

# INTENT EXPLICITO

## ¿Qué es un Intent?

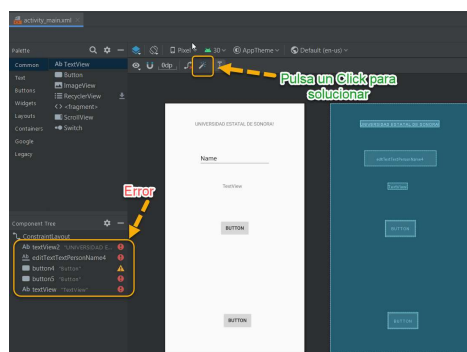
- Es un objeto de mensajería que puedes usar para solicitar una acción de otro componente de una APP.
- Los **intent** facilitan la comunicación entre componentes.
- Los **intent** explícitos especifican que aplicación las administrará, ya sea incluyendo el nombre del paquete de la APP de destino o bien el nombre de la clase del componente.
- Normalmente, el usuario usa un **intent** explícito para iniciar un componente en su propia aplicación porque conoce el nombre de la actividad o el servicio que desea iniciar.
- **Ejemplo:**
  - La puedes utilizar para iniciar una nueva actividad en respuesta a una acción del usuario.
- En otras palabras el intent nos permite facilitar la comunicación entre dos actividades, es decir cambiar de un **Activity a otro** y para hacerlo se utilizan los **index** y se dice que son explícitos ya sabemos de qué **Activity** a que **Activity** van a pasar.

## EJERCICIO: Ejemplo de Intent Explícito

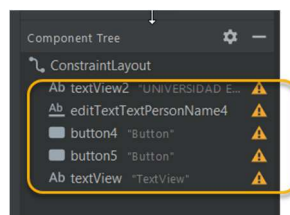
1. Lo primero que haremos será realizar un diseño que nos permita visualizar el concepto, para hacerlo, nos vamos a dirigir a nuestro **Activity\_main.xml** en vista de Diseño y le agregaremos los siguientes componentes:

- 1 **Edit Text-Numérico**
- 1 **TextView**
- 2 **Button**

**NOTA:** Observa que al agregar los componentes nos muestra un error, y esto se debe a la distribución del **Layout**, para resolverlo, selecciona los componentes y pulsa un click sobre el botón **INFER CONSTRAINT**

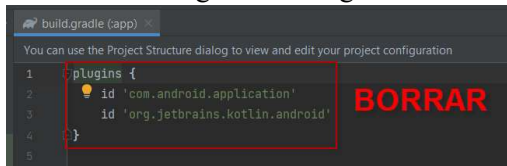


Observar cómo **han cambiado** los iconos de color **rojo** a color **naranja**, una vez que aplicas **Infer Constraint**

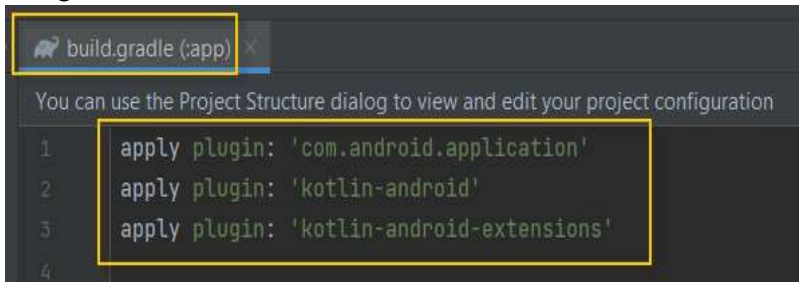


## AJUSTAR EL ARCHIVO DE CONFIGURACIÓN GRADLE

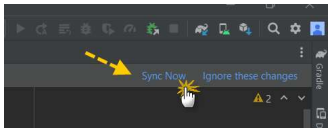
- Elimínanos el siguiente código



- Apertura el archivo que se identifica con el nombre “**Build.gradle**” y colocar el siguiente código

