

UNIVERSIDAD DE GUAYAQUIL

INGENIERÍA DE SOFTWARE

MANUAL TÉCNICO

SISMOAPP
(nombre provisional)

PROYECTO DE VINCULACIÓN
EVALUACIÓN DE LA VULNERABILIDAD SÍSMICA
DE LAS EDIFICACIONES UBICADAS EN LAS PARROQUIAS
OLMEDO Y BOLIVAR DE LA CIUDAD DE GUAYAQUIL

GUAYAQUIL - ECUADOR

AGOSTO 2025

Tabla de contenido

MANUAL TÉCNICO.....	1
1. INTRODUCCIÓN (borrador)	4
2. REQUERIMIENTOS TÉCNICOS (borrador)	5
3. ARQUITECTURA DEL SOFTWARE.....	8
2. Arquitectura de Alto Nivel.....	8
La arquitectura general del sistema se divide en tres capas interconectadas:	8
3. Componentes y Tecnologías	8
Front-end:	8
Back-end:.....	8
Infraestructura de Datos:	9
4. Flujo de Procesos	9
Detalle del Flujo de Peticiones:	9
4. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO	11
Android studio 2024.3.2.....	11
¿Por qué usarlo?.....	11
Beneficios	11
Flutter	11
¿Por qué usarlo?.....	12
¿Cómo funciona?	12
Supabase.....	12
¿Por qué usarlo?.....	12
¿Cómo funciona?	12
JAVA.....	12
¿Por qué usarlo?.....	13
¿Cómo funciona?	13
GitHub.....	13
¿Por qué usarlo?.....	13
¿Cómo funciona?	14
5. INSTALACIÓN	15
CONFIGURACIÓN DEL APLICATIVO	15
Instalación e implemetacion de Flutter	19
FASE 1: Descargar la carpeta Flutter.....	19

FASE 2: Implementación de Flutter en el entorno de trabajo.....	24
Visual Studio	26
Fase 3: Verificación e instalación.....	27
Android Studio	28
Visual studio.....	29
COMPILEADOR DEL PROYECTO.....	34
Opcion 1: Emulador de Android Studio.....	34
Opcion 2: Sacar APK del Programa	35
Opción 3: Terminal de Flutter para dispositivo físico	37
Opción 4: Extensión ADB desde el entorno de Android.....	37
Conexiones posteriores con wifi.....	39
6. CASOS DE USO	42
7. MÓDULOS	43
7.1. Módulo de Autenticación de Usuario.....	43
7.2. Módulo de Registro de Usuarios	43
7.3. Módulo de Consulta de Edificaciones	43
7.4. Módulo de Registro de Edificios	44
8. MODELO DE ENTIDAD RELACIÓN.....	46
9. DICCIONARIO DE DATOS DEL MODELO ENTIDAD RELACIÓN	47
1. Autenticación y seguridad.....	47
2. Formulario.....	51
3. Resultados.....	61
4. Historial y Auditoría	63
11. PROTOTIPOS DE PANTALLAS DEL APIACTIVO	66
12. Anexo de sugerencias y soluciones a problemas	67

1. INTRODUCCIÓN (borrador)

La evaluación de la vulnerabilidad sísmica de edificaciones es un proceso fundamental dentro de la gestión del riesgo, especialmente en zonas de alta sismicidad como las parroquias Olmedo y Bolívar de Guayaquil. Tradicionalmente, este proceso se realiza de manera manual, lo que implica una recopilación lenta de datos, una mayor probabilidad de errores y limitaciones al momento de consolidar resultados.

Para superar estas dificultades, se plantea el desarrollo de una **aplicación móvil denominada *SismosApp***, basada en la metodología **FEMA P-154 (2015)** para la inspección rápida de edificaciones. El proyecto busca **digitalizar el formulario FEMA P-154**, permitiendo a los inspectores capturar información en campo de manera más ágil, precisa y segura, reduciendo la carga operativa y facilitando la generación de reportes estandarizados.

La solución propuesta no se limita únicamente a la digitalización de formularios, sino que integra un ecosistema tecnológico completo que combina:

- **Una aplicación móvil en Flutter** (Android/iOS), para la captura y visualización de datos en campo.
- **Un back-end desarrollado en Node.js con Express.js**, que actúa como intermediario entre la app y los servicios en la nube.
- **Supabase** como plataforma de base de datos relacional (PostgreSQL), autenticación segura mediante JSON Web Tokens (JWT), y almacenamiento de objetos (fotografías y evidencias gráficas).

Este enfoque garantiza no solo la **eficiencia en la recolección de datos**, sino también la **consistencia, escalabilidad y seguridad** de la información, aspectos esenciales para un sistema que busca apoyar decisiones críticas en la gestión de riesgos sísmicos.

La aplicación incorpora un **motor de cálculo (SL1 Engine)** encargado de procesar automáticamente los parámetros establecidos en la metodología FEMA P-154, entregando resultados cuantificables sobre el nivel de vulnerabilidad de las edificaciones. Estos resultados pueden convertirse en **reportes PDF/Excel**, generados en segundo plano mediante procesos automáticos (Background Workers), listos para ser consultados por usuarios o instituciones.

El proyecto representa un **avance significativo en la transformación digital de la evaluación de vulnerabilidad sísmica**, aportando una herramienta moderna, confiable y de impacto directo en la resiliencia urbana de Guayaquil.

Alcance

El alcance del proyecto comprende:

- El diseño y desarrollo de una aplicación móvil con **formularios dinámicos** basados en FEMA P-154.
- La implementación de un **sistema de autenticación y roles de usuario** a través de Supabase Auth.
- El almacenamiento seguro y escalable de la información mediante **PostgreSQL** y Supabase Storage.
- La integración de un **motor de cálculo (SL1 Engine)** para procesar automáticamente las puntuaciones de vulnerabilidad.
- La generación de **reportes digitales en formato PDF/Excel**, gestionados mediante procesos en segundo plano.
- La disponibilidad de un **back-end RESTful** que permita la comunicación fluida entre la aplicación móvil y la base de datos.

Objetivo General (borrador)

Desarrollar una aplicación móvil multiplataforma para la **evaluación rápida de vulnerabilidad sísmica de edificaciones** en Guayaquil, basada en la metodología FEMA P-154, que permita **digitalizar, almacenar, procesar y reportar datos estructurales y no estructurales** mediante un ecosistema tecnológico moderno y escalable.

Objetivos Específicos (borrador)

1. **Digitalizar** el formulario FEMA P-154 en una aplicación móvil multiplataforma (Flutter).
2. **Implementar un sistema seguro de usuarios y roles** mediante Supabase Auth y JWT.
3. **Diseñar la base de datos en PostgreSQL** para gestionar de manera estructurada la información de inspecciones.
4. **Integrar un motor de cálculo (SL1 Engine)** que ejecute automáticamente las reglas de vulnerabilidad definidas por FEMA.
5. **Optimizar la captura de datos en campo** con soporte para fotografías y almacenamiento en Supabase Storage.
6. **Desarrollar un módulo de generación de reportes digitales (PDF/Excel)** en procesos asíncronos gestionados por Background Workers.
7. **Facilitar la consulta y análisis de resultados** a través de la app móvil y la API RESTful.
8. **Validar la aplicación en campo** en edificaciones seleccionadas de las parroquias Olmedo y Bolívar, asegurando su usabilidad y precisión.

2. REQUERIMIENTOS TÉCNICOS (borrador)

Requerimientos mínimos de software

Requisitos de Software

COMPONENTE	Mínimo	Recomendado
SO MÓVIL	Android 10 / iOS 13	Android 12 / iOS 15
FRAMEWORK MÓVIL	Flutter 3.x (Dart SDK)	Última versión estable de Flutter
IDE	Android Studio Flamingo / VS Code	Android Studio Giraffe + extensiones
BACKEND	Node.js 16.x + Express	Node.js 18.x LTS + Express
BASE DE DATOS	PostgreSQL 12 (Supabase)	PostgreSQL 14+ (Supabase)
AUTENTICACIÓN	Supabase Auth con JWT	Supabase Auth + MFA
ALMACENAMIENTO	Supabase Storage	Supabase Storage + backup automático
REPORTES	pdfkit / exceljs	pdfkit / exceljs + workers en contenedores
CONTROL DE VERSIONES	Git	GitHub/GitLab con CI/CD

Requerimientos mínimos de hardware

2. Requerimientos de hardware

a) Dispositivo móvil (usuarios en campo)

REQUISITO	Mínimo	Recomendado
PROCESADOR	Quad-core 1.8 GHz	Octa-core ≥ 2.0 GHz
RAM	3 GB	6 GB
ALMACENAMIENTO	32 GB	128 GB
CÁMARA	8 MP	12 MP+
CONECTIVIDAD	3G / Wi-Fi	4G LTE o superior
PANTALLA	5.5" HD	6"+ Full HD

b) Servidor / Back-end

REQUISITO	Mínimo	Recomendado
CPU	2 núcleos	4 núcleos
RAM	2 GB	8 GB
ALMACENAMIENTO	20 GB SSD	50 GB SSD
SO	Ubuntu 20.04 LTS	Ubuntu 22.04 LTS
SERVICIOS	Supabase (Auth, Storage, Postgres)	Supabase + backups automáticos

c) Estación de desarrollo

REQUISITO	Mínimo	Recomendado
CPU	Intel i3 8va gen / Ryzen 3	Intel i5 10ma gen / Ryzen 5
RAM	8 GB	16 GB
DISCO	128 GB SSD	256 GB SSD+
GPU	Integrada	Dedicada o integrada con soporte emulador
SO	Windows 10 / Ubuntu 20.04 / macOS 11	Windows 11 / Ubuntu 22.04 / macOS 13
EXTRAS	SDK Android, Node.js, Git	SDK Android + Docker + PostgreSQL client

3. ARQUITECTURA DEL SOFTWARE

Arquitectura técnica propuesta para el desarrollo de un sistema de evaluación de vulnerabilidad sísmica. La solución se basa en una arquitectura de microservicios y se compone de tres elementos principales: una aplicación móvil como front-end, un back-end RESTful en la nube, y una plataforma de servicios de bases de datos y almacenamiento.

2. Arquitectura de Alto Nivel

La arquitectura general del sistema se divide en tres capas interconectadas:

1. **Capa de Presentación (Front-end):** Una aplicación móvil desarrollada con Flutter, compatible con iOS y Android. Su función principal es la captura de datos a través de formularios dinámicos y la presentación de información al usuario.
2. **Capa de Lógica de Negocio (Back-end):** Un servidor de aplicaciones desarrollado con Node.js y Express.js, que expone una REST API para la comunicación con el front-end. Esta capa contiene el motor de cálculo de vulnerabilidad y gestiona las tareas en segundo plano.
3. **Capa de Datos y Almacenamiento:** Se utiliza Supabase como plataforma de servicios.
 - a. PostgreSQL: Almacena los datos estructurados, incluyendo la información de los levantamientos y los reportes generados.
 - b. Supabase Storage: Un servicio de almacenamiento de objetos diseñado para guardar archivos binarios, como las fotos capturadas durante las inspecciones.
 - c. Supabase Auth: Un servicio de autenticación que gestiona la seguridad y el acceso de los usuarios mediante JSON Web Tokens (JWT).

3. Componentes y Tecnologías

Front-end:

1. Tecnología: Flutter
2. Función:
 - a. Interfaz de usuario (UI) para la interacción del usuario.
 - b. Form Builder para la creación y gestión de formularios de levantamiento de datos.
 - c. Cache para el almacenamiento temporal de datos y mejora del rendimiento.
 - d. Manejo de la autenticación y autorización del usuario a través de JWT.

Back-end:

1. Tecnología: Node.js, Express.js
2. Función:

- a. REST API: Sirve como punto de entrada para todas las solicitudes del front-end.
- b. SL1 Engine: Un módulo de lógica de negocio que implementa las reglas de evaluación de vulnerabilidad sísmica FEMA P-154.
- c. Report Worker API: Un servicio de API para gestionar la cola de tareas de generación de reportes.
- d. Background Workers: Procesos que se ejecutan en segundo plano para tareas que requieren tiempo, como la generación de archivos PDF/Excel.

Infraestructura de Datos:

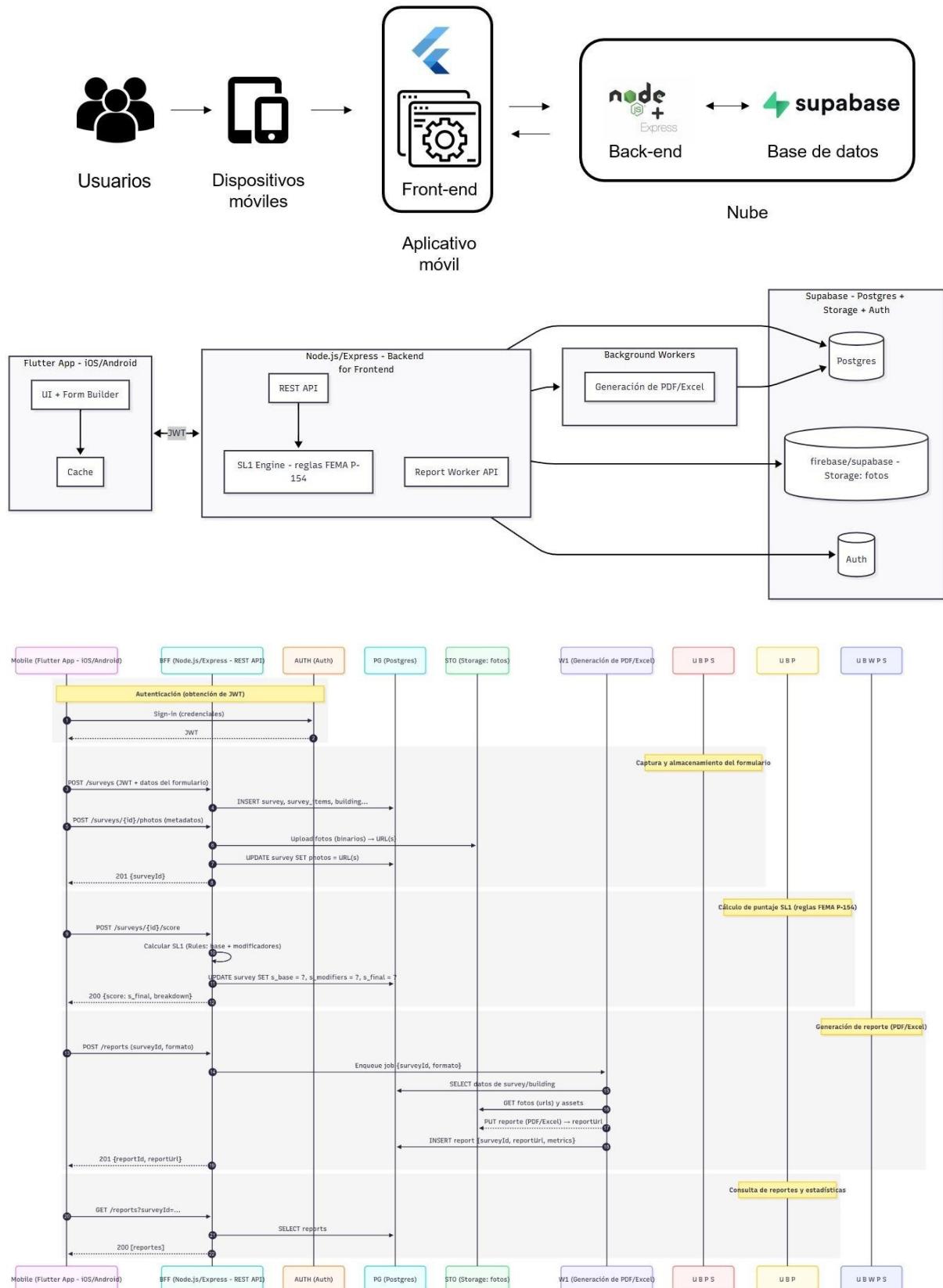
1. Tecnología: Supabase
2. Servicios Utilizados:
 - a. Postgres: Base de datos principal para la persistencia de datos relacionales.
 - b. Storage: Para almacenar las fotos asociadas a los levantamientos.
 - c. Auth: Provee un sistema de autenticación de usuarios y gestión de tokens.

4. Flujo de Procesos

El siguiente diagrama de secuencia ilustra el flujo de trabajo para la captura de datos y la generación de reportes:

Detalle del Flujo de Peticiones:

1. Autenticación: El usuario se autentica a través de la API de Auth para obtener un JWT.
2. Envío de Datos: La aplicación móvil envía los datos del formulario (metadata) a la API de encuestas (/surveys). El back-end inserta estos datos en la tabla de encuestas en Postgres.
3. Subida de Fotos: Las fotos se suben directamente a Supabase Storage, y la API del back-end actualiza las URLs de las fotos en el registro de la encuesta.
4. Cálculo de Puntuación: La aplicación móvil solicita el cálculo de la puntuación SL1. El back-end ejecuta el SL1 Engine y actualiza la base de datos con el resultado.
5. Generación de Reporte: El usuario solicita un报告, lo que encola una tarea en los Background Workers. Estos workers recuperan la información de Postgres y Storage, generan el reporte (PDF/Excel) y lo almacenan. El back-end actualiza el registro de la encuesta con el ID del reporte.
6. Consulta de Reportes: La aplicación móvil puede obtener los reportes o estadísticas a través de la API, que a su vez consulta la base de datos Postgres.



4. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

Android studio 2024.3.2

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android. Desarrollado por Google, ofrece herramientas poderosas y características para crear aplicaciones móviles y dispositivos Android. Basado en IntelliJ IDEA, permite a los desarrolladores diseñar, probar y depurar aplicaciones para teléfonos, tabletas, wearables, televisores y dispositivos de auto.

¿Por qué usarlo?

- **Desarrollo optimizado para Android:** Es la herramienta más completa para desarrollar aplicaciones Android, con características específicas para este sistema operativo.
- **Compatibilidad con lenguajes modernos:** Soporta Java, Kotlin y otros lenguajes usados para aplicaciones Android.
- **Integración con herramientas de Google:** Está diseñado para funcionar de manera fluida con las APIs y servicios de Google, como Firebase y Google Maps.
- **Diseño eficiente de interfaces:** Permite crear interfaces de usuario mediante un editor visual (Layout Editor) y proporciona acceso a plantillas de diseño.

Beneficios

- Emuladores de dispositivo Android: Facilita las pruebas de aplicaciones sin necesidad de un dispositivo físico.
- Herramientas de depuración: Ofrece herramientas avanzadas para depurar y optimizar el código, como el Android Debugger (ADB) y análisis de rendimiento.
- Plantillas de código y ejemplos: Proporciona plantillas preconfiguradas para crear rápidamente aplicaciones Android con diseños y funcionalidades comunes.
- Compatibilidad con Kotlin: Es compatible de forma nativa con Kotlin, el lenguaje preferido para el desarrollo Android.
- Soporte para pruebas unitarias y de interfaz: Facilita la ejecución de pruebas para asegurar la calidad de las aplicaciones.
- Actualizaciones constantes: Recibe actualizaciones frecuentes con nuevas características, herramientas y mejoras de seguridad.

Flutter

Flutter es un framework de desarrollo de aplicaciones creado por Google. Te permite crear aplicaciones móviles, web, de escritorio y embebidas usando una sola base de código.

