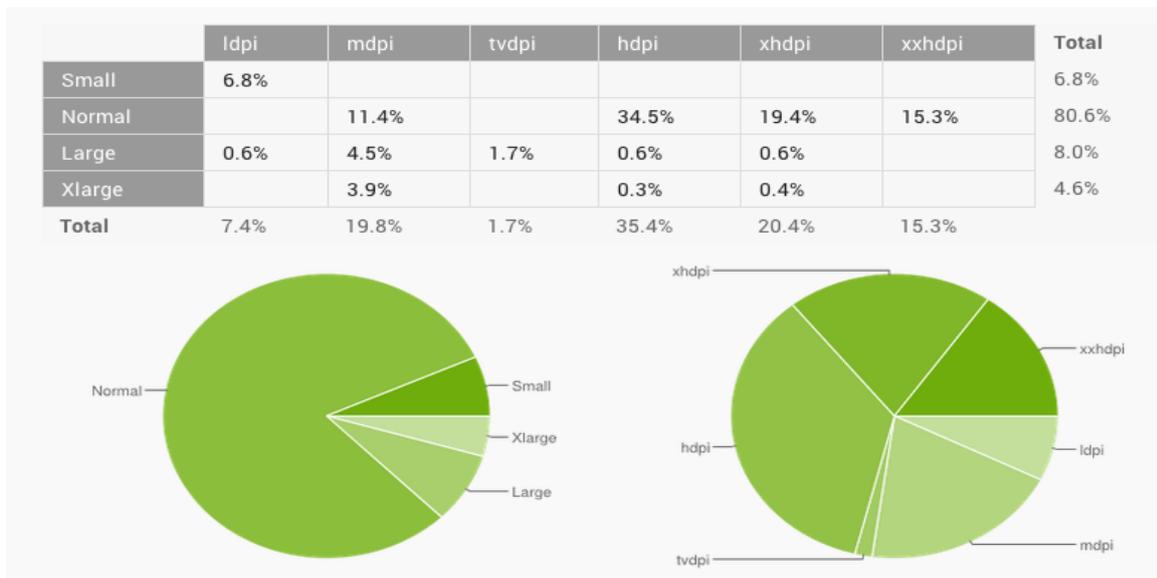


Figura 2.1: Datos recogidos durante 7 días a finales de Julio 2014. Cualquier configuración de pantalla con una cuota menor de 0.1 % no se incluye [7].

Se puede observar como el tamaño predominante es el Normal, seguido del largo y pequeño, dejando en menor importancia el extra largo (utilizado por los tablets).

Requisitos de hardware

No existen requisitos de hardware para esta aplicación ya que está preparada para almacenar hasta 10MB de imágenes según se vayan necesitando, además la carga y visualización depende de la interacción que el usuario tenga en cada momento, liberando memoria cuando sea necesario y cargando la información desde la caché cuando se precisen imágenes que ya hayan sido vistas.

Requisitos de software

Existe únicamente un requisito software referente a la compatibilidad de la aplicación respecto a obtener el mayor número de dispositivos Android que hay en el mercado. Después de investigar y analizar los datos, se determinó que la aplicación tendría que ser compatible como mínimo con la **api 15**.

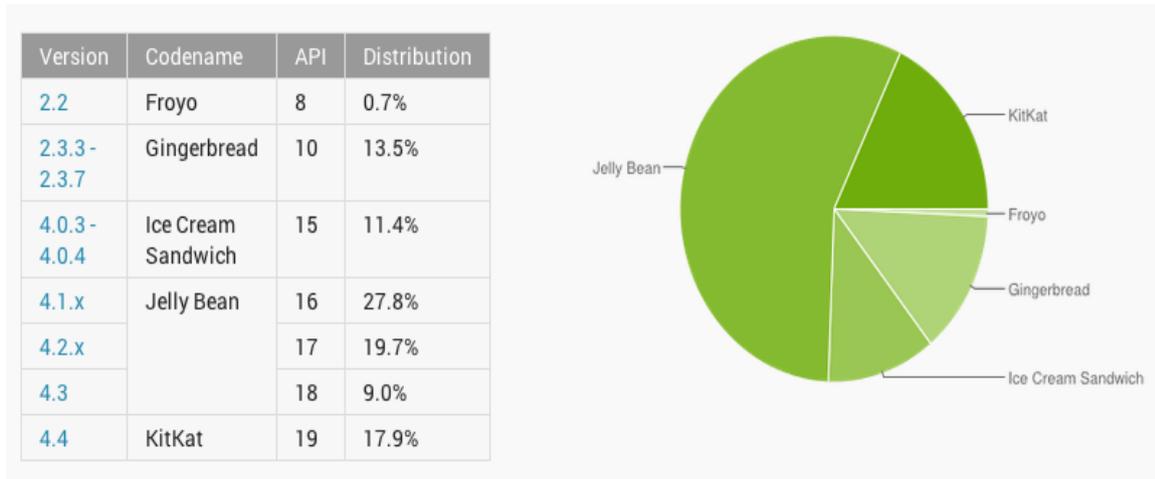


Figura 2.2: Datos recogidos durante 7 días a finales de Julio 2014. Cualquier versión con una cuota menor de 0.1 % no se incluye [7].

Estudiando la figura 2.2, la aplicación podría estar presente actualmente en el 85.8 % de los dispositivos, ya que la mínima versión del SDK [37] en el Manifest[23] está establecida en la 15.

Requisitos de comunicaciones

Al tratarse de una aplicación móvil, debe poder conectarse al sistema mediante la tecnología GPRS, EDGE, 3G, HSDPA, HSUPA, 4G o Wifi. Si el usuario tiene activada alguna de ellas, será suficiente.

Se debe utilizar el protocolo HTTPS para la transferencia segura de información ya que se estará tratando con datos personales, contraseñas y fotografías privadas.

2.3.2. Requisitos funcionales

Los requisitos funcionales se disponen siguiendo el modelo de usuario. Estos requisitos explican las funcionalidades o casos de uso que cada usuario puede llevar a cabo [45].

Una funcionalidad viene documentada en un breve resumen (descripción de la funcionalidad), un actor (tipo de usuario), una precondition, una postcondición, un proceso de acciones que suceden secuencialmente y un listado de excepciones (comportamientos erróneos o errores de ejecución). Estas funcionalidades se mostrarán en diagramas de casos de uso.

Acto seguido se resume cada uno de los requisitos funcionales del sistema agrupados por tipo de usuario:

Usuario no registrado

- **Consultar términos y condiciones del servicio:** El usuario en el proceso de registro, antes de aceptar las condiciones, podrá leer las cláusulas del contrato a las que estará sujeto.
- **Registro:** El usuario que quiera acceder al sistema por primera vez deberá registrarse en el sistema.

Usuario registrado

- **Aceptar/cancelar invitación:** El usuario podrá aceptar/cancelar la invitación para pertenecer a un grupo.
- **Activar/desactivar la vibración de las notificaciones:** Las notificaciones pueden tener vibración y esta característica puede ser modificada por el usuario.
- **Actualizar lista de amigos:** El usuario podrá actualizar la lista de usuarios para comprobar si existen nuevos usuarios con la aplicación instalada.
- **Cambiar contraseña de usuario:** El usuario podrá cambiar su contraseña de inicio de sesión.
- **Cambiar el tono de notificación:** El sistema dispone de notificaciones y éstas pueden contener un tono de notificación personalizado. El usuario podrá personalizarlas a su gusto.
- **Cambiar foto de perfil:** El usuario puede cambiar su foto de perfil. El resto de usuarios verán ésta foto.
- **Cerrar sesión:** El usuario cierra la sesión manualmente y los datos almacenados en la aplicación se eliminan.
- **Compartir app:** El usuario podrá compartir el enlace de descarga de la aplicación utilizando otras aplicaciones.
- **Consultar datos del perfil:** El usuario puede visualizar sus datos personales.
- **Consultar grupos:** El usuario podrá visualizar todos los grupos de los cuales es miembro.

- **Consultar invitaciones:** El usuario podrá visualizar todos las invitaciones a los grupos que tiene pendientes de responder.
- **Consultar preguntas frecuentes:** Un usuario puede tener alguna duda sobre el funcionamiento de la aplicación. Para ello se ha dispuesto de una página web que recoge las preguntas más frecuentes que se puedan hacer los usuarios.
- **Contactar con el administrador del sistema:** Un usuario podrá enviar mensajes al administrador del sistema para resolver dudas y problemas o para solicitar información.
- **Crear nuevo grupo:** El usuario puede crear un nuevo grupo donde compartirá imágenes con otros usuarios.
- **Editor datos del perfil:** El usuario puede cambiar sus datos personales tantas veces como desee.
- **Eliminar cuenta:** El usuario puede eliminar su cuenta del sistema en cualquier momento.
- **Invitar usuario:** Invitar a un amigo a un grupo existente.
- **Login:** El usuario se autentificará en el sistema mediante su usuario y contraseña.
- **Modificar grupo:** El usuario puede editar el nombre del grupo y la imagen de un grupo.
- **Obtener los usuarios de un grupo:** El usuario podrá saber que usuarios son miembros de un grupo al cual él pertenezca.
- **Recuperar contraseña:** Un usuario que no recuerde su contraseña puede solicitar una nueva.
- **Salir del grupo:** El usuario podrá salirse de un grupo cuando desee.
- **Subir una foto:** El usuario puede subir tantas fotos como quiera a cada uno de los grupos que es miembro.
- **Visitar el perfil de un amigo:** Visitar perfil de un amigo donde aparecerán todas sus fotos y el ranking de las mejores fotos.
- **Visitar el perfil de usuario:** Visitar el perfil del usuario logueado. Se podrán ver todas las imágenes subidas al sistema y las mejores 10 imágenes puntuadas por otros usuarios.

- **Visitar el perfil de un usuario miembro de un grupo:** Visitar el perfil de usuario de un miembro del grupo.
- **Visualizar el ránking de un grupo:** El usuario podrá ver las fotografías de un grupo ordenadas por su puntuación media.
- **Visualizar las fotos nuevas de un grupo:** El usuario podrá ver las fotos nuevas que los otros miembros del grupo han subido al grupo.
- **Visualizar listado de amigos:** El usuario podrá visualizar todos los usuarios que son amigos.
- **Visualizar ránking de fotos de un usuario:** Un usuario podrá visualizar el ránking de fotos de otros usuarios. Solo serán visibles aquellas fotografías que estén en grupos en común.
- **Visualizar todas las fotos de un usuario:** Un usuario podrá visualizar todas las fotos de otros usuarios. Solo serán visibles aquellas fotografías que estén en grupos en común.
- **Votar una foto:** El usuario que tenga fotos nuevas de un grupo podrá puntuarlas de 0 a 5.

Usuario administrador de un grupo

- **Eliminar usuarios de un grupo:** El administrador de un grupo tiene la capacidad de eliminar a usuarios que sean miembros de ese grupo.

Usuario desactivado

- **Reactivar la cuenta:** Un usuario puede retornar al sistema tras un tiempo después de eliminar su cuenta.

2.3.3. Requisitos no funcionales

Un requisito no funcional en la ingeniería de software es un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. No describen información a guardar, ni funciones a realizar [46].

Rendimiento y escalabilidad

En este subapartado se especificará los requisitos vinculados con la carga que el sistema debe soportar. Se quiere lograr a medio-largo plazo unos buenos resultados en cuanto a número de usuarios inscritos en el sistema. Por tanto, a medida que el número de usuario que se registren vaya aumentando, el servidor deberá mejorarse para que el servicio que ofrece sea capaz de dar respuesta a la fuerte demanda, estar activo ante cualquier pico de carga. También se tendrá que controlar la capacidad de información del mismo para que pueda almacenar mayor cantidad de imágenes y datos.

Actualmente el servicio dispone de una capacidad de 30GB, 1GB de memoria y una sola base de datos instalada.

Restricciones de diseño

En esta subapartado se desglosan los elementos que limitan las opciones del desarrollador. Las restricciones de diseño representan decisiones que se han tomado y que se tienen que cumplir.

Para el desarrollo de la aplicación móvil se ha utilizado el lenguaje de programación Java (ver sección 5.2.1) ya que es el lenguaje con el que se programa para Android. Para el control del proyecto se debe que utilizar un control de versiones que permite el seguimiento del trabajo y el desarrollo en paralelo por los componentes del equipo.

La aplicación debe realizar el menor número de consultas al servidor, pudiendo obtener la mayor cantidad de datos al inicio de la misma, ayudándose de una base de datos, de las notificaciones push enviadas por parte del servidor y una caché de imágenes que almacene las últimas fotografías descargadas.

Seguridad

Al tratarse de una aplicación que comparte fotografías privadas en grupos que se accede mediante invitación, la seguridad es un característica trascendental de la *app*.

El sistema autentificará cada vez a un usuario cuando desee acceder al sistema, mediante su usuario y su contraseña, permitiendo el inicio de sesión automático a menos que se indique lo contrario. Ningún usuario podrá ver ningún grupo y su contenido si no tiene permiso para ello. El usuario que acceda al perfil de otro, sólo podrá acceder a las fotos que los dos tengan en común. La gestión de permisos estará gestionada por el

servidor y éste será quien compruebe las acciones que pueda realizar en cada momento un usuario.

Fiabilidad

La base de datos del servidor deberá realizar copias y luego almacenarlas, de tal manera que puedan restaurarse los datos ante problemas, posibles ataques o fallos del sistema. Las imágenes subidas por los usuarios no pueden ser accedidas por terceras personas, además se deberán realizar copias de seguridad en otro servidor.

Estas tareas de seguridad serán llevadas a cabo periódica y automáticamente en el servidor.

Disponibilidad

La aplicación estará disponible para los usuarios en cualquier hora del día y todos los días del año, siempre y cuando, no existan fallos en el sistema.

Mantenibilidad

Para la gestión de la aplicación móvil no se requerirá a ninguna persona, dejando la parte de supervisar el contenido de los grupos y la gestión de comentarios, al administrador del sistema. El administrador estará en posesión de una herramienta gestión, como será un Panel de administración web en el que podrá asistir a los problemas de los usuarios.

Las tareas de mantenimiento y conservación del sistema, como se ha mencionado anteriormente, las realizará el propio servidor.

Ante los posibles errores que se produzcan en la *app*, habrá un programador para solucionarlos, para ello la aplicación deberá ser estable en la fecha del lanzamiento oficial.

Portabilidad

La aplicación para la primera versión, debe ser completa y estar acabada para el terminales Android, diseñando el sistema de tal manera que para la futura salida a la plataforma iOS ^[20] no se deba modificar ni código ni estructura del sistema.

La tecnología utilizada para el sistema y que se ejecuta en un hosting, tiene que poder ser portada a otro que posea las mismas características.

Eficiencia

La aplicación está categorizada como una aplicación de fotografía. El principal contenido con el que tratará serán imágenes. La subida de imágenes debe ser veloz y el mismo terminal las tendrá que comprimir a una relación óptima entre calidad-tamaño, para que posteriormente, cuando otro usuario requiera ese fichero, el tiempo de respuesta para la descarga y procesamiento sea el menor posible. Si además, la *app* se respalda con una caché ^[7] que almacena en disco las imágenes descargadas, el tiempo de carga se reduce considerablemente.

El usuario no podrá interactuar con el sistema y tener toda la funcionalidad hasta que los datos que se requieran hayan sido obtenidos del servidor y estén cargados. Mientras tanto el sistema quedará bloqueado ante cualquier interacción del usuario.

Los datos que se soliciten al servidor deberán ser exactamente los necesarios para la siguiente pantalla con la que interactuar, así se evitará realizar peticiones constantes e innecesarias, produciendo una posible saturación del servidor si existen cuotas altas de usuarios.

Capítulo 3

Análisis

3.1. Introducción

Este capítulo describe la parte de análisis del proyecto. Se analiza la aplicación desarrollada, se documenta su estructura y las funcionalidad de las que dispone mediante diagramas que fijan el funcionamiento y las peculiaridades del sistema.

Para llevar a término el análisis se utilizará UML [20]. UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. El uso que se le concede es el de visualizar, especificar, construir y documentar un sistema. No puede compararse con la programación estructurada, no es programación, ya que solo se diagrama la realidad de una utilización en un requerimiento. UML cuenta con varios tipos de diagramas, entre ellos destacan los diagramas de estructura, de comportamiento y de interacción.

3.2. Diagramas de casos de uso

En esta sección se detalla la aplicación utilizando diagramas de casos de uso que son equivalentes a los requisitos funcionales que se han especificado en la subsección 2.3.2 del capítulo de especificación de requisitos.

Un diagrama de casos de uso es un tipo de diagrama de comportamiento UML mejorado. Aportan una vista general simple de un caso o conjunto de casos de uso. Los diagramas de caso de uso suelen ser confundidos por los casos de uso, ya que están internamente relacionados, mientras que los segundos muestran más de detalle que los primeros.

Un caso de uso es una descripción de los pasos o actividades que deberán realizarse

para llevar a cabo algún procesos. Los personajes o entidades que participan en un caso de uso se denominan actores. En la aplicación, un caso de uso [33] es una sucesión de interacciones que se despliegan entre los actores y el sistema como réplica a un acontecimiento que da comienzo un actor hacia el sistema.

3.2.1. Actores de la aplicación

En la figura 3.1 se muestra el diagrama relativo a los actores. Existen 4 actores que operan en el sistema. Existe una relación de generalización/especialización. El actor *usuario administrador de un grupo* es una especialización del actor general *usuario registrado*. Además una persona puede ejercer varios roles diferentes como se ha descrito en la subsección 2.2.3 donde se referenciaba a los usuarios del sistema.

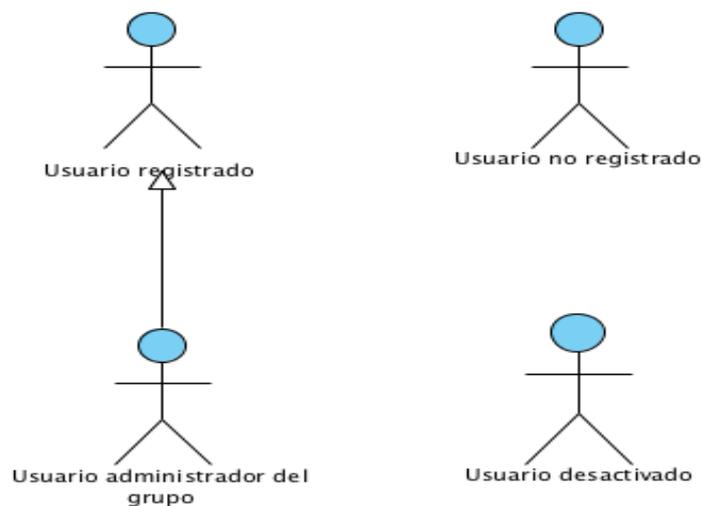


Figura 3.1: Actores

3.2.2. Usuario no registrado

En la figura 3.2 se muestra los dos caso de uso que el *Usuario no registrado* puede realizar: *Registro* en el sistema y *Consultar los términos y condiciones del contrato*.

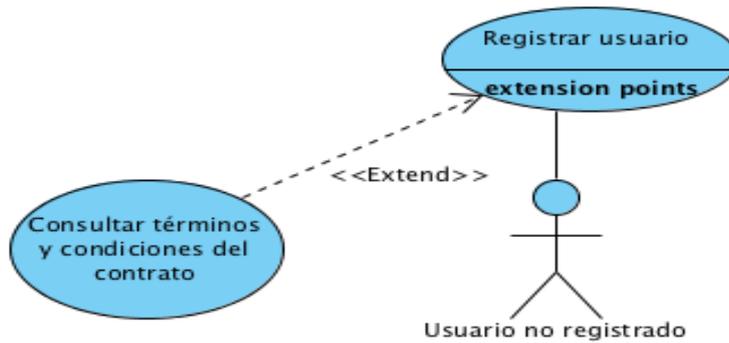


Figura 3.2: Usuario no registrado

Nombre	Consultar términos y condiciones del servicio.
Casos de Uso relacionados	Registrar usuario.
Precondición	Usuario no registrado.
Postcondición	Pantalla con los términos y condiciones del contrato.
Proceso	
Usuario	Sistema
1- El usuario pulsa en el texto de los <i>Términos y condiciones del servicio</i> cuando está registrándose.	
	2- El sistema redirige al usuario a la página web de los términos y condiciones del servicio.

Nombre	Registrar usuario.
Casos de Uso relacionados	-
Precondición	Usuario no registrado.
Postcondición	Usuario registrado.
Proceso	
Usuario	Sistema
1- El nuevo usuario pulsa el botón <i>Regístrate</i> de la pantalla de inicio.	
2- El usuario introduce el nombre de usuario, teléfono, correo electrónico, la contraseña, el sexo y acepta los términos y condiciones del servicio.	
	3- El sistema valida los datos del usuario.
	4- El sistema registra el nuevo usuario en el servidor.
	5- El sistema envía un correo al usuario.
Excepciones	
· La contraseña no puede ser la misma que el nombre.	
· Contraseña debe contener al menos 8 caracteres.	
· Usuario existente.	
· Correo en uso.	
· Faltan campos por completar.	

3.2.3. Usuario registrado

En la siguiente subsección se exponen los diagramas de casos de uso para el principal actor del sistema. Estarán segmentados en cuatro grupos que engloban las características trascendentales. *Gestión de usuario*, *Gestión de grupos*, *Gestión de los datos de la aplicación* y finalmente *Gestión de los perfiles de usuario*.

Gestión de usuario

El usuario podrá administrar y modificar su perfil privado tantas veces como desee. Las principales características con las que se encontrará son: *Login*, *Cerrar sesión*, *Consultar los datos del perfil*, *Editar datos del perfil*, *Cambiar foto de perfil*, *Cambiar contraseña de acceso*, *Eliminar cuenta*, *Cambiar tono de notificación* y *Activar/desactivar la vibración de las notificaciones*.

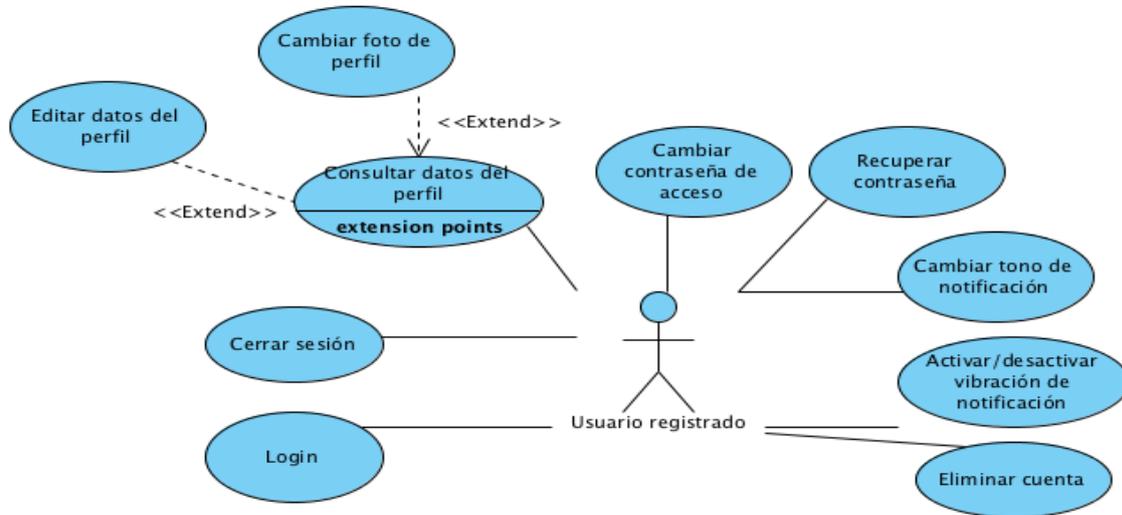


Figura 3.3: Gestión de usuario

Nombre	Login de Usuario.
Casos de Uso relacionados	-
Precondición	Usuario registrado.
Postcondición	Usuario logueado en el sistema.
Proceso	
Usuario	Sistema
1- El usuario introduce su correo electrónico o teléfono y su contraseña.	
	2- El sistema valida que los datos son correctos.
	3- El sistema asigna un token al usuario para las futuras peticiones al servidor y le devuelve los nuevos datos.
	4- El sistema redirige al usuario a la <i>Home</i> .
Excepciones	
· Servidor no disponible.	
· Usuario o contraseña erróneos.	
· Usuario con cuenta cancelada.	
· Usuario no existente.	

