Glosario CH31

# Indice

[Frontend](#tfjhq7lczxfl)

[Backend](#9eid2irnas88)

[API](#fospf747wgii)

[Interfaz](#7dfdvuzbocd7)

[Lenguajes de programación](#du3jzi8lol5m)

[Internet](#hwfvpyge417z)

[Cliente/Servidor](#rseg9qwwfvs)

[Solicitudes HTTP](#m6qsk8smom9t)

[CRUD](#np1qszae3p85)

[Respuestas del servidor](#pn0xsubbx9e1)

[TCP/IP](#dot0i5wrub38)  
  
CLI (Interfaz de Línea de Comando): Una forma de interactuar con una computadora o programa ingresando comandos de texto en lugar de usar una interfaz gráfica.

GUI (Interfaz Gráfica de Usuario): Un entorno de usuario que utiliza elementos visuales como ventanas, botones y menús para interactuar con programas y sistemas en lugar de texto.

Anidación: La práctica de colocar elementos o estructuras dentro de otros elementos o estructuras similares, como colocar un bucle dentro de otro bucle en la programación.

git --version: Un comando que muestra la versión de Git instalada en una computadora.

git help: Un comando para obtener información de ayuda sobre cómo usar Git y sus comandos.

Framework: Una estructura o conjunto de herramientas que facilita el desarrollo de aplicaciones al proporcionar una base sólida y componentes predefinidos.

Footer: La sección inferior de una página web que a menudo contiene información de contacto, enlaces útiles u otros datos.

Operador Ternario: Un operador de programación que permite tomar decisiones basadas en una condición, generalmente en una sola línea de código.

Switch: Una estructura de control en programación que permite seleccionar una acción a ejecutar entre muchas opciones, similar a una serie de "if" y "else if".

If: Una estructura de control que permite ejecutar un bloque de código si una condición es verdadera.

Else If: Una extensión de la estructura "if" que permite verificar múltiples condiciones.

Datos Primitivos: Tipos de datos básicos en programación, como números, cadenas, booleanos y valores nulos.

Operadores de Igualdad e Igualdad Estricta: Operadores utilizados para comparar valores en JavaScript, donde la igualdad estricta verifica también el tipo de dato.

Manipulación del DOM: La capacidad de cambiar la estructura y el contenido de una página web utilizando JavaScript.

Rejects: Puede ser un error tipográfico. Si te refieres a "Promesas Rechazadas," son promesas que no se cumplieron correctamente debido a un error.

Slice / Splice: Métodos en JavaScript para trabajar con arreglos. slice crea una copia de parte de un arreglo, mientras que splice modifica el arreglo original al agregar o eliminar elementos.

Arreglos .map: Un método de arreglo en JavaScript que permite aplicar una función a cada elemento del arreglo y crear un nuevo arreglo con los resultados.

forEach: Un método de arreglo que itera sobre cada elemento y ejecuta una función proporcionada.

Sort: Un método de arreglo que ordena los elementos de un arreglo alfabéticamente o numéricamente.

Stream Template: Puede ser un error tipográfico. Si te refieres a "Template Strings," son cadenas de texto en JavaScript que permiten incrustar variables dentro de ellas.

Wireframes (Baja, Media y Alta Fidelidad): Representaciones visuales de una interfaz de usuario que van desde bocetos simples hasta diseños detallados, utilizados en diseño de sitios web y aplicaciones.

Tipos de Nodos en el DOM: Elementos, atributos, texto y otros componentes que componen la estructura de un documento HTML en el Modelo de Objetos del Documento (DOM).

Objetos JSON: Objetos JavaScript en formato JSON (JavaScript Object Notation), comúnmente utilizado para el intercambio de datos.

POO (Programación Orientada a Objetos): Un paradigma de programación que utiliza objetos y clases para modelar y organizar el código.

Request Body: Los datos enviados al servidor HTTP como parte de una solicitud, comúnmente en formato JSON o XML.

localStorage: Un mecanismo de almacenamiento en el navegador web que permite almacenar datos de manera persistente en el dispositivo del usuario.

Constructor: Una función en programación que se utiliza para crear objetos de una clase en la programación orientada a objetos.

Herencia: Un concepto en programación orientada a objetos donde una clase (subclase) puede heredar propiedades y métodos de otra clase (superclase).

Funciones Regex: Funciones que se utilizan para trabajar con expresiones regulares, que son patrones utilizados para buscar y manipular texto.

Arreglos (Arrays): Estructuras de datos que almacenan una colección ordenada de elementos.

Promesas: Objetos en JavaScript que representan un valor que puede estar disponible en el futuro, ya sea correctamente resuelto o rechazado.

Fetch: Una función en JavaScript que se utiliza para realizar solicitudes HTTP a servidores web y obtener datos.

Parse: La acción de convertir una cadena de texto en un tipo de dato específico, como convertir una cadena JSON en un objeto JavaScript.

Full Stack - Front End - Back End: Términos que describen diferentes roles y responsabilidades en el desarrollo de aplicaciones web, donde "Full Stack" implica trabajar tanto en el lado del cliente como en el servidor.

