



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO



# **TECNOLÓGICO NACIONAL DE MEXICO**

## **INSTITUTO TECNOLÓGICO DE CIUDAD MADERO**

Ingeniería en Sistemas Computacionales.

### **Tarea 9\_ Unidad 3 Como usar Listas y material Design**

Asignatura: Programación Nativa para Moviles

Docente: Jorge Peralta Escobar

Hora: 14:00 – 15:00 pm

Integrantes:

Del Angel Del Angel Erika Yaneth #20071926

Villaseñor Grimaldo Alejandro #19071548

Semestre: Enero – Junio 2025.



GITHUB - [Workspace/Programacion\\_movil/Tarea9U3 at master · Alejandro19071548/Workspace](#)

# RUTA DE APRENDIZAJE 1 - MAS ASPECTOS BASICOS DE KOTLIN

## VIDEO 1 – RUTA DE APRENDIZAJE 1

En esta lección se presentan conceptos fundamentales para escribir código más limpio y potente en Kotlin. Se explican las CLA, que permiten crear estructuras reutilizables, las clases enum, ideales para definir valores constantes, y las clases de datos, útiles para representar información de forma sencilla. También se introduce el uso de era, que permiten ampliar el comportamiento de clases existentes sin necesidad de modificarlas.

Otro tema clave son las colecciones como listas, conjuntos y mapas, y cómo aprovechar funciones como map, filter y reduce para procesar datos de forma más funcional y eficaz.

Finalmente, se destaca la relevancia de dominar estos conceptos para crear apps Android más robustas y escalables, incentivando la práctica a través de ejercicios interactivos (codelabs) que fortalecen el aprendizaje.

## CUESTIIONARIO RUTA 1

1. ¿Cuál de las siguientes opciones es una forma válida de definir una clase de datos en Kotlin?

☒ `data class Person(val name: String, val age: Int)` ¡Correcto!

☐ `class Person(val name: String, val age: Int): data`

☐ `class Person(val name: String, val age: Int)`

☐ `data class Person{val name: String, val age: Int}`

2. Cuando se usa una clase sellada, todas las subclases directas deben estar en el mismo paquete.

☒ verdadero ¡Correcto!

☐ falso

3. Cuando se usan elementos genéricos, el tipo de datos genéricos va dentro de \_\_\_\_.

☐ `[]`

☐ `{}`

☐ `()`

☒ `<>` ¡Correcto!

4. Una clase \_\_\_ es útil cuando tienes un conjunto fijo de valores.

- ☒ enum ¡Correcto!
- ☐ sellada
- ☐ datos
- ☐ heredada

5. Para crear un objeto de lista que tenga la capacidad de cambiar su tamaño, deberías llamar a \_\_\_.

- ☐ `modifiableListOf()`
- ☐ `immutableListOf()`
- ☐ `listOf()`
- ☒ `mutableListOf()` ¡Correcto!

6. ¿Cuáles de las siguientes son funciones de orden superior?

*Selecciona todas las respuestas que consideres correctas.*

- ☒ `map` ¡Correcto!

7. En la siguiente línea de código, ¿cuál de los siguientes comandos mostrará `Blue`?  
(Sugerencia: Si no lo sabes con certeza, ejecuta el código en el Playground de Kotlin).

```
val colors = listOf("Red", "Green", "Blue")
```

*Selecciona todas las respuestas que consideres correctas.*

- ☒ `println(colors[2])` ¡Correcto!
- ☒ `println(colors.get(2))` ¡Correcto!
- ☐ `println(colors.contains(2))`
- ☐ `println(colors.getOrDefaultValue(index = 2, defaultValue = 10))`

8. El concepto de programación de una clase que tiene una sola instancia se denomina \_\_\_.

- ☐ Exclusividad
- ☒ Singleton ¡Correcto!
- ☐ Mono-object
- ☐ Lambda

9. ¿Cuál de las siguientes afirmaciones es verdadera respecto de los conjuntos y los mapas?

- ☒ Un conjunto debe tener valores distintos, y las claves de mapa deben ser distintas. ✓ ¡Correcto!
- ☐ Un conjunto debe contener valores distintos, y las claves de mapa pueden contener valores duplicados.
- ☐ Un conjunto puede contener valores duplicados, y los valores de un mapa también pueden estar duplicados.
- ☐ Un conjunto puede contener valores duplicados, y las claves de mapa deben ser distintas.

