

Flutter

Curso práctico

**Crea Apps para iOS, Android,
Web y Escritorio**



www.ra-ma.es

Desde www.ra-ma.es podrá
descargar material adicional.

Luis Ayala



Ra-Ma®

edü

Ediciones de la U

Flutter

Curso práctico

Crea Apps para iOS, Android,
Web y Escritorio

Luis Ayala



Ra-Ma®

edü®
Ediciones de la U

Ayala, Luis; *et al*

Flutter. Curso práctico. Crea Apps para iOS, Android, Web y Escritorio / Luis Ayala.

Bogotá: Ediciones de la U, 2026

308 p. ; 24 cm.

ISBN 978-628-7912-41-0 ePUB 978-628-7912-43-4 PDF 978-628-7912-42-7

1. Informática 2. Programación web 3. Computadores I. Tít.

519.7 cd. - Thema: UMW

Edición original publicada por © Editorial Ra-Ma (España)

Edición autorizada a Ediciones de la U para Colombia

Área: Informática

Primera edición: Bogotá, Colombia, julio de 2026

ISBN 978-628-7912-41-0

© Luis Ayala

© Ra-ma Editorial

Calle Jarama, 3-A (Polígono Industrial Igarsa) 28860 Paracuellos de Jarama

www.ra-ma.es y www.ra-ma.com / E-mail: editorial @ra-ma.com

Madrid, España

© Ediciones de la U - Carrera 27 #27-43 - Tel. (+57) 601 6455049

www.edicionesdelau.com - E-mail: editor@edicionesdelau.com

Bogotá, Colombia

Ediciones de la U es una empresa editorial que, con una visión moderna y estratégica de las tecnologías, desarrolla, promueve, distribuye y comercializa contenidos, herramientas de formación, libros técnicos y profesionales, e-books, e-learning o aprendizaje en línea, realizados por autores con amplia experiencia en las diferentes áreas profesionales e investigativas, para brindar a nuestros usuarios soluciones útiles y prácticas que contribuyan al dominio de sus campos de trabajo y a su mejor desempeño en un mundo global, cambiante y cada vez más competitivo.

Coordinación editorial: Daniela Parra G.

Carátula: Ediciones de la U

Impresión: DGP Editores SAS

Calle 63 No. 70 D - 34, Pbx. (571) 7217756

Impreso y hecho en Colombia

Printed and made in Colombia

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro y otros medios, sin el permiso previo y por escrito de los titulares del Copyright.

*Al profesor Edgardo Caballero
por sus enseñanzas de programación y
apoyo en mis proyectos.*

100 PROYECTOS CON FLUTTER

Accede a un paquete exclusivo con 100 proyectos totalmente ejecutables con Flutter, diseñados para que aprendas desarrollando aplicaciones modernas, funcionales y listas para el mundo profesional por 9,99€ a través de la web del libro.

¿Por qué elegir este paquete?

- Aprende haciendo: cada proyecto está pensado para desarrollar habilidades reales.
- Domina interfaces modernas, integración con APIs, bases de datos, Firebase y autenticación.
- Ahorra cientos de horas de búsqueda con ejemplos estructurados y listos para ejecutar.
- Ideal para estudiantes, desarrolladores que quieren especializarse en desarrollo móvil.
- Tendrá proyectos actualizados hechos con las últimas versiones de Flutter y Android Studio.

Qué encontrarás:

- Aplicaciones completas y funcionales.
- Código organizado siguiendo buenas prácticas.
- Ejemplos progresivos: desde apps básicas hasta desarrollos avanzados.
- Proyectos multiplataforma compatibles con iOS, Android, Windows, macOS y Web.

Adquiere tu paquete en este enlace.



ÍNDICE

100 PROYECTOS CON FLUTTER.....	6
ACERCA DEL AUTOR	9
INTRODUCCIÓN	10
OBJETIVOS DEL LIBRO.....	11
CAPÍTULO 1. PREPARANDO EL ENTORNO DE DESARROLLO.....	13
CAPÍTULO 2. CONFIGURAR ANDROID STUDIO.....	23
CAPÍTULO 3. CONOCIENDO ANDROID STUDIO, FLUTTER Y DART.....	30
CAPÍTULO 4. COMENTARIOS EN FLUTTER.....	42
CAPÍTULO 5. ERRORES Y ADVERTENCIAS O WARNINGS.....	45
CAPÍTULO 6. ANALIZANDO EL PROYECTO DEMO DE FLUTTER	60
CAPÍTULO 7. ARCHIVOS DE PROYECTO FLUTTER.....	67
CAPÍTULO 8. MÁS SOBRE EL IDE.....	86
CAPÍTULO 9. INTRODUCCIÓN A LOS WIDGETS.....	130
CAPÍTULO 10. LENGUAJE DART.....	149
CAPÍTULO 11. MÁS SOBRE LOS WIDGETS	169
CAPÍTULO 12. ESTILO: MATERIAL DESIGN Y CUPERTINO EN FLUTTER	248

CAPÍTULO 13. INTERACCIÓN Y NAVEGACIÓN.....	251
CAPÍTULO 14. GESTIÓN CON PROVIDERS.....	256
CAPÍTULO 15. PROYECTOS EN FLUTTER	281
CAPÍTULO 16. PUBLICACIÓN DE APP EN GOOGLE PLAY Y APP STORE	299
16.1 PREPARACIÓN PARA GOOGLE PLAY (ANDROID).....	300
16.2 PREPARACIÓN PARA APP STORE (IOS)	301
16.3 ASO: OPTIMIZACIÓN PARA QUE TU APP SEA ENCONTRADA	302
16.4 REGLAS A TOMAR EN CUENTA	303
16.5 CONSEJOS DE ORO PARA EVITAR EL RECHAZO	304
CONCLUSIÓN	305
¿QUE SIGUE DESPUÉS DE ESTUDIAR ESTE LIBRO?	306
MATERIAL ADICIONAL	307
100 PROYECTOS CON FLUTTER	308



ACERCA DEL AUTOR

Luis Alonso Ayala Ponce, es programador de Apps con tecnología Flutter, tiene publicadas 27 apps en Google Play y App Store.

