

Bootcamp de Full Stack

Bienvenidos a la clase N°51

- Fundamentos
- Nodemon
- Variables de Entorno
- Módulos de Node
- REPL & Objetos Globales
- Servidor HTTP
- Módulo http

Backend

Fundamentos

¿Qué es el Backend?

El backend son los datos y la infraestructura que permiten que la aplicación funcione. Almacena y procesa los datos de las aplicaciones para los usuarios.

¿Qué es Node JS?

Es un entorno de ejecución de JavaScript que funciona fuera del navegador. Es código abierto, multiplataforma y permite crear aplicaciones del lado del servidor, como APIs y servicios web. Proporciona acceso a funcionalidades del sistema operativo (como archivos y red) que no están disponibles en los navegadores.

Características:

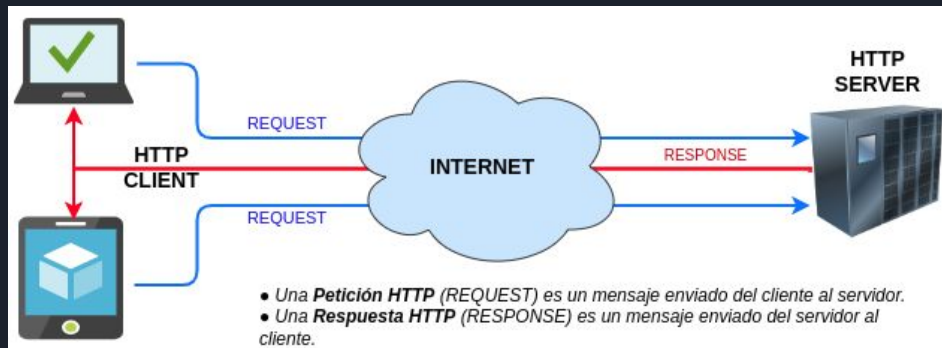
- Asíncrono y orientado a eventos: usa un modelo de entrada/salida (I/O) no bloqueante.
- Motor V8: convierte el código JavaScript en código máquina de forma muy rápida.
- Gestión de paquetes (NPM): incluye npm (Node Package Manager), el repositorio de librerías más grande del mundo, que facilita la instalación de dependencias y herramientas.
- Multiplataforma: puede ejecutarse en Windows, Linux y macOS sin cambios importantes en el código.
- Aplicaciones típicas: se usa para crear APIs REST, servidores web, aplicaciones en tiempo real, microservicios y herramientas de línea de comandos.

Backend

Fundamentos

¿Qué es HTTP?

Hypertext Transfer Protocol (HTTP) es un protocolo de la capa de aplicación para la transmisión de documentos (HTML, XML, JSON, etc). Fue diseñado para la comunicación entre los navegadores y servidores web, aunque se puede utilizar para otros propósitos también. Sigue el clásico modelo cliente-servidor, en el que un cliente establece una conexión con el servidor, realiza una petición y espera hasta que recibe una respuesta del mismo. Se trata de un protocolo sin estado, lo cual significa que el servidor no guarda ningún dato (estado) entre dos peticiones.



Backend

Fundamentos

¿Qué son los códigos de estado de respuesta HTTP?

Los códigos de estado de respuesta HTTP indican si se ha completado satisfactoriamente una solicitud.

Las respuestas se agrupan en cinco clases:

- Respuestas informativas (100–199),
- Respuestas satisfactorias (200–299),
- Redirecciones (300–399),
- Errores en los clientes (400–499),
- Errores en los servidores (500–599).

Principales códigos de estado de respuesta HTTP:

- | | |
|-----------------------------|------------------------------------|
| • 200 Ok | • 400 Bad request |
| • 201 Created | • 401 Unauthorized (autenticación) |
| • 500 Internal Server Error | • 403 Forbidden (permisos) |
| | • 404 Not found |
| | • 409 Conflict |
| | • 422 Unprocessable content |

Backend

Nodemon

¿Qué es Nodemon?

