



La Clase Intent

CONCEPTOS.

Un **Intent** es una estructura pasiva de datos que mantiene una descripción abstracta para realizar una acción. Se puede utilizar `startActivity` para iniciar una **Activity**, `broadcastIntent` para enviarla a cualquiera de los componentes **BroadcastReceiver**, y `startService (Intent)` o `bindService (Intent, ServiceConnection, int)` para comunicarse con un **Service** en segundo plano.

Un **Intent** explícito contiene un nombre de componente; un **Intent** implícito no tiene un nombre de componente y depende de la acción y de los datos.

Un **Intent** proporciona las herramientas para realizar la unión final del código en tiempo de ejecución en diferentes aplicaciones. Su aplicación más importante es el lanzamiento de actividades y puede visualizarse como el pegamento entre las actividades. Una aplicación está formada por un conjunto de actividades. Cuando una actividad interactúa con otras actividades también puede intercambiar datos.

La información principal que utiliza un **Intent** es:

action La acción general por realizar, por ejemplo `ACTION_VIEW`, `ACTION_EDIT`, `ACTION_MAIN`, etc.

data Los datos por operar, por ejemplo un registro en una base de datos, expresado como **Uri**.

El nombre de la acción es un **String**, que puede tener como prefijo el nombre del paquete. La sección de datos de un **Intent** realmente no son datos, sino referencias a los datos y están representados por la cadena que representa a una **Uri**. Una **Uri** de un **Intent** puede contener argumentos, en forma similar a las **Urls** de **Web**.

Constructores públicos:

<code>Intent()</code>	Crea un Intent vacío.
<code>Intent(Intent o)</code>	Copia un constructor.
<code>Intent(String action)</code>	Crea un Intent con una determinada acción.
<code>Intent(String action, Uri uri)</code>	Crea un Intent con una acción y una determinada url de datos.
<code>Intent(Context packageContext, Class<?> cls)</code>	Crea un Intent para un componente específico.
<code>Intent(String action, Uri uri, Context packageContext, Class<?> cls)</code>	Crea un Intent para un componente específico con una acción específica y datos.

Un **Intent** admite, además de acciones y datos, un atributo adicional llamado extras. Este tipo de dato tiene la forma clave-valor, donde el nombre de la clave inicia con el nombre del paquete y el valor puede ser de un tipo primitivo, o un objeto arbitrario si es que se implanta la interface `android.os.Parcelable`. La información extra se representa con la clase `android.os.Bundle`.

Las actividades se clasifican y se buscan por categorías. Cuando el sistema se inicia, busca las actividades en la categoría `CATEGORY_LAUNCHER` de `android.intent.category.LAUNCHER`.

Las categorías son:

<code>CATEGORY_DEFAULT</code>	Si se declara una actividad <code>DEFAULT</code> , se invoca con Intent implícito, en caso contrario se invoca como Intent explícito.
<code>CATEGORY_BROWSABLE</code>	La actividad se invoca con seguridad por el navegador y muestra los datos referenciados por un enlace.
<code>CATEGORY_GADGET</code>	La actividad se puede anidar dentro de otra actividad que pueda poseer gadgets.
<code>CATEGORY_HOME</code>	La actividad de este tipo es la pantalla principal y se muestra al iniciar el móvil o al digitar el botón Home . Si existe más de una, se pregunta al usuario cuál puede elegir.



Cuando se utiliza un **Intent** para iniciar una actividad se especifica qué tipo de actividad se desea especificando la categoría:

```
Intent i = new Intent(Intent.ACTION_MAIN, null);
i.addCategory(Intent.CATEGORY_LAUNCHER);
PackageManager pm = getPackageManager();
List<resolveinfo> l = pm.queryIntentActivities(i, 0);
</resolveinfo>
```

Las categorías se declaran en el **AndroidManifest.xml** en la forma siguiente:

```
<activity android:name=".PrincipalActivity" android:label="@string/app_name">
    <intent -filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent>
</activity>
```

DESARROLLO

EJEMPLO 1.

Paso 1. Crear un nuevo proyecto. En la actividad principal **MainActivity.java** predeterminado, capturar el siguiente código:

```
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.view.View.*;
import android.content.Intent;
import android.widget.*;

public class MainActivity extends Activity{
    EditText    jet;
    Button      jbn;
    Bundle      bdl;
    Intent      itn;
    public void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_main);
        jet = (EditText) findViewById(R.id.xet);
        jbn = (Button) findViewById(R.id.xbn);
        jbn.setOnClickListener(new OnClickListener() {
            public void onClick(View arg0) {
                itn = new Intent(MainActivity.this, SegundaActivity.class);
                bdl = new Bundle();
                bdl.putString("NOMBRE", jet.getText().toString());
                itn.putExtras(bdl);
                startActivity(itn);
            }
        });
    }
}
```

