



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

Ingeniería en Sistemas Computacionales.

Tarea 7_ Unidad 1 Tu primera App para Android

Asignatura: Programación Nativa para Móviles

Docente: Jorge Peralta Escobar

Hora: 14:00 – 15:00 pm

Integrantes:

Del Angel Del Angel Erika Yaneth #20071926

Villaseñor Grimaldo Alejandro #

Semestre: Enero – Junio 2025.



RUTA 1- INTRODUCCION A KOTLIN

GitHub - [Workspace/Programacion_movil at master · Alejandro19071548/Workspace](#)

Video 1 ruta de Aprendizaje 1 (¡Te damos la bienvenida a Aspectos básicos de Android con Compose!)

El video introductorio del curso “Android Básico con Compose” presenta una guía completa para aprender a desarrollar aplicaciones Android desde cero, dirigida por el equipo de Android de Google. Los instructores, Dan, Kat y Meghan, enfatizan la importancia y el alcance de las aplicaciones Android, que funcionan en más de 2.5 mil millones de dispositivos en todo el mundo — desde teléfonos y tabletas hasta televisores, relojes y autos. El curso está estructurado en unidades y caminos de aprendizaje que combinan videos, ejercicios prácticos (codelabs) y cuestionarios, incentivando el aprendizaje a través de insignias que reflejan el progreso y la adquisición de habilidades.

El objetivo principal es enseñar los fundamentos de programación y los conceptos clave dentro del desarrollo Android con un enfoque en el lenguaje Kotlin y el entorno Android Studio. No se requiere experiencia previa en programación, pero sí conocimientos básicos de computación y matemáticas, esenciales para resolver problemas de lógica y desarrollo.

Se destaca la importancia motivacional para completar el curso, y se aclara que este programa prepara sólidamente para continuar en niveles más avanzados, pero por sí solo no garantiza un empleo inmediato como desarrollador Android.

Video 2 ruta de Aprendizaje 1 (¿Que sigue?)

El curso incluye prácticas y un cuestionario evaluativo para reforzar el aprendizaje, permitiendo a los usuarios tomar el examen cuantas veces quieran y consultar el contenido si tienen dudas. Al finalizar, los estudiantes reciben una insignia digital que se añade a su perfil de Google Developer, destacando su progreso. Además, se anima a compartir estos logros en redes sociales usando el hashtag #AndroidBasics para fomentar la comunidad. Finalmente, se anticipa el siguiente módulo, en el cual se aplicará el conocimiento de Kotlin para construir la primera aplicación Android utilizando Android Studio, incluyendo la ejecución en un dispositivo real.

Puntos Clave

- Completar el módulo básico de Kotlin es un logro fundamental para el desarrollo en Android.
- Se aprende sobre programas simples, variables y funciones.
- Se incluye un cuestionario para evaluar y reforzar conocimientos.
- La finalización otorga una insignia visible en el perfil de Google Developer.
- El próximo paso es crear y ejecutar tu primera app en Android Studio.
- Se puede repetir el quiz y consultar el contenido para afianzar el aprendizaje.
- Se recomienda compartir los logros en redes sociales con el hashtag #AndroidBasics.

Cuestionario 1

1. ¿Cuál de las siguientes declaraciones de variable es válida?

☐ `var hello: Int? = ""`

☐ `String "hello" = hello`

☒ `val hello = "hello"`

✓ ¡Correcto!

☐ `hello: String = "hello"`

2. Se considera una práctica recomendada declarar una variable que no cambiará cuando se usa `var` en lugar de `val`.

☐ Verdadero

☒ Falso

✓ ¡Correcto!

3. ¿Cuáles de las siguientes opciones son formas válidas de actualizar una variable?

Selecciona todas las respuestas que consideres correctas.

☒ `total++`

✓ ¡Correcto!

☐ `total - 1`

☒ `total--`

✓ ¡Correcto!

4. En Kotlin, los comentarios pueden ser de una o varias líneas, y el compilador los ignora.

☒ Verdadero

✓ ¡Correcto!

☐ Falso

5. ¿Cuál de las siguientes opciones no es un tipo de datos en Kotlin?

☐ `String`

☒ `Decimal`

✓ ¡Correcto!

☐ `Int`

☐ `Boolean`

6. `Float` también representa un decimal, pero es menos preciso que `Double`.

☒ Verdadero

✓ ¡Correcto!

☐ Falso

7. En Kotlin, el punto de entrada de un programa es ____.

☐ Sentencia `println()`

☐ Variable `val`

☒ Función `main()`

✓ ¡Correcto!

☐ Sentencia `return`

8. ¿Cuáles de las siguientes afirmaciones sobre los valores que se muestran de las funciones son verdaderas?

Selecciona todas las respuestas que consideres correctas.

☒ Parcialmente correcto.

☒ Si una función no especifica un tipo de datos que se muestra, este es `Unit`.

