INFORME ACADÉMICO / TÉCNICO

1. Datos Generales

Título del Informe:	Laboratorio1: Configuración, instalación y administración del SDK.
Autor(a):	Mesias Mariscal Bryan Quispe Gabriel Murillo
Carrera:	Ingeniería en Software
Asignatura o Proyecto:	Desarrollo De Aplicaciones Móviles
Tutor o Supervisor:	Doris Karina Chicaiza Angamarca
Institución:	Universidad de las Fuerzas Armadas ESPE – Sede Sangolquí
Fecha de entrega:	12 de octubre del 2025

Índice de Contenido

Contenido

2. Introducción

Este informe abordará la configuración de un entorno de trabajo para el framework Flutter que servirá para el desarrollo de aplicaciones Android, para lo cual se tomará en cuenta la configuración, instalación y administración del SDK, esto se desenvolverá a través de Android Studio como entorno de desarrollo integrado principal.

Android Studio es un IDE que se integra perfectamente con Flutter, un framework multiplataforma de Google que utiliza Dart, lenguaje de programación creado en 2011 y optimizado para interfaces de usuario modernas. Esto incluye un editor de código con autocompletado, depurador, inspector de widgets, emulador de Android.

La implementación de este entorno permitirá aprovechar al máximo las capacidades de Flutter para el desarrollo de aplicaciones móviles modernas. El trabajo abarcará desde la obtención del Flutter SDK y su configuración en el sistema operativo, pasando por la instalación de Android Studio y sus plugins necesarios, hasta la configuración completa del Android SDK.

3. Objetivos

3.1 Objetivo General

Realizar la instalación y configuración del entorno de desarrollo Flutter, empleando Android Studio, con el propósito de desarrollar y ejecutar aplicaciones multiplataforma.

3.2 Objetivos Específicos

1. Instalar Flutter y sus dependencias, siguiendo la documentación oficial y las herramientas necesarias, para disponer de un entorno base funcional.
2. Configurar Android Studio, añadiendo los complementos de Flutter y Dart.

3. Crear y ejecutar un proyecto de prueba en Flutter, utilizando el entorno configurado, para comprobar el correcto funcionamiento del framework.

4. Marco Teórico

4.1 Flutter

Flutter es un framework de código abierto desarrollado por Google para crear aplicaciones mutiplataforma: Android, iOS, web y escritorio. Se presenta como un SDK (kit de desarrollo de software) móvil que utiliza el lenguaje Dart como herramienta principal.

La arquitectura de Flutter se basa en widgets: cada elemento de la interfaz de usuario es un widget anidado dentro de otros, formando un árbol de widgets, esto permite un control fino sobre la interfaz y una experiencia visual consistente en todas las plataformas. Además, Flutter incluye características como la compilación Ahead-Of-Time (AOT) del código Dart a código nativo para lograr un alto rendimiento, así como la compilación Just-In-Time (JIT) (Córdova Arévalo & Rocano Ortega, 2022).

4.2 Dart

Dart es el lenguaje de programación desarrollado por Google que subyace a Flutter. Es un lenguaje orientado a objetos, fuertemente tipado y optimizado para interfaces de usuario dinámicas. Entre sus características destacan la compilación Just-In-Time (JIT) y Ahead-Of-Time (AOT) (Córdova Arévalo & Rocano Ortega, 2022).

4.3 Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial de Android, creado por Google sobre la base de IntelliJ IDE. Está diseñado para el desarrollo de aplicaciones Android, también ofrece soporte completo para Flutter mediante plugins (Flutter, 2023). Cuenta con un sistema de compilación basado en Gradle (Compilador de Java) que automatiza tareas, un

emulador para probar apps en distintos dispositivos, así como un entorno unificado de edición (Google, 2023).

AOT

La compilación AOT transforma el código intermedio en código nativo antes del tiempo de ejecución, eliminando la sobrecarga de compilar durante la ejecución. Según Wade et al. (2017), AOT evita el coste de recopilar perfiles en tiempo real y de compilar dinámicamente, aunque puede no alcanzar la calidad de optimización que permite JIT.

