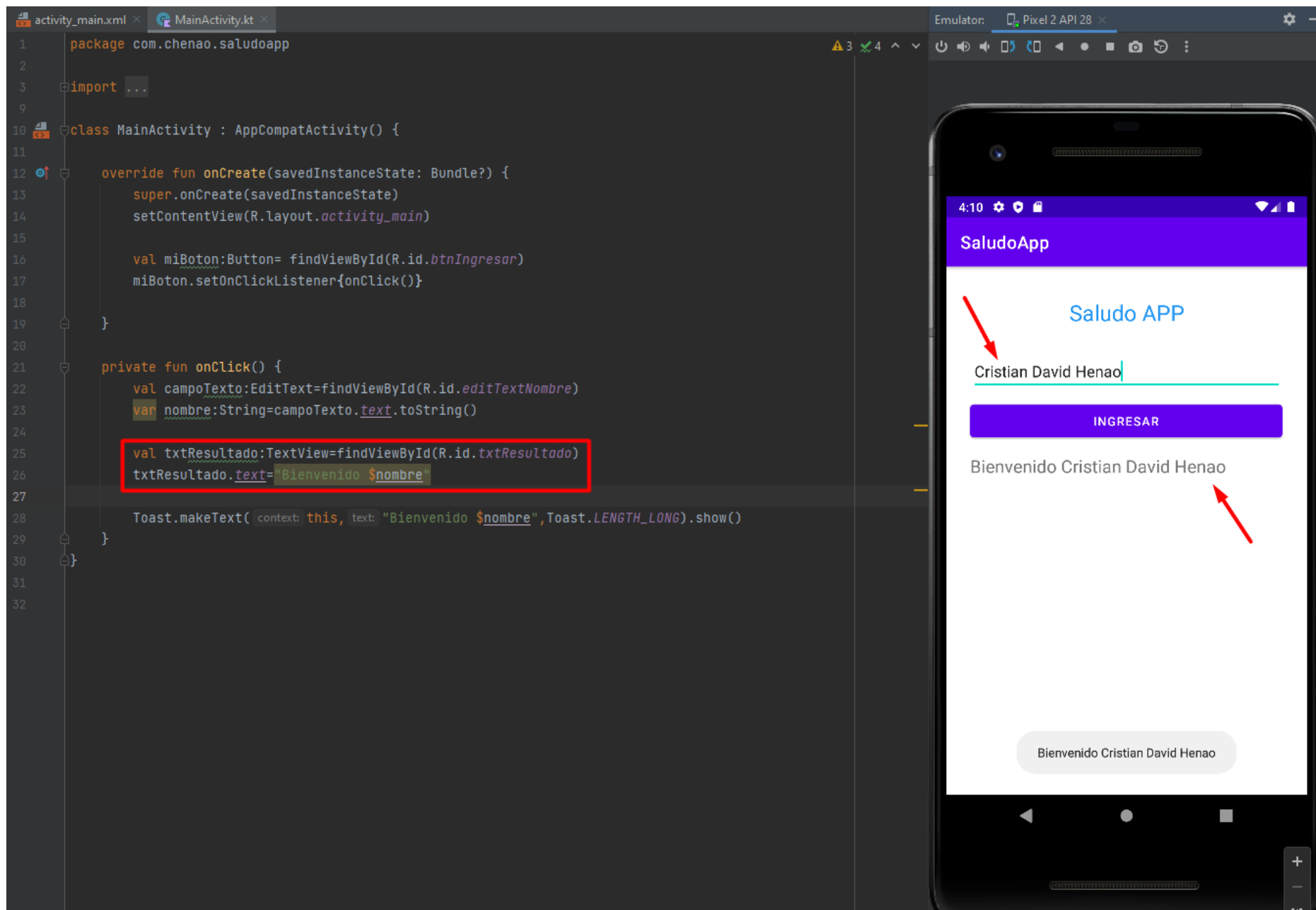




Navegar entre Activities

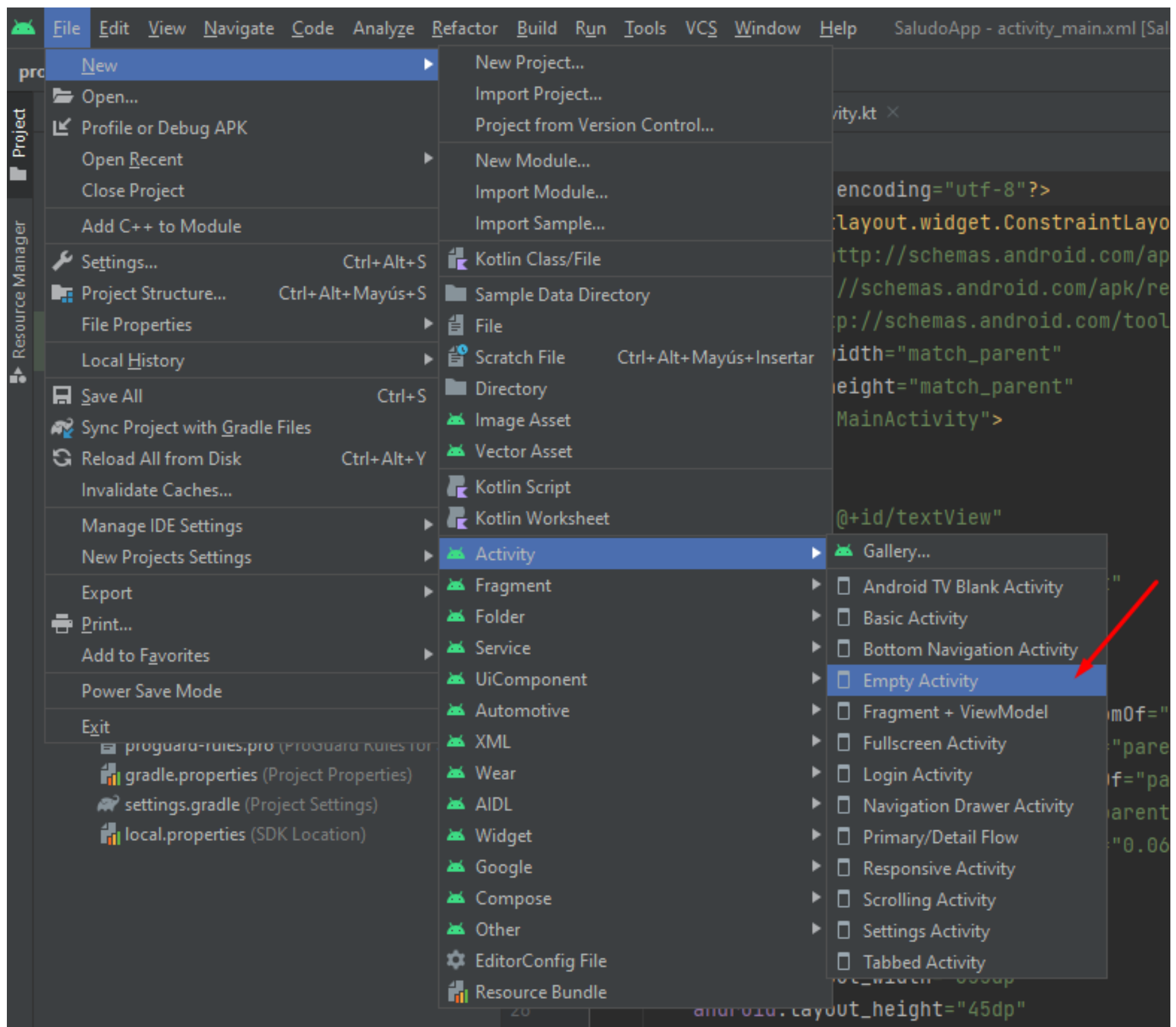
En guías anteriores vimos como crear un proyecto en Android Studio y generamos la siguiente aplicación



