

Contenido



Estructura del proyecto

Interfaz de usuario

- Ventanas de herramientas
- Cómo completar el código
- Búsqueda de código de muestra
- Navegación
- Estilo y formato
- Aspectos básicos del control de versiones

Sistema de compilación de Gradle

- Variantes de compilación
- Compatibilidad con varios APK
- Reducción de recursos
- Administración de dependencias

Herramientas de depuración y perfil

Depuración integrada

- Generador de perfiles de rendimiento
- Volcado de montón
- Perfil de memoria
- Acceso al archivo de datos
- Inspecciones de código
- Anotaciones en Android Studio
- Mensajes de registro
- Generación de perfiles de rendimiento

Cómo acceder a tu cuenta de acceso

Introducción a Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android, basado en [IntelliJ IDEA](#) . Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas aplicaciones para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle
- Un emulador rápido y cargado de funciones
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android
- Aplicación de cambios para insertar cambios de códigos y recursos a la aplicación en ejecución sin reiniciar la aplicación
- Integración con GitHub y plantillas de código para ayudar a compilar funciones de aplicaciones comunes y también importar código de muestra
- Variedad de marcos de trabajo y herramientas de prueba
- Herramientas de pelusa para identificar problemas de rendimiento, usabilidad y compatibilidad de la versión, entre otros
- Compatibilidad con C ++ y NDK
- Compatibilidad integrada con [Google Cloud Platform](#) , que facilita la integración con Google Cloud Messaging y App Engine

En esta página, encontrará una introducción a las funciones básicas de Android Studio. Para acceder a un resumen de los cambios más recientes, consulte las [notas de la versión de Android Studio](#) .

Estructura del proyecto

Cada proyecto de Android Studio incluye uno o más módulos con archivos de código fuente y archivos de recursos. Entre los tipos de módulos se incluyen los siguientes:

- Módulos de aplicaciones para Android
- Módulos de biblioteca
- Módulos de Google App Engine

De manera predeterminada, Android Studio muestra los archivos de su proyecto en la vista de proyecto de Android, como se ve en la Figura 1. Esta vista está organizada en módulos para que pueda acceder rápidamente a los archivos fuente clave de su proyecto.

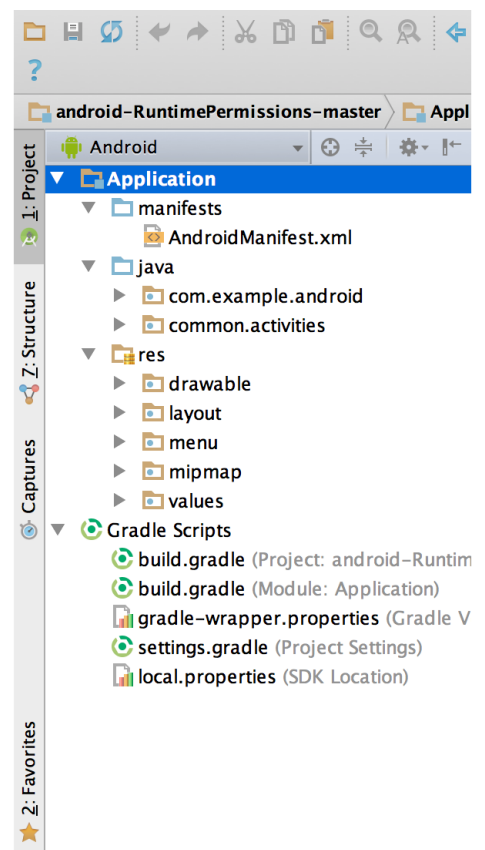
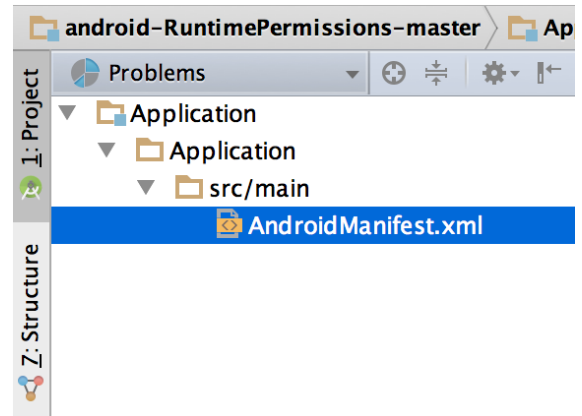