Nombre	Cerrar sesión.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario selecciona la opción <i>Cerrar sesión</i> en el <i>Drawer</i> .	
	2- El sistema muestra un diálogo de confirmación.
3- El usuario acepta el cierre de sesión.	
	4- El sistema borra el token de acceso, la base de datos y la caché de imágenes.
	5- El sistema redirige al usuario a la pantalla de <i>Login</i> .

Nombre	Consultar datos del perfil.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> .	
2- El usuario pulsa en la imagen de perfil.	
	3- El sistema abre una pantalla y muestra los datos personales del usuario.

Nombre	Editar datos del perfil.
Casos de Uso relacionados	Consultar datos del perfil.
Precondición	Usuario logueado.
Postcondición	Los datos del usuario han cambiado.
Proceso	
Usuario	Sistema
1- El usuario accede a su perfil desde el <i>Drawer</i> .	
2- El usuario pulsa en su imagen de perfil.	
3- El usuario visualiza los datos.	
4- El usuario modifica algún dato (nombre, descripción o sexo).	
5- El usuario pulsa en el menú la opción <i>ACTUALIZAR</i> .	
	6- El sistema valida que los datos sean correctos.
	7- El sistema envía la información al servidor.
	8- El sistema redirige al usuario a la pantalla <i>Home</i> .
Excepciones	
· Algún campo se encuentra vacío.	

Nombre	Cambiar foto de perfil.
Casos de Uso relacionados	Consultar datos del perfil.
Precondición	Usuario logueado.
Postcondición	Foto de perfil actualizada.
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> .	
2- El usuario pulsa en su imagen de perfil situada en la parte posterior del <i>Drawer</i> .	
	3- El sistema muestra la pantalla con la información del perfil del usuario.
4- El usuario pulsa en su imagen de perfil.	
	5- El sistema muestra la galería de imágenes.
6- El usuario selecciona una foto.	
7- El usuario corta la foto a una dimensión preestablecida.	
	8- El sistema automáticamente sube la foto al servidor.
	9- El sistema redirige al usuario a la pantalla <i>Home</i> .

Nombre	Cambiar contraseña de acceso.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	Contraseña modificada.
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa en <i>Ajustes</i> .	
2- El usuario pulsa en <i>Info de cuenta</i> .	
3- El usuario pulsa en <i>Privacidad</i> .	
	4- El sistema muestra un formulario.
5- El usuario introduce la contraseña actual.	
6- El usuario introduce la nueva contraseña.	
7- El usuario repite la nueva contraseña.	
	8- El sistema valida los datos.
	9- El sistema envía los datos actualizados del usuario al servidor.
	10- El servidor envía la nueva contraseña al usuario mediante correo electrónico.
Excepciones	
· La nueva contraseña contiene menos de 8 caracteres.	
· La nueva contraseña no coincide. Escríbala de nuevo.	
· Algún campo está vacío.	

Nombre	Recuperar contraseña.
Precondición	-
Postcondición	Usuario con contraseña nueva.
Proceso	
Usuario	Sistema
1- El usuario en la pantalla de inicio pulsa en <i>Recordar contraseña</i> .	
	2- El sistema abre un diálogo y solicita el número de teléfono o correo electrónico al usuario.
3- El usuario introduce el dato.	
4- El usuario pulsa en el botón <i>Solicitar</i> .	
	5- El sistema se comunica con el servidor para la recuperación de la contraseña del usuario.
	6- El servidor envía la nueva contraseña por correo electrónico al usuario.

Nombre	Cambiar tono de notificación.
Casos de Uso relacionados	-
Precondición	Usuario logueado
Postcondición	Tono de notificación cambiado.
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa en <i>Ajustes</i> .	
2- El usuario pulsa en <i>Ajustes de chat</i> .	
3- El usuario pulsa en <i>Tono de notificación</i> .	
	4- El sistema muestra el listado de tonos de notificación disponibles.
5- El usuario selecciona un tono.	
6- El usuario acepta.	
	7- El sistema cambia el tono de notificación.

Nombre	Activar/desactivar la vibración de las notificaciones.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	Vibración activada/desactivada.
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa en <i>Ajustes</i> .	
2- El usuario pulsa en <i>Ajustes de chat</i> .	
3- El usuario activa/desactiva la opción <i>Vibrar</i> .	
	4- El sistema activa/desactiva la vibración de las notificaciones.

Nombre	Eliminar cuenta.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	Cuenta de usuario cancelada.
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa en <i>Ajustes</i> .	
2- El usuario pulsa en <i>Info de cuenta</i> .	
3- El usuario pulsa en <i>Eliminar cuenta</i> .	
	4- El sistema muestra un campo de texto.
5- El usuario introduce su correo electrónico.	
6- El usuario pulsa el botón <i>Cancelar cuenta</i> .	
	7- El sistema comprueba el correo electrónico.
	8- El sistema muestra un diálogo solicitando la confirmación.
9- El usuario acepta la confirmación.	
	10- El sistema solicita al servidor la cancelación definitiva de la cuenta.
	11- El servidor envía un correo electrónico al usuario.
	12- El sistema cierra sesión y redirige al usuario a la pantalla de inicio de sesión.
Excepciones	
· El correo electrónico no concuerda con el del usuario.	

Gestión de grupos

Seguidamente se exponen las funcionalidades disponibles para la gestión de los grupos. Destacan las características *Crear nuevo grupo*, *Invitar usuario*, *Modificar grupo*, *Obtener los usuarios del grupo*, *Consultar grupo*, *Visualizar ránking fotos*, *Visualizar fotos nuevas*, *Puntuar fotos* y *Salir del grupo*.

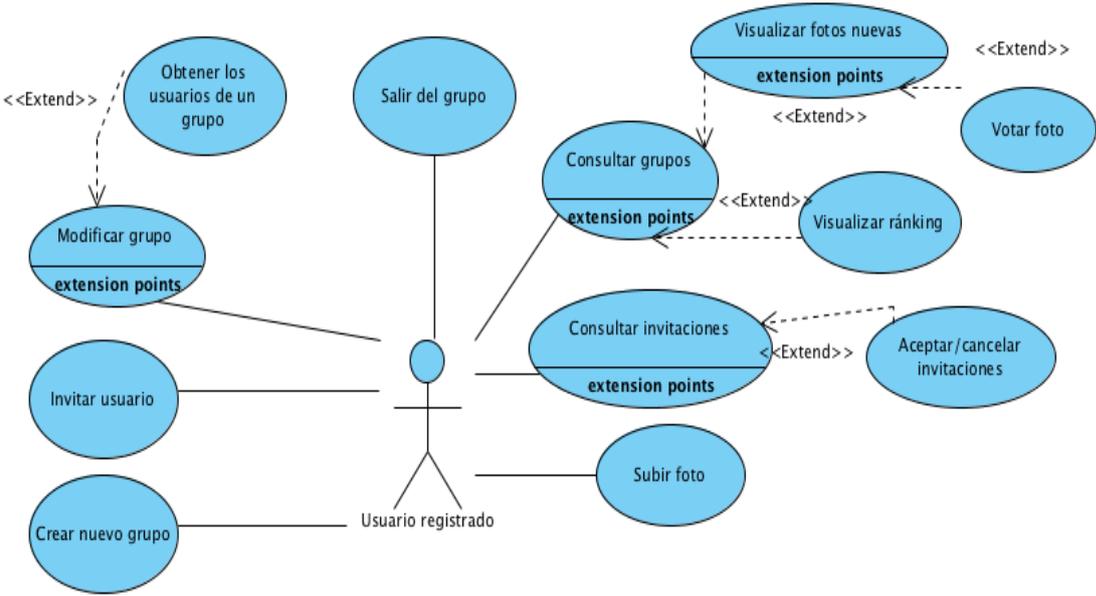


Figura 3.4: Gestión de grupos

Nombre	Crear nuevo grupo.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	Grupo nuevo y el usuario registrado convierte su rol a usuario administrador del grupo.
Proceso	
Usuario	Sistema
1- El usuario situado en la pantalla <i>Home</i> pulsa la opción del menú o el último icono que aparece al final de los grupos.	
	2- El sistema muestra un breve formulario.
3- El usuario, opcionalmente, pulsa sobre el icono de image.	
	4- El sistema abre la galería de imágenes del teléfono.
5- El usuario selecciona la imagen.	
	6- El sistema cambia la imagen por la seleccionada.
7- El usuario introduce obligatoriamente el nombre del grupo y opcionalmente una descripción.	
8- El usuario pulsa el botón <i>Siguiente</i> .	
	9- El sistema abre una nueva pantalla con el listado de amigos del usuario.
10- El usuario puede seleccionar los usuarios que serán miembros del grupo.	
11- El usuario pulsa la opción del menú <i>Enviar</i> .	
	12- El sistema envía toda la información al servidor.
	13- El sistema crea el grupo y redirige al usuario a la pantalla <i>Home</i> con el nuevo grupo.
Excepciones	
· El grupo debe tener un nombre.	

Nombre	Invitar usuario.
Casos de Uso relacionados	Consultar grupos.
Precondición	El usuario debe ser miembro del grupo.
Postcondición	Invitación enviada.
Proceso	
Usuario	Sistema
1- El usuario en la pantalla <i>Home</i> selecciona un grupo.	
	2- El sistema muestra una nueva pantalla con las fotos del grupo seleccionado.
3- El usuario pulsa el menú y selecciona la opción <i>Invitar usuario</i> .	
	4- El sistema muestra el listado de amigos del usuario que no pertenecen a ese grupo.
5- El usuario selecciona los amigos.	
6- El usuario pulsa la opción <i>Invitar usuarios</i> del menú.	
	7- El sistema envía las invitaciones a los usuarios seleccionados.
	8- El sistema redirige al usuario al grupo actual.

Nombre	Modificar grupo.
Casos de Uso relacionados	-
Precondición	El usuario debe ser miembro del grupo.
Postcondición	Grupo modificado.
Proceso	
Usuario	Sistema
1- El usuario en la pantalla <i>Home</i> selecciona un grupo.	
	2- El sistema muestra una nueva pantalla con las fotos del grupo seleccionado.
3- El usuario pulsa el menú y selecciona la opción <i>Modificar</i> .	
	4- El sistema muestra la descripción del grupo.
5- El usuario, opcionalmente, pulsa en la imagen del grupo.	
	6- El sistema abre la galería de imágenes del teléfono.
7- El usuario selecciona una foto.	
	8- El sistema cambia la foto del grupo.
9- El usuario, opcionalmente, cambia el nombre del grupo.	
10- El usuario hace <i>scroll</i> hasta llegar al final de la pantalla y pulsa el botón <i>Siguiente</i> .	
	11- El sistema envía la información actualizada del grupo al servidor.
	12- El sistema redirige al usuario a las fotos del grupo.
Excepciones	
· El grupo debe contener un nombre, no puede estar vacío.	

Nombre	Obtener los usuarios de un grupo.
Casos de Uso relacionados	Modificar grupo
Precondición	Usuario logueado
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario en la pantalla <i>Home</i> selecciona un grupo.	
	2- El sistema muestra una nueva pantalla con las fotos del grupo seleccionado.
3- El usuario pulsa el menú y selecciona la opción <i>Modificar</i> .	
	4- El sistema muestra el título del grupo, la descripción, la imagen y el listado de miembros del grupo.

Nombre	Salir del grupo.
Casos de Uso relacionados	-
Precondición	El usuario debe ser miembro del grupo.
Postcondición	El usuario no pertenece al grupo.
Proceso	
Usuario	Sistema
1- El usuario en la pantalla <i>Home</i> selecciona un grupo.	
	2- El sistema muestra una nueva pantalla con las fotos del grupo seleccionado.
3- El usuario pulsa el menú y selecciona la opción <i>Salir del grupo</i> .	
	4- El sistema envía la información al servidor.
	5- El sistema redirige el usuario a la pantalla <i>Home</i> , donde ya no aparecerá este grupo.

Nombre	Consultar grupos.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa en <i>Home</i> o en el icono de la izquierda de la barra inferior (dibujo de una casita).	
	2- El sistema realiza una consulta a la base de datos.
	3- El sistema procesa los datos y devuelve todos los grupos.

Nombre	Visualizar fotos nuevas.
Casos de Uso relacionados	Consultar grupos.
Precondición	Consultar grupos.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario situado en la pantalla <i>Home</i> selecciona el grupo que desea ver las fotos nuevas.	
	2- El sistema realiza una consulta al servidor para obtener las fotos nuevas.
	3- El sistema procesa los datos y visualiza las imágenes.

Nombre	Votar foto.
Casos de Uso relacionados	Visualizar fotos nuevas.
Precondición	Consultar grupos.
Postcondición	Foto votada.
Proceso	
Usuario	Sistema
1- El usuario situado en la pantalla <i>Home</i> selecciona el grupo que contiene fotos nuevas.	
	2- El sistema realiza una consulta al servidor para obtener las fotos de un grupo que están por puntuar por dicho usuario.
	3- El sistema procesa la información y muestra las imágenes en un Grid.
4- El usuario pulsa sobre una imagen.	
	5- El sistema muestra la imagen con su título, el nombre del propietario y cinco estrellas en con fondo blanco.
6- El usuario puntúa la foto.	
	7- El sistema envía dicha puntuación al servidor.
	8- El sistema solicita el listado de puntuaciones de dicha foto.
	9- El sistema muestra el listado de puntuaciones que tiene esa foto.

Nombre	Visualizar el ránking de un grupo.
Casos de Uso relacionados	Consultar grupo.
Precondición	Consultar grupos.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario situado en la pantalla <i>Home</i> selecciona el grupo que desea visualizar.	
	2- El sistema realiza una consulta al servidor para obtener las fotos puntuadas y ordenadas por puntuación.
	3- El sistema procesa los datos y visualiza las imágenes.

Nombre	Consultar invitaciones.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa en <i>Invitaciones</i> .	
	2- El sistema realiza una consulta a la base de datos.
	3- El sistema procesa los datos y devuelve el listado de invitaciones del usuario.

Nombre	Aceptar/Cancelar invitación.
Casos de Uso relacionados	Consultar invitaciones.
Precondición	Usuario logueado.
Postcondición	Invitación aceptada/cancelada.
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa en <i>Invitaciones</i> .	
	2- El sistema realiza una consulta a la base de datos.
	3- El sistema procesa la datos y los muestra por pantalla.
4- El usuario acepta/cancela la invitación.	
	5- El sistema envía la información al servidor.
	6- El sistema elimina la invitación de la base de datos.
	7- El sistema redirige al usuario a la pantalla <i>Home</i> (sólo si es la última invitación por responder).

Nombre	Subir foto.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	Foto subida al grupo.
Proceso	
Usuario	Sistema
1- El usuario, en cualquier pantalla, pulsa el icono situado en el centro de la barra inferior (imagen del objetivo de una cámara).	
	2- El sistema ofrece dos opciones: subir desde la cámara de fotos del teléfono o desde la galería.
3- El usuario selecciona una opción.	
	4a- El sistema lanza la cámara de fotos.
	4b- El sistema muestra la galería de imágenes.
5a- El usuario toma una foto.	
5b- El usuario selecciona una imagen.	
	6- El sistema muestra la foto.
7- El usuario introduce opcionalmente un título y pulsa el botón <i>Enviar a</i> .	
	8- El sistema muestra todos los grupos del usuario.
9- El usuario selecciona un grupo.	
	10- El sistema envía la foto al servidor al grupo seleccionado.
	11- El sistema redirige el usuario a la <i>Home</i> .

Gestión perfiles de usuario

En la figura 3.5 se muestran los casos de uso que un usuario puede realizar en el ámbito de los perfiles de usuario. Principalmente hacen referencia a la acción de visitar los perfiles desde diferentes lugares de la aplicación. Cabe destacar que existe la relación *extend* entre dos de ellos. Dicha relación documenta que el caso es opcional y que se puede utilizar cuando se desee.

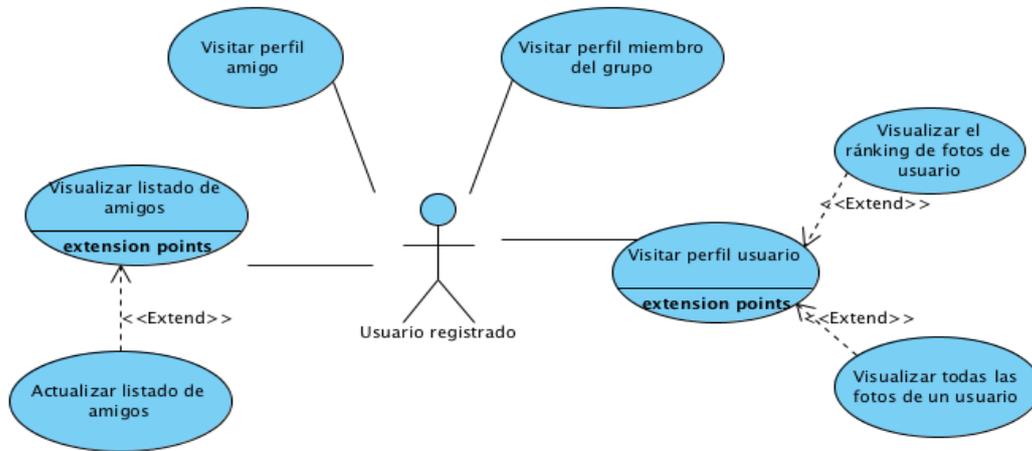


Figura 3.5: Gestión perfiles de usuario

Nombre	Visualizar listado de amigos.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa en <i>Amigos</i> .	
	2- El sistema realiza una consulta a la base de datos.
	3- El sistema devuelve todos los amigos del usuario.

Nombre	Actualizar lista de amigos.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	Lista de amigos actualizada.
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa en <i>Amigos</i> .	
2- El pulsa el icono del menú.	
	3- El sistema realiza una consulta al servidor.
	4- El sistema procesa los datos y devuelve la lista de amigos actualizada.

Nombre	Visitar perfil de un amigo.
Casos de Uso relacionados	Consultar amigos.
Precondición	Usuario logueado.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> .	
2- El usuario pulsa en <i>Mis Amigos</i> .	
	3- El sistema muestra el listado de amigos del usuario.
4- El usuario pulsa sobre el amigo.	
	5- El sistema realiza una consulta al servidor.
	6- El sistema procesa los datos y muestra el perfil del amigo.

Nombre	Visitar perfil de un miembro del grupo.
Casos de Uso relacionados	Modificar grupo.
Precondición	Usuario logueado.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario desde <i>Home</i> selecciona un grupo.	
	2- El sistema visualiza el grupo con sus fotografías.
3- El usuario pulsa el botón de menú.	
4- El usuario selecciona <i>Modificar</i> .	
	5- El sistema muestra el título del grupo, la descripción, la foto y el listado de usuarios.
6- El usuario pulsa sobre un usuario del listado.	
	7- El sistema realiza una consulta al servidor.
	8- El sistema redirige al usuario al perfil del usuario seleccionado.

Nombre	Visitar el perfil de un usuario.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario, en cualquier pantalla, pulsa en la barra inferior el icono de la derecha (imagen de una estrella).	
	2- El sistema realiza una petición al servidor.
	3- El sistema procesa y carga el perfil del usuario logueado.

Nombre	Visualizar el ránking de fotos de un usuario.
Casos de Uso relacionados	Visitar perfil usuario.
Precondición	Usuario situado en el perfil de un usuario.
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario pulsará en la pestaña <i>RÁN-KING</i> .	
	2- El sistema realiza una consulta al servidor.
	3- El sistema procesa los datos y visualiza el ránking de imágenes del usuario seleccionado.

Nombre	Visualizar todas las fotos de un usuario.
Casos de Uso relacionados	Visitar perfil usuario.
Precondición	Usuario situado en el perfil de un usuario.
Postcondición	Listado de todas las fotos de un usuario
Proceso	
Usuario	Sistema
1- El usuario pulsará en la pestaña <i>TODAS</i> .	
	2- El sistema realiza una consulta al servidor.
	3- El sistema procesa los datos y visualiza todas las imágenes del usuario.

Gestión de datos de la aplicación

Acto seguido se describen las acciones con las que el usuario podrá tratar con datos son: *Compartir la aplicación*, *Consultar FAQ* y *Contactar con el administrador del sistema*.

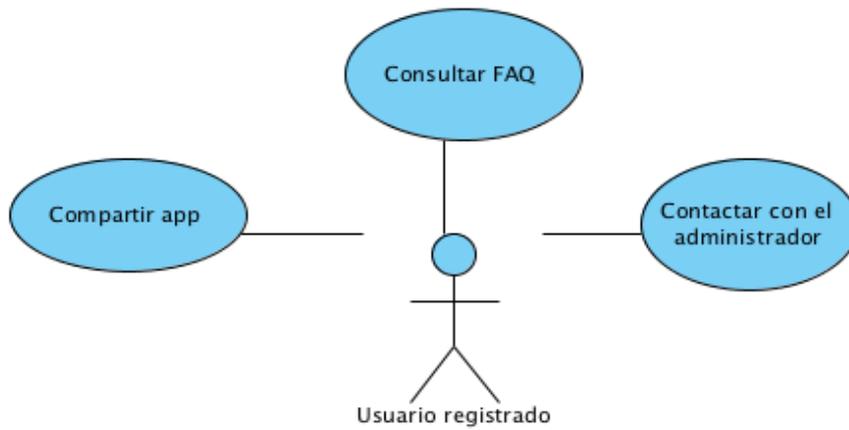


Figura 3.6: Gestión de datos de la aplicación

Nombre	Compartir app.
Casos de Uso relacionados	-
Precondición	Usuario logueado
Postcondición	-
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa en <i>Compartir</i> o desde la pantalla <i>Home</i> utilizando el icono <i>Invita a tus amigos</i> .	
	2- El sistema muestra las aplicaciones que admiten recibir datos desde otras aplicaciones.
3- El usuario selecciona una aplicación.	
	4- El sistema abre la aplicación con el texto y el enlace.

