



PicDay

**PicDay**

Elena Kaloya-nova Popova y  
Andrés Plaza Hernando

# Índice general

|  |           |
|--|-----------|
| <b>Índice general</b>                          | <b>2</b>  |
| <b>1 Manual de Instalación</b>                 | <b>3</b>  |
| 1.1. Requisitos mínimos . . . . .              | 3         |
| <b>2 Manual de Usuario</b>                     | <b>5</b>  |
| 2.1. Resumen de la aplicación . . . . .        | 5         |
| 2.2. Funcionalidades . . . . .                 | 5         |
| <b>3 Diseño</b>                                | <b>7</b>  |
| 3.1. Casos de uso . . . . .                    | 7         |
| 3.2. Diseño de la arquitectura . . . . .       | 8         |
| 3.3. Diagramas de secuencia . . . . .          | 16        |
| <b>4 Consideraciones técnicas</b>              | <b>26</b> |
| <b>5 Metodología de trabajo y conclusiones</b> | <b>39</b> |

# Capítulo 1

# Manual de Instalación

Manual orientado a los aspectos más técnicos de la aplicación.

## 1.1. Requisitos mínimos

A continuación se muestran los distintos requisitos para la utilización de la aplicación.

### Versión de Android mínima

Se requiere el nivel de API mínima de 22, correspondiente a la versión 5.1 Marshmellow.

### Permisos requeridos

La aplicación precisa de los siguientes permisos:

- Acceso a la red.
- Cámara.
- Galería.
- Ubicación (basada en red y GPS).

En la figura 1.1 se observan los permisos solicitados en el manifiesto de la aplicación.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fdi.ucm.picday">

