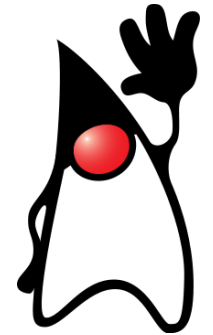




# Android Con Java



## Ejercicio 18



Configuración de  
Widgets en Android

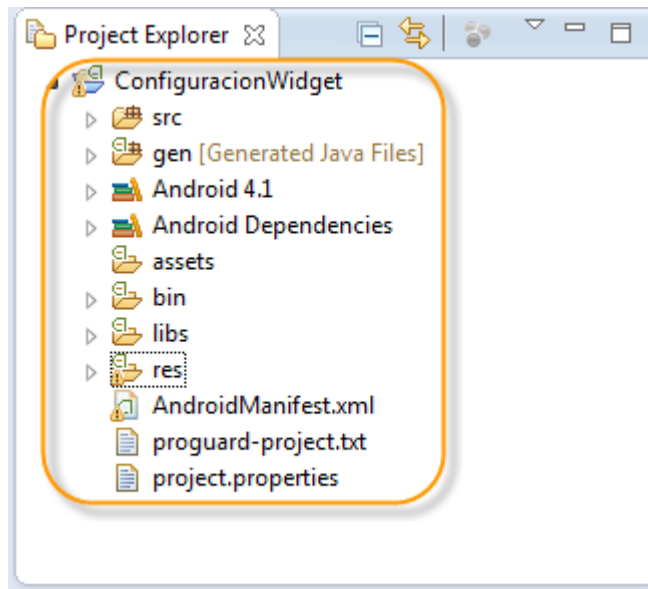
## Objetivo del Ejercicio

- El objetivo del ejercicio crear un Widget que antes de utilizarlo nos permita configurarlo. En este caso, solicitar la URL que desea visualizar. Al finalizar deberemos observar la siguiente aplicación:



# Paso 1. Crear el proyecto ConfiguracionWidget

Creamos el proyecto ConfiguracionWidget:





## Paso 2. Modificar el archivo strings.xml

Modificamos el archivo strings.xml ubicado "res/values":

```
<resources>
```

```
    <string name="app_name">ConfiguracionWidget</string>
    <string name="menu_settings">Settings</string>
    <string name="title_activity_main">ConfiguracionWidget</string>
    <string name="abrir">Abrir URL</string>
    <string name="title_activity_configuracion">Configuracion</string>
    <string name="crear">Crear Widget</string>
```

```
</resources>
```



## Paso 3. Modificar la actividad activity\_main.xml

Modificamos la actividad activity\_main.xml:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

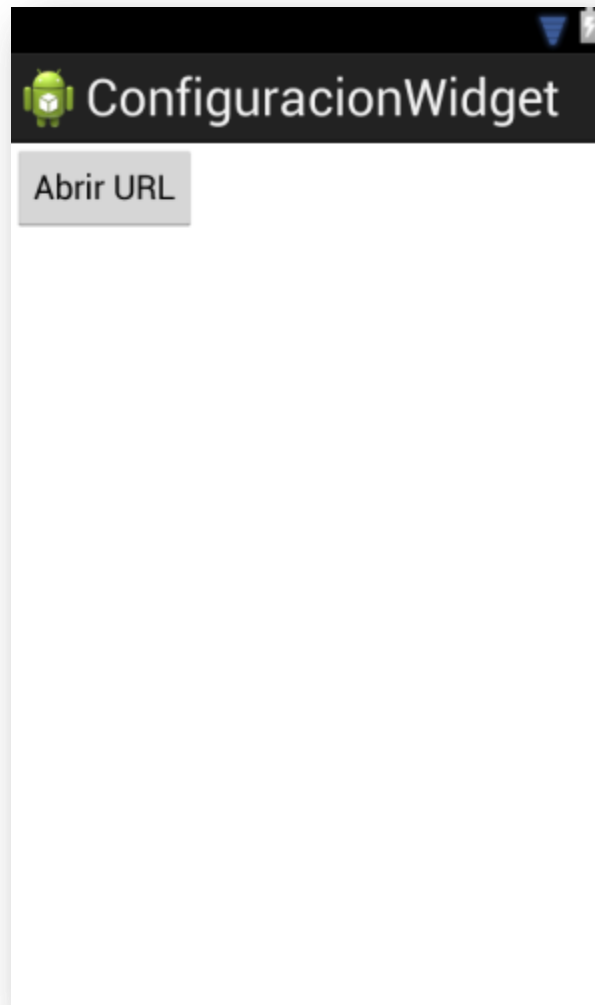
    <Button
        android:id="@+id/botonAbrirNavegador"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:text="@string/abrir" />