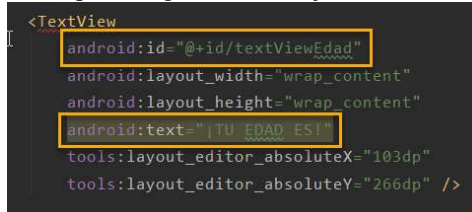


- Con la finalidad de aplicar los cambios, pulsaremos un click sobre la opción que se identifica con el nombre “**Sync Now**”



2. Como siguiente paso, dirígete a la **VISTA DEL CÓDIGO** y en el **TextView** que acabas de incorporar cambiaremos el

- También cambia el **ID** por **textViewEdad**
- **Text** por el siguiente mensaje “**TU EDAD ES**”



- 2.1 Al **EditText** le vamos a agregar

- Cambiamos el **ID** por **editTextEdad**
- Colocamos un **Hint**, con la leyenda “**Ingresa tu año de nacimiento**”



## 2.2 Ahora dirígete a la sección de los BOTONES

- Cambiaremos los nombres de los identificadores, el **primer botón** tendrá el nombre **buttonCalcular**
- y el **segundo** **buttonSiguiente**,
- También le vamos a cambiar el atributo del **text**, en ambos botones
  - El primero será **CALCULAR**
  - y el segundo será **SIGUIENTE**

```
<Button
    android:id="@+id/buttonCalcular"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="CALCULAR"
    tools:layout_editor_absoluteX="103dp"
    tools:layout_editor_absoluteY="323dp" />

<Button
    android:id="@+id/buttonSiguiente"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="SIGUIENTE"
    tools:layout_editor_absoluteX="108dp"
    tools:layout_editor_absoluteY="419dp" />
```

## 2.3 Ahora dirígete al **buttonSiguiente**

- Cambiaremos el ancho, haciendo uso del atributo **layout\_width = "match\_parent"**
  - con la finalidad que el botón cubra el ancho de la ventana.
- Agregaremos el color de fondo con el atributo
  - **android:background="@color/design\_default\_color\_primary\_dark"**
- También cambiaremos el atributo del color del texto

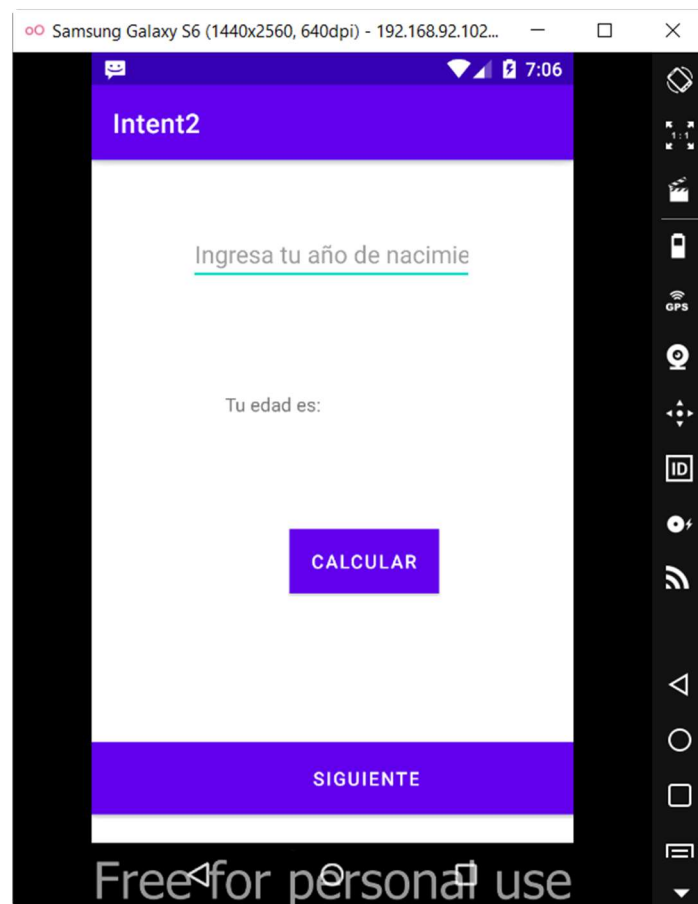
```
<Button
    android:id="@+id/buttonSiguiente"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="SIGUIENTE"
    android:background="@color/design_default_color_primary_dark"
    android:textColor="@color/white"
    tools:layout_editor_absoluteX="158dp"
    tools:layout_editor_absoluteY="421dp" />
```