¿Por qué usarlo?

- Una sola base de código para múltiples plataformas (¡no tienes que programar separado para Android e iOS!).
- Interfaz de usuario muy fluida y personalizable.
- Hot Reload: puedes ver los cambios en tiempo real mientras desarrollas.
- Respaldado por Google, usado en apps como Google Pay y eBay Motors.

¿Cómo funciona?

- Usa Dart para la lógica.
- Tiene su propio motor de renderizado gráfico (no usa componentes nativos como React Native, sino que dibuja todo desde cero).

Supabase

Supabase es una plataforma de desarrollo backend como servicio (BaaS) de código abierto. Te permite crear la infraestructura de base de datos, autenticación, almacenamiento y APIs sin tener que programar todo desde cero.

¿Por qué usarlo?

- Todo en uno: Base de datos, autenticación, almacenamiento y APIs listos para usar.
- Autenticación integrada: Manejo de usuarios con seguridad mediante JWT.
- PostgreSQL gestionado: Base de datos robusta y escalable sin tener que instalar nada local.
- Storage: Guardado de archivos (fotos, PDFs, videos) vinculado a la base de datos.
- Rápida implementación: En minutos tienes un backend funcional.
- Código abierto y respaldado por la comunidad.

¿Cómo funciona?

- Base de datos (Postgres): Permite crear tablas y relaciones con control de acceso por usuario (RLS).
- Auth: Sistema de autenticación (correo, redes sociales, SSO).
- Storage: Almacenamiento de archivos con control de acceso.
- APIs automáticas: Cada tabla tiene una API REST lista para consultar y modificar datos.
- SDKs: Tiene librerías para integrarse fácilmente con apps en Flutter, Node.js, Python, etc.

JAVA

Java 17 es una versión LTS del lenguaje y plataforma Java, altamente estable, segura y optimizada, fundamental para la compilación nativa de aplicaciones Android en Flutter y Gradle.

¿Por qué usarlo?

- **LTS oficialmente soportado:** Java 17 es la versión Long-Term Support vigente, con actualizaciones de seguridad garantizadas hasta 2029, asegurando estabilidad en entornos de producción.
- **Compatibilidad total con Flutter 3.x, Android Studio y AGP 8.x:** Evita incompatibilidades de compilación, alinear la toolchain de Flutter con el ecosistema Android moderno.
- **Optimización de rendimiento:** Introduce recolectores de basura de baja latencia (ZGC y Shenandoah), que reducen stop-the-world pauses y mejoran la experiencia en procesos intensivos como builds y pruebas automatizadas.
- **Nuevas características del lenguaje:** Incorpora Records, Sealed Classes y Pattern Matching, que disminuyen boilerplate y permiten una modelación más expresiva y segura de datos.
- **Seguridad reforzada:** Mitiga vulnerabilidades críticas (CVEs) presentes en versiones anteriores como Java 8 y 11, fortaleciendo la capa de ejecución en sistemas sensibles.
- **Interoperabilidad Kotlin/Java:** Garantiza plena integración con plugins nativos de Android y librerías legacy escritas en Java, coexistiendo con el ecosistema Kotlin en Android Studio.

¿Cómo funciona?

- **JVM 17 ejecuta bytecode multiplataforma:** Utiliza JIT (Just-In-Time) y soporte para AOT (Ahead-Of-Time) en Android ART, optimizando bytecode en tiempo de ejecución y en instalación.
- **Gradle + Android Studio:** La compilación se alinea mediante la propiedad org.gradle.java.home=jdk-17, garantizando que la toolchain use el mismo JDK en entornos locales y de CI/CD.
- **AGP 8.x con D8/R8:** Convierte el bytecode Java/Kotlin en DEX (Dalvik Executable) y aplica optimización/minificación, reduciendo tamaño final del APK/AAB y mejorando tiempo de carga.
- **JVM TI y Flight Recorder:** Proporcionan capacidades avanzadas de depuración, profiling y monitoreo en Android Studio Profiler, permitiendo análisis de rendimiento granular en tiempo real.
- **Configuración centralizada:** La definición de la versión de Java se establece en gradle.properties y variables de entorno (.bashrc, .zshrc, pipelines CI/CD), unificando la toolchain para desarrollo y despliegue continuo.

GitHub

GitHub es una plataforma colaborativa basada en Git que centraliza control de versiones, integración continua (CI/CD) y gestión de proyectos, garantizando trazabilidad, seguridad y automatización en equipos de desarrollo.

¿Por qué usarlo?

- **Control de versiones distribuido:** Basado en Git, permite mantener el historial completo de cambios en repositorios locales y remotos, soportando ramas paralelas y revertibilidad sin dependencia de un servidor central activo.
- **Flujo colaborativo escalable:** Los Pull Requests, revisiones de código (Code Review) e Issues facilitan la integración continua y la trazabilidad de decisiones técnicas, asegurando calidad en proyectos multi-equipo.
- **CI/CD nativo con GitHub Actions:** Proporciona pipelines automatizados para compilar proyectos Flutter, ejecutar pruebas unitarias/integración y desplegar artefactos (APK, AAB, IPA) en entornos controlados. (Nota: la generación y firma de IPA requiere runners macOS o integración con servicios externos como Codemagic o Fastlane).
- **Gestión de secretos y entornos:** Permite configurar variables cifradas (tokens de acceso, credenciales de keystore Android, certificados iOS) garantizando seguridad en la ejecución de workflows de CI/CD.
- **Integración con ecosistemas externos:** A través de Webhooks y GitHub Actions, se puede notificar a servicios como Supabase Edge Functions, Slack o Discord, desencadenando despliegues automáticos o alertas en tiempo real.
- **Comunidad y transparencia:** Issues públicos, wikis, GitHub Projects (tableros Kanban) y discusiones permiten la gestión ágil de tareas y la documentación colaborativa con visibilidad global.

¿Cómo funciona?

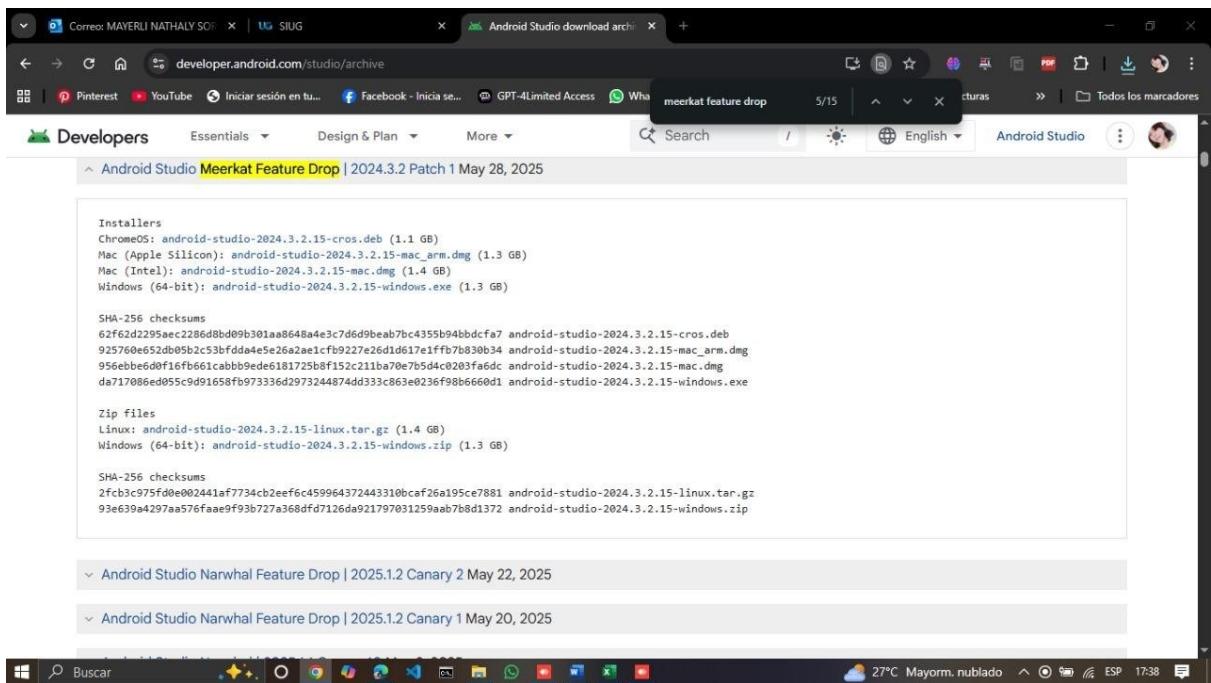
- **Repositorio Git remoto en la nube:** El flujo estándar incluye commits en local y sincronización mediante git push origin main, replicando cambios, etiquetas (tags) y versiones en GitHub.
- **Workflows YAML en GitHub Actions:** Definidos en .github/workflows/, permiten orquestar pipelines que incluyen comandos como flutter build apk --release, ejecución de tests automatizados y análisis estático con SonarCloud.
- **Protección de ramas y PR checks:** Se configuran reglas de branch protection que exigen pasar pruebas (lint, tests, build) antes de permitir merges a ramas críticas como main o release.
- **Releases y artefactos:** GitHub Releases almacena paquetes versionados (APK, AAB, IPA) y notas de versión. Los artefactos pueden ser firmados digitalmente con SHA-256 a través de workflows configurados en Actions.
- **Webhooks REST y APIs:** GitHub expone eventos HTTP POST para integrarse con sistemas externos. Esto permite, por ejemplo, que al cerrar un PR o fallar un pipeline, se ejecute una función en Supabase, un microservicio backend o una notificación en un canal de monitoreo.

5. INSTALACIÓN

CONFIGURACIÓN DEL APLICATIVO

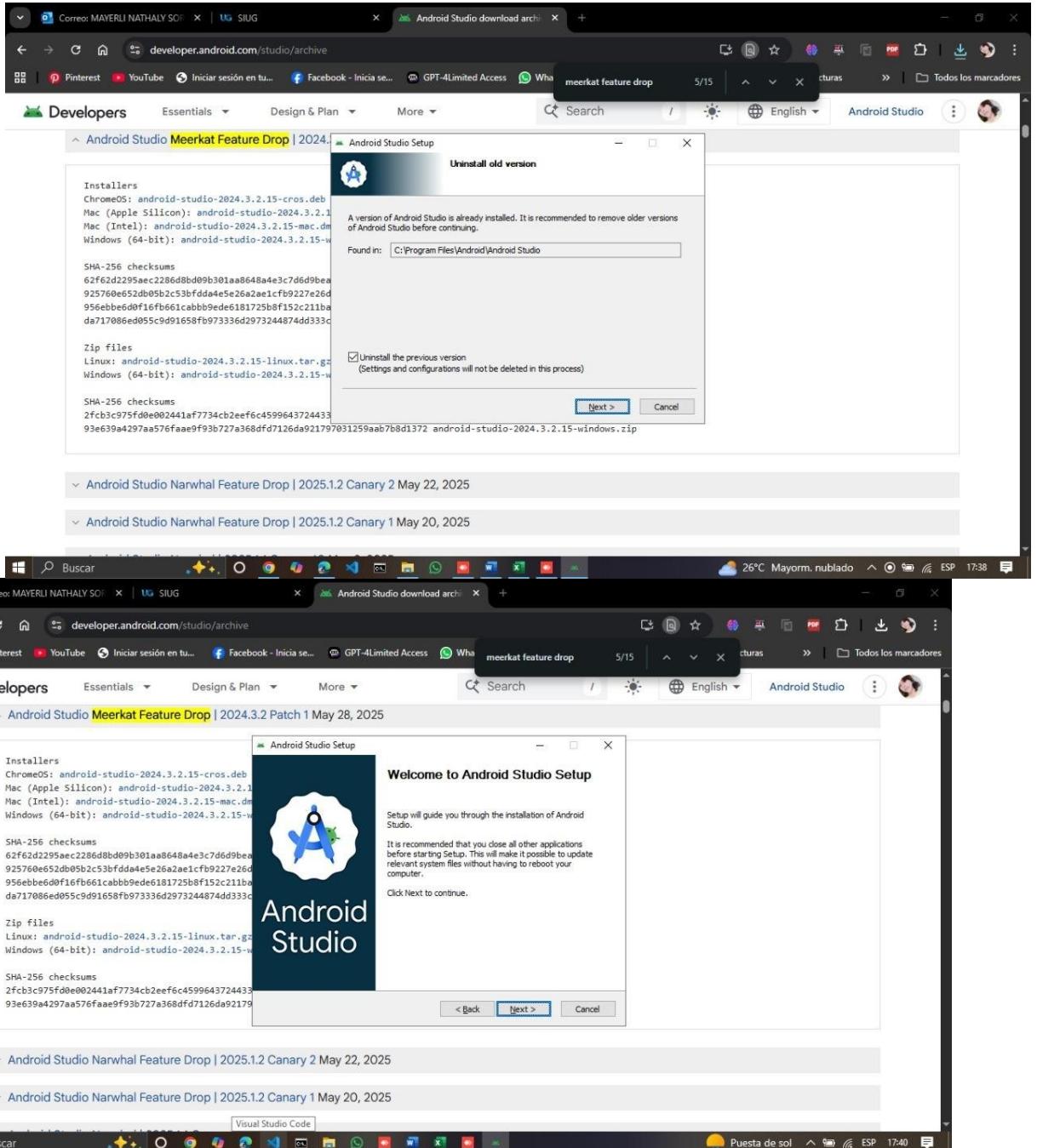
Descarga de Android Studio

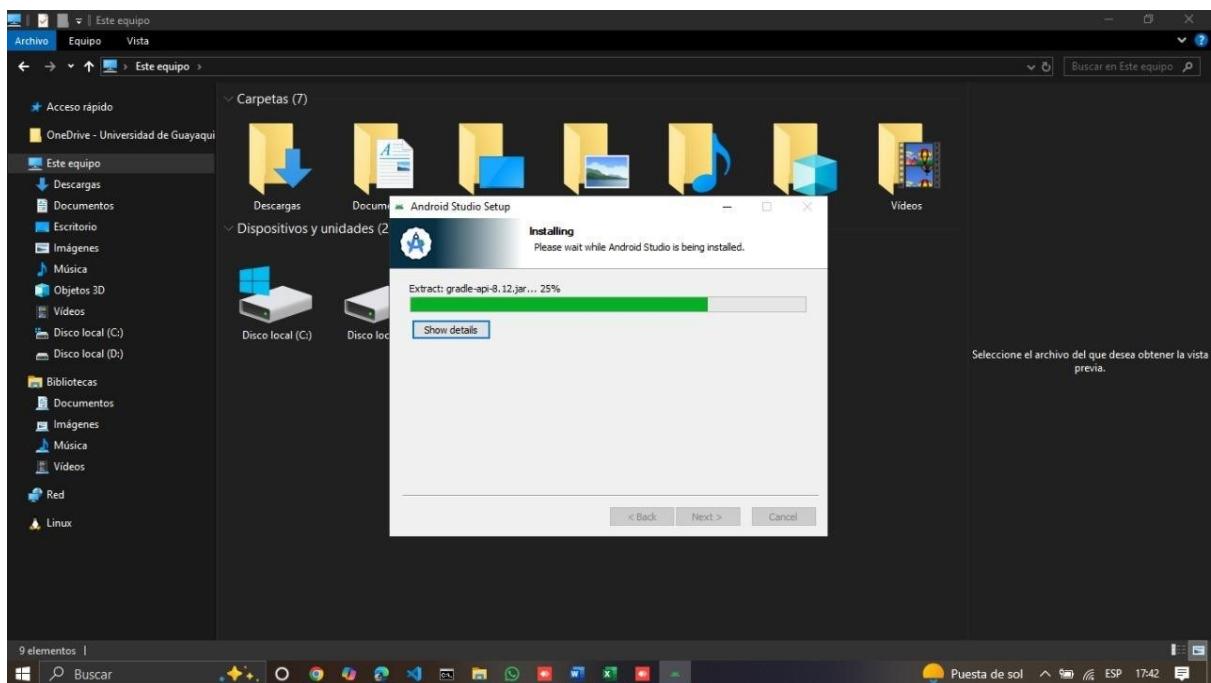
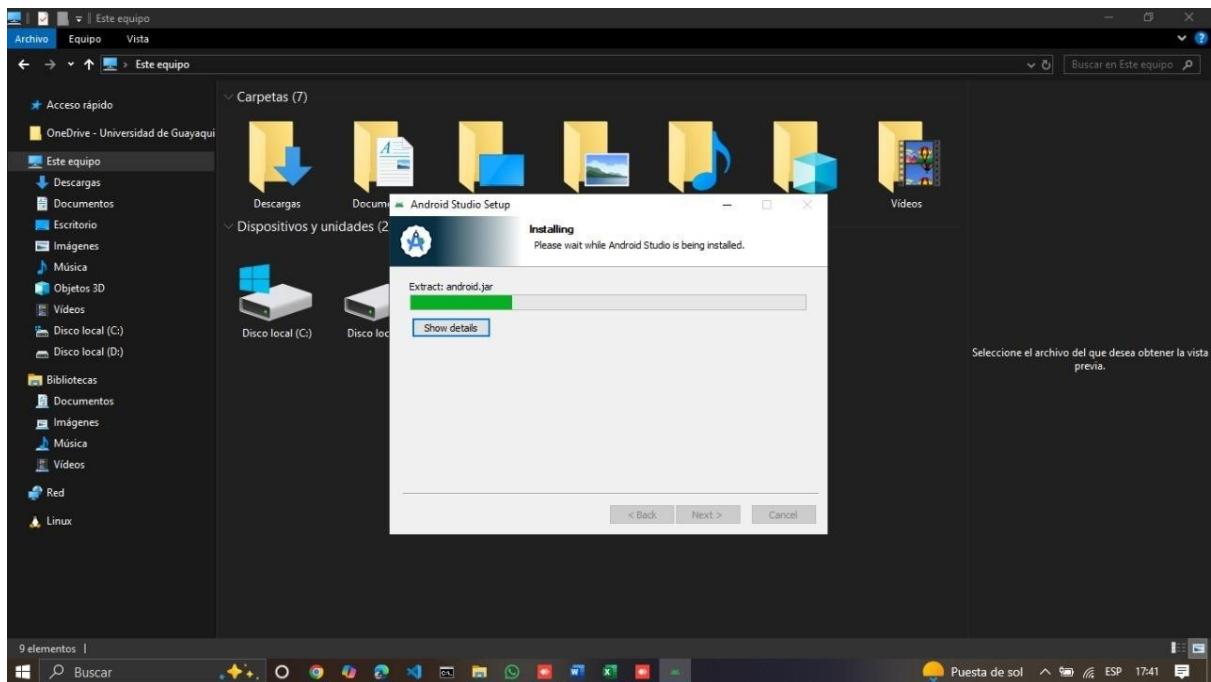
1. Ingresar y descargar el instalador de Android Studio desde <https://developer.android.com/studio/archive>
2. Seleccionar la versión recomendada (Android Studio 2024.3.2)
3. Descargar el archivo de instalación.