10. Si tienes una variable de colección denominada `records`, puedes llamar a \_\_\_ para determinar la cantidad de elementos que contiene.

- ☐ `records.length`
- ☐ `records.quantity`
- ☐ `len(records)`
- ☒ `records.size` ✓ ¡Correcto!

## RUTA DE APRENDIZAJE 2 - CREA UNA LISTA DESPLAZABLE

### VIDEO 1 – RUTA DE APRENDIZAJE 2

Esta lección enseña cómo desarrollar una aplicación que muestra una lista desplazable de afirmaciones positivas, cada una con una imagen ilustrativa. Se utiliza Columna diferida de Jetpack Compose para crear listas efectivas tarjetas como elementos visuales que combinan texto e imagen.

Se destaca la importancia de estructurar el código en funciones reutilizables y de aplicar modificadores para controlar el estilo y la disposición de los componentes en pantalla. Además, se promueven buenas prácticas en el desarrollo Android, como separar la lógica de la interfaz y usar recursos externos para textos e imágenes, facilitando así el mantenimiento y la escalabilidad del código.

### CUESTIONARIO RUTA 2

1. Los íconos de selector adaptables de apps para Android consisten en elementos de diseño vectoriales de primer y segundo plano.

☒ Verdadero

✓ ¡Correcto!

☐ Falso

2. Cuando se crea una IU que requiere una lista de desplazamiento con una cantidad variable de elementos, ¿qué elementos componibles deberías usar?

☐ Column

☐ Row

☒ LazyColumn

✓ ¡Correcto!

☐ Card

3. ¿Qué tipo de archivo se utiliza para los vectores de primer y segundo plano del ícono de selector?

☐ .txt

☐ .jpg

☐ .png

☒ .xml

✓ ¡Correcto!

4. ¿Qué elemento componible debería usar para crear una cuadrícula desplazable verticalmente con una cantidad indeterminada de elementos?

☐ LazyColumn

☒ LazyVerticalGrid

✓ ¡Correcto!

☐ LazyHorizontalGrid

☐ Row

5. El elemento componible `Column` no es una buena opción para una lista de artículos de cantidad desconocida debido a \_\_\_\_.

- ☒ Solo puede contener una cantidad pequeña y limitada de elementos predefinidos. ¡Correcto!
- ☐ Ordena elementos de forma vertical.
- ☐ Proporciona desplazamiento de forma predeterminada sin código adicional.
- ☐ Puede agregar contenido on demand

6. El método \_\_\_\_ se usa para agregar contenido a un elemento componible `LazyColumn`.

- ☐ `painterResource()`
- ☐ `Modifier.padding()`
- ☒ `items()` ¡Correcto!
- ☐ `onCreate()`

7. ¿Qué elemento componible organiza los elementos de forma horizontal?