✓ ¡Correcto!

☒ Se puede almacenar un valor que se muestra en una variable.

✓ ¡Correcto!

☐ Las funciones con un tipo de datos que se muestra `Unit` deben incluir una sentencia `return`.

☐ Un tipo de valor que se muestra debe coincidir con el tipo de datos que se muestra de una función.

9. ¿Cuáles de las siguientes afirmaciones sobre las funciones son verdaderas?

Selecciona todas las respuestas que consideres correctas.

☒ Las funciones pueden tomar parámetros o variables como entradas.

✓ ¡Correcto!

☐ Los parámetros de la función deben tener argumentos predeterminados.

☒ Cuando se llama a una función con parámetros, los valores que se pasan se denominan argumentos.

✓ ¡Correcto!

☒ Dividir el código en funciones independientes facilita su mantenimiento.

✓ ¡Correcto!

10. Con los argumentos con nombre, puedes cambiar el orden en que pasas los argumentos a una función.

☒ Verdadero

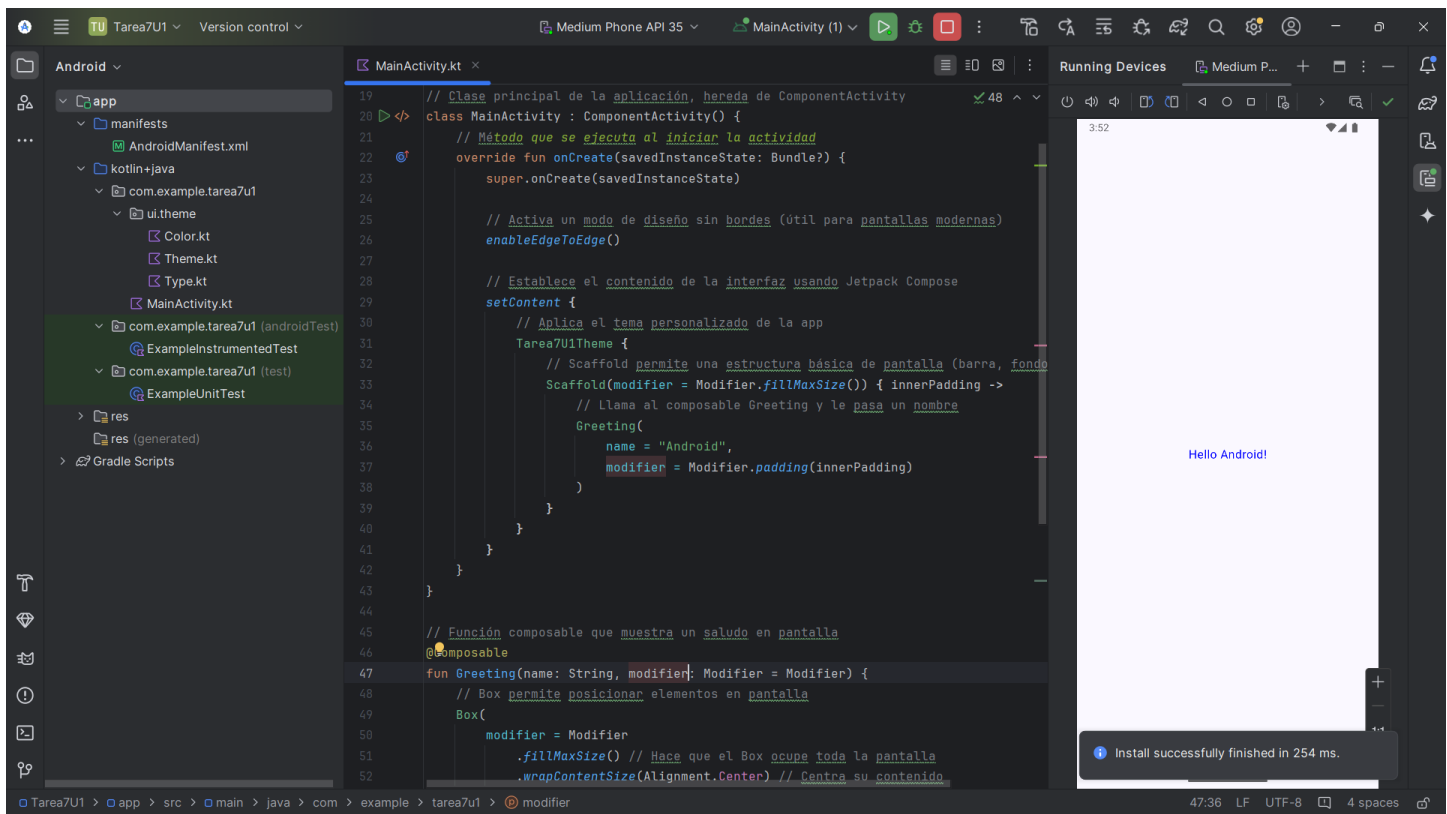
✓ ¡Correcto!

