

Contenido

Introducción	2
Conceptos básicos de Compose	3
Codelab (Como escribir tu primera app de compose)	3
Codelab(Como implementar un diseño real)	4
Codelab (Como comenzar a usar el estado).....	5
Codelab(Migración práctica)	8
Cuestionario de Conceptos básicos de Compose.....	10
Diseños, temas y animaciones	10
Codelab(Como aplicar un tema en tu app con Material Desing 3)	10
Codelab(Como animar elementos)	12
Cuestionario de Diseños, temas y animación	14
Arquitectura y estado	14
Codelab(Efectos Secundarios y estados avanzados)	14
Codelab (Navegación)	16
Cuestionario de Arquitectura y estado.....	19
Accesibilidad, pruebas y rendimiento	19
Codelab (Pruebas)	19
Codelab (Accesibilidad)	21
Cuestionario de Accesibilidad, pruebas y rendimiento.....	23
Factores de formas	23
Codelab (Compila apps adaptables).....	23
Codelab (Wear OS con Compose)	25
Cuestionario de Factores de forma	28
Conclusión	29

Introducción

En el mundo actual del desarrollo de aplicaciones móviles, la eficiencia y la calidad de la experiencia del usuario son aspectos cruciales. Jetpack Compose ha emergido como una herramienta innovadora que permite a los desarrolladores crear interfaces de usuario de manera más sencilla y efectiva. Esta bitácora recopila los aprendizajes adquiridos a través de una serie de cursos y codelabs enfocados en Jetpack Compose, abarcando desde sus fundamentos hasta aplicaciones específicas en dispositivos wearables.

Los cursos están diseñados para guiar a los participantes en la implementación de prácticas de diseño modernas, utilizando Material Design y adaptando las aplicaciones a diferentes plataformas, incluidas las apps para Wear OS. A través de la exploración de temas como el manejo del estado, la optimización del rendimiento, la accesibilidad y la adaptabilidad, los desarrolladores pueden transformar su enfoque hacia la creación de aplicaciones, asegurando que sean no solo funcionales, sino también inclusivas y atractivas.

La estructura de estos cursos combina teoría y práctica, permitiendo a los estudiantes aplicar inmediatamente lo aprendido en proyectos reales. En esta bitácora, se documentarán los conceptos clave, las técnicas adquiridas y las experiencias vividas durante el proceso de aprendizaje, proporcionando una referencia valiosa para futuros desarrollos y la evolución profesional en el ámbito del desarrollo de aplicaciones Android.

Conceptos básicos de Compose

Codelab (Como escribir tu primera app de compose)

En este codelab se dio un pequeño repaso de lo que es la herramienta Jetpack Compose, Combina un modelo de programación reactivo con la concisión y facilidad de uso del lenguaje de programación Kotlin. Es totalmente declarativo, lo que significa que describes tu IU si llamas a una serie de funciones que transforman datos en una jerarquía de IU. Cuando los datos subyacentes cambian, el framework vuelve a ejecutar estas funciones automáticamente y actualiza la jerarquía de la IU por ti.

En este codelab, se aprendió lo siguiente:


- Qué es Compose
- Cómo compilar IUs con Compose
- Cómo administrar el estado en funciones de componibilidad
- Cómo crear una lista de rendimiento
- Cómo agregar animaciones
- Cómo aplicarle estilos y temas a una app

The screenshot shows the Codelab interface for 'Cómo escribir tu primera app de Compose'. At the top, there's a title bar with a green checkmark icon, the title 'Cómo escribir tu primera app de Compose', the Codelab logo, and a back arrow. Below the title bar, a description states: 'Obtén experiencia práctica y aprende los aspectos básicos de la IU declarativa, el trabajo con el estado, los diseños y los temas. Verás qué son los elementos componibles y los modificadores, cómo trabajar con elementos básicos de la IU como Row y Column, y cómo asignarle un estado a tu app.' A 'Realizar codelab' button is on the right. Below this is a progress bar showing 'Aspectos básicos de Jetpack Compose' and a timer 'Queda 1 minuto'. On the left, a list of 14 steps is shown, with step 6 'Cómo crear columnas y filas' selected. The main content area shows a terminal with the command '\$ git clone https://github.com/android/codelab-android-compose' and a list of links for '¿Qué sigue?'. A modal overlay in the center-right says '¡Obtuviste una insignia En aprendizaje!' and includes a 'Ver perfil' button. At the bottom, there are 'Atrás', 'Descartar', and 'Listo' buttons.

Codelab(Como implementar un diseño real)

Contenido del Codelab

1. **Requisitos Previos:** Antes de iniciar este codelab, es esencial haber completado el codelab de aspectos básicos, donde se introducen conceptos fundamentales como:
 - **Surfaces:** Componente base para crear áreas visuales.
 - **Rows y Columns:** Estructuras de diseño que organizan los elementos en filas y columnas.
 - **Modificadores:** Herramientas para alterar y ajustar el comportamiento y la apariencia de los componentes, como padding, fillMaxWidth y size.
2. **Objetivo del Codelab:** En este codelab, se busca implementar un diseño más complejo y realista que el aprendido anteriormente. Durante el curso, se explorarán:
 - **Elementos Componibles:** Componentes listos para usar que facilitan la creación de interfaces.
 - **Modificadores Avanzados:** Nuevos modificadores que permiten un mayor control sobre el diseño y la disposición de los elementos.
3. **Resultados Esperados:** Al finalizar este codelab, los participantes deberán ser capaces de convertir un diseño básico de aplicación en un código funcional utilizando Jetpack Compose. Esto incluye una comprensión más profunda de cómo trabajar con elementos composables y modificadores para crear interfaces de usuario atractivas y eficientes.



Cómo implementar un diseño real

En este codelab, aprenderás a implementar diseños reales con elementos componibles y modificadores que Compose proporciona de forma predeterminada.

Realizar codelab

X Diseños básicos en ComposeQueda 1 minutoEspañol - A

4 Barra de búsqueda: Modificadores

5 Align your body: Alineación

6 Tarjeta de Favorite collection: Elemento Surface de Material

7 Fila de Align your body: Disposiciones

8 Cuadrícula de Favorite collections: Cuadrículas diferidas

9 Sección principal: APIs de ranuras

10 Pantalla principal: Desplazamiento

11 Navegación inferior: Material

12 App de MySoothe: Scaffold

13 Fila de navegación: Material

14 App de MySoothe: Tamaño de la ventana

15 Felicitaciones

15. Felicitaciones

¡Felicitaciones! Completaste correctamente este codelab y aprendiste más sobre los diseños en Compose. A través de la implementación de un diseño real, aprendiste sobre los modificadores, las alineaciones, las disposiciones, los diseños diferidos, las APIs de ranuras, el desplazamiento, los componentes de Material y los diseños específicos.


