

# **Análisis comparativo entre supabase y pocketbase**

Trabajo en grupo ADI

Hecho por:

Erardo Aldana Pessoa  
David Mas Almendros  
Darío Suárez Domenech

## **Índice:**

1. *¿Qué son los BaaS?*
2. *Pocketbase.*
3. *Supabase.*
4. *Comparación directa.*
  - 4.1. *Tecnologías.*
  - 4.2. *Experiencia de desarrollo.*
  - 4.3. *Escalabilidad.*
  - 4.4. *Despliegue.*
5. *Casos de uso*
6. *Conclusión.*

# 1. ¿Qué son los BaaS?

El Backend as a Service o BaaS es un modelo de servicio en la nube que hace posible externalizar y automatizar el backend. Es una infraestructura de backend configurada para la mayoría de casos que normalmente requiere un backend tradicional.

Esto permite un tiempo de desarrollo menor y hace que los programadores puedan enfocarse en el frontend y la lógica de negocio sin preocuparse por los siguientes aspectos que son **proporcionados por el proveedor**:

- Gestión de la base de datos.
- Autenticación de usuarios.
- Almacenamiento de datos y archivos.
- Actualizaciones en tiempo real.
- Funciones en la nube.

Esto aporta una serie de ventajas frente al uso de un backend común pero también tiene sus desventajas y problemas.

## **Ventajas:**

- Reducción del tiempo de desarrollo.
- Menor complejidad técnica.
- Escalabilidad automática.
- Reducción de costes.

## **Desventajas:**

- Menor flexibilidad.
- Dependencia del proveedor.
- Costes variables.
- Personalización limitada (Menor control).

## 2. Pocketbase.

Pocketbase es una plataforma BaaS de código abierto que está programada en el lenguaje de programación Go. Es ligera y completamente auto alojable ya que se distribuye como un único ejecutable el cual pesa solo aproximadamente 15 MB.

### **Cuenta con:**

- Base de datos (SQLite embebido).
- Autenticación.
- OAuth.
- Comunicación en tiempo real.
- API REST.
- Panel de administración.

## 3. Supabase.

Supabase también es una plataforma BaaS de código abierto pero esta se presenta como una alternativa a Firebase con la diferencia de que se basa en PostgreSQL. En vez de ser un binario único, es una colección de herramientas profesionales que trabajan juntas.

### **Cuenta con:**

- Base de datos PostgreSQL.
- Funciones Edge.
- Suscripciones y almacenamiento en tiempo real.
- Generación de APIs automáticamente.
- Autenticación robusta (OAuth y RLS).
- SDKs oficiales.
- Gran comunidad.
- Varias opciones de despliegue.

## 4. Comparación directa.

### 4.1. Tecnologías.

Característica	Pocketbase	Supabase
Lenguaje	Go	TypeScript, Rust y Go
Base de datos	SQLite embebido	PostgreSQL
Arquitectura	Único binario ejecutable	Sistema distribuido de microservicios
Tamaño	Aproximadamente 15 MB	Varios contenedores Docker
Licencia	MIT	Apache 2.0
Tipo de datos	Limitados a SQLite	Avanzados de PostgreSQL
Consultas	REST	REST, GraphQL, SQL directo

Supabase gana en robustez ya que usa PostgreSQL y esto permite consultas complejas, tipos de datos avanzados y procedimientos almacenados. Pocketbase gana en agilidad gracias a SQLite ya que es más rápido y sencillo.

### 4.2. Experiencia de desarrollo.

Característica	Pocketbase	Supabase
Setup	Descargar y ejecutar	Crear cuenta y proyecto o configurar Docker
Curva de aprendizaje	Baja	Media
Panel de admin	Integrado en localhost	Dashboard web con editor SQL

<b>Desarrollo local</b>	Ejecutar binario	CLI de Supabase o Docker Compose
<b>SDKs</b>	JavaScript/TypeScript y Dart	JavaScript/TypeScript, Dart, Python, Kotlin, Swift, C# y Rust
<b>Documentación</b>	Buena pero menos extensa	Excelente y muy completa
<b>Comunidad</b>	Creciente	Muy grande

Pocketbase gana en setup por su simplicidad, pero Supabase ofrece una experiencia de desarrollo más pulida con mejor documentación, un ecosistema mayor y más opciones en todo.

### 4.3. Escalabilidad.

<b>Característica</b>	<b>Pocketbase</b>	<b>Supabase</b>
<b>Tipo de escalado</b>	Solo vertical	Vertical y réplicas de lectura
<b>Concurrencia</b>	Limitada por SQLite	Alta concurrencia
<b>Capacidad de la base de datos</b>	GBs	Hasta 60TB
<b>Límite de usuarios</b>	Aproximadamente entre 10.000 y 20.000	Millones

Esta es una clara victoria para Supabase ya que es prácticamente la única opción si se planea escalar significativamente o tener una alta concurrencia de escritura.

#### 4.4. Despliegue.

<b>Característica</b>	<b>Pocketbase</b>	<b>Supabase</b>
<b>Complejidad</b>	Muy baja	Media-Alta
<b>Opciones cloud</b>	Auto alojado en un VPS	Servicio gestionado o auto alojado
<b>Requisitos mínimos</b>	VPS con 2 vCPU y 4 GB de RAM	VPS más potente o servicio gestionado
<b>Configuración</b>	Copiar binario y ejecutar	Docker compose o servicio gestionado
<b>Backups</b>	No	Automáticos en tier pagado
<b>Monitorización</b>	Manual	Dashboard integrado en servicio gestionado
<b>SSL/HTTPS</b>	Configurar reverse proxy	Incluido en servicio gestionado
<b>CI/CD</b>	Scripts personalizados	CLI de Supabase

Pocketbase es imbatible en simplicidad de despliegue pero Supabase ofrece características profesionales como backups automáticos, mejor monitorización y SSL/HTTPS incluidos.

## 5. Casos de uso

Después de la comparación directa, se puede ver claramente que cada una de las plataformas está pensada para unos casos de uso distintos.

**Pocketbase** brilla para MVPs y prototipos rápidos o side projects y aplicaciones indie. También es ideal para herramientas internas de empresa y aplicaciones donde no se espera un tráfico mayor a 10.000 usuarios.

**Supabase** es excelente para startups con ambiciones de escalar, aplicaciones que requieran datos relacionales complejos, productos SaaS multi-tenant y aplicaciones en tiempo real colaborativas. También destaca para proyectos que requieran un ecosistema maduro (integraciones y documentación) o aplicaciones con componentes de IA.

## 6. Conclusión.

Tanto Pocketbase como Supabase son soluciones BaaS de código abierto excelentes, pero están pensadas para saciar diferentes necesidades.

**Pocketbase** se destaca por su simplicidad, portabilidad y costes mínimos. Es ideal cuando se necesita:

- Lanzar algo rápido.
- Costes predecibles y bajos.
- Control total y auto alojamiento.

**Supabase** ofrece una plataforma robusta, escalable y con un ecosistema maduro. Es mejor elección cuando se necesita:

- Escalar a millones de usuarios.
- La potencia de PostgreSQL para datos complejos.
- Ecosistema excelente de integraciones.
- Funcionalidades empresariales (RLS, backups, monitoring).