



EBook Gratis

APRENDIZAJE android-studio

Free unaffiliated eBook created from
Stack Overflow contributors.

#android-
studio

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con Android Studio.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	3
Instalación o configuración.....	3
Windows.....	3
Mac.....	4
Linux.....	4
Notas.....	5
Problema con la descarga.....	5
Atajos útiles.....	6
Atajos de navegación.....	6
Atajos de navegación - Mac OS X.....	6
Programación de atajos.....	6
Vista previa de diferentes tamaños de pantalla (dispositivos) y orientaciones.....	7
1. Previsualizar diferentes dispositivos.....	7
2. Orientación de conmutación.....	8
Usa tus atajos de herramientas favoritos en Android Studio.....	10
Capítulo 2: Actualizaciones de Android Studio.....	11
Examples.....	11
Actualizando Android Studio en Ubuntu.....	11
Canales de actualización de Android Studio.....	11
Visión general.....	11
Seleccionando un canal de actualización.....	12
Capítulo 3: Consejos y trucos de Android Studio.....	13
Examples.....	13
Emulador para pruebas.....	13
Plantilla personalizada en vivo.....	13

Ejemplo:.....	13
Para lograr este propósito, debe agregar una plantilla personalizada en vivo:.....	14
Use estilos de código personalizados, comparta con otros miembros del equipo y formatee au.....	17
Capítulo 4: Optimización de Android Studio.....	20
Parámetros.....	20
Examples.....	20
Personaliza la opción VM.....	20
Windows:.....	20
Mac:.....	20
Linux:.....	20
Configuración predeterminada para Android Studio de 64 bits.....	20
Ajuste optimizado.....	21
Creditos.....	22

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [android-studio](#)

It is an unofficial and free android-studio ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official android-studio.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con Android Studio

Observaciones

Android Studio es el IDE oficial de Google para el desarrollo de aplicaciones para Android, basado en IntelliJ IDEA. [1]

Utiliza un sistema de compilación basado en Gradle a través del Android Plugin para Gradle. [2]

Versiones

Versión	Fecha de lanzamiento
0.1.x	2013-05-01
0.2.x	2013-07-01
0.3.2	2013-10-01
0.4.2	2014-01-01
0.4.6	2014-03-01
0.5.2	2014-05-01
0.8.0	2014-06-01
0.8.6	2014-08-01
0.8.14	2014-10-01
1.0	2014-12-01
1.0.1	2014-12-02
1.1.0	2015-02-01
1.2.0	2015-04-01
1.2.1	2015-05-01
1.2.2	2015-06-01
1.3.0	2015-07-01
1.3.1	2015-08-01
1.3.2	2015-08-02

Versión	Fecha de lanzamiento
1.4.0	2015-09-01
1.4.1	2015-10-01
1.5.0	2015-11-01
1.5.1	2015-12-01
2.0	2016-04-01
2.1.0	2016-04-02
2.1.1	2016-05-01
2.1.2	2016-06-01
2.1.3	2016-08-01
2.2.0	2016-09-01
2.3.0	2017-03-02
2.3.1	2017-04-02

Examples

Instalación o configuración

Windows

1. Verifique que tiene el JDK correcto. Puede verificarlo abriendo el símbolo del sistema (presione la tecla de Windows y escriba `cmd`). En el símbolo del sistema escriba `javac -version`, esto mostrará la versión actual de JDK instalada en su coincidencia o un error * si falta Java. Si el JDK no está disponible o la versión es inferior a 1.8, descargue el [Kit de desarrollo de Java SE 8](#).
2. Descarga la última [versión de Android Studio](#).
3. Inicie el archivo `.exe` descargado.
4. Sigue el asistente para instalar Android Studio.
5. Una vez completada la instalación, abra Android Studio desde el acceso directo que se realizó en el escritorio durante el proceso de instalación.
6. Cuando abre Android Studio por primera vez, puede pedirle que copie sus configuraciones

anteriores, pero como es la primera vez, simplemente puede elegir no copiar nada.

7. Luego, Android Studio pide descargar las API necesarias para desarrollar aplicaciones de Android. Descárgalos, después de descargar el IDE se abrirá y podrás escribir tu primera aplicación.

Mac

1. Verifique que tiene el JDK correcto. Puede verificarlo abriendo el terminal (presione comando + espacio y escriba el terminal). En la línea de comando escriba `javac -version`, esto mostrará la versión actual de JDK instalada en su coincidencia o un error * si falta Java. Si el JDK no está disponible o la versión es inferior a la 1.8, descargue el [Kit de desarrollo de Java SE 8](#). **
2. Descarga la última [versión de Android Studio](#) .
3. Inicie el archivo `.dmg` descargado.
4. Arrastre y suelte Android Studio en la carpeta Aplicaciones, luego inicie Android Studio.
5. Abre Android Studio.
6. Cuando abre Android Studio por primera vez, puede pedirle que copie sus configuraciones anteriores, pero como es la primera vez, simplemente puede elegir no copiar nada.
7. Luego, Android Studio pide descargar las API necesarias para desarrollar aplicaciones de Android. Descárgalos, después de descargar el IDE se abrirá y podrás escribir tu primera aplicación.

Linux

1. Verifique que tiene el JDK correcto. Puede verificarlo abriendo el terminal (presione `comando + espacio` y escriba el `terminal`). En la línea de comando escriba `javac -version` , esto mostrará la versión actual de JDK instalada en su coincidencia o un error * si falta Java. Si el JDK no está disponible o la versión es inferior a 1.8, descargue el [Kit de desarrollo de Java SE 8](#) . **
2. Descarga la última [versión de Android Studio](#) .
3. Descomprima / extraiga Android Studio en una carpeta específica.
4. Abre el terminal y sigue la ruta donde has extraído el Android Studio. (Luego, use el comando `cd` para ir dentro de la carpeta de Android Studio). Después de entrar necesitamos ir dentro de la carpeta `bin` nuevo, use el comando `cd bin` y ingrese.
5. Ahora necesitamos cambiar el mod de nuestro archivo requerido, es decir, `studio.sh`, para hacerlo ingrese el comando `sudo chmod 777 -R studio.sh` , presione enter y escriba su

contraseña (si corresponde) e ingrese. (También puede ver la lista de archivos presentes dentro de bin mediante el comando `ls`).

- Después de cambiar el mod, solo tenemos que ejecutar el archivo `.studio.sh` para hacerlo, ingrese el comando `./studio.sh`
- Cuando ejecute el comando anterior El lanzamiento de Android Studio por primera vez, puede pedirle que copie sus configuraciones anteriores, pero como es la primera vez, simplemente puede elegir no copiar nada.
- Luego, Android Studio pide descargar las API necesarias para desarrollar aplicaciones de Android. Descárgalos, después de descargar el IDE se abrirá y podrás escribir tu primera aplicación.