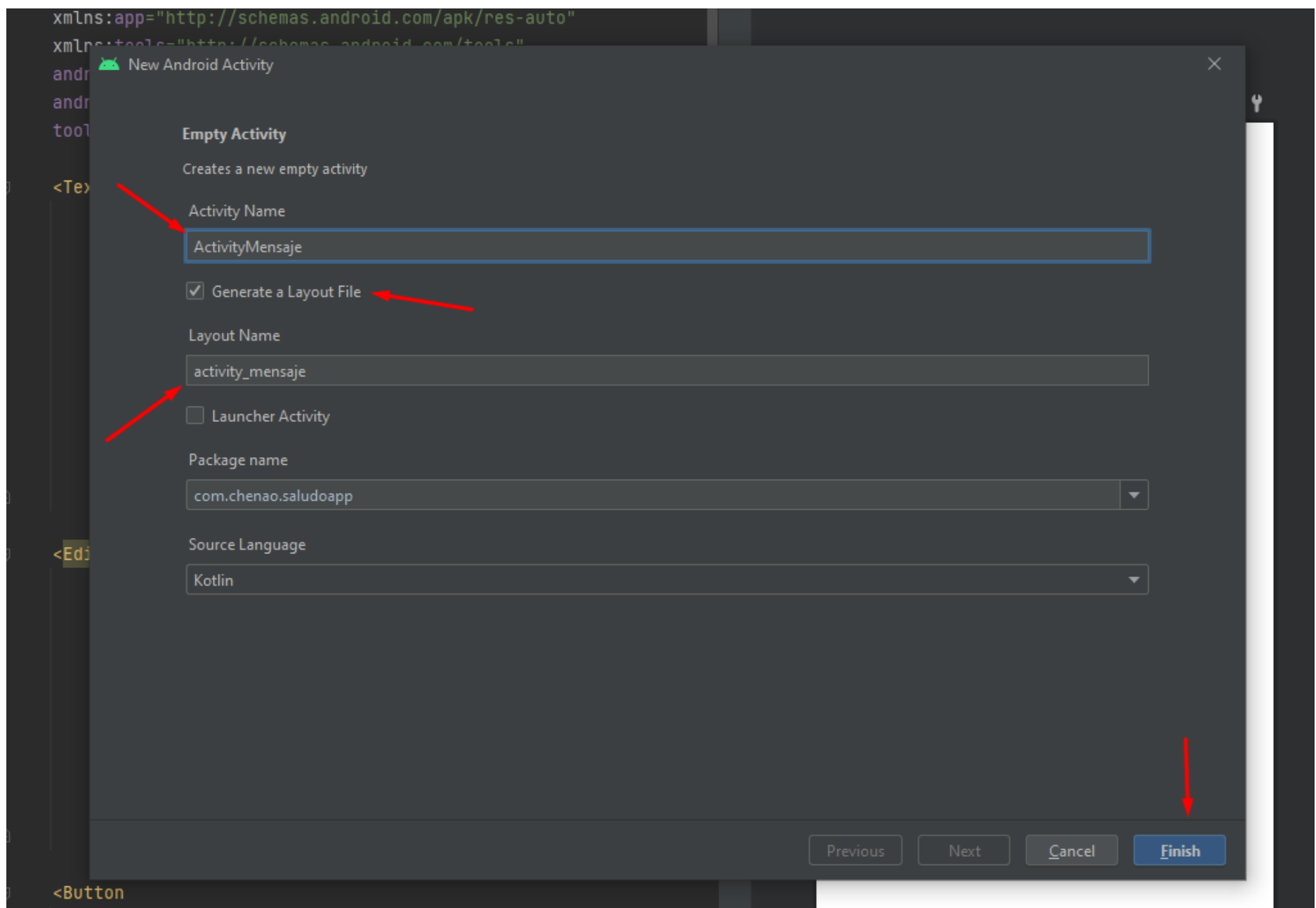
En ella lo único que hicimos era crear un Activity que nos solicitaba el ingreso de un nombre, al presionar el botón el sistema procesa la información ingresada y muestra en la misma pantalla el valor que pusimos en el campo correspondiente. En esta guía vamos a aumentar un poco la complejidad de nuestra aplicación haciendo que el valor ingresado sea transportado hacia otra Activity que nos procesará el valor y lo mostrará en pantalla.

