

# Comunicación entre Actividades

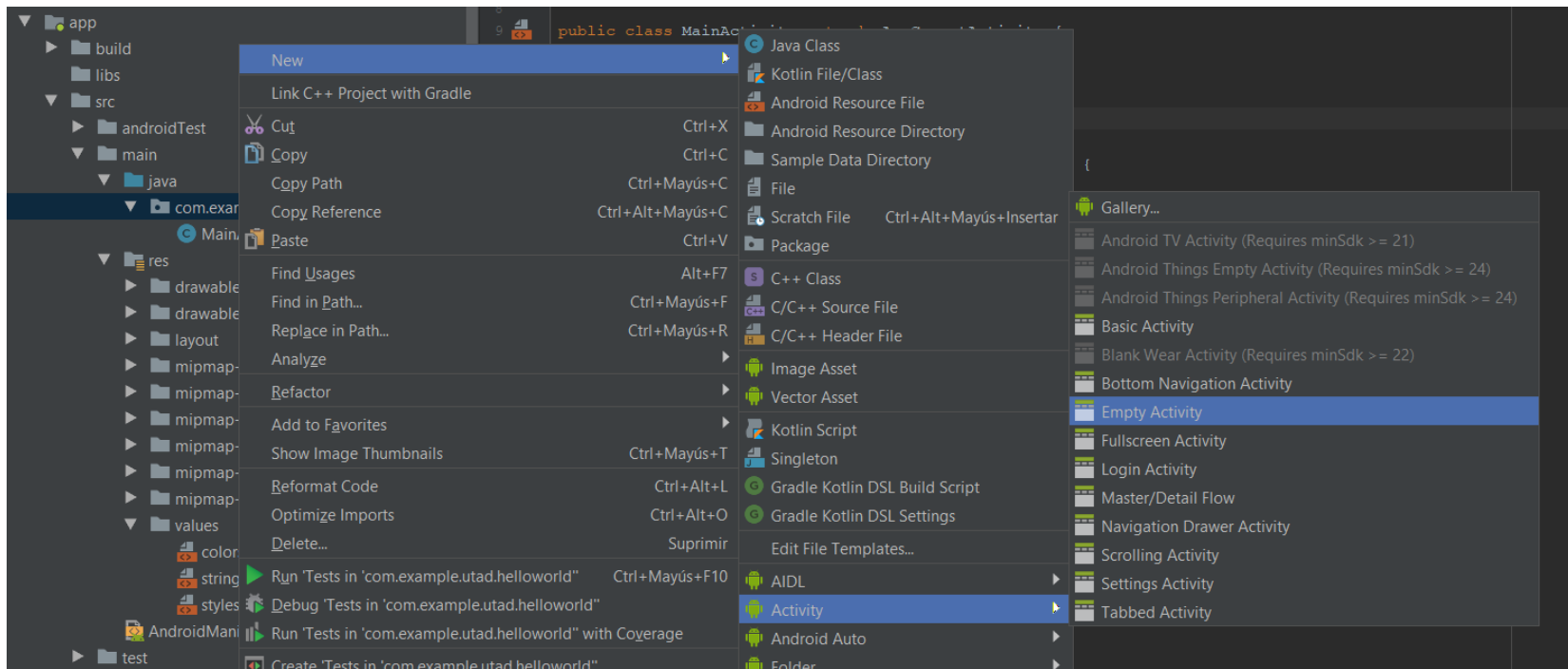
Desarrollo para plataformas móviles



Borja Martin Herrera  
[Borja.herrera@u-tad.com](mailto:Borja.herrera@u-tad.com)