Puede encontrar "no se puede ejecutar la herramienta SDK de mksdcard" cuando está instalando Android Studio en el sistema operativo de Ubuntu de 64 bits porque el estudio requiere unos 32 binarios. Para superar este error, finalice y cierre Android Studio y vaya a la terminal y ejecute `sudo apt-get install lib32z1 lib32ncurses5 lib32bz2-1.0 lib32stdc++6` . Una vez completada la instalación para estos binarios, vuelva al paso 6 y reanude el proceso de instalación.

Notas

- Si ya instaló JDK y aún lo está consiguiendo, asegúrese de haber configurado `JAVA_HOME` en sus `System Variables` . Puede comprobar esta [respuesta](#) sobre cómo configurar uno.
- Sin embargo, hay problemas de estabilidad conocidos en Android Studio en Mac cuando se usa JDK 1.8. Hasta que se resuelvan estos problemas, puede mejorar la estabilidad al degradar su JDK a una versión anterior (pero no inferior a JDK 1.6).
- Mientras se completa la descarga de Android Studio, verifique qué versión de **JDK** tiene: abra una línea de comandos y escriba `javac -version` . Si el JDK *no* está disponible o la versión es inferior a *1.8* , descargue el [Kit de desarrollo de Java SE 8](#) .
- Si está ejecutando una versión de 64 bits de **Ubuntu** , necesita instalar algunas bibliotecas de 32 bits con los siguientes comandos:

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install lib32z1 lib32ncurses5 libbz2-1.0:i386 lib32stdc++6
```

- Si está ejecutando **Fedora de 64 bits**, el comando es:

```
sudo yum install zlib.i686 ncurses-libs.i686 bzip2-libs.i686
```

Problema con la descarga

- Si se enfrenta a algún problema, como el mensaje de que su sistema no es compatible con Android Studio (puede ocurrir cuando utiliza un navegador web diferente a Chrome), descargue `Android Studio` desde: <http://tools.android.com/>

¡Eso es!

Atajos útiles

Atajos de navegación

Go to class	Ctrl+N
Go to file	Ctrl + Shift + N
Navigate open tabs	ALT + Left-Arrow; ALT + Right-Arrow
Lookup recent files	CTRL + E
Go to line	CTRL + G
Navigate to last edit location	CTRL + SHIFT + BACKSPACE
Go to declaration	CTRL + B
Go to implementation	CTRL + ALT + B
Go to source	F4
Go to super Class	CTRL + U
Show Call hierarchy	Ctrl + Alt + H
Search in path/project	CTRL + SHIFT + F

Atajos de navegación - Mac OS X

Go to line number	CMD + L
-------------------	---------

Programación de atajos

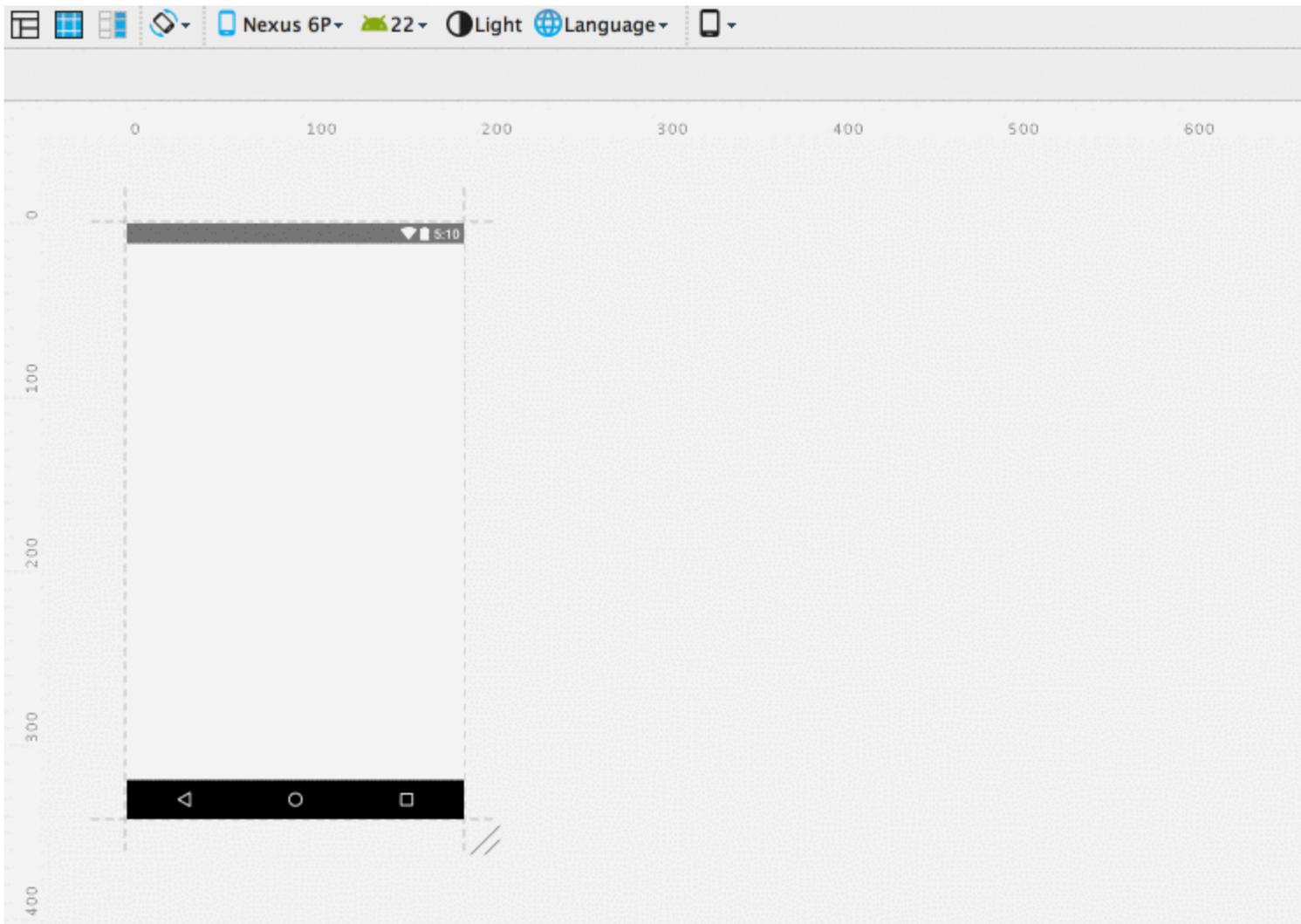
Reformat code	CTRL + ALT + L
Optimize imports	CTRL + ALT + O
Code Completion	CTRL + SPACE
Issue quick fix	ALT + ENTER
Surround code block	CTRL + ALT + T
Rename and refactor	Shift + F6
Line Comment or Uncomment	CTRL + /
Block Comment or Uncomment	CTRL + SHIFT + /
Go to previous/next method	ALT + UP/DOWN
Show parameters for method	CTRL + P
Quick documentation lookup	CTRL + Q
Project	Alt+1
Version Control	Alt+9
Run	Shift+F10
Debug	Shift+F9
Android Monitor	Alt+6
Return to Editor	Esc
Hide All Tool Windows	CTRL +Shift+F12
Auto generate code (constructor, getter/setter etc)	Alt+insert
Code completion	CTRL+Space
Smart code completion (by expected type)	CTRL+Shift+Space
Show quick fix	CTRL+Enter
Duplicate Line	Ctrl+D
Delete Line	Ctrl+Y

Vista previa de diferentes tamaños de pantalla (dispositivos) y orientaciones

2.1.3

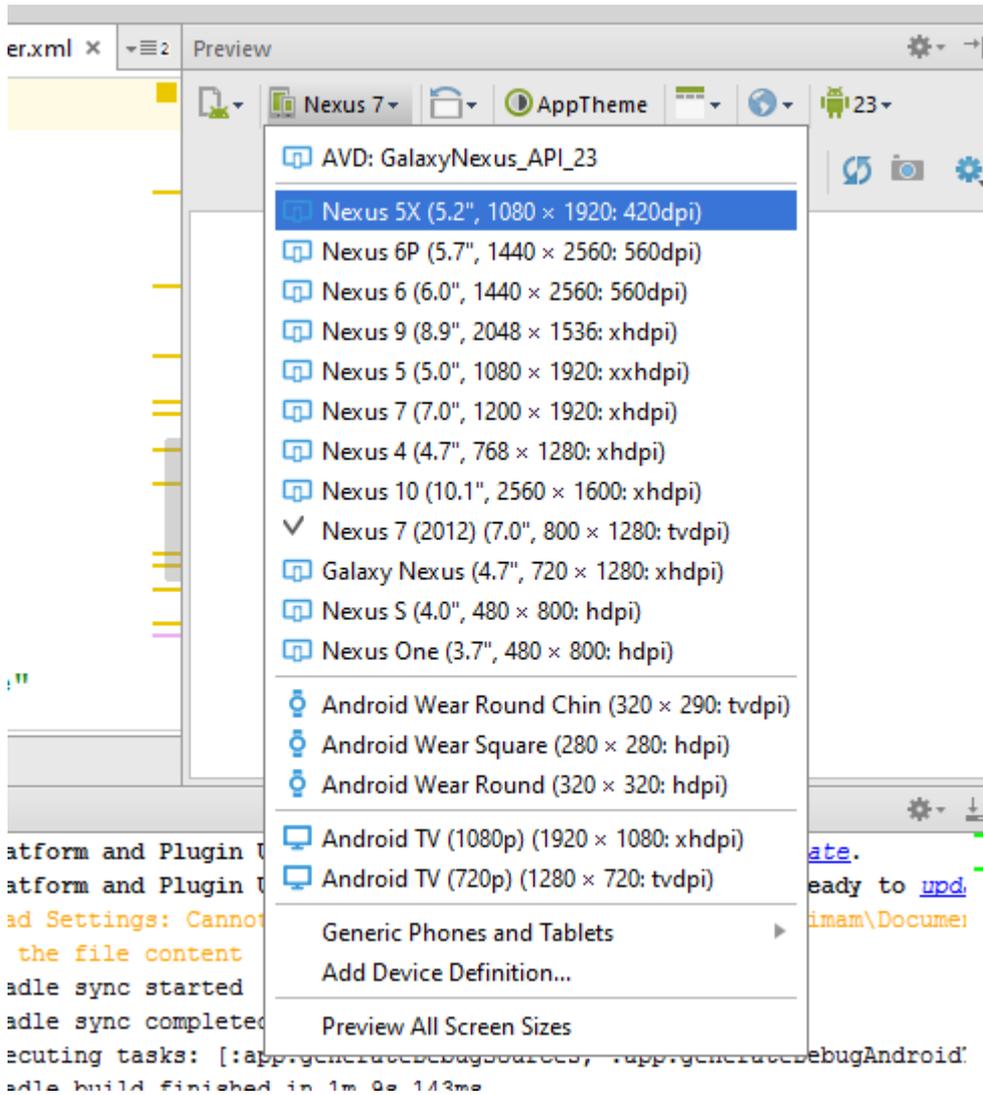
1. Previsualizar diferentes dispositivos

Hay un panel de vista previa a la derecha del estudio de Android. En este panel hay un botón con el nombre del dispositivo con el que está previsualizando la UI de su aplicación de esta manera.



Haga clic en el pequeño indicador desplegable de este y aparecerá un panel flotante con todos los dispositivos predefinidos. Puede seleccionar cualquiera de ellos para obtener una vista previa de la aplicación de la UI con diferentes dispositivos que mencionan sus tamaños de pantalla.

