

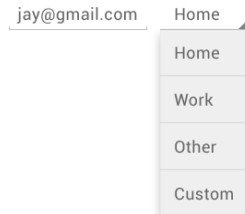


La Clase Spinner

CONCEPTOS

Un Spinner es un componente gráfico, del tipo lista desplegable, que muestra un elemento hijo a la vez y permite al usuario elegir uno de ellos. Los elementos del Spinner provienen del adaptador asociado con este componente.

En el estado predeterminado, se muestra su valor seleccionado en ese momento. Al digitar el Spinner se muestra un menú desplegable con todos los demás valores disponibles, de los cuales el usuario puede seleccionar uno nuevo.



Se puede agregar un control a la plantilla con el objeto Spinner. Para utilizarlo, se incluye en la plantilla XML una etiqueta `<Spinner>`. Por ejemplo:

```
<Spinner
    android:id="@+id/planets_spinner"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
```

Para rellenar el Spinner con una lista de opciones, es necesario especificar un `SpinnerAdapter` en la actividad o fragmento de código fuente.

Los elementos del Spinner.

Los elementos que se agregan al Spinner pueden venir de cualquier fuente, pero deben proporcionarse a través de un `SpinnerAdapter`, por ejemplo un `ArrayAdapter` si las opciones están disponibles en un arreglo o en un `CursorAdapter` si las opciones están disponibles a partir de una consulta a una base de datos.

Si las opciones disponibles están predeterminadas, se pueden proporcionar con un arreglo de cadenas definido en el archivo `strings.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="planets_array">
        <item>Mercury</item>
        <item>Venus</item>
        <item>Earth</item>
        <item>Mars</item>
        <item>Jupiter</item>
        <item>Saturn</item>
        <item>Uranus</item>
        <item>Neptune</item>
    </string-array>
</resources>
```

Con un arreglo como éste, se puede utilizar el siguiente código en la Activity o Fragment para abastecer al Spinner con el arreglo mediante una instancia de `ArrayAdapter`:

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
// Create an ArrayAdapter using the string array and a default spinner layout
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
    R.array.planets_array, android.R.layout.simple_spinner_item);
```



```
// Specify the layout to use when the list of choices appears
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
// Apply the adapter to the spinner
spinner.setAdapter(adapter);
```

El método `createFromResource()` permite crear un `ArrayAdapter` con el arreglo de cadenas. El tercer argumento del método es una plantilla que define la forma en que se muestra la opción seleccionada en el `Spinner`. La plantilla `simple_spinner_item` es proporcionado por la plataforma y es la predeterminada que se debe utilizar, a menos que se desee definir una propia plantilla para la apariencia del `Spinner`.

A continuación, se debe llamar a `setDropDownViewResource(int)` para especificar la plantilla que el adaptador debe utilizar para mostrar la lista de opciones (`simple_spinner_dropdown_item` es otra plantilla estándar definida por la plataforma). Llamar a `setAdapter()` para aplicar el adaptador al `Spinner`.

Respuesta a las selecciones del usuario.

Cuando el usuario selecciona un elemento de la lista, el objeto `Spinner` recibe un evento `on-item-selected`.

Para definir el control del evento de selección, se implanta la interfaz `AdapterView.OnItemSelectedListener` y el correspondiente método `onItemSelected()`. Enseguida se muestra una implantación de la interfaz en una actividad:

```
public class SpinnerActivity extends Activity implements OnItemSelectedListener {
    :
    public void onItemSelected(AdapterView<?> parent, View view, int pos, long id) {
        // Se extrae el elemento seleccionado con parent.getItemAtPosition(pos)
    }
    public void onNothingSelected(AdapterView<?> parent) {
        // Otra llamada a la interface
    }
}
```

El `AdapterView.OnItemSelectedListener` necesita los métodos `onItemSelected()` y `onNothingSelected()`. Por tanto, se requiere especificar la implantación de la interfaz invocando a `setOnItemSelectedListener()`:

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
spinner.setOnItemSelectedListener(this);
```

Si implanta la interfaz `AdapterView.OnItemSelectedListener` en la `Activity` o `Fragment` (como en el ejemplo anterior), se puede pasar esto como la instancia de la interfaz.

DESARROLLO

EJEMPLO 1.

En este ejemplo, las opciones del `Spinner` se toman de un arreglo especificado en el archivo `strings.xml`.

Paso 1. Crear un proyecto `Spinner1`. En la carpeta `java/com.example.mipaquete`, abrir y modificar el archivo Java predeterminado `MainActivity.java`, con el siguiente código:

```
import android.os.Bundle;
import android.app.Activity;
import android.view.*;
import android.widget.*;
import android.widget.AdapterView.*;
public class MainActivity extends Activity{
    Spinner s;
    public void onCreate(Bundle b) {
        super.onCreate(b);
```



```

setContentView(R.layout.activity_main);
s = (Spinner) findViewById(R.id.xsp);
s.setOnItemSelectedListener(new OnItemSelectedListener() {
    public void onItemSelected(AdapterView<?> av, View v, int i, long l) {
        Toast.makeText(MainActivity.this, s.getSelectedItem().toString(),
            Toast.LENGTH_LONG).show();
    }
    public void onNothingSelected(AdapterView<?> arg0) { }
});
}
}

```

Paso 2. En la carpeta res/layout, abrir y modificar el archivo XML predeterminado activity_main.xml, con 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="match_parent"
    android:orientation="vertical" >
    <Spinner
        android:id="@+id/xsp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:prompt="@string/xsops"
        android:entries="@array/xops" />
</LinearLayout>

```

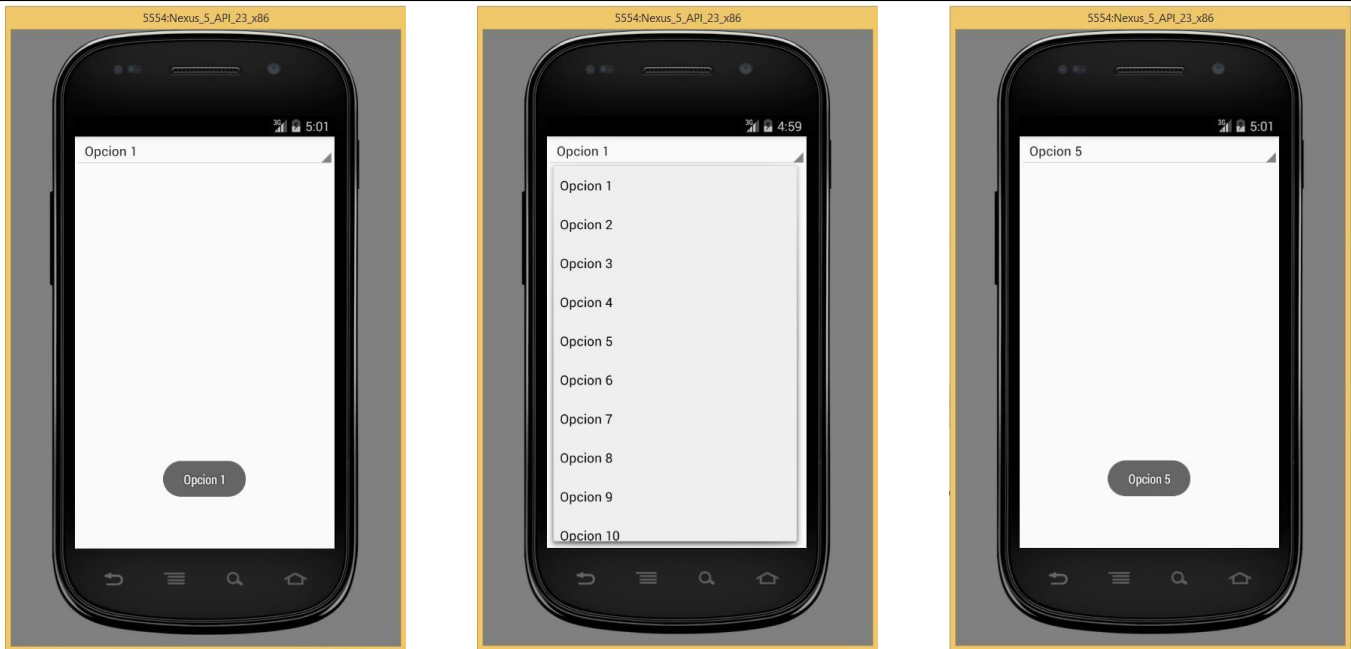
Paso 3. En la carpeta res/values, abrir y modificar el archivo XML predeterminado strings.xml, con el siguiente código:

```

<?xml version="1.0" encoding="utf-8" ?>
<resources>
    <string name="app_name">Ejemplo de Spinner </string>
    <string name="xsops">Opciones</string>
    <string name="action_settings">Ajustes</string>
    <string-array name="xops">
        <item>Opcion 1</item>
        <item>Opcion 2</item>
        <item>Opcion 3</item>
        <item>Opcion 4</item>
        <item>Opcion 5</item>
        <item>Opcion 6</item>
        <item>Opcion 7</item>
        <item>Opcion 8</item>
        <item>Opcion 9</item>
        <item>Opcion 10</item>
    </string-array>
</resources>

```

Paso 4. Por último, ejecutar la aplicación. Al inicio, se muestra solamente la primera opción del Spinner. Digitar sobre el primer elemento para desplegar toda la lista. Al seleccionar una opción, se muestra un Toast conteniendo el texto del elemento seleccionado. Al final, se pliega el Spinner y muestra de nuevo la primera opción.