Consulta los otros codelabs sobre la [ruta de aprendizaje de Compose](#). No olvides consultar las [muestras de código](#) también.

Documentación

Para obtener más información y orientación sobre estos temas, consulta la siguiente documentación:

- [Diseños en Compose](#)
- [Modificadores](#)
- [Cómo brindar compatibilidad con diferentes tamaños de pantalla](#)

¡Obtuviste otra insignia En aprendizaje!



[Volver a la ruta de aprendizaje](#)

[Ver perfil](#)

[Descartar](#)

[Atrás](#)

[Listo](#)

[Informar un error](#)

Codelab (Como comenzar a usar el estado)

Este codelab está diseñado para profundizar en los conceptos clave relacionados con el manejo del estado en Jetpack Compose, una herramienta esencial para construir interfaces de usuario reactivas y dinámicas en aplicaciones Android.

Contenido del Codelab

1. Conceptos de Estado y Eventos:

- **Pensamiento en Estado:** Se aprenderá a conceptualizar cómo el estado de la aplicación influye en los elementos visibles de la IU, incluyendo cómo y cuándo se deben actualizar.

2. Determinación de Elementos con Estado:

- **Uso del Estado en Compose:** Se explorará cómo Jetpack Compose utiliza el estado para decidir qué elementos deben mostrarse en la pantalla, permitiendo una UI reactiva que responde a cambios en el estado de la app.

3. Elevación de Estado:

- **Qué es la Elevación de Estado:** Se explicará el concepto de elevar el estado a un nivel superior en la jerarquía de componentes, lo que permite compartir el estado entre diferentes partes de la IU.

4. Funciones de Componibilidad:

- **Con y Sin Estado:** Se abordará cómo funcionan las funciones de componibilidad, tanto aquellas que manejan su propio estado como las que no lo hacen, facilitando un diseño más claro y mantenible.

5. Seguimiento Automático del Estado:

- **API de State<T>:** Se aprenderá cómo Compose maneja automáticamente el seguimiento del estado utilizando la API de State<T>, lo que simplifica el proceso de actualización de la IU.

6. Memoria y Estado Interno:

- **APIs de remember y rememberSaveable:** Se analizará cómo gestionar la memoria y el estado interno en las funciones de componibilidad mediante estas APIs, lo que permite conservar el estado durante cambios de configuración.

7. Trabajo con Listas y Estados:

- **APIs mutableStateListOf y toMutableStateList:** Se enseñará a trabajar con listas de estado utilizando estas APIs, lo que facilita la manipulación y visualización de datos dinámicos en la IU.

8. Integración de ViewModel:

- **Uso de ViewModel con Compose:** Se explorará cómo integrar ViewModels en el contexto de Compose, promoviendo la separación de la lógica de negocio de la UI y facilitando el manejo del estado de manera efectiva.

Objetivos de Aprendizaje

Al finalizar este codelab, los participantes deberían:

- Comprender cómo el estado afecta la IU y cómo manejar eventos dentro de Compose.
- Ser capaces de implementar y gestionar el estado en sus aplicaciones utilizando las herramientas y APIs que ofrece Jetpack Compose.
- Adquirir habilidades para optimizar la estructura de sus funciones de componibilidad, utilizando patrones de elevación de estado y el modelo de ViewModel.



Cómo comenzar a usar el estado

Codelab

Aprende cómo el estado de la app determina qué se muestra en la IU, cómo Compose mantiene actualizada la IU cuando cambia el estado, cómo optimizar la estructura de tus funciones de componibilidad y cómo trabajar con ViewModels en una app de Compose.

Realizar codelab

13. Felicitaciones

¡Bien hecho! Completaste correctamente este codelab y aprendiste todas las APIs básicas para trabajar con el estado de una app de Jetpack Compose.

Aprendiste a pensar en el estado y los eventos para extraer elementos de componibilidad sin estado en Compose, y cómo Compose usa las actualizaciones de estado a fin de impulsar el cambio en la IU.

¿Qué sigue?

Consulta los otros codelabs sobre la [ruta de aprendizaje de Compose](#).

Apps de ejemplo

- [JetNews](#) muestra las prácticas recomendadas que se explican en este codelab.

Más documentación

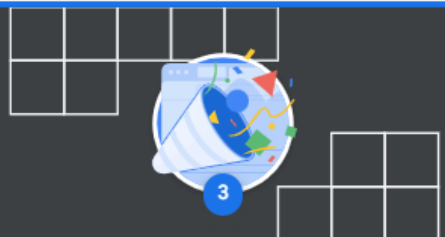
- [Cómo pensar en Compose](#)
- [El estado y Jetpack Compose](#)
- [Flujo de datos unidireccional en Jetpack Compose](#)

APIs de

- [r](#)
- [r](#)
- [S](#)
- [M](#)
- [S](#)

¡Obtuviste otra insignia En aprendizaje!

Tu perfil obtuvo el logro.



[Volver a la ruta de aprendizaje](#)

Ver perfil



[Descartar](#)

Codelab(Migración práctica)

Este codelab está diseñado para guiar a los desarrolladores en el proceso de migrar aplicaciones existentes de la interfaz basada en Views a Jetpack Compose, utilizando como ejemplo la pantalla de detalles de plantas de la aplicación Sunflower.

Contenido del Codelab

1. Integración de Compose y View:

- **Compatibilidad:** Se explicará cómo Jetpack Compose y el sistema de Views de Android pueden coexistir en una misma aplicación, permitiendo una migración gradual y flexible.

2. Migración de la App Sunflower:

- **Proyecto de Ejemplo:** Se proporcionará una copia del proyecto Sunflower, permitiendo a los participantes experimentar la migración de una app realista a Compose.

3. Objetivos de Aprendizaje: Al finalizar este codelab, los participantes aprenderán a:

- **Rutas de Migración:** Identificar las diferentes estrategias que se pueden seguir al migrar a Compose, lo que les permitirá elegir el enfoque que mejor se adapte a sus necesidades.
- **Migración Incremental:** Comprender cómo llevar a cabo la migración de una aplicación de manera incremental, lo que facilita el proceso y minimiza el riesgo de introducir errores.
- **Agregar Compose a Pantallas Existentes:** Aprender cómo incorporar Compose a una pantalla que ya está construida utilizando Views, lo que permite aprovechar ambas tecnologías durante la transición.
- **Uso de Objetos View en Compose:** Conocer cómo utilizar objetos de Views tradicionales desde el entorno de Compose, asegurando que las funciones existentes no se vean interrumpidas durante la migración.
- **Creación de Temas en Compose:** Explorar cómo definir y aplicar temas en Compose, lo que ayuda a mantener la coherencia visual a lo largo de la aplicación.
- **Pruebas de Pantallas Mixtas:** Aprender a probar pantallas que contienen tanto Views como Compose, garantizando que ambos componentes funcionen correctamente juntos.



