



**¡Les damos la
bienvenida!**

¿Comenzamos?

Esta clase va a ser

- grabada



COMISIÓN N°70365

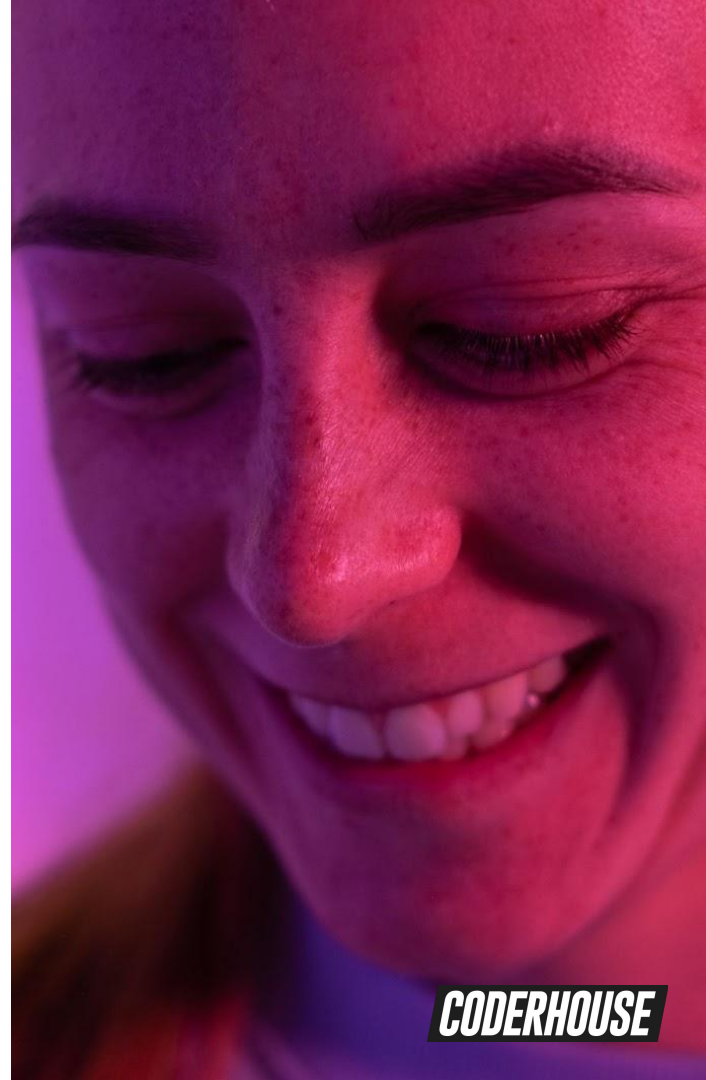
Presentación del equipo

- ✓ **Profesor/a responsable:** Sergio Regalado Alessi
- ✓ **Tutor Adjunto:** Saúl Belbey
- ✓ **Tutores y tutoras:**
 - Mauro Calodolce
 - Martin Alberti