EJEMPLO 2.

En este ejemplo, las opciones del Spinner se toman desde una base de datos SQLite.

Paso 1. Crear un proyecto Spinner2. En la carpeta java/com.example.mipaquete, abrir y modificar el archivo Java predeterminado MainActivity.java, con el siguiente código.

```

import java.util.List;
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.inputmethod.InputMethodManager;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.*;

public class MainActivity extends Activity implements OnItemSelectedListener {
    Spinner      jsp1;
    Button        jbn1;
    EditText      jet1;
    @Override
    public void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_main);
        jsp1 = (Spinner) findViewById(R.id.xsp1);
        jbn1 = (Button) findViewById(R.id.xbn1);
        jet1 = (EditText) findViewById(R.id.xet1);
        jsp1.setOnItemSelectedListener(this);
        cargaSpinner();
        jbn1.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                String s = jet1.getText().toString();
                if (s.trim().length() > 0) {
                    DatabaseHandler db = new DatabaseHandler(getApplicationContext());
                    db.insertLabel(s);
                    jet1.setText("");
                    InputMethodManager imm = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);

```



```

        imm.hideSoftInputFromWindow(jet1.getWindowToken(), 0);
        cargaSpinner();
    } else {
        Toast.makeText(getApplicationContext(), "Escribir elemento",
Toast.LENGTH_SHORT).show();
    }
}
});
}
private void cargaSpinner() {
    DatabaseHandler db = new DatabaseHandler(getApplicationContext());
    List<String> l = db.getAllLabels();
    ArrayAdapter<String> aa = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, l);
    aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    jspl.setAdapter(aa);
}
@Override
public void onItemSelected(AdapterView<?> av, View v, int i, long l) {
    String s = av.getItemAtPosition(i).toString();
    Toast.makeText(av.getContext(), "Selección: " + s, Toast.LENGTH_LONG).show();
}
@Override
public void onNothingSelected(AdapterView<?> arg0) {
    // TODO Auto-generated method stub
}
}
}

```

Paso 2. En la carpeta java/com.example.mipaquete, crear el archivo DatabaseHandler.java y agregar el siguiente código:

```

import java.util.*;
import android.content.*;
import android.database.Cursor;
import android.database.sqlite.*;
public class DatabaseHandler extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "EjemploDeSpinner";
    private static final String TABLE_LABELS = "nombres";
    private static final String KEY_ID = "id";
    private static final String KEY_NAME = "nombre";
    public DatabaseHandler(Context c) {
        super(c, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_CATEGORIES_TABLE = "CREATE TABLE " + TABLE_LABELS + "(" + KEY_ID +
            " INTEGER PRIMARY KEY," + KEY_NAME + " TEXT)";
        db.execSQL(CREATE_CATEGORIES_TABLE);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_LABELS);
        onCreate(db);
    }
    public void insertLabel(String s){
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put(KEY_NAME, s);
        db.insert(TABLE_LABELS, null, cv);
        db.close();
    }
}

```



```

public List<String> getAllLabels() {
    List<String> l = new ArrayList<String>();
    String q = "SELECT * FROM " + TABLE_LABELS;
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor c = db.rawQuery(q, null);
    if (c.moveToFirst()) {
        do {
            l.add(c.getString(1));
        } while (c.moveToNext());
    }
    c.close();
    db.close();
    return l;
}
}

