

FRONTEND VS BACKEND

LA ANATOMIA DEL DESARROLLO WEB

INTRODUCCIÓN

EL **DESARROLLO WEB** ES EL MOTOR DE NUESTRO MUNDO DIGITAL, DESDE TIENDAS EN LÍNEA HASTA REDES SOCIALES. TODO LO QUE USAMOS TIENE DOS COMPONENTES ESENCIALES:

Frontend

Es la interfaz (**lo visible**). Todo lo que el usuario ve y con lo que interactúa directamente.

Backend

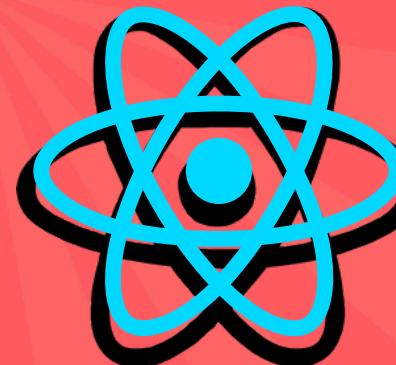
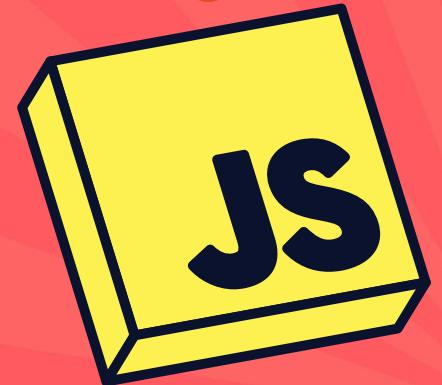
Es la lógica (**lo invisible**). Se encarga de procesar datos, conectar con bases de datos y asegurar el funcionamiento correcto del sistema.

DE LA UNIÓN DE ESTAS DOS ÁREAS SURGE **EL DESARROLLADOR FULLSTACK**, UN PROFESIONAL CAPAZ DE CONSTRUIR PROYECTOS DESDE CERO CON UNA VISIÓN COMPLETA, TRABAJANDO TANTO EN LA INTERFAZ COMO EN LA LÓGICA INTERNA.

¿QUÉ ES BACK END Y FRONT END?



FRONTEND



VS

BACKEND





FRONTEND

EXPO

Presentado por Nicolas Barrancos

¿QUÉ ES?

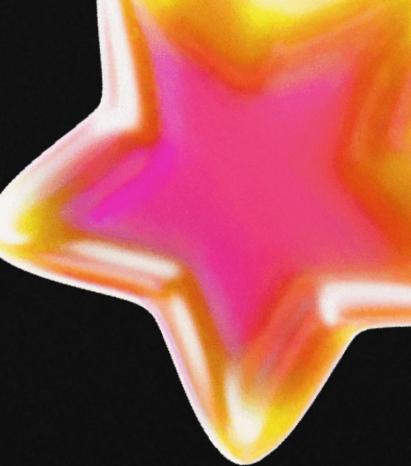


Frontend también conocido como programación del lado del cliente, con la que el usuario interactúa directamente, es decir, todo lo que se ve y se toca en el navegador. A diferencia del Backend, que opera en el servidor

El código Frontend se descarga y ejecuta directamente en el navegador del usuario, actuando como el puente entre el ser humano y la lógica de la máquina.

