



**¡Les damos la  
bienvenida!**

¿Comenzamos?

Esta clase va a ser

- grabada



COMISIÓN N°41815

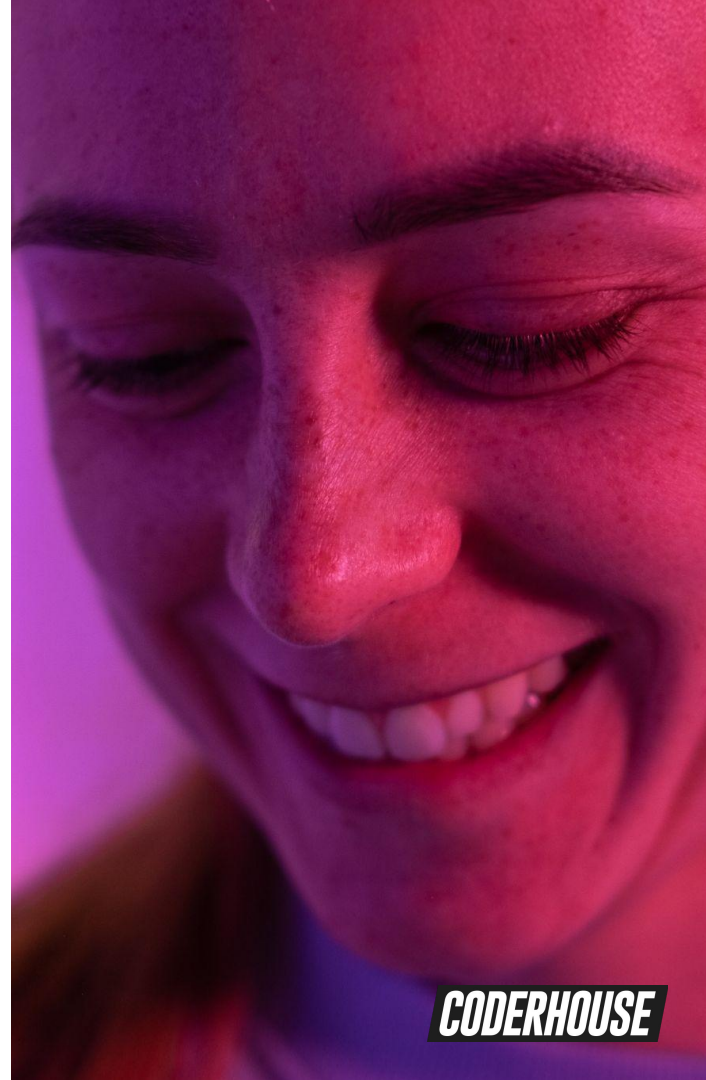
# Presentación del equipo

- ✓ Profesor responsable: Gonzalo Manoukian
- ✓ Tutores y tutoras:
  - Marcelo Furlong (Tutor Sincrónico)
  - Luis Luciuk (Tutor Sincrónico)
  - Fernanda Arrieta (Tutora Correctora)
  - Jesus Gabriel Jorge (Tutor Corrector)
  - Ezequiel Madrid (Tutor Corrector)

# Presentación de estudiantes

Por encuestas de Zoom

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



**CODERHOUSE**

Clase 00. JAVASCRIPT

# Introducción a JavaScript

# Temario

00

## Introducción a JavaScript

- ✓ [Aplicaciones web](#)
- ✓ [JavaScript](#)
- ✓ [Herramientas del curso](#)

