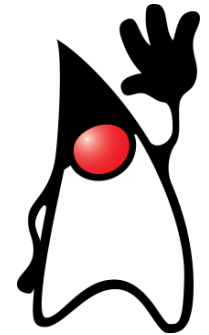




Android Con Java



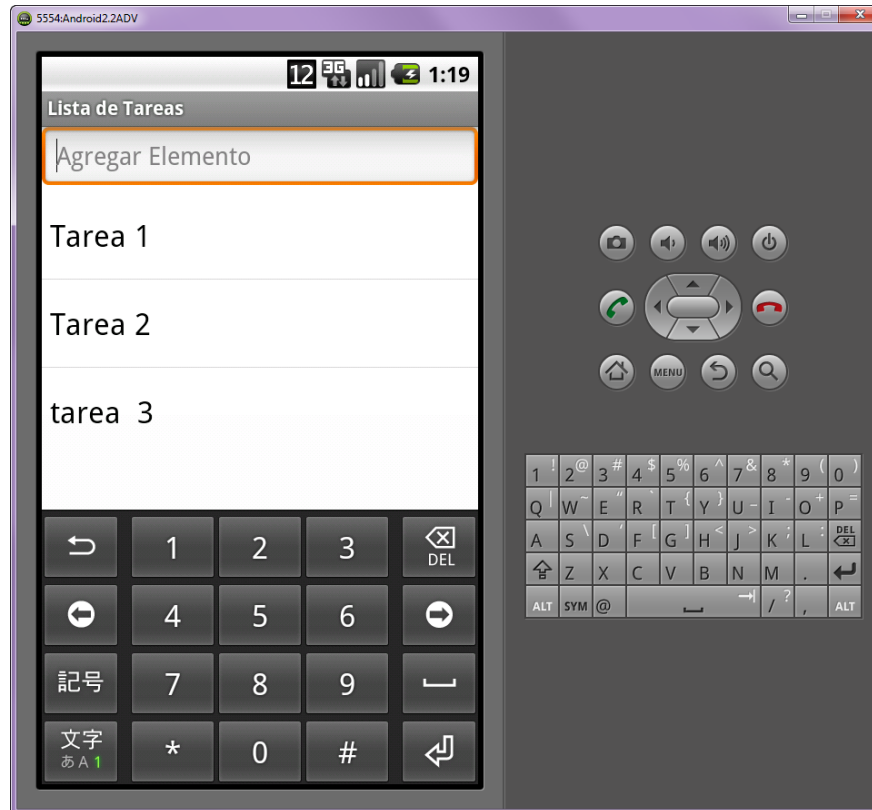
Ejercicio 4



Manejo de Layouts
en Android

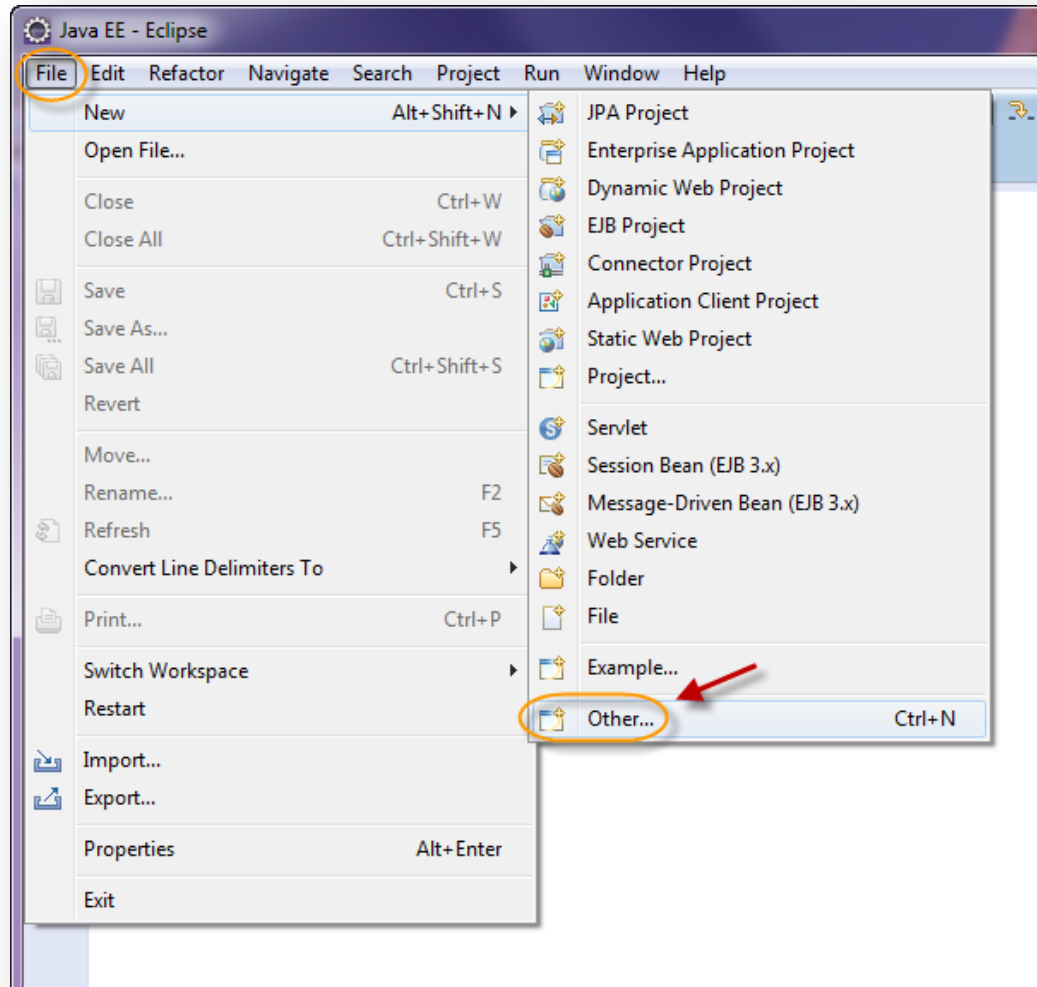
Objetivo del Ejercicio

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



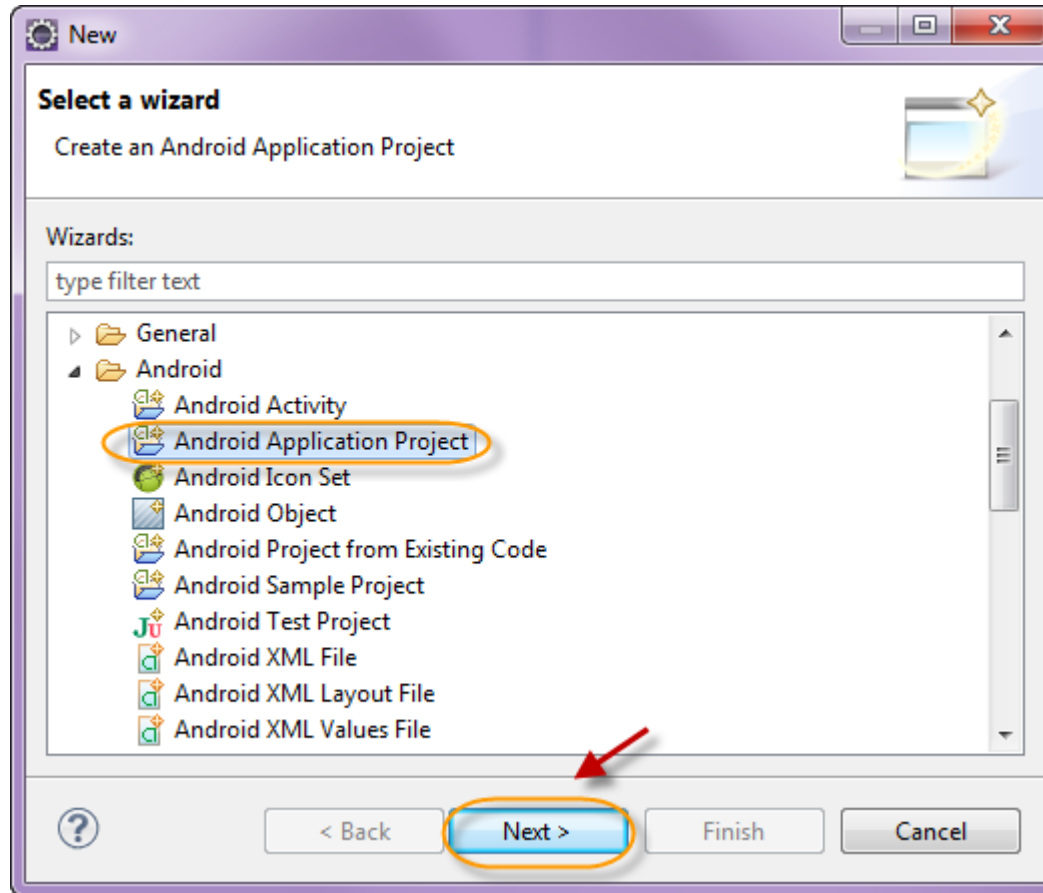
Paso 1. Creación proyecto ListaTareas

Creamos el proyecto ListaTareas:



Paso 1. Creación proyecto ListaTareas (cont)

Creamos el proyecto ListaTareas con Android:



Paso 1. Creación proyecto ListaTareas (cont)

Creamos el proyecto ListaTareas con Android:

New Android App

New Android Application
Creates a new Android Application

Application Name: ListaTareas

Project Name: ListaTareas

Package Name: mx.com.gm.listatareas

Build SDK: Android 4.1 (API 16) Choose...

Minimum Required SDK: API 8: Android 2.2 (Froyo)

