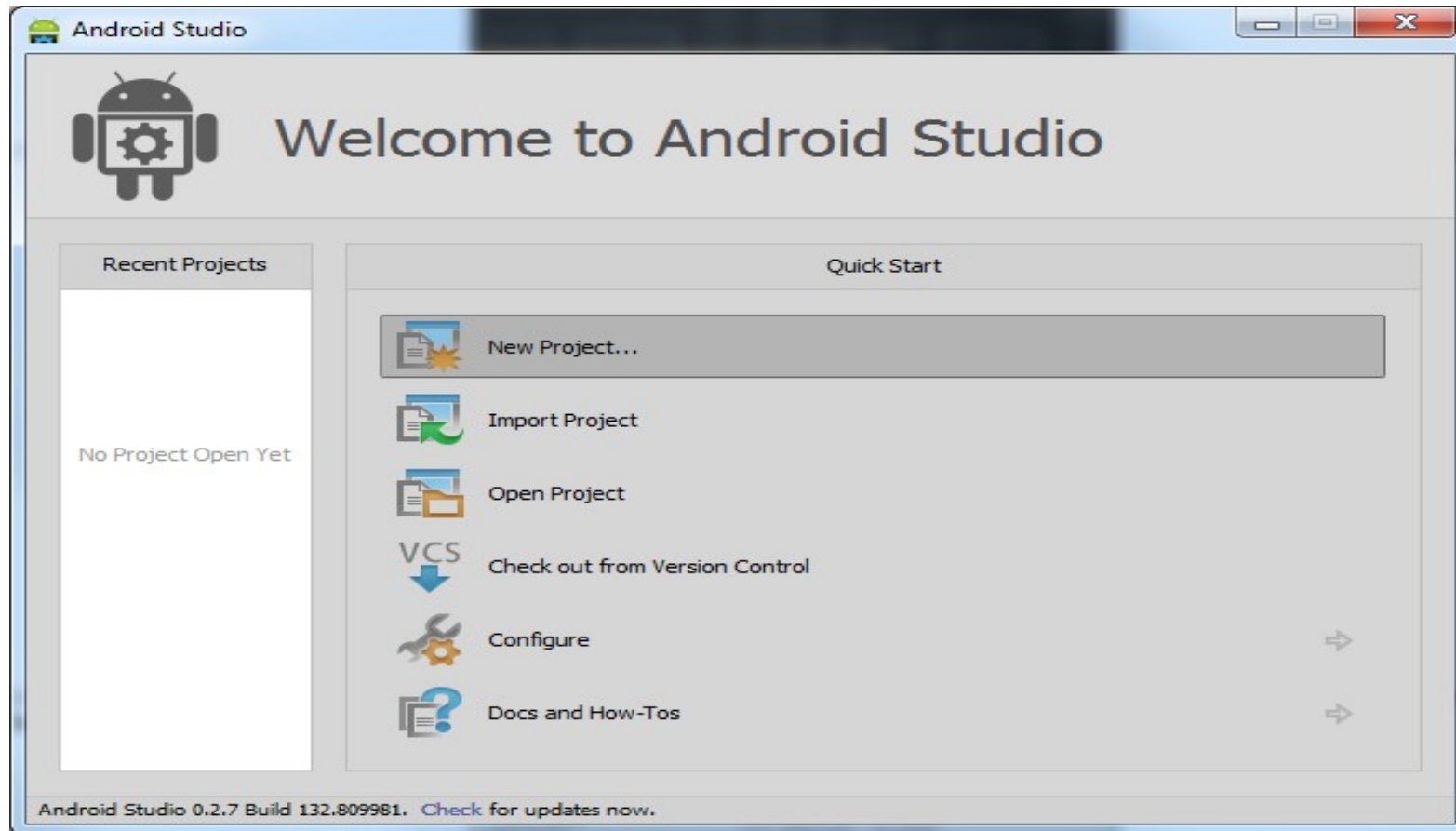
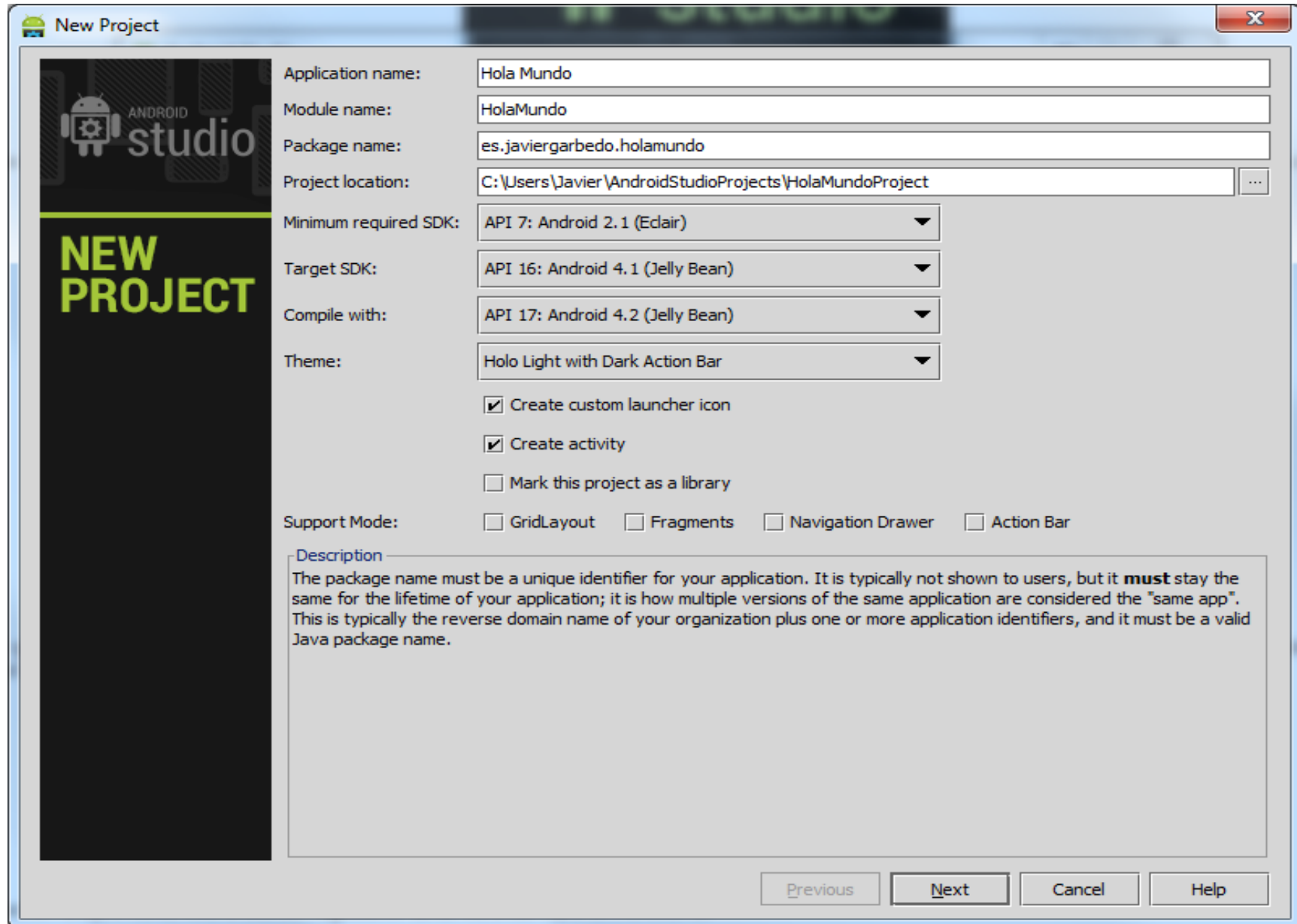


Primer proyecto (Hola Mundo) con Android Studio



Primer proyecto (Hola Mundo) con Android Studio



New Project

Application name: Hola Mundo

Module name: HolaMundo

Package name: es.javiargarbedo.holamundo

Project location: C:\Users\Javier\AndroidStudioProjects\HolaMundoProject

Minimum required SDK: API 7: Android 2.1 (Eclair)

Target SDK: API 16: Android 4.1 (Jelly Bean)

Compile with: API 17: Android 4.2 (Jelly Bean)

Theme: Holo Light with Dark Action Bar

☒ Create custom launcher icon

☒ Create activity

☐ Mark this project as a library

Support Mode: ☐ GridLayout ☐ Fragments ☐ Navigation Drawer ☐ Action Bar

NEW PROJECT

Description

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 must be a valid Java package name.