JIT

La compilación JIT convierte código intermedio (bytecode) en código nativo durante la ejecución del programa, focalizándose especialmente en “fragmentos muy usados” para aplicar optimizaciones basadas en el perfil real de ejecución. (Wade et al., 2017)

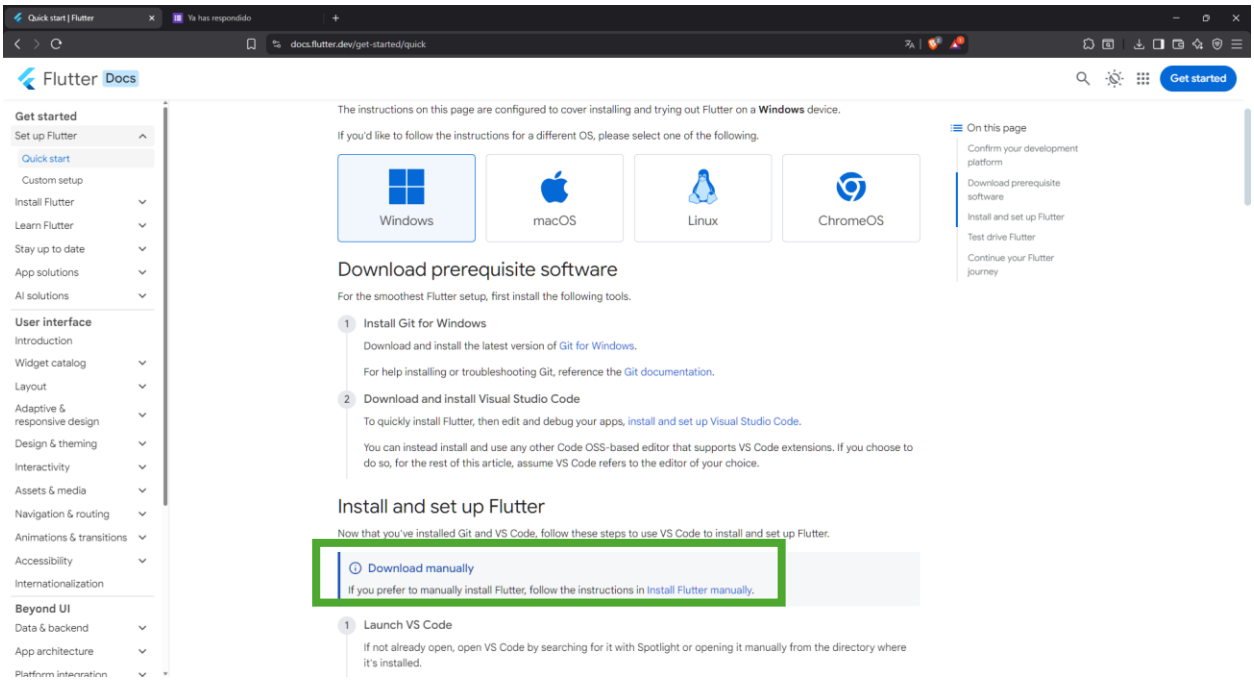
5. Desarrollo

5.1 Descarga de Flutter

Para iniciar con la instalación de Flutter, es necesario acceder a la página oficial de documentación en https://docs.flutter.dev/get-started/quick. Como se observa en la Figura 1, se debe seleccionar el sistema operativo Windows y proceder con la descarga manual del SDK como se señala en la Figura 1.