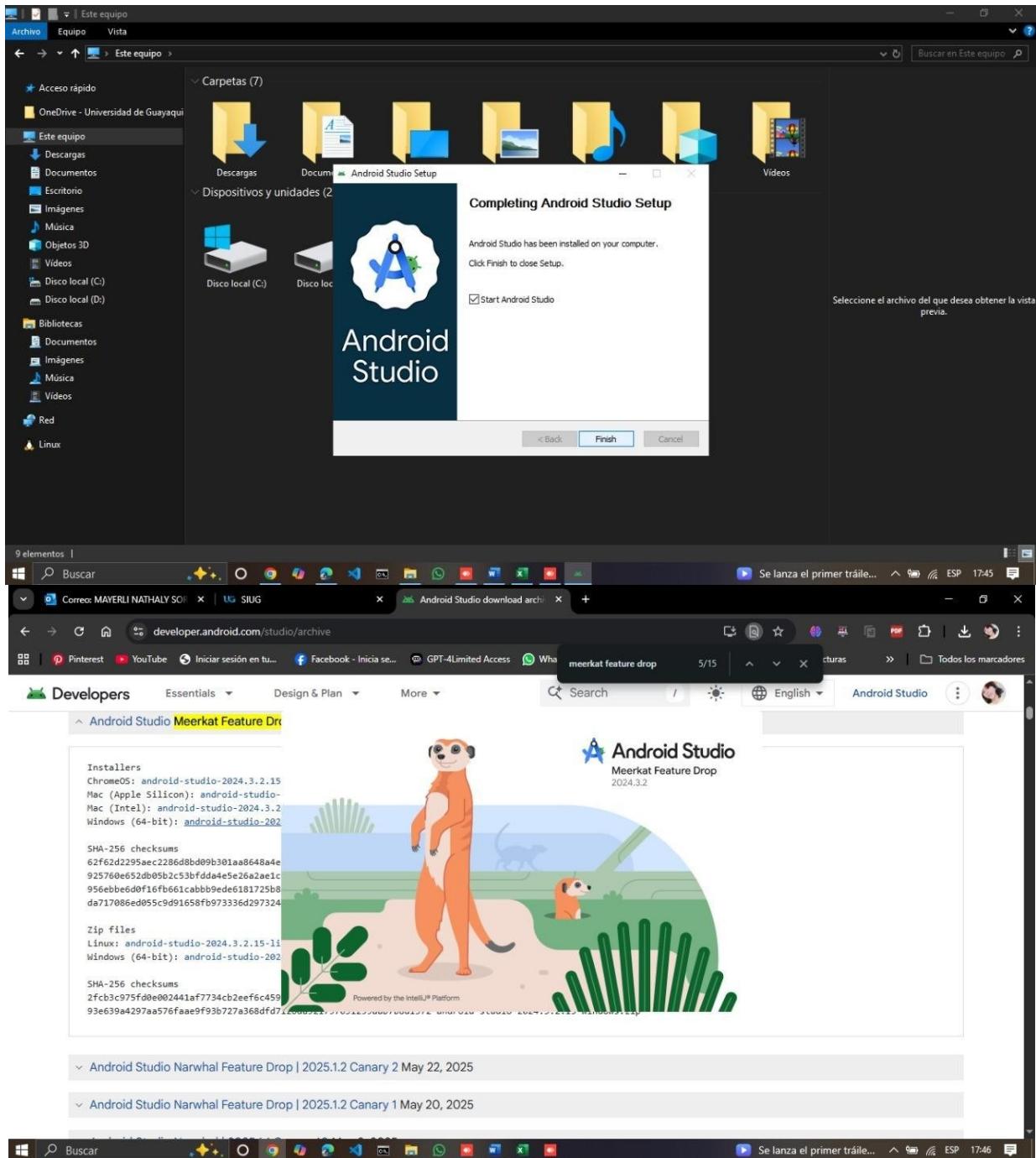


Instalador de Android Studio

1. Ejecutar el instalador descargado.
2. Seleccionar la ruta por defecto.
3. Asegurarse de marcar:
 - **Android SDK**
 - **AVD (Android Virtual Device)**
 - **Herramientas de compilación**
4. Finalizar la instalación y abrir Android Studio.







Instalación e implementación de Flutter

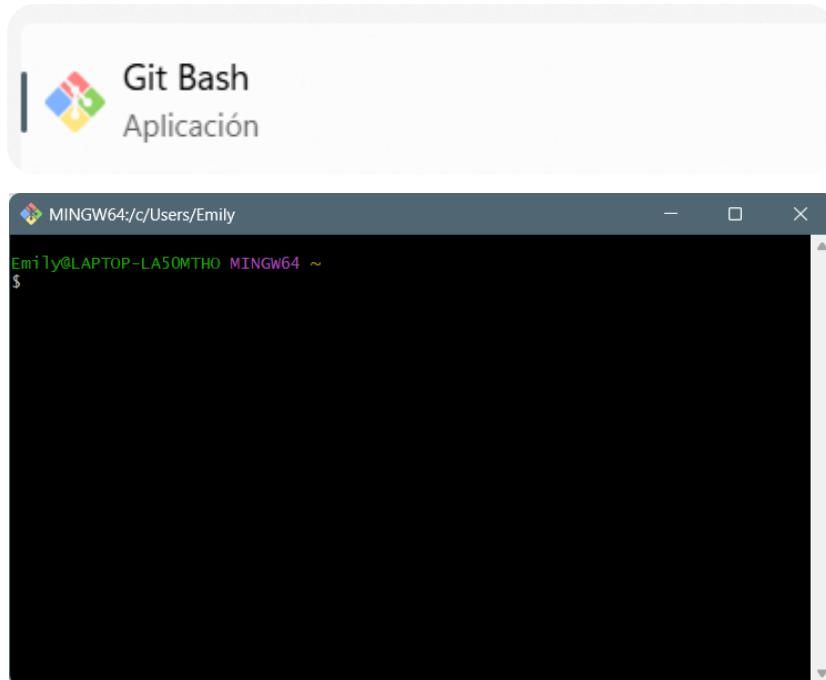
FASE 1: Descargar la carpeta Flutter

Se descarga la carpeta Flutter, esta acción se puede realizar de diversas formas según su página oficial <https://docs.flutter.dev> algunas de ellas se explicarán a continuación:

- Opción 1: Descarga del repositorio de Flutter a través de Git.

Advertencia: Se debe tener instalado Git : <https://git-scm.com/>

Paso 1: Abre GIT BASH..



Paso 2: Una vez abierta , se debe colocar el siguiente comando en la pantalla:

```
$ git clone -b main https://github.com/flutter/flutter.git
```

y luego se debe ejecutar (se puede hacer ENTER para ejecutarlo). Se debe esperar a que se descargue completamente, debe ser similar a la siguiente pantalla:

```
EVELIN@DESKTOP-OD8LGT4 MINGW64 ~
$ git clone -b main https://github.com/flutter/flutter.git
Cloning into 'flutter'...
remote: Enumerating objects: 1075483, done.
remote: Counting objects: 100% (117/117), done.
remote: Compressing objects: 100% (104/104), done.
remote: Total 1075483 (delta 55), reused 13 (delta 13), pack-reused 1075366 (from 2)
Receiving objects: 100% (1075483/1075483), 350.07 MiB | 15.65 MiB/s, done.
Resolving deltas: 100% (766740/766740), done.
Updating files: 100% (14946/14946), done.
```

Este repositorio se descargará de manera automática en el directorio:

C:\Users\(nombre del usuario)

Paso 3: Verifica que Dart se haya configurado correctamente, similar a la acción anterior se coloca este comando en el cmd y se ejecuta:

\$./flutter/bin/flutter

La información que se debe presentar en la pantalla debe ser similar a las imágenes a continuación:

```

$ ./flutter/bin/flutter
Unpacking dart sdk...
Downloaded flutter engine...
Expanding downloaded archive with PowerShell...
Building Flutter tool...
Running pub upgrade...
Resolving dependencies...
Downloaded packages... (23.6s)
Got dependencies.
Manage your Flutter app development.

Common commands:
  flutter create <output directory>
    Create a new Flutter project in the specified directory.
  flutter run [options]
    Run your Flutter application on an attached device or in an emulator.
Usage: flutter <command> [arguments]

Global options:
  -h, --help          Print this usage information.
  -v, --verbose       Noisy logging, including all shell commands
                      executed.
  If used with "--help", shows hidden options. If used
  with "Flutter doctor", shows additional diagnostic
  information. Use "--vv" to force verbose logging in
  those cases.
  -d, --device-id     Target device id or name (prefixes allowed).
  --version           Reports the version of this tool.
  --enable-analytics Enable telemetry reporting each time a flutter or
  dart command runs, until it is re-enabled.
  --disable-analytics Disable telemetry reporting each time a flutter or
  dart command runs, until it is re-enabled.
  --suppress-analytics Suppress analytics reporting for the current CLI

Project
  analyze
  build
  clean
  create
  drive

channel
  Populate the Flutter tool's cache of binary artifacts.
  Upgrade your copy of Flutter.

  precache
  upgrade

  attach
  custom-devices
  devices
  emulators
  install
  logs
  screenshot
  symbolize
  widget-preview

  Attach to a running app.
  List, reset, add and delete custom devices.
  List all connected devices.
  List, launch and create emulators.
  Install a Flutter app on an attached device.
  Show log output for running Flutter apps.
  Take a screenshot from a connected device.
  Symbolize a stack trace from an AOT-compiled Flutter app.
  Manage the widget preview environment.

Run "Flutter help <command>" for more information about a command.
Run "Flutter help -v" for verbose help output, including less commonly used
options.

```

Paso 4: Si se desea verificar que versión de Flutter y Dart se tiene, se pude colocar el siguiente comando:

\$./flutter/bin/flutter --version

```

EVELIN@DESKTOP-OD8LGT4 MINGW64 ~
$ ./flutter/bin/flutter --version
Flutter 3.35.0-1.0.pre-1488 • channel main •
https://github.com/flutter/flutter.git
Framework • revision d61f45178f (75 minutes ago) • 2025-08-19 22:42:38 -0400
Engine • hash 1e134652bf69d49e3ed8110ddc3cd0b146ecd79 (revision d61f45178f) (60
minutes ago) • 2025-08-20 02:42:38.000Z
Tools • Dart 3.10.0 (build 3.10.0-115.0.dev) • DevTools 2.49.0

EVELIN@DESKTOP-OD8LGT4 MINGW64 ~
$ 

```

b. Opción 2: Descarga de la carpeta de Flutter a través de la página oficial <https://docs.flutter.dev>

The screenshot shows the 'Install Flutter manually' section of the Flutter documentation. It includes a 'Choose your development platform' dropdown set to 'Windows', a 'Get started' button, and links for 'Choose your development platform', 'Download prerequisite software', 'Install and set up Flutter', 'Add Flutter to your PATH', and 'Continue your Flutter journey'.

Paso 1: En el apartado de **Install manually** de la página oficial se va la sección **de Install and set up Flutter:**

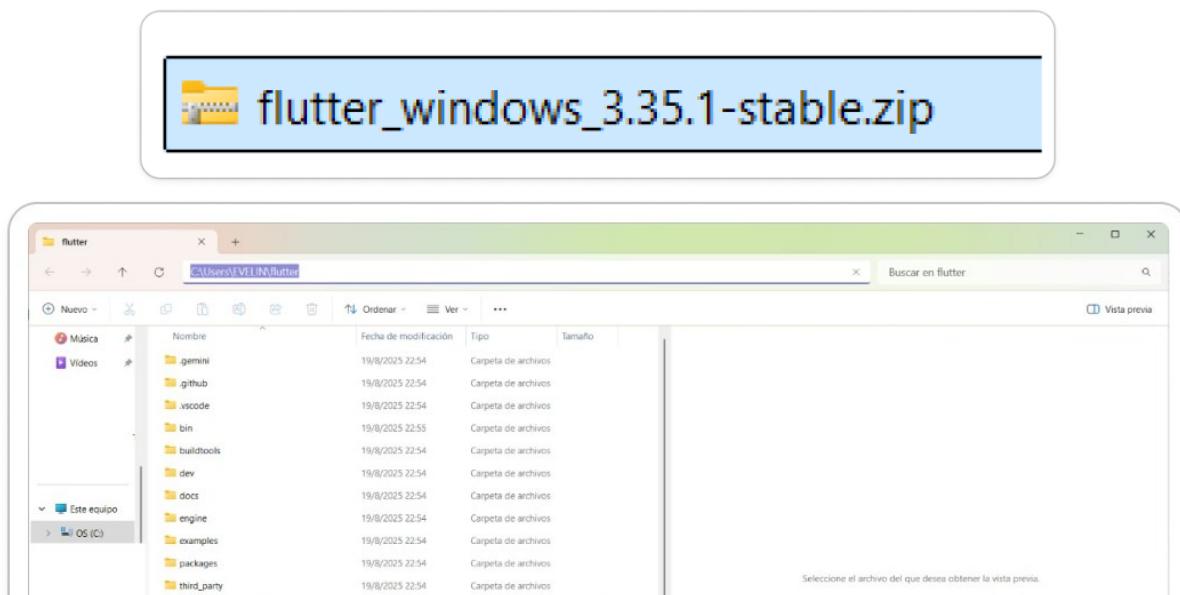
The screenshot shows the 'Install manually' section of the Flutter Docs. It includes a sidebar with navigation links like 'Get started', 'Set up Flutter', 'Install Flutter', 'Overview', 'Install with VS Code', 'Install manually' (which is highlighted), 'Upgrade SDK', 'SDK archive', 'Add to path', 'Troubleshoot', 'Uninstall SDK', 'Learn Flutter', 'Stay up to date', 'App solutions', 'Create with AI', 'User interface', 'Introduction', and 'Widget catalog'. The main content area is titled 'Install and set up Flutter' and contains three steps: 1. Download the Flutter SDK bundle (with a link to 'flutter_windows_3.35.2-stable.zip'), 2. Create a folder to store the SDK (with a note about selecting a path), and 3. Extract the SDK. A note at the bottom states that 'docs.flutter.dev uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.' There is also an 'OK, got it' button.

Se realiza clic en [flutter_windows_3.35.2-stable.zip](#), para descargar la versión más estable (actual) e manera automática.

Paso 2: Una vez descargada, extráela en la carpeta que deseas; es recomendable que se coloque en el directorio del usuario en el disco C:

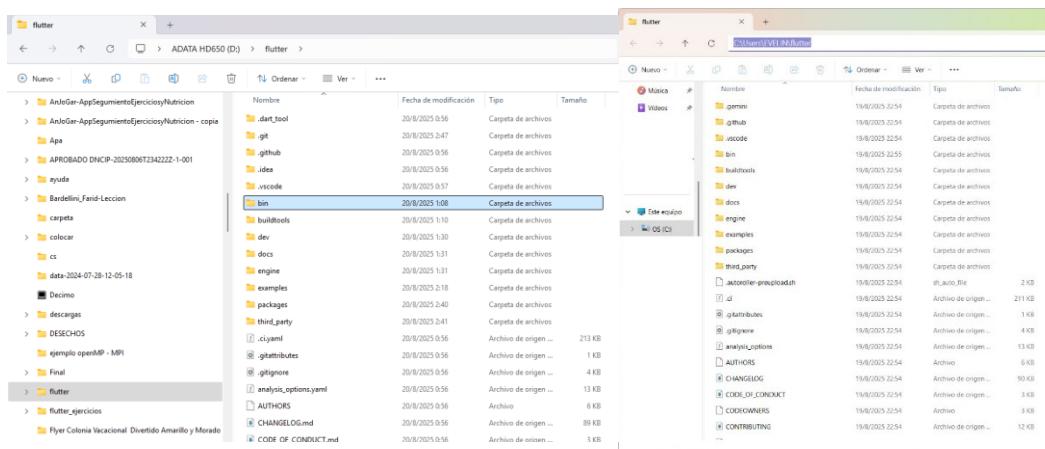
C:\Users\(\nombre del usuario)

Dado que se extraerá una carpeta llamada **flutter** y puede que en ciertos casos generar problemas con encontrar la carpeta.



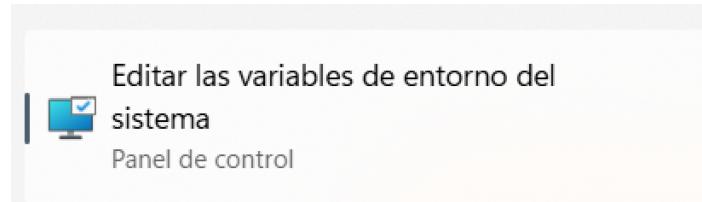
Con eso se debe establecer que Flutter se instaló, pero en ciertas circunstancias puede que las herramientas no la detecten (en ambas instalaciones) por lo que es recomendable realizar las siguientes acciones:

Paso 1: Verificamos donde está la carpeta de flutter/bin



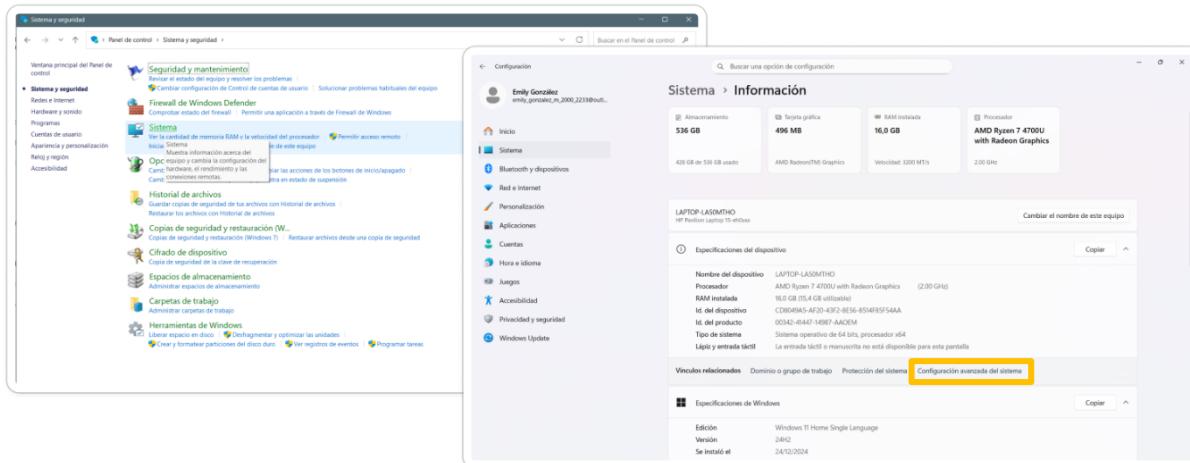
Paso 2: Nos dirigimos a las propiedades del sistema específicamente a las variables de entorno:

Opcion 1: Busca en tu computador “Editar Variables de entorno”

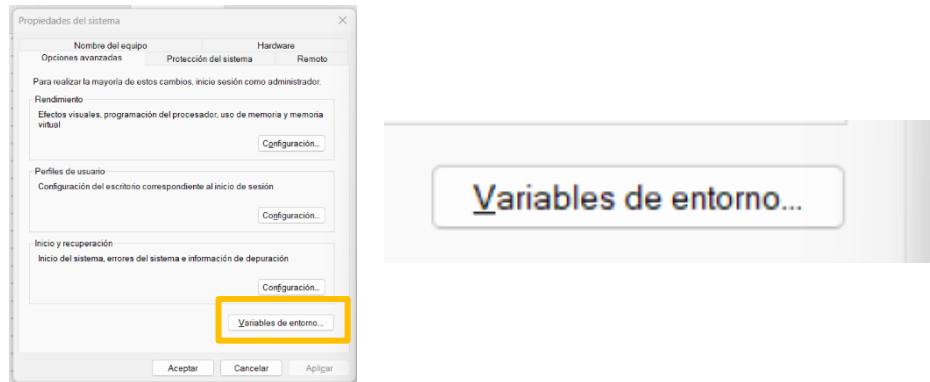


Opción 2: Se realiza la siguiente secuencia:

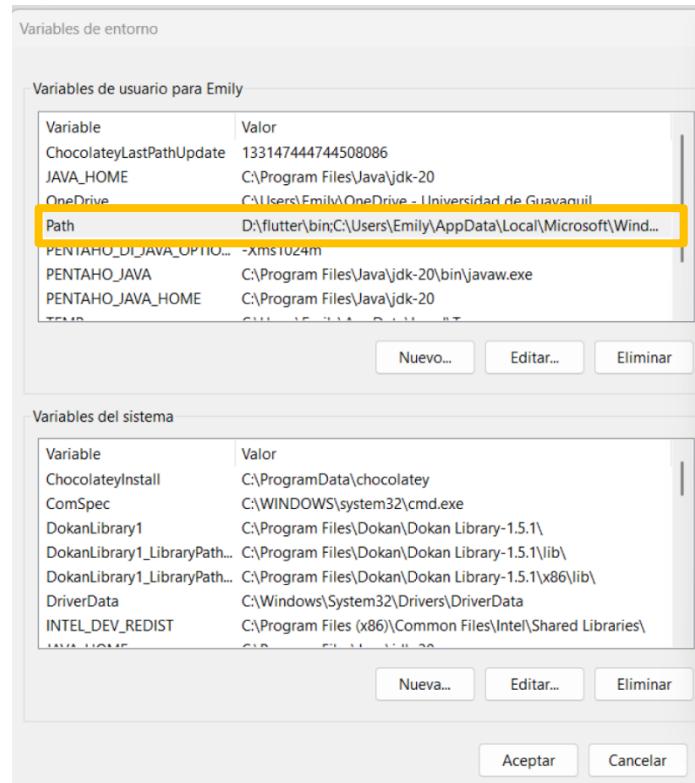
Abre el Panel De Control, nos dirigimos a la sección de Sistema y seguridad. Luego hacemos clic en la parte de Sistema y buscamos la opción de Configuraciones avanzada del sistema. Una vez encontrada se realiza clic en ella.



