

Descripcion General

Compilación y ejecución básicas.....	1
Cómo supervisar el proceso de compilación.....	2
Ver tareas de compilación.....	4
Funciones avanzadas de compilación y ejecución.....	4
Cómo implementar de forma incremental con Apply Changes.....	5
Cómo usar Apply Changes.....	5
Cómo habilitar la ejecución de resguardo para Apply Changes.....	6
Limitaciones de Apply Changes.....	6
Cambios de código que requieren reiniciar la app.....	6
Bibliotecas y complementos.....	7
Cómo configurar tu proyecto nuevo.....	7
Cómo habilitar Ediciones en vivo.....	10
Cómo realizar y revisar cambios.....	11
Solución de problemas.....	11
Limitaciones.....	12
Cómo cambiar la configuración de ejecución o depuración.....	12

Descripcion general: Cómo crear y ejecutar tu app

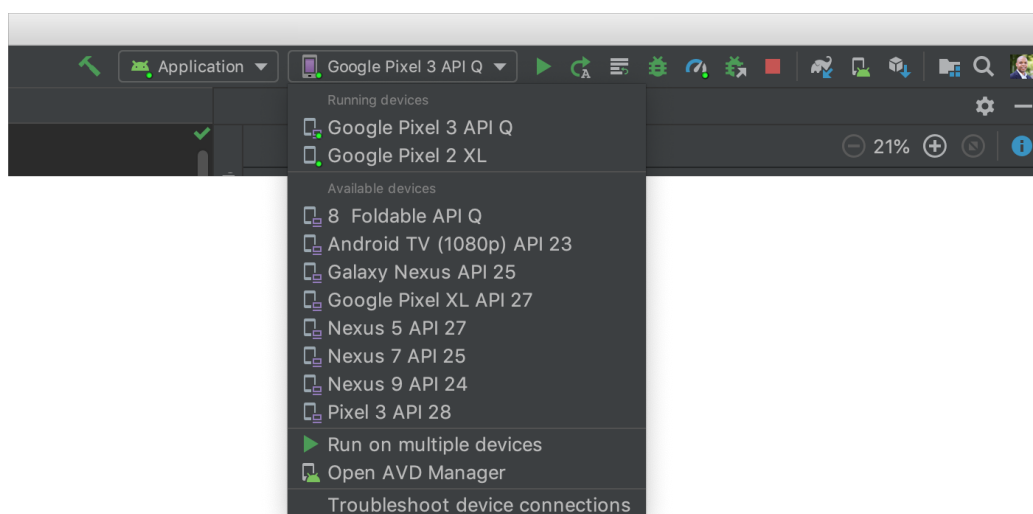
Cuando quieras ver cómo se ve y se comporta tu app en un dispositivo, debes compilarla y ejecutarla.

Android Studio configura proyectos para que puedas implementar tu app en un dispositivo virtual o físico con unos pocos clics.

Compilación y ejecución básicas

Para compilar y ejecutar la app, sigue estos pasos:

1. En la barra de herramientas, selecciona la app en el menú desplegable de configuraciones de ejecución.
2. En el menú desplegable del dispositivo de destino, selecciona el dispositivo en el que deseas ejecutar la app.



Si no tienes ningún dispositivo configurado, debes crear un dispositivo virtual de Android para usar Android Emulator o conectar un dispositivo físico.


3. Haz clic en **Run** .

Android Studio te advertirá si intentas iniciar el proyecto en un dispositivo que tiene un error o una advertencia asociados.

La iconografía y los cambios estilísticos diferencian entre:

- **errores** (selecciones de dispositivos que den como resultado una configuración dañada)
- **advertencias** (selecciones del dispositivo que pueden dar como resultado un comportamiento inesperado, pero aún se pueden ejecutar).

Cómo supervisar el proceso de compilación

Para ver los detalles del proceso de compilación, haz clic en **View > Tool Windows > Build** (o en **Build** , en la barra de ventanas de herramientas).

La ventana muestra las tareas que ejecuta Gradle para compilar la app, como se muestra en la Figura 1.

