

Esmeraldas 20 de noviembre del 2025

Nombres: Jandry Steven Zambrano Palacios

Carrera: Desarrollo de Software

Tema: Listas Mutables en Android Studio

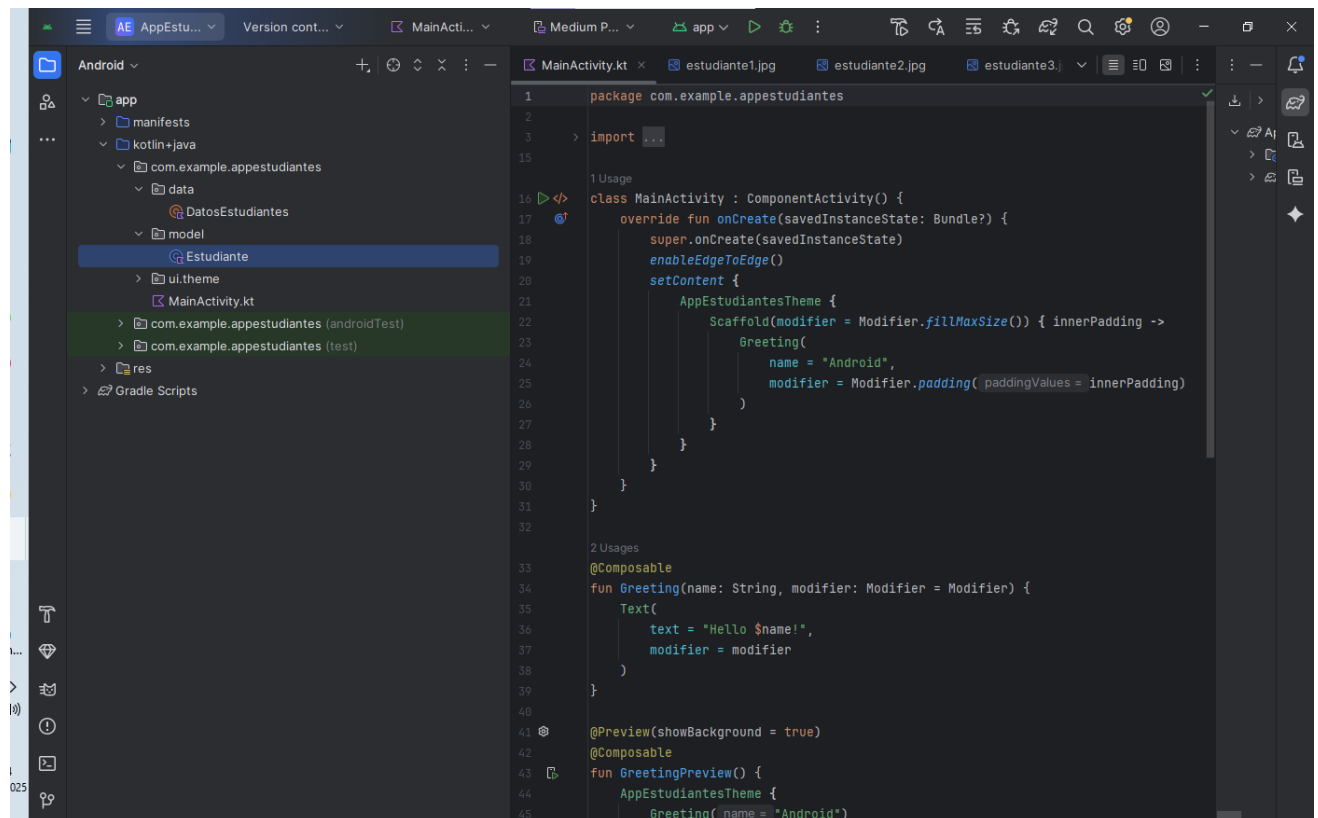
Introducción.