Presentación de estudiantes

Por encuestas de Zoom

1. País
2. Conocimientos previos
3. ¿Por qué elegiste este curso?



CODERHOUSE



Lo que debes saber
antes de empezar

Acuerdos y compromisos

ACUERDOS Y COMPROMISOS

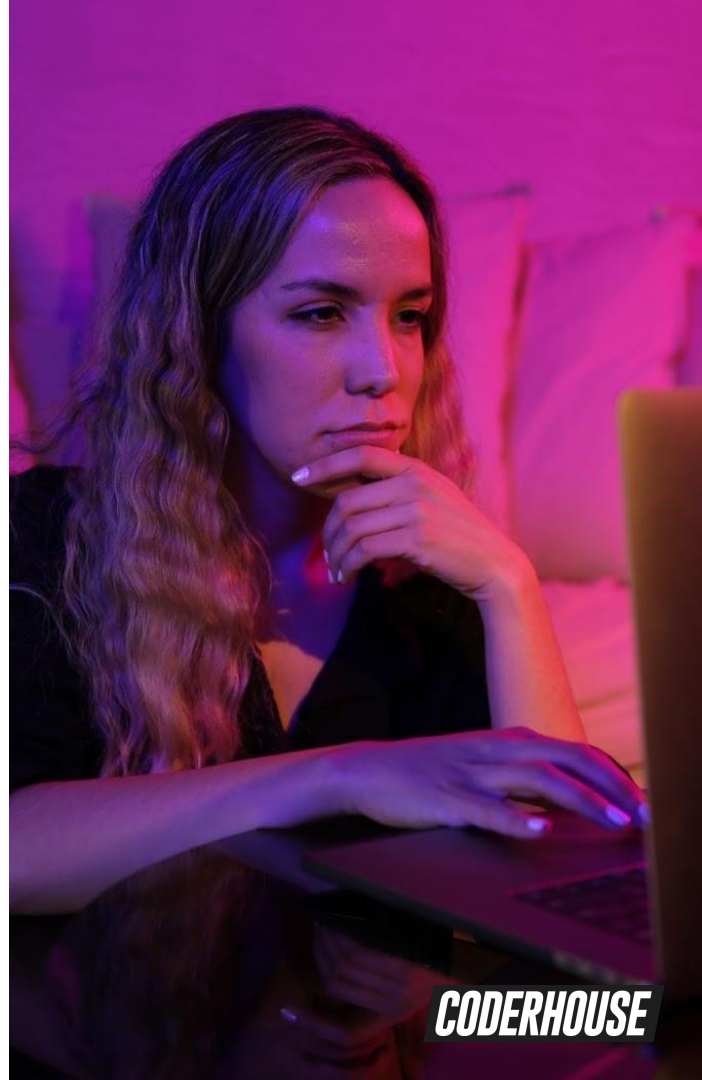
Convivencia

- ✓ Conoce aquí nuestro [código de conducta](#) y ayúdanos a generar un ambiente de clases súper ameno.
- ✓ Durante las clases, emplea los medios de comunicación oficiales para canalizar tus dudas, consultas y/o comentarios: **chat Zoom público y privado, y plataforma de Coderhouse.**
- ✓ Ten en cuenta **las normas** del buen hablante y del buen oyente, que nunca están de más.
- ✓ Verifica el estado de **la cámara y/o el micrófono** (on/off) de manera que esto no afecte la dinámica de la clase.

ACUERDOS Y COMPROMISOS

Distractores

- ✓ Encuentra tu espacio y crea el momento oportuno para **disfrutar de aprender**
- ✓ Evita dispositivos y aplicaciones que puedan **robar tu atención**
- ✓ Mantén la **mente abierta y flexible**, los prejuicios y paradigmas no están invitados



CODERHOUSE

ACUERDOS Y COMPROMISOS

Herramientas

- ✓ Mantén a tu alcance **agua, mate o café**
- ✓ Si lo necesitas, ten a mano lápiz y papel para que no se escapen las ideas. Pero recuerda que **en Google Drive tienes archivos que te ayudarán a repasar, incluidas las presentaciones.**
- ✓ Conéctate desde algún equipo (laptop, tablet) que te permita **realizar las actividades** sin complicaciones.
- ✓ Todas las clases quedarán grabadas y serán compartidas tanto en la **plataforma de Coderhouse como por Google Drive.**



ACUERDOS Y COMPROMISOS

Equipo

- ✓ **¡Participa de los After Class!** Son un gran espacio para atender dudas y mostrar avances
- ✓ Intercambia ideas por **el chat de la plataforma de Coderhouse**
- ✓ Siempre **interactúa respetuosamente**
- ✓ No te olvides de **valorar tu experiencia** educativa y de contarnos cómo te va