    <uses-feature android:name="android.hardware.camera" android:required="true"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Figura 1.1: Permisos necesarios para la aplicación.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.support:recyclerview-v7:27.1.1'
    implementation 'com.android.support:cardview-v7:27.1.1'
    implementation 'com.github.bumptech.glide:glide:3.7.0'
    implementation 'com.android.support:design:27.1.1'
    implementation 'com.google.android.gms:play-services-maps:15.0.1'
}
```

Figura 1.2: Dependencias declaradas en el build.gradle de la App.

## Cómo instalar APK

Una vez comprobado que el dispositivo cumple los requisitos mínimos se puede proceder a la instalación. Para ello hay que permitir la instalación de aplicaciones de origen desconocido en nuestro dispositivo. Para ello accedemos a *Ajustes* y una vez dentro a *Seguridad* donde simplemente hay que marcar la casilla de *Orígenes Desconocidos*.

Ya puede ser instalado el archivo apk de la aplicación.

## Cómo configurar el proyecto en Android Studio

Primero hay que importar el proyecto de la carpeta «PicDay». Una vez importado hay que incluir las bibliotecas externas que han sido utilizadas a lo largo del desarrollo para que el proyecto compile de forma correcta. Las librerías que hay que importar son las que se observan en la figura 1.2. También es necesario importar la librería Volley para la integración con la base de datos.

# Capítulo 2

# Manual de Usuario

Manual orientado al uso de la aplicación.

## 2.1. Resumen de la aplicación

PicDay es una aplicación de fotografía que se centra en eventos diarios o de localización. Cada día se abre un nuevo evento temático para que los usuarios suban una fotografía original relacionada con ese tema a un repositorio común, el resto de usuarios valoran las fotos y al final del día el que más puntuación haya recibido es el ganador del evento y sube en el ranking de usuarios. Además de los eventos diarios habrá eventos que se mantendrán siempre abiertos y que también darán puntos a los usuarios según la puntuación que hayan obtenido sus fotos en ellos. Los usuarios también tendrán la opción de buscar retos por localización, se detectarán aquellos retos especiales localizados a menos de 500 metros. Las fotos del repositorio se borran al acabar el día pero en el perfil de cada usuario se guardan las fotos que haya hecho y aquellas que ha marcado como favoritas.

## 2.2. Funcionalidades

A continuación se muestran las distintas funcionalidades de las que dispone nuestra app, aparte de los típicos formularios de registro e inicio de sesión:

### Página inicial

La primera página a la que accede el usuario una vez entra en su sesión es la que muestra la lista de todos los retos disponibles. La interfaz dispone de

una barra de navegación inferior compuesta por tres botones que permite al usuario acceder a:

- **Su perfil:** Se le mostrará su nombre de usuario y la lista de todas las fotos que haya subido a la aplicación, incluidas las de los retos diarios.
- **Buscar retos:** Obtendrá la localización actual del usuario y le llevará a otra página igual que la inicial pero la lista de retos que se muestra estará compuesta por aquellos retos especiales cercanos al usuario (a menos de 500 metros). Consideramos como retos especiales aquellos que se han creado específicamente para una localización concreta y sólo están activos si se está cerca de esta.
- **Ranking:** Un ranking de usuarios según las puntuaciones que han ido obteniendo.

En el extremo derecho de la barra de herramientas el usuario encontrará un botón marcado como una interrogante que le llevará a una sección que le permita ponerse en contacto con nosotros los desarrolladores.