</RelativeLayout>
```



## Paso 3. Modificar la actividad activity\_main.xml (cont)

El resultado debe ser similar al siguiente:





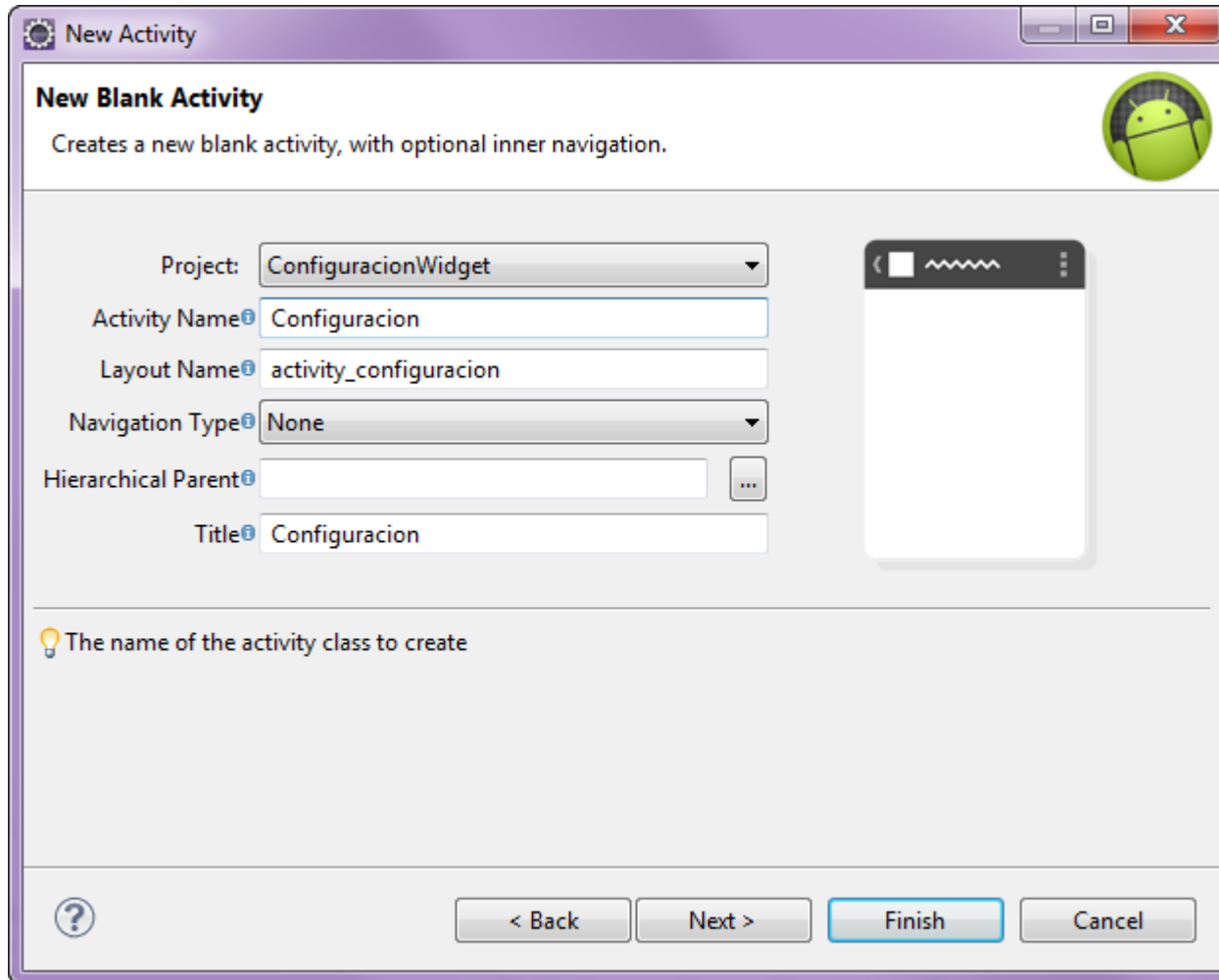
## Paso 4. Modificamos la clase MainActivity.java

El código de la clase MainActivity debe ser el siguiente:

```
package mx.com.gm.configuracionwidget;  
  
import android.appwidget.AppWidgetProvider;  
  
public class MainActivity extends AppWidgetProvider {  
  
}
```

## Paso 5. Creamos la actividad activity\_configuracion.xml

Creamos la actividad activity\_configuracion.xml



**New Activity**

**New Blank Activity**  
Creates a new blank activity, with optional inner navigation.

Project: ConfiguracionWidget

Activity Name: Configuracion

Layout Name: activity\_configuracion

Navigation Type: None

Hierarchical Parent:

Title: Configuracion

The name of the activity class to create

< Back Next > Finish Cancel





## Paso 5. Creamos la actividad activity\_configuracion.xml (cont)

La actividad activity\_configuracion.xml debe quedar como sigue:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
```

```
<EditText
```

```
    android:id="@+id/editTextURL"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:ems="10"
    android:text="http://icursos.net" >
```

```
<requestFocus />
```

```
</EditText>
```

```
<Button
```

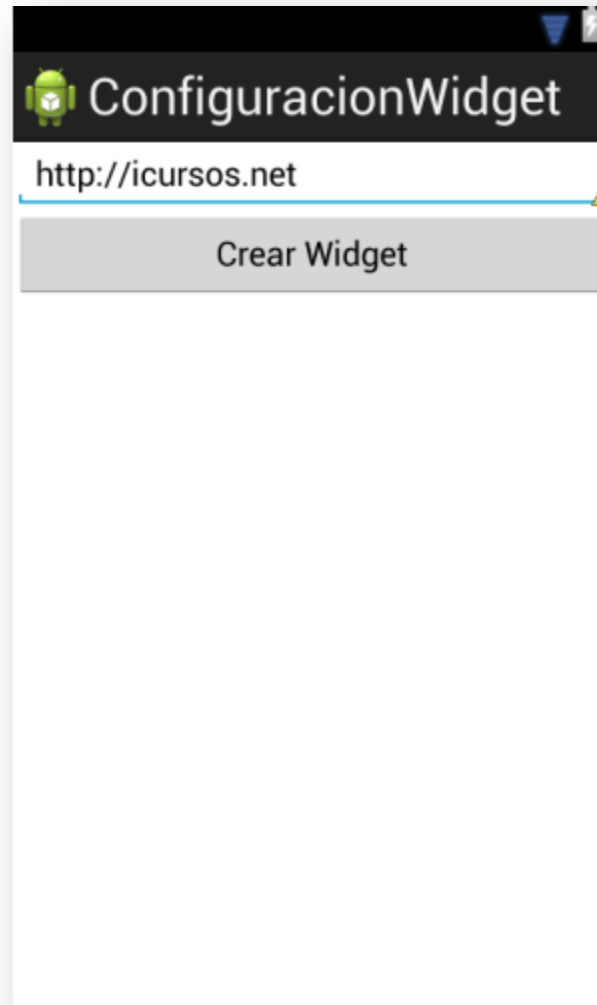
```
    android:id="@+id/botonCrearWidget"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/editTextURL"
    android:text="@string/crear" />
```

```
</RelativeLayout>
```



