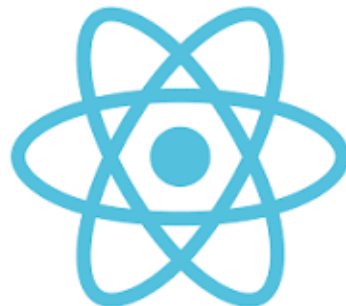


Agencia de
Aprendizaje
a lo largo
de la vida

ESPECIALIZACIÓN REACT



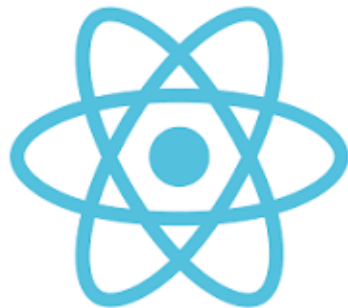


Agencia de
Aprendizaje
a lo largo
de la vida

Clase 01. REACT JS

NIVELACIÓN

NIVELACIÓN HTML, CSS Y JS



React JS es una biblioteca para **desarrollo web**,
por lo cual debemos contar con conocimientos
mínimos sobre los lenguajes que el navegador web
interpreta.



React es una biblioteca de JavaScript de código abierto para construir interfaces de usuario. Está basada en la componetización de la UI: la interfaz se divide en componentes independientes, que contienen su propio estado. Cuando el estado de un componente cambia, React vuelve a renderizar la interfaz.

Esto hace que React sea una herramienta muy útil para construir interfaces complejas, ya que permite dividir la interfaz en piezas más pequeñas y reutilizables.

Fue creada en 2011 por Jordan Walke, un ingeniero de software que trabajaba en Facebook y que quería simplificar la forma de crear interfaces de usuario complejas.

Es una biblioteca muy popular y es usada por muchas empresas como Facebook, Netflix, Airbnb, Twitter, Instagram, etc.