Es escritor de libros de programación Dart, Flutter, Inteligencia Artificial y de otros temas.

Reflexión

“La vida es un bucle finito, algún día todo terminará”.

Luis Ayala

INTRODUCCIÓN

Flutter es un software para desarrollo de aplicaciones con único código base para; iOS, Android, Windows, macOS, Linux y web.

El lenguaje de Flutter es Dart y el entorno de desarrollo es Android Studio, utilizado en este libro. Hay otros IDEs para Flutter, como Visual Studio Code. Si el lector prefiere, puede hacer los ejercicios en ese entorno.

Flutter, su lenguaje Dart y Android Studio son propiedad de Google, formando un ecosistema con el soporte ideal para las tareas de programación.

Flutter fue presentado en 2015 con el nombre Sky.

El 4 de diciembre del 2018 se presentó en Londres la versión de Flutter 1.0, la 2.0 se publicó en 2021 y en 2022 la versión 3.0.

La versión 3.38.7 fue publicada en enero del 2026.

Dart, el lenguaje de Flutter fue creado por Google y presentado en octubre del 2011 en Dinamarca. Es un lenguaje de programación orientado a objetos, la versión 1.0 fue publicada en noviembre del 2013, la 2.0 en 2018 y la última versión es la **3.10** publicada el 12 de noviembre del 2025.

Tiene una sintaxis parecida a C#, java y JavaScript. Y como lenguaje de POO tiene los pilares de todo lenguaje de esta categoría como ser: Herencia, Abstracción, Polimorfismo y Encapsulación.

Hay un trinomio genial para el desarrollo fácil de aplicaciones multiplataforma que son Flutter con su lenguaje Dart y el entorno de desarrollo integrado **Android Studio**.

Además de esto, se cuenta con valiosas herramientas de apoyo logístico para facilitar la programación como **dartpad.dev** el editor online para Dart, útil para

probar código Dart, **pub.dev** que proporciona paquetes necesarios en el desarrollo de muchas apps.

Así que no te has equivocado, estás en el camino correcto para aprender o reforzar tus habilidades en el desarrollo de apps para Android, iOS, Windows, macOS, Linux y Web.

¡Adelante!

OBJETIVOS DEL LIBRO

El contenido de este libro está preparado para enseñar a desarrollar apps para dispositivos con sistema operativo Android, iOS, Windows, macOS, Linux y la Web.

Forma parte de una serie de 8 libros del mismo autor, sobre tecnologías Dart, Flutter y Android Studio.

1. Dart: programe fácil.
2. Dart: aprende por ejemplos.
3. Programe en Dart: preguntas y respuestas.
4. Dart a Flutter.
5. **Flutter para todos.**
6. Flutter: aprende por ejemplos.
7. Programe en Dart: preguntas y respuestas.
8. Android Studio para Flutter.

Cada libro aun perteneciendo a una serie es independiente de los otros, por lo que se puede adquirir y leer en cualquier orden.

¿A quién va dirigido?

Personas que deseen aprender a desarrollar aplicaciones multiplataformas, es decir, para diversos sistemas operativos como los mencionados en el párrafo anterior.

Los que no conocen la plataforma obtendrán el conocimiento desde cero hasta nivel principiante y los expertos encontraran cosas nuevas e interesantes porque Flutter, su lenguaje Dart y Android Studio están actualizándose constantemente.

Este libro capta los últimos cambios en estas tecnologías.

Se maneja un lenguaje sencillo que puede entenderlo desde niños de temprana edad que ya sepan leer hasta el adulto mayor que nunca haya programado y en su jubilación desee ponerle más encanto a su vida haciendo apps.

¿Cómo utilizar este libro para maximizar el aprendizaje?

Lee los conceptos teóricos, analiza el código de los ejemplos, haz los ejercicios.

Puedes leer o estudiar los temas en cualquier lugar que estés, incluso en receso en tu trabajo o mientras viajas.

Hazte preguntas y responde sobre las explicaciones y sobre los ejemplos.

Trata de memorizar y explicar con tus propias palabras conceptos fundamentales.

Si estas en casa o en cualquier lugar donde tengas una computadora disponible y configurada para programar en Flutter, haz los ejercicios y ejemplos.

Adquiere y lee los otros libros de la serie para reforzar tus conocimientos teóricos y prácticos sobre el desarrollo de apps.

La constancia en lectura y práctica es la base de tu avance en tu camino en el aprendizaje para desarrollo de apps en Flutter.

1

PREPARANDO EL ENTORNO DE DESARROLLO

Objetivo

Conocer e instalar el hardware, software y herramientas necesarias para programar en macOS y Windows.

Introducción

En este capítulo conocerás el hardware, software y herramientas necesarias para programar en Flutter, es decir el equipo y programas necesarios para el desarrollo de apps en Flutter.

