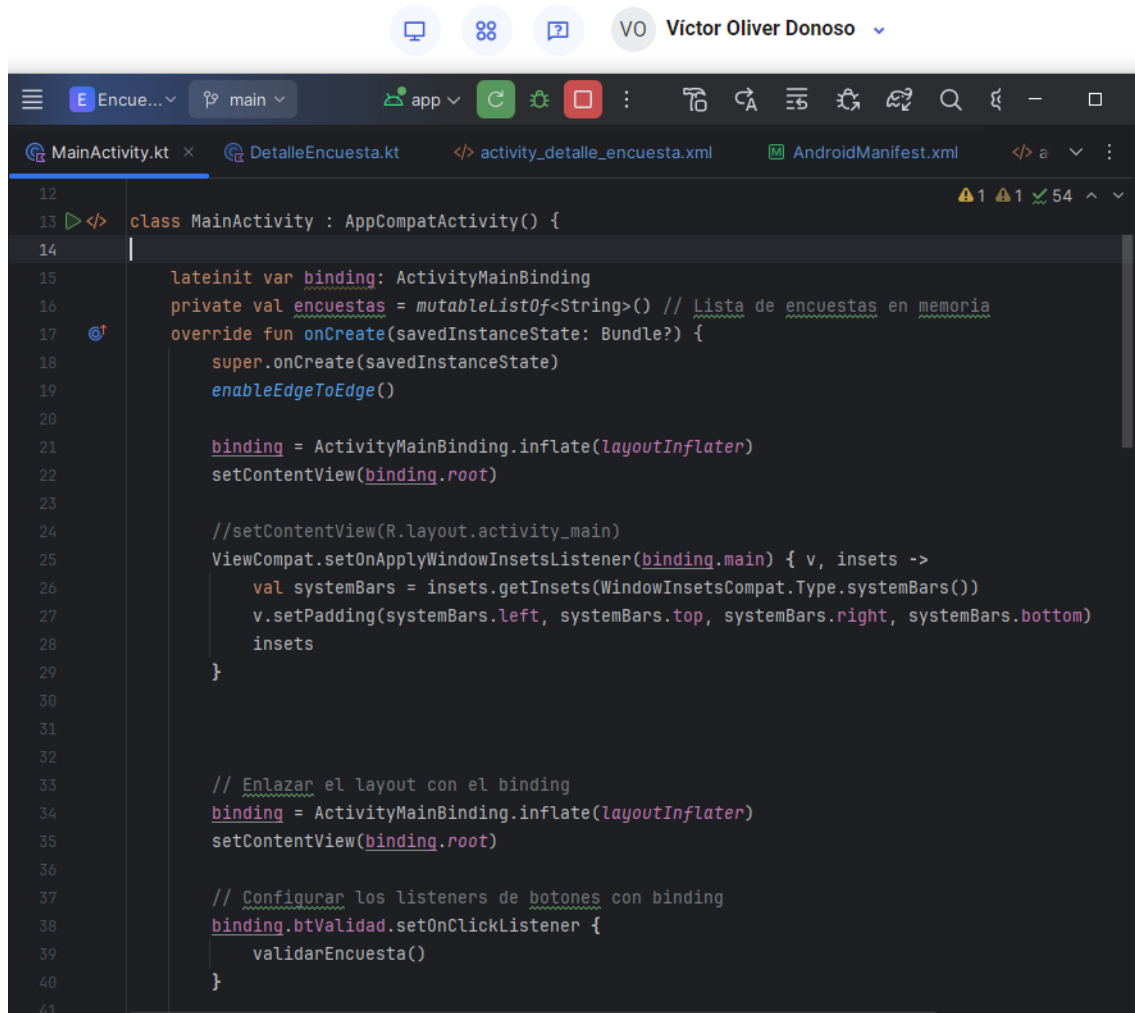


ENCUESTA

Pondré comentarios para facilitar el entendimiento.