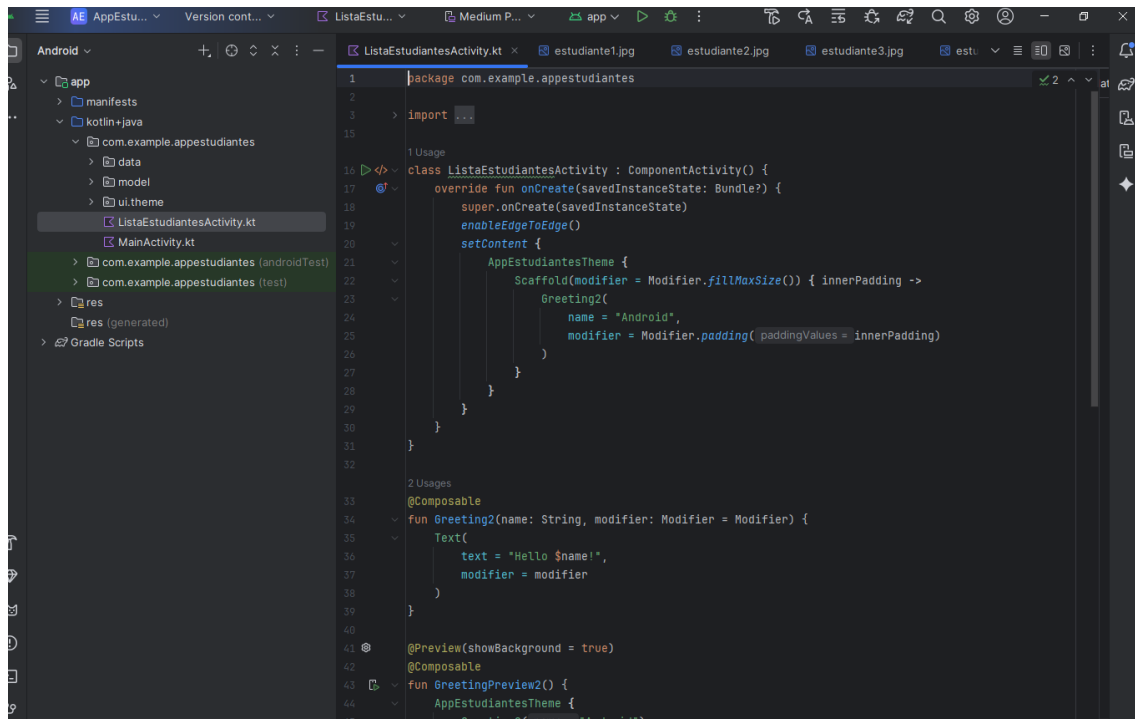
Este informe documenta el desarrollo de una aplicación móvil