Si vas a desarrollar apps para Android debes hacerlo en computadora con Windows. Si es para iPhone, debe ser en computadora macOS.

Al desarrollar, aplicaciones para iPhone, las pruebas debes hacerlas en móviles iPhone y para Android será en dispositivos Android.

Debes tener conexión a internet para descargar los programas necesarios para el desarrollo de apps.

Programar en Windows

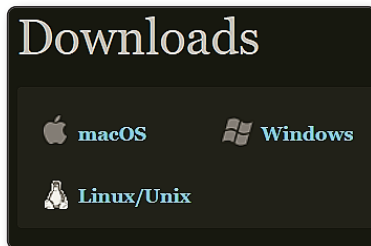
1. Computadora Windows.
2. Sistema operativo Windows 10 ó posterior, x86_64 CPU Cores.

3. Tener instalada la herramienta Windows PowerShell 5, ya viene incluida en Windows 10 y posterior.
4. Tener instalado el Git para Windows 2.27 o posterior.
Para saber si tienes el Git en tu maquina ejecuta en la consola el comando.
Git --versión.
Si está instalado mostrará la versión.

Git versión 2.52.0.windows.1

Si no está instalado, ve a la página de descarga y selecciona macOS o Windows según tu computadora.

<https://git-scm.com/downloads>.

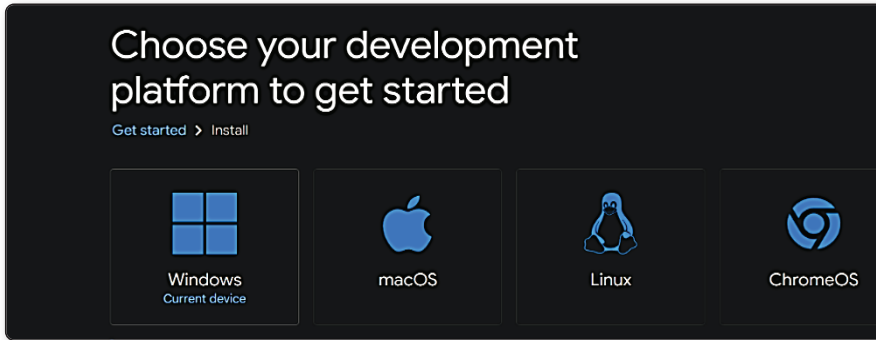


Descargarlo y seguir las instrucciones para instalarlo.

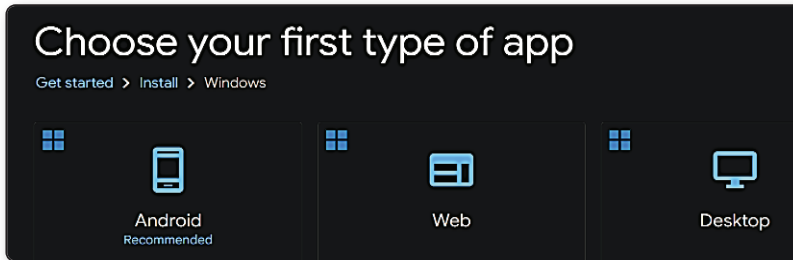
5. Memoria disponible en disco, mínimo 11 GB, ideal 60 GB.
6. Memoria RAM mínima 8 GB, ideal 16 GB o más.
7. Dispositivos móviles físicos Android (teléfonos) con diferentes modelos, tamaño y resolución de pantallas, para probar las apps.
8. Resolución de pantalla en pixeles: fHD (1920 x 1080) ideal y WXGA (1366 x 768).
9. Descargar Android Studio para Windows, ultima version desde la página de Android.
<https://developer.android.com/studio?hl=es-419>.
10. Descargar Flutter para Windows desde la página de Flutter.
https://storage.googleapis.com/flutter_infra_release/releases/stable/windows/flutter_windows_3.29.1-stable.zip.

Encontraras varias opciones que debes elegir según tu máquina y si vas a programar en Windows o macOS.

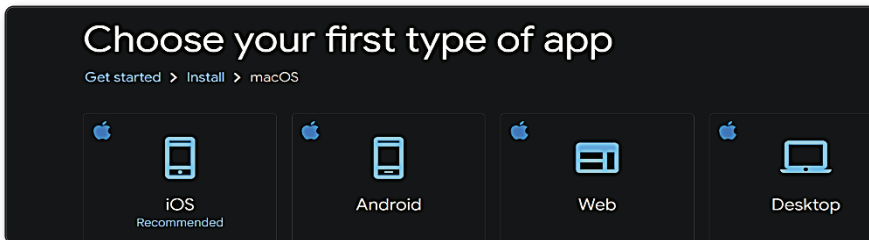
Escoge aquí el sistema operativo en que vas a desarrollar.



Si son Apps para móviles Android elija Android.



Seleccione uno u otro dispositivo.



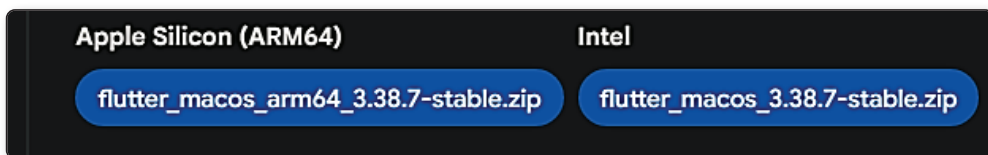
Recuerda para iOS programar en macOS, para Android programar en Windows.

