Farit Alexander Reasco Torres 18/11/2025

Desarrollo Móvil

1. Documentación de Pruebas

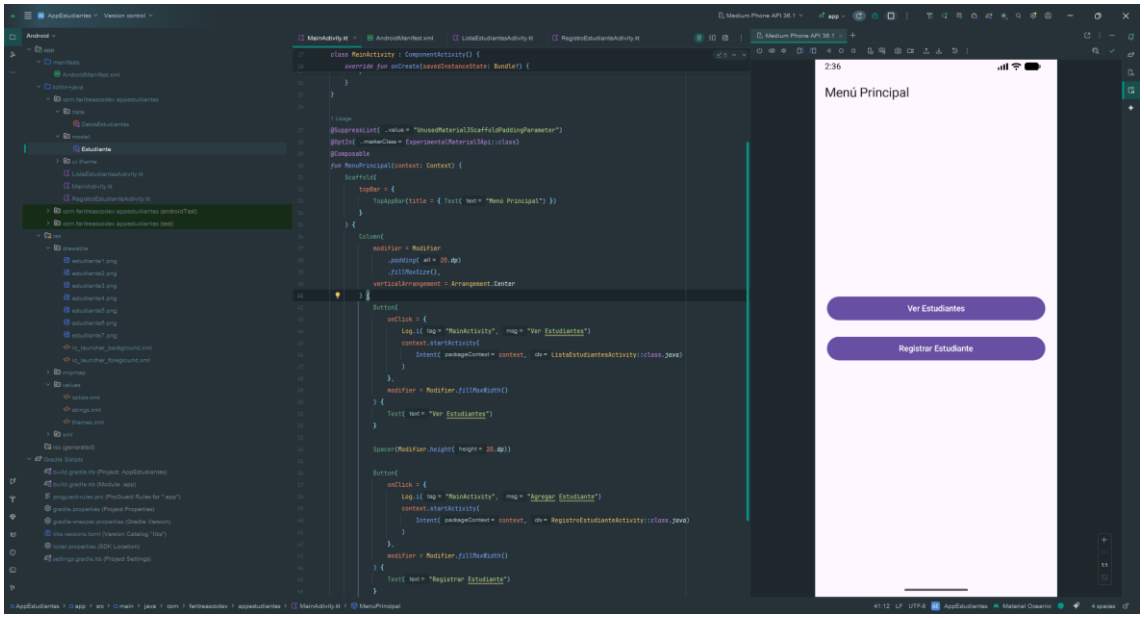
Ejercicio con listas mutables – 10AppEstudiantes

Se creó una app llamada AppEstudiantes en Kotlin que permite gestionar información básica de estudiantes mediante dos funcionalidades principales: ver la lista de los estudiantes en cards con sus datos y una pequeña imagen, y registrar nuevos estudiantes mediante un formulario. La aplicación en sí, implementa el patrón Singleton con la finalidad de compartir datos entre actividades con MutableListOf, usa, además, el Jetpack Compose para UI moderna y reactiva, y aplica Material Design.

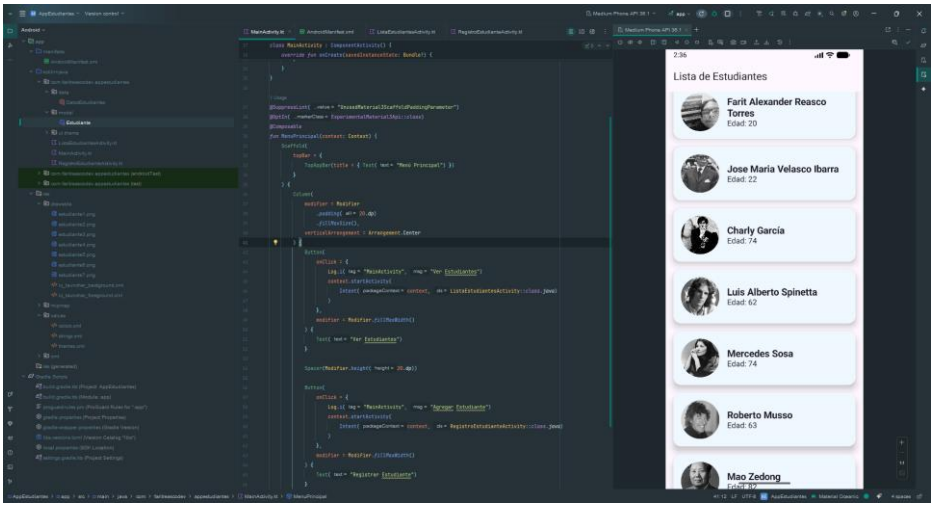
Tal y como se detalló en el documento, se implemetó una arquitectura en capas: model con el Data Class Estudiante, data con Singleton DatosEstudiantes y lista mutable compartida, y Activities siendo MainActivity la principal, ListaEstudiantes para mostrar los estudiantes en cards y finalmente el RegistroActivitu para agregar estudiantes. De con esta estructura “modular” pienso que será más fácil mantener, escalar y comprender el código.

Pruebas en el Emulador Medium Phone API 31.1:

MainActivity:

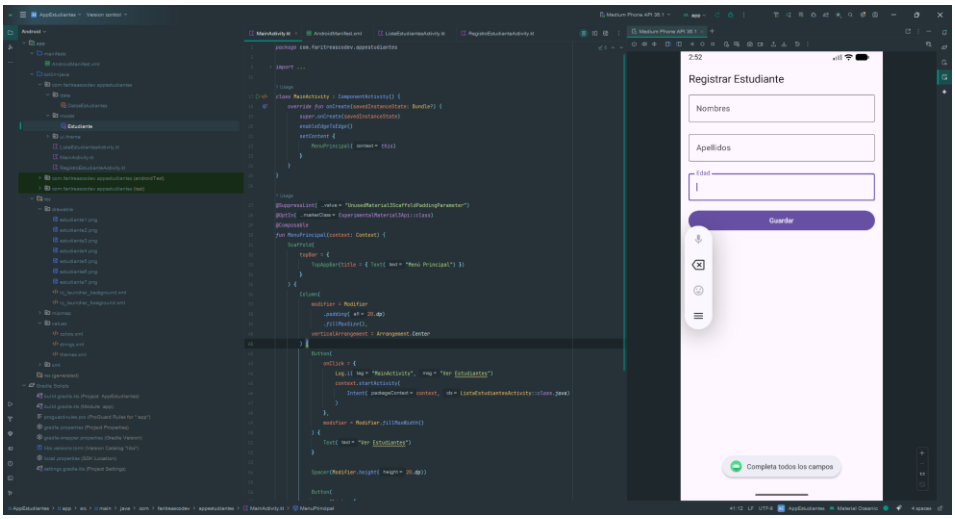


ListaEstudianteActivity

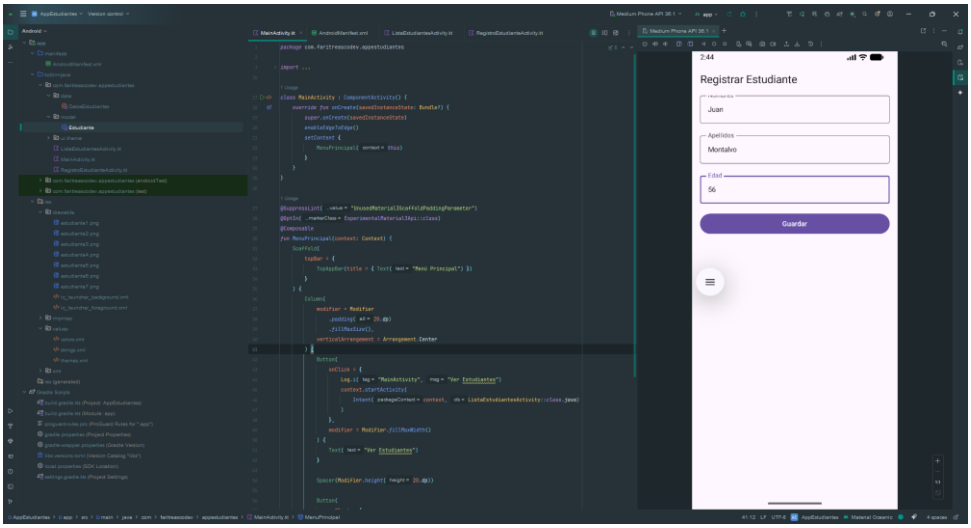


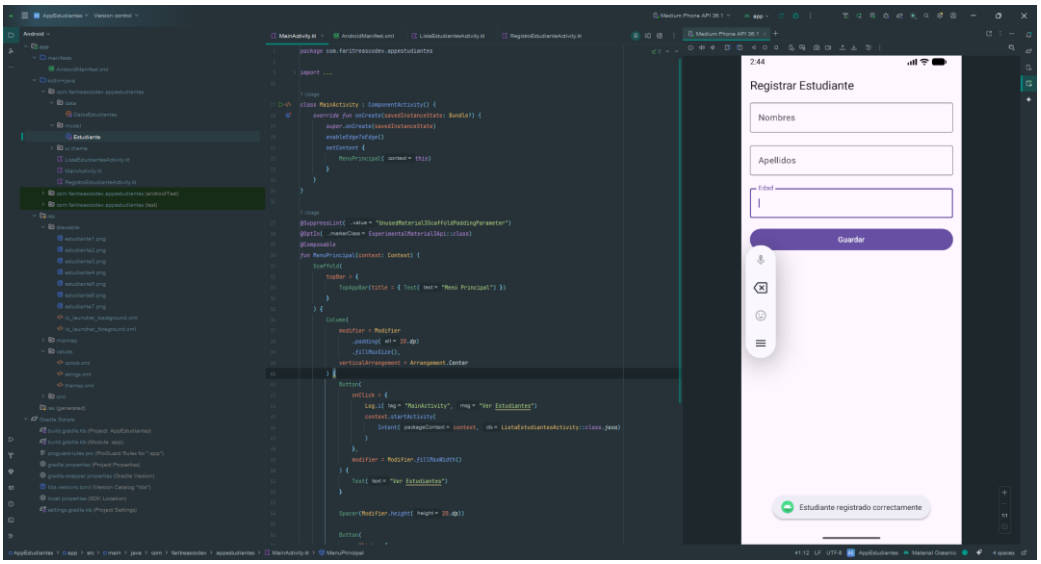
RegistroEstudianteActivity

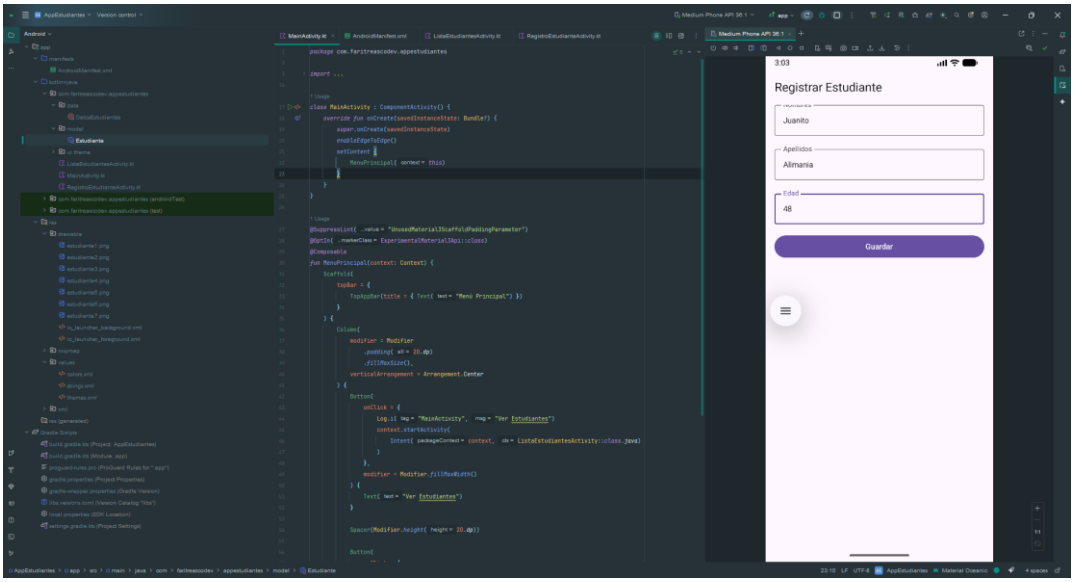
Con campos vacíos:

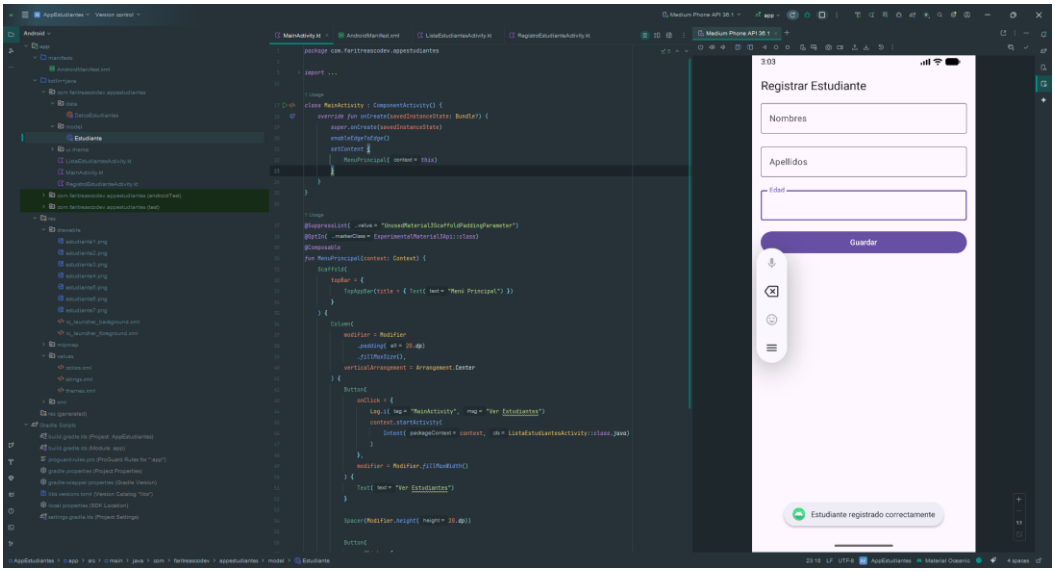


Carga de datos:

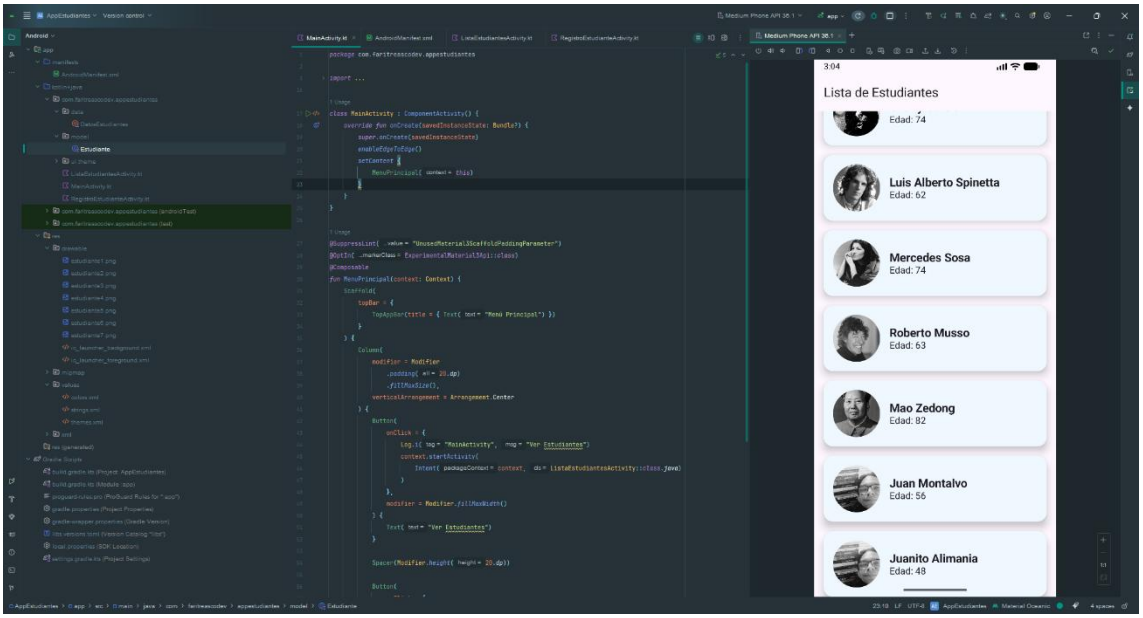



Registrar Estudiante
Nombres
Apellidos
Edad
Guardar
Estudiante registrado correctamente