Figura 1

Selección de Windows y descarga manual de Flutter SDK

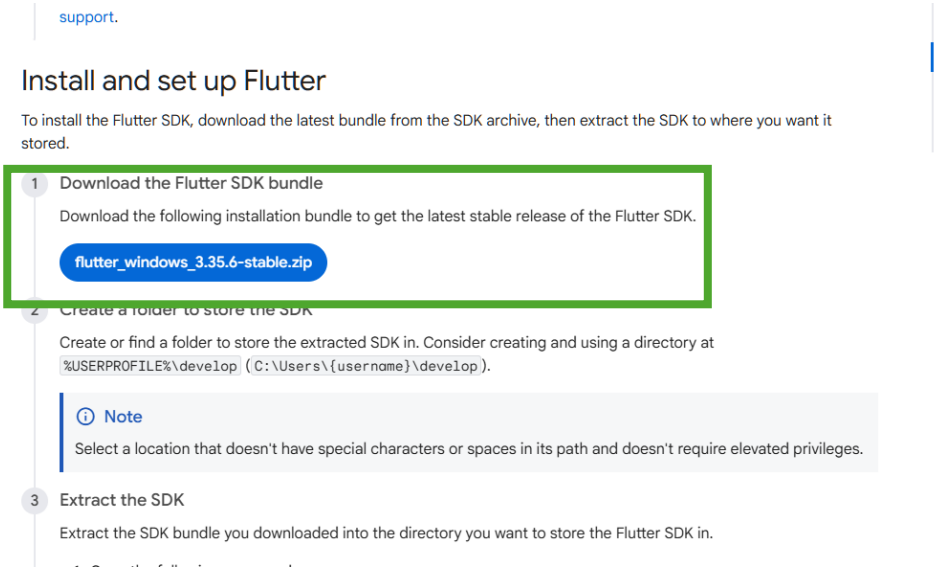


Nota. Adaptado de Flutter Docs (https://docs.flutter.dev/get-started/quick), 2025

Posteriormente, como se muestra en la Figura 2, se debe seleccionar la opción " Download the Flutter SDK bundle" como se señala en la Figura 2.

Figura 2

Selección de la opción Download the Flutter SDK bundle



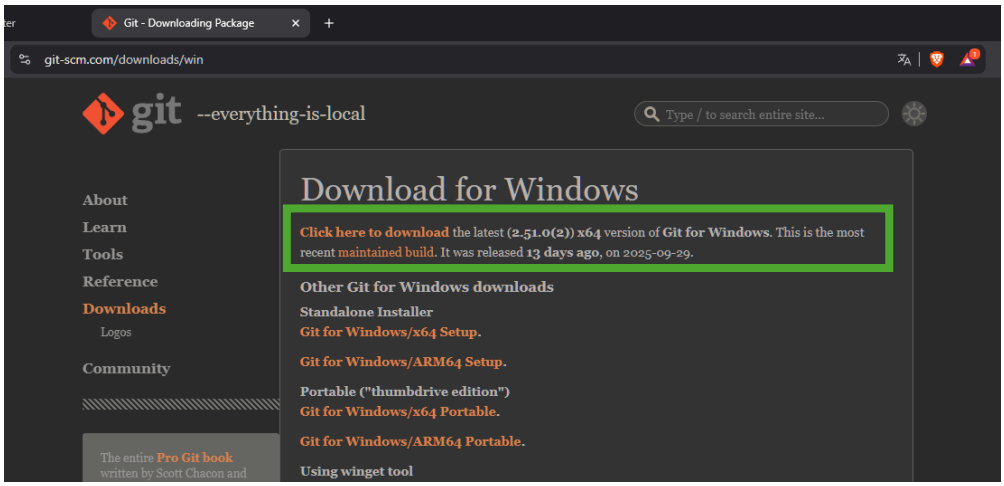
Nota. Adaptado de Flutter Docs (https://docs.flutter.dev/install/manual), 2025.

5.2 Descarga e instalación de Git

Como requisito para el correcto funcionamiento de Flutter es necesario la instalación de Git, para lo cual se accede a la página oficial de descargas en https://git-scm.com/download/win. Como se muestra en la Figura 3, se descarga el instalador para Windows y se ejecuta el archivo .exe para iniciar el asistente de instalación.