Si programas en Windows descarga la última versión de Flutter para Windows.

Al momento de la actualización de este libro la versión para Windows es:

flutter_windows_3.38.7-stable.zip

Si programas en macOS descarga la última versión de Flutter para macOS.



Instalación de Flutter

Sigue los pasos que muestra el asistente para la instalación de Flutter.

Instalación de Dart

No es necesario descargar Dart por separado porque Flutter ya lo incluye. Solo descargaras los Plugin Dart y Flutter desde Android Studio.

Instalación de Android Studio

Lo haces desde este sitio: <https://developer.android.com/studio?hl=es-419>

Te mostrará una pantalla donde elegirás la descarga recomendada para Windows o para MAC

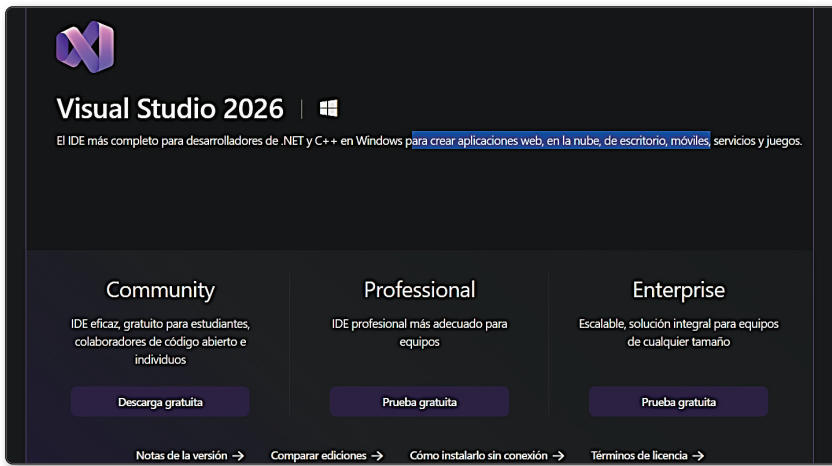
Descargas de Android Studio			
Descarga la versión más reciente de Android Studio. Para obtener más información, consulta las notas de la versión de Android Studio .			
Plataforma	Paquete de Android Studio	Tamaño	Suma de comprobación SHA-256
Windows (64 bits)	android-studio-2025.2.2.8-windows.exe Recomendado	1.4 GB	3acc0587882f81281714098e440673f44de2ecaf75e9f005b334ec5fd2d29da8
Windows (64 bits)	android-studio-2025.2.2.8-windows.zip Sin instalador .exe	1.5 GB	d63603d6bbc433484b4059060319cc116a50c9bdf0d37610dd83ac68d77a1294
Mac (64 bits)	android-studio-2025.2.2.8-mac.dmg	1.5 GB	aa1bb1027fe0999b3154a9f017c30eaa7fe56b59afaf7640e9ff72a361ab047f

Más adelante hay más explicaciones acerca de Android Studio, el entorno de desarrollo integrado que usamos en este libro.

Instalación de Visual Studio Community 2026

Es necesario tener instalado este software para crear aplicaciones web, en la nube, de escritorio y móviles. A continuación el link de descarga.

<https://visualstudio.microsoft.com/es/downloads/>



Instalación del JDK de java

Descargar JDK de java desde la página de Oracle.

<https://www.oracle.com/java/technologies/downloads/#jdk25-windows>
Programar en.

Lo llevará a otra pantalla en donde busca la versión para Windows o Mac:

<https://www.oracle.com/java/technologies/downloads/#jdk25-windows>

Java 25, Java 21, and earlier versions available now

JDK 25 is the latest *Long-Term Support (LTS)* release of the Java SE Platform. [Learn about Java SE Subscription](#)

JDK 21 is the previous *Long-Term Support (LTS)* release of the Java SE Platform.

Earlier JDK versions are available below.

JDK 25 JDK 21

Java SE Development Kit 25.0.1 downloads

JDK 25 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC).

JDK 25 will receive updates under the NFTC, until September 2028, a year after the release of the next LTS. Subsequent JDK 25 updates will be licensed under the Java SE OTN License (OTN) and production use beyond the limited free grants of the OTN license will require a fee.

Linux macOS Windows

Product/file description	File size	Download
x64 Compressed Archive	204.33 MB	https://download.oracle.com/java/25/latest/jdk-25_windows-x64_bin.zip (sha256)
x64 Installer	182.29 MB	https://download.oracle.com/java/25/latest/jdk-25_windows-x64_bin.exe (sha256)
x64 MSI Installer	181.05 MB	https://download.oracle.com/java/25/latest/jdk-25_windows-x64_bin.msi (sha256)

Instalaciones en Mac

1. Computadora macOS.
2. Sistema operativo macOS 11 (Big Sur) o posterior.
3. Memoria disponible en disco, mínimo 44 GB, ideal 70 GB.
4. CPU Cores, ideal 8, mínimo 4.
5. Instalar Rosseta 2 ejecutando el siguiente comando.
6. Sudo softwareupdate --install-rosetta --agree-to-license.
7. Instalar Xcode 16 o última versión.
8. Instalar CocoaPods 1.16 o última versión.
9. Acceso a Internet.
10. Descargar Android Studio para macOS, última versión desde la página de Android.
11. Descargar Flutter para macOS desde la página de Flutter.
<https://docs.flutter.dev/get-started/install/macos/mobile-ios>.