Creación Nueva Activity

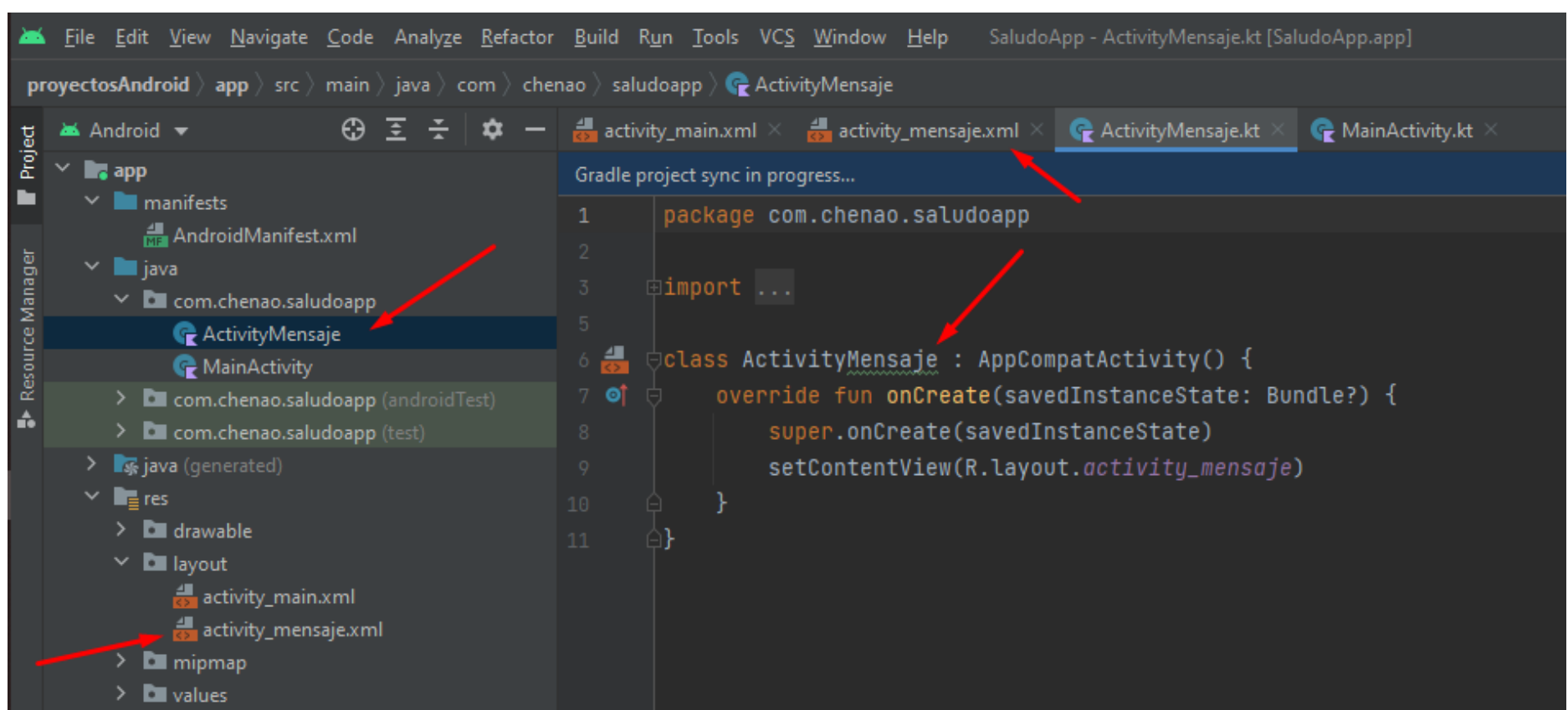
Para el desarrollo del ejercicio vamos a crear un nuevo activity con su respectivo layout, para eso vamos a *file/New/Activity/Empty Activity*



