



PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

2º DAM



NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

INICIAR OTRA ACTIVITY

Para navegar a través de las distintas pantallas que puede tener una aplicación, es necesario iniciar una nueva Activity en cada ocasión. Es decir, por cada cambio de pantalla dentro de la aplicación, se debe lanzar o iniciar una nueva Activity, de lo contrario el cambio de pantalla no se realizará.

Para realizar esto, es necesario crear un objeto de tipo **Intent**.

Una Intent es un objeto de acción que podemos usar para solicitar una acción de otro componente de la aplicación.

El Intent representa la “intención de hacer algo” de una app. Se pueden usar las intents para varias tareas, pero en esta práctica el intent **iniciará otra activity**.



NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

CREAR UNA INTENT

Para crear una **intent**, basta con crear un objeto:

```
Intent i = new Intent();
```

Posteriormente le asignamos dos parámetros:

- El primer parámetro es el Activity donde nos encontramos
- El segundo parámetro es el Activity a donde queremos ir

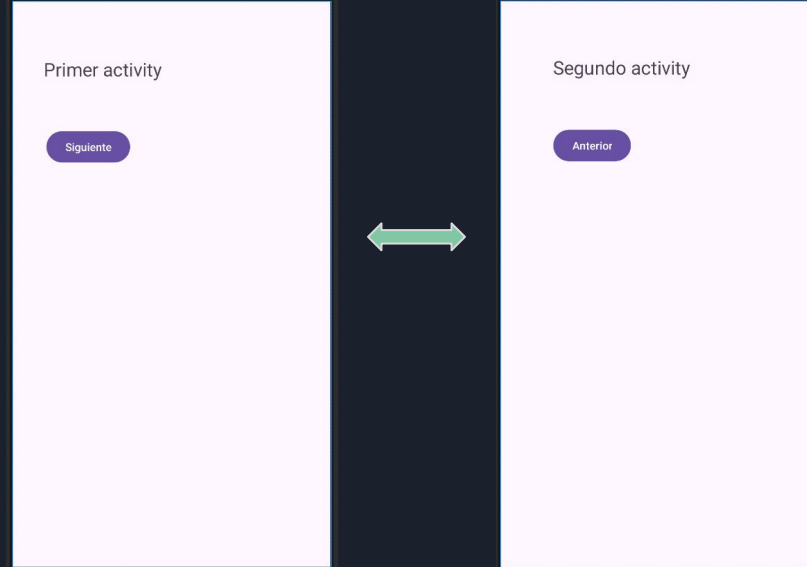
```
Intent i = new Intent(this, Activity2.class);
```

Finalmente, hay que hacer uso del método **startActivity**, colocando dentro de sus paréntesis el nombre del objeto, el cual nos permitirá realizar el cambio de pantalla.

```
startActivity(i);
```

NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

EJEMPLO A CREAR





NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

EJEMPLO A CREAR

- Creamos un nuevo proyecto llamado CambioDeActivities
- Seleccionamos una activity vacía llamada MainActivity
- Agregamos un botón y un TextView (como en el ejemplo)
- Con blueprint creamos las distancias
- Con `strings.xml` modificamos el texto de cada uno de nuestros controles
- Creamos un segundo activity dando click derecho en app > New > Activity > EmptyViewsActivity y la llamamos “SegundoActivity”
- Comprobamos cómo se nos ha creado una parte lógica y una gráfica.



NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

EJEMPLO A CREAR

- Continuamos agregando un botón y un TextView (como anteriormente)
- Con blueprint creamos las distancias
- Con strings.xml modificamos el texto de cada uno de nuestros controles
- Creamos un nuevo método en **MainActivity**:

```
public void siguiente(View view){  
    Intent siguiente = new Intent(this, SegundoActivity.class);  
    startActivity(siguiente);  
}
```