## Paso 5. Creamos la actividad `activity_configuracion.xml` (cont)

El resultado debe ser similar al siguiente:



# Paso 6. Modificamos la clase Configuracion.java

El código debe ser el siguiente:

```
package mx.com.gm.configuracionwidget;
```

```
import mx.com.gm.configuracionwidget.R;
import android.app.Activity;
import android.app.PendingIntent;
import android.appwidget.AppWidgetManager;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RemoteViews;
```

```
public class Configuracion extends Activity {
```

```
    private Configuracion contexto;
    private int widgetId;
```

Override

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

```
    contexto = this;
```

```
    setResult(RESULT_CANCELED);
```

```
    Bundle extras = getIntent().getExtras();
```

```
    if (extras != null) {
```

```
        widgetId = extras.getInt(AppWidgetManager.EXTRA_APPWIDGET_ID,
AppWidgetManager.INVALID_APPWIDGET_ID);
    }
```

```
    final AppWidgetManager widgetManager =
```

```
AppWidgetManager.getInstance(contexto);
```

```
    final RemoteViews views = new RemoteViews(contexto.getPackageName(),
R.layout.activity_main);
```

```
final EditText editText = (EditText) findViewById(R.id.editTextURL);
Button boton = (Button) findViewById(R.id.botonCrearWidget);
boton.setOnClickListener(new OnClickListener() {
```

```
    public void onClick(View v) {
        Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse(editText.getText().toString()));
        PendingIntent pendingIntent = PendingIntent.getActivity(contexto, 0,
intent, 0);
        views.setOnClickPendingIntent(R.id.botonAbrirNavegador, pendingIntent);
        widgetManager.updateAppWidget(widgetId, views);

        Intent resultadoIntent = new Intent();
        resultadoIntent.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID,
widgetId);
        setResult(RESULT_OK, resultadoIntent);
        finish();
    }
});
}
```



## Paso 7. Creamos el archivo widget.xml

Creamos el archivo de configuración widget.xml, y lo depositamos en "res/xml/widget.xml". A diferencia del ejercicio anterior, en este caso indicamos que existe una clase de configuración antes de mostrar el widget al usuario. El xml queda como sigue:

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:updatePeriodMillis="0" android:initialLayout="@layout/activity_main"
    android:minHeight="72dp" android:minWidth="144dp"
    android:configure="mx.com.gm.configuracionwidget.Configuracion">