☐ Falso

Results

Tu puntuación es **9 de 10**. ¡Felicitaciones! Aprobaste el cuestionario.

INTRODUCCION A KOTLIN



Este código crea una app sencilla con Jetpack Compose que muestra un texto centrado en pantalla con el mensaje "Hello Android!" en color azul.

¿Qué hace cada parte?

- **MainActivity:** Es el punto de entrada. Usa setContent para definir la interfaz gráfica con Compose.
- **Tarea7U1Theme:** Aplica el tema visual de tu app (colores, tipografía, etc.).
- **Scaffold:** Estructura base de UI, útil si más adelante agregas barras o botones.
- **Greeting:** Función que muestra un mensaje con el nombre pasado como parámetro.
- **GreetingPreview:** Permite ver cómo se verá la UI directamente en el editor sin ejecutar la app.

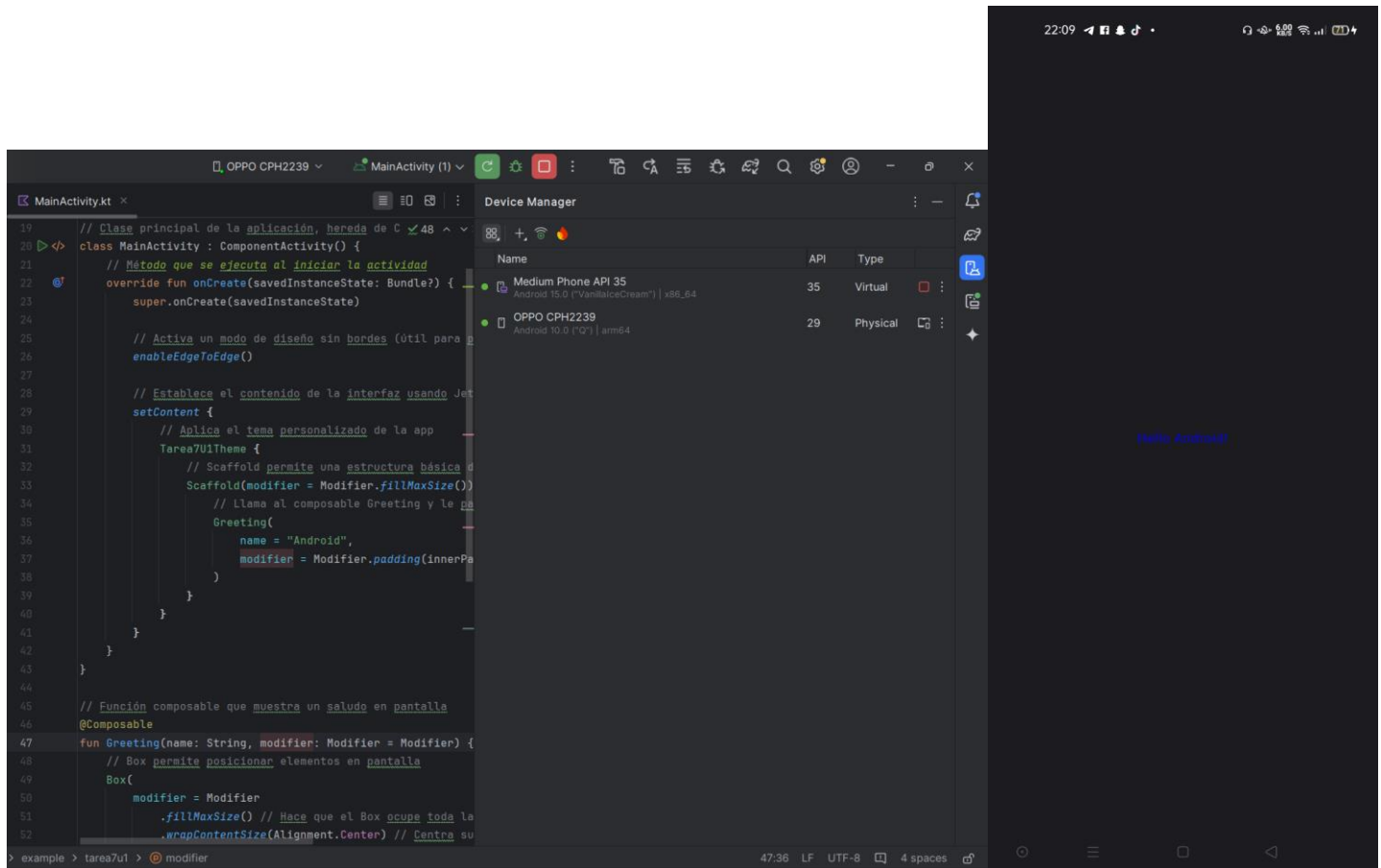
ruta 2 - CONFIGURACION DE ANDROID STUDIO

