

# Android Essentials

## Intents

Hasta ahora aprendimos como hacer y manejar `Activity's` aisladas

¿Y si necesitamos que las actividades interactúen?

Es necesario un elemento que nos permita saltar de pantalla a pantalla.

A diferencia de los sistemas tradicionales, donde una interfaz de usuario contiene ventanas hijas y cuadro de diálogos, en Android dada su filosofía, el sistema es el producto de acoplar pequeños subsistemas

Una interfaz casi es un única aplicación que si necesita presentar una segunda UI deberá conectarse con otra interfaz.

un **Intent** es un objeto que sirve de mensaje entre dos componentes

Sirve para indicarle a Android que se quiere interactuar con otro componente en particular

Con un **Intent** podemos:

- Ir de un **Activity** a otro **Activity**
- Llamar a un servicio
- Invocar a una app externa

Toda la información que se desee pasar de un componente a otro se colocará dentro del **Intent** y se recuperará como un objeto **Bundle** en el **onCreate()** de nuestro componente objetivo

# Intents explícitos



Desde una actividad queremos comunicarnos con otra **actividad específica**

Y con esto me refiero a una actividad propia de nuestra App, **no externa**

Para esto se utilizará un **Intent** explícito

```
Intent i = new Intent(MainActivity.this, Clase1Activity.class);  
startActivity(i);
```

Solo hace falta indicarle el contexto actual ( **MainActivity**) y el **\*.class** con el cual queremos interactuar

Para enviar información se usa el método `putExtra()`, el recibe dos parámetros, un TAG que se usa para identificar la información y un objeto que puede ser :

- `boolean`
- `byte`
- `char`
- `short`
- `int`
- `long`
- `float`
- `double`
- `String`
- `CharSequence`
- `Parcelable`
- `Serializable`
- `Bundle`
- O `Array<T>` de cualquiera de estos

```
Intent i = new Intent(MainActivity.this, Clase1Activity.class);  
i.putExtra("un nombre", new String("Hola"));  
startActivity(i);
```

```
// en el onCreate() del otro activity  
Bundle extras = getIntent().getExtras();  
Serializable serializable = (Serializable) extras.get("Mi TAG");
```

**Como podrán notar, si quiere pasar alguna Clase propia, tiene que heredar Serializable o Parcelable !!!**

# Intents implícitos



Cuando no sabemos qué actividad puede llegar a resolver la acción que necesitamos resolver, pero sabemos que quizás en el dispositivo exista una app que resuelve lo que queremos, usamos un Intent implícito

Especificamos:

- Acción a soportar por la actividad
- MIME Type a soportar
- Categorías a soportar
- Datos a transferir

```
Intent intent = new Intent();  
intent.setType("image/*");  
intent.setAction(Intent.ACTION_GET_CONTENT);  
intent.addCategory(Intent.CATEGORY_OPENABLE);  
startActivityForResult(intent, REQUEST_CODE);
```

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

```
Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:928-76-34-26"));  
startActivity(intent);
```

# Intents implícitos



## ACTIONS y CATEGORIES

<https://developer.android.com/reference/android/content/Intent.html>

# Comunicación entre Activities



Creamos el Intent, luego de esto tenemos 2 opciones

Llamamos a `startActivity()`, lo que hará que android busque la mejor alternativa y pasa el intent para que lo maneje, pero la actividad inicial nunca se enterará si la actividad secundaria finalizó

```
public void startActivity(Intent intent)
```

Podemos invocar `startActivityForResult()`, pasando el intent y un número único para la actividad origen. La actividad origen será notificada del fin de la secundaria mediante el "callback" `onActivityResult()`

```
public void startActivityForResult(Intent intent, int requestCode)
```



```
protected void onActivityResult(int requestCode, int resultCode, Intent data)
```

**requestCode** : código original utilizado en la invocación del **Intent**

**resultCode** : es el resultado (o código de error) del llamado. Este valor debería ser cero si todo fue correcto y valor distinto para indicar que algo falló. Existen algunos códigos predefinidos que pueden ser utilizados, pero también pueden definirse nuevos códigos para errores más específicos.

**data**: és un Intent, el cual puede retornar datos al llamador en sus extras.

# Intent Filters

Especifica los tipos de Intents que la actividad, servicio o receiver puede responder

Declaran las capacidades del componente padre. Qué tipos de Intents puede recibir y manejar

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

```
<intent-filter>
  <action android:name="android.intent.action.EDIT" />
  <action android:name="android.intent.action.VIEW" />
  ...
</intent-filter>
```

```
<intent-filter>
  <category android:name="android.intent.category.DEFAULT" />
  <category android:name="android.intent.category.BROWSABLE" />
  ...
</intent-filter>
```