Creación de interfaces sencillas en Android: Listas

En Android, una lista representa un conjunto de elementos que se muestran unos a continuación de los otros.

Algunas aplicaciones de las listas, son por ejemplo en agendas, lista de contactos, lista de precios, etc.

Gustavo Márquez Flores Facultad de Ciencias

Creación de interfaces sencillas en Android

Cada elemento de una lista puede poseer de una a tres líneas como máximo y cada fila de una lista puede personalizarse mediante distintos componentes (***TextView***, ***Button***, ***ImageView***...).

Existen dos métodos para crear una lista. La actividad que integra la lista puede:

- Heredar de la clase ***ListActivity***.
- O heredar de la clase ***Activity***.

Para insertar datos en una lista, se utiliza un ***adapter*** (clase adaptador) que permite insertar datos a una lista.

Android proporciona varios tipos de adaptadores, entre ellos:

- ***ArrayAdapter***: inserta datos en una lista a partir de un array o de una colección.
- ***SimpleCursorAdapter***: inserta datos en una lista a partir de una base de datos.

Se puede también crear un ***adapter*** propio, simplemente heredando de la clase ***BaseAdapter*** o bien de un adaptador ya existente.

Creación de interfaces sencillas en Android

Crear una lista

- **ListActivity**

El primer método consiste en crear una lista que herede de la clase ***ListActivity***.

Para comenzar, crear un archivo *XML* que represente una vista que contenga únicamente una lista (carpeta ***layout***).

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

    <ListView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@android:id/list" />

</LinearLayout>
```

Creación de interfaces sencillas en Android

Para poder manipular una ***ListView*** gracias a una **ListActivity**, la lista debe tener, necesariamente, un identificador: ***@android:id/list***.

A continuación, declarar una actividad que herede de la clase ***ListActivity***.

```
public class ListActivityActivity extends ListActivity {

@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

    }
}
```

De momento, la lista está vacía y no contiene ningún dato. Vamos a utilizar un array de cadenas de caracteres como origen de datos.

```
private String[] androidVersion = { "Cupcake", "Donut", "Eclair", "Froyo",
      "Gingerbread", "Honeycomb", "Ice Cream Sandwich", "Jelly Bean", "KitKat", "Lollipop" };
```

Creación de interfaces sencillas en Android

Crear ahora un ***ArrayAdapter*** para agregar en la lista, los datos que provienen del array que acabamos de definir:

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

El método que permite construir un ***ArrayAdapter*** recibe tres parámetros:

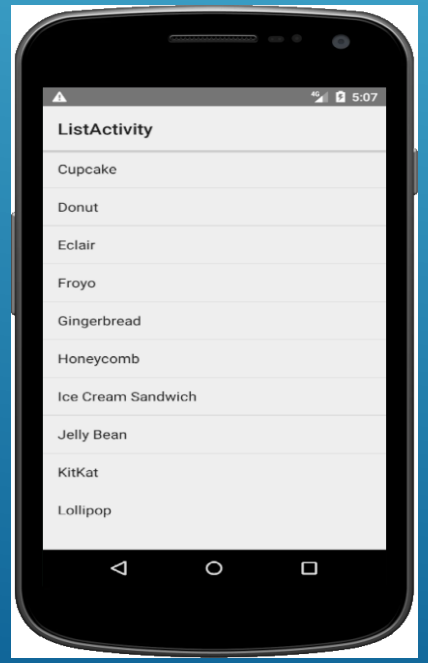
- El primer parámetro representa el contexto de la actividad en curso.
- El segundo parámetro representa el *layout* que se aplicará a cada fila de la lista. Se puede crear un *layout* personalizado o bien utilizar los que proporciona *Android*.
- El tercer parámetro representa el array de datos que queremos insertar en la lista.

Creación de interfaces sencillas en Android

La última etapa consiste en asociar el ***adapter*** que contiene los datos que queremos agregar, con la lista (***ListActivity***), mediante el método **setListAdapter** :

setListAdapter(adapter);

A continuación, ejecutar el ejemplo y comprobar el resultado obtenido:



Ver el programa Ch5_ListActivity

Gustavo Márquez Flores Facultad de Ciencias

Creación de interfaces sencillas en Android

- **ListView**

La segunda manera de crear una lista consiste en utilizar una simple actividad (la clase debe heredar de la clase *Activity*).

Para ello, crear el archivo *XML layout* que representa a la lista:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
...
    <ListView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/myList" />
...
</LinearLayout>
```

Este método ofrece la posibilidad de especificar el identificador que se desee.