Video 1 – Ruta de Aprendizaje 2 (Introducción a Android Studio)

Android Studio es un IDE (Entorno de Desarrollo Integrado) que reúne múltiples herramientas de programación en un solo lugar, facilitando la creación y prueba de aplicaciones. Se explica que este entorno ofrece funcionalidades como revisión automática de errores en el código y sugerencias para acelerar el proceso de escritura. Además, se destaca que Android Studio permite ejecutar y probar las aplicaciones directamente en dispositivos Android reales o en un emulador, que simula un dispositivo en la computadora. Usar este IDE reduce errores y hace el desarrollo mucho más eficiente. Durante el curso, los estudiantes aprenderán a familiarizarse con Android Studio y tendrán numerosas oportunidades para practicar.

- **Android Studio como entorno integrado:** La integración de múltiples herramientas en un solo programa hace que el proceso de desarrollo sea mucho más fluido. No es necesario cambiar entre diferentes softwares para escribir código, diseñar interfaces o probar aplicaciones, lo que incrementa la productividad

y reduce errores. Esto es especialmente útil para quienes están aprendiendo, ya que el entorno está pensado para asistir durante todo el desarrollo.



Conexión de mi dispositivo Oppo CPH2239 a Android Studio activando el modo desarrollador y la depuración USB

PASOS PARA CONECTAR TU DISPOSITIVO A ANDROID STUDIO

1. Activa las opciones de desarrollador en tu celular:

1. Abre la app Configuración (Ajustes).
2. Ve a Acerca del teléfono.
3. Toca varias veces donde dice "Número de compilación" (Build Number) hasta que aparezca "¡Ahora eres un desarrollador!".

2. Activa la depuración por USB:

1. Regresa a Configuración.
2. Entra en Opciones de desarrollador (o "Opciones para desarrolladores").
3. Activa Depuración por USB (USB Debugging).

3. Conecta tu teléfono a la computadora por USB.

- Usa un cable de datos original o funcional.
- Asegúrate de seleccionar "Transferencia de archivos" o "Transferencia de datos" como modo de conexión USB.

4. Confirma la conexión desde el dispositivo:

- Aparecerá un mensaje en tu teléfono que dice: "¿Permitir depuración USB?"
- Presiona "Permitir" (y puedes marcar "Recordar esta computadora").

5. Android Studio reconocerá tu dispositivo:

- En la barra superior de Android Studio (donde seleccionas el emulador), aparecerá el nombre de tu dispositivo.
- Selecciónalo.

6. Ejecuta tu app:

- Haz clic en Run (botón de ejecutar).
- Android Studio compilará y enviará tu app directamente al dispositivo.

RUTA 3 - CREAR UN DISEÑO BASICO

Video 1 – Ruta de aprendizaje 3 (Crear un Diseño Básico)

El video presenta Jetpack Compose, una moderna herramienta para construir interfaces de usuario (UI) en aplicaciones Android. Desarrollada por el equipo de Android con más de una década de experiencia, Compose es una biblioteca de código abierto que forma parte del conjunto Jetpack y permite diseñar la UI utilizando Kotlin, el mismo lenguaje que se usa para la lógica del app. Esto facilita la comunicación entre la UI y el resto de la aplicación. A diferencia del sistema anterior basado en Views, Compose utiliza un enfoque declarativo con funciones anotadas como `@Composable`, que describen visualmente cada parte de la interfaz sin devolver un valor directo. La UI es inmutable y se actualiza automáticamente mediante un proceso llamado recomposición, que reactualiza solo las partes que han cambiado, optimizando el rendimiento. Compose incluye componentes integrados (botones, textos, imágenes, layouts como filas y columnas) que pueden configurarse mediante parámetros para adaptarse a las necesidades. Además, a lo largo del curso se aprenderán funciones avanzadas como listas, temas, animaciones y pruebas UI. Compose simplifica y agiliza la creación de interfaces en Android, promoviendo una experiencia de desarrollo más intuitiva y eficiente.

Video 2 – Ruta de aprendizaje 3 (Diseña una app de tarjeta de cumpleaños)