Paso 3: Debe aparecer la siguiente pantalla y hacemos clic en el botón de Variables de entorno



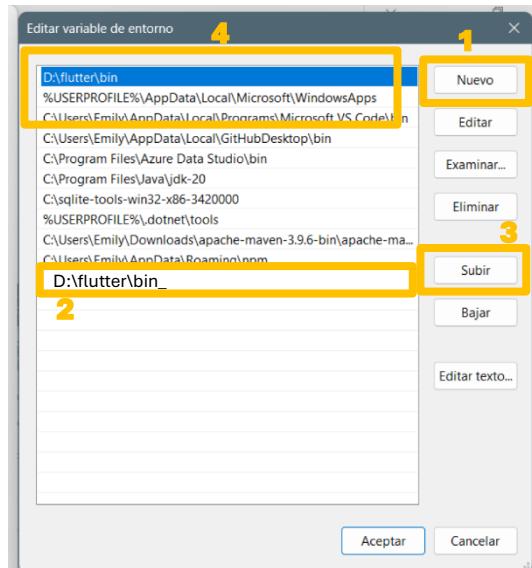
Paso 4: La siguiente pantalla es la que aparece, buscamos la variable PATH y hacemos doble clic en ella



Paso 5: En esta pantalla, se debe seguir los siguientes pasos:

1. Hacemos clic en el botón Nuevo
2. Colocamos la ruta donde esta la carpeta flutter/bin
3. Luego subimos al inicio

4. Debe quedar como esta en la imagen



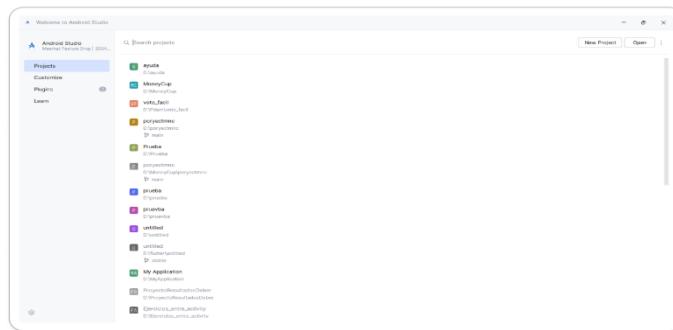
Paso 6: Aceptamos todo y para mayor seguridad reiniciamos el dispositivo.

FASE 2: Implementación de Flutter en el entorno de trabajo

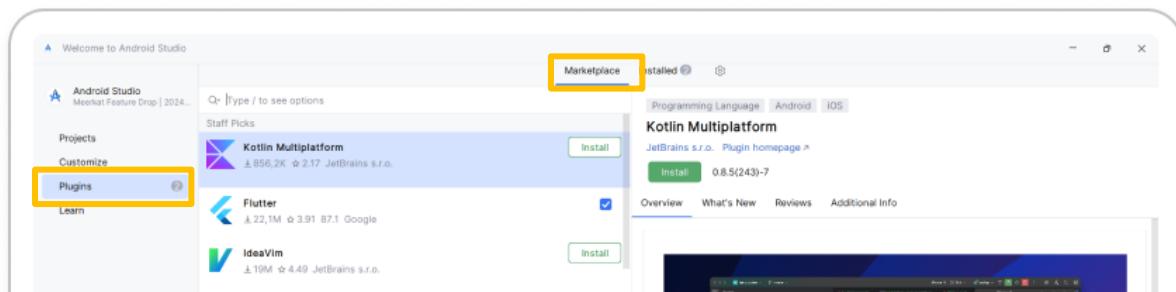
A continuación, se explicará de manera detallada de la implementación de las extensiones o plugins de Flutter en el entorno de trabajo, ya sean en Android Studio o Visual Studio.

Visual Studio

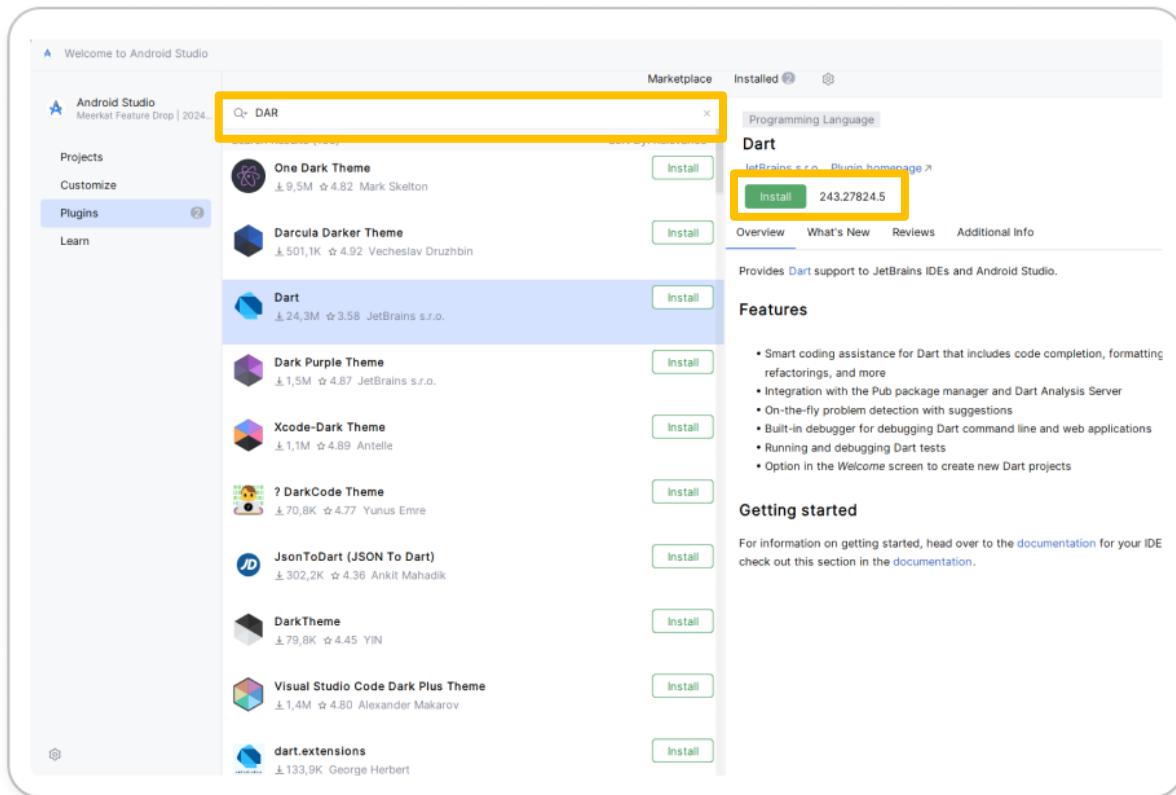
Paso 1: Iniciamos el programa



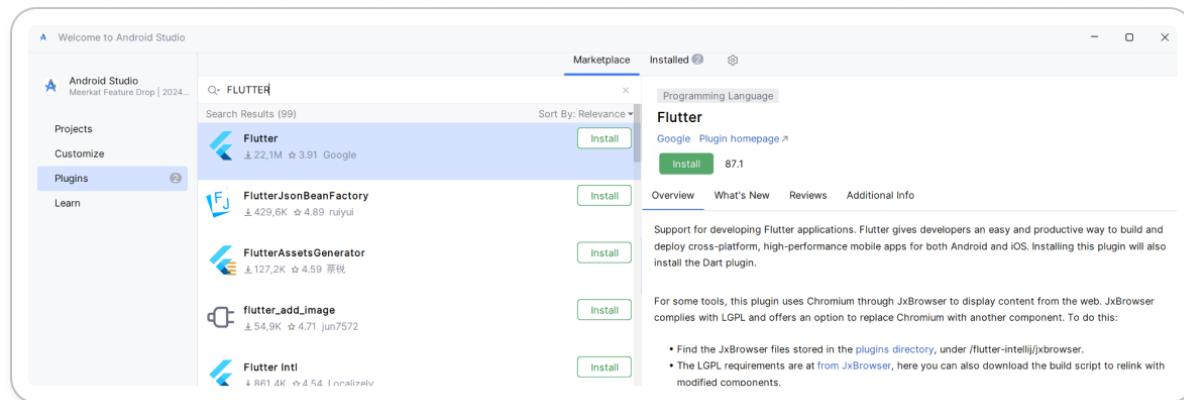
Paso 2: Nos dirigimos a la sección de Pluggins, en la parte de MARKETPLACE



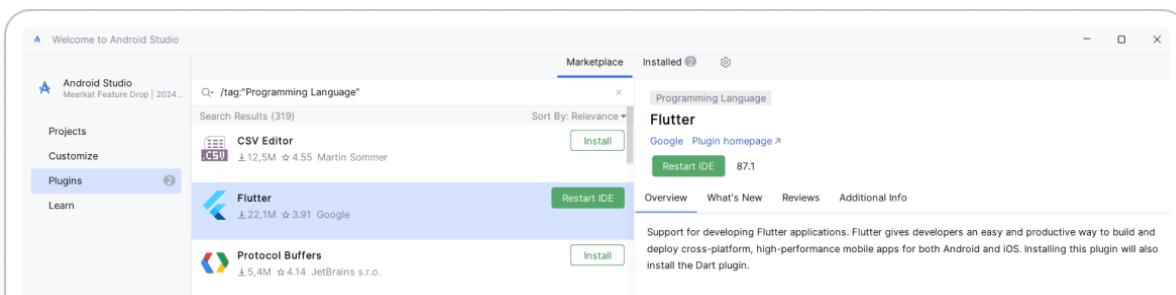
Paso 3: Escribimos “DART” en la barra de búsqueda y realizamos la instalación



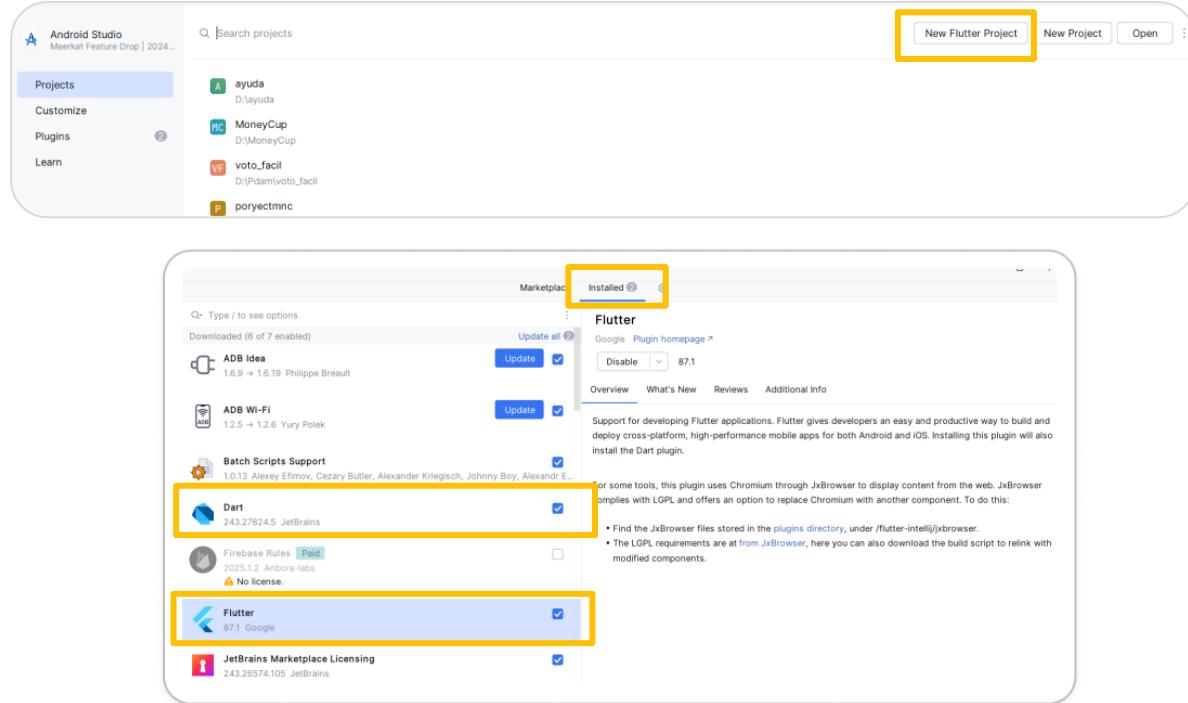
Hacemos lo mismo con Flutter:



Paso 4: Una vez instalado, se tiene que reiniciar el aplicativo.

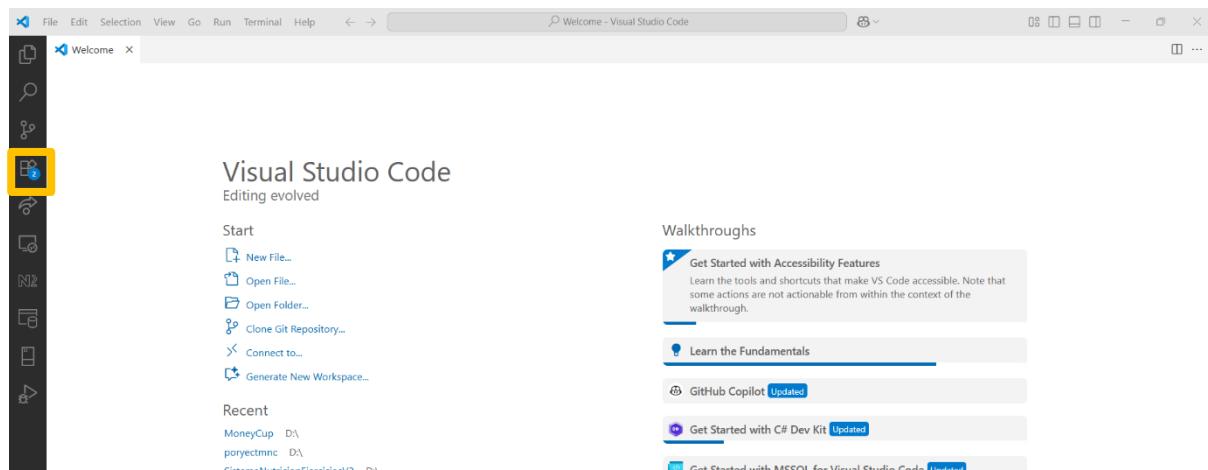


Paso 5: Para verificar que se ha instalado de manera correcta el plugin; se debe observar el botón de New Flutter Project al iniciar el aplicativo y los plugins deben estar en la sección de Installed.

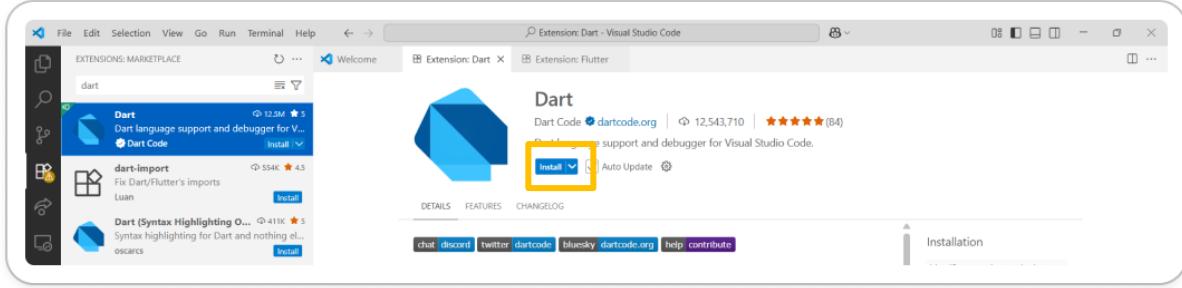


Visual Studio

Paso 1: Abrimos el Visual Studio y nos dirigimos a las Extensiones colocada en un parte lateral



Paso 2: Escribimos DART o FLUTTER en la barra de búsqueda e instalamos. Un aviso SI buscas e instalas Flutter primero te sale un aviso que las dos extensiones (FLUTTER Y DART) se van instalar



Paso 3: Verificamos la instalación de ambas extensiones.

No es necesario reiniciar el aplicativo, pero es recomendable.