01

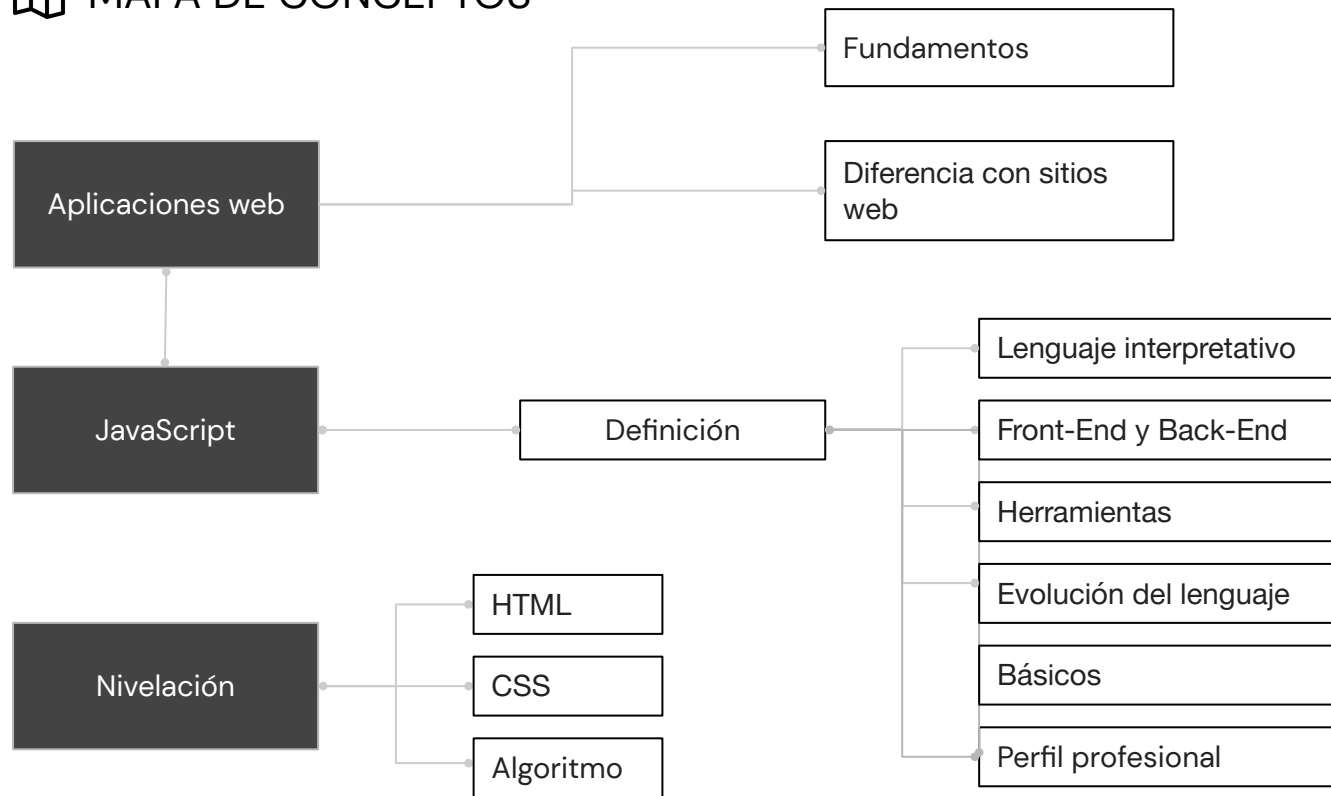
## Sintaxis y variables

- ✓ Sintáxis y código
- ✓ Variables y Valores
- ✓ Prompt, consola y alert

# Objetivos de la clase

- **Aprender** la diferencia entre sitios y aplicaciones web.
- **Reconocer** qué es un algoritmo y desarrollar capacidad resolutiva frente a un problema.
- **Comprender** los fundamentos de Javascript.

## MAPA DE CONCEPTOS

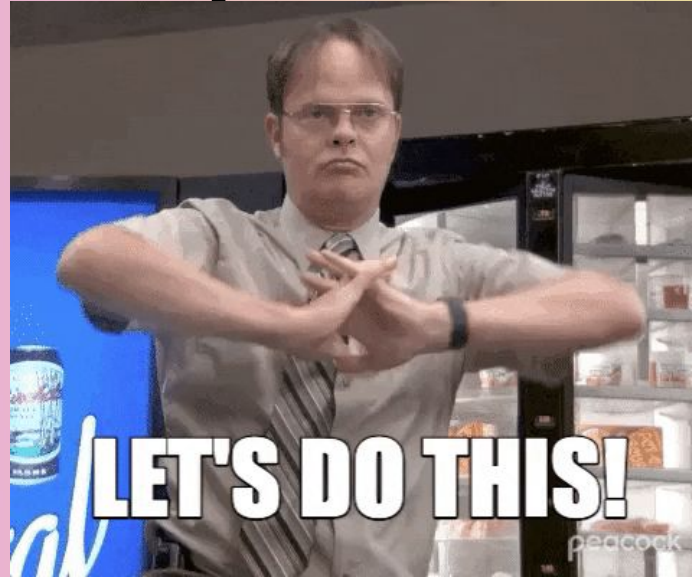




# Módulos del programa



# ¿Empezamos?



**LET'S DO THIS!**

# Aplicaciones web



PARA RECORDAR

# Aplicación Web

Mientras que los sitios web buscan brindar información estática, las web apps permiten a los usuarios realizar múltiples tareas.