Previous Next Cancel Help

Primer proyecto: elementos a definir

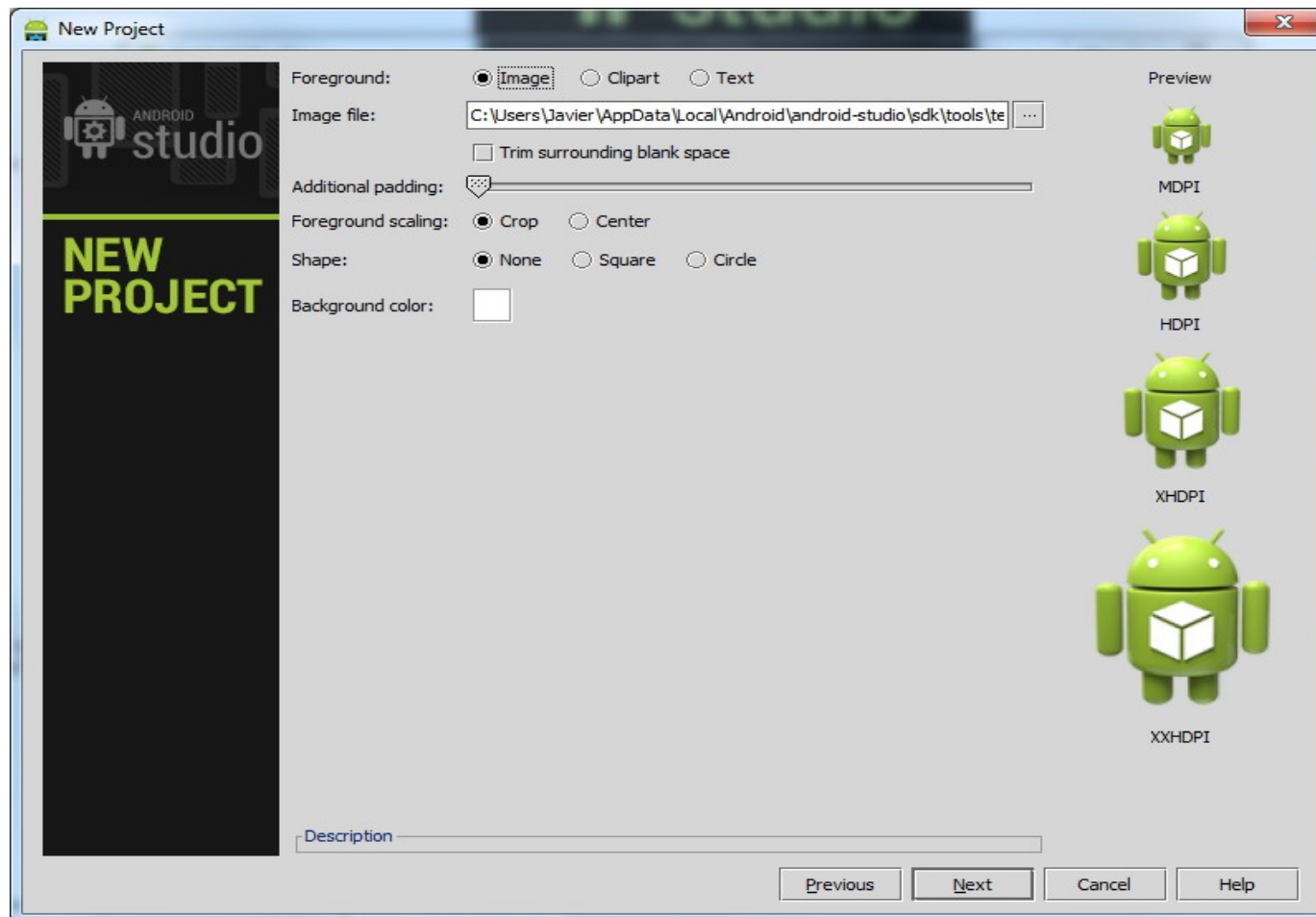
- **Application name:** Es el nombre que deseas dar a tu aplicación, el cual será mostrado en el repositorio de aplicaciones Play Store así como en la lista de Aplicaciones dentro de los Ajustes del dispositivo.
- **Module name:** Este nombre de módulo (proyecto) es utilizado por el entorno de desarrollo (Android Studio). Suele ser igual que el nombre de la aplicación, aunque no podrás usar caracteres especiales como el espacio.
- **Package name:** Nombre del paquete, que debe ser un identificador único de la aplicación. Este nombre no se muestra al usuario, pero se debe mantener el mismo nombre de paquete durante todo el tiempo de vida de la aplicación, ya que es la manera en la que distintas versiones de la aplicación se consideran la misma aplicación. Se suele utilizar el nombre de dominio al revés de la empresa o del autor que realiza la aplicación, seguido de un identificador de la aplicación. Este nombre de paquete debe ser un nombre válido de paquete Java, por lo que no se admitirán determinados caracteres. Por defecto aparece el prefijo com.example, pero en una aplicación real debe ser modificado por otro nombre.
- **Project location:** Carpeta donde se almacenan los archivos del proyecto.
- **Minimum required SDK:** Versión de API más baja que tu aplicación puede soportar. Cuanto más bajo sea el número de versión mínima, más dispositivos podrán utilizar tu aplicación, pero tendrás menos características disponibles.

Primer proyecto: elementos a definir

- **Target SDK:** Versión de API más alta con la que la aplicación puede funcionar correctamente. Este dato informa al sistema que la aplicación con esa versión funciona sin problemas de compatibilidad, y que puede ejecutarse en versiones anteriores, hasta la versión mínima indicada anteriormente.
- **Compile with:** Versión de API que deseas utilizar para compilar el proyecto, seleccionándola entre las versiones instaladas en el SDK. Conviene que sea la versión más moderna.
- **Theme:** Tema (combinaciones de colores) del interfaz de usuario que se tomará como base para la aplicación.
- **Create custom launcher icon:** Debes dejarlo activado si deseas que el asistente de creación del proyecto muestre una ventana que te permite crear un icono personalizado para la aplicación.
- **Create activity:** Te permite crear, también durante este asistente, una pantalla para la aplicación con un diseño prefijado.
- **Mark this project as a library:** Marcar el proyecto como una librería utilizable en otros proyectos, en lugar de una aplicación ejecutable por el usuario.
- **GridLayout, Fragments, Navigation Drawer, Action Bar:** Selecciona los elementos que desees utilizar en la aplicación entre los indicados, para que se utilice en el proyecto la librería de soporte (Support Library) que permite utilizar estos elementos en versiones antiguas que no soportaban esos elementos

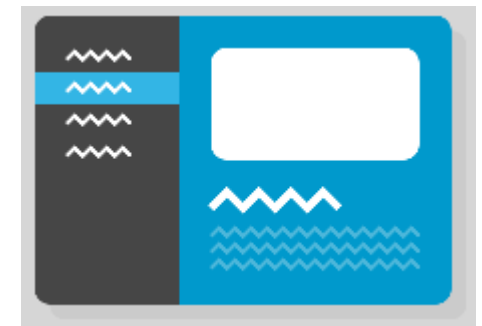
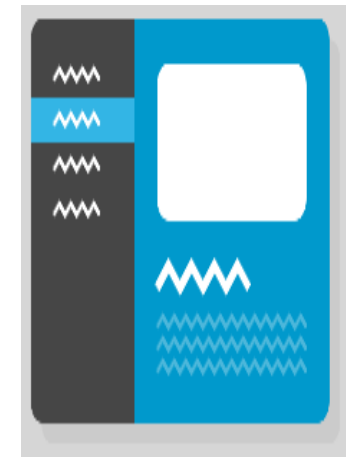
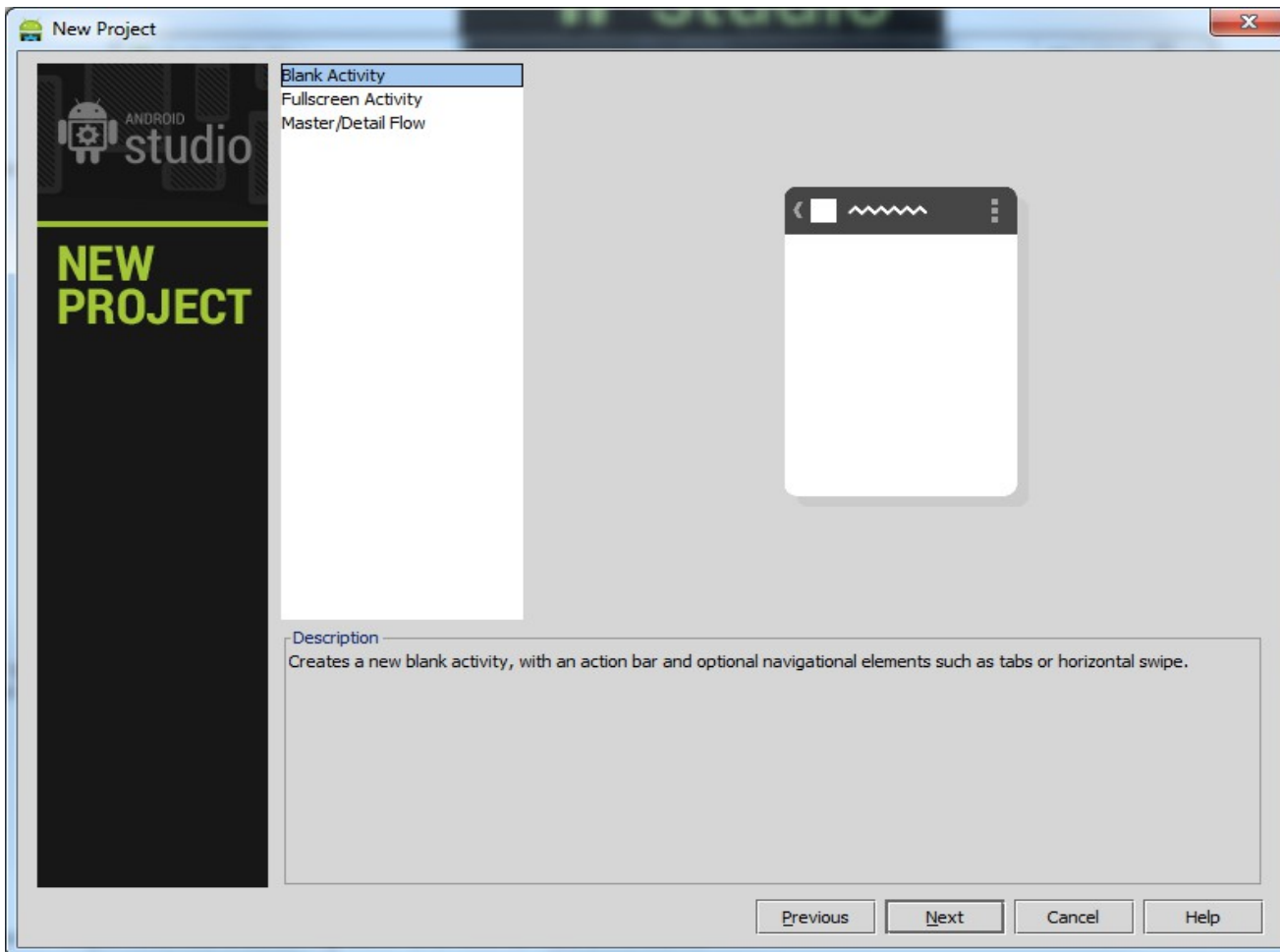
Primer proyecto: elementos a definir