Paso 2. En el archivo **activity_main.xml** predeterminado, capturar el siguiente código:

```
<?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:text="@string/escribir"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
```



```

<EditText
    android:id="@+id/xet"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/xbn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/aceptar" />
</LinearLayout>

```

Paso 3. En la carpeta `java/com.example.escom.mipaquete` del repositorio izquierdo, crear una segunda actividad `SegundaActivity.java` y capturar el siguiente código:

```

import android.os.Bundle;
import android.app.Activity;
import android.widget.*;
public class SegundaActivity extends Activity{
    EditText    jet;
    Bundle      bdl;
    public void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_segunda);
        jet = (EditText) findViewById(R.id.xet1);
        bdl = getIntent().getExtras();
        jet.append("Hola " + bdl.getString("NOMBRE"));
    }
}

```

Paso 4. En la carpeta `res/layout` del repositorio izquierdo, crear el archivo `activity_segunda.xml` y capturar el siguiente código:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:id="@+id/xtv1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Segunda actividad:" />
    <EditText
        android:id="@+id/xet1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>

```

Paso 5. En la carpeta `manifests` del repositorio izquierdo, abrir el archivo `AndroidManifest.xml`. Insertar la siguiente etiqueta:

```

<activity
    android:name=".SegundaActivity">
</activity>

```

Entre las etiquetas `<application></application>` y después del primer par de etiquetas `<activity></activity>`, como se indica enseguida, con letras negritas:

```

<?xml version="1.0" encoding="utf-8"?>

```

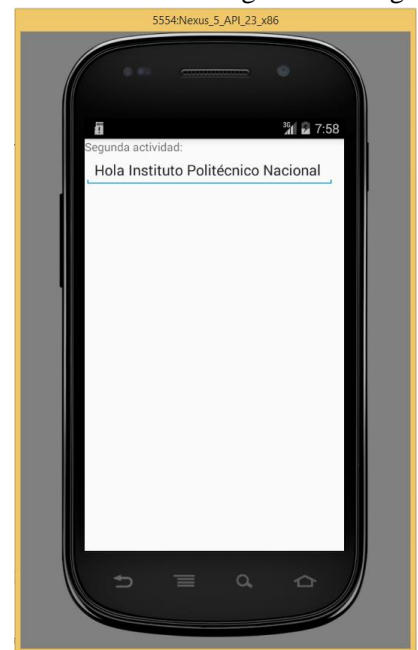
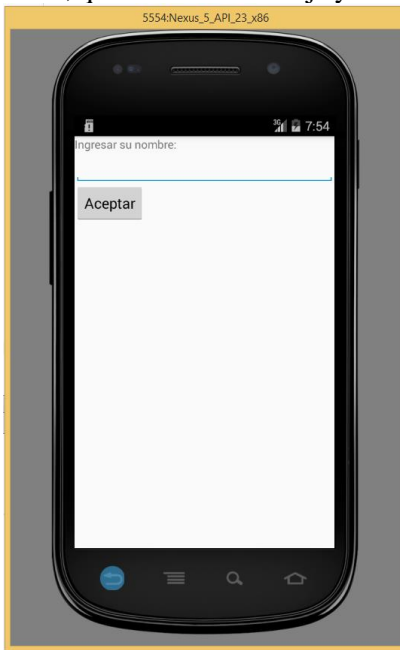


```
<manifest
:
package="com.example.escom.intentos">
<application
:
:
>
<activity
android:name=".MainActivity"
:
:
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>

<activity
android:name=".SegundaActivity">
</activity>

</application>
</manifest>
```

Paso 6. Por último, ejecutar la aplicación. Ingresar el nombre y enseguida digitar el botón **Aceptar**. Se invoca a la segunda actividad, que recibe el mensaje y lo muestra en el campo de texto de su plantilla, como se indica en las siguientes imágenes:



EJERCICIO 1.

Diseñar una aplicación que solicite su nombre y apellido en una primera actividad. Al digitar un botón, se invoca a una segunda actividad que muestra las dos cadenas en su plantilla. Utilizar las instancias `bundle` e `intent` necesarios para cada uno de los datos; observar las letras en negritas.

Para enviarlos, en la primera actividad:

```
bundle.putString("DATO_X", editText.getText().toString());
intent.putExtras(bundle);
```

Para recibirlos en la segunda actividad:



```
bundle = getIntent().getExtras();  
editText.append("Hola " + bundle.getString("DATO_X"));
```

EJERCICIO 2.

Diseñar una aplicación que resuelva una ecuación de segundo grado $ax^2 + bx + c = 0$.

En una primera actividad, el usuario ingresa los valores para a, b, c, y con un botón se envían a una segunda actividad. La segunda actividad calcula las raíces de la solución y los muestra en su plantilla.