

Actividad | 2 | Pantalla de registro.

Desarrollo de aplicaciones móviles 1.

Ingeniería en Desarrollo de
Software.



academiaglobal

TUTOR: Humberto Jesús Ortega Vázquez

ALUMNO: Carlos Ariel Nicolini

FECHA: 01/12/2024

Índice

Introducción	3
Descripción	4
Justificación	5
Desarrollo.....	6
Conclusión.....	20
Referencias.....	21

Introducción

Android Studio permite la integración de características y funciones bastantes positivas para las aplicaciones que, con el tiempo, se perfeccionan. De esta forma, tenemos lo siguiente:

- El sistema de compilación es flexible, además de ser compatible con Gradle, la cual permite la automatización de compilaciones de forma flexible y con gran rendimiento. Groovy y Kotlin DSL son los lenguajes utilizados para los scripts de compilación.
- La intención de este entorno es la permitir al usuario trabajar de forma fluida y con una gran cantidad de funciones prácticas y útiles.
- Contiene plantillas de compilación que te ayudan a otorgar funciones comunes de otras apps de forma más rápida, además de importar códigos de muestra.
- Modificar fragmentos de códigos y recursos de una app sin necesidad de que esta se reinicie.
- Proporciona compatibilidad con servicios en la nube tal como Google Cloud Plataform.
- Compatibilidad con lenguaje como NDK y C++.

Una de las maravillas de lenguaje de programación Kotlin, es que permite una interoperabilidad natural con Java, pudiendo incluso desarrollar código para proyectos utilizando ambos lenguajes a la vez sin ningún problema.

Descripción

Contextualización:

La unidad de negocio enfocada a servicios bancarios solicita una aplicación móvil. Por lo que, requieren que un ingeniero en desarrollo de software atienda a su necesidad.

Actividad:

Configurar un proyecto, instalar los paquetes de la plataforma e instalar las herramientas SDK. Crear un dispositivo virtual, configurarlo y dejarlo funcional.

Como nos comentó el profesor, en este ejercicio aprenderemos sobre como configurar un proyecto nuevo, configurar las herramientas SDK, configurar un emulador con sus opciones y hacerlo funcionar, que arranque demostrando que esta funcional con las mínimas especificaciones.

En toda esta materia conoceremos el software, a generar un nuevo proyecto y las opciones donde se encuentran, ya que tiene un abanico gigante de posibilidades para elegir a la hora de realizar nuestro proyecto.

Espero que el programa presentado sea del agrado y cumpla con lo solicitado. Me divertí mucho realizándolo. Gracias por todo el apoyo recibido y las explicaciones del profesor.

Justificación

En esta actividad aprenderemos más sobre este potente software llamado Android Studio, en esta oportunidad en un proyecto configuraremos e instalaremos las SDK Platforms para que nuestro proyecto pueda funcionar en una amplia gama de versiones de Android y dispositivos (instalaremos los recomendados por el profesor en la clase). También instalaremos las herramientas de SDK tools que son necesarias para probar y realizar cambios de mejoras en las aplicaciones de Android. Se nos pide además crear y configurar un dispositivo virtual que es donde se realizarán las pruebas y se verá el proyecto funcionando (se debe configurar y dejarlo funcionando de manera normal).

En esta materia seguimos aprendiendo más sobre este software el cual es muy impresionante y tiene muchísimo potencial para poder crear proyectos y programas para Android con multitud de herramientas y oportunidades, pero también este programa necesita muchísimos recursos a nivel hardware.

Este trabajo fue subido al siguiente enlace de GitHub

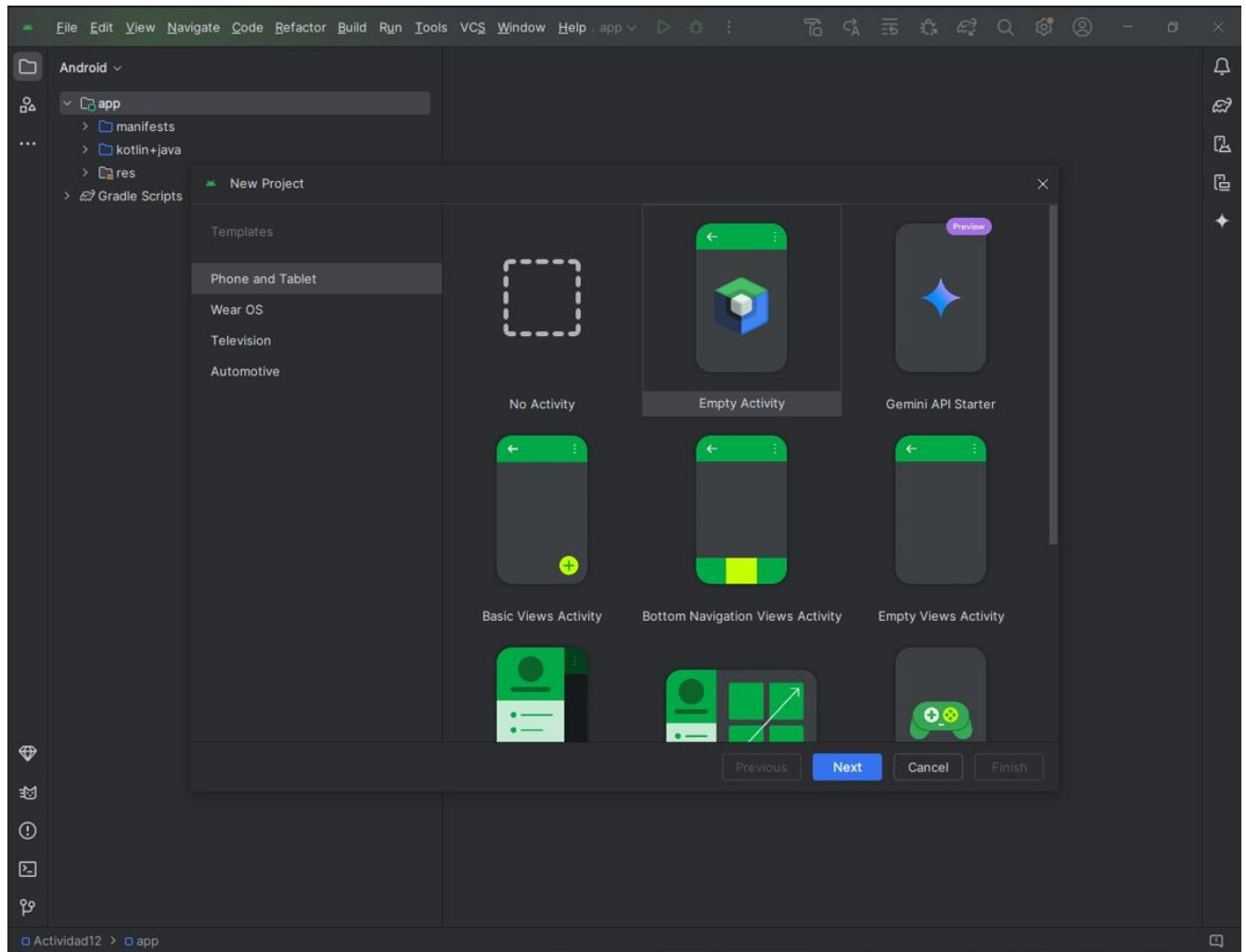
<https://github.com/CarlosNico/Desarrollodeaplicacionesmoviles1>