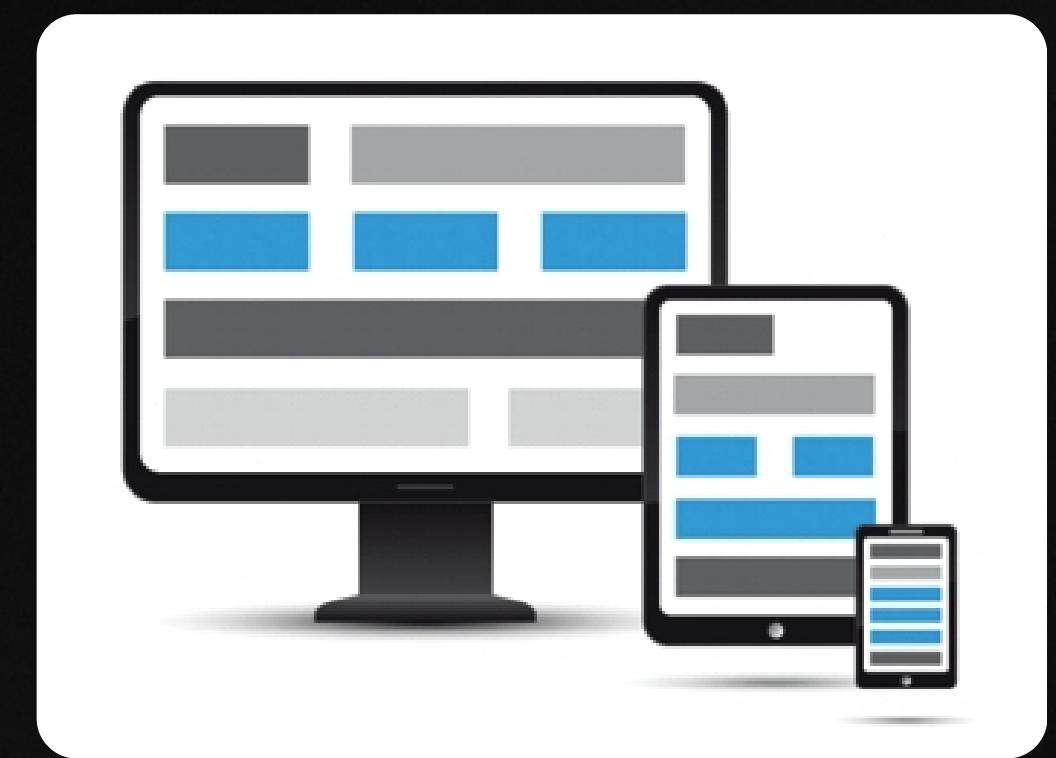


OBJETIVO



EXPERIENCIA DE USUARIO UX & UI:

Implementar diseños que sean intuitivos, guiando al usuario a través de la aplicación sin fricción



DISEÑO RESPONSIVO

Garantizar, mediante técnicas como Media Queries, que la aplicación se adapte fluidamente a diferentes resoluciones de pantalla, desde teléfonos móviles hasta monitores de escritorio



OPTIMIZACIÓN DEL RENDIMIENTO (WPO)

Asegurar tiempos de carga mínimos mediante la compresión de recursos y la ejecución eficiente de scripts, ya que la velocidad impacta directamente en la retención de usuarios

LENGUAJES MAS USADOS EN EL MUNDO



01

HTML5 (HYPERTEXT MARKUP LANGUAGE) QUE DEFINE LA ESTRUCTURA SEMÁNTICA Y EL CONTENIDO



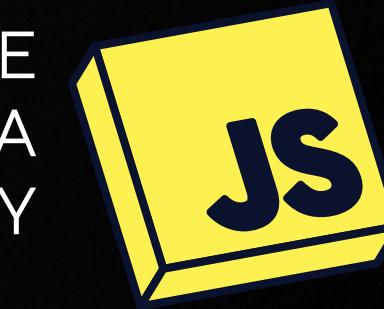
02

CSS3 (CASCADING STYLE SHEETS) QUE DETERMINA LA PRESENTACIÓN VISUAL, COLORES Y TIPOGRAFIAS

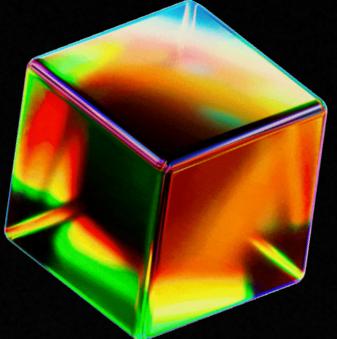


03

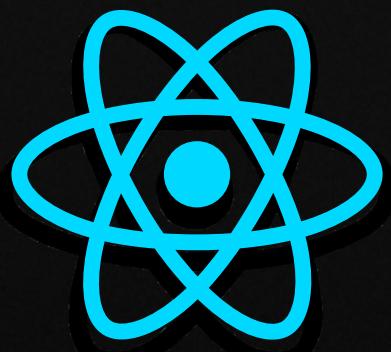
JAVASCRIPT UN LENGUAJE DE PROGRAMACIÓN QUE DOTA AL SITIO DE INTERACTIVIDAD Y LÓGICA DINÁMICA.



LA SANTÍSIMA TRINIDAD



REACT



ANGULAR



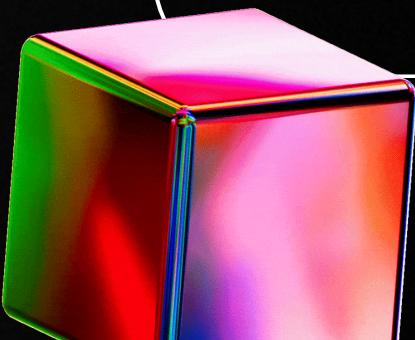
VUE



Construir interfaces de usuario interactivas y reutilizables (componentes).