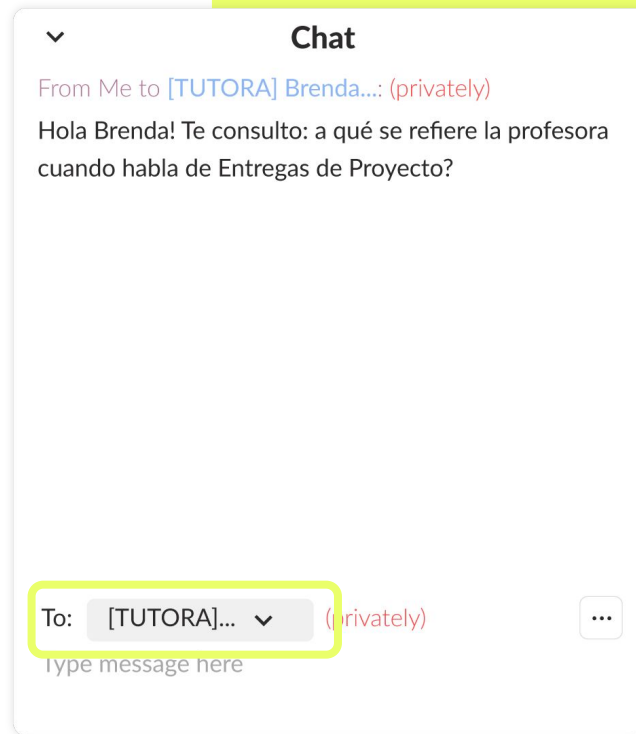
Interacciones en clase

INTERACCIONES EN CLASE

Mientras el profesor explica

Para mantener una comunicación clara y fluida a lo largo de la clase, te proponemos mantener 2 reglas:

1. Si tienes dudas durante la explicación, debes consultarle directamente por privado a tu tutor por el chat de Zoom.

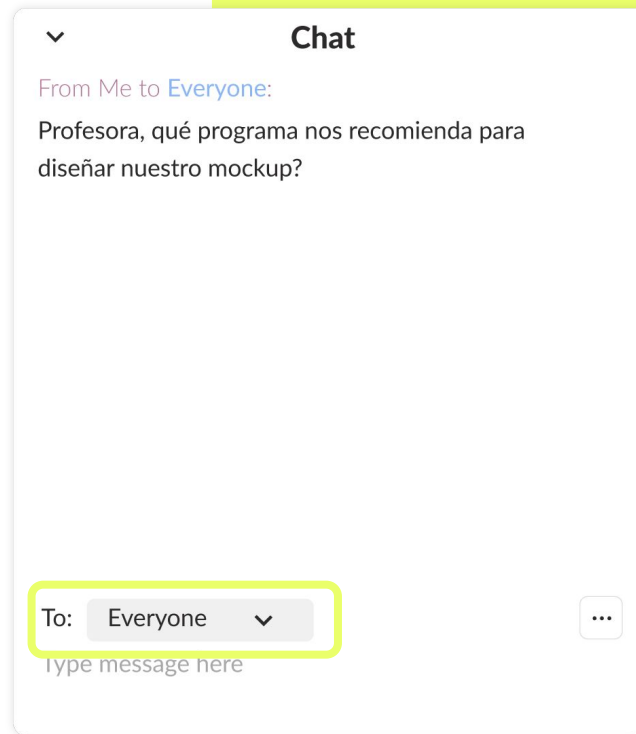


Espacios para consultas

2. Entre contenido y contenido, se abrirán breves espacios de consulta. Allí puedes escribir en el chat tu pregunta.

¡Tu duda puede ayudar a otras personas!

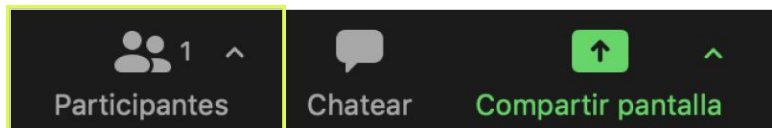
No olvides seleccionar “todos” para que todos puedan leerte (y no solo tu tutor).



