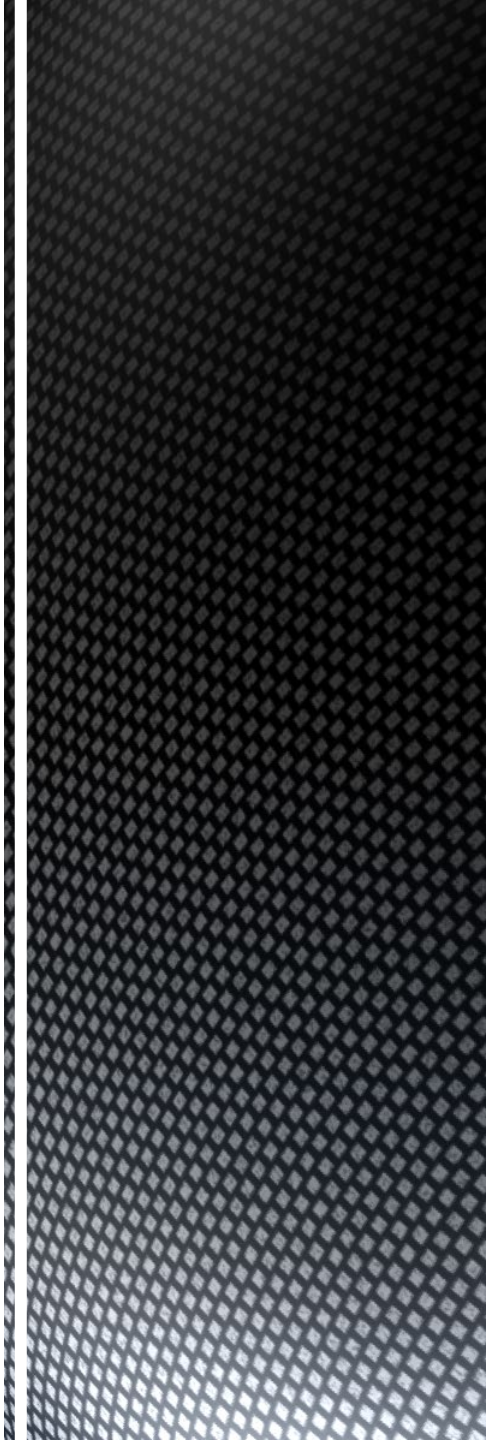
Actividades

Varias Actividades en una Aplicación

Intents

Implicit Intents

Ciclo de vida



Nueva Activity

El concepto de actividad en Android representa una unidad de interacción con el usuario, es lo que coloquialmente llamamos una pantalla de la aplicación.

Una aplicación suele estar formada por una serie de actividades, de forma que el usuario puede ir navegando entre actividades.

Una aplicación estará formada por un conjunto de actividades independientes, es decir se trata de **clases** independientes que **no comparten variables**, aunque todas trabajan para un objetivo común.

Toda actividad ha de ser una subclase de Activity.

Para finalizar una actividad se llama al método finish() que termina su ejecución.

Nueva Activity

El paradigma de programación Android no tiene un método main() donde empieza la ejecución.

Android inicia el código en una instancia de Activity invocando métodos de devolución de llamada específicos (métodos Callback) que corresponden a etapas específicas de su ciclo de vida.

- Le decimos a Android qué Activity es la principal.
- Android, el Sistema Operativo, llama a los métodos de esa clase en función de la evolución del ciclo de vida de esa Activity.
- El programador debe preparer código en estos Callbacks
- Android gestiona los ciclos de vida de las Activities y otros objetos, no el programador.

Cuando una App invoca a otra, la App que llama invoca una actividad en la otra, no a la App en si (no hay main()).

De esta manera, una actividad sirve como punto de entrada para la interacción de una App con el usuario.

Intents

Un Intent representa la voluntad de realizar alguna acción o tarea

Un Intent nos permite lanzar una actividad o servicio de nuestra aplicación o de una aplicación diferente.

Podemos ver los Intents como un tipo de lenguaje para especificar operaciones que queremos realizar

- Quiero seleccionar un contacto
- Quiero hacer una foto
- Quiero llamar a un numero de teléfono
- Etc.

Intents

Un Intent es un objeto de mensajería que se puede usar para solicitar una acción a otro componente de la aplicación. Hay tres casos de uso fundamentales:

Para lanzar una Activity:

El Intent describe, (se construye indicando) la actividad que hay que lanzar y puede almacenar (llevar consigo) datos que sirvan a dicha actividad.
LLamando a startActivity() pasándole el intent se lanza la otra actividad.

Si se quiere recibir datos de la actividad lanzada cuando esta finalice, usaremos el método startActivityForResult() (en vez de startActivity()). En ese caso, el método callback onActivityResult() de la actividad que llama será puesto en ejecución por Android cuando termine la otra actividad, y recibirá como parámetro un Intent que llevará los datos de retorno

Para lanzar un Service:

Un Service realiza operaciones en background sin interface de usuario.
Se puede lanzar un servicio para realizar una operación pasando un Intent al método startService(). El Intent describe el servicio a lanzar y puede almacenar datos válidos para dicho servicio.

