Esmeraldas 18 de diciembre del 2025

Desarrollo Móvil

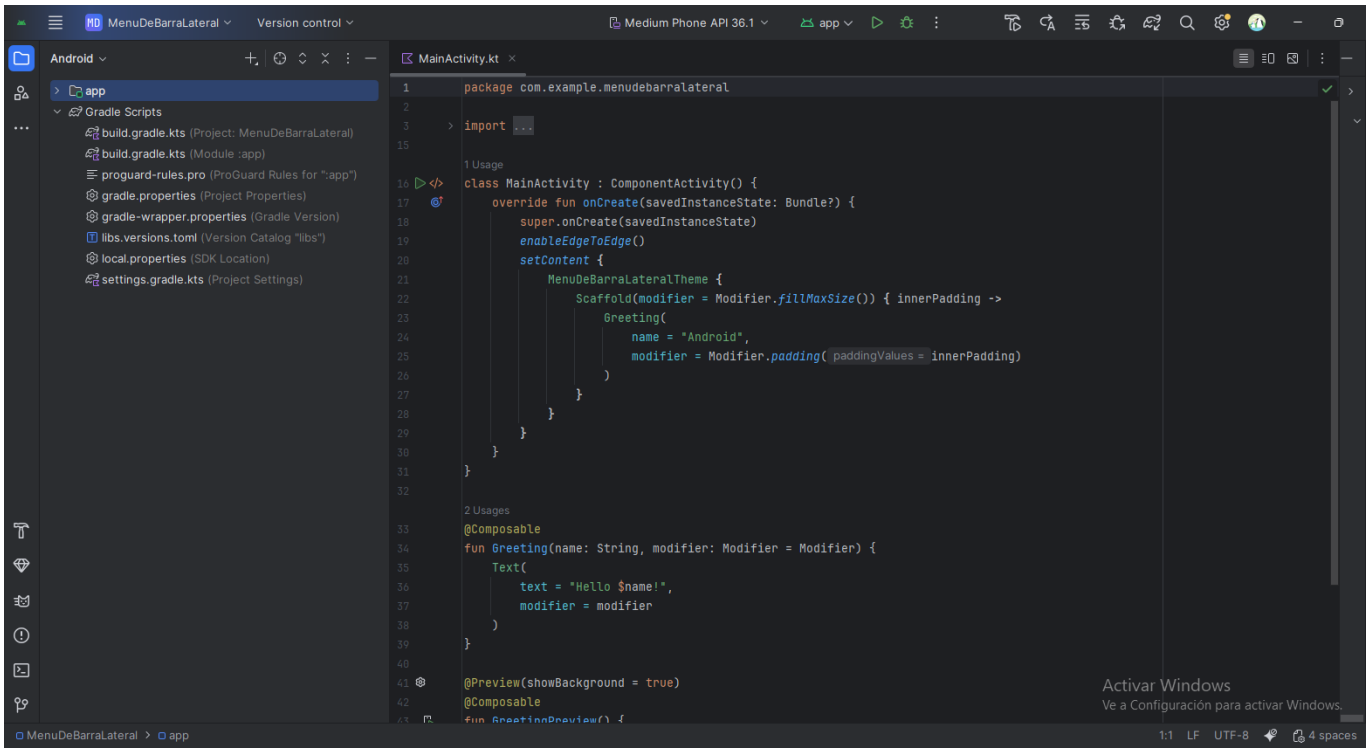
nombres: Zambrano palacios Jandry Steven

Maestro: Kleber Posligua

Tarea 2:

Menú Barra lateral con Jetpack Compose

Primer paso de la tarea es que se debe crear el proyecto menú barra lateral y ubicar en el proyecto en Empy Activity esperar a que cargue