☐ Create custom launcher icon

☐ Mark this project as a library

☒ Create Project in Workspace

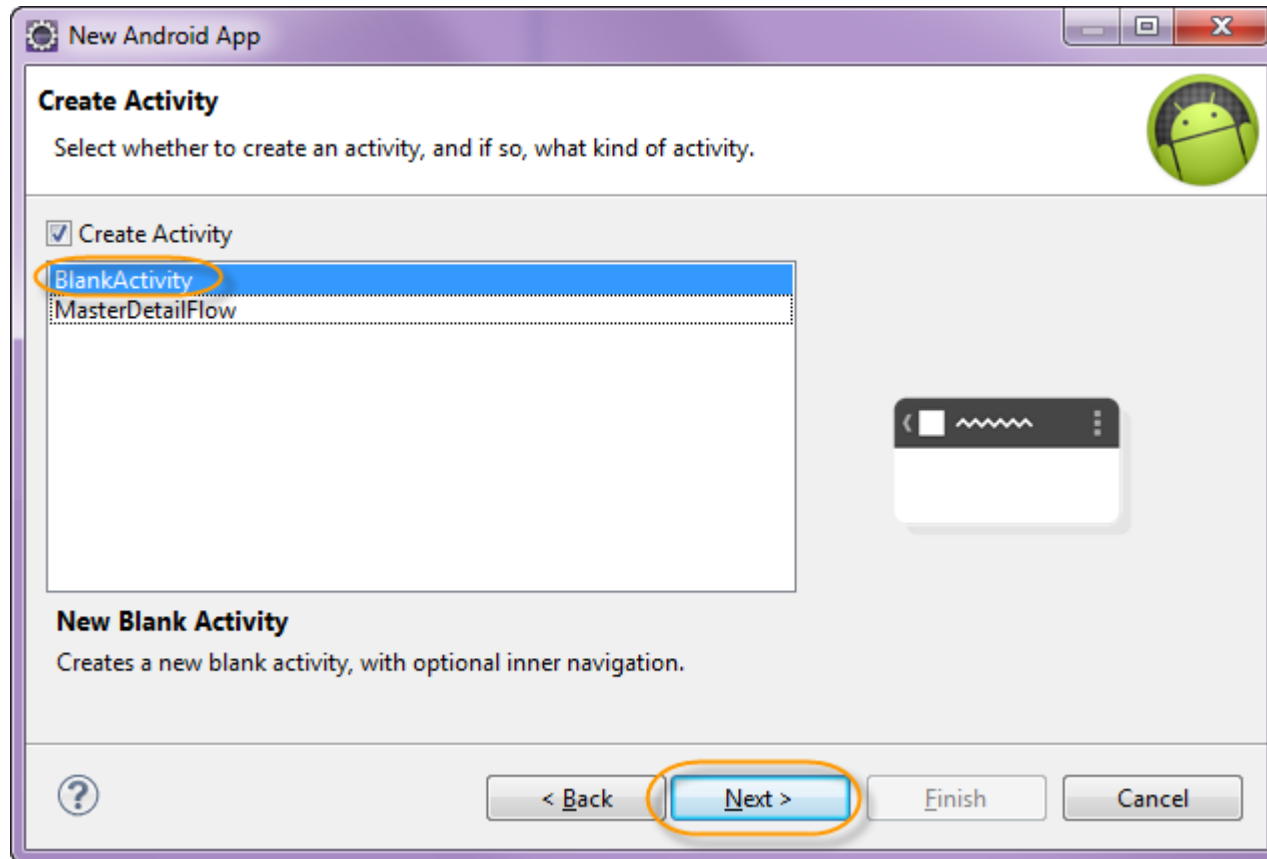
Location: C:\AndroidWorkspace\ListaTareas Browse...

The package name must be a unique identifier for your application. It is typically not shown to users, but it *must* stay the same for the lifetime of your application; it is how multiple versions of the same application are considered the "same app". This is typically the reverse domain name of your organization plus one or more application identifiers, and it

< Back Next > Finish Cancel

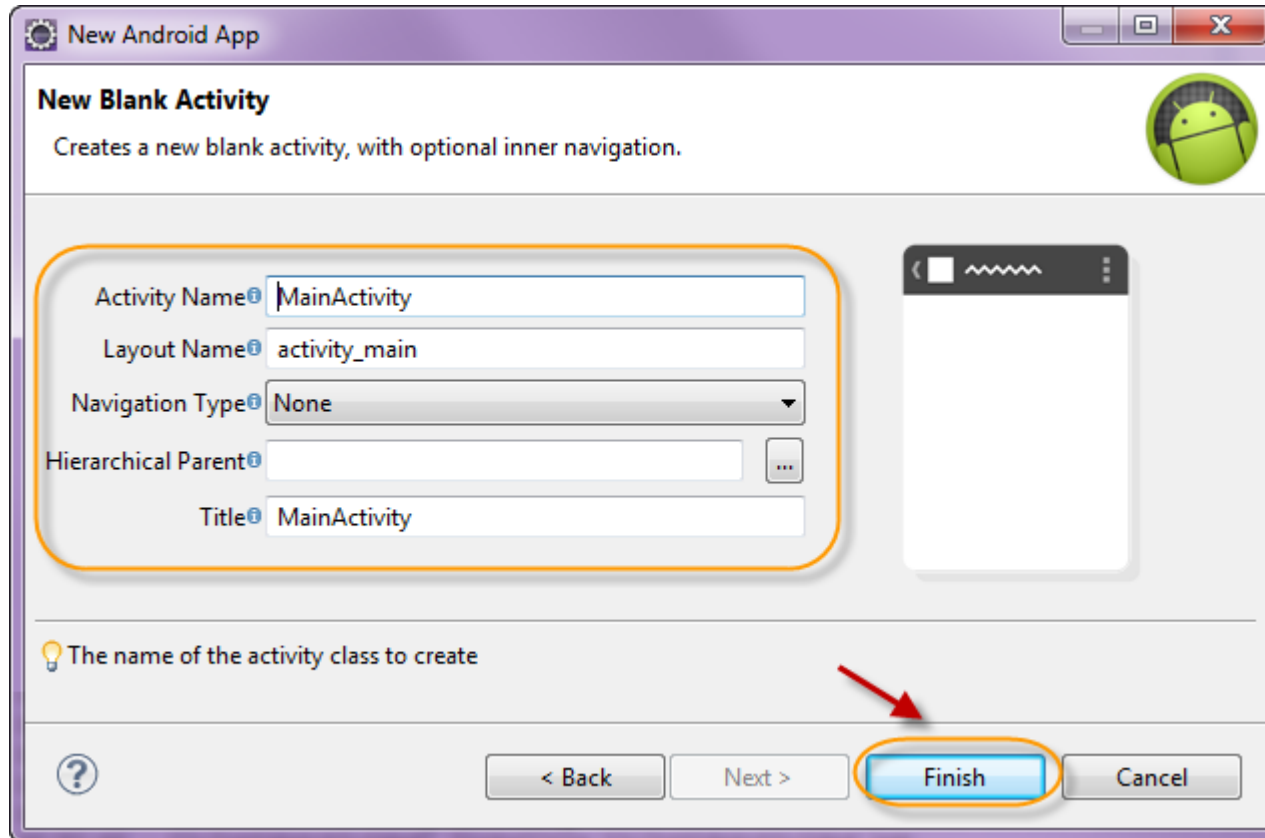
Paso 1. Creación proyecto ListaTareas (cont)