## 2.4 Nos vamos a dirigir al **buttoncalcular**

- Cambiaremos el *color del fondo* con el atributo *background*
- También el *color de la letra* con el atributo *textColor*.

```
27 <Button
28     android:id="@+id/buttonCalcular"
29     android:layout_width="wrap_content"
30     android:layout_height="wrap_content"
31     android:text="CALCULAR"
32     android:background="@color/design_default_color_primary_dark"
33     android:textColor="@android:color/white"
34     tools:layout_editor_absoluteX="158dp"
35     tools:layout_editor_absoluteY="328dp" />
```

## DISEÑO DE LA APP QUE EJEMPLIFICA EL INTENT



## INTENT EXPLICITO -- (codificar)

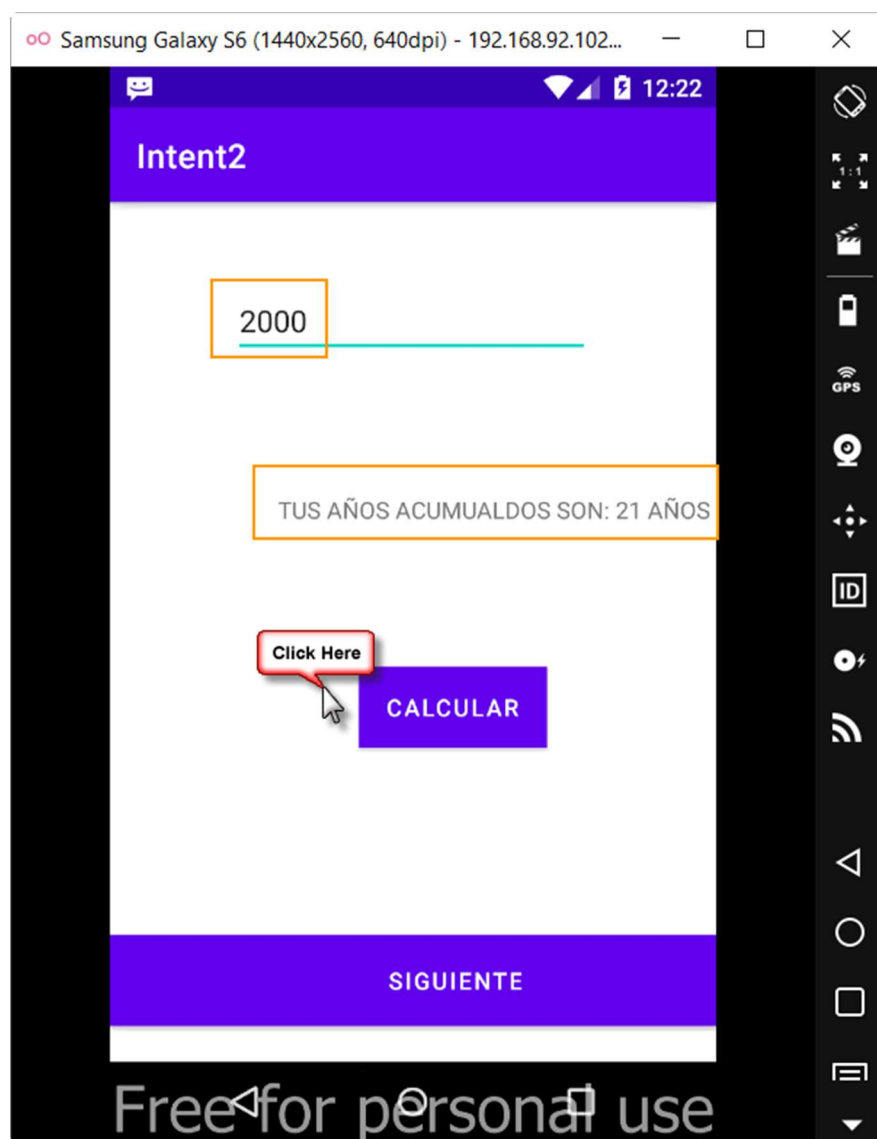
1. Lo que haremos, será dirigirnos a la pestaña que corresponde al **MainActivity.kt**, en donde vamos a generar la interacción al **buttonCalcular** que hemos asignado en el diseño, utilizaremos los **ID** para tener acceso a los objetos e invocamos al evento **onClickListener ()** y dentro de las llaves **{ }** le colocaremos la lógica de lo que sucederá cuando el usuario presione el botón.
  - Crearemos una variable llamada **nacimiento** de **tipo entero** y le asignaremos el valor contenido en el **EditText** en donde utilizaremos la función que convierte el **texto a número**, con la finalidad de recoger el dato que ha sido capturado por el usuario.
  - También declaramos una variable llamada **actual** y le asignamos la función **calendar**, con la finalidad de obtener la fecha de acuerdo al dispositivo
  - Declaramos otra variable con el nombre **edad**, la cual contendrá el resultado de la **resta entre el año actual y el año de nacimiento**
  - El **resultado** se lo pasaremos al **TextView** por medio de su atributo **text**, y enviando un mensaje referenciado a la variable **edad**

```
1 package com.example.intent2
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContentView(R.layout.activity_main)
9
10        //Declaramos las variables y utilizamos el identificador
11        val buttonCalcular: Button = findViewById(R.id.buttonCalcular)
12        val editTextEdad: EditText = findViewById(R.id.editTextEdad)
13        val textViewEdad: TextView = findViewById(R.id.textViewEdad)
14
15        //Asignaremos el evento onClickListener
16        buttonCalcular.setOnClickListener {
17            //Declarar las variables
18            val nacimiento: Int = editTextEdad.text.toString().toInt()
19            //Declaramos la variable y le asignamos la función calendario
20            val actual: Int = Calendar.getInstance().get(Calendar.YEAR)
21            //Declaramos la variable, en donde vamos a realizar el calculo
22            val edad: Int = actual-nacimiento
23
24            //Enviaremos a impresión, utilizando un TOAST
25            textViewEdad.text = "TUS AÑOS ACUMULADOS SON: $edad AÑOS"
26        }
27    }
28}
```

