



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

título del TFG



Presentado por Pablo Ahíta del Barrio
en Universidad de Burgos — 4 de marzo
de 2023

Tutor: Pedro Luis Sánchez Ortega



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Pablo Ahita del Barrio, con DNI 71566290L, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 4 de marzo de 2023

Vº. Bº. del Tutor:

D. nombre tutor

Vº. Bº. del co-tutor:

D. nombre co-tutor

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vii
Introducción	1
Objetivos del proyecto	3
Conceptos teóricos	5
3.1. Secciones	5
3.2. Referencias	5
3.3. Imágenes	6
3.4. Listas de items	6
3.5. Tablas	7
Técnicas y herramientas	9
4.1. <i>Azure</i>	9
4.2. <i>Microsoft Visual Studio Code</i>	9
4.3. <i>Android Studio</i>	9
Aspectos relevantes del desarrollo del proyecto	41
5.1. Aspectos relevantes	42
Trabajos relacionados	45
Conclusiones y Líneas de trabajo futuras	49

Índice de figuras

3.1. Autómata para una expresión vacía	6
4.1. Página de descarga del instalador de Android Studio	11
4.2. Sección de términos y condiciones anterior a la descarga del instalador de Android Studio	11
4.3. El fichero de instalación ya ha sido descargado	12
4.4. Mensaje de bienvenida del instalador de Android Studio	13
4.5. Selección de componentes a instalar en el instalador de Android Studio	14
4.6. Selección del directorio de instalación	15
4.7. Selección del directorio de creación de accesos directos en el menú de inicio	16
4.8. Inicio de instalación de Android Studio	16
4.9. Finalización de la instalación de Android Studio	17
4.10. Mensaje de finalización de la instalación	17
4.11. Ventana emergente de importación de configuración	18
4.12. Mensaje de bienvenida del asistente de configuración inicial de Android Studio	19
4.13. Selección del tipo de instalación de los componentes de Android Studio	20
4.14. Selección del directorio predeterminado del JDK	20
4.15. Selección del diseño de la interfaz de Android Studio	21
4.16. Selección de componentes SDK a instalar en Android Studio	22
4.17. Advertencia por falta de componentes a instalar	22
4.18. Configuración de la memoria RAM utilizada por el emulador de Android	23
4.19. Verificación de componentes a instalar en Android Studio	23
4.20. Licencia de software del paquete SDK	24

4.21. Licencia de software del acelerador de emulación de Intel	24
4.22. Instalación de los componentes en la configuración inicial	25
4.23. Finalización del asistente de instalación de componentes	25
4.24. Pantalla de selección de proyecto	26
4.25. Pantalla de selección de actividad principal	27
4.26. Pantalla de configuración de la actividad	27
4.27. Desarrollo del proceso de instalación del soporte JDK	28
4.28. Finalización del proceso de instalación del soporte JDK	28
4.29. Interfaz de Android Studio con el proyecto Android recién creado	29
4.30. Acceso a las opciones de Android Studio	29
4.31. Pantalla de control de cuentas de GitHub	30
4.32. Pantalla de tokens de usuario antes de ir a la pantalla de generación de tokens	30
4.33. Pantalla de generación de tokens con los permisos de usuario .	31
4.34. Siguientes permisos de usuario	31
4.35. Últimos permisos de usuario	32
4.36. Pantalla de tokens de usuario con el token de usuario generado .	32
4.37. Pantalla de control de cuentas de GitHub lista para añadir usuario	33
4.38. Añadir token de usuario de GitHub en Android Studio	33
4.39. Cuenta de GitHub recién añadida	34
4.40. Acceso al administrador de dispositivo mediante el menú desplegable de dispositivos	34
4.41. Administrador de dispositivos	35
4.42. Selección del nuevo dispositivo virtual	35
4.43. Selección de sistema operativo para el nuevo dispositivo virtual	36
4.44. Proceso de instalación del sistema operativo y de configuración del dispositivo virtual	37
4.45. Finalización del proceso de instalación del sistema operativo y de configuración del dispositivo virtual	37
4.46. Finalización del asistente de creación del dispositivo y comprobación de la configuración	38
4.47. Acceso a la configuración del nuevo commit	39
4.48. Configuración del nuevo commit	39
4.49. Soluciones a warnings durante el proceso de commit	40
4.50. Menú de publicación de cambios realizados	40
4.51. Commit y push realizados satisfactoriamente	40
5.1. Diagrama de casos de uso de la aplicación	42
5.2. Diagrama de clases de la aplicación	44

Índice de tablas

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto 7

Introducción

Descripción del contenido del trabajo y del estrucutra de la memoria y del resto de materiales entregados.

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

Conceptos teóricos

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de L^AT_EX¹.

3.1. Secciones

Las secciones se incluyen con el comando section.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.2. Referencias

Las referencias se incluyen en el texto usando cite [3]. Para citar webs, artículos o libros [2].

¹Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de L^AT_EX, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 3.1: Autómata para una expresión vacía

3.4. Listas de items

Existen tres posibilidades:

- primer item.
 - segundo item.
1. primer item.
 2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

■

3.5. Tablas

Igualmente se pueden usar los comandos específicos de L^AT_EX o bien usar alguno de los comandos de la plantilla.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket	X		X	X	X
MikT _E X					X
T _E XMaker					X
Astah					X
Balsamiq Mockups	X				
VersionOne	X		X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Técnicas y herramientas

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

4.1. *Azure*

4.2. *Microsoft Visual Studio Code*

4.3. *Android Studio*

Para el desarrollo de la aplicación se ha decidido utilizar Android Studio, el cual es el entorno de desarrollo integrado oficial de Google para aplicaciones en Android. Desde el 7 de marzo del 2019 Kotlin es el lenguaje de programación preferido de Google para el desarrollo de aplicaciones en Android, aunque esta IDE también permite la implementación de las mismas en otros lenguajes de programación, como C++ o Java. Como lenguaje de programación se ha utilizado este último, ya que es un lenguaje que se ha utilizado a lo largo de la carrera en diferentes asignaturas, siendo uno

de los lenguajes de programación más utilizados en los últimos años. Las principales ventajas de utilizar Android Studio son las siguientes:

- Como se ha mencionado con anterioridad, es la IDE oficial de Google para el desarrollo de aplicaciones de Android, desbancando a Eclipse en el año 2013.
- Permite la conversión de código Java a código Kotlin, algo que es imposible en otras IDEs como Eclipse, ya que Kotlin es un lenguaje el cual se ejecuta sobre una máquina virtual de Java, permitiendo también utilizar sus librerías.
- Permite programar la interfaz de la aplicación de forma interactiva, todo ello mediante los ficheros .xml del directorio /res/layout, pudiendo intercalar de forma sencilla entre la forma interactiva y el código .xml.
- Permite simular el funcionamiento de la aplicación sobre diferentes dispositivos, ya sean virtuales mediante su emulador, o físicos pudiendo conectar diferentes dispositivos mediante las opciones de desarrollador de los dispositivos Android.
- Permite inicializar proyectos a partir de plantillas preestablecidas, siendo de gran utilidad tanto para principiantes como para expertos.
-

Los principales inconvenientes de Android Studio son los siguientes:

-

Instalación

Los pasos a seguir para la instalación del programa son los siguientes

1. En la página de descarga de la aplicación [1, Android Studio] se selecciona el botón **Android Studio Electric Eel** para descargar el instalador de Android Studio:

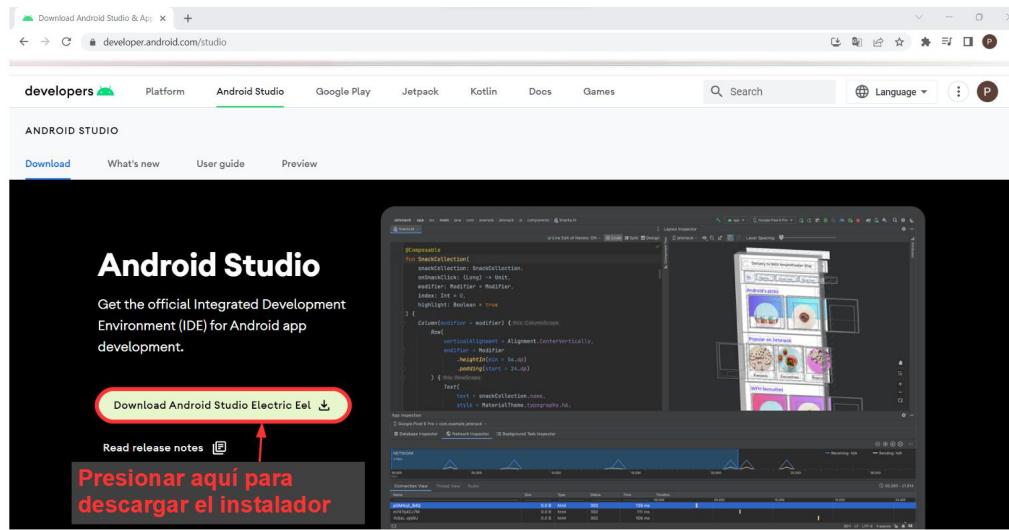


Figura 4.1: Página de descarga del instalador de Android Studio

2. Posteriormente, se tienen que aceptar los términos y condiciones marcando la opción *I have read and agree with the above terms and conditions*, para presionar posteriormente el botón **Download Android Studio Electric Eel | 2022.1.1 for Windows**.



Figura 4.2: Sección de términos y condiciones anterior a la descarga del instalador de Android Studio

3. Posteriormente se obtiene el instalador:

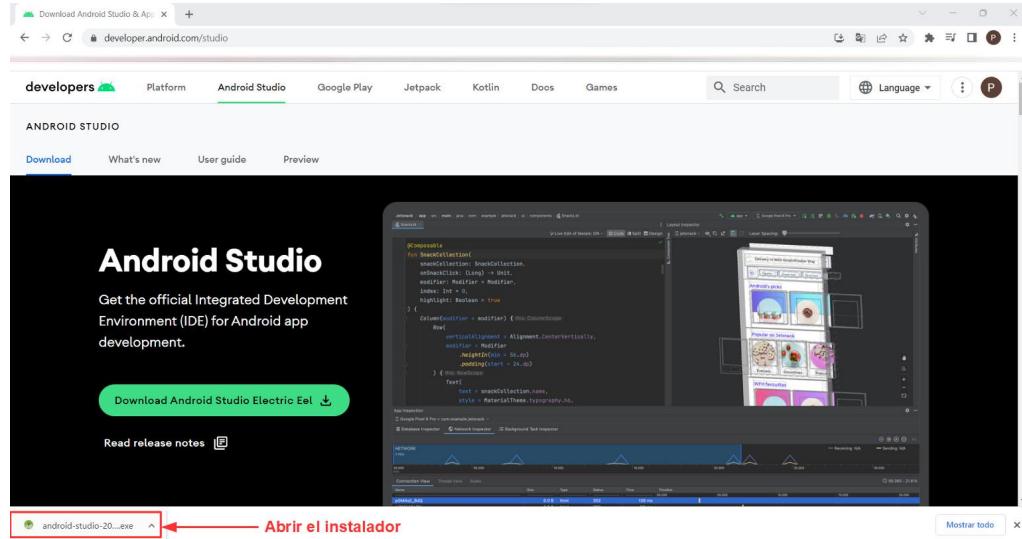


Figura 4.3: El fichero de instalación ya ha sido descargado

4. Ya abierto el asistente de instalación, se tienen que seguir los pasos que nos piden:

- a) En primer lugar se muestra el mensaje de bienvenida del asistente de instalación:

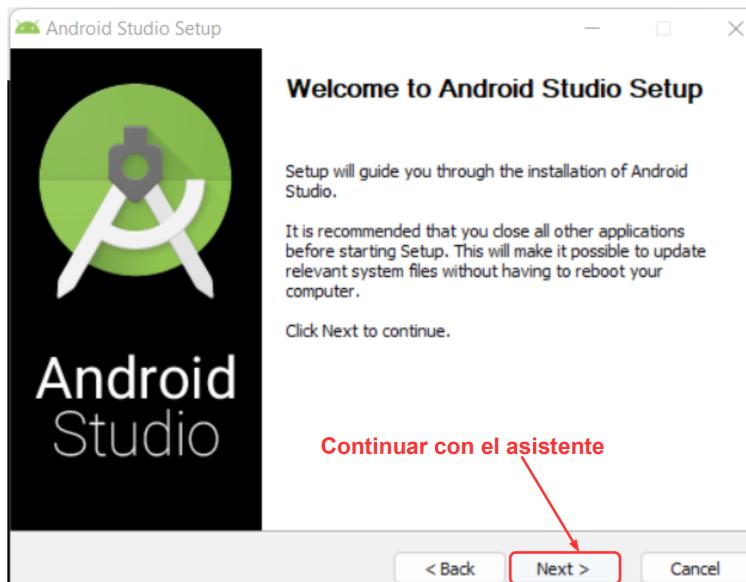


Figura 4.4: Mensaje de bienvenida del instalador de Android Studio

- b) Posteriormente se seleccionan los componentes a instalar. Se seleccionan tanto *Android Studio*, que es la propia IDE de Google para el desarrollo de aplicaciones para Android, como *Android Virtual Device*, que se utiliza de forma integrada en Android Studio para la simulación del funcionamiento de las aplicaciones para diferentes aplicaciones:

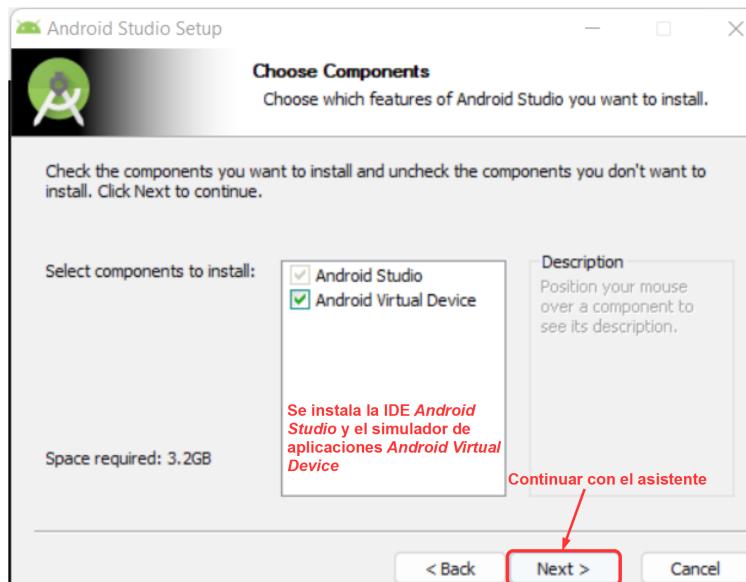


Figura 4.5: Selección de componentes a instalar en el instalador de Android Studio

- c) Más adelante se establece un directorio para la instalación del programa. Se puede elegir libremente dicho directorio, pero en este caso se va a escoger el directorio por defecto:

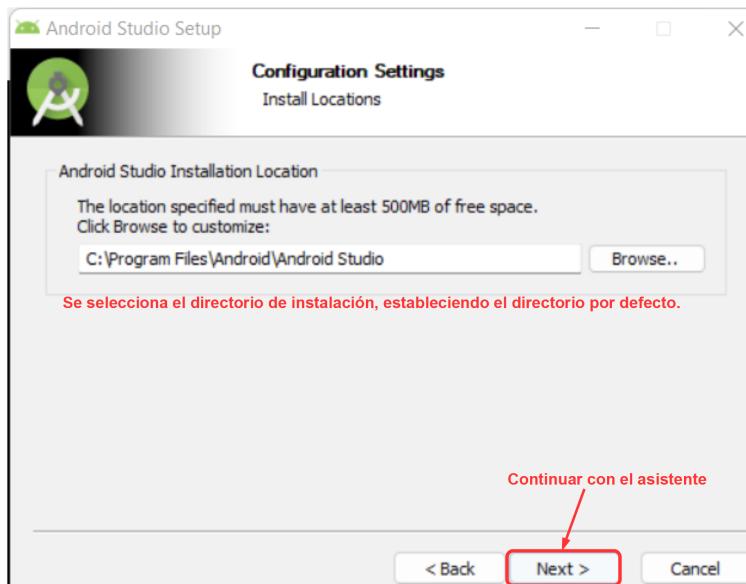


Figura 4.6: Selección del directorio de instalación

- d) Además del directorio de instalación se puede establecer un directorio donde guardar el acceso directorio al programa en el menú de inicio. No es un paso obligatorio, pero se establecerán las opciones por defecto:

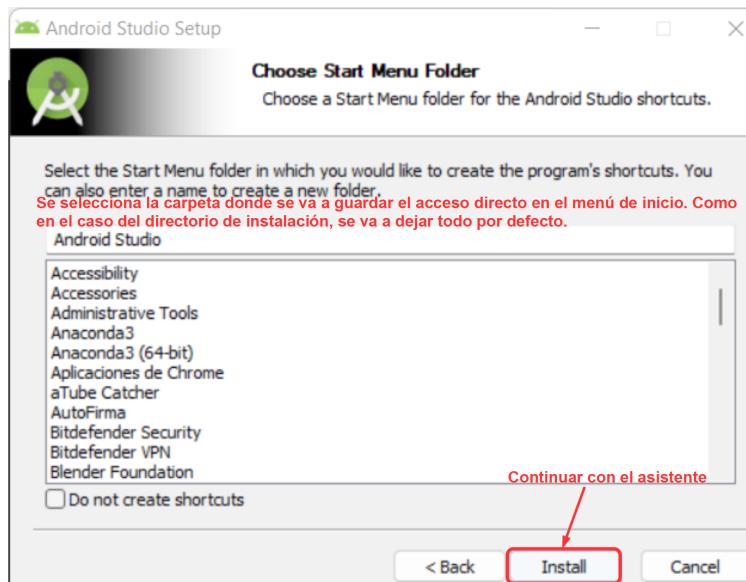


Figura 4.7: Selección del directorio de creación de accesos directos en el menú de inicio

- e) Cuando se han seguido los pasos anteriores, se procede con la instalación:

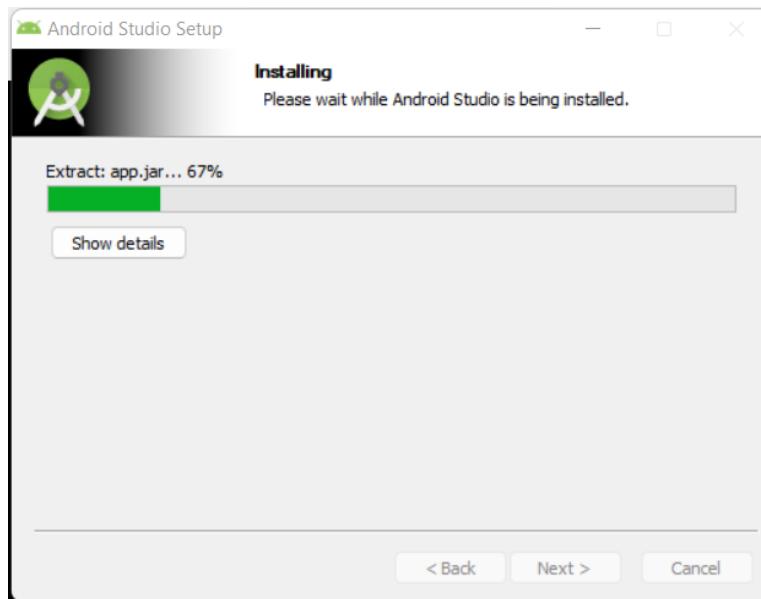


Figura 4.8: Inicio de instalación de Android Studio

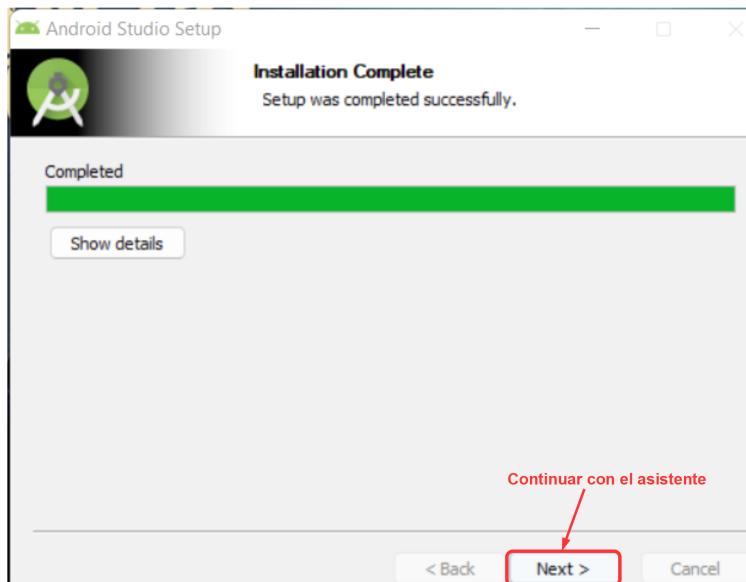


Figura 4.9: Finalización de la instalación de Android Studio

f) Por último se marca la opción de iniciar Android Studio para proceder así con la configuración del mismo:

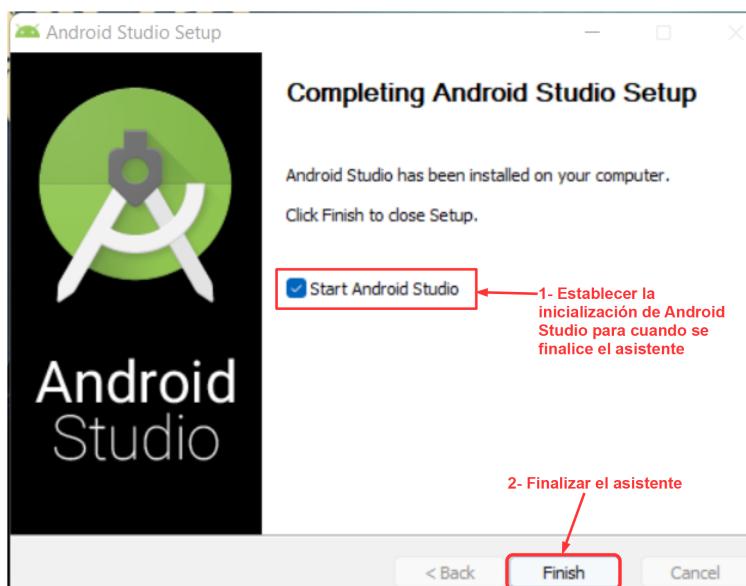


Figura 4.10: Mensaje de finalización de la instalación

Tras haber finalizado con la instalación de *Android Studio*, se procede con su configuración inicial.