NETFLIX

HTML es un **lenguaje de etiquetas**,
el cual dará la estructura para
nuestras páginas web.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Especializacion Codo a Codo</title>
  </head>
  <body>
    <h1>Primer Clase de React</h1>
    <p>Especializacion Codo a Codo</p>
  </body>
</html>
```

¿CUÁLES SON LAS NOVEDADES DE HTML5?

HTML 5 incluye novedades significativas en diversos ámbitos. Este nuevo estándar supone mejoras en áreas que hasta ahora quedaban fuera del lenguaje y para las que se necesitaba utilizar otras tecnologías:

- Estructura del `<body>`
- Etiquetas para contenido específico
- Canvas
- Bases de datos locales
- Web Workers
- Aplicaciones web Offline
- Geolocalización

Estructura de una Página Web (código)



<!DOCTYPE html> Indica que la versión corresponde a HTML5.

<html lang= “es”> Es la etiqueta principal que engloba al resto de las etiquetas, el atributo *lang* define el tipo de lenguaje.

<head> Es la cabeza del documento que contiene los metadatos de la página web.

<meta charset="utf-8"> Indica al navegador qué tipo de caracteres contiene la página, con el atributo *charset* vamos a indicar el conjunto de caracteres que vamos a usar y con el valor "utf-8" abarcamos a la mayoría de los sistemas de escritura.

<title> Indica el título de la página Web, que se visualiza en la barra de título del navegador.

<body> Es el cuerpo del documento donde va a estar todo el contenido que vamos a mostrar.

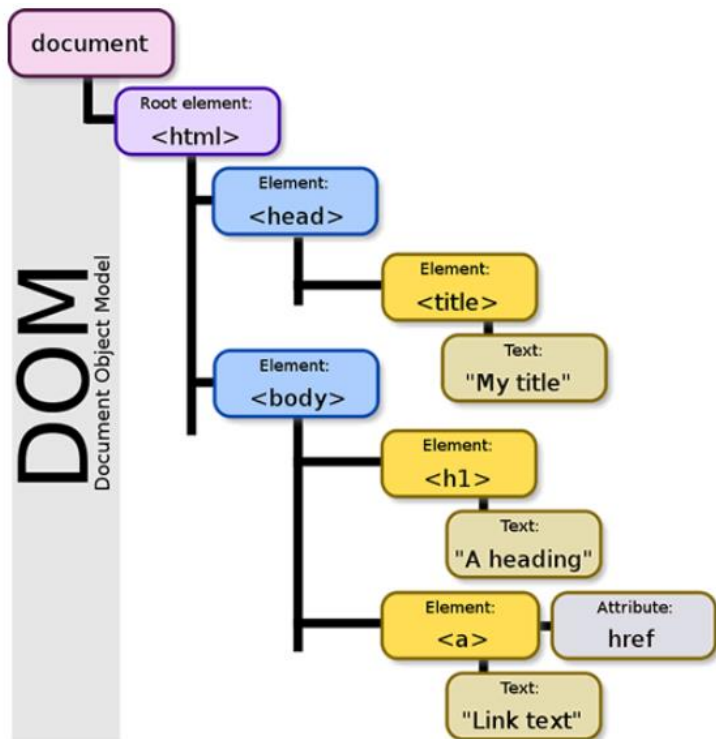
¿Entonces que es el DOM?

El DOM es otra forma de ver el contenido de la página. Con el DOM, todos los elementos HTML se insertan en un árbol cuyos nodos son las marcas HTML y las hojas, los valores propiamente dichos de las marcas. Por medio de JavaScript podemos acceder y modificar este árbol de marcas y hacer que la página varíe luego que ya se haya mostrado en el navegador. De aquí el nombre de esta tecnología DHTML es decir Dynamic Hyper Text Markup Language.

Mediante el DOM podemos acceder al contenido y estilo de cada marca del documento y modificarlo de acuerdo a algún evento.

Mediante el DOM podemos insertar, borrar, modificar marcas HTML. Podemos acceder a la hoja de estilo definida a la página y dinámicamente agregar, modificar o borrar propiedades. Todo esto sin tener que recargar la página del servidor, es decir todo se hace en el cliente (navegador) mediante JavaScript.

Importancia del DOM (Document Object Model)



El modelo de objetos de documento (DOM) es una interfaz multiplataforma e independiente del lenguaje que trata un documento XML o HTML como una estructura de árbol en la que cada nodo es un objeto que representa una parte del documento.

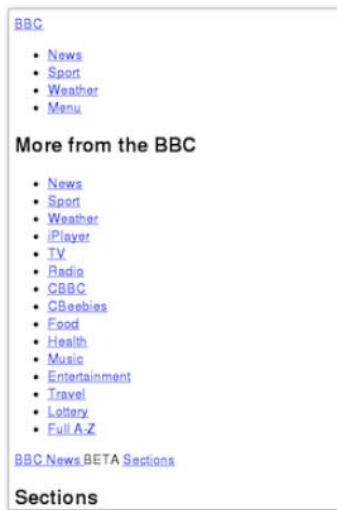
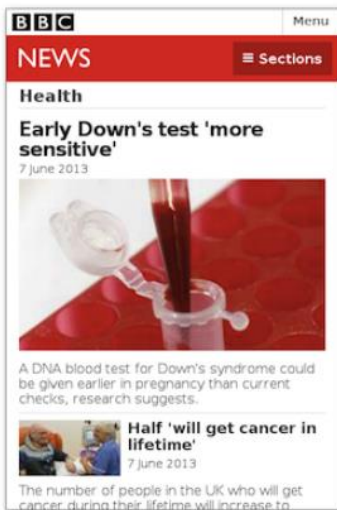
El DOM representa un documento con un árbol lógico. Cada rama del árbol termina en un nodo y cada nodo contiene objetos.

Los métodos DOM permiten el acceso programático al árbol; con ellos se puede cambiar la estructura, estilo o contenido de un documento. Los nodos pueden tener controladores de eventos adjuntos. Una vez que se activa un evento, se ejecutan los controladores de eventos.

¿Qué significa CSS? |

Cascading Style Sheets

- Hojas de estilo en cascada.
- Compuestas de REGLAS, SELECTORES y DECLARACIONES.



CSS (cascading style sheets - hojas de estilo en cascada) es un lenguaje de diseño gráfico con el cual podremos **dar estilos** (diseño, colores, márgenes) a nuestras webs desarrolladas con HTML. Veremos la sintaxis básica de CSS

¿QUÉ ES?

JavaScript es el **lenguaje de programación web** por excelencia.

Decimos que se trata de un lenguaje de programación **interpretado**.

Su uso más conocido es del lado del cliente (**client-side**), corriendo en el navegador web, permite mejoras en la interfaz de usuario.

React está desarrollado en JS, por eso lo llamamos **reactjs**.



¿Qué es JavaScript de vainilla?

Hablar de Vanilla JavaScript (o Vanilla JS) es hablar de JavaScript sin ningún tipo de framework o librería adicional, es decir, JavaScript puro

Cada vez los frameworks son más sofisticados, más grandes y cada vez se parecen menos a JavaScript. De hecho, algunos directamente tienen un pseudolenguaje que tienes que compilar, para lo que tienes que tener instalado nodeJS, npm, webpack, etc.



Diferencias entre librerías y frameworks

Una librería te permite solucionar un problema concreto, mientras que un framework te brinda un set de herramientas para desarrollar sistemas o aplicaciones. Un framework, por lo general, contiene librerías (algunas veces no), provee buenas prácticas y resulta toda una experiencia de desarrollo.

var

`var` es la forma declarar una variable en ES5 . Esta es una _palabra clave_ genérica para "variable".

let

`let` es una nueva palabra clave de ES6, esto asignará una variable muy similar a `var`, pero con un comportamiento un poco diferente. Lo más notable es que difiere al crear un "nivel de _scope_" (hablaremos sobre esto más adelante).

const

`const` también es nuevo en ES6. Un `const` es una variable que no se podrá cambiar. Esto es la abreviatura de "constante".

funcion flecha, se elimina la palabra function y se coloca la flecha luego de los parentesis

ejemplos

si solamente usamos un parámetro no lleva paréntesis, y si tiene una sola expresión o línea de códigos

NO LLEVA LLAVES