Las aplicaciones web son **plataformas dinámicas e interactivas y sus funcionalidades están en constante mantenimiento y mejora.**

# Aplicación Web

Plataformas como  
Mercado Libre,  
Youtube, Gmail,  
Facebook,  
CoderHouse son web  
apps por la cantidad  
de funcionalidades  
que ofrecen.



# Aplicación Web

En la actualidad, el desarrollo desde cero de sitios web estáticos, es decir, aquellos cuya información no cambia en respuesta a las acciones del usuario, es poco frecuente.

Lo que se busca construir en el ámbito laboral, son plataformas que ofrezcan un alto nivel de interactividad, y un variado número de funcionalidades. Ya no hablamos de sitios, sino de aplicaciones web que permiten realizar tareas importantes a sus usuarios.

# Aplicación Web

Por **funcionalidades** entendemos diversas tareas que los usuarios y clientes pueden realizar y son típicamente demandadas hoy.

La interactividad en la página es una de ellas. Por ejemplo implementar animaciones y transiciones complejas, respuestas a ciertos eventos de los usuarios (como clicar un botón), o capturar y enviar datos mediante formularios.

Es normal también consumir alguna API o servicio de backend y/o base de datos, con la cual podemos cargar y administrar la información de la página.

# JavaScript





# ¿Qué es JavaScript?

JavaScript es un **lenguaje de programación** que se utiliza **principalmente para aportar dinamismo a *sitios web* y *aplicaciones web*.**

Funciona en complemento con los lenguajes web **HTML Y CSS3**, permitiendo crear nuevas funcionalidades e interacciones avanzadas con los usuarios, así como interactuar con otras aplicaciones o servicios de backend.

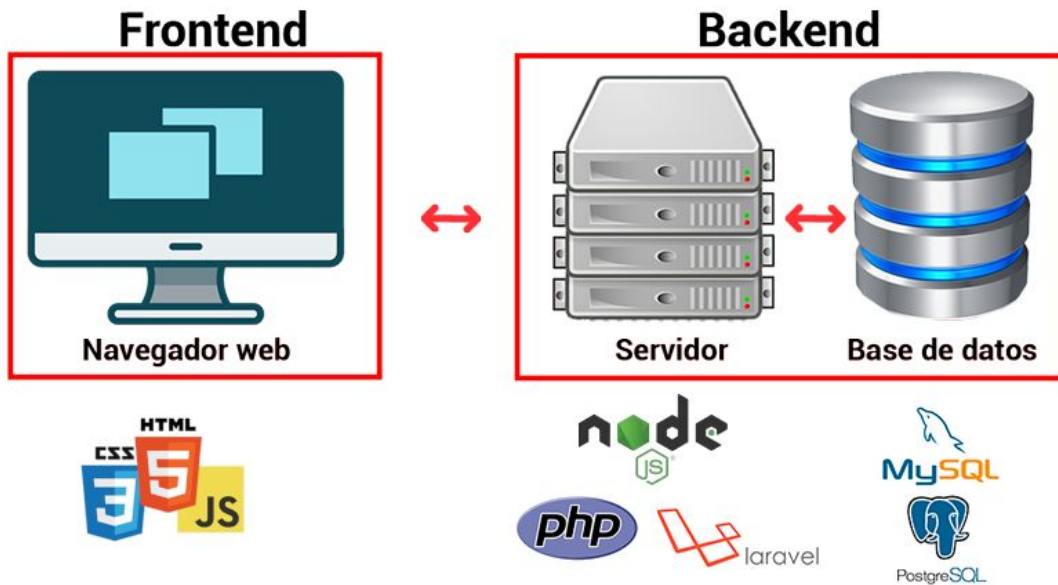
# Lenguaje Interpretado

A diferencia de un lenguaje de programación compilado, JavaScript es un lenguaje **interpretado**, lo que significa que **se ejecuta por medio de un programa intérprete**.