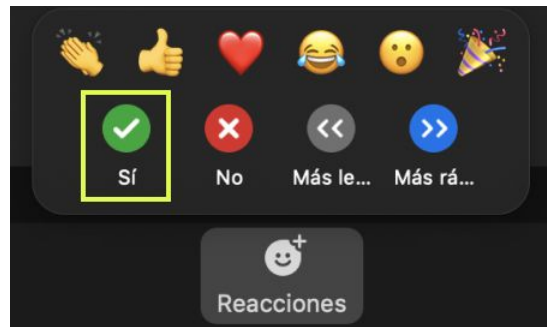
INTERACCIONES EN CLASE

Funcionalidades

Para **evitar saturar el chat de mensajes**, utiliza los signos que figuran en el apartado **Participantes**, dentro de Zoom.**



Por ejemplo: si se pregunta si se escucha correctamente, debes seleccionar la opción "Sí" o "No".



**Para quitar el signo, presiona el mismo botón nuevamente o la opción "clear all".

After Class

AFTER CLASS

¿Qué son?

Te acompañamos para resolver tus consultas sobre el contenido en estos espacios.

Si hay **temas que no se entendieron o necesitan refuerzo** se trabajarán en una clase de 1 hs que opera como **espacio de consulta**.

No son obligatorias ni se toma asistencia, pero son el espacio uno a uno con tu profesor/a** para responder dudas puntuales o reforzar conceptos.

Tu profesor/a está comprometido con tu educación, por lo tanto:

- ✓ Se responderán **dudas puntuales** que hayan quedado sobre los temas dados. ¡Vení preparado, queremos escucharte!
- ✓ Se verán temas de **conocimientos básicos** para la nivelación de saberes.

Proyecto Final



PROYECTO FINAL

Creando una API REST

Desarrollar el **backend de una aplicación de e-commerce** para poder vender productos de un rubro a elección.

El servidor se basará en un diseño de capas, orientado a MVC y su código contendrá las estructuras de programación más sólidas por parte del lenguaje JavaScript.





PROYECTO FINAL

Entrega	Requisito	Fecha
Primera pre entrega	Primera integración de tu PF	Clase 8
Segunda pre entrega	Segunda integración Websockets	Clase 12
Entrega del Proyecto Final	Mejorando la arquitectura de tu PF	Clase 17



GRILLA DE PRÁCTICAS Y PREENTREGAS

Clases	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Prácticas de pre-entregas																	
PF																	

⚠ Ten en cuenta los siguientes lapsos de tiempo al momento de cargar tus trabajos en la plataforma:

- ✓ Las preentregas cuentan con un lapso de **7 días continuos**.
- ✓ Una vez finalizada la cursada contarás con **10 días** continuos para entregar tu **Proyecto final**.

Clase 00. Desarrollo Avanzado de Backend

Principios de programación Backend

Temario

00

Principios de programación Backend

- ✓ [Programación web](#)
- ✓ [Distintas maneras de probar Javascript](#)
- ✓ [Tipos de datos en JavaScript](#)

01

Principios básicos de JavaScript

- ✓ Tipos de datos y variables en Javascript
- ✓ Javascript y ES6
- ✓ Funciones en Javascript