Creamos el proyecto ListaTareas con Android:



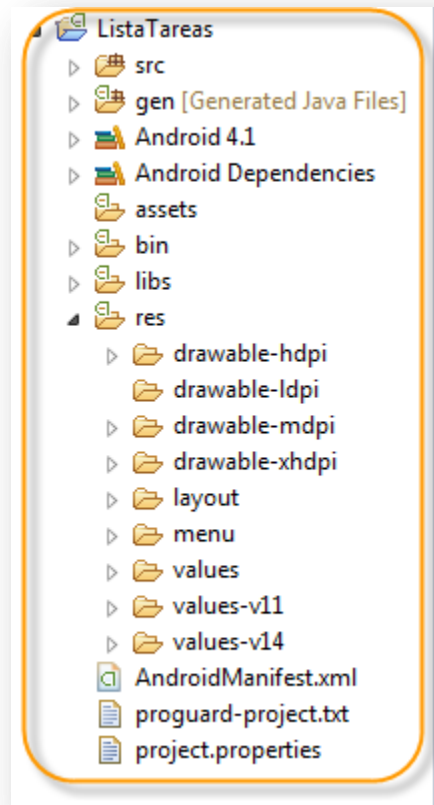
Paso 1. Creación proyecto ListaTareas (cont)

Creamos el proyecto ListaTareas con Android:



Paso 1. Creación proyecto ListaTareas (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">ListaTareas</string>
```

```
    <string name="menu_settings">Configuración</string>
```

```
    <string name="title_activity_main">Lista de Tareas</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:

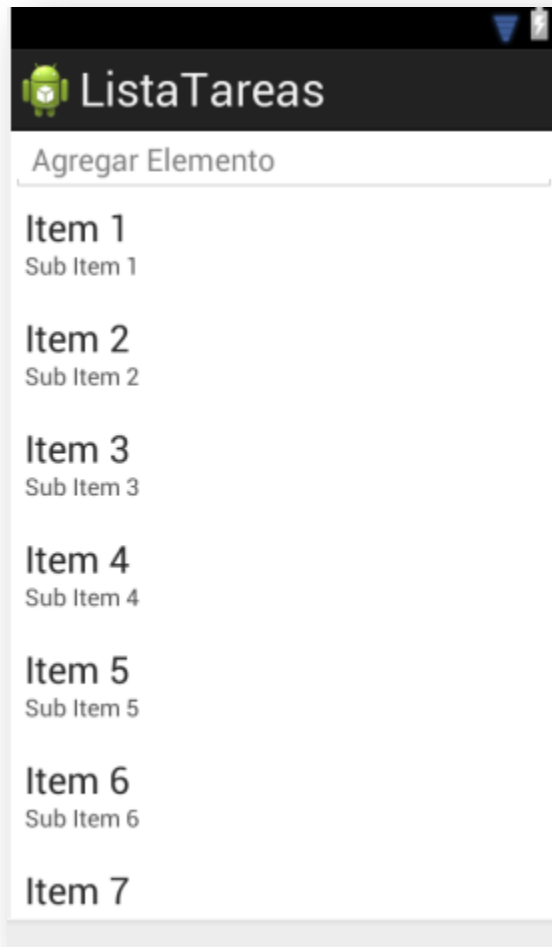
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <EditText
        android:id="@+id/editTextTarea"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/agregarElemento"
        android:contentDescription="@string/agregarDescElemento">

    </EditText>
    <ListView
        android:id="@+id/listViewTarea"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>
```



Paso 3. Modificamos el layout activity_main.xml (cont)

El resultado debe ser similar al siguiente:



Paso 4. Modificamos la clase MainActivity.java

Agregamos el siguiente código a la clase MainActivity.java:

```
package mx.com.gm.listatareas;
```

```
import java.util.ArrayList;
import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.EditText;
import android.widget.ListView;
```

```
public class MainActivity extends Activity {
```

```
    Override
```

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

```
        //1. Obtenemos las referencias a los elementos UI
```

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

```
        final EditText editTextTarea = (EditText)
        findViewById(R.id.editTextTarea);
```

```
        //2. Creamos un arrayList de tareas, y agregamos algunas tareas
```

```
        final ArrayList<String> tareas = new ArrayList<String>();
        tareas.add("Tarea 1");
        tareas.add("Tarea 2");
```

```
        //3. Creamos el adaptador que enlazara el arreglo con el ListView
```

```
        //Utilizamos un layout que ya nos proporciona Android: simple_list_item_1
```

```
        final ArrayAdapter<String> adaptador =
            new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, tareas);
```

```
        //4. Enlazamos el adaptador de datos con el ListView
```

```
        listViewTarea.setAdapter(adaptador);
```

```
        //5. Capturamos el evento Enter para agregar nuevos elementos
```

```
        editTextTarea.setOnClickListener(new OnClickListener() {
```

```
            public boolean onKey(View v, int keyCode, KeyEvent event) {
```

```
                if (event.getAction() == KeyEvent.ACTION_DOWN) {
                    if (keyCode == KeyEvent.KEYCODE_DPAD_CENTER
                        || keyCode == KeyEvent.KEYCODE_ENTER) {
```