### Página de reto

Una vez seleccionado un reto el usuario accederá a una pantalla donde se le muestran todas las fotos que contiene dicho reto. Podrá seleccionar aquella que desee para verla en detalle y valorarla o añadir su propia foto, bien haciéndola en el momento o bien importándola desde su galería de fotos. Si decide hacer la foto en el momento esta se guardará tanto en su perfil como en la memoria interna de su teléfono.

### Página de foto

Esta pantalla muestra la foto seleccionada por el usuario, ya sea desde un reto o desde su perfil, junto con el nombre de usuario que la ha hecho. Si la foto no ha sido tomada por él podrá valorarla de 0.5 a 5 estrellas y descargarla a su teléfono. Si es suya únicamente podrá consultar la valoración que tiene.

Tanto esta como las funcionalidades comentadas anteriormente requieren conexión a Internet para ser accesibles ya que obtienen la información mediante nuestra base de datos que estará almacenada en un servidor. Localmente el usuario podrá ver las fotos hechas por él y aquellas que haya guardado.

# Capítulo 3

## Diseño

En esta sección vamos a presentar el proceso de diseño de la aplicación partiendo de los casos de uso para después explicar la arquitectura que va a tener. Para ello vamos a mostrar los diagramas más generales tanto de clase como de secuencia, lo que nos ayudará a la hora de implementar la aplicación.

### 3.1. Casos de uso

Diseñar los casos de uso nos permite ver la aplicación desde el punto de vista del usuario, lo que nos proporciona unos primeros requisitos que debe cumplir nuestra aplicación.

Comenzamos por lo más básico, registro e inicio de sesión (Figura 3.1). Lo primero que se le mostrará al usuario al abrir la aplicación es un formulario de inicio de sesión. Si aún no está registrado podrá hacerlo mediante un formulario al que puede acceder desde la misma pantalla de Login.

Una vez dentro de la aplicación, como ya se ha comentado en la sección inicial, lo primero que se va a encontrar en la pantalla inicial («StartPage») que es la que contiene la mayoría de accesos de la aplicación (Figura 3.2). Podrá seleccionar uno de los retos que incluye la lista de retos que gestionamos o buscar retos especiales mediante su localización y nuestra lista de retos especiales. Podrá también acceder a su perfil para ver sus fotos subidas, consultar el ranking de usuarios que incluye la lista de todos los usuarios registrados ordenados por puntuación o contactar con los desarrolladores para obtener más información. Cómo se accede a estas secciones se ha comentado en la Sección 2.2.1.

Dentro de la pantalla que muestra un reto determinado («ChallengeActivity») cuenta con los casos de uso ilustrados en la Figura 3.3. Se le mostrará

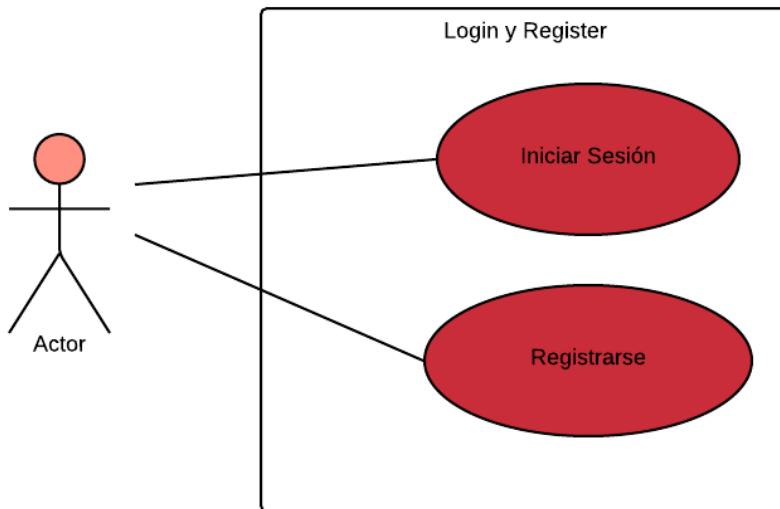


Figura 3.1: Casos de uso de las pantallas de Login y Register.

al usuario la lista de todas las fotos que pertenecen al reto en cuestión mediante miniaturas acompañadas del nombre de usuario del dueño de la foto y la puntuación de esta hasta el momento. El usuario podrá seleccionar una de las fotos o añadir una nueva pulsando en el botón de añadir del extremo derecho de la barra de herramientas.

Por último, la pantalla que contiene la foto en sí y sus detalles («Picture-Page») tiene dos casos de uso (Figura 3.4), valorar la foto con desde media a cinco estrellas que aparecen debajo de la foto o descargarla a su teléfono pulsando un botón de descarga. En esta pantalla se muestra también el nombre de usuario del dueño de la foto pero no se puede interactuar con él, es un dato meramente informativo.

### 3.2. Diseño de la arquitectura

Nuestra aplicación se sustenta en cuatro clases fundamentales, que nos sirven como modelo. Se puede observar una definición de estas clases en la Figura 3.5.

Toda nuestra aplicación girará en torno a estas clases:

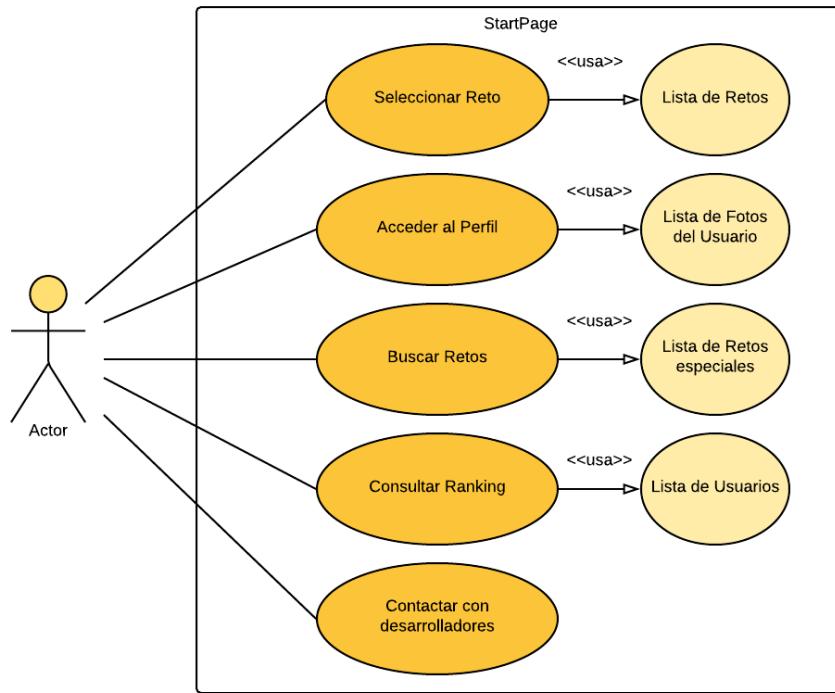


Figura 3.2: Casos de uso de la pantalla inicial.

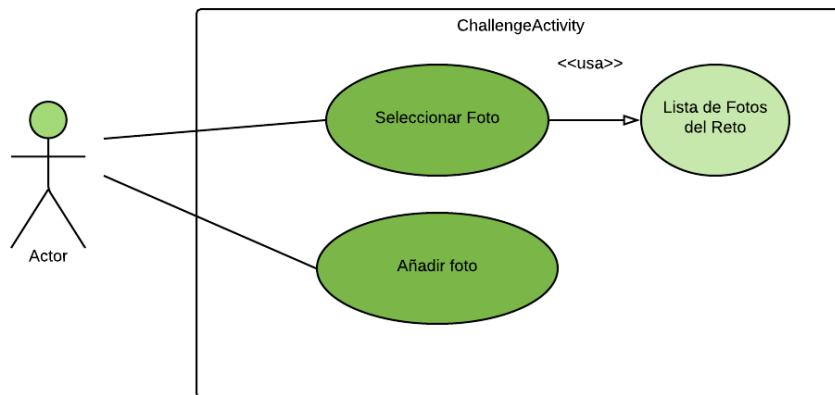