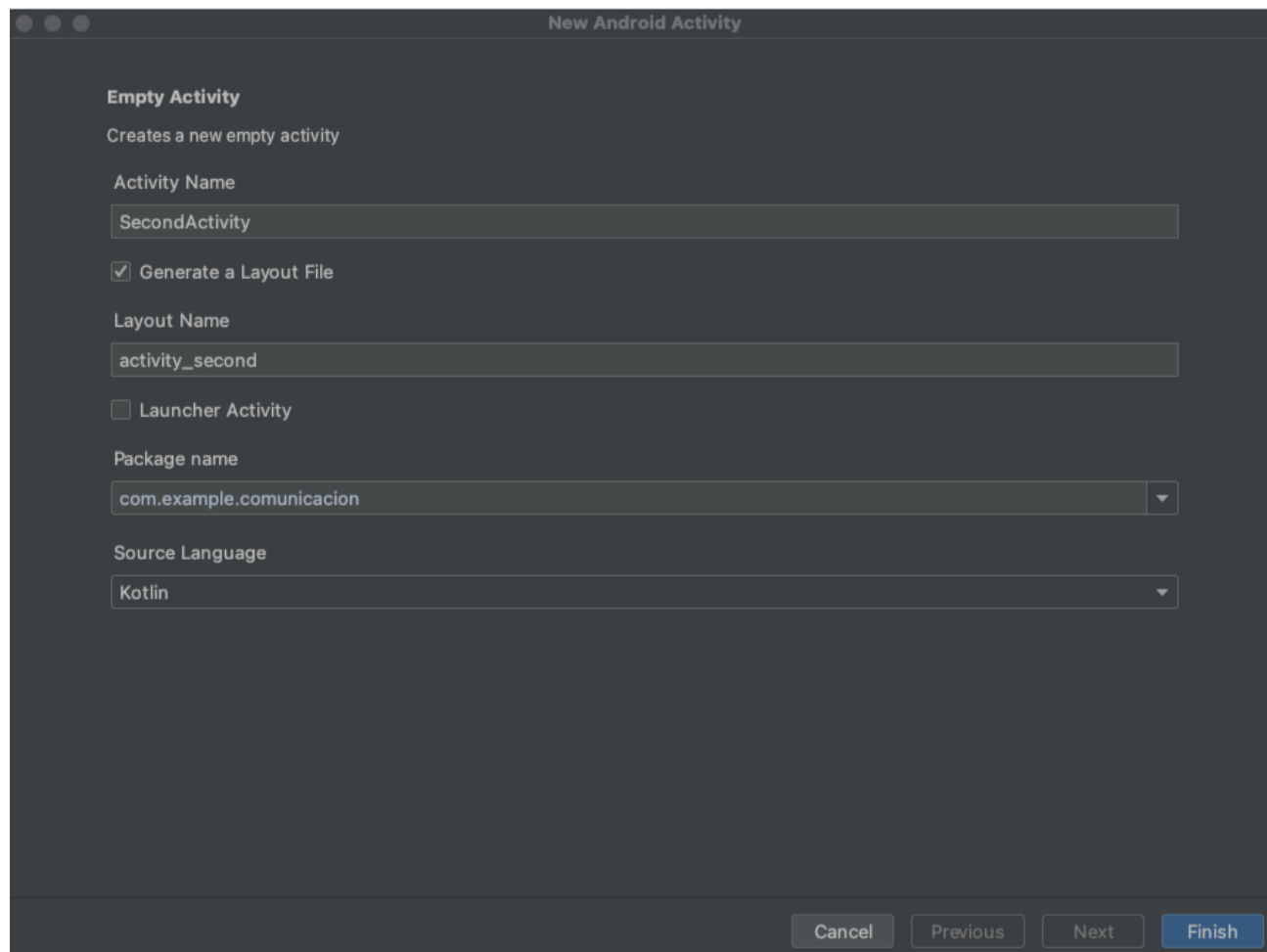
# Crear segunda actividad

- Hasta ahora, sólo hemos tenido una actividad en la app.
- Vamos a ver cómo crear una segunda actividad y como podemos pasar de una a otra.



# Crear segunda actividad

- Le ponemos nombre y dejamos que *AndroidStudio* cree automáticamente el *Layout* de la actividad.



The screenshot shows the 'New Android Activity' dialog box in Android Studio. The dialog is titled 'New Android Activity' and has a dark theme. It contains the following fields and options:

- Empty Activity**: Creates a new empty activity.
- Activity Name**: Text field containing 'SecondActivity'.
- Generate a Layout File**: Checked checkbox.
- Layout Name**: Text field containing 'activity\_second'.
- Launcher Activity**: Unchecked checkbox.
- Package name**: Text field containing 'com.example.comunicacion'.
- Source Language**: Dropdown menu set to 'Kotlin'.

