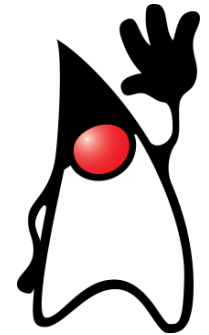




# Android Con Java



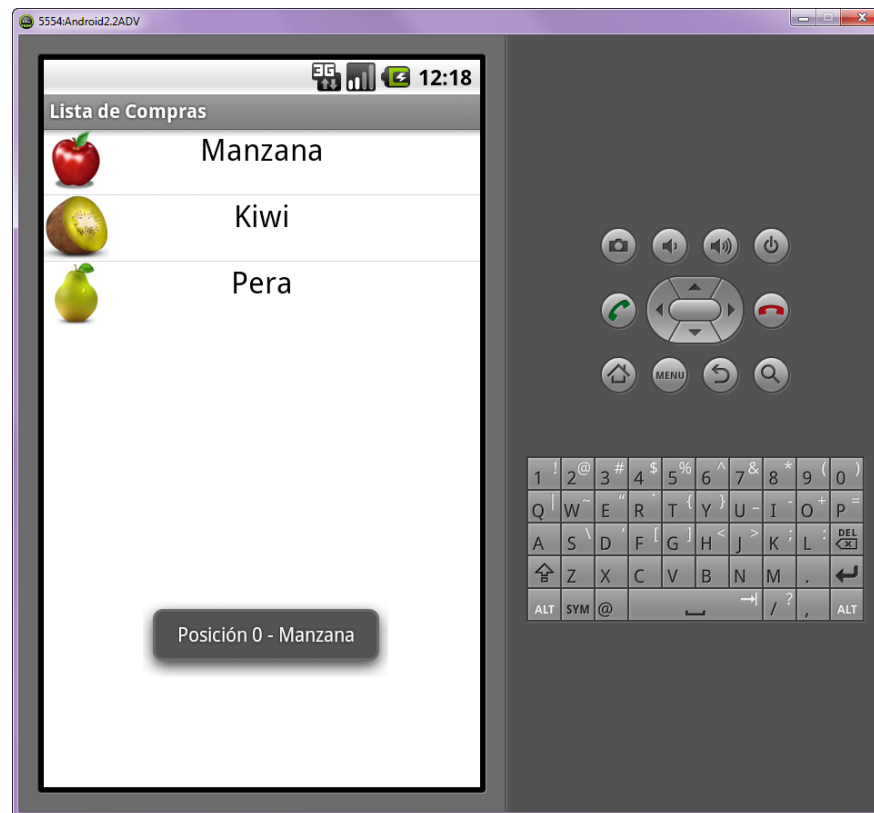
## Ejercicio 5



Manejo de  
Adaptadores Personalizados  
en Android

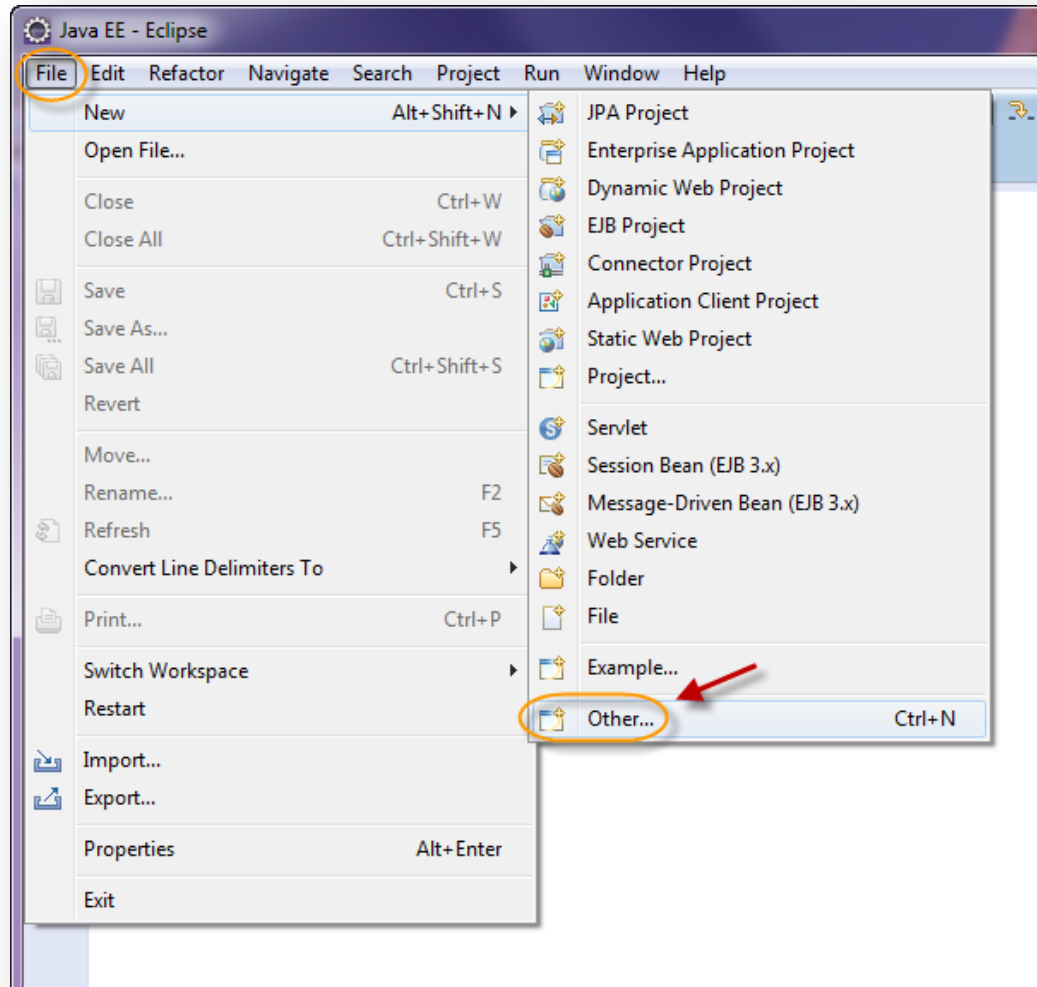
## Objetivo del Ejercicio

- El objetivo del ejercicio es entender el manejo de Layout y los Adaptadores Personalizados en Android.
- Al finalizar deberemos observar la siguiente aplicación Android:



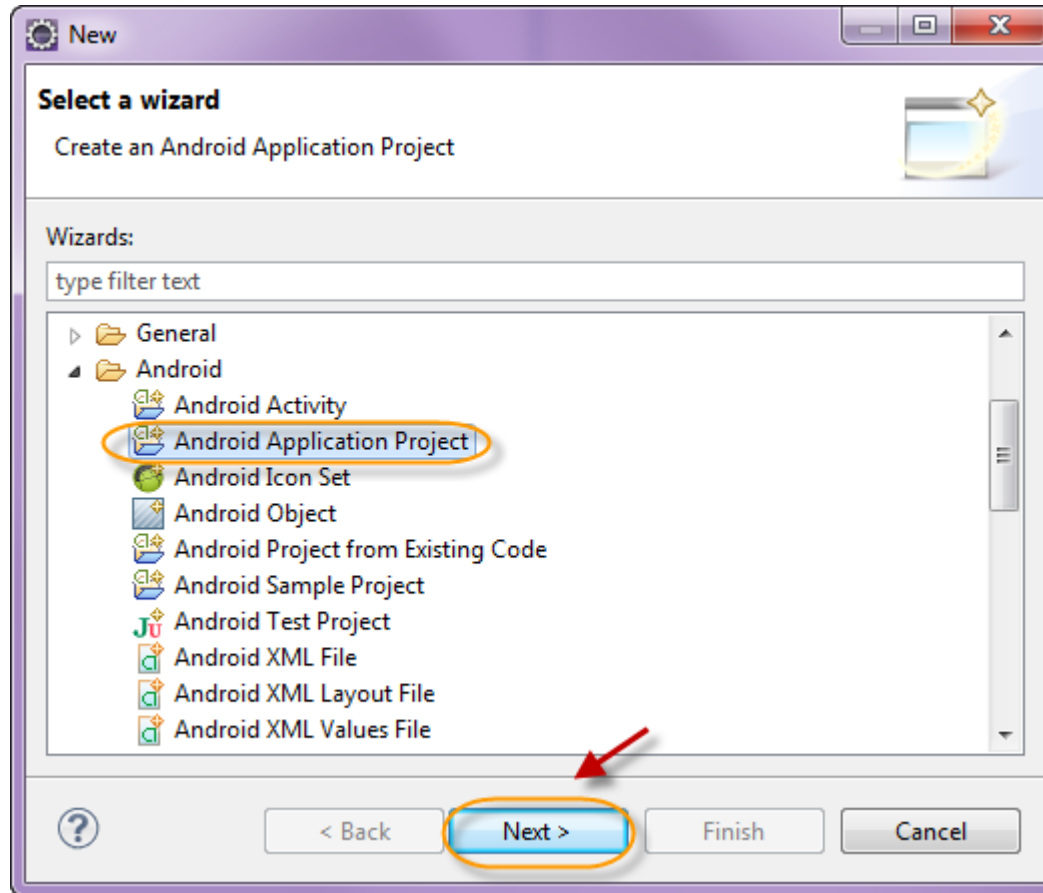
# Paso 1. Creación proyecto ListaCompras

Creamos el proyecto ListaCompras:



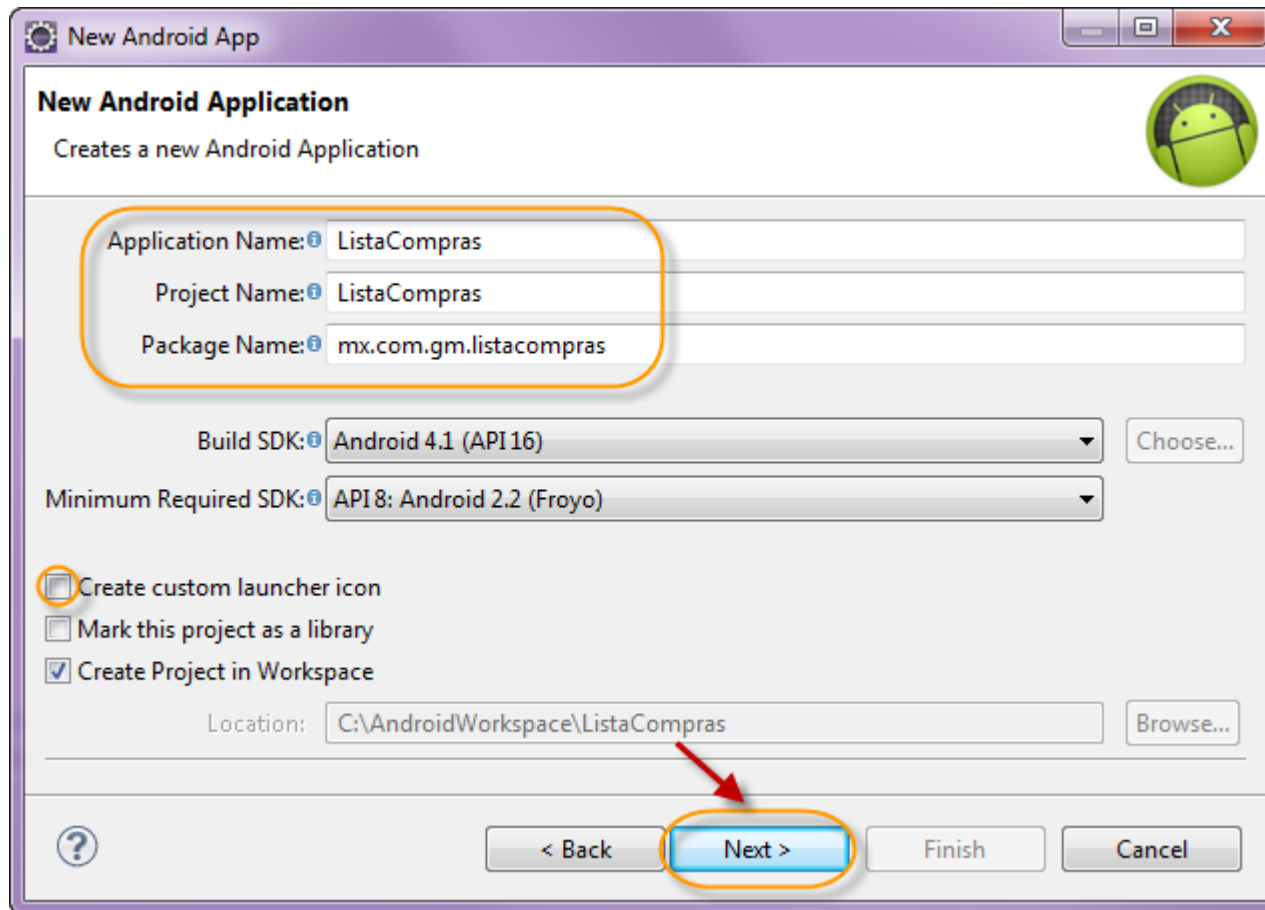
# Paso 1. Creación proyecto ListaCompras (cont)