Figura 3.3: Casos de uso de un reto.

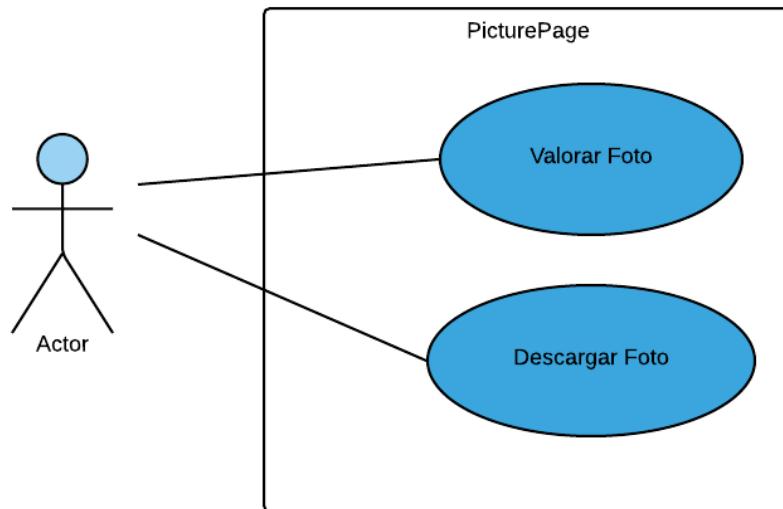


Figura 3.4: Casos de uso de una foto.

- **User:** Representa a los usuarios de la aplicación. Guarda la información típica: nombre de usuario, email y contraseña. Además almacena la fecha de registro del usuario y el número de puntos que ha conseguido hasta el momento.
- **Challenge:** Representa los retos con los que trabaja la aplicación. Almacena su id (un identificador aleatorio), su nombre, su descripción de no más de 140 caracteres y un booleano que nos permite diferenciar los retos diarios de los persistentes. Además de esta clase, contamos con LocatedChallenge para representar retos. Esta clase hereda de Challenge y lo único que aporta es una localización al reto mediante su longitud y latitud.
- **Picture:** Representa las fotos que habrá en los distintos retos y en el perfil de los usuarios. Almacena el identificador de la foto, un Bitmap que la represente, su puntuación (de cero a 5) y las veces que ha sido puntuada. Además, guarda el nombre de usuario de la persona que ha hecho la foto y el identificador del reto al que pertenece. Estos dos últimos atributos nos permitirán enlazar todo en nuestra base de datos.

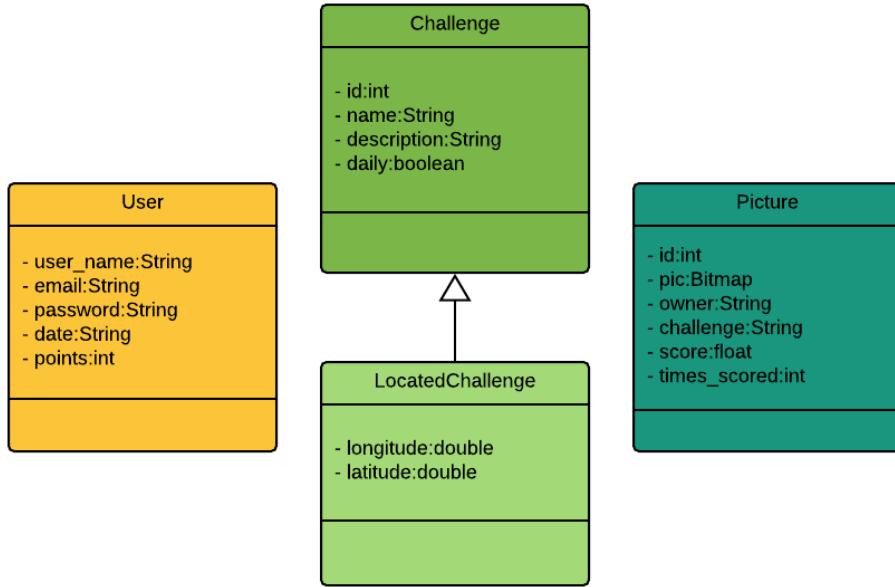


Figura 3.5: Clases principales de la aplicación.

Utilizaremos objetos de estas clases para encapsular las entidades que manejamos en la base de datos. En la figura 3.6 se puede observar el diagrama Entidad-Relación que hemos realizado. Lo implementaremos mediante una base de datos MySql alojada en un servidor, actualmente funciona en local mientras hacemos pruebas, y sobre la que se realizarán consultas con PHP. La librería de HTTP, Volley, nos permitirá integrar las consultas programadas en PHP con Android. Utilizaremos como parte central la tabla «pictures» que contendrá todas las fotos almacenadas en nuestro servidor. Como cada foto guarda el nombre de usuario del dueño y el identificador del reto al que pertenece la carga de fotos se realizará de manera simple según estos parámetros: en la pantalla de un determinado reto se cargarán únicamente aquellas fotos cuyo id\_challenge sea igual al id\_challenge del reto (un simple JOIN) y en el perfil de usuario se hará algo similar, sólo se mostrarán las fotos cuyo user-Name sea igual al userName del usuario. Los retos especiales (Figura 3.7) se almacenan en una tabla distinta a los retos normales pero su funcionamiento es igual, la consulta a la tabla «pictures» es la misma. Lo único que varía para la tabla «located\_challenges» es que sólo se realizan consultas sobre ella si el

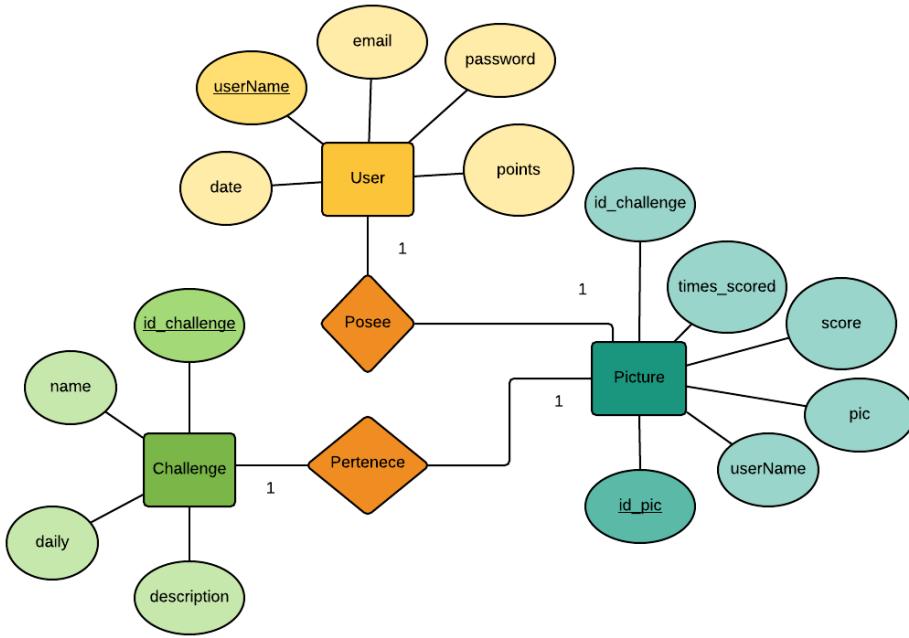


Figura 3.6: Diagrama ER general de la aplicación.

usuario indica que quiere buscar retos y se utilizan los parámetros adicionales de longitud y latitud para saber si el reto cumple la condición de estar a menos de 500 metros o no.

Gran parte de la información que se mostrará en la aplicación será estará representada mediante listas (Recyclers). Para que los elementos se muestren de forma correcta y personalizada necesitamos crear adaptadores para cada una de las clases principales: User, Challenge y Picture. Estos adaptadores serán clases que hereden de RecyclerView.Adapter y nos permitirán personalizar cómo se verán los elementos a la hora de añadirlos a la vista.

Tanto la página inicial («StartPage») como la página de buscar retos («FindPics») muestran listas de retos (Figura 3.8), la única diferencia es que FindPics muestra sólo los retos especiales cercanos por eso la clase hereda de StartPage para añadir el atributo de MAX\_DISTANCE que representa los 500 metros de distancia ya mencionado y el método isClose que recibe las coordenadas de la posición del usuario y del reto para hayar la distancia entre ambos y devolver true en caso de que esta sea menor que MAX\_DISTANCE.