- Agregamos el método al botón con onClick



NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

EJEMPLO A CREAR

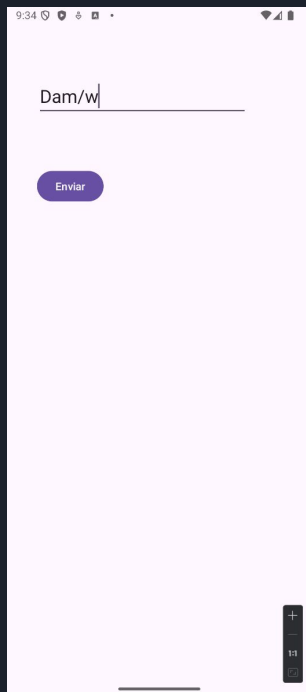
- Hacemos lo mismo en nuestra SegundoActivity.java

```
public void anterior(View view){  
    Intent anterior = new Intent(this, MainActivity.class);  
    startActivity(anterior);  
}
```

- Añadimos en onClick

NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

PROBEMOS AHORA A PASAR DATOS O PARÁMETROS ENTRE ACTIVITIES





NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

EJEMPLO A CREAR

- Creamos un nuevo proyecto llamado PasoParametros
- Seleccionamos una activity vacía llamada MainActivity
- Agregamos un botón y un Plain Text (como en el ejemplo)
- Con blueprint creamos las distancias
- Con strings.xml modificamos el texto de cada uno de nuestros controles
- Creamos un segundo activity dando click derecho en app > New > Activity > EmptyActivity y la llamamos “SegundoActivity”
- Comprobamos cómo se nos ha creado una parte lógica y una gráfica.



NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

EJEMPLO A CREAR

En Android, los métodos `putExtra` y `getStringExtra` son utilizados para enviar y recibir datos entre Activities mediante `Intents`.

- `putExtra`: se utiliza para agregar datos a un `Intent` antes de que se inicie una nueva activity. Permite enviar diferentes tipos de datos (strings, enteros, booleanos, etc.) desde una actividad a otra.
- `getStringExtra`: se utiliza para recuperar el dato enviado con `putExtra` en la nueva activity. Devuelve el valor asociado con la clave que especificamos, en este caso, una cadena (`String`).



NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

EJEMPLO A CREAR

- Continuamos agregando un botón y un TextView (como anteriormente)
- Con blueprint creamos las distancias
- Con strings.xml modificamos el texto de cada uno de nuestros controles
- Creamos un nuevo método en MainActivity:

```
public class MainActivity extends AppCompatActivity {  
  
    private EditText et1;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        ...  
  
        et1 = findViewById(R.id.et1);  
    }  
  
    //Método para el botón enviar  
    public void enviar(View view){  
        Intent i = new Intent(this, SegundoActivity.class);  
        i.putExtra("dato", et1.getText().toString());  
        startActivity(i);  
    }  
}
```

- Agregamos el método al botón con onClick



NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

EJEMPLO A CREAR

- Ahora nos vamos a nuestra segunda activity
- Conectamos el textView y creamos la relación lógica/gráfica
- Alojamos/establecemos el dato de la anterior activity en nuestro textView

```
public class SegundoActivity extends AppCompatActivity {  
  
    private TextView tv1;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.SegundoActivity);  
  
        tv1 = findViewById(R.id.tv1);  
  
        String nombre = getIntent().getStringExtra("dato");  
        tv1.setText("Hola " + nombre);  
    }  
}
```