El main



```
12
13 <?> class MainActivity : AppCompatActivity() {
14
15     lateinit var binding: ActivityMainBinding
16     private val encuestas = mutableListOf<String>() // lista de encuestas en memoria
17     override fun onCreate(savedInstanceState: Bundle?) {
18         super.onCreate(savedInstanceState)
19         enableEdgeToEdge()
20
21         binding = ActivityMainBinding.inflate(layoutInflater)
22         setContentView(binding.root)
23
24         //setContentView(R.layout.activity_main)
25         ViewCompat.setOnApplyWindowInsetsListener(binding.main) { v, insets ->
26             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
27             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
28             insets
29         }
30
31
32
33         // Enlazar el layout con el binding
34         binding = ActivityMainBinding.inflate(layoutInflater)
35         setContentView(binding.root)
36
37         // Configurar los listeners de botones con binding
38         binding.btValidad.setOnClickListener {
39             validarEncuesta()
40         }
41     }
```

VO Víctor Oliver Donoso

Encue... main ... app ...

MainActivity.kt x DetalleEncuesta.kt activity_detalle_encuesta.xml AndroidManifest.xml activity_main.xml

```
13 class MainActivity : AppCompatActivity() {
17     override fun onCreate(savedInstanceState: Bundle?) {
41
42         binding.btReiniciar.setOnClickListener {
43             reiniciarEncuestas()
44         }
45
46         binding.btCuantas.setOnClickListener {
47             mostrarCantidadEncuestas()
48         }
49
50         binding.btResumen.setOnClickListener {
51             mostrarResumenEncuestas()
52         }
53     }
54
55
56
57
58
59
60     private fun validarEncuesta() {
61         // Validar datos de la encuesta
62         val nombre = if (binding.swAnonimo.isChecked) "Anónimo" else binding.etTuNombre.text.toString()
63         val horasEstudio = binding.sbHorasDeEstudio.progress
64         val especialidad = getEspecialidadSeleccionada()
65         val sistemaOperativo = getSistemaOperativoSeleccionado()
66
67         if (especialidad == null || sistemaOperativo == null) {
68             Toast.makeText(context: this, text: "Faltan datos por completar", Toast.LENGTH_SHORT).show()
69         }
70     }
71 }
```

Problems File 56 Project Errors

VO Víctor Oliver Donoso

```
13 class MainActivity : AppCompatActivity() {
60     private fun validarEncuesta() {
69         return
70     }
71
72     // Guardar la encuesta en la lista
73     val encuesta = "Nombre: $nombre, Horas: $horasEstudio, Especialidad: $especialidad, SO: $sistemaOperativ
74     encuestas.add(encuesta)
75     Toast.makeText(context: this, text: "Encuesta guardada", Toast.LENGTH_SHORT).show()
76
77     // Aquí puedes abrir un segundo activity para mostrar la encuesta
78
79     val intent = Intent(packageContext: this, DetalleEncuesta::class.java)
80     intent.putExtra(name: "ENCUESTA_DETALLE", encuesta)
81     startActivity(intent)
82 }
83
84 private fun reiniciarEncuestas() {
85     encuestas.clear()
86     Toast.makeText(context: this, text: "Encuestas reiniciadas", Toast.LENGTH_SHORT).show()
87 }
88
89 private fun mostrarCantidadEncuestas() {
90     Toast.makeText(context: this, text: "Total encuestas: ${encuestas.size}", Toast.LENGTH_SHORT).show()
91 }
92
93 private fun mostrarResumenEncuestas() {
94     val resumen = encuestas.joinToString(separator: "\n")
95     binding.textView2.text = resumen
96 }
```