Para lanzar un Broadcast:

Un Broadcast es un mensaje que cualquier app puede recibir.
El sistema android lanza distintos broadcast para informar de eventos del sistema, por ejemplo el *sistema arranca*, *el dispositivo comienza a cargar*...
Se puede lanzar un broadcast pasando un Intent a sendBroadcast().

Nueva Activity

Para crear una nueva actividad debemos:

- Crear un nuevo Layout para la activity:
  El interfaz gráfico de la activity (XML)
- Crear una nueva clase para la activity:
  El código Java que gestiona la activity.
- Registrarla en el AndroidManifest.xml:
  Indicándole a Android que tenemos una nueva actividad con las capacidades que indiquemos.

Crear un nuevo Layout para la actividad:

New -> Other -> Android XML Layout File

Crear una nueva clase descendiente de Activity (en el package donde queramos)

New -> Class ->

Dar el nombre a la Activity, por ejemplo saludo

La superclass debe ser: android.app.Activity

Registrarla obligatoriamente en el Android Manifest (Android Developers Añadir una Activity al Manifiesto)

Añadir la actividad usando:

<activity android:name="saludo" />

Para que nuestra nueva Activity sea visible será necesario activarla (llamarla) desde otra actividad. Por ejemplo en el onClick() de un botón realizamos la llamada (usando un intent).

Nueva Activity

Preparamos el código de la nueva Activity para la llamada a los Callback que hará el Sistema Operativo Android.

En la nueva clase descendiente de Activity sobrecargaremos el método OnCreate

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

  ....
}
```

Para lanzar la nueva actividad desde el OnClick() de un button podemos utilizar el siguiente código, que hará uso de un Intent, utilizará el método putExtra del Intent para enviar datos con el Intent y llamará al método startActivity() pasándole el Intent.

```
Intent intent = new Intent(MainActivity.this, saludo.class);
intent.putExtra("texto_saludo", "Hola Caracola");
startActivity(intent);
```

Nueva Activity

Asociar actividad a la aplicación. (en AndroidManifest.xml) Ejemplos:

```
<!-- Main Activity-->
    <activity android:name="YourActivityName" >
        <intent-filter>
      <!-- MAIN represents that it is the Main Activity-->
            <action android:name="android.intent.action.MAIN" />
      <!-- Launcher Denotes that it will be the first launching activity-->
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

 <!--Add Other activities like this-->
    <activity android:name="YourActivityName2" >
    <!--Default Intent Filter-->
        <intent-filter>
            <action android:name="android.intent.action.DEFAULT" />
        </intent-filter>
    </activity>

 <!--OR Other activities like this And  intent filter is not necessary in other activites-->
    <activity android:name="YourActivityName3" >
    </activity>

 <!--OR Add Other activities like this-->
    <activity android:name="YourActivityName4" />
```

```
<activity android:name="saludo" android:label="Saludo"/>
```

Intents

Dos tipos de Intents

Explicit

Se indica el componente a lanzar por nombre (*fully qualified*)
Por ejemplo, desde una actividad lanzamos la activity *AcercaDe* por medio de una intención explicita que mostrará la pantalla Acerca De de nuestra app.

Implicit

No especifican un componente por su nombre.
Declaran una acción general a ser realizada, lo que permite a un componente de otra aplicación que la realice.

Pueden solicitar tareas abstractas como:
"quiero tomar una foto" o "quiero enviar un mensaje", "quiero mostrar un mapa"

Se resuelven en tiempo de ejecución.
Android mirará qué Apps han registrado la posibilidad de ejecutar ese tipo de actividad.
Si hay varias el sistema puede preguntar al usuario la actividad que prefiere utilizar.

Android Developers: Intent Comunes

Intents

Campos de un Intent:

- ACTION
- DATA & TYPE
- CATEGORY
- COMPONENT

Esta propiedades (nombre de componente, acción, datos y categoría) representan las características definitorias de una intent.

Mediante la lectura de estas propiedades, el sistema Android puede resolver qué componente de la aplicación debe iniciar.

- EXTRAS
- FLAGS

Sin embargo, una intent puede tener información adicional que no afecta cómo se resuelve en un componente de la aplicación. Una intent también puede incluir Extras y Flags

Intents

COMPONENT

Este elemento es opcional pero si queremos que sea un Explicit Intent es obligatorio, pues determina específicamente a qué componente de qué aplicación se le debe pasar el Intent. Si se omite, el intent será un Implicit Intent.
Para lanzar un servicio es obligatorio también.

Indicamos pues qué componente recibirá este Intent.

Para indicarlo en el Intent en el constructor usamos:

```
Intent newInt = Intent(Context packageContext, Class<?> cls);
```

Donde Class<?> debe ser la clase Activity que queremos lanzar.
También podemos crear el inten con

```
Intent newInt = new Intent ();
```

Y luego usamos uno de estos métodos del nuevo Intent

- .setComponent()
- .setClass()
- .setClassName()

Intents

ACTION

Una cadena de caracteres donde indicamos la acción a realizar o nombres de operaciones a realizar, algunos ejemplos son:

ACTION_DIAL -> Llamar a un Numero
ACTION_EDIT -> Mostrar información a Editar
ACTION_SYNC -> Sincronizar información del dispositivo con un servidor
ACTION_MAIN -> Empezar como actividad inicial de una aplicación

¿Cómo indicamos el ACTION en un Intent? Con una de estas dos formas.