Figura 3

Página de descarga de Git para Windows



Nota. Adaptado de la página web Download for Windows de Git, 2025 (https://git-scm.com/download/win)

Se debe continuar con la ejecución del archivo .exe, no hay configuraciones relevantes mas que darle aceptar términos y continuar, finalizado esto se podrá emplear los siguientes pasos

5.3 Configuración del Path y Verificación

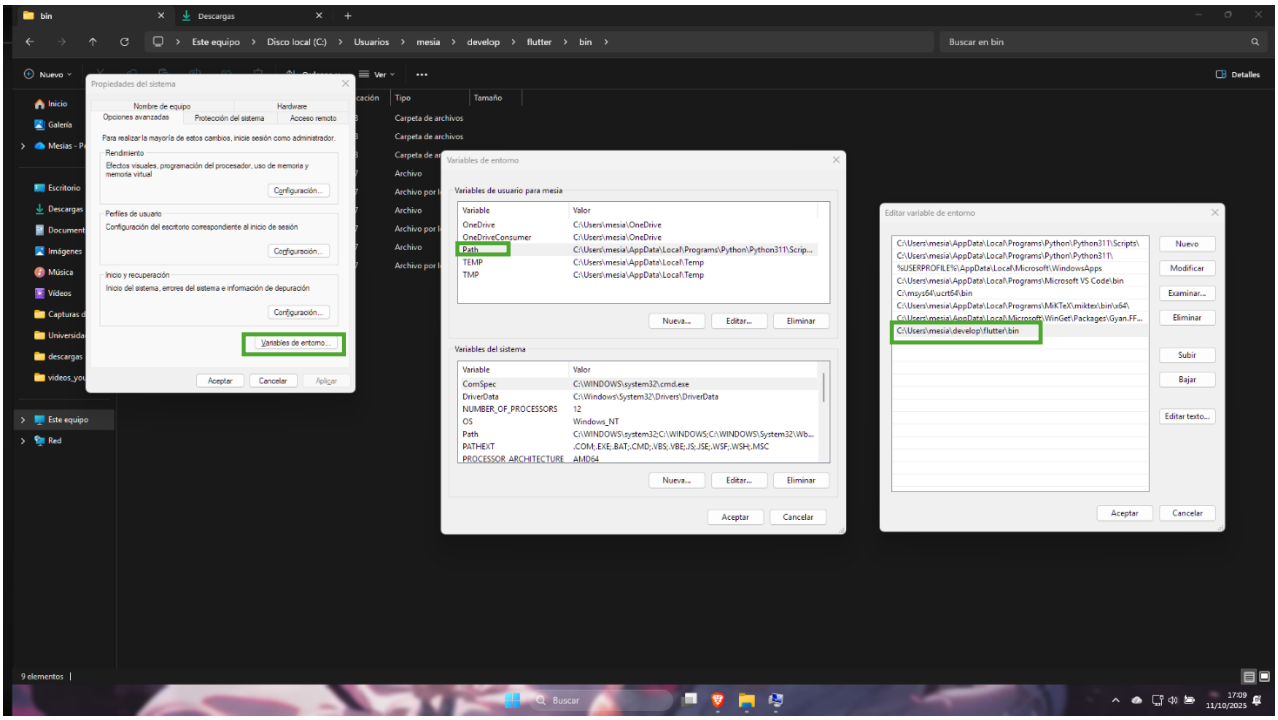
Previo a la configuración, siguiendo la recomendación oficial, se crea la ruta `C:\Users\{username}\develop`. Dentro de este directorio se descomprime el SDK

descargado en la Figura 2, resultando en la ubicación final `C:\Users\{username}\develop\flutter`.

Con el SDK ya ubicado, se añade la ruta de sus binarios (`C:\Users\{username}\develop\flutter\bin`) a las **variables de entorno del usuario**, como se muestra en la Figura 4.

Figura 4

Adición de la Ruta de Flutter a las Variables de Entorno del Usuario



Nota. Captura de pantalla del editor de variables de entorno del sistema operativo. Adaptado de *Windows 11*, por Microsoft, 2025.

