Material complementario

CODERHOUSE

Clase 0: Principios de programación Backend

Programación web

¿Qué es el desarrollo web?

Cuando hablamos de desarrollo web, nos referimos a la construcción y mantenimiento de estos sitios que pueden encontrarse en la World Wide Web (www) y sus redes derivadas.
Originalmente éste sólo se basaba en el desarrollo de páginas estáticas. Sin embargo, con el paso del tiempo estas fueron requiriendo más dinamismo, al punto de comenzar a desarrollar no sólo páginas estáticas, sino sitios web completos o incluso aplicaciones web.

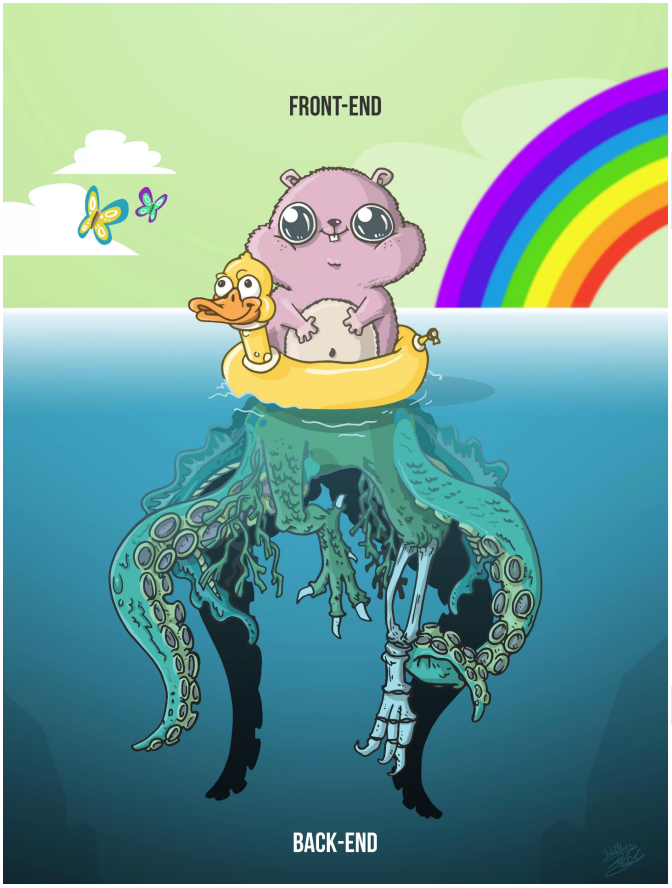
Las dos caras de una misma moneda

Conforme las necesidades del desarrollo web van creciendo, surge la necesidad de comenzar a separar el desarrollo en dos aspectos importantes: FrontEnd y BackEnd.
El **FrontEnd** comprende toda la **parte visual y de interacción directa con el usuario**. Su fin es dar la mejor impresión al usuario sobre la página o aplicación.

- ✓ Imágenes
- ✓ Colores
- ✓ Botones
- ✓ Interacciones

El **BackEnd** comprende toda la parte **lógica y de manejo de información**. Es decir, la base lógica de nuestra página o aplicación. El usuario NO sabe lo que hay detrás.

- ✓ El almacenamiento de información
- ✓ Los cálculos complejos
- ✓ Los servidores donde viven las páginas
- ✓ Manejo de información en general



Mientras que el **FrontEnd** se centra en la experiencia del usuario, ¡que esta sea lo más presentable y amigable posible!

...el **BackEnd** tiene una estructura interna para que todo funcione correctamente. ¡Que no lo vea el usuario!

¿FrontEnd vs BackEnd?

A menudo escuchamos a otras personas decir “odio el BackEnd” o “qué aburrido el frontEnd”. Esto ha polarizado a la comunidad del desarrollo, pensando que el mundo sería mejor si alguno de los dos no existiera. ¡FALSO! Ambos necesitan el uno del otro para poder estructurar páginas y aplicaciones. De esta manera, podremos resolver los problemas del mundo real con soluciones tecnológicas.
¡Hay que hacer las paces!

