



La Clase `ListView`

CONCEPTOS

La clase `ListView` muestra una colección de vistas desplazables verticalmente, donde cada vista se coloca inmediatamente debajo de la vista anterior en la lista. Para un enfoque más moderno, flexible y de rendimiento para mostrar listas, se puede utilizar `RecyclerView`.

Para mostrar una lista, puede incluir una vista de lista en su archivo XML de diseño:

```
<ListView
    android:id="@+id/list_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

Donde:

```
public class ListView
    extends AbsListView
    java.lang.Object
        android.view.View
            android.view.ViewGroup
                android.widget.AdapterView<android.widget.ListAdapter>
                    android.widget.AbsListView
                        android.widget.ListView
```

Subclase directa:

```
ExpandableListView
```

Un componente `ListView` es una vista adaptadora que no conoce los detalles, como el tipo y el contenido, de las vistas que contiene. En su lugar, el `ListView` solicita vistas a un `ListAdapter` según sea necesario, por ejemplo para mostrar nuevas vistas a medida que el usuario se desplaza hacia arriba o hacia abajo.

Para mostrar elementos en la lista, se invoca a `setAdapter(ListAdapter)` para asociar un adaptador con la lista.

Para mostrar una vista más personalizada para cada elemento en su conjunto de datos, se implanta un `ListAdapter`. Por ejemplo, heredar de `BaseAdapter` y crear y configurar la vista para cada elemento de datos en `getView (...)`:

```
private class MyAdapter extends BaseAdapter {
    // sobrescribir los métodos abstractos aquí.
    @Override
    public View getView(int position, View convertView, ViewGroup container) {
        if (convertView == null) {
            convertView = getLayoutInflater().inflate(R.layout.list_item, container, false);
        }
        ((TextView) convertView.findViewById(android.R.id.text1))
            .setText(getItem(position));
        return convertView;
    }
}
```

El componente `ListView` intenta reutilizar objetos de vista para mejorar el rendimiento y evitar un retraso en respuesta a las volutas del usuario. Para aprovechar esta función, compruebe si `convertView` proporcionado a `getView (...)` es nulo antes de crear o inflar un nuevo objeto de vista. Consulte [Hacer que ListView Scrolling Smooth busque más formas de garantizar una experiencia de usuario fluida](#).



DESARROLLO

EJEMPLO 1.

Este ejemplo muestra una lista con la clase `ListView`. La lista incluye una imagen y texto, los cuales son seleccionables con un escucha `setOnItemClickListener` para mostrar un mensaje `Toast`.

Paso 1. Crear un nuevo proyecto `Listalimagenes`. En el archivo Java predeterminado `MainActivity.java`, capturar el siguiente código:

```
import java.util.ArrayList;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.*;

public class MainActivity extends Activity {
    private ListView lv;
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.listado);
        ArrayList<ListaEntrada> al = new ArrayList<ListaEntrada>();
        al.add(new ListaEntrada(R.drawable.buho,
"BUHO", "Búho es el nombre común de aves de la familia Strigidae, del orden de las estrigiformes o
aves rapaces nocturnas. Habitualmente designa especies que, a diferencia de las lechuzas, tienen
plumas alzadas que parecen orejas."));
        al.add(new ListaEntrada(R.drawable.colibri,
"COLIBRÍ", "Los troquilinos (Trochilinae) son una subfamilia de aves apodiformes de la familia
Trochilidae, conocidas vulgarmente como colibríes, quindes, tucusitos, picaflores, chupamirtos,
chuparrosas, huichichiquis (idioma nahuatl), mainumby (idioma guaraní) o guanumby."));
        al.add(new ListaEntrada(R.drawable.cuervo,
"CUERVO", "El cuervo común (Corvus corax) es una especie de ave paseriforme de la familia de los
córvidos (Corvidae). Presente en todo el hemisferio septentrional, es la especie de córvido con la
mayor superficie de distribución."));
        al.add(new ListaEntrada(R.drawable.flamenco,
"FLAMENCO", "Los fenicopteriformes (Phoenicopteriformes), los cuales reciben el nombre vulgar de
flamencos, son un orden de aves neognatas, con un único género viviente: Phoenicopterus."));
        al.add(new ListaEntrada(R.drawable.kiwi,
"KIWI", "Los kiwis (Apteryx, gr. 'sin alas') son un género de aves paleognatas compuesto por cinco
especies endémicas de Nueva Zelanda.1 2 Son aves no voladoras pequeñas, aproximadamente del tamaño
de una gallina."));
        al.add(new ListaEntrada(R.drawable.loro,
"LORO", "Las Psitácidas (Psittacidae) son una familia de aves psitaciformes llamadas comúnmente
loros o papagayos, e incluye a los guacamayos, las cotorras, los periquitos, los agapornis y formas
afines."));
        al.add(new ListaEntrada(R.drawable.pavo,
"PAVO", "Pavo es un género de aves galliformes de la familia Phasianidae, que incluye dos especies,
el pavo real común (Pavo cristatus) y el pavo real cuelliverde (Pavo muticus).1"));
        al.add(new ListaEntrada(R.drawable.pinguino,
"PINGÜINO", "Los pingüinos (familia Spheniscidae, orden Sphenisciformes) son un grupo de aves
marinas, no voladoras, que se distribuyen únicamente en el Hemisferio Sur, sobre todo en sus altas
latitudes."));
        lv = (ListView) findViewById(R.id.xlv_listado);
        lv.setAdapter(new ListaAdapter(this, R.layout.activity_main, al){
            public void onEntrada(Object o, View v) {
                if (o != null) {
                    TextView texto_superior_entrada = (TextView)
v.findViewById(R.id.textView_superior);
                    if (texto_superior_entrada != null)
                        texto_superior_entrada.setText(((ListaEntrada) o).get_textoEncima());
                    TextView texto_inferior_entrada = (TextView)
v.findViewById(R.id.textView_inferior);
                    if (texto_inferior_entrada != null)
```



