

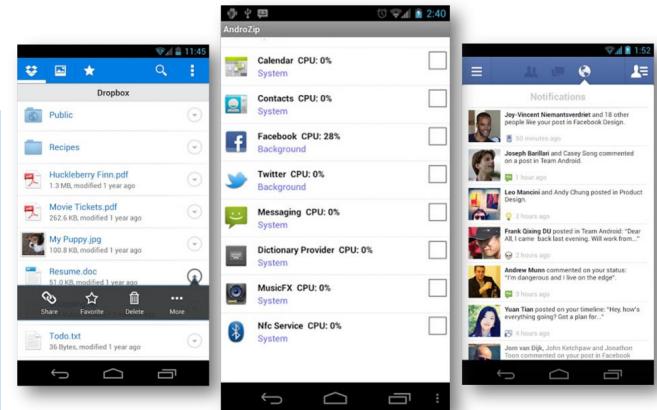
U.T.6 PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES. LISTAS



LISTAS (*ListView*)

Las listas son muy utilizadas en las aplicaciones de Android, muestran múltiples elementos (items). Cada elemento se muestra a través de un componente visual diferente e independiente de los propios datos que muestra y de su origen.

Un *ListView* puede ser considerado un layout, porque incluye otros componentes como mínimo un *textView*.



ListView: widget que muestra una lista de elementos con desplazamiento vertical.

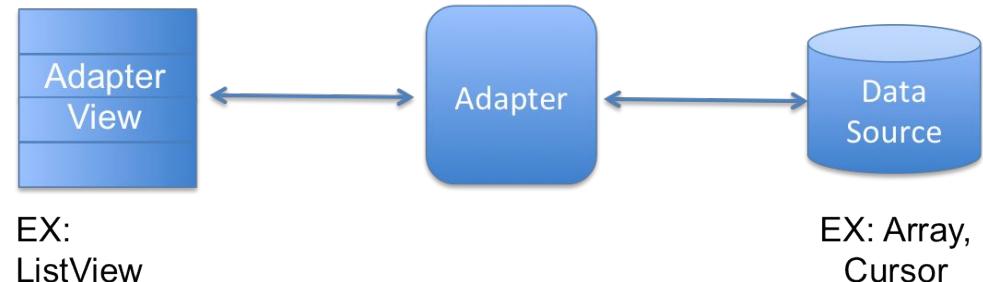
Ej. añade un listView a un layout

The screenshot shows the Android Studio interface. On the left, there's a project navigation pane with a tree view of files and folders. A tooltip for the **ListView** component is displayed, stating: "A view that shows items in a vertically scrolling list. The items come from the ListAdapter associated with this view." Below the navigation pane is the main content area, which contains a palette of various Android UI components. The **ListView** component is highlighted with a blue selection bar. To the right of the palette is a preview window showing a smartphone screen with a list titled "Lista de los meses del año" containing eight items, each with a sub-item below it. At the bottom of the interface, there are tabs for "Gradle Scripts" and "AndroidManifest.xml".

Construir un ListView

Para construir un ListView se utiliza un adaptador, a través del cual ListView es capaz de obtener cada uno de los componentes gráficos que debe mostrar.

Un adaptador es un objeto que implementa la interfaz Adapter. Puede ser `ArrayAdapter`, `CursorAdapter`, `SimpleCursorAdapter`...



La clase **ArrayAdapter**

ArrayAdapter recibe un array de objetos y para cada uno construye un TextView para mostrar el resultado.

Por tanto, al conectar un ListView con un ArrayAdapter veremos la lista de los elementos del array que le hayamos dado al adaptador.

```
String meses[]={"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
 "Agosto","Septiembre", "Octubre","Noviembre","Diciembre"};
```

```
ArrayAdapter aa = new ArrayAdapter <String> (this,android.R.layout.simple_list_item_1,
meses);
```

El constructor de [ArrayAdapter](#)

[ArrayAdapter\(Context context, int resource, T\[\] objects\)](#)

recibe tres parámetros:

1. El objeto que se usará como contexto para la creación de las vistas.
2. El layout que se iniciará para cada item. Debe ser un TextView.
3. El array de elementos de tipo T (o, alternativamente, un List<T>) del que se obtendrán los items.