Nombre	Consultar preguntas frecuentes.
Casos de Uso relacionados	-
Precondición	Usuario logueado
Postcondición	Página web de FAQ
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa <i>Ajustes</i> .	
2- El usuario pulsa en <i>Ayuda</i> .	
3- El usuario pulsa en <i>Preguntas frecuentes</i> .	
	4- El sistema muestra en pantalla la página web de FAQ.

Nombre	Contactar con el administrador.
Casos de Uso relacionados	-
Precondición	Usuario logueado.
Postcondición	Mensaje enviado.
Proceso	
Usuario	Sistema
1- El usuario abre el <i>Drawer</i> y pulsa <i>Ajustes</i> .	
2- El usuario pulsa en <i>Ayuda</i> .	
3- El usuario pulsa en <i>Contáctanos</i> .	
	4- El sistema muestra en pantalla texto con un campo donde introducir la solicitud.
5- El usuario escribe su problema o duda.	
6- El usuario pulsa el botón <i>Enviar</i>	
	7- El sistema envía el comentario al servidor.
	8- El servidor envía una copia del contenido del mensaje al correo electrónico.
	9- El sistema redirige al usuario a la pantalla anterior.
Excepciones	
· El campo está vacío. Escribe tu pregunta o comentario antes de enviar.	

3.2.4. Usuario administrador del grupo

El usuario que posee el rol de administrador de un grupo podrá expulsar usuarios que suban las fotografías que no se adecuen con la descripción o eliminar a aquellos que hayan sido invitados sin consentimiento del resto del grupo (ver tabla 3.1). Además podrá realizar todas las funcionalidades que el *Usuario registrado*.



Figura 3.7: Usuario administrador de un grupo.

3.2.5. Usuario desactivado

Este usuario se caracteriza por haber eliminado anteriormente su cuenta en el sistema. En el momento que realice la única funcionalidad disponible, *Reactivar cuenta* (ver tabla 3.2), tomará el rol de *Usuario registrado* y volverá a tener acceso normal en la aplicación.

Nombre	Eliminar usuarios de un grupo
Casos de Uso relacionados	Invitar usuario.
Precondición	El usuario debe ser el creado del grupo ^[1] .
Postcondición	Usuarios eliminados del grupo.
Proceso	
Usuario	Sistema
1- El usuario en la pantalla <i>Home</i> selecciona un grupo.	
	2- El sistema muestra una nueva pantalla con las fotos del grupo seleccionado.
3- El usuario pulsa el menú y selecciona la opción <i>Eliminar usuarios</i> .	
	4- El sistema muestra el listado de usuarios que pertenecen al grupo.
5- El usuario selecciona los usuarios que desea expulsar del grupo.	
6- El usuario pulsa la opción <i>Eliminar</i> del menú.	
	7- El sistema envía al servidor el listado de usuarios seleccionados.
	8- El sistema redirige el usuario a la pantalla <i>Home</i> .

Cuadro 3.1: Funcionalidad disponible sólo para aquellos usuarios que posean el rol de administrador del grupo.

Nombre	Reactivar cuenta.
Casos de Uso relacionados	-
Precondición	Usuario con cuenta eliminada.
Postcondición	Usuario con cuenta activa y pasa a tener el rol de usuario registrado.
Proceso	
Usuario	Sistema
1- El usuario inicia la <i>app</i> .	
2- El usuario introduce su usuario y contraseña.	
	3- El sistema realiza un intento de login.
	4- El sistema recibe el estado de la cuenta.
	5- El sistema muestra un diálogo de activación de la cuenta.
6- El usuario pulsa en el botón <i>Aceptar</i> .	
	7- El sistema reactiva la cuenta.
	8- El servidor envía un correo electrónico al usuario.
	9- El sistema inicia sesión automáticamente y muestra la pantalla <i>Home</i> .

Cuadro 3.2: Funcionalidad disponible para aquellos usuarios que hayan eliminado su cuenta anteriormente.



Figura 3.8: Usuario desactivado.

3.3. Diagrama de clases

En este apartado se presenta el diagrama de clases del sistema. En este diagrama se describe la estructura de un sistema. En él se puede identificar las clases que contiene, las propiedades que tienen los objetos, las operaciones que pueden realizar, las relaciones estructurales y de herencia que se establecen entre las entidades.

3.3.1. Usuarios de la aplicación

Acto seguido se presenta el diagrama de clases que simboliza toda la información que se genera sobre los usuarios, las imágenes que suben, las puntuaciones, los grupos, las invitaciones, las relaciones de amistad y los comentarios que envía. Todo ello quedará almacenado en el servidor.

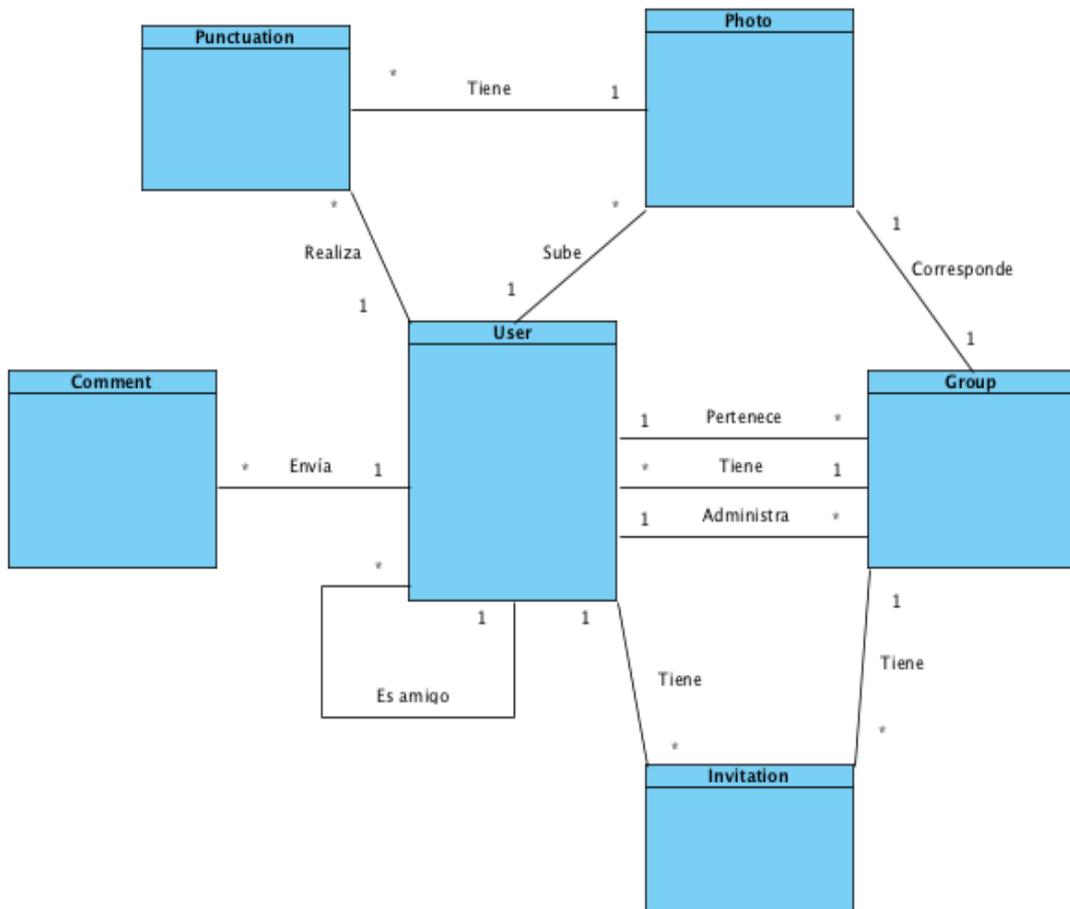


Figura 3.9: Diagrama de clases: Usuarios.

La clase principal de la aplicación es la clase **Usuario** a partir de la cual se genera toda la información.

Un **Usuario** puede ser miembro de muchos grupos así como puede administrar varios de ellos a la vez. Un **Grupo** posee muchos usuarios y un único administrador. A cada uno de los grupos se accede mediante **Invitación**. Una **Invitación** es creada por un usuario y en ella se incluye el **Usuario** destinatario y el identificador del **Grupo** al que se invita.

Un **Usuario** puede subir muchas fotografías a los grupos, pero cada fotografía pertenecerá exclusivamente a un **Grupo**. Una vez la foto está subida a un **Grupo**, el resto de usuarios de ese grupo tienen que votar esa **Foto** una única vez. Por tanto una **Foto** tiene asociadas **Puntuaciones** y las puntuaciones las llevan a cabo otros **Usuarios** de ese **Grupo**.

Un **Usuario** tiene relación, conoce o es amigo de otros usuarios, por tanto esa relación de muchos a muchos se ilustra con la tabla **Amigo**. En la cual se almacenan los identificadores de los **Usuarios**.

Como última entidad a comentar, la clase **Comentario**, donde un **Usuario** crea y envía tantos comentario al administrador del sistema como quiera. En ella se guarda el identificador del **Usuario** que escribe el comentario junto con el contenido y la fecha de creación.

Capítulo 4

Diseño

4.1. Introducción

Para el desarrollo y diseño de la aplicación móvil se ha seguido el patrón de arquitectura conocido como **Modelo Vista Controlador** (MVC), cuya arquitectura separa los datos de la aplicación, la interfaz del usuario y la lógica de negocio en tres componentes. El MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, que prácticamente define por un lado los componentes para la representación de la información, y por otro lado, la interacción del usuario [42].

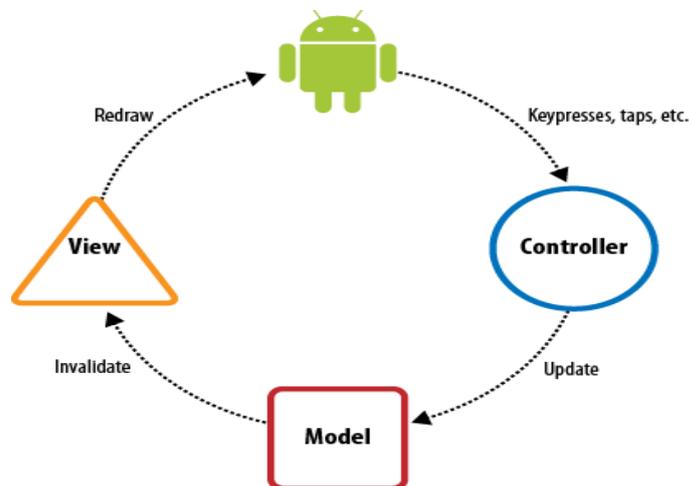


Figura 4.1: Modelo Vista Controlador.

- El modelo simboliza la información con la que trabaja la aplicación (lógica de negocio).

- La **vista** convierte el modelo en una pantalla con la que el usuario puede interactuar.
- El **controlador** es el encargado de procesar las interacciones y eventos del usuario y realizar los cambios apropiados en el modelo y/o en la vista.

4.2. Arquitectura del sistema

Se define como el diseño o conjunto de relaciones entre las partes que constituyen un sistema. Android está formado por varias capas que ayudan al desarrollador para crear aplicaciones. También, se otorgan permisos para acceder a las capas inferiores mediante librerías para el usuario no tenga que programar a un nivel más bajo las funcionalidades que se precisen para que la aplicación haga uso de las características del hardware del terminal.

Android utiliza la arquitectura de tipo Pila ya que una capa puede hacer uso de elementos de la capa inferior.

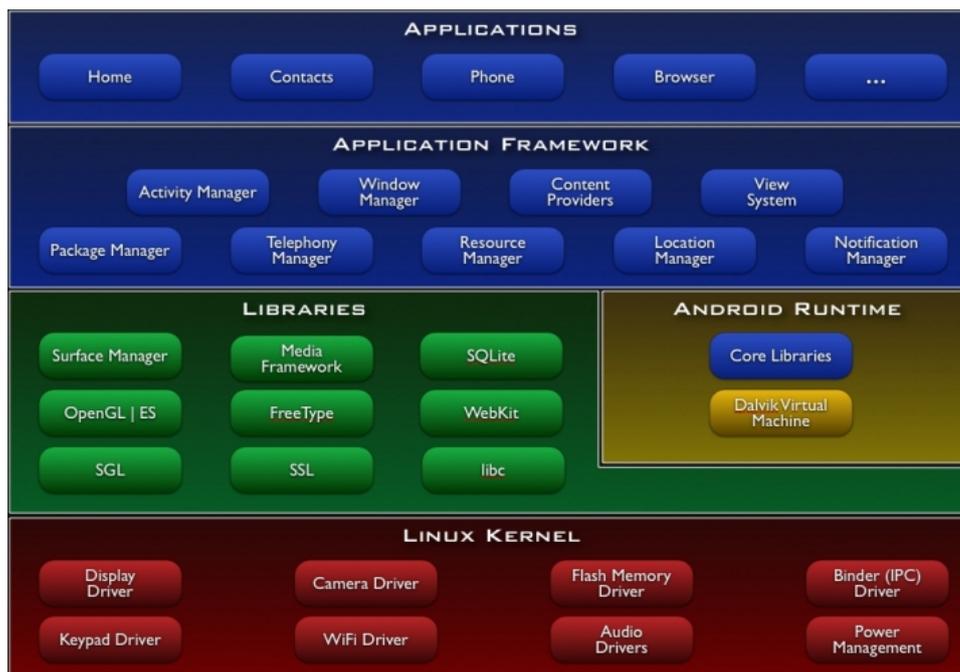


Figura 4.2: Arquitectura de tipo Pila utilizada por Android.

A continuación se describe en forma general las capas empezando desde abajo hacia arriba [2]:

- **Kernel de Linux:** situada en la parte inferior de las capas. Gestiona los diferentes recursos del teléfono y del sistema operativo. Proporciona la funcionalidad básica para la gestión de procesos, la gestión de la memoria, la gestión de dispositivos como la cámara, teclado y pantalla (entre otros). El núcleo del sistema está basado en el Kernel de Linux versión 2.6, con la característica de estar adaptado a las características del hardware para dispositivos móviles. Actúa como una capa de abstracción entre el hardware y el resto de capas. Al desarrollador no se le permite acceder directamente a esta capa, sino que debe utilizar las librerías situadas en la capa superior.
- **Librerías:** situadas en la parte superior del Kernel, son un conjunto de librerías nativas de Android compiladas específicamente para la arquitectura hardware del teléfono. Producidas, por regla general, por los fabricantes del teléfono, cuya función principal es suministrar funcionalidades a las aplicaciones para tareas que se repiten con frecuencia. Se pueden encontrar librerías de motor gráfico como *OpenGL*, navegador web *WebKit*, bibliotecas multimedia para audio, imagen y video, base de datos *SQLite* o cifrado *SSL* para las comunicaciones.
- **Entorno de ejecución:** no se considera una capa como tal porque está formada por librerías. En ella se incluyen librerías con funcionalidades de Java y de Android. Como se aprecia, contiene una máquina virtual conocida como *Dalvik*. Las aplicaciones se codifican en Java y se compilan en un formato especial para que dicha máquina virtual las pueda ejecutar¹. Las aplicaciones se compilan una primera y única vez, así estarán disponibles para ser distribuidas para cualquier dispositivo Android con la mínima versión del sistema operativo que la aplicación necesite.
- **Framework de aplicaciones:** ofrece muchos servicios de alto nivel a las aplicaciones en forma de clases de Java. Los desarrolladores tienen autorización para hacer uso de estos servicios en sus aplicaciones. Prácticamente, la mayoría de elementos de esta capa son librerías que acceden a recursos de capas inferiores mediante Dalvik. A continuación se citarán aquellas más importantes que han sido utilizadas mediante el desarrollo:
 - *Activity Manager*: administra la pila de actividades de la aplicación y el ciclo de vida de éstas.
 - *Windows Manager*: organiza aquello que será mostrado por pantalla.

¹Dalvik es una variación de la máquina virtual de Java, por esta razón, no es compatible con el bytecode de Java. Los ejecutables que genera el SDK de Android tienen la extensión *.dex* que es característica para Dalvik, por tanto, las aplicaciones Java no pueden ser ejecutadas en Android y viceversa.

- *Content Provider*: encapsula los datos para que puedan ser compartidos entre aplicaciones y facilita el control para el acceso a la información.
 - *Notification Manager*: clase que notifica al usuario de los acontecimientos que suceden en primer o segundo plano en la barra de estado. Autoriza a modificar y activar/desactivar los sonidos, la vibración, los iconos y el LED.
 - *Resource Manager*: gestiona elementos que forman parte de la aplicación y que están fuera del alcance del programador mediante código, como sería la traducción del texto de la aplicación dependiendo del lenguaje actual del dispositivo o sonidos, entre otros..
 - *Cámara*: permite obtener vídeos e imágenes desde la cámara del dispositivo.
- **Aplicaciones**: última capa de la arquitectura donde están comprendidas las aplicaciones del dispositivo, como aquellas que tienen o no interfaz de usuario, nativas (escritas en C o C++), administrativas (escritas en Java), las reinstaladas y otras instaladas por el propio usuario. Cabría destacar también la aplicación principal del sistema, Inicio o *Home*, ya que autoriza la ejecución de otras aplicaciones mediante una lista, además de mostrar todas la aplicaciones en diferentes escritorios con iconos de acceso directo o widgets.

4.3. Capa de persistencia

En programación orientada a objetos, esta capa confiere a las demás, el servicio de persistencia de datos o información y su posterior recuperación. Esta capa encapsula el acceso a los datos de las aplicaciones que la utilicen, siendo posible trabajar con varios gestores de datos sin necesidad de cambiar el código.

Android ofrece varias opciones para persistir los datos. Entre todas ellas, se destacarán las utilizadas.

4.3.1. SQLite

Es un sistema de gestión de bases de datos relacional contenida en una pequeña biblioteca escrita en C. El motor de SQLite no es un proceso independiente con el cual el programa o la aplicación se comunica, sino que esta biblioteca se enlaza con el programa pasando a ser parte íntegra del mismo. El programa utiliza las características de SQLite mediante llamadas simples a subrutinas y funciones. Es por ello que se reduce el tiempo de acceso a los datos. La base de datos se guarda como un sólo fichero estándar en la máquina host [51].

La figura 4.3 representa el diagrama Entidad-Relación que almacena los datos de un usuario cuando inicia sesión en el sistema. Cabe destacar que este diagrama difiere con el de la figura 3.9, ya que en este sólo se guarda la información necesaria para la navegación entre pantallas para un usuario determinado y no almacena más información que la estrictamente necesaria.

Resaltar la herencia entre la clase **Amigo** respecto a la clase **Usuario**, ya que en sí contienen los mismos atributos y realizan las mismas acciones, pero es la forma escogida para vincular el usuario principal con el resto de usuarios conocidos.

Por otra parte, la clase foto se especializa en dos subclases, **Foto ranking** y **Foto nueva**, que se utilizan respectivamente en las pestañas **Ranking** y **Nuevas**, cuando un usuario acceda a un grupo. Cuando una foto nueva sea votada por un usuario, para éste mismo, esta fotografía cambiará su clase por la de foto puntuada.

Simbólicamente se ha añadido la clase **Administrador Grupo** al diagrama para dar relevancia del papel de un usuario que crea un grupo. No existe realmente dicho usuario en el código de la aplicación, ya que un usuario obtiene este rol si al pulsar el menú de opciones de un grupo el sistema compara su identificador con el identificador del propietario del grupo y coincide, entonces se activará el caso de uso 3.1.

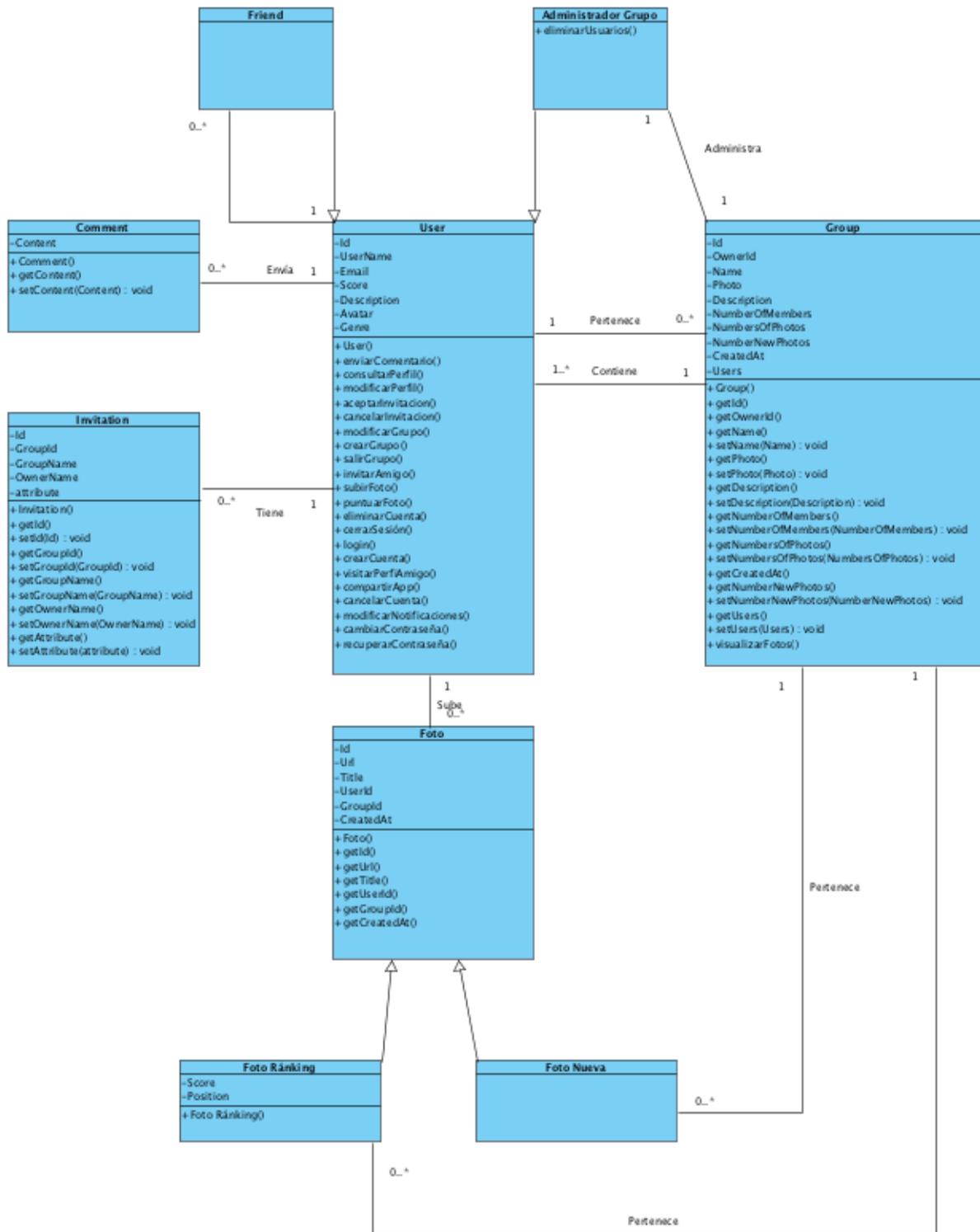


Figura 4.3: Diagrama Entidad-Relación de la aplicación.