Desarrollo

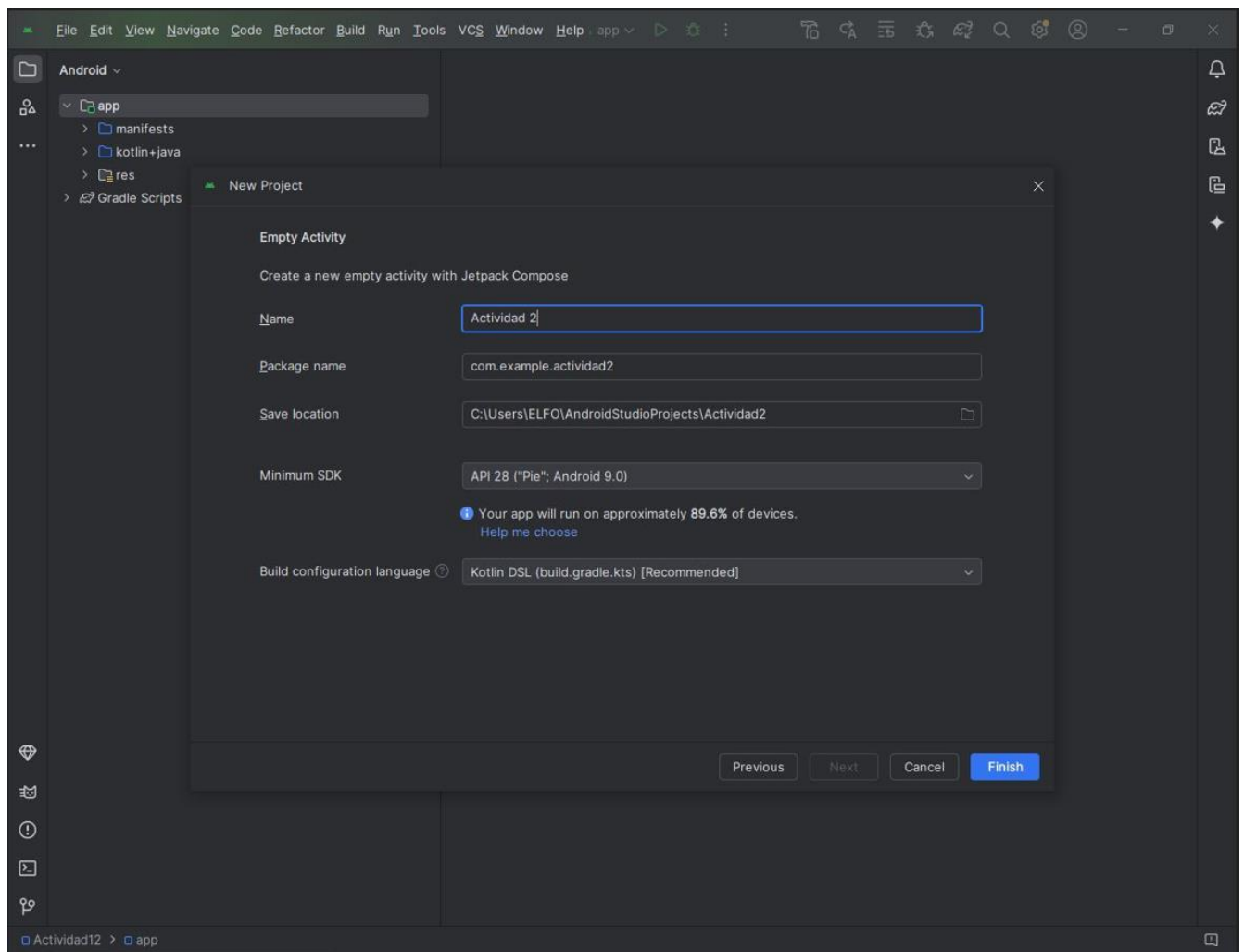
Configuración de nuevo proyecto

En el trabajo se nos pide realizar la configuración de un nuevo proyecto además de la creación del emulador.

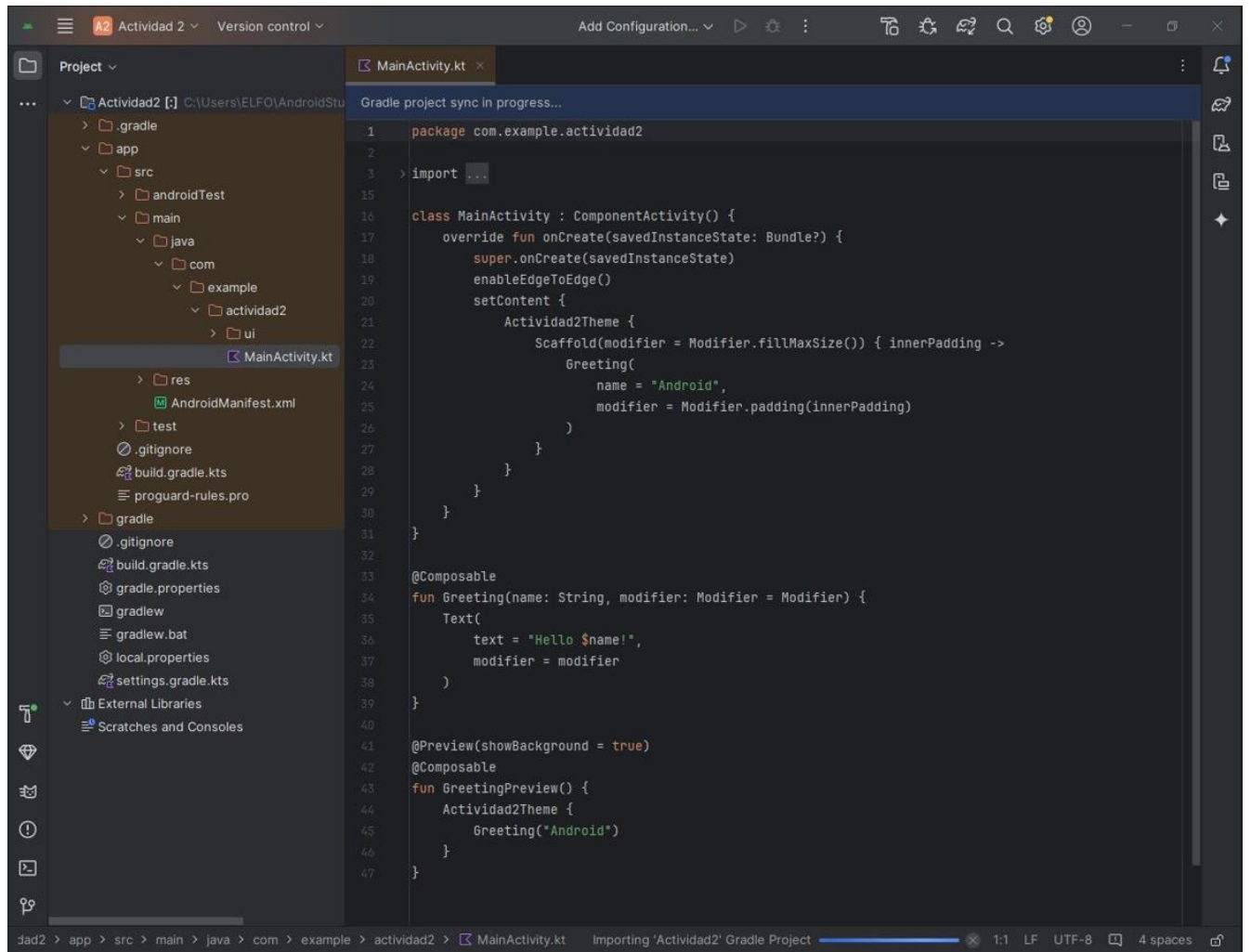
Empezamos creando un proyecto nuevo de la siguiente manera (como se vio en el trabajo anterior).