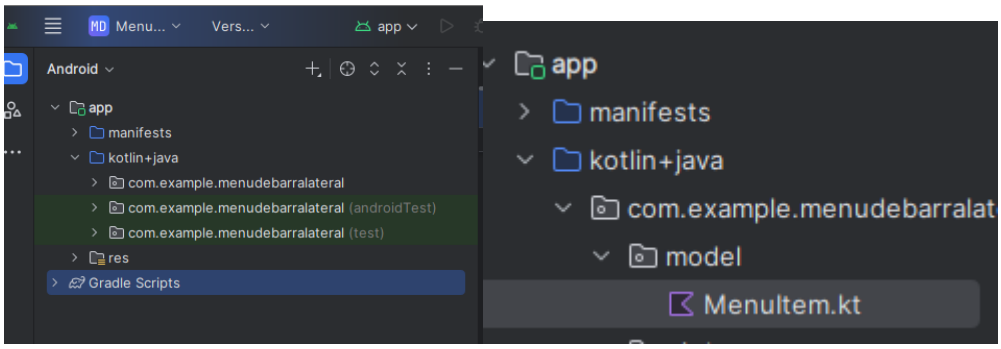


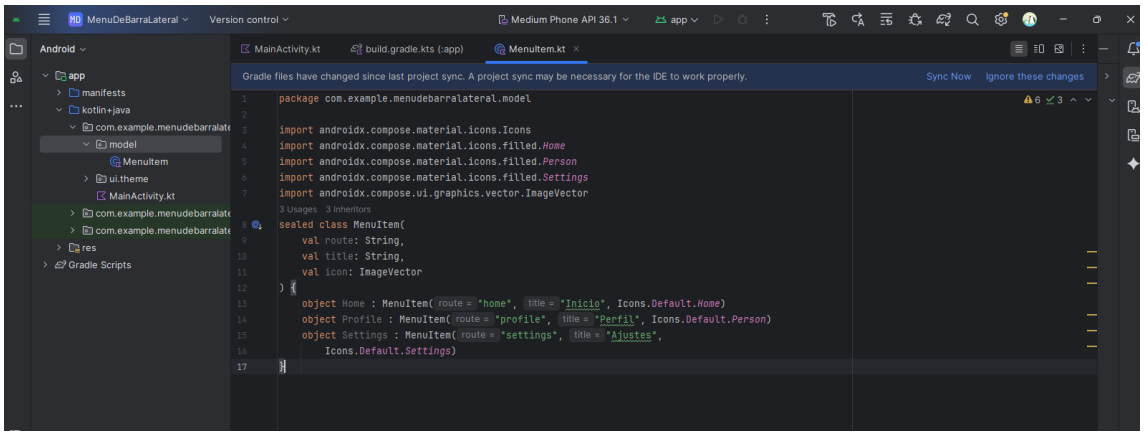
2 paso cuando está el proyecto creado tenemos que ir a buil.gradle: modulo app y ubicar las dependencias

implementation("androidx.navigation:navigation-compose:2.7.7") //o versión más reciente

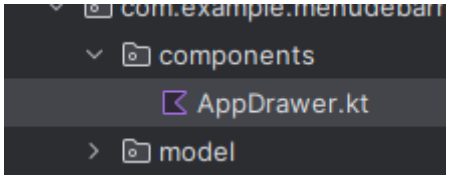
implementation("androidx.compose.material:material-icons-extended:1.6.0")

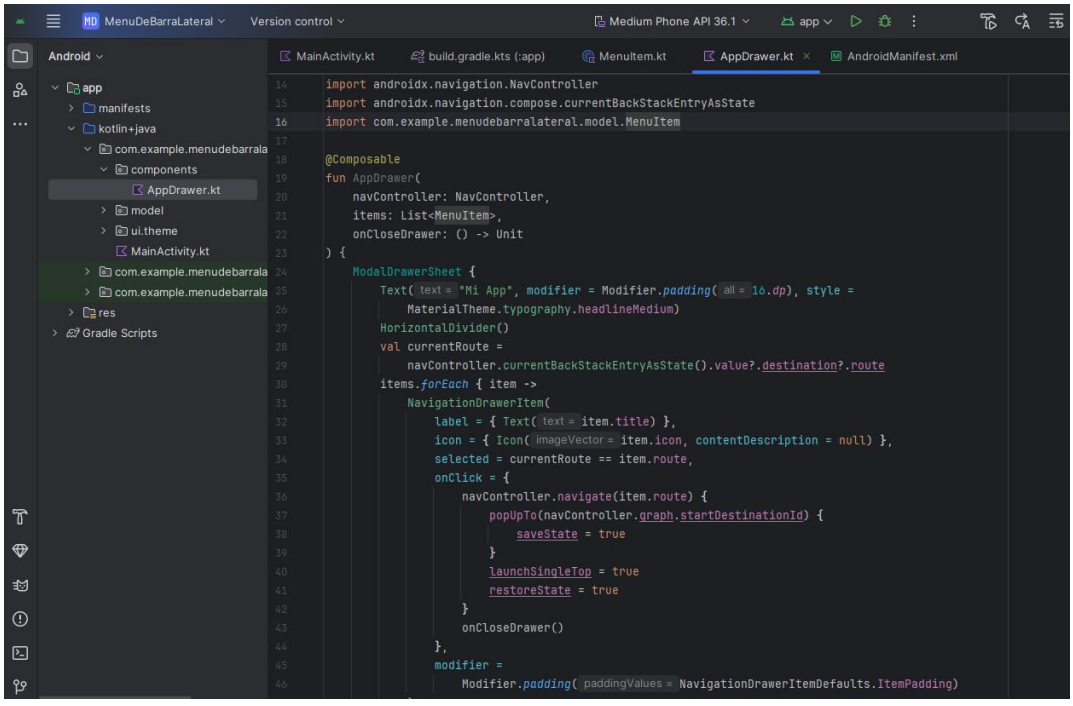
paso 3 crear un nuevo paquete llamado: model en class kotlin class file MenuItem