```
                        //Agregamos la nueva tarea
```

```
                        tareas.add(editTextTarea.getText().toString());
```

```
                        editTextTarea.setText("");
```

```
                        //Notificamos el cambio en la lista de tareas
```

```
                        adaptador.notifyDataSetChanged();
```

```
                        return true;
```

```
                    }
```

```
                }
```

```
                return false;
```

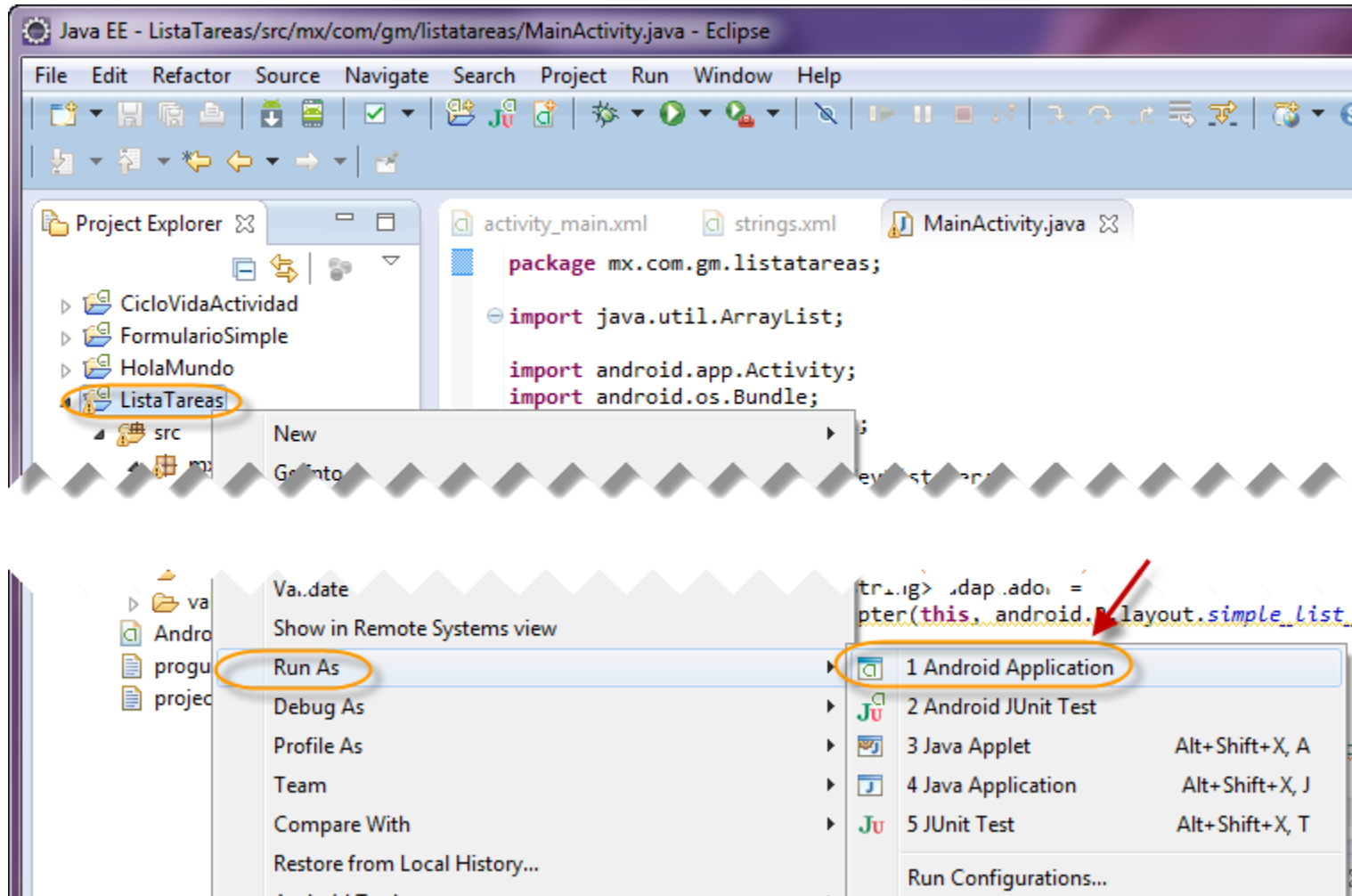
```
            }
```