Figura 1: Los archivos de proyecto en la vista de Android

Puedes ver todos los archivos de compilación en el nivel superior de **Secuencias de comandos de Gradle** y cada módulo de aplicación contiene las siguientes carpetas:

- **manifiestos** : Contiene el archivo `AndroidManifest.xml`.
- **java** : Contiene los archivos de código fuente Java, incluido el código de prueba de JUnit.
- **res** : Contiene todos los recursos sin código, como diseños XML, cadenas de IU e imágenes de mapa de bits.

La estructura del proyecto de Android en el disco difiere de esta representación plana. Para ver la estructura de archivos reales del proyecto, selecciona **Proyecto** en el menú desplegable **Proyecto** (en la figura 1, se muestra como **Android**).



También puedes personalizar la vista de los archivos del proyecto para concentrarte en aspectos específicos del desarrollo de tu aplicación. Por ejemplo, si seleccionas la vista **Problemas** de tu proyecto, muestra enlaces a los archivos que contienen errores conocidos de codificación y sintaxis, como una etiqueta de cierre que falla para un elemento XML en un archivo de diseño.

Figura 2: Archivos del proyecto en la vista Problems, en la que se muestra un archivo de diseño con un problema

Para obtener más información, consulta [Descripción general de proyectos](#) .

Interfaz de usuario

La ventana principal de Android Studio consta de varias áreas lógicas identificadas en la Figura 3.

