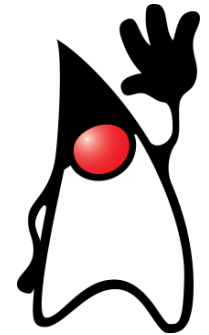




Android Con Java



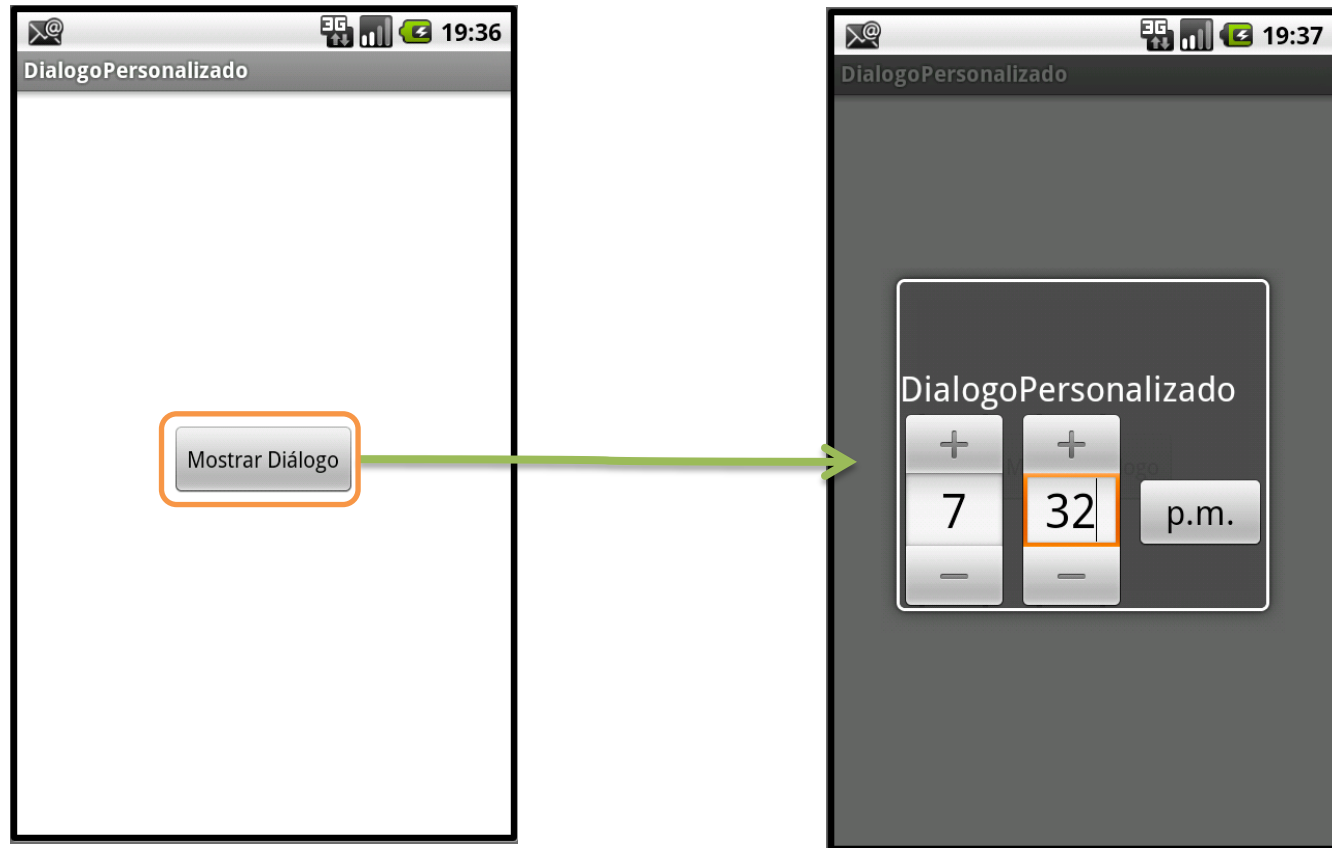
Ejercicio 14



Manejo de
Diálogos Personalizados

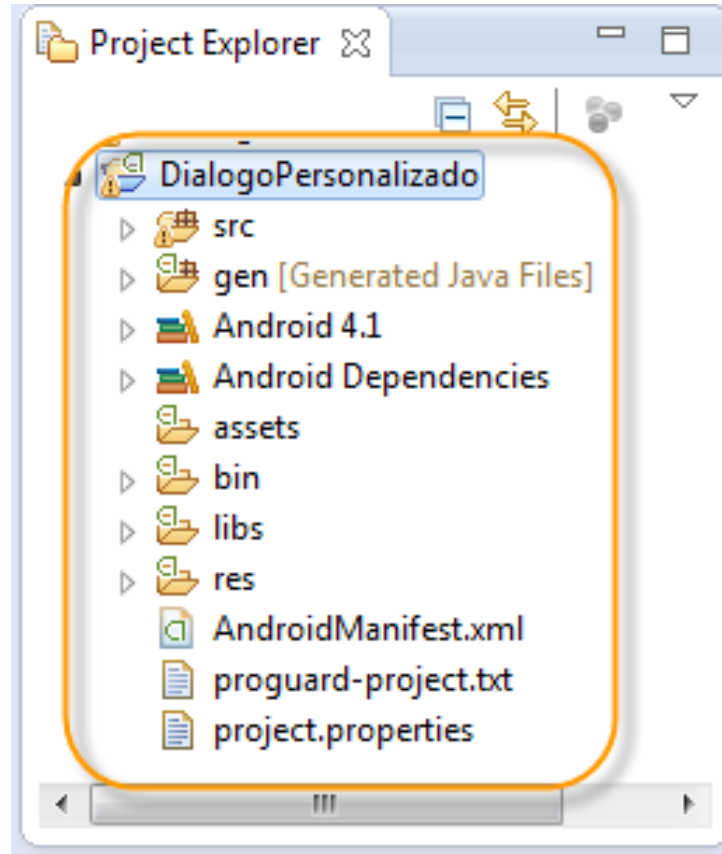
Objetivo del Ejercicio

- El objetivo del ejercicio es entender el manejo de Menú de Diálogos Personalizados. Al finalizar deberemos observar la siguiente aplicación Android:



Paso 1. Crear el proyecto DialogoPersonalizado

Creamos el proyecto DialogoPersonalizado:





Paso 2. Modificar el archivo strings.xml

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

```
<resources>
```

```
<string name="app_name">DialogoPersonalizado</string>  
<string name="menu_settings">Settings</string>  
<string name="title_activity_main">DialogoPersonalizado</string>  
<string name="dialogo">DialogoPersonalizado</string>  
<string name="mostrar">Mostrar Diálogo</string>
```

```
</resources>
```



