

Esmeraldas20 de noviembre del 2025

Nombres: Jandry Steven Zambrano Palacios

Carrera: Desarrollo de Software

Tema: Listas Mutables en Android Studio

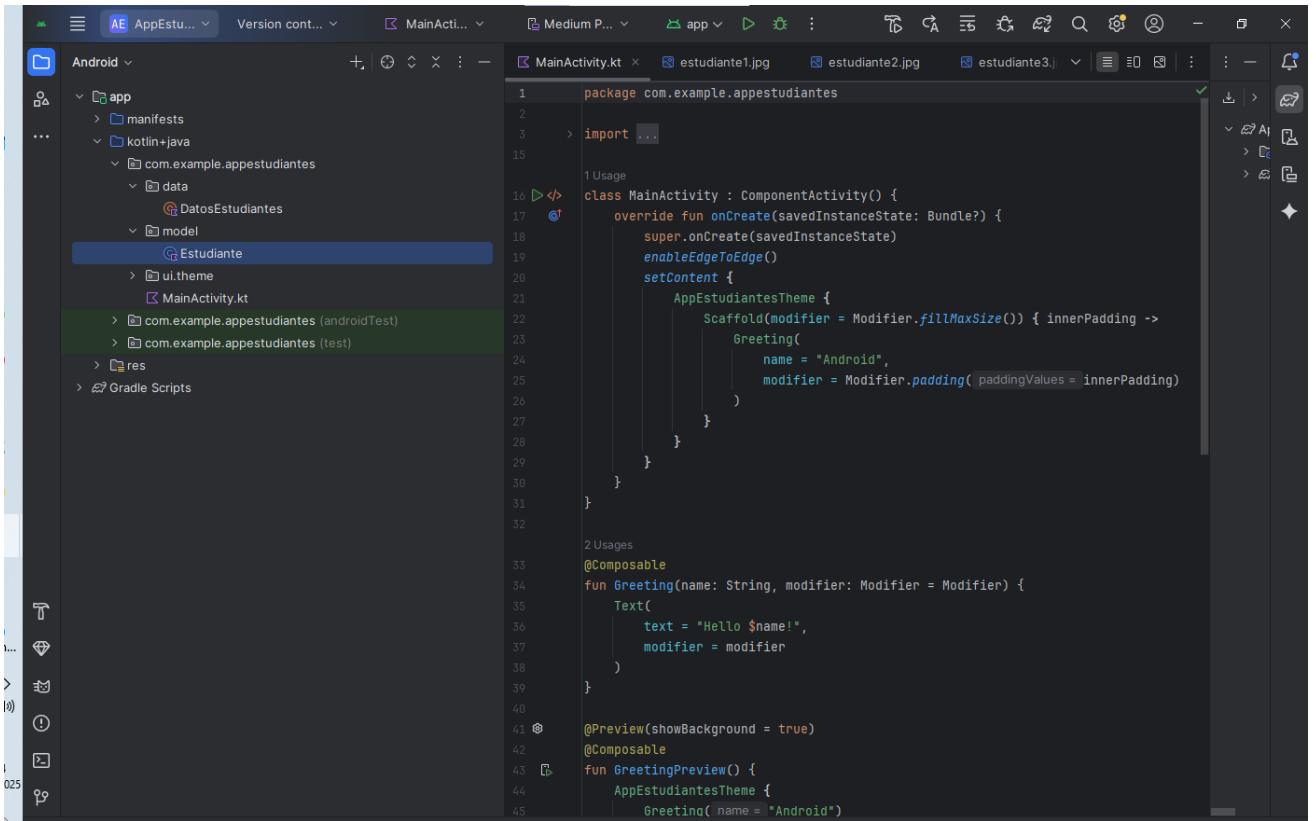
Introducción.

Este informe documenta el desarrollo de una aplicación móvil

Android utilizando Jetpack Compose y Kotlin. La aplicación permite gestionar un listado de estudiantes mediante dos funcionalidades principales:

Visualización de estudiantes en tarjetas (Cards) y registro de nuevos estudiantes a través de un formulario

La estructura de los paquetes estructurados.