4.3.2. Shared Preferences

Ofrece un marco general que permite al usuario guardar y recuperar datos con el patrón clave-valor, cada preferencia estará compuesta por un identificador único y un valor asociado a dicho identificador. Como diferencia a `SQLite`, los datos no se guardan en un fichero binario de tipo base de datos, sino en ficheros XML [13].

Una aplicación Android puede manejar diferentes preferencias que se diferencia por un identificador que también será único. Para obtener la referencia de una colección de datos almacenados, se utiliza el método `"getSharedPreferences()"` que recibe por parámetros el identificador de la colección y el modo de acceso:

- `MODE_PRIVATE`: únicamente la aplicación tiene acceso a los ficheros de preferencias.
- `MODE_WORLD_READABLE`: todas las aplicaciones tienen acceso a las preferencias pero sólo la aplicación creadora de las preferencias puede modificar los datos.
- `MODE_WORLD_WRITABLE`: todas las aplicaciones tiene acceso de lectura y escritura sobre las preferencias.

La aplicación contiene diferentes preferencias que almacenan configuraciones del usuario como el tono y la activación/desactivación de la vibración para las notificaciones, el correo electrónico/teléfono junto con la contraseña del usuario para el inicio de sesión automático y por último, un flag que determina si la aplicación ha sido ejecutada por primera vez para que el usuario visualice una pequeña presentación de imágenes al inicio. Puesto que se procesan con datos personales, el acceso será privado.

4.3.3. Disk LRU Cache

Se ha utilizado una api implementada en Java diseñada especialmente para terminales con Android [14].

Descripción

Una memoria caché utiliza una cantidad limitada de espacio en un sistema de ficheros. Cada entrada de la caché tiene asociada una clave. Cada clave debe coincidir con una expresión regular de la forma $[a-z0-9]^{\{1,64\}}$. Los valores son secuencias de bytes, accesibles como streams o archivos. Cada valor debe tener una longitud comprendida entre 0 y `Integer.MAX_VALUE`.

La memoria caché almacena sus datos en un directorio del sistema. Este directorio debe ser de acceso exclusivo para la caché. La memoria caché puede borrar o sobrescribir archivos de ese directorio.

Esta caché limita el número de bytes para almacenar en el sistema. El usuario realiza llamadas a la caché para crear nuevas entradas o para actualizar otras. Cuando el número de bytes almacenados ha excedido el límite, la caché eliminará entradas que lleven mayor tiempo almacenadas en disco hasta que haya suficiente para almacenar nuevos datos.

Problemática

Los datos de la aplicación son fotografías, así que la *app* continuamente solicita imágenes al servidor para descargarlas y mostrarlas al usuario. Los principales problemas de la descarga continua de imágenes, se reflejan en el aumento del consumo de datos, la disminución de energía en el dispositivo y la sobrecarga del servidor. Para solucionar estos problemas, la mejor opción es introducir un sistema de caché que se encargue de la gestión de las imágenes (descarga, almacén y obtención de fotos).

4.3.4. Protocolo remoto

La aplicación realiza de forma activa peticiones HTTP (Http Request) a la dirección ip del servidor. Éste se encuentra a la espera de recibir el token (identificador único) del usuario, acompañado de un listado de objetos de la clase `NameValuePair`, que son pares *clave-valor* separados por el carácter `&`, donde cada *clave* es conocida por el servidor y *valor* es la información que envía/solicita el cliente.

Código 4.1: Ejemplo de petición de datos al servidor.

```
http://54.77.55.82/app_dev.php/api?_r=getNewPhotos
&_token=eyJmJSUzI1NiIsIn*****xcVQ=&g=25}
```

El servidor responde a la solicitud mediante un objeto JSON el cual desglosándose tiene una forma como la siguiente:

Código 4.2: Ejemplo de datos en formato JSON.

```
1 {
2 "photos": [
3   {
```

```
4     "id":272,
5     "url":"http://54.77.55.82/images/
        upload_photos_group/25bbd58ac769cd232fae9b36f94
        7f94f4b0e123d.jpeg",
6     "title":"","
7     "groupId":"25",
8     "userId":"120"
9     },
10    {
11     "id":271,
12     "url":"http://54.77.55.82/images/
        upload_photos_group/7cea5344553cb5de21a8e5bf885
        6eaf0b7064496.jpeg",
13     "title":"","
14     "groupId":"25",
15     "userId":"120"},
16    {
17     "id":266,
18     "url":"http://54.77.55.82/images/
        upload_photos_group/f2b51bde61c6b6a7b965b3e3c87
        5e05358161290.jpeg",
19     "title":"Cala Comptesa (Mallorca)",
20     "groupId":"25",
21     "userId":"106"
22    }
23  ]
24 }
```

Se observar en el ejemplo que el servidor ha devuelto a la petición anterior un objeto JSON (ver sección 5.2.6) cuyo contenido es un `JSONArray` con la información de fotos nuevas que un usuario tiene por votar para el grupo que tiene con identificador 25.

4.4. Capa de presentación

En esta sección se explica como se presentan los datos al usuario de forma visual. La capa de presentación sienta las bases para la comunicación con la capa de negocio cuando se requiere almacenar datos o recuperarlos dependiendo de las acciones del usuario.

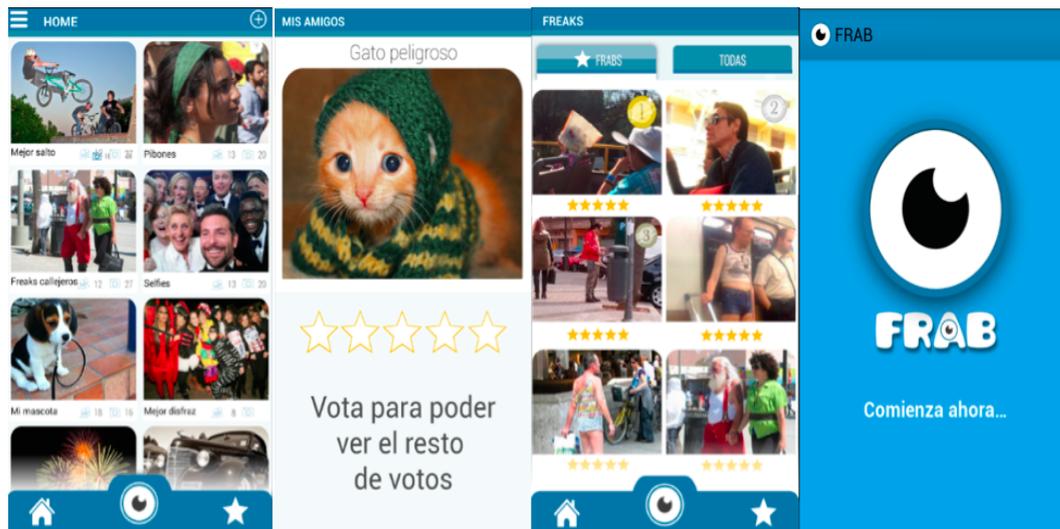


Figura 4.4: Pantallas de presentación de la aplicación.

En las aplicaciones Android la capa de presentación está formada por Actividades. Cada pantalla de la interfaz gráfica del usuario se crea a partir de la clase `Activity`. Las actividades utilizan objetos especiales conocidos como `Views` ^[47] que permiten la composición de la interfaz para que los usuarios puedan interactuar con la aplicación.

Una aplicación puede tener varias actividades y se pueden cambiar en tiempo de ejecución. Poseen su propio ciclo de vida.

Ciclo de vida de una `Activity`

Android inicia la aplicación con métodos específicos que corresponden a etapas de vida de sus actividades.

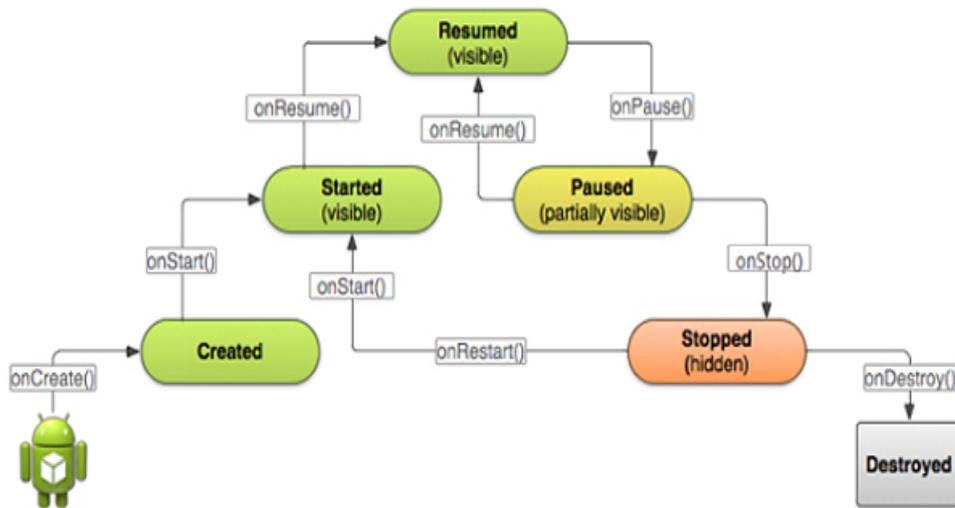


Figura 4.5: Ciclo de vida de una Actividad.

Android arranca mediante la invocación de métodos concretos que corresponden a etapas de vida. Inicia con el método `onCreate()` y finaliza en `onDestroy()`. Hacer notar que no siempre se utilizan todos los métodos del ciclo, depende las tareas que se vayan a realizar y si cuando se realizan [4].

Seguidamente se describirá el comportamiento de cada método:

- **onCreate()**: se ejecuta al inicio y define la interfaz del usuario, así como inicializa las variables de la clase. Se ejecuta una sola vez en el momento justo que se invoca la actividad. En este método se carga el archivo XML que da forma gráfica a la actividad o configura su interfaz.
- **onStart()**: la actividad está a punto de estar visible para el usuario.
- **onResume()**: en este momento la actividad se encuentra en primer plano y va a comenzar a interactuar con el usuario.
- **onPause()**: la aplicación está a punto de ser enviada a segundo plano, ya que otra actividad es lanzada. En este estado no se reciben datos del usuario y no se puede ejecutar código.
- **onStop()**: la actividad ya no va a ser visible para el usuario. Los valores de las variables se mantienen pero no se puede ejecutar código.
- **onRestart()**: la actividad volverá a ser mostrada al usuario después de haber pasado por el método `onStop()`. Seguidamente pasará por `onStart()` y `onResume()`.

- **onDestroy()**: el sistema antes de destruir la actividad la envía a este método. En este momento se deben liberar los recursos para evitar un consumo de memoria inútil o que pueda suceder un comportamiento inesperado.

4.5. Capa de control

Esta capa establece las reglas que deben cumplirse, realiza la función de controlar y administrar los datos que son enviados desde la capa de presentación y además solicita a la base de datos almacenar u obtener datos de ella. En función de los eventos que se reciban de la capa de presentación se modifica el modelo y se muestran los resultados.

4.5.1. Mapa de alcanzabilidad

El mapa de alcanzabilidad es el mapa que representa las transiciones disponibles desde una pantalla a otras, es decir, existe un camino que une una pantalla A con una pantalla B pasando por un número determinado de pantallas. Con este mapa se pueden determinar el número de caminos existen para llegar a un determinado estado accediendo desde diferentes partes.

Seguidamente se enumerarán las pantallas de las que consta la *app*:

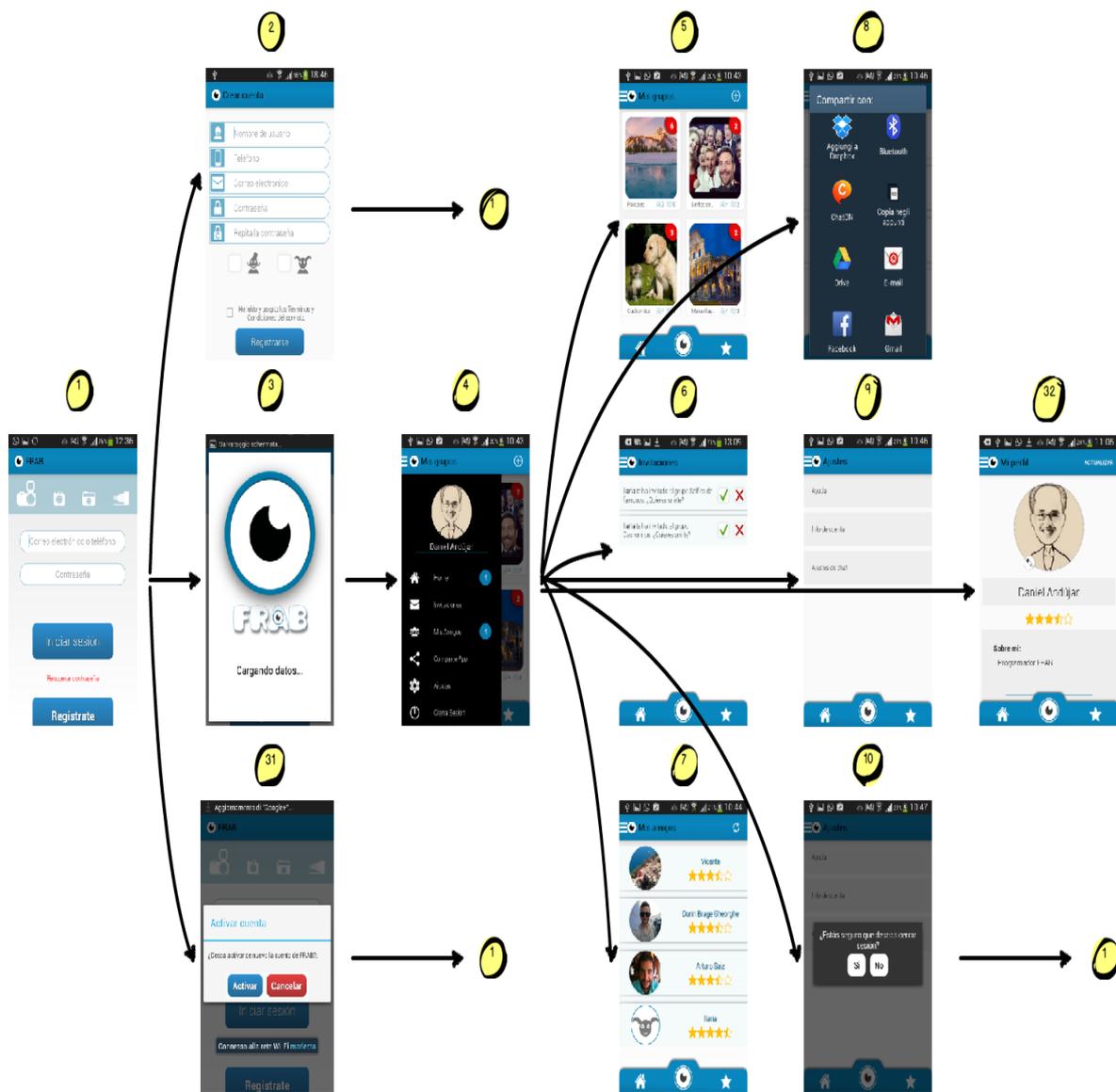


Figura 4.6: Primeras diez pantallas de la aplicación.



Figura 4.7: Pantallas accesibles desde un grupo.

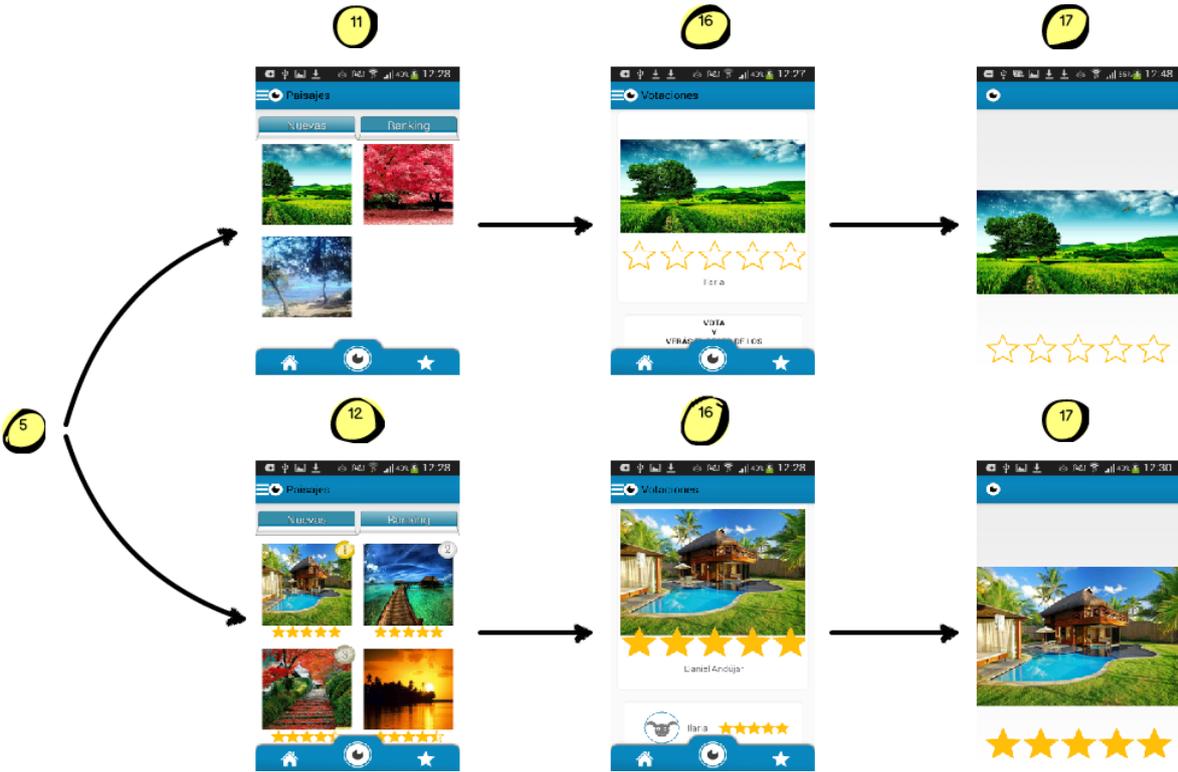


Figura 4.8: Pantallas para visualización/votación de fotografías de un grupo.

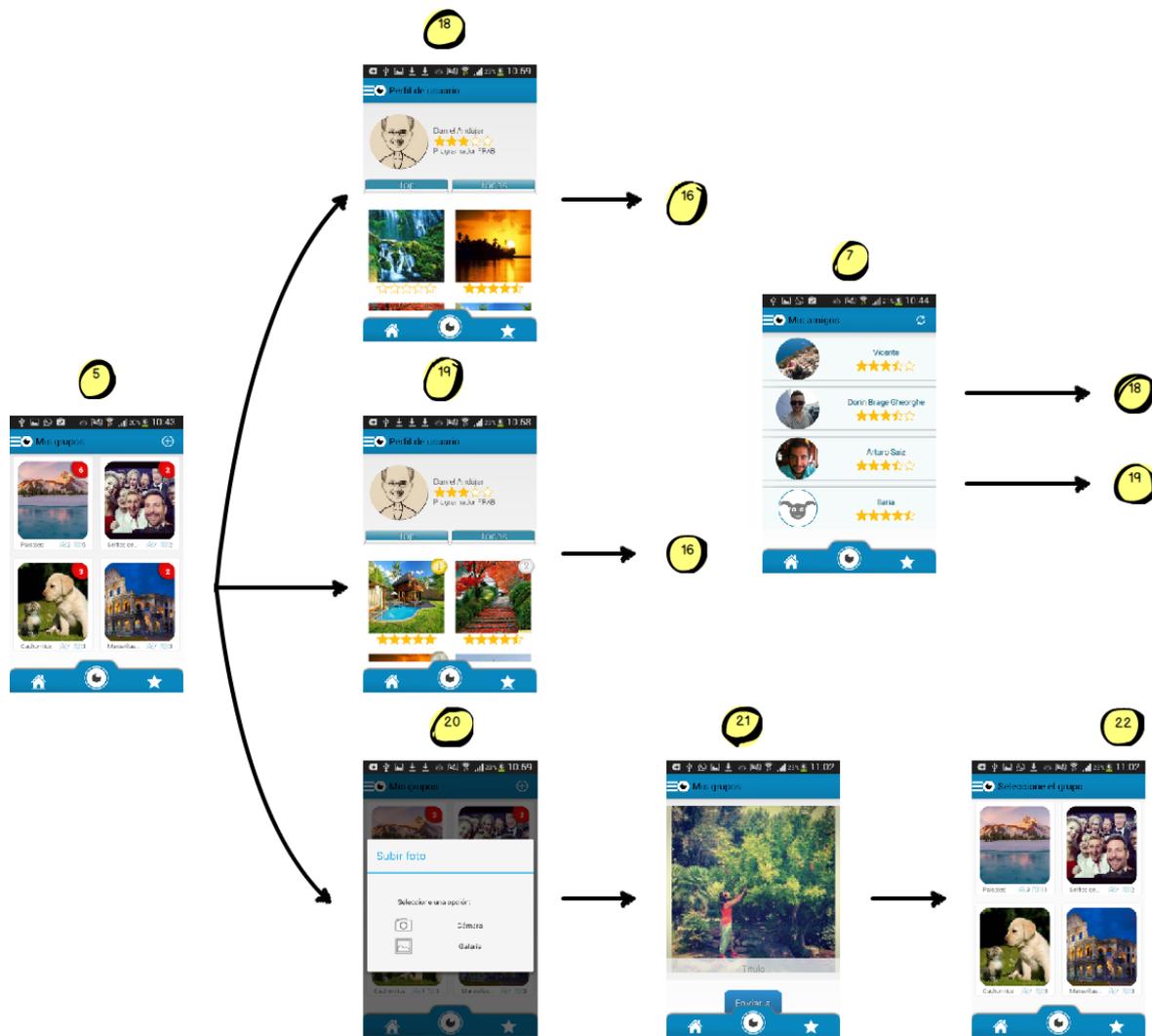


Figura 4.9: Pantallas del perfil de un usuario y subida de una foto.