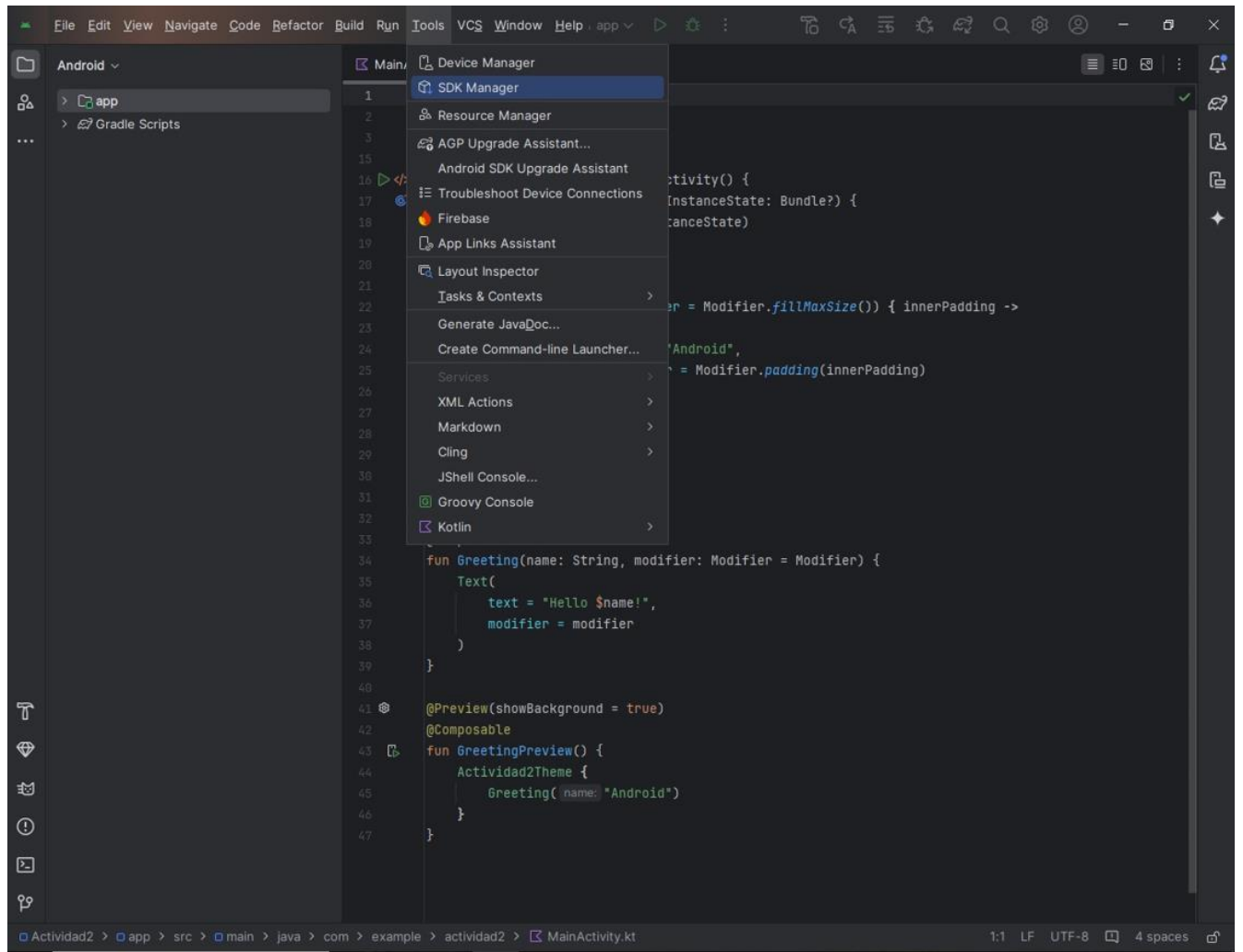
En la siguiente pantalla deberemos poner el nombre del proyecto, el package name (lo dejaremos de default), también la ruta donde se guardara el proyecto, en minimum SDK (esta opción se refiere a la versión mínima de Android API level que tu aplicación será capaz de soportar, versiones inferiores no podrían instalar ni ejecutar la aplicación) elegiremos API 28. En la opción Build configuration language (lenguaje utilizado para definir y gestionar la configuración del proyecto) elegiremos Kotlin DLS que es la recomendada y daremos en finish.



Ya tenemos el nuevo proyecto abierto.



En el menú superior buscaremos la opción de Tools y elegiremos la opción de SDK Manager.