### ¿Cómo usar un click listener?

- <https://developer.android.com/guide/topics/ui/controls/button?hl=es-419>

## VISTA DE LA APP CON LA FUNCIONALIDAD DEL BOTÓN CALCULAR



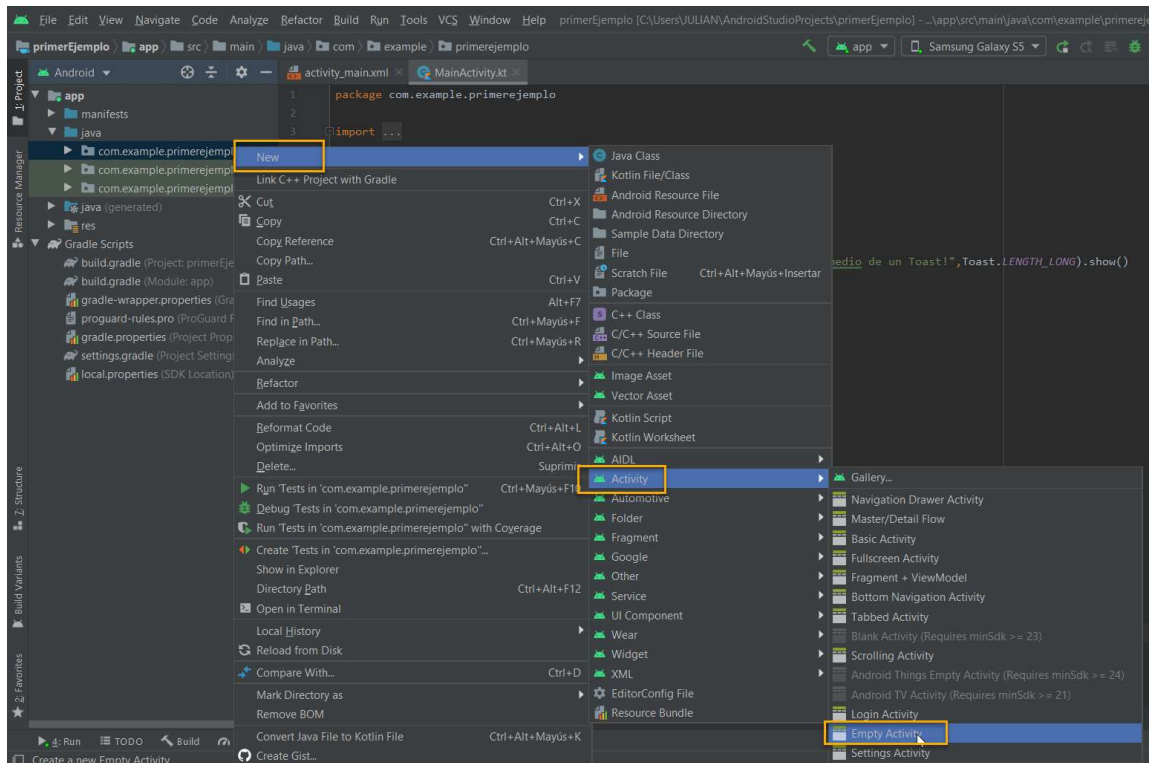
2. Lo siguiente que haremos, será **CREAR UNA FUNCIÓN** con el nombre **startActivity ( )**