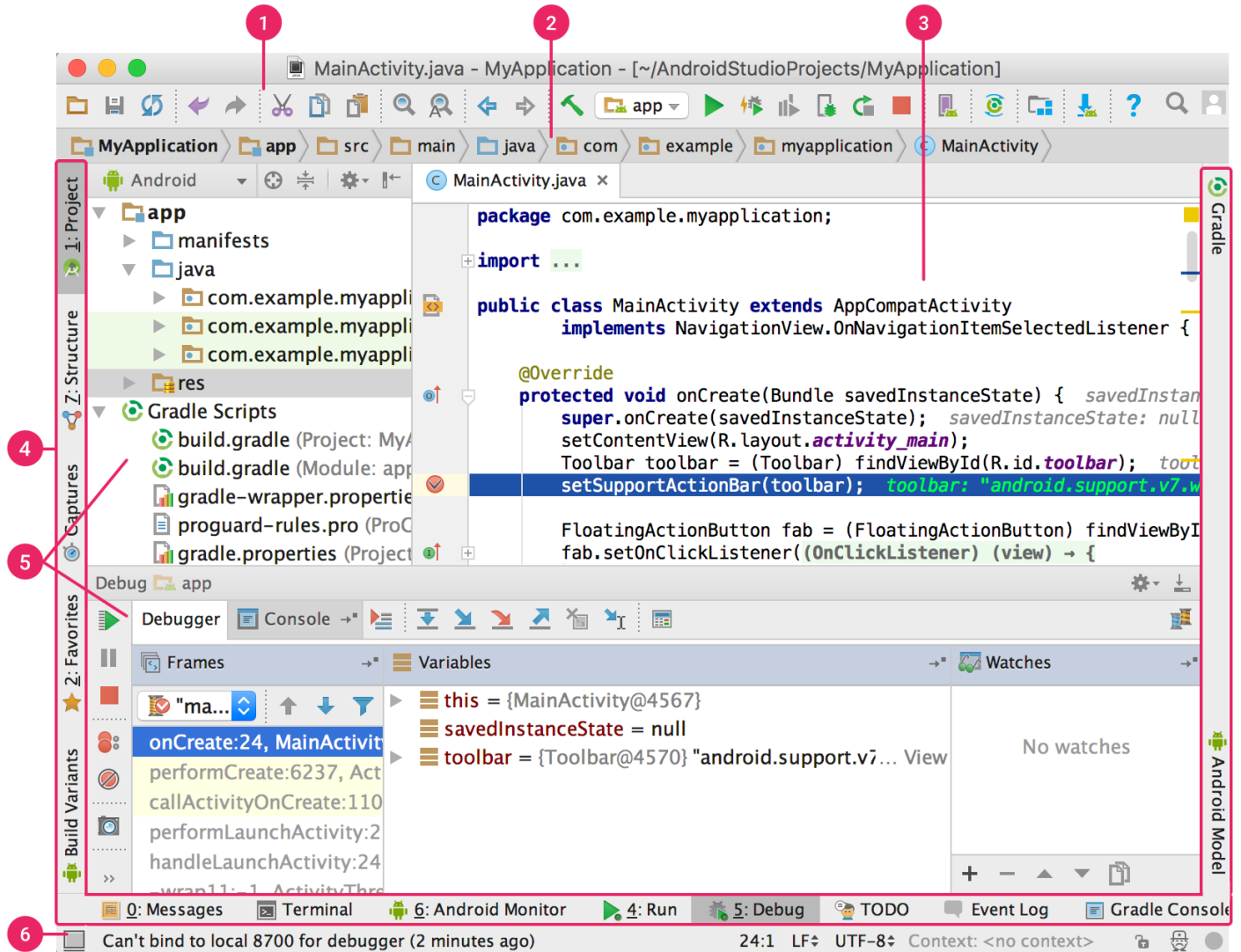


Figura 3 : Ventana principal de Android Studio

1. La **barra de herramientas** te permite realizar una gran variedad de acciones, como ejecutar tu aplicación e iniciar las herramientas de Android.
2. La **barra de navegación** te ayuda a explorar tu proyecto y abrir archivos para editar. Proporciona una vista más compacta de la estructura visible en la ventana **Proyecto**.
3. La **ventana del editor** es el área en la que puedes crear y modificar el código. Según el tipo de actividad actual, el editor puede cambiar. Por ejemplo, cuando ves un archivo de diseño, el editor muestra el Editor de diseño.
4. La **barra de la ventana de herramientas** se encuentra afuera de la ventana del IDE y contiene los botones que permiten expandir o contraer ventanas de herramientas individuales.
5. Las **ventanas de herramientas** te brindan acceso a tareas específicas, como la administración de proyectos, la búsqueda, el control de versiones, entre otras. Puedes expandirlas y contraerlas.
6. En la **barra de estado**, muestra el estado de tu proyecto y el IDE, además de advertencias o mensajes.

Puedes organizar la ventana principal para tener más espacio en pantalla si ocultas o desplaza las barras y ventanas de herramientas. También puedes usar combinaciones de teclas para acceder a la mayoría de las funciones del IDE.

En cualquier momento, puede realizar búsquedas en el código fuente, las bases de datos, las acciones y los elementos de la interfaz de usuario, entre otros. Para ello, presiona dos veces la tecla Mayús o haz clic en la lupa en la esquina superior derecha de la ventana de Android Studio. Esto puede ser muy útil, por ejemplo, si intentas localizar una acción específica del IDE que olvide cómo activar.

Ventanas de herramientas

En lugar de usar perspectivas preestablecidas, Android Studio sigue tu contexto y te ofrece automáticamente ventanas de herramientas relevantes mientras trabajas. En la configuración predeterminada, las ventanas de herramientas más usadas se fijan a la barra de la ventana de herramientas en los bordes de la ventana de la aplicación.