Para corrobar la correcta instalación de Flutter se ejecuta los siguientes comandos. `flutter --version` y dart `-version`

Figura 5

Resultado del Comando de Verificación flutter

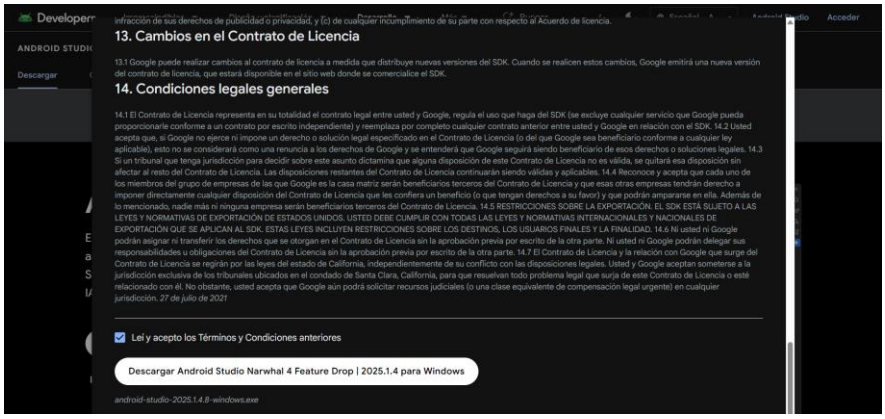
Nota. Resultado generado por la ejecución del comando `flutter --version` y `dart --version` en una terminal de PowerShell en el equipo local.

5.4 Descarga e Instalación de Android Studio

Descargamos Android Studio de la página oficial https://developer.android.com/studio?hl=es-419 y damos aceptar términos

Figura 6

Página de Descarga Oficial de Android Studio

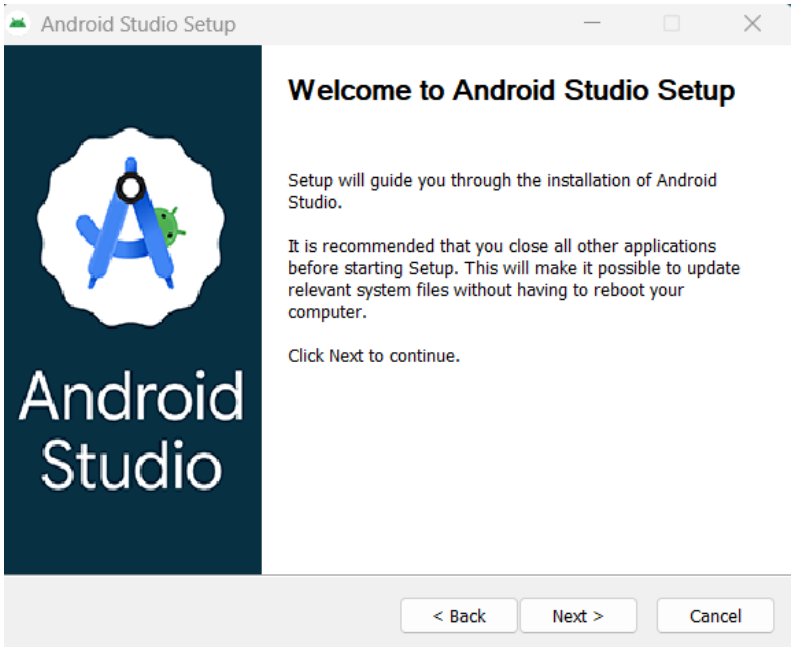


Nota. Adaptado del sitio web oficial *Download Android Studio*, por Google, 2025 (https://developer.android.com/studio).

Instalamos el Android Studio previamente descargado, ejecutando el archivo .ex, aceptando término y condiciones, damos siguiente y siguiente a cada parámetro no es necesario alguna modificación.

Figura 7

Instalación "Standard"