Construir aplicaciones web complejas y empresariales, con una estructura robusta

Desarrollar interfaces de usuario de forma progresiva, fácil de integrar o usar para proyectos completos

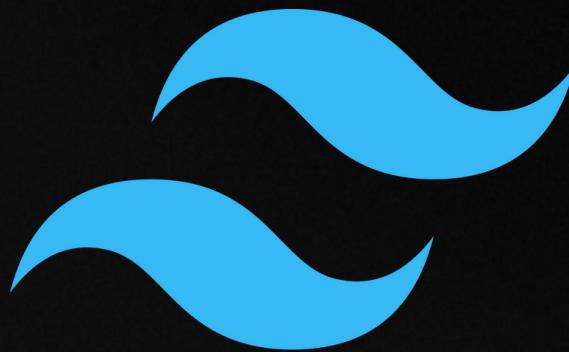


DISEÑO RÁPIDO Y EFICIENTE



BOOTSTRAP

Se basa en componentes prediseñados para desarrollo rápido y estandarizado, con una curva de aprendizaje suave



TAILWIND

Usa clases de utilidad para una personalización total y diseño desde cero, ideal para proyectos únicos, aunque con una curva más pronunciada y archivos más pequeños al final

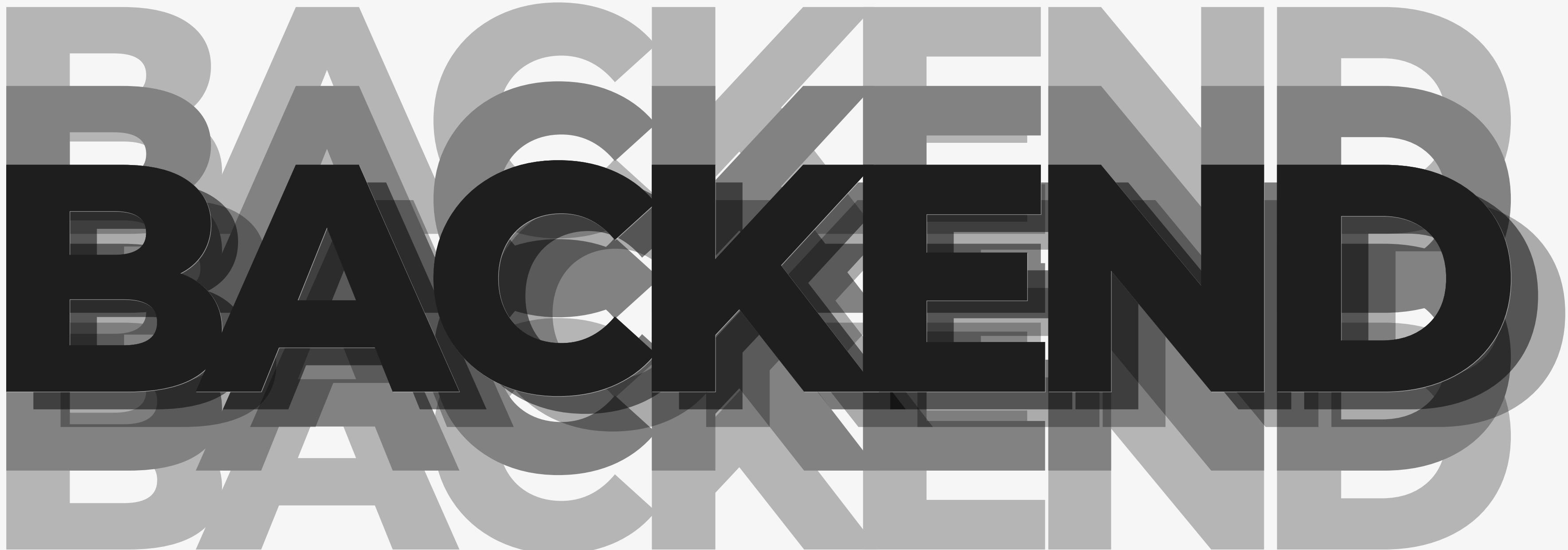


Presentado por



Daniel Maldonado Cespedes

08 de dic , 2025





QUE ES EL BACKEND???

Es la **parte lógica** e invisible de una aplicación web o móvil, que reside en el servidor.

