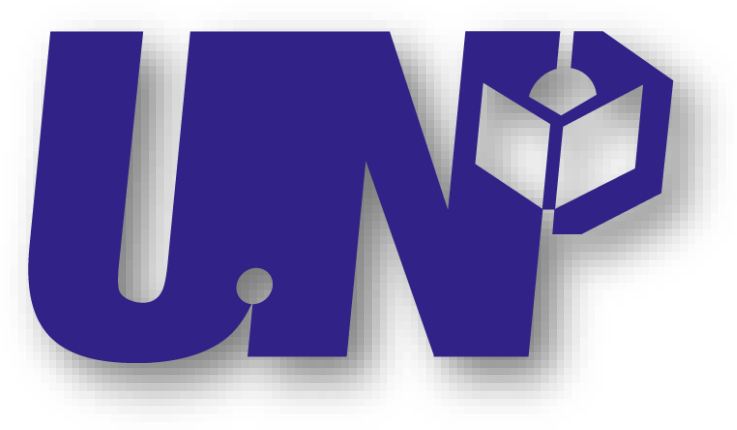

Universidad Nacional Politécnica

Programacion Movil

Estudiante:

- Adrian Eduardo Reynosa Flores

2010455

Docente:

- Ing. Luis Guido Calderon.

"Acumulado para el primer parcial - Investigacion"

Descripción general de proyecto de Android Studio

Android Studio es la plataforma esencial para desarrollar aplicaciones móviles en el ecosistema Android. En esta herramienta, los proyectos son la base de todo el trabajo, encapsulando todos los elementos necesarios para construir una aplicación. Desde código fuente hasta recursos y configuraciones de compilación, cada aspecto crucial de tu aplicación reside dentro de un proyecto.

Los proyectos se organizan en módulos, que actúan como unidades de funcionalidad discretas. Estos módulos permiten dividir la aplicación en partes manejables, lo que facilita la compilación, prueba y depuración independientes de cada uno. Puedes tener varios módulos en un proyecto, cada uno con sus propios archivos y configuraciones.

Android Studio ofrece diversos tipos de módulos para satisfacer las necesidades específicas de desarrollo. Por ejemplo, el módulo de app para Android es el contenedor principal para el código fuente de tu aplicación, recursos y configuraciones de nivel de app. Además, hay módulos de funciones que representan características modulares de la aplicación y módulos de biblioteca para el código reutilizable.

Cada tipo de módulo tiene su propósito y configuración específicos. Por ejemplo, los módulos de biblioteca pueden contener código Java o Kotlin que se puede compartir entre diferentes proyectos o módulos de la misma aplicación.

Dentro de cada módulo, la estructura de archivos se organiza en grupos como manifest, java, res, entre otros, que contienen diferentes tipos de archivos relacionados con la aplicación. Estos grupos facilitan la navegación y la gestión de los recursos de la aplicación.

Para ver la estructura de archivos real del proyecto, incluidos los archivos ocultos, puedes cambiar a la vista de proyecto en Android Studio. Esta vista te permite acceder a todos los archivos y directorios, incluidos los resultados de compilación y bibliotecas privadas.

Además, Android Studio proporciona herramientas para configurar la estructura del proyecto, como el diálogo de estructura del proyecto, que te permite establecer opciones de configuración específicas del proyecto y del módulo, como la versión de SDK, la firma de la aplicación y las dependencias de la biblioteca.

Create a project

Android Studio simplifica el proceso de creación de aplicaciones para Android en diversos dispositivos como teléfonos, tablets, televisores y dispositivos Wear. Aquí tienes un resumen sobre cómo crear un nuevo proyecto o importar uno existente:

1. Comenzar un Nuevo Proyecto:

- Si no tienes un proyecto abierto, haz clic en "Empezar un nuevo proyecto de Android Studio" en la pantalla de bienvenida.
- Si ya tienes un proyecto abierto, selecciona "Archivo > Nuevo > Nuevo Proyecto" desde el menú principal.

2. Seleccionar Tipo de Proyecto:

- En la pantalla "Nuevo Proyecto", verás categorías de factores de forma de dispositivos en el panel de Plantillas.
- Selecciona el tipo de proyecto que deseas crear (por ejemplo, teléfono, tablet).

3. Configurar Proyecto:

- Una vez que hayas seleccionado el tipo de proyecto, haz clic en "Siguiente".
- En la siguiente pantalla, configurarás algunas opciones importantes:
- Nombre del proyecto: Especifica un nombre para tu proyecto.
- Nombre del paquete: Define el nombre del paquete para tu proyecto, que se utilizará como el espacio de nombres del proyecto y el ID de la aplicación.
- Ubicación de Guardado: Elige dónde quieres almacenar localmente tu proyecto en tu sistema.
- Lenguaje: Selecciona si quieres usar Kotlin o Java para el código de tu proyecto. Ten en cuenta que no estás limitado a usar solo ese lenguaje en el proyecto.
- Nivel de API Mínimo: Decide el nivel de API mínimo que quieres que tu aplicación admita. Un nivel de API más bajo significa que tu aplicación puede ejecutarse en más dispositivos Android, pero también limita las funciones modernas que puedes utilizar.
- Si necesitas ayuda para elegir el nivel de API, puedes hacer clic en "Ayúdame a elegir". Esto te mostrará una pantalla con la distribución acumulativa para el nivel de API seleccionado y te permitirá ver el impacto de usar diferentes niveles de API.