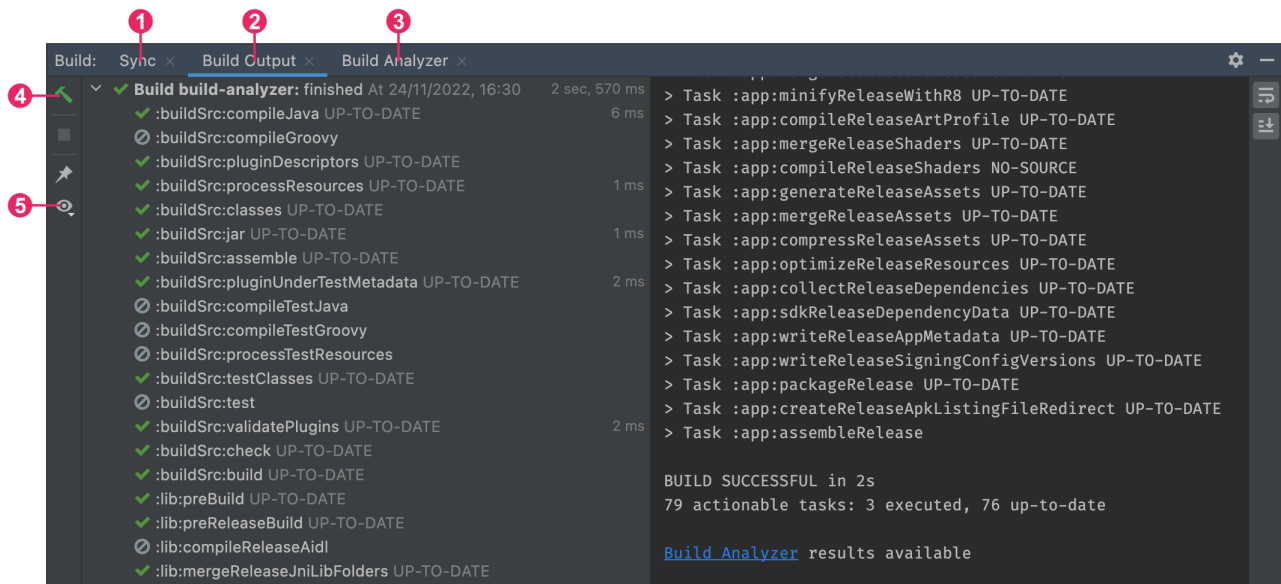


Figura 1: La ventana Build Output de Android Studio

1. **Pestaña Build:** Muestra las tareas que ejecuta Gradle en forma de árbol, donde cada nodo representa una fase de compilación o un grupo de dependencias de tareas.
 - Si recibes errores de compilación o de tiempo de compilación, inspecciona el árbol y selecciona un elemento para leer el resultado del error, como se muestra en la Figura 2.