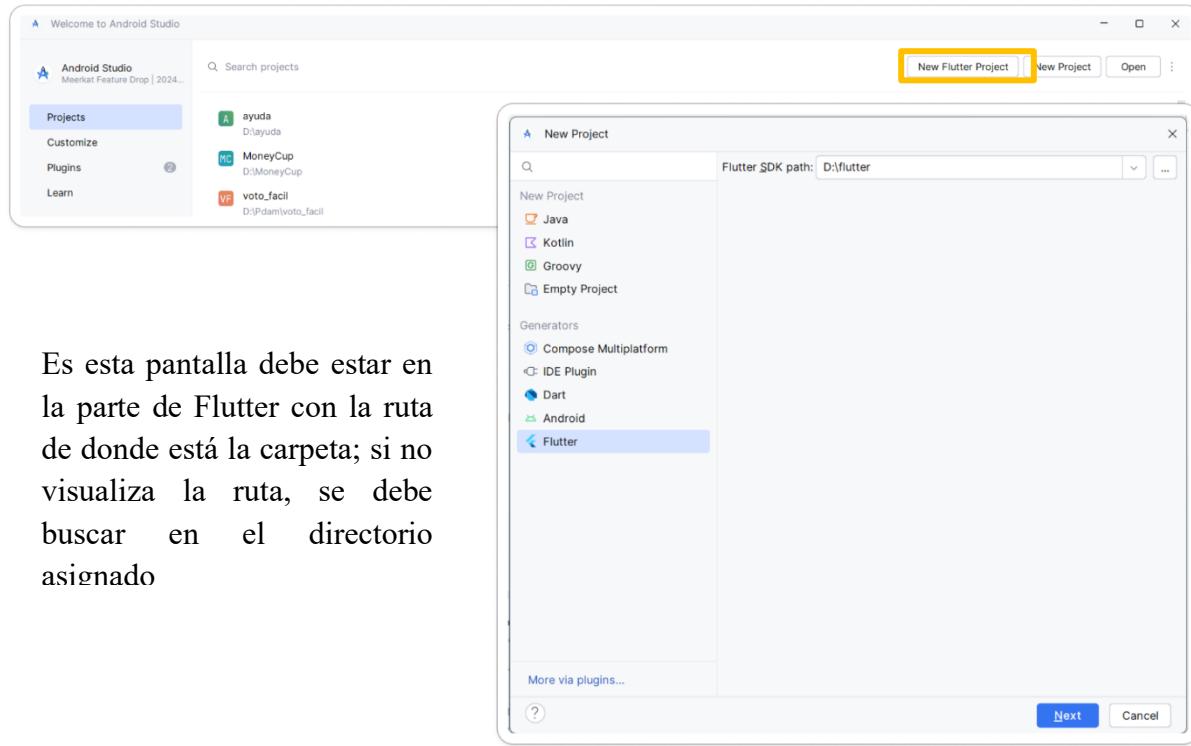
Paso 4: Para verificar si se ha implantado la extensión, nos dirigimos al área de búsqueda y colocamos >flutter si sale **Flutter: New Project**, se ha realizado con éxito.

Fase 3: Verificación e instalación.

En esta parte verificaremos que todo este correctamente implementado para la generación y compilación de cualquier proyecto Flutter.

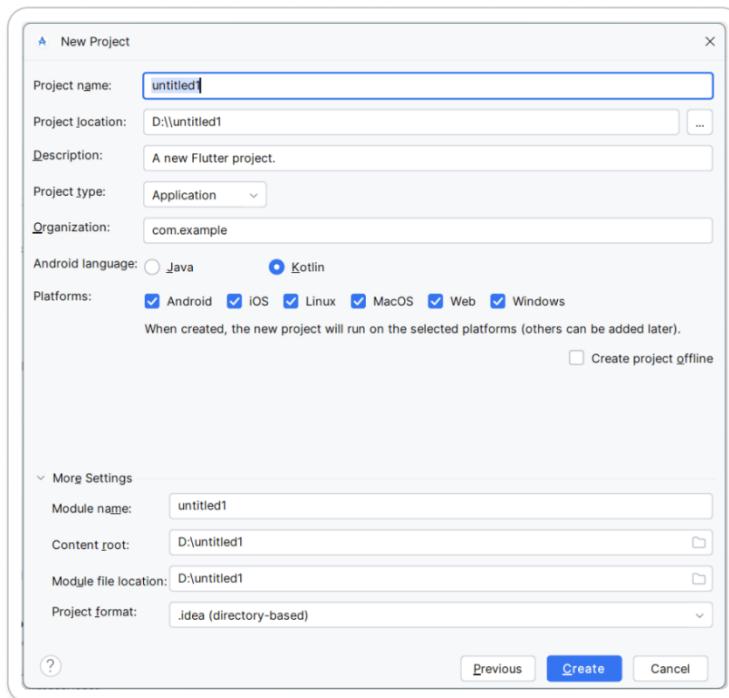
Android Studio

Paso 1: Hacemos clic en botón de New Flutter Project, se debe abrir la ventana de New Project



Es esta pantalla debe estar en la parte de Flutter con la ruta de donde está la carpeta; si no visualiza la ruta, se debe buscar en el directorio asignado

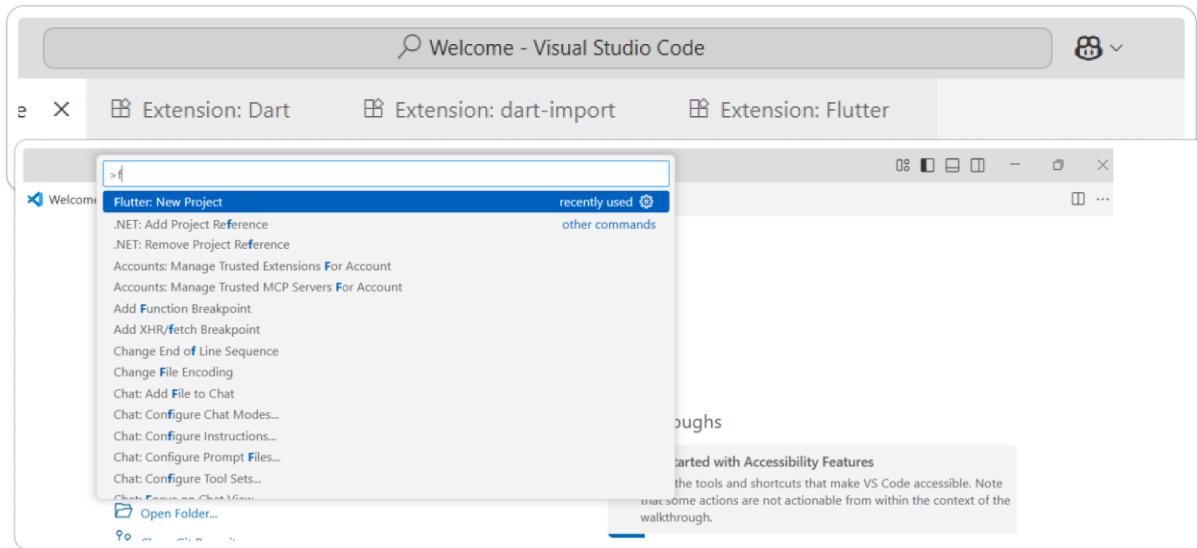
Paso 2: Si después de hacer el clic en el botón de Next aparece la siguiente ventana, entonces todo esta correctamente implementado.



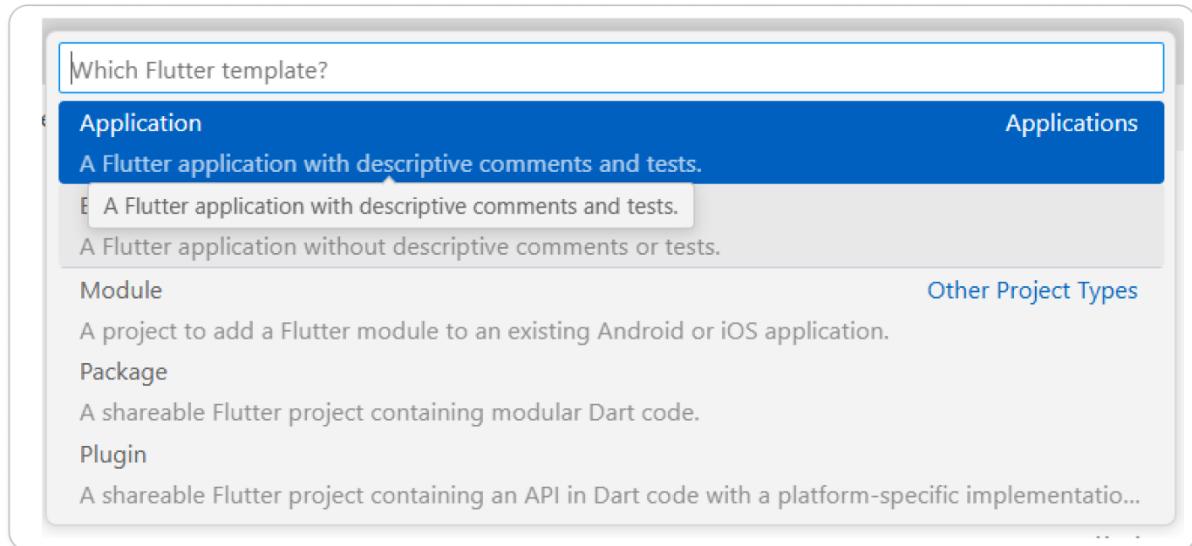
Si sale alguna advertencia o no parece la pantalla verificar las variables de entorno.

Visual studio

Paso 1: nos dirigimos al área de búsqueda y colocamos >flutter y si sale **Flutter: New Project**, y lo seleccionamos



Paso 2: Al seleccionar **Flutter: New Project** debe aparecer lo siguiente:



Si sale alguna advertencia o no parece la pantalla verificar las **Variables de entorno**.

Creación de nuevo proyecto con Supabase

Pasos a seguir:

1. Ingresar a <https://supabase.com/>
2. Registrarse o iniciar sesión con Google/GitHub.
3. Crear un **nuevo proyecto** → indicar nombre, región y contraseña de la base de datos.
4. Guardar la **URL del proyecto** y la **API Key** (necesarias para integrarlo con Flutter).

Construir en un fin de semana Escala a millones

Supabase es la plataforma de desarrollo de Postgres. Comienza tu proyecto con una base de datos Postgres, autenticación, API instantáneas, funciones Edge, suscripciones en tiempo real, almacenamiento e incrustaciones vectoriales.

[Comienza tu proyecto](#) [Solicitar una demostración](#)

Bienvenido de nuevo

Inicia sesión en tu cuenta

[Continuar con GitHub](#) [Continuar con SSO](#)

Correo electrónico

Contraseña [¿Has olvidado tu contraseña?](#)

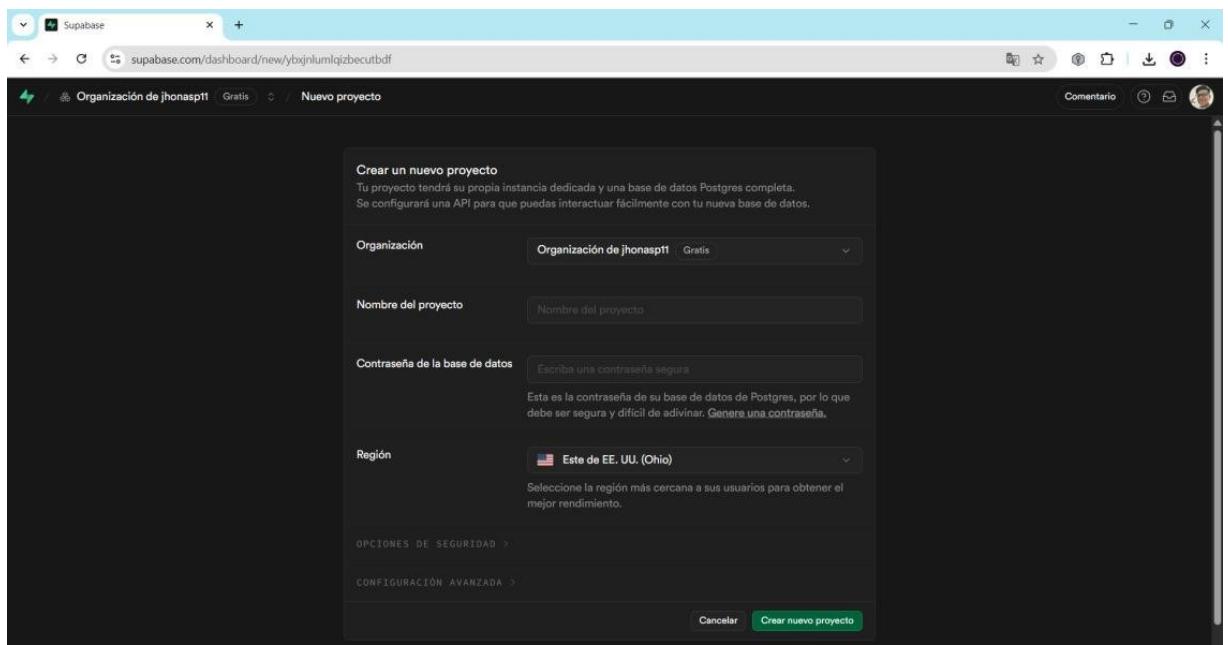
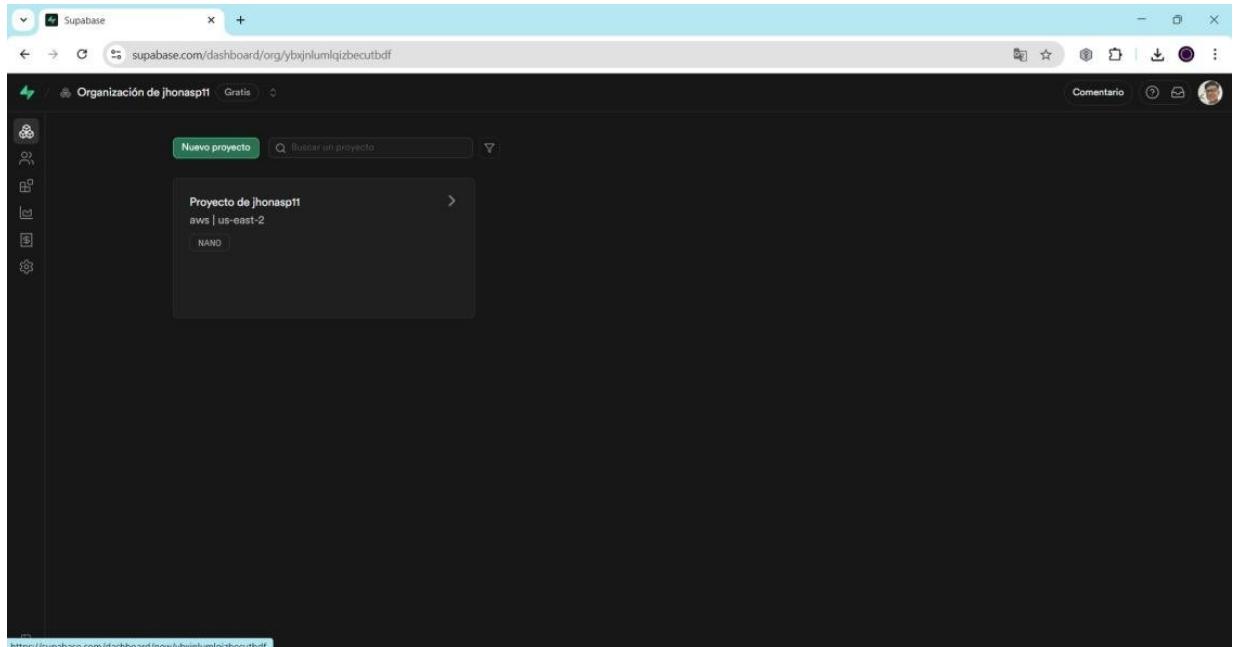
[Iniciar sesión](#)

¿No tienes cuenta? [Regístrate ahora](#)

Al continuar, aceptas los [Términos de servicio](#) y la [Política de privacidad](#) de Supabase y recibir correos electrónicos periódicos con actualizaciones.

¡Genial! Supabase es increíble.
Literalmente le ahorra a nuestro pequeño equipo el trabajo de un ingeniero entero constantemente. Los fundadores y todos con los que he hablado en Supabase también son gente increíble :)

 @KennethCassel



Bienvenido a tu nuevo proyecto
Su proyecto se ha implementado en su propia instancia, con su propia API configurada y lista para usar.

Comience a construir su base de datos
Empieza a crear tu aplicación creando tablas e insertando datos. Nuestro Editor de Tablas hace que Postgres sea tan fácil de usar como una hoja de cálculo, pero también puedes usar nuestro Editor SQL si necesitas algo más.

identificación	descripción	estado
1	Crear un proyecto	Completo
2	Ler la documentación	Completo
3	Crear aplicación	En curso
4	Conectar Supabase	En curso
5	Implementar proyecto	No iniciado
6	Conseguir usuarios	No iniciado
7	Actualizar a Dev	No iniciado

Creación del repositorio en github

Pasos a seguir:

1. Ingresar a GitHub y hacer clic en **New Repository**.
2. Definir nombre del repositorio (ejemplo: **SismoApp**).
3. Elegir visibilidad (Privado o Público).
4. Marcar la opción **Initialize with README**.
5. Crear el repositorio.

