

Clase 8

Introducción: Primeros pasos con Node

Comenzamos en 10 min

6:40pm (hora CDMX)



| Temas de la clase (180 min)

Node.js

¿Qué es?

¿Por qué es importante?

¿Qué podemos construir con Node?

Node.js en la practica

Instalación

Ejecución de archivos con Node

Servidor con Node.js

Tu primer servidor local

Express

Reto de la semana



| ¿Qué es Node.js?

Node.js es un entorno de ejecución para **JavaScript** basado en el motor V8 de Google Chrome, que permite ejecutar código JavaScript fuera de un navegador. Esto significa que ahora puedes usar JavaScript no solo para desarrollar sitios web interactivos, sino también para crear **aplicaciones de servidor**, herramientas de línea de comandos y mucho más.



DEV.E::



| ¿Por qué usar Node.js?

- **Velocidad:** Gracias al motor V8, Node.js ejecuta JavaScript de forma rápida y eficiente.
- **Modelo no bloqueante:** Utiliza un enfoque asíncrono y basado en eventos, lo que permite manejar muchas solicitudes al mismo tiempo sin bloquear el hilo principal.
- **Versatilidad:** Puedes usar JavaScript en el cliente (navegador) y en el servidor, lo que hace más consistente el desarrollo.
- **Gran comunidad:** Con un ecosistema masivo de paquetes y herramientas disponibles en npm (Node Package Manager), es fácil agregar funcionalidades a tus proyectos.



| ID

¿Para qué sirve? Aplicar estilos a un único elemento en toda la página. Es ideal para elementos con una función específica, como un menú de navegación (**#menu-principal**) o un pie de página (**#footer**).

En resumen:

- Los selectores de **elemento** son para estilos generales.
- Los selectores de **clase** son para estilos específicos en varios elementos.
- Los selectores de **ID** son para estilos únicos en un solo elemento.

DEV.F:



| Usos de Node.js

Node.js es ideal para una variedad de aplicaciones como:

- APIs REST
- Aplicaciones en tiempo real (chats, juegos en línea)
- Herramientas de línea de comandos
- Servidores web
- Procesamiento de datos y tareas en segundo plano



DEV.F:



Agregar imagenes

Si no has instalado Node.js puedes hacerlo desde la página oficial!

Ejecuta JavaScript en cualquier parte

Node.js® es un entorno de ejecución de JavaScript multiplataforma, de código abierto y gratuito que permite a los desarrolladores crear servidores, aplicaciones web, herramientas de línea de comando y scripts.

Descargar Node.js (LTS) 


Descarga Node.js **v22.13.1**¹ con soporte a largo plazo. Node.js también puede ser instalado a través de gestores de paquetes.

¿Quieres nuevas funciones más pronto? Consigue **Node.js v23.6.1**¹ en vez.

Create an HTTP Server Write Tests Read and Hash a File Streams Pipeline Work with Threads

```
1 // server.mjs
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with `node server.mjs`
15
```

JavaScript

 Copiar al portapapeles

Aprenda más sobre lo que Node.js puede ofrecer con nuestros [Materiales de aprendizaje](#).



| Comprobando instalación

Al instalar Node.js debemos verificar si el proceso se realizó de manera correcta, esto lo podemos hacer ejecutando desde nuestra terminal:

- **node -v**

Una vez la consola nos responda con la versión de Node que instalamos podemos proceder al siguiente paso.

```
C:\> Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.5371]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Diego>node -v
v20.11.0

C:\Users\Diego>
```


| Ejecutando archivos con Node

Una vez hayamos verificado que la instalación se realizó correctamente podemos proceder a ejecutar nuestro primer archivo en Node.js con el siguiente comando:

- **node <nombre del archivo>**

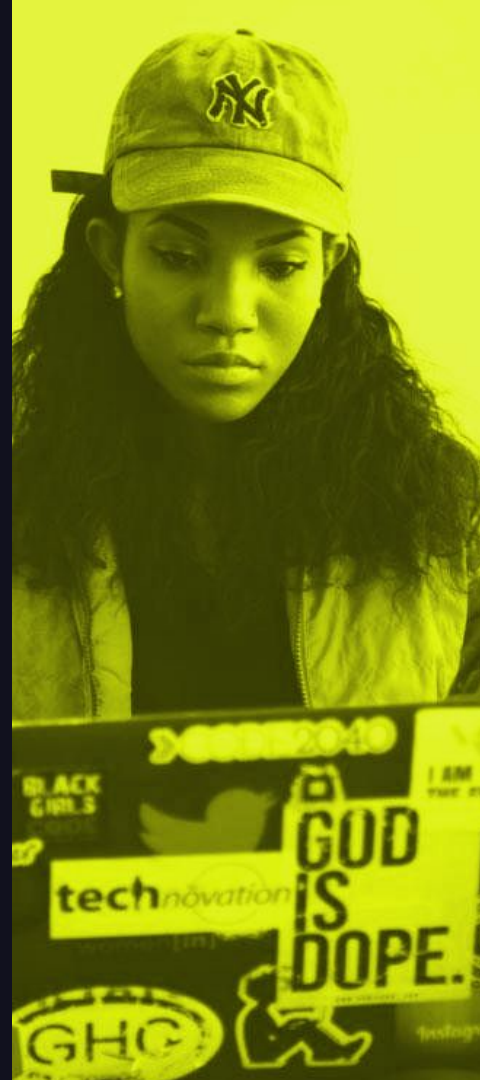
```
• PS C:\Users\Diego\Documents\HTML tests> node .\extra.js  
Este es un archivo de JavaScript ejecutado con Node.js  
❖ PS C:\Users\Diego\Documents\HTML tests> █
```



| Package.json

El archivo `package.json` es un archivo clave en cualquier proyecto de Node.js. Es un documento en formato **JSON** que sirve para gestionar y describir la configuración del proyecto, las dependencias, los scripts, y otra información importante. Básicamente, es el **"manual de instrucciones"** de tu aplicación Node.js.

```
{  
  "name": "sustain_be",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "author": "",  
  "license": "ISC"  
}
```





| Levantando nuestro servidor

Para poder levantar nuestro servidor tenemos que seguir los siguientes pasos:

- **npm init -y**
- **npm i express**
- **npm i nodemon**

Tras ejecutar los comandos anteriores podemos proceder a insertar nuestros scripts en nuestro archivo package.json para poder ejecutarlos de forma más sencilla:

```
1  {  
2    "dependencies": {  
3      "express": "^4.21.2",  
4      "nodemon": "^3.1.9",  
5      "vite-express": "^0.20.0"  
6    },  
7    "scripts": {  
8      "start": "node index.js",  
9      "dev": "nodemon index.js"  
10   }  
11 }  
12
```

| Express.js

Express.js es un **framework** web minimalista y flexible para Node.js. Proporciona un conjunto de herramientas que simplifican el proceso de crear aplicaciones web y **APIs**, permitiéndote construir servidores rápidos y escalables sin complicaciones.

Aunque Node.js ya tiene un módulo integrado llamado `http` para crear servidores, escribir código directamente con él puede ser tedioso y repetitivo. Express.js **simplifica** estas tareas, ofreciendo una estructura más clara y muchas funcionalidades útiles para construir aplicaciones de manera más eficiente.

DEV.E:

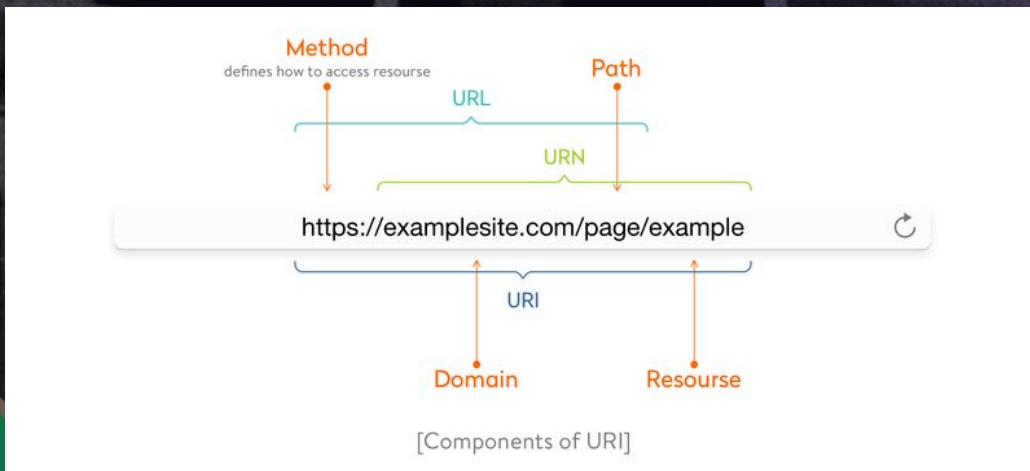


```
1  const express = require('express');
2
3  const app = express();
4
5  app.get('/', (req, res) => {
6    res.send('Mi primer servidor con express');
7  });
8
9  app.listen(3000, () => {
10    console.log('Server is running on port 3000');
11  });
```


Reto de la semana: Crear una nueva ruta



- Imprime un mensaje diferente creando una nueva ruta en tu servidor



No olviden



Las lecturas de **edu.devf.la**