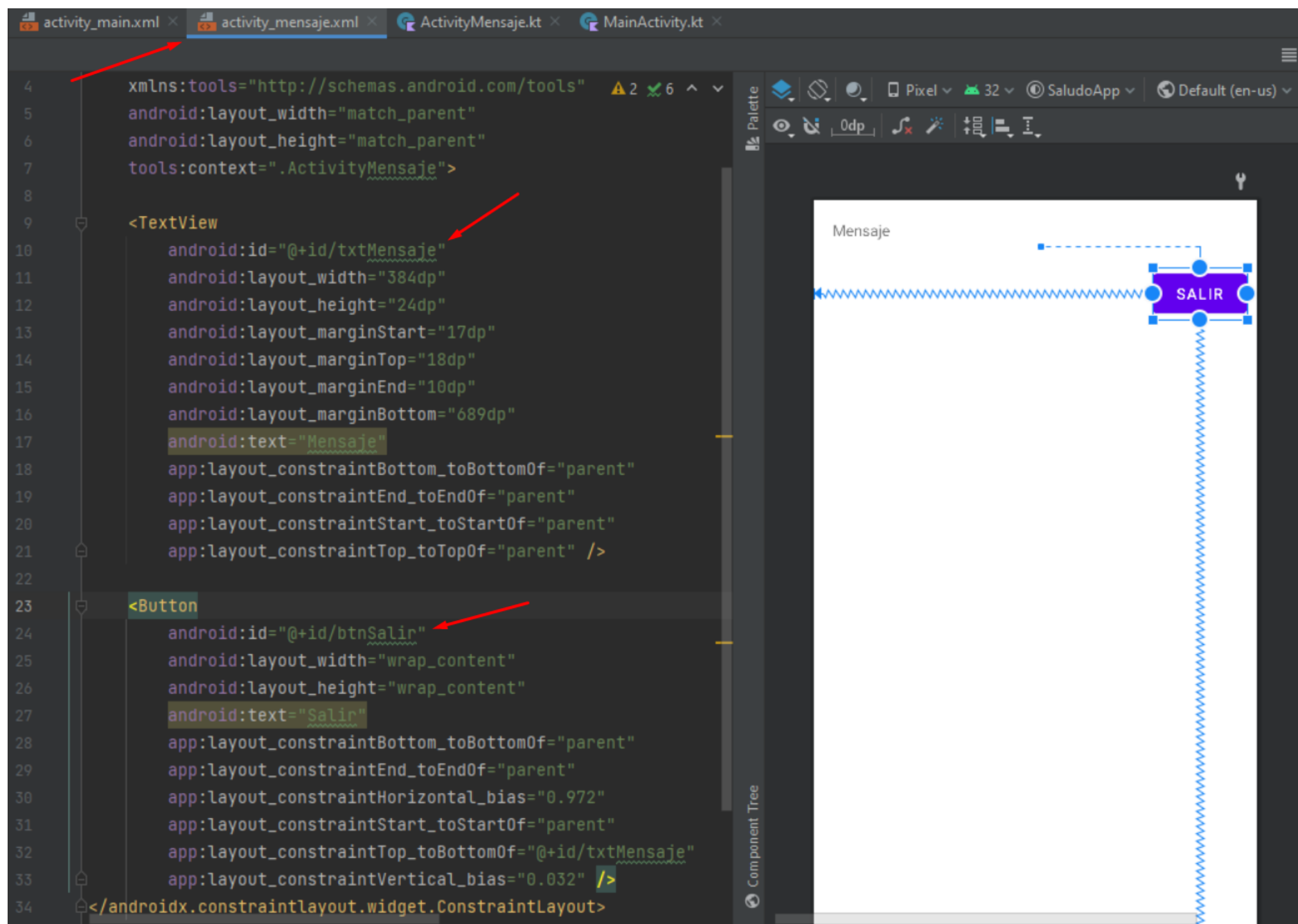
Al hacerlo se carga la siguiente ventana donde definimos el nombre y ruta de nuestra Actividad.



Al presionar finish vemos en el árbol de proyecto como se crearon la clase y layout respectivamente.