Layout predefinidos

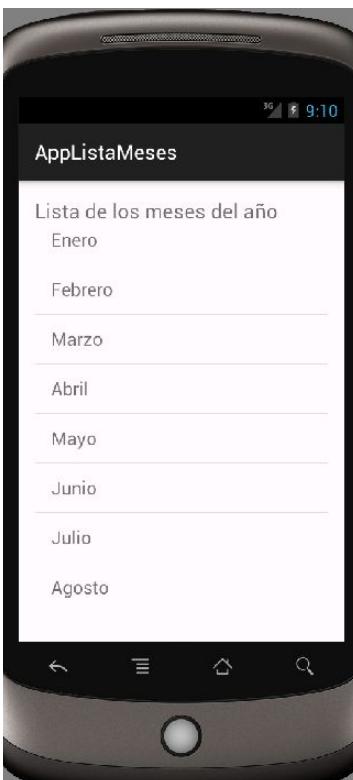
Android proporciona layout predefinidos que ahorran la creación del layout, y que además deja espacio suficiente para poder seleccionar cada elemento en función del tipo y configuración del dispositivo.

`android.R.layout.simple_list_item_1`. Es un layout predefinido que contiene un `TextView`

`android.R.layout.simple_list_item_2`, que incluye dos `TextView`

Ejemplo: Lista de los meses del año

utiliza [android.R.layout.simple_list_item_1](#)



```
package cintas.castro.applistameses;

import ...

public class AppListaMeses extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_app_lista_meses);
        String meses[]={"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto","Septiembre",
                        "Octubre","Noviembre","Diciembre"};
        ArrayAdapter<String> aa;
        aa = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,meses);
        ListView lv = (ListView) findViewById(R.id.listView);
        lv.setAdapter(aa);
    }
}
```

Ejemplo: Evento al pulsar sobre un ítem.

utiliza [setOnItemClickListener\(\)](#) método [onItemClick\(\)](#)

```
public class AppListaMeses extends ActionBarActivity {  
    ArrayAdapter<String> aa;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_app_lista_meses);  
        String meses[]={"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre",  
                      "Octubre", "Noviembre", "Diciembre"};  
  
        aa = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,meses);  
        ListView lv = (ListView) findViewById(R.id.listView);  
        lv.setAdapter(aa);  
        lv.setAdapter(aa);  
        lv.setOnItemClickListener(  
            new AdapterView.OnItemClickListener() {  
                public void onItemClick(AdapterView<?> parent, View view,  
                                    int position, long id) {  
  
                    Toast.makeText(getApplicationContext(),  
                        "Pulsado " + position + ", " + id +", "+aa.getItem(position),  
                        Toast.LENGTH_SHORT).show();  
                }  
            });  
    }  
}
```

[setOnItemClickListener\(\)](#):
recibe un
[OnItemClickListener](#)
(interfaz interno de
[AdapterView](#), superclase
de [ListView](#)). El interfaz
fuerza a la implementación
de un único método,
[onItemClick\(\)](#), que recibe
información sobre el
evento.

Para acceder al dato del
ítem utiliza el método
[getItem\(position\)](#) del
[ArrayAdapter](#).

El método `getView`

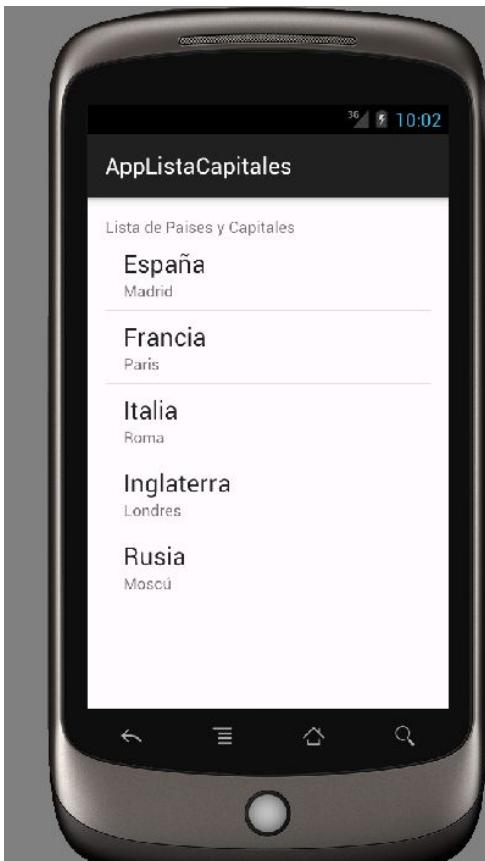
Es el método más importante de la clase `ArrayAdapter`, devuelve la vista (control) que mostrará cada elemento de la vista.

Se utiliza cuando lo que queremos visualizar en el `ListView` es algo más complejo que un simple `TextView`. Por ejemplo para visualizar una imagen o simplemente 2 etiquetas.

Para conseguirlo, necesitaremos hacer una subclase de `ArrayAdapter`, sobreescibir el método `getView()` y construir nosotros la vista del item.