- Para expandir o contraer una ventana de herramientas, haz clic en el nombre de la herramienta, en la barra de la ventana de herramientas. También puedes arrastrar, fijar, no fijar, adjuntar y ocultar ventanas de herramientas.
- Para volver al diseño predeterminado actual de la ventana de herramientas, haga clic en **Ventana> Restaurar diseño predeterminado** o personalice su diseño predeterminado haciendo clic en **Ventana> Guardar diseño actual como predeterminado**.
- Para ocultar o mostrar la barra de la ventana de herramientas completas, haz clic en el icono de la ventana  en la esquina inferior izquierda de la ventana de Android Studio.
- Para ubicar una ventana de herramientas específicas, despliegue sobre su ícono y selección en el menú.

También puedes usar combinaciones de teclas para abrir ventanas de herramientas. En la Tabla 1, se muestran las combinaciones de teclas para las ventanas más comunes.

Tabla 1: Combinaciones de teclas para algunas ventanas de herramientas útiles

Ventana de herramientas	Windows y Linux	Mac
Proyecto	Alt + 1	Comando + 1
Control de versiones	Alt + 9	Comando +9
Ejecutar	Mayús + F10	Control + R

Depurar	Mayús + F9	Control + D
Logcat	Alt + 6	Comando + 6
Volver al editor	Esc	Esc
Ocultar todas las ventanas de herramientas	Control + Mayús + F12	Comando + Mayús + F12

Si quieres ocultar todas las barras de herramientas, ventanas de herramientas y pestañas del editor, haz clic en **Ver> Ingresar al modo libre de distracciones** . De esta manera, se habilita el modo *Distraction Free Mode* . Para salir del modo Distraction Free Mode, haz clic en **Ver> Salir de Distraction Free Mode** .

Puedes usar la *Búsqueda rápida* para buscar y filtrar en la mayoría de las ventanas de herramientas en Android Studio. Para usar la Búsqueda rápida, selecciona la ventana de herramientas y, luego, escribe tu búsqueda.

Para obtener más sugerencias, consulta [Combinaciones de teclas](#) .

Cómo completar el código

Android Studio ofrece tres opciones para completar el código, a las que puedes acceder con combinaciones de teclas.

Tabla 2. Combinaciones de teclas para completar código

Tipo	Descripción	Windows y Linux	Mac
Completar de manera básica	Muestras sugerencias básicas para variables, tipos, métodos y expresiones, entre otras. Si completa de manera básica dos veces seguidas, verás más resultados. Entre otros, miembros privados y miembros estáticos sin importar.	Control + Barra espaciadora	Control + Barra espaciadora
Completar de manera inteligente	Muestras relevantes en función del contexto. La función Completar de manera inteligente reconoce el tipo y los cambios de datos previstos. Si completa de manera inteligente dos veces	Control + Mayús + Barra espaciadora	Control + Mayús + Barra espaciadora

seguidas, verás más resultados. Por ejemplo, cadenas.

Completar enunciados	Completa la instrucción actual agregando elementos que faltan, como paréntesis, corchetes, llaves y formato, entre otros.	Control + Mayús + Mayús + Introducción	+ Mayús + Comando + Introducción
----------------------	---	--	----------------------------------

También puedes realizar correcciones rápidas para mostrar acciones de intención si presionas **Alt + Intro**.

Búsqueda de códigos de muestra

El navegador de muestras de código de Android Studio te ayuda a buscar muestras de código de Android de alta calidad proporcional por Google según el símbolo actualmente destacado en tu proyecto. Para obtener más información, [consulte Cómo buscar código de muestra](#).

Navegación

Aquí te ofrecemos algunas sugerencias para ayudarte a desplazarte por Android Studio.