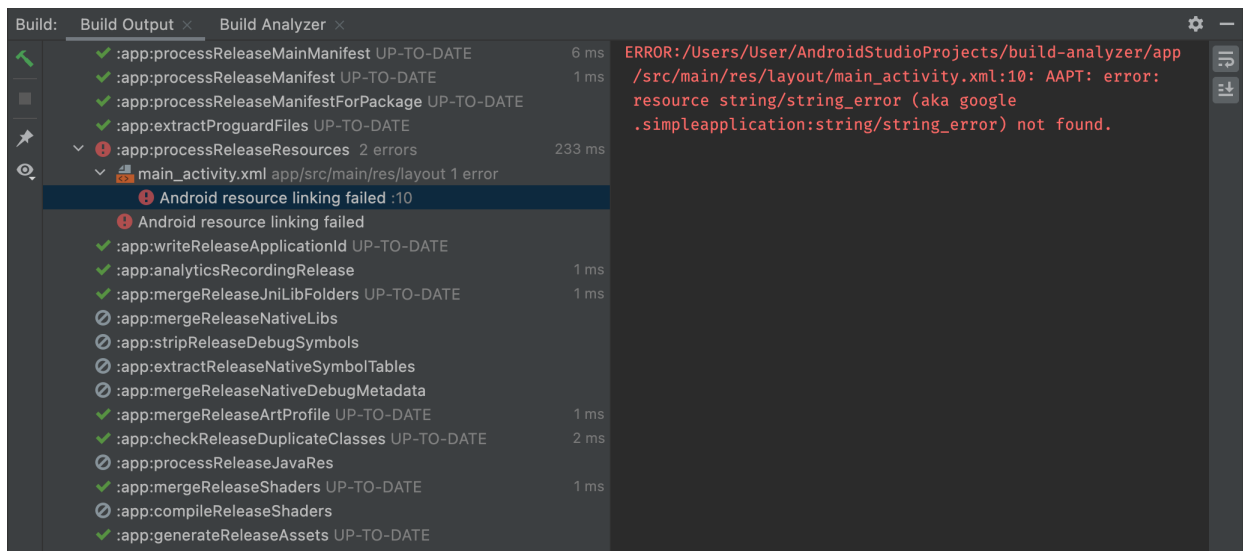




Figura 2: Inspecciona la ventana Build Output para ver los mensajes de error

2. **Pestaña Sync:** Muestra las tareas que ejecuta Gradle para sincronizarse con los archivos del proyecto. De manera similar a la pestaña **Build**
 - Si encuentras un error de sincronización, selecciona elementos del árbol para obtener más información al respecto.
3. **Restart:** Realiza la misma tarea que elegir **Build > Make Project** mediante la generación de archivos de compilación intermedia para todos los módulos del proyecto.
4. **Toggle view:** Activa o desactiva la visualización de la ejecución de tareas como un árbol gráfico, y muestra resultados de texto más detallados de Gradle.
 - Este es el mismo resultado que se muestra en la ventana **Gradle Console**  Gradle Console en Android Studio 3.0 y versiones posteriores.
 - Si las variantes de compilación usan variantes de productos, Gradle también invocará tareas para compilar esas variantes de producto.

Ver tareas de compilación

Para ver la lista de todas las tareas de compilación disponibles, haz clic en **View > Tool Windows > Gradle** (o en **Gradle** , en la barra de ventanas de herramientas).

Ver Errores

Si se produce un error durante el proceso de compilación, Gradle podría recomendar algunas opciones de la línea de comandos que te ayuden a resolver el problema, como:
--stacktrace o **--debug**.

Para usar opciones de la línea de comandos con el proceso de compilación, haz lo siguiente:


1. Abre el diálogo **Settings** o **Preferences**:
 - En Windows o Linux, selecciona **File > Settings** en la barra de menú.
2. Navega hasta **Build, Execution, Deployment > Compiler**.
3. En el campo de texto junto a *Command-line Options*, ingresa las opciones de la línea de comandos.
4. Haz clic en **OK** para guardar los cambios y salir.

Gradle aplicará estas opciones de la línea de comandos la próxima vez que intentes compilar la app.

Funciones avanzadas de compilación y ejecución

La forma predeterminada en que Android Studio implementa tu app, que se explica en la sección anterior, suele ser suficiente para probar una app simple.

Para casos de uso más avanzados, puedes activar o desactivar varios aspectos sobre cómo se compila y ejecuta tu app:

- Además, puedes implementar tu app en el modo de depuración. Para ello, haz clic en **Debug** . La ejecución de la app en el modo de depuración te permite configurar interrupciones en el código, examinar variables y evaluar expresiones durante el tiempo de ejecución, así como ejecutar herramientas de depuración.
- Si tienes una app más grande y compleja, **usar Apply Changes en lugar de hacer clic en Ejecutar puede ahorrarte tiempo**, ya que evitas reiniciar la app cada vez que desees implementar un cambio.
- Si usas Compose, Ediciones en vivo te permite actualizar elementos que admiten composición en tiempo real, sin volver a hacer clic en **Run**, de modo que puedas enfocarte en escribir código de IU con una interrupción mínima.
- Si tienes una app con múltiples variantes de compilación, o versiones, puedes **elegir qué variante de compilación implementar** mediante la ventana de herramientas de variantes de compilación.
- Para ajustar las opciones de instalación, inicio y prueba de la app, puedes **cambiar la configuración de ejecución y depuración**. La configuración de ejecución y depuración especifica si se debe implementar la app desde un APK o desde un Android App Bundle, y también especifica el módulo para ejecutar, el paquete para implementar, la actividad para iniciar, el dispositivo de destino, la configuración del emulador, las opciones de logcat y otros elementos.
- Deberías intentar usar Android Studio para tus necesidades de desarrollo, pero también puedes implementar tu app en un dispositivo virtual o físico desde la **línea de comandos**.