Objetivos de la clase



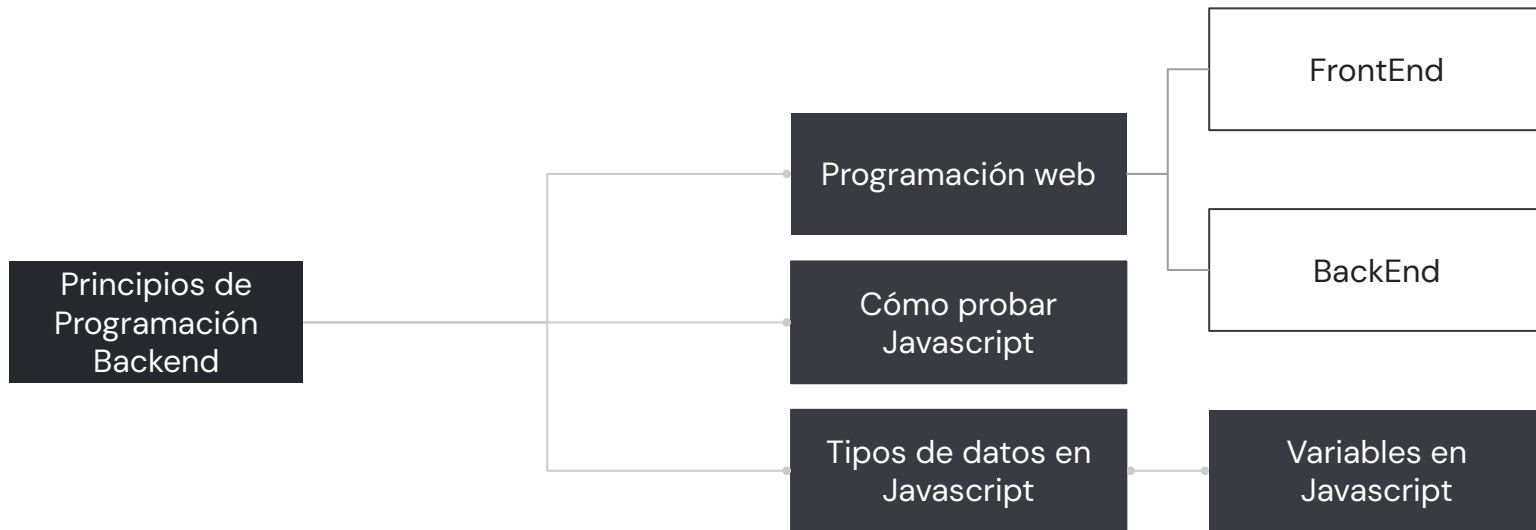
Conocer las diferencias entre programación Front end y Back end



Familiarizarse con las nociones básicas para programar utilizando Javascript y el MERN Stack



MAPA DE CONCEPTOS



FrontEnd y BackEnd

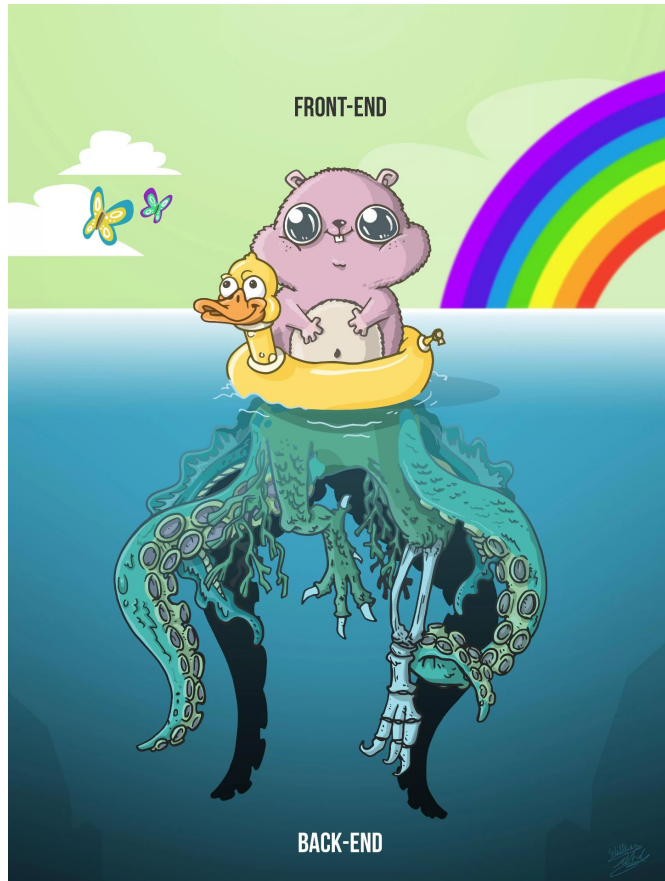
FrontEnd y Backend

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!