Migración práctica

Codelab Opcional



Analiza una migración práctica de una app basada en objetos View a Jetpack Compose para comprender cómo adoptar Compose cada vez más, y explora las implicaciones para tu arquitectura y tus pruebas.

Realizar codelab

X Cómo migrar a Jetpack Compose

⌚ Queda 1 minuto

- 1. Introducción
- 2. Estrategia de migración
- 3. Cómo prepararte
- 4. Compose en Sunflower
- 5. ¡Hola, Compose!
- 6. Cómo crear una función de componibilidad a partir de XML
- 7. ViewModels y LiveData
- 8. Más migraciones de código XML
- 9. Vistas en el código de Compose
- 10. ViewCompositionStrategy y
- 11. Temas de Material
- 12. Prueba
- 13. Felicitaciones

13. Felicitaciones

¡Felicitaciones! Completaste este codelab con éxito.

La rama [compose](#) del proyecto original de GitHub de Sunflower migra por completo la pantalla de detalles de las plantas a Compose. Además de lo que hiciste en este codelab, también simula el comportamiento del CollapsingToolbarLayout. Esto incluye lo siguiente:

- Carga de imágenes con Compose
- Animaciones
- Mejor control de las dimensiones
- Y mucho más

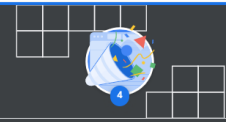
¿Qué sigue?

Consulta los otros codelabs sobre la [ruta de aprendizaje de Compose](#):

- [Conceptos básicos de Compose](#)
- [Diseños de Compose](#)
- [Temas de Compose](#)

Lect... ¡Obtuviste otra insignia En aprendizaje!

Tu perfil obtuvo el logro.



[Volver a la ruta de aprendizaje](#)

[Ver perfil](#)

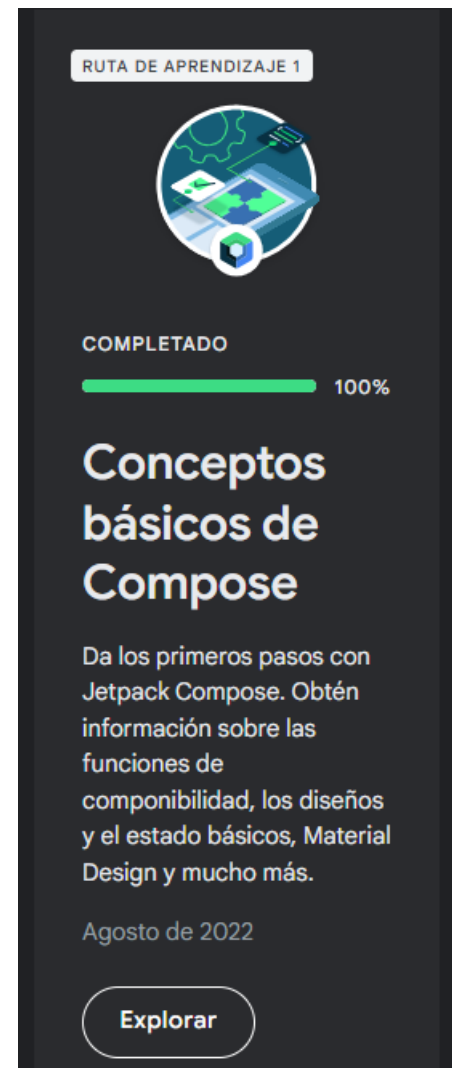


[Descartar](#)

Atrás

Listo

Cuestionario de Conceptos básicos de Compose



Diseños, temas y animaciones

Codelab(Como aplicar un tema en tu app con Material Design 3)

Este codelab está diseñado para enseñar a los desarrolladores cómo aplicar temas a sus aplicaciones utilizando Material Design 3 en Jetpack Compose. A través de este codelab, los participantes aprenderán sobre los componentes fundamentales que conforman un diseño atractivo y accesible.

Contenido del Codelab

1. Aspectos Clave de Material Design 3:

- Se presentarán los conceptos esenciales de los temas de Material Design 3, que proporcionan una base sólida para crear interfaces de usuario atractivas y funcionales.

2. Esquemas de Colores:

- **Generación de Temas:** Los participantes aprenderán sobre los diferentes esquemas de colores de Material Design 3 y cómo utilizar estos esquemas para generar temas personalizados para sus aplicaciones.

3. Soporte para Temas:

- **Temas Claro y Oscuro:** Se explorará cómo implementar soporte para temas claros, oscuros y dinámicos, permitiendo que la aplicación se adapte a las preferencias del usuario y a las condiciones de iluminación.

4. Tipografía y Formas:

- **Personalización de la App:** Los asistentes aprenderán a utilizar tipografía y formas de Material Design 3 para personalizar sus aplicaciones, asegurando una experiencia de usuario coherente y estilizada.

5. Componentes de Material 3:

- **Estilo y Personalización:** Se enseñará cómo trabajar con los diversos componentes de Material Design 3 y cómo personalizarlos para adaptarse al estilo de la aplicación, mejorando así la estética y la funcionalidad.


6. Temas Dinámicos y Niveles de Énfasis:

- **Compatibilidad con Temas Dinámicos:** Los participantes explorarán cómo implementar temas dinámicos que cambien en función de las preferencias del usuario y cómo aplicar diferentes niveles de énfasis para mejorar la jerarquía visual de la información.

Objetivos de Aprendizaje

Al finalizar este codelab, los participantes deberían ser capaces de:

- Comprender y aplicar los conceptos fundamentales de Material Design 3 en sus aplicaciones.
- Generar y utilizar esquemas de colores, soportar diferentes temas y personalizar la tipografía y formas.
- Mejorar la accesibilidad y la experiencia del usuario a través de la implementación de temas personalizados y dinámicos.



Cómo aplicar un tema en tu app con Material Design 3

Codelab

En este codelab, aprenderás a usar las APIs de asociación de temas de Jetpack Compose para definir el estilo de tu app. Veremos la personalización de colores, formas y tipografías para que se usen de manera coherente en toda tu app y admitan temas claros, oscuros y dinámicos

Realizar codelab

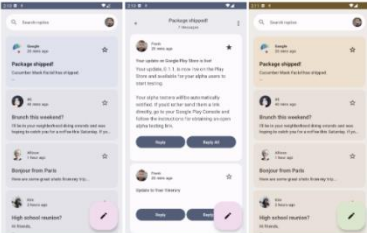
X Temas en Compose con Material 3

Queda 1 minuto Español - A

- Introducción
- Prepárate
- Temas de Material 3
- Esquemas de colores
- Cómo agregar colores dinámicos en la app
- Tipografía
- Formas
- Énfasis
- Felicitaciones

9. Felicitaciones

¡Felicitaciones! Completaste correctamente este codelab. Implementaste los temas de Material con Compose usando colores, tipografía y formas junto con colores dinámicos para aplicar temas en tu aplicación y ofrecer una experiencia personalizada.



