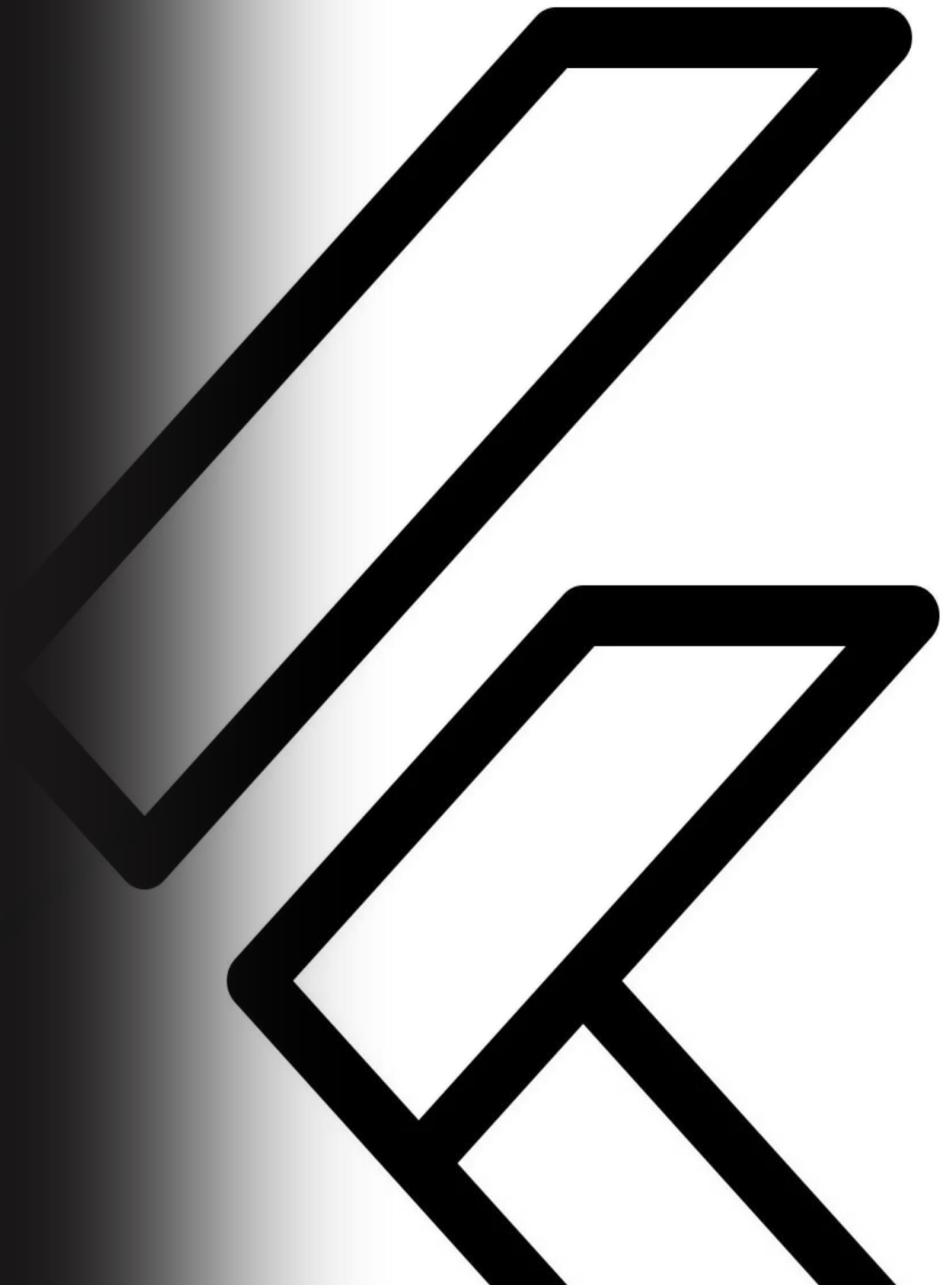


# FLUTTER: Desarrollo Multiplataforma

*Flutter es un **framework de código abierto** de **Google** para crear aplicaciones **multiplataforma** con un solo código base. Desarrolla para **Android, iOS, Web, Windows, macOS** y **Linux** con diseño moderno y alto rendimiento.*

*Utiliza **Dart**, destacando por su rapidez y facilidad para interfaces atractivas.*





# ¿Para qué sirve Flutter?

*Flutter permite desarrollar aplicaciones que funcionan en varios sistemas operativos sin código diferente para cada uno. Es ideal para:*