Start coding with Codespaces
Add a README file and start coding in a secure, configurable, and dedicated development environment.
[Create a codespace](#)

Add collaborators to this repository
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before
[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/DomenicaBrigitte/Vulnerabilidad-Sismica.git>
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# Vulnerabilidad-Sismica" >> README.md
git init
```

Colaboradores en Github

Pasos a seguir:

1. Entrar al repositorio.
2. Ir a **Settings > Collaborators**.
3. Agregar los usuarios de tu equipo.
4. Cada colaborador recibirá una invitación en su cuenta GitHub.

The screenshot shows the 'Collaborators' section of a GitHub repository settings page. On the left, there's a sidebar with options like 'Pages', 'Security', 'Advanced Security', 'Deploy keys', 'Secrets and variables', 'Integrations', 'GitHub Apps', and 'Email notifications'. The main area lists nine pending invite recipients, each with an email icon, their name, 'Awaiting response', and a 'Pending invite' button with a trash icon. At the bottom, there's a note about team access controls and discussions for contributors in an organization, along with a 'Create an organization' button.

Proyecto en Github

Terminal de Android Studio mostrando comandos Git:

1. Abrir la terminal de Android Studio en el proyecto.
2. Inicializar Git:
git init
git remote add origin https://github.com/usuario/SismoApp.git
3. Subir cambios iniciales:
git add .
git commit -m "Proyecto inicial"
git push -u origin main
4. Verificar en GitHub que el proyecto aparece cargado.

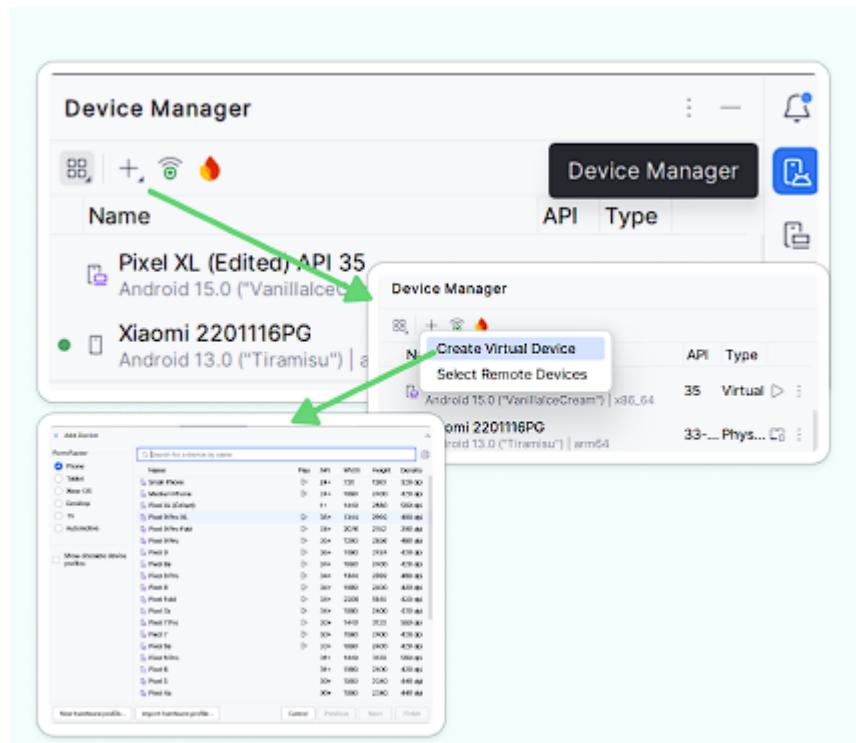
The screenshot shows a GitHub repository page for 'Vulnerabilidad-Sismica'. The top navigation bar includes links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The repository name 'Vulnerabilidad-Sismica' is displayed with a blue icon. Below the header, it shows 'master' branch, '1 Branch', '0 Tags', and a search bar. The commit history lists four commits by 'DomenicaBrigitte': 'Primer commit: creación de proyecto' (1 minute ago), 'node_modules' (1 minute ago), 'vulnerabilidad_sism' (1 minute ago), and 'package-lock.json' (1 minute ago). To the right, there's an 'About' section with project details ('Proyecto de vinculación - Vulnerabilidad Sísmica'), an 'Activity' section ('0 stars', '0 watching', '0 forks'), a 'Releases' section ('No releases published'), a 'Packages' section ('No packages published'), and a 'README' file link.

COMPILADOR DEL PROYECTO

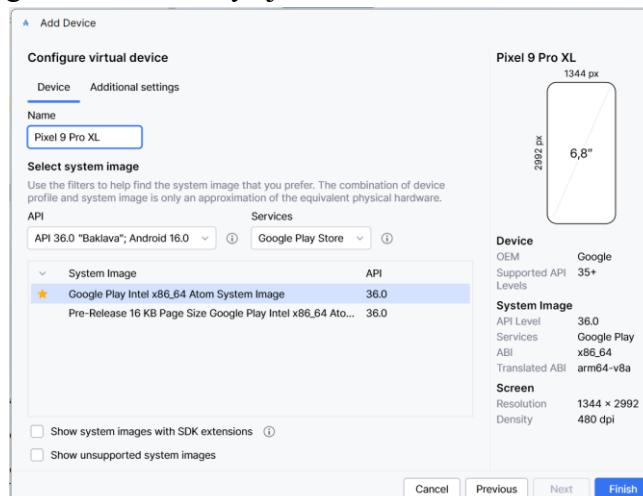
Opcion 1: Emulador de Android Studio

Si el proyecto no es demasiado pesado, el proyecto se puede ejecutar de manera directa con el Emulador que viene por defecto en Android Studio:

- **Paso 1:** Busca la herramienta de Device Manager y crea el emulador adecuado para el proyecto



- **Paso 2:** Configura el emulador y ejecútalo



Opcion 2: Sacar APK del Programa

Paso 1: Abrimos la terminal del proyecto colocamos los siguientes comandos

```
flutter build apk --release
```

Y esperamos que se ejecute hasta que aparezca el mensaje de :

Built build\app\outputs\flutter-apk\app-release.apk (45.6MB)

```
Terminal Local + 
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS D:\prueba2025> flutter build apk --release
Font asset "MaterialIcons-Regular.otf" was tree-shaken, reducing it from 1645184 to 1368 bytes (99.9% reduction). Tree-shaking can be disabled by providing the
--no-tree-shake-icons flag when building your app.
Running Gradle task 'assembleRelease'...
174,6s
PS D:\prueba2025> v Built build\app\outputs\flutter-apk\app-release.apk (39.4MB)
PS D:\prueba2025>
```

Paso 2: Escribimos luego el siguiente comando:

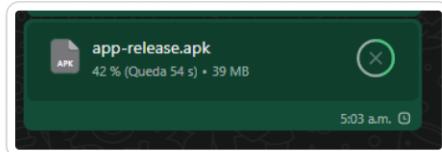
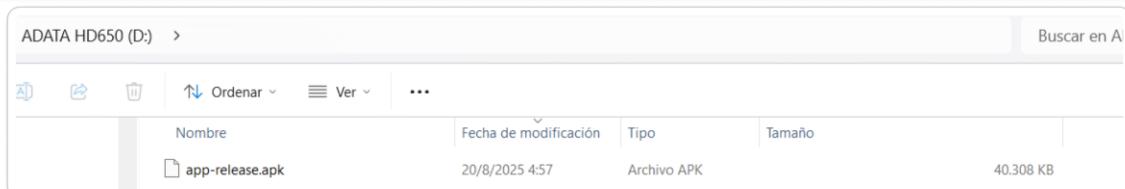
```
cd build\app\outputs\flutter-apk
```

```
PS D:\prueba2025> cd build\app\outputs\flutter-apk
PS D:\prueba2025\build\app\outputs\flutter-apk> [REDACTED]
```

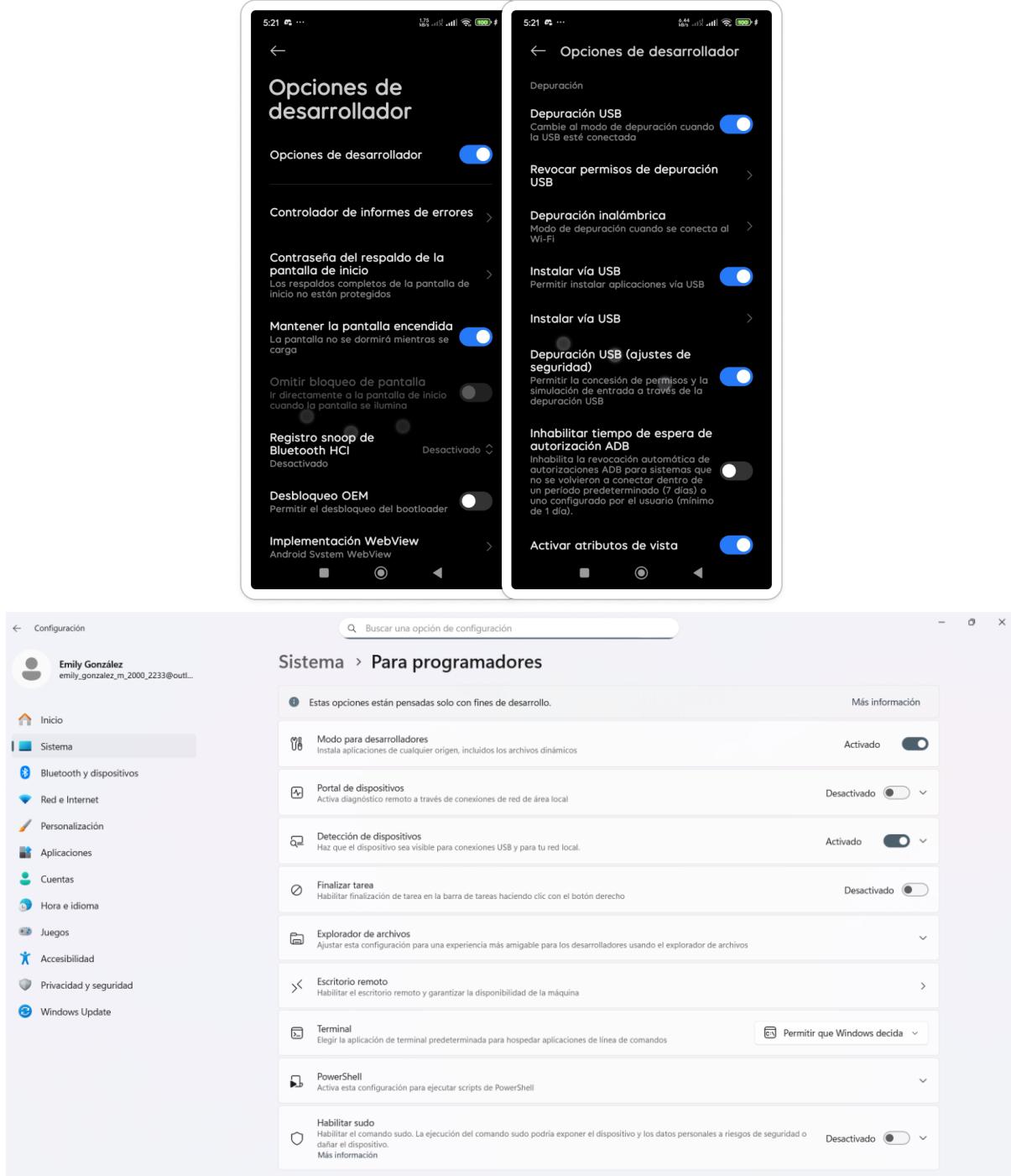
Paso 3: Luego el siguiente:

copy app-release.apk D:\← donde vas a copiar el apk

```
PS D:\prueba2025\build\app\outputs\flutter-apk> copy app-release.apk D:\[REDACTED]
PS D:\prueba2025\build\app\outputs\flutter-apk> [REDACTED]
```



Para las siguientes formas necesitas tener configurado el dispositivo para la depuración USB o WIFI (tanto desde el computador como para el teléfono)



Opción 3: Terminal de Flutter para dispositivo físico

Paso 1: Conecta tu teléfono con cable USB

Paso 2: Abre la terminal del proyecto y coloca el siguiente comando:

```
flutter devices
```

Y esperamos que se visualicen los dispositivos:

```
PS C:\flutterproyectos\Examen_2p_Domenica_Ortiz\vulnesis\prueba> flutter devices
>> # Esto usa ADB internamente
Found 5 connected devices:
  SM S918B (mobile) • R5CW711M44E      • android-arm64      • Android 15 (API 35)
  SM S918B (mobile) • 192.168.3.27:5555  • android-arm64      • Android 15 (API 35)
  Windows (desktop) • windows           • windows-x64       • Microsoft Windows [Versión 10.0.26100.4946]
  Chrome (web)      • chrome            • web-javascript   • Google Chrome 139.0.7258.139
  Edge (web)        • edge              • web-javascript   • Microsoft Edge 139.0.3405.111
```

Paso 3: Se coloca el comando de

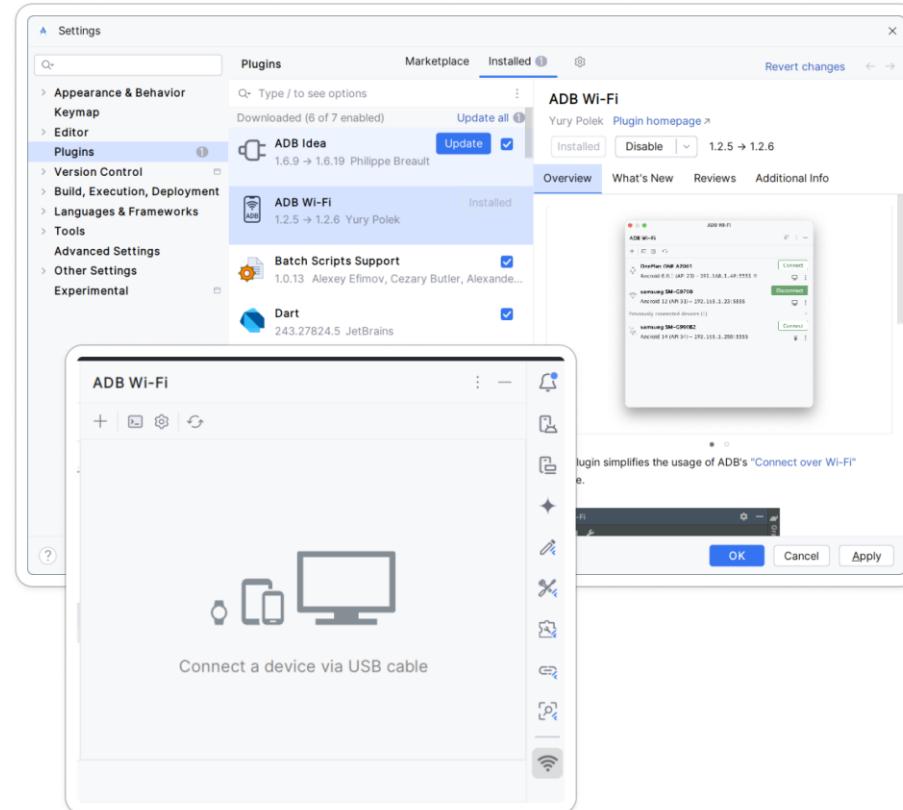
```
flutter run -d (nombre del dispositivo)
```

hacemos Enter y esperamos que se ejecute

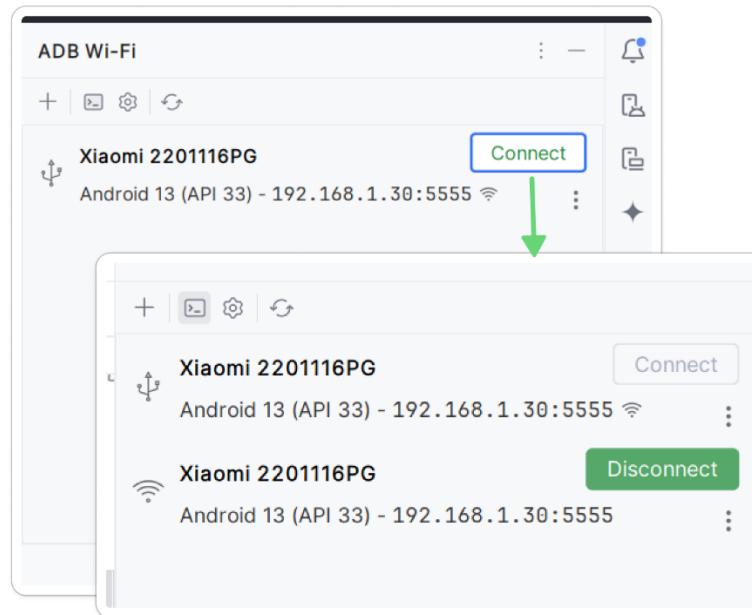
```
If you expected another device to be detected, please run "flutter doctor" to diagnose potential issues. You may also try increasing the time to wait for
connected devices with the "--device-timeout" flag. Visit https://flutter.dev/setup/ for troubleshooting tips.
PS C:\flutterproyectos\Examen_2p_Domenica_Ortiz\vulnesis\prueba> # Ejecutar en el dispositivo por USB (más rápido)
>> flutter run -d R5CW711M44E
>>
>> # O ejecutar en el dispositivo por WiFi
>> flutter run -d 192.168.3.27:5555
Launching lib\main.dart on SM S918B in debug mode...
Running Gradle task 'assembleDebug'...
```

Opción 4: Extensión ADB desde el entorno de Android

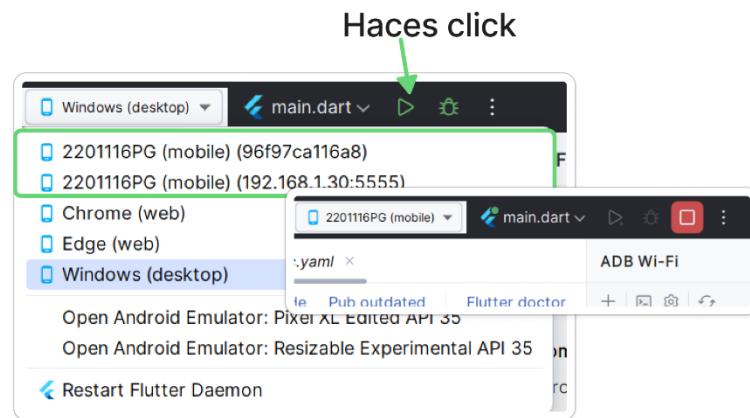
Paso 1: Descargar plugin ADB wifi en Android Studio



Paso 2: Conectar el cable y realiza la conexión:



PASO 3: Vamos al compilador en la parte superior y escogemos el dispositivo. Finalmente ejecutamos:



Paso 4: Aceptamos los permisos que nos pidan:

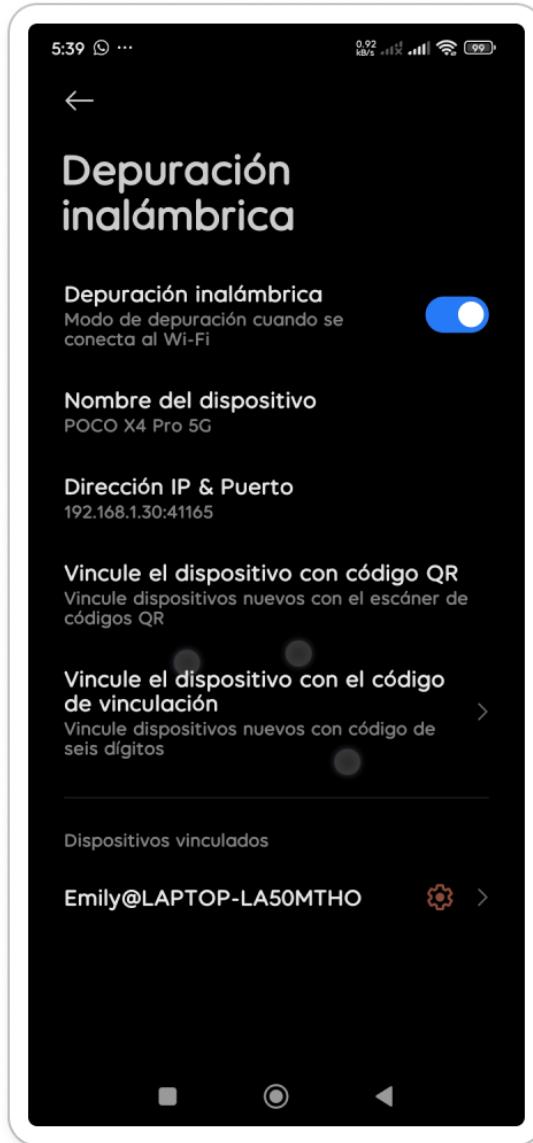


Conexiones posteriores con wifi

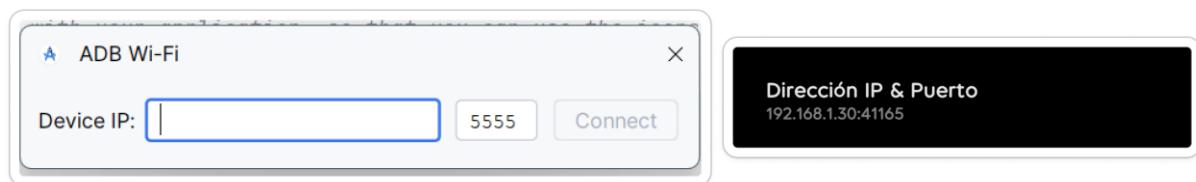
Paso 1: Realizamos una conexión:



Paso 2: Nos dirigimos en nuestro dispositivo y vamos a la parte de Depuración Inalámbrica y vemos el Id & puerto

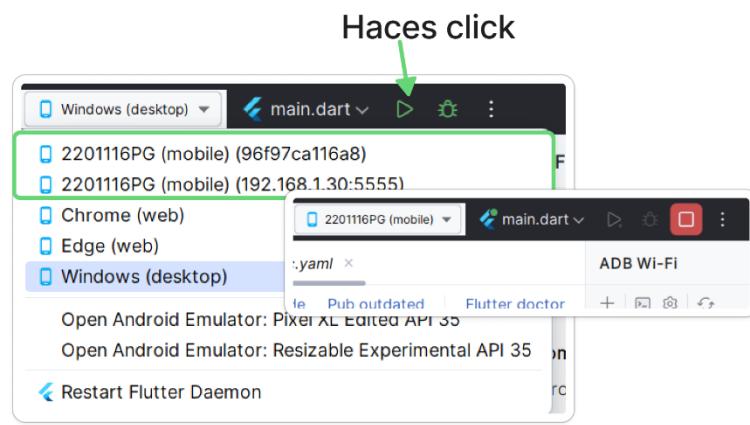


Paso 2: Colocamos el id en la parte izquierda y el puerto en el derecho



Realizamos la conexión, si llega la notificación del dispositivo acerca de la conexión, puedes cerrar la ventana de ADB wifi.

Paso 3: Vamos al compilador en la parte superior y escogemos el dispositivo. Finalmente ejecutamos:



6. CASOS DE USO

Se presenta una lista inicial de casos de uso del sistema "SismosApp", basada en el diagrama de flujo general, los wireframes y las funcionalidades descritas en el proyecto. Esta lista es preliminar y se enfoca en las interacciones principales de los usuarios según sus roles (Administrador, Inspector y Lector). Los casos de uso serán desarrollados en detalle posteriormente por todo el equipo, incluyendo diagramas, actores involucrados, precondiciones, postcondiciones y escenarios alternativos.

1. Registro de usuario: El usuario ingresa datos como nombre, rol, correo, teléfono y contraseña para crear una cuenta nueva.
2. Inicio de sesión: El usuario ingresa correo y contraseña para acceder al sistema, redirigiendo según su rol.
3. Recuperación de contraseña: El usuario ingresa correo o teléfono para restablecer la contraseña mediante un enlace o código.
4. Visualización y edición de perfil: El usuario ve y actualiza sus datos personales (nombre, correo, rol, etc.).
5. Cierre de sesión: El usuario finaliza su sesión activa.
6. Visualización de panel de estadísticas (para Administrador): Muestra resúmenes de usuarios registrados, activos, inactivos, edificios evaluados y gráficos de tendencias.
7. Registro de nuevo edificio (para Inspector): Ingreso de datos del edificio como nombre, dirección, código postal, uso, latitud, longitud, inspector asignado, fecha, número de pisos, área total, año de construcción/ampliación, ocupación y tipo de suelo.
8. Evaluación de vulnerabilidad sísmica (para Inspector): Completar formulario FEMA con detalles como irregularidades verticales/planta, código sísmico, matriz de resultados, extensión de revisión, peligros geológicos y acciones requeridas.
9. Búsqueda de edificios o informes: Realizar búsquedas por inspector, edificio o código de informe.
10. Visualización de detalles de edificio o informe: Ver información completa de un edificio, incluyendo inspectores, fecha, resultados y opciones para ver detalles o resultados.
11. Consulta de informes FEMA (para Lector): Acceso a pantallas de consulta para ver resultados de evaluaciones existentes.
12. Visualización de resultados de evaluación: Mostrar matriz de resultados, índice sísmico y recomendaciones (e.g., evaluación estructural detallada).
13. Subida de archivos gráficos/fotos: Adjuntar imágenes o gráficos durante el registro o evaluación de edificios.

7. MÓDULOS

7.1. Módulo de Autenticación de Usuario

Este módulo gestiona el acceso de los usuarios al sistema. Su función principal es validar la identidad del usuario a través de la autenticación de credenciales, como el **nombre de usuario** y la **contraseña**.

Además de las funciones básicas de inicio de sesión, el módulo incluye las siguientes opciones para mejorar la experiencia de usuario y la seguridad:

- **¿Olvidaste tu contraseña?**: Permite a los usuarios restablecer su contraseña si la han olvidado. Esto puede redirigirlos a un formulario de recuperación o a un proceso de envío de enlace al correo electrónico.
- **¿Aún no tienes una cuenta?**: Dirige a los usuarios nuevos al proceso de registro.
- **Registrarse**: Inicia el flujo de creación de una nueva cuenta de usuario, donde se solicitará información como nombre, correo electrónico y una contraseña.

7.2. Módulo de Registro de Usuarios

Este módulo se encarga de gestionar la creación de nuevas cuentas de usuario en el sistema. Su propósito es capturar y almacenar de manera segura la información necesaria para que un nuevo usuario pueda acceder y utilizar las funcionalidades del proyecto.

El proceso de registro solicita los siguientes datos al usuario:

- **Usuario**: Nombre de usuario único que servirá como identificador para iniciar sesión.
- **Rol**: Define los permisos y el nivel de acceso que tendrá el usuario dentro del sistema.
- **Correo**: Dirección de correo electrónico del usuario, utilizada para verificaciones de cuenta y comunicaciones importantes.
- **Teléfono**: Número de contacto del usuario.
- **Contraseña y Confirmar contraseña**: Permite al usuario establecer una contraseña segura para su cuenta, la cual debe ingresarse dos veces para asegurar que no haya errores de tipeo.

Una vez completado el registro, el módulo valida la información y crea una nueva entrada en la base de datos, lo que permite al usuario iniciar sesión en el **Módulo de Autenticación de Usuario**.

7.3. Módulo de Consulta de Edificaciones

Este módulo provee una interfaz para buscar y visualizar la información detallada de todas las edificaciones evaluadas o programadas para ser evaluadas. Su objetivo es facilitar el acceso a

los datos de las inspecciones sísmicas, permitiendo a los usuarios generar informes completos de cada estructura.

El módulo ofrece dos criterios de búsqueda principales:

- **Búsqueda por Inspector:** Permite a los usuarios encontrar edificios basándose en el inspector asignado.
- **Búsqueda por Edificio:** Permite buscar una edificación específica usando su nombre, código u otro identificador.

Una vez que se ha realizado la búsqueda, se muestra un listado de edificaciones que coinciden con los criterios. Al seleccionar un edificio del listado, el sistema despliega un resumen del informe de inspección que incluye el código de informe, la fecha de la evaluación y el nombre del inspector. Este resumen también puede mostrar imágenes relevantes de la edificación.

Finalmente, el módulo incluye dos opciones para obtener más detalles:

- **Ver Detalle del Informe:** Muestra el informe completo de la evaluación, que contiene datos técnicos, observaciones y recomendaciones del inspector.
- **Ver Resultados:** Presenta los resultados de la evaluación de vulnerabilidad sísmica, incluyendo la calificación o el nivel de riesgo asignado a la estructura.

7.4. Módulo de Registro de Edificios

Este módulo gestiona la entrada de datos de las edificaciones que serán evaluadas. Su propósito es capturar de manera sistemática toda la información relevante de cada estructura, lo que servirá como base para los análisis de vulnerabilidad sísmica.

El proceso de registro incluye:

- **Archivos adjuntos:**
 - Gráfico:** Archivo de la vista de la edificación (plano o diagrama).
 - Foto del edificio:** Imagen fotográfica de la edificación.
- **Detalles básicos:**
 - Nombre del edificio:** Nombre de la edificación a registrar.
 - Dirección:** Ubicación completa de la estructura.
 - Código postal:** Código postal asociado a la dirección.
 - Uso del edificio:** Categorización del uso (ej. residencial, comercial, industrial, educativo, etc.).
 - Latitud y longitud:** Coordenadas geográficas exactas de la edificación para su geolocalización.
- **Detalles técnicos:**
 - Número de pisos:** Cantidad de pisos o niveles de la edificación.

Área total de piso (m²): Área construida total del edificio expresada en metros cuadrados.

Año de construcción: Año en el que se finalizó la construcción inicial del edificio.

Ampliación: Opción de Sí o No. Si se selecciona Sí, se debe ingresar el **año de ampliación**.

- **Información del registro:**

Inspector: Nombre del inspector a cargo de la evaluación.

Fecha y hora del registro: Momento exacto en que se ingresan los datos al sistema.

- **Tipo de ocupación:** Categorización del uso del edificio, incluyendo las opciones: asamblea, comercial, servicios empresariales, industria, oficina, escuela, almacén, residencial y herramientas.
- **Tipo de suelo:** Clasificación del suelo donde se encuentra la edificación. Los tipos se categorizan por letras:

A: Roca dura

B: Roca semi-dura

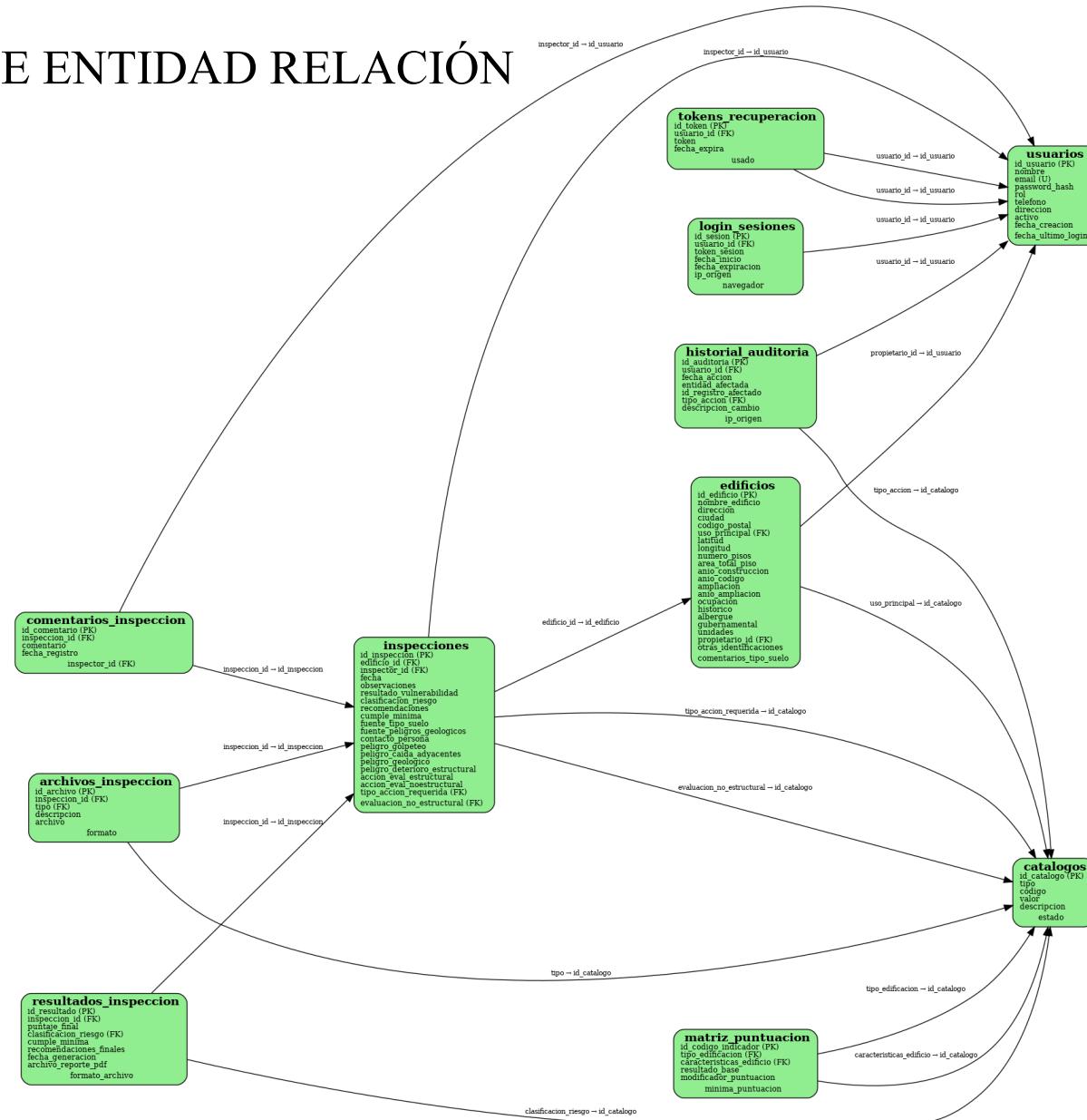
C: Suelo denso

D: Suelo rígido

E: Suelo blando

F: Suelo pobre

8. MODELO DE ENTIDAD RELACIÓN



9. DICCIONARIO DE DATOS DEL MODELO ENTIDAD RELACIÓN

1. Autenticación y seguridad

1.1. Usuarios

Tabla de Usuarios								
<u>id_usuario</u>	<u>Identificador único del usuario</u>	UUID	=	<u>true</u>	<u>false</u>	<u>true</u>	=	<u>Autoincremental</u>
<u>cedula</u>	<u>Número de cédula del usuario</u>	VARCHAR	10	<u>flase</u>	<u>false</u>	<u>true</u>	=	<u>Solo números, longitud 10 dígitos ('0923456789')</u>
<u>nombre</u>	<u>Nombre completo del usuario</u>	VARCHAR	100	<u>false</u>	<u>false</u>	<u>false</u>	=	<u>Solo letras y espacios ('Carlos Pérez')</u>
<u>photo_perfil</u>	<u>Fotografía de perfil del usuario</u>	VARCHAR	255	<u>false</u>	<u>true</u>	<u>false</u>	=	<u>URL o ruta de la imagen ('/images/perfil/usuario1.png')</u>

<u>email</u>	<u>Correo electrónico para autenticación</u>	<u>VARCHAR</u>	<u>150</u>	<u>false</u>	<u>false</u>	<u>true</u>	<u>=</u>	<u>Formato válido de email ('usuario@correo.com')</u>
<u>password_hash</u>	<u>Contraseña cifrada del usuario</u>	<u>VARCHAR</u>	<u>255</u>	<u>false</u>	<u>false</u>	<u>false</u>	<u>=</u>	<u>Hash bcrypt o equivalente</u>
<u>rol</u>	<u>Tipo de usuario en el sistema</u>	<u>VARCHAR</u>	<u>20</u>	<u>false</u>	<u>false</u>	<u>false</u>	<u>=</u>	<u>Valores: 'admin', 'inspector', 'ayudante' Pendiente.</u> <u>Debe existir al menos un admin. El rol inicial de un nuevo usuario es Pendiente</u>
<u>telefono</u>	<u>Número telefónico</u>	<u>VARCHAR</u>	<u>20</u>	<u>false</u>	<u>true</u>	<u>false</u>	<u>=</u>	<u>Formato E.164 (+593987654321)</u>
<u>direccion</u>	<u>Dirección física del usuario</u>	<u>VARCHAR</u>	<u>255</u>	<u>false</u>	<u>true</u>	<u>false</u>	<u>=</u>	<u>Ej: 'Av. Central 123, Quito'</u>
<u>activo</u>	<u>Estado del usuario (activo/inactivo)</u>	<u>BOOLEAN</u>	<u>=</u>	<u>false</u>	<u>false</u>	<u>false</u>	<u>=</u>	<u>DEFAULT TRUE. Si es FALSE, no puede iniciar sesión</u>

<u>fecha_creacion</u>	<u>Fecha y hora de registro del usuario</u>	timestamptz	=	<u>false</u>	<u>false</u>	<u>false</u>	=	<u>DEFAULT now()</u>
<u>fecha_ultimo_login</u>	<u>Fecha y hora del último inicio de sesión</u>	timestamptz	=	<u>false</u>	<u>true</u>	<u>false</u>	=	<u>Se actualiza en cada inicio exitoso</u>

Roles propuestos

- **Admin (Super Admin / Coordinador General)**

- Asigna roles (inspector, ayudante).
- Gestiona catálogos (tipos de suelo, materiales, tipologías).
- Ve auditoría completa.

- **Inspector (rol operativo principal)**

- Realiza inspecciones y genera reportes.
- Puede asignar ayudantes a sus inspecciones.

- **Ayudante (rol de apoyo)**

- Registra datos de campo, sube fotos, llena formularios.
- No puede cerrar ni validar inspecciones.

- **Pendiente (rol inicial)**

- Solo puede loguearse, completar su perfil, y esperar asignación de rol.

1.2. Tokens de recuperación

Tokens para recuperación de contraseña

<u>id_token</u>	<u>Identificador único del token</u>	<u>integer</u>	<u>=</u>	<u>true</u>	<u>false</u>	<u>true</u>	<u>=</u>	<u>Autoincremental</u>