Ejemplo: Lista Paises y Capitales

utiliza [android.R.layout.simple_list_item_2](#)



Project: 1: App

Structure: `AppListaCapitales.java` `MyArrayAdapter.java` `Paises.java`

```
package cintas.castro.applistacapitales;

import ...

public class AppListaCapitales extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_app_lista_capitales);

        ArrayList <Paises> listaPaises=new ArrayList<Paises>();
        listaPaises.add(new Paises("España", "Madrid"));
        listaPaises.add(new Paises("Francia", "París"));
        listaPaises.add(new Paises("Italia", "Roma"));
        listaPaises.add(new Paises("Inglaterra", "Londres"));
        listaPaises.add(new Paises("Rusia", "Moscú"));

        MyArrayAdapter aa;
        aa = new MyArrayAdapter(this,listaPaises);
        ListView lv = (ListView) findViewById(R.id.listView);
        lv.setAdapter(aa);
    }
}
```

clase que hereda de
ArrayAdapter y contiene
método getView

Ejemplo: Lista Paises y Capitales

utiliza [android.R.layout.simple_list_item_2](#)

1. Crea la clase Paises.

```
AppListaCapitales.java x | MyArrayAdapter.java x | 
package cintas.castro.applistacapitales;

/* Created by Nina on 08/01/2016.
public class Paises {
    String nombrePais;
    String capital;

    public Paises(String nombrePais, String capital) {
        this.nombrePais = nombrePais;
        this.capital = capital;
    }

    public String getNombrePais() {
        return nombrePais;
    }

    public String getCapital() {
        return capital;
    }

    public void setNombrePais(String nombrePais) {
        this.nombrePais = nombrePais;
    }

    public void setCapital(String capital) {
        this.capital = capital;
    }
}
```

carga el layout
y asocia los
datos que se
visualizarán
en cada
textView.

2. Crea la clase MyArrayAdapter.
3. sobrescribe el método getView

```
/*
public class MyArrayAdapter extends ArrayAdapter<Paises> {

    public MyArrayAdapter(Context context, Paises[] paises) {
        super(context, 0, paises);
    }

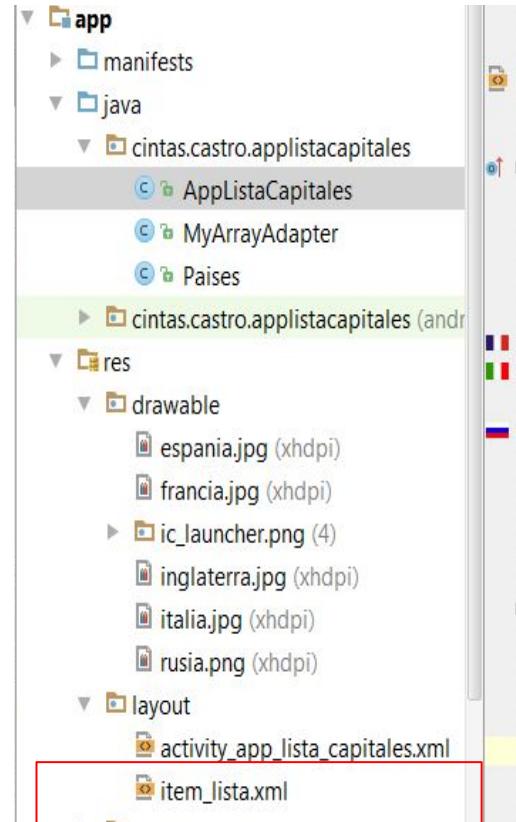
    public MyArrayAdapter(Context context, List<Paises> LPaises) {
        super(context, 0, LPaises);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        Paises p = getItem(position);
        View v;
        v = LayoutInflater.from(getContext()).inflate(
            android.R.layout.simple_list_item_2,
            parent,
            false);
        TextView tv;
        tv = (TextView) v.findViewById(android.R.id.text1);
        tv.setText(p.nombrePais);
        tv = (TextView) v.findViewById(android.R.id.text2);
        tv.setText(p.capital);

        return v;
    } // getView
}
```