Cómo implementar de forma incremental con Apply Changes

En Android Studio 3.5 y versiones posteriores, Apply Changes te permite enviar cambios de código y recursos a la app en ejecución sin reiniciarla y, en algunos casos, sin reiniciar la actividad actual. Esta flexibilidad te permite controlar qué partes de tu app se reinician cuando quieres implementar y probar pequeños cambios incrementales y, al mismo tiempo, preservar el estado actual del dispositivo.

Apply Changes utiliza funciones de la implementación de JVMTI para Android que son compatibles con dispositivos que ejecutan Android 8.0 (nivel de API 26) o versiones posteriores.

Requisitos

La acción de Apply Changes solo está disponible si se cumplen las siguientes condiciones:

- Si compilas un APK de la app mediante una variante de compilación de depuración.
- Si implementas la app en un emulador o un dispositivo de destino que ejecuta Android 8.0 (nivel de API 26) o versiones posteriores.

Cómo usar Apply Changes

Utiliza las siguientes opciones cuando desees implementar los cambios en un dispositivo compatible:

Apply Changes and Restart Activity

- Reinicia la actividad para intentar aplicar los cambios de código y de recurso, pero sin reiniciar la app.
- Por lo general, puedes usar esta opción cuando modificas un código en el cuerpo de un método o cuando modificas un recurso existente.

Apply Code Changes


- Intenta aplicar solo los cambios de código sin reiniciar nada.
- Por lo general, puedes usar esta opción cuando modificas el código en el cuerpo de un método, pero no has modificado ningún recurso.
- Si modificaste el código y los recursos, entonces usa **Apply Changes and Restart Activity**.

Run

- Implementa todos los cambios y reinicia la app.
- Usa esta opción cuando los cambios que hayas realizado no se puedan aplicar mediante ninguna de las opciones de Apply Changes.

Cómo habilitar la ejecución de resguardo para Apply Changes

Después de hacer clic en **Apply Changes and Restart Activity** o **Apply Code Changes**, Android Studio compila un nuevo APK y determina si se pueden aplicar los cambios.

- Cuando no se pueden aplicar los cambios debido a que se produciría un error en Apply Changes, Android Studio te solicita que vuelvas a ejecutar  la app.
- Sin embargo, si no quieres recibir la solicitud cada vez que ocurra esto, puedes configurar Android Studio para que vuelva a ejecutar automáticamente la app cuando no se puedan aplicar los cambios.

Para habilitar este comportamiento, haz lo siguiente:

1. Abre el diálogo **Settings** o **Preferences**:
- En Windows o Linux, selecciona **File > Settings** en la barra de menú.
2. Ve a **Build, Execution, Deployment > Deployment**.
3. Selecciona las casillas de verificación para habilitar el resguardo de ejecución automático para cualquiera de las acciones de Apply Changes.
4. Haz clic en **OK**.

Nota: Algunos tipos de cambios no provocan la falla de Apply Changes, pero aun así requieren que reinicies tu app manualmente a fin de poder verlos. Por ejemplo, si realizas cambios en el método **onCreate()** de una actividad, solo surtirán efecto una vez que se vuelva a iniciar la actividad, por lo que debes reiniciar la app para verlos.

Cambios que dependen de la plataforma

Algunas funciones de Apply Changes dependen de versiones específicas de la plataforma de Android. Para aplicar estos tipos de cambios, tu app debe implementarse en un dispositivo que ejecute esa versión de Android (o una versión posterior).

Limitaciones de Apply Changes

Apply Changes está diseñado para acelerar el proceso de implementación de la app. Sin embargo, existen algunas limitaciones respecto de cuándo se puede usar.

Cambios de código que requieren reiniciar la app





No es posible aplicar algunos cambios de código y recursos hasta que se reinicie la app, incluidos los siguientes:

- Agregar o quitar un campo
- Quitar un método
- Cambiar las firmas de métodos
- Cambiar modificadores de métodos o clases
- Cambiar la herencia de clase
- Cambiar los valores de enumeraciones
- Agregar o quitar un recurso
- Cambiar el manifiesto de la app
- Cambiar las bibliotecas nativas (archivos .so)

Bibliotecas y complementos

Algunas bibliotecas y complementos realizan cambios automáticamente en los archivos de manifiesto de la app o en los recursos a los que se hace referencia en el manifiesto.