- Alterna entre los archivos a los que accediste recientemente mediante la acción *Archivos recientes*. Presiona **Control + E** (**Comando + E** en Mac) para activar la acción Archivos recientes. De forma predeterminada, selecciona el último archivo al que accediste. También puedes acceder a cualquier ventana de herramientas en la columna izquierda de esta acción.
- Usa la acción *File Structure* para visualizar la estructura del archivo actual. Para activar la acción Estructura de archivos, presiona **Control + F12** (**Comando + F12** en Mac). Con esta acción, puedes navegar rápidamente hacia cualquier parte del archivo actual.
- Busca una clase específica en tu proyecto y navega hacia ella con la acción *Navega a clase*. Para activar la acción, presiona **Control + N** (**Comando + O** en Mac). Navegue a Clase admite expresiones sofisticadas, como camel jorobas, rutas, líneas de navegación, coincidencia de segundo nombre y otras. Si la llama dos veces seguidas, tendrá los resultados fuera de las clases de proyectos.
- Para navegar en un archivo o una carpeta, usa la acción *Navega hasta Archivo*. Si quieres activar la acción Navega hasta Archivo, presiona **Control + Mayús + N** (**Comando + Mayús + O** en Mac). Para buscar carpetas en lugar de archivos, agrega una / al final de la expresión.
- *Navegar a un método o campo por nombre* con la acción *Navegar a Símbolo*. Para activar la acción Navegar a Símbolo, presiona **Control + Mayús + Alt + N** (**Comando + Opción + O** en Mac).

- Encuentra todas las partes del código que hacen referencia a la clase, el método, el campo, el parámetro o la instrucción en la posición actual. Para ello, presiona **Alt + F7** (**Opción + F7** en Mac).

Estilo y formato

Mientras editas, Android Studio aplica automáticamente formatos y estilos según lo especificado en tu configuración de estilo de código. Puede personalizar la configuración de estilo de código programando el idioma, que incluye la especificación de convenciones para pestañas y sangrías, espacios, ajuste y llaves, y líneas en blanco. Para personalizar la configuración de estilo de tu código, haz clic en **Archivo > Configuración > Editor > Estilo de código** (**Android Studio > Preferencias > Editor > Estilo de código** en Mac).

Si bien el IDE aplica automáticamente formatos mientras trabajas, también puedes llamar explícitamente a la acción *Reformatear Código* si presionas **Control + Alt + L** (**Opción + Comando + L** en Mac), o bien aplica sangrías automáticas a todas las líneas si presionas **Control + Alt + I** (**Control + Opción + I** en Mac).

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mActionBar = getSupportActionBar();
    mActionBar.setDisplayHomeAsUpEnabled(true);
}
```

Figura 4: Código antes de la aplicación de formato

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mActionBar = getSupportActionBar();
    mActionBar.setDisplayHomeAsUpEnabled(true);
}

// Get reference to the drawer layout and set event listener
```

Formatted 7 lines
Show reformat dialog: ⌘⇧⌘L

Figura 5: Código después de la aplicación de formato

Aspectos básicos del control de versiones

Android Studio admite diferentes sistemas de control de versión (VCS), incluidos Git, GitHub, CVS, Mercurial, Subversion y Google Cloud Source Repositories.

Después de importar tu aplicación a Android Studio, usa las opciones del menú del VCS de Android Studio a fin de habilitar la compatibilidad con VCS para el sistema de control de versión deseada, crear un repositorio, importar los archivos nuevos al control de versión y realizar otras operaciones de control de versión:

1. En el menú del **VCS** de Android Studio, haz clic en **Habilitar integración de control de versiones**.
2. En el menú desplegable, selecciona un sistema de control de versión para asociarlo con la raíz del proyecto y, luego, haz clic en **OK**.

En el menú del VCS se mostrarán diversas opciones de control de versión según el sistema que hayas seleccionado.

Nota : También puede usar la opción del menú **Archivo> Configuración> Control de versión** para configurar y modificar los ajustes de control de versión.