At the bottom right, there are four buttons: 'Cancel', 'Previous', 'Next', and 'Finish'.

# Crear segunda actividad

- Automáticamente la añade a AndroidManifest.xml

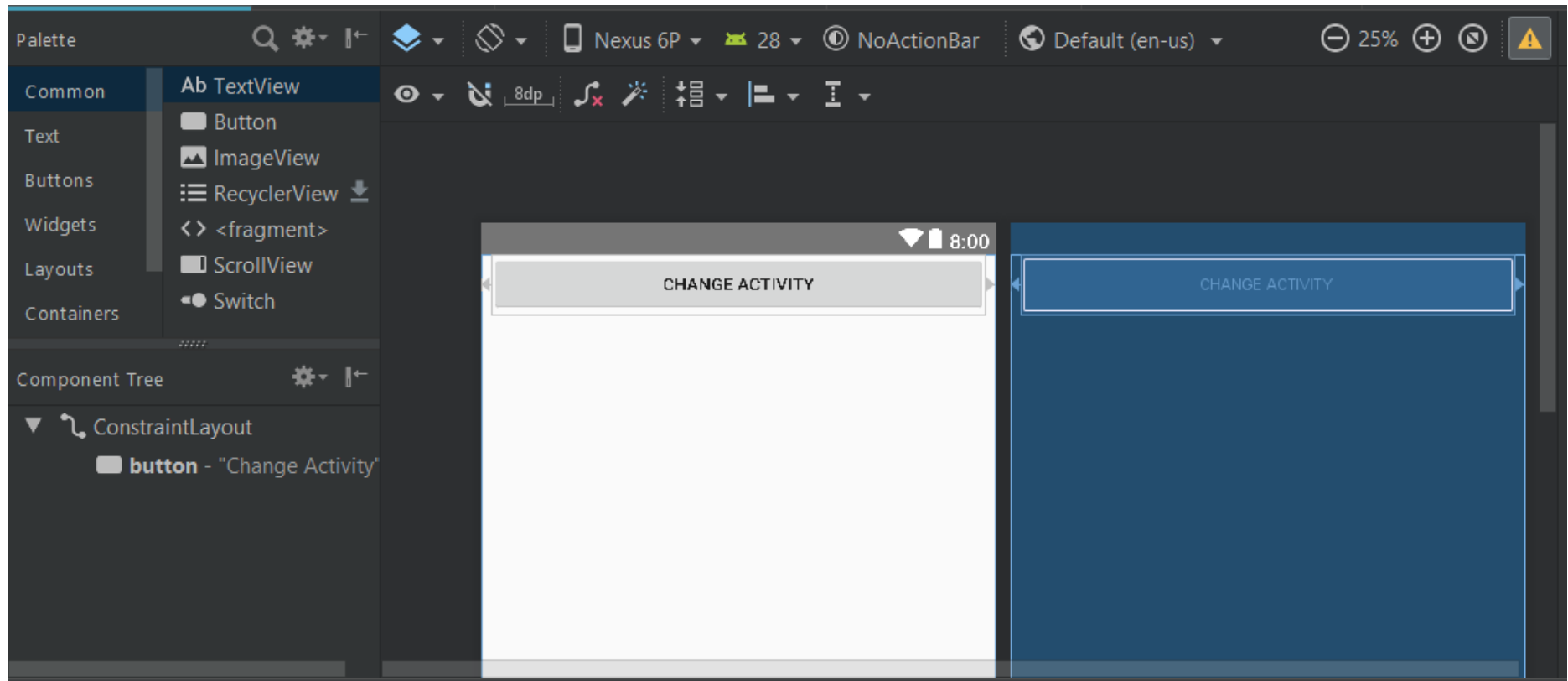
```
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="comunicacion"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Comunicacion"
    tools:targetApi="31">
    <activity
        android:name=".SecondActivity"
        android:exported="false">
        <meta-data
            android:name="android.app.lib_name"
            android:value="" />
    </activity>
    <activity
        android:name=".MainActivity"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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

        <meta-data
            android:name="android.app.lib_name"
            android:value="" />
    </activity>
```