- Por defecto, tu proyecto estará configurado para utilizar las bibliotecas AndroidX, que reemplazan a las bibliotecas de soporte de Android. Puedes optar por utilizar las bibliotecas de soporte heredadas, pero no se recomienda, ya que no reciben soporte.
- Una vez que hayas configurado todas las opciones según tus necesidades, haz clic en "Finalizar".

Con estos pasos, habrás creado y configurado tu proyecto en Android Studio, y estarás listo para comenzar a desarrollar tu aplicación para Android. Si más adelante deseas agregar soporte para otro tipo de dispositivo o compartir código y recursos entre módulos, puedes hacerlo mediante la adición de un nuevo módulo o la creación de una biblioteca de Android.

Como migrar a Android Studio

Para migrar tus proyectos a Android Studio, necesitas adaptarte a una nueva estructura de proyecto, funcionalidades del IDE y un sistema de compilación basado en Gradle. Si ya estás utilizando IntelliJ y tu proyecto emplea Gradle, puedes abrirlo directamente en Android Studio. Sin embargo, si tu proyecto de IntelliJ no utiliza Gradle, necesitas prepararlo manualmente antes de la importación. A continuación, se detallan los pasos básicos para realizar la migración desde IntelliJ:

1. Organización del proyecto y módulos: Android Studio organiza los proyectos en unidades llamadas módulos, que permiten una división funcional del proyecto. Cada proyecto puede contener varios módulos que se pueden compilar y probar de forma independiente.

2. Sistema de compilación basado en Gradle: Android Studio utiliza Gradle como su sistema de compilación, lo que ofrece ventajas como compatibilidad con bibliotecas binarias, variantes de compilación y fácil configuración. Es importante comprender cómo funciona Gradle y cómo configurar tu compilación según las necesidades de tu proyecto.

3. Dependencias: Android Studio utiliza declaraciones de dependencias de Gradle y dependencias de Maven para gestionar las bibliotecas en tu proyecto. Debes asegurarte de declarar correctamente las dependencias de tus bibliotecas para que el sistema de compilación las gestione adecuadamente.

Si estás migrando desde IntelliJ, puedes seguir estos pasos:

- Si ya utilizas Gradle en tu proyecto de IntelliJ, simplemente importa tu proyecto a Android Studio.

- Si tu proyecto de IntelliJ no utiliza Gradle, puedes optar por crear un nuevo proyecto vacío en Android Studio y luego copiar tu código fuente, o bien, crear un archivo de compilación de Gradle personalizado para tu proyecto existente.

Para migrar creando un nuevo proyecto vacío:

1. Abre Android Studio y crea un nuevo proyecto vacío.

2. Copia tus archivos fuente y recursos desde tu proyecto de IntelliJ al nuevo proyecto en Android Studio.

3. Ajusta las configuraciones necesarias en el diálogo "Project Structure" y agrega las dependencias de Gradle según sea necesario.

4. Realiza pruebas de compilación para asegurarte de que todo funcione correctamente.

Si optas por migrar creando un archivo de compilación de Gradle personalizado:

1. Crea un archivo build.gradle o build.gradle.kts en el directorio de tu proyecto.

2. Configura adecuadamente los repositorios y ajusta los conjuntos de orígenes según sea necesario.

3. Identifica y declara las dependencias de tus bibliotecas en el archivo de compilación de Gradle.

4. Borra los archivos de configuración de IntelliJ y luego importa tu proyecto a Android Studio.

Una vez completada la migración, es importante ajustar la configuración adicional según sea necesario, como la configuración del control de versión y la firma de apps. Además, debes tener en cuenta las actualizaciones de software y el ajuste del tamaño máximo de montón de Android Studio para mejorar el rendimiento según tus necesidades específicas.

Version de control basic

Android Studio puedes gestionar el control de versiones de tu aplicación utilizando varios sistemas, como Git, GitHub, CVS, Mercurial, Subversion y Google Cloud Source Repositories. Aquí te explico cómo habilitar el control de versiones en Android Studio:

1. Menú de VCS de Android Studio: Después de importar tu aplicación en Android Studio, puedes habilitar el control de versiones desde el menú de VCS (Sistema de Control de Versiones) de Android Studio.

2. Habilitar la integración del control de versiones: Selecciona la opción "Enable Version Control Integration" (Habilitar Integración de Control de Versiones) desde el menú de VCS.

3. Seleccionar el sistema de control de versiones: A continuación, elige el sistema de control de versiones que deseas utilizar, como Git, GitHub, CVS, Mercurial, Subversion o Google Cloud Source Repositories.

4. Confirmar la selección: Haz clic en "OK" para confirmar la asociación del sistema de control de versiones con el proyecto.