Creación de interfaces sencillas en Android

Modificar la actividad para recuperar la instancia de la lista (mediante el método ***findViewById***):

```
ListView myList = (ListView) findViewById(R.id.myList);
```

Como en el caso anterior, el ejemplo utiliza un array de cadenas de caracteres así como un ***ArrayAdapter:***

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

Para terminar, vincular el ***adapter*** con la lista.

```
myList.setAdapter(adapter);
```

Se recomienda utilizar ***ListActivity*** si la vista solo contiene una lista.

Creación de interfaces sencillas en Android

Con ***ListView*** se permiten utilizar varias Listas en una vista.

Personalizar una lista

También se pueden crear ***adapters*** personalizados para gestionar mejor la visualización y los datos de una lista.

En seguida se muestra un ejemplo que permite mostrar una lista que contiene el nombre y el número de distintas versiones de *Android* mediante un ***adapter*** personalizado.

En primer lugar, crear una clase ***AndroidVersion*** que represente, simplemente, una versión de *Android*. Representa un renglón de la lista.

Creación de interfaces sencillas en Android

```
package com.eni.android.customadapter;

public class AndroidVersion {

    private String versionName;
    private String versionNumber;

    public String getVersionName() {
        return versionName;
    }

    public void setVersionName(String versionName) {
        this.versionName = versionName;
    }

    public String getVersionNumber() {
        return versionNumber;
    }

    public void setVersionNumber(String versionNumber)
{
        this.versionNumber = versionNumber;
    }

}
```

Creación de interfaces sencillas en Android

El ejemplo utilizará un ***adapter*** personalizado que hereda de la clase ***ArrayAdapter*** (utiliza un array para agregar los datos en la lista).
Para ello, crear en primer lugar un archivo *XML layout* que represente la lista:

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

    <ListView
        android:id="@+id/myList"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />

</LinearLayout>
```

Creación de interfaces sencillas en Android

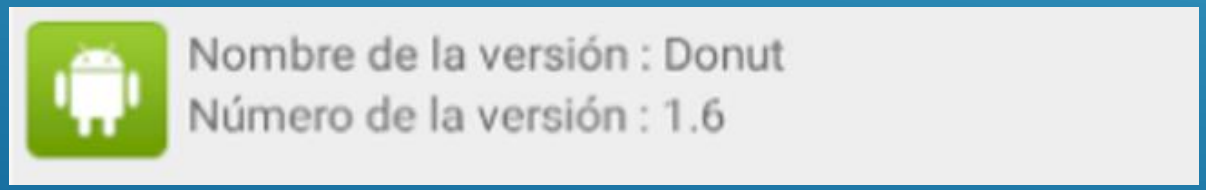
Para personalizar la lista y hacerla algo más rica y más agradable, es preciso crear un *layout* personalizado que servirá para especificar la interfaz correspondiente a cada fila de la lista:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="?android:attr/listPreferredItemHeight"
    android:padding="6dip" >

    <ImageView
        android:id="@+id/icon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="6dip"
        android:src="@drawable/list_icon" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="fill_parent"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/title"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:gravity="center_vertical"
            />

        <TextView
            android:id="@+id/description"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            />

    </LinearLayout>

</LinearLayout>
```

Creación de interfaces sencillas en Android

Cada fila de la lista se compone de los siguientes elementos:

- Un **LinearLayout** horizontal.
- Una imagen.
  - La imagen estará situada en la carpeta ***drawable***.
  - La imagen posee un espacio exterior (margin) derecho de 6dp.
- Un segundo LinearLayout (vertical) utilizado para mostrar ambas zonas de texto (título y descripción).
  - Un primer texto que indica el título de la fila.
  - Un segundo texto que indica la descripción de la fila.