La descarga será de acuerdo al procesador que tenga tu máquina.

Configuración de Flutter para iOS en macOS

Luego de la descarga, para la instalación sigue los pasos correspondientes que se encuentran en la documentación de Flutter.

En el siguiente enlace encontraras las instrucciones más detalladas.

<https://docs.flutter.dev/get-started/install/macos/mobile-ios>.

Acciones para instalación y configuración

1. Instala los programas descargados si es que aún no lo has hecho.
Para Windows: android Studio, Flutter, Visual Studio Community 2022, JDK de Java.
Para macOS: android Studio, Flutter, Rosetta, Xcode, CocoaPods.
Busca siempre las últimas versiones.
2. **Configurar la variable de entorno JAVA_HOME:**
 - Haz clic derecho en el botón “Inicio” y selecciona “Sistema”.
 - En el menú que se muestra, haz clic en “Configuración avanzada del sistema”.
 - En la ventana “Propiedades del sistema”, haz clic en el botón “Variables de entorno”.

- En la sección “Variables del sistema”, verifica si existe la variable JAVA_HOME. Si no existe, créala:
 - Haz clic en el botón “Nueva”.
 - En “Nombre de la variable”, escribe JAVA_HOME.
 - En “Valor de la variable”, escribe la ruta de instalación del JDK (por ejemplo: c:\Program Files\Java\jdk-17).
 - Haz clic en “Aceptar”.

3. Configurar la variable de entorno Path:

- En la sección “Variables del sistema” de la ventana “Variables de entorno”, busca la variable Path.
- Haz clic en “Editar”.
- Haz clic en “Nuevo” y agrega la ruta de la carpeta bin del SDK de Flutter (por ejemplo: c:\flutter\bin).

Nota

Debes ir a la carpeta donde instalaste Flutter y allí buscas el bin copias la ruta del bin y eso es lo que copiaras en la variable Path, no tienes que borrar las variables que ya están, solo debes poner punto y coma “;” al final de la última variable y luego agrega la ruta del bin de Flutter.

- Haz clic en “Aceptar” en todas las ventanas abiertas para guardar los cambios.

4. Ejecutar flutter doctor:

- Abre la consola de PowerShell.
- Ejecuta el comando flutter doctor en la consola para verificar la instalación de Flutter y sus dependencias.
Flutter doctor.
El doctor hará un diagnóstico.

```
git version 2.32.0.windows.1
PS C:\Users\luisa> flutter doctor
Downloading package sky_engine... 397ms
Downloading package flutter_gpu... 104ms
Downloading flutter_patched_sdk tools... 614ms
Downloading flutter_patched_sdk_product tools... 425ms
Downloading windows-x64 tools... 2.917ms
Downloading windows-x64/font-subset tools... 201ms
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.38.7, on Microsoft Windows [Version 10.0.26200.7462], locale es-HN)
[✓] Windows Version (11 Home Single Language 64-bit, 25H2, 2009)
[✓] Android toolchain - develop for Android devices (Android SDK version 36.1.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.7.4)
[✓] Connected device (3 available)
[✓] Network resources

• No issues found!
PS C:\Users\luisa>
```

El doctor Flutter te dará un reporte del estado de todo tu entorno de desarrollo. En el caso que muestra la imagen, no hay errores, todo está instalado y configurado correctamente.

- Si el informe del doctor indica que faltan herramientas o acciones por realizar, sigue las instrucciones para completar la instalación.

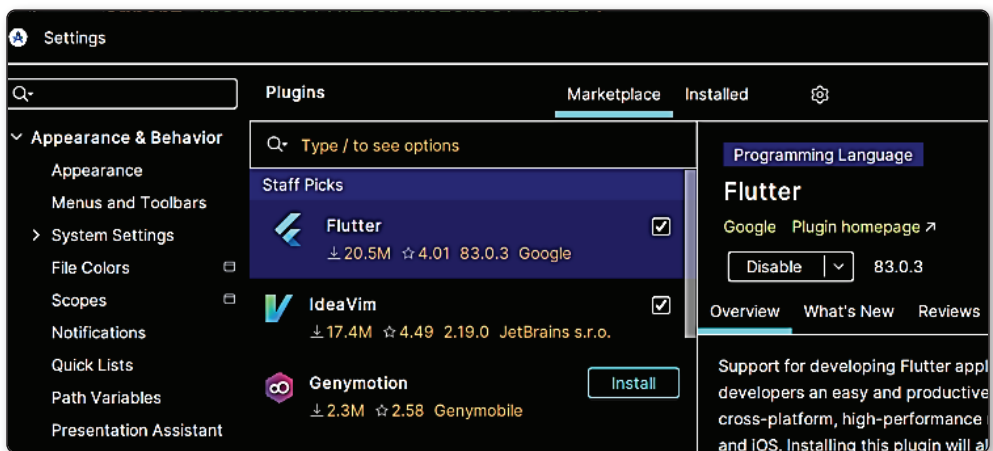
5. Instalar los plugin Flutter y Dart en Android Studio:

Es requisito para instalar los Plugin Dart y Flutter tener instalado Android Studio. Flutter, Visual Studio 2022 o 2026

- Abre Android Studio.



- Ve a File > Settings > Plugins.
- Busca los plugins “Flutter” y “Dart” en el Marketplace e instálalos.