- Si has dejado activada la opción de crear un icono personalizado, la siguiente pantalla será la que te permita diseñarlo. Podrás hacerlo desde una imagen almacenada en un archivo, una imagen prediseñada (Clipart) o sólo texto. Podrás cambiar algunas características del icono, como los márgenes, colores y formas.



Primer proyecto: elementos a definir

- De igual manera, si dejaste activada la opción de crear una Activity (pantalla de la aplicación), la siguiente pantalla te permitirá elegir el estilo deseado para la pantalla. Seleccionando los distintos tipos mostrados en la lista podrás ver en la parte derecha el estilo correspondiente a cada uno. Posteriormente podrás crear otras Activities.



Primer proyecto: elementos a definir

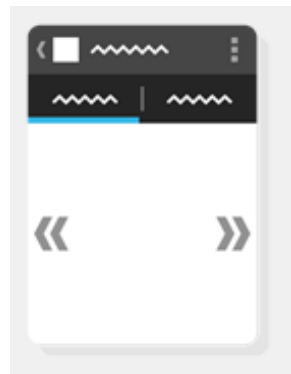
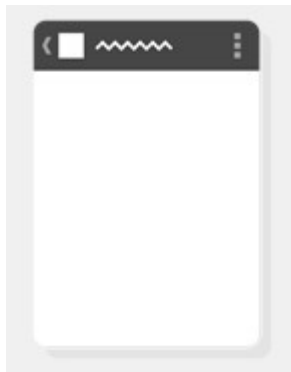
Para finalizar, indica el nombre que le deseas dar a la clase Java asociada a la Activity, el nombre para el archivo que almacenará la estructura de los elementos que contendrá (Layout Name), y el tipo de navegación (Navigation Type) que podrá utilizar el usuario. Ten en cuenta que los tipos de navegación requieren al menos la versión 11 de la API (Android 3.0).

None: Sin ninguna herramienta de navegación inicialmente.

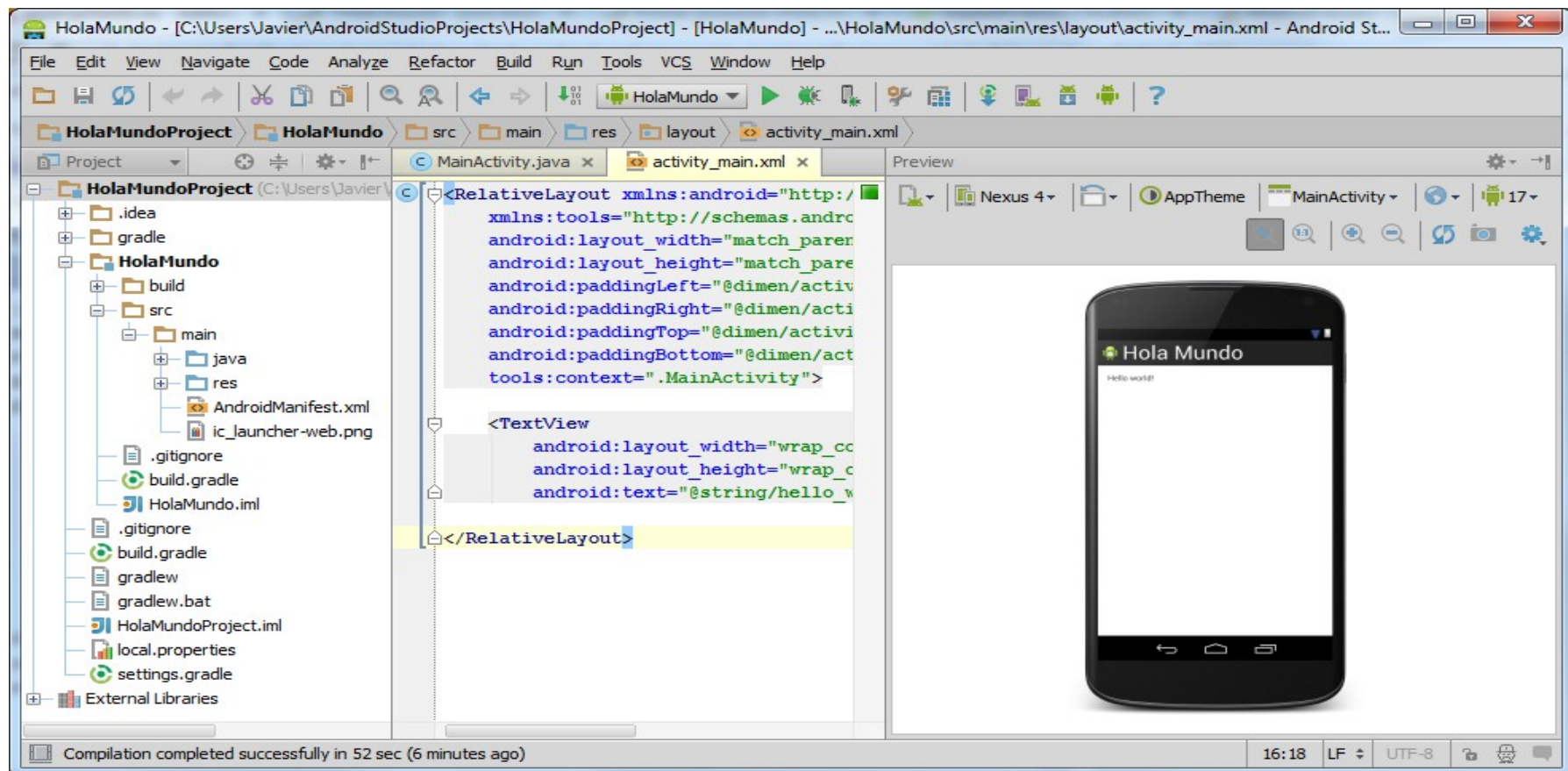
Fixed Tabs + Swipe: Pestañas fijas en la parte superior y desplazamiento lateral.

Scrollable Tabs + Swipe: Desplazamiento lateral con títulos.

Dropdown: Menú desplegable.



- La aplicación es mostrar el mensaje Hello World en una pantalla blanca.
- En la parte central de la ventana podrás ver el código fuente de la Activity que se ha creado, separado en dos archivos seleccionables en las pestañas que hay sobre el código (MainActivity.java y activity_main.xml).
- El lado izquierdo esta r las Tool Windows, siendo una de esas posibles ventanas la de estructura del proyecto (project). Esta sección, puedes hacerla visible con el icono que aparece en la parte inferior izquierda (opción Project), con el menú View > Tool Windows > Project, o con la combinación de teclas Alt+F1.
- La sección derecha de la pantalla está ocupada por la vista previa de la Activity, simulando uno de los dispositivos Android. Para ver la vista previa, debe estar visible el código fuente del Layout de la Activity (archivo XML con la estructura de la pantalla).