Paso 3. Modificar el menú activity_main.xml

Modificamos el menú 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" >

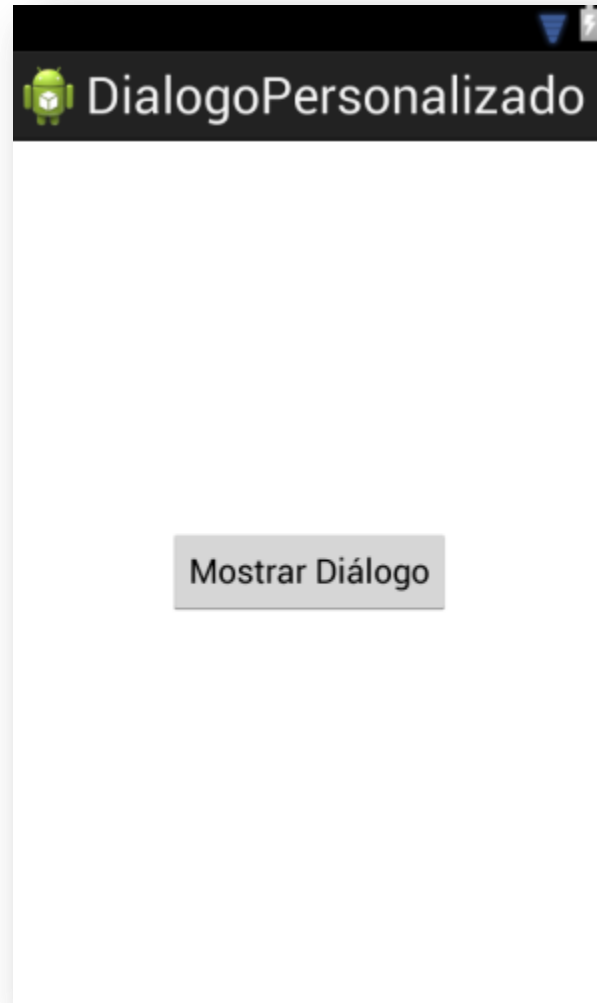
    <Button
        android:id="@+id/botonMostrarDialogo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/mostrar" />