- Pasándole como parámetros la variable con el nombre **activity** que va a ser el **objeto :Activity**
- Le pasamos un segundo parámetro que es **nextActivity** y le pasamos la **clase**.
  - Dentro de las llaves le vamos a indicar de que **activity** a que **activity** vamos a pasar,
    - Para hacerlo vamos a declarar una variable con el nombre **intent** y le asignamos la **clase Intent**, pasándole como argumentos **activity, nextActivity**
  - Iniciamos la **actividad** y le pasamos el objeto la cual tiene por nombre **intent**
  - Por último indicamos la finalización del activity
- También le declararemos una variable llamada **SALUDO** y le agregaremos un saludo, por medio de un string
- En la función agregaremos una línea de código en donde referenciamos al **intent.putExtra**, pasándole como **parámetros** un nombre que en este caso es el **saludo** y el nombre de la variable que contiene el saludo

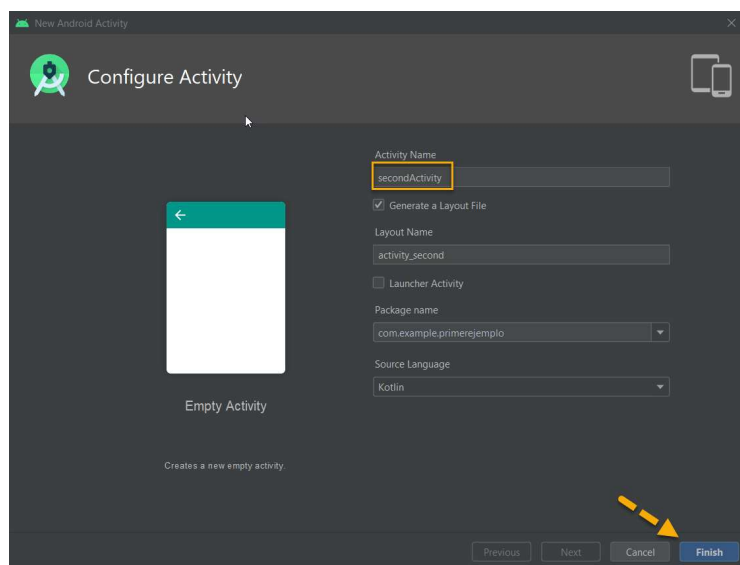
```
12 class MainActivity : AppCompatActivity() {
13
14     //Declaramos la variable saludo
15     val SALUDO = "Saludo desde el Activity Principal"
16
17     override fun onCreate(savedInstanceState: Bundle?) {
18         super.onCreate(savedInstanceState)
19         setContentView(R.layout.activity_main)
20
21         //Declaramos las variables y utilizamos el identificador
22         val buttonCalcular: Button = findViewById(R.id.buttonCalcular)
23         val editTextEdad: EditText = findViewById(R.id.editTextEdad)
24         val textViewEdad: TextView = findViewById(R.id.textViewEdad)
25
26         //Asignaremos el evento onClickListener
27         buttonCalcular.setOnClickListener {
28             //Declaramos las variables
29             val nacimiento: Int = editTextEdad.text.toString().toInt()
30             //Declaramos la variable y le asignamos la función calendario
31             val actual: Int = Calendar.getInstance().get(Calendar.YEAR)
32             //Declaramos la variable, en donde vamos a realizar el calculo
33             val edad: Int = actual - nacimiento
34
35             //Enviaremos a impresión, utilizando un TOAST
36             textViewEdad.text = "TUS AÑOS ACUMULADOS SON: $edad AÑOS"
37         }
38     }
39
40     //Declaramos la función, que activa el siguiente ACTIVITY
41     //Tienes que crear un segundo Activity, para poder construir esta función
42     fun startActivity(activity: Activity, nextActivity: Class<*>){
43         //Declaramos la variable para el INTENT
44         val intent = Intent(activity, nextActivity)
45
46         //Haremos uso de la variable, que contiene el saludo
47         intent.putExtra("saludo", SALUDO)
48
49         //Iniciamos la actividad
50         activity.startActivity(intent)
51
52         //Finalizar la actividad
53         activity.finish()
54     }
55 }
```