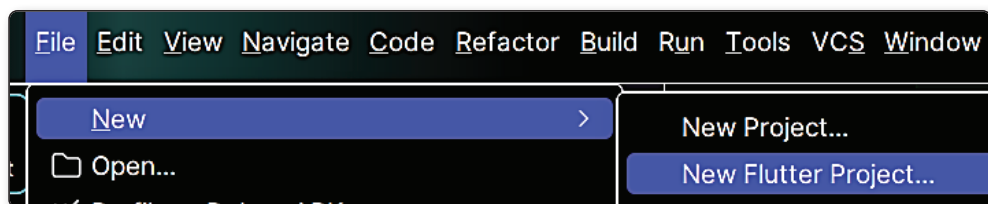
Los plugins de Dart y Flutter debes agregarlos cuando ya instalaste Android Studio y te vas a File, Setting, Plugins, Marketplace y luego directamente o en el buscador encontrar los plugins Dart y Flutter y luego los instalas.

En macOS la ruta para instalar Plugins es File→preferences→Plugin→luego buscar los plugins Dart y Flutter.

Finalizados todos los procedimientos de instalación indicados, tu máquina debería estar lista para actividades de programación, compruébalo con el comando flutter doctor en la consola y también verificar si aparece la opción para desarrollo de proyectos flutter en el menú File, New, New Flutter Project.

6. Verificar la instalación:

Si las herramientas están bien instaladas al abrir el menú File o archivo, podrás ver al dar clic en New, la opción New Flutter Project.



Es la opción para crear nuevo proyecto de Flutter.

Instalación en macOS

Las herramientas para desarrollo en macOS varían un poco en relación a Windows, no obstante todo es intuitivo y las instrucciones fácil de seguir.

Para esta tarea te recomiendo sigas las instrucciones directamente de la documentación oficial, es más recomendable porque las formas y herramientas se cambian y actualizan con frecuencia.

Resumen

En este capítulo, se mostró cómo instalar y configurar las herramientas necesarias para desarrollar aplicaciones Flutter en Android Studio, tanto en Windows como en macOS. Se explicó cómo verificar la instalación de Flutter y sus dependencias con el comando flutter doctor y se enseñó a instalar los plugins necesarios en Android Studio. Además, se instruyó para la instalación de Visual Studio Community 2026 o 2022 para el desarrollo en Windows.

Ya sabes que para programar para iPhone se necesita computadora macOS y para móviles Android se necesita computadora Windows.

Preguntas

1. ¿Cuáles son las herramientas esenciales para desarrollar aplicaciones Flutter en Windows?
2. ¿Cuáles son las herramientas esenciales para desarrollar aplicaciones Flutter en macOS?
3. ¿Cómo se instala el SDK de Flutter en Windows?
4. ¿Cómo se configura la variable de entorno JAVA_HOME?
5. ¿Cómo configurar la variable entorno PATH?
6. ¿Cómo se verifica la instalación de Flutter y sus dependencias?
7. ¿Por qué es necesario instalar Visual Studio 2022 o 2026 para desarrollar aplicaciones Flutter en Windows?
8. ¿Qué diferencia hay entre las herramientas para programar en iOS y Android?
9. ¿Para qué sirve el comando flutter doctor?

Ejercicios

1. Instala Android Studio, Visual Studio community 2022 o 2026 (solo en Windows), el SDK de Flutter y las herramientas necesarias en tu sistema operativo.
2. Ejecuta el comando **flutter doctor** y verifica que todas las dependencias estén instaladas correctamente.
3. Crea un nuevo proyecto Flutter en Android Studio.
4. Usando un comando en la consola revisa si tienes instalado el Git para Windows.
5. Revisa qué variables de entorno hay instaladas y comprueba si están las variables de entorno PATH con la ruta del bin de Flutter y la variable de entorno JAVA_HOME con la variable del JDK.

2

CONFIGURAR ANDROID STUDIO

Objetivos

1. Continuar configuración del entorno de desarrollo.
2. Instalar las herramientas necesarias para desarrollo.
3. Crear emuladores en Device Manager.
4. **Configurar teléfonos reales para desarrollo.**

Introducción

Las herramientas básicas para programar en Android Studio son: el software Android Studio que nos proporcionará el IDE (Integrated Development Environment) o entorno de desarrollo integrado, el JDK de java (Java Development kit) o Kit de Desarrollo de Java, el SDK Flutter que incluye el lenguaje Dart. Además se necesita para programar en Windows tener instalado Visual Studio Community 2022 o 2026.

Hacer algunas configuraciones en el equipo y tener conexión a internet para descargar las herramientas de soporte de Android Studio, así como las frecuentes actualizaciones.

Se ocupa computadora macOS para programar para iPhone y computadora Windows para hacerlo para Android.

Probar o ejecutar la aplicación

Las aplicaciones se pueden probar en el mismo emulador, en el navegador Edge o Chrome o en un dispositivo real.

Configurar el dispositivo móvil para probar aplicaciones.

En la configuración de tu teléfono. Habilita Opciones de desarrollador en Seguridad.



También debe estar habilitada la Depuración de USB.



En algunos teléfonos, las opciones de desarrollador están ocultas. Por lo que debes dar clic 7 veces sobre el número de compilador del teléfono para que se muestre; el cual se encuentra en la sección Seguridad en la configuración de tu móvil.

Algunos modelos de móviles ya traen habilitados el modo desarrollador.

Si aun así tu dispositivo no es visible en el IDE, deberás revisar en tu teléfono, las opciones de seguridad y habilitar la que permita la depuración e instalación de app externas.

