

# **ANDROID STUDIO**

**AUTOR: MAX JOSÉ BERMÚDEZ LEÓN**

**MARZO: 2021**



**San Marcos**

## Introducción

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está 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 apps 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 código y recursos a la app en ejecución sin reiniciarla
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra
- Variedad de marcos de trabajo y herramientas de prueba
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros
- Compatibilidad con C++ y NDK
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine



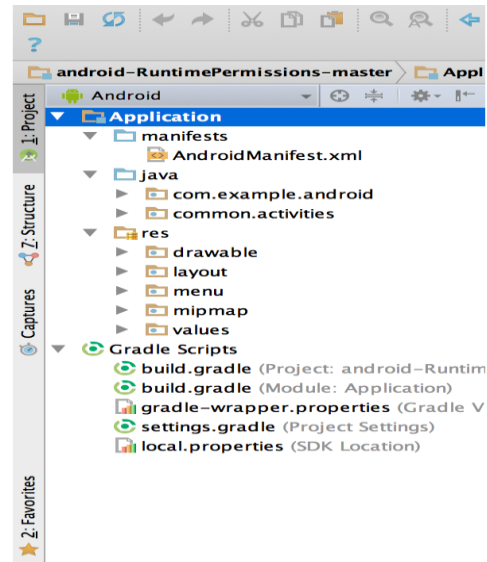
## Tabla de contenido

Introducción.....	1
.....	1
Estructura del proyecto .....	4
Interfaz de usuario .....	6
Ventanas de herramientas.....	8
Cómo completar el código .....	10
Cómo buscar código de ejemplo .....	11
Navegación .....	11
Estilo y formato .....	13
Aspectos básicos del control de versiones .....	13
<b>Sistema de compilación de Gradle .....</b>	<b>14</b>
Variantes de compilación.....	15
Compatibilidad con varios APK.....	15
Reducción de recursos.....	16
Administración de dependencias.....	16
Herramientas de depuración y perfil .....	17
Depuración integrada .....	17
Generador de perfiles de rendimiento .....	18
Volcado de montón .....	18
Memory Profiler .....	18
Acceso al archivo de datos.....	19
Inspecciones de código.....	19
Anotaciones en Android Studio.....	20
Mensajes de registro .....	20
Generación de perfiles de rendimiento.....	20
<b>Cómo acceder a tu cuenta de desarrollador .....</b>	<b>20</b>
Cómo instalar Android Studio.....	22
<b>Windows</b> .....	<b>22</b>
<b>Mac</b> .....	<b>22</b>
<b>Linux</b> .....	<b>23</b>
Bibliotecas requeridas para máquinas de 64 bits .....	24
<b>Sistema operativo Chrome</b> .....	<b>25</b>

Conclusiones y recomendaciones.....	28
Referencias bibliográficas .....	29

**Las formas normales proporcionan el criterio necesario para determinar el grado de vulnerabilidad a inconsistencias lógicas y anomalías en los datos. Cuanto más alto sea el número de la forma normal, menos vulnerable será nuestra tabla. Y sí, he dicho nuestra tabla, porque las formas normales se aplican a ellas, individualmente. Para poder decir que una base de datos cumple una determinada forma normal, todas sus tablas deberán cumplirla.**

## Estructura del proyecto



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

Android

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

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

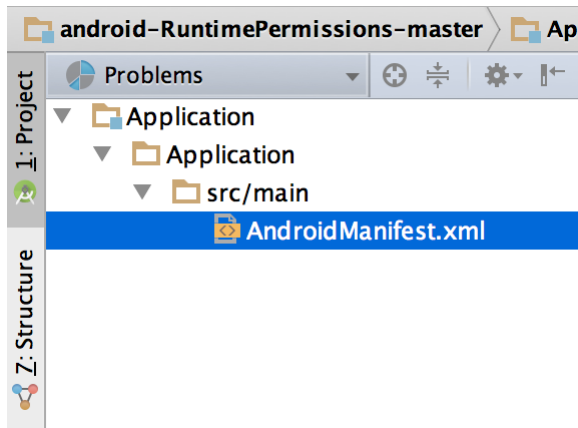
Puedes ver todos los archivos de compilación en el nivel superior de **Secuencias de comando de Gradle** y cada módulo de app contiene las siguientes carpetas:

- **manifests:** 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, strings 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 real de archivos del proyecto, selecciona **Project** en el menú desplegable **Project** (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 app. Por ejemplo, si seleccionas la vista **Problems** de tu proyecto, se mostrarán vínculos a los archivos fuente que contengan errores conocidos de codificación y sintaxis, como una etiqueta de cierre faltante en un elemento XML en un archivo de diseño.

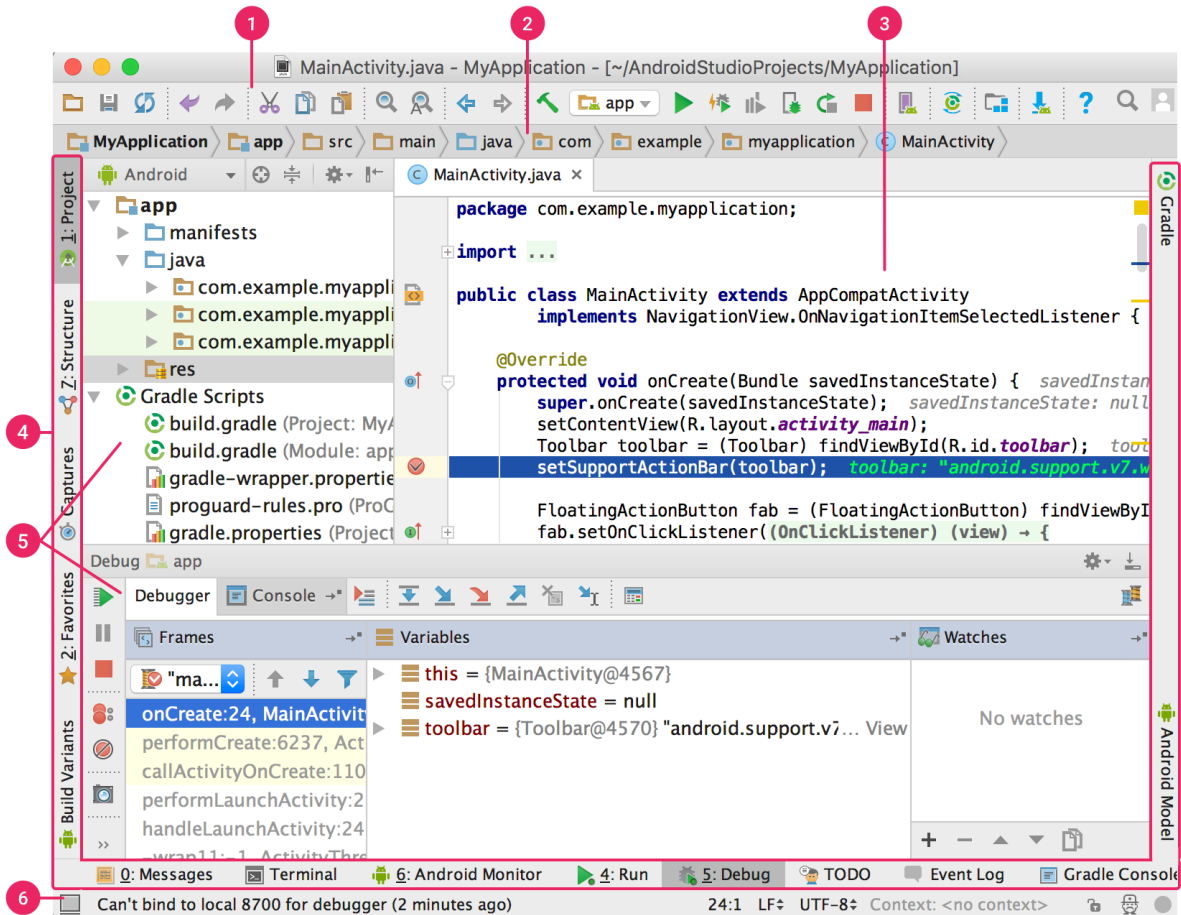




**Figura 2:** Archivos del proyecto en la vista Problems, en la que se muestra un archivo de diseño con un problema

Interfaz de usuario

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



1. La **barra de herramientas** te permite realizar una gran variedad de acciones, como ejecutar tu app 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 **Project**.
3. La **ventana del editor** es el área en la que puedes crear y modificar 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 te 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**, se 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 ocultando o desplazando 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, puedes 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 olvidaste cómo activar.