</RelativeLayout>
```



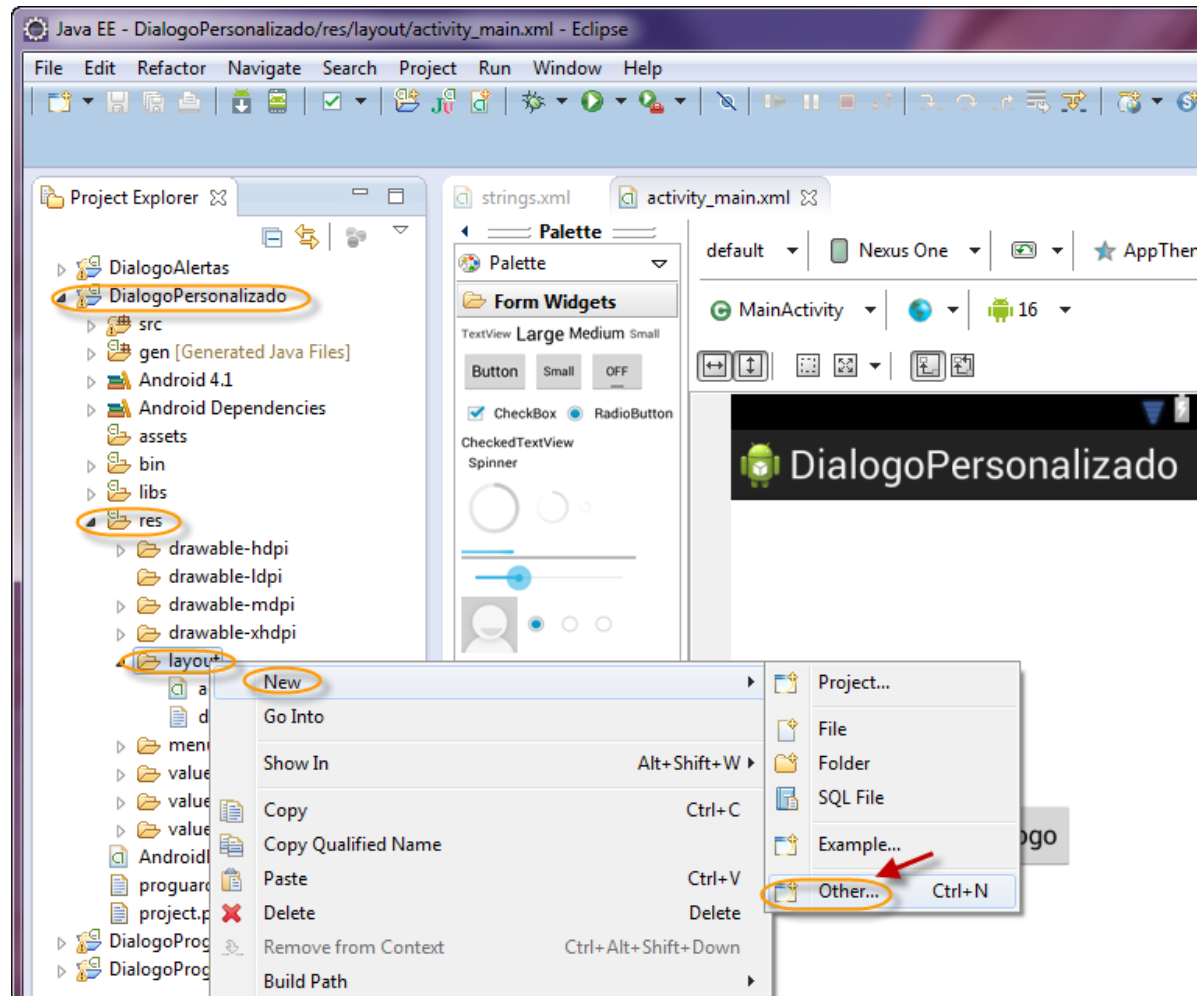
Paso 3. Modificar el menú activity_main.xml (cont)

El resultado debe ser similar al siguiente:



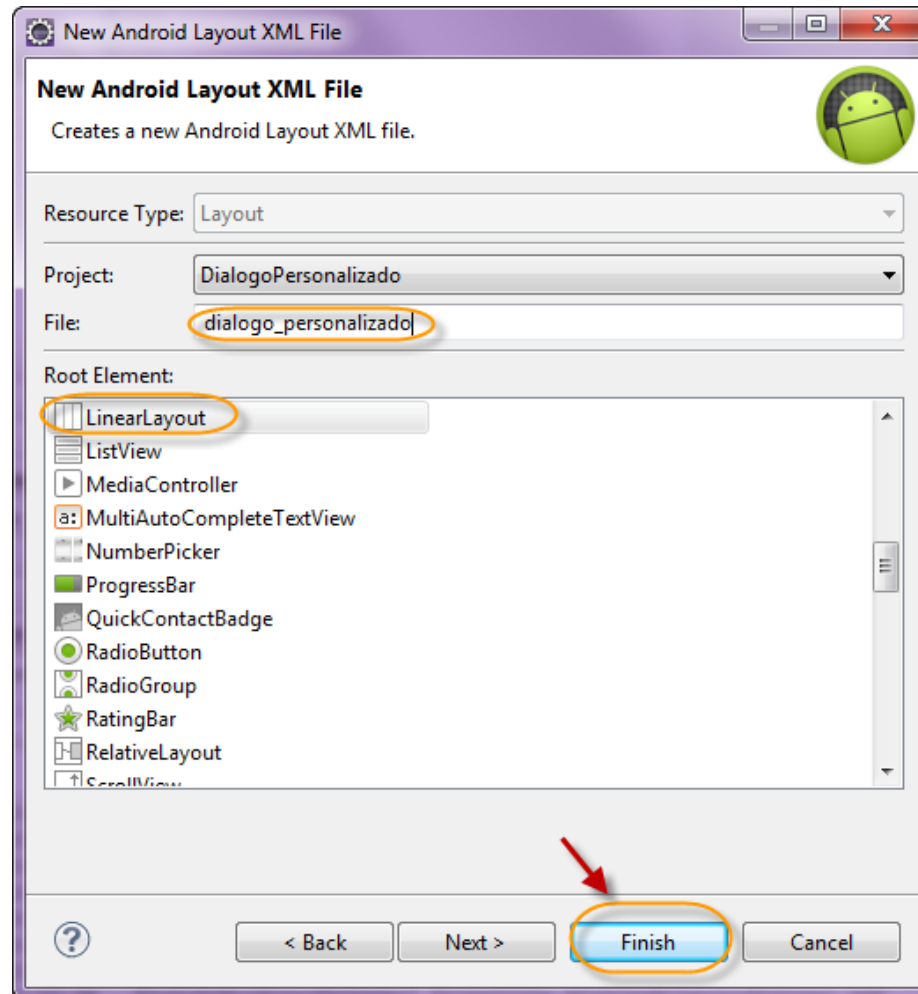
Paso 4. Crear el layout del diálogo

Creamos el archivo xml del diálogo personalizado (layout):



Paso 4. Crear el layout del diálogo (cont)

Creamos el archivo xml del diálogo personalizado (layout):





Paso 4. Crear el layout del diálogo (cont)

Modificamos el menú dialogo_personalizado.xml:

```
<?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:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/dialogo"
        android:textAppearance="?android:attr/textAppearanceLarge" />

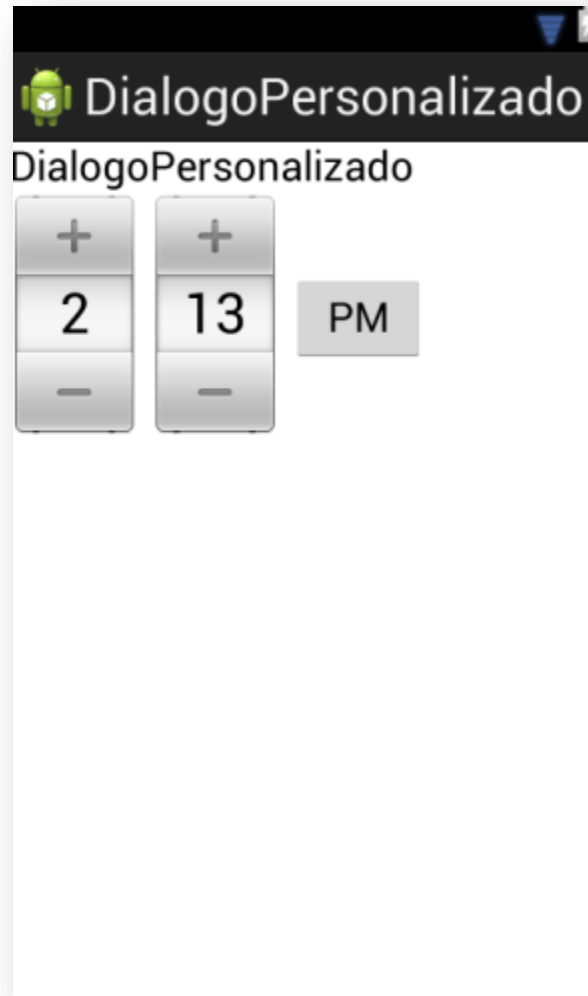
    <TimePicker
        android:id="@+id/timePicker1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```



Paso 4. Crear el layout del diálogo (cont)

El resultado debe ser similar al siguiente:



Paso 5. Modificamos la clase MainActivity.java

Modificamos la clase MainActivity.java.

```
package mx.com.gm.dialogopersonalizado;
```

```
import android.app.Activity;
import android.app.Dialog;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
```

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

```
    final private int MOSTRAR_DIALOGO = 0;
```

```
    private Dialog dialogo = null;
```

```
    private Handler handler = new Handler() {
```

```
        public void handleMessage(android.os.Message msg) {
            dialogo.dismiss();
        }
    };
```

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    Button botonMostrarDialogo = (Button) findViewById(R.id.botonMostrarDialogo);
```

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