Creamos el proyecto ListaCompras con Android:



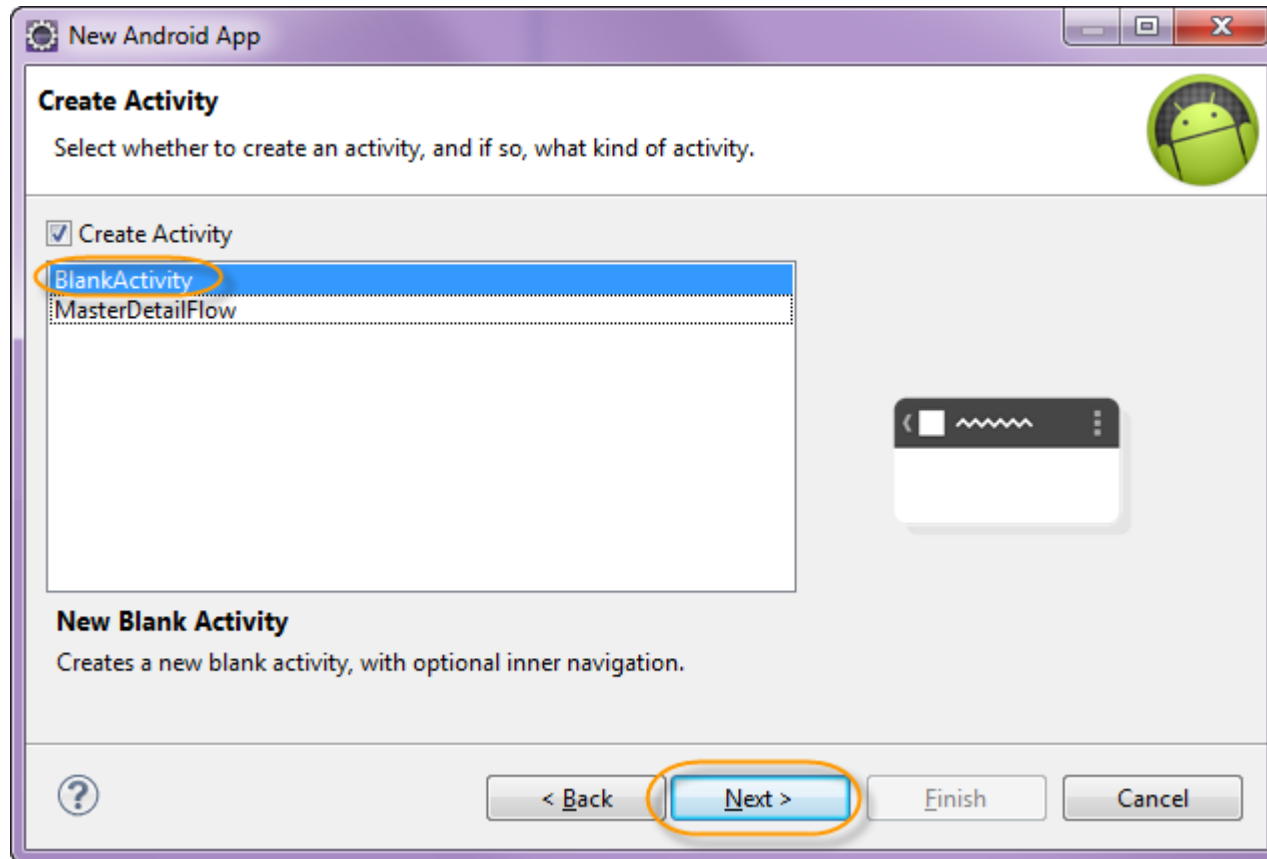
# Paso 1. Creación proyecto ListaCompras (cont)

Creamos el proyecto ListaCompras con Android:



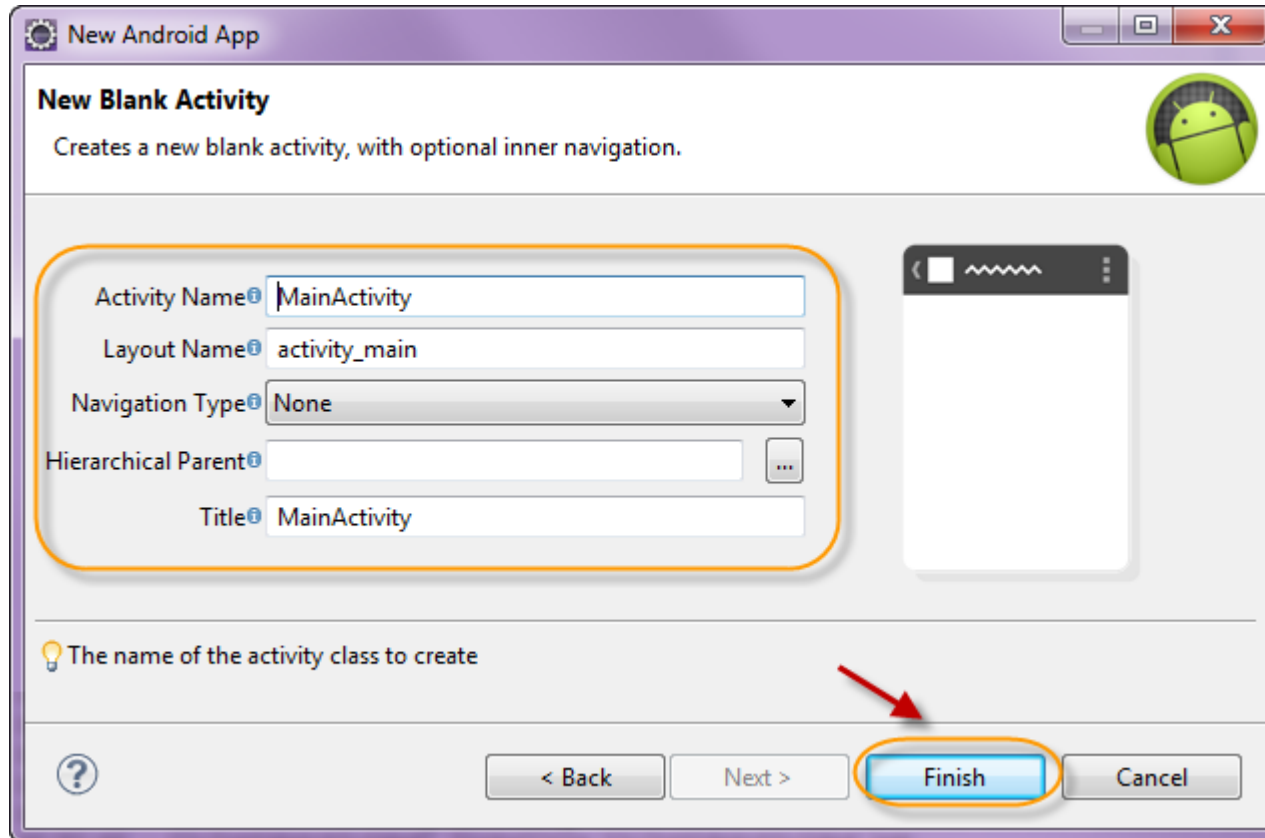
# Paso 1. Creación proyecto ListaCompras (cont)