Configuración del Emulador, o simulador de móvil

Los proyectos en Flutter se pueden probar en dispositivos móviles, en emuladores que la plataforma trae entre sus paquetes o en emuladores externos. También en navegadores Chrome, Edge y Windows para escritorio.

Para usar los emuladores de Android estudio y Flutter es necesario que el pc tenga tecnología de virtualización. Condición que las computadoras modernas ya traen.

Para usar emuladores externos (es decir software que hay para servir como emuladores) es necesario configurar la variable `ANDROID_HOME`, además de tecnología de virtualización (VT).

Alternativa a lo anterior es usar los dispositivos móviles, se recomienda que tengas varios modelos de teléfonos de baja, media y alta gama. Si una app funciona en un teléfono de baja gama es casi seguro que lo hará también en uno de media o alta gama. Por eso debes asegurarte de que en el código fuente y recursos de la app se apunte a las API (Application Programming Interfaz) correspondientes para modelos antiguos y nuevos.

Para crear un emulador para depuración o prueba de proyectos, debes conocer acerca del `DEVICE MANAGER`, antes conocido como AVD o Android Virtual Device.

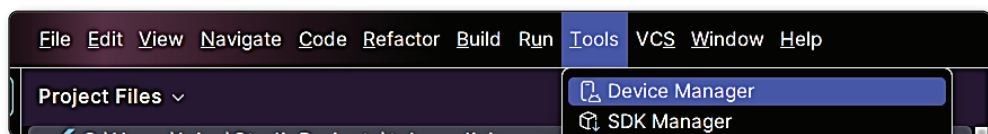
Device Manager (Dispositivo Virtual de Android) proporciona una interfaz gráfica, en la cual se crean y manejan dispositivos virtuales similares a los reales y que son requeridos por el emulador de Android.

Son indispensables para probar los proyectos y ver como estos se verían una vez terminados.

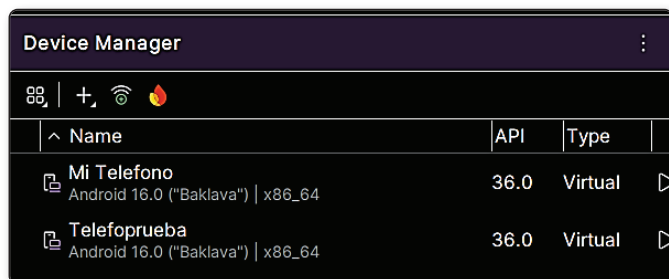
El `DEVICE MANAGER` antes llamado AVD permite crear varios dispositivos con diferente tipo, tamaño y resolución de pantalla.

Para abrir el `DEVICE MANAGER` hay varias opciones:

Clic en el icono con forma de rejilla en la esquina superior izquierda, (es el que abre el menú luego abre Tool y veras el submenú Device Manager que al presionarlo mostrará la opción para crear uno o varios dispositivos virtuales.



Elija Device Manager y se abre un cuadro de dialogo.



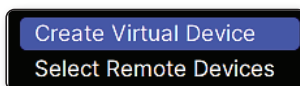
Luego muestra el cuadro de dialogo donde se ven los dispositivos disponibles que ya tienes y si no, estará vacío. En el caso de la imagen anterior se muestran dos modelos de teléfonos.

Crear un nuevo dispositivo o emulador virtual.

Clic en el icono +.



Selecciona Create Virtual Device.

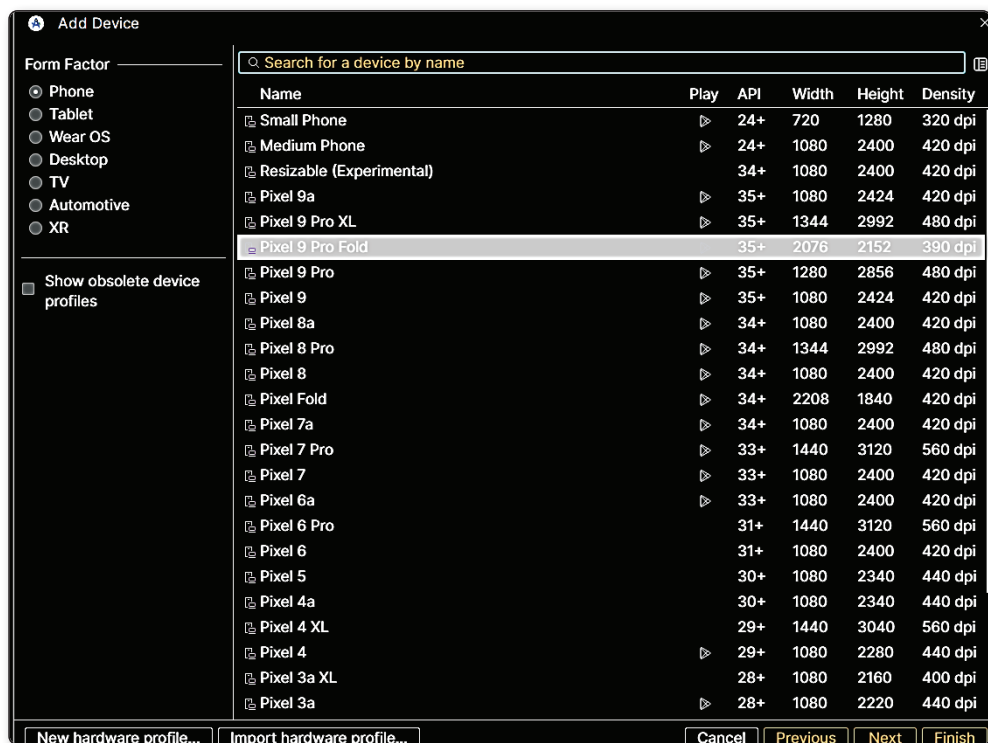


En el siguiente cuadro de dialogo escoges un modelo por ejemplo Pixel 9, API 35+ y luego presiona el botón Next.

