Aplicaciones Telemáticas
El Tiempo

Sergio Jiménez Roger

Xavier Bernat Correas

1. Manual de funcionamiento

Primera pantalla

El elemento principal es un Spinner el cual al pinchar en él aparecen todas las provincias de España ordenadas por comunidades autónomas (estas van en orden alfabético).

Hay un botón Seleccionar para seleccionar la provincia y pasar a la segunda activity.

Segunda Pantalla

Primero te indica arriba la provincia en la que te encuentras.

Spinner con la misma función que en la pantalla anterior, pero en el spinner ahora salen los municipios de la provincia seleccionada en la pantalla anterior, ordenados alfabéticamente.

Tercera pantalla

Lo primero que vemos es el municipio y provincia de donde queremos ver el tiempo. Además de la fecha del día en el que estás.

Podemos observar las temperaturas máximas y mínimas de cada día la próxima semana. También la imagen que aparece al lado de estos datos muestra el estado del cielo ese día.

Finalmente, el icono de información, a la derecha de cada día, es un botón el cual si pulsas te envía a la información más específica de ese día, situada en el siguiente Activity.

Cuarta Pantalla

Esta última pantalla es muy simple, pues en ella se muestra por pantalla los datos específicos del día seleccionado en la anterior.

Donde los diferentes iconos son las temperaturas, probabilidad de precipitación, las características del viento y el estado del cielo.

Además incluye dos botones desde los cuales se puede acceder a las activitys de selección de la provincia y el municipio.

En el caso del municipio se podría cambiar entre los municipios de la provincia seleccionada inicialmente.

En el caso de las provincias la función es la misma que al iniciar la App.

Explicación del código

Antes de empezar con las diferentes activities que hemos realizado, cabe comentar que para las peticiones hemos reutilizado el bloque API_REST que se nos proporcionó en la práctica 4. De este bloque solo hemos modificado que en la línea, charset hemos puesto UTF-8 para poder leer los caracteres: tildes, ñ, ç …

```
new InputStreamReader(conn.getInputStream() , "utf-8" ));
```

Primera Activity (MainActivity)

```
class ServiciosWebEncadenados extends Thread {

    String url_inicial;

    // constructor
    ServiciosWebEncadenados(String url_inicial) { this.url_inicial = url_inicial; }

    // tarea a ejecutar en hilo paralelo e independiente

    @Override public void run() {
        // Gestiónese oportunamente las excepciones
        try {
            // Primera peticion
            final String respuesta = API_REST(url_inicial);
            // Impresión de resultados en el hilo de la UI (User Interface thread): runOnUiThread
            runOnUiThread(() -> ListProvincias(respuesta));

        } catch (Exception e) {
            Toast.makeText( context: MainActivity.this,  text: "Se ha produciodo un ERROR ", Toast.LENGTH_LONG).show();
        }
    } // run

} // ServiciosWebEncadenados
```

Clase ServiciosWebEncaminado. Esta clase va a ser una constante en las activities pero con variaciones. En este caso solo hay una petición para hacer la lista de las provincias.

Spinner para Provincias

```
public void ListProvincias(String provincias) {
    Log.i(TAG,  msg: "informacion");

    spinner_prov = (Spinner) findViewById(R.id.spiner_prov);

    int i = 0;
    try {    /**Con el do while nos recorremos todo el Json para sacar los nombres de las provincias y añadirlas a la list
        JSONArray arr = new JSONArray(provincias);
        do {
            String prov = arr.getJSONObject(i).getString( name: "nm");
            lista_Prov.add(prov);
            i += 1;
        } while (i < arr.length());

        arrayAdapter_Prov = new ArrayAdapter<String>( context: this, android.R.layout.simple_spinner_item, lista_Prov);
        spinner_prov.setAdapter(arrayAdapter_Prov);
```

Para esto hacemos la petición a la página web:

```
"https://raw.githubusercontent.com/IagoLast/pselect/master/data/provincias.json"
```

La cual contiene todas las Provincias de España con sus respectivas “id”. Para ello se hace un bucle do-while para sacar todos los nombres y meterlos en la lista que se utiliza en el spinner

```
//Seleccionamos la id de la provincia selecionada
spinner_prov.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener(){
    @Override       //Una vez selecionado un elemento del spinner obtenemos su nombre y la id
    public void  onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        try {
            id_prov = arr.getJSONObject(position).getString( name: "id");
            nm_prov = arr.getJSONObject(position).getString( name: "nm");

        } catch (JSONException e) {
            Toast.makeText( context: MainActivity.this,  text: "Se ha producido un error", Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {
        Toast.makeText( context: MainActivity.this,  text: "NothingSelected", Toast.LENGTH_SHORT).show();
    }
});
```

En esta parte lo que hacemos es obtener el nombre y la id de la provincia seleccionada.