1

Aplicaciones Móviles

*Android e iOS.*

2

Aplicaciones Web

*Experiencias interactivas.*

3

Aplicaciones de Escritorio

*Windows, macOS, Linux.*

4

Prototipos Rápidos

*Desarrollo ágil.*

# Ventajas de Flutter



Código Único

*Para múltiples  
plataformas.*



Hot Reload

*Cambios en tiempo real.*



Alto Rendimiento

*Cercano al nativo.*



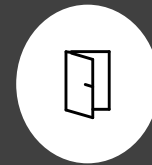
Diseño Flexible

*Con widgets  
personalizables.*



Gran Comunidad

*Soporte de Google.*



Código Abierto

*Y gratuito.*

# Desventajas de Flutter

- *Tamaño inicial de la aplicación algo grande.*
- *Lenguaje Dart menos conocido que otros.*
- *Algunas librerías todavía en desarrollo.*


*Aunque Flutter ofrece muchas ventajas, es importante considerar estos puntos al elegirlo para un proyecto.*



# ¿Qué es Dart?

*Dart es el lenguaje de programación de Flutter. Es moderno, fácil de aprender y optimizado para aplicaciones rápidas y eficientes.*

*Combina características de lenguajes como Java, JavaScript y C++.*

-  *Dart fue también creado por Google, lo que asegura una integración perfecta con Flutter.*



# Dart



# ¿Cómo instalar Flutter?

## Requisitos Previos

- *Computadora (Windows, macOS o Linux)*
- *Conexión a Internet*
- *Git instalado*

## Pasos Básicos

1. *Descargar Flutter (página oficial).*
2. *Extraer en una carpeta.*
3. *Agregar Flutter al PATH.*
4. *Verificar con flutter doctor.*
5. *Instalar Android Studio o VS Code.*