Se encarga de las operaciones detrás de escena, a las que el **usuario final no tiene acceso directo**.

Actúa como **el cerebro de la aplicación** , comunicándose constantemente con el Frontend y la Base de Datos.

OBJETIVO PRINCIPAL DEL BACKEND

Lógica de Negocio: Implementar las reglas y procesos que definen cómo funciona la aplicación (ej. calcular precios, gestionar inventario, autorizar transacciones).

Gestión de Datos: Almacenar, recuperar y manipular datos en la base de datos (CRUD: Crear, Leer, Actualizar, Borrar).

Autenticación y Autorización: Verificar la identidad del usuario (autenticación) y definir a qué recursos puede acceder (autorización).

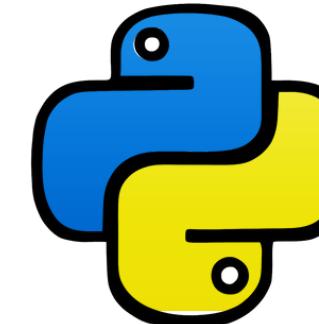
Comunicación: Servir la información al Frontend a través de APIs (Application Programming Interfaces).



Lenguajes de Programación

Estos lenguajes se ejecutan en el servidor y son la base de la lógica.

01 **PYTHON** QUE DEFINE LA LÓGICA DE NEGOCIO Y ES USADO EN CIENCIA DE DATOS.



02 **JAVASCRIPT (NODE.JS)** QUE DEFINE EL MANEJO DE SOLICITUDES ASÍNCRONAS Y EL DESARROLLO FULLSTACK.



03

JAVA QUE DEFINE LA ROBUSTEZ EMPRESARIAL Y LA ESCALABILIDAD DE GRANDES SISTEMAS.



04

PHP (HYPERTEXT PREPROCESSOR) QUE DEFINE EL PROCESAMIENTO EN EL LADO DEL SERVIDOR Y LA GENERACIÓN DINÁMICA DE CONTENIDO.



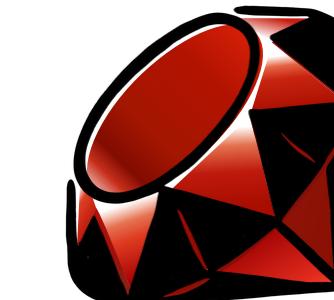
05

GO (GOLANG) QUE DEFINE LA EFICIENCIA DE LA CONCURRENCIA Y EL DESARROLLO DE MICROSERVICIOS.

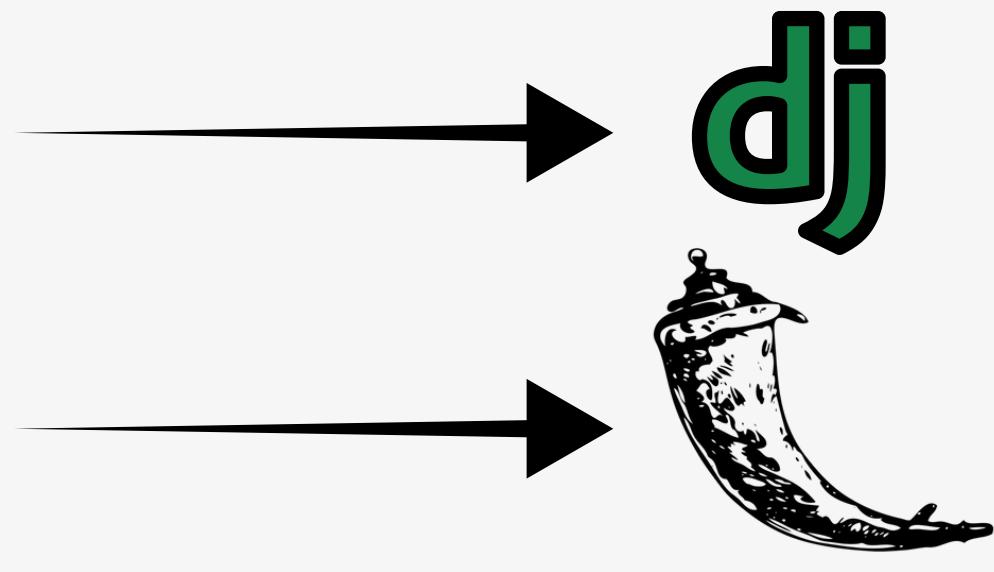


06

RUBY QUE DEFINE LA SINTAXIS EXPRESIVA Y EL DESARROLLO RÁPIDO CON EL FRAMEWORK RAILS.