```
Intent newint=new Intent (Intent.ACTION_DIAL);

Intent newint=new Intent();
   newint.setAction(Intent.ACTION_DIAL);
```

Reference Andriod Developers: Intent Class ← Lista con los Actions disponibles

Intents

DATA

Información asociada al Intent, en formato URI (Uniform Resource Identifier) que referencia a los datos sobre los que se va a trabajar. Adicionalmente se indica su tipo MIME. El tipo generalmente depende del tipo de acción,

Por ejemplo si el Action es ACTION_EDIT, el DATA contendrá el URI del archivo a editar.

Información para ver en un Mapa

```
Uri.parse("geo:47.6,-122.3")    (geo:latitud,longitud)
```

Número a marcar con el "phone dialer"

```
Uri.parse("tel:+15555555555")
```

¿Cómo indicamos DATA en un Intent? Con una de las siguientes formas.

```
Intent newInt = new Intent(Intent.ACTION_DIAL,Uri.parse("tel:+15555555555"));

Intent newInt = new Intent(Intent.ACTION_DIAL);
newInt.setData(Uri.parse("tel:+15555555555"));
```

Formatos de imagen –> image/*, image/png, image/jpg

Formato de texto –> TEXT/PLAIN, TEXT/HTML

NOTA: Si no se especifica el tipo MIME, Android lo inferirá

(Listado completo MIME-Types) (Los más importantes)

Para añadirlo al Intent:

```
Intent.setType(String type)
```

Si se van a especificar ambos, data y type, hay que usar esta función. (setData anula Type y viceversa.)

```
Intent.setDataAndType(Uri data, String type)
```

Intents

EXTRAS

Almacena información adicional relacionada con el Intent

Se configura como un MAP con parejas <key-value>

La actividad receptora del Intent debe conocer tanto el nombre como el tipo de datos de la información contenida en el campo EXTRAS:

Reference Andriod Developers: Intent Class ← Lista con los Extra Data disponibles

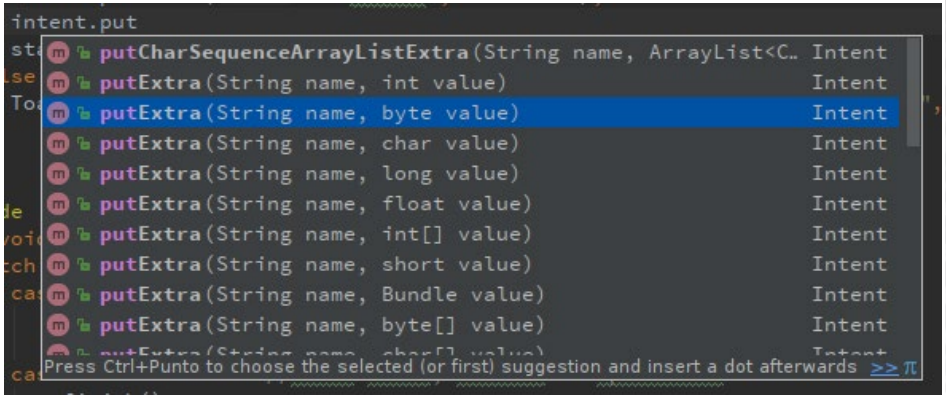
EJEMPLO:

```
Intent newInt = new Intent(Intent.ACTION_SEND);
newInt.putExtra("email_list",
                new String[]{"usuario@empresa.es",
                             "alumno@empresa.es",
                             "profesor@empresa.es"}
               );
```

Diferentes formas de añadir EXTRAS dependiendo del tipo de datos:

```
putExtra(String name, String value);
putExtra(String name, float[] value);
```

Etc.



Intents

FLAGS

Representa información sobre cómo debe manejarse el Intent

Ejemplos:

FLAG_ACTIVITY_NO_HISTORY

Al aplicarlo a un intent:

Hace que una actividad lanzada con él, NO se coloque en la pila de históricos (History stack)

FLAG_DEBUG_LOG_RESOLUTION

Le indica a Android que imprima (log) información extra cuando se procese este Intent (Útil cuando lo usemos con Intents implicitos)

FLAG_ACTIVITY_SINGLE_TOP

Si se active, la activity no será lanzada si ya está corriendo en la cima del history stack.

Añadiremos los flags usando:

setFlags(int)

addFlags(int)

Ver más en:

http://developer.android.com/guide/components/intents-filters.html

http://developer.android.com/reference/android/content/Intent.html