En nuestro caso, los **navegadores** son los programas que se encargan de interpretar y ejecutar el código JavaScript que creamos.

Cuando accedemos a alguna página a través del navegador, éste se encarga de **leer y ejecutar todos los archivos que conforman la aplicación** (HTML, CSS, JS).

# FRONT-END Y BACK-END



En el curso nos focalizaremos en el desarrollo front-end de la aplicación web

# Front-end y Back-end

Javascript se utiliza tanto para construir aplicaciones de Frontend como de Backend.

Por **Frontend** entendemos a la parte de la aplicación que corre en el navegador y con la cual interactúan los usuarios.

Como tal, estaremos creando aplicaciones con **JavaScript, HTML y CSS**; vinculando los tres lenguajes en el desarrollo un único producto.

Nuestra aplicación de Frontend también consume *datos y servicios ofrecidos por algún **Backend***.

JavaScript será la herramienta que nos permitirá comunicarnos e intercambiar información con APIs u otras aplicaciones.

# Historia de JavaScript

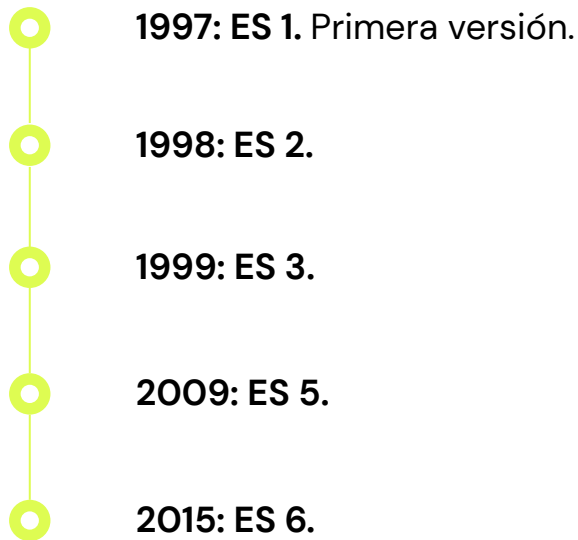
# ¿Cuándo nace?



La primera aparición pública de Javascript la encontramos en el año 1995 cuando se utiliza como herramienta del navegador **Netscape Navigator**.

El **objetivo** fue **permitir agregar programas a páginas web**.

# Evolución de Javascript



En el curso nos focalizamos en las versiones **ES6** y posteriores. Actualmente JS se encuentra en la versión 11 y desarrollo de la 12.

# Diferencia con JAVA

Si bien tanto Java con Javascript son lenguajes de programación que comparten base de su sintaxis, el uso de cada uno es distinto.

Son **dos lenguajes diferentes**, y no es correcto mencionarlos como sinónimos.