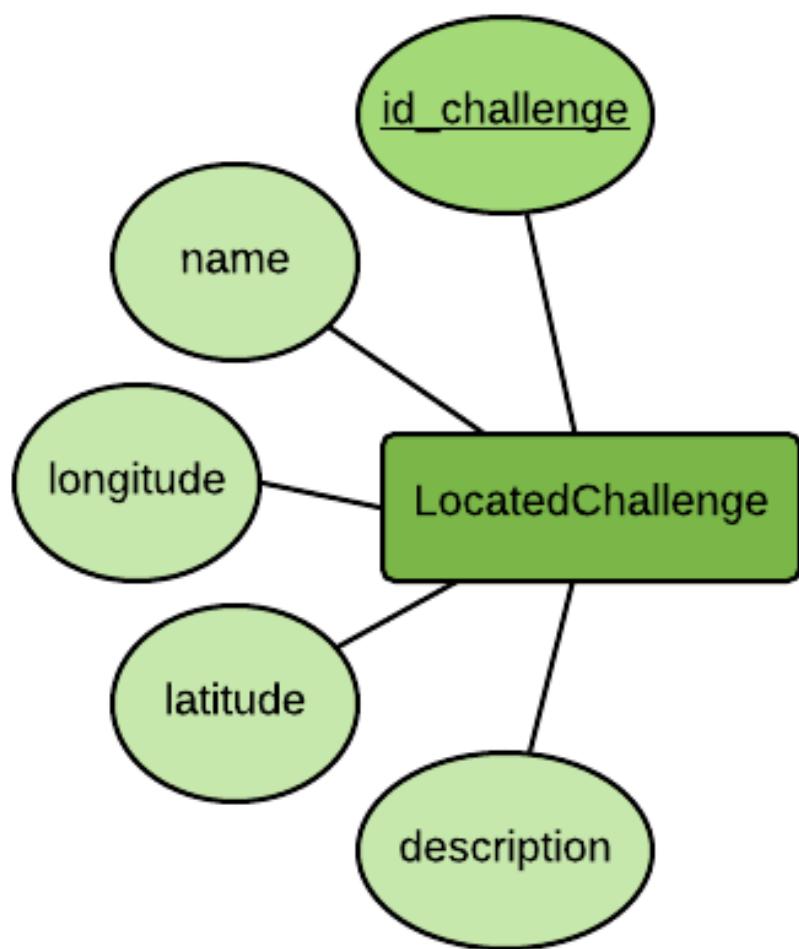


Figura 3.7: Representación de la entidad LocatedChallenge.

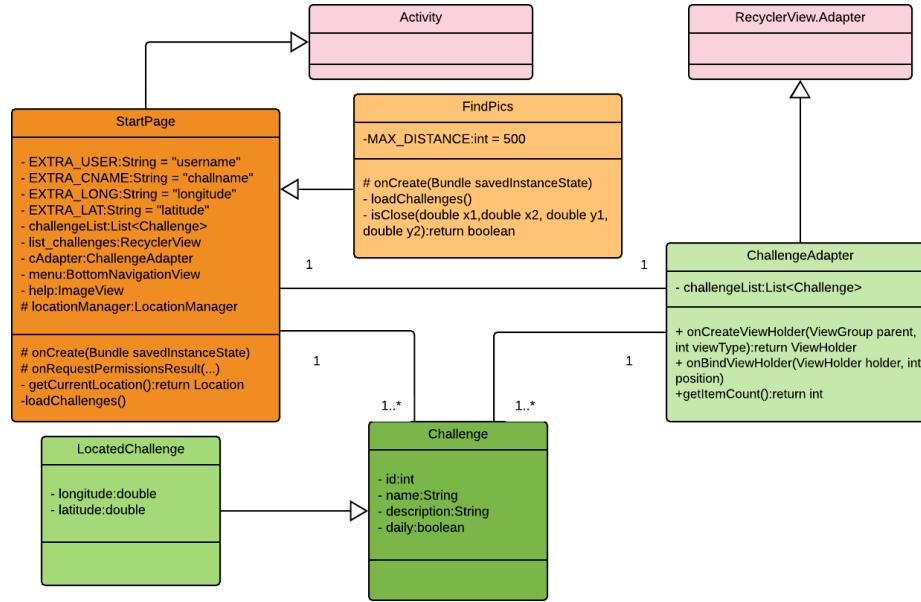


Figura 3.8: Diagramas de clases para la parte de retos.

Ambas clases tienen una barra de navegación inferior, un objeto de tipo BottomNavigationView que implementa un menú de tres elementos. Pinchar en cualquiera de estos tres elementos llevará a otra pantalla: la de perfil, la de FindPics o la del ranking de usuarios. Se trata de transmitir el menor número de información mediante los nuevos Intent: a la actividad Profile y a la de Ranking se les pasa únicamente el nombre del usuario, la de FindPics se le pasan solo las coordenadas de la localización del usuario. A parte de la barra de navegación inferior, las clases también cuentan con un objeto de la clase ImageView que utilizamos a modo de botón, este botón con forma de interrogante llevará al usuario a una pantalla cuyo layout es un WebView que mostrará información sobre la aplicación y los desarrolladores. El objetivo principal de este modulo es que el usuario seleccione un reto después de haber visto su nombre y su descripción para ver las fotos subidas hasta el momento y/o añadir alguna.

Una vez que el usuario ha seleccionado el reto se le mostrará una nueva pantalla con el nombre del reto y todas las fotos que tiene hasta el momento (Figura 3.9). Esta clase y Profile (que hereda de ella) utilizan el adaptador de fotos («PictureAdapter») para mostrar las fotos (combinando Recycler View

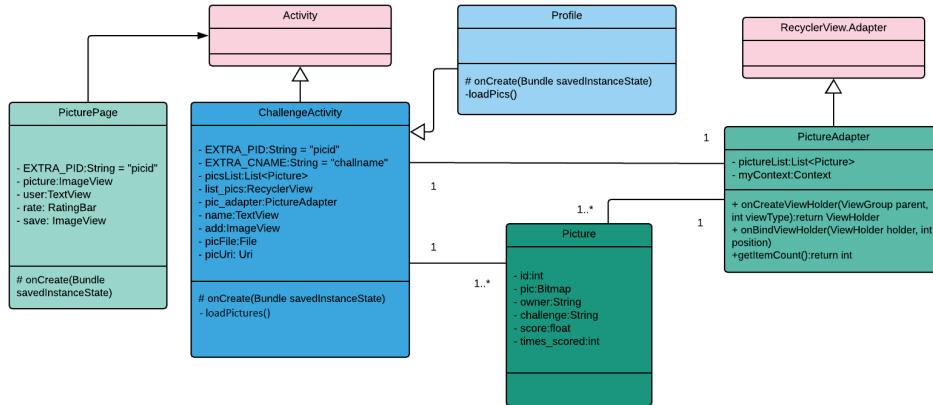


Figura 3.9: Diagramas de clases para la parte de fotos.

y CardView) del reto correspondiente o del usuario respectivamente. A parte de esta lista tendrán un objeto de tipo TextView para mostrar una etiqueta con el nombre del reto en el caso de ChallengeActivity o del usuario en el caso de Profile y otro de tipo ImageView que utilizaremos como botón para que el usuario pueda añadir desde ahí sus nuevas fotos. Pulsar sobre este botón abrirá la cámara del dispositivo y permitirá al usuario hacer la foto. Una vez hecha esta se guardará tanto en el dispositivo de forma local como en nuestra base de datos junto al nombre de usuario del usuario y el id del reto desde el que se ha hecho la foto. Los objetos picFile y picUri permiten el almacenamiento local de la foto. Si la foto se hace desde el perfil en vez de desde un reto se pedirá al usuario que especifique a qué reto quiere añadir la nueva foto.

Si se selecciona una foto desde cualquiera de las dos vistas que acabamos de comentar se iniciará una nueva actividad a la que le pasaremos el identificador de la foto para cargarla y mostrarla junto al nombre de usuario de la persona que la haya hecho y la valoración que tiene. La foto se mostrará sobre un ImageView en el que insertaremos el Bipmap de la foto, el nombre de usuario aparecerá sobre un TextView y la valoración se mostrará con mediante un RatingBar de cinco estrellas. En esta pantalla también contará con un objeto ImageView que nos servirá como botón para descargar la imagen al dispositivo del usuario.

Lo último que nos queda por comentar es el adaptador de usuarios («UserAdapter») que será utilizado en la pantalla de Ranking (Figura 3.10) que simplemente mostrará los usuario por orden decreciente de puntuación junto

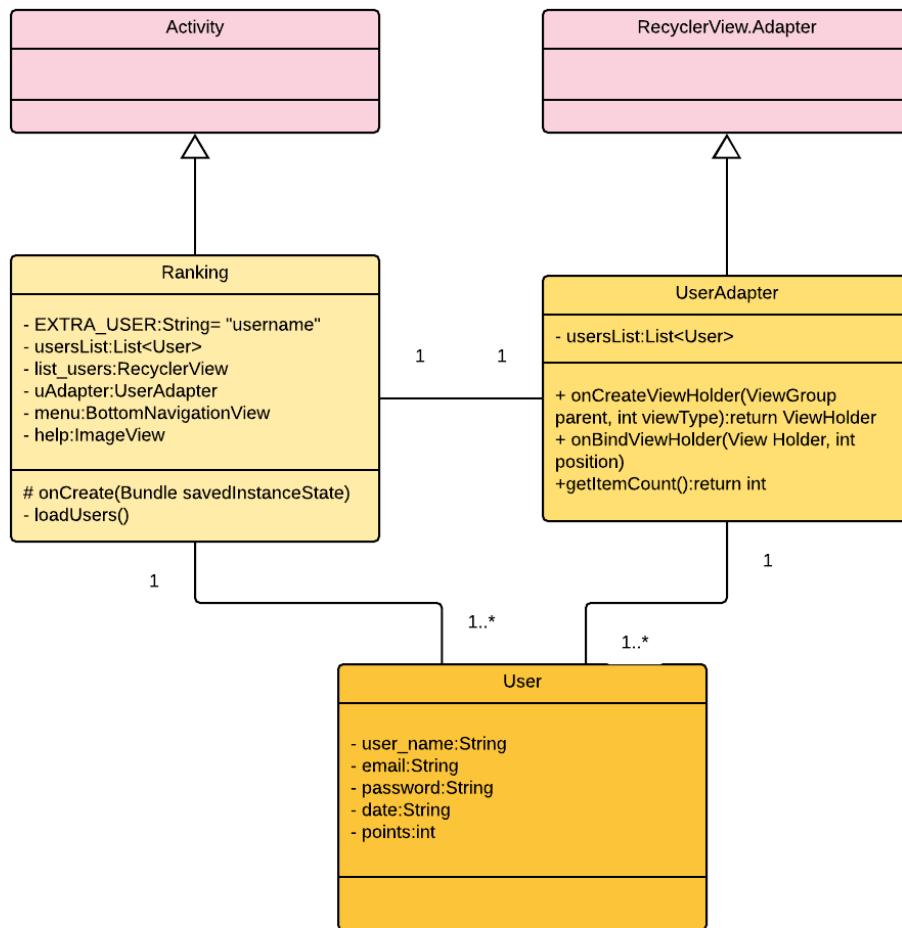


Figura 3.10: Diagramas de clases para la parte de usuarios.

a sus puntos. Esta pantalla tiene, al igual que la de StartPage y FindPics, la barra de navegación inferior y el botón de ayuda.

### 3.3. Diagramas de secuencia

En esta sección se mostrarán los diagramas de secuencia para mostrar de forma concisa el funcionamiento de los modulos de los que dispone actualmente la aplicación.

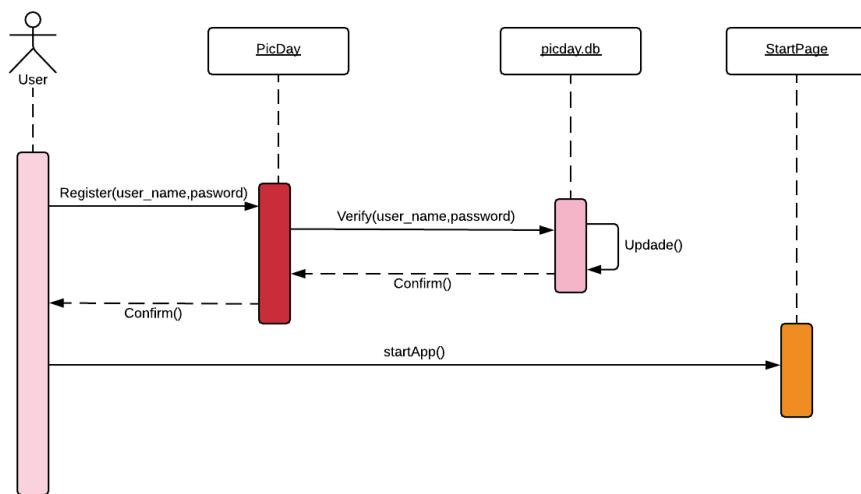


Figura 3.11: Diagrama de secuencia del formulario de registro.