Mira esta imagen

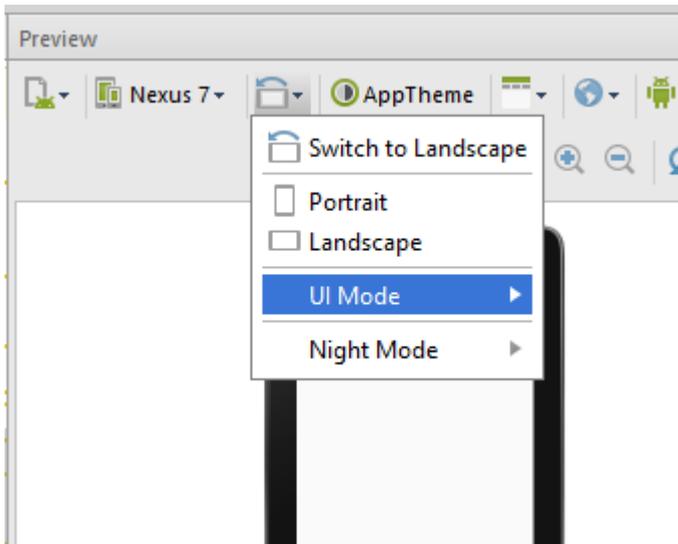


2. Orientación de conmutación

Y al lado de este botón hay otro botón como

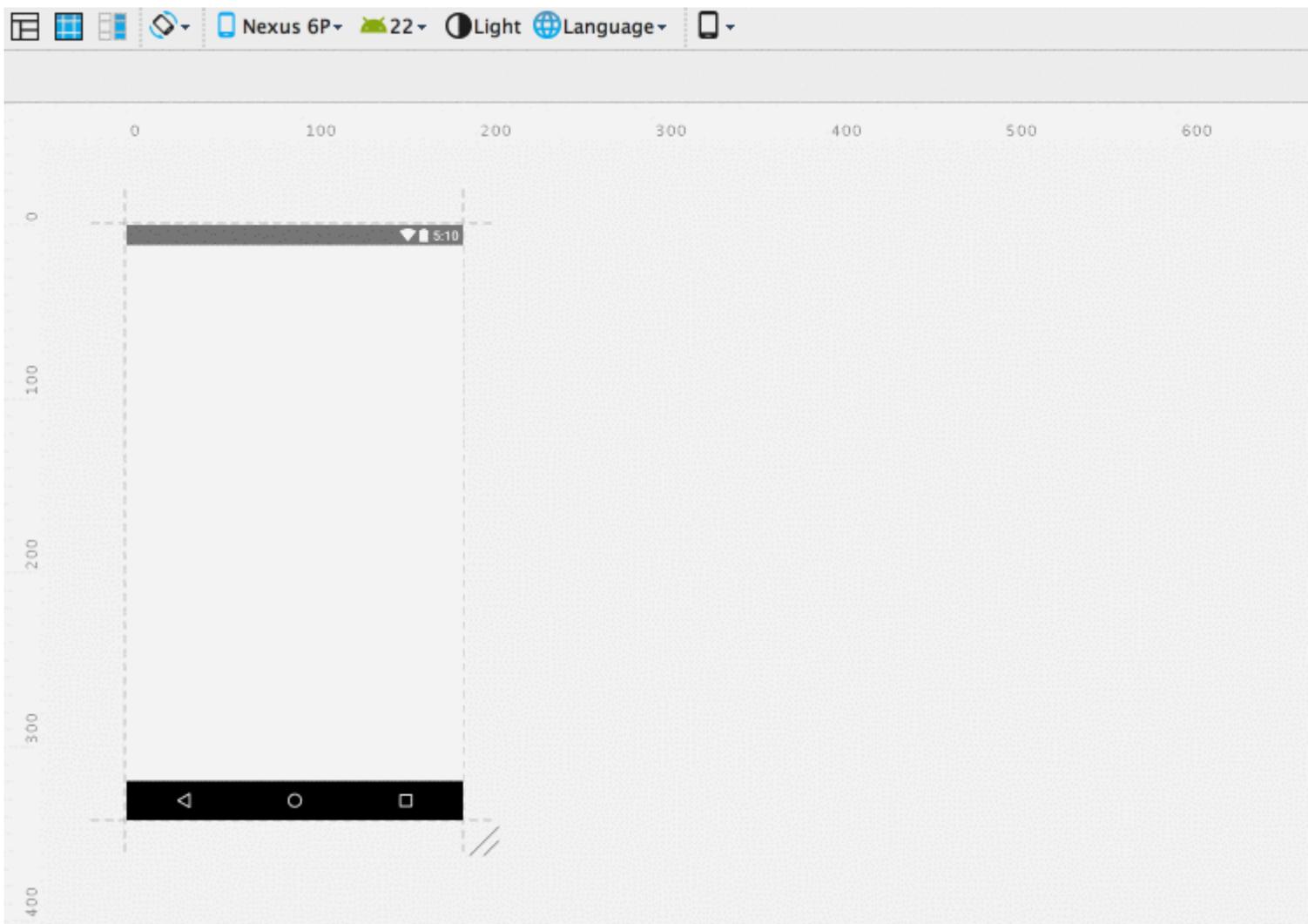


Al hacer clic en su menú desplegable, aparecerá un panel flotante con algunas opciones como `portrait` , `landscape` , etc. Seleccione una de ellas para previsualizar en diferentes orientaciones. Mira esta imagen



2.2.0

La opción de vista previa de todos los tamaños de pantalla no está disponible en Android Studio 2.2 Beta 2. Sin embargo, puede cambiar el tamaño de la pantalla de vista previa para ver cómo se vería su diseño en diferentes tamaños de pantalla, como se muestra en el adjunto.



Compruebe: [Android Studio 2.2 Vista previa de todos los tamaños de pantalla que faltan](#)

Usa tus atajos de herramientas favoritos en Android Studio

Vaya a Archivo> Configuraciones> Mapa de teclas y seleccione la opción de Mapas de teclas en:

- Mac OS X
- Emacs
- Estudio visual
- Eclipse
- Netbeans
- Jbuilder

y otros, para mapear los accesos directos a las herramientas buscadas.

Lea **Empezando con Android Studio en línea**: <https://riptutorial.com/es/android-studio/topic/830/empezando-con-android-studio>

Capítulo 2: Actualizaciones de Android Studio

Examples

Actualizando Android Studio en Ubuntu

Si puedes descargar una actualización de Android Studio, pero después de que se reinicia, no pasa nada, revisa el siguiente ejemplo:

1. Después de descargar el parche y cerrar Android Studio, abra el terminal.
2. Vaya a la carpeta de Android Studio, por ejemplo, `cd ~/android-studio`
3. Ir a la subcarpeta `bin`: `cd bin`
4. Asegúrese de que su archivo `studio.sh` tenga permisos de ejecución: `chmod +x studio.sh`
5. Ejecute Android Studio desde aquí: `./studio.sh`

Después de que Android Studio encuentre el parche y lo instale. Luego puedes cerrar Android Studio y ejecutarlo de la manera habitual (como prefieras, lo ejecuto desde el Launcher Pad).

Canales de actualización de Android Studio

Visión general

El mecanismo de actualización incorporado de Android Studio se puede configurar para recibir actualizaciones a través de cualquiera de estos **cuatro** canales:

- **Canario:** borde sangrante, publicado aproximadamente semanalmente. Estas son las primeras vistas preliminares publicadas para obtener retroalimentación del mundo real durante el desarrollo. El canal canario siempre tendrá la versión más reciente, incluidas las actualizaciones a versiones beta o estables. Recomendamos ejecutar compilaciones canarias en paralelo con una instalación beta o estable.
- **Dev:** Canary compila después de que una ronda completa de pruebas internas sean promovidas a Dev Channel.
- **Beta:** los candidatos de lanzamiento se basan en construcciones canarias estables, se publican y actualizan para obtener comentarios antes del lanzamiento estable. El canal beta se actualizará con nuevas compilaciones estables hasta que una nueva compilación de canarios se convierta en beta.
- **Estable:** la versión estable oficial, disponible en el sitio del desarrollador de Android.

Descargue las instalaciones completas de la compilación ofrecida en cada uno de estos canales:

[Canary](#) , [Dev](#) , [Beta](#) , [Stable](#) .

Alternativamente, puede compilarlo usted mismo, siguiendo las instrucciones en [Descripción general de la compilación](#) .

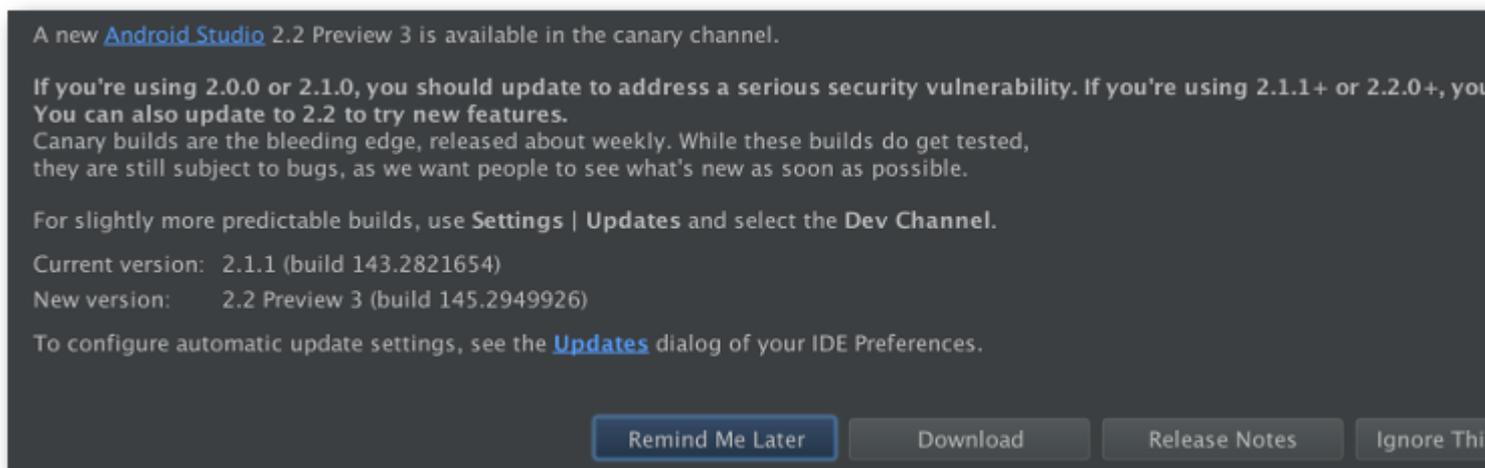
Seleccionando un canal de actualización

Para seleccionar el canal de actualización para una instalación de Android Studio, vaya a través de:

File > Settings > System Settings > Updates

y elija el canal apropiado en el que buscar actualizaciones:

Cuando hay una actualización disponible, el IDE te preguntará:



Al seleccionar actualizar, se mostrará el cuadro de diálogo de información actualizada con información detallada sobre el parche disponible, su tamaño y su canal:

Lea Actualizaciones de Android Studio en línea: <https://riptutorial.com/es/android-studio/topic/1998/actualizaciones-de-android-studio>

Capítulo 3: Consejos y trucos de Android Studio

Examples

Emulador para pruebas

Si no hay necesidades específicas, elija imágenes del sistema `x86_64` con API de Google para crear el emulador.

Funciona mucho más rápido que `armeabi-v7a` en computadoras basadas en Intel x86 x64.

Hay algunas bibliotecas de SDK compiladas y diseñadas con arquitectura ARM.

Si intenta instalarlos en emuladores basados en Intel, recibirá un mensaje de error que dice:

```
"INSTALL_FAILED_NO_MATCHING_ABIS"
```

En tales ocasiones, puede [optar](#) por un emulador de terceros como [Genymotion](#) o [el emulador de Android independiente de Visual Studio](#) .

Plantilla personalizada en vivo

Ejemplo:

```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        fastFinish();
    }

    public void fastFinish(){

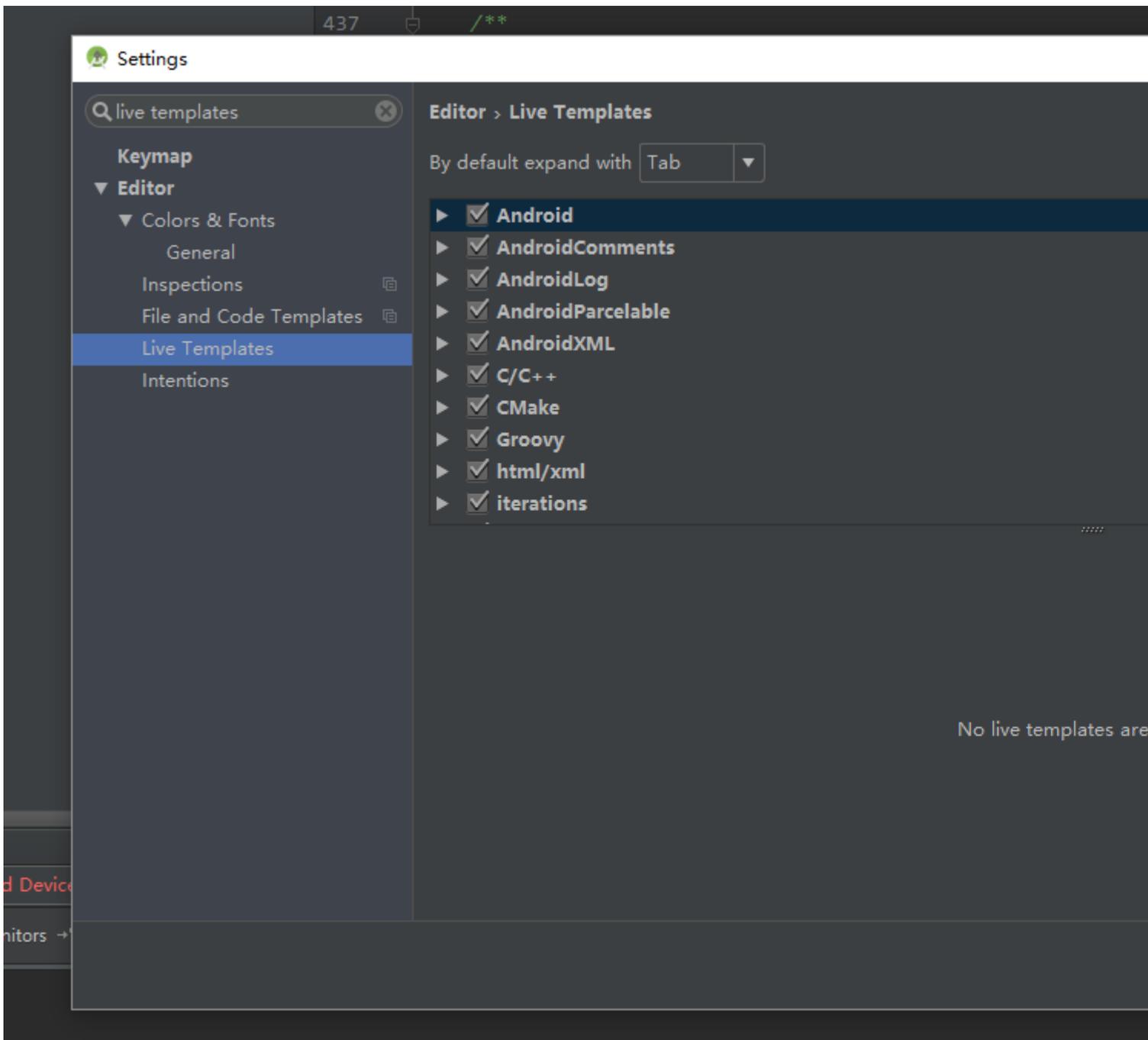
    }

}

```

Para lograr este propósito, debe agregar una plantilla personalizada en vivo:

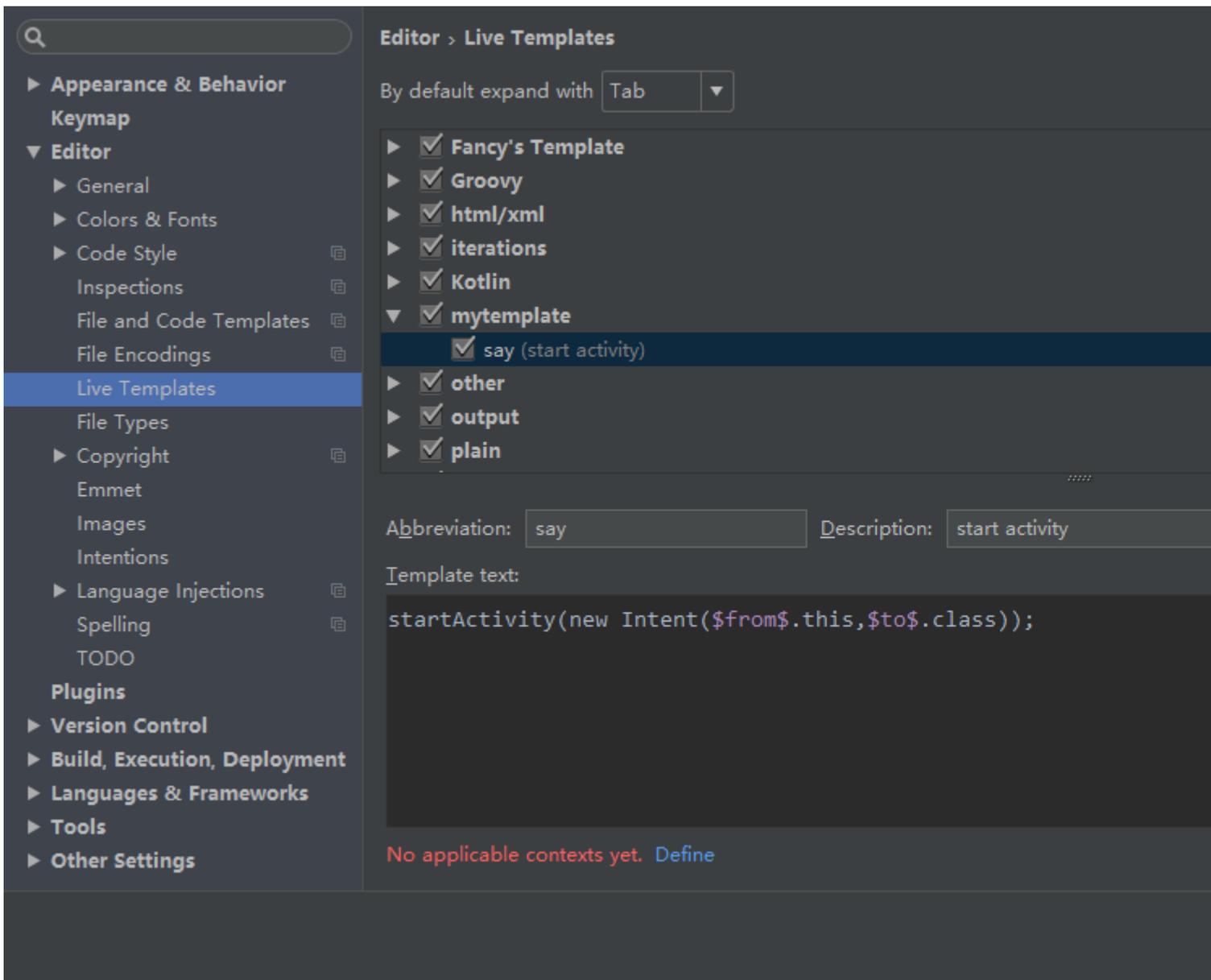
1. Configuración abierta [`Ctrl + Alt + S`]
2. Escriba "Plantillas en vivo" en la barra de búsqueda superior izquierda.
3. Haga clic en + para agregar un "Grupo de plantillas" y escriba un nombre de grupo (por ejemplo: Mi plantilla) para continuar



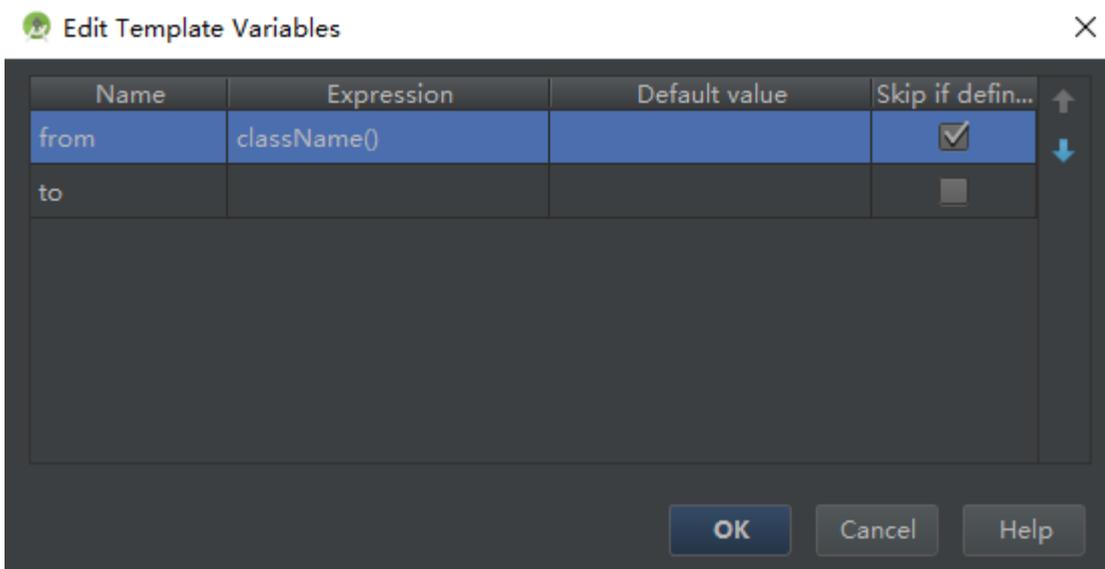
4. Elija su grupo de plantillas personalizadas (por ejemplo: Mi plantilla), luego haga clic en + y elija "Plantilla en vivo".
5. Tomaré "decir" por ejemplo:

```
say -> startActivity(new Intent($from$.this,$to$.class));
```

6. Escriba la clave que desea simplificar en "Abreviatura" (p. Ej .: say) y escriba la declaración en el "Texto de plantilla" (p. Ej .: startActivity(new Intent(\$from\$.this,\$to\$.class));)



7. Haga clic en "definir" en la parte inferior izquierda y elija la situación (por ejemplo, Java)
8. Haga clic en "Editar variables", defina la expresión. (Vea el detalle: [Cuadro de diálogo Editar variables de la plantilla](#))



9. Haga clic en "Aceptar" y "Aplicar". Y trata de escribir "say" en tu editor.

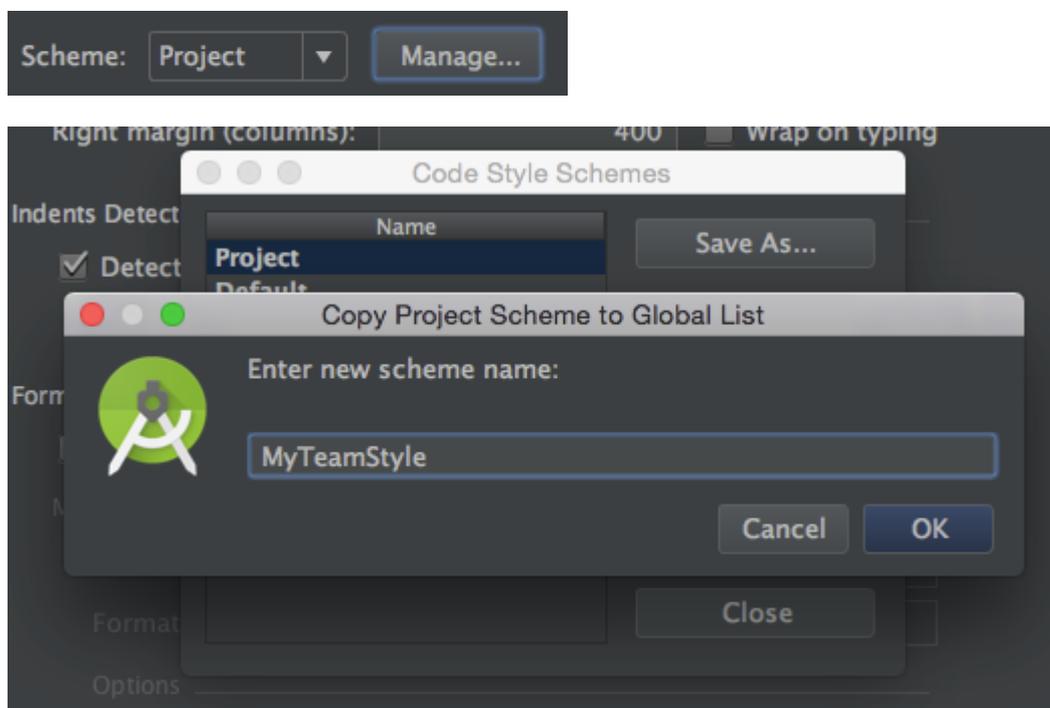
Use estilos de código personalizados, comparta con otros miembros del equipo y formatee automáticamente con acceso directo

Es posible tener sus propios estilos de código personalizados, compartirlos con otros miembros del equipo y usar un atajo para formatear automáticamente el código en un archivo.

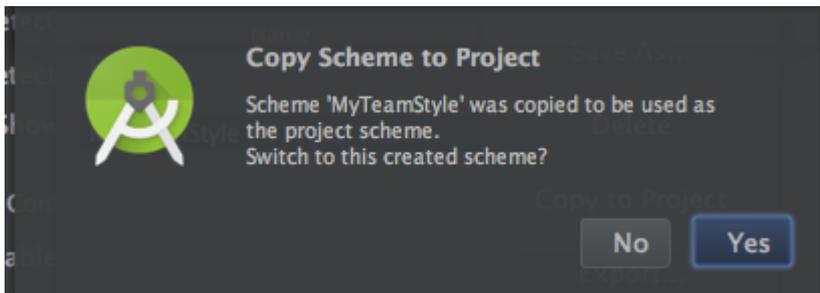
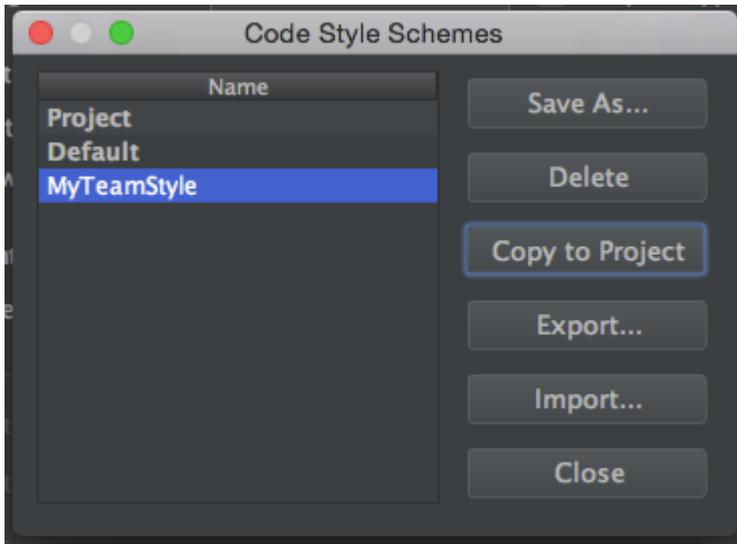
Para crear su propio estilo de código personalizado, vaya a: Preferencias -> Editor -> Estilo de código

Hay algunas configuraciones generales de estilo de código aquí. También puede seleccionar el idioma (Java para Android) y configurar los parámetros como mejor le parezca. Hay *muchos* ajustes.

Una vez que hayas hecho esto, guarda tu estilo de código. Esto es sólo para su custodia.



Una vez que lo haya guardado, seleccione "Copiar al proyecto" en el cuadro de diálogo "Administrar". Se le preguntará si desea cambiar a este esquema creado. Contesta que si



Una vez hecho esto, cierre el diálogo de preferencias de Estilo de código.

Ahora, verifique que sus configuraciones de estilo de código se hayan guardado en: `.idea / codeStyleSettings.xml`

Asegúrese de que este archivo no se ignore en su sistema de control de versiones para que pueda compartirlo con sus compañeros de equipo.

Una vez que sus compañeros de equipo tengan este archivo, también deben tener la misma configuración.

Ahora, al editar un archivo, puede formatear el código seleccionando:

Código -> Reformatear Código

Atajos para el código de reformateo (tomado de [esta](#) respuesta; vea la respuesta para obtener detalles sobre cómo resolver problemas con el atajo en Ubuntu):

Ganar

`Ctrl + Alt + L`

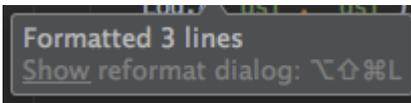
Linux:

`Ctrl + Shift + Alt + L`

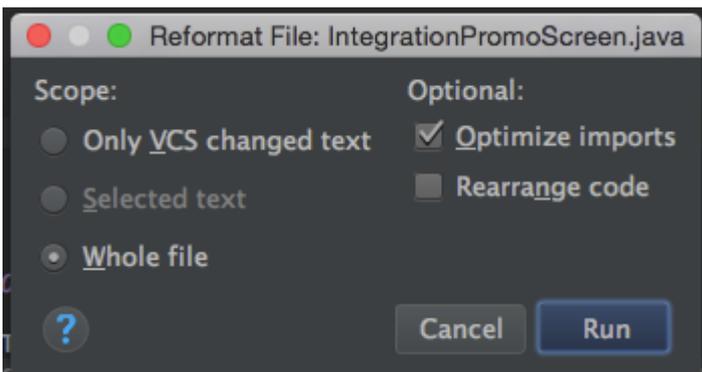
Mac:

Opción + Comando + L

Cuando realice un cambio de formato de código, aparecerá un pequeño cuadro de diálogo que le informará sobre los cambios realizados. Puede hacer clic en "Mostrar" en este cuadro de diálogo para abrir el "Cuadro de diálogo de archivo de reformateo".



También puede abrir este cuadro de diálogo desde el menú Código y su acceso directo correspondiente.



Tenga en cuenta que "Solo el texto cambiado de VCS" no siempre funciona según cómo se haya editado el código (puede ignorar una regla si parte del código no se ha editado).

También puede seleccionar texto y mostrar el cuadro de diálogo de cambio de formato para dar formato solo al texto seleccionado.

Lea Consejos y trucos de Android Studio en línea: <https://riptutorial.com/es/android-studio/topic/2228/consejos-y-trucos-de-android-studio>

Capítulo 4: Optimización de Android Studio

Parámetros

Parámetro	Detalle
Xms	Asignación de memoria inicial
Xmx	Asignación de memoria máxima

Examples

Personaliza la opción VM

Puede anular los `vmoptions` predeterminados con sus propias configuraciones personales seleccionando **Ayuda > Editar opciones de VM personalizadas** en la barra de herramientas de Android Studio. Esto creará una copia local que puede editar libremente.

Alternativamente, puede editar las `vmoptions` predeterminadas directamente usando las rutas que se indican a continuación. Tenga en cuenta que este método no es recomendable y que sus cambios pueden sobrescribirse al actualizar Android Studio.

Windows:

```
%USERPROFILE%\.{FOLDER_NAME}\studio.exe.vmoptions and/or  
%USERPROFILE%\.{FOLDER_NAME}\studio64.exe.vmoptions  
%USERPROFILE%\.{FOLDER_NAME}\idea.properties
```

Mac:

```
~/Library/Preferences/{FOLDER_NAME}/studio.vmoptions  
~/Library/Preferences/{FOLDER_NAME}/idea.properties
```

Linux:

```
~/.{FOLDER_NAME}/studio.vmoptions  
~/.{FOLDER_NAME}/studio64.vmoptions  
~/.{FOLDER_NAME}/idea.properties
```

Configuración predeterminada para Android Studio de 64 bits

```
-Xms128m
-Xmx750m
-XX:MaxPermSize=350m
-XX:ReservedCodeCacheSize=96m
-ea
-Dsun.io.useCanonCaches=false
-Djava.net.preferIPv4Stack=true
-Djna.nosys=true
-Djna.boot.library.path=

-Djna.debug_load=true
-Djna.debug_load.jna=true
-Djsse.enableSNIExtension=false
-XX:+UseCodeCacheFlushing
-XX:+UseConcMarkSweepGC
-XX:SoftRefLRUPolicyMSPerMB=50
-Didea.platform.prefix=AndroidStudio
-Didea.paths.selector=AndroidStudio
```

Ajuste optimizado

```
-Xms1024m
-Xmx4096m
-XX:MaxPermSize=1024m
-XX:ReservedCodeCacheSize=256m
-ea
-Dsun.io.useCanonCaches=false
-Djava.net.preferIPv4Stack=true
-Djna.nosys=true
-Djna.boot.library.path=

-Djna.debug_load=true
-Djna.debug_load.jna=true
-Djsse.enableSNIExtension=false
-XX:+UseCodeCacheFlushing
-XX:+UseConcMarkSweepGC
-XX:SoftRefLRUPolicyMSPerMB=50
-Didea.platform.prefix=AndroidStudio
-Didea.paths.selector=AndroidStudio
```

Lea Optimización de Android Studio en línea: <https://riptutorial.com/es/android-studio/topic/3247/optimizacion-de-android-studio>

Creditos

S. No	Capítulos	Contributors
1	Empezando con Android Studio	Ajay , Anurag , ARUN.G , Atif Farrukh , CJBS , Community , Daniel Nugent , H. Pauwelyn , Inzimam Tariq IT , kalan , murf , Nissim R , Paul Thompson , petrumo , piotrek1543 , pRaNaY , Rob R. , SaravInfern , Shashanth , sudo , Valentin Kuhn
2	Actualizaciones de Android Studio	piotrek1543 , Yury Fedorov
3	Consejos y trucos de Android Studio	Damindu , hellyale , meunicorn , piotrek1543 , Shashanth , the_new_mr
4	Optimización de Android Studio	Kevin Cooper , manfcas , piotrek1543 , UmAnusorn , Yury Fedorov