# Diferencia con JAVA

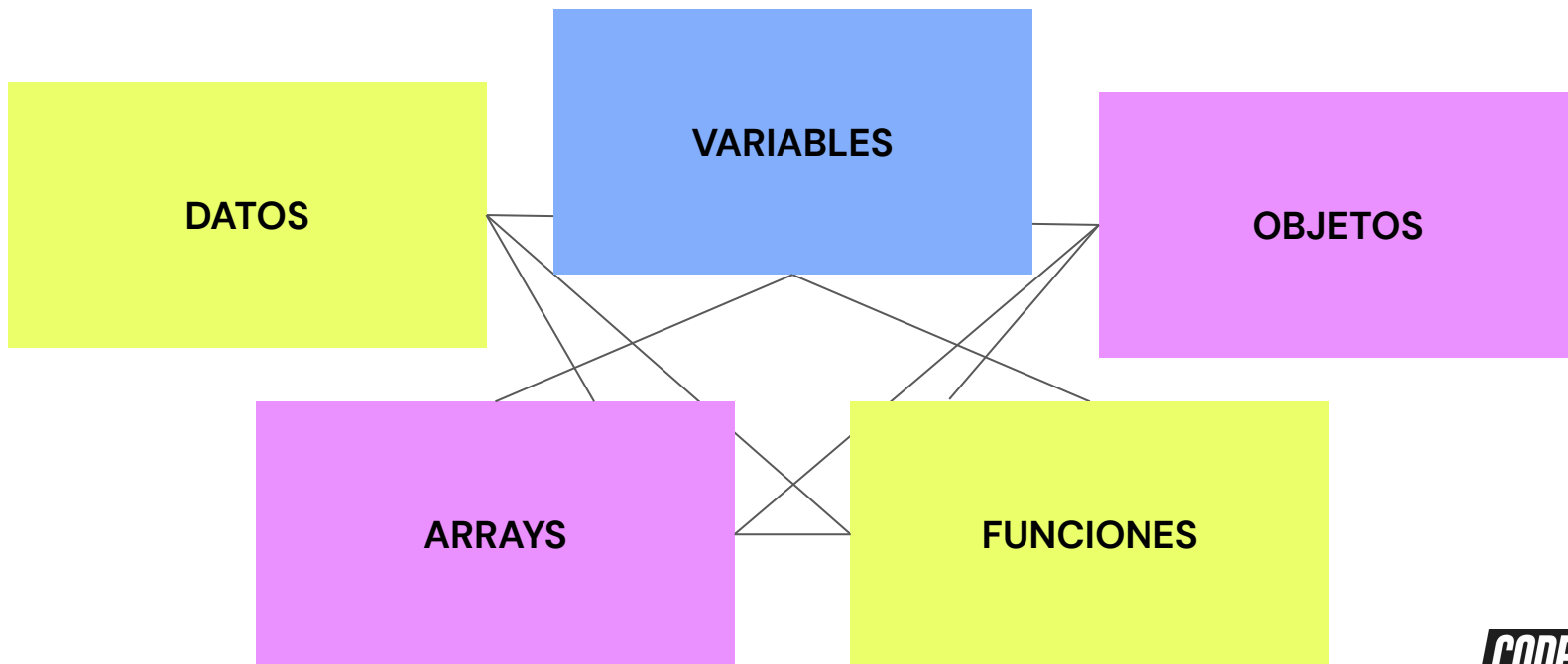
**Java** funciona a través de declaraciones que se establecen a partir de un sistema de clases en tiempo de compilación.

En cambio, **JavaScript** se basa en un sistema de tiempo de ejecución donde cada dato distinto puede representar valores booleanos, de cadena o numéricos.