Configuración inicial

En cuanto a la configuración inicial de *Android Studio* y de la aplicación, los pasos a seguir son los siguientes:

1. En primer lugar, *Android Studio* da la opción de importar la configuración de otro directorio, pero como es una instalación nueva, no se precisa de importar configuración alguna:
2. Posteriormente se muestra un mensaje para recolectar información sobre *Android Studio* y sus diferentes herramientas, el cual se va a rechazar:

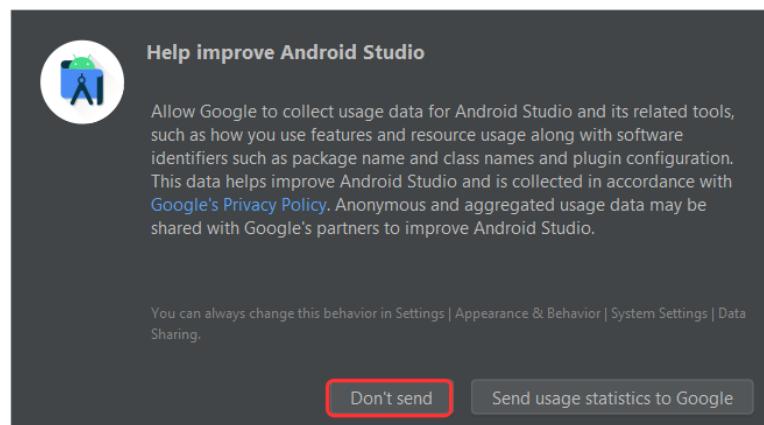


Figura 4.11: Ventana emergente de importación de configuración

- 3.
4. A continuación aparece el asistente de configuración de *Android Studio*, el cual aparece únicamente en la primera ejecución del programa. Para ir avanzando con el asistente se presiona el botón **Next**, el cual lleva al usuario al siguiente paso de la configuración:

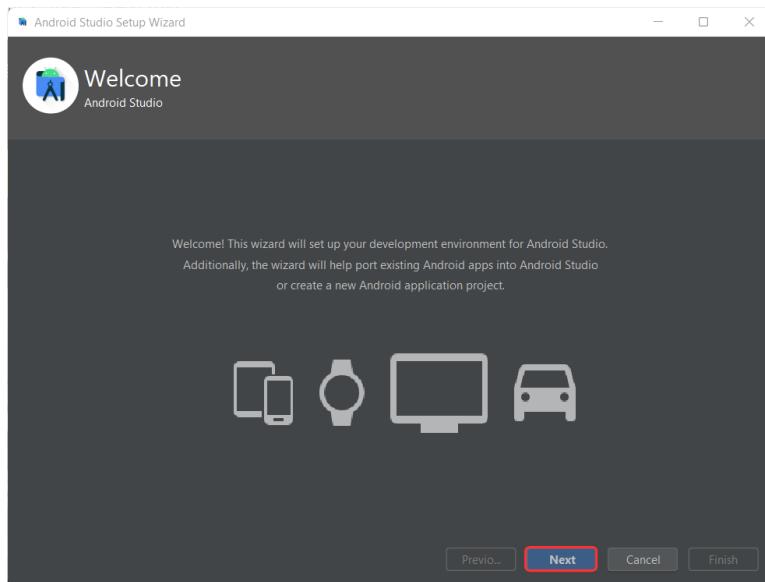


Figura 4.12: Mensaje de bienvenida del asistente de configuración inicial de Android Studio

5. Como tipo de instalación, Android Studio proporciona dos diferentes opciones, **la opción estándar**, la cual instala los componentes predefinidos, y **la opción personalizada**, la cual permite al usuario seleccionar la configuración y los componentes a instalar:

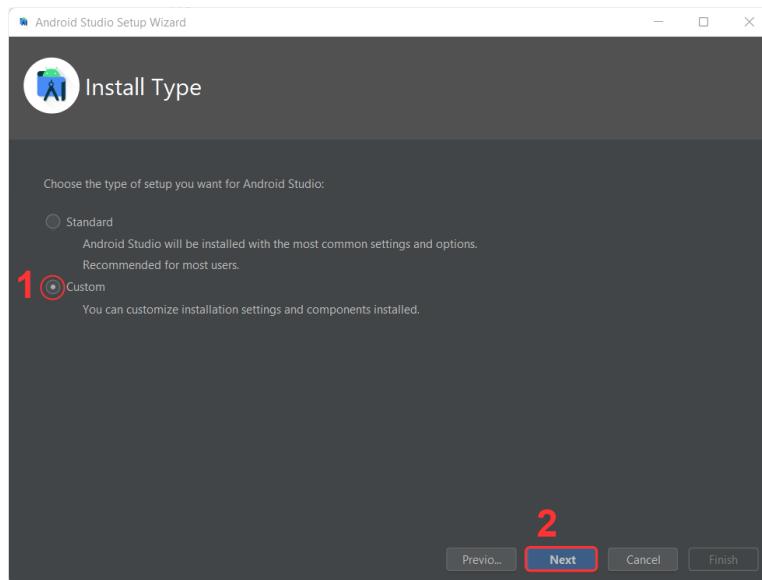


Figura 4.13: Selección del tipo de instalación de los componentes de Android Studio

6. En cuanto a la selección del directorio del JDK, Android Studio mostrará el JDK instalado más reciente. También se puede personalizar si se dispone de más de un JDK en el equipo.

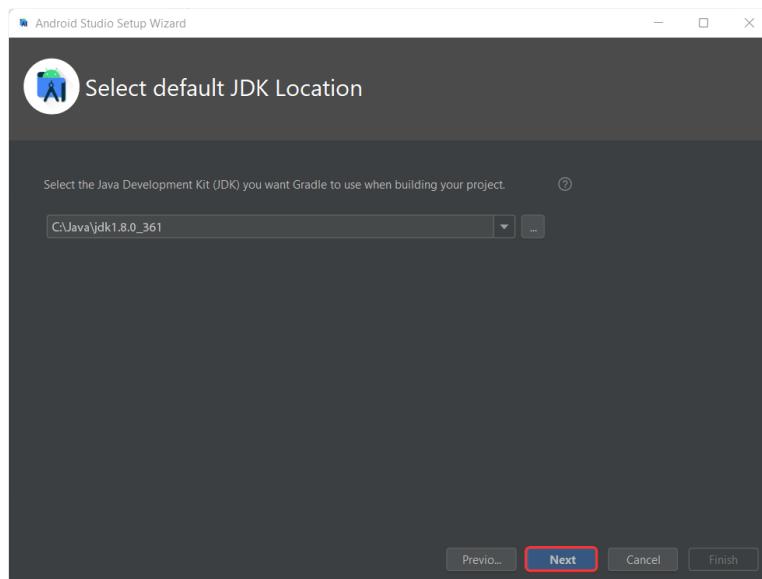


Figura 4.14: Selección del directorio predeterminado del JDK

7. A continuación se selecciona el diseño de la interfaz, el cual es irrelevante para el correcto funcionamiento de la herramienta:

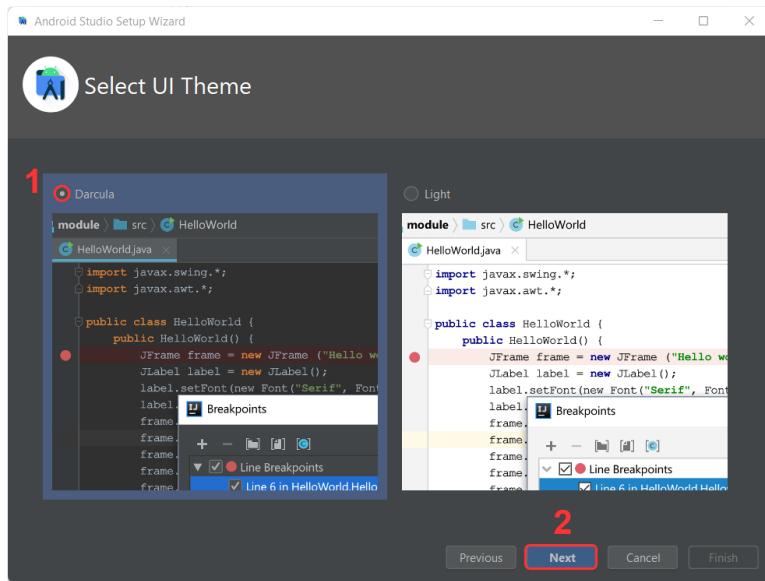


Figura 4.15: Selección del diseño de la interfaz de Android Studio

8. Posteriormente se seleccionan los componentes SDK (*Software Development Kit*) a instalar, los cuales permiten desarrollar las aplicaciones para Android. En este caso se seleccionarán los componentes de la captura posterior:

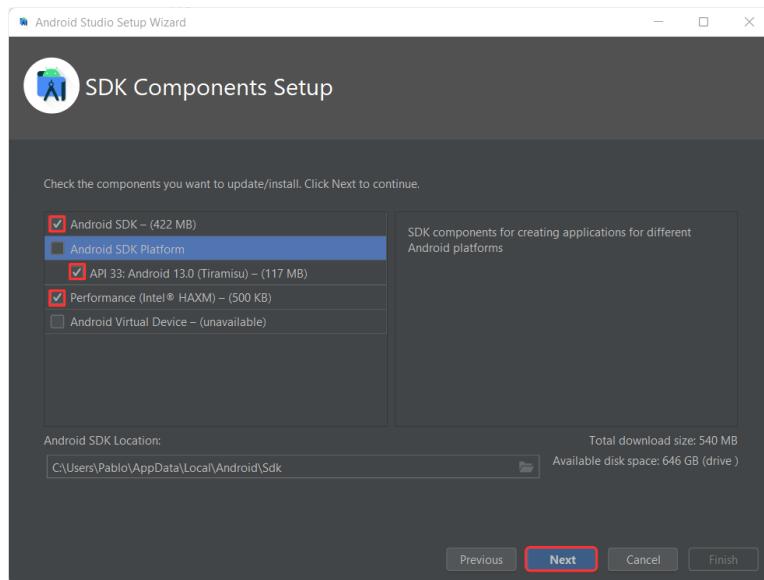


Figura 4.16: Selección de componentes SDK a instalar en Android Studio

Como se puede comprobar, en el momento de haberse realizado la captura anterior, el componente Android Virtual Device no está disponible. Aun así, podemos proseguir con la instalación presionando **OK**:

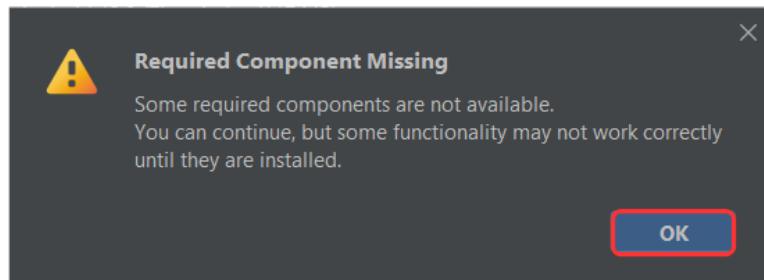


Figura 4.17: Advertencia por falta de componentes a instalar

9. Posteriormente se configura la memoria RAM máxima que utilizará el emulador de aplicaciones. En este caso dejamos la opción recomendada de 2GB de RAM:

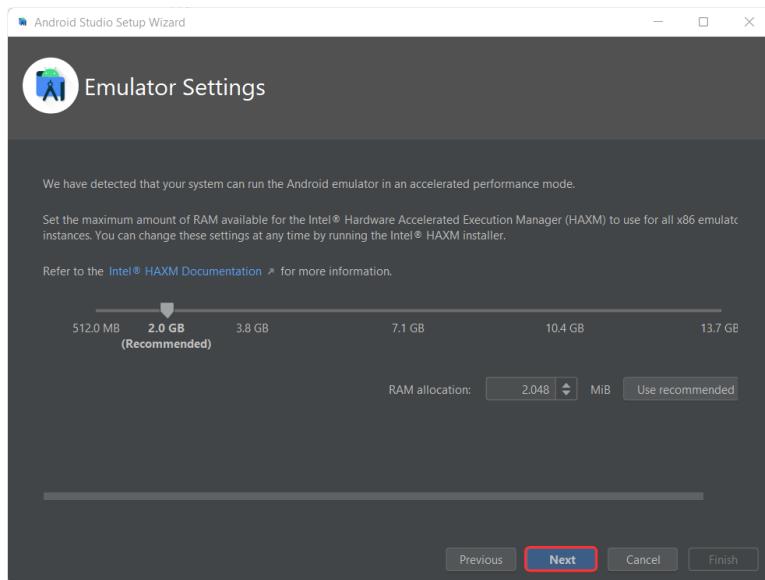


Figura 4.18: Configuración de la memoria RAM utilizada por el emulador de Android

10. Más adelante se comprueban todos los componentes a instalar y a configurar durante la configuración inicial de Android Studio:

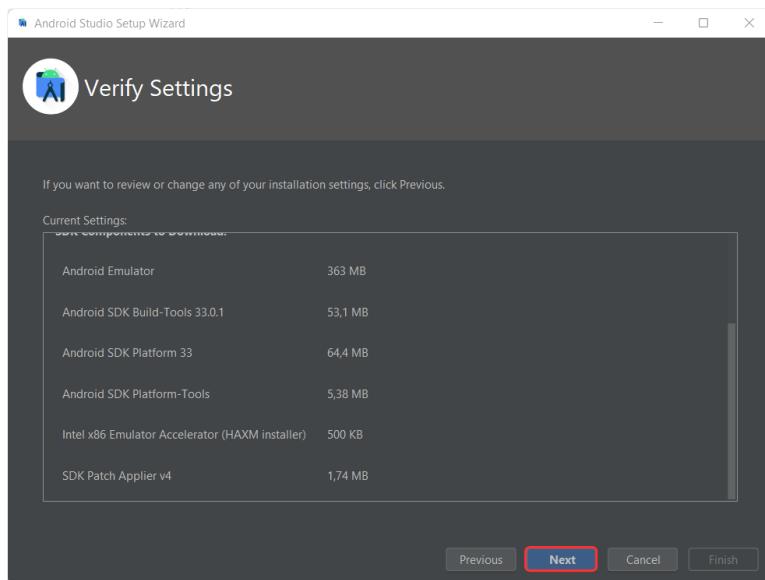


Figura 4.19: Verificación de componentes a instalar en Android Studio

11. Antes de instalar los componentes, se tienen que aceptar la licencia del software tanto del *paquete SDK* como del *acelerador de emulación de Intel*:

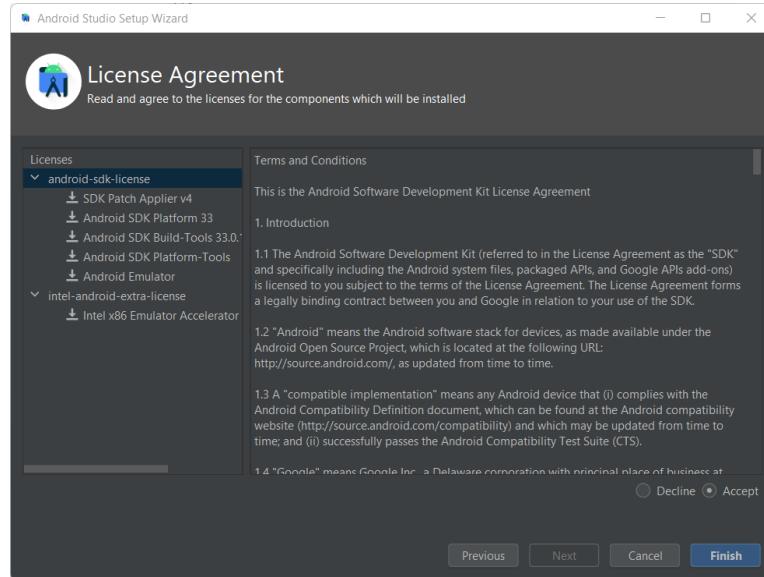


Figura 4.20: Licencia de software del paquete SDK

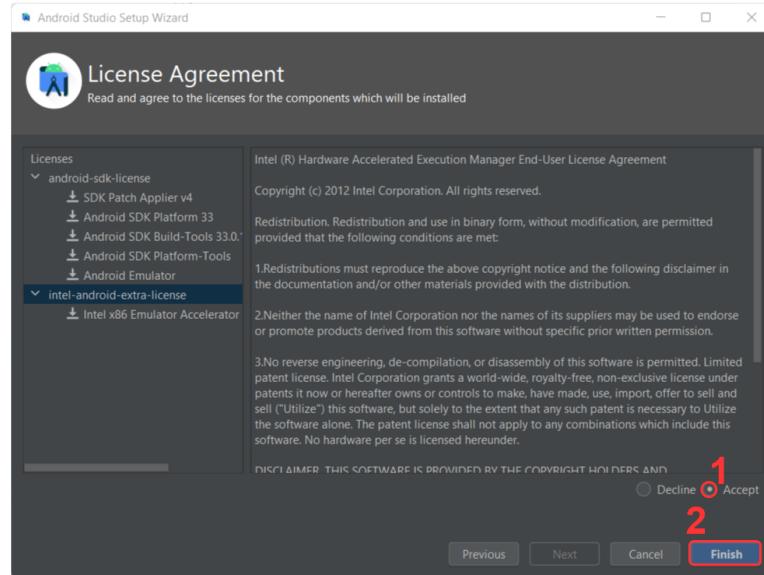


Figura 4.21: Licencia de software del acelerador de emulación de Intel

12. Por último se procede con la instalación de los componentes mencionados con anterioridad y se finaliza con el asistente de configuración inicial presionando el botón **Finish**:

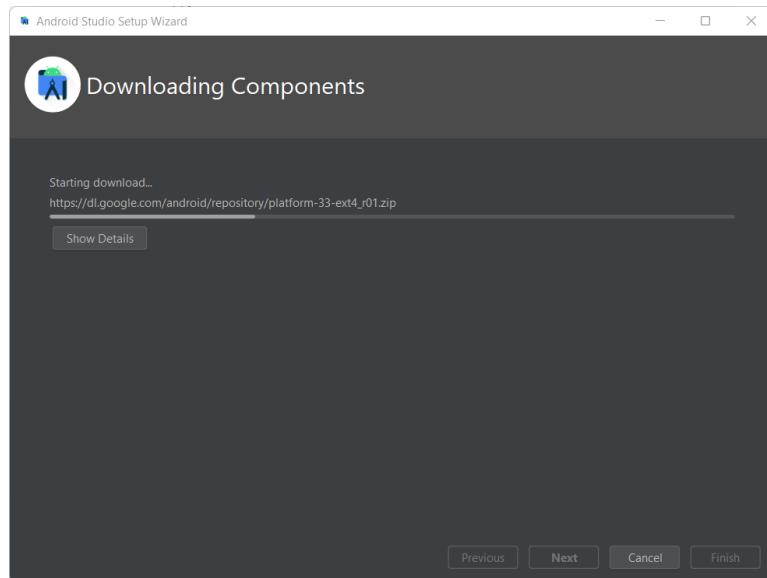


Figura 4.22: Instalación de los componentes en la configuración inicial

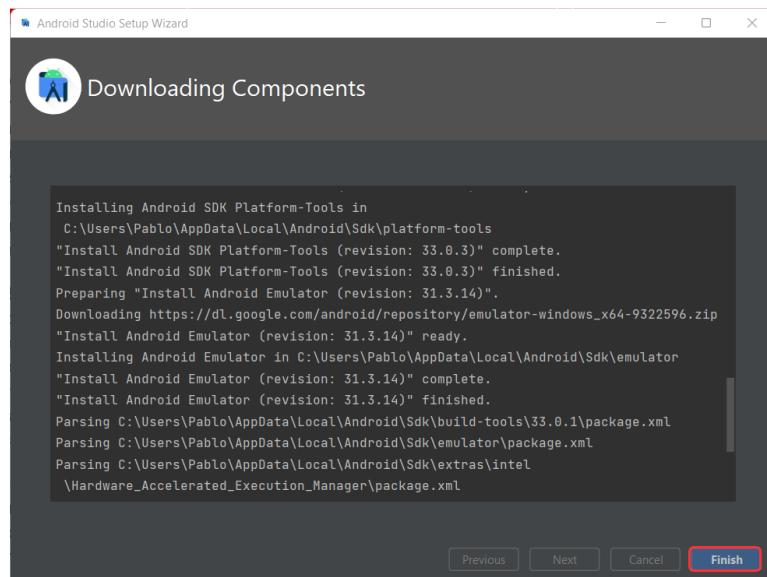


Figura 4.23: Finalización del asistente de instalación de componentes

Creación del proyecto nuevo en Android Studio y configuración de GitHub

Tras haber finalizado con la instalación y con la configuración inicial de Android Studio, se tiene que iniciar un proyecto inicial donde se va a desarrollar la aplicación. Los pasos a seguir son los siguientes:

1. En primer lugar en la pantalla de bienvenida se presiona sobre el botón **New Project** para crear un proyecto de aplicación nuevo:

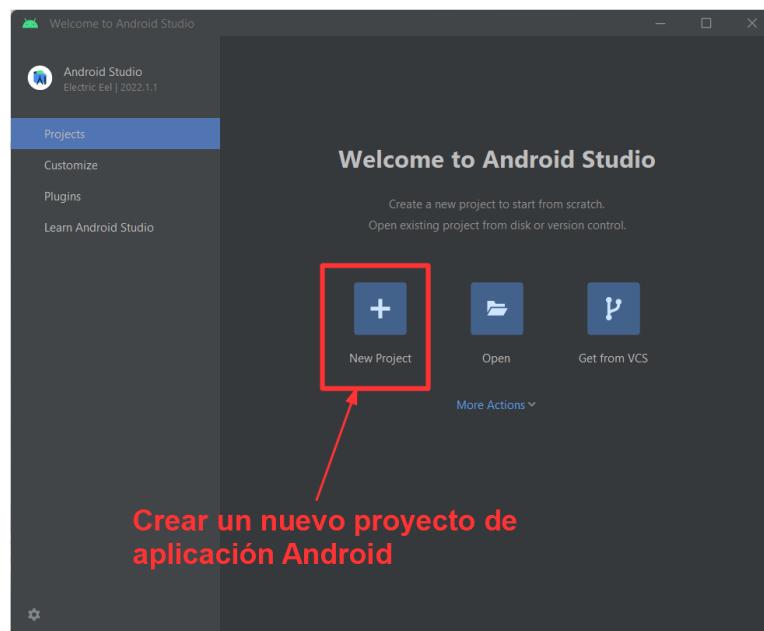


Figura 4.24: Pantalla de selección de proyecto

2. Posteriormente se selecciona el diseño de la actividad principal de la aplicación. Para simplificar la estructura inicial del proyecto se selecciona la actividad vacía o *Empty Activity*. Al tener la actividad ya seleccionada se presiona el botón **Next** para continuar con la configuración de la actividad:

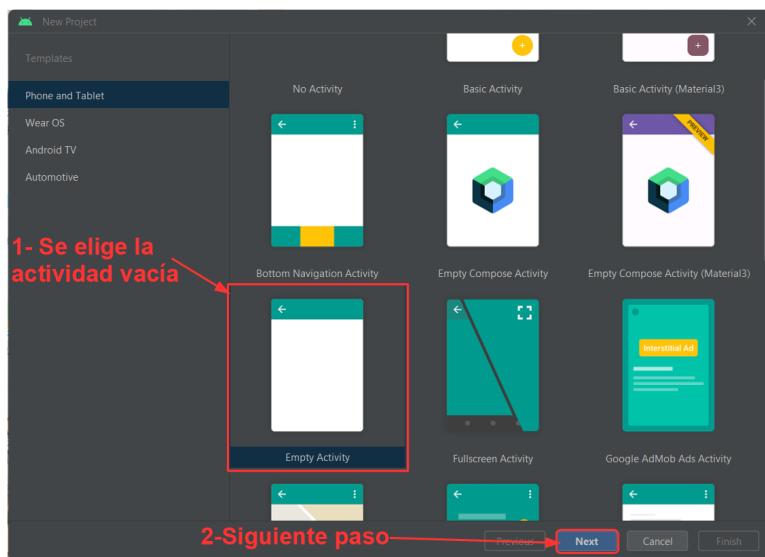


Figura 4.25: Pantalla de selección de actividad principal

3. Ya en la pantalla de configuración de la actividad, se puede modificar el nombre de la misma, el nombre del paquete a la que pertenece, el directorio donde se va a guardar, el lenguaje de programación en la que va a ser programada y la versión mínima de Android en la que puede ser ejecutada:

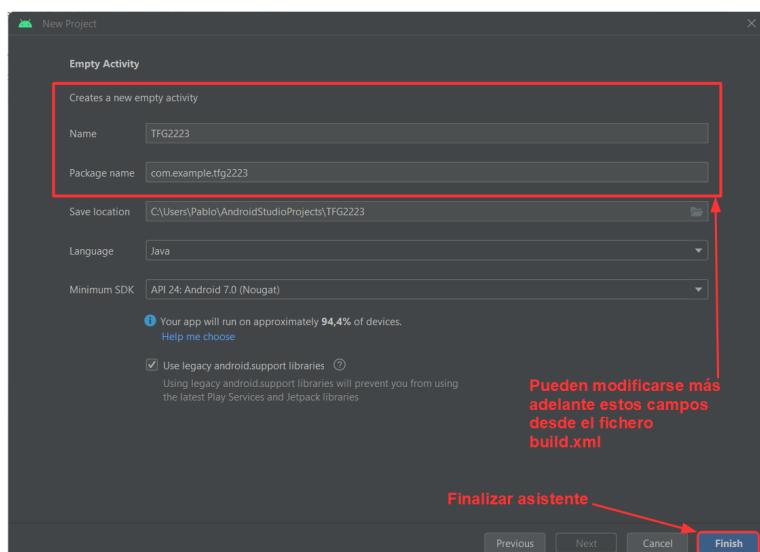


Figura 4.26: Pantalla de configuración de la actividad

Cuando ya se ha configurado la actividad, se instalará el soporte JDK para la emulación de la misma:

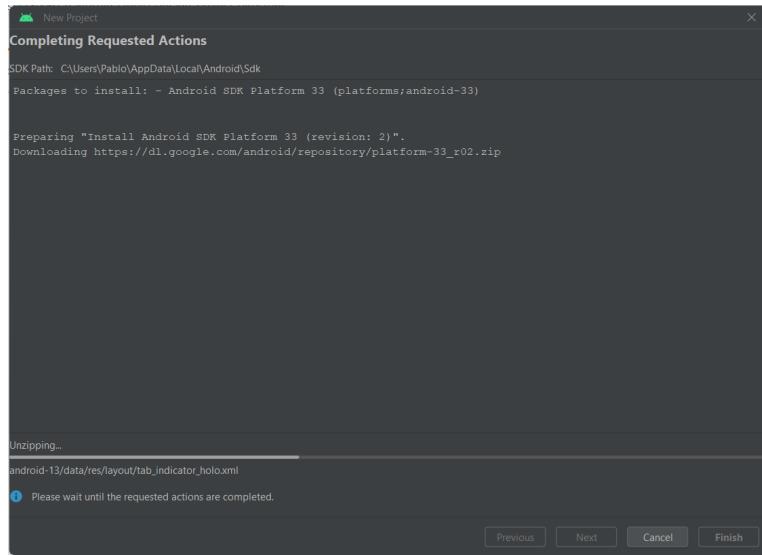


Figura 4.27: Desarrollo del proceso de instalación del soporte JDK

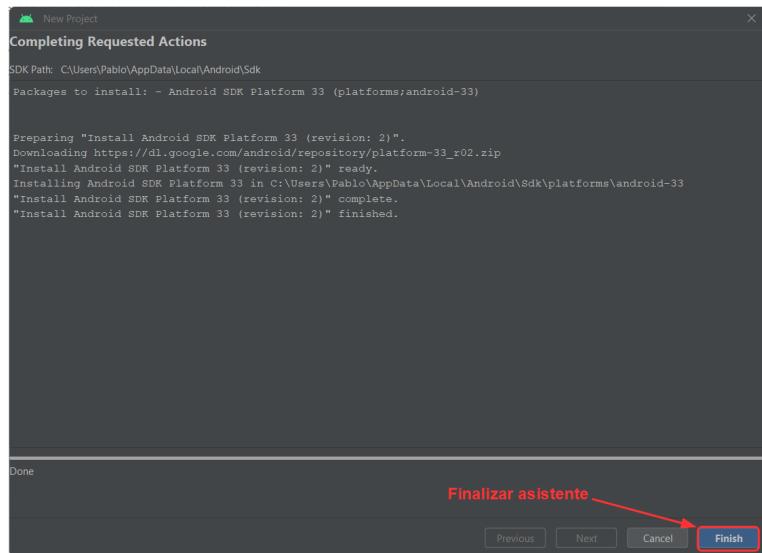


Figura 4.28: Finalización del proceso de instalación del soporte JDK

Cuando ya se haya terminado con el asistente de creación del proyecto, ya nos mostrará el código Java de esta actividad vacía que acabamos de

crear, además de que en la parte izquierda de la pantalla encontramos todas las rutas creadas dentro del directorio del proyecto.

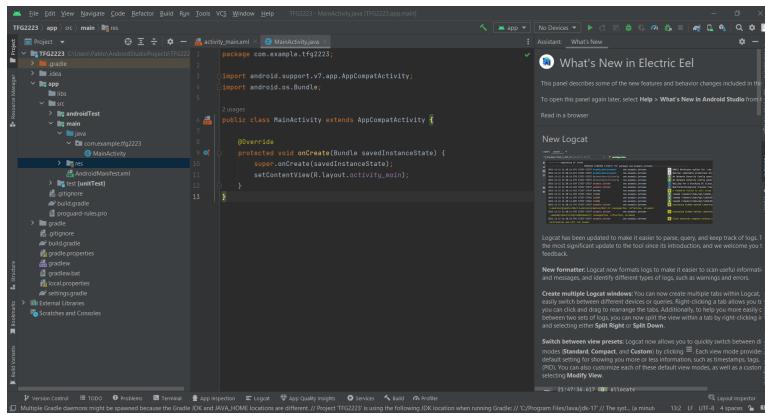


Figura 4.29: Interfaz de Android Studio con el proyecto Android recién creado

4. Tras haberse creado el proyecto, se va a proceder a añadir una cuenta de GitHub para publicar los progresos que vayan realizándose durante el desarrollo de la aplicación. Para ello en primer lugar se va a **File** → **Settings**:

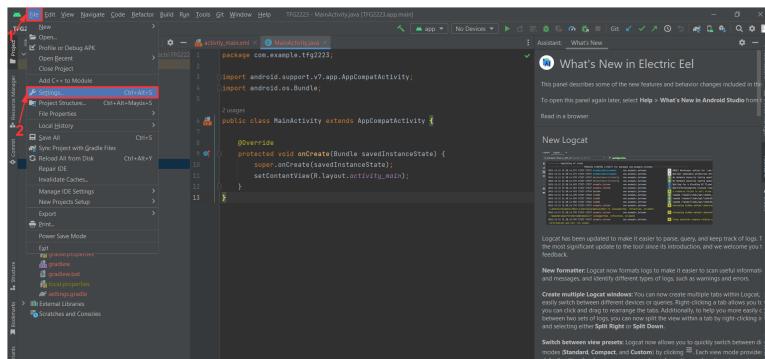


Figura 4.30: Acceso a las opciones de Android Studio

5. Posteriormente se tiene que acceder a **Version Control** → **GitHub**

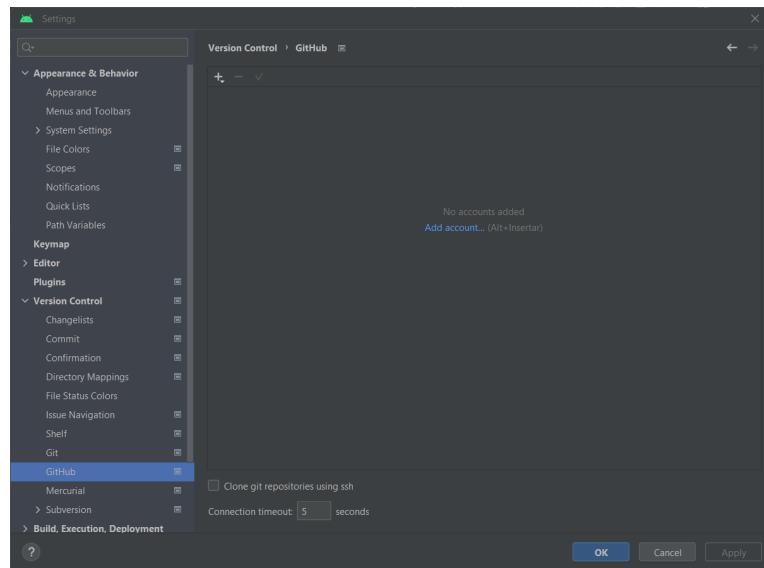


Figura 4.31: Pantalla de control de cuentas de GitHub

6. Más adelante se accede a la configuración de GitHub, concretamente en ***Settings*** → ***Developer Settings*** → ***Personal access tokens*** → ***Tokens (classic)***, para ir a la página de muestra de tokens de usuario ***Generate new token*** → ***Generate new token (classic)***:

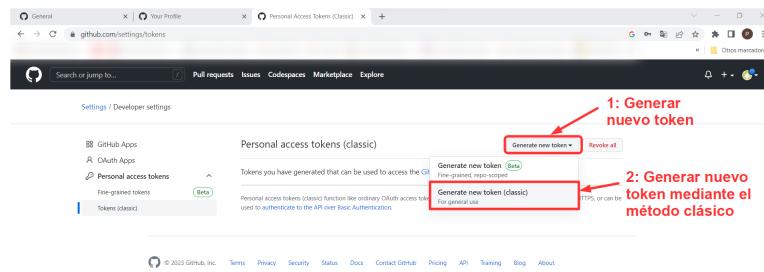


Figura 4.32: Pantalla de tokens de usuario antes de ir a la pantalla de generación de tokens

7. En cuanto a la generación del token, se menciona en una nota para qué queremos utilizar ese token, la fecha de expiración del mismo (la

cual se aconseja que sea de duración hasta junio, fecha de la defensa) y se seleccionan todos los permisos de uso del token:

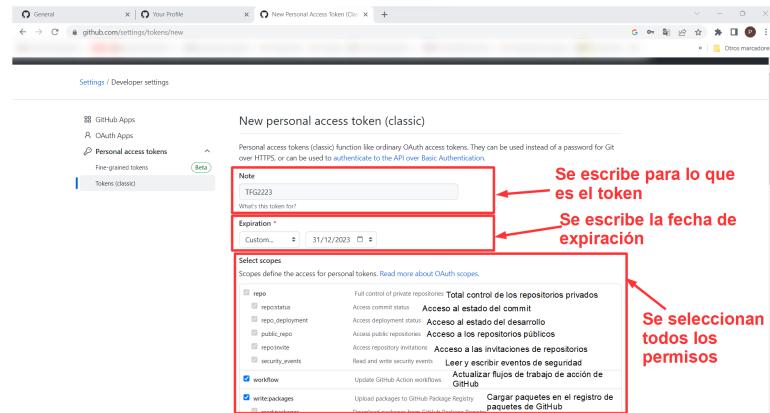


Figura 4.33: Pantalla de generación de tokens con los permisos de usuario

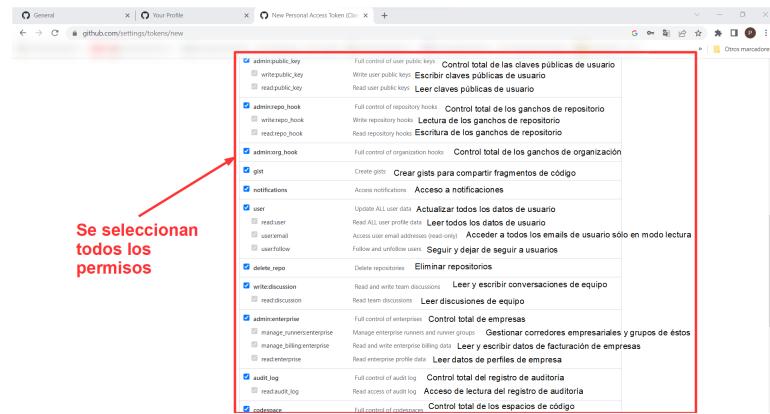


Figura 4.34: Siguientes permisos de usuario

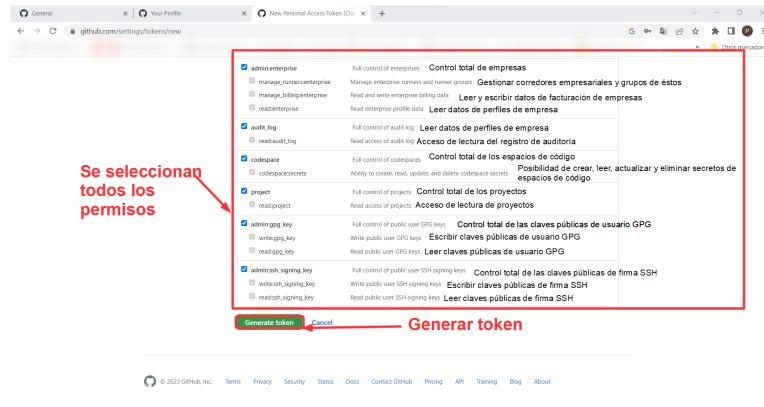


Figura 4.35: Últimos permisos de usuario

- Con el token recién generado, se copia para poder añadirlo a Android Studio:

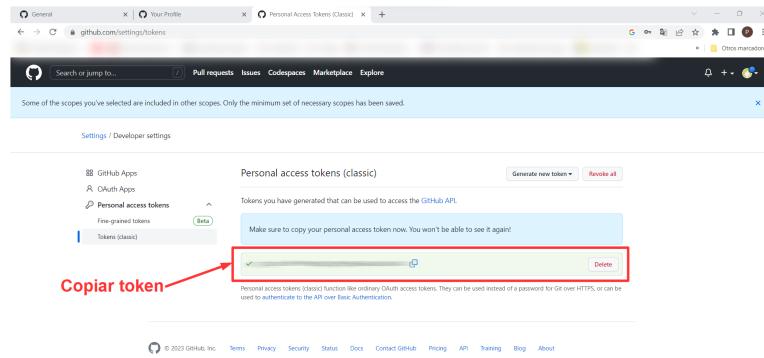


Figura 4.36: Pantalla de tokens de usuario con el token de usuario generado

En la pantalla de control de cuentas de GitHub en Android Studio, se presiona en **Add account** y se añade el token

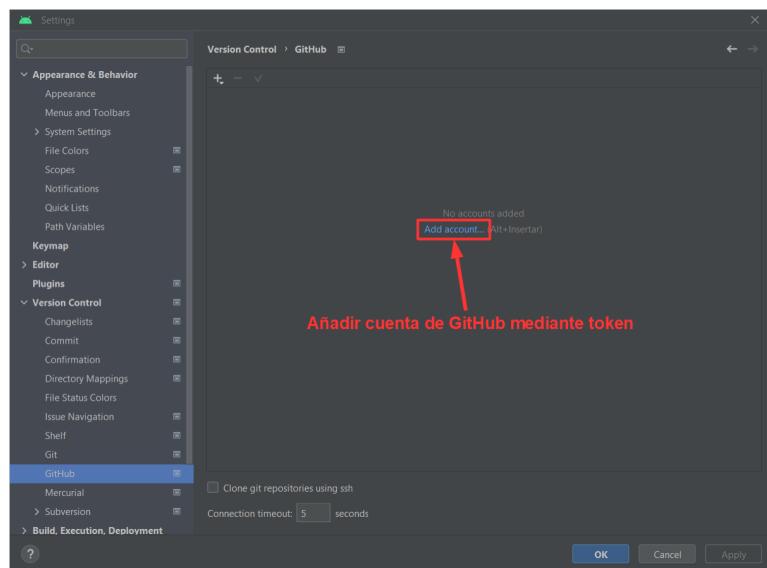


Figura 4.37: Pantalla de control de cuentas de GitHub lista para añadir usuario

Para confirmar la agregación del token, se presiona sobre el botón **Add Account**:

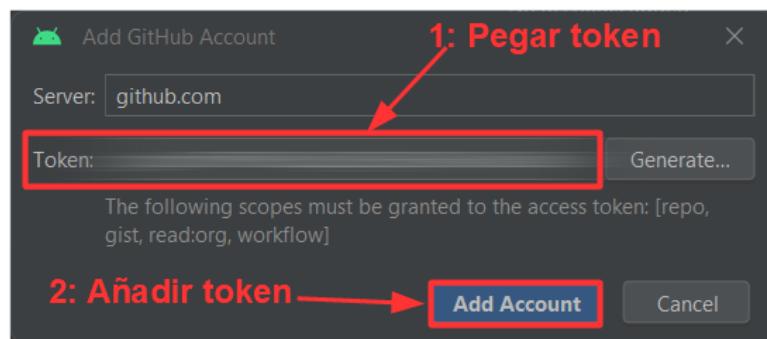


Figura 4.38: Añadir token de usuario de GitHub en Android Studio

Con el token ya añadido, se presiona el botón **Ok**:

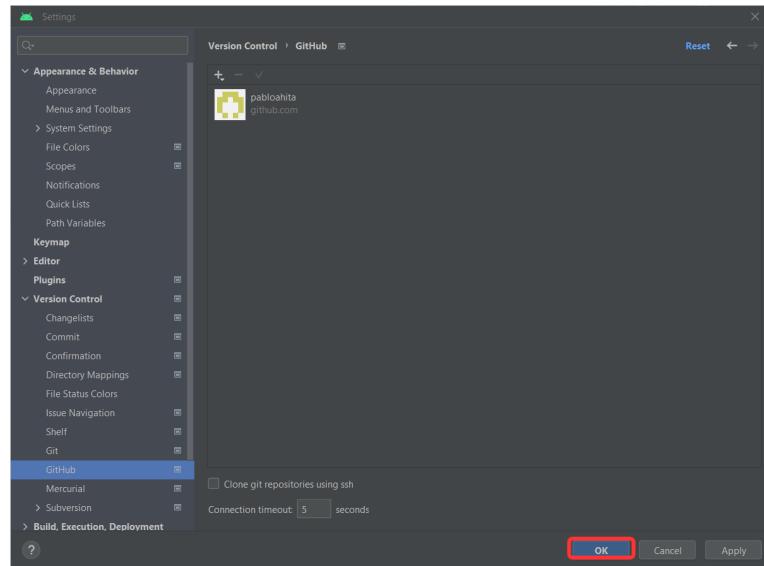


Figura 4.39: Cuenta de GitHub recién añadida

9. Tras haber añadido satisfactoriamente la cuenta de GitHub a Android Studio, se va a proceder a añadir un dispositivo virtual. Para ello se accede al menú desplegable con los dispositivos y posteriormente al administrador de dispositivos, es decir, **No Device → Device Manager**:

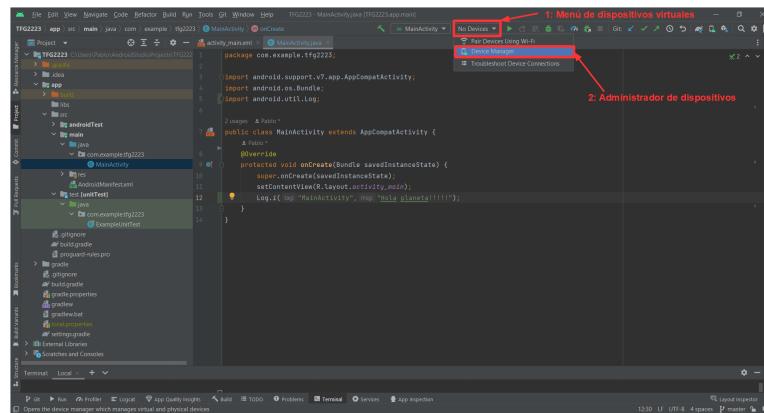


Figura 4.40: Acceso al administrador de dispositivo mediante el menú desplegable de dispositivos

10. Posteriormente sale en la parte derecha de la pantalla el administrador de dispositivos (**Device Manager**), el cual muestra todos

los dispositivos virtuales que se utilizan durante las simulaciones de funcionamiento. Para añadir un dispositivo virtual basta con presionar el botón **Create Device**:

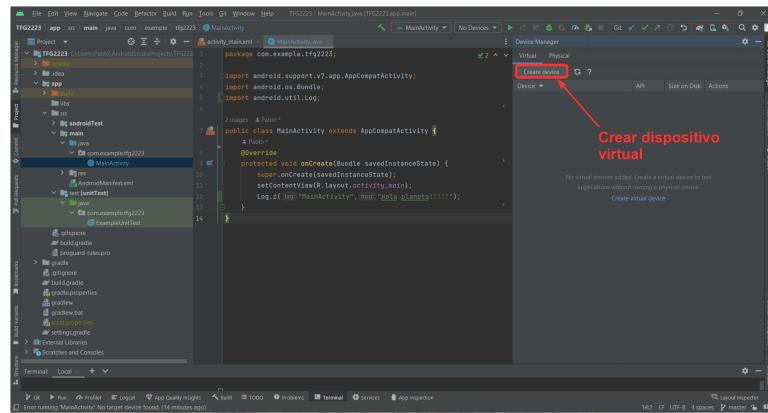


Figura 4.41: Administrador de dispositivos

11. En cuanto a la configuración del nuevo dispositivo virtual, existen diferentes opciones para elegir, pudiendo simular aplicaciones para smartphones, tablets, dispositivos wearable, escritorio, smart TV y Android Auto. Como primer dispositivo se va a añadir el smartphone *Pixel 6 Pro* de Google. Posteriormente se presiona el botón **Next** para seguir con el siguiente paso:

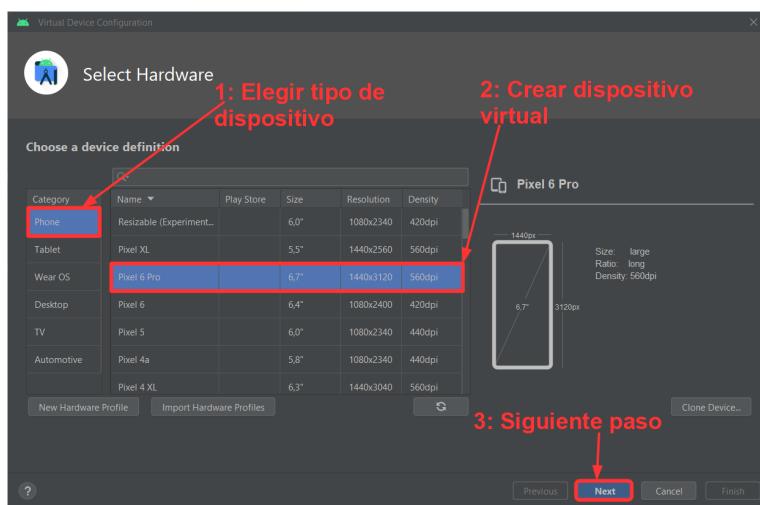


Figura 4.42: Selección del nuevo dispositivo virtual

12. Posteriormente se elige el sistema operativo de este dispositivo virtual. Para ello se va a seleccionar dicho sistema operativo para descargarlo para el dispositivo y posteriormente se presiona sobre **Next**:

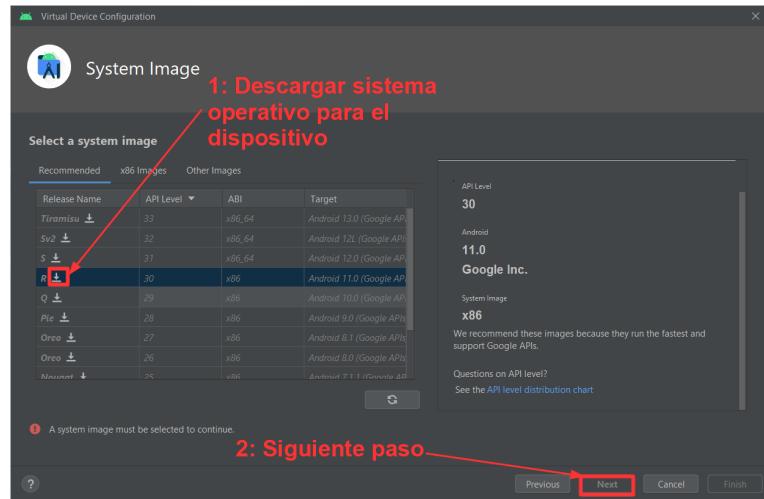


Figura 4.43: Selección de sistema operativo para el nuevo dispositivo virtual

13. Tras haber seleccionado la configuración tanto de hardware como de software se procede con la instalación tanto del dispositivo como de su sistema operativo. Al finalizar se presiona el botón **Finish**:

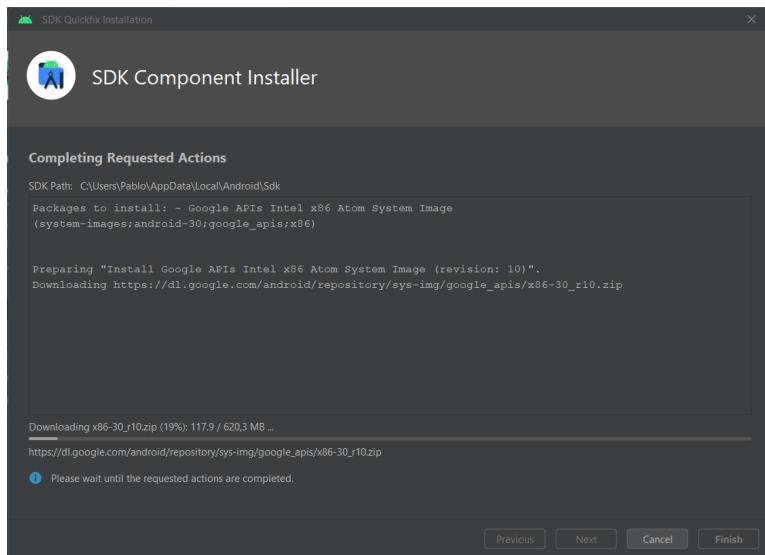


Figura 4.44: Proceso de instalación del sistema operativo y de configuración del dispositivo virtual



Figura 4.45: Finalización del proceso de instalación del sistema operativo y de configuración del dispositivo virtual

14. Para finalizar se puede comprobar la configuración que se ha establecido para el dispositivo nuevo, además de poder cambiar la orientación inicial del mismo. Cuando se quiera finalizar con la creación del dispositivo virtual, se presiona el botón **Finish**:



Figura 4.46: Finalización del asistente de creación del dispositivo y comprobación de la configuración

Hacer un commit

Al tener la cuenta de GitHub agregada en Android Studio, se pueden realizar todas las operaciones de git desde la propia interfaz. Por lo tanto, los pasos a seguir para hacer un commit y publicarlo en GitHub son los siguientes:

1. En primer lugar hay que desplazarse a la barra de herramientas superior y ahí acceder a la ruta **Git → Commit**. Alternativamente se puede acceder a la configuración del nuevo commit mediante el comando ***Ctrl+K***:

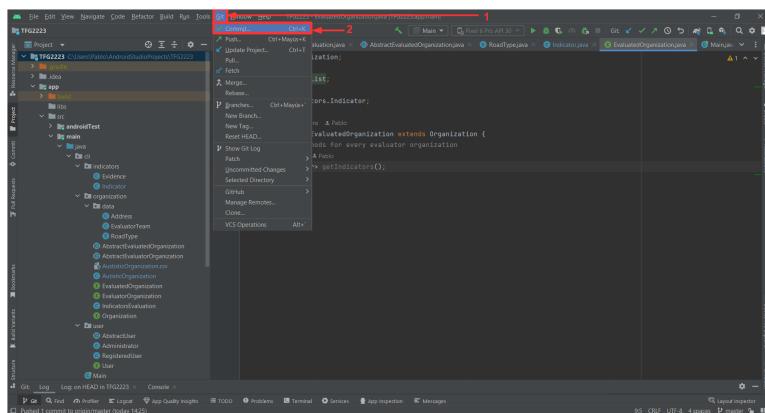


Figura 4.47: Acceso a la configuración del nuevo commit

2. Posteriormente se seleccionan los ficheros que hayan tenido cambios, se escribe el mensaje de commit y se presiona en el botón ***Commit and Push***:

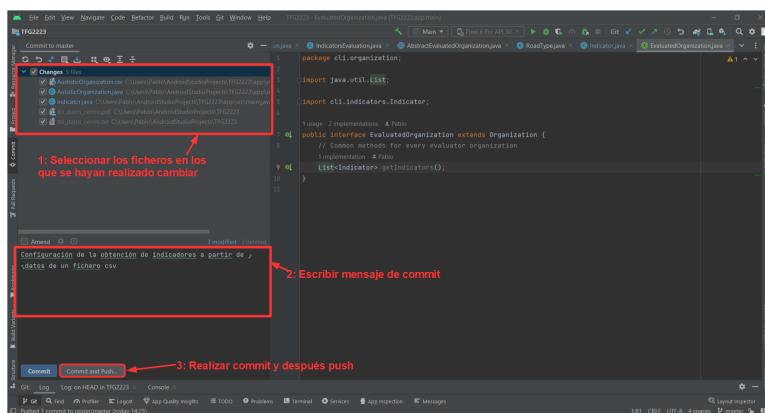


Figura 4.48: Configuración del nuevo commit

Tras presionar ese botón analizará los posibles fallos y warnings que tenga el código de la aplicación. En la siguiente captura se han detectado cuatro warnings, dos en la clase **Indicator** y dos en la clase **AutisticOrganization**. Se puede optar por solucionar esos warnings y reiniciar la comprobación anteriormente mencionada, o por realizar las operaciones de commit y push mediante el botón ***Commit Anyway and Push***:

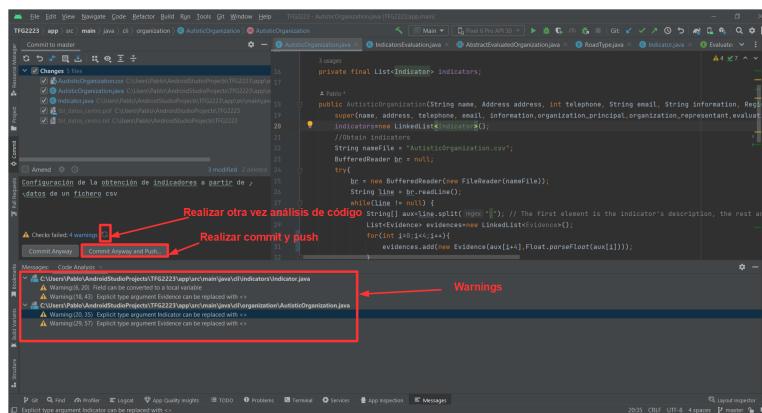


Figura 4.49: Soluciones a warnings durante el proceso de commit

3. Tras haber solucionado todos los warnings, se presiona el botón ***Push*** para publicar los cambios realizados en GitHub:

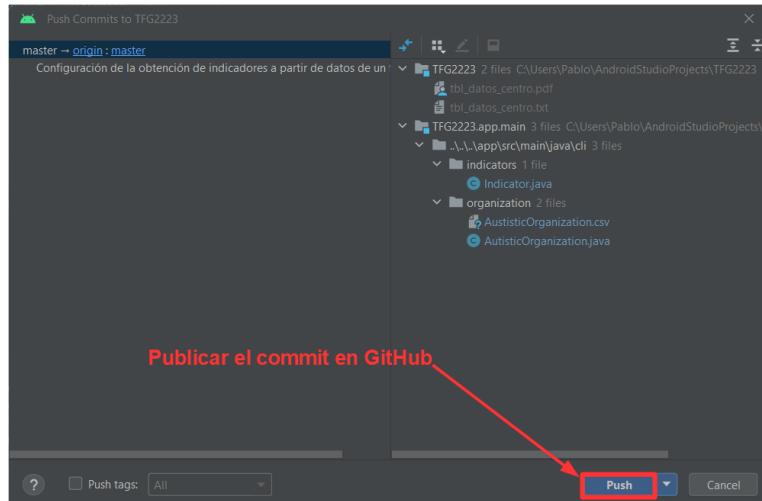


Figura 4.50: Menú de publicación de cambios realizados

Al acabar con todo este proceso sale este mensaje:

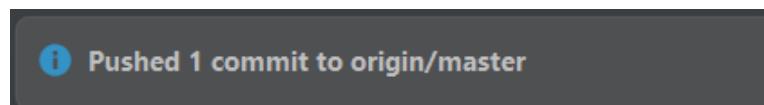


Figura 4.51: Commit y push realizados satisfactoriamente

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros³, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

5.1. Aspectos relevantes

Resumen del funcionamiento esperado de la aplicación Indicadores

Costes del desarrollo

Análisis del entorno

Roles de usuario y casos de uso

En esta aplicación se distinguen cuatro diferentes niveles de usuarios, teniendo cada uno de ellos diferentes funciones y permisos, siendo éstos últimos representados mediante los diferentes casos de uso dentro de la aplicación:

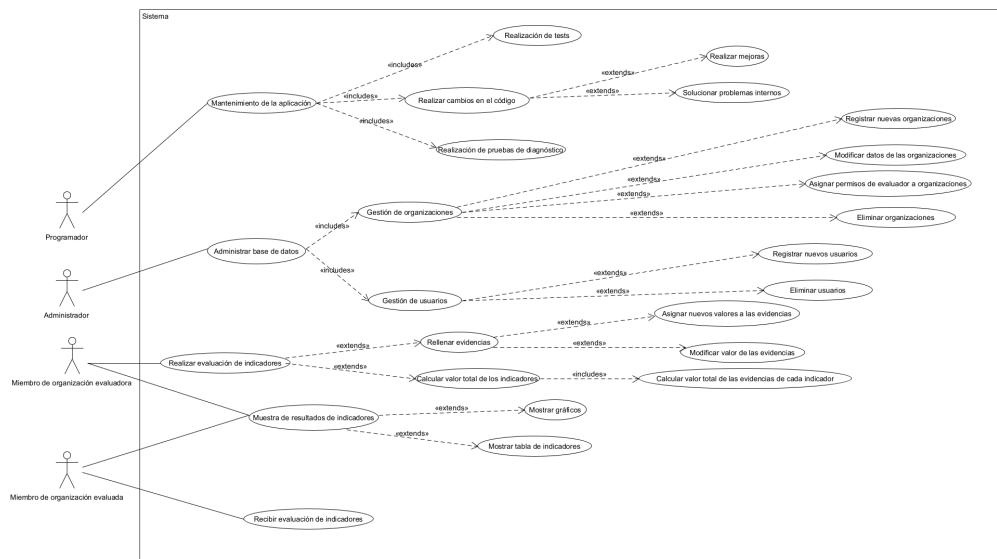


Figura 5.1: Diagrama de casos de uso de la aplicación

Como se puede comprobar en este diagrama, se distinguen cuatro tipos distintos de usuario con diferentes permisos dentro de la aplicación:

- **Programador:** Es el usuario encargado del desarrollo de la aplicación. Dentro del funcionamiento de la aplicación tiene permisos de mantenimiento de la aplicación, donde se realizan test para comprobar que la aplicación funcione de la manera deseada. También es el encargado de

realizar cambios en el código, incluyendo la realización de mejoras y la solución de problemas internos en el funcionamiento de la aplicación, como es el caso de incompatibilidades y fallos internos de la misma. Para poder arreglar fallos, el programador es el encargado de realizar las diferentes pruebas de diagnóstico, ayudando a comprobar el tipo de fallo que pueda haber en la aplicación.

- **Administrador:** Es el usuario encargado de gestionar todas las operaciones relacionadas con las organizaciones y con los usuarios almacenados en la base de datos. En cuanto a las organizaciones, este usuario se encarga de dar de alta o de baja a las organizaciones que así lo deseen, además de proporcionar los permisos de evaluador a las organizaciones evaluadoras y modificar datos de las mismas. En cuanto a los usuarios, las operaciones a realizar son las mismas que en el caso de las organizaciones, con la excepción de dar permisos de evaluación, al ser éste un requisito exclusivo de las organizaciones.
- **Miembro de la organización evaluadora:** Es un usuario cuya organización a la que pertenece posee permisos de evaluación sobre otras organizaciones que no los tienen, rellenando los valores de las evidencias y calculando el valor total de cada indicador a partir de las mismas, también calculando el valor total obtenido mediante el valor total de cada indicador. También tiene permisos para modificar dichos indicadores durante la evaluación. Además de eso puede observar los resultados de cada evaluación de indicadores, ya sea mediante tablas o mediante gráficos.
- **Miembro de la organización evaluada:** Es un usuario cuya organización a la que pertenece posee no permisos de evaluación, por lo tanto, dicho usuario pertenece a una organización que está a disposición de ser evaluada. Además de recibir la evaluación por parte de la organización evaluadora, también puede observar los resultados de las diferentes evaluaciones, ya sea mediante tablas o mediante gráficos.

Requisitos funcionales

Requisitos no funcionales

Diseño del entorno

Diagrama de clases

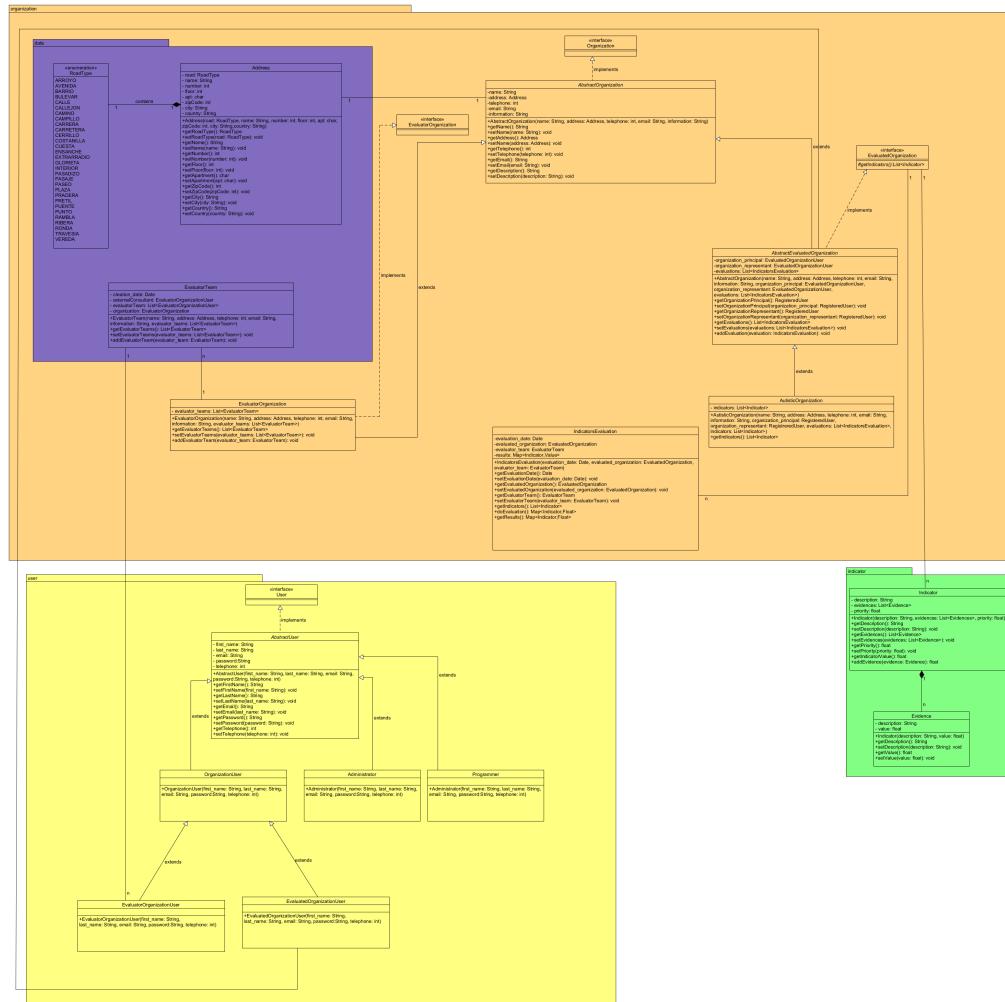


Figura 5.2: Diagrama de clases de la aplicación

Patrones de diseño utilizados

Herramientas a utilizar

Pseudocódigos

Trabajos relacionados

Los trabajos final de grado de referencia que se han utilizado son los siguientes:

- **Prueba de concepto Azure Monitor:** Es un Trabajo Final de Grado que trata sobre aplicaciones APM (*Application Performance Monitor*), que son herramientas que diagnostican el rendimiento de las aplicaciones, con la finalidad de encontrar fallos en el programa, cuellos de botella , e incluso poder solucionarlos, evitando degradaciones. La funcionalidad utilizada en este TFG es *Azure Monitor*, el cual es la herramienta de este estilo proporcionada por Microsoft, cuyo objetivo es averiguar si el proyecto puede ser utilizado por una organización que tiene dos entornos completamente diferentes: un entorno de nube *Azure* donde se encuentra su sitio web y que podrá albergar en el futuro alguno de los servicios que ofrece, y otro entorno en el sitio donde residen el resto de los servicios.
- **Desarrollo de una plataforma de tratamiento y streaming de vídeo para difusión de la cultura utilizando instancias de Azure:** Es un Trabajo Final de Grado que trata sobre una herramienta que es utilizada por artistas de cine independientes para darse a conocer a través de publicidad en diferentes dispositivos, ya sea a través de aplicaciones móviles o a través de navegadores. Azure es utilizado como soporte para la base de datos y la posterior gestión de transacciones, lo que se busca para el proyecto de la Fundación Miradas.
- **Servicios en la nube con Microsoft Azure : desarrollo y operación de una aplicación Android con DevOps:** Este proyecto

trata sobre el desarrollo de una aplicación de Android con almacenamiento en la nube, utilizándose para el almacenamiento, visualización y procesado de fotografías. Dicho proyecto utiliza *Azure* para desplegar la aplicación en la nube, con ayuda de DevOps para poder alargar el ciclo de vida de la misma. Puede ser de utilidad también para poder desarrollar la aplicación tanto a nivel de frontend como a nivel de backend.

- **JIZT. Generación de resúmenes abstractivos en la nube mediante Inteligencia Artificial:** JIZT es un servicio de generación automática de resúmenes basado en la corriente *Cloud Native*, que se basa en los principios de los sistemas escalables, elasticidad y agilidad. Dicho servicio es sustentado por una arquitectura de micro-servicios dirigido por eventos, garantizando la alta disponibilidad del servicio, aparte de los tres principios mencionados con anterioridad. Dicha aplicación es multiplataforma, por lo que consume la API REST del servicio en la nube, donde cualquier usuario dispone de los resúmenes que deseé.
- **UBUNurse:** Este proyecto consiste en una aplicación multi-dispositivo el cual almacena en la nube registros sobre la evaluación de la atención domiciliaria por parte de dicho personal hacia un determinado paciente. El procedimiento a seguir consiste en elegir un paciente en la lista de la cual dispone el enfermero, posteriormente se elige el test a realizar y por último se obtienen los resultados de la realización de dicho test. Este software pretende automatizar dicho proceso para mejorar la eficiencia del personal sanitario y también para mejorar el proceso de evaluación de cada paciente.
- **Machine learning mediante Microsoft Azure: una aplicación sobre real-state:** En este proyecto se trata más a fondo las herramientas de las cuales dispone *Azure*, las cuales son utilizadas para la creación de elementos de machine learning, los cuales explican el funcionamiento de *Azure* con fines estadísticos. Se utiliza en dicho proyecto *Azure Machine Learning* con los datos empíricos de una inmobiliaria estadounidense para la predicción y posterior clasificación del valor de las viviendas, comparando posteriormente con otros modelos de clasificación y de regresión. A pesar de no ser un proyecto de bases de datos, se le da mucho hincapié a los gráficos y a la muestra de resultados, lo que puede servir de utilidad para la parte final de muestra de resultados.

- **Desarrollo de un Bot en la plataforma *Azure* para ayudar en el aprendizaje del lenguaje de programación C:** Este proyecto consiste en el desarrollo de un bot mediante *Azure* para ayudar al alumnado de 1º curso del Grado en Ingeniería Electrónica y Automática de la Universidad Politécnica de Cartagena al aprendizaje del lenguaje de programación C. Aparte de eso trata también sobre las herramientas de Inteligencia Artificial de *Azure*, como *Luis (Language Understanding)*, aparte de que el software de dicho bot tiene soporte de incorporación de diferentes idiomas, lo que es un factor de gran importancia para el desarrollo de la app. Se ha escogido este trabajo de final de grado puesto que se trata de un proyecto bastante completo en cuanto a contenido a sacar de él y en cuanto a estructuración del contenido del mismo.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] Google. Android studio, 2023. [Página de descarga del instalador de Android Studio].
- [2] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [3] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].