# Crear segunda actividad

- El Layout de la actividad principal sólo tiene un botón:



# Paso de una actividad a otra

- Para poder cambiar de una actividad a otra, tenemos que usar el método:
  - `startActivity(Intent intent)`
  - Tenemos que pasarle un objeto de la clase `Intent`
  - Los objetos `Intent` son cómo mensajes con “intención” de hacer cosas (siempre abrir otra actividad de nuestra app o de una app ajena).
  - Tienen un emisor, un receptor y datos.
  - La actividad avisa a Android de que se quiere abrir una actividad

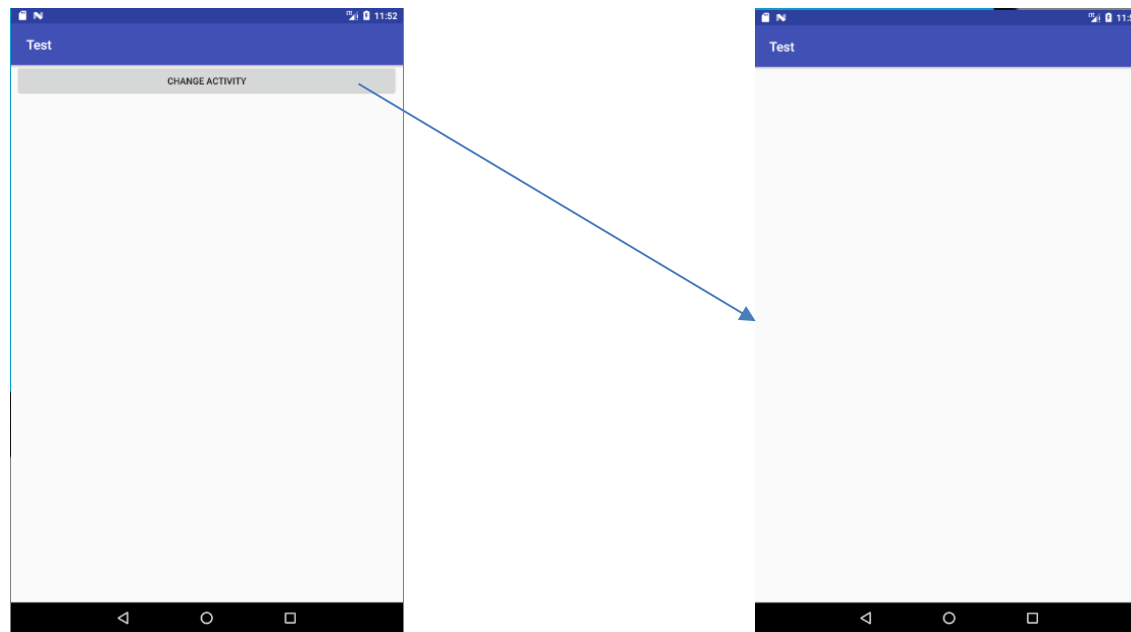
```
val intent = Intent(applicationContext, SecondActivity::class.java)
startActivity(intent)
```



Para: ViewMessage Activity  
De: Esta actividad

# Paso de una actividad a otra

- Dejamos la segunda actividad en blanco.
- Si ejecutamos la app y pulsamos el botón vemos cómo nos lleva a la segunda actividad.



# Paso de una actividad a otra

- Se pueden mandar datos entre actividades.
- Tenemos que añadirlos al objeto de la clase *Intent*.
- Usamos el método *putExtra* del objeto *Intent* para añadir la información que queremos transmitir.
- Tiene como parámetros un id del mensaje y el valor.





