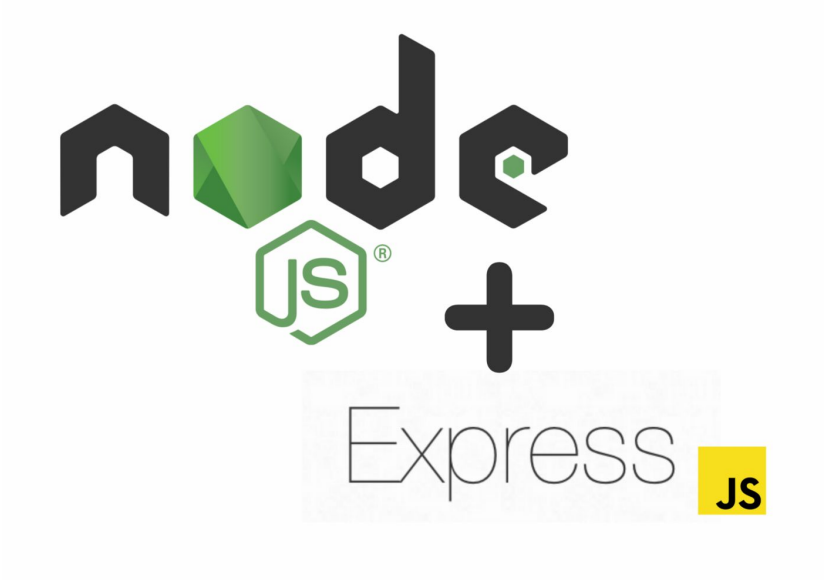
releevant.

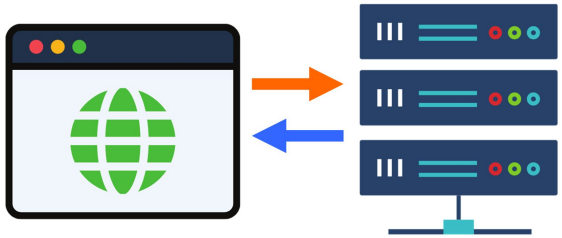


linkedin.com/in/maortizolid

maortizolid@gmail.com

releevant.

Modelo cliente - servidor



Cliente - Cuando intentas acceder a una página web en tu navegador, el navegador (**cliente**) envía una solicitud HTTP al servidor.

Servidor - Es un programa que escucha solicitudes y genera respuestas apropiadas. Estas respuestas incluyen:

- Enviar información al cliente.
- Ejecutar tareas o procesos.
- Trabajar con la información de una base de datos y/o actualizarla.

releevant.

¿Qué es Node.js?

Es un intérprete de JavaScript que se ejecuta en servidor basado en el motor de JavaScript que utiliza Google Chrome (V8), escrito en C++

Node.js NO es un:

- Lenguaje de programación.
- Framework.
- Librería (biblioteca).

En resumen, es un **entorno de ejecución** que se utiliza para ejecutar JavaScript fuera del navegador. Se basa en un modelo de E/S no bloqueante y orientado a eventos.

Impacto que Node.js está teniendo en el área del desarrollo web.
https://survey.stackoverflow.co/2023/#most-popular-technologies-webframe

releevant.

¿Qué es Express?

- Framework de aplicaciones web específicamente diseñado para Node.js.
- Express tiene muchos métodos HTTP útiles y middleware que puedes usar para crear APIs robustas (Application Programming Interfaces), las cuales son fundamentales para el desarrollo web back-end y full-stack.

releevant.

Conceptos básicos de Node.js

Node.js, es el entorno de ejecución de JavaScript por el lado del servidor, construido sobre el motor **JavaScript V8 de Google Chrome.**

Características de Node.js

- **Velocidad -** El código JavaScript que se ejecuta puede llegar a ser el doble de rápido, que el código de los lenguajes compilados como Java o C. Tiempos de ejecución en órdenes más rápidos, que los lenguajes interpretados como Python o Ruby.

- **Simple -** Node.js es una plataforma de bajo nivel, existen miles de bibliotecas, construidas sobre Node.js para hacer las cosas más fáciles e interesantes para los desarrolladores (Frameworks y herramientas).

- **JavaScript -** Node.js ejecuta código **JavaScript** en el lado del servidor permitiendo integrar el mismo lenguaje de programación tanto en FrontEnd como en BackEnd.