Estas actualizaciones automáticas pueden interferir en Apply Changes de las siguientes maneras:

- Si una biblioteca o un complemento realiza cambios en el manifiesto de la app, no podrás usar **Apply Code Changes**  ni **Apply Changes and Restart Activity** , y deberás reiniciar la app para poder ver los cambios.
- Si una biblioteca o un complemento realiza cambios en los archivos de recursos de la app, no podrás usar **Apply Code Changes**  y deberás usar **Apply Changes and Restart Activity**  para ver esos cambios.

A fin de evitar estas limitaciones, inhabilita todas las actualizaciones automáticas para las variantes de compilación de depuración.

Cómo configurar tu proyecto nuevo

1. Antes de comenzar, asegúrate de tener instalada la versión más reciente de Android Studio Electric Eel y de que el nivel de API de tu dispositivo físico o emulador sea al menos 30.
2. Abre Android Studio y selecciona **New Project** en la ventana emergente **Welcome to Android Studio**. Si ya tienes un proyecto abierto, puedes crear uno nuevo si vas a **File > New > New Project**.
3. Elige la plantilla **Empty Compose Activity** para **Phone and Tablet** y, luego, haz clic en **Next**.

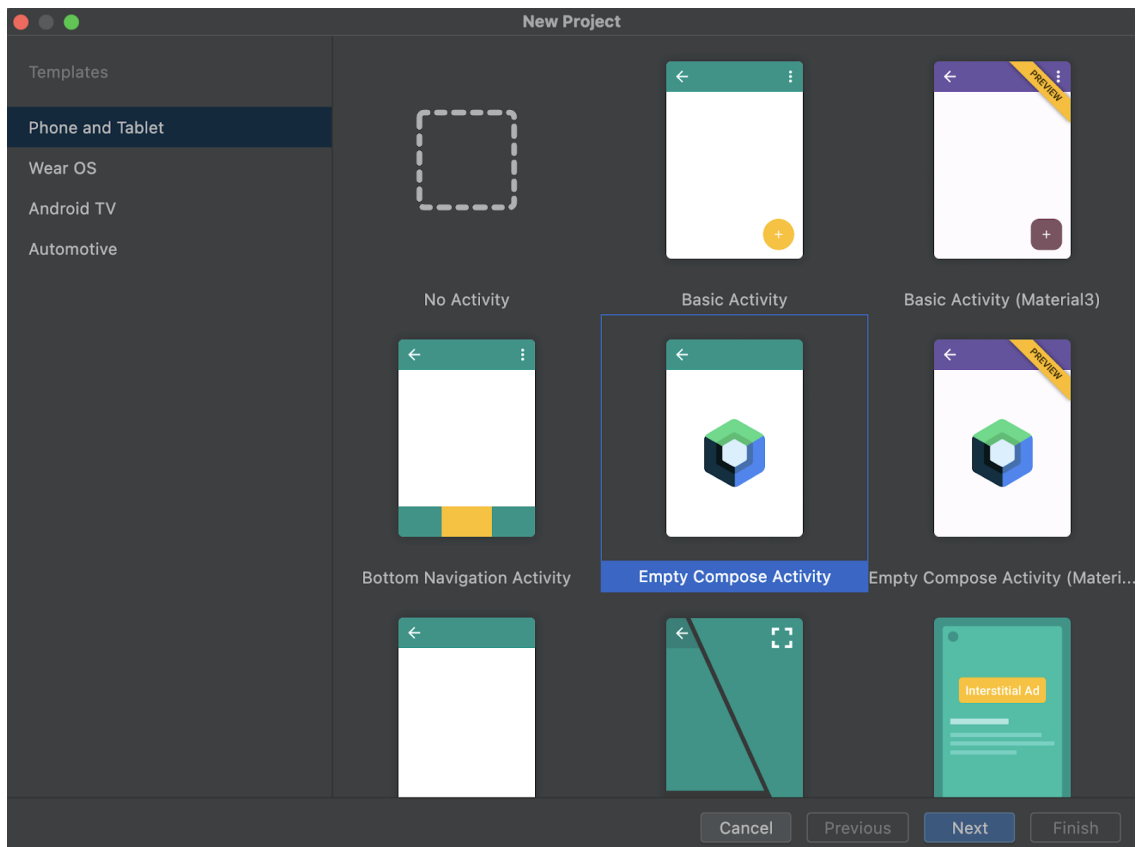


Figura 4: Las plantillas que puedes elegir En Ediciones en vivo, elige **Empty Compose Activity**.