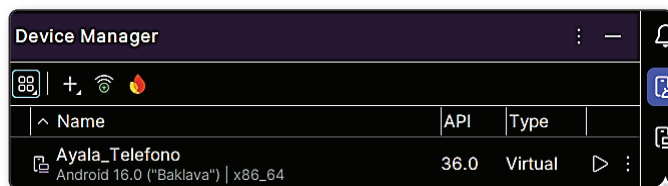
Aparece en donde debes poner el nombre al dispositivo, lo puedes dejar como está pero te puedes confundir.



Asígnale un nombre al emulador.



Al presionar Finish ya podrás ver en la ventana de Device Manager el dispositivo creado.



Al dar play en el icono Play.

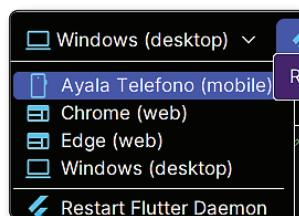


Después de alrededor de dos minutos se mostrará un lindo emulador del móvil creado.



El emulador creado aparecerá en la lista de emuladores disponibles para probar proyectos.

Ayala_Telefono aparece en la lista.



Al seleccionar y dar Play y después de un tiempo de carga que es más prolongado la primera vez y luego se mostrará la app que esté en ejecución.

Resumen

En este capítulo se explicó cómo configurar los teléfonos móviles reales para actividades de desarrollo y se enseñó cómo crear emuladores de teléfonos en Device Manager.

Preguntas

1. ¿Cuáles son las formas que hay en el IDE para probar los proyectos?
2. ¿Qué es Devices Management?
3. ¿Qué navegadores aparecen en tu IDE para probar proyectos?

Ejercicios

1. En la terminal del Windows PowerShell o en la terminal de Android Studio ejecuta el comando **flutter doctor** y revisa el informe que el doctor te da.

Nota

La terminal en Android Studio está abajo a la izquierda en la línea lateral de iconos.

2. Revisa si ya tienes instalado Android Studio, Flutter, Visual Studio 2022 o 2026, configurada la variable `Java_Home`, la variable de entorno para Flutter.
3. Crea un emulador llamado Mi Teléfono y pruébalo.

CONOCIENDO ANDROID STUDIO, FLUTTER Y DART

Objetivos

1. Comprender la relación entre Android Studio, Flutter y Dart.
2. Escribir el primer programa en Flutter.

Introducción

Para desarrollar aplicaciones con Flutter, es fundamental entender cómo interactúan las tres herramientas principales: Android Studio, Flutter y Dart.

Android Studio es la plataforma que proporciona el IDE (Entorno de Desarrollo Integrado), la “fábrica” donde construirás tus aplicaciones.

Flutter es el SDK (Kit de Desarrollo de Software), que proporciona las herramientas y bibliotecas necesarias para compilar y ejecutar aplicaciones en diferentes plataformas.

Flutter fue presentado en 2015, la versión 1.0 fue publicada en Londres en el año 2018 y luego han publicado varias versiones con la mejora de la capacidad en cada actualización.

Dart: el lenguaje de Flutter

Dart es el lenguaje de programación, orientado a objetos (POO), presentado el 10/10/2011 en la conferencia GOTO desarrollada en Aarhus Dinamarca.

Desarrollado por Google por los programadores Lars Bak, Kasper Lund y Gilad Bracha.

La versión 1.0 fue publicada en noviembre del 2013, y la última versión estable es la 3.7.2 publicada el 12 de marzo del 2025.

La versión Dart 3.10 beta fue publicada a finales del 2025.

Características de Dart:

- Es lenguaje de Programación Orientado a Objetos (POO).
- Sintaxis similar a C#, Java y JavaScript.
- Incluye variables, clases, funciones, operadores, métodos, constructores, expresiones de control de flujo, comentarios y bloques de código.
- Ofrece herramientas para facilitar el aprendizaje, como DartPad, un editor online que permite escribir y probar código Dart directamente en el navegador.
- Tiene la característica de Hot Reload que permite visualizar los cambios hechos en el código al instante en la interfaz.
- Su compilación es Ahead Of Time (AOT) y Just in time (JIT) la primera la implementa en tiempo de ejecución y lanzamiento y la segunda en tiempo de desarrollo y depuración.
- Tiene amplia disposición de librerías y paquetes tanto externas como internas que apoyan el desarrollo de los proyectos.
- Dart y Flutter fueron desarrollados por Google, Android Studio también pertenece a Google, lo que crea un ecosistema formidable y robusto para el desarrollo de apps con estas herramientas.

¡Hola Mundo! en Dart.

Para que te familiarices con Dart, escribe un programa que muestra el mensaje “¡Hola Mundo!” en la pantalla. Utiliza DartPad, el editor online de Dart.

1. Abre tu navegador web y ve a <https://dartpad.dev/>.
2. Borra el código que aparece por defecto.
3. Escribe el siguiente código:

```
void main() {  
    print("¡Hola Mundo!");  
}
```