Segunda Activity (ActivityMun)

Es muy parecida a la primera, cambiando principalmente que ahora queremos sacar el municipio, para ello hemos utilizado la página:

```
"https://raw.githubusercontent.com/IagoLast/pselect/master/data/municipios.json"
```

Y el cambio más significativo del código es que a la hora de obtener la lista hemos comparado los dos primeros dígitos de las id's de los municipios con la id de la provincia.

Por otra parte, para obtener la id final, sacamos el nombre del pueblo y lo comparamos uno a uno hasta que sea el seleccionado en el spinner, de ahí obtenemos su posición en la matriz y sacamos su id.

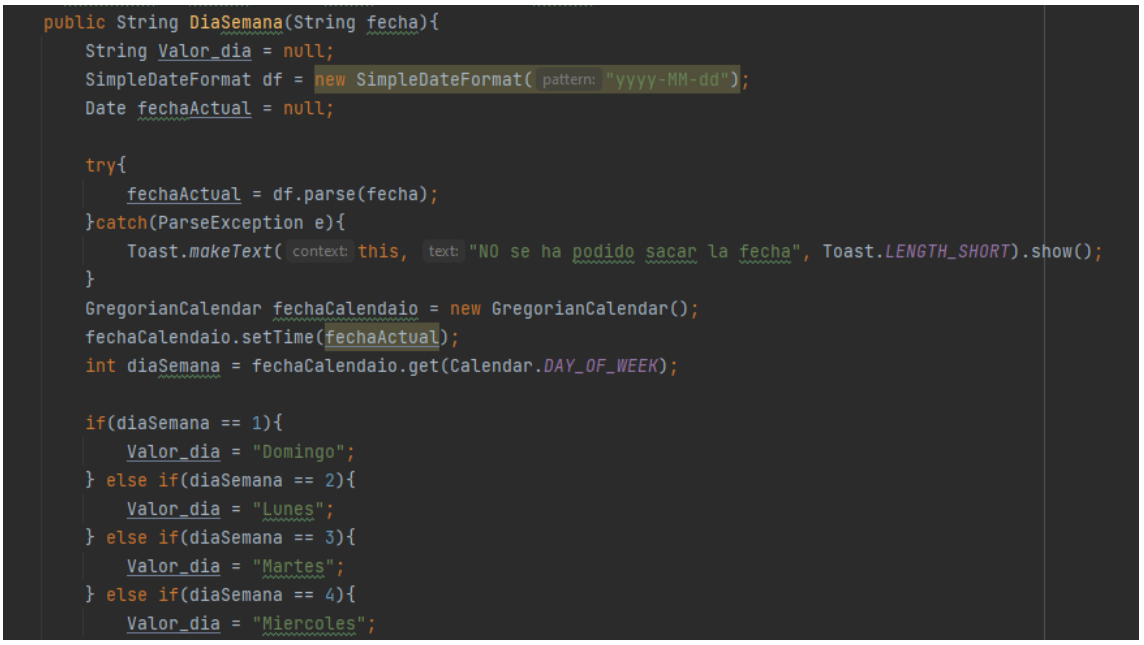
Tercera Activity (ActivityDias)

Esta activity es bastante extensa, por una parte tiene las peticiones al igual que las anteriores, pero en esta ya hay dos peticiones al igual que la práctica 4. La variedad ha sido a la hora de poder sacar por pantalla la imagen del estado del cielo de cada día, para ello hemos utilizado la estructura switch-case:

```
public void Imagenes(String estado_cielo, String dias){
    Drawable x = null;
    switch (estado_cielo) {
        case ("Bruma"):
            x = getResources().getDrawable(R.drawable.bruma);
            break;
        case ("Calima"):
            x = getResources().getDrawable(R.drawable.calima);
            break;
        case ("Cubierto"):
            x = getResources().getDrawable(R.drawable.cubierto);
            break;
        case ("Cubierto con lluvia"):
            x = getResources().getDrawable(R.drawable.cubiertolluvia);
            break;
        case ("Cubierto con lluvia escasa"):
            x = getResources().getDrawable(R.drawable.cubiertolluviaescasa);
            break;
```

También para obtener los datos de los 7 días de la semana, hemos hecho una array para cada String de información

Finalmente, para acabar la activity, cómo se puede observar en la tercera pantalla, mostramos que dia de la semana es, para ello hemos utilizado esta parte del código



Donde a través de la fecha que hay en el array y utilizando la función GregorianCalendar. También con un código semblante hemos obtenido el més en el que nos encontramos.

Cuarta Activity (ActivityFin)

En esta, simplemente obtenemos los valores obtenidos en la activity anterior y se muestran por pantalla.

En la cual reutilizamos el case para los estados del cielo de cada día.