</appwidget-provider>
```



## Paso 8. Modificamos el archivo de manifiesto

Modificamos el archivo AndroidManifest.xml, quedando como sigue:

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

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

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <receiver android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
            </intent-filter>

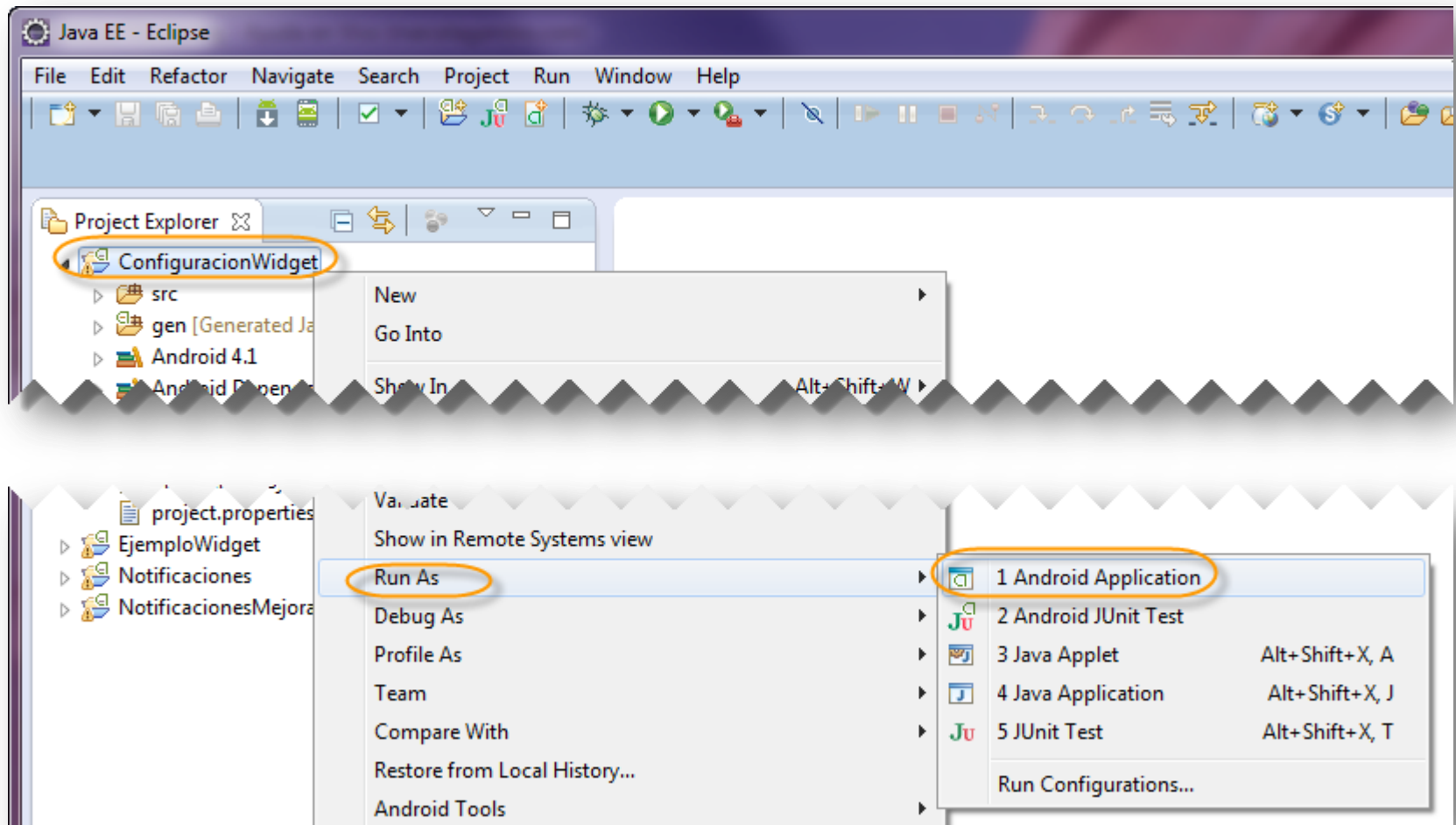
            <meta-data
                android:name="android.appwidget.provider"
                android:resource="@xml/widget" />
        </receiver>

        <activity
            android:name=".Configuracion"
            android:label="@string/title_activity_configuracion" >
            <intent-filter>
                <action android:name="android.appwidget.action.APPWIDGET_CONFIGURE" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

## Paso 9. Ejecutamos el proyecto

Para ejecutar el proyecto ConfiguracionWidget:



## Paso 9. Ejecutamos el proyecto (cont)

Una vez instalada la aplicación, debemos instalar el Widget para que pueda visualizarse. Para ello debemos estar en la pantalla de inicio, dejamos presionado la pantalla por algunos segundos (long touch) y aparecerá un menú:





## Paso 7. Ejecutamos el proyecto (cont)

Una vez visualizado nuestro Widget, a diferencia del ejercicio anterior, podemos observar que se nos muestra la actividad de configuración del widget. Una vez establecido el URL, damos click en Crear Widget y listo. Podemos crear varias versiones de un widget, por lo que podemos tener botones apuntando a distintas URLs.







## Conclusión

Con este ejercicio hemos revisado cómo configurar un Widget antes de mostrarlo y visualizarlo en la pantalla de inicio de nuestro dispositivo Android.

Con esto ya tenemos las bases para la creación de Widgets en Android, y así concluimos esta lección.



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

*Pasión por la tecnología Java*

Experiencia y Conocimiento para tu vida