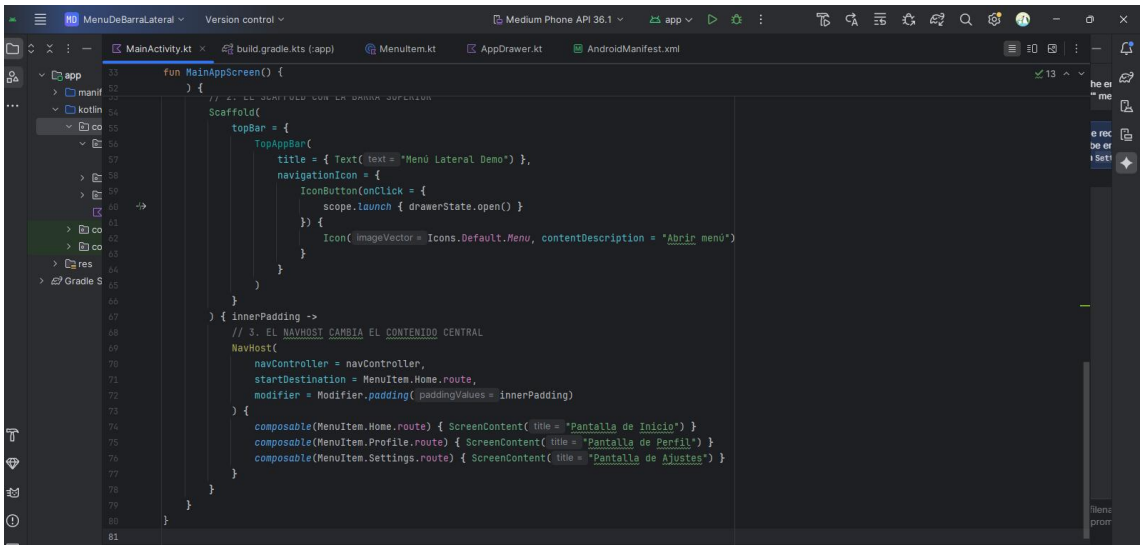




Paso 4 crear un package Compose y crear un kotlin poniendo nombre de AppDrawer.kt