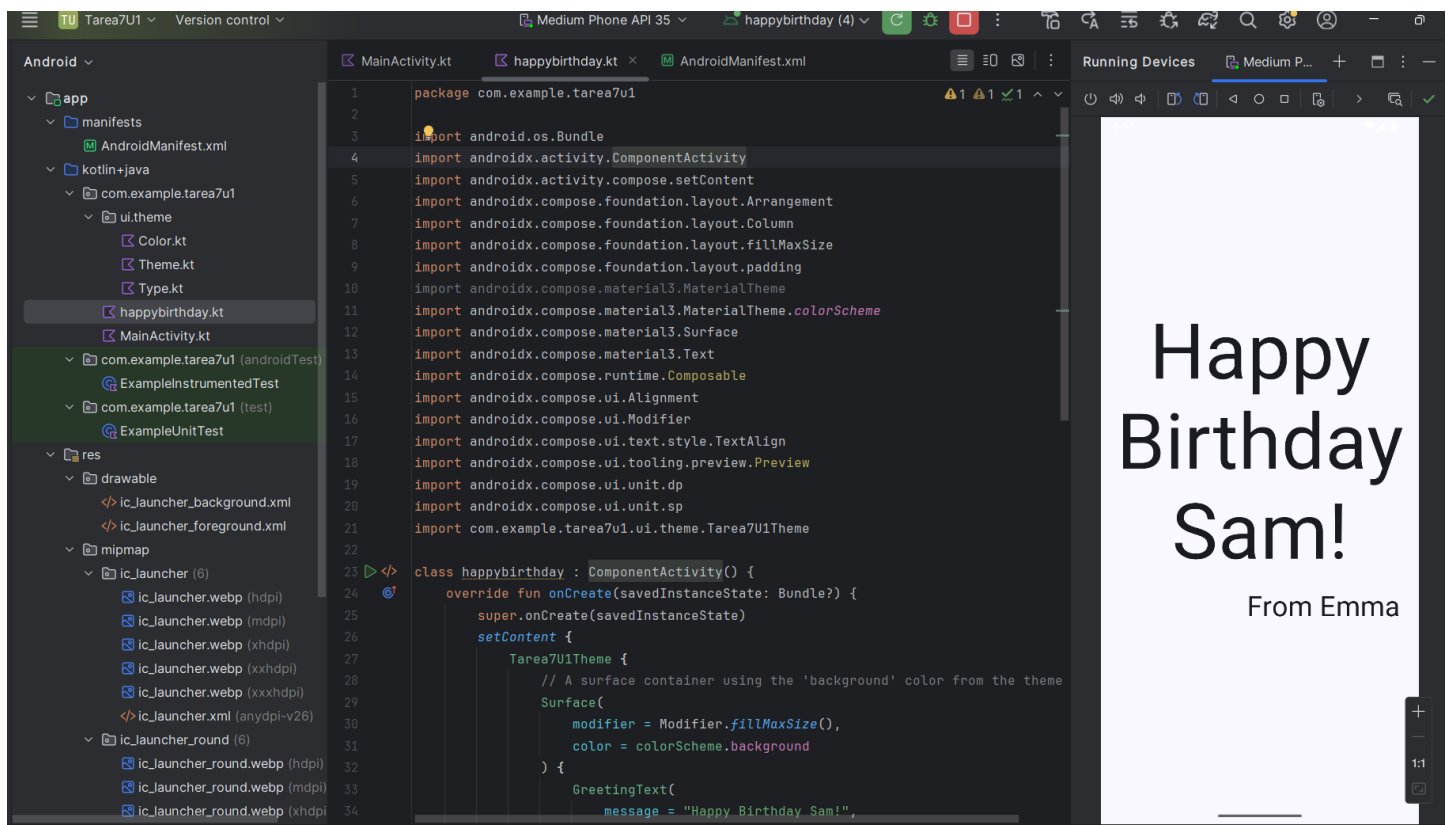
El video comienza enfatizando que el primer paso para crear una aplicación es tener una idea increíble, y en este caso, la idea es desarrollar una app que celebre un día especial: el cumpleaños del usuario. Antes de construir la app, es vital entender qué elementos debe contener. Se realiza una analogía con la cocina, donde para preparar un platillo se necesitan ingredientes específicos y un método para combinarlos; de igual forma, para hacer una app exitosa hay que planificar sus componentes y funciones.

Se invita a reflexionar sobre las aplicaciones favoritas que uno utiliza, identificando que las mejores son efectivas, eficientes, visualmente agradables y accesibles. Estas características aseguran que el usuario alcance su objetivo con rapidez y facilidad en un entorno organizado. Sin embargo, es importante primero definir cuál es la tarea o “trabajo” que la app debe cumplir, sin necesidad aún de programar.

Se utiliza como ejemplo una tarjeta de cumpleaños para ilustrar qué elementos suelen tener: un mensaje de “feliz cumpleaños”, un saludo personalizado con el nombre y edad del cumpleaños, y una firma con el nombre de quien envía la tarjeta. La app propuesta tendrá una imagen grande y texto organizado, animando al usuario a personalizarla con imágenes y mensajes propios, además de esbozar o dibujar diferentes ideas para la tarjeta digital.

En resumen, el video aborda la importancia de planificar detalladamente antes de desarrollar una aplicación, centrándose en la experiencia del usuario y fomentando la creatividad en el diseño de una app personalizada de tarjetas de cumpleaños.