Stack MERN

Son ciertas tecnologías que, en conjunto, nos brindarán la posibilidad de desarrollar sistemas completos, debido a su máxima compatibilidad.
Podríamos decir que es la forma en la que el FrontEnd y el backEnd hacen las pases, ya que trabajan en conjunto. En este curso trabajaremos sobre el Stack MERN.

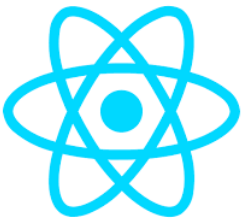
Componentes de MERN Stack

MongoDB

base de datos no relacional

ExpressJS

framework para crear servidores en NodeJS



ReactJS

librería para desarrollar interfaces de usuario

NodeJS

entorno de ejecución de Javascript

Distintas maneras de probar Javascript

¿Cómo probamos Javascript?

Cliente web

Lo utilizamos desde la consola del navegador, no necesitamos instalar nada ya que todo el motor vive en el navegador.

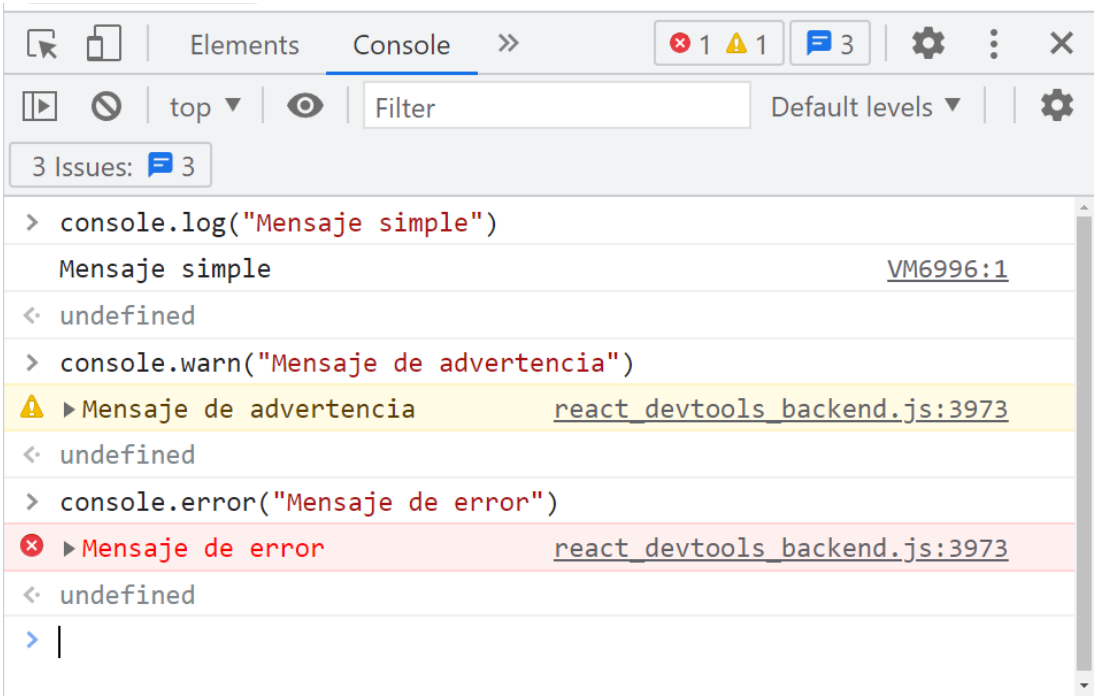
Node Js

Debemos instalarlo ya que se usa fuera del navegador, por lo que podremos escribir Javascript desde nuestro computador directamente. Éste es el que utilizaremos a lo largo del curso.

Cliente web

Se utiliza desde cualquier navegador. Algunos de los comandos que más usarás y verás a lo largo de un debug en cliente web son:

- ✓ **console.log("texto")** Mostrará un texto simple
- ✓ **console.warn("texto")** Mostrará una advertencia
- ✓ **console.error("texto")** Mostrará un error
- ✓ **console.clear()** Limpiará la consola para evitar que se acumule el código.