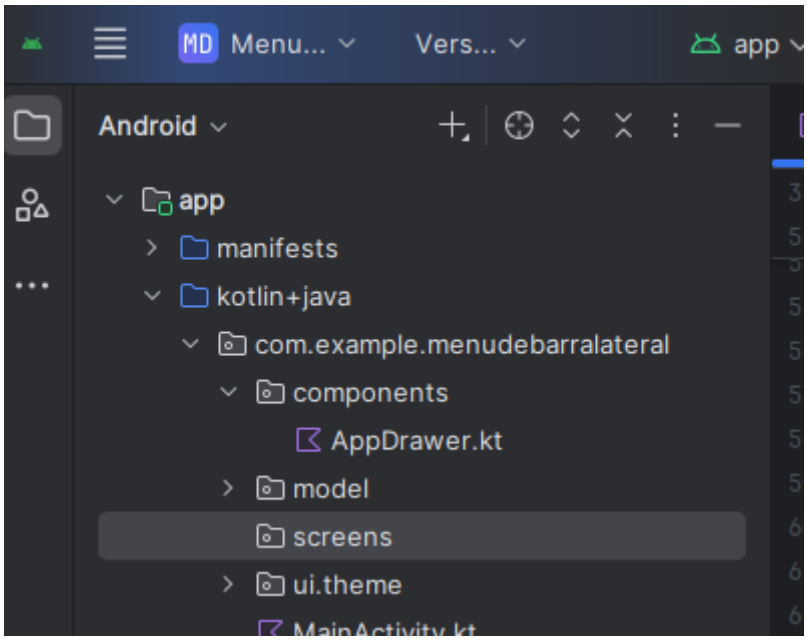


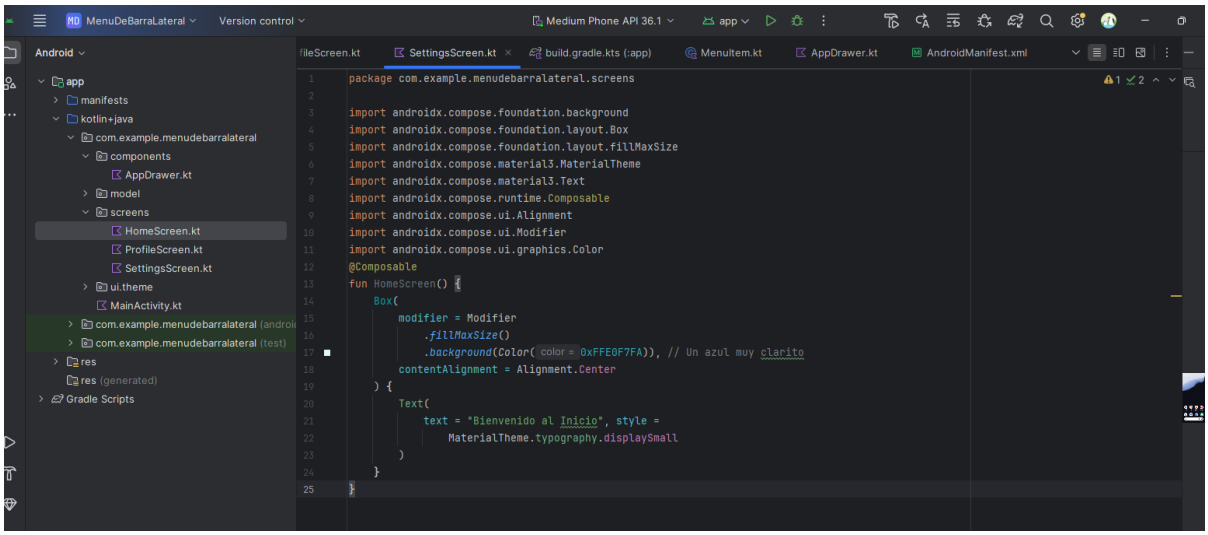


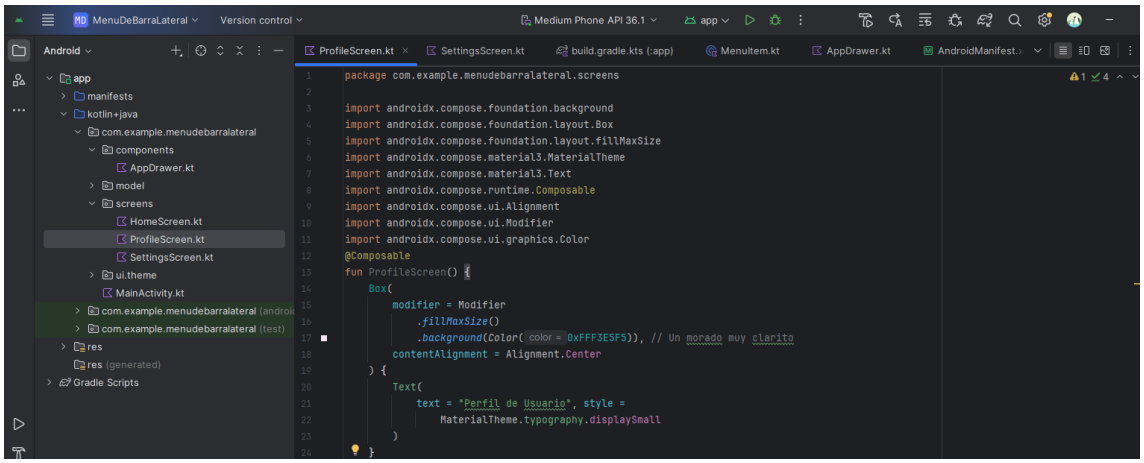

MenuDeBarraLateral
Version control
Medium Phone API 36.1
app
MainActivity.kt
build.gradle.kts (:app)
MenuItem.kt
AppDrawer.kt
AndroidManifest.xml
fun MainAppScreen() {
) {
Scaffold(
topBar = {
TopAppBar(
title = { Text(text = "Menú Lateral Demo") },
navigationIcon = {
IconButton(onClick = {
scope.launch { drawerState.open() }
}) {
Icon(imageVector = Icons.Default.Menu, contentDescription = "Abrir menú")
}
}
)
}
) { innerPadding ->
// 3. EL NAVHOST CAMBIA EL CONTENIDO CENTRAL
NavHost(
navController = navController,
startDestination = MenuItem.Home.route,
modifier = Modifier.padding(paddingValues = innerPadding)
) {
composable(MenuItem.Home.route) { ScreenContent(title = "Pantalla de Inicio") }
composable(MenuItem.Profile.route) { ScreenContent(title = "Pantalla de Perfil") }
composable(MenuItem.Settings.route) { ScreenContent(title = "Pantalla de Ajustes") }
}
}
}
}