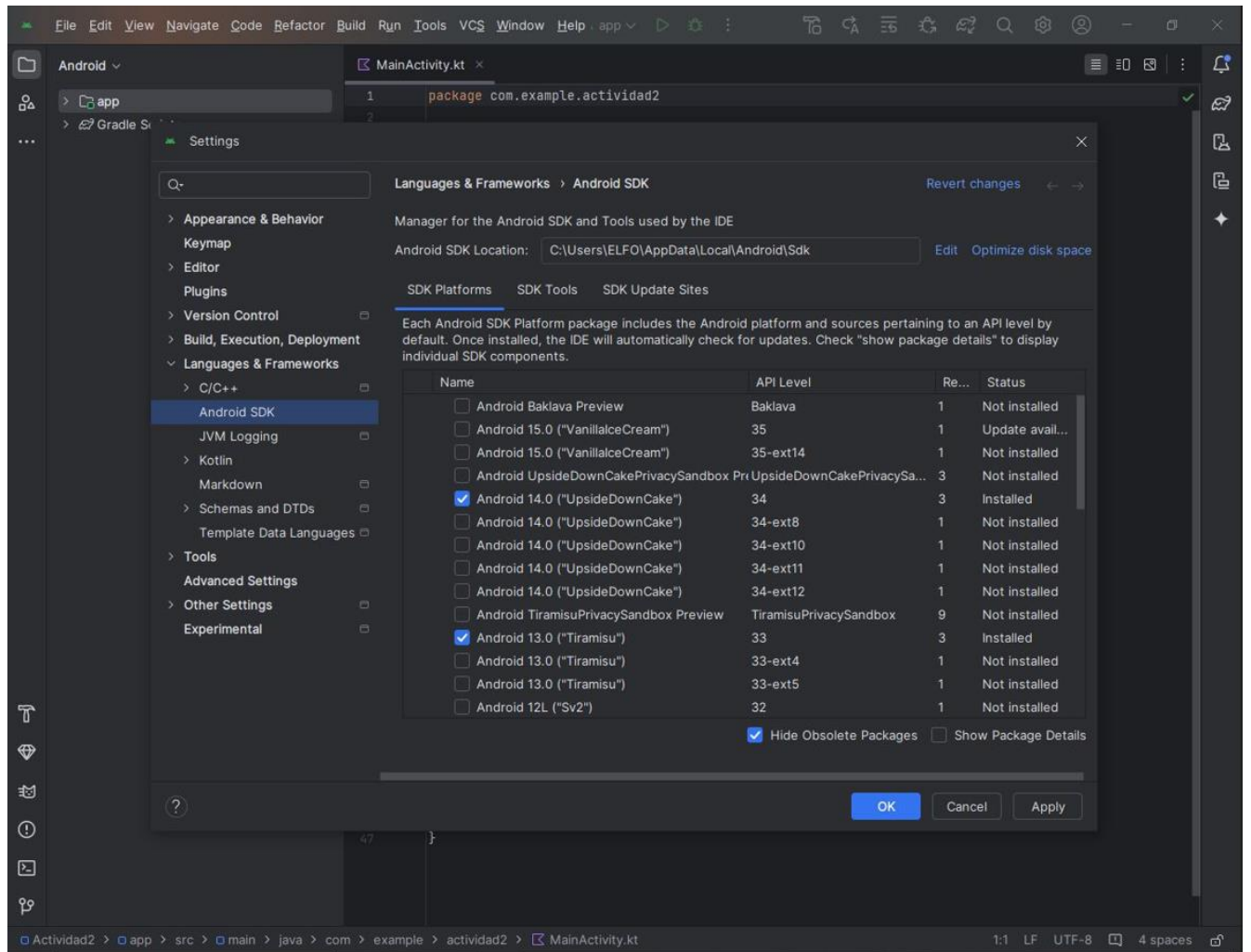


Se nos abrirá la opción, en donde debemos seleccionar plataformas y herramientas.

En el apartado de SDK Platforms (donde se descargan y gestionan las diferentes versiones de SDK (Software Development Kit) necesarias para desarrollar aplicaciones para diferentes versiones del sistema operativo Android). En la cual elegiremos las siguientes:

- Android 14.0 (34)
- Android 13.0 (33)
- Android 11.0 (30)
- Android 10.0 (29)
- Android 9.0 (28)
- Android 8.1 (27)
- Android 8.0 (26)
- Android 7.1.1 (25)
- Android 7.0 (24)

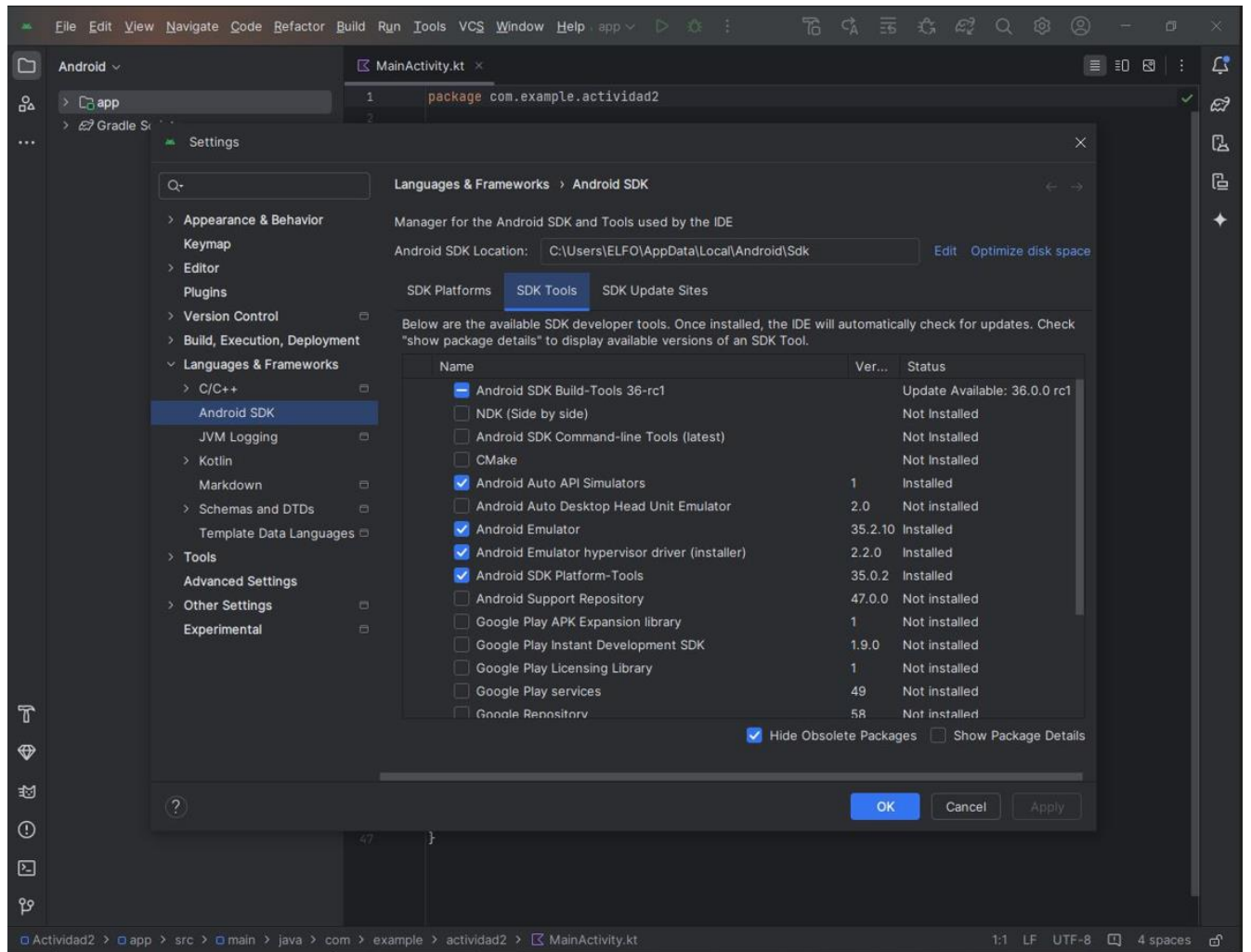
Una vez elegidas daremos en apply, tras lo cual saldrá un cartel informativo donde daremos ok a las versiones a instalar, esperar que se instalen (y revisar los posibles mensajes de error si es que se presentan para su remediación). Al terminar tendremos esos paquetes instalados.