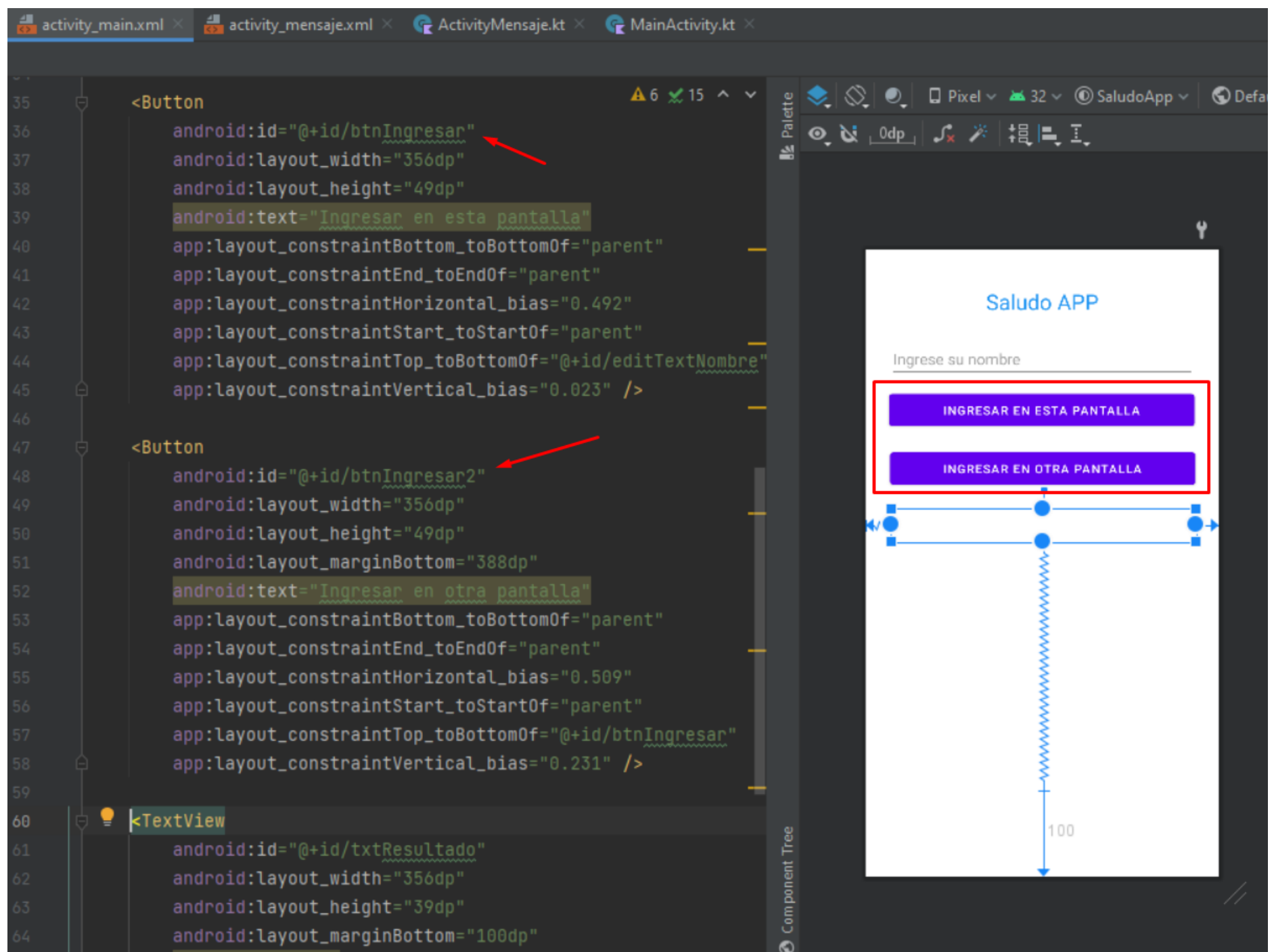


En el nuevo Activity agregaremos un textView y le ponemos un identificador, así como un botón Salir que nos permitirá cerrar esta actividad y regresar a la principal.



Modificación de la actividad principal.

Para este ejercicio vamos a agregar una nueva funcionalidad a la actividad principal, pero dejaremos la anterior tal como estaba, por esa razón vincularemos un nuevo botón que permita la comunicación.



Posteriormente definimos la lógica del nuevo botón en la actividad principal, para esto asignamos un parámetro al método `onClick()` el cual recibirá un entero dependiendo del botón que sea presionado (esto no es necesario ya que podríamos simplemente llamar otro método pero se realiza para reutilizar la misma función)


```

10
11 class MainActivity : AppCompatActivity() {
12     var campoTexto: EditText? = null
13     var txtResultado: TextView?=null
14
15     override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17         setContentView(R.layout.activity_main)
18
19         campoTexto=findViewById(R.id.editTextNombre)
20         txtResultado=findViewById(R.id.txtResultado)
21
22         val miBoton:Button= findViewById(R.id.btnIngresar)
23         miBoton.setOnClickListener{onClick( boton: 1)}
24
25         val miBoton2:Button= findViewById(R.id.btnIngresar2)
26         miBoton2.setOnClickListener{onClick( boton: 2)}
27     }
28
29     private fun onClick(boton: Int) {