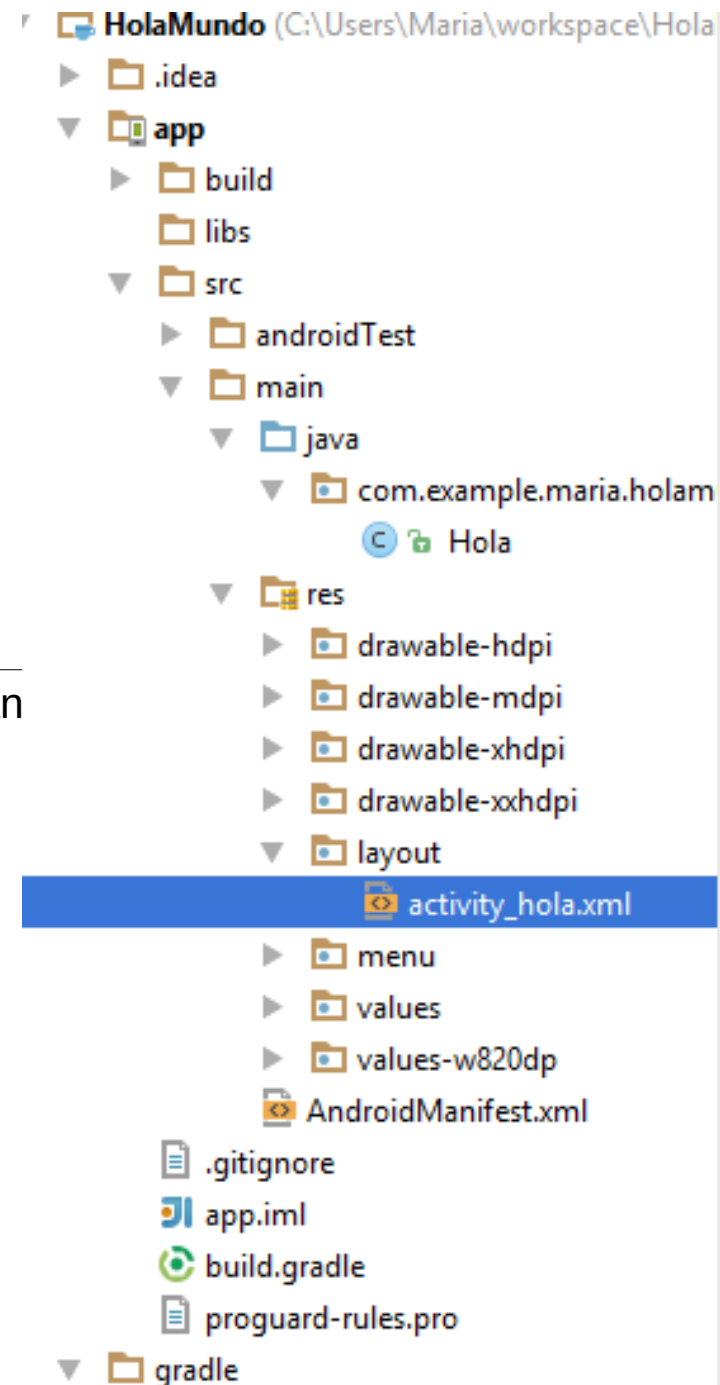
Estructura proyecto(Hola Mundo)

Contenido de activity_hola.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"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".Hola">

    <TextView
        android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```



```
import android.app.Activity;
import android.os.Bundle;
```

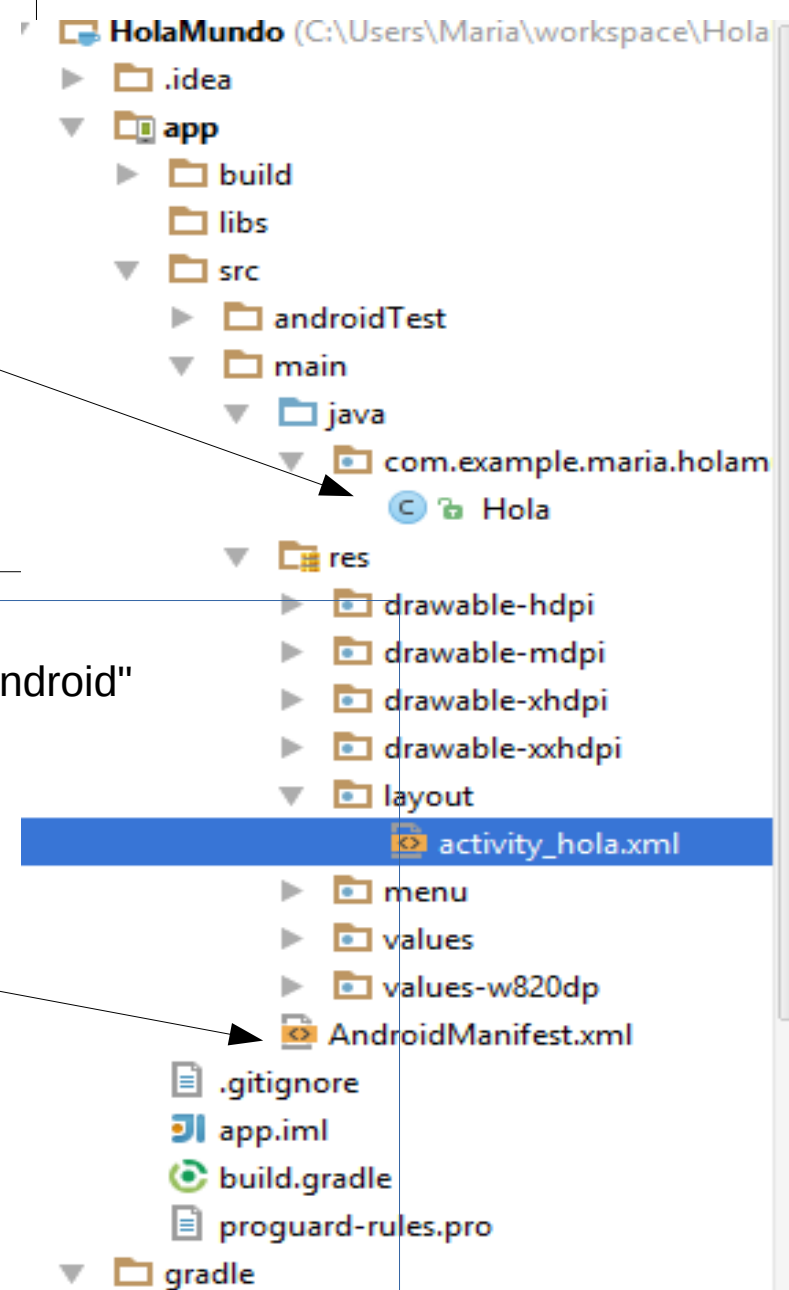
```
public class Hola extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hola);
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.maria.holamundo" >
```

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".Hola"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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



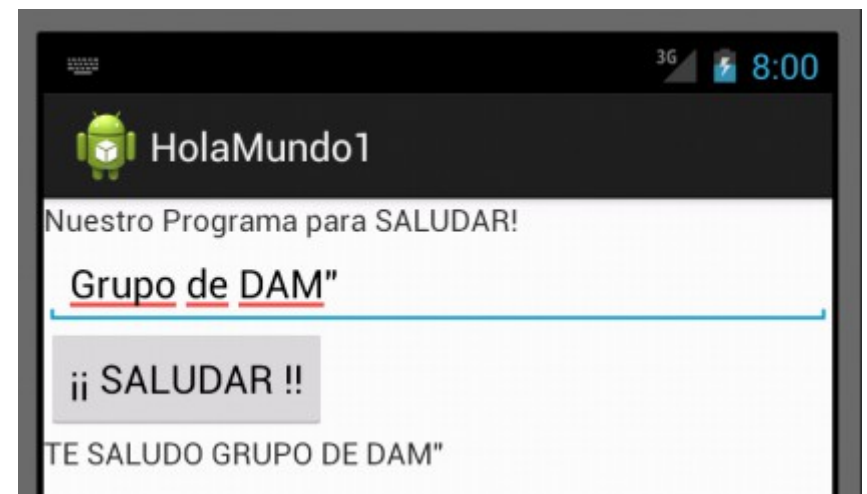
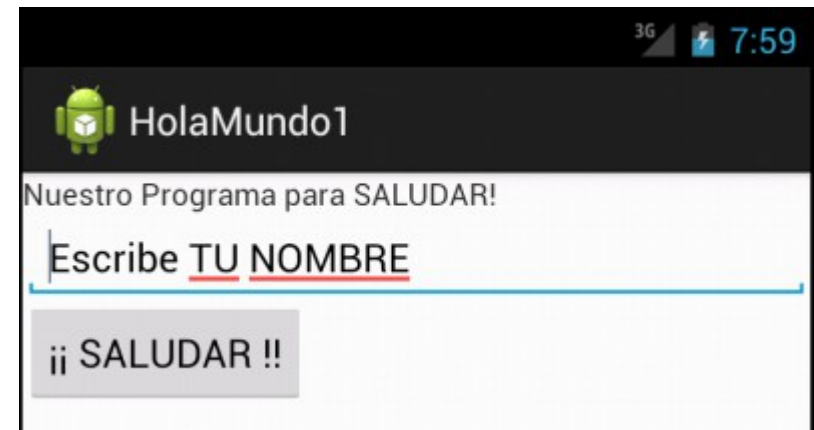
Mejorando: Hola Mundo1

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"

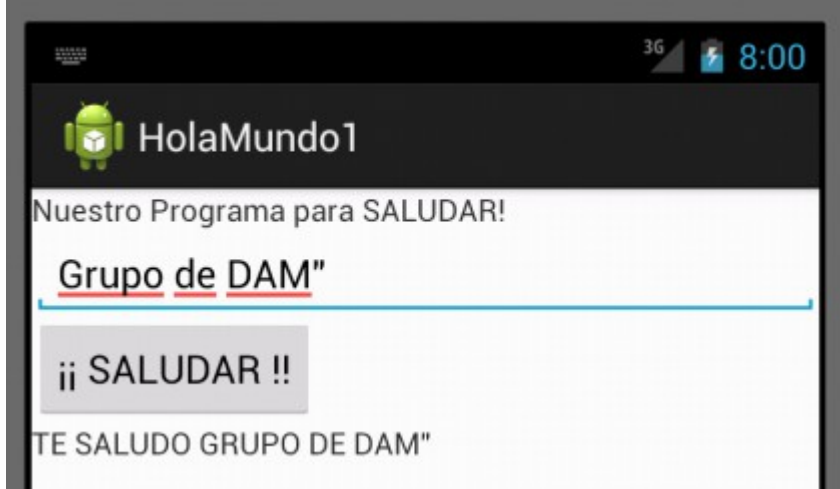
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TextView
        android:text="@string/saludo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <EditText
        android:id="@+id/miTxt"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/escribe" />
    <Button
        android:id="@+id/miBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/pulsar" />
    <TextView
        android:id="@+id/miLbl"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

Añadir las siguientes componentes:
un campo de texto, un botón, una etiqueta
resultado.