# Paso de una actividad a otra

- El método *putExtra* puede tener parámetros de diferente tipo:

```
val intent = Intent(applicationContext, SecondActivity::class.java)
intent.put
```

	Method Signature	Return Type
star	<code>putExtras(extras: Bundle)</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: FloatArray?)</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: LongArray?)</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: CharSequence?)</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: Byte)</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: Float)</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: IntArray?)</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: Array&lt;(out) Pa...</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: Parcelable?)</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: Long)</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: ByteArray?)</code>	<code>Intent</code>
	<code>putExtra(name: String!, value: Array&lt;(out) Ch</code>	<code>Intent</code>

Press ^ to choose the selected (or first) suggestion and insert a dot afterwards [Next Tip](#) ⋮



# Paso de una actividad a otra

- El método para cambiar entre actividades quedará de la siguiente manera:

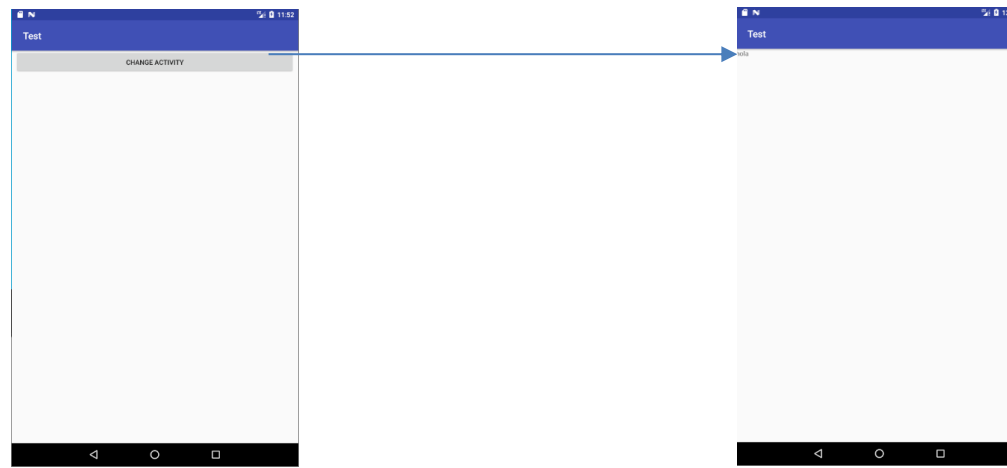
```
val intent = Intent(applicationContext, SecondActivity::class.java)
val bundle = Bundle()
bundle.putString("name", "dato pasar")
bundle.putString("surname", "dato pasar")
bundle.putInt("age", 5)
intent.putExtras(bundle)
startActivity(intent)
```

- Tenemos que recoger el mensaje en la segunda actividad.
  - Primero tenemos que “recuperar” el *intent* con el método *getIntent()* de la actividad.
  - Luego usamos *getTipoDeDatoExtra(String id)* para leer el valor.

```
class SecondActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_second)
        val nombre = intent.extras?.getString(key: "name")
        val apellido = intent.extras?.getString(key: "surname")
        val edad = intent.extras?.getInt(key: "age")
    }
}
```

# Ejemplo de App

Ejercicio: Crear un TextView en la segunda actividad y que se cargue el texto que estamos pasando desde la primera actividad.



# Apertura de otras Apps

- La clase *Intent* es muy potente, podemos usarla para mandar mensajes a otras apps.
- Vamos a indicarle en el constructor el tipo de *Intent* que queremos hacer especificando la acción que tiene que realizar como parámetro.

```
Intent intent = new Intent( packageContext: this, ViewMessage.class);
```



```
Intent intent = new Intent(Intent.ACTION_SEND);
```

- La lista de acciones posibles es enorme. Lo mejor es consultarla cuando necesitemos hacer algo concreto.