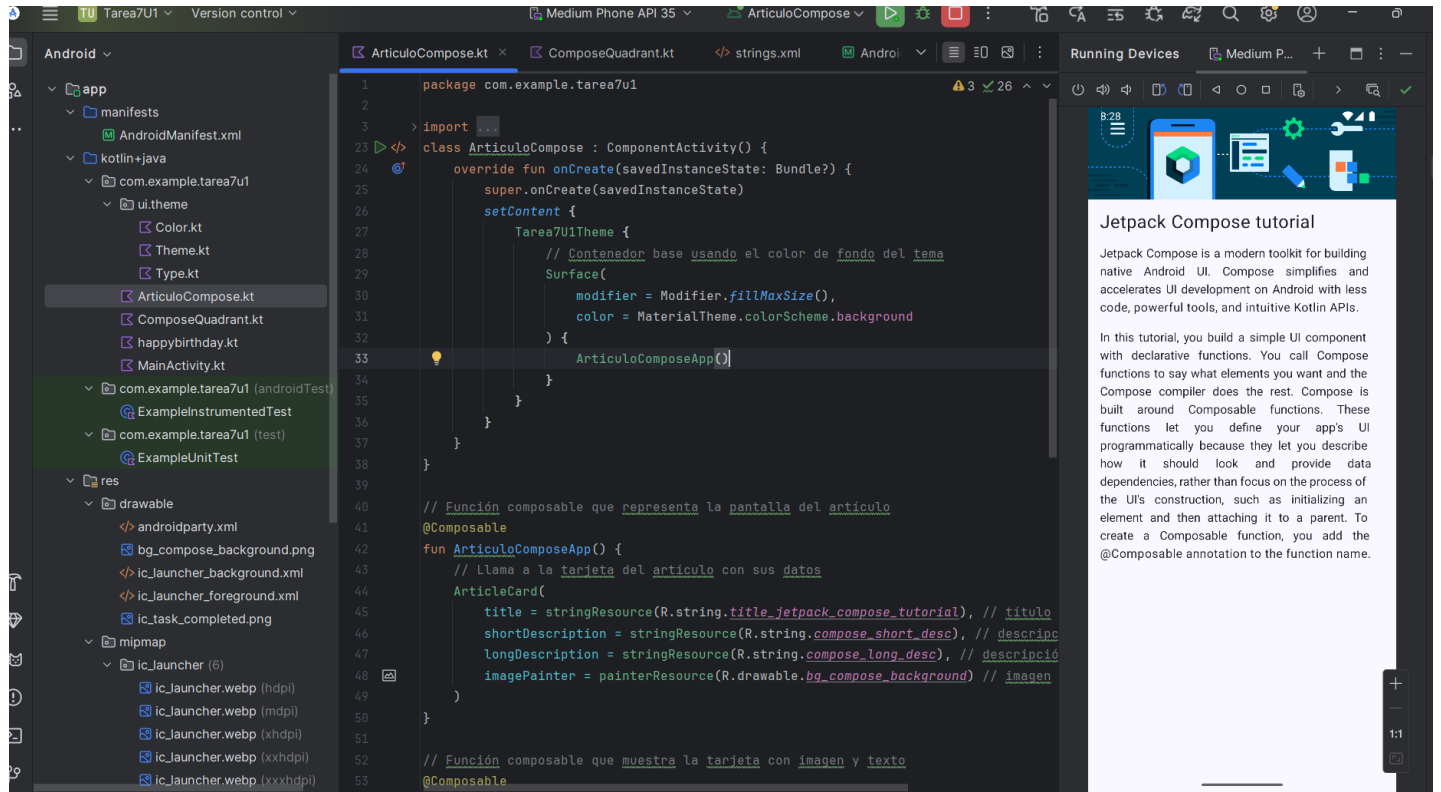
TARJETA DE CUMPLEAÑOS



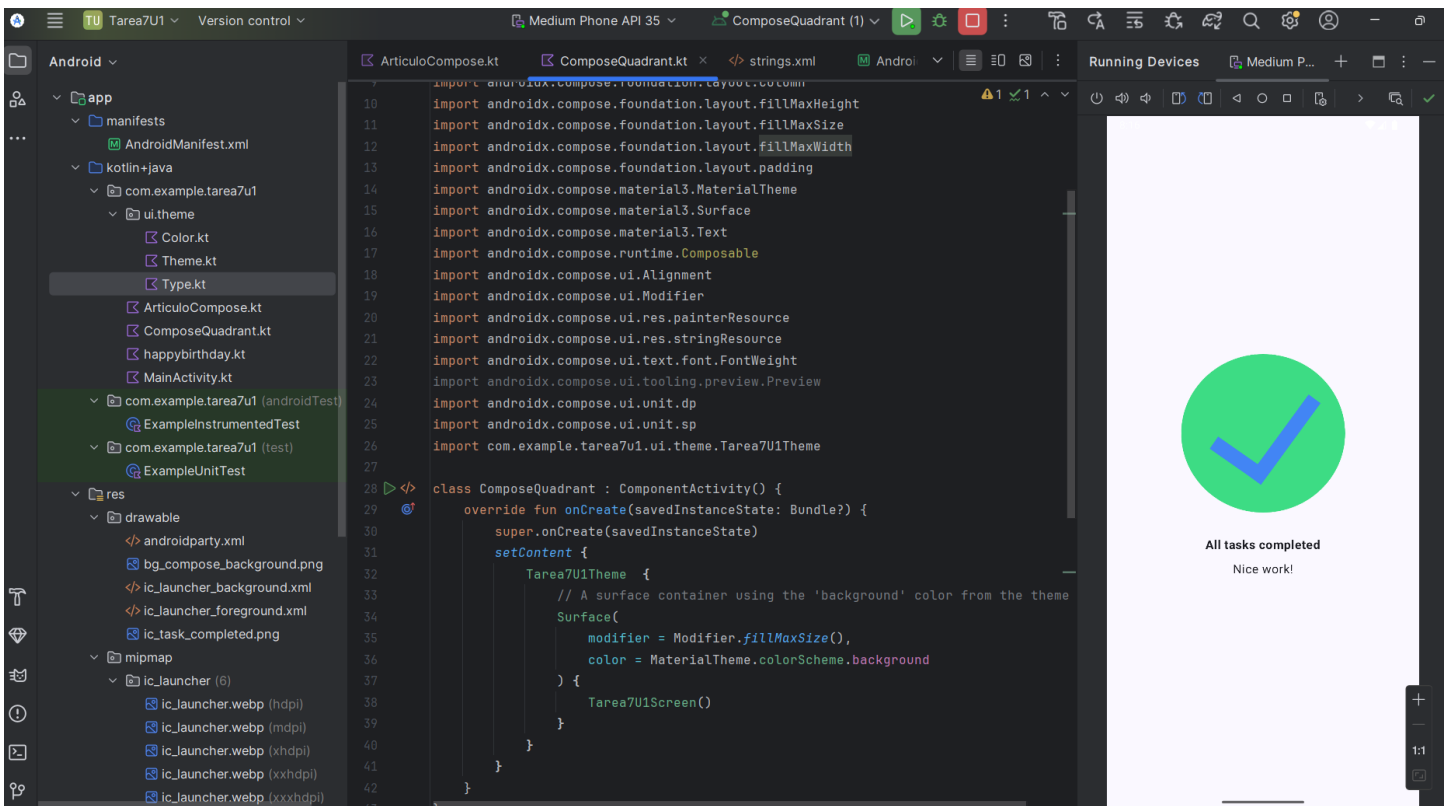
Este código define una pequeña aplicación de cumpleaños en Jetpack Compose. Muestra un mensaje de felicitación como tarjeta de cumpleaños con dos textos: el mensaje principal y el remitente.

- **happybirthday:** Es la actividad principal que configura la interfaz usando `setContent` con un tema personalizado (`Tarea7U1Theme`), un contenedor `Surface`, y un composable llamado `GreetingText`.
- **GreetingText:** Es un composable personalizado que organiza el mensaje de cumpleaños y el nombre del remitente en una columna. Usa estilos como tamaño de fuente, alineación y padding.
- **BirthdayCardPreview:** Es una función anotada con `@Preview`, que permite ver una vista previa del diseño en Android Studio sin necesidad de ejecutar la app en un emulador o dispositivo físico.