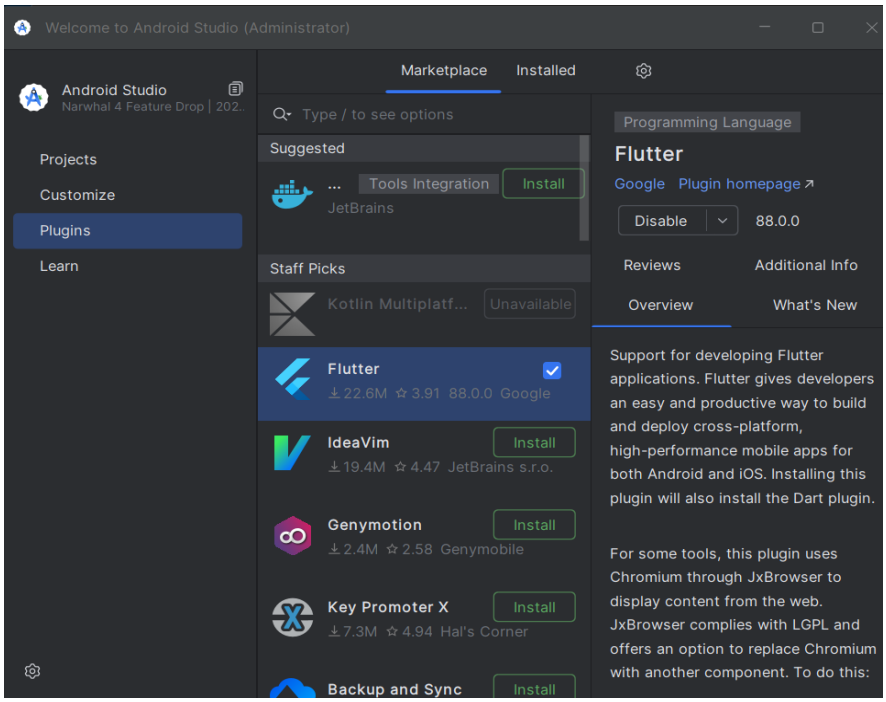
Nota. Captura de pantalla del asistente de instalación. Adaptado de *Android Studio*, por Google, 2025.

5.5 Instalación de Plugins para Flutter en Android Studio

Instalamos el plugin de flutter en Android Studio desde su apartado de plugins y buscamos el que diga Flutter y lo instalamos

Figura 8

Búsqueda e Instalación del Plugin de Flutter



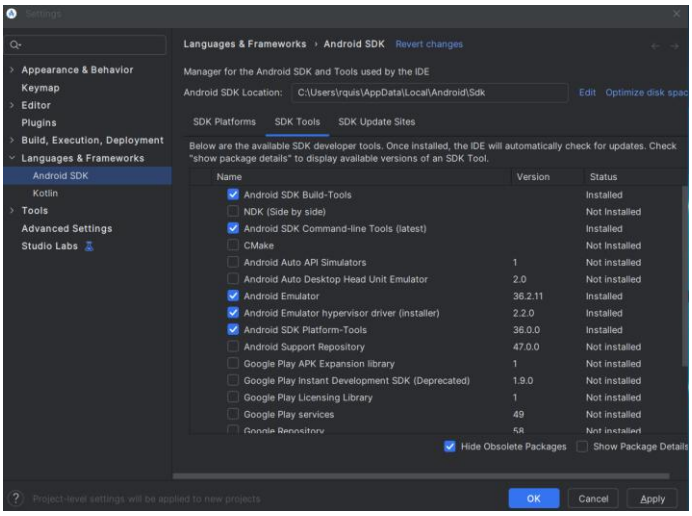
Nota. Captura de pantalla del marketplace de plugins. Adaptado de *Android Studio*, por Google, 2025.

Verificar tener las SDK necesarias sino agregarlas, se debe acceder al **SDK Manager** dentro de la configuración de Android Studio.

Como se muestra en la Figura 9, en la pestaña "SDK Tools" es importante asegurarse de que estén instalados todas las Tools que muestra en la Figura 9. Si no lo están, se deben seleccionar y aplicar los cambios para su descarga.

Figura 9

Verificación de Herramientas en el SDK Manager



Nota. Captura de pantalla del gestor de SDK. Adaptado de *Android Studio*, por Google, 2025.