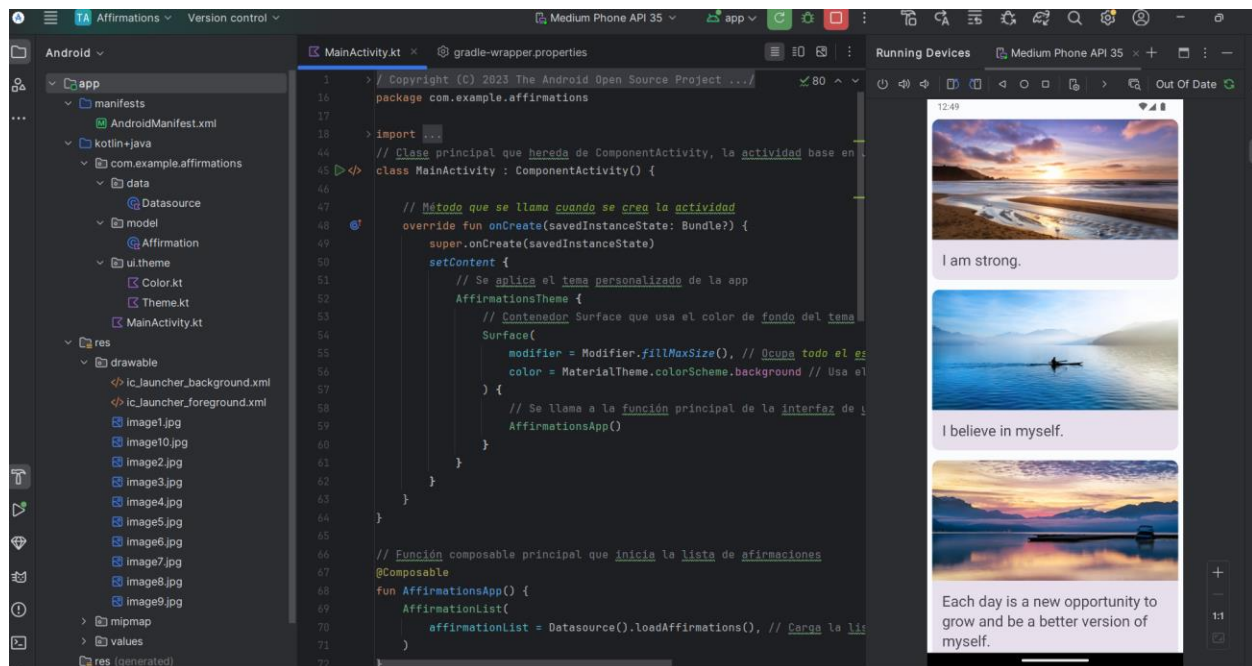
- ☐ `Column`
- ☐ `Card`
- ☐ `Image`
- ☒ `Row` ¡Correcto!

8. ¿Qué elemento componible organiza los elementos de forma vertical?

- ☒ `Column` ¡Correcto!
- ☐ `Card`
- ☐ `Image`
- ☐ `Row`

## Results

Tu puntuación es **8 de 8**. ¡Felicitaciones! Aprobaste el cuestionario.



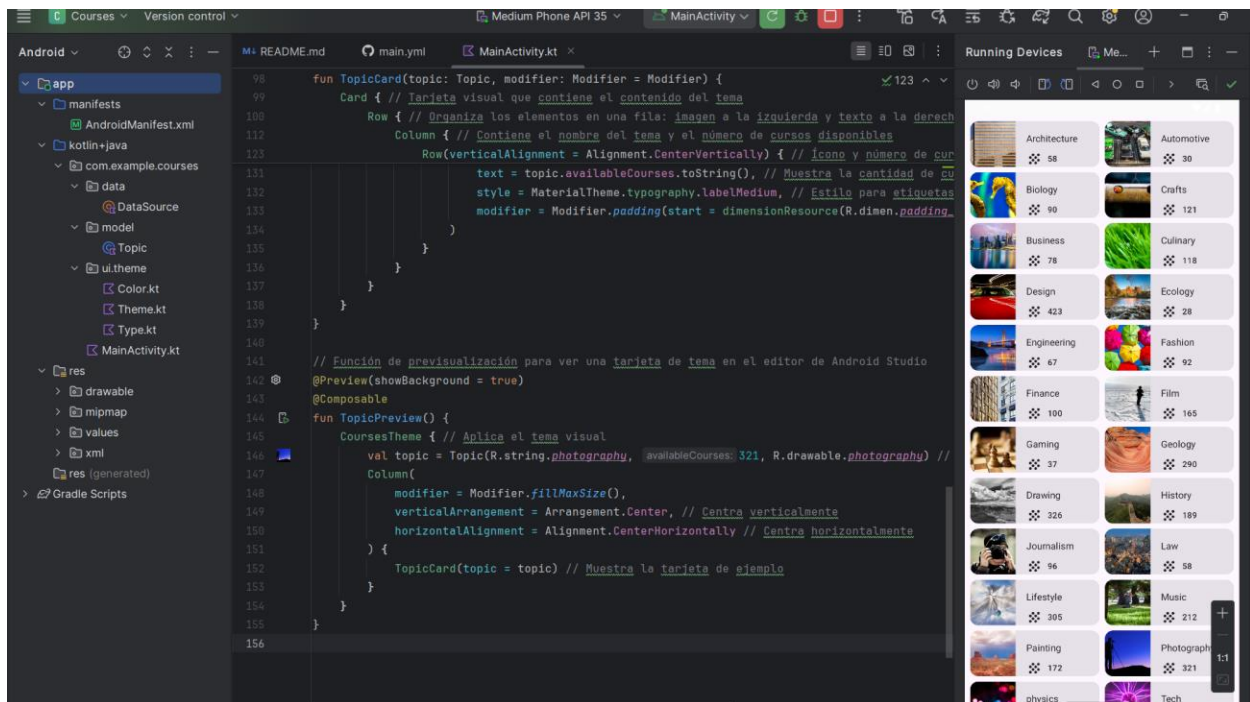
Este código crea una aplicación Android que muestra una lista de afirmaciones positivas, cada una acompañada de una imagen representativa. Utiliza Jetpack Compose para construir la interfaz de forma declarativa.