#### Ventanas de herramientas

En lugar de usar perspectivas predeterminadas, Android Studio sigue tu contexto y te ofrece automáticamente ventanas de herramientas relevantes mientras trabajas. De manera

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 app.

- 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, haz clic en **Window > Restore Default Layout** o personaliza tu diseño predeterminado haciendo clic en **Window > Store Current Layout as Default**.
- Para mostrar u ocultar la barra de la ventana de herramientas completa, haz clic en el ícono de ventana , en la esquina inferior izquierda de la ventana de Android Studio.
- Para ubicar una ventana de herramientas específica, desplázate sobre su ícono y selecciónala 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 **View > Enter Distraction Free Mode**. De esta manera, se habilita el modo *Distraction Free Mode*. Para salir del modo Distraction Free Mode, haz clic en **View > Exit 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.

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	Muestra sugerencias básicas para variables, tipos, métodos y expresiones, entre otras. Si completas 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	Muestra opciones relevantes en función del contexto. La función Completar de manera inteligente reconoce el tipo y los flujos de datos previstos. Si completas de manera inteligente dos veces seguidas, verás más resultados. Por ejemplo, cadenas.	Control + Mayús + Barra espaciadora	Control + Mayús + Barra espaciadora
Completar instrucciones	Completa la instrucción actual agregando elementos que faltan, como paréntesis, corchetes, llaves y formato, entre otros.	Control + Mayús + Intro	Mayús + Comando + Intro

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

### Cómo buscar código de ejemplo

El Navegador de muestras de código de Android Studio te ayuda a buscar muestras de código de Android de alta calidad proporcionadas por Google según el símbolo actualmente destacado en tu proyecto.

### 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 *Recent Files*. Presiona **Control + E** (**Comando + E** en Mac) para activar la acción Archivos recientes. De forma predeterminada, se 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 File Structure, presiona **Control + F12 (Comando + F12 en Mac)**. Con esta acción, podrás navegar rápidamente hacia cualquier parte del archivo actual.
- Busca una clase específica en tu proyecto y navega hacia ella con la acción *Navigate to Class*. Para activar la acción, presiona **Control + N (Comando + O en Mac)**. Navigate to Class admite expresiones sofisticadas, como camel humps, rutas, líneas de navegación, coincidencia de segundo nombre y otras. Si la llamas dos veces seguidas, te mostrará los resultados fuera de las clases de proyectos.
- Para navegar a un archivo o una carpeta, usa la acción *Navigate to File*. Si quieres activar la acción Navigate to File, presiona **Control + Mayús + N (Comando + Mayús + O en Mac)**. Para buscar carpetas en lugar de archivos, agrega una / al final de la expresión.
- Navega a un método o campo por nombre con la acción *Navigate to Symbol*. Para activar la acción Navigate to Symbol, presiona **Control + Mayús + Alt + N (Comando + Opción + O en Mac)**.
- Encuentra todas las partes de código que hagan 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. Puedes 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 **File > Settings > Editor > Code Style (Android Studio > Preferences > Editor > Code Style en Mac)**.

Si bien el IDE aplica formato de manera automática mientras trabajas, también puedes llamar explícitamente a la acción *Reformat Code* si presionas **Control + Alt + L (Opción + Comando + L en Mac)**, o bien aplicar 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 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 app 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 deseado, 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 **Enable Version Control Integration**.
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 puedes usar la opción del menú **File > Settings > Version Control** 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 ejecuta en una herramienta integrada desde el menú de Android Studio, y lo hace independientemente de 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 app; 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 app. Los archivos de compilación de Android Studio se denominan `build.gradle`. Son archivos de texto sin formato que usan la sintaxis [Groovy](#) a fin de 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 independientes para cada módulo. Cuando importas un proyecto existente, Android Studio genera automáticamente los archivos de compilación necesarios.