```
const saludarOk = nombre => document.write(`Hola ${nombre} como estas`);  
saludarOk("Nicolino")
```

```
let suma = (num1, num2) => {  
  let resultado = num1+ num2  
  document.write(resultado);  
  document.write("<br>")  
}  
suma (32,32) ; //64
```


REACT JS: LOS INICIOS

¿Qué es React?

React es una biblioteca de JavaScript de código abierto para construir interfaces de usuario. Está basada en la componetización de la UI: la interfaz se divide en componentes independientes, que contienen su propio estado. Cuando el estado de un componente cambia, React vuelve a renderizar la interfaz.

Esto hace que React sea una herramienta muy útil para construir interfaces complejas, ya que permite dividir la interfaz en piezas más pequeñas y reutilizables.

Fue creada en 2011 por Jordan Walke, un ingeniero de software que trabajaba en Facebook y que quería simplificar la forma de crear interfaces de usuario complejas.

Es una biblioteca muy popular y es usada por muchas empresas como Facebook, Netflix, Airbnb, Twitter, Instagram, etc.

¿Cuáles son las características principales de React?

Las características principales de React son:

Componentes: React está basado en la componetización de la UI. La interfaz se divide en componentes independientes, que contienen su propio estado. Cuando el estado de un componente cambia, React vuelve a renderizar la interfaz.

Virtual DOM: React usa un DOM virtual para renderizar los componentes. El DOM virtual es una representación en memoria del DOM real. Cuando el estado de un componente cambia, React vuelve a renderizar la interfaz. En lugar de modificar el DOM real, React modifica el DOM virtual y, a continuación, compara el DOM virtual con el DOM real. De esta forma, React sabe qué cambios se deben aplicar al DOM real.

Declarativo: React es declarativo, lo que significa que no se especifica cómo se debe realizar una tarea, sino qué se debe realizar. Esto hace que el código sea más fácil de entender y de mantener.

Unidireccional: React es unidireccional, lo que significa que los datos fluyen en una sola dirección. Los datos fluyen de los componentes padres a los componentes hijos.

Universal: React se puede ejecutar tanto en el cliente como en el servidor. Además, puedes usar React Native para crear aplicaciones nativas para Android e iOS.

¿Qué significa exactamente que sea declarativo?



No le decimos cómo debe renderizar la interfaz a base de instrucciones. Le decimos qué debe renderizar y React se encarga de renderizarlo.

Un ejemplo entre declarativo e imperativo:

// Declarativo

```
const element = <h1>Hello, world</h1>
```

// Imperativo

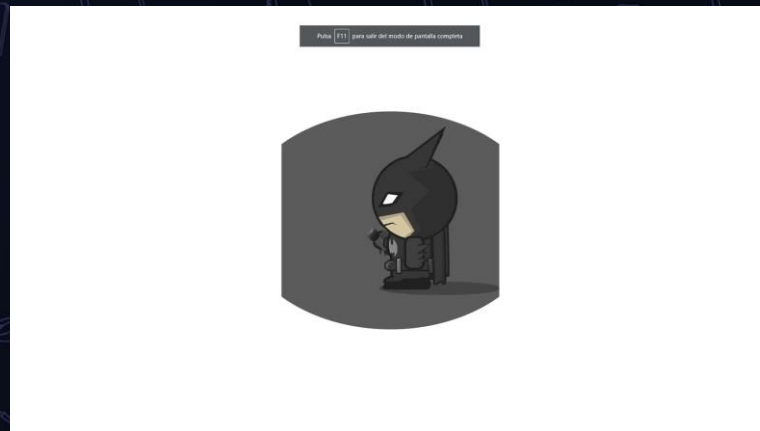
```
const element = document.createElement('h1')
```

```
element.innerHTML = 'Hello, world'
```

PROYECTO DE REPASO Y TAREA 1

En la carpeta de la clase 1 en drive encontraran los archivos necesarios para la creación del proyecto

HTML
CSS
JAVASCRIPT
MANIPULACION DOM
EVENTOS



Luego de finalizar el proyecto creado en clase, te invitamos a crear tu propio proyecto teniendo en cuenta todo lo aprendido.

Tu practica inspirada en el proyecto tiene que tener animación y sonido

PROYECTO FINAL

El Proyecto Final se construye a partir de los **desafíos** que se realizan
clase a clase

El objetivo es que cada estudiante pueda utilizar su Proyecto Final como
parte de su portfolio personal.

El **proyecto final** se debe subir a la plataforma la ante-última o última
clase del curso.

PROYECTO FINAL

Recomendaciones

- Empezar a formar un grupo/equipo de 4 personas.
- Comenzar a pensar en el contenido del proyecto web.
- Recuerden que la temática es libre.
- Recomendamos visitar sitios web de similares temáticas.