La aplicación inicia en la clase MainActivity, donde se establece el tema visual y se llama a la función principal de la interfaz (AffirmationsApp). Esta función carga una lista de afirmaciones desde un origen de datos y la pasa a otra función que construye la lista visual (AffirmationList).

Cada afirmación se muestra dentro de una tarjeta (AffirmationCard) que contiene una imagen y un texto. La lista es desplazable y eficiente gracias al componente LazyColumn. Además, se organiza el código en funciones reutilizables y se aplican estilos usando modificadores.

En conjunto, el código presenta una estructura clara y moderna para mostrar contenido en una app Android, siguiendo buenas prácticas como el uso de temas, componentes reutilizables y separación de responsabilidades.

## CUADRICULA



Este código crea una aplicación Android con Jetpack Compose que muestra una cuadrícula de tarjetas, donde cada tarjeta representa un tema con su imagen, nombre y la cantidad de cursos disponibles.

La actividad principal (MainActivity) configura la interfaz usando un tema personalizado y muestra el contenido en una cuadrícula de dos columnas usando LazyVerticalGrid. Cada tarjeta se construye con una fila que contiene una imagen a la izquierda y una columna con texto e ícono a la derecha.

Se aplica espaciado y estilo mediante recursos de dimensiones y tipografía definidos en el tema. También incluye una función de previsualización para ver una tarjeta de ejemplo dentro del editor de Android Studio.

En resumen, el código implementa una interfaz visual moderna, eficiente y bien organizada para presentar contenido educativo de manera atractiva y funcional.

## RUTA DE APRENDIZAJE 3 - COMPILA APPS FABULOSAS

### VIDEO 1 – RUTA DE APRENDIZAJE 3

Ofrece y aplicacionesMochila propulsora. Se explica que el Diseño de Materiales y



La lección destaca cómo Jetpack Compose simplifica el diseño de interfaces al ser un enfoque declarativo y reactivo, lo que permite mayor control y rapidez en el desarrollo. También se enseña a incorporar elementos clave del diseño como temas personalizados, paletas de colores, tipografías y formas, mostrando cómo adaptarlos fácilmente dentro de Compose para lograr una apariencia moderna y profesional en las aplicaciones.

### CUESTIONARIO RUTA 3

1. Las animaciones de tu app para Android pueden hacer lo siguiente:

- ☐ Agregar indicaciones visuales sobre lo que sucede en tu app
- ☐ Agregar un estilo refinado a tu app
- ☐ Ayudar al usuario a ver qué cambió
- ☒ Todas las opciones anteriores

✓ ¡Correcto!

2. La animación de resorte se basa en lo siguiente:

- ☐ Valores de inicio y finalización para la duración especificada
- ☒ Proporción de amortiguamiento y rigidez
- ☐ Valores de instantáneas especificados en diferentes marcas de tiempo
- ☐ Interpolación entre dos valores de fotogramas clave

✓ ¡Correcto!

3. La animación de resorte es una animación basada en la física impulsada por la fuerza de resorte.

- ☒ Verdadero

✓ ¡Correcto!

4. En los temas de Material, el color \_\_\_\_ es el que se muestra con mayor frecuencia en las pantallas y componentes de tu app.

- ☒ principal ✓ ¡Correcto!
- ☐ secundario
- ☐ superficie
- ☐ fondo

5. El siguiente archivo se usa para definir formas de componentes en Compose.

- ☐ Theme.kt
- ☐ Color.kt
- ☒ Shape.kt ✓ ¡Correcto!
- ☐ Colors.kt

6. Solo puedes tener un elemento `@Preview` componible

- ☐ Verdadero
- ☒ Falso ✓ ¡Correcto!