A continuación, crear una clase que represente el ***adapter*** personalizado. Ésta tendrá las siguientes características:

Creación de interfaces sencillas en Android

- Hereda de la clase **ArrayAdapter**, de modo que los datos se almacenarán mediante un array.
- Cada elemento de la lista representa una versión de Android. Un renglón de la lista.
- La clase tiene los siguientes tres métodos :
  - Un constructor.
  - Un método ***getView***: cada llamada a este método permite recuperar una fila (dato y vista) de la lista que se encuentra en una posición determinada.
  - Un método ***getCount***: devuelve el número de elementos de la lista.

```
public class AndroidAdapter extends ArrayAdapter<AndroidVersion> {

    private ArrayList<AndroidVersion> androidVersionList;
    private Context context;
    private int viewRes;
    private Resources res;
```

Creación de interfaces sencillas en Android

```
public AndroidAdapter(Context context, int textViewResourceId,
                      ArrayList<AndroidVersion> versions) {
    super(context, textViewResourceId, versions);
    this.androidVersionList = versions;
    this.context = context;
    this.viewRes = textViewResourceId;
    this.res = context.getResources();
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View view = convertView;
    if (view == null) {
        LayoutInflater layoutInflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        view = layoutInflater.inflate(viewRes, parent, false);
    }
    final AndroidVersion androidVersion = androidVersionList.get(position);
    if (androidVersion != null) {
        final TextView title = (TextView) view.findViewById(R.id.title);
        final TextView description = (TextView) view.findViewById(R.id.description);
        final String versionName = String.format(res.getString(R.string.list_title), androidVersion.getVersionName());
        title.setText(versionName);
        final String versionNumber = String.format(res.getString(R.string.list_desc), androidVersion.getVersionNumber());
        description.setText(versionNumber);
    }
    return view;
}

@Override
public int getCount() {
    return androidVersionList.size();
}
```

Creación de interfaces sencillas en Android

El constructor sirve para almacenar los tres argumentos necesarios para implementar el ***adapter***:

- El contexto de la vista.
- El layout correspondiente a la vista personalizada, aplicado a cada fila de la lista.
- El array que representa los datos que se desea insertar en la lista.

El método ***getView(int position, View convertView, ViewGroup parent)*** posee las siguientes características:

- Este método permite recuperar la vista que se desea aplicar a una fila concreta (argumento ***position***).
- La vista que representa una fila de la lista se pasa como parámetro al método (argumento ***convertView***).
- La vista padre se pasa como parámetro (argumento ***parent***).

Creación de interfaces sencillas en Android

- La vista personalizada debe cargarse mediante el método ***inflate***. Una vez cargada, se pasa como parámetro al método ***getView*** (parámetro ***convertView***) para utilizarla en las siguientes llamadas, lo que reduce el número de llamadas al método ***inflate*** (llamada bastante costosa).
- A continuación, se recupera el dato correspondiente a la lista que se desea mostrar.
- Por último, se recuperan los dos ***TextView*** (título y descripción de una fila) para completarlas con los datos recuperados en la etapa anterior.

Sin olvidar el método ***getCount*** que permite indicar el número de elementos contenidos en la lista.

Para terminar, crear la actividad que permite inicializar la vista, el ***adapter*** personalizado y los datos.

Creación de interfaces sencillas en Android

```
public class CustomAdapterActivity extends Activity {

   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);

      ArrayList<AndroidVersion> androidList = new ArrayList<AndroidVersion>();

      initList(androidList);

      AndroidAdapter adapter = new AndroidAdapter(this, R.layout.list_layout, androidList);
      final ListView list = (ListView) findViewById(R.id.myList);
      list.setAdapter(adapter);
      list.setOnItemClickListener(new OnItemClickListener() {

         @Override
         public void onItemClick(AdapterView<?> adapter, View v, int position,
               long id) {
            AndroidVersion selectedItem = (AndroidVersion) adapter.getItemAtPosition(position);
            Log.v("CustomAdapter", "Elemento seleccionado: " + selectedItem.getVersionName());
         }
      });
   }
```

