



## Comunicación entre activities

### Paso a paso

1. **Crear un activity:** SenderActivity  
Automáticamente nos crea un archivo java, su correspondiente XML y lo declara en el manifest.
2. **Agregar un botón** en el layout del SenderActivity.  
Asignarle el nombre botonApretado al atributo onClick.
3. **Definir el método** botonApretado  
Recordar la firma: `public void apretarBoton(View view){}`
4. **Imprimir un toast** por pantalla que informe que el botón fue presionado.  
Recordar: `Toast.makeText(this, "Boton apretado", Toast.LENGTH_LONG).show();`
5. **Crear un nuevo Activity:** ReceptorActivity  
Automáticamente nos crea un archivo java, su correspondiente XML y lo declara en el manifest.
6. **Agregar un TextView vacío** en el layout del ReceptorActivity.  
Recordar que debemos agregarle un id para luego poder identificarlo desde el código.
7. **Definir una constante** en el ReceptorActivity  
`public static final String MENSAJE = "mensaje"`  
Recordar que por cada dato que el ReceptorActivity espera recibir, debemos definir una constante.

### En el SenderActivity, dentro del método botonApretado:

8. **Crear un Intent**  
Es un mensaje que me permite, entre otras cosas, pedirle al sistema operativo que dispare un nuevo Activity.
9. **Crear un Bundle** para poder enviar información.  
Recordar que un Bundle, es una especie de paquete que contiene información que queremos enviar.

10. **Cargar el Bundle** con un mensaje "Hola mundo conectado" Recordar que cuando cargamos información en el Bundle, utilizamos una clave de identificación (constante que definimos en el ReceptorActivity) y el valor que queremos enviar.

11. **Asociar el Bundle al Intent**

Recordar que para esto utilizamos el método `putExtras(bundle);`

12. **Realizar el start del Activity** utilizando el Intent.

Recordar que para esto utilizamos el método `startActivity(intent);`

**En el ReceptorActivity, dentro del método onCreate:**

13. **Obtenemos el Intent**

Recordar que para esto utilizamos el método `getIntent();`

14. **Obtenemos el Bundle**

Recordar que para esto utilizamos el Intent anterior y el método `getExtras();`

15. **Obtenemos el dato**

Recordar que para esto utilizamos la clave que definimos anteriormente y el método `getString();`

16. **Identificamos el textView y le seteamos el mensaje**

Recordar que para esto usamos el `findViewById(R.id.idDelTextView)` y el `setText(textoAMostrar);`