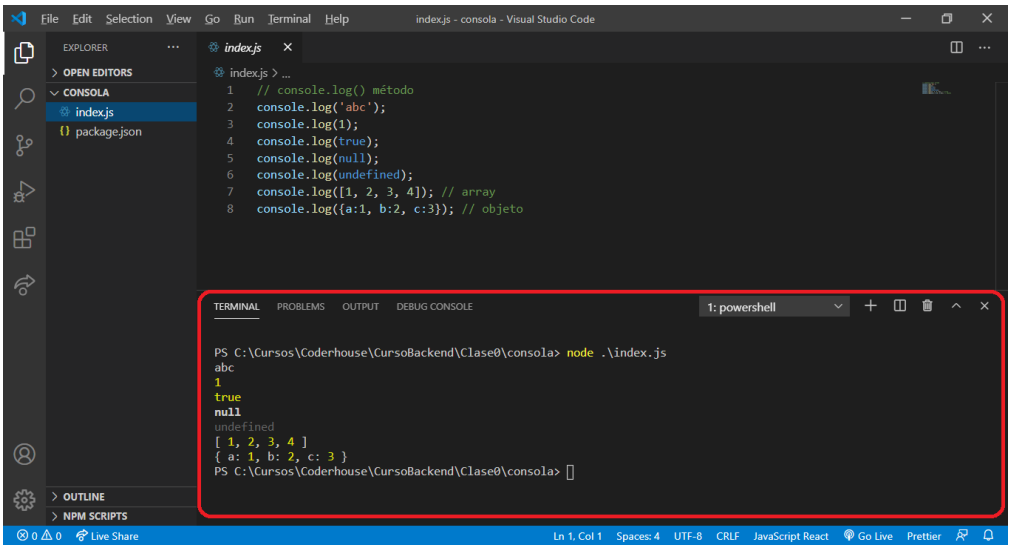


Consola de cliente web

Node Js

Cuando trabajamos en backend, no tenemos un navegador, es por ello que se precisa de alguna tecnología que nos permita correr código de Javascript, sin necesidad de abrirlo en el navegador. Ahí es cuando utilizamos Node js, para ello, toca considerar:

- ✓ Node js levantará un entorno completo para probar nuestras funciones.
- ✓ Debe correrse desde un CLI para poder visualizar los avances del código.
- ✓ Usualmente usaremos el CLI de Visual Studio Code.



Código de Javascript corriendo desde Node Js

Tipos de datos en Javascript

Tipo de dato

Es el atributo que especifica la clase de dato que almacena la variable.

Especifica con qué estaremos trabajando, para que la computadora reconozca qué operaciones puede hacer con él.

Tipos de datos

- ✓ **Tipo Primitivos:** Incluyen a las cadenas de texto (String), variables booleanas cuyo valor puede ser true o false (Boolean) y números (Number). Además hay dos tipos primitivos especiales que son Null y Undefined. La copia es por valor.
- ✓ **Tipo Objeto:** Incluyen a los objetos (Object), a los arrays (Array) y funciones. La copia es por referencia.

		Nombre	Descripción
Tipos de datos en Javascript	**Tipos primitivos**	String	Cadenas de texto
		Number	Valores numéricos
		Booleans	True/false
		Null	Tipo especial, contiene null
		Undefined	Tipo especial, contiene undefined.
	Tipos objeto	Tipos predefinidos de Javascript	Date (fecha) RegExp (expresiones regulares) Error (datos de error)
		Tipos definidos por el programador/usuario	Funciones simples Clases
		Arrays	Serie de elementos o formación tipo vector o matriz. Lo consideraremos un objeto especial que carece de métodos.
		Objetos especiales	Objeto global
			Objeto prototipo
			Otros

Variables en Javascript

Ahora que hemos comprendido cuáles son los tipos de datos, es importante comprender dónde serán guardados dichos datos. Una variable es un espacio de memoria apartado por la computadora, para poder guardar un dato.
Tal cual lo dice su nombre, una variable puede cambiar su valor si así el programa lo necesita. Ello permite la reutilización de una sola variable, para los casos que se vayan requiriendo.