FRAMEWORKS



Express.js

El framework más popular para Node.js. Excelente para construir APIs REST y microservicios de alto rendimiento gracias a su naturaleza asíncrona.

NestJS

Un framework avanzado, basado en TypeScript. Utiliza patrones de diseño para construir aplicaciones empresariales y escalables de manera modular.



Django

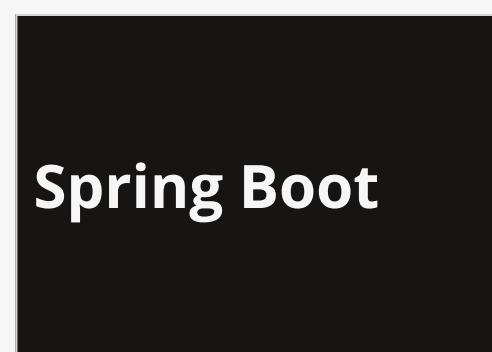
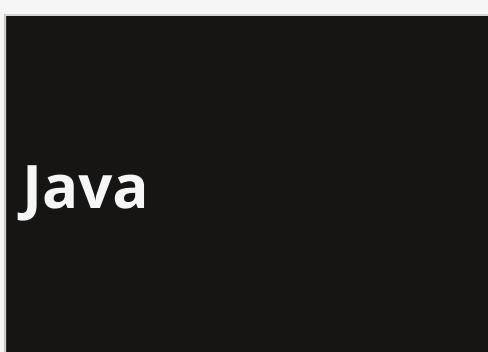
Flask

Un framework "todo incluido". Ideal para proyectos grandes, complejos y con necesidades de seguridad. Sigue el patrón MTV (Model-Template-View).

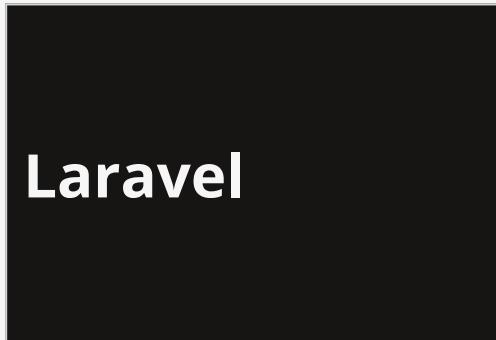
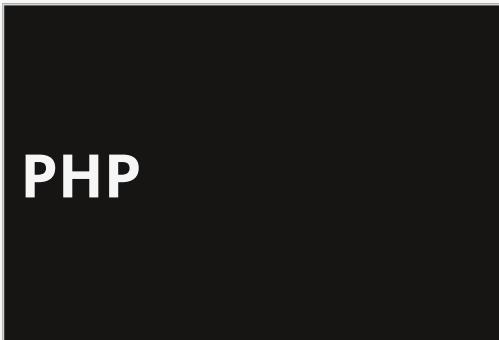
Un micro-framework ligero y minimalista. Permite una gran libertad y flexibilidad, perfecto para prototipos rápidos o APIs sencillas.



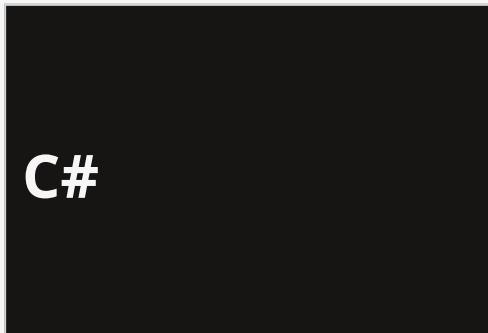
FRAMEWORKS



Domina el sector empresarial. Simplifica drásticamente la creación de aplicaciones robustas, standalone y listas para producción con mínima configuración.



Reconocido por su sintaxis elegante y expresiva. Facilita tareas comunes como la autenticación, routing y caching mediante herramientas integradas.



Moderno, de código abierto y multi-plataforma. Ideal para construir APIs rápidas, Microservicios y aplicaciones empresariales a gran escala.

Bases de Datos

**Relacionales
(SQL)**

PostgreSQL

Muy avanzado, conocido por su robustez y fuerte integridad de datos. Ideal para sistemas que requieren transacciones complejas.