FOLLOW

THESE

STEPS

# Editores Recomendados

*Estos editores ofrecen extensiones que facilitan el desarrollo con Flutter:*

Visual Studio Code

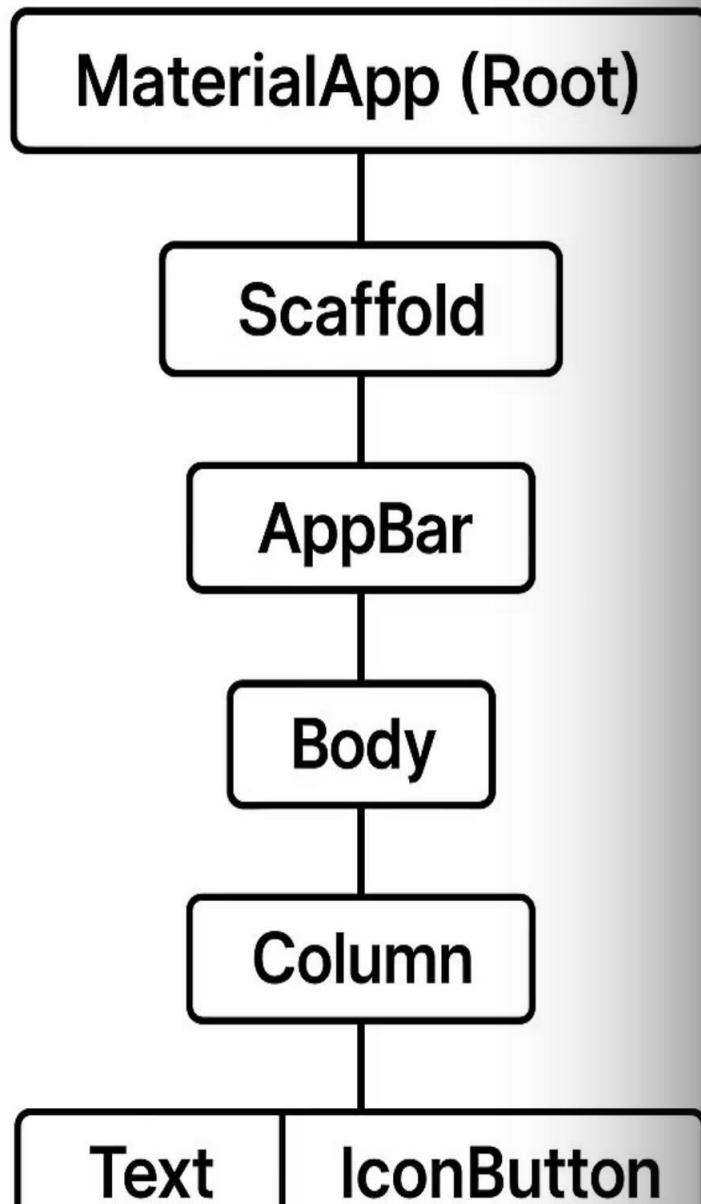
Android Studio

IntelliJ IDEA





# Understand the Widget Tree 🌳



## ¿Cómo funciona Flutter?

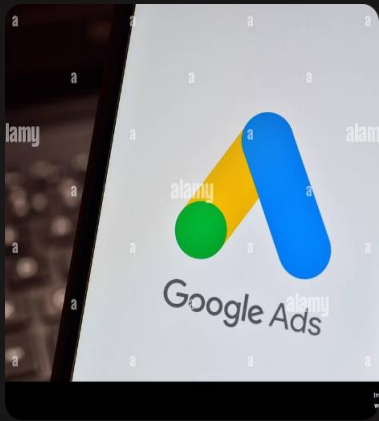
*Flutter se basa en **widgets**. Todo en Flutter es un widget: botones, textos, imágenes, etc. Se combinan para crear interfaces completas.*

*No usa componentes nativos, sino que dibuja su interfaz con el motor gráfico **Skia**, asegurando consistencia en todas las plataformas.*



# Ejemplos de Apps con Flutter

*Grandes empresas confían en Flutter para sus aplicaciones:*



Google Ads

Google Pay



Alibaba



BMW App



eBay Malaysia



# ¿Por qué usar Flutter?

*Flutter es ideal si buscas **rapidez, ahorro de tiempo, buen diseño y compatibilidad multiplataforma.***

*Es una excelente opción para desarrolladores de todos los niveles.*

#DESARROLLO

#MULTIPLATAFORMA

#INNOVACIÓN

# EL PUENTE

Conexión Flutter Backend  
MongoDB





# Introducción

*En el desarrollo de aplicaciones modernas, la seguridad y organización son fundamentales. En Tap Master, implementamos una arquitectura de PUENTE para conectar Flutter con MongoDB de forma segura y profesional.*

## Símbolos de diagrama de flujo



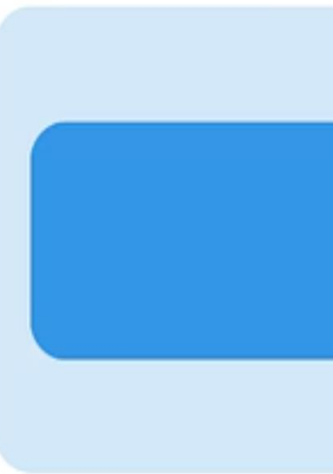
de flujo

el orden que  
lleven las  
tividades u  
operaciones



**Entrada /  
Salida**

Son las lectura de  
los datos de la  
entrada y la  
impresión de  
datos en la salida



**Proceso**

Representa  
operaciones  
cualquier t

# ¿Qué es el PUENTE?

*El PUENTE es el backend desarrollado en Flask (Python) que actúa como intermediario entre:*



Flutter (Frontend)



API REST (Flask)



MongoDB (Base de datos)

*Flutter no se conecta directamente a MongoDB; toda la comunicación pasa por el PUENTE.*

# ¿Por qué no conectar Flutter directamente a MongoDB?

*Conectar una app directamente a la base de datos no es buena práctica por:*

1

Exposición

*Expone la base de datos a internet.*

2

Credenciales

*Revela credenciales sensibles.*

3

Sin Validación

*No permite validar datos.*

4

Falta de Control

*No hay control de lógica ni seguridad.*

*El FUENTE controle todo en el backend, resolviendo estos problemas*



# Cómo funciona el PUENTE



*Este flujo asegura una comunicación controlada y segura entre la aplicación y la base de datos.*