#### Variantes de compilación

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

#### Compatibilidad con varios APK

La compatibilidad con varios APK te permite crear de manera eficiente varios APK basados en la densidad de la pantalla o en ABI. Por ejemplo, puedes crear APK individuales de una





app para las densidades de pantalla hdpi y mdpi, y considerarlos una misma variante de modo que compartan la configuración de APK, javac, dx y ProGuard para la prueba.

## Reducción de recursos

La reducción de recursos en Android Studio quita automáticamente los recursos sin usar del paquete de tu app ni las dependencias de bibliotecas. Por ejemplo, si tu aplicación usa 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 de elemento de diseño de los botones `SignInButton`.

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

## 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).

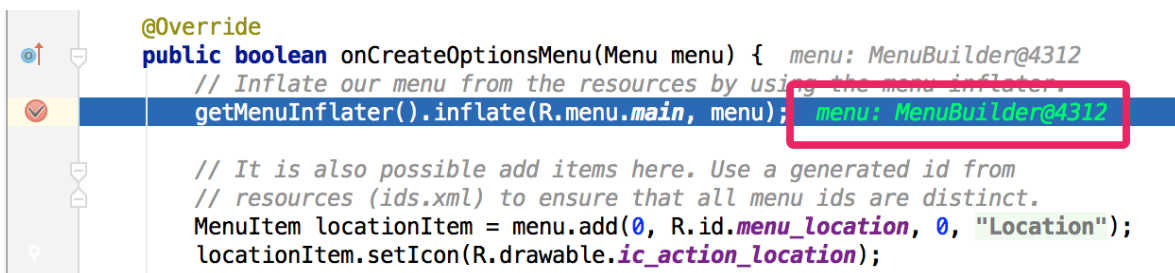
## Herramientas de depuración y perfil

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




```

@Override
public boolean onCreateOptionsMenu(Menu menu) { menu: MenuBuilder@4312
    // Inflate our menu from the resources by using the menu inflater.
    getMenuInflater().inflate(R.menu.main, menu); menu: MenuBuilder@4312

    // It is also possible add items here. Use a generated id from
    // resources (ids.xml) to ensure that all menu ids are distinct.
    MenuItem locationItem = menu.add(0, R.id.menu_location, 0, "Location");
    locationItem.setIcon(R.drawable.ic_action_location);
  
```

**Figura 6:** Valor de una variable integrada



A fin de habilitar la depuración integrada, en la ventana **Debug**, haz clic en **Settings**  y selecciona la casilla de verificación para activar la opción **Show Values Inline**.

### 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 app, encontrar objetos asignados, ubicar pérdidas de memoria, optimizar el rendimiento de los gráficos y analizar las solicitudes de red. Con tu app ejecutándose en un dispositivo o emulador, abre la pestaña del **Android Profiler**.

### Volcado de montón

Cuando controlas el uso de la memoria en Android Studio, puedes iniciar simultáneamente la recolección de elementos no utilizados 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 ayudarte a realizar el seguimiento del uso de memoria y encontrar fugas de memoria.

### Memory Profiler

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 determinadas acciones. Conocer estas asignaciones te permite optimizar el rendimiento de tu app y el uso de la memoria ajustando las llamadas de método relacionadas con esas acciones.

## Acceso al archivo de datos

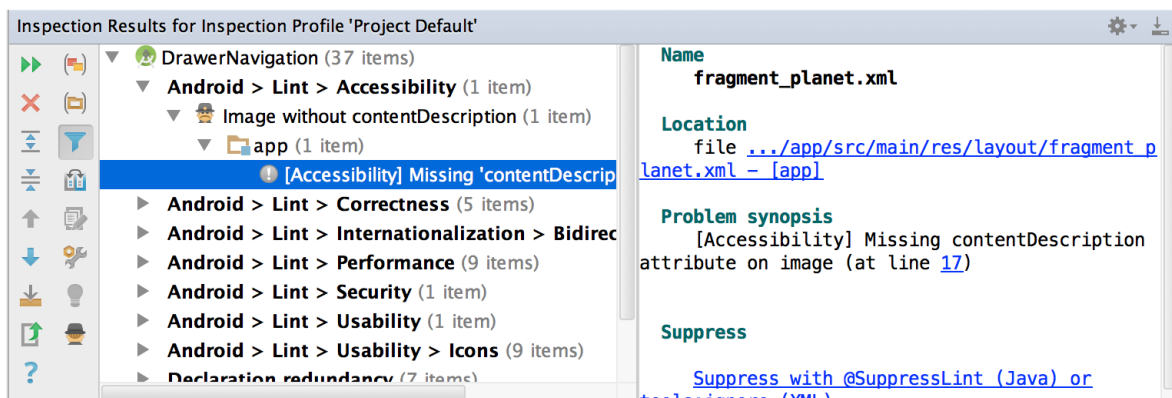
Las herramientas del SDK de Android, como Systrace y logcat, generan datos de rendimiento y depuración para un análisis detallado de la app.

Para ver los archivos de datos generados disponibles, abre la ventana de herramientas Captures. 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 ejecuta automáticamente inspecciones de Lint y otras inspecciones de IDE para ayudarte a identificar y corregir problemas con la calidad estructural de tu código de manera sencilla.

La herramienta lint comprueba los archivos de origen 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 valida anotaciones para simplificar tu flujo de trabajo de codificación.

### Anotaciones en Android Studio

Android Studio admite anotaciones de variables, parámetros y valores de retorno a fin de ayudarte a detectar errores, como excepciones 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.

### Mensajes de registro

Cuando creas y ejecutas tu app con Android Studio, puedes ver los resultados de adb y los mensajes de registro del dispositivo en la ventana de **Logcat**.

### Generación de perfiles de rendimiento


Si deseas perfilar el rendimiento de red, memoria y CPU de tu app, abre Android Profiler haciendo clic en **View > Tool Windows > Android Profiler**.

### Cómo acceder a tu cuenta de desarrollador

Puedes acceder a tu cuenta de desarrollador en Android Studio para acceder a herramientas adicionales que requieren autenticación, como Cloud Tools for Android Studio y

la herramienta de prueba Acciones en apps. Al acceder, le otorgas a esas herramientas permiso para ver y administrar tus datos en todos los servicios de Google.

Después de abrir un proyecto en Android Studio, puedes acceder a tu cuenta de desarrollador o cambiar de cuenta de desarrollador 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.

## Cómo instalar Android Studio

Primero, asegúrate de descargar la versión más reciente de Android Studio.

<https://developer.android.com/studio?hl=es-419>

### Windows

Para instalar Android Studio en Windows, haz lo siguiente:

1. Si descargaste un archivo `.exe` (recomendado), haz doble clic en él para iniciarlo.

Si descargaste un archivo `.zip`, extráelo y copia la carpeta **android-studio** en la carpeta **Archivos de programa**. A continuación, abre la carpeta **android-studio > bin** y, luego, inicia `studio64.exe` (para máquinas de 64 bits) o `studio.exe` (para las de 32 bits).

2. Sigue los pasos del asistente de configuración en Android Studio y asegúrate de instalar los paquetes de SDK que recomiende.

Cuando haya herramientas nuevas y otras API disponibles, Android Studio te lo informará por medio de una ventana emergente. También puedes buscar actualizaciones si haces clic en **Help > Check for Update**.

### Mac



Para instalar Android Studio en tu Mac, haz lo siguiente:

1. Ejecuta el archivo DMG de Android Studio.
2. Arrastra y suelta Android Studio en la carpeta Aplicaciones y, luego, inícialo.
3. Elige si deseas importar configuraciones anteriores de Android Studio y, luego, haz clic en **OK**.
4. El asistente de configuración de Android Studio te guiará por el resto de la configuración, lo que incluye la descarga de los componentes del SDK de Android que se necesiten para el desarrollo.

Cuando haya herramientas nuevas y otras API disponibles, Android Studio te lo informará por medio de una ventana emergente. También puedes buscar actualizaciones si haces clic en **Android Studio > Check for Updates**.

★ **Nota:** Si usas Android Studio en macOS Mojave o versiones posteriores, es posible que veas un mensaje para permitir que el IDE acceda a tus calendarios, contactos o fotos. Este mensaje se muestra debido a mecanismos de protección de la privacidad nuevos destinados a las aplicaciones que acceden a archivos dentro del directorio principal. Por lo tanto, si tu proyecto incluye archivos y bibliotecas del directorio principal y ves este mensaje, puedes seleccionar **Don't Allow**.

## Linux

Para instalar Android Studio en Linux, haz lo siguiente:





1. Extrae el archivo `.zip` que descargaste en una ubicación apropiada para tus aplicaciones, como dentro de `/usr/local/` para tu perfil de usuario o `/opt/` para usuarios compartidos.

Si usas una versión de Linux de 64 bits, asegúrate de instalar primero las bibliotecas requeridas para máquinas de 64 bits.

2. Para iniciar Android Studio, abre una terminal, navega al directorio `android-studio/bin/` y ejecuta `studio.sh`.
3. Selecciona si deseas o no importar configuraciones anteriores de Android Studio y, luego, haz clic en **OK**.
4. El asistente de configuración de Android Studio te guiará por el resto de la configuración, lo que incluye la descarga de los componentes del SDK de Android que se necesiten para el desarrollo.

★ **Sugerencia:** Para que Android Studio esté disponible en tu lista de aplicaciones, selecciona **Tools > Create Desktop Entry** en la barra de menú de Android Studio.

Bibliotecas requeridas para máquinas de 64 bits

Si ejecutas una versión de Ubuntu de 64 bits, deberás instalar algunas bibliotecas de 32 bits con el siguiente comando:

```
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 libbz2-1.0:i386
```

Si usas Fedora de 64 bits, el comando es el siguiente:

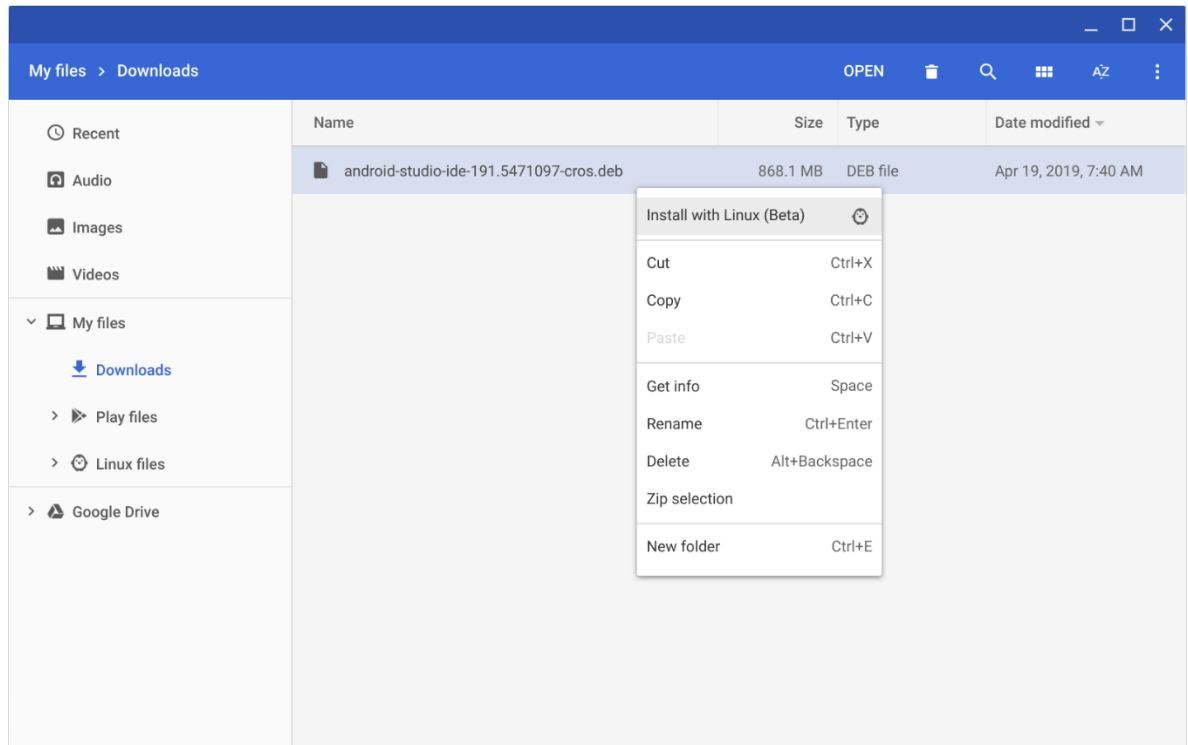
```
sudo yum install zlib.i686 ncurses-libs.i686 bzip2-libs.i686
```

Cuando haya herramientas nuevas y otras API disponibles, Android Studio te lo informará por medio de una ventana emergente. También puedes buscar actualizaciones si haces clic en **Help > Check for Update**.

## Sistema operativo Chrome

Sigue estos pasos para instalar Android Studio en el Sistema operativo Chrome:

1. Si aún no lo hiciste, [instala Linux para Chrome OS](#).
2. Abre la app de **Archivos** y busca el paquete DEB en la carpeta **Descargas** en **Mis archivos**.
3. Haz clic con el botón derecho en el paquete DEB y selecciona **Instalar con Linux (Beta)**.



- Si ya instalaste Android Studio, selecciona si deseas importar la configuración anterior de Android Studio y, luego, haz clic en **OK**.
4. El **asistente de configuración** de Android Studio te guiará por el resto de la configuración, lo que incluye la descarga de los componentes del SDK de Android que se necesiten para el desarrollo.
  5. Una vez que se complete la instalación, inicia Android Studio desde el Selector o desde la terminal de Linux del Chrome OS ejecutando `studio.sh` en el directorio de instalación predeterminado:

```
/opt/android-studio/bin/studio.sh
```

Se ve así. Cuando haya herramientas nuevas y otras API disponibles, Android Studio te lo informará por medio de una ventana emergente. También puedes buscar actualizaciones si

haces clic en **Help > Check for Update**.

★ **Nota:** Actualmente, Android Studio en Chrome OS permite implementar tu app solo en un dispositivo de hardware conectado. Si deseas obtener más información, lee [Cómo ejecutar apps en un dispositivo de hardware](#).

## Conclusiones y recomendaciones

El modelo entidad-relación ER es un modelo de datos que permite representar cualquier abstracción, percepción y conocimiento en un sistema de información formado por un conjunto de objetos denominados entidades y relaciones, incorporando una representación visual conocida como diagrama entidad-relación.

Otro aspecto importante sería el diseño y creación de la base de datos, donde existen distintos modos de organizar la información y representar las relaciones entre por datos los tres modelos lógicos principales dentro de una base de datos son el jerárquico, de redes y el relacional, los cuales tiene ciertas ventajas de procesamiento y de negocios.

## Referencias bibliográficas

- Sanz, Rafael León - Ramón Galán López. (2014). Introducción a la movilidad: 4G/LTE y el desarrollo de aplicaciones Android. Dextra Editorial
- Torres, Manuel. (2017). Desarrollo de aplicaciones móviles con Android. Editorial Marcombo.
- Developers Android (2021). Android Studio. <https://developer.android.com/studio/install?hl=es-419>
- Wikipedia (2021). Android Studio. Recopilado de: [https://es.wikipedia.org/wiki/Android\\_Studio](https://es.wikipedia.org/wiki/Android_Studio)



[www.usanmarcos.ac.cr](http://www.usanmarcos.ac.cr)

San José, Costa Rica