Stack MERN

¿Qué es un “stack”?

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



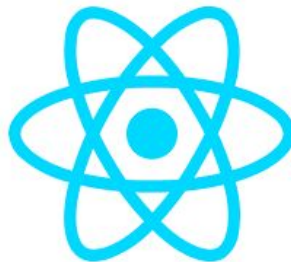
MongoDB

base de datos no
relacional



ExpressJS

framework para crear
servidores en NodeJS



ReactJS

librería para desarrollar
interfaces de usuario



el corazón de todo

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.



Ejemplo en vivo

Consola web: object (.log, .error, .warn y .clear)



Ejemplo en vivo

Consola CLI en Visual Studio Code



Para pensar

¿Cuándo utilizarías un cliente web, y cuándo utilizarías Node js al utilizar Javascript?

¿Una es reemplazo de la otra, o podríamos considerarlas complementarias?



Break

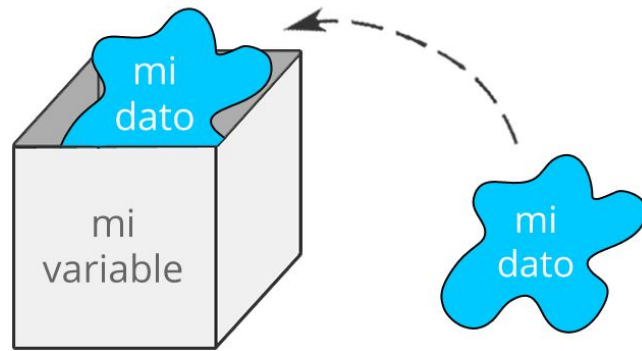
¡10 minutos y volvemos!

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.

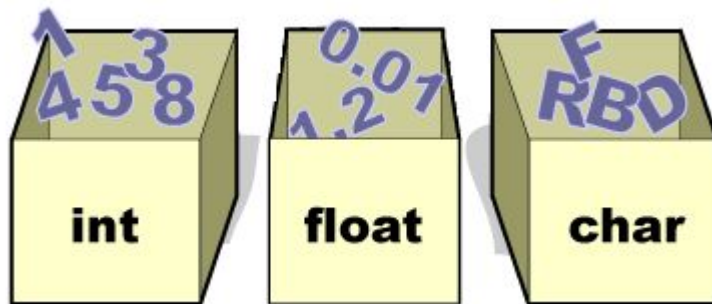
		Nombre	Descripción
Tipos de datos en Javascript	Tipos primitivos	String	Cadenas de texto
		Number	Valores numéricos
		Boolean	True/false
		Null	Tipo especial, contiene null
		Undefined	Tipo especial, contiene undefined.
	Tipos objeto	Object	Un objeto es una colección de propiedades clave-valor.
		Array	Serie de elementos o formación tipo vector o matriz.
		Function	En JavaScript, las funciones también son objetos. Es decir, se pueden almacenar en variables, pasarlas como argumentos y/o retornarlas.
		Objetos especiales	Objeto global (Infinity, NaN, etc.)
			Objeto prototipo
			Otros



ACTIVIDAD EN CLASE

Datos y variables

Definir variables que almacenen datos (nombre, edad, precio, nombres de series y películas), mostrar esos valores por consola, incrementar la edad en 1, una serie a la lista y volver a mostrarlas. Compartir la definición en el Visual Studio Code.



¿Preguntas?

Opina y valora
esta clase

Muchas gracias.