El evento sobre el botón escribirá en la
etiqueta resultado el saludo al nombre del
campo de texto.



Mejorando: Hola Mundo1



```
public class Hola extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hola);

        final EditText miTexto= (EditText)findViewById(R.id.miTxt);
        final Button miBoton= (Button)findViewById(R.id.miBtn);
        final TextView elSaludo= (TextView)findViewById(R.id.miLbl);

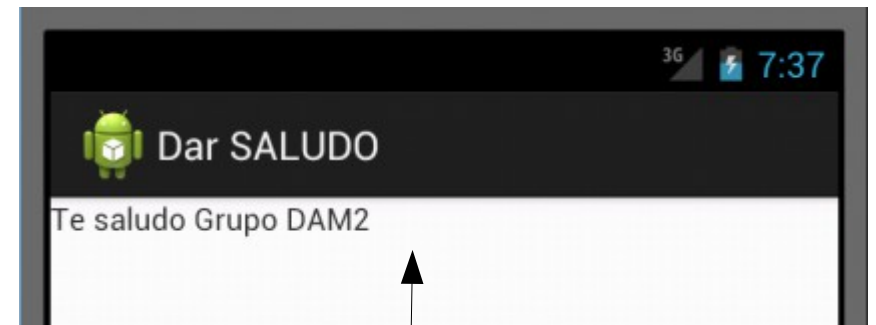
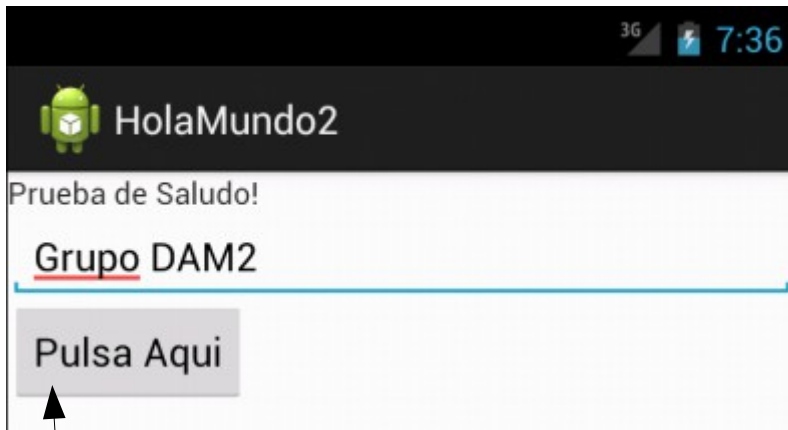
        miBoton.setOnClickListener( new View.OnClickListener() {
            public void onClick(View v){
                String mensajePaso= "Te saludo " + miTexto.getText();
                elSaludo.setText(mensajePaso);
            }
        });
    }
}
```

Mejorando: Hola Mundo2

Programar DOS ACTIVIDADES:

En la primera tenemos: un campo de texto y un botón,

En la segunda solo se incluye la etiqueta resultado



El evento sobre el botón de la primera actividad escribirá en la etiqueta resultado de la segunda actividad el saludo al nombre del campo de texto.

Mejorando: Hola Mundo2

```
public class Hola2 extends Activity {
    public static int COD_RESPUESTA=0;
    TextView elSaludo;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hola);

        final EditText miTexto= (EditText)findViewById(R.id.miTxt);
        final Button miBoton= (Button)findViewById(R.id.miBtn);
        elSaludo= (TextView)findViewById(R.id.miLbl);

        //Borrar el texto inicial del EditText
        miTexto.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                miTexto.setText("");
            }
        });
        miBoton.setOnClickListener( new View.OnClickListener() {
            public void onClick(View v) {
                Intent miIntent= new Intent(Hola2.this, Pantalla2.class);
                Bundle miBundle=new Bundle();
                String mensajePaso= "Te saludo " + miTexto.getText();
                miBundle.putString("TEXTO", mensajePaso);
                miIntent.putExtras(miBundle);
                startActivity(miIntent);
            }
        });
    }
}
```

```
<LinearLayout
```

```
<TextView
```

```
    android:text="@string/saludo"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content" />
```

```
<EditText
```

```
    android:id="@+id/miTxt"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="@string/escribe" />
```

```
<Button
```

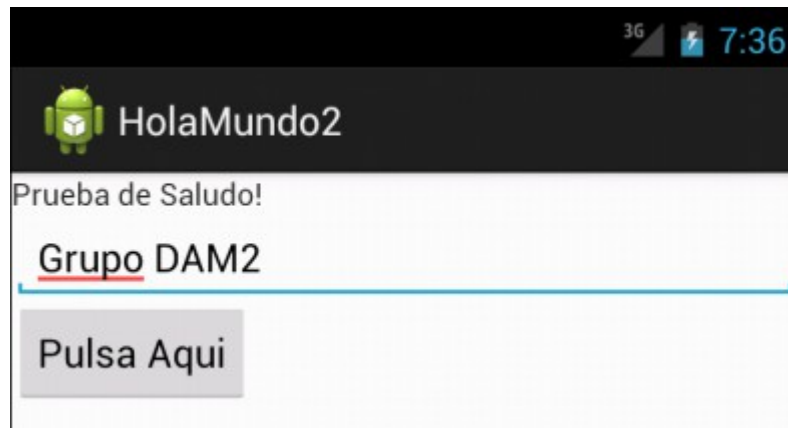
```
    android:id="@+id/miBtn"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="@string/pulsar" />
```

```
</LinearLayout>
```



```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <TextView
        android:id="@+id/miMensaje"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/saludo" />

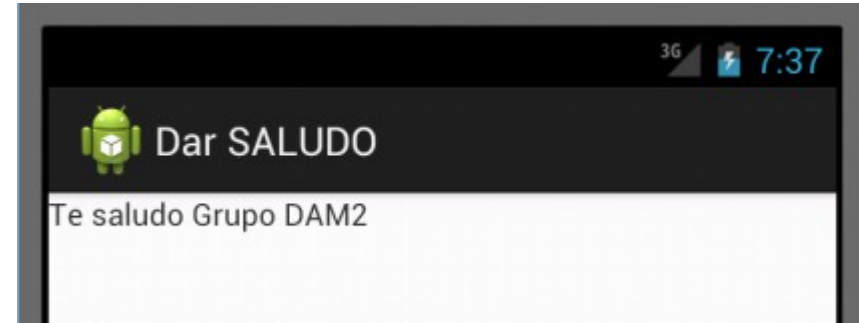
</LinearLayout>
```

```
public class Pantalla2 extends Activity {

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pantalla2);

        final TextView otroSaludo= (TextView)findViewById(R.id.miMensaje);
        Bundle  miBundleRecoger = getIntent().getExtras();
        otroSaludo.setText(miBundleRecoger.getString("TEXTO"));

    }
}
```



Mejorando: Hola Mundo2

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.maria.holamundo2" >
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".Hola2"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Pantalla2"
            android:label="@string/app2_name" >
        </activity>
    </application>
</manifest>
```

Recordar:

- Incluir en el manifest.xml la segunda actividad.
- Definir en string.xml todas las propiedades @text

string.xml

```
<resources>
    <string name="app_name">HolaMundo</string>
    <string name="app2_name">DarSaludo</string>
    <string name="saludo">Prueba de Saludo!</string>
    <string name="escribe">Introduce TU NOMBRE</string>
    <string name="pulsar">Pulsa Aqui</string>
    <string name="action_settings">Settings</string>
</resources>
```


Mejorando más: Hola Mundo3

```
public class Hola3 extends Activity {
    public static int COD_RESPUESTA=0;
    TextView elSaludo;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hola3);

        final EditText miTexto= (EditText)findViewById(R.id.miTxt);
        final Button miBoton= (Button)findViewById(R.id.miBtn);
        elSaludo= (TextView)findViewById(R.id.miLbl);