5.6 Configuración y Ejecución en un Dispositivo Físico (inalámbrico)

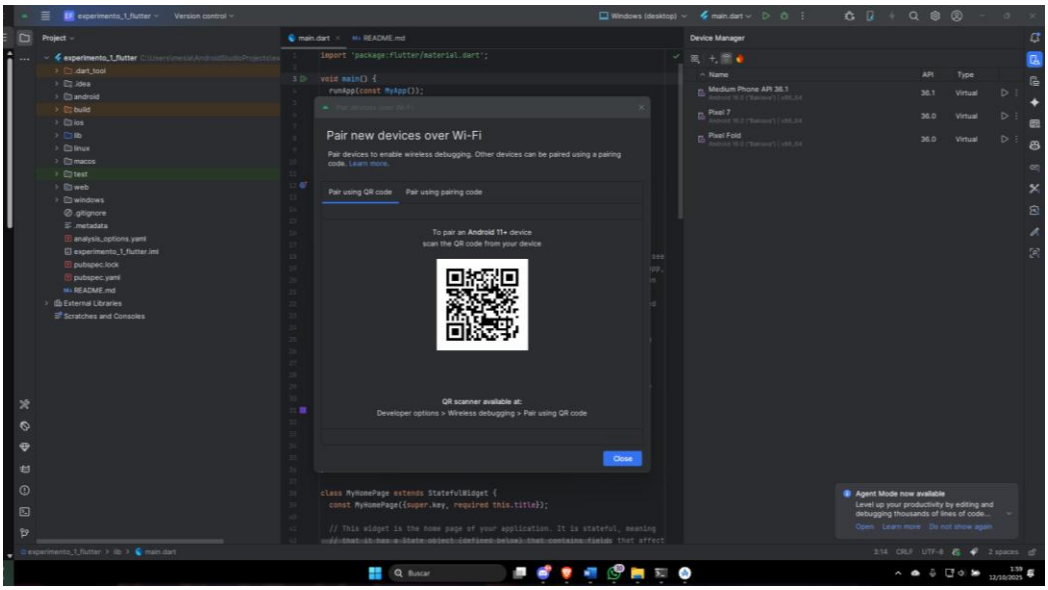
El primer paso es crear un nuevo proyecto de Flutter en Android Studio. Se debe especificar el nombre del proyecto y confirmar que la ruta del SDK de Flutter sea la correcta.

Una vez creado el proyecto, el siguiente paso es conectar un dispositivo físico. Para ello, en el móvil se activan las **Opciones de desarrollador** y la **Depuración inalámbrica** (este método es válido para Android 11 o superior).

A continuación se selecciona Android Studio “Device Manager” luego en el icono de wifi, se selecciona "Vincular usando código QR" para mostrar el código en la pantalla, como se observa en la Figura 10.

Figura 10

Código QR para Vinculación Inalámbrica



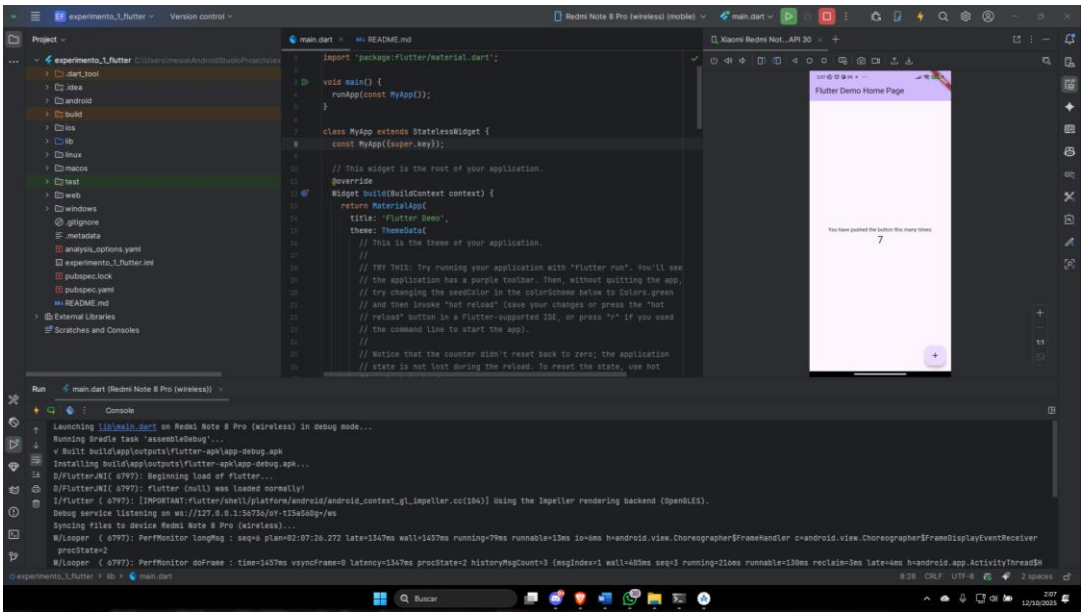
Nota. Captura de pantalla del sistema operativo. Adaptado de *Android Studio*, por Google, 2025.