3. **CREAR UN SEGUNDO ACTIVITY**: Lo que haremos, será **crear un segundo activity**, y para hacerlo nos posicionaremos en la siguiente ruta **Java/com.example** y pulsaremos el botón derecho del mouse, al hacerlo nos mostrara un menú emergente desde donde seleccionaremos **New/ Activity/ Empty Activity**

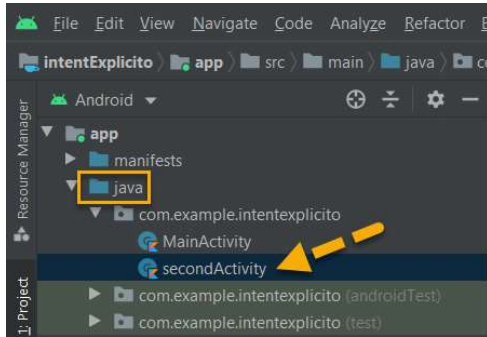


Establecemos el nombre **secondActivity** y pulsamos un click en **finalizar**





## LOGRANDO EL SIGUIENTE RESULTADO



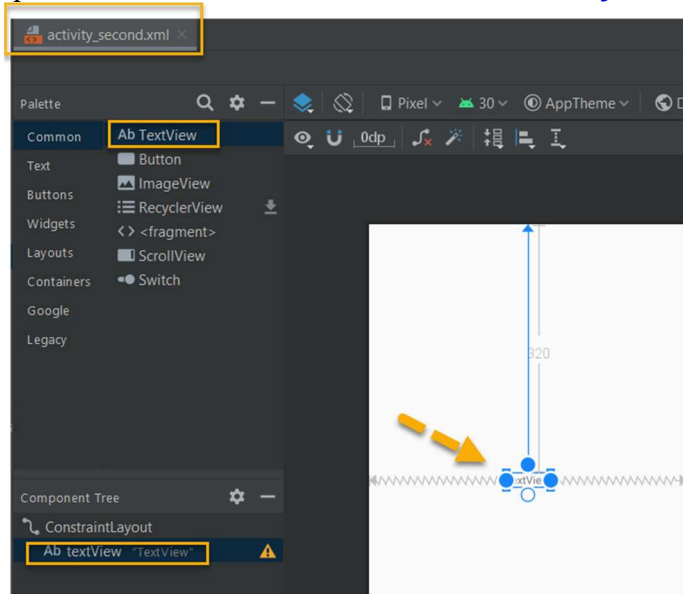
3. Ahora nos dirigiremos al **MainActivity.kt** e invocamos el evento **setOnClickListener**, para el **buttonSiguiete**