```
        public void onClick(View v) {
```

```
            showDialog(MOSTRAR_DIALOGO);
```

```
        }
    });
```

```
}
```

```
protected android.app.Dialog onCreateDialog(int id) {
```

```
    switch (id) {
```

```
        case MOSTRAR_DIALOGO:
```

```
            dialogo = new Dialog(this);
```

```
            dialogo.setContentView(R.layout.dialogo_personalizado);
```

```
            return dialogo;
```

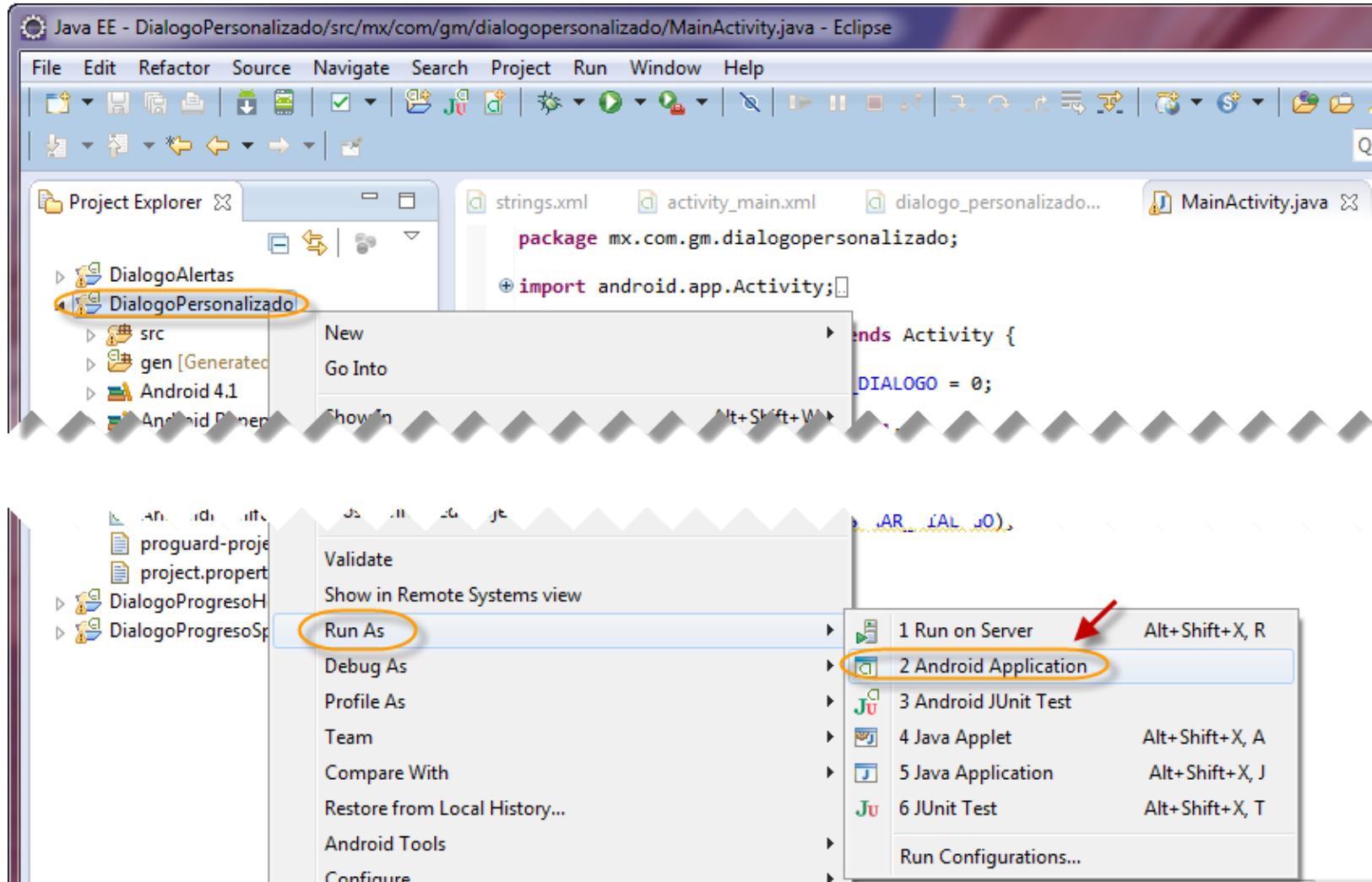
```
        }
```