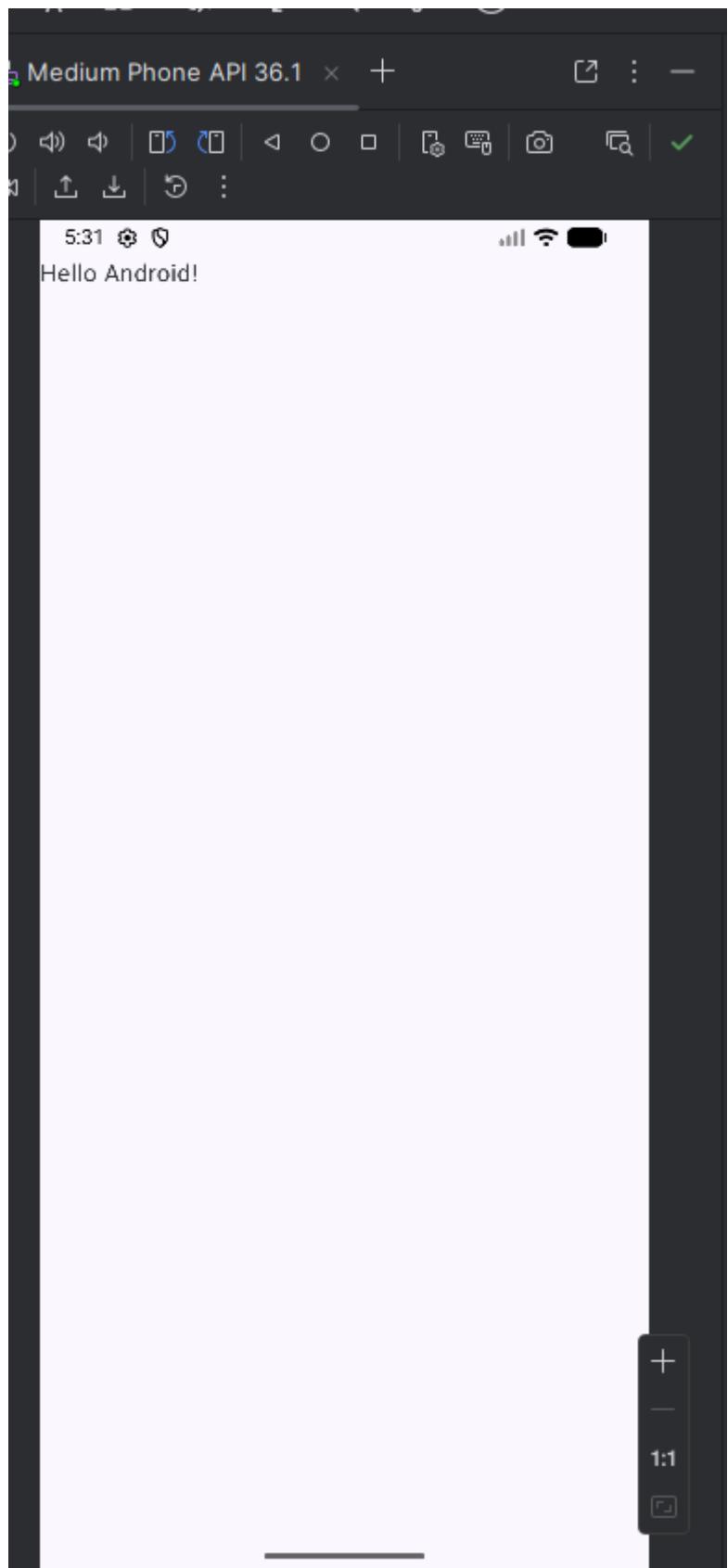


The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "Android". It includes the "app" module with subfolders like "manifests", "kotlin+java", and "res". Inside "kotlin+java", there's a package named "com.example.appestudiantes" containing "MainActivity.kt", "ui.theme", and "model" subfolders. The "model" folder contains a file named "Estudiante.kt", which is currently selected and highlighted in blue.
- Code Editor:** The main right pane displays the content of "Estudiante.kt". The code defines a class "Estudiante" with a constructor taking a string parameter "name". It also contains a companion object with a factory method "Greeting" that returns a "Text" composable with the greeting message "Hello \$name!".
- Toolbars and Status Bar:** The top bar shows tabs for "AppEstu...", "Version control...", "MainActiv...", and "Medium P...". The status bar at the bottom shows "025" and other standard Android Studio icons.

```
1 package com.example.appestudiantes
2
3 import ...
4
5
6 class ListaEstudiantesActivity : ComponentActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         enableEdgeToEdge()
10        setContent {
11            AppEstudiantesTheme {
12                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
13                    Greeting2(
14                        name = "Android",
15                        modifier = Modifier.padding(paddingValues = innerPadding)
16                    )
17                }
18            }
19        }
20    }
21
22    companion object {
23        @Composable
24        fun Greeting2(name: String, modifier: Modifier = Modifier) {
25            Text(
26                text = "Hello $name!",
27                modifier = modifier
28            )
29        }
30    }
31
32    @Preview(showBackground = true)
33    @Composable
34    fun GreetingPreview2() {
35        AppEstudiantesTheme {
36            Greeting2(name = "Android")
37        }
38    }
39}
40
41
```

```
1 package com.example.appestudiantes
2
3 import ...
4
5
6 class RegistroEstudianteActivity : ComponentActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         enableEdgeToEdge()
10        setContent {
11            RegistroEstudianteUI(context = this)
12        }
13    }
14
15    companion object {
16        @OptIn(ExperimentalMaterial3Api::class)
17        @Composable
18        fun RegistroEstudianteUI(context: Context) {
19            var nombres by remember { mutableStateOf("") }
20            var apellido by remember { mutableStateOf("") }
21            var edad by remember { mutableStateOf("") }
22
23            Scaffold(
24                topBar = {
25                    TopAppBar(title = { Text(text = "Registrar Estudiante") })
26                }
27            ) {
28                Column(
29                    modifier = Modifier
30                        .padding(top = 112.dp)
31                        .fillMaxSize(),
32                    verticalArrangement = Arrangement.spacedBy(space = 16.dp)
33                ) {
34
35                
```



- ¿Para qué sirve el archivo tipo Data class, ¿cuál es su utilidad?

Un Data class es una clase especial diseñada principalmente para almacenar y representar datos de forma eficiente y con menos códigos.

¿En que consiste el patrón Singleton ?

El patrón Singleton es un patrón de diseño que garantiza que una clase tenga una única instancia (un solo objeto) en toda la aplicación y proporciona un punto de acceso global a esa instancia.

- Define que es un elemento Card, y su utilidad

Una Card es esencialmente una "caja" rectangular que contiene contenido organizado, con bordes definidos, sombra opcional y padding interno. Funciona como una unidad independiente de información que puede incluir texto, imágenes, botones y otros elementos.