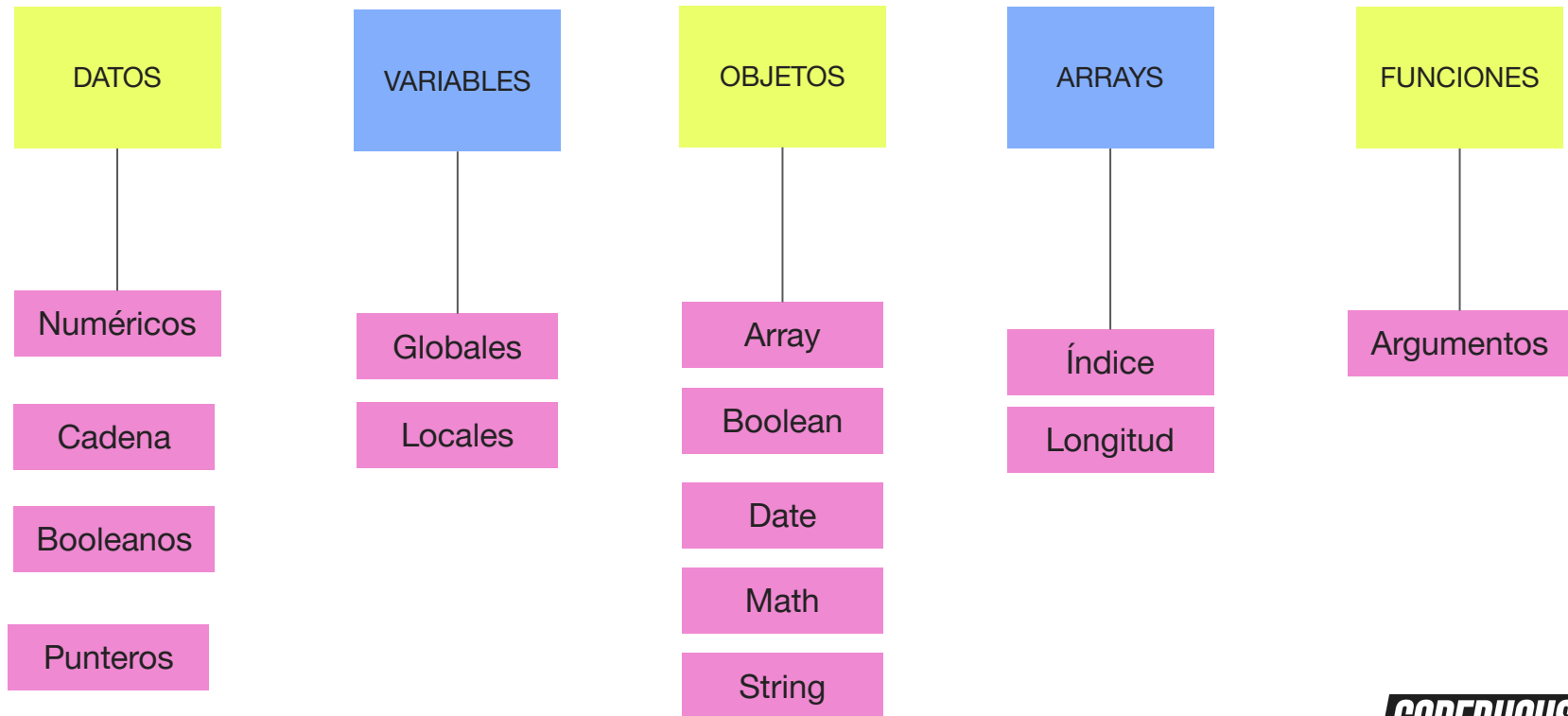
# Básicos de JavaScript

# Elementos básicos

Si bien a lo largo del curso se desarrollarán en profundidad, es necesario saber que para trabajar en y con Javascript existen ciertos elementos básicos e imprescindibles.



# Elementos básicos



**¿Cuál es el perfil de un profesional en JavaScript?**

# Algunas competencias

- ✓ Capacidad para realizar tareas de programación, desarrollo, puesta en funcionamiento y procesos de mejoras de sitios web y apps.
- ✓ Alto manejo y comprensión del área frontend en desarrollo de apps y software.
- ✓ Habilidad para pruebas y depuración de código.
- ✓ Versatilidad para trabajar con otros profesionales en procesos de optimización y mejoras integrales, como diseñadores de experiencia de usuario o profesionales del backend.

# ¿Por qué Javascript?

- ✓ Porque es uno de los lenguajes más usados en el mundo.
- ✓ Porque es muy demandado en el mercado laboral debido a su sencillez, amplitud y adaptabilidad.
- ✓ Porque la mayoría de sitios web y apps funcionan con este lenguaje.
- ✓ Porque es escalable y te brindará herramientas para profundizar luego tus conocimientos en programación y desarrollo.

**BREAK!**

**21:32 ARG VOLVEMOS!**



# Nivelación

## Encuesta!

# Relación con Desarrollo Web

# HTML

Es un "lenguaje" de marcado de etiquetas, que permite crear documentos para web.

Los términos de uso frecuente:

- ✓ Etiqueta
- ✓ Atributo
- ✓ Estructura

```
>Hello world!  
>_
```



# CSS

El CSS, en español «hojas de estilo en cascada», es un **lenguaje de diseño gráfico**, utilizado para **definir y crear la presentación de un documento estructurado**, escrito en un lenguaje de marcado.

Los términos de uso frecuente:

- ✓ **Estilo**
- ✓ **Reglas**
- ✓ **Medidas**
- ✓ **Fuente**

# Relación entre HTML, CSS & JS

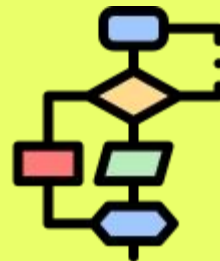
- ✓ **HTML** nos aportará a través de sus etiquetas, la estructura básica de sitio web que creemos. Funcionará en consonancia y en relación con CSS y JS.
- ✓ **CSS** lo utilizaremos para darle el estilo que queremos mostrar en nuestro sitio web a través de la definición de formato y diseño de nuestra presentación
- ✓ **JavaScript** entonces utilizará a ambos lenguajes(HTML y CSS) para controlar el comportamiento y funcionalidad de los elementos de nuestros sitio web.