<u>usuario_id</u>	<u>Usuario que solicitó el token</u>	<u>integer</u>	<u>=</u>	<u>false</u>	<u>false</u>	<u>false</u>	<u>usuarios.usuario_id</u>	<u>Debe existir en usuarios</u>
<u>token</u>	<u>Código único para recuperación</u>	<u>varchar</u>	<u>255</u>	<u>false</u>	<u>false</u>	<u>true</u>	<u>=</u>	<u>Hash aleatorio</u>

<u>fecha_expira</u>	<u>Fecha y hora de expiración</u>	timestamp ptz	=	false	false	false	=	2025-08-12 10:00:00
usado	<u>Indica si ya fue usado</u>	boolean	=	false	false	false	=	Default FALSE

2. Formulario

2.1. Entity Name: edificios

Entity Description: Información general y única de cada edificación registrada en el sistema.

Formulario para la creación de edificios								
id_edificio	Identificador único del edificio	UUID	-	true	false	true	-	Autoincremental
nombre_edificio	Nombre o referencia del edificio	VARCHAR	100	false	false	false	-	Ej: "Torre Central"
direccion	Dirección completa	VARCHAR	255	false	false	false	-	Ej: "Av. Central 123"
ciudad	Ciudad donde se ubica	VARCHAR	100	false	false	false	-	Ej: "Quito"

codigo_postal	Código postal de la ubicación	VARCHAR	10	false	true	false	-	Ej: "110101"
uso_principal	Tipo de uso principal del edificio	VARCHAR	50	false	false	false	catalogos.id_catalogo	Ej: "Residencial", "Comercial"
latitud	Coordenada geográfica norte-sur	DECIMAL	9,6	false	false	false	-	Ej: -0.225219
longitud	Coordenada geográfica este-oeste	DECIMAL	9,6	false	false	false	-	Ej: -78.524948
numero_pisos	Número total de pisos incluyendo subsuelos	SMALLINT	-	false	false	false	-	Ej: 12
area_total_piso	Área total construida en m ²	NUMERIC	10,2	false	false	false	-	Ej: 1500.50
anio_construcion	Año en que se construyó el edificio	SMALLINT	-	false	false	false	-	Ej: 1998
anio_codigo	Año del código de construcción aplicado	SMALLINT	-	false	false	false	-	Ej: 2011
ampliacion	Indica si el edificio ha tenido ampliaciones	BOOLEAN	-	false	false	false	-	Valores: true/false

anio_ampliacion	Año en que se realizó la ampliación	SMALLINT	-	false	true	false	-	Solo aplica si ampliacion = true
ocupacion	Actividad principal	VARCHAR	50	false	false	false	-	Ej: "Oficinas"
historico	Indica si el edificio tiene valor histórico	BOOLEAN	-	false	false	false	-	Valores: true/false
albergue	Indica si es usado como albergue	BOOLEAN	-	false	false	false	-	Valores: true/false
gubernamental	Indica si pertenece a una entidad gubernamental	BOOLEAN	-	false	false	false	-	Valores: true/false
unidades	Número de unidades (viviendas/locales)	SMALLINT	-	false	false	false	-	Ej: 20
propietario_id	Usuario propietario o cliente	INT	-	false	false	false	usuarios.id_usuario	Debe existir en usuarios
otras_identificaciones	Identificadores adicionales	TEXT	-	false	true	false	-	Ej: "Parcela 45, Registro Catastral 1234"
comentarios_tipo_suelo	Comentarios adicionales sobre el tipo de suelo	TEXT	-	false	true	false	-	Ej: "Suelo denso con capa superficial arenosa"

2.2. Entity Name: inspecciones

Entity Description: Registro detallado de cada visita de inspección realizada a un edificio.

Detalles de la Inspección								
Nombre del Campo	Descripción	Tipo de Datos	Llave Extranjera	Notable	Notable	Notable	Notable	Notable
id_inspeccion	Identificador único de inspección	UUID	-	true	false	true	-	Autoincremental
edificio_id	Edificio inspeccionado	INT	-	false	false	false	edificios.id_edificio	Debe existir en edificios
inspector_id	Usuario inspector	INT	-	false	false	false	usuarios.id_usuario	Debe existir en usuarios
fecha	Fecha y hora de la inspección	TIMESTAMP	-	false	false	false	-	Ej: "2025-08-11 14:30:00"
observaciones	Observaciones generales	TEXT	-	false	true	false	-	Ej: "Desgaste en fachada"
resultado_vulnerabilidad	Puntaje de vulnerabilidad calculado (FEMA)	DECIMAL	5,2	false	false	false	-	Ej: 3.5
clasificacion_riesgo	Nivel de riesgo	VARCHAR	20	false	false	false	-	Valores: "Bajo", "Medio", "Alto"

recomendaciones	Acciones sugeridas	TEXT	-	false	true	false	-	Ej: "Reforzar estructura"
cumple_minima	Indica si la puntuación cumple el mínimo FEMA	BOOLEAN	-	false	false	false	-	Valores: true/false
fuente_tipo_suelo	Fuente para determinar el tipo de suelo	TEXT	-	false	true	false	-	Ej: "Estudio geotécnico 2023"
fuente_peligros_geologicos	Fuente para peligros geológicos	TEXT	-	false	true	false	-	Ej: "Mapa IGEPN 2024"
contacto_persona	Contacto que proporcionó información	TEXT	-	false	true	false	-	Ej: "Arq. Luis Pérez"
peligro_golpeteo	Riesgo de golpeteo con edificios cercanos	BOOLEAN	-	false	true	false	-	Valores: true/false
peligro_caida_adyacentes	Riesgo de caída de edificios adyacentes	BOOLEAN	-	false	true	false	-	Valores: true/false
peligro_geologico	Peligro geológico o suelo tipo F	BOOLEAN	-	false	true	false	-	Valores: true/false

peligro_deterioro_estructural	Deterioro estructural significativo	BOOLEAN	-	false	true	false	-	Valores: true/false
accion_eval_estructural	Requiere evaluación estructural detallada	BOOLEAN	-	false	true	false	-	Valores: true/false
accion_eval_noestructural	Requiere evaluación no estructural	BOOLEAN	-	false	true	false	-	Valores: true/false
tipo_accion_requerida	Tipo de acción requerida según FEMA	VARCHAR	255	false	true	false	catalogos.id_catalogo	Ej: "Si, resultado menor al mínimo"
evaluacion_no_estructural	Evaluación de elementos no estructurales según FEMA	VARCHAR	255	false	true	false	catalogos.id_catalogo	Ej: "No se sabe"

2.3. Entity Name: archivos_inspeccion

Entity Description: Archivos asociados a una inspección, como fotos, planos y documentos.)

Archivos asociados a una inspección								
Nombre de campo	Descripción	Tipo de dato	Llave	Autoincremental	Not null	Default	Relación	Restricciones
id_archivo	Identificador único	UUID	-	true	false	true	-	Autoincremental
inspeccion_id	Inspección asociada	INT	-	false	false	false	inspecciones.id_inspeccion	Debe existir en inspecciones
tipo	Tipo de archivo	VARCHAR	20	false	false	false	catalogos.id_catalogo	Ej: "foto_fachada", "plano"
descripcion	Descripción opcional	TEXT	-	false	true	false	-	Ej: "Vista principal desde el norte"
archivo	Archivo binario o ruta	BYTEA	-	false	true	false	-	Formatos: JPEG, PNG, PDF
formato	Formato del archivo	TEXT (URL de Supabase Storage)	10	false	true	false	-	Ej: "JPEG"

2.4. Entity Name: comentarios_inspeccion

Entity Description: Observaciones textuales realizadas durante una inspección.

Comentarios de Inspección								
Nombre Columna	Descripción	Tipo de dato	Not null	Default	Unique	Index	Relación	Restricciones
id_comentario	Identificador único	UUID	-	true	false	true	-	Autoincremental
inspeccion_id	Inspección asociada	INT	-	false	false	false	inspecciones.id_inspeccion	Debe existir en inspecciones
comentario	Texto con la observación	TEXT	-	false	false	false	-	Ej: "Suelo tipo D por falta de información"
fecha_registro	Fecha y hora registro	timestamptz	-	false	false	false	-	Formato: YYYY-MM-DD HH:MM:SS
inspector_id	Inspector que ingresó	INT	-	false	false	false	usuarios.id_usuario	Debe existir en usuarios

2.5. Entity Name: catalogo_inspeccion

Entity Description: Lista unificada de catálogos y sus elementos para uso en campos controlados.

<u>id_catalogo</u>	<u>Identificador único</u>	UUID	=	<u>true</u>	<u>false</u>	<u>true</u>	=	<u>Autoincremental</u>
<u>tipo</u>	<u>Tipo de catálogo</u>	VARCHAR	<u>50</u>	<u>false</u>	<u>false</u>	<u>false</u>	=	Ej: "tipo_suelo", "tipo_material", "tipo_archivo", "tipo_accion_requerida"
<u>codigo</u>	<u>Código del elemento</u>	VARCHAR	<u>20</u>	<u>false</u>	<u>false</u>	<u>false</u>	=	Ej: "SUELO_A"
<u>valor</u>	<u>Valor descriptivo</u>	VARCHAR	<u>100</u>	<u>false</u>	<u>false</u>	<u>false</u>	=	Ej: "Arcilla"
<u>descripcion</u>	<u>Descripción del elemento</u>	TEXT	=	<u>false</u>	<u>true</u>	<u>false</u>	=	Ej: "Suelo arcilloso de baja resistencia"
<u>estado</u>	<u>Estado del elemento</u>	CHAR	<u>1</u>	<u>false</u>	<u>false</u>	<u>false</u>	=	'A' (Activo) o 'I' (Inactivo)

2.6. Entity Name: matriz_puntuacion

Entity Description: Contiene los valores base y modificadores de puntuación FEMA P-154 para cada tipo y característica de edificación, permitiendo calcular el puntaje final y evaluar si cumple con la mínima puntuación requerida.

Matriz de Puntuación								
<u>id_codigo_indicador</u>	<u>Identificador único del indicador</u>	UUID	=	<u>true</u>	<u>false</u>	<u>true</u>	=	<u>Autoincremental</u>
<u>tipo_edificacion</u>	<u>Tipo de edificación según catálogo FEMA</u>	VARCHAR	10	<u>false</u>	<u>false</u>	<u>false</u>	<u>catalogos.id_catalogo</u>	Ej: "W1", "S1", "C1", "URM"
<u>caracteristicas_edificio</u>	<u>Código de característica específica del edificio</u>	SMALLINT	=	<u>false</u>	<u>false</u>	<u>false</u>	<u>catalogos.id_catalogo</u>	Ej: 1 = Irregularidad vertical severa, 4 = Pre-código sísmico
<u>resultado_base</u>	<u>Puntaje base conforme al tipo y características de la edificación</u>	NUMBER	8.2	<u>false</u>	<u>false</u>	<u>false</u>	=	Ej: 3.50
<u>modificador_puntuacion</u>	<u>Valor que se suma o resta al puntaje base según la característica</u>	NUMBER	8.2	<u>false</u>	<u>false</u>	<u>false</u>	=	Ej: -0.50 (reduce), +0.30 (aumenta)
<u>minima_puntuacion</u>	<u>Puntaje mínimo permitido para considerar aceptable la edificación</u>	NUMBER	8.2	<u>false</u>	<u>false</u>	<u>false</u>	=	Ej: 2.50

3. Resultados

Entity Name: resultados_inspeccion

Entity Description: Contiene los datos calculados de cada inspección FEMA P-154, incluyendo puntaje final, clasificación de riesgo, recomendaciones y vínculo con el reporte PDF.

id_resultado	Identificador único del resultado	UUID	-	true	false	true	-	Autoincremental
inspeccion_id	Inspección asociada	INT	-	false	false	false	inspecciones.id_inspeccion	Debe existir en inspecciones
puntaje_final	Puntaje final calculado según metodología FEMA P-154	NUMERIC	8,2	false	false	false	-	Ej: 3.25
clasificacion_riesgo	Clasificación final de riesgo	VARCHAR	20	false	false	false	catalogos.id_catalogo	Ej: "Bajo", "Medio", "Alto"
cumple_minima	Indica si cumple la puntuación mínima FEMA	BOOLEAN	-	false	false	false	-	Valores: true/false
factores_modificadores	Factores que aumentaron/disminuyeron el puntaje	TEXT	-	false	true	false	-	Ej: "Irregularidad vertical, Suelo tipo E"

peligros_detectados	Resumen de peligros adicionales observados	TEXT	-	false	true	false	-	Ej: "Golpeteo entre edificios, riesgo por suelos blandos"
recomendaciones_finales	Recomendaciones consolidadas según riesgo y peligros detectados	TEXT	-	false	true	false	-	Ej: "Refuerzo estructural urgente, inspección nivel 2"
fecha_generacion	Fecha y hora en que se generó el resultado	timestamptz	-	false	false	false	-	DEFAULT now()
archivo_reporte_pdf	Ruta o archivo binario del reporte PDF generado	BYTEA	-	false	true	false	-	Formato binario PDF
formato_archivo	Formato del archivo de reporte	VARCHAR	10	false	true	false	-	Ej: "PDF"

4. Historial y Auditoría

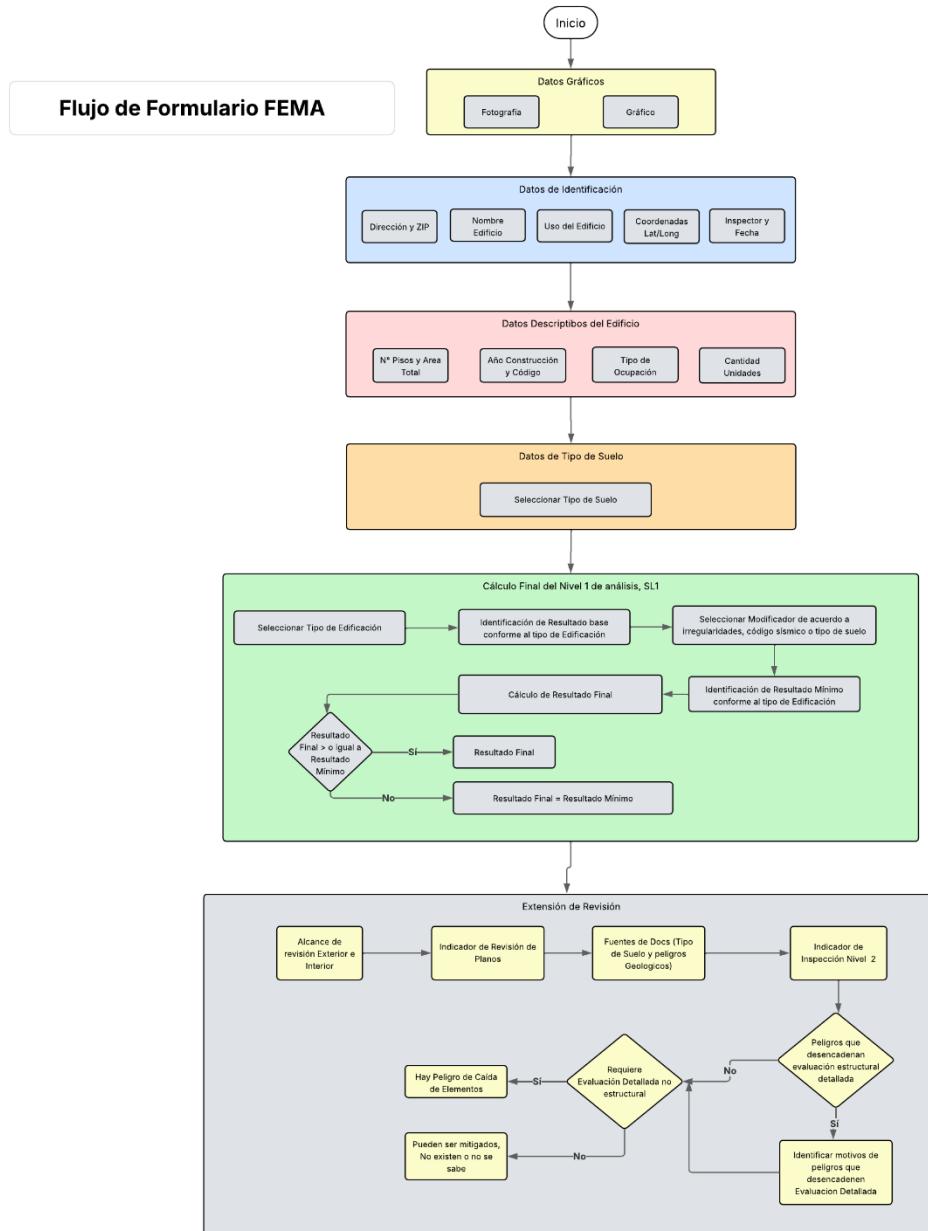
Entity Name: historial_auditoria

Entity Description: Registra cambios realizados en las entidades del sistema, incluyendo usuario que ejecutó la acción, fecha, entidad afectada y descripción del cambio.

Historial y Auditoría								
id_auditoria	Identificador único del registro de auditoría	UUID	-	true	false	true	-	Autoincremental
usuario_id	Usuario que realizó la acción	INT	-	false	false	false	usuarios.id_usuario	Debe existir en usuarios
fecha_accion	Fecha y hora en que se realizó la acción	timestamptz	-	false	false	false	-	DEFAULT now()
entidad_afectada	Nombre de la entidad (tabla) modificada	VARCHAR	100	false	false	false	-	Ej: "edificios", "inspecciones"
id_registro_afectado	Identificador del registro modificado	INT	-	false	false	false	-	Ej: 15
tipo_accion	Tipo de acción ejecutada	VARCHAR	20	false	false	false	catalogos.id_catalogo	Ej: "INSERT", "UPDATE", "DELETE"

descripcion_cambio	Descripción detallada del cambio realizado	TEXT	-	false	true	false	-	Ej: "Se actualizó la dirección del edificio"
valores_anteriores	Valores previos de los campos modificados (formato JSON)	JSONB	-	false	true	false	-	Ej: {"direccion": "Av. Central 123", "telefono": "+593987654321"}
valores_nuevos	Valores actualizados de los campos modificados (formato JSON)	JSONB	-	false	true	false	-	Ej: {"direccion": "Av. Amazonas 456", "telefono": "+593912345678"}
ip_origen	Dirección IP desde la cual se realizó la acción	VARCHAR	45	false	true	false	-	Ej: "192.168.0.15"

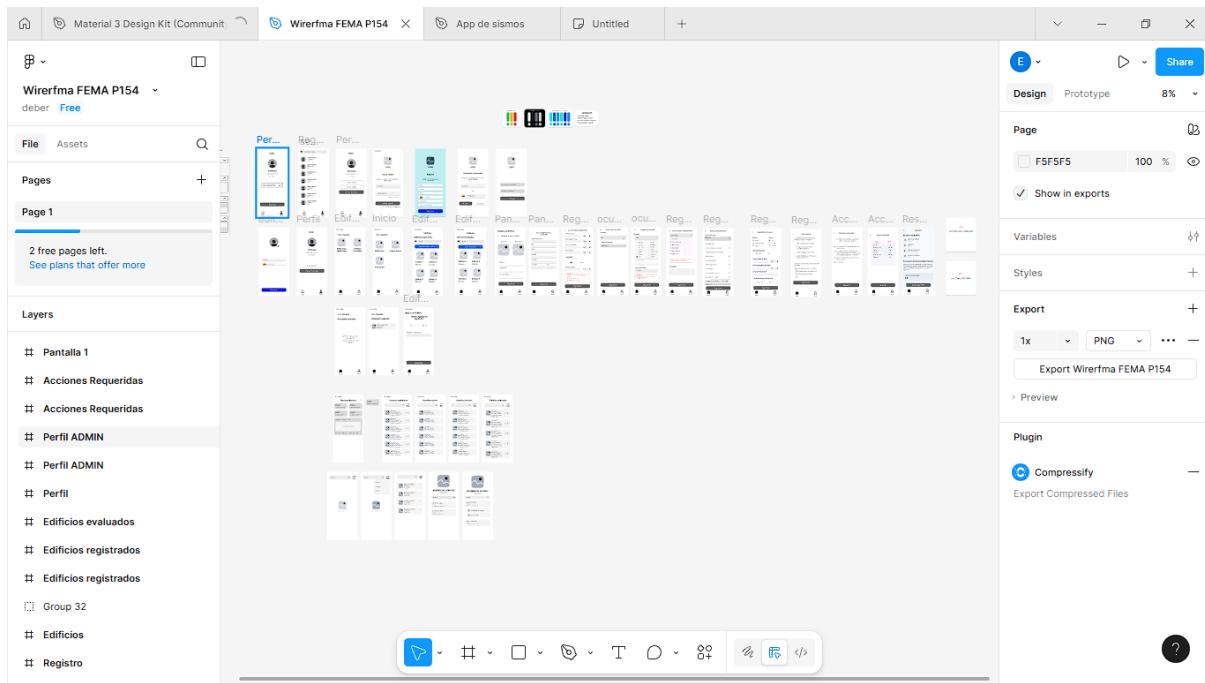
10. Diagrama de flujos



11. PROTOTIPOS DE PANTALLAS DEL APIACTIVO

Enlace de figma donde se encuentra el prototipado:

<https://www.figma.com/design/JN0WS5ulUw3K6Lbw2ujmtD/Wirerfma-FEMA-P154?node-id=0-1&t=U4Ste6H0ATR87FyP-1>



12. Anexo de sugerencias y soluciones a problemas

Problemas de Sdk: Revisar la parte de variables de entorno.

Problemas con la compilación:

Abre el terminal del flutter y ejecuta el siguiente comando:

Flutter config --jdk-dir=(ruta donde esta el jdk)*

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. <https://aka.ms/PSWi>

```
› C:\Users\User\AndroidStudioProjects\untitled1> flutter config --jdk-dir="C:\Program Files\Java\jdk-17"
```