Una vez se escanea utilizando la opción “Vincular dispositivo con código QR” en Android. Al vincularse exitosamente, se reflejara en Android Studio la su pantalla.

Finalmente, se selecciona el dispositivo recién conectado y se ejecuta el proyecto. La correcta compilación y el despliegue de la aplicación de demostración en la pantalla del móvil, como se observa en la Figura 11, confirman que todo el entorno de desarrollo de Flutter se ha instalado y configurado de manera correcta.

Figura 11

Ejecución Exitosa de la Aplicación por Defecto en el Dispositivo Físico



Nota. Captura de pantalla de la aplicación de demostración de Flutter ejecutándose en un dispositivo Android.

6. Conclusiones

La instalación de Flutter y sus dependencias se completó siguiendo la documentación oficial, permitiendo crear un entorno funcional para el desarrollo. Este informe incluyó la descarga del Flutter SDK desde el sitio oficial, la extracción en una ubicación específica del sistema y la configuración de las variables de entorno necesarias para que el sistema operativo reconociera los comandos de Flutter en la terminal e IDE.

La configuración de Android Studio mediante la instalación de los complementos de Flutter y Dart agilizó el proceso de programación y depuración del proyecto. Los plugins instalados desde la tienda del IDE proporcionaron funcionalidades como el autocompletado inteligente de código en Dart, las herramientas de depuración avanzadas y la integración completa con el emulador de Android.

La creación y ejecución en Flutter comprobó el correcto funcionamiento del entorno. El proyecto inicial se ejecutó sin problemas, validando que todas las dependencias, configuraciones y herramientas estaban correctamente instaladas y comunicándose entre sí de manera eficiente.

7. Recomendaciones

Se recomienda el uso del comando en PowerShell de `flutter doctor`, para verificar con más detalle si algún error o advertencia pueda afectar el uso de flutter y dart a futuro, además de que muestra si esta listo para usarse en Visual Studio Code o Android Studio, de esta manera dando mas seguridad antes de entrar a configurar Android Studio con los plugis y tolos .

Para mejorar el rendimiento se recomienda optar la funcionalidad de “Hot Reload”, permite visualizar cambios en la interfaz de usuario de forma casi instantánea, lo que reduce drásticamente los tiempos de compilación y acelera significativamente los ciclos de desarrollo y depuración.

Es recomendable usar un dispositivo físico externo como se usa en este informe debido a que no consume tanto recurso al no intentar emular un dispositivo, lo cual menora la carga en ejecutar código, sin embargo de tener los recursos suficientes en hardware se puede optar por una emulación dentro del IDE mismo.

8. Referencias Bibliográficas

Córdova Arévalo, J. F., & Rocano Ortega, T. S. (2022). *Desarrollo e implementación de una aplicación móvil para el control de rutas de motorizados monitoreados en tiempo real* [Trabajo de titulación, Universidad Politécnica Salesiana]. Repositorio Institucional UPS.

Flutter. (2023). *Android Studio and IntelliJ*. Flutter documentation. https://docs.flutter.dev/tools/android-studio

Google. (2023). *Introducción a Android Studio*. Android Developers. https://developer.android.com/studio/intro?hl=es-419

Wade, A. W., Kulkarni, P. A., & Jantz, M. R. (2017). *AOT vs. JIT: Impact of profile data on code quality*. En *Proceedings of the 18th ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES 2017)*. https://doi.org/10.1145/3140582.3081037