Es una herramienta que se utiliza en el desarrollo de aplicaciones con [Node.js](#). Su función principal es reiniciar automáticamente la aplicación cada vez que detecta cambios en los archivos del proyecto. Esto evita que el programador tenga que detener y volver a ejecutar el servidor manualmente en cada modificación, lo que agiliza mucho el desarrollo.

Características:

- Monitorea los archivos del proyecto en tiempo real.
- Reinicia la aplicación automáticamente cuando detecta un cambio.
- Se usa solo en entornos de desarrollo (no en producción).
- Es fácil de instalar y ejecutar con npm.

Instalación: `npm install --save-dev nodemon`

Añadir al script del package.json (se ejecuta con `npm run dev`): `"dev": "nodemon ./src/app.js"`

Backend

Variables de Entorno

¿Qué son las Variables de Entorno?

En Node.js, las variables de entorno son valores globales que pueden ser accedidos por cualquier parte de una aplicación. Estas variables son utilizadas para configurar y personalizar el comportamiento de la aplicación según el entorno en el que se está ejecutando. Algunos ejemplos comunes de variables de entorno incluyen información de conexión a bases de datos, claves de API, configuraciones de seguridad, entre otros.

Instalación: `npm install dotenv`

Importar dentro de `app.js`: `import "dotenv/config"`

Backend

Módulos de Node

¿Qué son los Módulos de Node JS?

Son bloques de código reutilizables y encapsulados que ayudan a organizar y estructurar las aplicaciones. Cada archivo se considera un módulo independiente, y se puede exportar parte de su contenido para que otros módulos lo utilicen. Esta modularidad permite dividir una aplicación en piezas más pequeñas, lo que facilita el mantenimiento, la escalabilidad y la legibilidad del código.

Tipos de Módulos:

- Módulos propios → Archivos creados por el programador.
- Módulos internos (core modules) → Incluidos en Node.js como path, fs, http, util.
- Módulos externos → Instalados desde npm, como eslint, moment, express, etc.

Backend

REPL & Objetos Globales

¿Qué es el REPL?

REPL por las siglas en inglés de (Read-Eval-Print-Loop), también conocido como alto nivel interactivo o consola de lenguaje, es un entorno de programación interactiva basado en un proceso cíclico donde se lee un fragmento de código ingresado por el usuario, se evalúa y se devuelve un resultado al usuario; las partes de programa escrito en un entorno pueden compartir un estado global mientras el programa esté activo.

Consigna:

- Emplear la CLI de REPL
- Ejecutar un archivo de JavaScript desde la consola
- Objetos Globales (<https://nodejs-es.github.io/api/globals.html>):
 - El comando `__filename` representa el nombre de archivo que se está ejecutando en el código.
 - El comando `__dirname` representa el nombre del directorio en el que reside el script que se está ejecutando.

BREAK

Descansemos 10 minutos



Backend

Servidor HTTP

Consigna:

- Crear archivo principal (dentro de src) `app.js`
- Inicializar el proyecto de Node JS: `npm init -y`
- Configurar el "type": "module"
- Implementar :
 - Nodemon: `npm install --save-dev nodemon`
 - Eslint: `npm install --save-dev eslint@latest @eslint/js@latest`
 - Dotenv: `npm install dotenv`
- Ignorar la carpeta `node_modules` en **Git**
- Agregar scripts en el package.json (start, dev y lint)
- Ejecutar en modo **watch**: `npm run dev`



Backend

Módulo http y path

Consigna:

- Emplear el módulo http
- Demostrar el uso de algunos códigos de estado HTTP y formatos de transferencia de datos:
 - Texto
 - HTML
 - JSON
- Emplear variables de entorno
- Controlar error de ruta inexistente



CIERRE DE CLASE

Continuaremos en la próxima clase