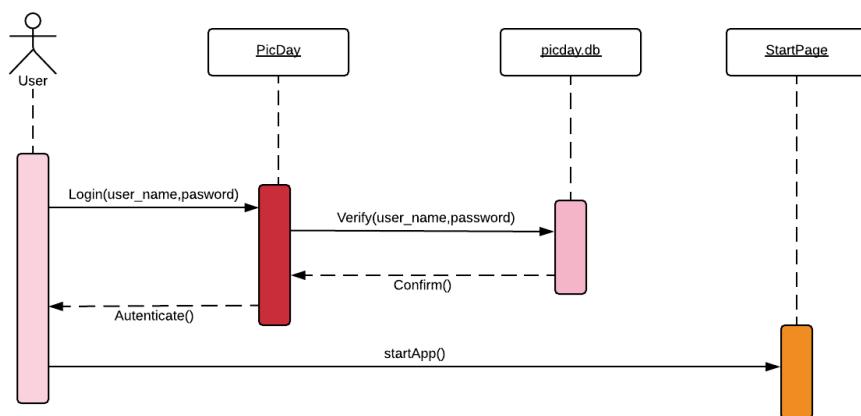


Figura 3.12: Diagrama de secuencia del formulario de Login.

En primer lugar, como en toda aplicación tenemos los formularios de registro (Figura 3.11) y de inicio de sesión (Figura 3.12). Estos se encargarán de coger los datos introducidos por el usuario y acceder a la base de datos para comprobar si son correctos. Si el usuario intenta iniciar sesión se buscará si el nombre introducido existe y la contraseña que le corresponde es la misma que la que ha introducido el usuario. Si todo está bien se autenticará al usuario y se le llevará a la pantalla inicial. Si lo que quiere el usuario es registrarse proporcionará sus datos y, si no hay un usuario con el nombre introducido, se actualizará la base de datos para almacenar al nuevo usuario. De nuevo, si todo ha ido bien se le llevará a la pantalla inicial.

Una vez en la pantalla inicial (Figura 3.13) se crea la vista tras solicitar a la base de datos la lista de retos normales. Esta pantalla permite al usuario acceder a prácticamente todo el resto de vistas, como ya se ha mencionado: si pulsa en alguno de los retos se le llevará a la ChallengeActivity correspondiente, si pulsa en cualquiera de los botones de la barra de navegación inferior se le llevará a la vista correspondiente (Profile, FindPics y Ranking, de izquierda a derecha) y si pulsa en el ícono con forma de interrogación se le llevará al formulario de contacto. A continuación vamos a explorar las posibles vistas y sus diagramas de secuencia correspondientes siguiendo el orden de la figura a la que acabamos de hacer referencia.

Si el usuario selecciona un reto de la lista se le mostrará la lista de fotos que componen dicho reto (Figura 3.14) tras obtenerla de la base de datos. Dentro de esta vista lo que podrá hacer es seleccionar una de las fotos, lo que le llevará a la vista de PicturePage (Figura 3.15), o pulsar en el ícono con forma de aspa para añadir una nueva foto. Si decide añadir una foto se abrirá la cámara del dispositivo (tras comprobar que esta está disponible y tenemos los permisos necesarios para acceder a ella) y el usuario podrá hacer su foto. Una vez hecha se actualiza la base de datos y se recarga la pantalla del reto con la nueva información.

La vista de PicturePage se creará solicitando a la base de datos la información sobre la foto con el correspondiente identificador. Una vez se le muestre al usuario este puede valorar la foto, lo que provocará una actualización de los atributos «score» y «times\_scored» en la base de datos, o guardarla en su dispositivo.

En su perfil (Figura 3.16), el usuario verá la lista de las fotos de las que es dueño tras obtenerla de la base de datos. Lo único que puede hacer en esta vista es seleccionar una de las fotos para verla mejor (no podrá valorarla y ya la tiene en su dispositivo local), lo que le llevará a la vista que hemos acabamos de mencionar: PicturePage.

La vista de FindPics (Figura 3.17), como ya hemos explicado, es idéntica

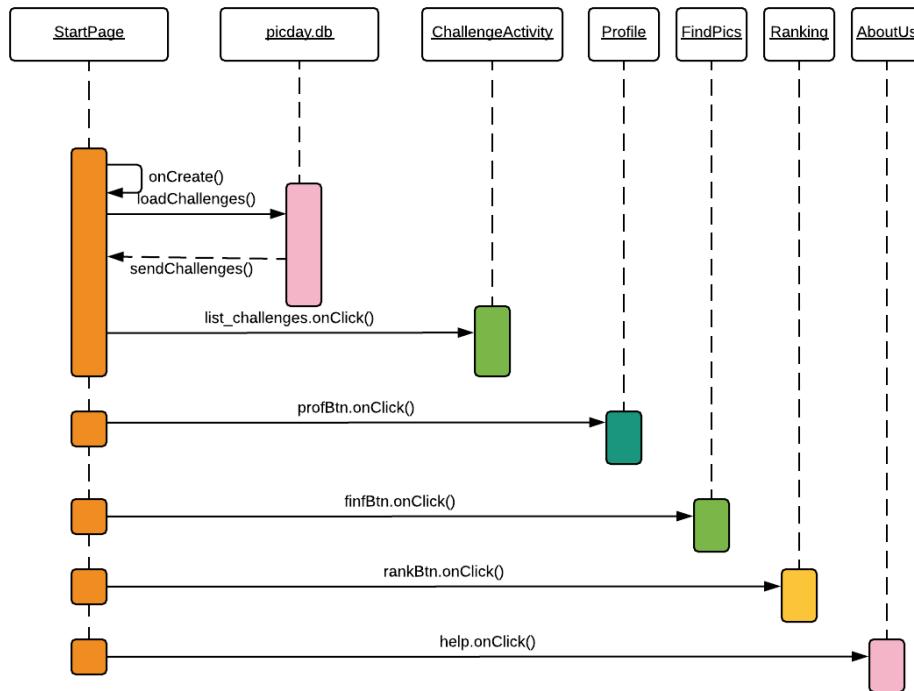


Figura 3.13: Diagrama de secuencia de la página inicial.

visualmente a la inicial, lo único que cambia son los retos que se muestran. Antes de acceder a esta vista se le deben pedir permisos al usuario para acceder a su localización, si no se tenían ya. Desde esta vista, al igual que desde la inicial o la de Ranking se puede utilizar la barra de navegación inferior para acceder a otras secciones y el ícono de interrogación para contactar con los desarrolladores. La vista de Ranking se encarga, por su parte de obtener la lista de todos los usuarios de la base de datos y los muestra tras clasificarlos por puntuación. Es una lista no interactiva, su objetivo es únicamente mostrar la clasificación.