        //Borrar el texto inicial del EditText
        miTexto.setOnFocusChangeListener(new View.OnFocusChangeListener(){
            public void onFocusChange(View v, boolean b){
                if (b) miTexto.setText("");
            }
        });

        miBoton.setOnClickListener( new View.OnClickListener(){
            public void onClick(View v){
                Intent miIntent= new Intent(Hola3.this, Pantalla2.class);
                Bundle miBundle=new Bundle();
                String mensajePaso= "Te saludo " + miTexto.getText();
                miBundle.putString("TEXTO", mensajePaso);
                miIntent.putExtras(miBundle);
                startActivityForResult(miIntent, COD_RESPUESTA);
            }
        });
    }
}
```

- Importante:

- 1.- Llamar a la actividad Hija con **startActivityForResult(miIntent, COD_RESPUESTA)**

- Importante:

- 2.- Recoger el resultado al sobrescribir el método **public void onActivityResult(int cod_resp, int cod_result, Intent intent)**

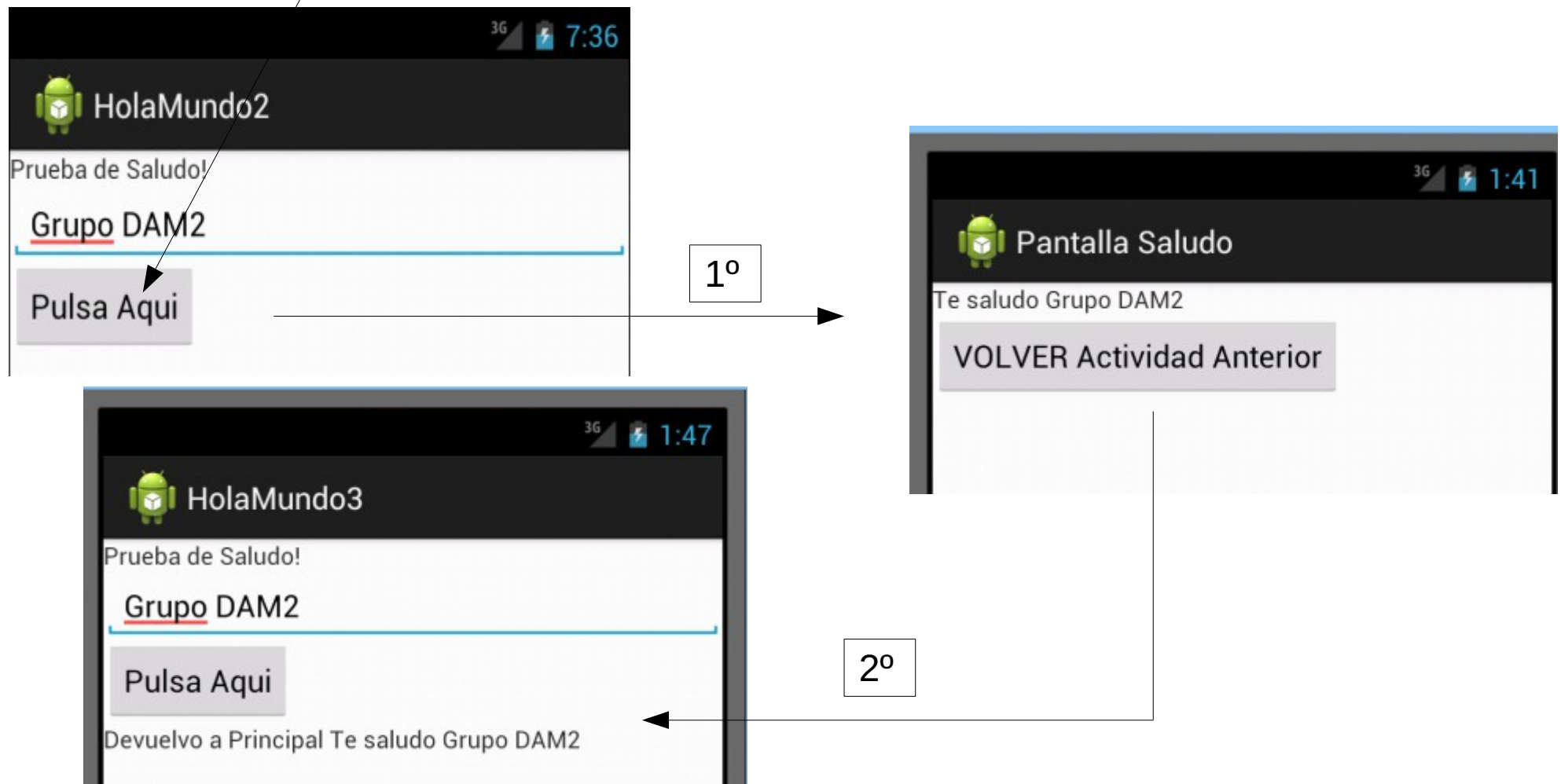
```
public void onActivityResult(int cod_resp, int cod_result, Intent intent){
    if (cod_result== RESULT_OK) {
        Bundle otroBundle = intent.getExtras();
        elSaludo.setText(otroBundle.getString("DEVI DEVI TO"));
```

Mejorando más: Hola Mundo3

Programar DOS ACTIVIDADES con paso de información entre ellas

- La Actividad **Hola3** pasa información a la Actividad **Pantalla2**
- La Actividad **Pantalla2** devuelve valores a la Actividad **Padre (Hola3)**

En la primera tenemos: un campo de texto y un botón que pasa información.
En la segunda incluye un boton de vuelta al Padre devolviendole valores



Tostadas y Diálogos

- **Tostadas**

```
protected void showToast() {  
    Context context = getApplicationContext();  
    CharSequence text = getResources().getString(R.string.noNameMsg);  
    int duration = Toast.LENGTH_SHORT;  
    Toast toast = Toast.makeText(context, text, duration);  
    toast.show();  
}
```

Version reducida: `Toast.makeText(this, "el texto", Toast.LENGTH_SHORT).show();`

- **Diálogos:**

```
protected void showAlert() {  
    CharSequence text = getResources().getString(R.string.noNameMsg);  
    AlertDialog.Builder alert = new AlertDialog.Builder(this);  
    alert.setMessage(text);  
    alert.setPositiveButton(android.R.string.ok, null);  
    alert.show();  
}
```

Version reducida:

```
new AlertDialog.Builder(this).setMessage("el  
texto").setPositiveButton(android.R.string.ok, null).show();
```

Indicar con Mensajes: Donde estamos

```
public class Hola3 extends Activity {
    public static int COD_RESPUESTA=0;
    TextView elSaludo;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hola3);

        final EditText miTexto= (EditText)findViewById(R.id.miTxt);
        final Button miBoton= (Button)findViewById(R.id.miBtn);
        elSaludo= (TextView)findViewById(R.id.miLbl);

        miBoton.setOnClickListener( new View.OnClickListener(){
            public void onClick(View v){
                Intent miIntent= new Intent(Hola3.this, Pantalla2.class);
                Bundle miBundle=new Bundle();
                String mensajePaso= "Te saludo " + miTexto.getText();
                miBundle.putString("TEXT0", mensajePaso);
                miIntent.putExtras(miBundle);
                startActivity( miIntent);
            }
        });
    }
}
```



```
• protected void showToast() {
    Context context = getApplicationContext();
    CharSequence text = getResources().getString(R.string.noNameMsg);
    int duration = Toast.LENGTH_SHORT;
    Toast toast = Toast.makeText(context, text, duration);
    toast.show();
}
```

Pantalla2

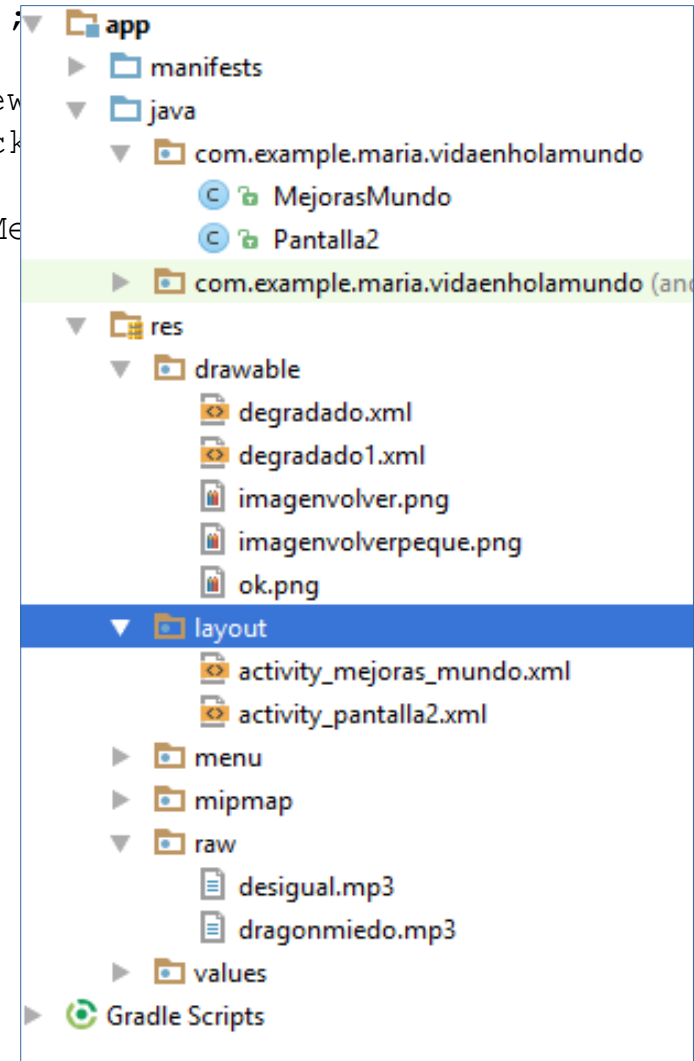
```
public class Pantalla2 extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pantalla2);

        final TextView otroSaludo = (TextView) findViewById(R.id.miMensaje);

