

INGENIERÍA MECATRÓNICA



DI\_CERO

DIEGO CERVANTES RODRÍGUEZ

DESARROLLO FULL STACK

HTML, CSS & JAVASCRIPT

Frontend y Backend

# Contenido

Frontend y Backend .....	2
Bases de Datos .....	2
¿Qué es Backend? .....	3
Servidores .....	4
Frontend.....	5



# Frontend y Backend

En la programación normalmente existen dos conceptos principales, uno es llamado Backend y el otro es llamado Frontend. El Frontend es lo que corre en el navegador y el Backend es lo que corre en el servidor.



Siempre existe un servidor asignado para cada sitio web, que carga todos los datos necesarios para correr la página y sus diferentes características como: los datos de cada usuario, el código en sí del HTML que maqueta la página, etc. Todo esto está guardado en una base de datos.

Existen muchos tipos de servidores y éstos suelen ser identificados por su puerto, el puerto es una serie de números para poder abrir un cierto huequito de las computadoras y que así nos podamos conectar a algún servidor.

Técnicamente existen 64,000 puertos, el puerto de las páginas web usualmente es el puerto 80.

El servidor en cuestión hardware, es una sola máquina o un conjunto de máquinas que en conjunto tienen guardados todos nuestros datos y se encargan de mandar o recibir nuestros datos; nosotros nos conectamos a estas máquinas para poder acceder a nuestros datos.

## Bases de Datos

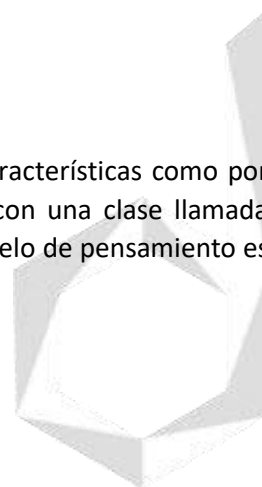
Las diferentes bases de datos que existen son las siguientes: Microsoft Access no es una base de datos, es un simulador de bases de datos, SQL (Sequel) Server si es una base de datos, Oracle también es una base de datos, pero la base de datos más famosa del mundo se llama MySQL que es una base de datos open source que corre del lado del servidor.

Cada gran empresa como Facebook o como Google debe tener una grandísima base de datos en donde se guardan todo tipo de datos como nuestro nombre, nuestra fecha de nacimiento, nuestros amigos, etc. Todo esto se accede a través de los servidores de estas empresas.

Existen dos tipos de bases de datos:

- Una es llamada base de datos relacional.
- La otra es llamada base de datos no relacional.

El modelo relacional es una herencia donde varios objetos mantienen ciertas características como por ejemplo tú con tu padre y tu padre con tu abuelo y todo ello tiene relación con una clase llamada persona ya que tanto tu como tu padre y como tu abuelo son personas, este modelo de pensamiento es el que siguen las bases de datos relacionales.



Las bases de datos no relacionales son las que no generan un vínculo entre sí, sino que generan piezas de documentos que son como listas de elementos sueltas, como por ejemplo en un chat que se tienen diferentes datos para cada usuario como el mensaje que se está enviando en forma de texto, el nombre del usuario que lo está mandando, la foto del usuario, etc. Pero la novedad que presentan las bases de datos no relacionales es que, si por ejemplo yo quiero que no solo se tenga el nombre de usuario, sino que ciertos usuarios en específico muestren su fecha de nacimiento y su lugar de trabajo, se puede hacer ya que en las bases de datos no relacionales se pueden agregar o quitar los datos que yo quiera y en las bases de datos relacionales esto no se podía hacer.

Las bases de datos NO relacionales debido a su modelo de elementos sueltos, para buscar algún dato el programa no debe entrar primero a un objeto y seguir la relación o herencia de los datos hasta encontrar el que quiero buscar como en la base de datos relacional, por lo que en la NO relacional es más rápido encontrar cualquier dato, aunque el problema es que en estas bases de datos es más difícil tener un buen orden ya que es muy flexible en los datos que me deja meter y quitar. Por este aspecto no hay una base de datos mejor que la otra más bien dependiendo de para qué la vayamos a usar es donde decidimos un modelo en vez del otro.

## ¿Qué es Backend?

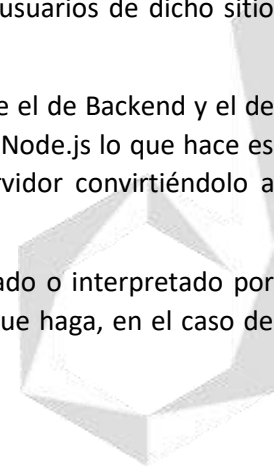
A los lenguajes de programación que interactúan con el usuario y sirven para hacer que la página se vea bonita como HTML (que sirve para maquetar nuestro sitio web), CSS (que sirve para darle estética a nuestra página) Y JavaScript (que se usa para darle dinamismo cuando se haga clic en algún botón y ese tipo de cosas) se les clasifica como Frontend. Pero resulta que existen lenguajes de programación que sirven solamente para conectarse a las bases de datos, a eso es a lo que se le llama Backend.

Hay varios lenguajes de programación que manejan el Backend, existe .NET, existe Java (aunque los anteriores no sean los más recomendables para programar en Backend), existe PHP, existe Python con Django y existe JavaScript con Node.js.