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

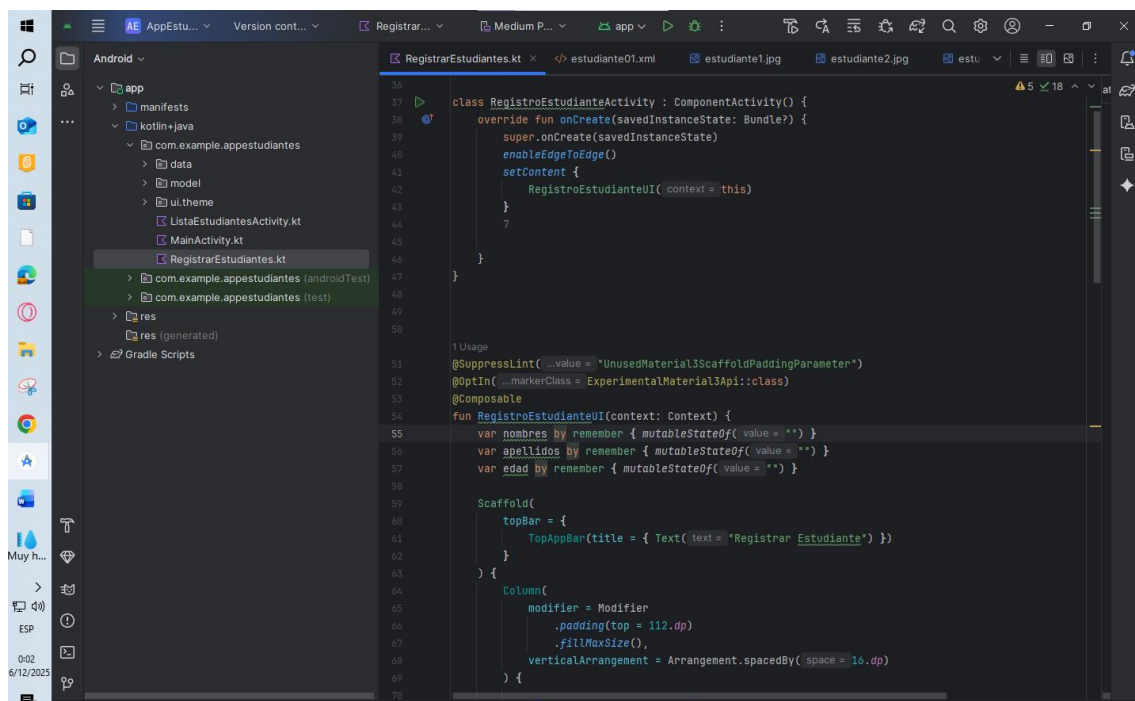
Visualización de estudiantes en tarjetas (Cards) y registro de nuevos estudiantes a Través de un formulario

La estructura de los paquetes estructurados.