        Bundle miBundleRecoger = getIntent().getExtras();
        otroSaludo.setText(miBundleRecoger.getString("TEXTO"));

        final ImageButton miBotonImagenVolver = (ImageButton) findViewById(R.id.imagenvolver);
        miBotonImagenVolver.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Intent miIntent = new Intent(Pantalla2.this, MejorasMundo.class);
                startActivity(miIntent);
            }
        });
    }
}
```



LinearLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/degradado">

    <TextView
        android:id="@+id/miMensaje"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        android:layout_marginStart="50dp"
        android:fontFamily="Arial"
        android:textSize="20dp"
        android:background="@color/colorFondo"
        android:textColor="@color/colorTexto"
        android:text="@string/saludoRecoger" />

    <LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="40dp" >
    <Button
        android:id="@+id/botonVolver"
        android:background="@color/colorBoton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:text="@string/pulsarvolver" />

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnImagen"
        android:layout_marginStart="80dp"
        android:src="@drawable/imagenvolverpeque"/>
    </LinearLayout>

</LinearLayout>
```

Pantalla2



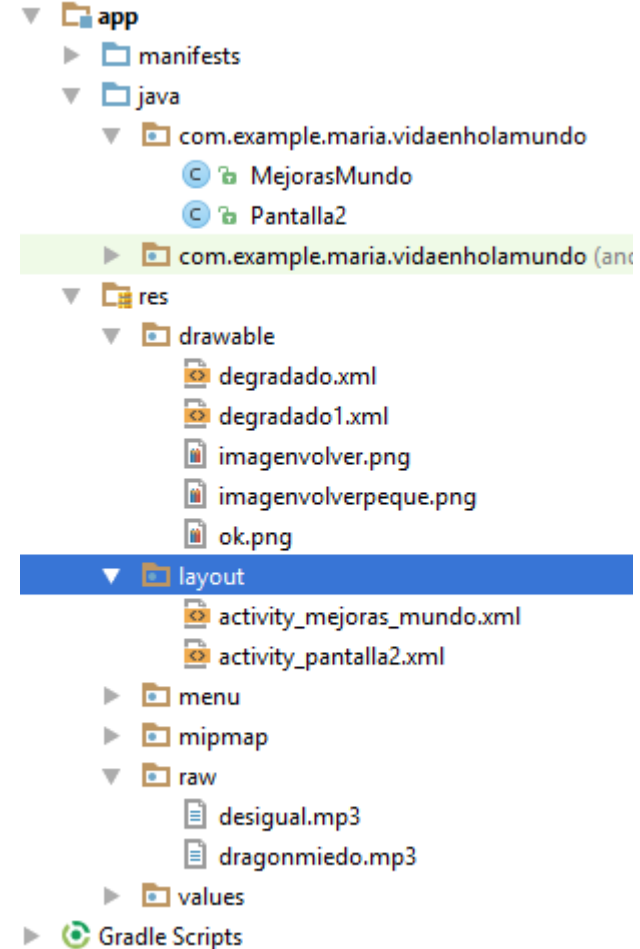
Añadir un recurso Musical

```
public class Hola3 extends Activity {
    public static int COD_RESPUESTA=0;
    TextView elSaludo;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hola3);

        final EditText miTexto= (EditText)findViewById(R.id.miTxt);
        final Button miBoton= (Button)findViewById(R.id.miBtn);
        elSaludo= (TextView)findViewById(R.id.miLbl);

        miBoton.setOnClickListener( new View.OnClickListener(){
            public void onClick(View v){
                Intent miIntent= new Intent(Hola3.this, Pantalla2.class);
                Bundle miBundle=new Bundle();
                String mensajePaso= "Te saludo " + miTexto.getText();
                miBundle.putString("TEXTO", mensajePaso);
                miIntent.putExtras(miBundle);
                startActivity( miIntent);
            }
        });
    }
}
```



```
MediaPlayer miMusica=
MediaPlayer.create(getApplicationContext(), R.raw
.desigual);
miMusica.start();
```

Ciclo de Vida de Una aplicación Android

- Que es un Actividad y Comunicación entre Actividades.
- La pila de llamadas entre Actividades.
- Orden de ejecución de los Métodos que secuencializan el paso entre actividades.
-

Ciclo de Vida: Actividades Android

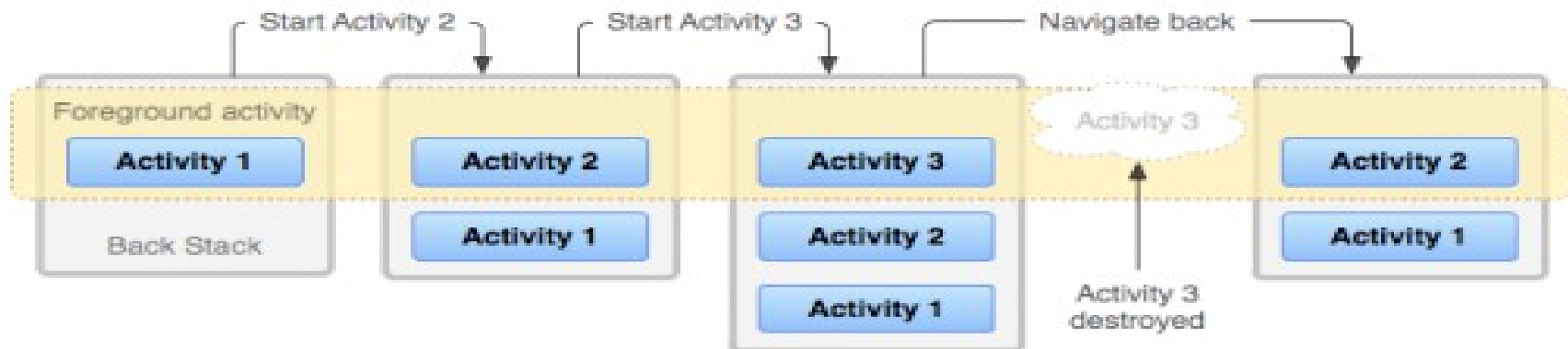
- Una aplicación en Android va a estar formada por un conjunto de elementos de visualización: Actividades.
- Además de varias actividades una aplicación también puede contener servicios.
- Son las actividades las que realmente controlan el ciclo de vida de las aplicaciones, dado que el usuario no cambia de aplicación, sino de actividad.
- El sistema va a mantener una pila con las actividades previamente visualizadas, el usuario puede regresar a la actividad anterior pulsando la tecla "atrás".
- Una aplicación Android corre dentro de su propio proceso Linux. Este proceso es creado para la aplicación y continuará vivo hasta el sistema reclame su memoria para asignársela a otra aplicación.

Ciclo de Vida: Actividades Android

- Una característica importante, y poco usual, de Android es que la destrucción de un proceso no es controlado directamente por la aplicación. Es el sistema quien determina cuando destruir el proceso.
- Si tras eliminar el proceso de una aplicación, el usuario vuelve a ella, se crea de nuevo el proceso, pero se habrá perdido el estado que tenía esta aplicación.
- Es responsabilidad del programador almacenar el estado de las actividades, si queremos que cuando sea reiniciada conserve su estado.

Ciclo de Vida: Actividades Android

- Se puede lanzar nuevas actividades desde otras actividades, de tal forma que la actividad lanzadora es pausada, pero el sistema la mantiene en memoria en una cola denominada back stack.
- Esta cola consiste en una cola tipo LIFO (Last In, First Out), o lo que es lo mismo, la última actividad que fue añadida, será la primera en la cola. Así, cuando el usuario pulse el botón atrás (Back), el sistema nos quitará la actividad actual y nos mostrará justo la anterior en la cola, aunque este comportamiento por defecto puede ser modificado según nuestro interés.