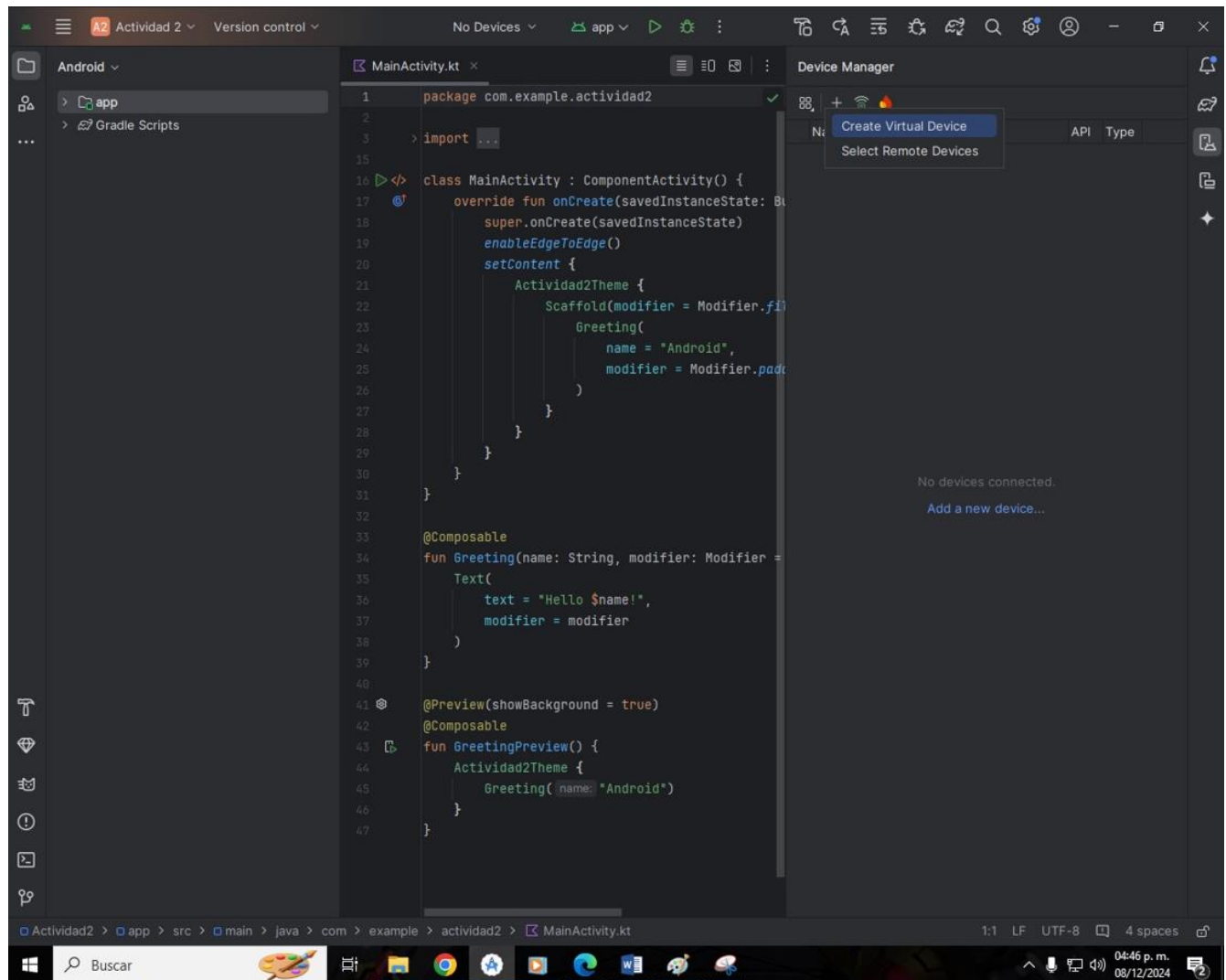
A continuación elegiremos la pestaña de SDK tools(es una sección clave dentro de A>Android Studio donde puedes administrar las herramientas complementarias del SDK necesarias para desarrollar, depurar, probar y empaquetar aplicaciones Android. Estas herramientas no están vinculadas directamente a una versión específica de Android, sino que ofrecen funcionalidades adicionales para el desarrollo).En la cual elegiremos las siguientes:

- Android Auto API Simulators (simuladores para probar aplicaciones compatibles con Android Studio).
- Android Emulator (Simulador que permite ejecutar y probar aplicaciones en dispositivos virtuales (AVDs).
- Android Emulator hypervisor driver (installer) Componente diseñado para mejorar el rendimiento de los emuladores en máquinas con procesadores AMD o Intel que no soportan completamente HAXM.
- Android SDK Platform-tools (Incluye herramientas como adb (Android Debug Bridge) para depurar aplicaciones y manejar dispositivos conectados.
- Intel x86 Emulator Accelerator (HAXM installer) Mejora el rendimiento de los emuladores en sistemas con procesadores Intel mediante virtualización.

Una vez elegidas daremos en apply, tras lo cual saldrá un cartel informativo donde daremos ok a las versiones a instalar, esperar que se instalen (se deben aceptar licencias). Al terminar tendremos esos SDK instalados.