Fin de la ruta de aprendizaje

Desbloqueaste un nuevo logro: En aprendizaje

Proximas acciones

Consultar

Volver a la ruta de aprendizaje Ver perfil

Discontinuar

Listo

Codelab(Como animar elementos)

Este codelab está diseñado para introducir a los desarrolladores en el uso de las APIs de animación disponibles en Jetpack Compose. Estas herramientas permiten crear interfaces de usuario más dinámicas y atractivas mediante el uso de animaciones efectivas.

Contenido del Codelab

1. Introducción a Jetpack Compose:

- Se proporcionará un contexto sobre Jetpack Compose como un kit de herramientas moderno que simplifica el desarrollo de interfaces de usuario en Android, destacando su enfoque en la creación de UI declarativas y eficientes.

2. APIs de Animación:

- Uso de Animaciones Básicas:** Los participantes aprenderán a utilizar varias APIs básicas de animación en Jetpack Compose, lo que les permitirá implementar animaciones sencillas y efectivas en sus aplicaciones.

Objetivos de Aprendizaje

Al finalizar este codelab, los participantes deberían ser capaces de:

- Comprender cómo funcionan las APIs de animación en Jetpack Compose.
- Implementar animaciones básicas para mejorar la interacción y la experiencia del usuario en sus aplicaciones.

Cómo animar elementos

Codelab

Aprende a usar las APIs de Compose Animation. Comenzaremos con la animación de valor más simple y aprenderemos otras API a través de algunos patrones de animación comunes. También analizaremos temas más avanzados, como las personalizaciones de animaciones y las animaciones con gestos táctiles.

Realizar codelab

X Cómo animar elementos en Jetpack Compose ⌚ Queda 1 minuto 🌐 Español - A... ⚙️

1 Introducción

2 Cómo prepararte

3 Cómo animar un cambio simple de valor

4 Cómo animar la visibilidad

5 Cómo animar el cambio de tamaño del contenido

6 Cómo animar varios valores

7 Cómo repetir animaciones

8 Animación gestual

9 ¡Felicitaciones!

9. ¡Felicitaciones!

¡Felicitaciones! Aprendiste las APIs básicas de Animation de Compose.

En este codelab, aprendimos a usar lo siguiente:

APIs de Animation generales:

- `AnimatedContentSize`
- `AnimatedVisibility`

APIs de Animation detalladas:

- `animate*asState` para animar un solo valor
- `updateTransition` para animar varios valores
- `InfiniteTransition` para animar valores indefinidamente
- `Animatable` para compilar animaciones personalizadas con gestos táctiles

¿Qué sigue?

Consulta la documentación de Jetpack Compose para obtener más información.

Para obtener más información, consulta la documentación de Jetpack Compose.

¡Obtuviste otra insignia! En aprendizaje!

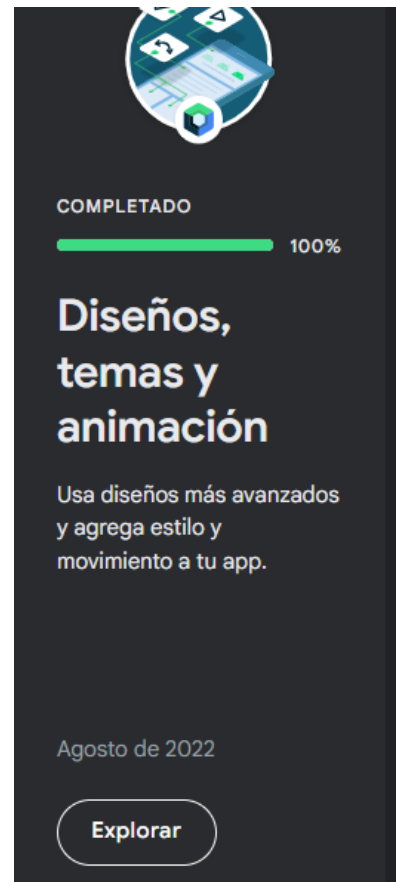
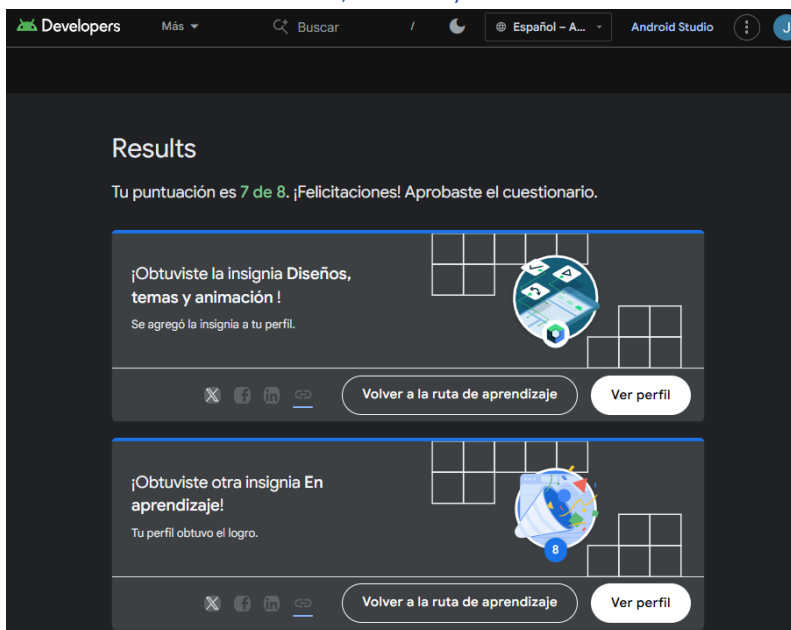
Volver a la ruta de aprendizaje Ver perfil

Atrás Descargar

Informar un error

Listo

Cuestionario de Diseños, temas y animación



Arquitectura y estado

Codelab(Efectos Secundarios y estados avanzados)

Este codelab está diseñado para profundizar en conceptos avanzados sobre la gestión del estado y la implementación de efectos secundarios en Jetpack Compose. Los participantes aprenderán a manejar elementos componibles con lógica compleja y a utilizar corrutinas para mejorar la interacción y el rendimiento de la UI.

Contenido del Codelab

1. Observación de Flujos de Datos:

- Se explicará cómo observar flujos de datos desde el código de Compose para mantener la interfaz de usuario actualizada de manera reactiva, garantizando que los cambios en el estado se reflejen automáticamente en la UI.

2. Contenedores de Estado:

- **Creación de Contenedores de Estado:** Los participantes aprenderán a crear contenedores de estado para elementos componibles que requieren lógica más compleja, lo que les permitirá gestionar el estado de manera más estructurada.