# Funciones principales del PUENTE



Recibir Peticiones

*Desde Flutter.*



Autenticar Usuarios

*Con JWT.*



Aplicar Lógica

*Del sistema y juego.*



Operaciones CRUD

*En MongoDB.*



Proteger Base de Datos

*De accesos directos.*



Enviar Respuestas

*En formato JSON.*

# Seguridad en el PUENTE

*El PUENTE implementa varias medidas de seguridad:*



Contraseñas Encriptadas

*Con bcrypt.*



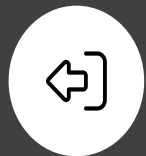
Autenticación JWT

*Tokens de seguridad.*



Expiración de Tokens

*Para mayor seguridad.*



Revocación de Sesiones

*Al cerrar sesión.*



Validación de Datos

*En cada petición.*

*Gracias a esto, los datos de los usuarios permanecen protegidos.*

# Ejemplo práctico del PUENTE (Login)

01

---

Usuario Ingresa Datos

*En la aplicación Flutter.*

02

---

Flutter Envía POST

*A /api/login.*

03

---

PUENTE Valida

*Las credenciales.*

04

---

MongoDB Devuelve

*El usuario.*

05

---

PUENTE Genera Token

*JWT.*

06

---

Flutter Recibe Token

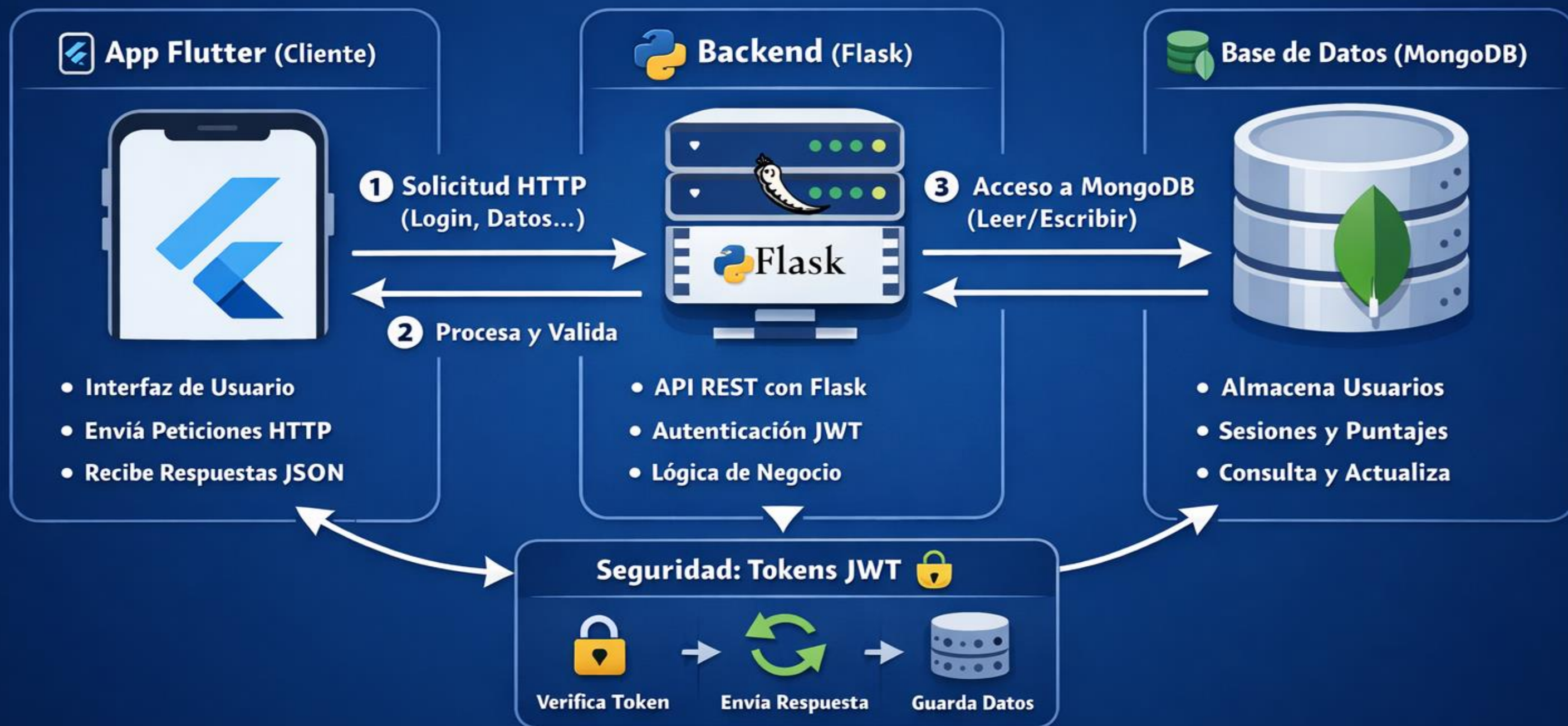
*Y mantiene la sesión.*

# Ventajas del PUENTE

- *Alta seguridad*
- *Arquitectura ordenada*
- *Separación de responsabilidades*
- *Escalabilidad del sistema*
- *Mantenimiento sencillo*
- *Compatible con múltiples plataformas*

# EL PUENTE: Conexión Flutter ↔ Backend ↔ MongoDB

Integración entre Flutter, Flask y MongoDB





# Importancia del PUENTE en Tap Master

*El PUENTE es el núcleo del sistema, permitiendo que Flutter y MongoDB trabajen juntos sin riesgos. Gracias a esta arquitectura, la aplicación puede crecer, agregar nuevas funciones y mantenerse segura.*

