Actividad M1-01

Instalación de Android Estudio

Objetivo: Identificar el IDE y conocer las ventanas esenciales para el desarrollo en Android.

Carlos Alberto Ramírez Sánchez

No cuenta: 303044651

PROGRAMACIÓN DE DISPOSITIVOS MÓVILES

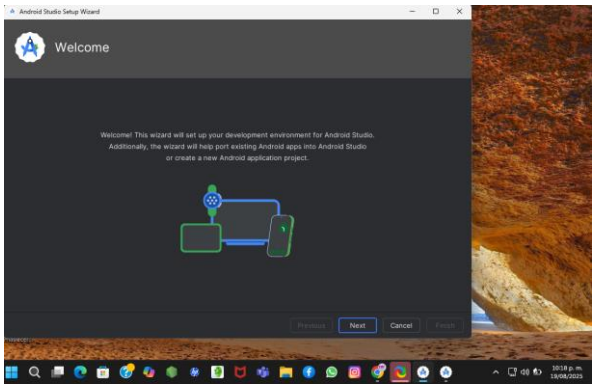
Agosto 2025

Introducción

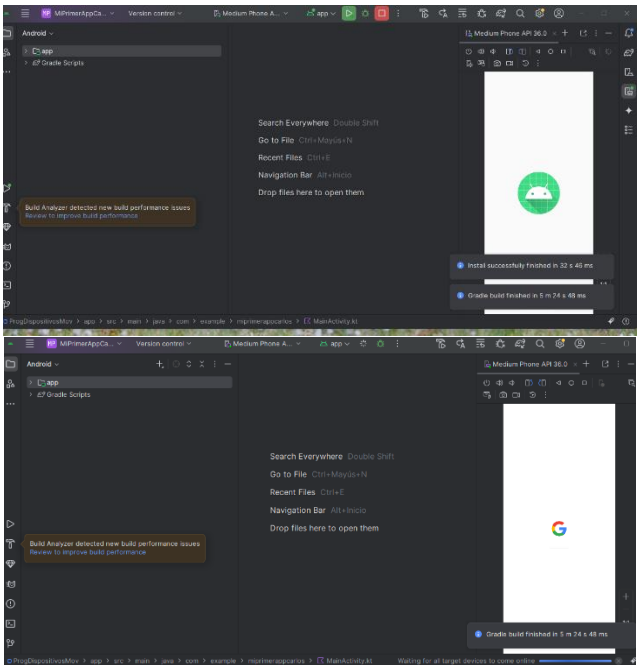
El propósito de esta actividad es instalar Android Studio correctamente en el sistema operativo Windows y familiarizarse con las principales ventanas del entorno de desarrollo (IDE) para el desarrollo de aplicaciones móviles en Android. Esta práctica permite identificar visualmente los elementos clave del IDE, como el explorador de recursos, el diseño de interfaces, el logcat, entre otros, lo cual es fundamental para comenzar a programar de manera efectiva.

Instalación de Android Studio en Windows

La instalación de Android Studio en Windows comenzó descargando el instalador oficial desde el sitio web de developer.android.com. Se siguieron los pasos recomendados por el asistente de instalación, el cual configuró el SDK de Android, el emulador y los componentes esenciales.



Una vez instalado, se creó un nuevo proyecto con una actividad vacía. El emulador fue configurado usando el dispositivo virtual Medium Phone API 36.0, y la aplicación corrió correctamente mostrando el ícono de Android en la pantalla de inicio.

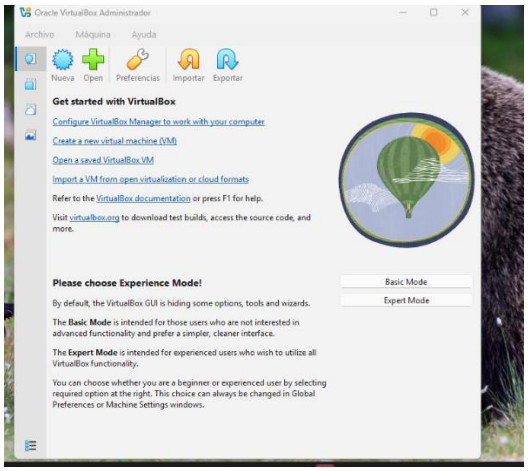


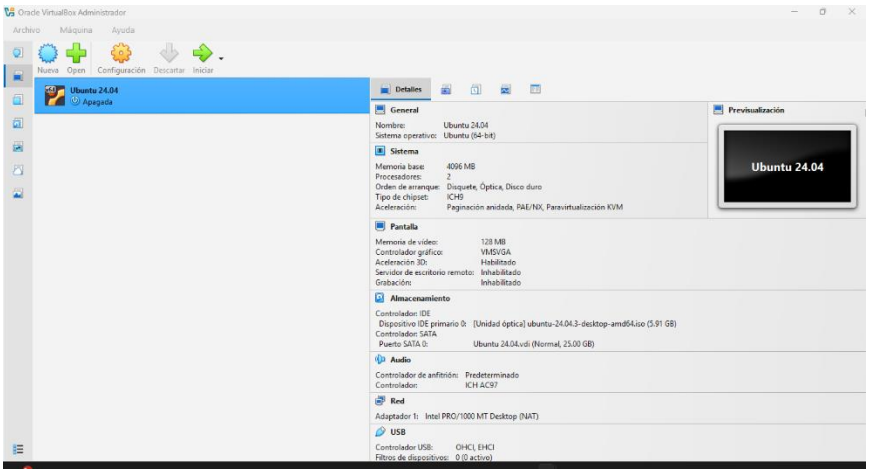
Instalación de Android Studio en GNU/Linux (Ubuntu 24.04 en VirtualBox)

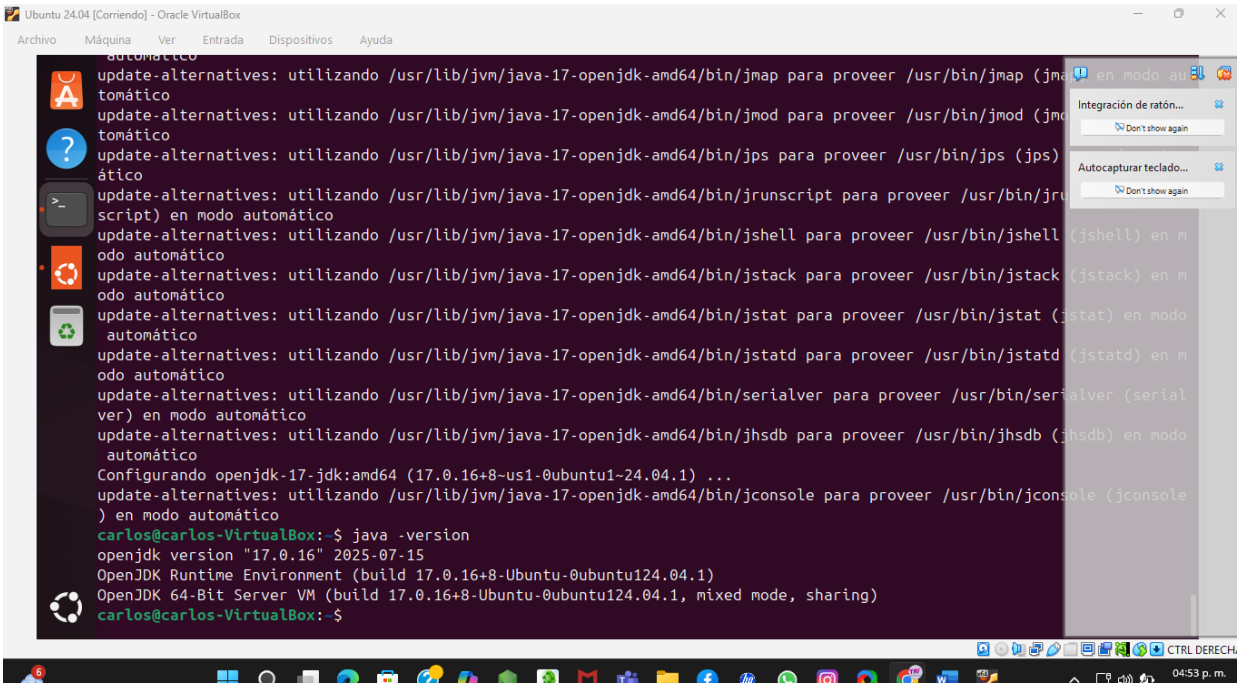
La instalación en GNU/Linux requirió algunos pasos adicionales, comenzando con la configuración de Java JDK 17 y la instalación de Android Studio mediante Snap.

Pasos principales:

1. **Verificación de Java**
    - Se instaló y verificó **OpenJDK 17**.



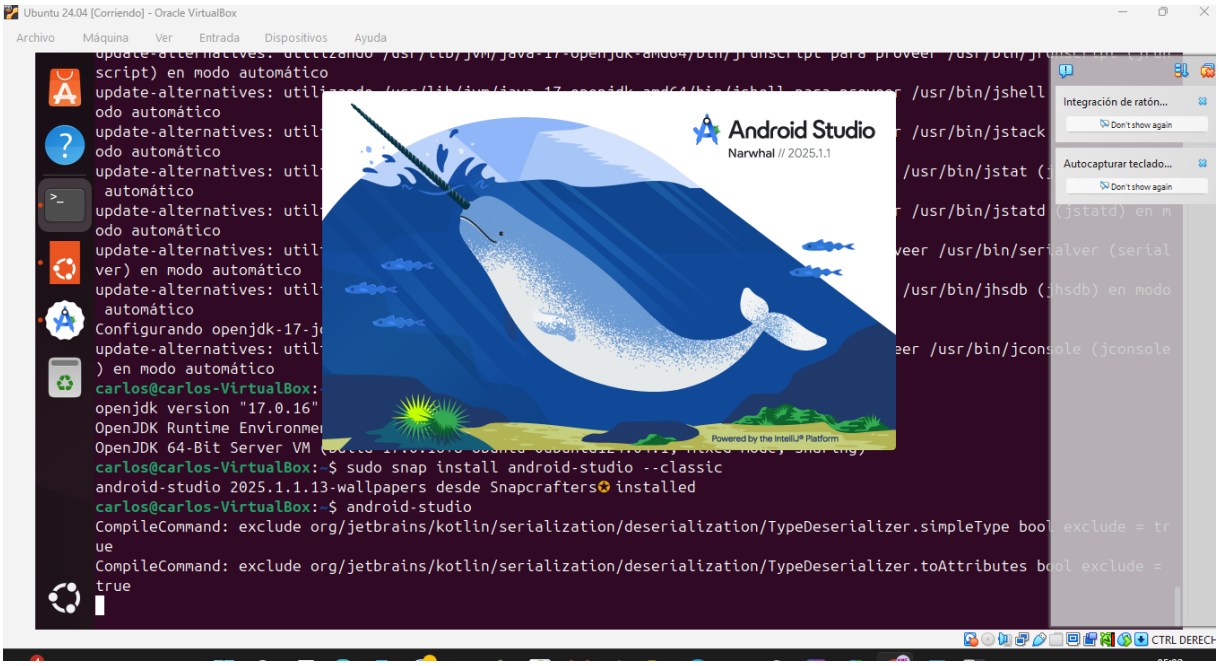


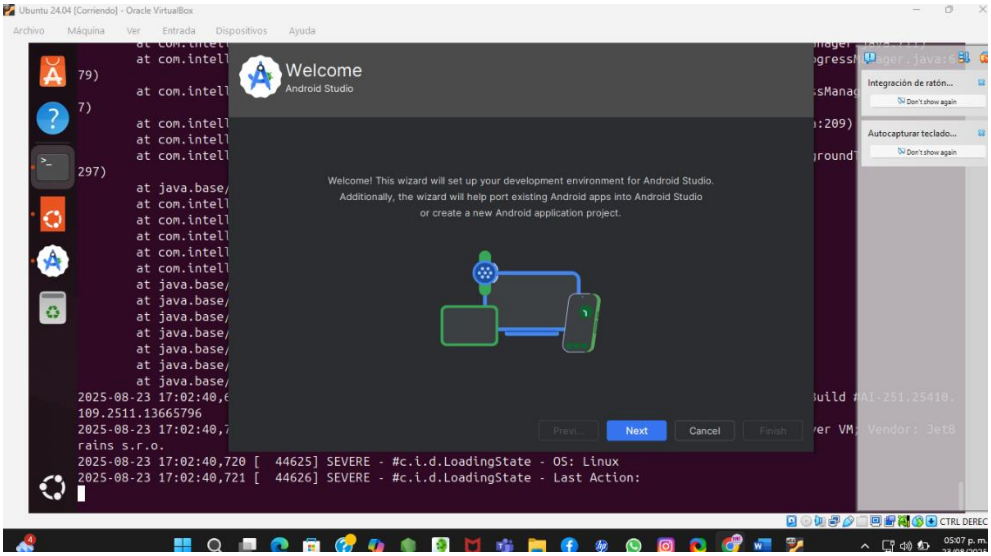


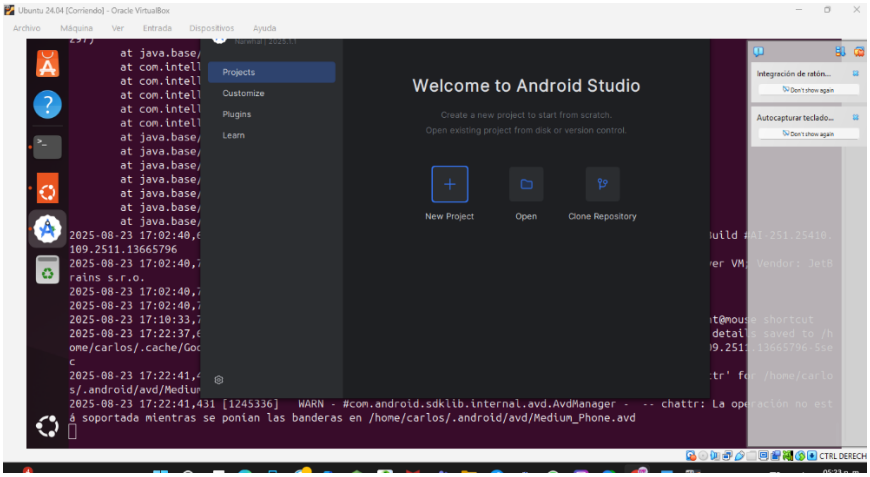
Instalación de Android Studio con Snap

- Se ejecutó el comando:

sudo snap install android-studio –classic


Ubuntu 24.04 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Android Studio
Narwhal // 2025.1.1
Powered by the IntelliJ® Platform
Integración de ratón...
Don't show again
Autocapturar teclado...
Don't show again
carlos@carlos-VirtualBox:~$ sudo snap install android-studio --classic
android-studio 2025.1.1.13-wallpapers desde Snapcrafters✪ installed
carlos@carlos-VirtualBox:~$ android-studio


Ubuntu 24.04 [Corriendo] - Oracle VirtualBox
Welcome
Android Studio
Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.
Next
Cancel
2025-08-23 17:02:40,720 [44625] SEVERE - #c.i.d.LoadingState - OS: Linux
2025-08-23 17:02:40,721 [44626] SEVERE - #c.i.d.LoadingState - Last Action:

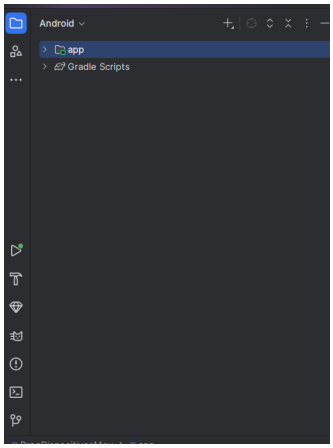


Desglose de ventanas del IDE

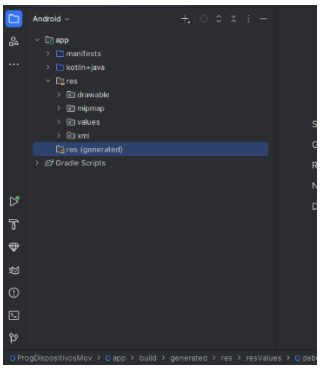
A continuación, se muestran y explican las ventanas esenciales del entorno de desarrollo Android Studio:

1. Project / Resource Explorer

En el panel izquierdo se encuentra el explorador de archivos del proyecto, donde se organiza el código fuente y los recursos de la app. Dentro de la carpeta res, se encuentran subcarpetas importantes como layout, values, mipmap y drawable.



1. Resources (res folder)

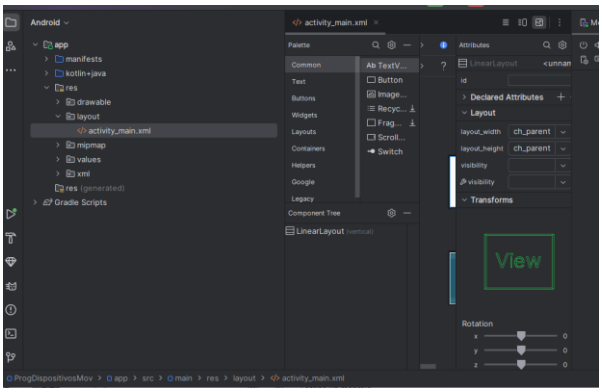


Ruta:
app > res

Descripción:
La carpeta res contiene todos los recursos que utiliza la app. Dentro de ella se encuentran subcarpetas como:

- drawable: imágenes e íconos.
- mipmap: íconos específicos para distintas densidades de pantalla.
- values: archivos XML donde se definen textos, colores, estilos, dimensiones, etc.
- xml: configuraciones adicionales como preferencias o mapas.

2. Layout

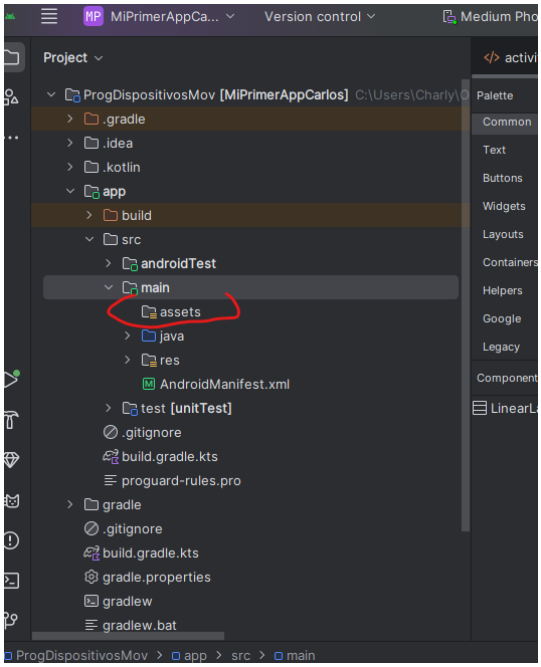


Ruta del archivo:
app > res > layout > activity_main.xml

Descripción:
La carpeta layout contiene los archivos XML donde se define la interfaz visual de cada pantalla de la app. En este caso, el archivo activity_main.xml es el layout principal de la aplicación.
En la vista del editor de diseño (Design), se puede trabajar con una interfaz visual para agregar botones, textos, imágenes, etc., sin necesidad de escribir todo el código

manualmente. También es posible configurar propiedades como el tamaño, posición, orientación o visibilidad de los elementos desde el panel de atributos.

3. Assets



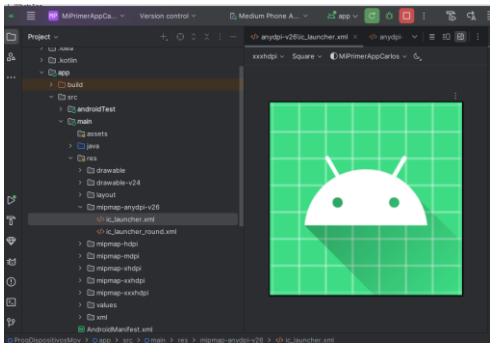
Ruta del archivo:
app > src > main > assets

Descripción:
La carpeta assets permite incluir archivos externos no estructurados como parte de la app, tales como archivos .txt, .json, .html, imágenes, sonidos, entre otros. A diferencia de otras carpetas como res, los archivos dentro de assets no son procesados ni compilados, sino que se conservan tal como están, permitiendo acceder a ellos mediante código Java/Kotlin.

Por ejemplo, si se agrega un archivo llamado datos.txt, se puede acceder desde la actividad principal (MainActivity.kt) usando el objeto assets. Esto es útil para mostrar textos personalizados, cargar configuraciones o leer datos sin conexión.

Este tipo de contenido se accede mediante funciones como assets.open("datos.txt"), las cuales permiten leer el contenido y usarlo dentro de la lógica de la app.

4. Mipmap



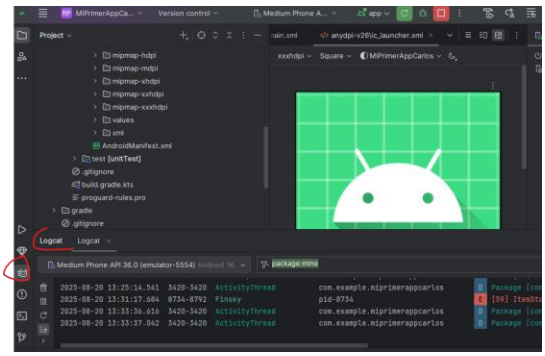
Ruta del archivo:
app > res > mipmap-anydpi-v26 > ic_launcher.xml

Descripción:
La carpeta mipmap contiene los íconos de la aplicación en diferentes resoluciones, asegurando que se vean correctamente en cualquier dispositivo Android, sin importar la densidad de pantalla.

En este caso, el archivo ic_launcher.xml corresponde al ícono adaptable de la app, y se acompaña de carpetas como mipmap-mdpi, mipmap-hdpi, mipmap-xhdpi, mipmap-xxhdpi y mipmap-xxxhdpi, que almacenan versiones del ícono en distintas calidades.

Estos recursos se utilizan principalmente para mostrar el logo de la aplicación en el launcher del dispositivo, la pantalla de inicio y la lista de aplicaciones. Gracias a las carpetas mipmap, Android selecciona automáticamente la imagen más adecuada según la pantalla del teléfono o tableta.

5. Logcat



Ruta de acceso:
View > Tool Windows > Logcat

Descripción:
La ventana Logcat muestra los registros en tiempo real del sistema y de la aplicación que se está ejecutando. Estos mensajes incluyen información del flujo de la app, advertencias y errores que ayudan al desarrollador a monitorear y depurar el proyecto.

En la parte superior se selecciona el dispositivo/emulador y el proceso de la app que se desea observar. También es posible aplicar filtros (por nombre de paquete o nivel de log como *Debug*, *Info* o *Error*) para facilitar la localización de eventos importantes.

En este ejemplo se observa la ejecución de la app en el emulador *Medium Phone API 36.0*, donde aparecen mensajes del sistema y del paquete com.example.miprimerappcarlos.

Conclusión

Conclusiones

La instalación de Android Studio en Windows y GNU/Linux (Ubuntu 24.04 en VirtualBox) permitió comprobar las diferencias y similitudes en los procesos de configuración del entorno de desarrollo. Mientras que en Windows el procedimiento resultó más directo gracias al asistente gráfico de instalación, en Ubuntu fue necesario realizar pasos adicionales, como la instalación de OpenJDK 17 y la configuración mediante Snap, lo cual refuerza la importancia de conocer los comandos básicos de administración en sistemas basados en Linux.

Una vez completada la instalación en ambos sistemas, se logró ejecutar Android Studio y crear un proyecto inicial, verificando el correcto funcionamiento del IDE y del emulador. Esto no solo aseguró que la herramienta estuviera lista para su uso, sino que también brindó la oportunidad de explorar las ventanas principales como Project Explorer, res folder, layout, assets, mipmap y Logcat, que son esenciales para la organización de los recursos, la construcción de interfaces y la depuración de las aplicaciones.

La práctica proporcionó una visión comparativa sobre la instalación en diferentes plataformas, demostrando que el dominio del IDE es independiente del sistema operativo y que lo realmente importante es comprender la estructura de proyectos y el uso de las herramientas internas. En consecuencia, esta actividad sentó las bases necesarias para continuar con el aprendizaje y desarrollo de aplicaciones móviles en Android, resaltando la importancia de familiarizarse tanto con el proceso de instalación como con el entorno de trabajo que ofrece Android Studio.

Fuentes:

- Android Developers. (s. f.). *Install Android Studio*. Recuperado de: https://developer.android.com/studio/install
- JetBrains. (s. f.). *IntelliJ IDEA and Android Studio*. JetBrains. https://www.jetbrains.com/idea/
- Canonical. (s. f.). *Snapcraft – Android Studio package*. Snapcraft.io. https://snapcraft.io/android-studio
- Oracle. (s. f.). *Java SE Development Kit 17 Documentation*. Oracle. https://docs.oracle.com/en/java/javase/17/
- Ubuntu Documentation. (s. f.). *Install additional software*. Ubuntu Official Docs. https://help.ubuntu.com/stable/ubuntu-help/addremove-install.html