```
        });
```

```
    }
```

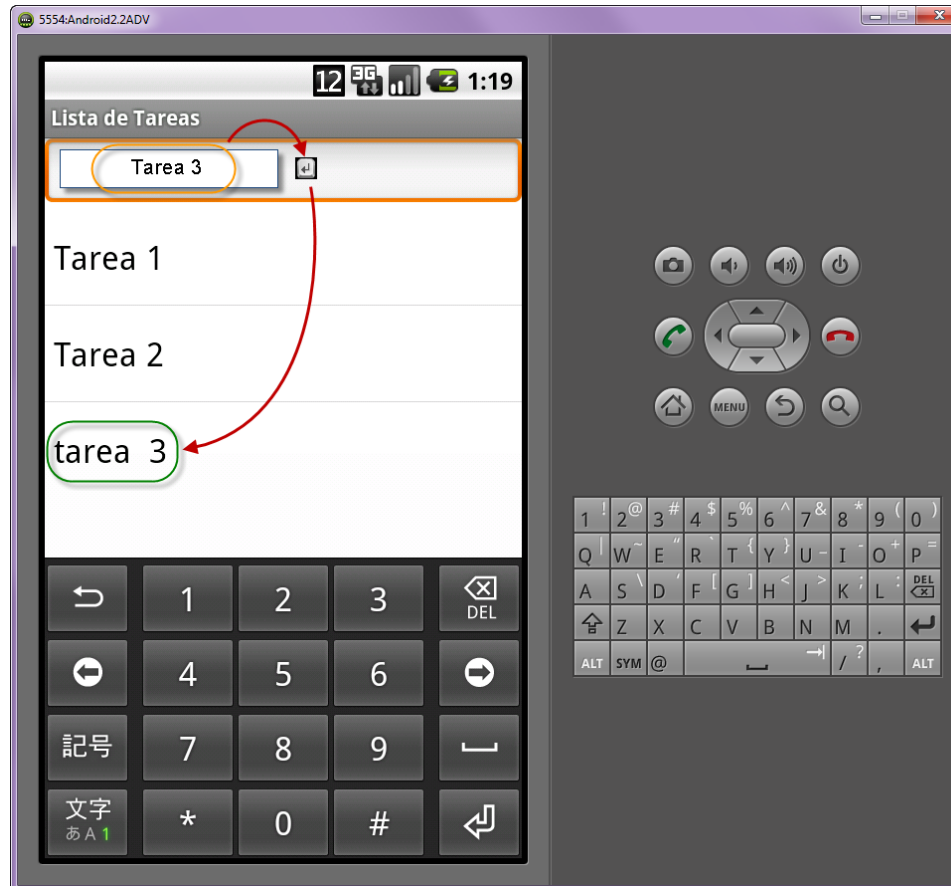
Paso 5. Ejecutamos el proyecto Lista de Tareas

Ejecutamos la aplicación como sigue:



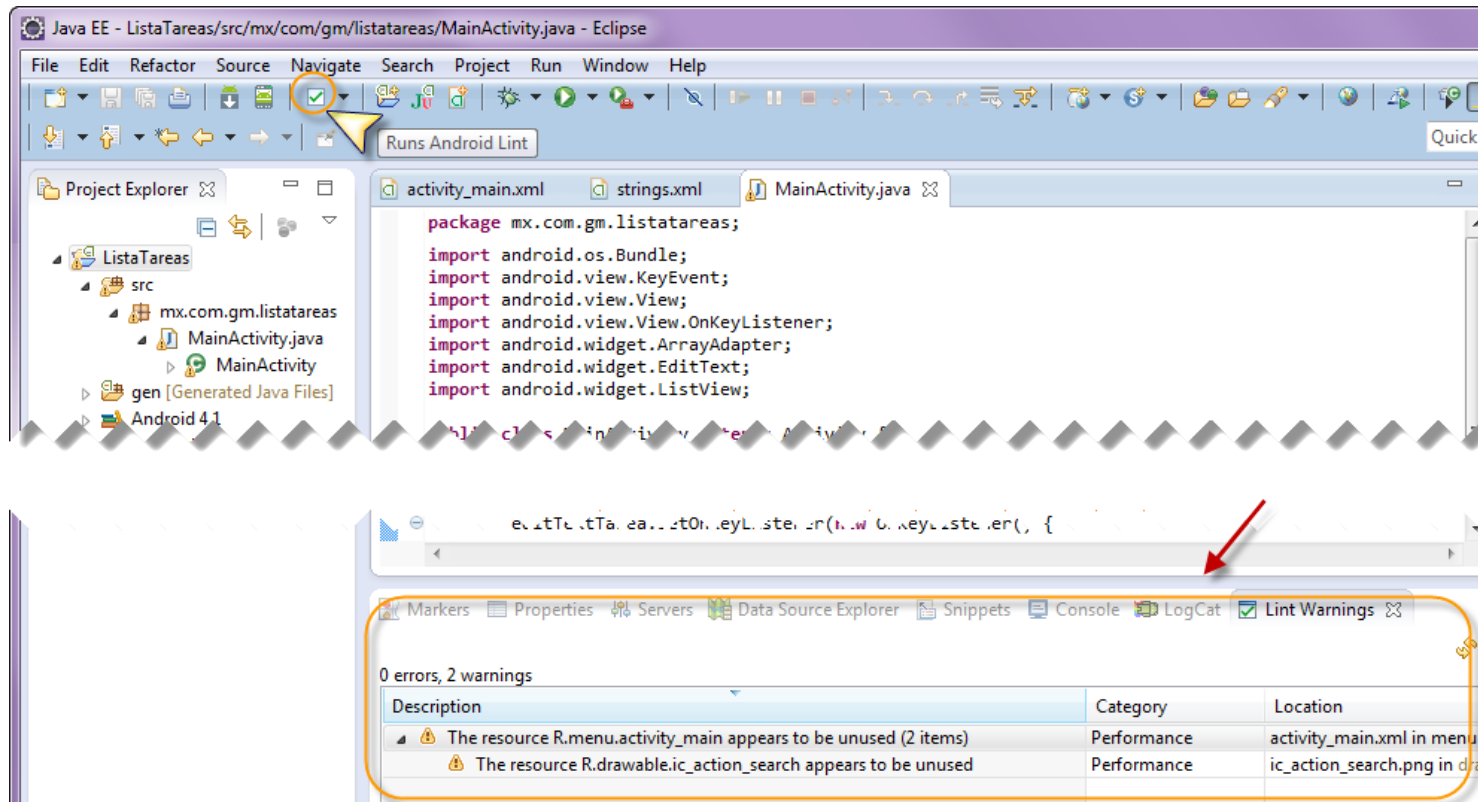
Paso 5. Ejecutamos el proyecto Lista de Tareas (cont)

Podemos observar que al escribir un texto en el campo de "Agregar Elemento", y presionar el botón de Enter, estamos respondiendo al evento y se agrega una nueva tarea a la lista:



Paso 6. Revisión de la Aplicación con Lint

Android proporciona la herramienta Lint, con el objetivo de verificar si tenemos algún problema en nuestra aplicación con el objetivo de optimizarla. Para ejecutar Lint en Eclipse damos click en:

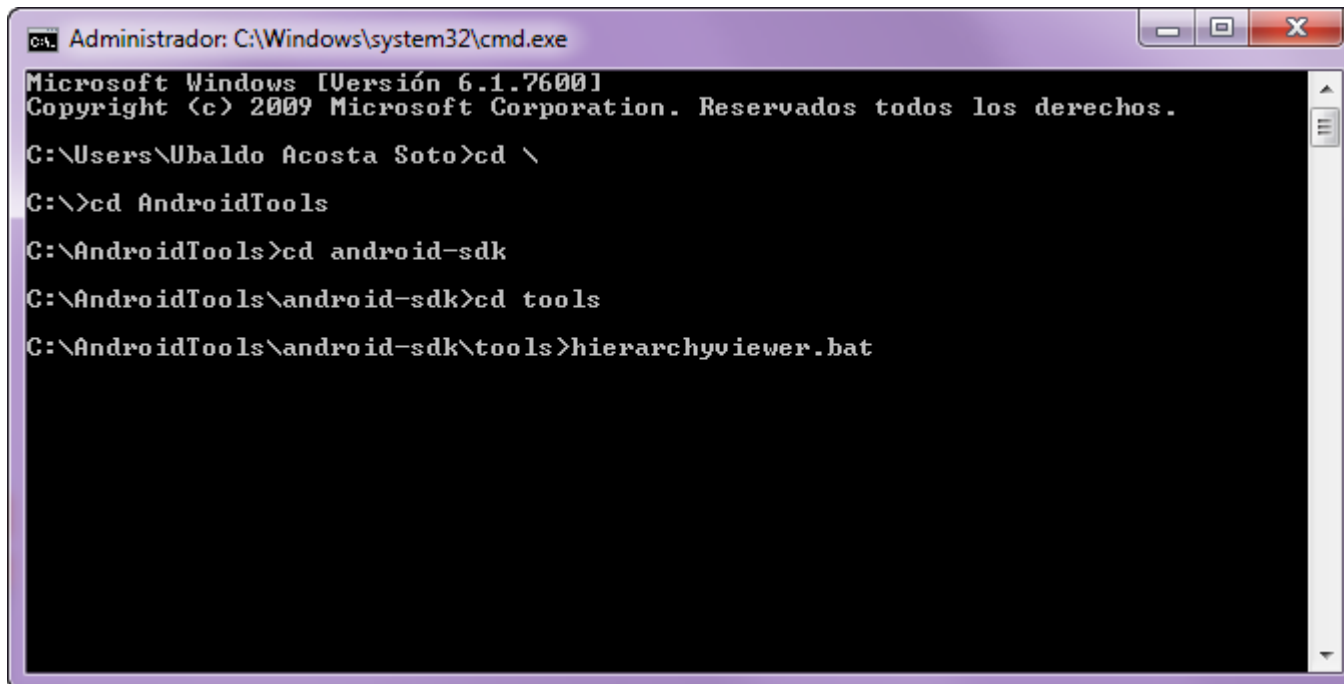


Para conocer todas las características que ofrece Lint, pueden consultar: <http://tools.android.com/tips/lint-checks>

Paso 7. Revisión del Layout con Hierarchy Viewer

Android proporciona la herramienta Hierarchy Viewer con el objetivo revisar a detalle una Actividad, y realizar tareas de optimizaciones:

Para ejecutar esta herramienta, entramos a la instalación del sdk de Android: `C:\AndroidTools\android-sdk\tools\hierarchyviewer.bat`

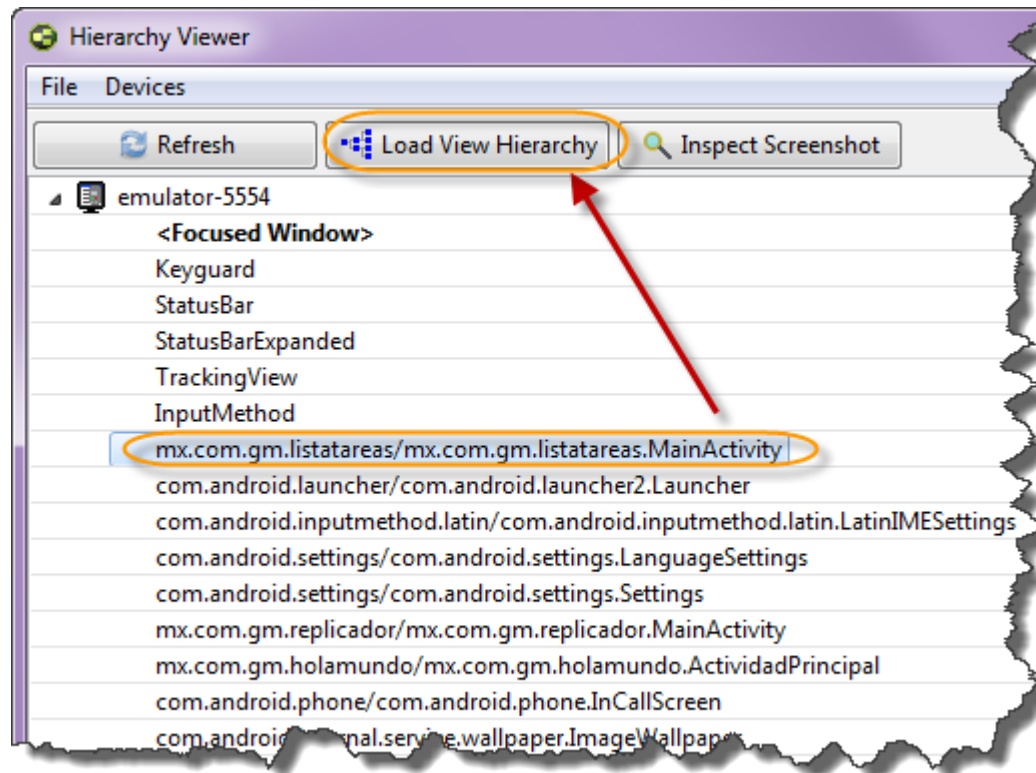