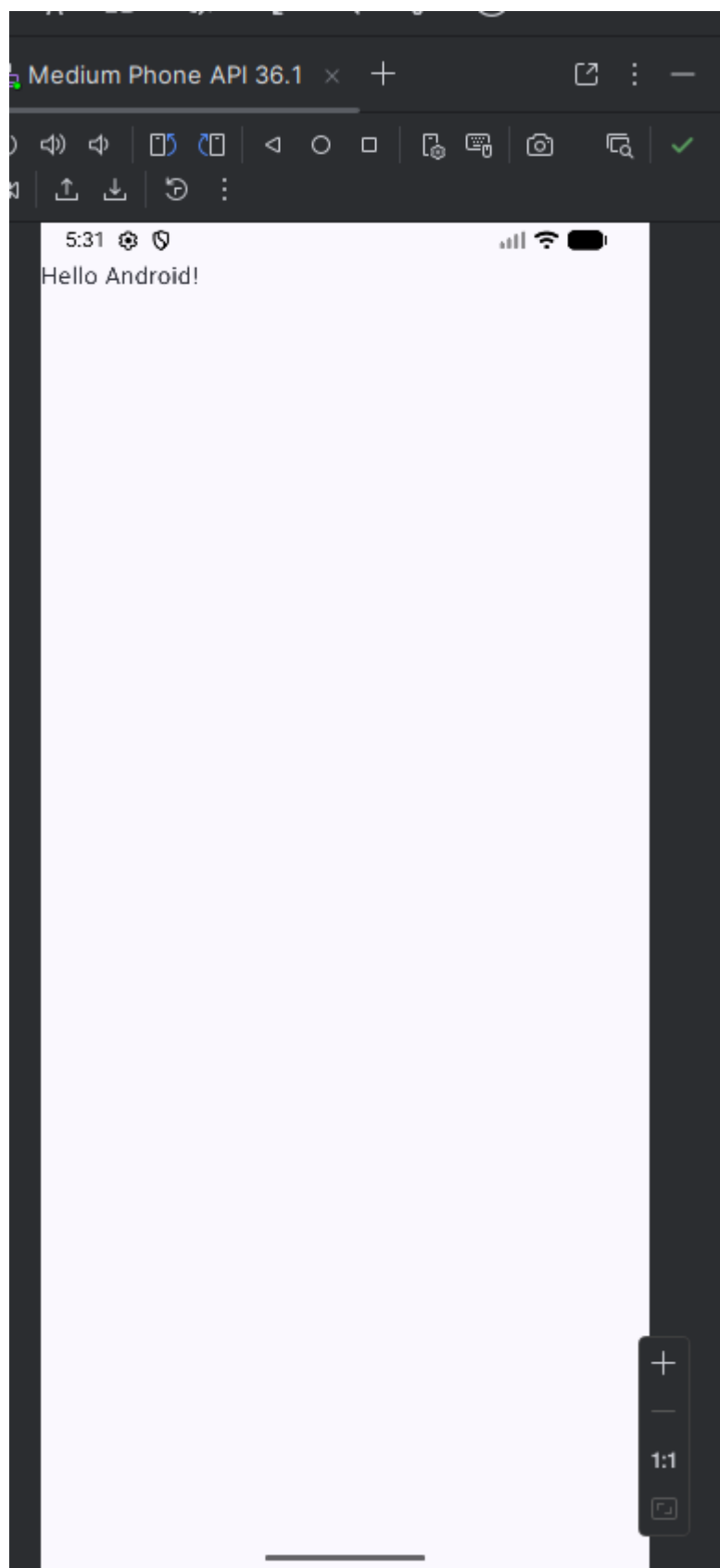




```
1 package com.example.appestudiantes
2
3 import androidx.appcompat.app.AppCompatActivity
4 import androidx.compose.foundation.layout.fillMaxSize
5 import androidx.compose.foundation.layout.padding
6 import androidx.compose.material3.Scaffold
7 import androidx.compose.material3.Text
8 import androidx.compose.runtime.Composable
9 import androidx.compose.ui.Modifier
10 import androidx.compose.ui.text.TextStyle
11 import androidx.compose.ui.unit.dp
12 import androidx.core.os.BundleCompat
13 import androidx.core.os.SavedInstanceStateCompat
14
15
16 class ListaEstudiantesActivity : AppCompatActivity() {
17     override fun onCreate(savedInstanceState: Bundle?) {
18         super.onCreate(savedInstanceState)
19         enableEdgeToEdge()
20         setContent {
21             AppEstudiantesTheme {
22                 Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
23                     Greeting2(
24                         name = "Android",
25                         modifier = Modifier.padding(paddingValues = innerPadding)
26                     )
27                 }
28             }
29         }
30     }
31 }
32
33
34 @Composable
35 fun Greeting2(name: String, modifier: Modifier = Modifier) {
36     Text(
37         text = "Hello $name!",
38         modifier = modifier
39     )
40 }
41
42 @Preview(showBackground = true)
43 @Composable
44 fun GreetingPreview2() {
45     AppEstudiantesTheme {
46         Greeting2(name = "Android")
47     }
48 }
```



```
36
37 class RegistrarEstudianteActivity : AppCompatActivity() {
38     override fun onCreate(savedInstanceState: Bundle?) {
39         super.onCreate(savedInstanceState)
40         enableEdgeToEdge()
41         setContent {
42             RegistrarEstudianteUI(context = this)
43         }
44     }
45 }
46
47
48
49
50
51 @SuppressLint(...)
52 @OptIn(...)
53 @Composable
54 fun RegistrarEstudianteUI(context: Context) {
55     var nombres by remember { mutableStateOf(value = "") }
56     var apellidos by remember { mutableStateOf(value = "") }
57     var edad by remember { mutableStateOf(value = "") }
58
59     Scaffold(
60         topBar = {
61             TopAppBar(title = { Text(text = "Registrar Estudiante") })
62         }
63     ) {
64         Column(
65             modifier = Modifier
66                 .padding(top = 112.dp)
67                 .fillMaxSize(),
68             verticalArrangement = Arrangement.spacedBy(space = 16.dp)
69         ) {
70
71         }
72     }
73 }
```



- ¿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.