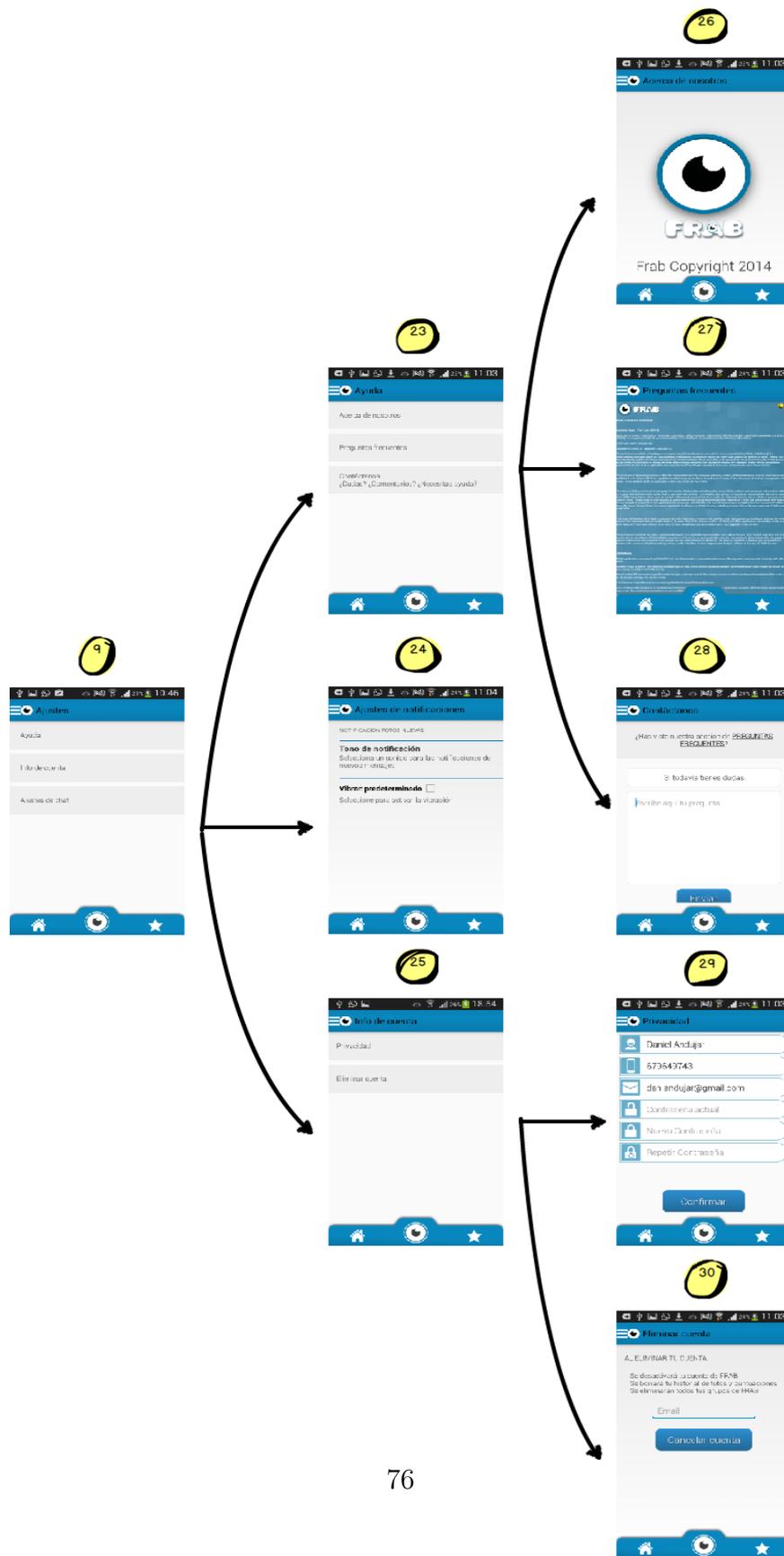


Figura 4.10: Pantallas de ajustes, información y contacto.

Acto seguido se enumerarán las pantallas que parecen en el mapa de alcanzabilidad:

- 1- Pantalla de login.
- 2- Pantalla de registro en el sistema.
- 3- Diálogo de inicio de sesión.
- 4- Drawer desplegado.
- 5- Pantalla de los grupos del usuario.
- 6- Pantalla de invitaciones pendientes del usuario.
- 7- Pantalla con el listado de amigos del usuario.
- 8- Listado de aplicaciones en las cuales poder compartir la *app*.
- 9- Pantalla de ajustes.
- 10- Diálogo para cerrar sesión.
- 11- Pantalla con las fotos nuevas de un grupo.
- 12- Pantalla con el ránking de fotos de un grupo.
- 13- Pantalla para invitar amigos a un grupo que no sean miembros.
- 14- Pantalla para modificar/consultar un grupo.
- 15- Pantalla para eliminar usuarios de un grupo.
- 16- Pantalla de visualización/votación de una foto.
- 17- Pantalla de visualización/votación de una foto en pantalla completa.
- 18- Pantalla del perfil de un usuario. Pestaña seleccionada *TOP* fotos.
- 19- Pantalla del perfil de un usuario. Pestaña seleccionada *TODAS* las fotos.
- 20- Diálogo que solicita el método de subida de una foto.
- 21- Pantalla que visualiza la foto seleccionada.
- 22- Pantalla de selección del grupo al que se subirá la foto.
- 23- Pantalla de ayuda.

- **24-** Pantalla para ajustes de notificaciones.
- **25-** Pantalla de información de la cuenta.
- **26-** Pantalla acerca de la empresa.
- **27-** Pantalla FAQ.
- **28-** Pantalla de contacto.
- **29-** Pantalla para el cambio de contraseña.
- **30-** Pantalla eliminar cuenta.
- **31-** Diálogo para reactivar la cuenta.
- **32-** Pantalla de consulta/modificación del perfil propio del usuario.
- **33-** Pantalla para crear un grupo.

4.5.2. Manejador de Fragments para el Drawer

Como ya se ha definido en la sección 2.1.3, el **Drawer** muestra las principales pantallas de navegación, aquellas en las que el usuario deberá navegar con frecuencia o desde las cuales utilizará como punto de partida para otras partes de la aplicación. En este caso en particular, el **Drawer** viene representado mediante la pantalla 4 a partir de la cual se podrá acceder a las pantallas 5, 6, 7, 8, 9, 10 y 32.

Un usuario cuando lleva a cabo una navegación entre pantallas en una aplicación, desea conocer el camino que le ha llevado hasta dicha pantalla, para ello hay que guardar el recorrido por el cual ha pasado. No se ha utilizado la clase que ofrece Android, sino que se ha diseñado una propia que resuelve eficientemente el desplazamiento.

Se utiliza una pila como estructura de datos, en la cual se van almacenando los **Fragments** junto con el nombre de la pantalla que representan. El usuario logueado siempre deberá iniciar la navegación desde el **Drawer**, siendo éste el elemento principal que ocupa la última posición de la pila. Para cada nueva pantalla que acceda el usuario, se creará el **Fragment** que representa la pantalla y se apilará en la cima. Así sucesivamente para cada transición. Si el usuario desea volver a la pantalla anterior, deberá pulsar el botón **Back** y la clase desapilará el elemento de la cima, realizando las operaciones necesarias para mostrar en pantalla el posterior elemento cargado en la pila.

4.5.3. Adaptador

Android dispone de un objeto que actúa como puente entre un `AdapterView` (cuyos elementos se repiten) y los datos subyacentes de esa misma vista. El adaptador permite acceder a elementos de datos, es el encargado de crear una vista personalizada para cada elemento del conjunto de datos y además replica el comportamiento a todos ellos [5].

En el siguiente apartado se puede observar como en las pantallas 4, 5, 6, 7, 8, 11, 12, 13, 15, 18, 19 y 22 se hace uso de esta clase que suministra Android.

4.5.4. PagerAdapter

Clase base que proporciona el adaptador para llenar páginas de un `ViewPager` [48]. `PagerAdapter` es más general que los utilizados para `AdapterViews`. En lugar de proporcionar un mecanismo de reciclaje de sobre el `ViewPager` utiliza llamadas para indicar las medidas que se deben adoptar para actualizar la pantalla. `ViewPager` asocia a cada página una llave de tipo objeto en lugar de trabajar con `Views` directamente. De esta manera, la clave se utiliza para buscar e identificar de forma única una página determinada, independientemente de la posición que en el adaptador [11].

Incluido en la pantalla 17 donde las imágenes aparecen en tamaño grande acompañadas de cinco estrellas. Estas estrellas si se encuentran todas vacías el usuario podrá votar dicha foto, en caso contrario, indicarán la puntuación media de la foto. El usuario podrá votar más fotos deslizando el dedo de izquierda a derecha y viceversa.

4.5.5. FragmentStatePagerAdapter

Es una implementación de `PagerAdapter` que utiliza `Fragments` para gestionar cada página. Esta clase se encarga también de guardar y restaurar el estado de cada `Fragment` [9].

Esta versión específica de adaptador es muy útil cuando hay un gran número de pantallas con las que se está trabajando. Cuando las pantallas no son visibles para el usuario, éstas pueden ser destruidas y se guardará sólo el estado de ese `Fragment`. Permite reservar memoria a cambio de aumentar la potencia de procesamiento cuando se cambia entre pantallas.

La pantalla 16 es un reflejo de la pantalla 14 salvo que en la primera, una vez el usuario haya votado dicha foto, debajo de la misma aparecerán todas las puntuaciones que tiene dicha foto. Las votaciones vienen representadas mediante un `Fragment` que

se carga al instante de la votación. El usuario podrá votar diferentes fotos a la vez deslizando el dedo por la pantalla desde la izquierda a la derecha y viceversa.

Capítulo 5

Detalles de implementación

5.1. Introducción

En esta sección se describen las tecnologías, los entornos de trabajo y las herramientas utilizadas para llevar a cabo el desarrollo de la aplicación.

5.2. Tecnologías

5.2.1. Java

Java es un lenguaje de programación desarrollado originalmente por James Gosling de Sun Microsystems. Su sintaxis deriva de en gran medida de C y C++ pero con menos utilidades a bajo nivel que cualquiera de ellos dos. Las aplicaciones Java son compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura subyacente del computador.

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo [40].

El desarrollo de aplicaciones para Android se hace habitualmente con el lenguaje de programación Java y el conjunto de herramientas de desarrollo.

5.2.2. SDK Android

Es el kit de desarrollo de software para Android, incluye un conjunto de herramientas de desarrollo. Contiene bibliotecas, simuladores de teléfonos, depuradores de código, documentación, tutoriales y ejemplos de códigos. Éste kit está soportado para Linux, Mac y Windows. La plataforma integral de desarrollo que soporta este SDK es Eclipse con el complemento ADT (Android Development Tools plugin). Posee actualizaciones que coordinan el desarrollo general de Android y versiones antiguas que facilitan las pruebas de compatibilidad.

5.2.3. Symfony

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón MVC. Separa la lógica de negocios, la lógica de servidor y la presentación de las aplicaciones web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación y automatiza tareas comunes.

Symfony está desarrollado en PHP ^[30], es compatible con la mayoría de gestores de bases de datos y puede ser ejecutado en todas las plataformas.

En este proyecto la parte de **Back-End** de la aplicación ha sido desarrollada completamente por el compañero de trabajo, Dorin Brage. La aportación propia en esta parte ha sido la especificación de requisitos, diseño de la base de datos y las consultas de la aplicación para obtener datos.

5.2.4. XML

Es un lenguaje de marcas desarrollado por el World Wide Consortium (W3C) utilizado para almacenar datos en forma legible. Permite definir la gramática de lenguajes específicos para estructurar documentos grandes, propone un estándar para el intercambio de información estructurada entre diferentes plataformas y es útil cuando varias aplicaciones deben comunicarse entre sí o integrar información [35].

Android proporciona un vocabulario XML sencillo que se corresponde con las clases y subclases de View, así como Widgets y Layouts.

5.2.5. API Google Cloud Messaging

Es un servicio que permite enviar datos desde el servidor a los dispositivos Android, así como recibir mensajes en la misma conexión. El servicio GCM gestiona todos los

aspectos de la gestión de la cola de mensajes y de la entrega a sus destinatarios. Es completamente libre, no importa el tamaño de los mensajes y no hay límite en el número de mensajes enviados [10].

Registro del dispositivo

Para poder recibir notificaciones en el dispositivo Android primero éste se deberá registrar dicho servicio en el servidor GCM (Paso 1). Si el proceso finalizó correctamente, el servidor devolverá un identificador de registro ¹ (Paso 2). Una vez obtenido el token identificativo, la siguiente acción será enviarlo al servidor de la aplicación, de manera que éste lo pueda almacenar en su base de datos y le permita comunicarse con GCM para indicarle que tiene que enviarle una notificación (Paso 3).



Figura 5.1: Proceso de registro en Google Cloud Messaging.

Envío de la notificación

Se puede enviar una notificación push a cualquier dispositivo Android que se tenga registrado en la base de datos del propio servidor.

Para llevar a cabo este proceso, el servidor deberá saber a priori qué información deberá enviar y a qué destinatarios (Paso 1). El servidor cuando tiene preparado el mensaje envía una petición a GCM para que mande las notificaciones a los usuarios seleccionados (Paso 2). Una vez enviada la petición a GCM con las credenciales del servidor, la información de la notificación y los destinatarios, GCM enviará la notificación push a todos y cada uno de ellos.

¹Cadena alfanumérica con más de 160 caracteres.



Figura 5.2: Proceso de envío de notificación push utilizado Google Cloud Messaging.

5.2.6. JSON

Es un formato ligero para el intercambio de datos. Resulta fácil su lectura y escritura. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML [17].

JSON se construye siguiendo dos estructuras (ver código 4.2):

- Una colección de pares nombre-valor. En varios lenguajes de programación, se serializa un objeto, estructurada, diccionario, tabla hash, lista con clave o un vector asociativo.
- Una lista ordenada de valores.

5.3. Herramientas y entorno

5.3.1. Base de datos

La base de datos que se ha instalado y utilizado ha sido MySQL [21]. Se ha escogido porque funciona en diferentes plataformas, tiene muchas APIs disponibles, tiene un sistema de privilegios y contraseñas seguro, posee librerías optimizadas y rápidas y está ampliamente utilizada por un gran número de usuarios.

5.3.2. Control de versiones

El desarrollo del proyecto se ha llevado a cabo en paralelo por dos miembros del equipo, así que se ha utilizado un sistema de control de versiones. El sistema utilizado ha sido GitLab [15] porque ofrecía un repositorio remoto gratuito y privado para la gestión

del proyecto. Además se ha respaldado el proyecto utilizando el programa SourceTree [26] que gestiona gratuitamente los repositorios locales y remotos.

5.3.3. Editor de imágenes

Para la formación, manipulación, edición y retoque de imágenes se ha utilizado Adobe Photoshop [27]. Este programa es muy popular incluso fuera del ámbito informático ya que permite realizar un sinnúmero de modificaciones a las imágenes y en este caso, a la capa de presentación de la aplicación.

5.3.4. Entorno de desarrollo

El entorno utilizado para el desarrollo de la aplicación móvil ha sido Eclipse con el plugin ADT y el SDK de Android. Para la parte del servidor se ha utilizado NetBeans [19] configurado por SFTP con el servidor.

Eclipse

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar aplicaciones. Esta plataforma ha sido utilizada para desarrollar entornos de desarrollo integrado, como el IDE de Java llamado *Java Development Toolkit* y el compilador que se entrega como parte de Eclipse [34].

5.3.5. Servidor Web

Un servidor web es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y generando una respuesta en cualquier lenguaje o Aplicación del lado del cliente [48]. Se ha utilizado un servidor web HTTP Apache [3].

5.4. Estructura de ficheros

En este apartado se presenta la estructura interna de la aplicación, en base a la experiencia adquirida sobre la tecnología utilizada y según lo presentado en apartados anteriores.

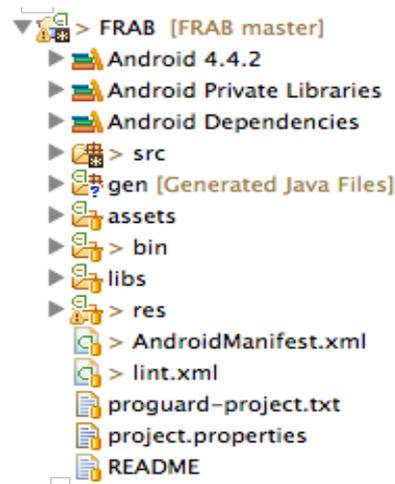


Figura 5.3: Estructura de ficheros en un proyecto Android.

5.4.1. Directorio src

Carpeta que contiene el código fuente de la aplicación. Los ficheros Java se almacenan en un espacio de nombres.

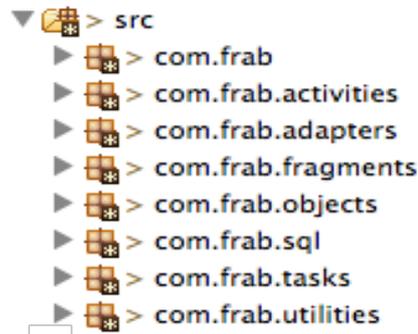


Figura 5.4: Directorio src de la aplicación.

La funcionalidad se distribuye en paquetes.

com.frab

En este paquete se encuentran las clases principales de *app*:

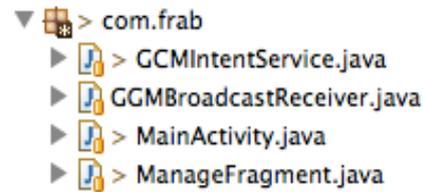


Figura 5.5: Paquete com.frab

- **GCMIntentService**: servicio destinado a la recepción y gestión de las notificaciones push.
- **GGMBroadcastReceiver**: broadcast encargado de despertar el tiempo necesario la aplicación antes de que termine la ejecución del servicio GCMIntentService.
- **MainActivity**: clase principal de la aplicación. Contiene un `FragmentPagerAdapter` que muestra las imágenes de presentación de la aplicación.
- **ManageFragment**: clase encargada de gestionar los `Fragments` del `Drawer`.

com.frab.activities

Este paquete recoge el resto de actividades de la aplicación:

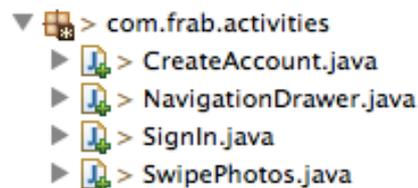


Figura 5.6: Paquete com.frab.activities

- **CreateAccount**: actividad encargada del registro de un nuevo usuario en el sistema.
- **NavigationDrawer**: actividad que contiene el `Drawer` desde donde el usuario accederá a todas las funcionalidades. Además contiene un `BroadcastReceiver` que es el encargado de capturar acciones que llegan desde `GCMIntentService`.
- **SignIn**: actividad supervisora del login del usuario.
- **SwipePhotos**: actividad encargada de mostrar las imágenes a tamaño completo pudiendo ser votadas. Contiene el adaptador `MyPagesPhotoAdapter`.

com.frab.adapters

Este paquete contiene los adaptadores que utilizan los Fragments:

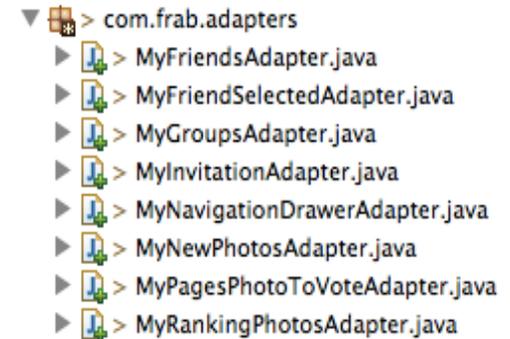


Figura 5.7: Paquete com.frab.adapters

- **MyFriendsAdapter**: utilizado en MyFriends. Muestra una imagen, el nombre del usuario y su puntuación media.
- **MyFriendSelectedAdapter**: utilizado en MyListOfFriends. Contiene una imagen, el nombre del usuario y un **CheckBox** para seleccionar el usuario.
- **MyGroupsAdapter**: utilizado en MyGroups. Contiene la imagen del grupo, el número de fotos, el número de miembros, el nombre del grupo y las foto nuevas que el usuario tiene por votar.
- **MyInvitationAdapter**: utilizado en MyInvitations. Contiene un texto y los botones de aceptar y cancelar la invitación.
- **MyNavigationDrawerAdapter**: utilizado en NavigationDrawer. Contiene el nombre de la pantalla a abrir y el número de elementos que existen en ella.
- **MyNewPhotosAdapter**: utilizado en NewPhotos. Contiene únicamente una imagen.
- **MyPagesPhotoToVoteAdapter**: utilizado en SwipePhotos. Contiene la foto, su título y las estrellas que indican su puntuación o para poder votarla.
- **MyRankingPhotosAdapter**: utilizado en RankingPhotos. Contiene una foto, su puntuación media y la posición que ocupa dentro del ránking.

com.frab.objects

Este paquete contiene objetos que son utilizados en los adaptadores una vez obtenidos los datos del servidor:



Figura 5.8: Paquete com.frab.objects

- **ItemNavigationDrawer**: utilizado por el adaptador MyNavigationDrawerAdapter.
- **NewPhotosOfGroup**: utilizado por el adaptador MyNewPhotosAdapter.
- **PhotoRanking**: utilizado por el adaptador MyRankingPhotosAdapter .
- **RowPhotoScore**: utilizado en UserRatingList. Referencia a la puntuación realizada por un usuario a una foto.

com.frab.sql

Paquete encargado de la gestión de la base de datos de la aplicación. Almacena la información necesaria con la que el usuario interactúa con mayor frecuencia.

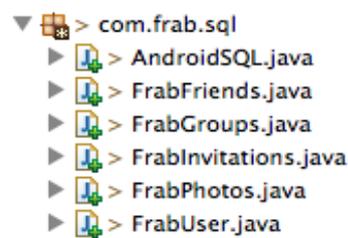


Figura 5.9: Paquete com.frab.sql

- **AndroidSQL**: implementa los métodos necesarios para la creación y actualización de la base de datos, así como la inserción, eliminación y modificación de las filas.
- **FrabFriends**: clase que representa un amigo del usuario logueado.

- **FrabGroups**: clase que contiene la información de un grupo.
- **FrabInvitations**: clase que desempeña la función de invitación.
- **FrabPhotos**: clase que posee la información de una foto.
- **FrabUser**: clase utilizada para obtener la información de usuarios.

com.frab.tasks

Este paquete contiene exclusivamente instancias de la clase `AsyncTask`. Esta clase permite realizar operaciones en segundo plano, como consultas al servidor, publicar los resultados en el hilo de la interfaz de usuario sin necesidad de manipular hilos y como beneficio no bloquea la ejecución de la aplicación.

com.frab.utilities

Paquete que contiene los diálogos, la gestión de la caché en disco de las imágenes, la API de conexión con el servidor, los estilos de las fuentes para las pantallas, las constantes utilizadas para el paso de información entre `Fragments` y `Activities` y funciones auxiliares.

com.frab.fragments

Este paquete contiene todos los `Fragments` de la aplicación. El usuario tendrá acceso a todos ellos mediante el `Drawer`. Cada uno de ellos aparece en el mapa de alcanzabilidad descrito en la sección 4.5.1:

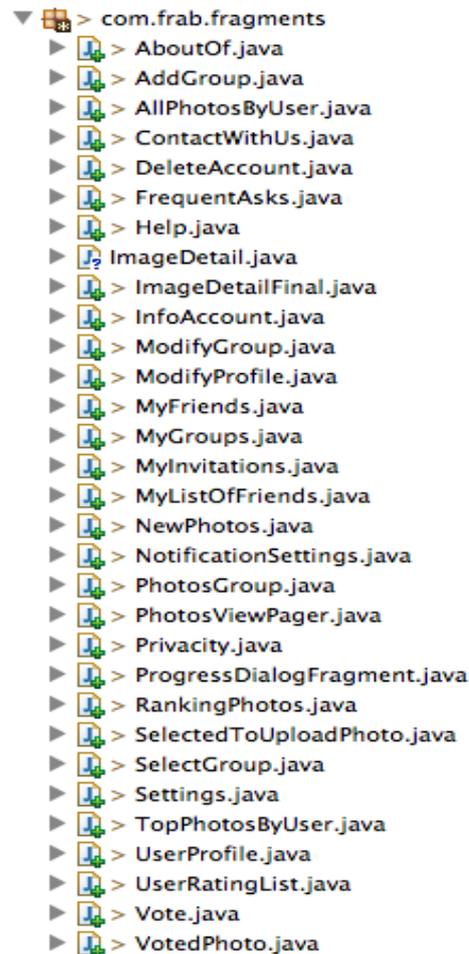


Figura 5.10: Paquete com.frab.fragments

- **AboutOf**: pantalla 26. Muestra información sobre la aplicación.
- **AddGroup**: pantalla 33. Crear grupo.
- **AllPhotosByUser**: pantalla 18. Muestra todas las fotos de un usuario. Utiliza el adaptador `MyRankingPhotosAdapter`.
- **ContactWithUs**: pantalla 28. Contiene un campo de texto y un botón donde el usuario escribe las dudas.
- **DeleteAccount**: pantalla 30. Contiene un campo de texto y un botón.
- **FrequentAsks**: pantalla 27. Contiene un `WebView` con un enlace FAQ de la web.
- **Help**: pantalla 9. Menú principal de ayuda.