MySQL

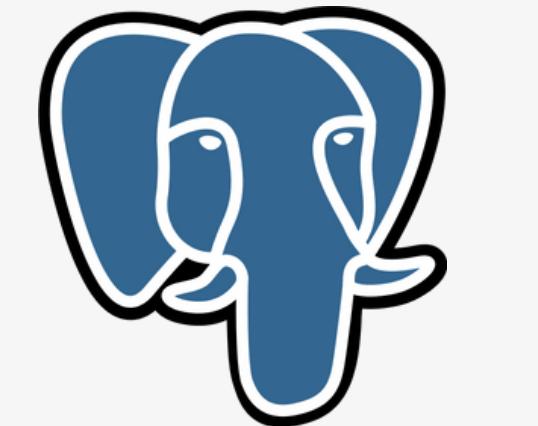
Ampliamente aceptado por la comunidad, conocido por su velocidad y fiabilidad. Utilizado en una vasta cantidad de aplicaciones web (ej. el stack LAMP).



**No Relacionales
(NoSQL)**

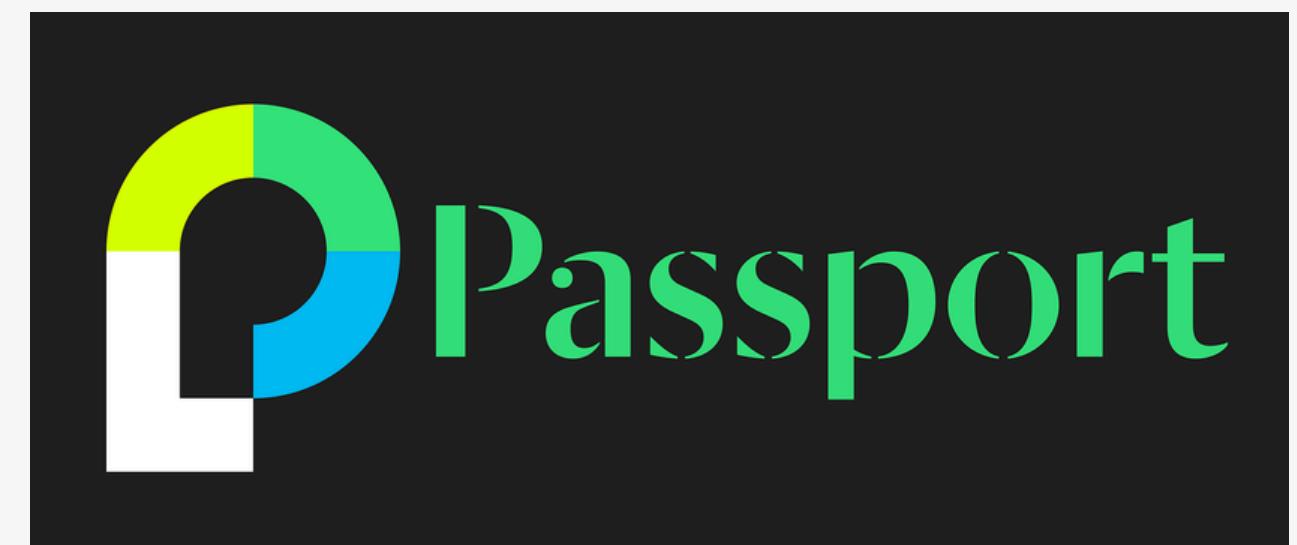
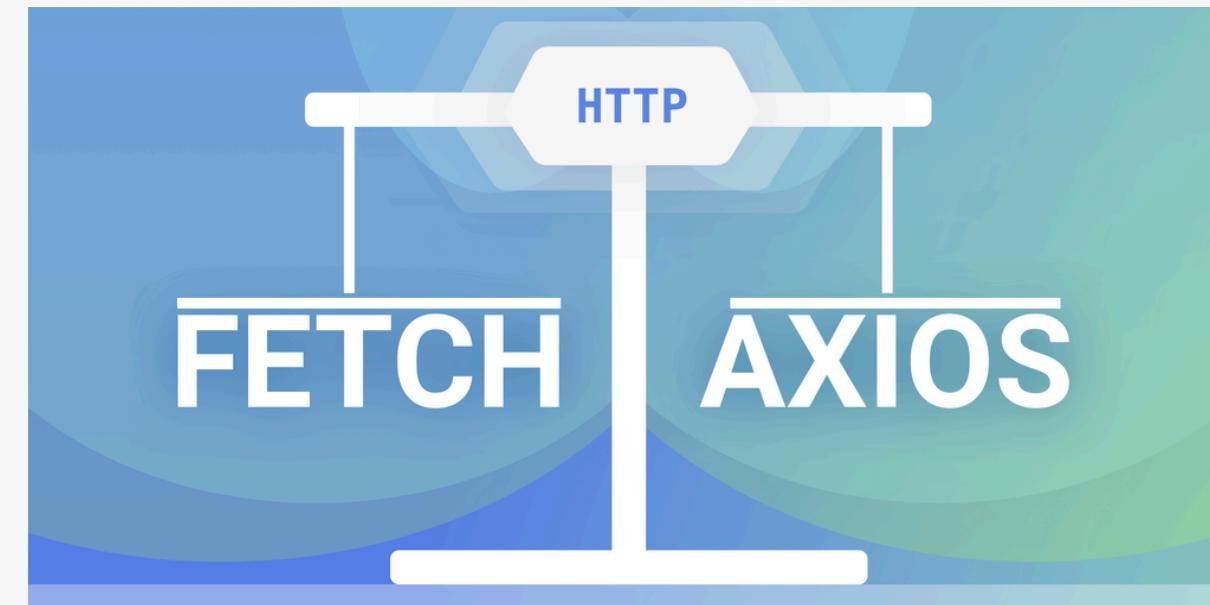
MongoDB