```

        texto_inferior_entrada.setText(((ListaEntrada) o).get_textoDebajo());
        ImageView imagen_entrada = (ImageView) v.findViewById(R.id.imageView_imagen);
        if (imagen_entrada != null)
            imagen_entrada.setImageResource(((ListaEntrada) o).get_idImagen());
    }
}
});
lv.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> av, View view, int i, long l) {
        ListaEntrada le = (ListaEntrada) av.getItemAtPosition(i);
        CharSequence cs = "Seleccionado: " + le.get_textoDebajo();
        Toast t = Toast.makeText(MainActivity.this, cs, Toast.LENGTH_SHORT);
        t.show();
    }
});
}
}
}

```

Paso 2. En la carpeta `java/com.example.mipaquete`, crear el archivo `ListaAdapter.java`, y capturar el siguiente código:

```

import java.util.ArrayList;
import android.content.Context;
import android.view.*;
import android.widget.BaseAdapter;
public abstract class ListaAdapter extends BaseAdapter {
    private ArrayList<?> al;
    private int R_layout_IdView;
    private Context c;
    public ListaAdapter(Context c, int R_layout_IdView, ArrayList<?> al) {
        super();
        this.c = c;
        this.al = al;
        this.R_layout_IdView = R_layout_IdView;
    }
    public View getView(int i, View v, ViewGroup vg) {
        if (v == null) {
            LayoutInflater vi = (LayoutInflater)
c.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            v = vi.inflate(R_layout_IdView, null);
        }
        onEntrada (al.get(i), v);
        return v;
    }
    public int getCount() {
        return al.size();
    }
    public Object getItem(int j) {
        return al.get(j);
    }
    public long getItemId(int k) {
        return k;
    }
    public abstract void onEntrada (Object o, View v);
}

```

Paso 3. En la carpeta `java/com.example.mipaquete`, crear el archivo `ListaEntrada.java`, y capturar el siguiente código:

```

public class ListaEntrada {
    private int idim;
    private String a;
    private String d;
    public ListaEntrada (int idI, String sa, String sd) {
        this.idim = idI;
        this.a = sa;
    }
}

```



```
        this.d = sd;
    }
    public String get_textoEncima() {
        return a;
    }
    public String get_textoDebajo() {
        return d;
    }
    public int get_idImagen() {
        return idim;
    }
}
```

Paso 4. En la carpeta `res/layout`, en el archivo XML predeterminado `activity_main.xml`, capturar el siguiente código:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <ImageView
        android:id="@+id/imageView_imagen"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:adjustViewBounds="true"
        android:scaleType="fitXY"
        android:contentDescription="Descripción del contenido de la imagen"
        android:src="@android:drawable/ic_menu_gallery" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical" >
        <TextView
            android:id="@+id/textView_superior"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Large Text"
            android:textAppearance="?android:attr/textAppearanceLarge" />
        <TextView
            android:id="@+id/textView_inferior"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Small Text"
            android:textAppearance="?android:attr/textAppearanceSmall" />
    </LinearLayout>
</LinearLayout>
```

Paso 5. En la carpeta `res/layout`, crear el archivo `listado.xml` y capturar el siguiente código:

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

Paso 6. En la carpeta `res/drawable`, guardar las imágenes JPG siguientes:
búho, colibrí, cuervo, flamenco, kiwi, loro, pavo y pingüino.



Paso 7. Por último, al ejecutar la aplicación se debe mostrar una imagen similar a la siguiente. El usuario puede desplazar la lista para mostrar más elementos. Si se digita una opción, se muestra un mensaje `Toast` con información de la opción correspondiente.



EJERCICIO.

Paso 1. Diseñar una aplicación que contenga una plantilla con un botón. Al digitar el botón se invoca, con un `Intent`, al mismo `ListView` del ejemplo anterior, mostrando un tema diferente con otras imágenes y texto.