4. Introduce lo siguiente y, luego, haz clic en **Finish**.

- **Name:** HelloWorld
- **Package name:** com.example.helloworld
- **Save location:** Predeterminado.
- **Language:** Java
- **Minimum SDK:** Predeterminado

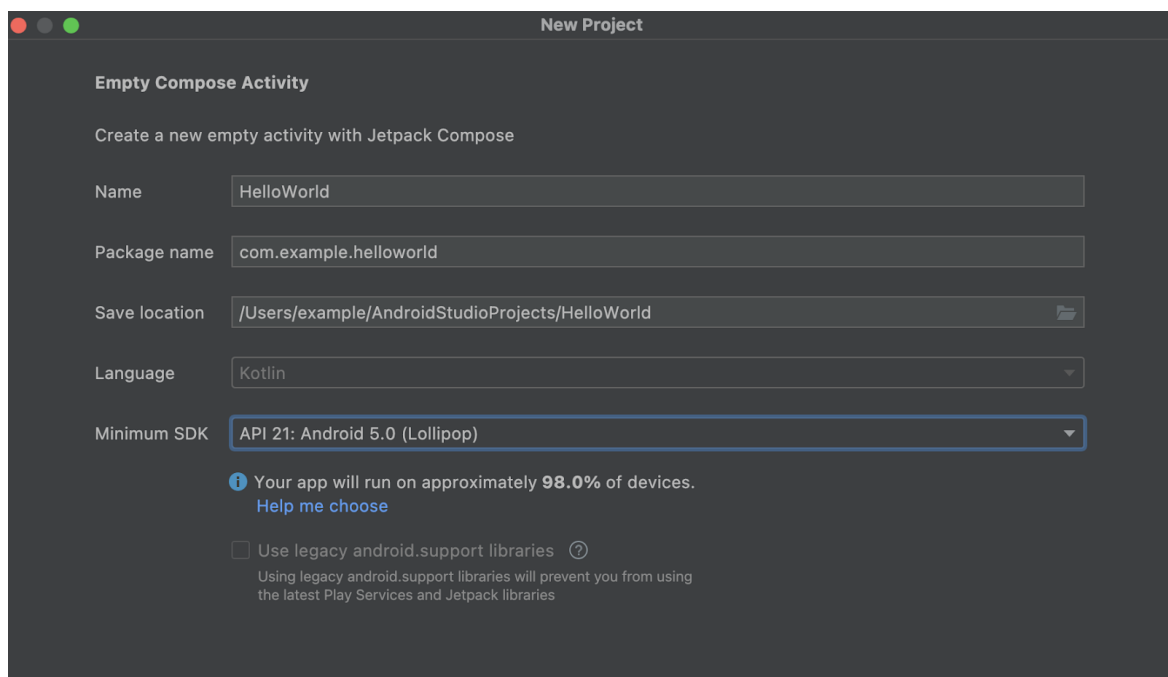


Figura 5: Ejemplo de la configuración del proyecto

Cómo habilitar Ediciones en vivo

1. En el IDE, navega a la configuración para habilitar Ediciones en vivo.
 - En Windows o Linux, ve a **File > Settings > Editor > Live Edit**.
 - En macOS, ve a **Android Studio > Preferences > Editor > Live Edit**.
2. Selecciona la opción **Live Edit** y el modo que quieres ejecutar desde la configuración. En el modo manual, los cambios de código se aplican cada vez que guardas manualmente con **Ctrl + S**
3. En el modo automático, los cambios de código se aplican en tu dispositivo o emulador a medida que los realizas.