DevOps: Una cultura y conjunto de prácticas que combina desarrollo de software (Dev) y operaciones de TI (Ops) para mejorar la entrega y el mantenimiento de aplicaciones.

Cloud - SQL - Base de Datos: Conceptos relacionados con la computación en la nube, bases de datos relacionales (SQL) y el almacenamiento y acceso de datos.

Redes: El estudio y la práctica de la comunicación de datos y recursos entre dispositivos y sistemas a través de redes.

Emmet Plugin: Una herramienta utilizada en editores de código para acelerar la escritura de código HTML y CSS.

Ciclos (for, while, do while): Estructuras de control en programación que permiten la repetición de una serie de acciones.

**Frontend** la parte visual

**Ux / UI:** User Experiencie / User Interface

**HTML:** Hypertext Markup Language

**CSS:** Cascade Style Sheet

**JS:** JavaScript

**Bootstrap:** framework de diseño y desarrollo web

**Validaciones**

**Consumo de APIs**

**Lógica en el cliente**

**Interface web**

**Backend** lo que no vemos, el cerebro, lógica

**Java:** Lenguaje de programación en la parte del servidor

**SQL:** Structured Query Language, lenguaje para gestionar y manipular bases de datos

**Spring Boot:** framework de desarrollo que facilita la creación de aplicaciones Java. Reduce configuraciones complejas y permite centrarse en la lógica de negocio.

**Bases de datos**

**Rendimiento**

**Lógica en el servidor**

**Seguridad**

**Microservicios**

**API** Interfaz de programación de aplicaciones



**Interfaz** Intermediario, puente, conexión



**Sintaxis**: Forma de “escribir”

**Lenguajes de alto nivel:** se asemeja al lenguaje humano y utiliza muchas palabras (Java, Python)

**Lenguajes de bajo nivel:** lenguaje binario (0 y 1)

**Tipado:** cómo se declaran las variables

**Declarar las variables:** especificar nombre y tipo de variable

**Variable:** donde se guardan los datos de un programa

**Internet:** redes interconectadas

**WWW:** World Wide Web / Red Mundial

Red:

**Cliente:** quien solicita y recibe servicios o recursos

**Servidor:** proporciona servicios, recursos o información

**Protocolo:** Forma de hacer las cosas

**Servicios:** Información, recursos, imágenes, videos, etc.

**Base de datos:** Información categorizada y organizada

**Solicitudes GET - POST - PUT - DELETE**

**Get:** obtener

**Post:** agregar

**Put:** modificar

**Delete:** borrar

**CRUD:** **C**reate, **R**ead, **U**pdate, **D**elete, representa operaciones básicas para gestionar información en bases de datos

Códigos de respuesta para conexiones del Servidor

**200:** Aquí está la hamburguesa, ok

**301:** La hamburguesa la venden en otro restaurante, Movido permanentemente, Redirección

**302:** Movido temporalmente

**401:** No tiene permitido comprar hamburguesas, Unauthorized

**404:** Hamburguesa no encontrada, Not found

**500:** La cocina se prendió fuego, Internal server error

**El servidor** (quien prepara nuestra hamburguesa) toma todos los datos de nuestra petición (carne, pan, jitomate, lechuga, etc), la manda **al internet** (el encargado del mostrador) quien es quien entrega **nuestra petición** tal cual la pedimos (el cliente).

**TCP/IP**: protocolo de control de transmisión / protocolo de internet

**HTTP:** Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol)

**Algoritmo:** conjunto de instrucciones o reglas que guían la realización de una tarea o resolución de un problema. Es como una receta que te dice exactamente qué hacer en cada paso para alcanzar un resultado deseado

**lista de comandos terminal**

**pwd:** print word directory te da una mejor explicaciónde donde estas trabajando (la ruta de dirección)

**ls:** te enlista los elementos que se encuentran en la ruta

-**parametros:** usas el - después del comando

- l : formato long

-a: all muestra todos

-R: hacer el comando recursivo

**clear:**  limpiar la ventana de comando

**history:** muestra un historial de los comando usados

**cd:** change directory/ cambiar de directorio **sintaxis: cd directorio**

parametro: .. (espacio punto punto) regresar un ruta del directorio

**mkdir:** make directory

**rmdir:** remove directory

el signo $ indica que esta listo para llevar a cabo un comando si no aparece esta llevando a cabo un proceso

**cp:** copiar sintaxis: cp Cama.txt ../CuartoMemo/

**rm:** remove

**GIT:** sistema de control de versionado

Git hub permite en un solo archivo tener versiones de tal manera que si quieres puedes volver a algún punto en específico cada versión viene con un comentario y continuar el flujo hasta el punto deseado y volver a a cualquier versión posible que tenga un comentario

**Repositorio:** Es un proyecto que se encuentra en la nube. esto permite que varias personas trabajen en un mismo código, algo similar como documentos de drive



la estructura de un comando es:

que quiero hacer / como lo quiero hacer / a quien lo voy a hacer

