

## Selecciones personalizables

Cualquier tipo de control es personalizable. Es decir, podemos definir la estructura de cada fila de la lista de selección como nosotros queramos. Para integrarlo en un Spinner o ListView hay que reescribir el funcionamiento del adaptador (`ArrayAdapter<T>`) de modo que cuando un selector demande los datos de las filas, devuelva una vista que sea la seleccionada. Para ello entender:

- El inflador: se pueden crear Vista (View) en tiempo de ejecución a partir de un fichero de recursos XML. Se puede crear un widget personalizado e “inflarlo” (crearlo) habiendo definido su estructura mediante un Layout definido en XML.
- Se puede hacer “override”. En concreto, del adaptador hay que reprogramar dos métodos `getDropDownView( int position, View convertView, ViewGroup parent)` y `getView(int position, View convertView, ViewGroup parent)` que retorna la vista personalizada de una determinada fila. Cuando el Spinner invoque a estos métodos debemos pasarle la fila personalizada de la posición que nos indique el parámetro posición.

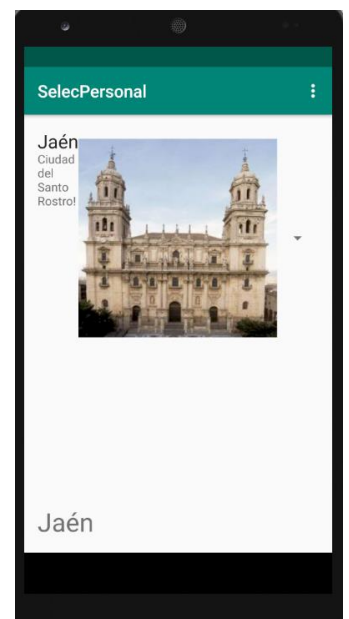
Veamos un ejemplo práctico. Usaremos un Spinner personalizado que muestre las ciudades andaluzas, pero con un formato personalizado..

La app tendrá dos ficheros XML, el propio de la actividad, con un Spinner y una caja de texto donde se escribirá el resultado de la elección, y el fichero XML con el diseño de una fila del Spinner. EL fichero de la fila llamado `lineasspinner.xml` tiene el siguiente código:

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Jaén, ciudad del Santo Rostro"
        android:id="@+id/descripcion"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:ellipsize="none"
        android:scrollHorizontally="false"
        android:layout_marginTop="26dp"
        android:layout_alignBottom="@+id/imagenCiudad"
        android:layout_alignRight="@+id/nombre"
        android:layout_alignEnd="@+id/nombre"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Jaén"
        android:id="@+id/nombre"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"/>
```



```

<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/jaen"
    android:id="@+id/imagenCiudad"
    android:layout_toRightOf="@+id/descripcion"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"/>
</RelativeLayout>

```

En el fichero java de la actividad, vamos a crear una subclase de la clase ArrayAdapter, con los dos métodos reescritos, esta clase se apoyará en tres arrays de datos para crear el objeto View que se mandará al Spinner, cada array tiene una información para los diferentes campos que componen la fila del Spinner:

```

String [] ciudades ={"Jaén", "Córdoba", "Sevilla", "Huelva", "Cádiz", "Málaga",
    "Granada", "Almería"};
String [] descripciones ={"Ciudad del Santo Rostro!", "Qué gran ciudad", "Ciudad
    del nuevo mundo", "Ciudad gastronómica", "Ciudad encantada", "Ciudad de la
    luz", "Ciudad encantada", "La gran desconocida"};
int imagenes [] = {R.drawable.jaen, R.drawable.cordoba, R.drawable.sevilla,
    R.drawable.huelva, R.drawable.cadiz, R.drawable.malaga, R.drawable.granada,
    R.drawable.almeria};

```

```

public class AdaptadorPersonalizado extends ArrayAdapter<String> {
    public AdaptadorPersonalizado(Context ctx, int txtViewResourceId, String[]
        objects){
        super(ctx, txtViewResourceId, objects);
    }

    @Override
    public View getDropDownView(int position, View convertView, ViewGroup parent){
        return crearFilaPersonalizada(position, convertView, parent);
    }

    @Override
    public View getView(int pos, View convertView, ViewGroup parent){
        return crearFilaPersonalizada(pos, convertView, parent);
    }

    public View crearFilaPersonalizada(int position, View convertView, ViewGroup
parent){
        LayoutInflater inflater = getLayoutInflater();
        View miFila = inflater.inflate(R.layout.lineaspiner, parent, false);

        TextView nombre = (TextView) miFila.findViewById(R.id.nombre);
        nombre.setText(ciudades[position]);

        TextView descripcion = (TextView) miFila.findViewById(R.id.descripcion);
        descripcion.setText(descripciones[position]);

        ImageView imagen = (ImageView) miFila.findViewById(R.id.imagenCiudad);
        imagen.setImageResource(imagenes[position]);
        return miFila;
    }
}

```

Observa cómo los dos métodos reescritos invocan a una tercera función `crearFilaPersonalizada()` que “*infla*” la vista del fichero xml `lineaspiner.xml` para crear un objeto View con los datos de los arrays y lo retorna.

Cuando crees el Spinner, acuérdate entonces de enviarle el adaptador de esta subclase:

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    Spinner selectorCiudades = (Spinner) findViewById(R.id.spin);  
    AdaptadorPersonalizado a=new AdaptadorPersonalizado(this,  
        R.layout.lineaspiner, ciudades);  
    selectorCiudades.setAdapter(a);  
    selectorCiudades.setOnItemClickListener(this);  
}
```

De esta forma quedan resultados tan espectaculares como estos:

