



UNIVERSIDAD NACIONAL POLITECNICA

Programación Móvil



Tarea 2

2- Android Studio Editor: Manage your project vale 4 puntos

Hacer una investigación de esos enlaces

<https://developer.android.com/studio/projects>,
<https://developer.android.com/studio/projects/create-project>,
<https://developer.android.com/studio/intro/migrate>,
<https://developer.android.com/studio/projects/version-control>,
<https://developer.android.com/studio/intro/studio-config>,
<https://developer.android.com/studio/intro/update>

Carrera:

Ingeniería en computación.

Elaborado por:

Keylin Marlene Reyes Ramírez

1. Descripción general de un proyecto en Android Studio

Un proyecto de Android Studio contiene todos los archivos y recursos necesarios para desarrollar una aplicación de Android. Está organizado en una estructura de directorios jerárquica que facilita la búsqueda y administración de los archivos.

Directorios principales:

- **app:** Este directorio contiene los archivos fuente Java o Kotlin de tu aplicación, así como los recursos de la aplicación, como diseños de interfaz de usuario, imágenes, cadenas de texto y archivos de configuración.
- **build:** Este directorio contiene los archivos de compilación generados por Gradle, el sistema de compilación de Android.
- **gradle:** Este directorio contiene los archivos de configuración de Gradle que definen cómo se compila y empaqueta tu aplicación.

Carpetas importantes dentro del directorio app:

- **java/kotlin:** Esta carpeta contiene los archivos fuente Java o Kotlin de tu aplicación.
- **res:** Esta carpeta contiene los recursos de la aplicación, organizados en subcarpetas para diferentes tipos de recursos:
 - **drawable:** Contiene imágenes de la aplicación.
 - **layout:** Contiene los diseños de la interfaz de usuario de la aplicación.
 - **values:** Contiene archivos de configuración, como cadenas de texto y colores.
 - **xml:** Contiene otros archivos XML de recursos, como menús y configuraciones de preferencias.

Archivos importantes:

- **AndroidManifest.xml:** Este archivo define la configuración básica de tu aplicación, como el nombre, la versión y los componentes que contiene.
- **build.gradle:** Este archivo define la configuración de Gradle para tu módulo de aplicación.

- **settings.gradle:** Este archivo define la configuración de Gradle para tu proyecto completo.

Abrir un proyecto en Android Studio:

1. Abre Android Studio.
2. Selecciona **Archivo > Abrir**.
3. Navega hasta la carpeta del proyecto que deseas abrir.
4. Haz doble clic en el archivo del proyecto.

Recursos adicionales:

- Estructura de un proyecto en Android Studio [se quitó una URL no válida]
- Documentación de Gradle para Android [se quitó una URL no válida]

2. Crear un proyecto en Android Studio

Crear un proyecto en Android Studio es un proceso sencillo que te guía a través de la configuración básica de tu aplicación. A continuación, te presento los pasos generales:

Abre Android Studio:

- Inicia Android Studio en tu computadora.

Selecciona "Nuevo proyecto":

- En la pantalla de bienvenida, haz clic en "**Iniciar un nuevo proyecto de Android Studio**".
- Si ya tienes un proyecto abierto, selecciona **Archivo > Nuevo > Nuevo proyecto** desde el menú principal.

Configura tu proyecto:

- **Nombre:** Ingresa un nombre descriptivo para tu aplicación.
- **Paquete:** Especifica el nombre del paquete de tu aplicación. Este nombre debe ser único y seguir las convenciones de nomenclatura de Java.
- **Ubicación:** Selecciona la ubicación en tu computadora donde deseas guardar los archivos de tu proyecto.
- **Idioma:** Elige el lenguaje de programación que deseas usar para desarrollar tu aplicación: Kotlin o Java.

Selecciona una plantilla:

- Android Studio ofrece diferentes plantillas para ayudarte a comenzar con tu proyecto.
- Puedes elegir entre:
 - **Actividad vacía:** Crea una aplicación con una pantalla vacía.
 - **Actividad con Compose:** Crea una aplicación utilizando la interfaz de usuario declarativa de Compose para Android.
 - **Actividad con fragmentos:** Crea una aplicación que utiliza fragmentos para administrar diferentes partes de la interfaz de usuario.
 - **Otras plantillas:** Hay más plantillas disponibles para casos de uso específicos.

Personaliza tu proyecto:

- Dependiendo de la plantilla que hayas elegido, es posible que puedas configurar opciones adicionales.
- Por ejemplo, puedes especificar la versión mínima de Android que tu aplicación admitirá o agregar actividades adicionales a tu proyecto.

Finaliza la creación del proyecto:

- Haz clic en **Finalizar** para crear tu proyecto.

Explora tu proyecto:

- Android Studio abrirá tu proyecto y mostrará la estructura de directorios y archivos.
- Puedes comenzar a editar el código, diseñar las interfaces de usuario y agregar recursos a tu aplicación.

Consejos:

- Es recomendable comenzar con una plantilla simple como "Actividad vacía" para familiarizarte con la estructura del proyecto y las herramientas de Android Studio.
- A medida que te sientas más cómodo, puedes explorar las diferentes plantillas y opciones disponibles para crear aplicaciones más complejas.

3. Cómo migrar a Android Studio

Existen dos métodos principales para migrar a Android Studio, dependiendo de la configuración actual de tu proyecto:

1. Migración desde IntelliJ IDEA:

Si tu proyecto actual de Android reside en IntelliJ IDEA y utiliza el sistema de compilación Gradle, puedes importarlo directamente a Android Studio sin necesidad de pasos adicionales. Sigue estos pasos:

- **Abre Android Studio.**
- Selecciona **Archivo > Abrir**.
- Navega hasta la carpeta raíz de tu proyecto IntelliJ IDEA.
- Selecciona el archivo build.gradle y haz clic en **Abrir**.

Android Studio importará tu proyecto y configurará el entorno de desarrollo.

2. Migración mediante la creación de un nuevo proyecto:

Si tu proyecto actual utiliza un sistema de compilación diferente a Gradle o si deseas reestructurarlo, puedes migrarlo creando un nuevo proyecto en Android Studio y copiando manualmente los archivos fuente. Sigue estos pasos:

- **Crea un nuevo proyecto vacío en Android Studio.**
- **Copia los archivos fuente de tu proyecto actual:**
 - **Java/Kotlin:** Copia todos los archivos .java o .kt de tu proyecto actual a la carpeta src/main/java o src/main/kotlin de tu nuevo proyecto en Android Studio.
 - **Recursos:** Copia las carpetas res/drawable, res/layout, res/values y res/xml de tu proyecto actual a las carpetas correspondientes de tu nuevo proyecto en Android Studio.
 - **AndroidManifest.xml:** Copia el archivo AndroidManifest.xml de tu proyecto actual al directorio raíz de tu nuevo proyecto en Android Studio.
- **Actualiza las dependencias:**
 - Abre el archivo build.gradle de tu nuevo proyecto en Android Studio.

- Revisa las dependencias declaradas y asegúrate de que estén actualizadas y sean compatibles con la versión de Android Studio que estás utilizando.
- **Configura el proyecto:**
 - Abre el archivo settings.gradle de tu nuevo proyecto en Android Studio.
 - Asegúrate de que la ruta al directorio de tu proyecto esté configurada correctamente.
- **Sincroniza el proyecto:**
 - En Android Studio, haz clic en **Archivo > Sincronizar proyecto con Gradle Files**.

Android Studio analizará tu proyecto, actualizará las dependencias y configurará el entorno de desarrollo.

Consejos:

- Antes de migrar, asegúrate de tener una copia de seguridad completa de tu proyecto actual.
- Si te encuentras con problemas durante la migración, consulta la documentación oficial de Android Studio o busca ayuda en comunidades en línea de desarrolladores de Android.
- Una vez que hayas migrado tu proyecto a Android Studio, aprovecha las funciones y herramientas avanzadas que ofrece este IDE para mejorar tu flujo de trabajo de desarrollo.

4. Conceptos básicos del control de versiones en Android Studio

El control de versiones es una práctica esencial para cualquier proyecto de software, y Android Studio ofrece una integración perfecta con Git, un sistema de control de versiones popular. Git te permite realizar un seguimiento de los cambios realizados en tu código fuente, colaborar con otros desarrolladores y revertir a versiones anteriores si es necesario.

A continuación, se presentan algunos conceptos básicos del control de versiones en Android Studio:

1. Repositorio:

Un repositorio es un lugar centralizado donde se almacena el historial de tu proyecto. Puede estar ubicado en un servidor remoto como GitHub o GitLab, o en tu computadora local.

2. Commit:

Un commit es una instantánea de los cambios realizados en tu código fuente en un momento determinado. Cada commit incluye un mensaje que describe los cambios realizados.

3. Branch:

Una rama es una línea de desarrollo independiente dentro de tu proyecto. Te permite trabajar en diferentes funciones o correcciones de errores sin afectar el código principal.

4. HEAD:

El puntero HEAD apunta al commit más reciente en tu rama actual.

5. Flujo de trabajo de Git:

El flujo de trabajo de Git es un proceso para realizar cambios en tu código fuente, revisarlos y enviarlos al repositorio. Existen diferentes flujos de trabajo, como Git Flow y GitHub Flow.

Integración de Git en Android Studio:

Android Studio proporciona herramientas integradas para trabajar con Git. Puedes realizar las siguientes acciones directamente desde el IDE:

- **Inicializar un repositorio Git en tu proyecto.**

- **Agregar, eliminar y modificar archivos.**
- **Realizar commits y empujarlos al repositorio remoto.**
- **Ver el historial de commits.**
- **Resolver conflictos de fusión.**
- **Cambiar de rama.**
- **Deshacer commits.**

Beneficios del control de versiones:

- **Historial de cambios:** Permite realizar un seguimiento de todos los cambios realizados en el código fuente.
- **Colaboración:** Facilita la colaboración con otros desarrolladores en el mismo proyecto.
- **Revertir cambios:** Permite revertir a versiones anteriores del código si es necesario.
- **Experimentación segura:** Permite experimentar con nuevas funciones o correcciones de errores sin afectar el código principal.
- **Comunicación:** Facilita la comunicación entre los miembros del equipo sobre los cambios realizados en el código.

Consejos:

- **Comienza a usar el control de versiones desde el inicio de tu proyecto.**
- **Realiza commits con frecuencia y mensajes descriptivos.**
- **Utiliza ramas para trabajar en diferentes funciones o correcciones de errores.**
- **Comunica los cambios a tu equipo mediante mensajes de commit y solicitudes de extracción.**
- **Resuelve los conflictos de fusión de manera oportuna.**
- **Realiza copias de seguridad de tu repositorio de forma regular.**

5. Configuración inicial de Android Studio

Configuración inicial:

- Al iniciar Android Studio por primera vez, te pedirá que configures algunos ajustes básicos. Se recomienda la instalación estándar para comenzar.
- **SDK Manager:**
 - Ve a **Configure > SDK Manager** (Configurar > Administrador de SDK).
 - Aquí puedes descargar las herramientas de desarrollo de Android (SDK) necesarias para crear aplicaciones.
 - Selecciona la versión más reciente de Android y las herramientas que desees (mínimo el SDK Platform y las Android Emulator Tools).
 - Haz clic en **OK** para iniciar la descarga e instalación.

Apariencia y comportamiento:

- Puedes personalizar la apariencia y el comportamiento de Android Studio para que se ajuste a tus preferencias.
 - Ve a **Archivo > Configuración** (o **Android Studio > Preferencias** en macOS) para acceder a las opciones de configuración.
 - Puedes ajustar el tema del IDE, los atajos de teclado, las fuentes y otras configuraciones.

Plugins:

- Android Studio admite una amplia variedad de plugins que pueden extender su funcionalidad.
 - Ve a **Archivo > Configuraciones > Plugins** (o **Android Studio > Preferencias > Plugins** en macOS).
 - Explora los plugins disponibles en el Marketplace o busca plugins específicos que te ayuden con tareas de desarrollo particulares.
 - Instala los plugins que consideres necesarios.

Emulador y dispositivos:

- Configura el emulador de Android para probar tus aplicaciones en diferentes versiones y dispositivos virtuales.

- Ve a **Avd Manager** (Administrador de AVD) en la ventana **Tools** (Herramientas).
- Crea un dispositivo virtual con las especificaciones deseadas (versión de Android, memoria RAM, etc.).
- Si tienes un dispositivo Android físico, conéctalo a tu computadora y habilita la depuración USB para poder probar tus aplicaciones directamente en él.

SDK y herramientas de desarrollo:

- Asegúrate de mantener el SDK de Android y las herramientas de desarrollo actualizadas para aprovechar las últimas funcionalidades y correcciones de errores.
 - Ve a **Configure > SDK Manager** (Configurar > Administrador de SDK) nuevamente.
 - Busca actualizaciones disponibles e instálalas.

6. Actualizar las herramientas IDE y SDK en Android Studio

Actualizar las herramientas IDE y SDK en Android Studio es crucial para garantizar que dispones de las últimas funcionalidades, correcciones de errores y mejoras de rendimiento para desarrollar aplicaciones Android. A continuación, te explico cómo actualizar tanto el IDE como el SDK:

Actualizar Android Studio:

1. **Abre Android Studio.**
2. En la barra de menú, selecciona **Ayuda > Buscar actualizaciones** (o **Android Studio > Buscar actualizaciones** en macOS).
3. Android Studio comprobará si hay actualizaciones disponibles.
4. Si hay una nueva versión disponible, se te pedirá que la descargues e instales.
5. Sigue las instrucciones en pantalla para completar la instalación de la actualización.

Actualizar el SDK de Android:

1. **Abre Android Studio.**
2. En la barra de menú, selecciona **Archivo > Configuraciones > Administrador de SDK** (o **Android Studio > Preferencias > Administrador de SDK** en macOS).
3. En la pestaña **SDK Platforms**, selecciona la versión de Android que deseas actualizar.
4. Si hay una actualización disponible, aparecerá una casilla junto a la versión.
5. Marca la casilla para seleccionar la actualización y haz clic en **Aplicar**.
6. Android Studio descargará e instalará la actualización del SDK.

Actualizar las herramientas de plataforma:

1. **Abre Android Studio.**
2. En la barra de menú, selecciona **Archivo > Configuraciones > Administrador de SDK** (o **Android Studio > Preferencias > Administrador de SDK** en macOS).
3. En la pestaña **Herramientas de plataforma**, selecciona las herramientas que deseas actualizar.

4. Si hay una actualización disponible para alguna herramienta, aparecerá una casilla junto a ella.
5. Marca las casillas para seleccionar las actualizaciones y haz clic en **Aplicar**.
6. Android Studio descargará e instalará las actualizaciones de las herramientas de plataforma.

Beneficios de actualizar:

- **Nuevas funcionalidades:** Las actualizaciones suelen incluir nuevas funcionalidades y mejoras para el desarrollo de aplicaciones Android.
- **Correcciones de errores:** Las actualizaciones corrigen errores conocidos que pueden afectar tu flujo de trabajo o el rendimiento de tus aplicaciones.
- **Mejoras de rendimiento:** Las actualizaciones pueden optimizar el rendimiento de Android Studio y las herramientas de desarrollo.
- **Seguridad:** Las actualizaciones pueden incluir parches de seguridad para proteger tu sistema de desarrollo y tus aplicaciones.

Bibliografía

- [Sitio web de Android Developers](#)
- [Documentación de Android Studio](#)
- [Guías para desarrolladores de Android](#)
- <https://developer.android.com/studio/intro/migrate>
- <https://m.youtube.com/watch?v=c2GXtbmgA7Y>
- <https://developer.android.com/studio/projects/create-project>
- https://m.youtube.com/watch?v=VKbnStw_O74
- <https://betterprogramming.pub/how-to-use-git-in-android-studio-part-1-a8a554006aad>
- <https://git-scm.com/book/en/v2>