

**Materia:**

**DISEÑO ELECTRÓNICO BASADO EN  
SISTEMAS EMBEBIDOS**

**Alumno:**

Posadas Pérez Isaac Sayeg

Paniagua Rico Juan Julian

García Azzúa Jorge Roberto

**Grado y grupo:**

8°G

**Profesor:**

Garcia Ruiz Alejandro Humberto

**Unidad 4 -Tarea 12:**

Software Node JS

# Software Node.js

## Introducción

**Node.js** es un entorno de ejecución para JavaScript en el servidor, basado en el motor **V8** de Google Chrome. Desde su creación en 2009 por **Ryan Dahl**, Node.js ha revolucionado el desarrollo de aplicaciones web, al permitir que el mismo lenguaje utilizado en el navegador también se ejecute en el servidor. Se caracteriza por ser **asíncrono**, **orientado a eventos** y altamente eficiente en el manejo de múltiples conexiones concurrentes. Node.js es ampliamente utilizado en aplicaciones web, APIs RESTful, microservicios, sistemas embebidos, y plataformas en tiempo real como chats y videojuegos en línea.

## Desarrollo

### ¿Qué es Node.js?

Node.js no es un lenguaje ni un framework, sino un **entorno de ejecución** que permite usar JavaScript para desarrollar aplicaciones del lado del servidor. A diferencia de lenguajes tradicionales como PHP o Java, Node.js opera sobre un modelo **single-threaded** (un solo hilo) pero **no bloqueante**, lo que lo hace muy eficiente para aplicaciones de red y sistemas con muchas operaciones de entrada/salida.

### Características principales:

- **Asincronía y no bloqueo:** Las funciones no esperan a que otras terminen; esto mejora el rendimiento en tareas como acceso a bases de datos o peticiones de red.
- **Orientado a eventos:** Usa un sistema de eventos y callbacks, ideal para manejar muchas conexiones simultáneas.

- **Ecosistema NPM (Node Package Manager):** Miles de bibliotecas de código abierto disponibles para extender sus capacidades.
- **Multiplataforma:** Funciona en Windows, Linux y macOS.
- **Escalabilidad:** Ideal para arquitecturas de microservicios y sistemas distribuidos.

### Usos comunes de Node.js:

- Backend de aplicaciones web (Express.js, NestJS).
- APIs RESTful o GraphQL.
- Aplicaciones en tiempo real (chats, videojuegos, IoT).
- Automatización de tareas y scripts.
- Controladores para hardware o sistemas embebidos (por ejemplo, con Johnny-Five en Arduino o Raspberry Pi).

### Ejemplo:

Un ejemplo simple de servidor HTTP con Node.js:

```
const http = require('http');  
  
const servidor = http.createServer((req, res) => {
```

```
res.writeHead(200, { 'Content-Type': 'text/plain' });

res.end('¡Hola desde Node.js!');

});

servidor.listen(3000, () => {

  console.log('Servidor ejecutándose en
http://localhost:3000');

});
```

Este pequeño script levanta un servidor web que responde con un mensaje de texto cuando se accede a la ruta principal. Puede usarse como base para construir una API o un servicio web más complejo.

También es común usar frameworks como **Express.js** para facilitar el enrutamiento y el manejo de peticiones:

```
const express = require('express');

const app = express();
```

```
app.get('/saludo', (req, res) => {  
  
    res.json({ mensaje: 'Hola, API RESTful con Node.js' });  
  
});  
  
app.listen(3000, () => console.log('Servidor en puerto  
3000'));
```

## Conclusión

Node.js es una herramienta poderosa y flexible que ha transformado la forma en que se desarrollan aplicaciones web y servicios backend. Su modelo asíncronico, su enorme ecosistema de paquetes y su capacidad para manejar múltiples conexiones con eficiencia lo convierten en una excelente elección para desarrollos modernos. Además, su uso con JavaScript lo hace accesible para una gran comunidad de desarrolladores. En contextos como APIs, microservicios, sistemas en tiempo real y hasta proyectos de IoT, Node.js ofrece una solución ligera y escalable.

## Bibliografía

1. Tilkov, S., & Vinoski, S. (2010). *Node.js: Using JavaScript to Build High-Performance Network Programs*. IEEE Internet Computing.
2. Cantelon, M., Harter, M., Holowaychuk, T., & Rajlich, N. (2014). *Node.js in Action*. Manning Publications.

3. Node.js Foundation. (2024). *Official Documentation*. <https://nodejs.org>
4. Rauschmayer, A. (2020). *Speaking JavaScript*. O'Reilly Media.
5. NPM Inc. (2024). *NPM Documentation*. <https://docs.npmjs.com>