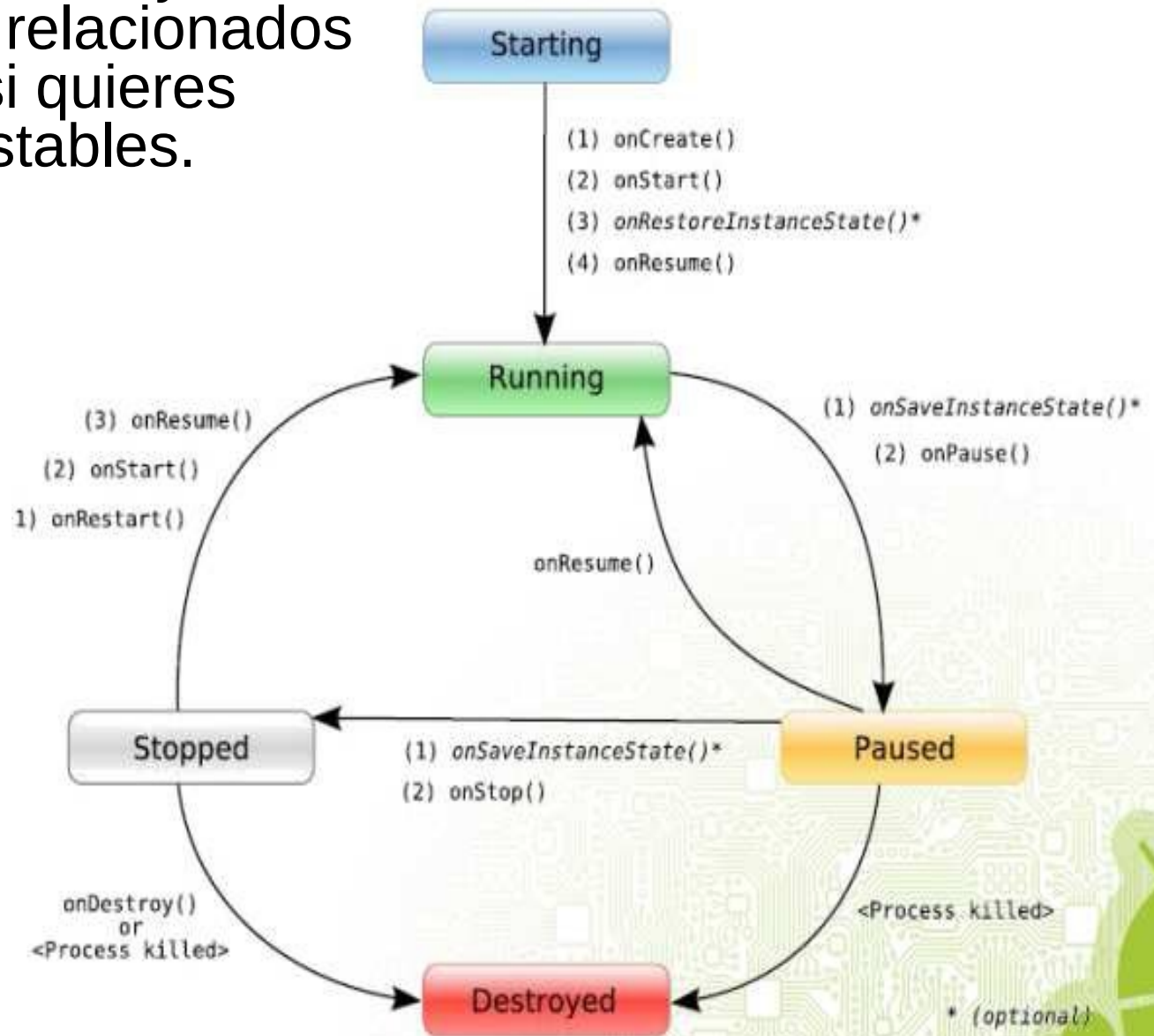


Estados de un Aplicación Android I

- Es necesario comprender y manejar los eventos relacionados con el ciclo de vida si quieres crear aplicaciones estables.

- Estados de un aplicación:

- Activa
- Visible (en pausa)
- Parada
- Destruída:



Estados de un Aplicación Android II

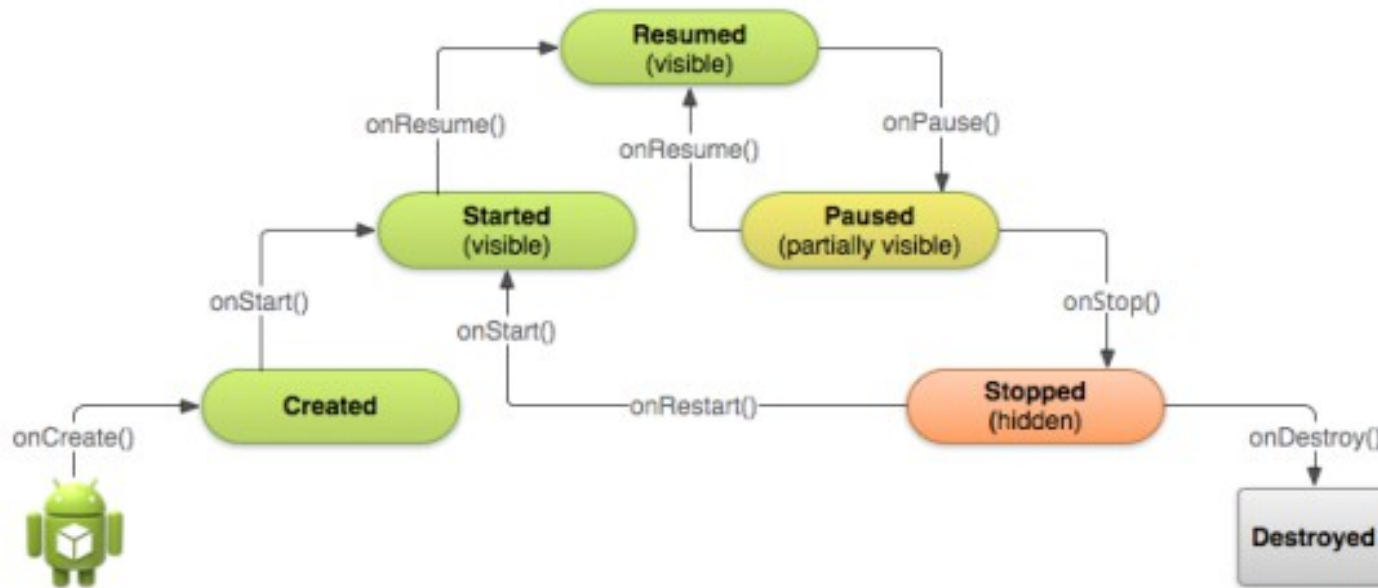
- Estados de un aplicación:

Activa (*Running*): La actividad está encima de la pila, lo que quiere decir que es visible y tiene el foco.

- **Visible** (*Paused*): La actividad es visible pero SIN el foco. Se alcanza este estado cuando pasa a activa otra actividad con alguna parte transparente o que no ocupa toda la pantalla. Cuando una actividad está tapada por completo, pasa a estar parada.
- **Parada** (*Stopped*): Cuando la actividad no es visible, se recomienda guardar el estado de la interfaz de usuario, preferencias, etc.
- **Destruida** (*Destroyed*): La actividad termina al invocarse el método *finish()*, o es matada por el sistema Android, sale de la pila de actividades.

Estados de un Aplicación Android II

- Las actividades pueden ser instanciadas más de una vez. Para crear una actividad, basta con que creamos una clase que herede de la clase Activity. Además de heredar de esta clase, deberemos sobrescribir algunos métodos que pertenecen al ciclo de vida de la actividad.
- Este ciclo de vida consiste en los diferentes estados por los que puede pasar una actividad y los métodos que nos permiten cambiar de un estado a otro.



Métodos

- **onCreate(Bundle):** Se llama en la creación de la actividad. Se utiliza para realizar todo tipo de inicializaciones, como la creación de la interfaz de usuario o la inicialización de estructuras de datos. Puede recibir información de estado de instancia (en una instancia de la clase Bundle), si se reanuda desde una actividad que ha sido destruida y vuelta a crear.
- **onStart():** Nos indica que la actividad está a punto de ser mostrada al usuario.
- **onResume():** Se llama cuando la actividad va a comenzar a interactuar con el usuario. Es un buen lugar para lanzar las animaciones y la música.
- **onPause():** Indica que la actividad está a punto de ser lanzada a segundo plano, porque otra aplicación es lanzada. Es el lugar adecuado para detener animaciones, música o almacenar los datos que estaban en edición.
- **onStop():** La actividad ya no va a ser visible para el usuario. Ojo si hay muy poca memoria, es posible que la actividad se destruya sin llamar a este método.
- **onRestart():** Indica que la actividad va a volver a ser representada después de haber pasado por *onStop()*.
- **onDestroy():** Se llama antes de que la actividad sea totalmente destruida. Por ejemplo, cuando el usuario pulsa el botón <volver> o cuando se llama al método *finish()*. Ojo si hay muy poca memoria, es posible que la actividad se destruya sin llamar a este método.

Ejemplo en HolaMundo2-VidaEnHolaMundos

- Introduce todos estos métodos en la clase "Principal.java"
- Observa la secuencia de mensajes Toast

```
@Override protected void onStart() {  
    super.onStart();  
    Toast.makeText(this, "onStart", Toast.LENGTH_SHORT).show();  
}  
  
@Override protected void onResume() {  
    super.onResume();  
    Toast.makeText(this, "onResume", Toast.LENGTH_SHORT).show();  
}  
  
@Override protected void onPause() {  
    Toast.makeText(this, "onPause", Toast.LENGTH_SHORT).show();  
    super.onPause();  
}  
  
@Override protected void onStop() {  
    super.onStop();  
    Toast.makeText(this, "onStop", Toast.LENGTH_SHORT).show();  
}
```


Ejemplo en HolaMundo2

```
@Override protected void onRestart() {  
    super.onRestart();  
    Toast.makeText(this, "onRestart", Toast.LENGTH_SHORT).show();  
}
```

```
@Override protected void onDestroy() {  
    Toast.makeText(this, "onDestroy", Toast.LENGTH_SHORT).show();  
    super.onDestroy();  
}
```

- Pulsa el botón *CLICK para SALUDAR* y al pasar a la otra pantalla observa la secuencia de Toast.
- Pulsa el botón *CLICK para VOVER* regresa a la actividad. Observa la secuencia de Toast.
- Sal de la actividad y observa la secuencia de Toast.

Bibliografía:

- <http://www.androidcurso.com/index.php/tutoriales-android/37-unidad-6-multimedia-y-ciclo-de-vida/158-ciclo-de-vida-de-una-aplicacion>
- <http://www.jtech.ua.es/cursos/apuntes/moviles/daa2013/index.html>
- http://www.elandroidelibre.com/2014/11/El%20Androide%20Libre_files/aprende-android-en-20-conceptos-conceptos-3-y-4.htm