- **ImageDetail**: pantalla de presentación de la aplicación.
- **ImageDetailFinal**: última pantalla de presentación de la aplicación.
- **InfoAccount**: pantalla 26. Muestra las opciones de *Privacidad* y *Eliminar cuenta*.
- **ModifyGroup**: pantalla 14. Muestra la información de un grupo.
- **ModifyProfile**: pantalla 32. Muestra el perfil del usuario logueado.
- **MyFriends**: pantalla 7. Muestra los amigos del usuario logueado.
- **MyGroups**: pantalla 5. Muestra los grupos que tiene un usuario.
- **MyInvitations**: pantalla 6. Muestra las invitaciones pendientes de un usuario..
- **MyListOfFriends**: pantalla 15. Muestra el listado de amigos que no pertenecen a un grupo o el listado de usuarios que pertenecen a un grupo y pueden ser eliminados.
- **NewPhotos**: pantalla 11. Utiliza un **GridView** con las fotografías por puntuar de un grupo.
- **NotificationSettings**: pantalla 24. Encargada de la gestión de las notificaciones.
- **PhotosGroup**: pantallas 11/12. Contiene dos pestañas donde carga los **Fragments** **NewPhotos** y **RankingPhotos**.
- **PhotosViewPager**: utiliza **FragmentStatePagerAdapter** para cargar las imágenes de un grupo y ver sus puntuaciones.
- **Privacy**: pantalla 29. Encargada de cambiar la contraseña del usuario.
- **RankingPhotos**: pantalla 12. Utiliza un **GridView** con las fotografías que ya han sido puntuadas.
- **SelectedToUploadPhoto**: pantalla 21. Contiene la imagen seleccionada para subirla a un grupo, además un campo de texto que indica el título de la foto y un botón.
- **SelectGroup**: pantalla 22. Muestra los grupos donde el usuario puede subir la foto.
- **Settings**: pantalla 9. Pantalla principal de acceso a los ajustes de la aplicación.
- **TopPhotosByUser**: pantalla 19. Contiene las 10 mejores fotos de un usuario.

- **UserProfile**: pantalla 18/19. Muestra el perfil de un usuario junto con todas sus fotos y las sus 10 mejores.
- **UserRatingList**: **Fragment** que tiene como función recuperar el listado de usuarios que han puntuado una foto. Utilizado en **VotedPhoto**.
- **Vote**: **Fragment** que muestra el texto *"VOTA Y VERÁS EL RESTO DE LOS VOTOS"*. Utilizado en **VotedPhoto**.
- **VotedPhoto**: pantalla 16. Utilizado para votar una imagen en tamaño reducido. Si un usuario vota una foto entonces cambiaría el **Fragment Vote** por el **Fragment UserRatingList**.

5.4.2. Directorio gen

Carpeta que contiene el código generado de forma automática por el SDK al compilar el proyecto. Dentro se encuentra:

- **BuildConfig.java**: define la constante **DEBUG** para que desde Java se conozca si la aplicación se encuentra en fase de desarrollo.
- **R.java**: define una clase que asocia los recursos de la aplicación con identificadores. De esta manera los recursos pueden ser accedidos desde Java.

5.4.3. Android X.X

Código JAR, el API de Android según la versión seleccionada.

5.4.4. Android Private Libraries

Librerías asociadas al proyecto.

5.4.5. Directorio assets

Carpeta que contiene ficheros o carpetas que pueden ser utilizadas en la aplicación como ficheros de datos o fuentes. No se puede modificar su contenido ni asociar ningún identificador.

5.4.6. Directorio bin

Esta carpeta contiene el código compilado y donde se genera el *.apk* que es el fichero comprimido que contiene la aplicación para ser instalada.

5.4.7. Directorio libs

Código JAR con las librerías que se usan en el proyecto. Automáticamente se añade la librería `android-support-vXX.jar` cuya función es permitir funcionalidades importantes dependiendo de la API mínima seleccionada.

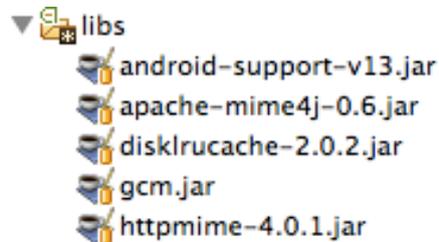


Figura 5.11: Librerías externas utilizadas en el proyecto.

- **android-support-v13**: clase utilizada para acceder a algunas características del paquete `android.app` a partir de la `api` 13 de manera que pueda ser compatible con versiones anteriores.
- **apache-mime4j-0.6**: ofrece un parser para la transmisión de streams en formato MIME.
- **disklrucache-2.0.2**: api de la caché de imágenes.
- **gcm**: api de `GoogleCloudMessaging` para las notificaciones push.
- **httpmime-4.0.1**: librería dirigida al intercambio, a través de internet, de todo tipos de archivos.

5.4.8. Directorio res

Carpeta que contiene todos los recursos que se utilizan en la aplicación. Esta carpeta contiene subcarpetas que tienen un sufijo que condiciona la carga de la aplicación dependiendo del dispositivo que se posea.

- **drawable**: contiene los ficheros de imágenes. Para definir diferentes recursos dependiendo de la resolución y densidad de la pantalla del dispositivo.
 - **drawable-ldpi**: densidad baja.
 - **drawable-mdpi**: densidad media.
 - **drawable-hdpi**: densidad alta.
 - **drawable-xhdpi**: densidad muy alta.

- **layout**: contiene los ficheros XML que corresponden a las vistas de la aplicación. Las vistas permiten configurar diferentes pantallas de la interfaz gráfica del usuario.
 - **layout**: orientación vertical.
 - **layout-land**: orientación horizontal.
- **menu**: ficheros XML con los menús de cada Fragment o Activity.
- **values**: permite indicar valores de tipo color, estilo o string. De esta manera se pueden cambiar valores sin tener que ser cambiados mediante código.
- **color**: contiene ficheros XML de definición de colores.

5.4.9. AndroidManifest.xml

Fichero que describe la aplicación. En él se reflejan las actividades, servicios y proveedores de contenido. Se declaran los permisos necesarios para su funcionamiento como la mínima versión de Android para poder ser ejecutada, el paquete principal de la aplicación y la versión Java, entre otros.

Capítulo 6

Pruebas

6.1. Pruebas de uso

En esta sección se explicará de forma guiada, todas las funcionalidades principales que ofrece la aplicación y la correcta forma para realizarlas.

6.1.1. Registro

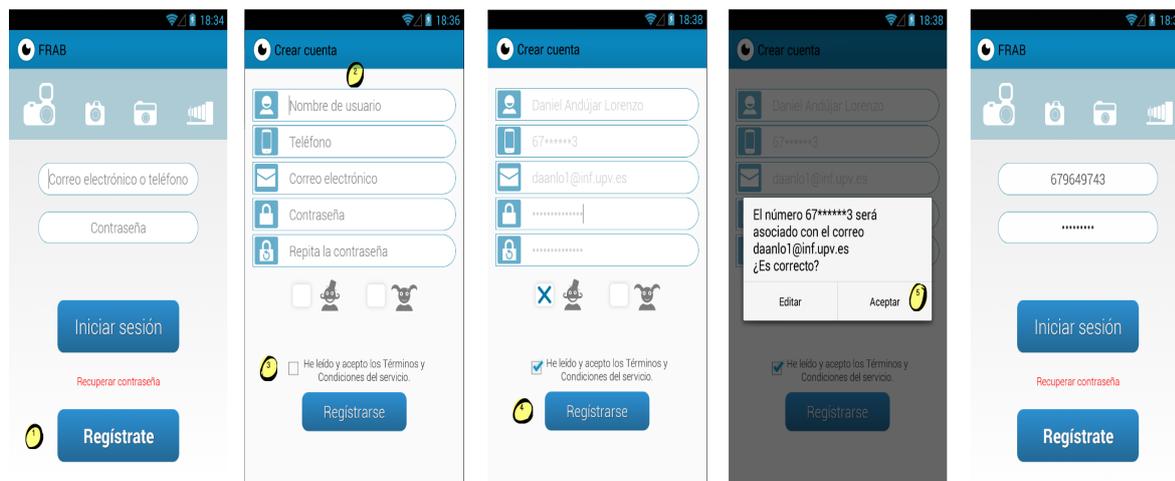


Figura 6.1: Registro de un nuevo usuario en el sistema.

Todo usuario que desee utilizar la aplicación, primero deberá registrarse en el sistema. Para ello, primero deberá iniciar la aplicación y situarse en la pantalla principal donde aparecerá el botón **Regístrate**, el cual deberá pulsar. Seguidamente se muestra un formulario con los datos del usuario: nombre de usuario, teléfono, correo electrónico, contraseña y sexo. Una vez completados los campos, el usuario deberá marcar la

opción de los términos y condiciones del servicio, pudiendo ser leídos si se pulsa sobre el mismo. Una vez terminado, se pulsa en Registrarse y la aplicación dará de alta al usuario.

6.1.2. Crear grupo

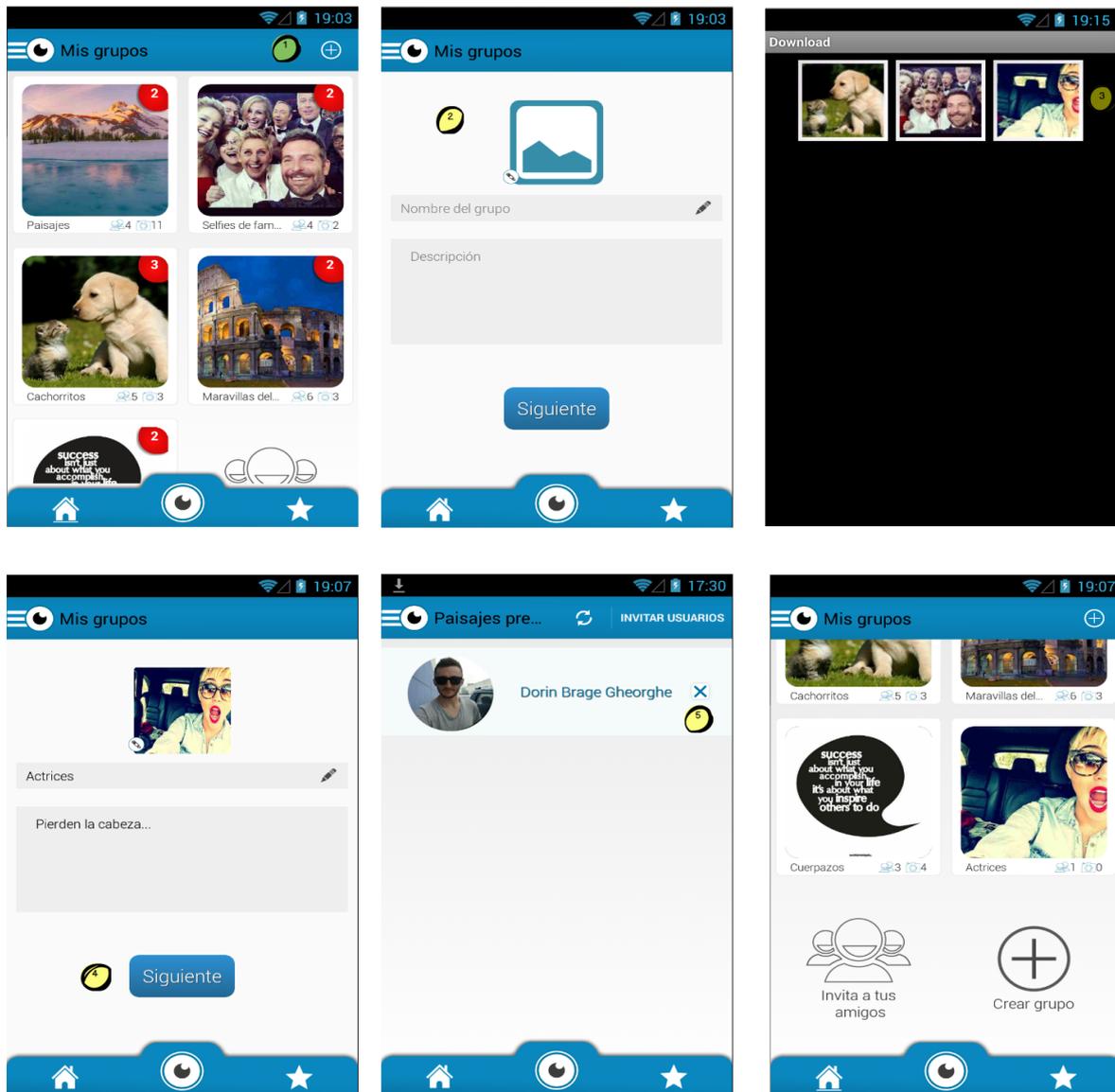


Figura 6.2: Registro de un nuevo usuario en el sistema.

Una vez iniciada la sesión, la pantalla principal que le aparecerá al usuario es la pantalla **Home**, donde aparecen todos los grupos que miembro. Si deseara crear uno nuevo, debería pulsar en el icono de la esquina superior derecha (símbolo \oplus). Seguidamente aparece una pantalla donde el usuario deberá introducir el título del grupo y la descripción del mismo. Para añadir una imagen como portada del mismo, pulsará sobre el icono de la montaña. La aplicación abrirá la galería de imágenes del teléfono donde el usuario deberá seleccionar una imagen. El siguiente paso será pulsar en el botón **Siguiente** y seleccionar a los usuarios a los que se quiere invitar al grupo. Por último, pulsar en **Invitar Usuarios** y el grupo ya se mostrará en la pantalla principal.

6.1.3. Aceptar/cancelar invitaciones

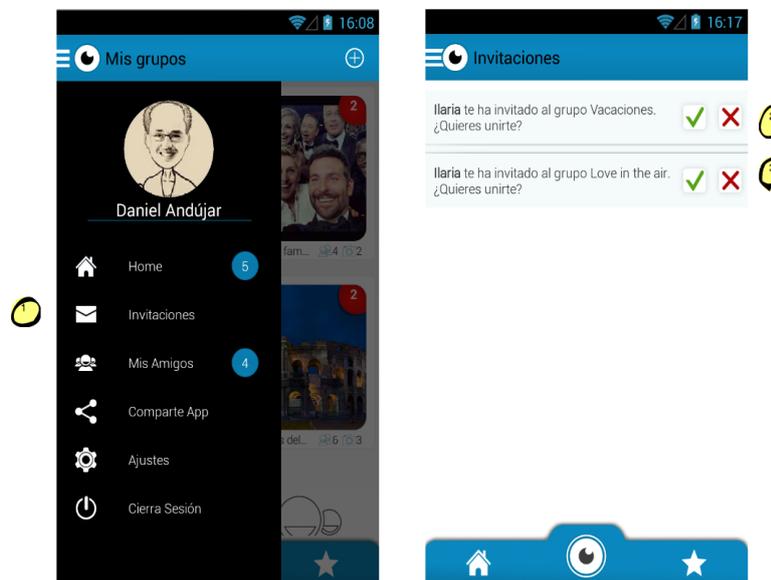


Figura 6.3: Responder a las invitaciones..

Un usuario puede tener invitaciones pendiente por responder, para acceder a ellas y responderlas, deberá abrir el **Drawer** y pulsar en la opción **Invitaciones**. La aplicación mostrará una pantalla con el listado de invitaciones pendientes de contestación. Para cada una de ellas se presenta como el *"El usuario X te invita al grupo Y. ¿Deseas unirte?"*. El usuario logueado deberá pulsar en \checkmark para aceptarla y en \times para cancelarla.

6.1.4. Visitar perfil del propio usuario

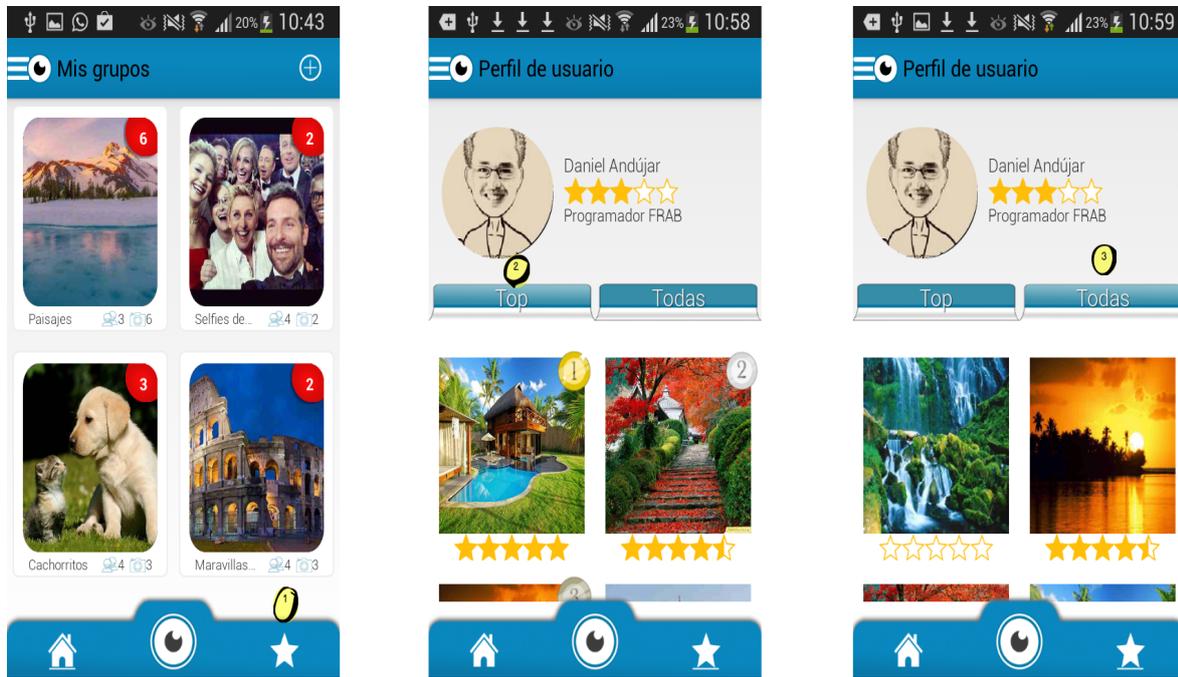


Figura 6.4: Perfil del usuario con sus fotos.

El usuario logueado (usuario que ha iniciado sesión), podrá ver todas sus fotos pulsando en el icono de la esquina inferior derecha, símbolo ✧. Dentro de la nueva pantalla, hay dos pestañas, en la primera aparecerán las 10 mejores fotos puntuadas por otros usuarios y en la otra pestaña, aparecerán todas las fotografías, incluso aquellas que no hayan sido puntuadas. Para visualizar una foto en tamaño grande, habrá que pulsar sobre una de ellas.

6.1.5. Subir fotos

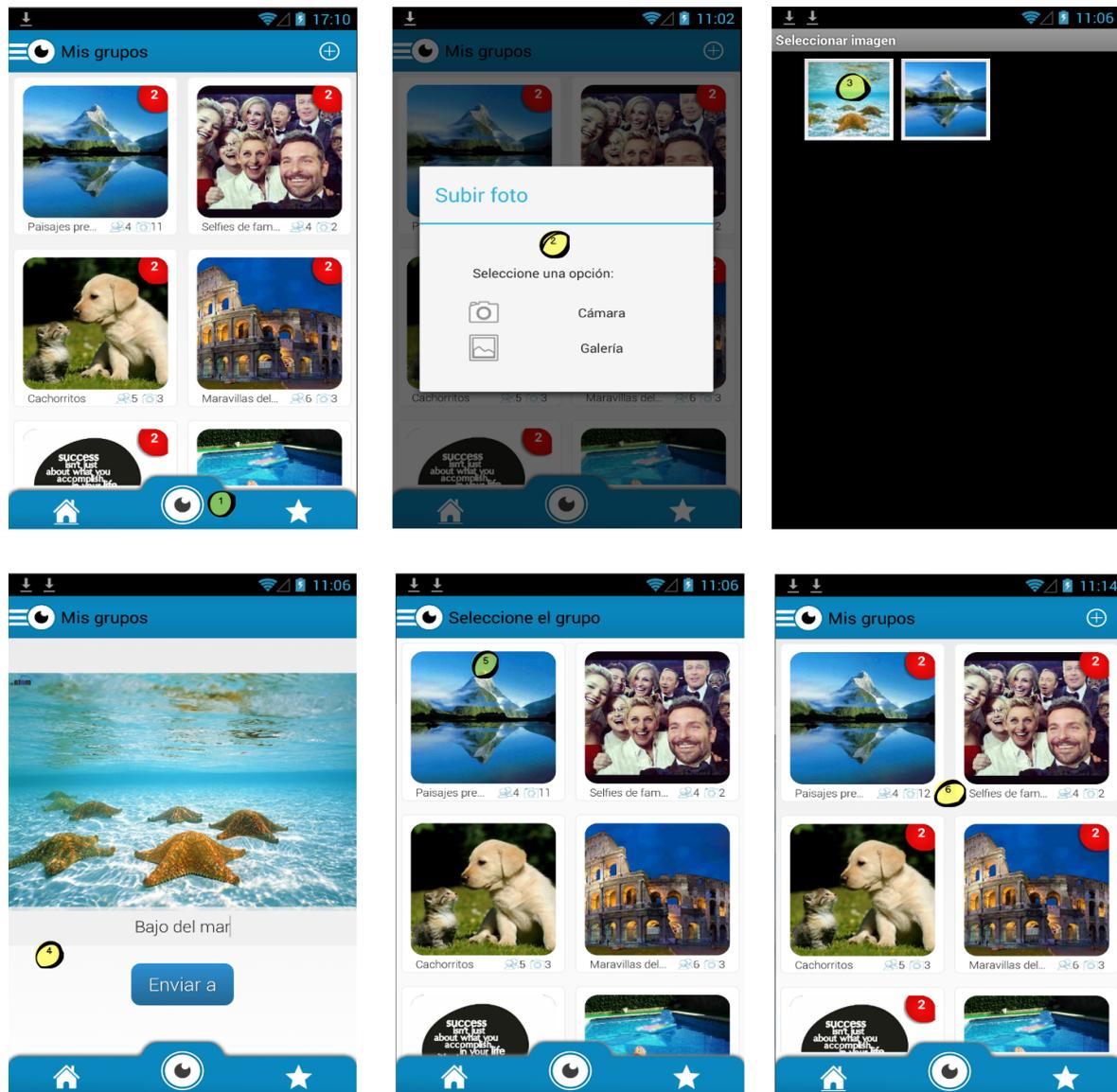


Figura 6.5: Subir fotografía a un grupo.