Creamos el proyecto ListaCompras con Android:



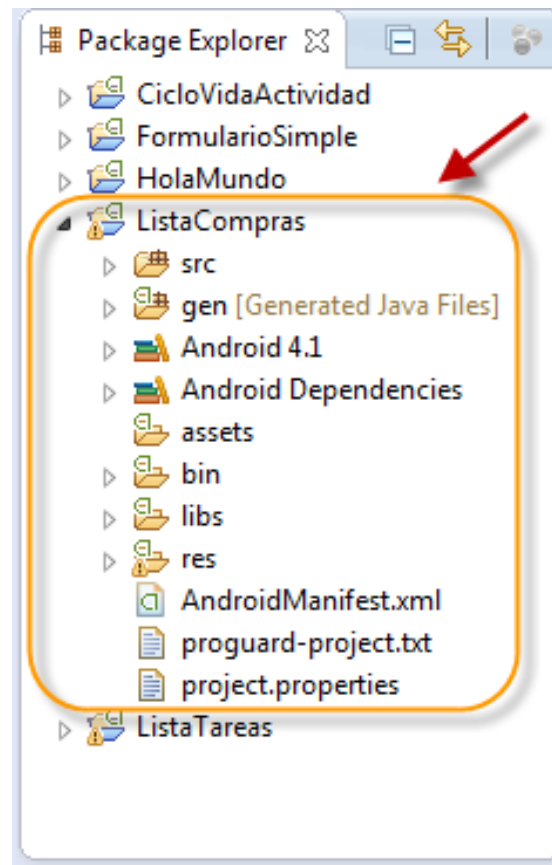
# Paso 1. Creación proyecto ListaCompras (cont)

Creamos el proyecto ListaCompras con Android:



## Paso 1. Creación proyecto ListaCompras (cont)

El proyecto debe visualizarse como sigue:







## Paso 2. Modificamos el archivo strings.xml

Modificamos algunas etiquetas, en el archivo de `res/values/strings.xml`:

```
<resources>
```

```
    <string name="app_name">ListaCompras</string>  
    <string name="menu_settings">Configuración</string>  
    <string name="title_activity_main">Lista de Compras</string>  
    <string name="agregarElemento">Agregar Elemento</string>  
    <string name="agregarDescElemento">Descripción Elemento</string>
```

```
</resources>
```