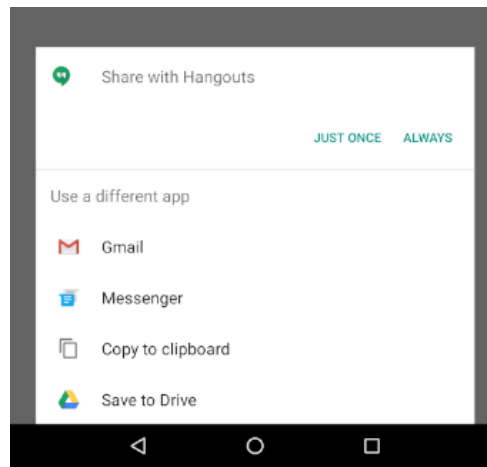


# Apertura de otras Apps

- Tenemos que especificar que queremos mandar y el propio dato en sí.
- Esto lo hacemos de la siguiente manera:

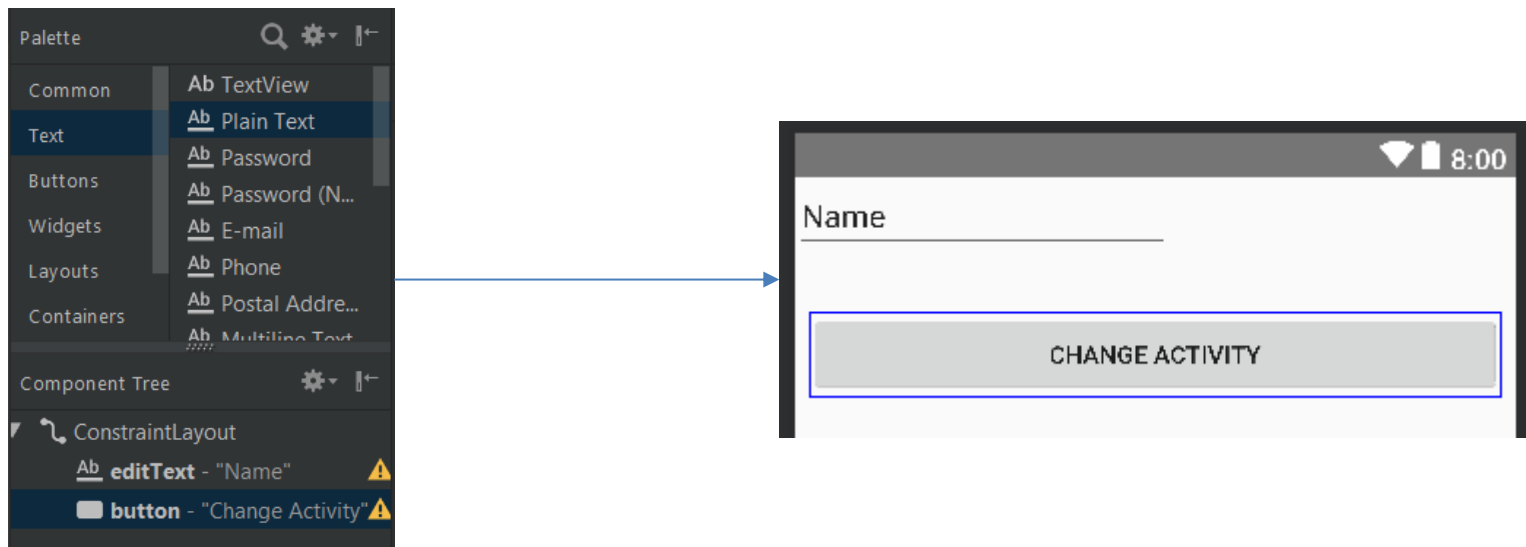
```
binding.botonEnvio.setOnClickListener { it: View!  
    val intent = Intent(Intent.ACTION_SEND)  
    intent.type = "text/plain"  
    intent.putExtra(Intent.EXTRA_TEXT, value: "Mensaje que se quiere enviar")  
    startActivity(intent)  
}
```

- Al hacer *startActivity*, Android buscará que actividades de las apps instaladas en el teléfono pueden manejar mensajes de texto plano.



# Ejemplo de App

Ejercicio: Realizad una app en la que se pueda escribir un mensaje y se rellene un email.