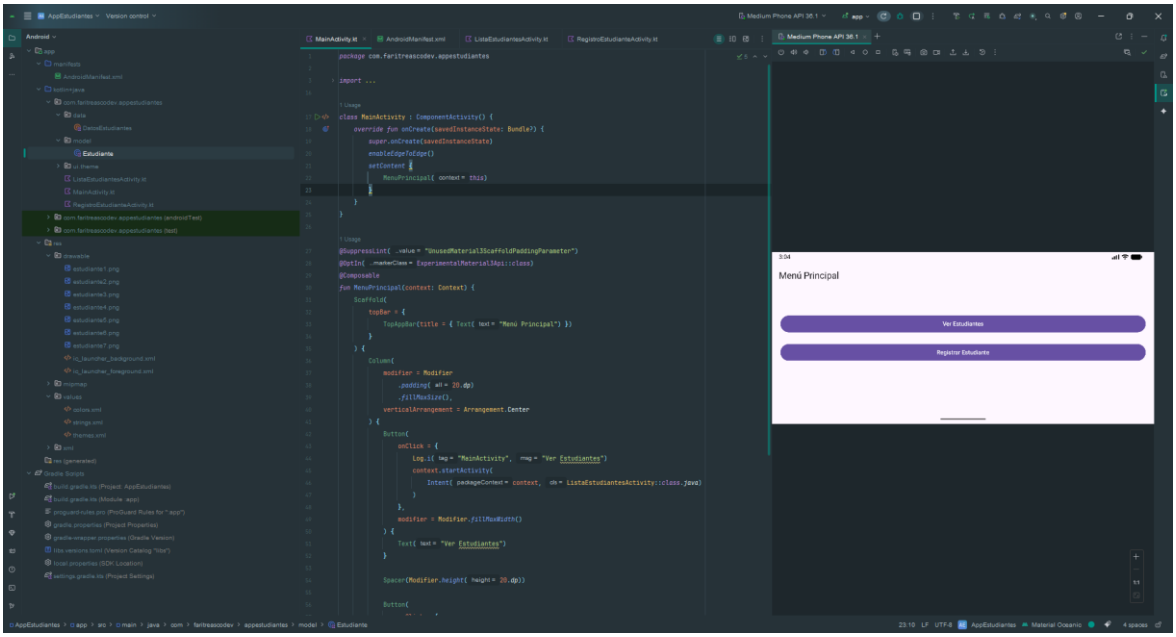

Registrar Estudiante
Juanito
Apellidos
Almania
Edad
48
Guardar

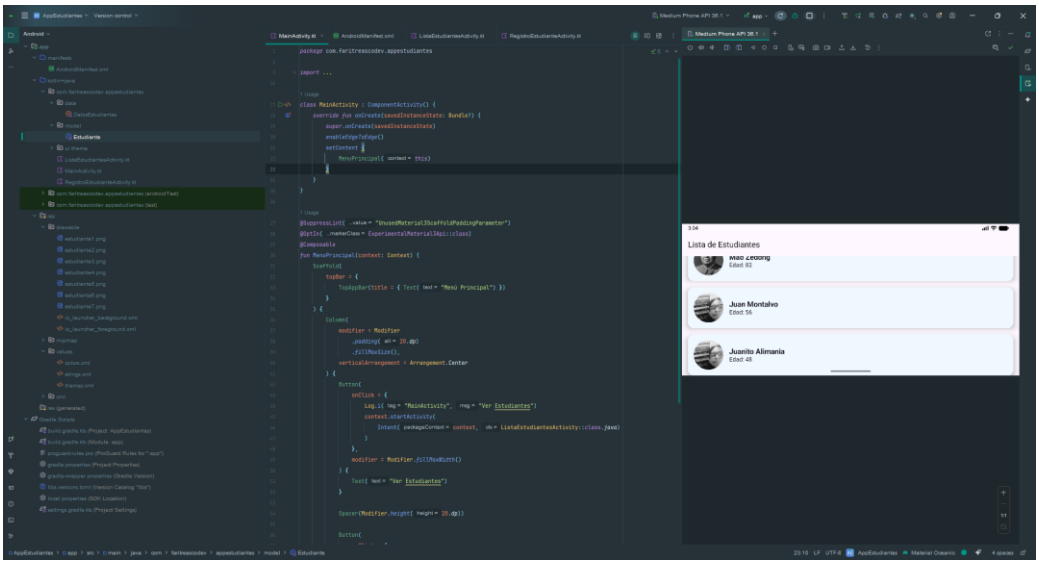

Registrar Estudiante
Nombres
Apellidos
Edad
Guardar
Estudiante registrado correctamente

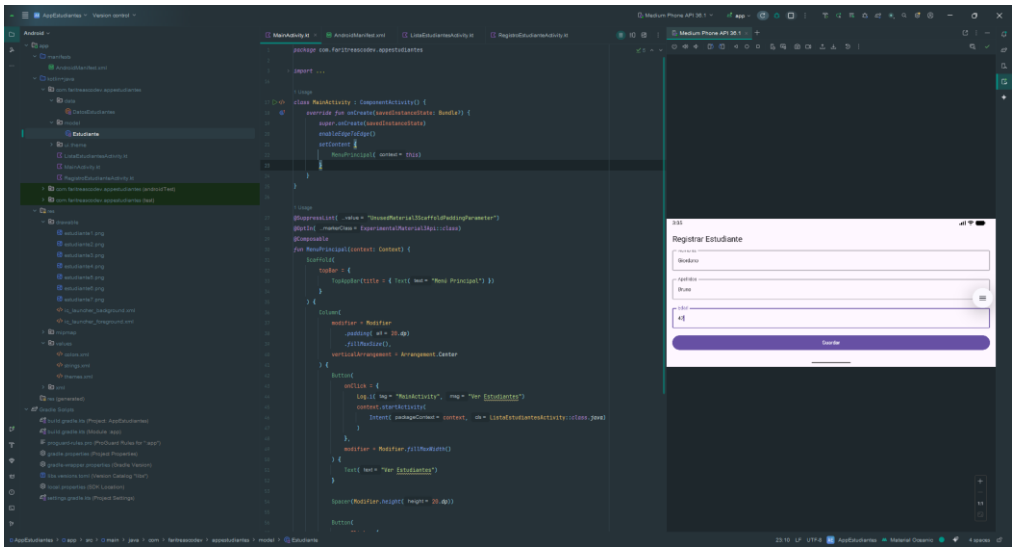
Se cargan directamente al listado, con la imagen “por defecto” que es la de estudiante1.png