## Paso 3. Modificamos el layout activity\_main.xml

Modificamos el layout activity\_main.xml:

```
<?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" >

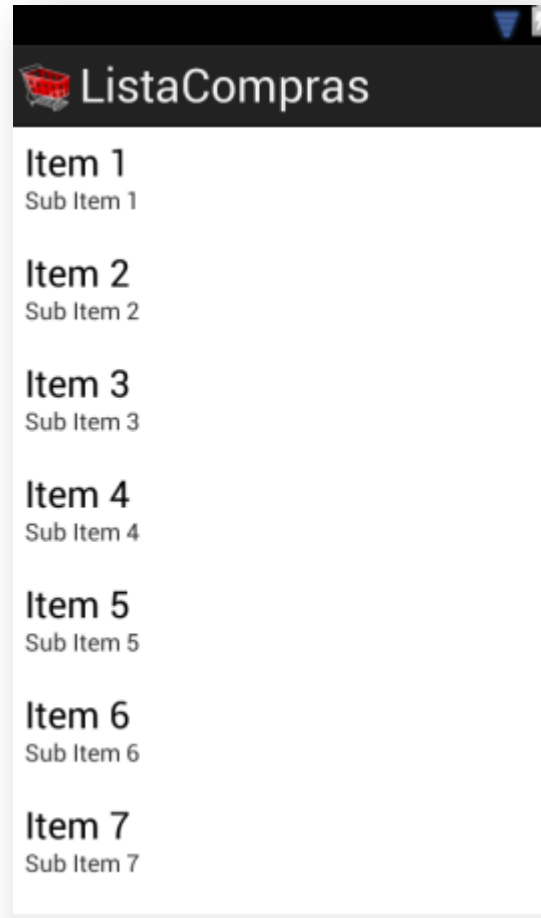
    <ListView
        android:id="@+id/listViewElementos"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```



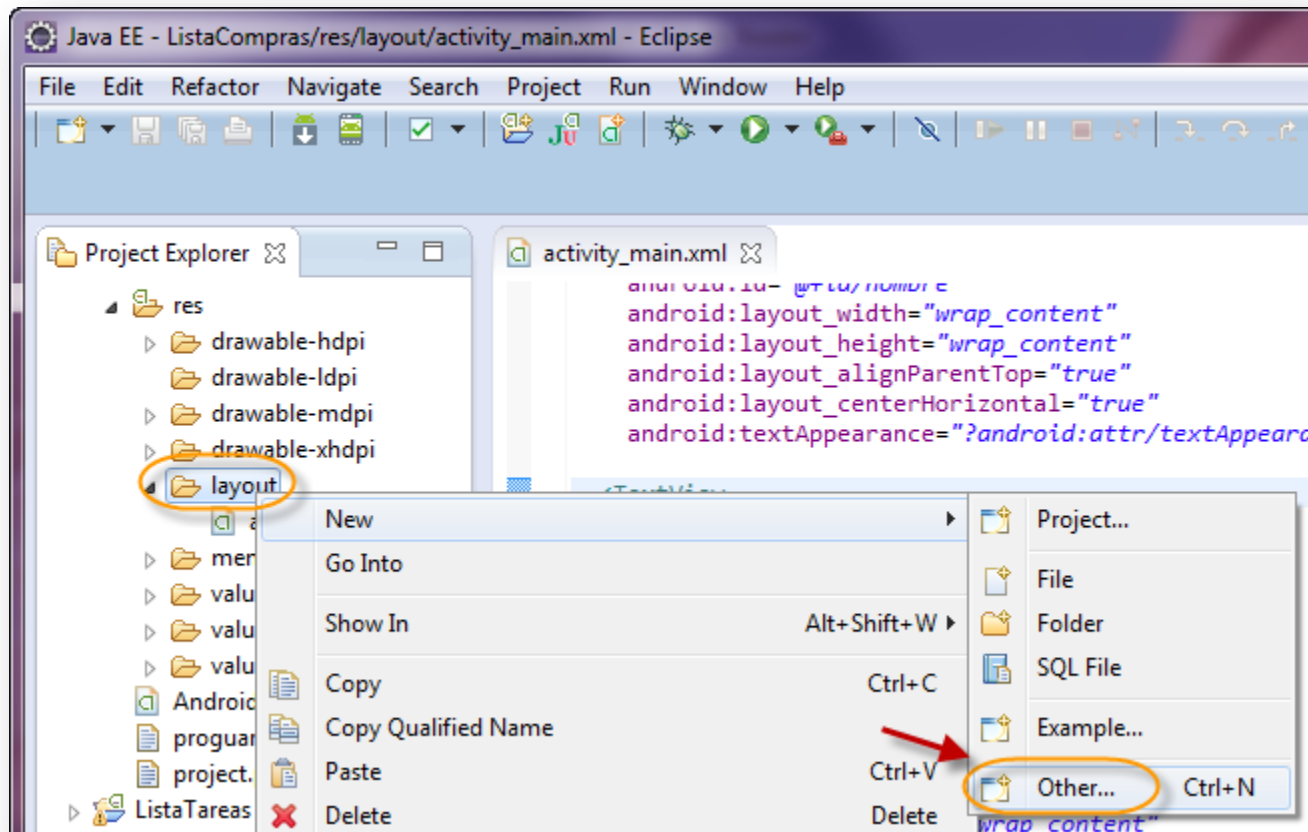
## Paso 3. Modificamos el layout activity\_main.xml (cont)

El resultado debe ser similar al siguiente. El ícono lo modificaremos posteriormente:



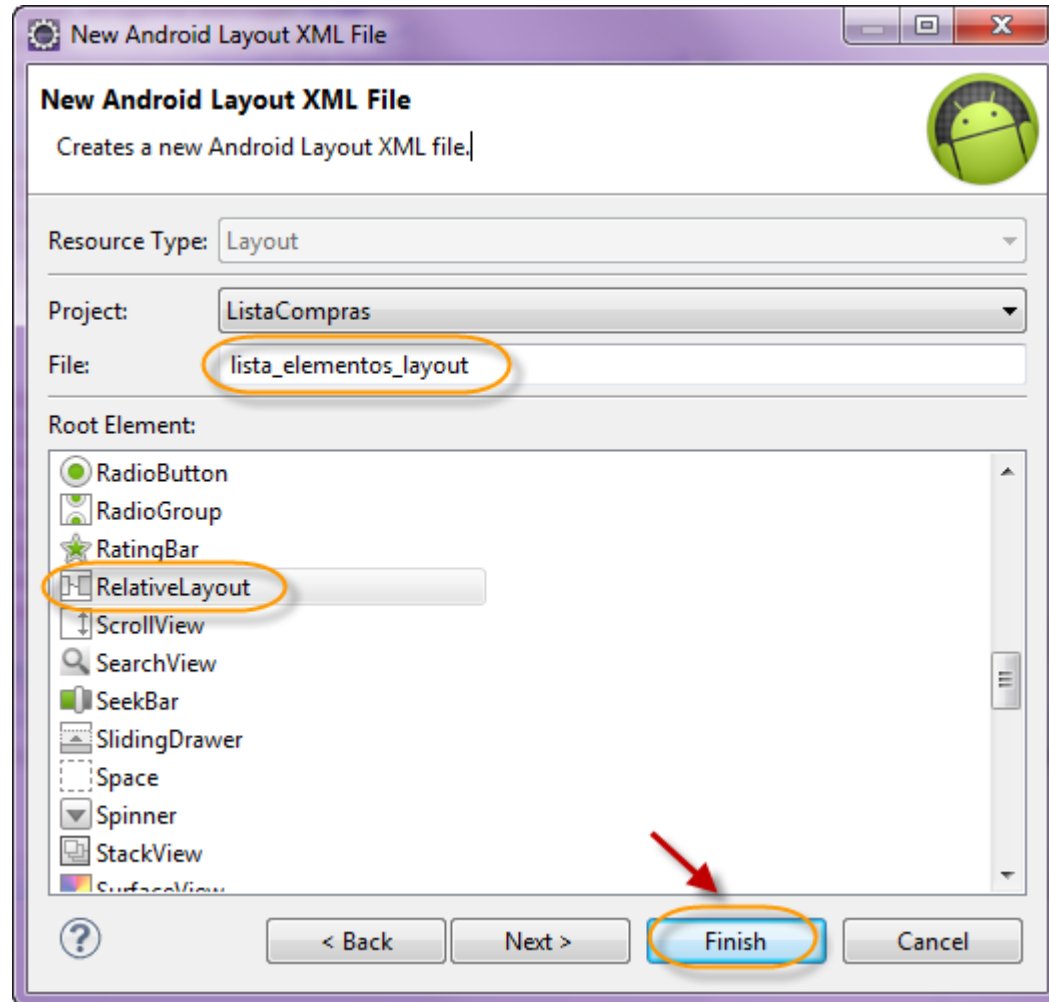
## Paso 4. Agregamos el layout lista\_elementos\_layout.xml

Creamos el layout lista\_elementos\_layout.xml:



## Paso 4. Agregamos el layout lista\_elementos\_layout.xml (cont)

Creamos el layout lista\_elementos\_layout.xml:





## Paso 4. Agregamos el layout lista\_elementos\_layout.xml (cont)

Creamos el layout lista\_elementos\_layout.xml:

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

    <ImageView
        android:id="@+id/imagenElemento"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:contentDescription="@string/elementoDescripcion"
        android:layout_centerInParent="true"/>

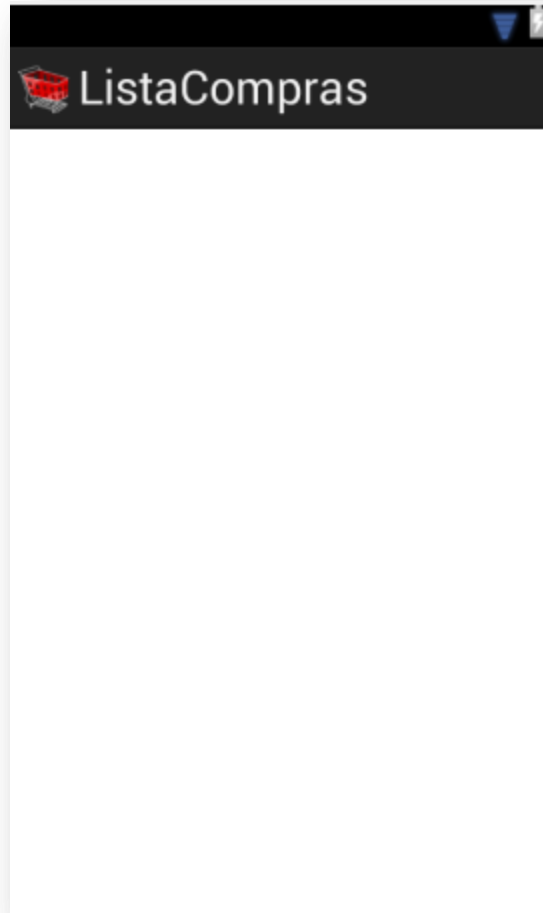
    <TextView
        android:id="@+id/nombre"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</RelativeLayout>
```