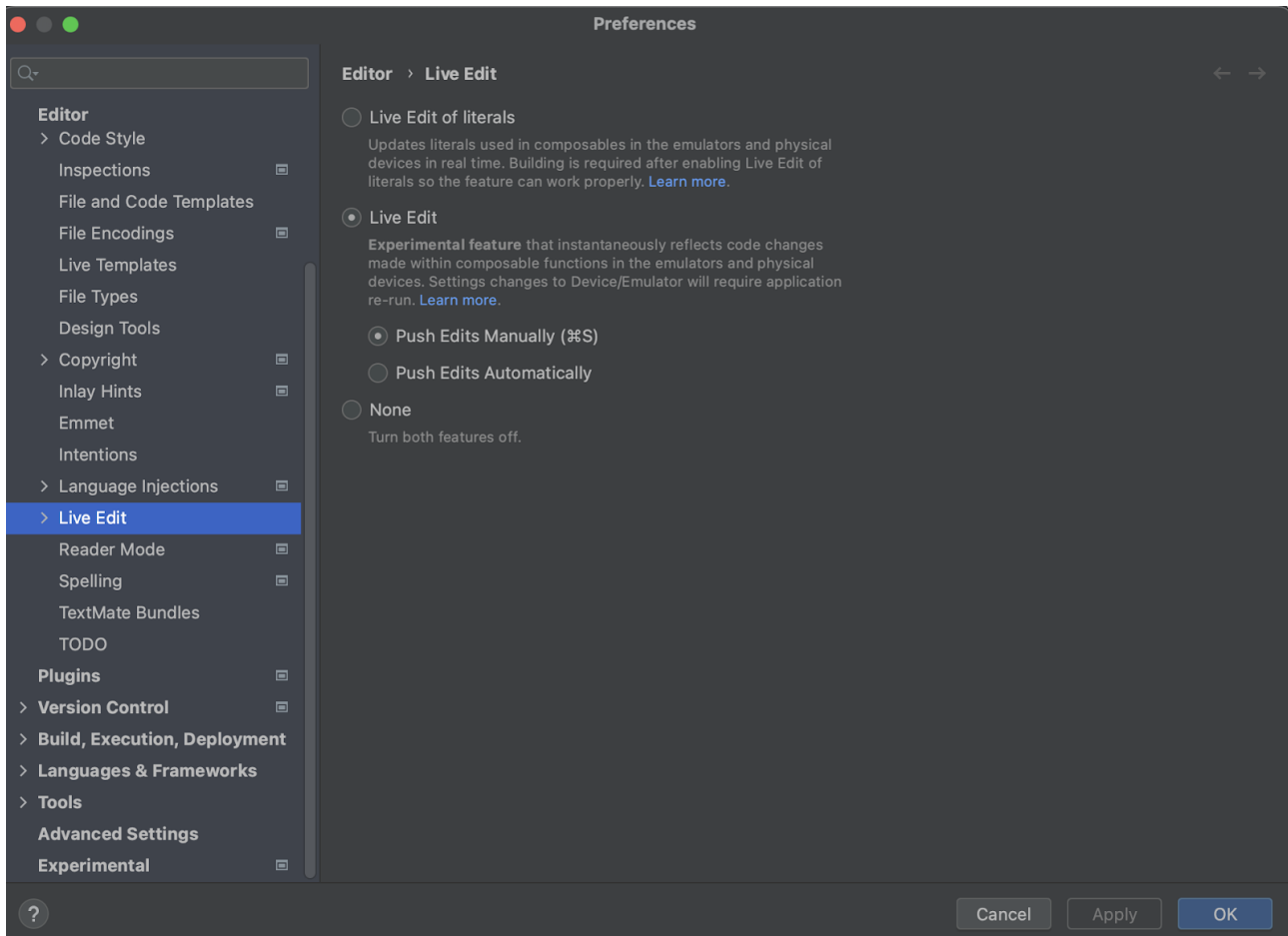

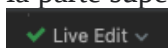


Figura 6: La opción **Live Edit** en la configuración

4. En el editor, abre el archivo **MainActivity**, que es el punto de entrada de tu app.
5. Haz clic en **Run**  para implementar tu app y, luego, en **Split** en la parte superior derecha del editor a fin de abrir la vista previa.
6. Después de activar Ediciones en vivo, deberías ver la marca de verificación verde de **Live Edit** en la parte superior derecha del editor.