- Dentro de las llaves **{ }** invocamos el nombre de la **función** que acabamos de crear **startActivity()**
- En donde le pasaremos como parámetros el **this**, que hace referencia desde esta actividad continuara al **segundo activity**, y al final le pasamos una clase

```
16  
17 override fun onCreate(savedInstanceState: Bundle?) {  
18     super.onCreate(savedInstanceState)  
19     setContentView(R.layout.activity_main)  
20  
21     //Declaramos las variables y utilizamos el identificador  
22     val buttonCalcular: Button = findViewById(R.id.buttonCalcular)  
23     val editTextEdad: EditText = findViewById(R.id.editTextEdad)  
24     val textViewEdad: TextView = findViewById(R.id.textViewEdad)  
25  
26     1 //Declaramos la variable y utilizamos el identificador para el boton Siguiete  
27     val buttonSiguiete: Button = findViewById(R.id.buttonSiguiete)  
28  
29     2 //Asignaremos el evento onClickListener, para el boton siguiente  
30     buttonSiguiete.setOnClickListener { // View:  
31         //Invocaremos la función que permite iniciar el activity  
32         startActivity( activity: this,secondActivity::class.java)  
33     }  
34  
35     //Asignaremos el evento onClickListener  
36     buttonCalcular.setOnClickListener{ // View:  
37         //Declarar las variables  
38         val nacimiento: Int = editTextEdad.text.toString().toInt()  
39         //Declaramos la variable y le asignamos la función calendario  
40         val actual: Int = Calendar.getInstance().get(Calendar.YEAR)  
41         //Declaramos la variable, en donde vamos a realizar el calculo  
42         val edad: Int = actual-nacimiento  
43  
44         //Enviaremos a impresión, utilizando un TOAST  
45         textViewEdad.text = "TUS AÑOS ACUMULADOS SON: $edad AÑOS"  
46     }  
47  
48  
49  
50
```

## DISEÑO DEL SECOND ACTIVITY

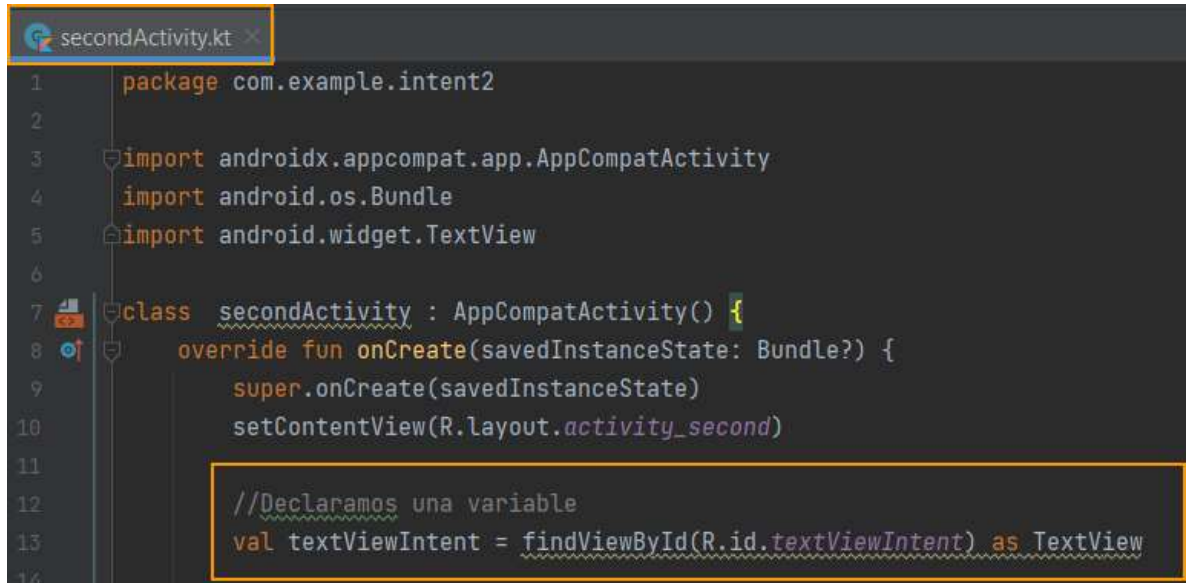
1. Lo que haremos, será dirigirnos al **activity\_second.xml**
  - y le agregaremos un **TextView** que es donde vamos a renderizar el mensaje que estamos enviando desde el **MainActivity**