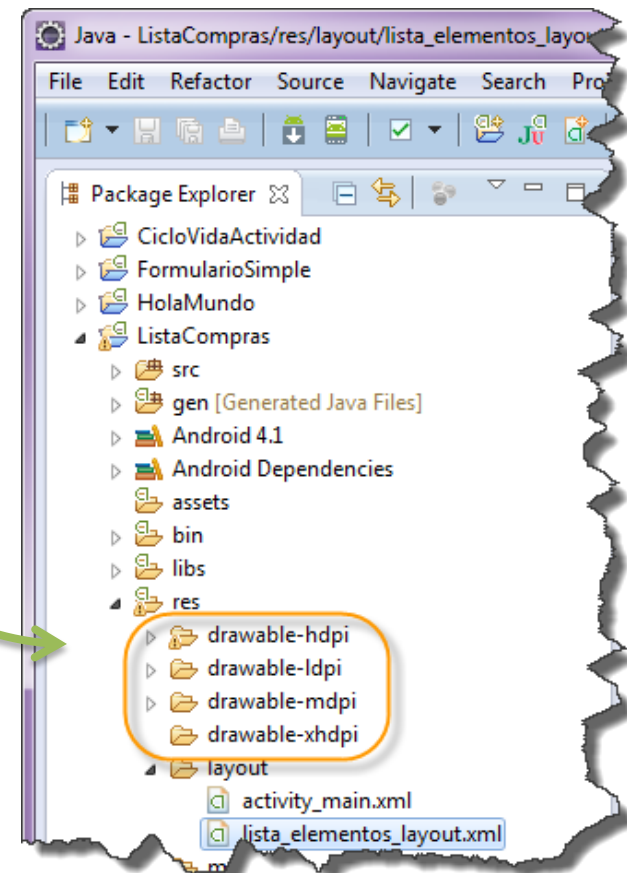
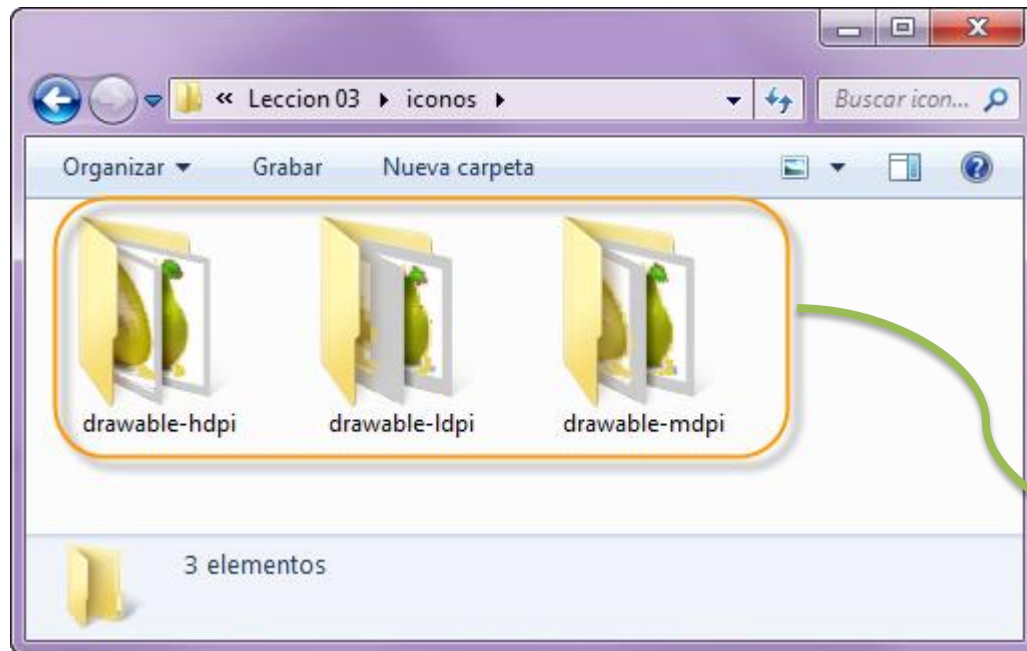
## Paso 4. Agregamos el layout lista\_elementos\_layout.xml (cont)

El resultado debe ser similar al siguiente. El ícono lo modificaremos posteriormente:



## Paso 6. Manejo de Imágenes

En su carpeta de ejercicios, les hemos agregado las imágenes e íconos para este ejercicio. Para agregarlo al proyecto, basta con arrastrar cada grupo de imágenes a su carpeta cada respectiva según la resolución indicada. La carpeta de drawable-xhdpi no agregaremos imágenes.