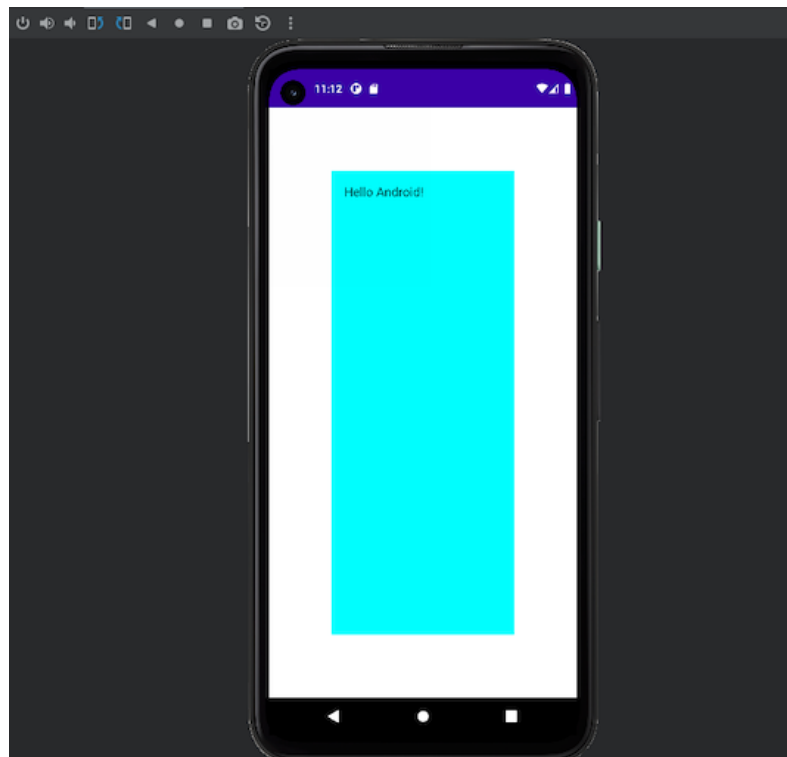
Cómo realizar y revisar cambios

En el editor, cambia el método **Greeting** actual en **MainActivity** por lo siguiente. Los cambios aparecerán al instante, como se muestra en la Figura 7.

@Composable

```
fun Greeting(name: String) {  
    Text(text = "Hello $name!",  
        Modifier.padding(80.dp) // Outer padding; outside background  
            .background(color = Color.Cyan) // Solid element background color  
            .padding(16.dp) // Inner padding; inside background, around text)  
    )  
}
```

Figura 7: Los cambios de Ediciones en vivo del método **Greeting** anterior aparecen al instante



Solución de problemas

Si no ves tus cambios en el panel de vista previa, es posible que Android Studio no las haya actualizado.

Comprueba si el indicador de la IU de **Ediciones en vivo** muestra un ícono de pausa, que sugiere un error de compilación.

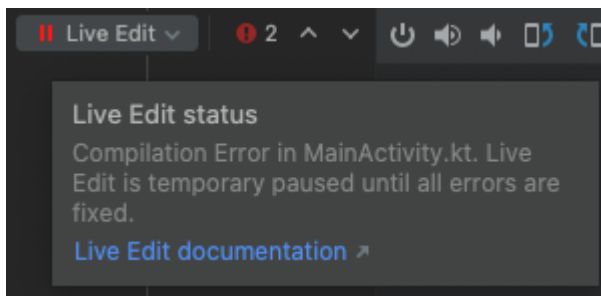


Figura 8: A fin de obtener más información sobre el error y las sugerencias para corregirlo, coloca el cursor sobre **Live Edit: ON** en la IU

Limitaciones

Nota: Ediciones en vivo es una función experimental en las compilaciones de Android Studio Flamingo, por lo que presenta algunos problemas y limitaciones.

Cómo cambiar la configuración de ejecución o depuración

Cuando ejecutas la app por primera vez, Android Studio usa una configuración de ejecución predeterminada.

La configuración de ejecución especifica si se debe implementar la app desde un APK o desde un Android App Bundle, y también especifica el módulo para ejecutar, el paquete para implementar, la actividad para iniciar, el dispositivo de destino, la configuración del emulador, las opciones de logcat y otros elementos.

La configuración de ejecución y depuración predeterminada compila un APK, lanza la actividad del proyecto predeterminado y usa el cuadro de diálogo **Select Deployment Target** para seleccionar dispositivos de destino.

Si la configuración predeterminada no se adapta al proyecto o al módulo, puedes personalizar la configuración de ejecución o depuración, o bien crear una nueva, en los niveles del proyecto, de la configuración predeterminada y del módulo.

Para editar una configuración de ejecución o depuración, selecciona **Run > Edit configurations**.

Para obtener más información, consulta [Cómo crear y editar configuraciones de ejecución y depuración](#).