Sistema de compilación de Gradle

Android Studio usa Gradle como base del sistema de compilación, y el [complemento de Android para Gradle](#) proporciona capacidades específicas de Android. Este sistema de compilación se ejecutará en una herramienta integrada desde el menú de Android Studio, y lo que modificará la línea de comandos. Puedes usar las funciones del sistema de compilación para lo siguiente:

- Personalizar, configurar y extender el proceso de compilación
- Crear varios APK para tu aplicación; diferentes funciones usan el mismo proyecto y los mismos módulos
- Volver a usar códigos y recursos en conjuntos de archivos fuente

Gracias a la flexibilidad de Gradle, puedes lograrlo sin modificar los archivos fuente de tu aplicación. Los archivos de compilación de Android Studio se denominan `build.gradle`. Son archivos de texto sin formato que usan la sintaxis [Groovy](#) para configurar la compilación con elementos que proporciona el complemento de Android para Gradle. Cada proyecto tiene un archivo de compilación de nivel superior para todo el proyecto y archivos de compilación de nivel de módulo independiente para cada módulo. Cuando se importa un proyecto existente, Android Studio genera automáticamente los archivos de compilación necesarios.

Para obtener más información sobre el sistema de compilación y su configuración, consulta [Cómo configurar tu compilación](#).

Variantes de compilación

El sistema de compilación puede ayudarte a crear diferentes versiones de la misma aplicación a partir de un solo proyecto. Esto resulta útil cuando tienes una versión gratuita o una versión de tu aplicación, o si quieres distribuir múltiples APK para diferentes dispositivos de dispositivos en Google Play.

Para obtener más información sobre el uso de variantes de compilación, consulte la sección [Cómo configurar variantes de compilación](#).

Compatibilidad con varios APK

La compatibilidad con varios APK te permite crear de manera eficiente varios APK modificados en la densidad de la pantalla o en ABI. Por ejemplo, puedes crear APK individual de una aplicación para las densidades de pantalla hdpi y mdpi, y considerar una misma variante de modo que compartan la configuración de APK, javac, dx y ProGuard para la prueba.

Para obtener más información sobre la compatibilidad con varios APK, lee [Cómo compilar varios APK](#).

Reducción de recursos

La reducción de recursos en Android Studio quita automáticamente los recursos sin usar del paquete de tu aplicación y de las dependencias de bibliotecas. Por ejemplo, si en su aplicación se usan [Servicios de Google Play](#) para acceder a la funcionalidad de Google Drive y actualmente no usas el [Acceso con Google](#), la reducción de recursos puede quitar los diferentes recursos del elemento de diseño de los botones `SignInButton`.

Nota : La reducción de recursos funciona con herramientas de reducción de código, como ProGuard.

Para obtener más información acerca de cómo reducir el código y los recursos, consulta [Cómo reducir tu código y tus recursos](#).

Administración de dependencias

Las dependencias para tu proyecto están especificadas por nombre en el archivo `build.gradle`. Gradle se ocupa de buscar tus dependencias y hacer que estén disponibles en tu compilación. Puedes declarar dependencias de módulos, dependencias binarias remotas y dependencias binarias locales en tu archivo `build.gradle`. Android Studio configura los proyectos para que usen el repositorio central de Maven de manera predeterminada. (Esta configuración está incluida en el archivo de compilación de nivel superior del proyecto). Para obtener más información sobre la configuración de dependencias, [consulte Cómo agregar dependencias de compilación](#).

Herramientas de depuración y perfil

Android Studio te ayuda a depurar y mejorar el rendimiento de tu código. Esto incluye herramientas integradas de depuración y análisis de rendimiento.