```

Posteriormente agregamos la lógica del método onClick(Int), en este caso hacemos uso de la estructura when con el fin de determinar la lógica específica para cada botón.

```

25     val miBoton2:Button= findViewById(R.id.btnIngresar2)
26     miBoton2.setOnClickListener{onClick( boton: 2)}
27 }
28
29 private fun onClick(boton: Int) {
30     when(boton){
31         1 -> {
32             var nombre:String= campoTexto!!.text.toString()
33             txtResultado!!.text="Bienvenido $nombre"
34             Toast.makeText( context: this, text: "Bienvenido $nombre",Toast.LENGTH_LONG).show()
35         }
36         2 ->{
37             //Creamos el intent encargado de la comunicación entre actividades
38             val intent =Intent( packageContext: this,ActivityMensaje::class.java)
39             //creamos el bundle para el paso de información entre activies
40             val miBundle:Bundle= Bundle()
41             miBundle.putString("Nombre",campoTexto!!.text.toString())
42             //Le agregamos la información al intent para que sea enviada
43             intent.putExtras(miBundle)
44             //Iniciamos el llamado a la nueva actividad
45             startActivity(intent)
46         }
47     }
48 }

```

Como se puede ver se agrega la lógica necesaria para comunicarnos entre actividades y adicionalmente transportar un valor.

La comunicación entre los distintos componentes y aplicaciones en Android se realiza mediante intents, por esa razón es necesario crear un objeto de este tipo, los intent brindan diferentes alternativas para su uso, con ellos podemos cambiar de actividades, enviar valores o invocar el uso de otras aplicaciones dentro de nuestro sistema.

En este caso usaremos el intent para invocar otra actividad desde la principal, por esa razón al momento de su instancia desde su constructor enviamos la actividad desde la que se hace el llamado (this) y la actividad a la que queremos acceder (ActivityMensaje).

Hasta este punto podríamos solamente invocar la actividad mediante startActivity() enviando como parámetro el objeto intent creado lo que permitiría simplemente llamar la actividad, pero como queremos enviar un mensaje a la otra actividad debemos agregar más procesos.

Para el envío de datos usamos un objeto Bundle que puede contener una lista de “claves – valor” con los datos a transportar, en este caso enviamos el valor ingresado en el campo de texto usando la clave “Nombre”.

Al finalizar debemos entregarle el dato al intent mediante el método putExtras() que contiene el bundle creado y posteriormente invocar el método startActivity() con el intent y el valor almacenado.

Ahora lo que debemos hacer es obtener la información y mostrarla en la nueva actividad