3. APIs de Efectos Secundarios:

- Se explorarán diversas APIs de efectos secundarios, incluyendo:
 - **LaunchedEffect:** Para ejecutar código que necesita efectos secundarios al ser lanzado.
 - **rememberUpdatedState:** Para recordar el estado actualizado dentro de la composición.
 - **DisposableEffect:** Para gestionar efectos que requieren limpieza al componer.
 - **produceState:** Para producir un estado que se puede observar.
 - **derivedStateOf:** Para crear un estado derivado basado en otros estados.

4. Uso de Corrutinas en Compose:

- **Creación de Corrutinas:** Los asistentes aprenderán a crear corrutinas y a llamar funciones de suspensión dentro de elementos componibles utilizando la API de `rememberCoroutineScope`. Esto les permitirá manejar operaciones asíncronas y realizar tareas de manera eficiente sin bloquear la UI.

Objetivos de Aprendizaje

Al finalizar este codelab, los participantes deberían ser capaces de:

- Observar y gestionar flujos de datos en Jetpack Compose para actualizar la interfaz de usuario.
- Implementar contenedores de estado para manejar lógica compleja en elementos componibles.
- Utilizar las APIs de efectos secundarios para implementar diversas funcionalidades y mejorar la estructura del código.
- Crear y manejar corrutinas en el contexto de Compose, permitiendo el uso de funciones de suspensión de manera efectiva.

Objetivos de Aprendizaje

Al finalizar este codelab, los participantes aprenderán:

1. Aspectos Básicos de Navigation:

- Comprender los fundamentos del uso de la biblioteca de Navigation de Jetpack con Jetpack Compose, incluyendo su configuración y características principales.

2. Navegación entre Composiciones:

- Cómo navegar entre diferentes funciones de componibilidad, permitiendo a los usuarios moverse entre distintas pantallas de la aplicación.

3. Integración de Barras de Pestañas:

- Implementar una barra de pestañas personalizada que admita composición dentro de la jerarquía de navegación, mejorando la organización y la usabilidad de la UI.

4. Navegación con Argumentos:


- Aprender a navegar utilizando argumentos, lo que permite pasar datos entre pantallas de manera efectiva y dinámica.

5. Navegación con Vínculos Directos:

- Cómo implementar la navegación con vínculos directos, facilitando la navegación entre diferentes secciones de la app.

6. Pruebas de Navegación:

- Métodos para probar la funcionalidad de navegación en la app, asegurando que todos los flujos de usuario se comporten como se espera.



Navegación

Codelab

Aprende a usar la biblioteca de Jetpack Navigation en Compose, a navegar dentro de tu aplicación, a navegar con argumentos, a admitir vínculos directos y a probar tu navegación.

Realizar codelab

X Navigation de Jetpack Compose

- 1 Introducción
- 2 Configuración
- 3 Descripción general de la app de Rally
- 4 Cómo migrar a la navegación de Compose
- 5 Cómo integrar RallyTabRow con la navegación
- 6 Cómo extraer de RallyDestinations elementos de pantalla componibles
- 7 Cómo navegar a SingleAccountScreen con argumentos
- 8 Cómo habilitar la compatibilidad con vínculos directos
- 9 Cómo extraer el NavHost en RallyNavHost
- 10 Cómo probar la navegación de Compose
- 11 Felicitaciones

Informar un error

11. Felicitaciones

¡Felicitaciones! Completaste este codelab con éxito. Puedes encontrar el [código de la solución aquí](#) y compararlo con el tuyo.

Agregaste la navegación de Jetpack Compose a la app de Rally y ahora conoces sus conceptos clave. Aprendiste a configurar un gráfico de navegación de destinos que admiten composición, definir tus rutas y acciones de navegación, pasar información adicional a las rutas mediante argumentos, configurar vínculos directos y probar tu navegación.

Si deseas obtener más información y temas, como la [integración de la barra de navegación inferior](#), la navegación en varios módulos y los [gráficos anidados](#), puedes consultar el repositorio [Now in Android GitHub](#) y ver el modo en que se implementaron allí.

¿Qué sigue?

Consulta estos materiales para continuar tu [ruta de aprendizaje de Jetpack Compose](#):

- [Codelab de pruebas de Compose](#)
- [Video de problemas comunes de rendimiento en Jetpack Compose](#)

Más

•

•

Doc

•

•

¡Obtuviste otra insignia En aprendizaje!

tu perfil obtiene el badge

10

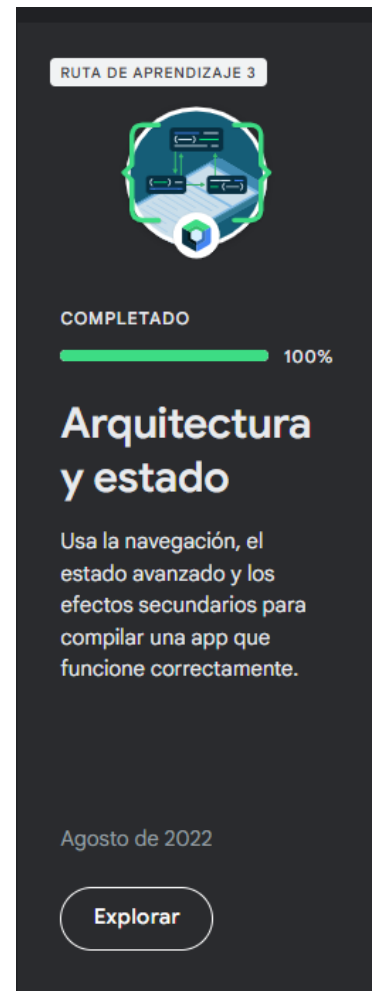
Volver a la ruta de aprendizaje

Ver perfil

Descartar

Listo

Cuestionario de Arquitectura y estado



Accesibilidad, pruebas y rendimiento

Codelab (Pruebas)

Este codelab está diseñado para enseñar a los desarrolladores cómo realizar pruebas efectivas de las interfaces de usuario (IU) que crean utilizando Jetpack Compose. A lo largo de este codelab, los participantes aprenderán diferentes enfoques y técnicas para garantizar que sus aplicaciones funcionen correctamente y ofrezcan una experiencia de usuario fluida.

Contenido del Codelab

1. **Introducción a las Pruebas de IU:**
 - Se explicará la importancia de realizar pruebas en las interfaces de usuario y cómo Jetpack Compose facilita este proceso.
2. **Pruebas Aisladas:**

- Los participantes aprenderán a escribir pruebas aisladas para sus componentes de IU, lo que les permitirá verificar el comportamiento de cada componente de manera individual.

3. Pruebas de Depuración:

- Se explorará cómo realizar pruebas de depuración para identificar y solucionar problemas en la IU, asegurando que los errores se detecten y se manejen adecuadamente.