Depuración integrada

Usa la depuración integrada para mejorar las revisiones de código en la vista del depurador con verificación integrada de referencias, expresiones y valores de variables. La información de depuración integrada incluye:

- Valores de variables integradas
- Referencia a objetos que hacen referencia a un objeto seleccionado
- Valores de retorno de métodos
- Expresiones lambda y de operador
- Valores de información sobre herramientas

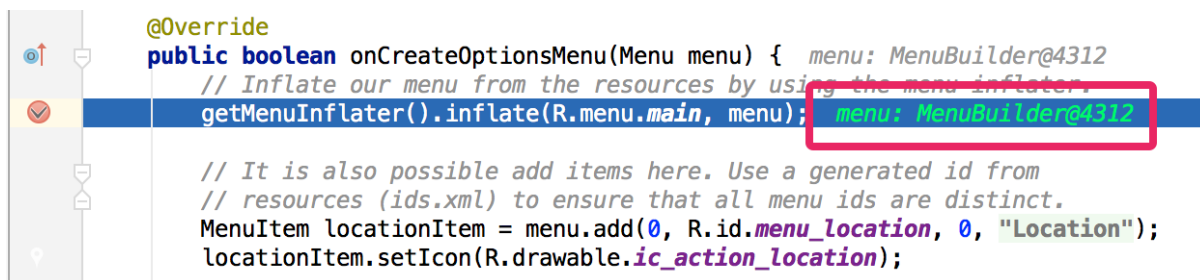


Figura 6 : Valor de una variable integrada

Para activar la depuración integrada, en la ventana **Depurar** , haga clic en **Configuración**  y seleccione la casilla de verificación para activar la opción **Mostrar valores en línea** .

Generador de perfiles de rendimiento

Android Studio ofrece generadores de perfiles de rendimiento para que puedas realizar un seguimiento más fácil del uso de CPU y memoria de tu aplicación, encontrar objetos asignados, ubicar pérdidas de memoria, optimizar el rendimiento de los gráficos y analizar las solicitudes de red. Con tu aplicación ejecutándose en un dispositivo o emulador, abre la etiqueta del **Android Profiler** .

Para obtener más información sobre los generadores de perfiles de rendimiento, [consulte Herramientas de perfilado de rendimiento](#) .

Volcado de montón

Cuando controlas el uso de la memoria en Android Studio, puedes activar simultáneamente la recolección de elementos no usados y volcar el montón de Java a una captura instantánea del montón en un archivo de formato binario HPROF específico de Android. El visor de HPROF muestra las clases, las instancias de cada clase y un árbol de referencia para ayudar a realizar el seguimiento del uso de la memoria y encontrar fugas de memoria.

Para obtener más información sobre cómo trabajar con volcados de montón, consulta [Cómo inspeccionar el montón y las asignaciones](#).

Generador de perfiles de memoria

Puedes usar el generador de perfiles de memoria para realizar un seguimiento de la asignación de memoria y ver dónde se asignan los objetos cuando realizas las acciones determinadas. Conocer estas asignaciones te permite modificar el rendimiento de tu aplicación y el uso de la memoria ajustando las llamadas de método relacionado con esas acciones.

Para obtener información sobre el seguimiento y análisis de asignaciones, consulta [Cómo inspeccionar el montón y las asignaciones](#).

Acceso al archivo de datos

Las herramientas del SDK de Android, como [Systrace](#) y [logcat](#), contienen datos de rendimiento y depuración para un análisis detallado de la aplicación.

Para ver los archivos de datos generados disponibles, abre la ventana de herramientas Capturas. En la lista de los archivos generados, haz doble clic en uno para ver los datos. Haz clic con el botón derecho en cualquier archivo `.hprof` para convertirlo al formato de archivo de [uso de RAM](#) estándar.

Inspecciones de código

Cada vez que compilas tu programa, Android Studio activa automáticamente las [inspecciones de pelusa](#) y otras [inspecciones de IDE](#) para ayudarte a identificar y corregir problemas fácilmente con la calidad estructural de tu código.

La herramienta de Lint comprueba los archivos fuente de tu proyecto de Android en busca de posibles errores y para realizar mejoras relacionadas con la precisión, la seguridad, el rendimiento, la usabilidad, la accesibilidad y la internacionalización.