Por último, el formulario de contacto (Figura 3.19) es simplemente una pequeña página HTML embebida en un WebView donde el usuario puede consultar información sobre la aplicación o los desarrolladores y dispone de unos links para contactar con nosotros en caso de duda o sugerencia.

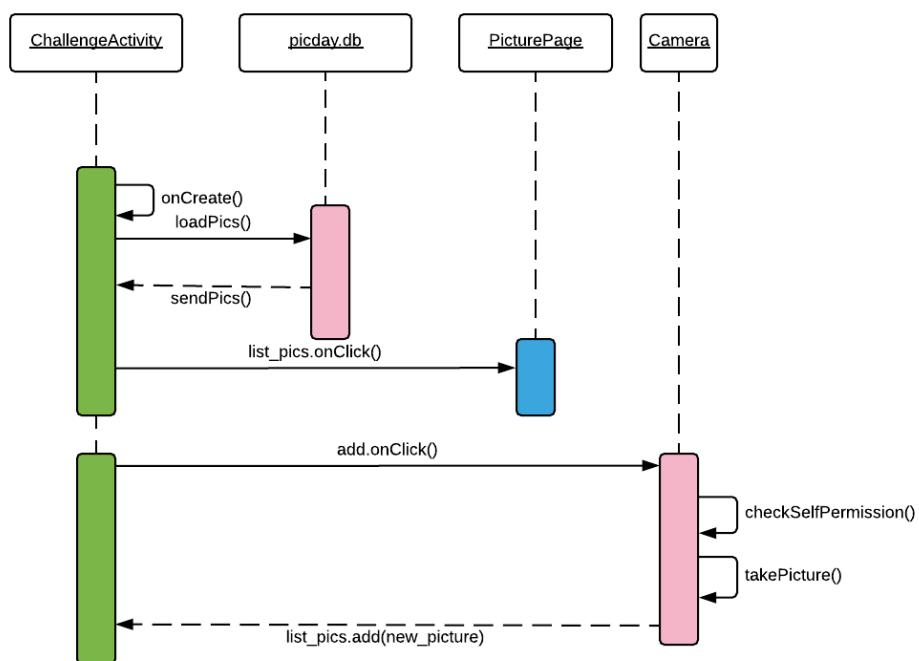


Figura 3.14: Diagrama de secuencia del formulario de un reto.

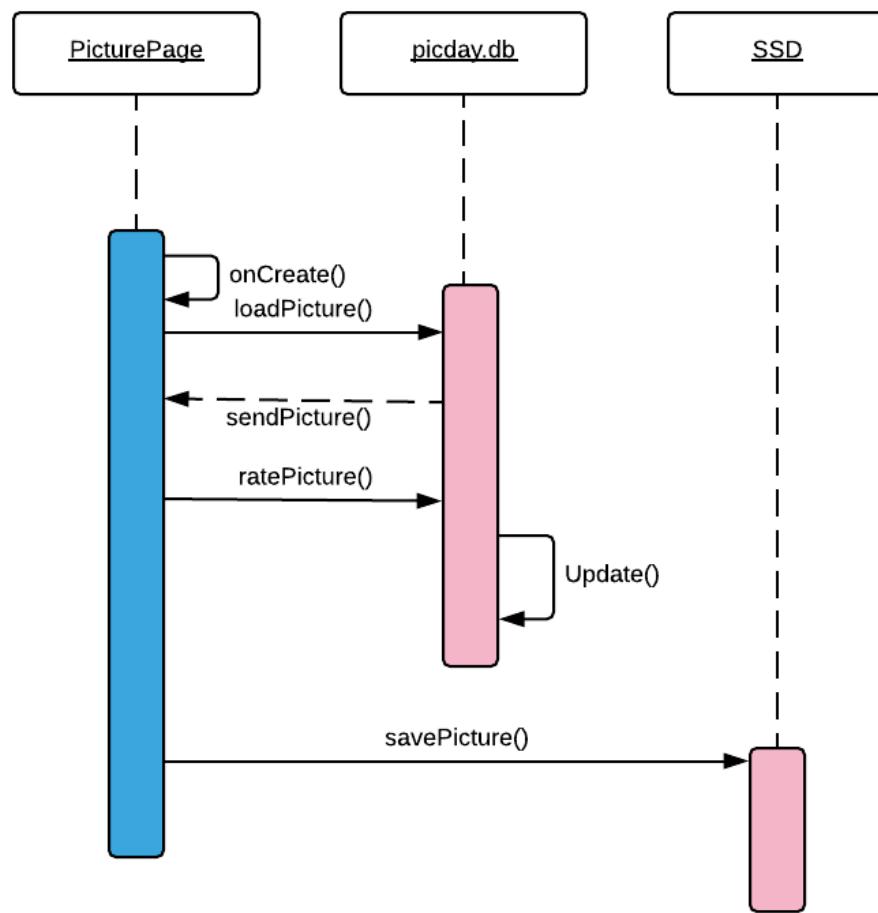


Figura 3.15: Diagrama de secuencia del formulario de una foto.

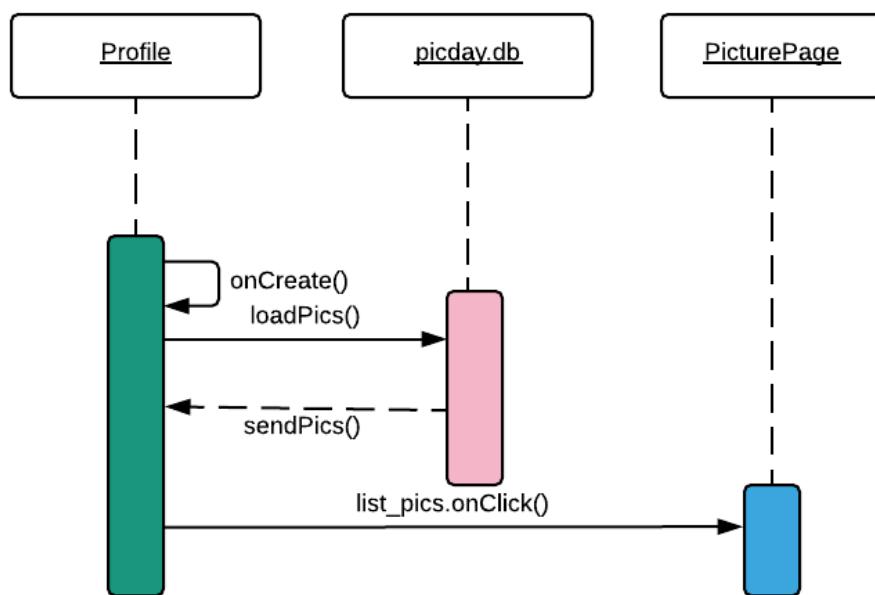


Figura 3.16: Diagrama de secuencia del formulario de perfil de usuario.

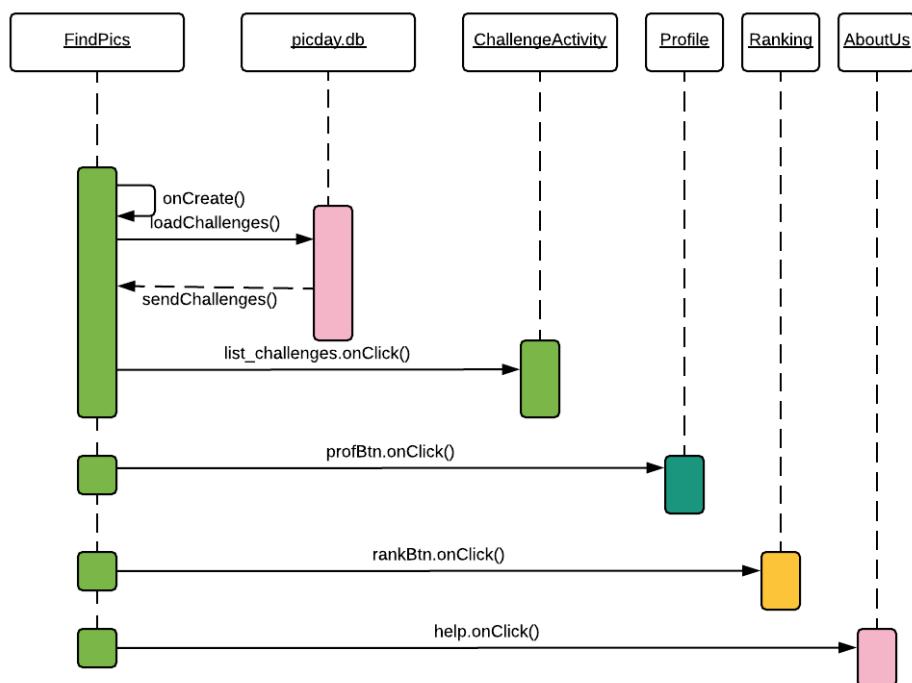


Figura 3.17: Diagrama de secuencia del formulario de retos especiales.

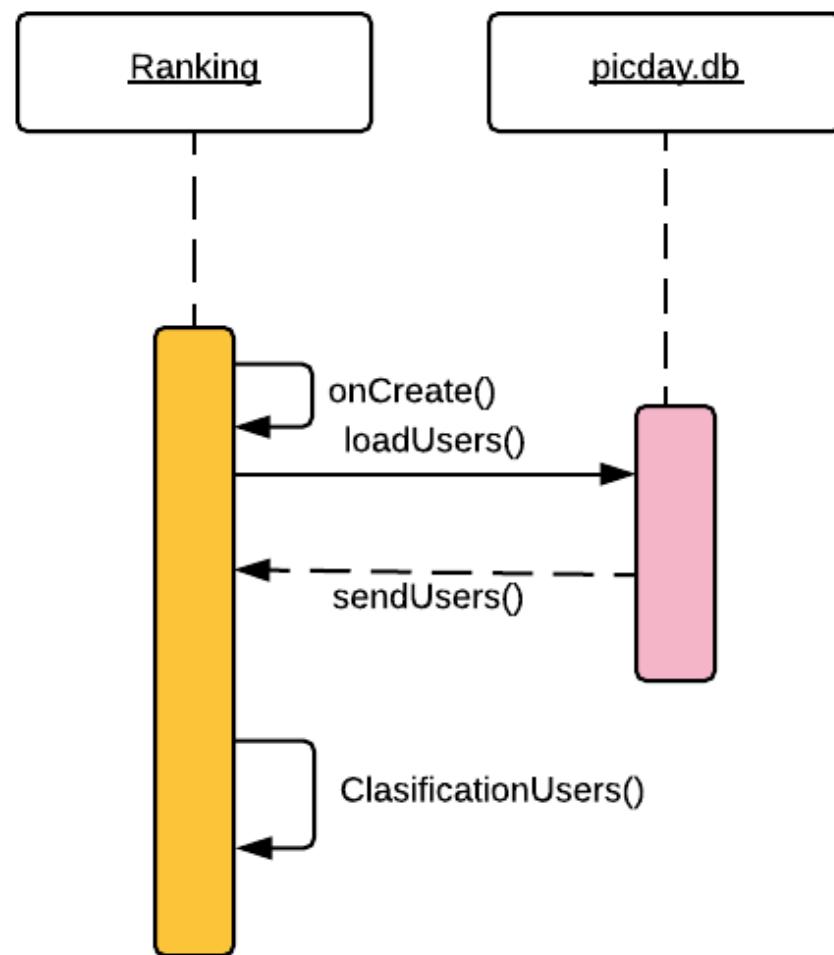


Figura 3.18: Diagrama de secuencia del formulario de ranking de usuarios.

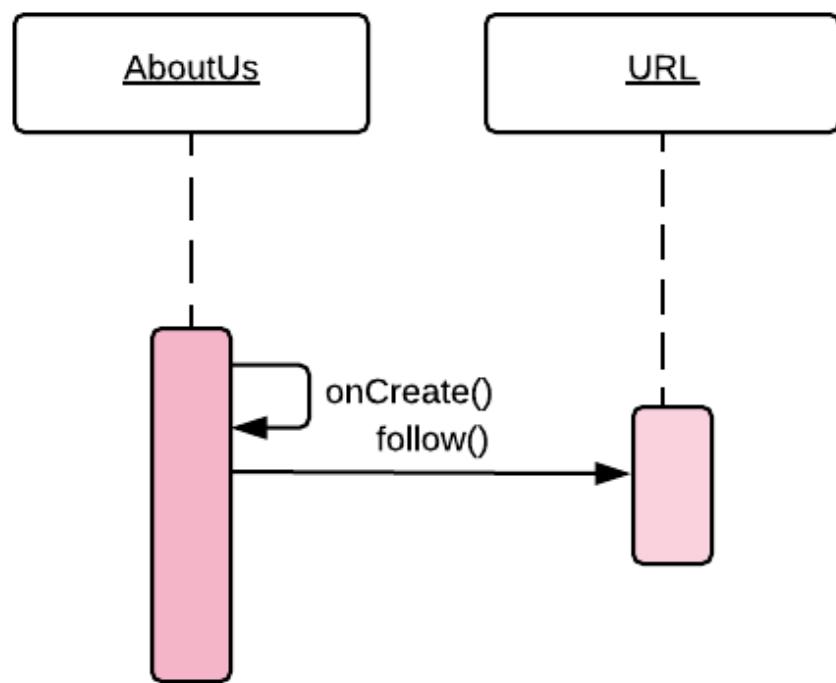


Figura 3.19: Diagrama de secuencia del formulario AboutUs.

## Capítulo 4

# Consideraciones técnicas

En esta sección explicaremos más en detalle los componentes utilizados en el desarrollo de la aplicación. Hemos creado cinco usuarios para realizar distintas pruebas y a continuación mostraremos capturas de la interfaz explicando cómo se han llevado a cabo las partes más importantes.

En primer lugar, igual que en la mayoría de aplicaciones tenemos un formulario de registro (Figura 4.1) y otro de inicio de sesión (Figura 4.2). Ambos formularios se componen de un ImageView con el icono de la aplicación, una serie de EditText para introducir los datos y unos Button para confirmar o cancelar (en el caso del registro). Para pasar al formulario del registro el usuario debe pulsar en el TextView «JOIN US!». Los botones, por su parte son intuitivos.

Hemos personalizado el Toolbar para que aparezca el nombre y el icono de la aplicación y se muestre de color rojo que será el principal que utilizaremos.

Una vez dentro de la aplicación el usuario se encuentra, como ya se ha comentado, con la lista de retos (Figura 4.3). La lista está representada gracias a un RecyclerView que utiliza el adaptador de retos («ChallengeAdapter») para representar cada uno de ellos en una tarjeta, definida en el «challenge\_list\_row.xml», compuesta por dos TextView: uno con el nombre del reto y otro debajo con la descripción de este.

Al seleccionar un reto el usuario se encontrará en una pantalla con la lista de las fotos (Figura 4.4) implementada también gracias a un RecyclerView pero con un adaptador diferente («PictureAdapter») que utiliza las tarjetas definidas en «pic\_card.xml». Estas tarjetas, además de dos TextView que representan el nombre del usuario que ha hecho la foto y la puntuación de esta, define un ImageView para mostrar una miniatura de la imagen. Esta vista permite al usuario ver el nombre del reto mediante un TextView y si

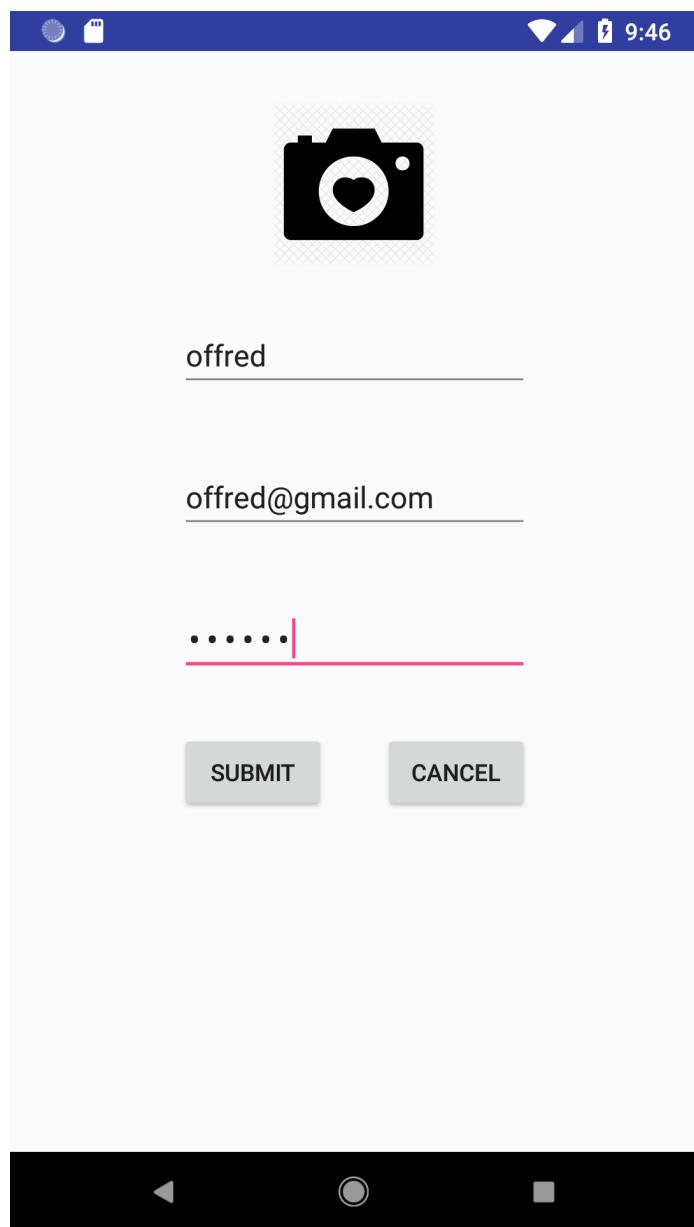


Figura 4.1: Captura pantalla de registro.

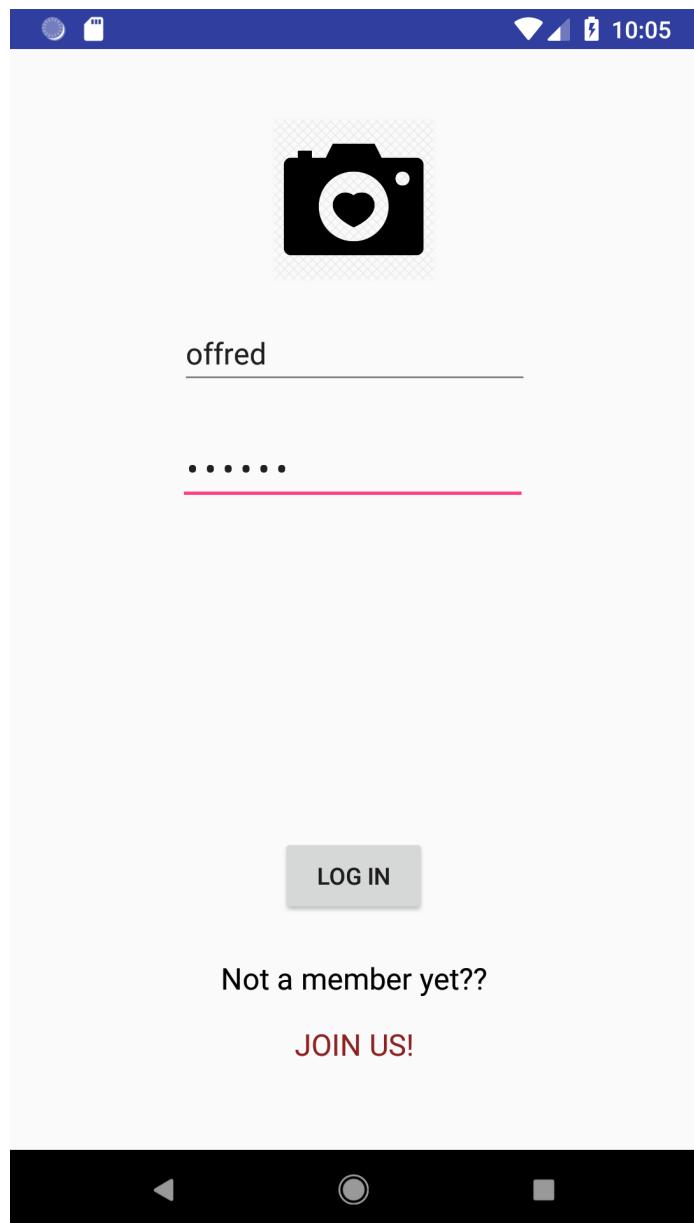


Figura 4.2: Captura pantalla de inicio de sesión.

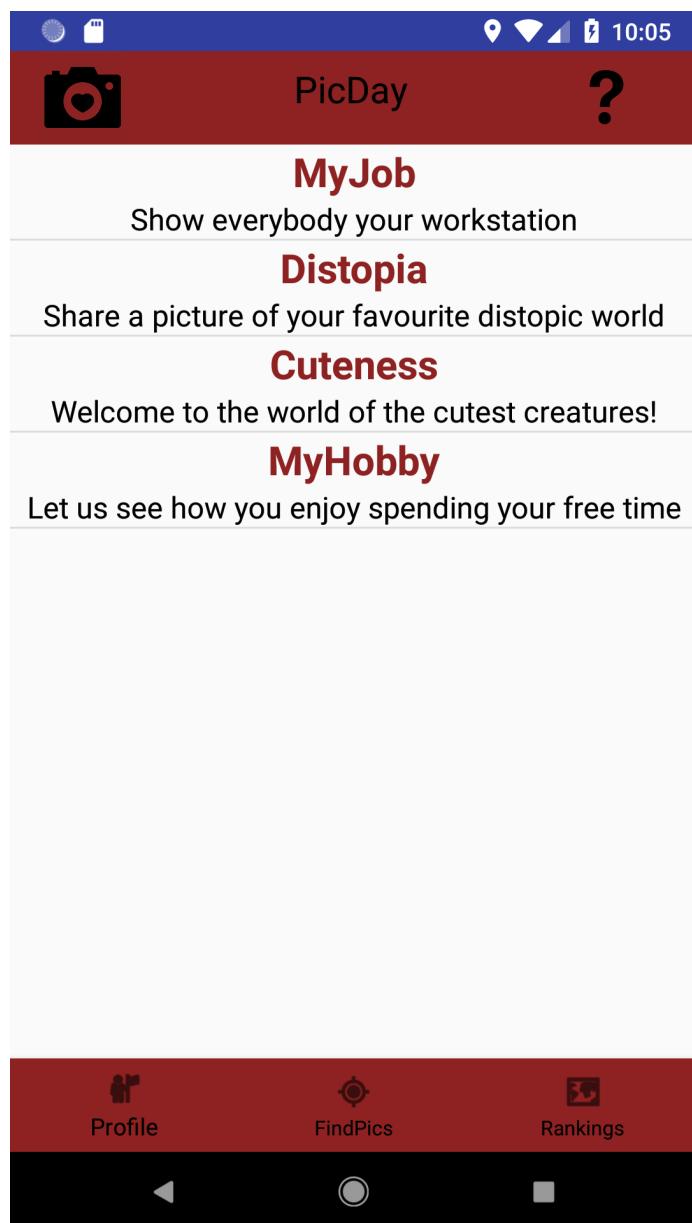


Figura 4.3: Captura pantalla de retos.

el reto es diario se le notifica al usuario mediante un Toast (Figura 4.5). El usuario podrá desde la pantalla de reto podrá añadir su imagen pulsando en el ImageView del extremo derecho del Toolbar. Esto abrirá la aplicación de cámara del dispositivo tras pedir los permisos necesarios (Figura 4.6) y el usuario podrá realizar una foto desde ella.

Una vista en detalle de una foto se puede obtener al seleccionarla desde la pantalla que acabamos de comentar o desde el perfil del usuario. Esta vista detalle consiste en un ImageView que contiene la imagen seleccionada, un RatingBar para mostrar la puntuación de la foto, un TextView que contiene el nombre de usuario del dueño de la foto y un ImageView que se utilizará a modo de botón para la descarga de la imagen.