## Paso 7. Modificar el archivo de Manifiesto

Cambiamos la ruta del icono en el archivo de AndroidManifest.xml. No agregamos la extensión, sólo el nombre de la imagen:



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="mx.com.gm.listacompras"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />

    <application
        android:icon="@drawable/icon"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```



## Paso 8. Agregamos la clase Elemento.java

Agregamos el siguiente código a la clase Elemento.java:

```
package mx.com.gm.listacompras;
```

```
public class Elemento {
```

```
    private int id;
```

```
    private String nombre;
```

```
    private String imagen;
```

```
    public Elemento(int id, String nombre, String imagen) {
```

```
        this.id = id;
```

```
        this.nombre = nombre;
```

```
        this.imagen = imagen;
```

```
    }
```

```
    public int getId() {
```

```
        return id;
```

```
    }
```

```
    public void setId(int id) {
```

```
        this.id = id;
```

```
    }
```

```
    public String getNombre() {
```

```
        return nombre;
```

```
    }
```

```
    public void setNombre(String nombre) {
```

```
        this.nombre = nombre;
```

```
    }
```

```
    public String getImagen() {
```

```
        return imagen;
```

```
    }
```

```
    public void setImagen(String imagen) {
```

```
        this.imagen = imagen;
```

```
    }
```

```
    Override
```

```
    public String toString() {
```

```
        return "Elemento [id=" + id + ", nombre=" + nombre + ", imagen=" +  
            imagen + "];"
```

```
    }
```

```
}
```



## Paso 9. Agregamos la clase ElementoAdaptador.java

Agregamos el siguiente código a la clase ElementoAdaptador.java:

```
package mx.com.gm.listacompras;
```

```
import java.util.ArrayList;
import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;
```

```
public class ElementoAdaptador extends BaseAdapter {
```

```
    private Activity actividad;
    private ArrayList<Elemento> elementos;
```

```
    public ElementoAdaptador(Activity actividad, ArrayList<Elemento>
elementos) {
        this.actividad = actividad;
        this.elementos = elementos;
    }
```

```
    public int getCount() {
        return elementos.size();
    }
```

```
    public Object getItem(int index) {
        return elementos.get(index);
    }
```

```
    public long getItemId(int index) {
        return elementos.get(index).getId();
    }
```

```
    public View getView(int posicion, View vistaACambiar, ViewGroup padre) {
        View vista = vistaACambiar;
        //Verificamos si la vista no se ha desplegado
        if (vista == null) {
            LayoutInflater inflater = (LayoutInflater)
actividad.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            vista = inflater.inflate(R.layout.lista_elementos_layout, null);
        }

        Elemento elemento = elementos.get(posicion);

        //Modificamos el contenido del nombre del elemento
        TextView nombre = (TextView) vista.findViewById(R.id.nombre);
        nombre.setText(elemento.getNombre());

        //Modificamos el contenido de la imagen del elemento
        ImageView imagen = (ImageView) vista.findViewById(R.id.imagenElemento);
        //Recuperamos el identificador de la imagen
        int recursolImagen = actividad.getResources().getIdentifier("drawable/" +
elemento.getImagen(), null, actividad.getPackageName());
        //Colocamos la imagen recuperada segun clase R
        imagen.setImageDrawable(actividad.getResources().getDrawable(recursolImagen));

        return vista;
    }
}
```



# Paso 10. Modificamos la clase MainActivity.java

Agregamos el siguiente código a la clase MainActivity.java:

```
package mx.com.gm.listacompras;

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

public class MainActivity extends Activity {

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

        //1. Recuperamos el ListView
        ListView viewLista = (ListView)
        findViewById(R.id.listViewElementos);

        //2. Obtenemos los elementos de la lista
        final ArrayList<Elemento> elementos = getElementos();

        //3. Iniciamos el adaptador, el cual llamará
        ElementoAdaptador adaptador = new ElementoAdaptador(this,
        elementos);
```

```
//4. Asociamos el adaptador a la vista
viewLista.setAdapter(adaptador);

//5. Registramos el evento OnItemClickListener
viewLista.setOnItemClickListener(new OnItemClickListener() {

    public void onItemClick(AdapterView<?> padre, View vista, int posicion,
        long id) {

        //Recuperamos el layout
        Elemento elementoSel = elementos.get(posicion);
        String mensaje = "Posición " + posicion + " - " +
        elementoSel.getNombre();

        //Enviamos una notificación
        Toast.makeText(MainActivity.this, mensaje,
        Toast.LENGTH_SHORT).show();
    }
});

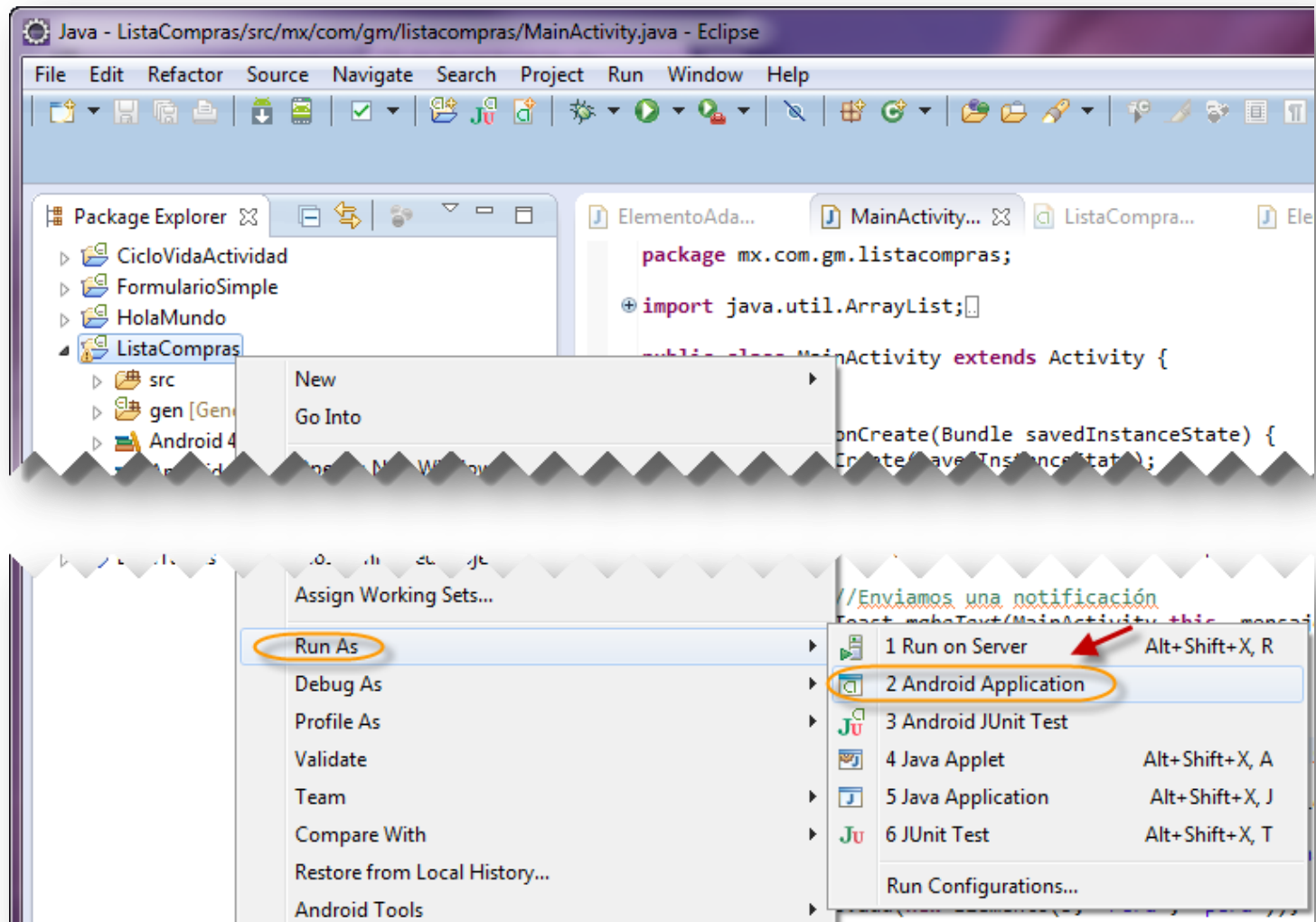
//Estos elementos se pueden obtener de otra fuente, como una BD
private ArrayList<Elemento> getElementos() {
    ArrayList<Elemento> elementos = new ArrayList<Elemento>();

    elementos.add(new Elemento(1, "Manzana", "manzana"));
    elementos.add(new Elemento(2, "Kiwi", "kiwi"));
    elementos.add(new Elemento(3, "Pera", "pera"));

    return elementos;
}
```