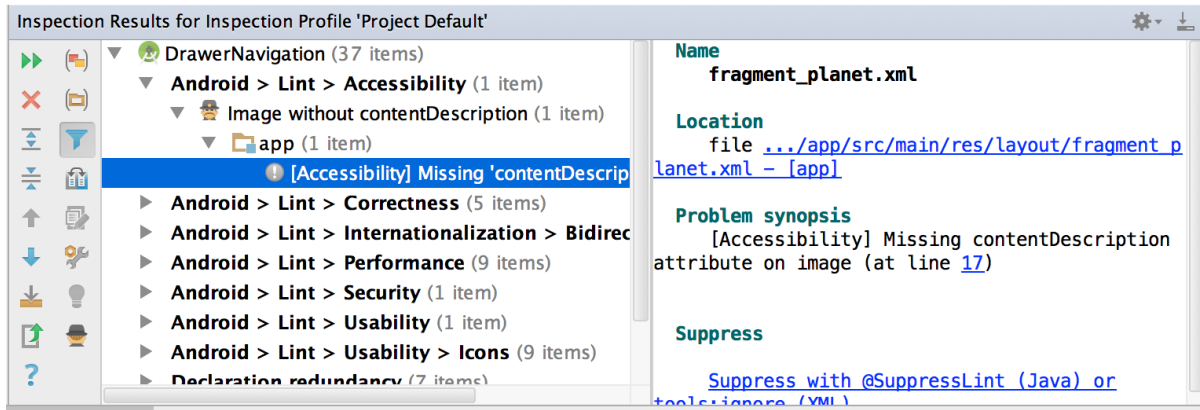


Figura 7: Resultados de una inspección de Lint en Android Studio.

Además de las verificaciones de Lint, Android Studio también realiza inspecciones de código de IntelliJ y validaciones de anotaciones para simplificar su flujo de trabajo de codificación.

Para obtener más información, [consulte Cómo mejorar su código con controles de Lint](#).

Anotaciones en Android Studio

Android Studio admite anotaciones de variables, parámetros y valores de retorno a fin de ayudar a detectar errores, como errores de puntero nulo y conflictos de tipos de recurso. Android SDK Manager empaqueta la biblioteca de compatibilidad-anotaciones en el Repositorio de compatibilidad de Android para usarla con Android Studio. Android Studio valida las anotaciones configuradas durante la inspección del código.

Para obtener más información sobre las anotaciones de Android, consulte [Cómo mejorar la inspección del código con anotaciones](#).

Mensajes de registro

Cuando creas y ejecutas tu aplicación con Android Studio, puedes ver los resultados de [adb](#) y los mensajes de registro del dispositivo en la [ventana Logcat](#).

Generación de perfiles de rendimiento

Si desea perfilar el rendimiento de red, memoria y CPU de tu aplicación, abre [Android Profiler](#) haciendo clic en **Ver > Herramienta Windows > Android Profiler**.

Cómo acceder a tu cuenta de acceso

Puede acceder a su cuenta de desarrollador en Android Studio para acceder a herramientas adicionales que requieren autenticación, como [Cloud Tools para Android Studio](#) y la [herramienta de prueba Acciones en aplicaciones](#) . Al acceder, le otorgo a esas herramientas permiso para ver y administrar tus datos en todos los servicios de Google.

Después de abrir un proyecto en Android Studio, puede acceder a su cuenta de software o cambiar la cuenta de procesador de la siguiente manera:

1. Haz clic en el ícono de perfil  al final de la barra de herramientas, como se muestra en la figura 8.



Figura 8: Haz clic en el ícono de perfil al final de la barra de herramientas para acceder.

2. En la ventana que aparece, realiza una de las siguientes acciones:
 - Si aún no accediste, haz clic en **Acceder** y permite que Android Studio acceda a los servicios de la lista.
 - Si ya accediste a tu cuenta, haz clic en **Agregar cuenta** para acceder con otra cuenta de Google. También puedes hacer clic en **Salir** y repetir los pasos anteriores para acceder a una cuenta diferente.