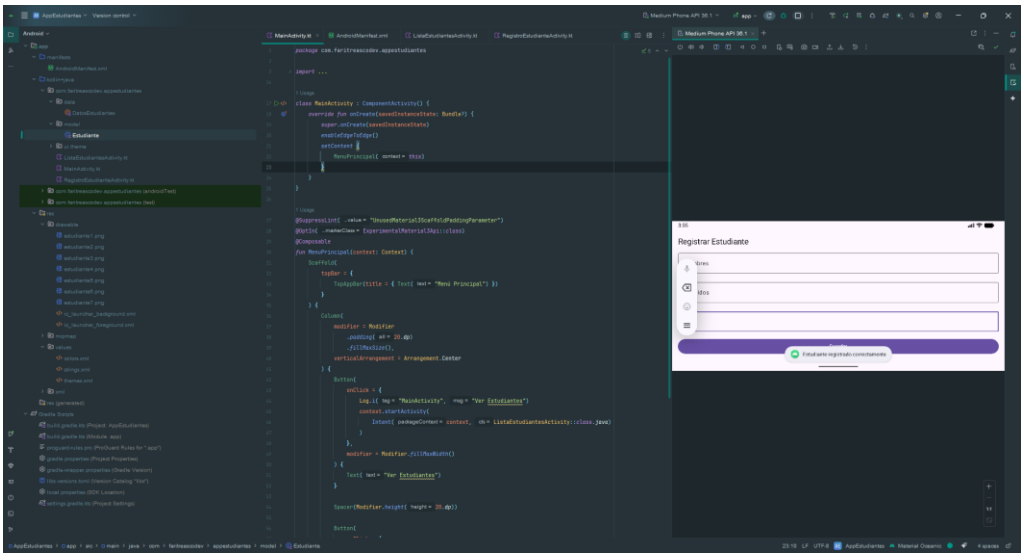


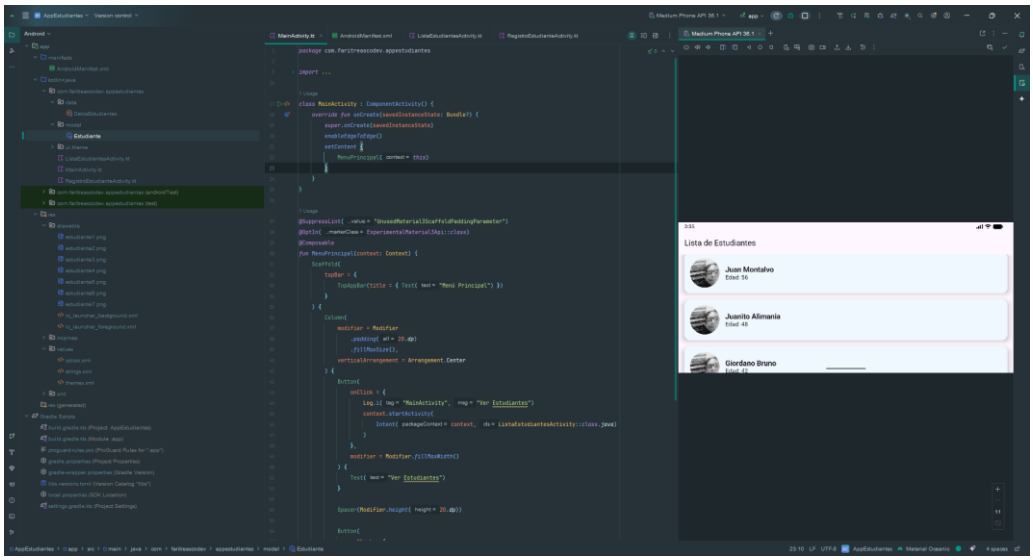
Vista en horizontal:




Lista de Estudiantes
Juan Montalvo
Edad 56
Juanito Alimania
Edad 48

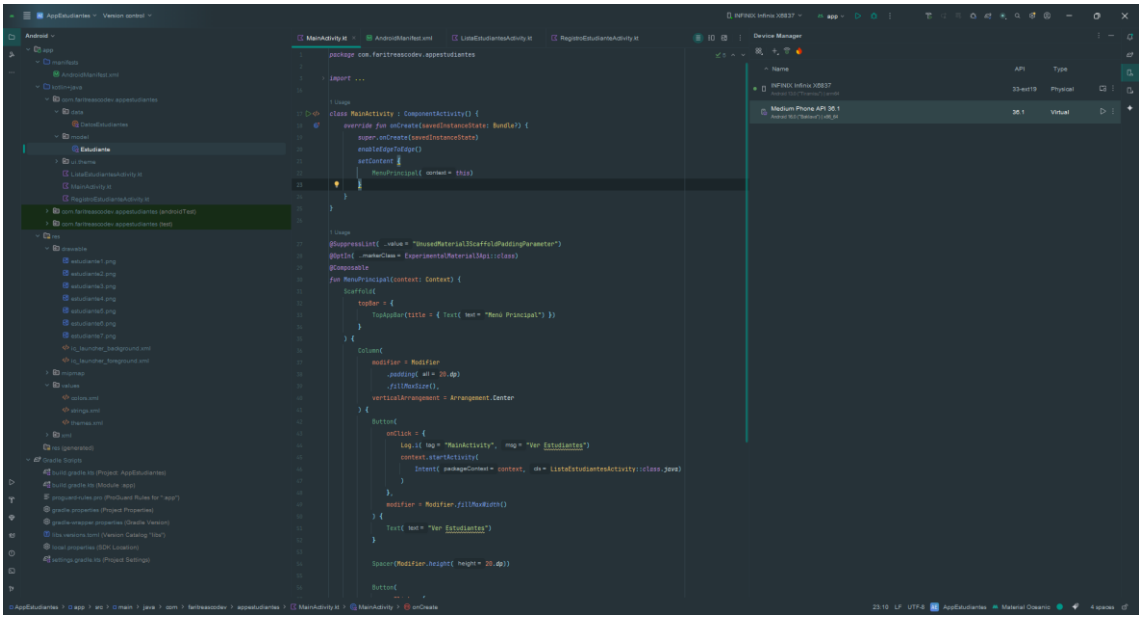

Registrar Estudiante
Guardar


Registrar Estudiante

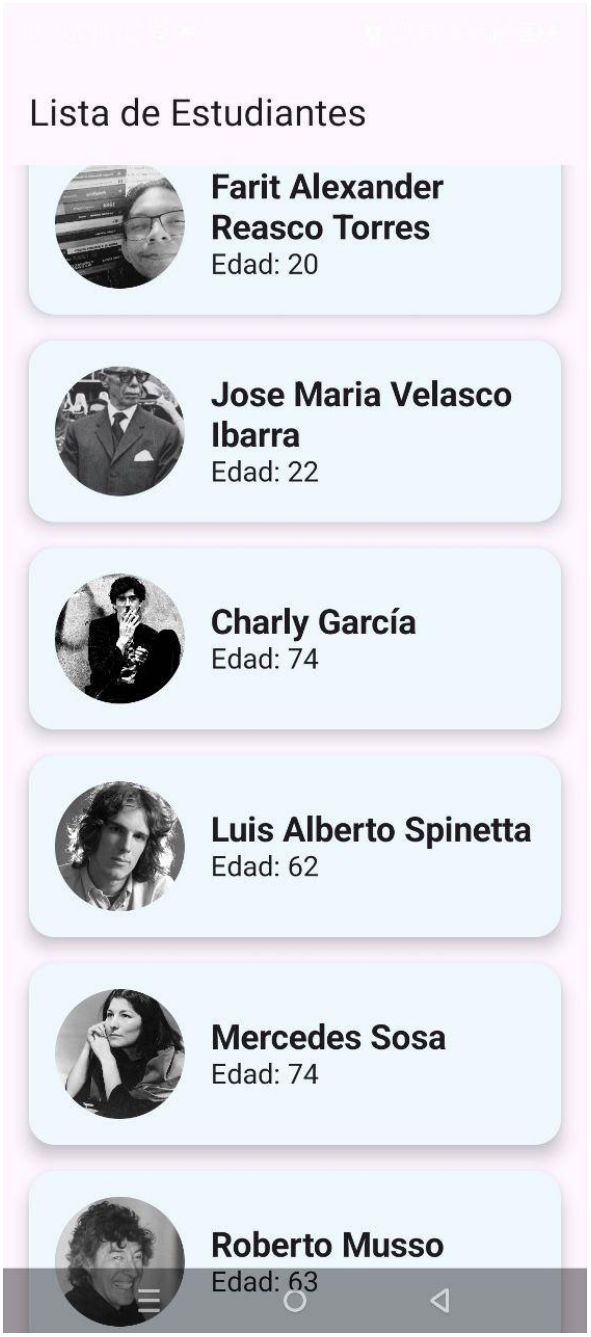


En mi celular:

Cambio de dispositivo:



Ejecución de la App:



Lista de Estudiantes

José María Velasco Ibarra
Edad: 22

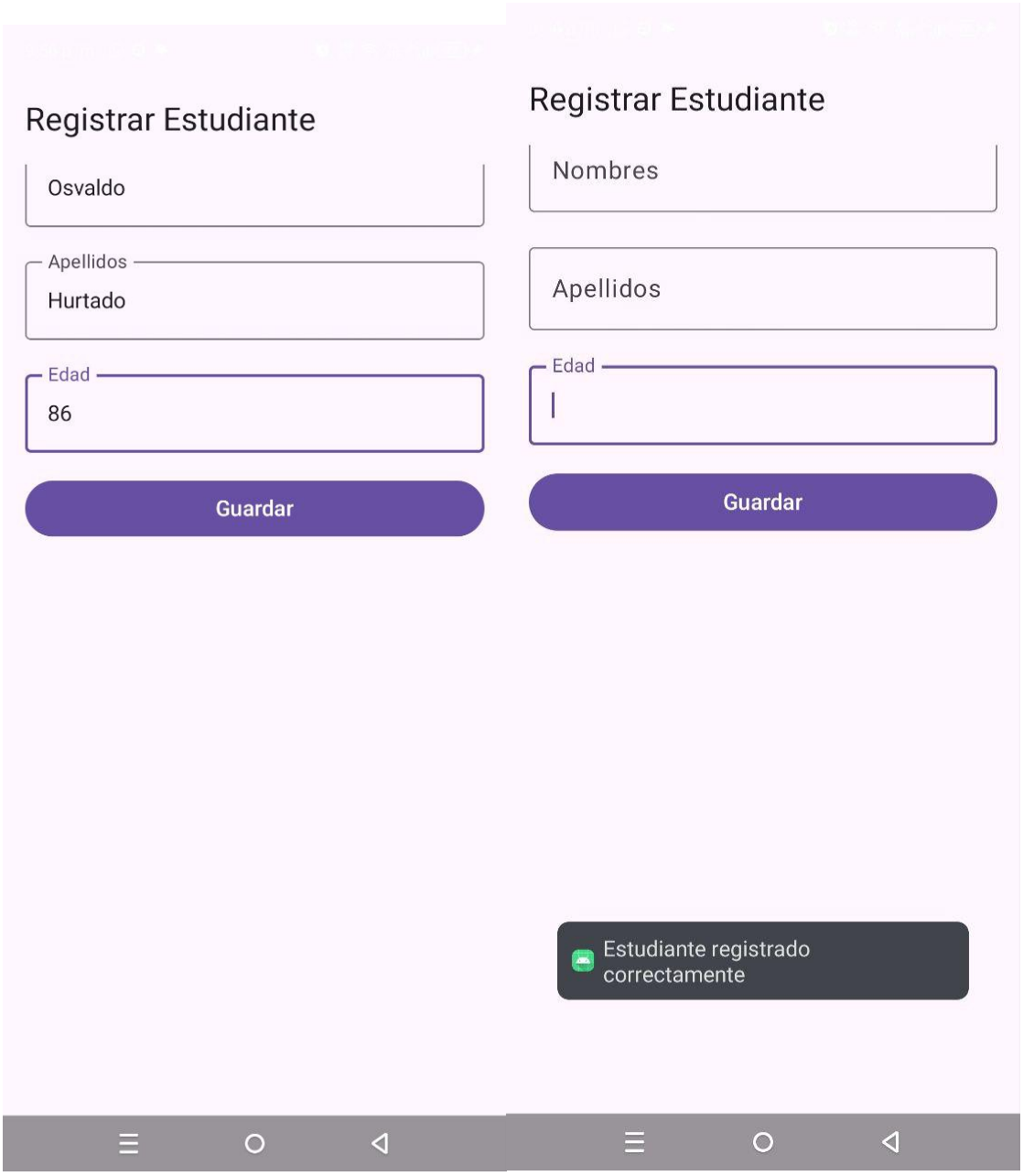
Charly García
Edad: 74

Luis Alberto Spinetta
Edad: 62

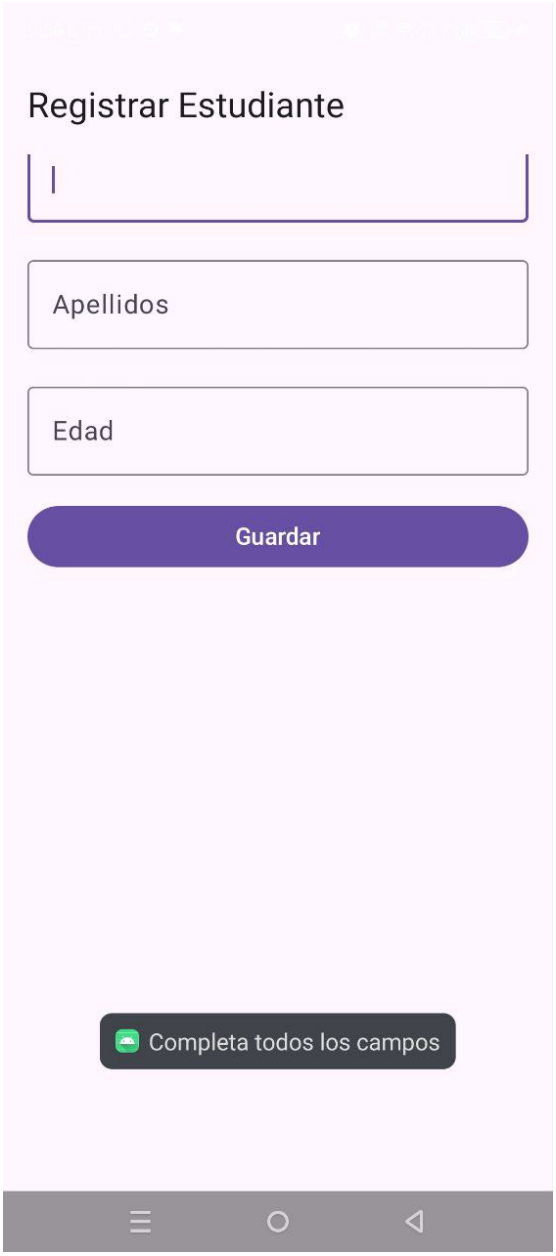
Mercedes Sosa
Edad: 74