Esta base de datos utiliza un modelo orientado a documentos . Almacena la información en formatos como JSON/BSON, lo que le permite ofrecer una alta flexibilidad de esquema.



LIBRERIAS

- **Axios/Fetch (Isomórfico):** Aunque se usan mucho en Frontend, en Backend (Node.js) se utilizan como librerías para realizar peticiones HTTP a APIs de terceros (ej. Stripe, APIs de clima)
- **Passport.js (Node.js):** Librería modular para manejar la autenticación (ej. login con Google, Facebook, etc.) en aplicaciones Node.js..



BAAS (BACKEND AS A SERVICE)

AWS Amplify /
Azure Mobile
Apps

Cloud BaaS

Soluciones de cloud que ofrecen herramientas para construir Backend y conectar aplicaciones de manera escalable.



 **Firebase**

Firebase
(Google)

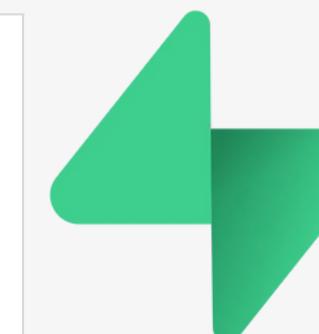
BaaS

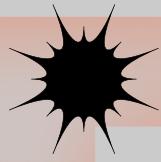
Popular para aplicaciones móviles y Frontend-pesadas. Ofrece base de datos en tiempo real (Firestore) y autenticación fácil.

Supabase

Open Source
BaaS

Se posiciona como una alternativa a Firebase, construida sobre PostgreSQL. Proporciona bases de datos, APIs automáticas y autenticación.

 **supabase**



NICOLAS
BARRANCOS

FRONTEND



BACKEND

DESEARROLLO
FULL STACK

2025

¿QUÉ ES?

Es la capacidad de trabajar y dominar ambas caras de una aplicación, frontend (la parte visual y de interfaz de usuario) como el backend (la lógica de negocio, servidores y bases de datos) de una aplicación web o software



UN DESARROLLADOR FULL STACK PUEDE...

Entender cómo se comunican las APIs del Backend con los componentes de React o Vue en el Frontend, facilitando planificación y la resolución de problemas.

Hacer un prototipado rápido e iterar sobre el producto de manera autónoma, sin depender de un especialista para cada tarea.

Conocer tecnologías en ambos lados (por ejemplo, JavaScript/React y Node.js/Express), y no necesariamente debe ser el mayor experto en cada una, sino un maestro en la integración de todas ellas.

Ejemplos de combinaciones comunes son MERN (MongoDB, Express, React, Node.js) y MEAN (MongoDB, Express, Angular, Node.js)

