Proyecto final de curso

Android: Fundamentos de Programación

(julio-noviembre 2015)

Nombre de la aplicación: **DataWeather**

Autor: Izquierdo Doménech, Juan Jesús

Que hace la aplicación:

Aplicación que **almacena datos sobre el tiempo** del lugar donde se encuentra el usuario automáticamente (ciudad, latitud, longitud y condición meteorológica).

Licencia:

El código de la aplicación se encuentra bajo el **sistema de control de versiones** Git, y esta alojado en la siguiente dirección de la plataforma GitHub (https://github.com/JuanIzquierdoDomenech/DataWeather)

A destacar:

- Uso del hilo principal y otros hilos para hacer peticiones HTTP.
- Leer datos de un JSON mediante llamadas a una API por HTTP GET.
- Localización de la app (Español e Inglés).
- Uso de imágenes para el icono de la app, animaciones (el latido que aparece en la actividad LoginActivity que muestra la condición meteorológica) y selectores de botones.
- Utilización de RelativeLayout y LinearLayout para maquetar las vistas.

- Uso de una lista y un adaptador propio (WeatherListItemAdapter y R.layout.activity_weatherlist_item).
- Uso de una clase como modelo para los datos, que, además, implementa la interfaz Parcelable para poder pasar instancias de la misma entre actividades con el método putExtra de la clase Intent.
- Uso de una base de datos SQLite.
- Se añadieron las clases de BaseGameUtils de google para poder loguearse con la cuenta de usuario de google y poder añadir logros, pero no tenía sentido en esta aplicación, y, aparte, se necesitaría dar de alta el correo de los usuarios como testers, puesto que la app no se tiene pensado subir a la store. El código está comentado y aparece en la LoginActivity.

Cómo lo hace:

La aplicación se inicia en la actividad LoginActivity, en la que hay ya un primer contacto con el sistema de localización del dispositivo móvil, comprobando si los proveedores de localización GPS y de red están habilitados.

Esta actividad muestra un campo de texto donde irá el nombre del usuario y un botón para ir a la siguiente actividad, pero esto sólo será posible si se encuentran datos sobre la localización y el tiempo, y el usuario ha introducido un nombre.

Una vez se haya encontrado la localización, se lanza una llamada a la API de OpenWeatherMap que devuelve los datos del tiempo en forma de JSON. Esta llamada se realiza por medio de un HTTP GET, en el que se requiere que el parámetro q=? sea el nombre de la ciudad sobre la cual se necesitan los datos, y que conseguimos traduciendo la latitud y la longitud con un método que se llama Reverse Geocoding.
Para utilizar la API de OpenWeatherMap se requiere de una clave que se puede conseguir de forma gratuita al registrarse en la página de OpenWeatherMap, y que se añade a la petición HTTP con el método addRequestProperty(arg1, arg2) de la clase HttpURLConnection, cuyo primer parámetro es la string “x-api-key” y el segundo la clave que te dan en la web.
Para el uso de esta API, se ha seguido el siguiente tutorial de Tuts+, que enseña como se pueden leer datos de un JSON desde Android.
La petición a la API se realiza en un thread diferente, y se comunica al hilo principal con una instancia de la clase Handler, ya que así podemos modificar elementos de la interfaz.

Con el nombre del usuario escrito en el campo de texto y viendo que se han recabado datos de localización y el tiempo, si pulsamos el botón de ‘!Empezemos¡”, pasaremos a la actividad WeatherListActivity.

La primera vez que entremos en esta actividad, se insertará en la base de datos el primer registro de los datos del lugar y el tiempo que se recopilaron en LoginActivity.

Esta actividad muestra una lista en donde se irán introduciendo de forma regular mediante un TimerTask los datos de localización y tiempo en la base de datos. Aparte, el usuario puede añadir manualmente una entrada en la base de datos con el botón '+' que se encuentra en la parte inferior derecha.

Desde esta actividad, con el menú podemos 'des-logearnos', lo que nos hace volver a la LoginActivity, acceder a las preferencias (se explica más abajo), y ordenar la lista tanto por fecha como por lugar.

Si el usuario pulsa en una fila de la lista, se le muestra la actividad WeatherDetailActivity, donde aparecen los datos de la entrada de forma más ampliada.

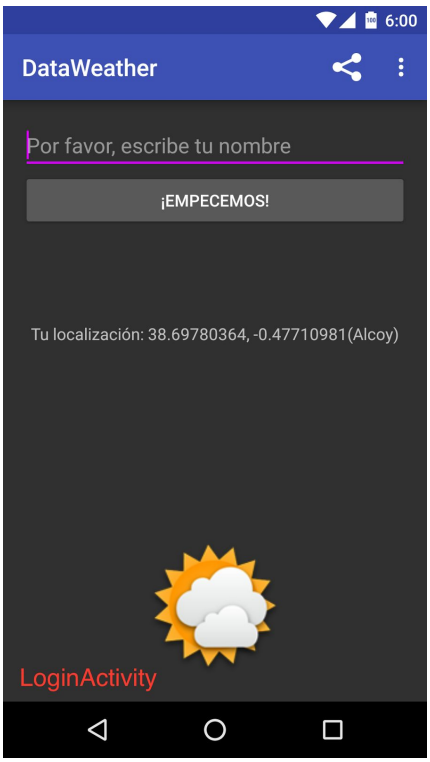
Si el usuario hace una pulsación larga en una fila de la lista, le aparece la opción de eliminar el registro mediante un AlertDialog.

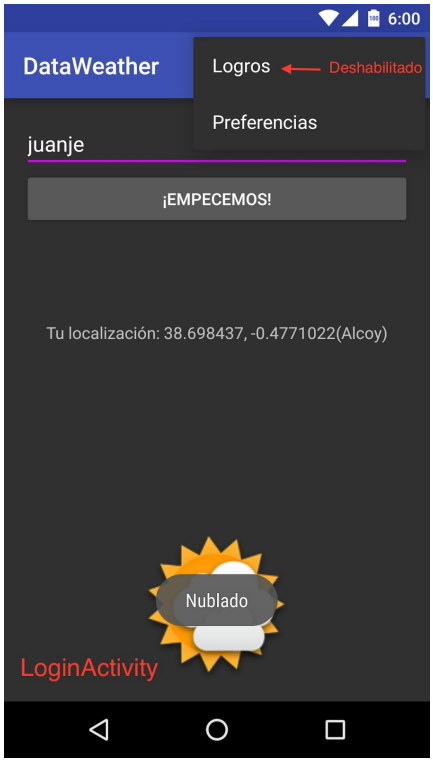
Aparte del flujo normal de la app (introducir nombre y que la app vaya introduciendo registros de datos), se ha experimentado con la opción de compartir datos de tipo texto con un intent a otras aplicaciones, siguiendo la documentación de Google (link).

Desde las preferencias, se permite cambiar la frecuencia de actualización del LocationManager, la distancia mínima entre actualizaciones del mismo, y la frecuencia de inserción en la base de datos, que por defecto ocurre cada 60 minutos.

Capturas de pantalla:
Flujo de la app (**LoginActivity** <-> **WeatherListActivity** -> **WeatherDetailActivity**)
-> Unidireccional (De una actividad a otra)
<-> Bidireccional (De una actividad a otra, y con el botón de atrás puedes volver)

LoginActivity





WeatherListActivity

WeatherDetailActivity