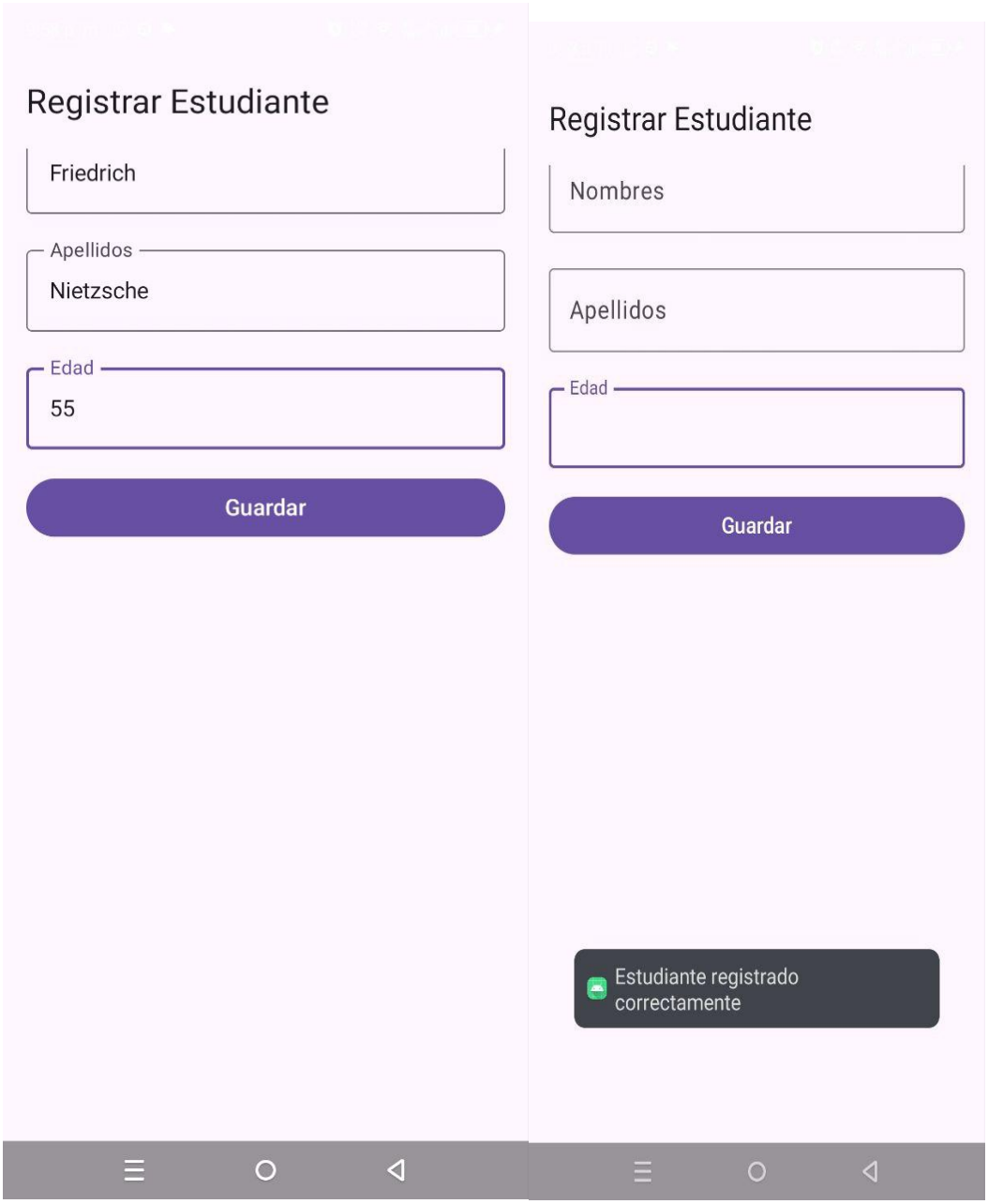
Roberto Musso
Edad: 63

Mao Zedong
Edad: 82

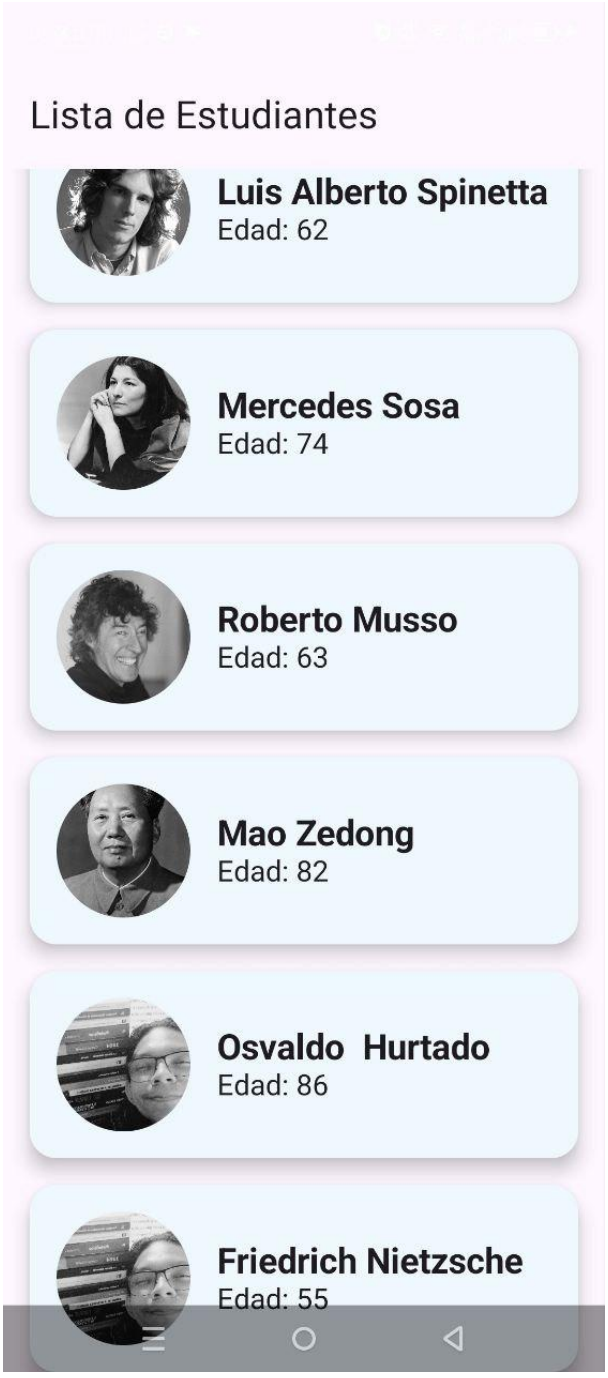

Registrar Estudiante
Osvaldo
Apellidos
Hurtado
Edad
86
Guardar
Registrar Estudiante
Nombres
Apellidos
Edad
Guardar
Estudiante registrado correctamente