EJEMPLOS

FRONTEND



POLLEN.DESIGN



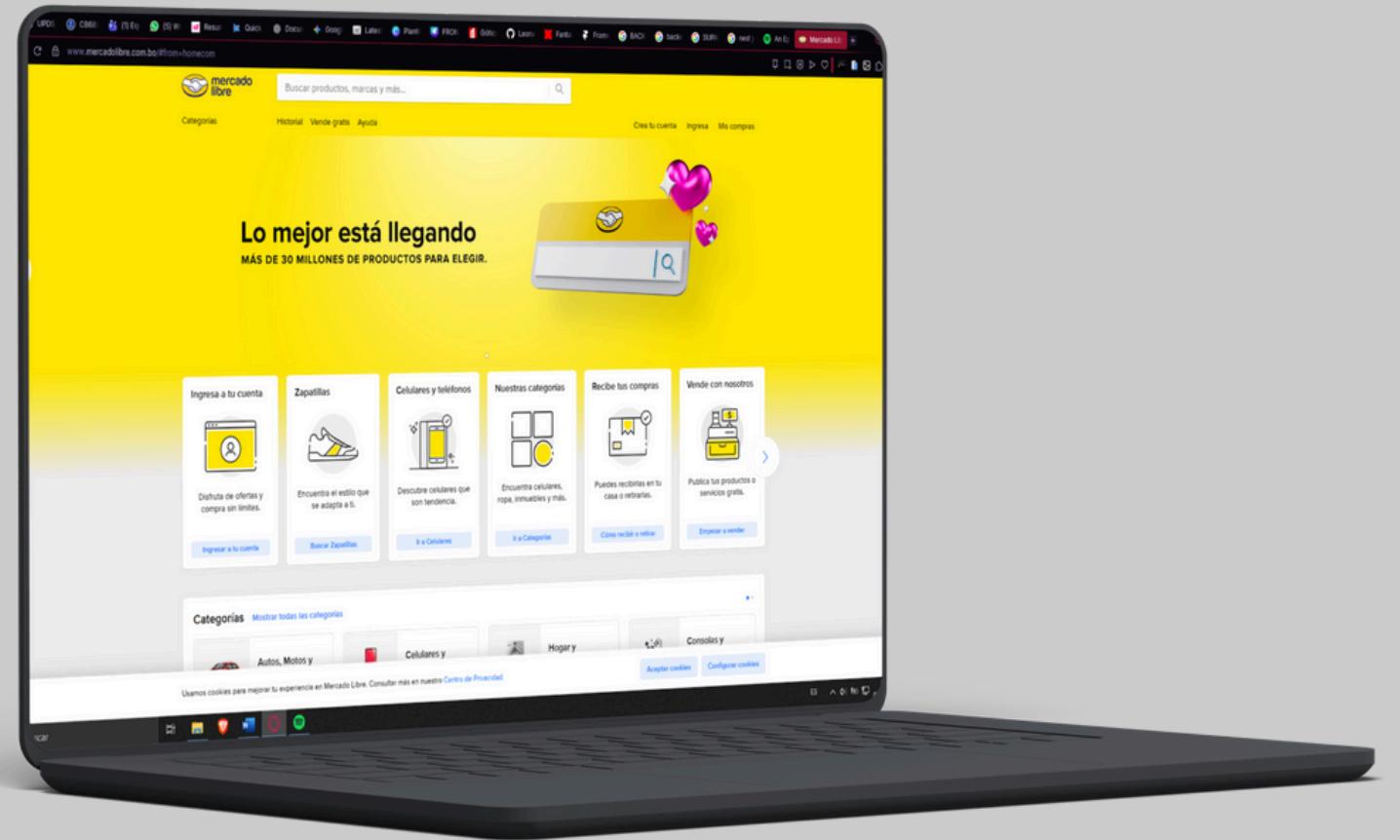
FANTASY.CO

EJEMPLOS

BACKEND



SPOTIFY.COM



MERCADOLIBRE.COM.BO



CONCLUSIONES

EL DESARROLLO WEB MODERNO NO ES UNA TAREA MONOLÍTICA, SINO UNA ORQUESTACIÓN DE ROLES QUE SE COMPLEMENTAN PARA CREAR UNA EXPERIENCIA DIGITAL COMPLETA Y FUNCIONAL.

EL FRONTEND (LA INTERFAZ) ES EL ARTISTA; SU MISIÓN ES LA EXPERIENCIA DEL USUARIO Y LA INTERACCIÓN VISIBLE.

EL BACKEND (LA LÓGICA) ES EL INGENIERO; SU MISIÓN ES LA SEGURIDAD, LA EFICIENCIA Y LA GESTIÓN DE LA INFRAESTRUCTURA DE DATOS.

EL DESARROLLADOR FULLSTACK ACTÚA COMO EL ARQUITECTO, UNIENDO AMBOS MUNDOS. SU VISIÓN COMPLETA PERMITE DISEÑAR Y CONSTRUIR SOLUCIONES DESDE LOS CIMENTOS (BACKEND) HASTA EL ACABADO FINAL (FRONTEND), OPTIMIZANDO LA COHERENCIA Y EL FLUJO DE TODO EL SISTEMA.

FOR
YOUR
AUDIENCE