```
Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Ubaldo Acosta Soto>cd \
C:\>cd AndroidTools
C:\AndroidTools>cd android-sdk
C:\AndroidTools\android-sdk>cd tools
C:\AndroidTools\android-sdk\tools>hierarchyviewer.bat
```

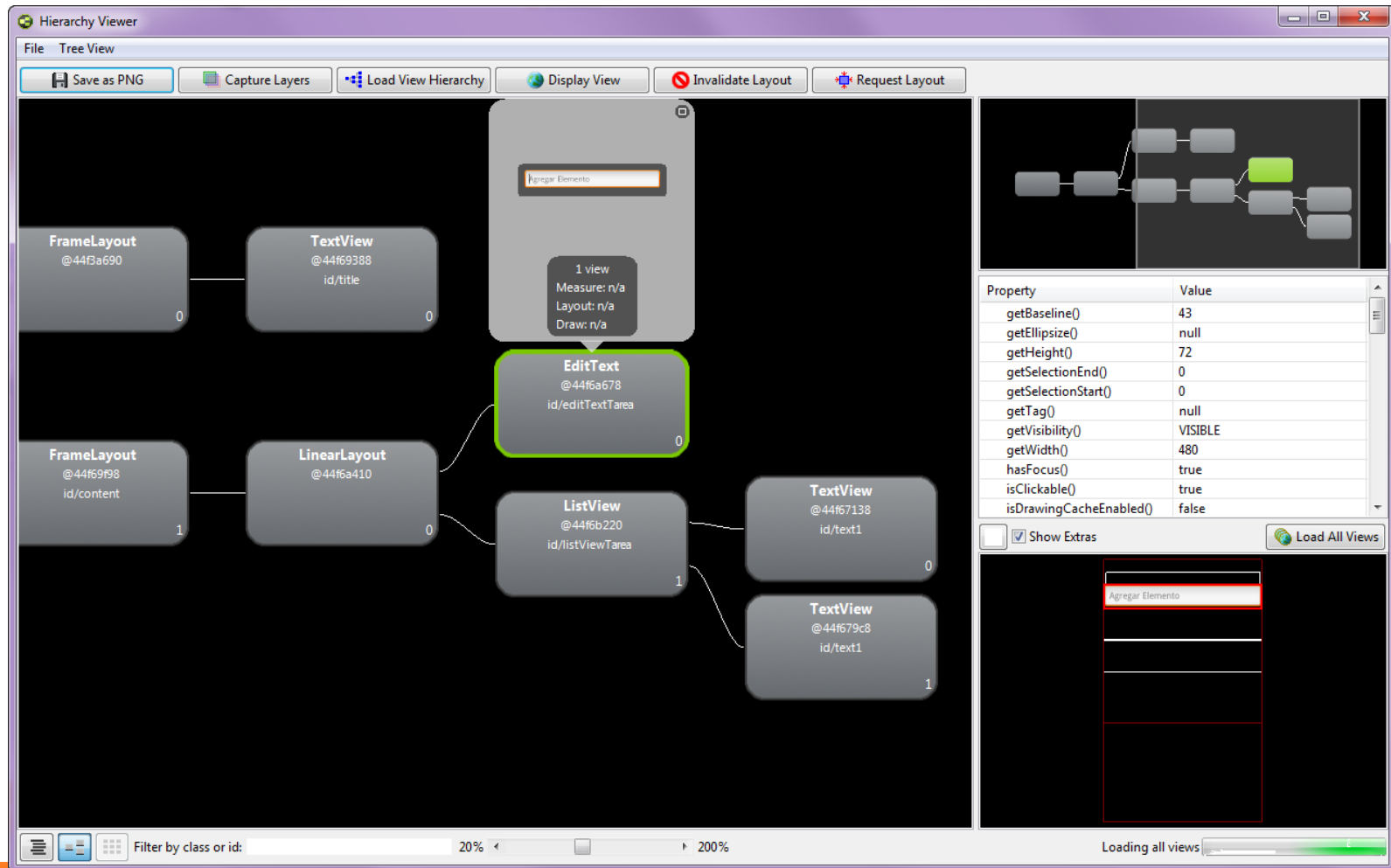

Paso 7. Revisión de Layout con Hierarchy Viewer (cont)

Una vez ejecutado, seleccionamos la actividad a analizar, y damos click en Load View Hierarchy:



Paso 7. Revisión de Layout con Hierarchy Viewer (cont)

Con esta herramienta podemos analizar la jerarquía de vistas, así como cada uno de los elementos de la actividad seleccionada.





Conclusión

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

Además, mencionamos las herramientas Lint y Hierarchy Viewer, las cuales son proporcionadas por el SDK de Android con el objetivo de optimizar nuestras aplicaciones.

Este ejercicio es muy simple, así que en el siguiente ejercicio personalizaremos la lista a mostrar, extendiendo el adaptador para soportar no solamente cadenas, sino objetos personalizados.



www.globalmentoring.com.mx

Pasión por la tecnología Java

Experiencia y Conocimiento para tu vida