- Le vamos a cambiar el **id** por **TextViewIntent** para poder identificarlo
- También le cambiaremos el tamaño de letra por un **tamaño 12**

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context=".secondActivity">
8
9   <TextView
10     android:id="@+id/textViewIntent"
11     android:layout_width="wrap_content"
12     android:layout_height="wrap_content"
13     android:text="TextView"
14     android:textSize="12dp"
15     app:layout_constraintBottom_toBottomOf="parent"
16     app:layout_constraintEnd_toEndOf="parent"
17     app:layout_constraintStart_toStartOf="parent"
18     app:layout_constraintTop_toTopOf="parent" />
19 </androidx.constraintlayout.widget.ConstraintLayout>
```

2. El siguiente paso, será dirigirte a la pestaña del **secondActivity.kt**
  - Declaramos una variable con el nombre **textView** y le asignamos la función **findViewById** con la finalidad de localizar el **textViewIntent** que creamos en el **activity\_second.xml**, estableciéndolo como texto.



```
1 package com.example.intent2
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.widget.TextView
6
7 class secondActivity : AppCompatActivity() {
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_second)
11
12        //Declaramos una variable
13        val textViewIntent = findViewById(R.id.textViewIntent) as TextView
14    }
15 }
```

3. Como siguiente paso creamos una **variable** con el nombre **bundle** a la cual le asignaremos la función **intent.extras**, con la finalidad de recoger los valores que están llegando al **activity**



```
7 class secondActivity : AppCompatActivity() {
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_second)
11
12        //Declaramos una variable
13        val textViewIntent = findViewById(R.id.textViewIntent) as TextView
14
15        //Creamos una segunda variable, a la cual le asignaremos el intent
16        val bundle = intent.extras
17    }
18 }
```

4. Lo siguiente es **hacer una comprobación** de que no está llegando vacío el **intent**, creamos una estructura condicional **IF** haciendo uso de la variable llamada **bundle** para validar que no esté nulo y además le pasamos el **Key** del **MainActivity** que para este caso fue **saludo**, y validamos que no esté nulo
- Dentro del cuerpo de la condición le pasaremos una variable con el nombre **saludo**, la cual recibe la función **bundle.getString** con el **Key**,
  - Le asignas al **textView** el contenido del variable **saludo**.
  - Para el caso que le llegue el mensaje vacío, colocamos un mensaje con un Toast

```
secondActivity.kt
3  import androidx.appcompat.app.AppCompatActivity
4  import android.os.Bundle
5  import android.widget.TextView
6  import android.widget.Toast
7
8  class secondActivity : AppCompatActivity() {
9      override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_second)
12
13         //Declaramos una variable
14         val textViewIntent = findViewById(R.id.textViewIntent) as TextView
15
16         //Creamos una segunda variable, a la cual le asignaremos el intent
17         val bundle = intent.extras
18
19         //Creamos la validación
20         if(bundle!=null && bundle.getString(key: "saludo")!=null){
21             //Declaramos una variable, la cual recoge el saludo
22             val saludo = bundle.getString(key: "saludo")
23             //Le asignamos el valor contenido en la variable saludo, al TextView, por medio de su atributo text
24             textViewIntent.text = saludo
25         }else{
26             //En caso de que llegue vacío el saludo, se lo haremos saber al usuario por medio de un TOAST
27             Toast.makeText(context: this, text: "Esta llegando vacío el mensaje", Toast.LENGTH_LONG).show()
28         }
29     }
30
31 }
32 }
```

## VISTA DEL ELMULADOR, AL MOMENTO DE PULSAR EL BOTON SIGUIENTE