Una vez completados estos pasos, Android Studio mostrará opciones adicionales en el menú de VCS según el sistema de control de versiones seleccionado. Desde aquí, puedes realizar operaciones de control de versiones, como commit, push, pull, merge, entre otras.

Es importante tener en cuenta que, dependiendo del sistema de control de versiones que elijas, es posible que necesites configurar credenciales y otros ajustes específicos para comenzar a utilizarlo en tu proyecto. Por ejemplo, si eliges Git o GitHub, deberás configurar tu cuenta y clonar el repositorio de tu proyecto. Si eliges Google Cloud Source Repositories, deberás configurar la autenticación con tu cuenta de Google y crear un repositorio en la nube.

El control de versiones es una práctica fundamental en el desarrollo de software, ya que te permite realizar un seguimiento de los cambios en tu código, colaborar con otros desarrolladores y mantener un historial de versiones de tu aplicación.

Configurar Android Studio

Android Studio simplifica la configuración inicial mediante asistentes y plantillas que verifican los requisitos del sistema, como el JDK y la memoria RAM disponible, y establecen ajustes predeterminados, como la emulación optimizada del AVD y las imágenes del sistema actualizadas. Además de estas configuraciones automáticas, Android Studio ofrece la posibilidad de personalizar aún más mediante dos archivos de configuración: `studio.vmoptions` para ajustes de la JVM y `idea.properties` para personalizar funciones del IDE.

El archivo `studio.vmoptions` permite ajustar parámetros como el tamaño de montón y la caché de la JVM de Android Studio, mientras que

`idea.properties` permite personalizar propiedades del IDE, como la ubicación de la carpeta de complementos y el tamaño máximo de archivo admitido. Estos archivos se pueden encontrar y editar desde el menú de ayuda de Android Studio, lo que facilita la personalización según las necesidades específicas del usuario.

Para aquellos que buscan información detallada sobre la configuración y el uso del emulador y el dispositivo, Android Studio ofrece documentación específica sobre la creación y administración de dispositivos virtuales, la ejecución de aplicaciones en dispositivos de hardware y la instalación de controladores USB de OEM, entre otros temas.

Además de los archivos de configuración, Android Studio permite establecer variables de entorno para señalar archivos de anulación específicos en otros lugares, lo que brinda flexibilidad adicional para la configuración del entorno de desarrollo. Estas variables, como `STUDIO_VM_OPTIONS` y `STUDIO_PROPERTIES`, permiten ajustar la configuración de manera más granular y adaptada a las necesidades individuales del usuario.

Para optimizar aún más el rendimiento de Android Studio, especialmente en equipos con recursos limitados, se pueden realizar ajustes específicos. Por ejemplo, reducir el tamaño máximo de montón disponible para Android Studio a 512 MB puede ser beneficioso en equipos con menos especificaciones recomendadas. Además, actualizar Gradle y el complemento de Android para Gradle a las versiones más recientes puede aprovechar las mejoras de rendimiento más recientes, mientras que habilitar el modo de ahorro de energía puede desactivar operaciones que consumen mucha memoria en segundo plano, como la compilación automática incremental.

Otras opciones para mejorar el rendimiento incluyen la reducción de las comprobaciones de lint innecesarias, realizando depuraciones en dispositivos físicos en lugar de emuladores para reducir el consumo de memoria y limitar las dependencias a los Servicios de Google Play necesarios para reducir el uso de memoria. Al ajustar adecuadamente estas configuraciones, los usuarios pueden lograr un rendimiento óptimo de Android Studio en una variedad de configuraciones de hardware y red.

Update the IDE and SDK tools

Para mantener actualizado tu entorno de desarrollo, es importante actualizar tanto el IDE de Android Studio como las herramientas del SDK de forma regular. Si instalaste Android Studio utilizando JetBrains Toolbox, este se

encargará de gestionar las actualizaciones automáticamente. JetBrains Toolbox te permite instalar versiones canary, beta y estable de Android Studio en paralelo, y también te permite retroceder a versiones anteriores si es necesario. Cuando haya una actualización disponible, Toolbox te mostrará una notificación, como se muestra en la figura 1.

Si instalaste Android Studio manualmente, el propio IDE te notificará cuando haya una actualización disponible. Para buscar actualizaciones manualmente, puedes hacer clic en File > Settings > Appearance & Behavior > System Settings > Updates (en macOS, Android Studio > Check for Updates). Las actualizaciones están disponibles en los siguientes canales: Canary, Beta y Estable, cada uno ofreciendo diferentes niveles de estabilidad y nuevas características.

Es importante mantener también actualizadas las herramientas del SDK utilizando el Administrador de SDK de Android. Este te permite descargar las herramientas, plataformas y otros componentes necesarios para el desarrollo de tus aplicaciones. Una vez descargadas, puedes encontrar cada paquete en el directorio indicado como la ubicación del SDK de Android. Para abrir el Administrador de SDK desde Android Studio, simplemente haz clic en Tools > SDK Manager o en el icono del SDK Manager en la barra de herramientas. Si no estás utilizando Android Studio, puedes descargar las herramientas utilizando la herramienta de línea de comandos sdkmanager.