# Recibir Intent desde otra app

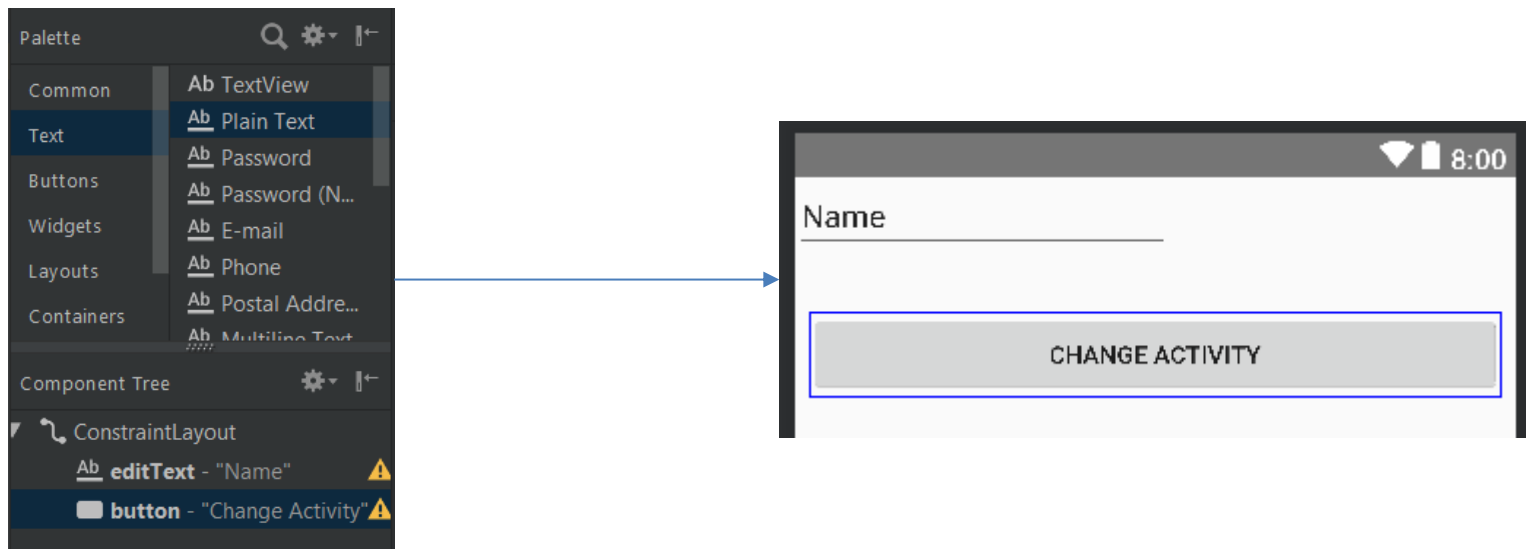
- ¿Y cómo sabe Android que actividades puede abrir?
  - Gracias al archivo AndroidManifest.xml
- Se puede especificar en cada actividad a que *Intents* puede responder.
- Para el ejemplo anterior haríamos:

```
<activity android:name=".ViewMessage">
  <intent-filter>
    <action android:name="android.intent.action.SEND" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType = "text/plain/" />
  </intent-filter>
</activity>
```



# Ejemplo de App

Ejercicio: Escribid una app con una actividad que pueda recibir un texto y lo imprima en pantalla.





# Selector de Android

- Podemos querer obligar al usuario a que tenga que seleccionar siempre la app a la que quiere enviar el mensaje.



# Selector de Android

- Para hacer esto, vamos a encapsular nuestro *Intent* dentro de un *Intent* que va a servir para abrir el selector.
- Este segundo *Intent* lo vamos a crear con el método *createChooser* de la clase *Intent*.

```
binding.botonEnvio.setOnClickListener { it: View!
    val intent = Intent(Intent.ACTION_SEND)
    intent.type = "text/plain"
    intent.putExtra(Intent.EXTRA_TEXT, value: "Mensaje que se quiere enviar")
    intent.putExtra(Intent.EXTRA_SUBJECT, value: "Mensaje importante")
    intent.putExtra(Intent.EXTRA_EMAIL, arrayOf("borja.herrera@ext.live.u-tad.com"))
    val intentChooser = Intent.createChooser(intent, title: "Mandar un correo personalizado")
    startActivity(intentChooser)
```

# Ejemplo de App

Ejercicio: Rehacer la app que rellena un email para que lo haga usando *createChooser*.