Para subir una fotografía a un grupo, el usuario deberá pulsar en el icono central de la barra inferior (icono del objetivo de una cámara). A continuación aparece un diálogo solicitando el método de subida de la imagen: seleccionar una imagen de la galería o tomar una foto de la cámara. Una vez seleccionada o tomada la foto, ésta última aparecerá en tamaño grande. Existe la posibilidad de introducir un título a la

foto. Seguidamente se debe pulsar en el botón **Enviar a** y seleccionar el grupo donde subir la foto.

6.1.6. Puntuar fotos

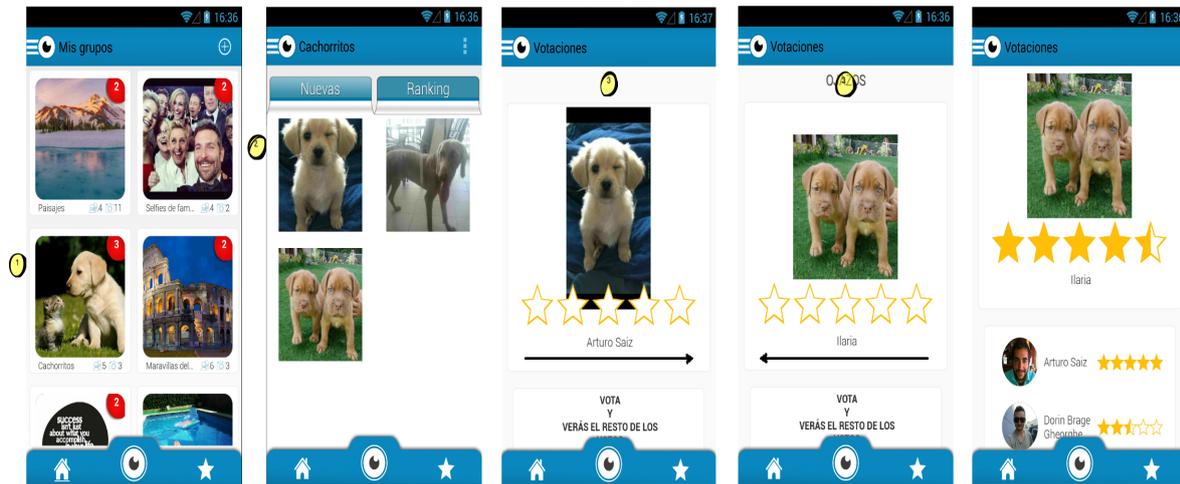


Figura 6.6: Votación de una fotografía.

Un usuario sabe que tiene fotos nuevas por puntuar porque en el pantalla Home los grupos puede tener un círculo minúsculo rojo con un número indicando la cantidad de fotos por votar. El usuario pulsa sobre el grupo que tiene fotos por puntuar, después debe pulsar sobre una fotografía y ésta aparecerá con un tamaño más grande acompañada de cinco estrellas en blanco. Para puntuarla se deberá seleccionar el número de estrellas que se considere como forma de voto. El usuario tiene la posibilidad de puntuar varias fotos a la vez únicamente deslizando el dedo desde la izquierda hacia la derecha o viceversa. El procedimiento de votación se repite para cada foto.

6.1.7. Enviar comentario

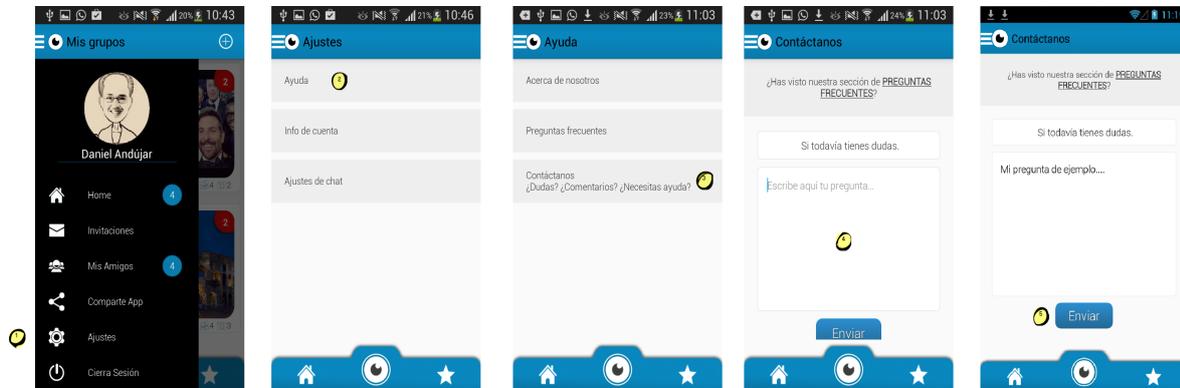


Figura 6.7: Enviar comentario al administrador del sistema.

El usuario puede enviar un comentario al administrador del sistema abriendo el Drawer y pulsando en Ajustes. Después deberá pulsar en Ayuda y a continuación en Contáctanos. La pantalla que muestra la aplicación tiene dos opciones: visualizar FAQ de la aplicación pulsando en Preguntas Frecuentes y escribir un comentario. El comentario debe escribirse donde dice "Escribe aquí tu pregunta...". Una vez terminado el escrito se deberá pulsar en el botón Enviar.

6.1.8. Cambiar contraseña

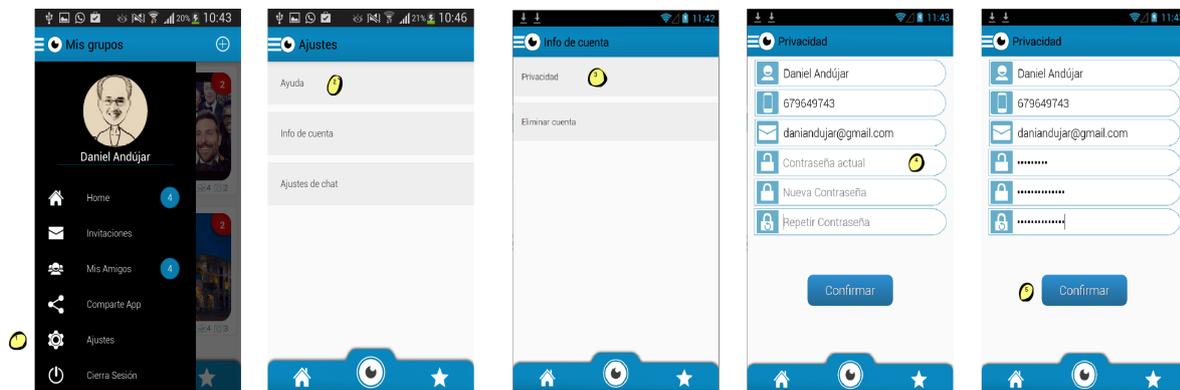


Figura 6.8: Pasos para cambiar la contraseña.

Para cambiar la contraseña de acceso de un usuario deberá abrir el Drawer y pulsar en Ajustes. A continuación pulsar en Ayuda y Privacidad. La pantalla resultante muestra la principal información de acceso del usuario. Éste deberá introducir en el apartado

correspondiente la actual contraseña y la nueva repitiéndola incluso una vez más. Para terminar, se debe pulsar en **Confirmar** y la contraseña de acceso habrá cambiado.

6.1.9. Eliminar cuenta

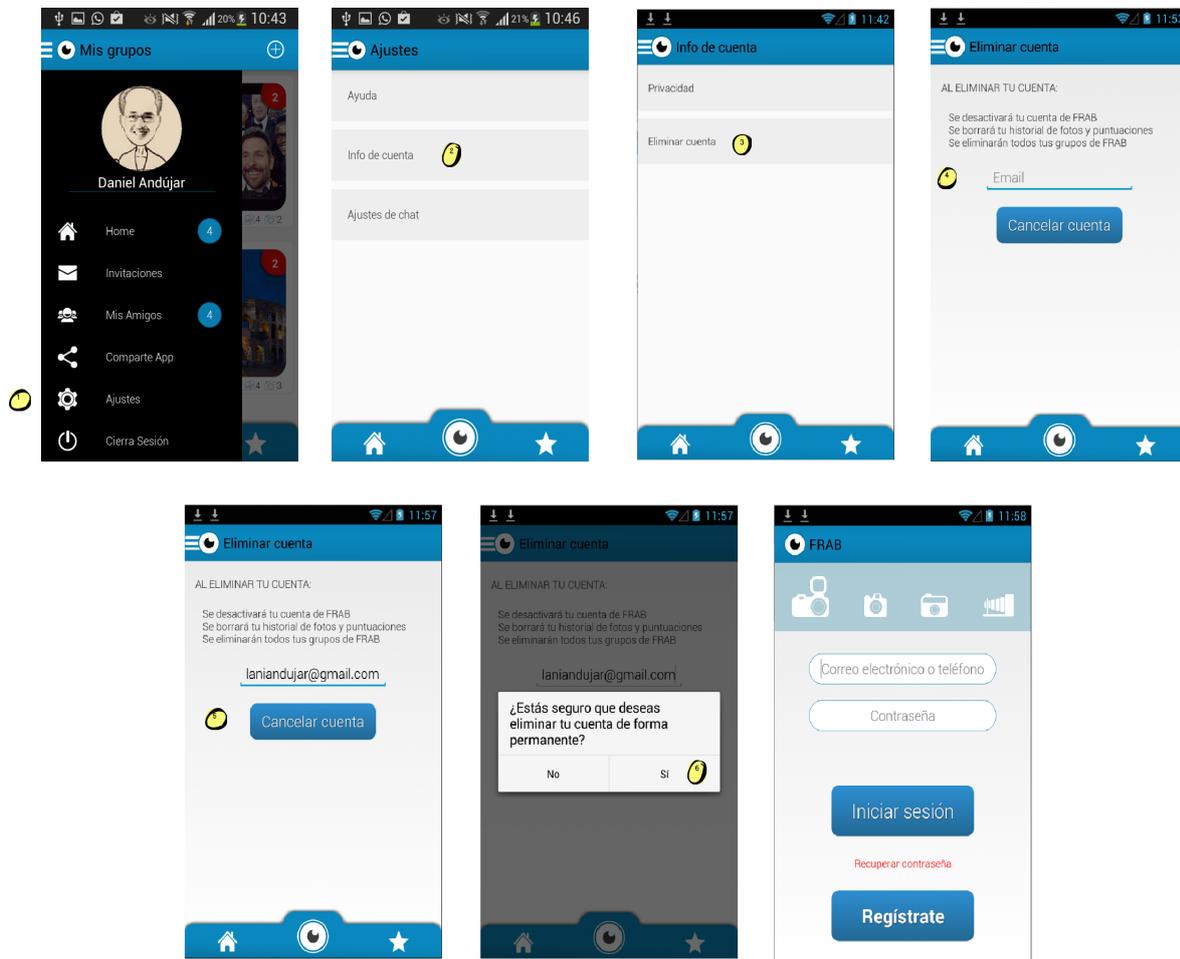


Figura 6.9: Instrucciones para eliminar la cuenta.

El usuario puede eliminar su cuenta en **Frab** accediendo desde el **Drawer** a **Ayuda**, pulsando posteriormente en **Info. de cuenta** y **Eliminar cuenta**. Situado en la pantalla para eliminar la cuenta, el usuario debe introducir su correo electrónico y aceptar el diálogo de confirmación. El sistema cerrará sesión automáticamente. Para reactivar la cuenta, se deberá iniciar sesión de nuevo y la aplicación realizará las acciones necesarias para permitir otra vez el acceso al usuario.

6.1.10. Modificar notificaciones

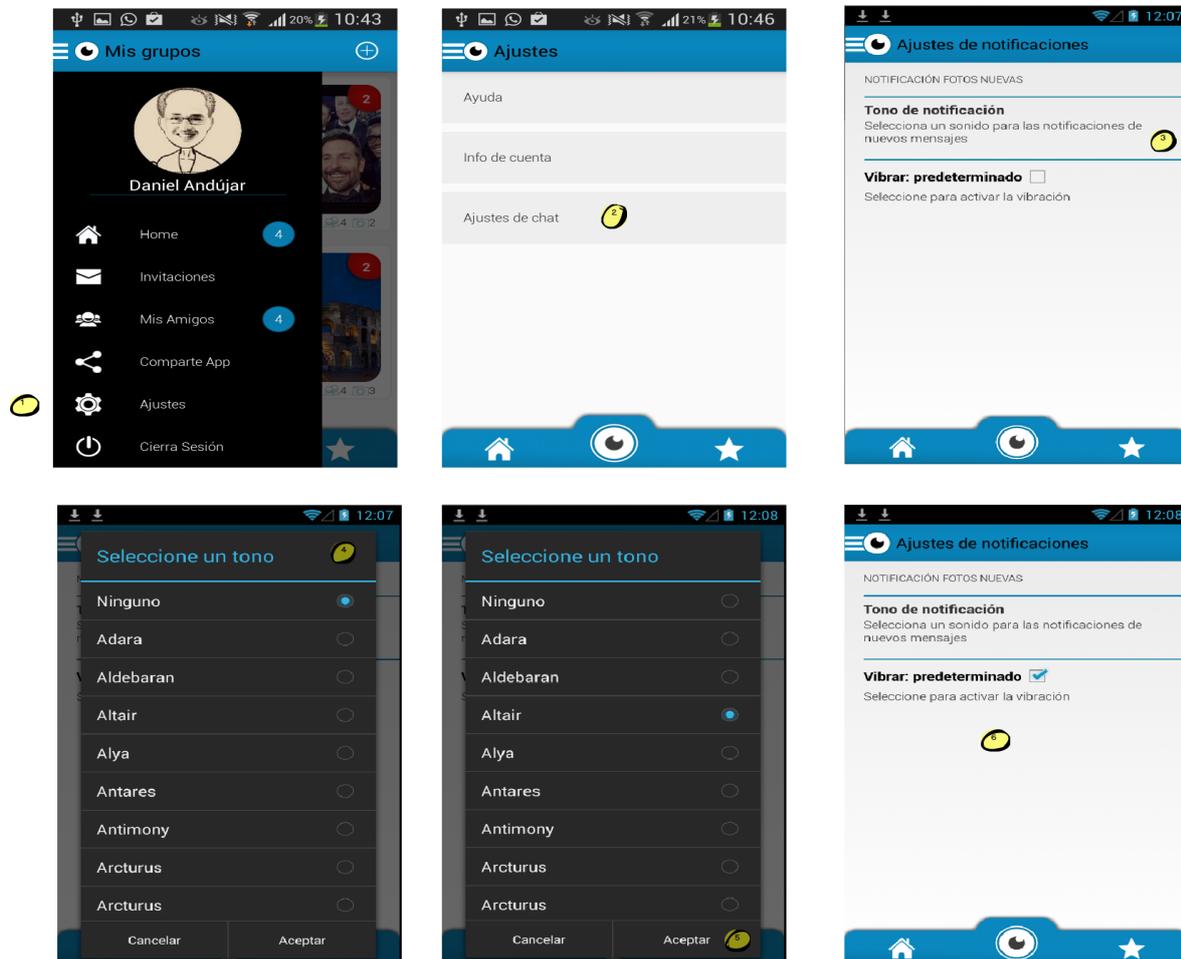


Figura 6.10: Cambiar el tono y activar o desactivar la vibración de las notificaciones.

Para personalizar las notificaciones que recibe un usuario, se deberá abrir el Drawer, pulsar en Ayuda y seguidamente en Ajustes de chat. La pantalla que muestra la aplicación es simple:

- Para cambiar el tono de notificación se deberá pulsar en Tono de notificación y de la lista de tonos que aparecen, seleccionar el que se quiera.
- Para activar/desactivar la vibración habrá que marcar/desmarcar la casilla de vibración.

6.1.11. Salir de un grupo

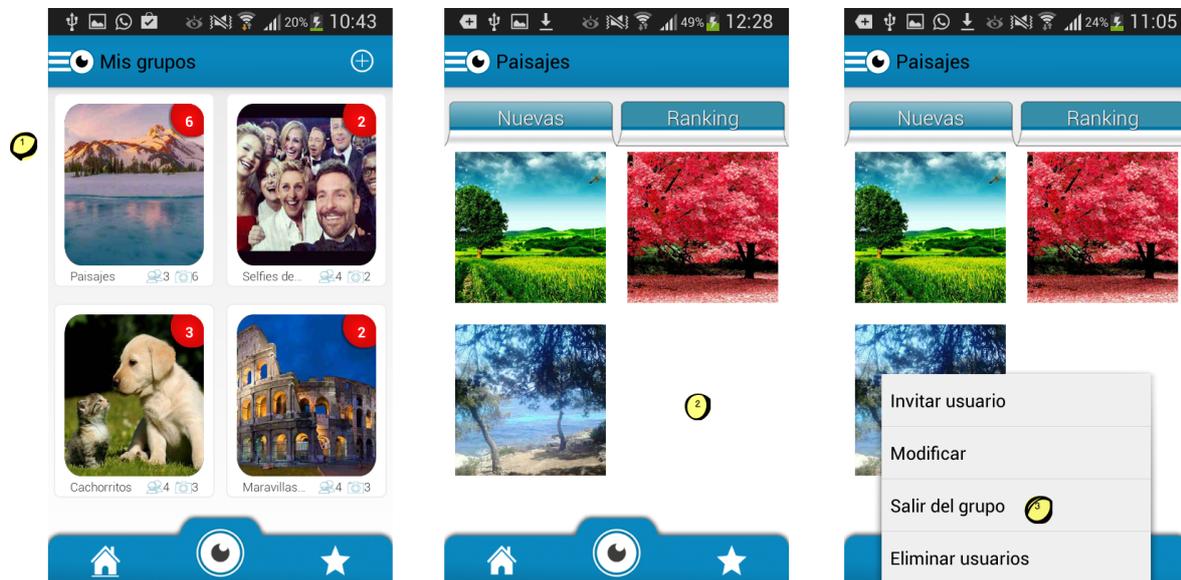


Figura 6.11: Abandonar un grupo.

Para salir de un grupo, el usuario deberá seleccionar desde la pantalla Home el grupo en cuestión. Una vez dentro del mismo, el usuario tiene que pulsar el botón de opciones del dispositivo y seleccionar la opción Salir del grupo.

6.1.12. Invitar un usuario a un grupo

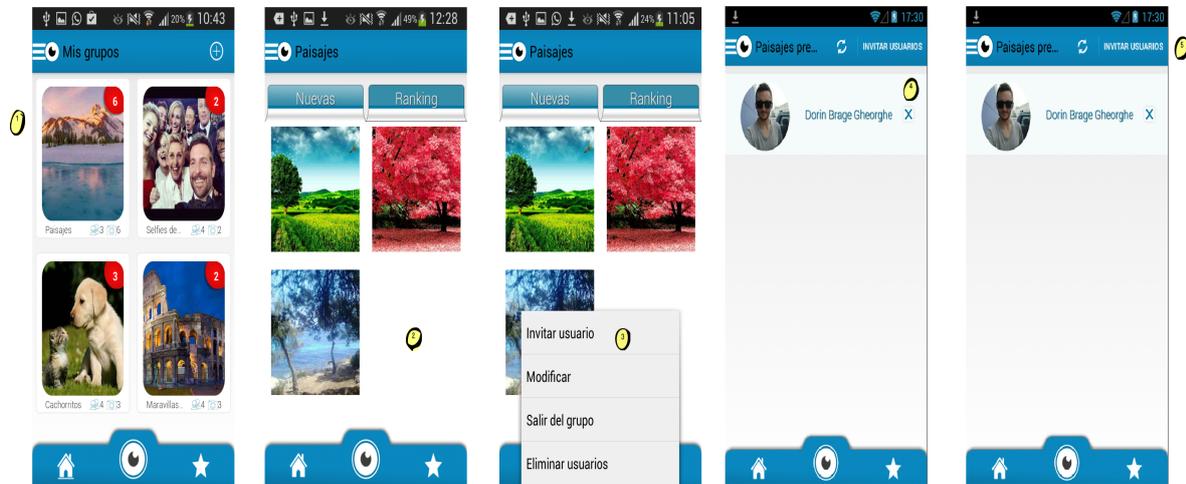


Figura 6.12: Invitar usuario a un grupo.

Para invitar nuevos usuarios a un grupo ya creado, el usuario deberá acceder al mismo y pulsar el botón de opciones de teléfono. Seguidamente pulsar en Invitar usuario. De la lista de usuarios que se presentan, seleccionar a los que se quiera invitar y pulsar en Invitar usuarios.

6.1.13. Eliminar usuarios de un grupo

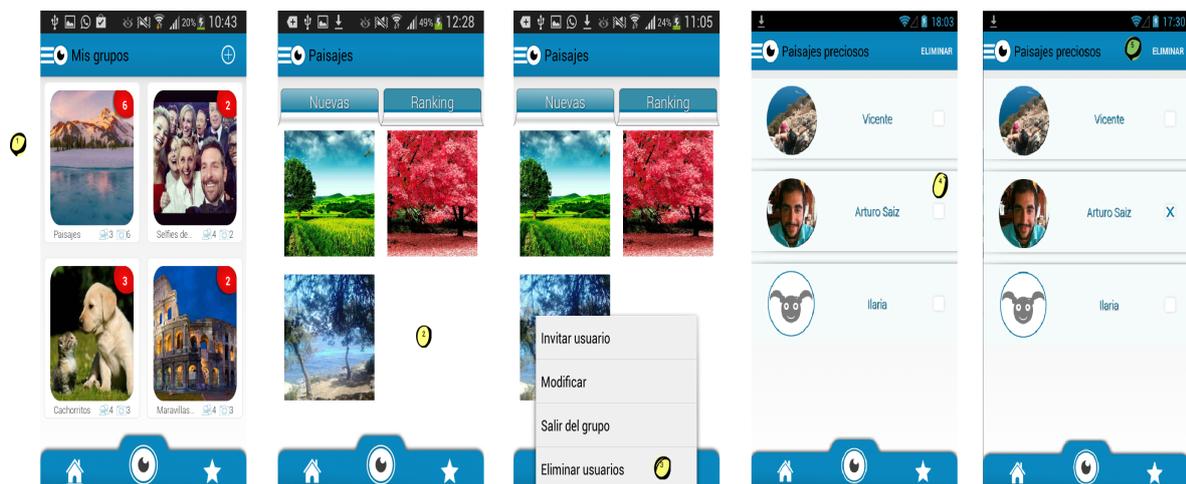


Figura 6.13: El administrador del grupo podrá eliminar usuarios de ese grupo.

Únicamente el administrador del grupo puede realizar esta acción. Se repite el mismo proceso que en el apartado 6.1.12 pero esta vez hay que pulsar en Eliminar usuarios y posteriormente en Eliminar cuando los usuarios ya hayan sido seleccionados.