- **Plataforma Asíncrona -** JavaScript permite crear código asincrónico y sin bloqueo de una manera muy simple, mediante el uso de un solo hilo, funciones de devolución de llamada y programación controlada por eventos.

releevant.

Ventajas de Node.js.

- **Escalabilidad -** Permite construir aplicaciones que escalan con el negocio, utilizando la arquitectura de microservicios y la contenerización.

- **Rendimiento -** Node.js ofrece ventajas de rendimiento mejoradas gracias al motor V8 de Chrome. Es un sistema de alto rendimiento.

- **Microservicio -** La naturaleza de los microservicios de Node.js es una gran ventaja para el mantenimiento. Si puedes dividir una base de código en pequeños trozos, es más fácil de mantener que una única base de código.

- **Usabilidad -** El proceso de desarrollo de aplicaciones web puede optimizarse y acelerarse cuando se utiliza Node.js debido a su arquitectura de microservicios, la capacidad de utilizar un solo lenguaje de programación tanto en el frontend como en el backend, y la disponibilidad de millones de bibliotecas a través de NPM (gestor de paquetes).

releevant.

Otros entornos de ejecución

Existen otros entornos de ejecución de Javascript que están ganando popularidad como Deno o Bun.

https://geekflare.com/best-javascript-runtime-environments/

releevant.

APIs y para qué se usan.

- **Application Program Interface**. Conjunto de protocolos que permiten conectar distintos softwares y aplicaciones entre sí.

- Permiten conectar con servicios ya existentes para la ejecución de tareas, de tal forma que no es necesario tocar el código o realizar la programación del software desde cero. Esto, a nivel empresarial, se traduce en una mayor estandarización y eficiencia en el uso de los recursos.

releevant.

Instalación de NodeJS

LTS vs. Last (Última Versión)

LTS (Long Term Support): para producción y estabilidad a largo plazo.

- Estabilidad y confiabilidad.
- Soporte a largo plazo (más de 30 meses).
- Actualizaciones menos frecuentes (aproximadamente cada 12 meses).
- Prioriza la compatibilidad con versiones anteriores.

Last (Última Versión): para acceder a las últimas características y mejoras.

- Últimas características y mejoras.
- Actualizaciones frecuentes (aproximadamente cada 6 meses).
- Menos tiempo de soporte.
- Posibles cambios disruptivos en la API y el comportamiento.

releevant.

Descargar e instalar NodeJS

S.O. Windows - https://nodejs.org/es/download/

S.O. Linux

```
sudo apt update
sudo apt install nodejs
sudo apt install npm
```

Verificar la versión de NodeJS

Desde la terminal

```
node --version
node -v
```

releevant.

NVM (Node Version Manager)

- **NVM** es una herramienta para administrar las versiones de Node en nuestro dispositivo.
- Permite instalar diferentes versiones de Node y cambiar entre estas versiones según el proyecto en el que estés trabajando a través de la línea de comando.

Instalar NVM

- Windows: https://github.com/coreybutler/nvm-windows/releases

- Linux/Mac:
  - curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v1.1.11/install.sh | bash
  - # o
  - wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v1.1.11/install.sh | bash

  - # cargar nvm para su uso (si no se ha incluido en el paso anterior):
  - export NVM_DIR="$([-z "${XDG_CONFIG_HOME-}"] && printf %s "${HOME}/.nvm" || printf %s "${XDG_CONFIG_HOME}/nvm")"
  - [-s "$NVM_DIR/nvm.sh"] && \. "$NVM_DIR/nvm.sh"
  - # recargar la configuración para utilizarla en la terminal:
  - source ~/.bashrc

releevant.

Con nvm instalado, ahora podemos instalar, desinstalar y cambiar entre diferentes versiones de Node en nuestro dispositivo:

- Versión de NVM : nvm -v
- Lista de las versiones instaladas en nuestro dispositivo: nvm ls
- Lista de todas las versiones de Node a instalar: nvm ls-remote

- Instalar la última version de Node: nvm install latest
- Instalar la version X.Y.Z de Node: nvm install vX.Y.Z
- Hacer que una versión sea la predeterminada: nvm alias default vX.Y.Z
- Usar una versión específica: nvm use vX.Y.Z

Resumiendo, NVM facilita la gestión de múltiples versiones de Node.js en diferentes proyectos que requieren diferentes versiones.