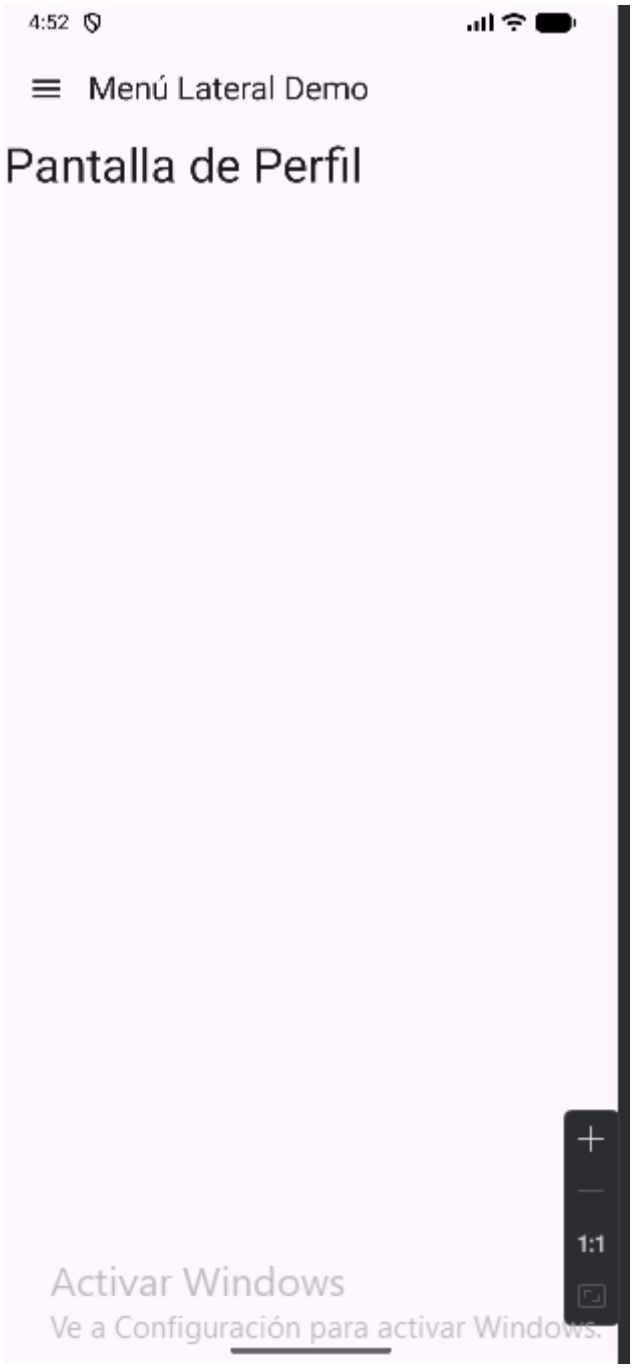

4:36
Menú Lateral Demo
Pantalla de Inicio

Paso 5 creamos un package Screens

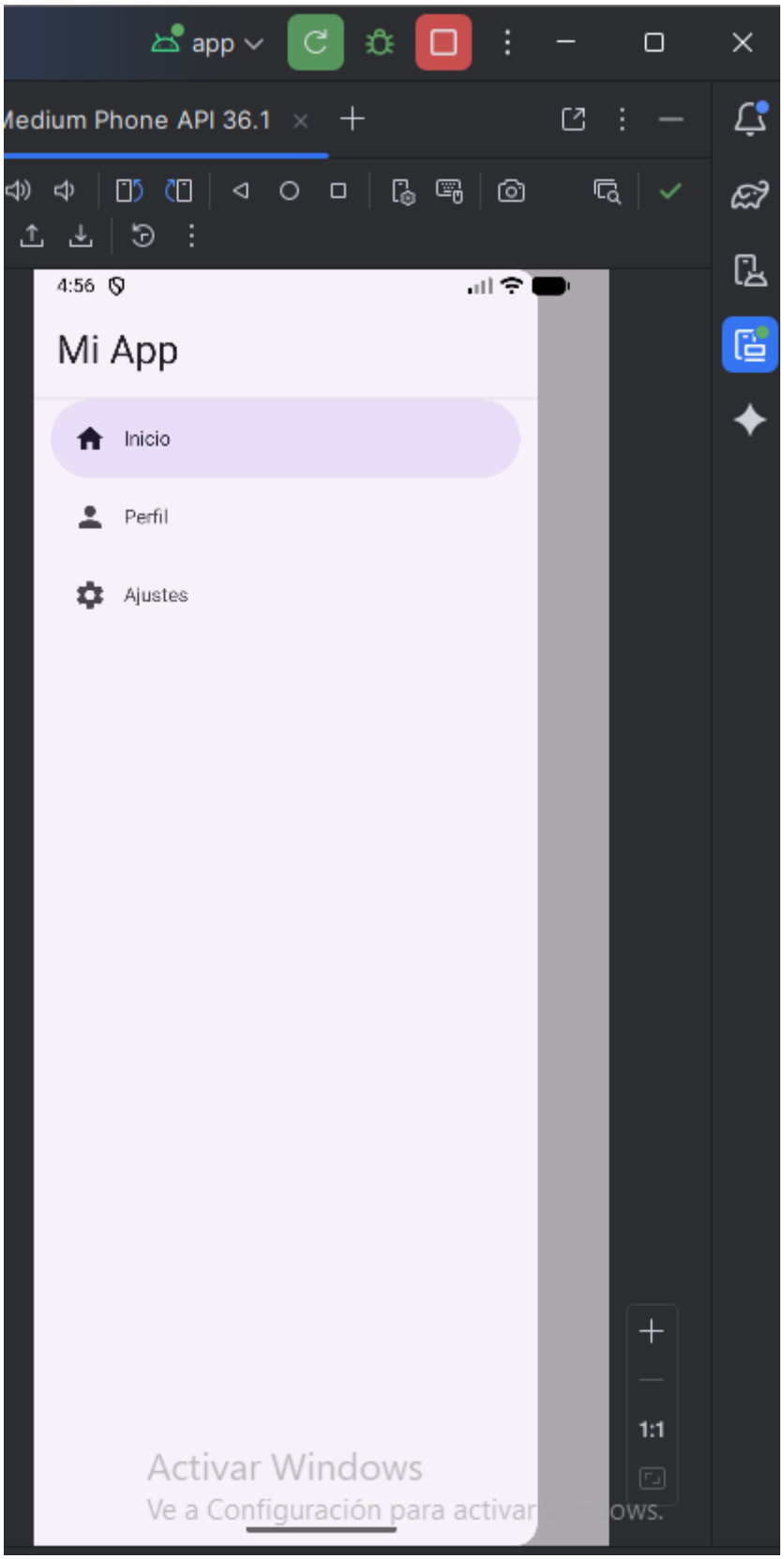







4:52
Menú Lateral Demo
Pantalla de Perfil
1:1
Activar Windows
Ve a Configuración para activar Windows.

Una **Sealed Class** (clase sellada) es una clase que restringe qué otras clases pueden heredarla, definiendo un conjunto **cerrado y limitado de subclases conocidas en tiempo de compilación**. Se recomienda usarla cuando necesitas representar un conjunto fijo y predecible de estados o tipos relacionados, como estados de UI (cargando, éxito, error), tipos de mensajes o resultados de operaciones, para **asegurar el manejo exhaustivo** de todos los casos posibles y evitar extensiones no deseadas, siendo una alternativa más potente que los enum.


app
Medium Phone API 36.1
4:56
Mi App
Inicio
Perfil
Ajustes
1:1
Activar Windows
Ve a Configuración para activar