6.1.14. Modificar grupo

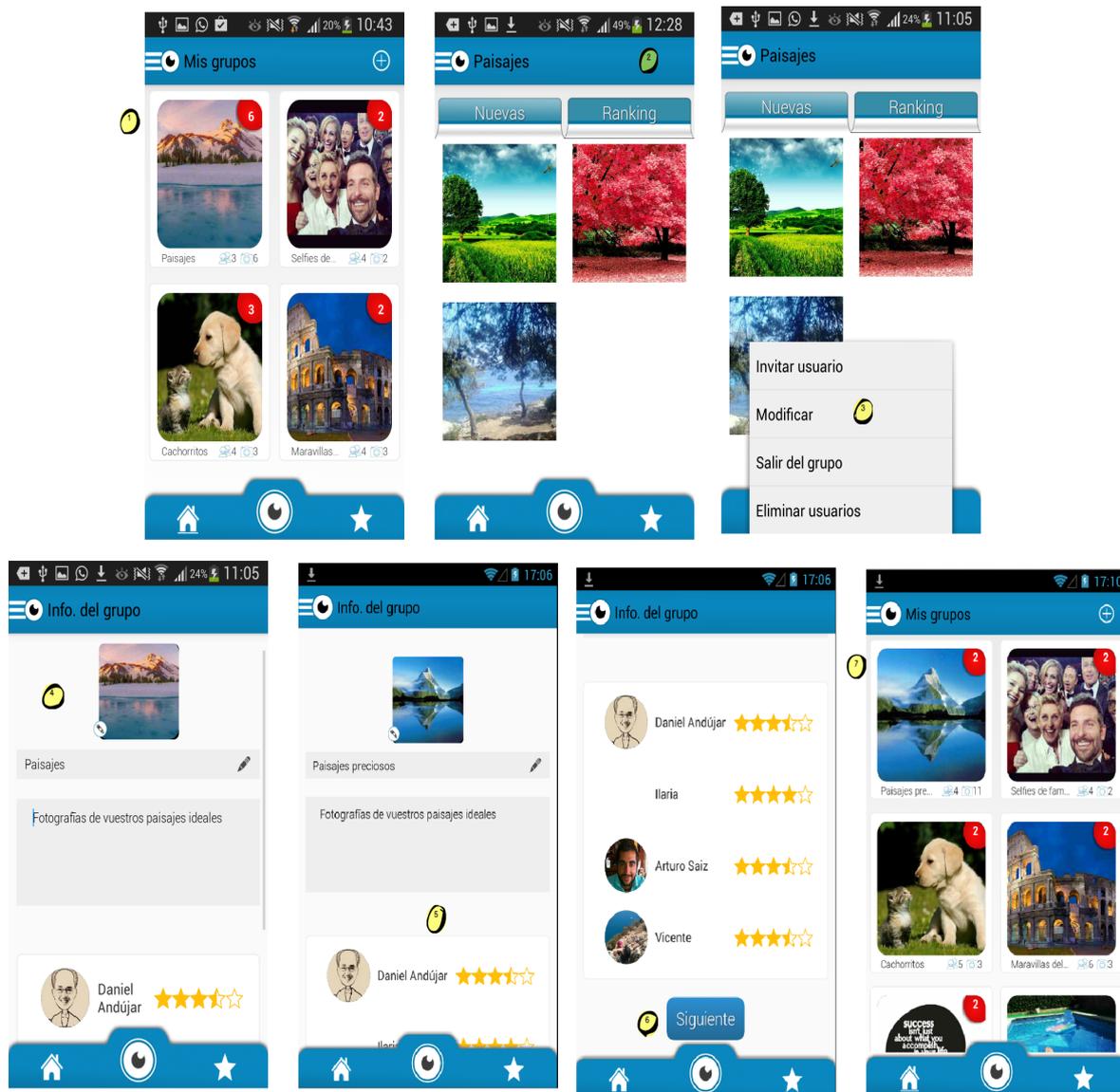


Figura 6.14: Perfil de un usuario de un grupo en común.

Un usuario puede cambiar el título y la imagen de un grupo accediendo al mismo y pulsando en el botón de opciones del teléfono seleccionando la opción **Modificar**. Si se quiere cambiar la imagen tiene que pulsar en la imagen para seleccionar una de la galería y para cambiar el título, pulsando en el título, parecerá el teclado para modificar el mismo. Una vez terminado se debe hacer *scroll* hasta situarse en la parte inferior de la pantalla y pulsar en el botón **Siguiente**. Seguidamente aparecerá la pantalla Home con el grupo modificado.

6.1.15. Ver perfil usuario miembro de un grupo

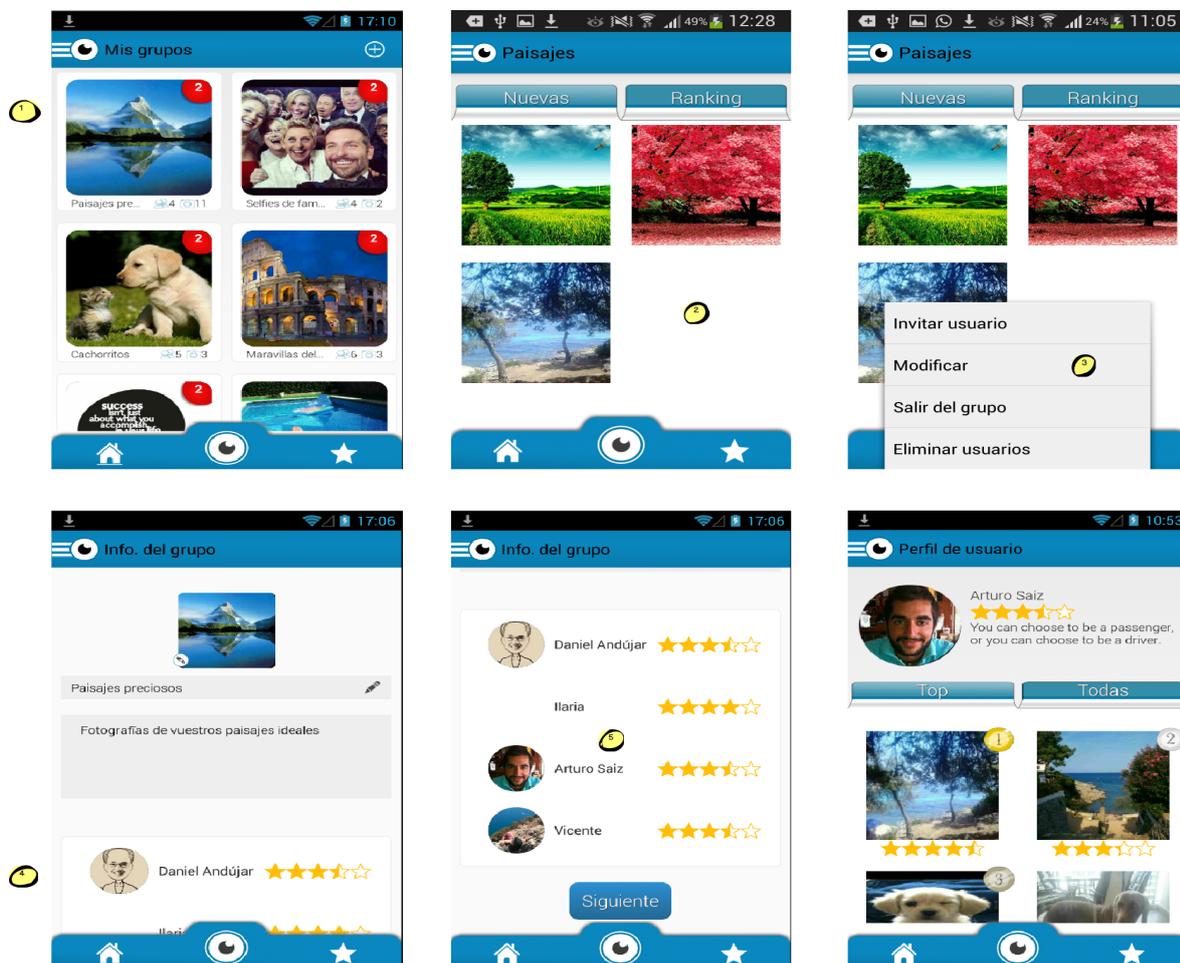


Figura 6.15: Perfil de un usuario de un grupo en común.

Un usuario puede ver las fotos de otro usuario que está en el mismo grupo accediendo a ese grupo, pulsando la tecla de opción y después en **Modificar**. En la pantalla que se

muestra se hace *scroll* hasta encontrar el usuario que se está buscando. Sólo se tendrá que pulsar sobre el mismo y se tendrá acceso a sus fotografías. Recordar que sólo se podrán visualizar aquellas que se encuentren en grupos en común entre los dos usuarios.

6.2. Resoluciones de pantalla

Android se ejecuta en una amplia variedad de dispositivos que tienen diferentes tamaños de pantallas y densidades. Android proporciona APIs que permiten controlar la interfaz de usuario de la aplicación para los tamaños y densidades de pantalla específicos, con el fin de optimizar el diseño para diferentes configuraciones de pantalla [12].

Por comodidad Android agrupa los dispositivos por tamaños de pantalla (*small*, *normal*, *large* y *extra large*) y por densidades¹ (*ldpi*, *mdpi*, *tvdpi*, *hdpi*, *xhdpi* y *xxhdpi*).

La aplicación está diseñada para dos tipos de pantalla, *small screen* y *normal screen*. Según refleja la figura 2.1, se podría llegar a un 87,4% del mercado basándose en el tamaño y al 83% basándose en las densidades de la pantalla.

6.2.1. Small screen

Estas pantallas tienen un tamaño de $470dp \times 320dp$.

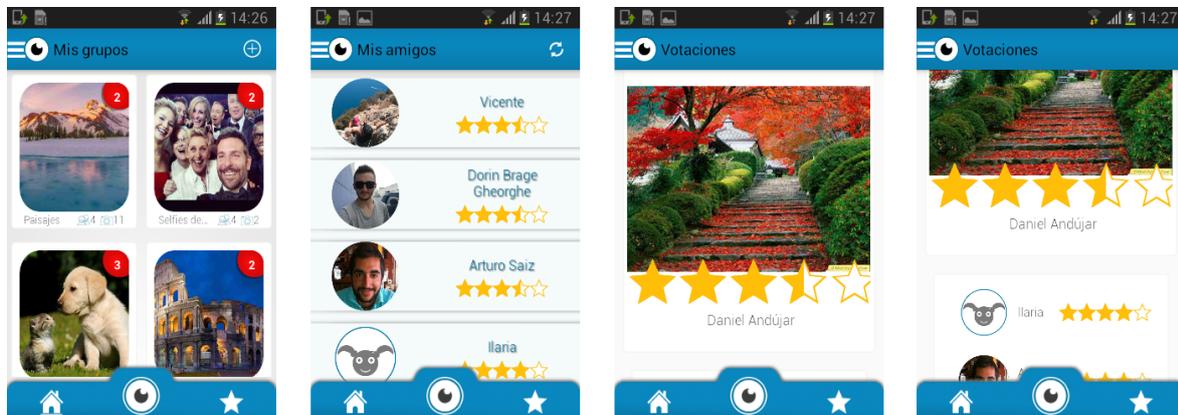


Figura 6.16: Capturas de pantalla en una pantalla de tamaño pequeño.

Se aprecia como faltan elementos por visualizar, teniendo que hacer *scroll* el usuario para ver todo el contenido.

¹Cantidad de píxeles dentro de una área física de la pantalla

6.2.2. Normal screen

Estas pantallas tienen un tamaño de 426dp x 320dp.

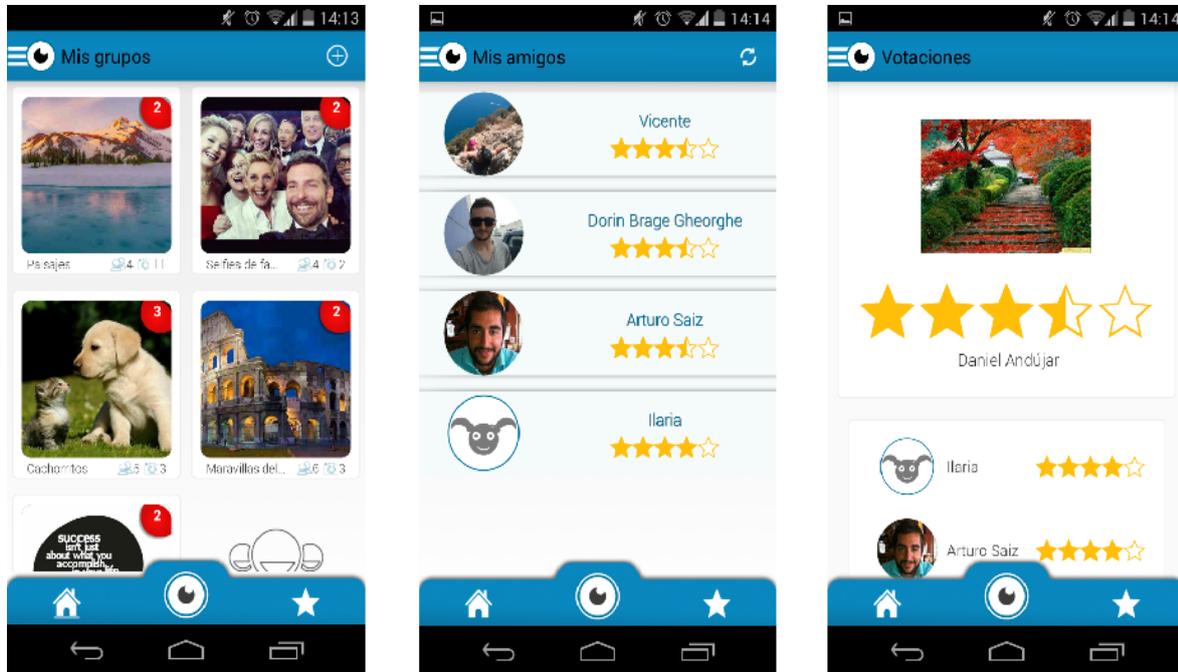


Figura 6.17: Capturas de pantalla en una pantalla de tamaño normal.

Capítulo 7

Conclusiones

En esta sección se presentan las conclusiones a las que se ha llegado después de realizar este proyecto final de carrera.

7.1. Técnicas

Se ha podido comprobar que los conocimientos adquiridos durante estos cinco años de universidad han servido, empezando desde cero, para realizar una correcta gestión del proyecto, una estructura de ficheros adecuada para el sistema que se requería y se han aplicado metodologías de *"buenas prácticas"* para conseguir rapidez y obtener eficiencia en el procesamiento de datos. Además se han obtenido conocimientos en la programación de dispositivos móviles Android, a trabajar utilizando un control de versiones y a planificar las tareas.

Ha supuesto un reto la búsqueda continua de información a través de páginas de desarrolladores, como la integración de diferentes apis y su modificación para adaptarlas a las necesidades requeridas del sistema.

Hacer hincapié en las posibilidad y facilidades que ofrece Android para cambiar aspectos de diseño de la aplicación pudiendo realizar innumerables variaciones, donde todo es modificable y adaptable a las necesidades existentes. En su contra mencionar la alta fragmentación con la que hay que combatir adaptando la *app* a todo tipo de terminales.

7.2. Personales

Destacar claramente el trabajo en equipo, que no siempre es perfecto y efectivo, ya que se ha tenido que debatir y discutir sobre ciertos aspectos de diseño. Se ha aprendido a ser paciente, a escuchar las opiniones e ideas del resto de miembros, a coordinarse para finalizar tareas y de este modo cumplir los tiempos de entrega.

Por otra parte se ha aprendido de los errores cometidos en el transcurso del proyecto, tanto en la programación y estructura de ficheros de Android como en la base de datos, desperdiciando trabajo realizado durante días porque se requerían o sobraban más datos de los estrictamente necesarios. Gracias a ello, en futuros proyectos se evitará cometer estos errores y si vuelven a ocurrir se solucionarán con mayor brevedad.

7.3. Agradecimientos

Agradecer al equipo de **Frab** la confianza depositada en mí para la realización de este proyecto aun sabiendo desde el principio mi desconocimiento práctico sobre el desarrollo de aplicaciones móviles.

A mi compañero de trabajo Dorin Brage, valoro y aprecio la paciencia, tu saber hacer y saber estar durante estos tres meses de aprendizaje.

A mi tutor del proyecto, Sergio Sáez, agradecerle el seguimiento que ha realizado a este proyecto y también por guiarme correctamente para mejorar el funcionamiento del mismo.

A mis dos compañeros y amigos de facultad, Tomás y Dani, recordarles que en gran parte estoy aquí gracias a vosotros. No os olvidaré.

A mi pareja Ilaria que me ha soportado durante estos dos últimos años en los buenos y malos momentos, gracias, eres única.

Para finalizar, hacer gran mención a mis padres. Simplemente gracias por todo, sin vosotros esto no hubiese sido posible.

Capítulo 8

Referencias bibliográficas

- [1] Adobe. Adobe Photoshop CS6. <http://www.adobe.com/es/products/photoshop.html>, may 2012.
- [2] Androideity. Arquitectura de Android, <http://androideity.com/2011/07/04/arquitectura-de-android/>, jul 2014.
- [3] Apache versión 2.4.7. <http://httpd.apache.org/>, ago 2014.
- [4] Desarrolladores web. Ciclo de vida de una actividad, <http://www.desarrolloweb.com/articulos/ciclo-vida-actividad-aplicacion.html>
- [5] Developer Android. Adaptador. <http://developer.android.com/reference/android/widget/Adapter.html>, ago 2014.
- [6] Developer Android. Android SDK versión 22.6.2, <http://developer.android.com/sdk/index.html>, mar 2014.
- [7] Developer Android. Dashboards. <https://developer.android.com/about/dashboards/index.html>, ago 2014.
- [8] Developer Android. Drawer. <http://developer.android.com/training/implementing-navigation/nav-drawer.html>
- [9] Developer Android. Fragment State Pager Adapter. <http://developer.android.com/reference/android/support/v4/app/FragmentManager.html>, ago 2014.
- [10] Developer Android. Google Cloud Messaging. <http://developer.android.com/google/gcm/index.html>, ago 2014.
- [11] Developer Android. Pager Adapter. <http://developer.android.com/reference/android/support/v4/view/PagerAdapter.html>, ago 2014.

- [12] Developer Android. Screens support. http://developer.android.com/guide/practices/screens_support.html.
- [13] Developer Android. Storage Options. <http://developer.android.com/guide/topics/data/data-storage.html>, ago 2014.
- [14] Github. JakeWharton (DiskLruCache). <https://github.com/JakeWharton/DiskLruCache>, jul 2014.
- [15] Gitlab, open source software to collaborate on code, versión 7.1, <https://about.gitlab.com/>, jul 2014.
- [16] *IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998, 1998.*
- [17] JSON. JavaScript Object Notation. <http://json.org/>, nov 2011.
- [18] *Ley Orgánica 15/1999, de Protección de Datos de Carácter Personal. Sección I. Disposiciones generales. Jefatura del Estado. Ref: BOE-A-1999-23750. 13 diciembre 1999*
- [19] NetBeans IDE versión 8.0. <https://netbeans.org/>, mar 2014.
- [20] OMG. Unified Modeling Language versión 2.5 <http://www.uml.org/>, oct 2012.
- [21] Oracle Corporations. MySQL versión 5.5.37. <http://www.mysql.com/>, mar 2014.
- [22] Oxford Dictionaries. <http://www.oxforddictionaries.com/definition/english/app>
- [23] RAE. Diccionario de la Real Academia Española de la Lengua. Edición 22 <http://lema.rae.es/drae/>
- [24] *Real Decreto 1720/2007, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre 1999, de protección de datos de carácter personal. Disposiciones generales. Sección I. Ministerio de Justicia. Ref: BOE-A-2008-979. 19 enero 2008*
- [25] SEOSEO7. Landing Page. <http://seoseo7.es/2012/01/30/que-es-una-landing-page/>, ene 2012.
- [26] SourceTree a free Git & Mercurial client, versión 1.9.6. <http://www.sourcetreeapp.com/>, may 2014.
- [27] Wikipedia. Adobe Photoshop. http://es.wikipedia.org/wiki/Adobe_Photoshop, jul 2014.

- [28] Wikipedia. Android. <http://es.wikipedia.org/wiki/Android>, ago 2014.
- [29] Wikipedia. Application Programming Interface. http://es.wikipedia.org/wiki/Interfaz_de_programación_de_aplicaciones, jun 2014.
- [30] Wikipedia. Autenticación. <http://es.wikipedia.org/wiki/Autenticación>, jun 2014.
- [31] Wikipedia. Base de datos. http://es.wikipedia.org/wiki/Base_de_datos, jul 2014.
- [32] Wikipedia. Caché, [http://es.wikipedia.org/wiki/Caché_\(informática\)](http://es.wikipedia.org/wiki/Caché_(informática)), jul 2014.
- [33] Wikipedia. Caso de uso. http://es.wikipedia.org/wiki/Caso_de_uso, jul 2014.
- [34] Wikipedia. Eclipse (Software). [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software)), jul 2014.
- [35] Wikipedia. Extensible Markup Language, http://es.wikipedia.org/wiki/Extensible_Markup_Language, jul 2014.
- [36] Wikipedia. Fotografía. <http://es.wikipedia.org/wiki/Fotografía>, ago 2014.
- [37] Wikipedia. Front and back ends. http://en.wikipedia.org/wiki/Front_and_back_ends, ago 2014.
- [38] Wikipedia. iOS. <http://es.wikipedia.org/wiki/IOS>, ago 2014.
- [39] Wikipedia. Iterative and incremental development. http://en.wikipedia.org/wiki/Iterative_and_incremental_development, jul 2014.
- [40] Wikipedia. Java (lenguaje de programación). [http://es.wikipedia.org/wiki/Java_\(lenguaje_de_programación\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programación)), ago 2014.
- [41] Wikipedia. Login. <http://es.wikipedia.org/wiki/Login>, jul 2014.
- [42] Wikipedia. Modelo Vista Controlador. <http://es.wikipedia.org/wiki/Modelo-vista-controlador>, ago 2014 .
- [43] Wikipedia. Php. <http://es.wikipedia.org/wiki/.php>, jul 2014.
- [44] Wikipedia. Push. http://es.wikipedia.org/wiki/Tecnología_Push, may 2014
- [45] Wikipedia. Requisitos funcionales. http://es.wikipedia.org/wiki/Requisito_funcional, ago 2014.

- [46] Wikipedia. Requisitos no funcionales. http://es.wikipedia.org/wiki/Requisito_no_funcional, jun 2014.
- [47] Wikipedia. SDK. http://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software, abr 2014.
- [48] Wikipedia. Servidor web. http://es.wikipedia.org/wiki/Servidor_web, ago 2014.
- [49] Wikipedia. Sistema de Gestión de Bases de Datos. http://es.wikipedia.org/wiki/Sistema_de_gestión_de_bases_de_datos, ago 2014.
- [50] Wikipedia. Social Network. http://en.wikipedia.org/wiki/Social_network, jul 2014.
- [51] Wikipedia. SQLite. <http://es.wikipedia.org/wiki/SQLite>, abr 2014.
- [52] Wikipedia. Structured Query Language. [http://es.wikipedia.org/wiki/SQL_\(informática\)](http://es.wikipedia.org/wiki/SQL_(informática)), jul 2014.