4. Árboles Semánticos:

- Se discutirá el uso de árboles semánticos en las pruebas, que ayudan a entender cómo se organizan y estructuran los elementos de IU dentro de la aplicación.

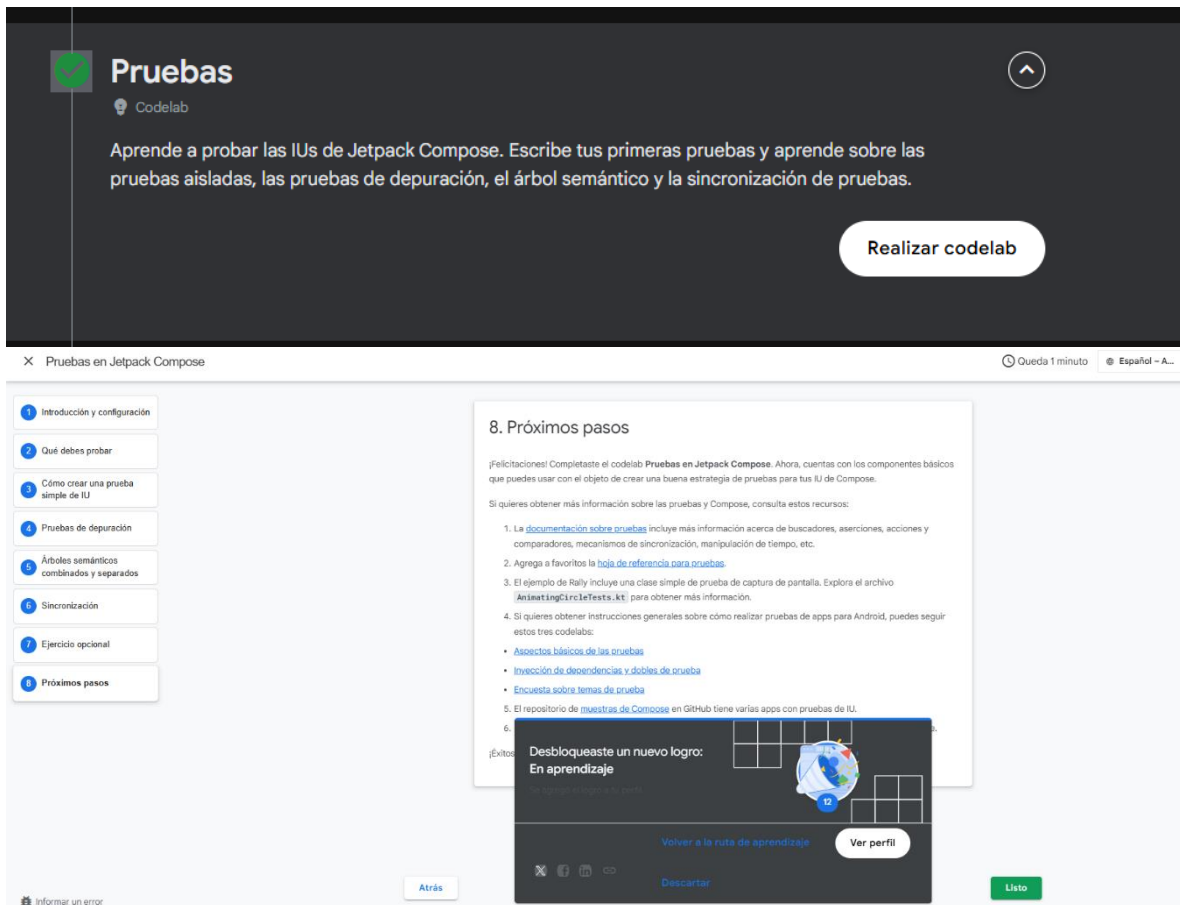
5. Sincronización:

- Los participantes aprenderán sobre la sincronización en pruebas, lo que es crucial para garantizar que las interacciones y actualizaciones en la IU se gestionen de manera fluida y sin problemas.

Objetivos de Aprendizaje

Al finalizar este codelab, los participantes deberían ser capaces de:

- Escribir y ejecutar pruebas de IU para sus aplicaciones de Jetpack Compose.
- Aplicar técnicas de pruebas aisladas y de depuración para mejorar la calidad de la aplicación.
- Comprender y utilizar árboles semánticos para estructurar las pruebas de manera efectiva.
- Manejar la sincronización en las pruebas para asegurar que las interacciones de la IU se comporten como se espera.



Codelab (Accesibilidad)

Este codelab está diseñado para enseñar a los desarrolladores cómo utilizar Jetpack Compose para hacer sus aplicaciones más accesibles, garantizando que todos los usuarios, incluidas aquellas personas con discapacidades, puedan utilizarlas de manera efectiva. A lo largo de este codelab, se mejorará una app de ejemplo mediante la implementación de diversas prácticas de accesibilidad.

Contenido del Codelab

1. Importancia de la Accesibilidad:

- Se discutirá la relevancia de desarrollar aplicaciones accesibles, teniendo en cuenta que muchos usuarios pueden tener condiciones como visión reducida, daltonismo, problemas de audición, trastornos de la motricidad, discapacidades cognitivas y otras afecciones.

2. Uso de TalkBack:

- Durante el codelab, se utilizará TalkBack, un servicio de accesibilidad para personas con discapacidad visual, para probar manualmente los cambios realizados en la app. También se fomentará la prueba de los cambios con otros servicios de accesibilidad, como la Accesibilidad con interruptores.

Objetivos de Aprendizaje

Al finalizar este codelab, los participantes aprenderán:

1. Tamaños de Objetivos Táctiles:

- Cómo aumentar los tamaños de los objetivos táctiles para satisfacer a los usuarios con trastornos de la motricidad, facilitando la interacción con la app.

2. Propiedades Semánticas:

- Qué son las propiedades semánticas en Jetpack Compose y cómo se pueden modificar para mejorar la accesibilidad de los elementos de IU.

3. Información de Elementos Componibles:

- Cómo proporcionar información a los elementos componibles para que sean más accesibles, lo que incluye el uso de descripciones de contenido y etiquetas de clics.

Accesibilidad

Codelab

Aprende a mejorar la usabilidad de tu app, en especial para los usuarios con discapacidad. Aumenta los tamaños de los objetivos táctiles, agrega descripciones de contenido, crea acciones personalizadas y mucho más.

Realizar codelab

X Accesibilidad en Jetpack Compose

Queda 1 minuto

Español - A...

1. Introducción

2. Cómo prepararte

3. Tamaño del objetivo táctil

4. Etiquetas de clics

5. Acciones personalizadas

6. Descripciones de elementos visuales

7. Encabezados

8. Combinaciones personalizadas

9. Interruptores y casillas de verificación

10. Descripciones de estados

11. ¡Felicitaciones!

11. ¡Felicitaciones!