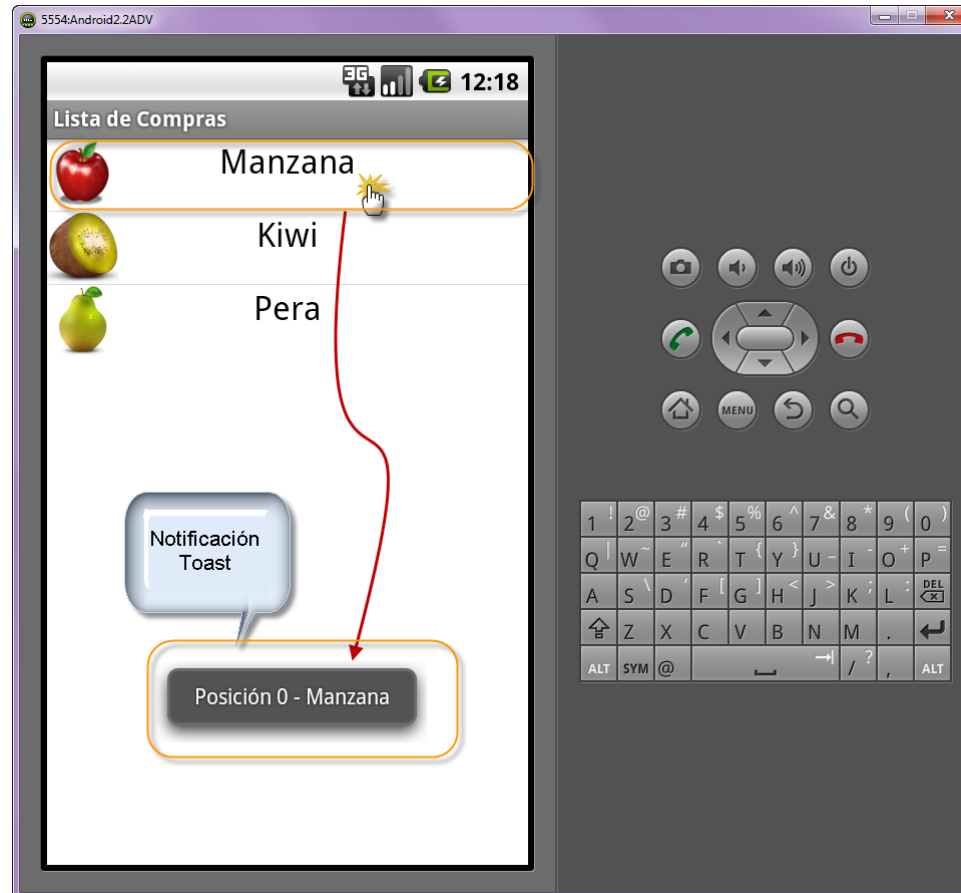
## Paso 11. Ejecutamos el proyecto Lista de Compras

Ejecutamos la aplicación como sigue:



## Paso 11. Ejecutamos el proyecto Lista de Compras (cont)

Podemos observar que la aplicación Lista los elementos agregados, y además al dar clic sobre cada elemento, nos manda el identificador de cada uno.





## Conclusión

Con este ejercicio hemos revisado cómo trabajar con Layouts y Adaptadores personalizados en Android.

Pudimos observar el uso de una plantilla personalizada, así como un adaptador personalizado para adecuar el contenido de String, a un objeto de tipo Elemento, y así mostrar las características del mismo.

Además, vimos cómo procesar el evento `setOnClickListener` del objeto `listView`, y así desplegar una notificación cada vez que seleccionamos un elemento.

Este ejercicio se puede hacer más completo con más actividades y el manejo de Intents para ir cambiando entre las actividades, lo cual estudiaremos en las siguientes lecciones.





## Recursos

Como complemento para la creación de íconos, pueden utilizar el siguiente link, y así generar las imágenes en las distintas resoluciones que trabajamos:

En esta página se pueden generar íconos para las diferentes secciones de una aplicación Android, tales como:

- Launcher icons
- Action bar and tab icons
- Notification icons
- Entre otros...

<http://android-ui-utils.googlecode.com/hg/asset-studio/dist/index.html>





[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

*Pasión por la tecnología Java*

Experiencia y Conocimiento para tu vida