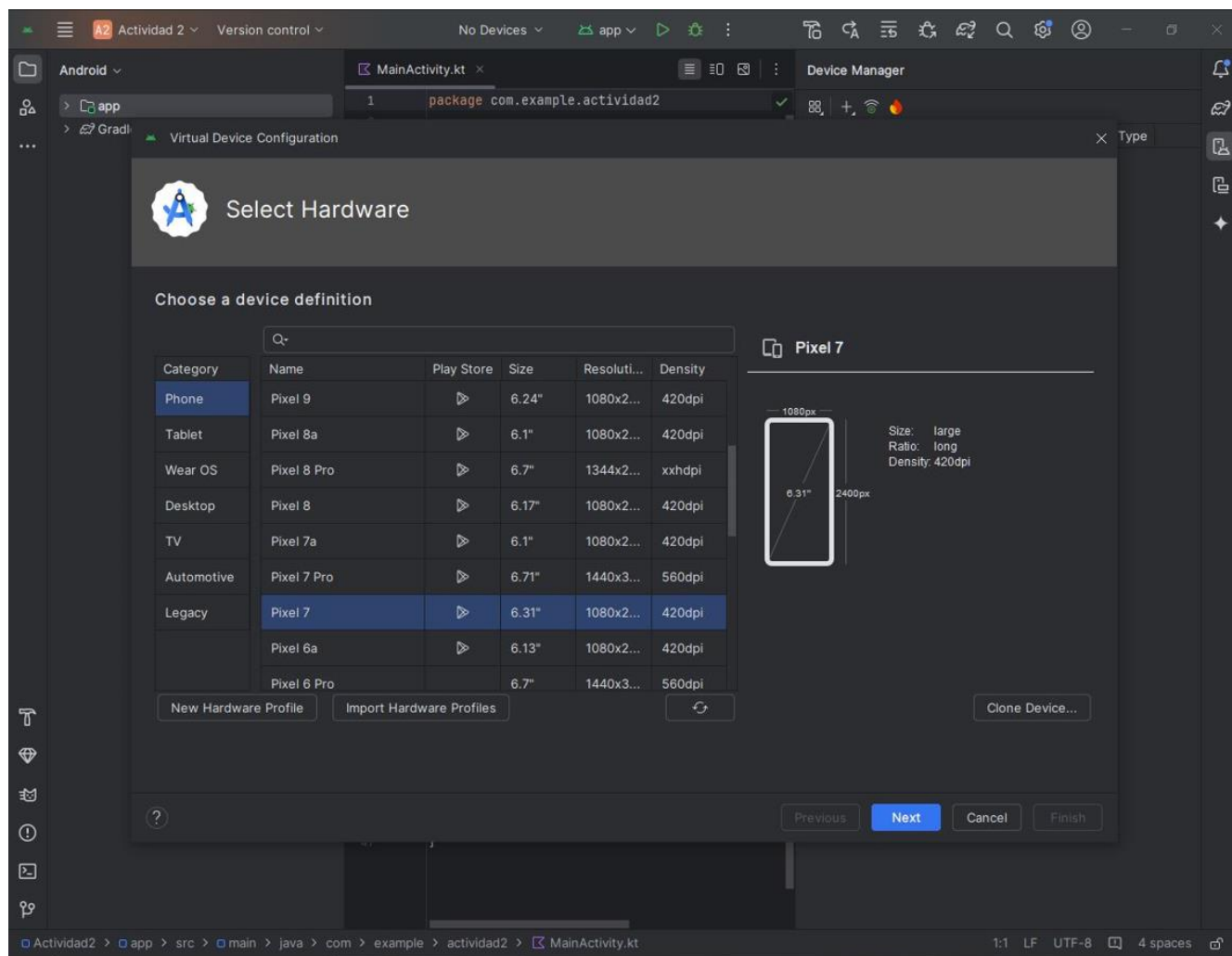


A continuación crearemos un dispositivo virtual para la emulación. Elegiremos la opción de Device Manager (el icono se encuentra en la parte derecha), tras lo cual elegiremos Create Virtual Device.