Creación de interfaces sencillas en Android

```
private void initList( ArrayList<AndroidVersion> androidList ) {

    AndroidVersion version = new AndroidVersion();
    version.setVersionName("Cupcake");
    version.setVersionNumber("1.5");
    androidList.add(version);

    AndroidVersion versionDonut = new AndroidVersion();
    versionDonut.setVersionName("Donut");
    versionDonut.setVersionNumber("1.6");
    androidList.add(versionDonut);

    AndroidVersion versionEclair = new AndroidVersion();
    versionEclair.setVersionName("Eclair");
    versionEclair.setVersionNumber("2.0.x");
    androidList.add(versionEclair);

    AndroidVersion versionFroyo = new AndroidVersion();
    versionFroyo.setVersionName("Froyo");
    versionFroyo.setVersionNumber("2.2.x");
    androidList.add(versionFroyo);

                .............
  }
}
```

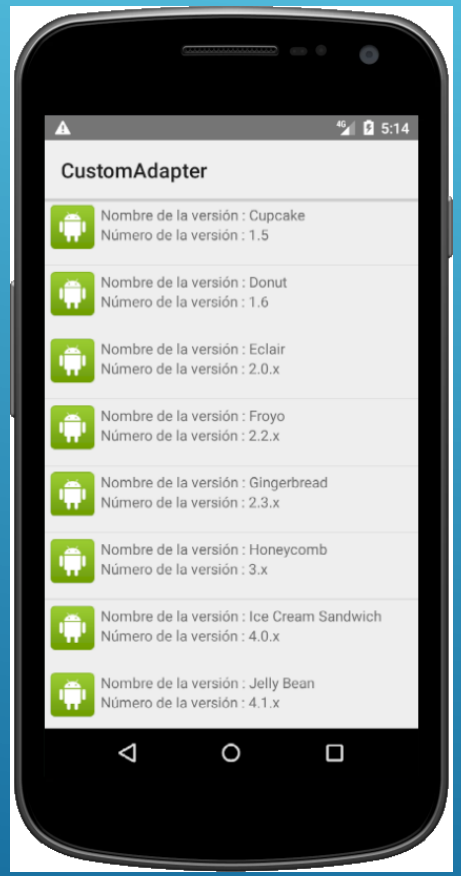
Creación de interfaces sencillas en Android

Esta actividad permite:

- Inicializar la lista de datos (método *initList*).
- Inicializar el ***adapter*** pasándole los parámetros necesarios para su correcto funcionamiento (contexto, vista y datos).
- Recuperar la lista y asociarla con el ***adapter***.

Ejecutar ahora el ejemplo y comprobar el resultado obtenido:

Creación de interfaces sencillas en Android



Ver el programa Ch5_CustomAdapter

Gustavo Márquez Flores Facultad de Ciencias

Creación de interfaces sencillas en Android

Gestión del clic en una lista

Para gestionar el click sobre los distintos elementos de una lista, utilizar el método ***setOnItemClickListener*** sobre la instancia de la lista correspondiente.

```
list.setOnItemClickListener(new OnItemClickListener() {

   @Override
   public void onItemClick(AdapterView<?> adapter, View v, int position, long id) {

      AndroidVersion selectedItem = (AndroidVersion) adapter.getItemAtPosition(position);
      Log.v("CustomAdapter", "Elemento seleccionado: " + selectedItem.getVersionName());
   }
});
```

Creación de interfaces sencillas en Android

El método **onItemClick** recibe cuatro argumentos:

- El *adapter* sobre el que se ha producido el click.
- La vista sobre la que se ha producido el click.
- La posición del click en el ***adapter***.
- El identificador de la posición del click en la vista.

A continuación, se puede recuperar la instancia de la clase ***AndroidVersion*** que se corresponde con la zona seleccionada por el usuario para mostrarla en un Log:

```
Log.v("CustomAdapter", "Elemento seleccionado: " + selectedItem.getVersionName());
```

Creación de interfaces sencillas en Android

Actualizar una lista.

Para actualizar el contenido de la lista durante la ejecución del programa, basta con agregar los nuevos datos en el ***adapter*** (una nueva lista, por ejemplo), y a continuación utilizar el método ***adapter.notifyDataSetChanged()*** para indicar al ***adapter*** y a la lista, que los datos que se utilizan se han modificado y se deben actualizar.

Creación de interfaces sencillas en Android

Gustavo Márquez Flores Facultad de Ciencias

Creación de interfaces sencillas en Android

Gustavo Márquez Flores Facultad de Ciencias