Es muy importante saber que el lenguaje que interactúa con el usuario jamás puede ser el mismo que interactúe con la base de datos porque sería un problema de seguridad, ya que con dar clic derecho al navegador web me sale un menú donde puedo darle clic a donde dice inspeccionar y ver el código fuente de la página en donde estoy, ya sabiendo esto ¿Qué pasaría si cualquier persona pudiera ver el código fuente y visualizar cómo esa página web se está conectando a su base de datos? Esto sería un problema ya que, si un lenguaje tiene acceso total a la base de datos, ese lenguaje tiene el poder para borrar, modificar o acceder a los datos de la página poniendo en peligro a los usuarios de dicho sitio web.

Pero aquí me puedo estar preguntando ¿no que no podían ser el mismo lenguaje el de Backend y el de Frontend? Entonces, ¿Por qué se repite JavaScript en los 2? Esto sucede porque Node.js lo que hace es agarrar mis líneas de código escritas en JavaScript y lo corre del lado del servidor convirtiéndolo a lenguaje C.

Cuando yo corro cualquier lenguaje de programación, éste necesita ser compilado o interpretado por algún otro programa y que así la computadora entienda qué le estoy diciendo que haga, en el caso de



JavaScript su compilador es el navegador web, pero para que JavaScript pueda correr en el servidor es donde se usa Node.js para que esas líneas escritas en código JavaScript puedan ser interpretadas y compiladas para correr del lado del servidor, en este proceso el navegador web no fue el compilador sino que fue Node.js el que hizo la interpretación de mi código.

Esto hace que el JavaScript que corre en el navegador no tenga nada de relación con el JavaScript que corre del lado del servidor.

En el Backend se puede usar cualquier lenguaje de programación y como no hay ninguna conexión entre el Frontend y el Backend esto no afectará en nada en cómo funciona mi programa, pero en el caso de Node.js lo que tiene de cómodo es que se puede programar todo en el mismo lenguaje y aun así no habrá relación entre el Frontend y el Backend.

## Servidores

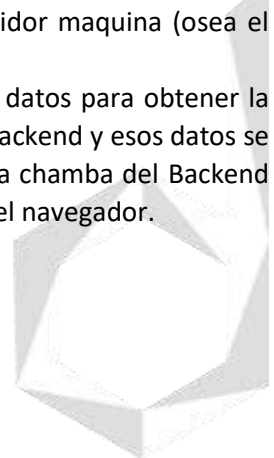
Los servidores son piezas de software que entienden la petición del navegador y me envían lo que requería, osea que el servidor es una aplicación especializada para entender la forma en que los navegadores piden cosas, de los más populares son Apache, NGINX, Node.js, etc.

Existen dos tipos de servidores; está el servidor máquina osea el hardware, la maquina física donde se guardan los datos y está conectado a un internet de alta velocidad en algún data center, que es un edificio lleno de servidores y está el servidor software que corre dentro de las computadoras a nivel de sistema operativo a través de los puertos, a través de cada puerto corre un servidor diferente, hay puertos diferentes para diferentes cosas como e-mails, páginas web, video llamadas, chats, etc.

Siempre que me conecte a páginas web estaré usando el puerto 80 a través de HTTP, entonces cuando yo me conecto al servidor hardware, éste verifica qué servidor de software está corriendo en el puerto 80 el cuál puede ser cualquiera de los mencionados previamente osea Apache, NGINX, Node.js, etc.

Entonces lo que sucede es lo siguiente:

1. Abrimos nuestro navegador web, este estará usando un puerto en específico de nuestra computadora.
  - a. El navegador estará usando el puerto 80 que es donde está nuestro servidor de software, el cuál sirve para conectarse a internet a través del protocolo http.
2. El servidor recibe la petición, si es una petición que tiene que ser procesada con bases de datos, se la manda al Backend. Si no es necesario usar nada del Backend como por ejemplo cuando pido una foto, que lo único que hace es meterse al disco duro del servidor maquina (osea el hardware), buscar la imagen y mandársela al navegador.
3. Cuando algo le mandemos al Backend, este se conectará con la base de datos para obtener la información que le haya solicitado, la base de datos le envía los datos al Backend y esos datos se los regresa al servidor para empujárselos al navegador y hasta ahí llego la chamba del Backend ya después de esto es donde entra el Frontend para mostrar los datos en el navegador.



# Frontend

El Frontend está compuesto por todas las cosas que corren del lado del navegador que es HTML, CSS y JavaScript; aunque también en el Frontend puedo tener imágenes, videos, etc.

En conclusión, la frontera divisora entre el Frontend y el Backend es que, el Backend corre del lado del servidor y el Frontend corre del lado del navegador, normalmente a esta diferencia se le llama cliente-servidor; donde cliente es el humano que está haciendo algo, osea el resultado final de mi página y servidor se le conoce a lo que corre detrás del humano osea lo que el humano no puede ver.

Este es el grafico que ejemplifica todos los pasos que acabamos de dar a conocer:



Ahora arranquemos:

¿Qué pasa cuando alguien entra a una URL equis a través de su navegador como por ejemplo [www.mejorando.la](http://www.mejorando.la)?

-Lo primero que pasa cuando alguien entra a alguna página web es que el nombre del sitio web es intercambiado por una IP, las IPs son 4 grupos de números separados por puntos si estamos hablando de la versión 4 de las IP, si estamos hablando de la versión 6 ya son más grupos de números y letras; a través de esta IP un servidor máquina puede identificar la ubicación exacta en el planeta real en donde se encuentra nuestro servidor, existen muchos servidores DNS que nos proporcionan el hosting para nuestra página. Evitar contratar con go daddy