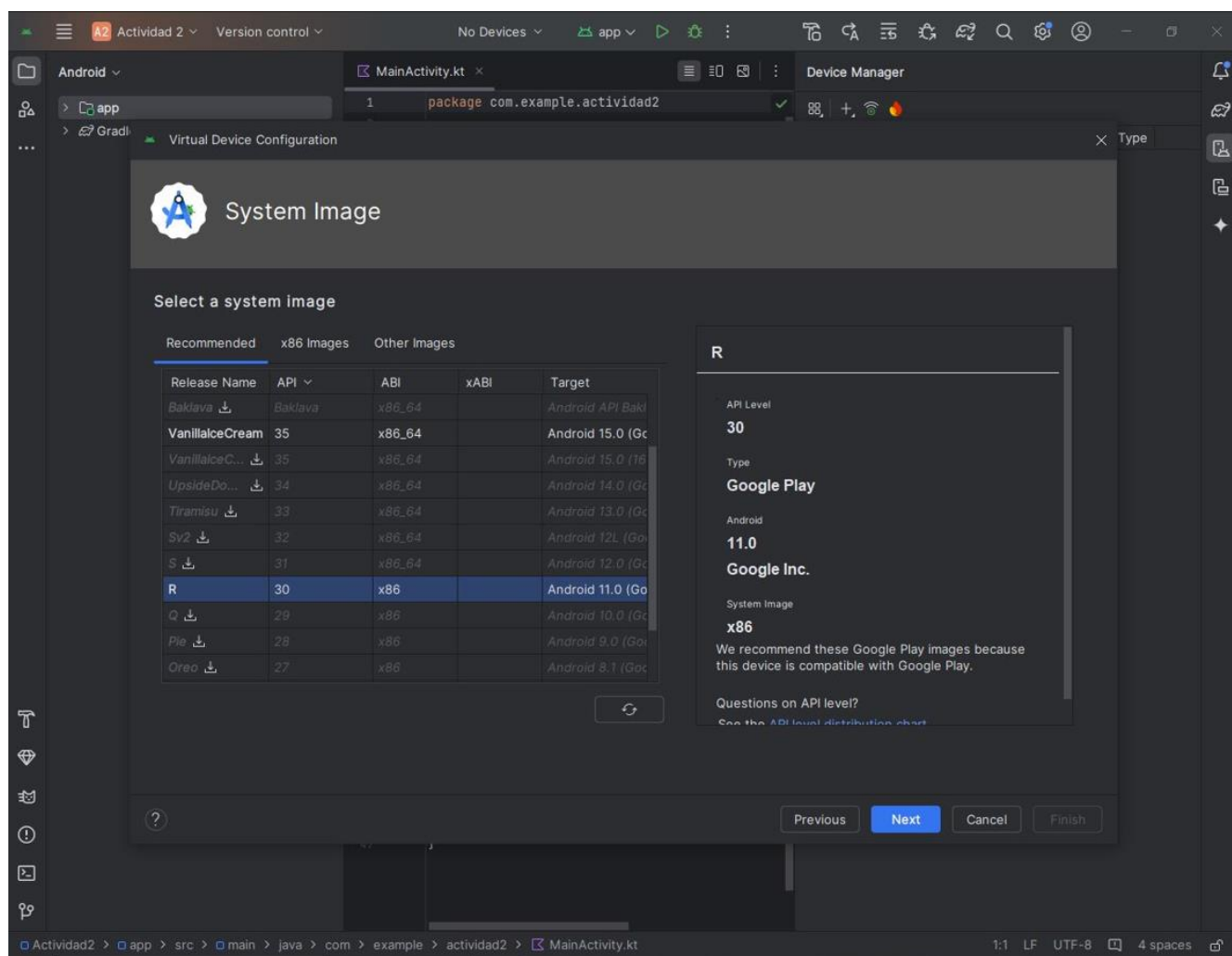


Configuración de emulador con Device Manager

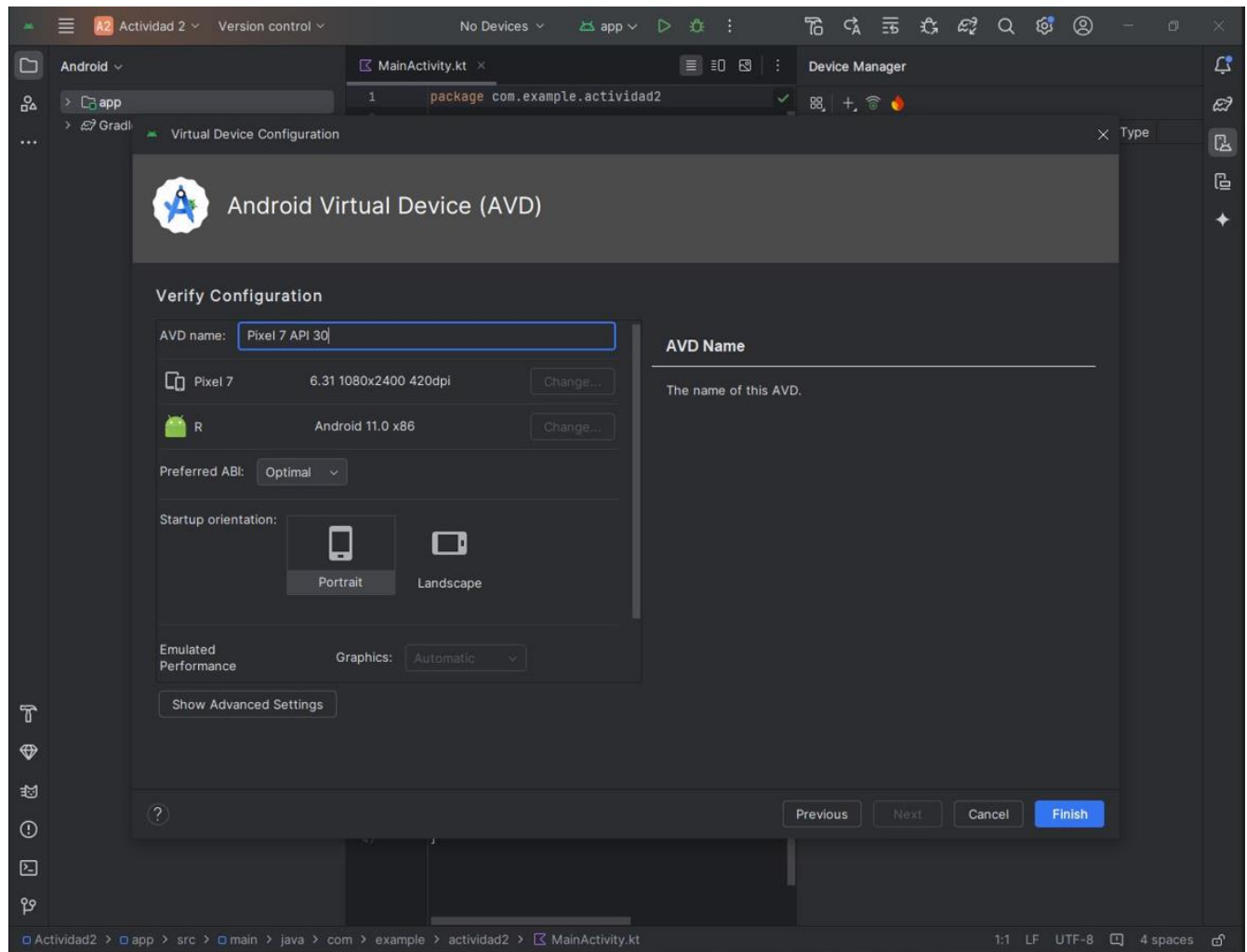
En la pantalla de creación de dispositivo en la cual nos permite elegir las especificaciones del dispositivo a emular, como el tipo de pantalla, RAM, procesador y otras características. Elegiremos un teléfono con un pixel de 7 y presionaremos en Next.



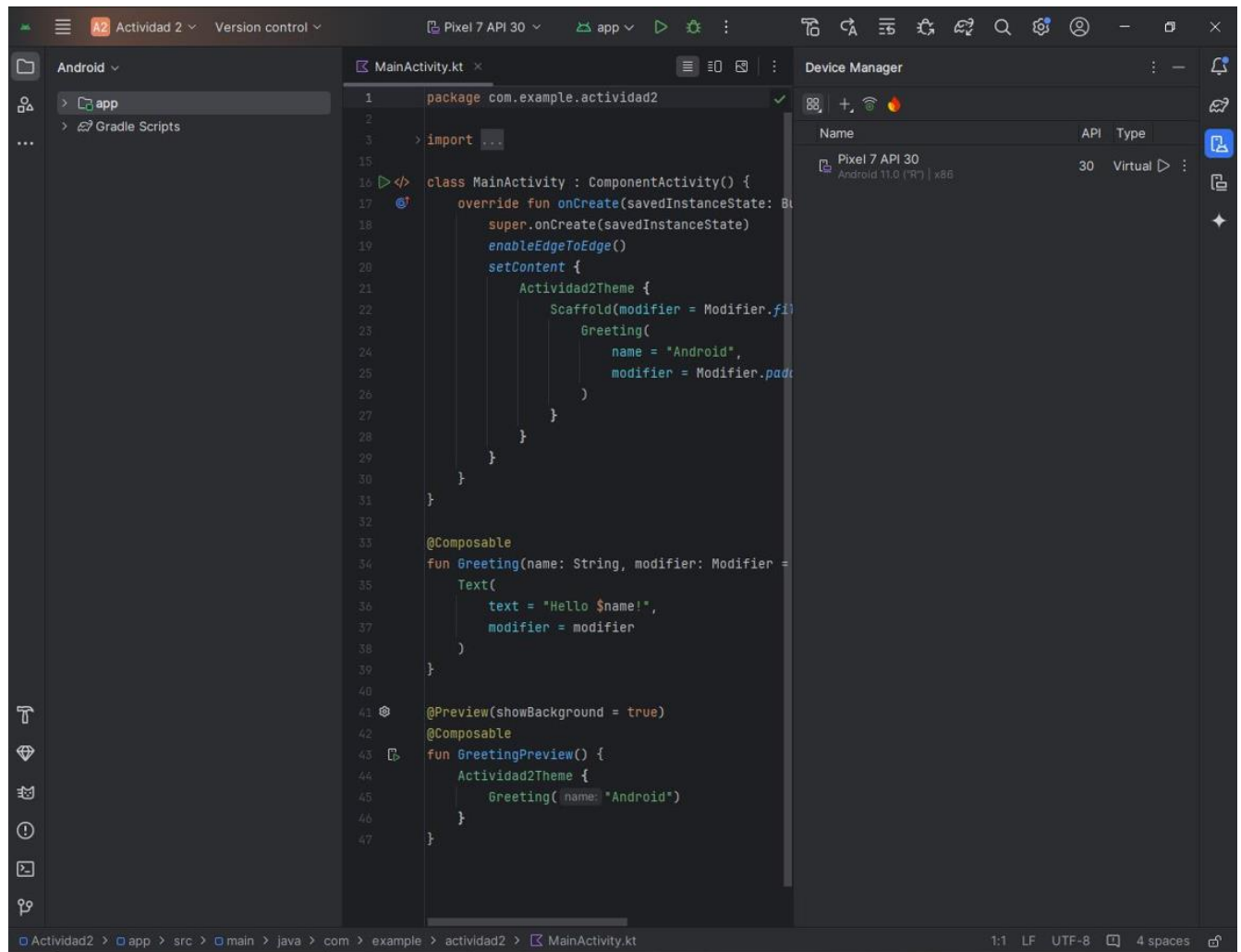
En la opción de system image en donde nos permitirá seleccionar la versión de Andorid que ejecutara el emulador y es clave para probarla en diferentes niveles de API y características del sistema. En esta opción elegiremos R y le daremos Next (si dicho paquete no está instalado, procederá a instalarse lo que puede tardar unos minutos).