# Algoritmos

# Algoritmo

En programación, un algoritmo es un **conjunto de procedimientos o funciones ordenados que se necesitan para realizar cierta operación o acción**. Por ejemplo, en una suma el algoritmo implica tomar un dato, sumarlo a otro y obtener un resultado.



# Algoritmo

Pensar en algoritmos es una práctica que debemos fortalecer como desarrolladores. Consiste en **encarar un problema complejo y dividir su resolución en diversos pasos, pensar cómo resolver cada uno y luego secuenciarlos correctamente** para llegar al **resultado esperado**.



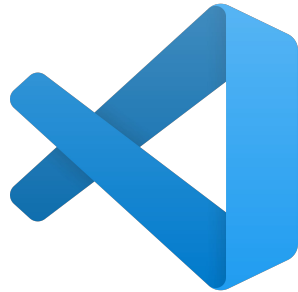
# Herramientas a utilizar en el curso

# Navegadores Web



[Chrome](#), [Firefox](#), [Edge](#) u [Opera](#).

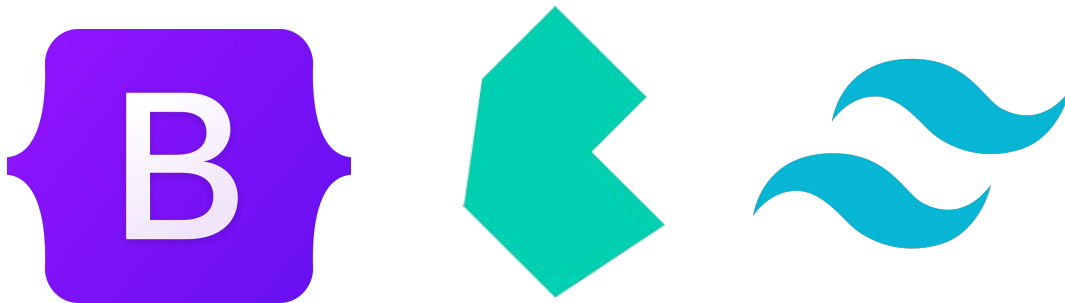
# Editores de código



[Visual Studio Code](#), [Sublime Text](#) o [Atom](#).

IMPORTANTE: Para la clase 1 ya tendrán que tener instalado/funcionando un editor de código

# Frameworks



[Bootstrap 5](#), [Bulma](#) o [Tailwind CSS](#)

# Librerías



[momentJS](#), [Sweet Alert](#), [Axios](#).

# Servidores



[Live Server \(VS Code\)](#), [XAMPP](#) , [WampServer](#).

¿Preguntas?



**Completa esta clase  
con los siguientes  
CoderTips**



# Videos y Podcasts

- ✓ [Aprende Programación Web y construye el futuro de nuestra humanidad](#) | Coderhouse
- ✓ [Desarrollo freelance](#) | Coderhouse
- ✓ [Desarrollo profesional](#) | Coderhouse
- ✓ [CoderNews](#) | Coderhouse
- ✓ [Serie de Branding](#) | Coderhouse
- ✓ [Serie para Emprendedores](#) | Coderhouse [Serie Aprende a Usar TikTok](#) | Coderhouse
- ✓ [Serie Finanzas Personales](#) | Coderhouse
- ✓ [CoderConf](#) | Coderhouse

¿Sabías que  
**premiamos a nuestros estudiantes**  
por su dedicación?

Conoce los [beneficios](#) del Top 10

# Resumen de la clase hoy

- ✓ Fundamentos de Desarrollo con JS.
- ✓ Básicos de Javascript
- ✓ Nivelación
- ✓ Contenido del curso

**Opina y valora**  
**esta clase**

**#DemocratizandoLaEducación**

**Muchas gracias.**