```
    return null;
```

```
}
```

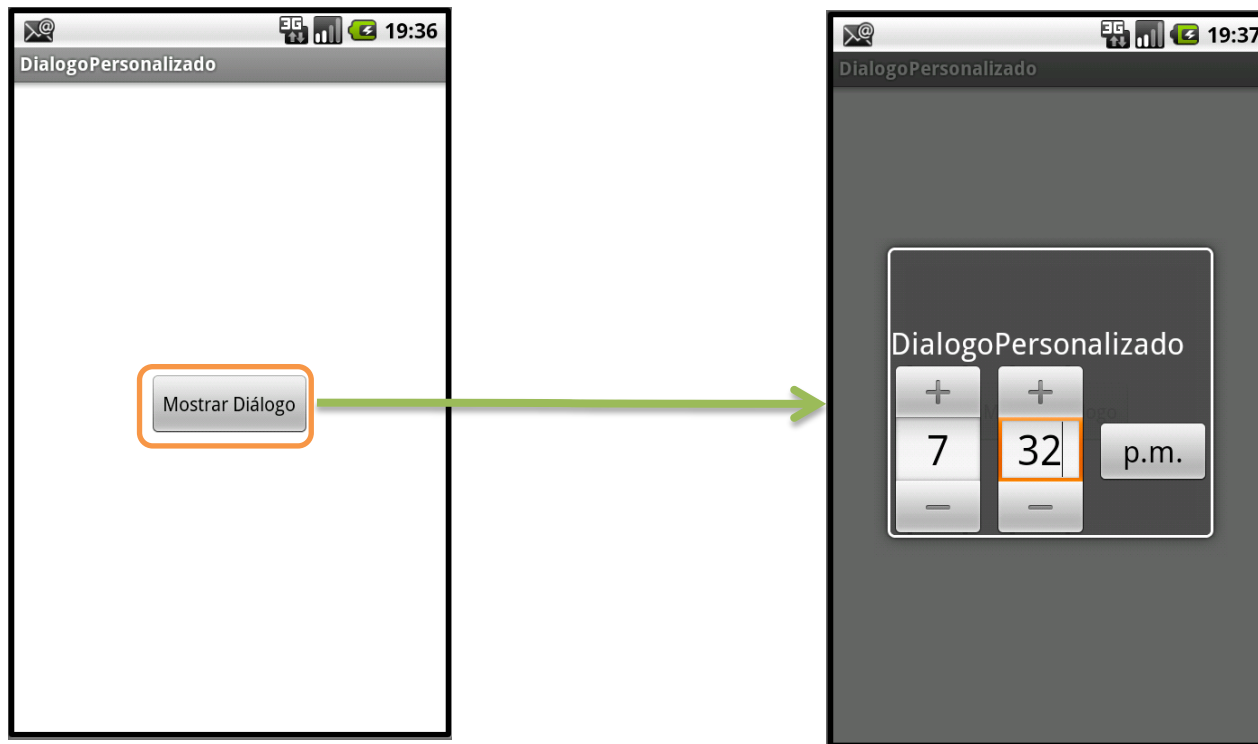
Paso 6. Ejecutamos el proyecto

Ejecutamos el proyecto DialogoPersonalizado:



Paso 6. Ejecutamos el proyecto (cont)

Observamos que al presionar el botón de Mostrar Diálogo, se muestra un diálogo con un layout personalizado. A este dialogo se le puede agregar cualquier tipo de elementos, e incluso interactuar con él.





Conclusión

Con este ejercicio hemos revisado cómo trabajar Diálogos Personalizados.

Este tipo de diálogos nos permite una mayor flexibilidad que cualquiera de los tipos de diálogos estudiados anteriormente. Sin embargo suele utilizarse sólo en ocasiones muy específicas.

Con este ejercicio concluimos el estudio de el manejo de diálogos en Android.



www.globalmentoring.com.mx

Pasión por la tecnología Java

Experiencia y Conocimiento para tu vida