Temari

1) Introducción
2) Framework Android
3) Proyectos Android y Android SDK
4) Activity
5) Fragments, Views y ListViews
6) Intents
7) Layouts y Custom Views
8) Resources y Themes
9) Dialogs, Menus y WebView
10) Persisténcia de datos
11) Tareas en Background e internet
12) SQLite y content providers
13) Notificaciones

5 – Intents

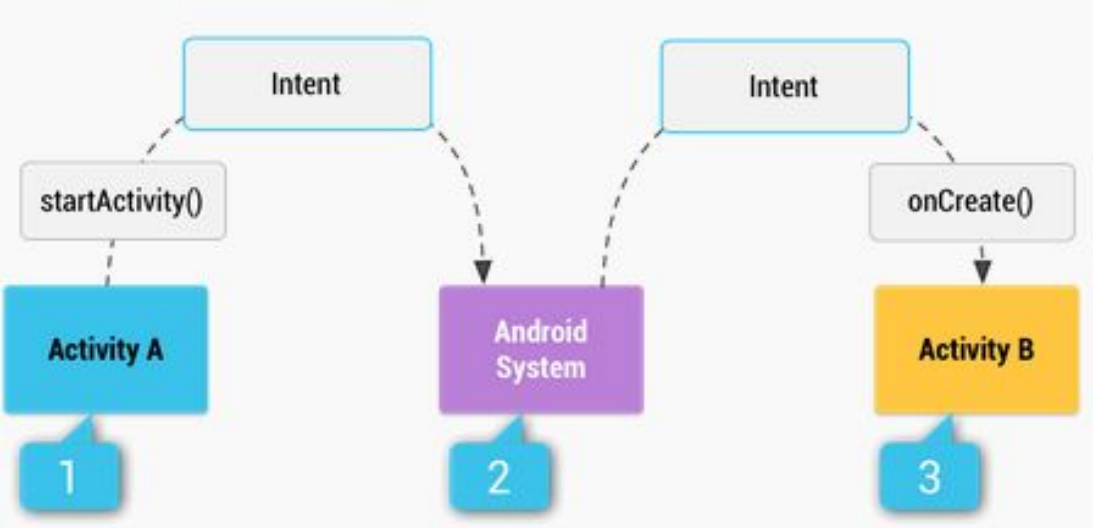
- Intents: Explícitos e Implícitos
- Intents implícitos: Acciones i URIs
- Datos en un Intent: Extras
- Sub-activities
- IntentFilters

Intents, qué son?

Representan una **interacción** entre componentes Android, un **mensaje**.

Pueden enviarse a todo el sistema (**broadcast**) o a una **Activity** o **Service** específico

P. ej., cuando desde una activity se inicia otra, y la 1ª va al background, se hace a través de un Intent



Intents explícitos

Se especifica explícitamente qué **componente** (activity, service...) **se ha de iniciar**

P. ej., en una activity:

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

Intents implícitos

En vez de especificar qué componente hay que iniciar, se especifica una **acción** (qué queremos hacer) y un **recurso** (sobre qué lo queremos hacer)

P. ej., abrir un navegador a una URL:

```
Intent i = new Intent( Intent.ACTION_VIEW ,
      Uri.parse("http://www.google.com");
startActivity(i);
```

Y cómo sabe que hay que hacer??

Intents implícitos

Native Android actions

- ACTION_ANSWER
- ACTION_DELETE
- ACTION_DIAL
- ACTION_EDIT
- ACTION_PICK
- ACTION_SEARCH
- ACTION_SEND
- ACTION_VIEW
- ACTION_WEB_SEARCH

https://developer.android.com/reference/android/content/Intent.html
https://developer.android.com/guide/components/intents-common.html

Combinaciones de URI i Actions

Envio de datos: Extras

Tanto el componente que llama como el que es llamado tienen acceso al intent, que continene una estructura de datos: Intent → extras del intent

- Para añadir:
  Void putExtra(String name, Bundle value);
- Para recuperar:
  Bundle b = getExtras();
  String value1 = b.getString(String);

Puede contener:
- Tipos primitivos
-String
- implements Parcelable
- implements Serializable
- array de los anteriores

P. ej.:

```
Intent i = new Intent(this, ActivityTwo.class);
i.putExtra(Intent.EXTRA_TEXT, "Hello!");
...
Bundle extras = getIntent().getExtras();
if(extras!=null){
      String value1=extras.getString(Intent.EXTRA_TEXT);
      // hacer algo
}
```


Sub-activities

Una activity se puede iniciar para recibir un resultado, p. ej. **Buscar y seleccionar un elemento**

En vez de startActivity(), se usa **startActivityForResult():**

```
Intent i = new Intent(this, ActivityTwo.class);
//put extras if required
startActivityForResult(i, SHOW_SUB_ACTIVITY);

...
Public void onActivityResult(int requestCode,
      Int resultCode, Intent data){

      Super.onActivityResult(requestCode, resultCode, data);
      if(requestCode == SHOW_SUB_ACTIVITY){
            Bundle resultData = data.getExtras();
            // do something
      }
}
```

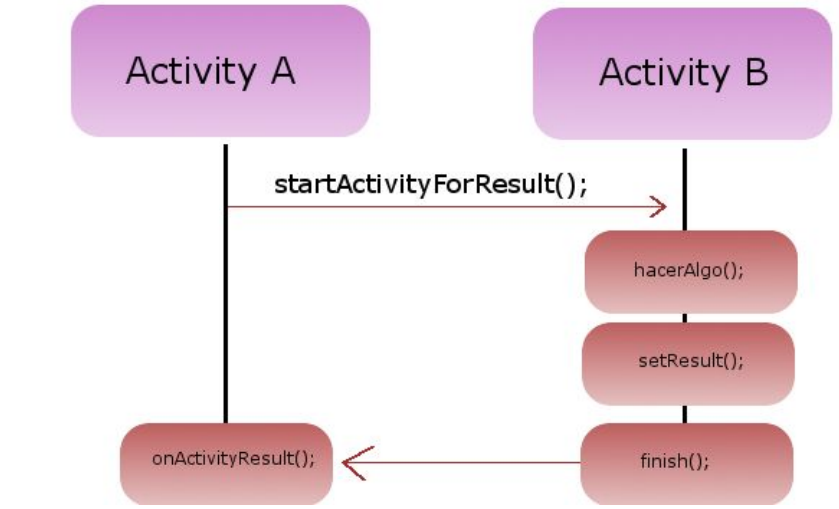
Integer constant!
Para identificar qué
Sub-activity nos devuelve
El resultado que recibimos

Sub-activities

En la sub-activity se devuelve el resultado, ejecutando el **setResult()** en el método **finish()** :

```
Public void finish(){
        Intent data = new Intent();
        Data.putExtra(''Text'', ''result'');
        setResult(RESULT_OK,data);
        Super.finish();

}
```



Intent Filters

Se peude hacer que nuestra appresponda a intents **implícitos** → definiendo un **Intent Filter** al **AndroidManifest.xml**

```
<activity android:name="MyBrowserActivity">
   <intent-filter>
      <action android:name="android.intent.action.VIEW"/>
      <category android:name="android.intent.category.DEFAULT"/>
      <data android:scheme="http"/>
   </intent-filter>
</activity>
```

Dentro de la Activity, con **getIntent() podemos obtener los datos de la llamada**:

```
Intent i = getIntent();
Uri data= i.getData();
i.getAction();
```