NAVEGACIÓN ENTRE DISTINTAS ACTIVITY

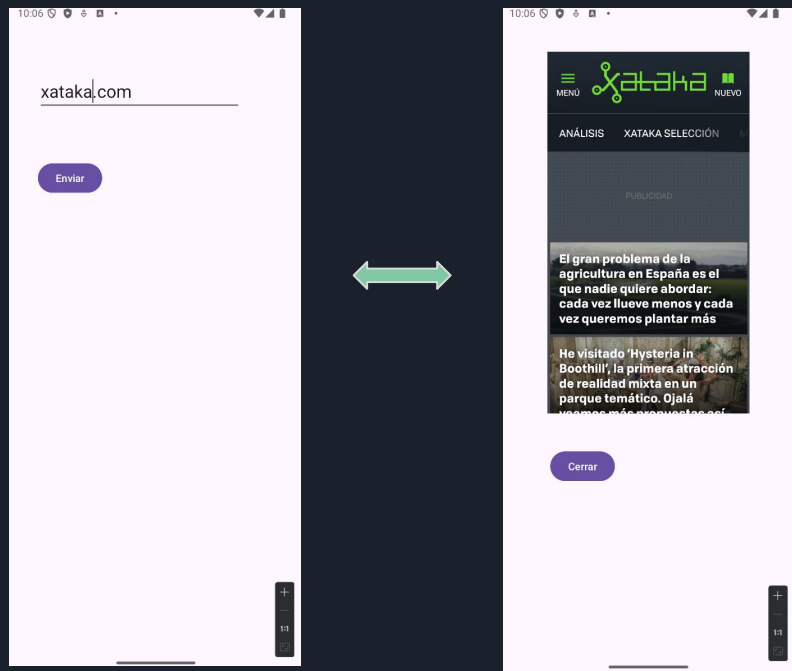
EJEMPLO A CREAR

- Creamos el método para regresar

```
//Método para el botón Regresar  
public void regresar(View view){  
    Intent i = new Intent(this, MainActivity.class);  
    startActivity(i);  
}
```

NAVEGACIÓN ENTRE DISTINTAS ACTIVITY | WEB VIEW

EJEMPLO A CREAR





NAVEGACIÓN ENTRE DISTINTAS ACTIVITY | WEB VIEW

- Crearemos un nuevo proyecto, con una nueva Activity vacía
- Añadiremos un **plain text** y un **button**
- Añadimos un id `txt_web` al plain text
- Al botón no le añadiremos id
- Desde blueprint estableceremos las distancias entre controles
- Desde el archivo strings.xml añadiremos las referencias necesarias (y las que tenemos previstas):

```
<resources>
```

```
<string name="app_name">WebView</string>
```

```
<string name="txt_web">Ingresa el sitio web</string>
```

```
<string name="botonIR">Ir al sitio</string>
```

```
<string name="botonCerrar">Cerrar</string>
```

```
</resources>
```

NAVEGACIÓN ENTRE DISTINTAS ACTIVITY | WEB VIEW

- Referenciamos esos strings desde la parte gráfica
- Ahora creamos nuestro segundo Activity App>New>Activity>Empty Views Activity
- Lo llamamos ActivityWeb
- Añadiremos el control **WebView** y un **button**
- Modificamos los tamaños y las distancias
- Añadimos las referencias de los strings
- El webView necesita la id ww1
- Creamos los objetos en MainActivity:

```
public class MainActivity extends AppCompatActivity {  
  
    private EditText et1;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        et1 = findViewById(R.id.txt_web);  
    }  
}
```




NAVEGACIÓN ENTRE DISTINTAS ACTIVITY | WEB VIEW

- Creamos el método del botón “Ir al sitio”:

//Método botón ir

```
public void navegar(View view){  
    Intent i = new Intent(this, ActivityWeb.class);  
    i.putExtra("sitioWeb", et1.getText().toString());  
    startActivity(i);  
}
```

NAVEGACIÓN ENTRE DISTINTAS ACTIVITY | WEB VIEW

- Creamos los objetos de nuestra segunda Activity
- WebView no lo pondremos de tipo private (será accesible desde el mismo paquete)
- Y creamos el método Cerrar

```
public class ActivityWeb extends AppCompatActivity {  
  
    WebView wv1;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_web);  
  
        wv1 = findViewById(R.id.wv1);  
  
        String uRL = getIntent().getStringExtra("sitioWeb");  
        wv1.setWebViewClient(new WebViewClient());  
        wv1.loadUrl("https://" + uRL);  
    }  
  
    public void cerrar(View view){  
        finish();  
    }  
}
```

NAVEGACIÓN ENTRE DISTINTAS ACTIVITY | WEB VIEW

- Ahora deberemos darle permisos a nuestra App para que pueda navegar por internet:
- En el archivo **AndroidManifest.xml** añadiremos:

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

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

```
  <application ...
```





NAVEGACIÓN ENTRE DISTINTAS ACTIVITY | WEB VIEW

Para finalizar, podemos buscar información sobre cómo hacer que nuestra **URL** se abra en un **navegador externo** en lugar de en otra activity.