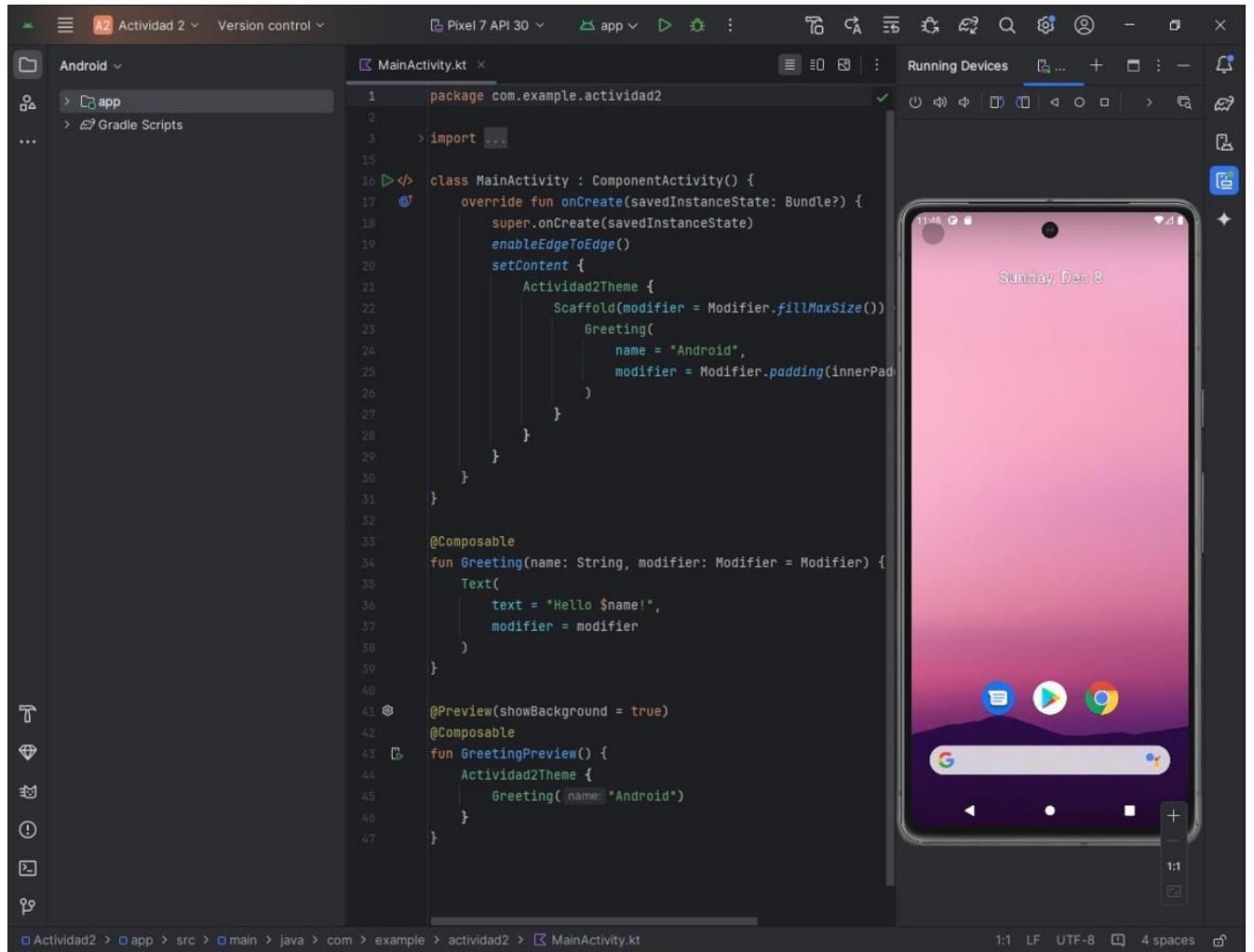
En la opción de Android Virtual Device (AVD) donde se personaliza el nombre del dispositivo, se selecciona la orientación, el tamaño reducido o completo de pantalla y además donde se pueden configurar la RAM y CPU. En esta opción lo dejaremos como esta y daremos Next.



Una vez realizado todos los pasos anteriores ya tendremos un dispositivo virtual configurado, tras lo cual le daremos en el icono de PLAY (triángulito acostado) para arrancar el emulador.



Y tendremos nuestro dispositivo virtual (teléfono) funcionando como se muestra en la pantalla.



Conclusión

En esta tarea aprendimos algunas de las opciones que trae este gran programa, sobre sus características y algunos apartados muy importantes que se deben tener en cuenta y configurar a la hora de empezar nuestro proyecto, como la elección de las correctas plataformas donde funcionara nuestro proyecto, lo cual nos permitirá si lo configuramos bien que nuestro software funcione en muchos dispositivos y versiones de Android, también de las herramientas necesarias para que se puede programar, depurar y probar (entre otras opciones) además de crear dispositivos virtuales que emulan un aparato (en este caso un teléfono con sistema Operativo Android) para todas las pruebas lo cual es una herramienta muy importante, porque nos permitirá realizar las pruebas de manera virtual y no tener que estar realizando pruebas iniciales en dispositivos físicos, lo cual nos puede ahorrar muchísimo tiempo en las pruebas iniciales.

Esta herramienta me gustó mucho poder seguir interactuando con ella, conocerla y poder empezar a trabajar en ella.

Muchas gracias profesor por toda su guía y apoyo en esta materia y con este software que es nuevo para mí pero que es impresionante y me puede ayudar muchísimo.

Este trabajo fue subido al siguiente enlace de GitHub

<https://github.com/CarlosNico/Desarrollodeaplicacionesmoviles1>

Referencias

[https://talently.tech/blog/que-es-android-](https://talently.tech/blog/que-es-android-studio/#:~:text=Esta%20plataforma%20te%20permite%20desarrollar,prueba%20con%20marcos%20de%20trabajo.)

[studio/#:~:text=Esta%20plataforma%20te%20permite%20desarrollar,prueba%20con%20marcos%20de%20trabajo.](https://talently.tech/blog/que-es-android-studio/#:~:text=Esta%20plataforma%20te%20permite%20desarrollar,prueba%20con%20marcos%20de%20trabajo.)