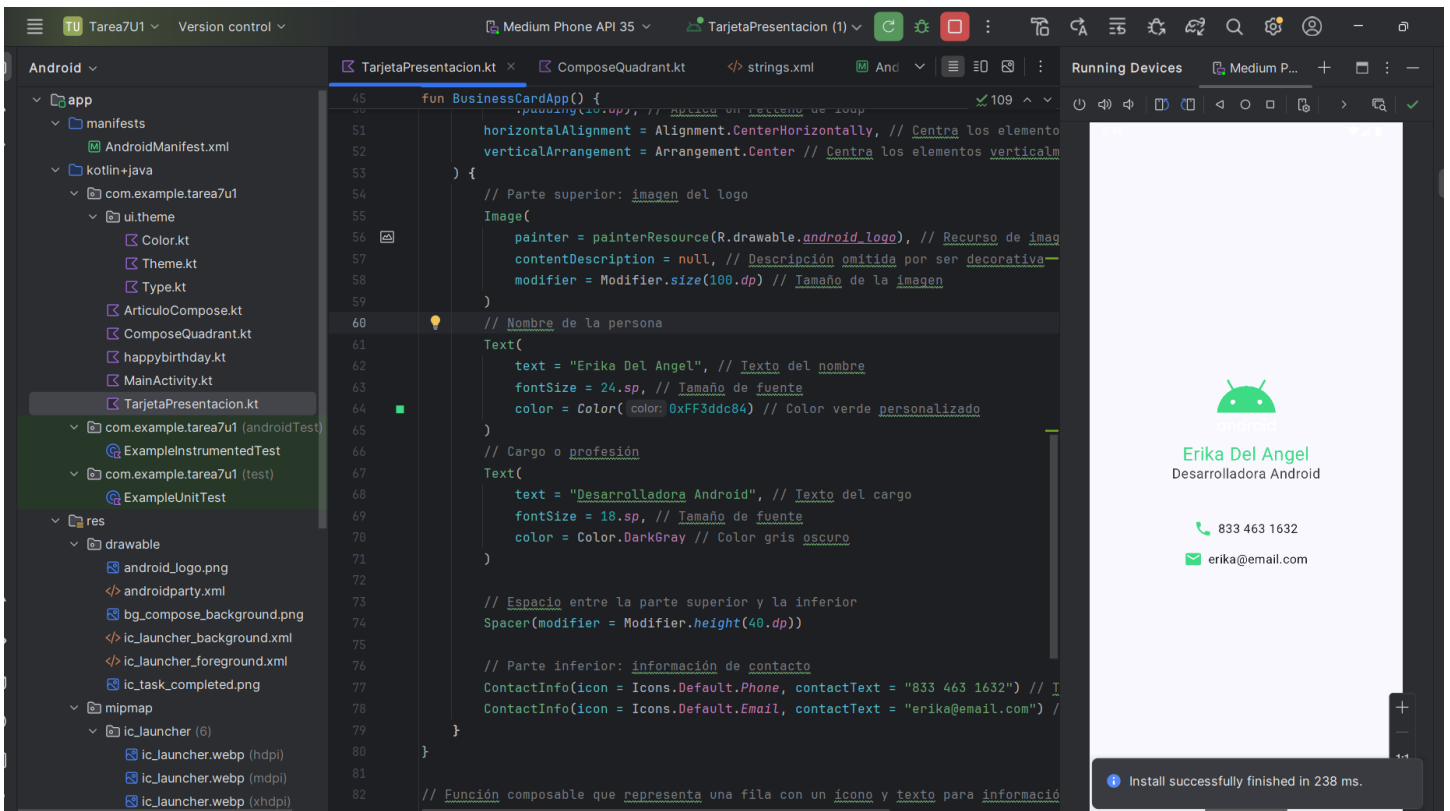
ARTICULO DE COMPOSE



ADMINISTRADOR DE TAREAS



PROYECTO: CREA UNA APP DE TARJETA DE PRESENTACIONES



Este código crea una app con una tarjeta de presentación digital usando Jetpack Compose. Muestra:

- Un logo en la parte superior.
- El nombre y el cargo de la persona.
- Dos líneas de información de contacto: teléfono e email.

Está dividida en una columna con espaciado y alineación centrada. Usa componentes como Text, Image, Icon, Row y Column para organizar los elementos de manera visualmente clara y profesional.

CUESTIONARIO RUTA 3

1. What is Jetpack Compose?

☒ A Modern toolkit to develop Android UI ¡Correcto!

☐ Toolkit to design libraries

☐ Database Interface

☐ Plugin to build APK

2. Composable functions are the basic building block of Compose.

☒ True ¡Correcto!

☐ False

3. What annotation is used to annotate a Composable function?

☐ @Annotation

☐ @ComposableFunction

☒ @Composable ¡Correcto!

☐ @Preview

4. The three basic standard layout elements in Compose are

Selecciona todas las respuestas que consideres correctas.

☒ Column ¡Correcto!

☒ Row ¡Correcto!

☐ Text

☒ Box ¡Correcto!

5. What is the tool window for importing, creating, managing, and using resources in your app?

☐ Application Manager

☒ Resource Manager ¡Correcto!

☐ Resource Tool

☐ Layout Manager

6. Which class is an automatically generated class by Android that contains the IDs of all resources in the project.

- ☐ The `Android` class
- ☒ The `Resource` class Incorrecto.
- ☐ The `R` Class
- ☐ The `ResourceID` class

7. Which function is used is to load a drawable image resource.

- ☐ The `stringResource()` function
- ☒ The `painterResource()` function ¡Correcto!
- ☐ The `ImageResource()` function
- ☐ The `loadResource()` function

8. What is the function parameter used to add accessibility text, used by talkback?

- ☐ `accessibilityText`
- ☐ `contentText`
- ☐ `accessibilityDescription`
- ☒ `contentDescription` ¡Correcto!

9. The Box layout stacks the UI elements on top of one another.

- ☒ True ¡Correcto!
- ☐ False

10. What parameter is used to align the child element to the beginning of the parent?

- ☐ `Alignment.End`
- ☐ `Alignment.Begin`
- ☒ `Alignment.Start` ¡Correcto!
- ☐ `Alignment.Top`

Results

Tu puntuación es **9 de 10**. ¡Felicitaciones! Aprobaste el cuestionario.