```

Paso 3. En la carpeta `res/layout`, abrir el archivo predeterminado `activity_main.xml` para modificarlo con el siguiente código:

```

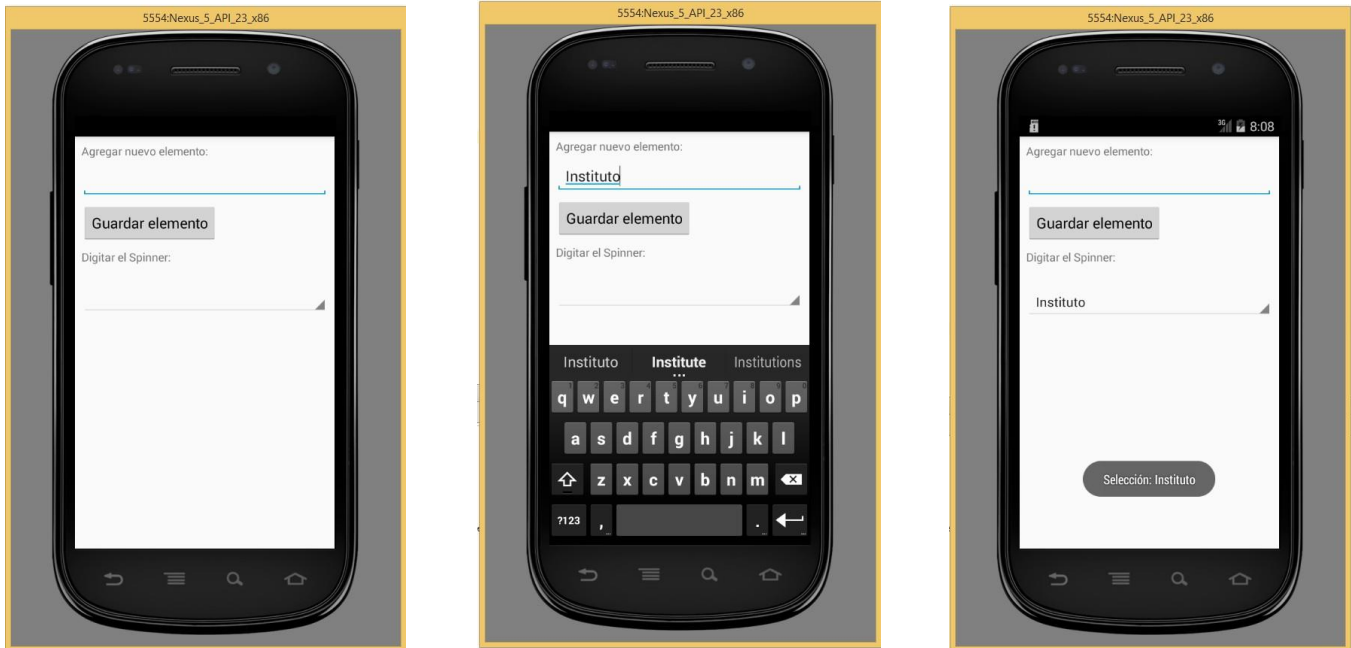
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Agregar nuevo elemento:"
        android:padding="8dip" />
    <EditText android:id="@+id/xet1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dip"
        android:layout_marginRight="8dip"/>
    <Button android:id="@+id/xbn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Guardar elemento"
        android:layout_marginLeft="8dip"
        android:layout_marginTop="8dip"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Digitar el Spinner:"
        android:padding="8dip" />
    <Spinner
        android:id="@+id/xsp1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:prompt="@string/spinner_title"
        android:layout_marginTop="20dip"
        android:layout_marginLeft="8dip"
        android:layout_marginRight="8dip" />
</LinearLayout>

