

Bootcamp de Full Stack

Bienvenidos a la clase N°54

- Fundamentos
- REPL & Objetos Globales
- Módulos de Node
 - Módulos path y util
- Servidor HTTP
 - Crear proyecto de Node
 - Emplear módulo http
- Variables de Entorno
- Proyecto Integrador II - Consultas

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.

Fuente: <https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend>

¿Qué es Node JS?

Es un entorno que trabaja en tiempo de ejecución, de código abierto, multiplataforma, que permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript. La ejecución en tiempo real está pensada para usarse fuera del contexto de un explorador web (es decir, ejecutarse directamente en una computadora o sistema operativo de servidor). Como tal, el entorno omite las APIs de JavaScript específicas del explorador web y añade soporte para APIs de sistema operativo más tradicionales que incluyen HTTP y bibliotecas de sistemas de ficheros.

Fuente: https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction

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.

Backend

Módulos de Node

¿Qué son los Módulos de Node JS?

En Node.js, los módulos son unidades de código reutilizable y encapsulada que se utilizan para organizar y estructurar el código en aplicaciones. Cada archivo en Node.js se trata como un módulo, y se puede exportar (hacer público) cierto código para que otros módulos lo utilicen. La modularidad es una característica clave de Node.js y permite dividir una aplicación en partes más pequeñas y manejables, lo que facilita el mantenimiento y la escalabilidad.

El módulo **path** proporciona utilidades para trabajar con rutas de archivos y directorios. Para lograr importar dicho módulo, se requiere de **CommonJS** que es una especificación que define un sistema para organizar y estructurar el código en entornos fuera del navegador. Es decir, en el lado del servidor.

Consigna:

- Uso de **CommonJS** (se exporta con **module.exports** y se importa con **require**)
- Emplear los módulos **path** y **util**

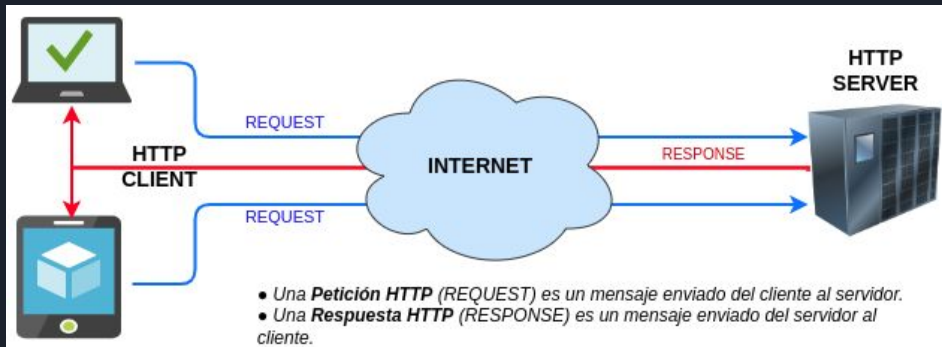
Backend

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.

Fuente: <https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend>



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 HTTP específica. 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).

Fuente: <https://developer.mozilla.org/es/docs/Web/HTTP/Status>

Principales códigos de estado de respuesta HTTP:

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

BREAK

Descansemos 15 minutos



Backend

Servidor

Consigna:

- Inicializar el proyecto de Node JS
- Emplear el módulo **http**
- Demostrar el uso de algunos códigos de estado HTTP y formatos de transferencia de datos:
 - Texto
 - HTML
 - JSON
- Agregar modo **watch** al **script** del archivo package.json

Backend

Fundamentos

¿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.

Consigna:

- Instalar el paquete **dotenv**
- Ignorar la carpeta **node_modules** en **git**
- Utilizar variables de entorno

Proyecto Integrador II

Espacios de Consultas



CIERRE DE CLASE

Continuaremos en la próxima clase