¡Felicitaciones! Completaste correctamente este codelab y aprendiste más sobre la accesibilidad en Compose. Aprendiste sobre objetivos táctiles, descripciones visuales de elementos y descripciones de estados. Agregaste etiquetas de clics, encabezados y acciones personalizadas. Sabes cómo agregar una combinación personalizada y cómo trabajar con interruptores y casillas de verificación. Si aplicas estos aprendizajes en tus apps, la accesibilidad mejorará de manera notable.

Consulta los otros codelabs sobre la [ruta de Compose](#) y otras [muestras de código](#), como Jetnews.

Documentación

Para obtener más información y orientación sobre estos temas, consulta la siguiente documentación:

- [Accesibilidad en Jetpack Compose](#)
- [Semántica en Jetpack Compose](#)

¡Obtuviste otra insignia En aprendizaje!

Obtuviste otra insignia

63

Volver a la ruta de aprendizaje

Ver perfil

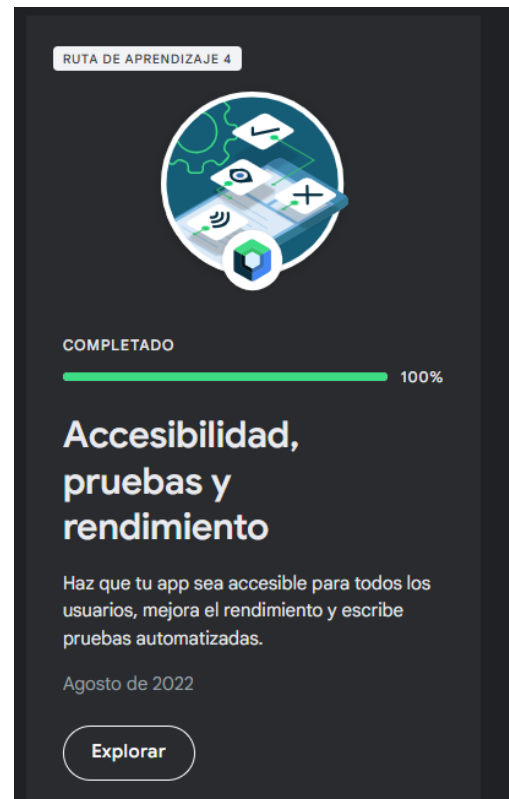
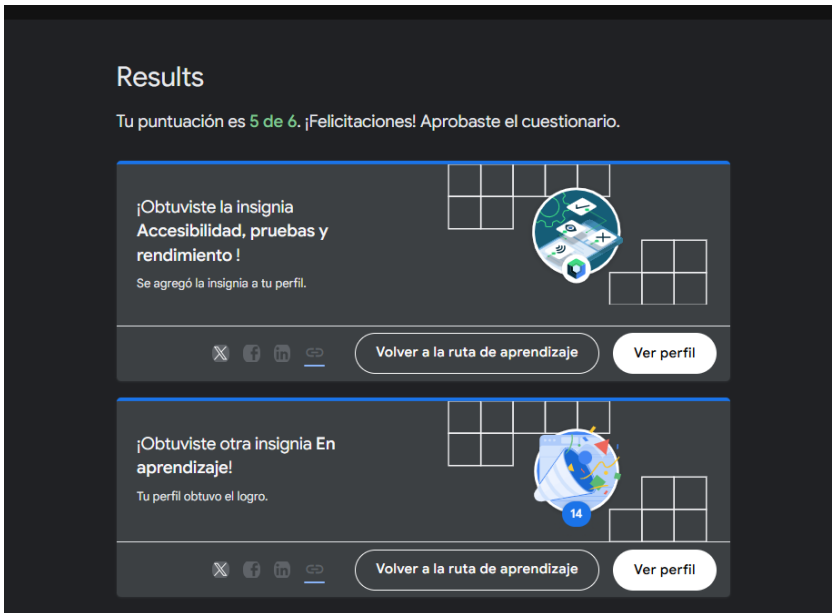
Descartar

Listo

Informar un error

Atrás

Cuestionario de Accesibilidad, pruebas y rendimiento



Factores de formas

Codelab (Compila apps adaptables)

Este codelab está diseñado para enseñar a los desarrolladores a compilar aplicaciones que sean adaptables a diferentes dispositivos, incluidos teléfonos, tablets y dispositivos plegables, mejorando así la accesibilidad a través de Jetpack Compose. Se explorarán las mejores prácticas para utilizar temas y componentes de Material 3, asegurando que las aplicaciones se adapten a una variedad de tamaños de pantalla y orientaciones.

Contenido del Codelab

1. Introducción a la Adaptabilidad:

- Se explicará el concepto de adaptabilidad, que se refiere a la capacidad de una IU para responder a diferentes tamaños de ventana, orientaciones y factores de forma. Se destacará la importancia de un diseño que se ajuste automáticamente al espacio disponible, desde ajustes simples hasta cambios de diseño completos.

2. Diseño Adaptable:

- Los participantes aprenderán a diseñar interfaces que se ajusten a los diversos tamaños de pantalla, incluyendo las diferencias entre teléfonos, tablets y dispositivos plegables.

3. Aplicación de Ejemplo: Reply:

- Se utilizará una aplicación de ejemplo llamada Reply para demostrar cómo implementar la adaptabilidad en Jetpack Compose, permitiendo a los desarrolladores ver ejemplos prácticos de las técnicas discutidas.

Objetivos de Aprendizaje

Al finalizar este codelab, los participantes aprenderán:

1. Orientación a Diferentes Tamaños de Ventana:

- Cómo diseñar la aplicación para que se oriente adecuadamente a todos los tamaños de ventana utilizando Jetpack Compose.

2. Segmentación para Dispositivos Plegables:


- Estrategias para segmentar la aplicación y optimizarla específicamente para dispositivos plegables, aprovechando su formato único.

3. Navegación y Accesibilidad:

- Cómo utilizar diferentes tipos de navegación en la aplicación para mejorar la accesibilidad y la experiencia del usuario.

4. Componentes de Material 3:

- Cómo aplicar los componentes de Material 3 para proporcionar la mejor experiencia de usuario, adaptando el diseño y la funcionalidad a cada tamaño de ventana.



Compila apps adaptables

Codelab

En este codelab, aprenderás a compilar apps adaptables para teléfonos, tablets y plegables, y también aprenderás sobre la accesibilidad. También conocerás las prácticas recomendadas para los temas y los componentes de Material 3.

Realizar codelab

1 Introducción

2 Prepárate

3 Cómo lograr que las apps sean adaptables

4 Navegación dinámica

5 Uso del espacio de la pantalla

6 Felicitaciones

6. Felicitaciones

¡Felicitaciones! Completaste correctamente este codelab y aprendiste a hacer apps adaptables con Jetpack Compose.