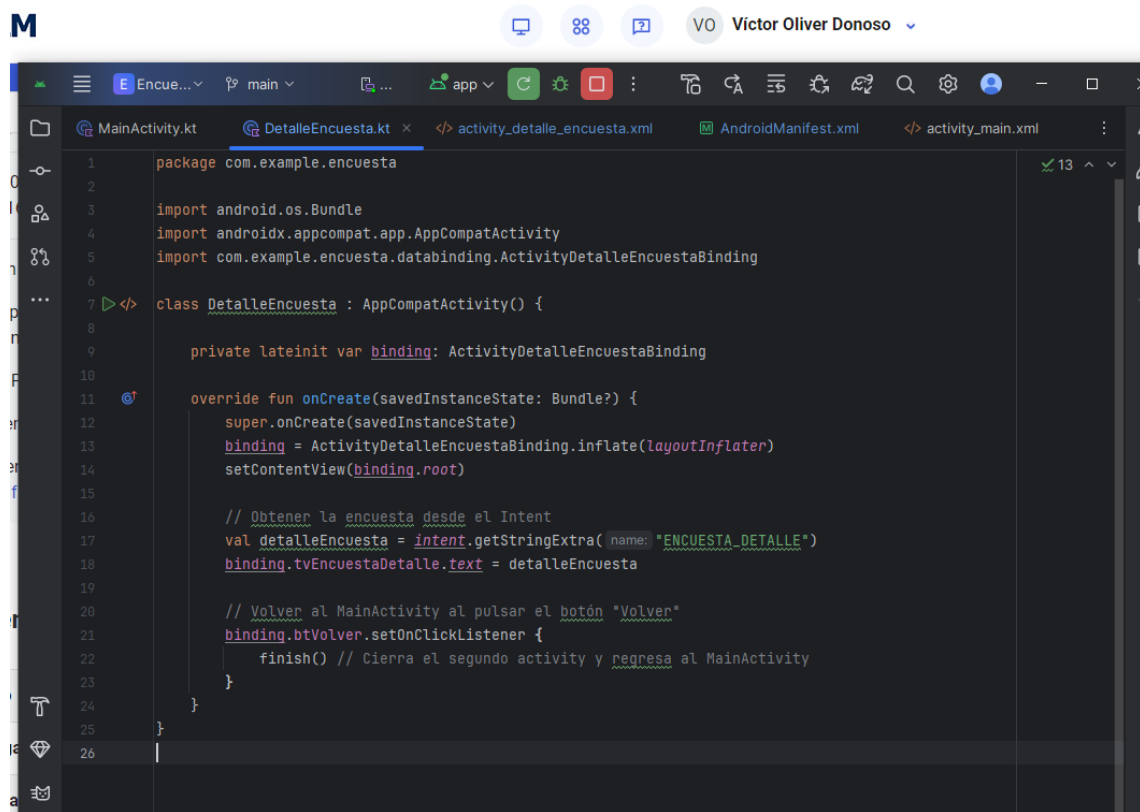
Problems File 56 Project Errors

VO Víctor Oliver Donoso

```
13 class MainActivity : AppCompatActivity() {
93     private fun mostrarResumenEncuestas() {
94         val resumen = encuestas.joinToString(separator: "\n")
95         binding.textView2.text = resumen
96     }
97
98     private fun getEspecialidadSeleccionada(): String? {
99         return when {
100             binding.cbDAM.isChecked -> "DAM"
101             binding.cbDAW.isChecked -> "DAW"
102             binding.cbASIR.isChecked -> "ASIR"
103             else -> null
104         }
105     }
106
107     private fun getSistemaOperativoSeleccionado(): String? {
108         return when {
109             binding.rbWindows.isChecked -> "Windows"
110             binding.rbMac.isChecked -> "Mac"
111             binding.rbLinux.isChecked -> "Linux"
112             else -> null
113         }
114     }
115 }
```

El detalle

M



The screenshot shows the Android Studio IDE with the following elements:

- Top Bar:** Includes icons for a monitor, a grid, a document, and a user profile labeled "VO Victor Oliver Donoso".
- Tab Bar:** Shows four open files: MainActivity.kt, DetalleEncuesta.kt (selected), activity_detalle_encuesta.xml, and AndroidManifest.xml. A fifth tab, activity_main.xml, is also visible.
- Code Editor:** Displays the Kotlin code for the DetalleEncuesta class. The code is as follows:

```
1 package com.example.encuesta
2
3 import android.os.Bundle
4 import androidx.appcompat.app.AppCompatActivity
5 import com.example.encuesta.databinding.ActivityDetalleEncuestaBinding
6
7 class DetalleEncuesta : AppCompatActivity() {
8
9     private lateinit var binding: ActivityDetalleEncuestaBinding
10
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         binding = ActivityDetalleEncuestaBinding.inflate(layoutInflater)
14         setContentView(binding.root)
15
16         // Obtener la encuesta desde el Intent
17         val detalleEncuesta = intent.getStringExtra("ENCUESTA_DETALLE")
18         binding.tvEncuestaDetalle.text = detalleEncuesta
19
20         // Volver al MainActivity al pulsar el botón "Volver"
21         binding.btVolver.setOnClickListener {
22             finish() // Cierra el segundo activity y regresa al MainActivity
23         }
24     }
25 }
26
```
- Left Sidebar:** Contains icons for the Project, Resource, and Run toolbars.
- Right Sidebar:** Shows a green checkmark and the number "13", indicating 13 lines of code.

Funcionalidad