Ej: Lista Paises, Capitales y Banderas

utiliza un layout creado: R.layout.item_lista.xml

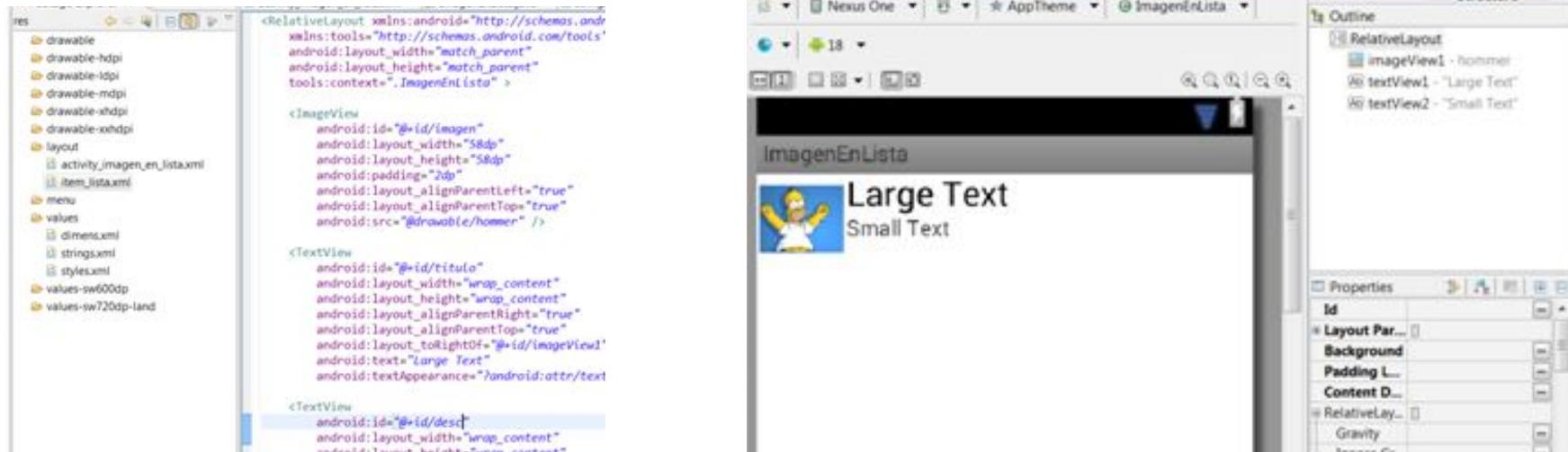


```
public class AppListaCapitales extends ActionBarActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_app_lista_capitales);  
  
        ArrayList <Paises> listaPaises=new ArrayList<>();  
        listaPaises.add(new Paises("España", "Madrid", R.drawable.espania));  
        listaPaises.add(new Paises("Francia", "París", R.drawable.francia));  
        listaPaises.add(new Paises("Italia", "Roma", R.drawable.italia));  
        listaPaises.add(new Paises("Inglaterra", "Londres", R.drawable.inglaterra));  
        listaPaises.add(new Paises("Rusia", "Moscú", R.drawable.rusia));  
  
        MyArrayAdapter aa;  
        aa = new MyArrayAdapter(this,listaPaises);  
        ListView lv = (ListView) findViewById(R.id.listView);  
        lv.setAdapter(aa);  
    }  
}
```

Crear un layout para cada item

Cuando lo que queremos mostrar en cada item no se puede representar con los layouts predefinidos, debemos crear un layout con los componentes (Widgets) necesarios y cargarlo el método `getView` con:

`View fila=inflater.inflate(R.layout.item_lista, parent, false);`



Ej: Lista Paises, Capitales y Banderas

1. Añade un atributo bandera a la clase Paises.

```
/*
 * Created by Nina on 08/01/2016.
 */
public class Paises {
    String nombrePais;
    String capital;
    int bandera;

    public Paises(String nombrePais, String capital, int bandera) {
        this.nombrePais = nombrePais;
        this.capital = capital;
        this.bandera=bandera;
    }

    public String getNombrePais() { return nombrePais; }

    public void setBandera(int bandera) { this.bandera = bandera; }

    public int getBandera() {
        return bandera;
    }

    public String getCapital() { return capital; }

    public void setNombrePais(String nombrePais) { this.nombrePais = nombrePais; }

    public void setCapital(String capital) { this.capital = capital; }
}
```

2. Añade la imagen en el método getView

```
public class MyArrayAdapter extends ArrayAdapter<Paises> {

    public MyArrayAdapter(Context context, Paises[] paises) { super(context, 0, paises); }

    public MyArrayAdapter(Context context, List<Paises> LPaises) { super(context, 0, LPaises); }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

        Paises p = getItem(position);
        View v;
        v = LayoutInflater.from(getContext()).inflate(R.layout.item_lista,
            parent,
            false);

        TextView tv;
        tv = (TextView) v.findViewById(R.id.textView2);
        tv.setText(p.nombrePais);
        tv = (TextView) v.findViewById(R.id.textView3);
        tv.setText(p.capital);
        ImageView im= (ImageView) v.findViewById(R.id.imageView);
        im.setImageResource(p.bandera);
        return v;
    }
}
```