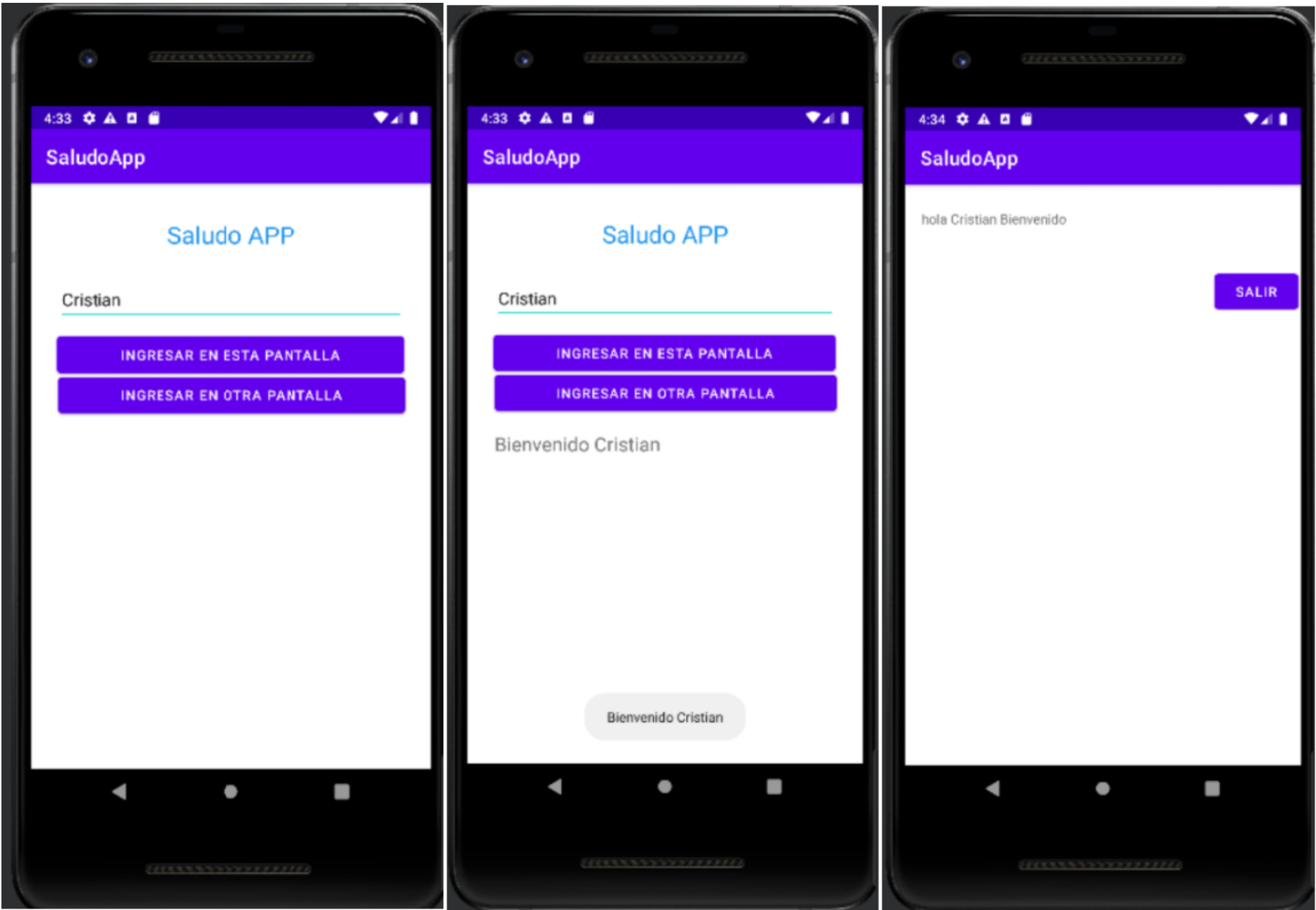
```
1 package com.chenao.saludoapp
2
3 import ...
4
5
6
7
8 class ActivityMensaje : AppCompatActivity() {
9
10 override fun onCreate(savedInstanceState: Bundle?) {
11     super.onCreate(savedInstanceState)
12     setContentView(R.layout.activity_mensaje)
13
14     val campoMensaje=findViewById<TextView>(R.id.txtMensaje)
15
16     //Recuperamos la información que llega en el intent
17     var miBundle: Bundle? =this.intent.extras
18
19     //Se asigna la información obtenida al textView
20     if (miBundle != null) {
21         campoMensaje.text="hola ${miBundle.getString( key: "Nombre")} Bienvenido"
22     }
23
24     val botonSalir:Button=findViewById(R.id.btnSalir)
25     botonSalir.setOnClickListener { onClick() }
26
27 }
28
29 private fun onClick() {
30     finish()
31 }
32 }
```

Luego de haber realizado el envío, desde la actividad receptora debemos crear otro objeto Bundle que se encargara de obtener el valor enviado y luego mediante el nombre del valor lo podemos asignar al text view de nuestra pantalla.

Notemos que lo hacemos gracias al bundle creado y debemos asociarlo directamente al nombre del parámetro que asociamos en el activity principal, en este caso lo llamamos “Nombre”

También agregamos la lógica del evento del botón, en este caso invocamos al método finish() que se encarga de cerrar el activity

actual.
Ahora al correr la aplicación tenemos las dos opciones, mostrar el mensaje en la misma actividad o mostrarlo en otra.



Actividad.

- 1. Modifique la aplicación sobre los componentes gráficos permitiendo que al momento de ingresar los datos se muestre una nueva pantalla con los resultados obtenidos.
- 2. Cree una aplicación que permita ingresar el nombre de un estudiante, una materia y 3 notas parciales, la app debe tener un botón calcular que al momento de presionarlo permita cargar una pantalla con la información del estudiante, las notas ingresadas, el promedio obtenido y el estado del mismo si aprobó o reprobó la materia, en caso de haber aprobado se debe presentar el texto del estado de color Verde, si reprueba se debe presentar el texto de color Rojo, tenga en cuenta que se aprueba con un promedio superior o igual a 3.5

Instructor: Cristian David Henao Hoyos