Aprendiste a verificar el tamaño y el estado de plegado de un dispositivo, y a actualizar la UI, la navegación y otras funciones de tu app adecuadamente. También aprendiste cómo la adaptabilidad mejora la accesibilidad y la experiencia del usuario.

Próximos pasos

Consulta los otros codelabs sobre la [ruta de aprendizaje de Compose](#).


Apps de ejemplo

- Las muestras de Compose son una colección de muchas apps que incorporan las prácticas recomendadas que se explican en codelabs.





Documentos de referencia

- [Diseño adaptable de Material 3](#)
- [Cómo diseñar para pantallas grandes](#)
- [Presentamos Material Theme Builder](#)

Desbloqueaste un nuevo logro:
En aprendizaje
Se agregó el logro a tu perfil.



Compartir



Volver a la ruta de aprendizaje

Ver perfil

Descartar

Lista

Informar un error

Atrás

Codelab (Wear OS con Compose)

Este codelab está diseñado para enseñar a los desarrolladores cómo aplicar su conocimiento previo de Jetpack Compose al desarrollo de aplicaciones para dispositivos wearables. Se explorarán las características y beneficios de Compose para Wear OS, facilitando la creación de interfaces de usuario atractivas y funcionales con un enfoque simplificado y menos código.

Contenido del Codelab

1. Introducción a Compose para Wear OS:

- Se explicará cómo Compose para Wear OS permite a los desarrolladores trasladar su experiencia con Jetpack Compose a dispositivos wearables, aprovechando la compatibilidad integrada con Material Design para simplificar el desarrollo de IU.

2. Uso de Horologist:

- Se presentará Horologist, un proyecto de código abierto que se basa en Jetpack Compose y ayuda a acelerar el desarrollo de aplicaciones para Wear OS. Los participantes utilizarán Horologist como parte del proceso de desarrollo.

Objetivos de Aprendizaje

Al finalizar este codelab, los participantes aprenderán:

1. Similitudes y Diferencias:

- Reconocer las similitudes y diferencias entre su experiencia anterior con Jetpack Compose y el desarrollo en Wear OS.

2. Elementos Componibles Simples:

- Cómo funcionan los elementos componibles simples en Wear OS y cómo se pueden implementar.

3. Elementos Componibles Específicos de Wear OS:

- Aprender a utilizar y crear elementos componibles que son específicos para la plataforma Wear OS, adaptándose a sus características únicas.

4. LazyColumn de Wear OS:

- Implementar ScalingLazyColumn, una versión de LazyColumn adaptada para Wear OS que permite un mejor manejo de listas en dispositivos con pantallas pequeñas.

5. Scaffold de Wear OS:

- Comprender y utilizar la versión de Scaffold diseñada para Wear OS, que proporciona una estructura básica para las interfaces de usuario.



Wear OS con Compose

Codelab



En este codelab, aprenderás a transferir tus conocimientos de Compose a wearables con el nuevo Compose para Wear OS. Al final, habrás creado elementos simples y avanzados que admiten composición en una app para reloj.

Realizar codelab

X Codelab de Compose para Wear OS

⌚ Queda 1 minuto

🌐 Español

- 1 Introducción
- 2 Cómo prepararte
- 3 Cómo consultar las dependencias
- 4 Cómo consultar MainActivity
- 5 Cómo agregar elementos componibles simples
- 6 Cómo agregar elementos componibles únicos de Wear
- 7 Cómo migrar a ScalingLazyColumn
- 8 Cómo agregar Scaffold
- 9 Felicitaciones

9. Felicitaciones

¡Felicitaciones! Ya aprendiste los conceptos básicos del uso de Compose en Wear OS.

Ahora puedes volver a aplicar todo tu conocimiento de Compose para crear apps atractivas de Wear OS.

¿Qué sigue?

Consulta los otros codelabs de Wear OS:

- [Cómo crear tu primera tarjeta de Wear OS](#)

Lecturas adicionales

- Entrada de blog [Compose para Wear OS ahora es 1.0](#)
- [Ejemplo simple](#) de Compose para Wear OS en GitHub
- [Colección de apps para la muñeca con Wear OS](#) y más guías

Comentarios

Nos encantaría conocer tu experiencia con Compose para Wear OS y sobre lo que puedes compilar. Únete al debate en el [foro de Stack Overflow](#).

(Diviértete)

¡Obtuviste otra insignia En aprendizaje!

¡Obtuviste otra insignia En aprendizaje!

[Volver a la ruta de aprendizaje](#)

[Ver perfil](#)

[Descartar](#)

🐛 [Informar un error](#)

[Atrás](#)

[Listo](#)


Cuestionario de Factores de forma

Results

Tu puntuación es **5 de 5**. ¡Felicitaciones! Aprobaste el cuestionario.

¡Obtuviste la insignia Factores de forma !


Se agregó la insignia a tu perfil.



[Volver a la ruta de aprendizaje](#) [Ver perfil](#)


¡Obtuviste otra insignia En aprendizaje!

Tu perfil obtuvo el logro.



[Volver a la ruta de aprendizaje](#) [Ver perfil](#)

RUTA DE APRENDIZAJE 5



1/6 ACTIVIDADES RESTANTES

83%

Factores de forma

Usa Compose a fin de desarrollar apps para varios formatos de pantalla y tipos de dispositivos.

Agosto de 2022

[Explorar](#)

Conclusión

Los cursos sobre Jetpack Compose ofrecen un recorrido integral por las herramientas y técnicas necesarias para desarrollar aplicaciones modernas y atractivas en plataformas Android, incluyendo Wear OS. A lo largo de los codelabs, los participantes adquieren habilidades clave en la creación de interfaces de usuario (IU) utilizando un enfoque declarativo, lo que simplifica el proceso de diseño y mejora la eficiencia del desarrollo.

Los cursos cubren desde los fundamentos de la UI con componentes básicos y modificadores, hasta conceptos más avanzados como el manejo del estado, la integración de ViewModels y la implementación de pruebas. La atención a la accesibilidad y la adaptabilidad asegura que las aplicaciones no solo sean visualmente atractivas, sino también funcionales y accesibles para todos los usuarios, independientemente de sus necesidades.

Además, la inclusión de herramientas como Horologist para Wear OS permite a los desarrolladores extender su conocimiento a dispositivos wearables, destacando la versatilidad y flexibilidad de Jetpack Compose en diferentes entornos. Los conceptos de Material Design 3 y la capacidad de implementar navegación efectiva refuerzan la creación de experiencias de usuario coherentes y de alta calidad.

En conjunto, estos cursos no solo capacitan a los desarrolladores en las últimas tendencias y tecnologías, sino que también fomentan un enfoque centrado en el usuario que es fundamental en el desarrollo de software moderno. La culminación de estos codelabs permite a los participantes sentirse seguros y preparados para enfrentar los desafíos del desarrollo de aplicaciones en el ecosistema Android.