```

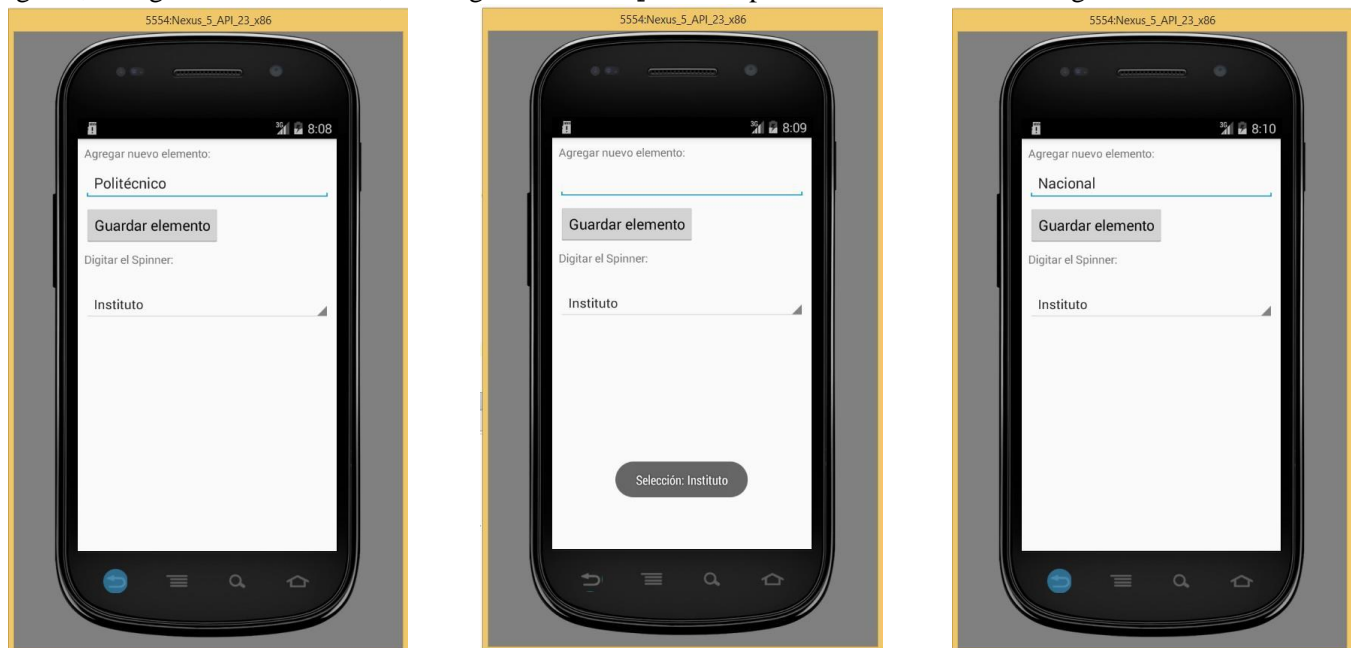
Paso 4. Por último, ejecutar la aplicación. Primero, ingresar un dato en el campo de texto. Si no se ingresa un dato, se muestra un `Toast` emergente que lo solicita. Al digitar el botón, se ingresa el dato en la base de datos. Enseguida, se puede ingresar otro dato o se puede mostrar la lista de elementos ingresados al `Spinner`, digitando sobre éste, como se muestra en las siguientes figuras.



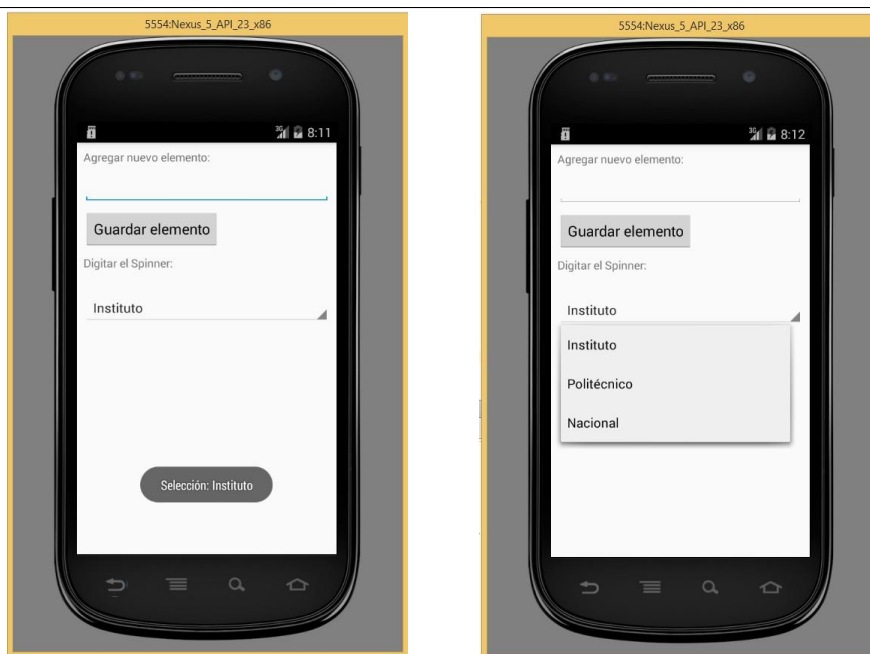
Al inicio, ingresar un primer elemento en el campo de texto y digitar el botón para guardarlo en la base de datos. Con un Toast se indica el elemento guardado:



Enseguida, se ingresa otro elemento, o se digita sobre el Spinner para mostrar los elementos guardados:



Después de ingresar más datos, en cualquier momento se puede mostrar el contenido del Spinner:



Nota. Ejecutar cada uno de los ejercicios y capturar la imagen de cada una de las aplicaciones. Guardar las imágenes obtenidas en un documento y asignarle la sintaxis `AlumnoTarea5Grupo.pdf` al archivo. Se pueden agregar comentarios que retroalimenten las aplicaciones y algún código extra corto que lo justifique. Enviar el documento al sitio indicado por el profesor.