La página inicial no sólo contiene la lista de retos si no también accesos al resto de módulos de la aplicación. En la esquina derecha del Toolbar hay un ImageView que representa un signo de interrogación y lleva al usuario a la pantalla de «Acerca de» (Figura 4.8). Esta vista es un WebView de un index.html que tenemos en el directorio de assets de la aplicación. En la parte inferior hay un BottomNavigationView de tres botones que implementa el menú definido en el «mainmenu.xml» del directorio de recursos «menu». Cada uno de estos botones lleva a una sección distinta:

- **Perfil** (Figura 4.9): Contiene, al igual que la vista de un reto, una lista de fotos. En este caso las fotos representadas son las que ha hecho el usuario en cuestión. Aparte de las fotos se muestra un TextView con el nombre de usuario.
- **Ranking** (Figura 4.10): Contiene una lista de usuarios representada mediante un RecyclerView que utiliza el adaptador «UserAdapter» para implementar las tarjetas definidas en «user\_list\_row.xml». Estas tarjetas están formadas por un TextView con el nombre del usuario y otro TextView en el extremo opuesto con su puntuación.
- **FindPics** (Figura 4.11): Vista idéntica a la que muestra los retos normales, la única diferencia es que sólo aparecen los retos especiales que estén cerca del usuario.

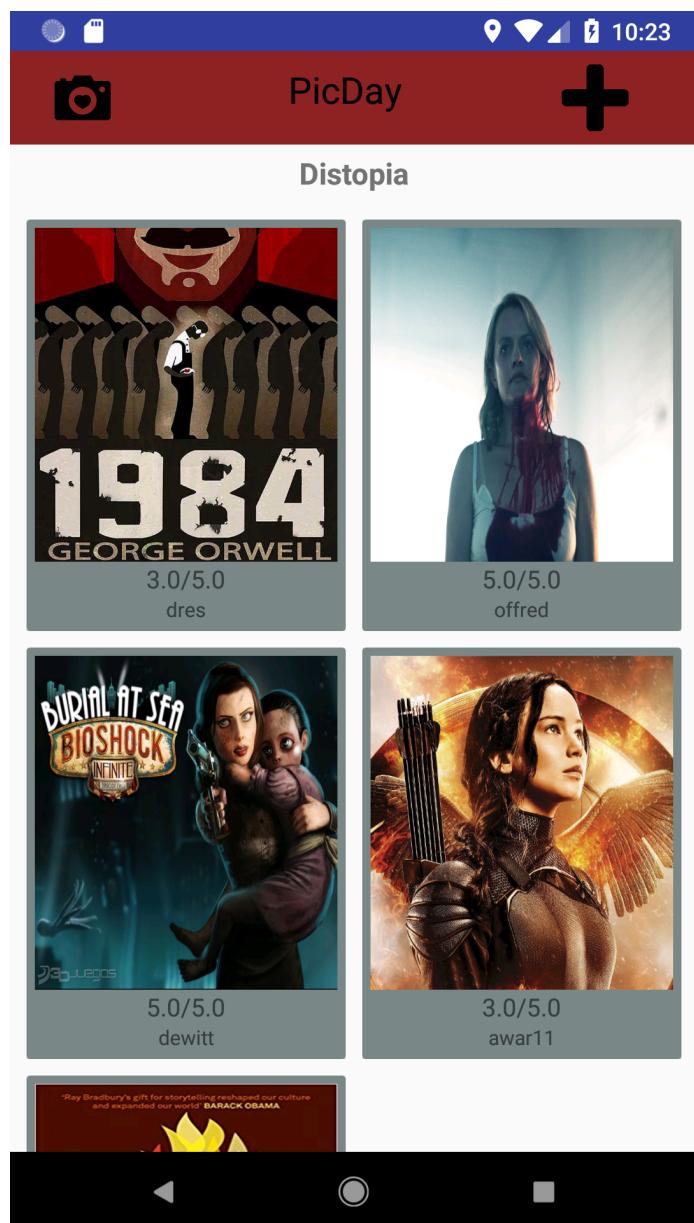


Figura 4.4: Captura pantalla del reto "Distopia".



Figura 4.5: Captura pantalla del reto diario "MyJobs".

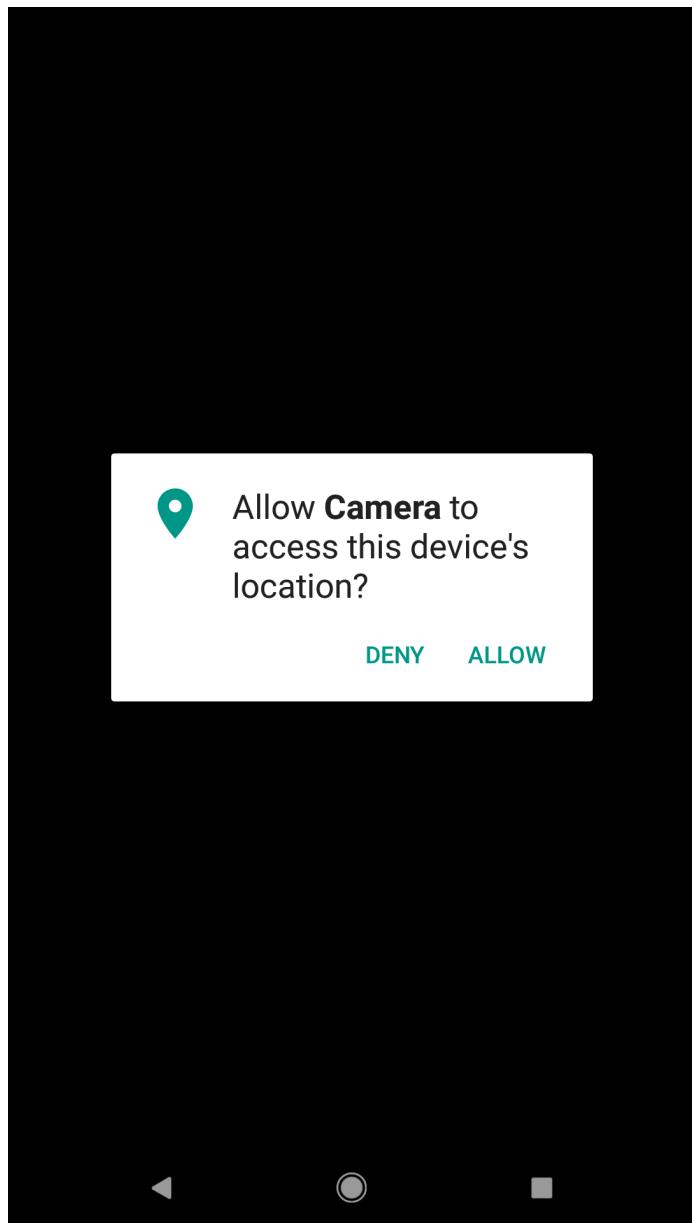


Figura 4.6: Captura solicitud de permisos para usar la cámara.

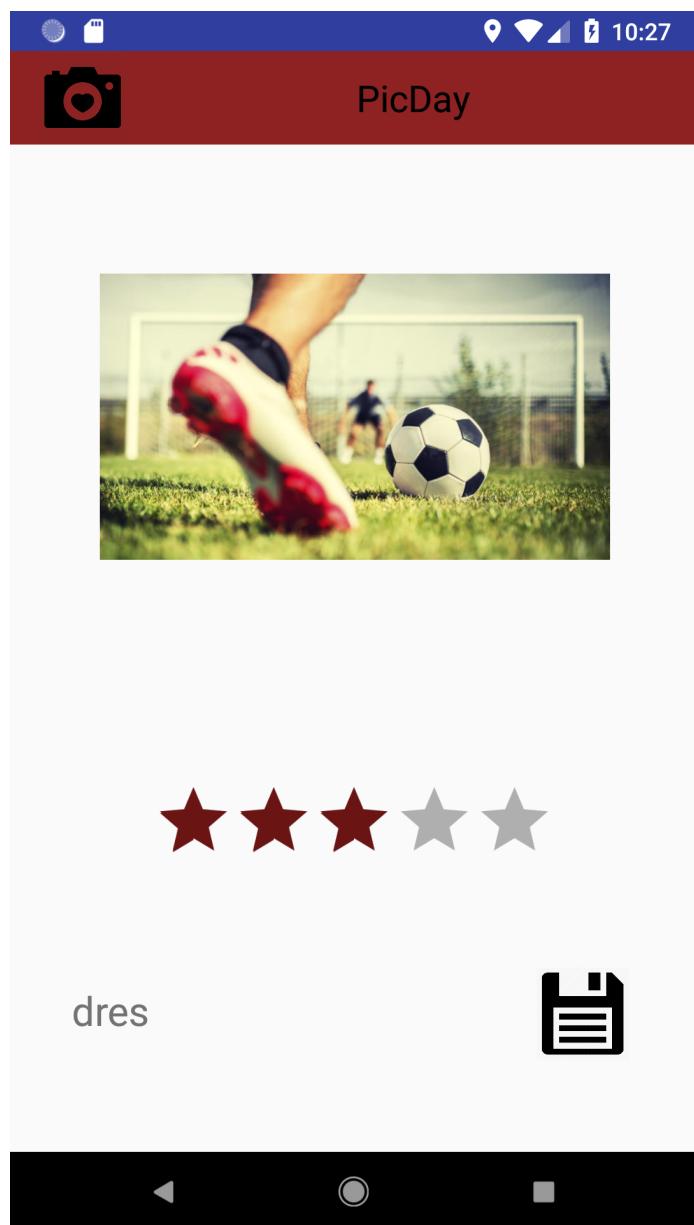


Figura 4.7: Captura pantalla detalle de una foto.



## This app was developed by:

- Elena Kaloyanova Popova
- Andrés Plaza Hernando

PicDay is the result of the final project of the subject "Programación de Aplicaciones en Dispositivos Móviles" and is meant to be a way to get yourself introduced into photography or improve your skills through lots of original challenges.

Don't hesitate to consult us if you have any questions:

- [Follow @dokrai11](#)
- Andrés Plaza Hernando



Figura 4.8: Captura pantalla Acerca De.



Figura 4.9: Captura pantalla de perfil.

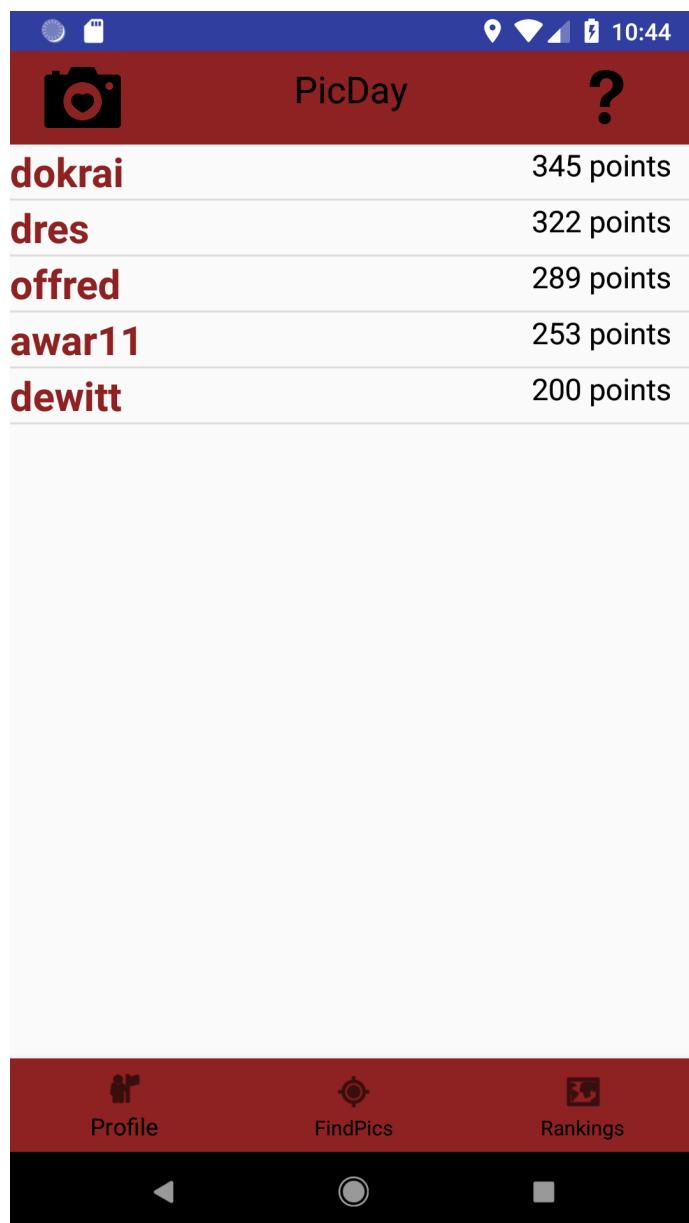


Figura 4.10: Captura pantalla de ranking.

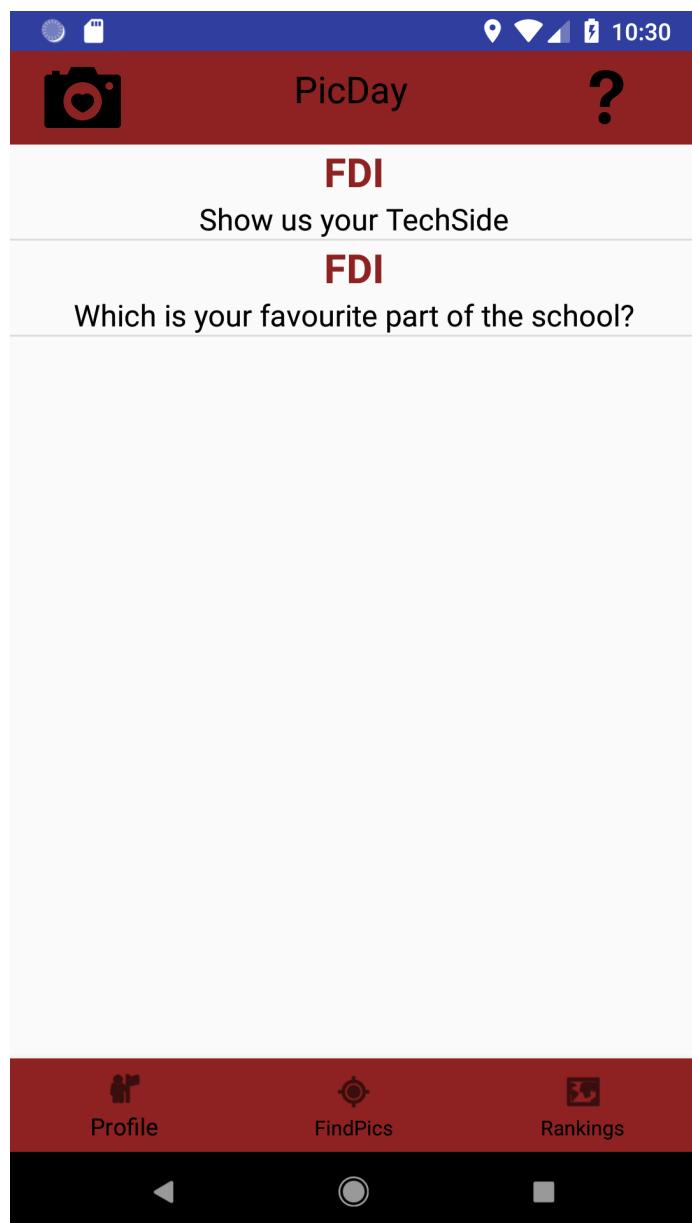


Figura 4.11: Captura pantalla de retos por cercanía.

## Capítulo 5

# Metodología de trabajo y conclusiones

Para el desarrollo de la práctica empezamos trabajando juntos para definir los objetivos de la aplicación y realizar el modelado y los diagramas. Una vez tuvimos claro en qué iba a consistir la aplicación y la interfaz que iba a tener abrimos un repositorio en GitHub con la aplicación y la memoria. Hemos contribuido de forma similar en el desarrollo ya que ninguno teníamos experiencia con Android y necesitábamos aprender. Una vez que obtuvimos una versión funcional Andrés se empezó a centrar en crear la base de datos y las consultas necesarias para la aplicación y Elena en depurar la interfaz. Cuando tengamos la base de datos volveremos a juntarnos para realizar la integración entre los dos.

Por ahora, debido a los problemas que nos hemos encontrado a la hora de crear e integrar la base de datos, los datos que utiliza la aplicación están metidos directamente desde el código. Tenemos una base de datos ya creada y gran parte de las consultas pero no hemos tenido tiempo para realizar la integración. Aprovecharemos hasta el día de la entrega para implementar esta parte y todo lo derivado de ella (se reduce el número de extras que le pasamos a los intents, el tratamiento de datos será más limpio y eficiente). También depuraremos ciertas partes. En los elementos pulsables el código está duplicado porque estamos haciendo las pruebas en un emulador y al hacer click nos lo detecta como un «onLongClick». Cuando la aplicación esté acabada y vayamos a lanzarla quitaremos esa parte para implementar otra funcionalidad para el «onLongClick».