Con campos vacíos

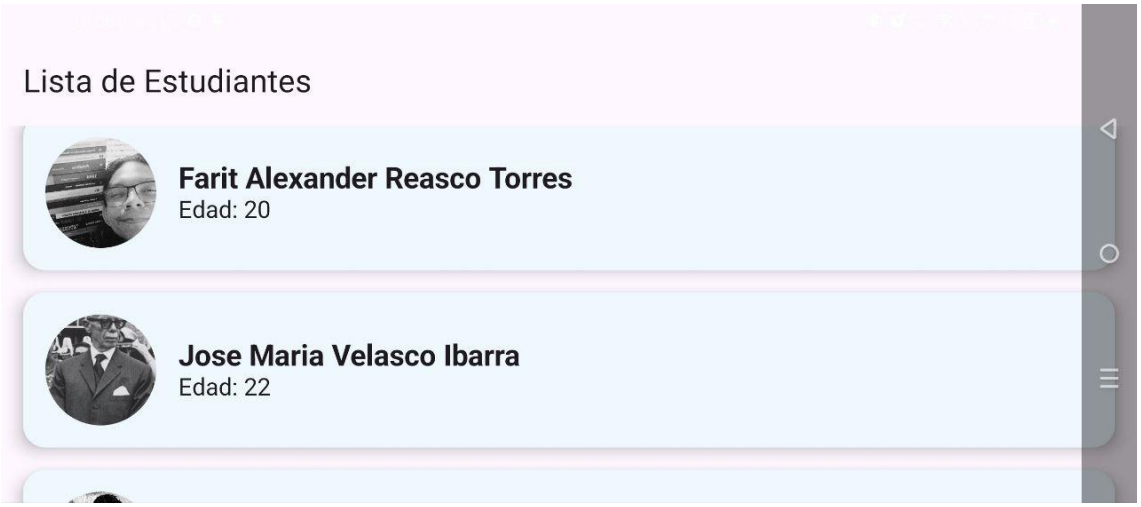



Registrar Estudiante
Friedrich
Apellidos
Nietzsche
Edad
55
Guardar
Registrar Estudiante
Nombres
Apellidos
Edad
Guardar
Estudiante registrado correctamente

Se cargan en el listado



En Horizontal



En conclusión, ambas plataformas demuestran que la aplicación funciona correctamente sin errores de compilación. El patrón Singleton con MutableListOf mantiene consistencia de datos entre Activities. Jetpack Compose renderiza correctamente en diferentes densidades de pantalla. La navegación mediante Intent y Context.startActivity() opera sin problemas. Todo funciona correctamente.

Preguntas

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

Las Data Class en Kotlin son clases diseñadas específicamente para almacenar datos de forma simple y eficiente. Su principal ventaja es que Kotlin genera automáticamente métodos como el equals(), hashCode(), toString() y copy(), ahorrándome escribir un montón de código repetitivo. Son perfectas para crear modelos de datos o cualquier estructura que solo necesite guardar información sin lógica tan compleja. Además, al usar val en sus propiedades, promueven la inmutabilidad, lo que hace el código más seguro y predecible.

¿En qué consiste el patrón Singleton?

El patrón Singleton es un patrón de diseño que garantiza que una clase tenga una única instancia en toda la aplicación y proporciona un punto de acceso global a ella. En resumidas cuentas, evita que puedas crear múltiples copias de una clase haciendo privado su constructor, y expone un método estático (como getInstance()) que siempre devuelve la misma instancia. Es súper útil cuando necesito compartir recursos como conexiones a bases de datos, configuraciones globales o gestores de logs, donde tener varias instancias causaría problemas o desperdiciaría memoria.

Define que es un elemento Card, y su utilidad

1. Un Card es un componente visual de Material Design que funciona como un contenedor elevado con bordes redondeados y sombra.
2. Su función principal es agrupar información relacionada de manera organizada y atractiva visualmente.
3. Los Cards pueden contener texto, imágenes, botones y otros elementos, y son ideales para mostrar listas de elementos como perfiles, productos o noticias.
    a. Son interactivos por naturaleza, ya que pueden responder a toques o gestos, lo que los hace atractivos para las apps.

¿Qué indican las siguientes anotaciones?

- **@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")**: Esta anotación le dice al analizador de código de Android que ignore una advertencia específica. El Scaffold de Material3 me da un padding automático para que el contenido no se tape con la TopAppBar, pero si decido no usarlo porque apliqué mi propio padding.

- **@OptIn(ExperimentalMaterial3Api::class)**: Aquí me indica que estoy usando APIs experimentales de Material3 que todavía no son estables, o sea son características en fase de prueba que pueden cambiar, renombrarse o eliminarse en futuras versiones. Al poner esta anotación, básicamente estoy aceptando el riesgo de que mi código pueda romperse cuando actualice la biblioteca. Sin ella, el compilador ni siquiera me deja usar componentes experimentales.

Bibliografía

Leiva, A. (s. f.). *Singleton - Patrones de diseño | DevExpert*. https://devexpert.io/blog/singleton-patrones-diseno

Singleton. (s. f.). https://reactiveprogramming.io/blog/es/patrones-de-diseno/singleton

Singleton: te explicamos cómo funciona el patrón singleton. (2021, 19 febrero). IONOS Digital Guide. https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/patron-singleton/

colaboradores de Wikipedia. (2025c, julio 22). *Singleton*. Wikipedia, la Enciclopedia Libre. https://es.wikipedia.org/wiki/Singleton

Singleton. (s. f.-b). https://refactoring.guru/es/design-patterns/singleton

Cómo comenzar a usar Jetpack Compose. (s. f.). Android Developers. https://developer.android.com/develop/ui/compose/documentation?hl=es-419

Inicio rápido. (s. f.). Android Developers. https://developer.android.com/develop/ui/compose/setup?hl=es-419

GeeksforGeeks. (2025, 15 junio). *Kotlin MutableListOf()*. GeeksforGeeks. https://www.geeksforgeeks.org/kotlin/kotlin-mutablelistof/

Trivedi, T. (2024, 3 junio). Mastering Kotlin Lists: Essential Techniques for Smart Data Management. *DhiWise*. https://www.dhiwise.com/post/kotlin-lists-essential-techniques-for-data-managemen