7. Un código de color hexadecimal comienza con un carácter numeral (#), seguido de seis letras o números que representan los componentes rojo, verde y azul (RGB) de ese color.

- ☒ Verdadero ✓ ¡Correcto!
- ☐ Falso

8. El archivo \_\_\_\_ contiene toda la información sobre el tema de la app, que se define a través del color, de la forma y la tipografía.

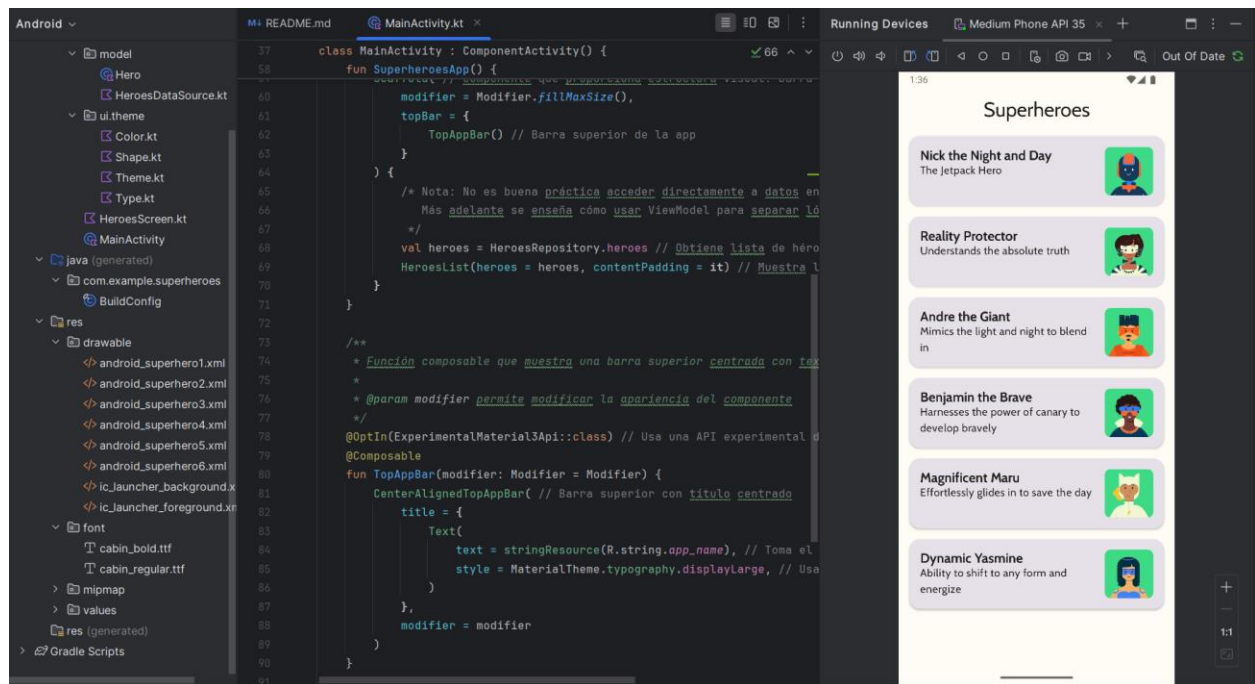
- ☒ Theme.kt ✓ ¡Correcto!
- ☐ Color.kt
- ☐ Shape.kt
- ☐ Colors.kt

9. \_\_\_\_ crea un contraste entre la tarjeta y el fondo agregando una sombra para que la app se vea más realista y atractiva a nivel visual.

- ☒ Elevation ✓ ¡Correcto!
- ☐ Forma



La interfaz utiliza componentes composables bien estructurados: `DogIcon` para mostrar `DogInformation` para mostrar el texto, y `DogItem` que los combina en una fila. La actividad principal configura el tema visual y carga la lista completa con `WoofApp`, que usa un `LazyColumn` para una lista eficiente. También incluye una función para previsualizar el diseño en Android Studio.



Finalmente, se incluye una función de vista previa ('SuperHeroesPrSuperHeroesPreview) que permite ver el diseño de la app dentro del entorno de desarrollo (Android Studio), sin necesidad de ejecutarla en un dispositivo físico o emulador. El código está organizado y preparado para escalarse mejor usando arquitecturas como ViewModel en el futuro.

