



**¡Les damos la
bienvenida!**

¿Comenzamos?

Esta clase va a ser

- grabada

Clase 02. REACT JS

Instalación y configuración del entorno

Temario

01

Introducción a React Js

- ✓ ¿Qué es React?
- ✓ ¿Qué es un componente?
- ✓ Código declarativo VS Código imperativo
- ✓ ¿Qué es una expresión?
- ✓ ¿Qué es una función?

02

Instalación y configuración del entorno

- ✓ Funcionamiento de React Js
- ✓ ¿Qué es virtual DOM?
- ✓ ¿Qué es NODE?
- ✓ NODE JS ¿Qué es NPM?
- ✓ Crear una aplicación utilizando el CLI

03

JSX y Transpiling

- ✓ Sugar Syntax
- ✓ Polyfills y la retrocompatibilidad
- ✓ Bundling con Webpack
- ✓ Transpiling
- ✓ Jsx

Objetivos de la clase

- Comprender el funcionamiento del virtual DOM en React.
- Instalar y configurar NodeJS
- Crear una app utilizando el CLI.

CLASE N°1

Glosario

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

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

DOCTYPE: no es una etiqueta, sino una instrucción para indicar al navegador qué versión de HTML vamos a utilizar.

DOM (Document Object Model o modelo de objetos del navegador): nos sirve para acceder a cualquiera de los componentes que hay dentro de una página.

CLASE N°#1

Glosario

HTML 5: es una nueva versión de diversas especificaciones, entre las que se encuentran: HTML 4–XHTML 1–CSS Nivel 2–DOM Nivel 2 (DOM = Document Object Model).

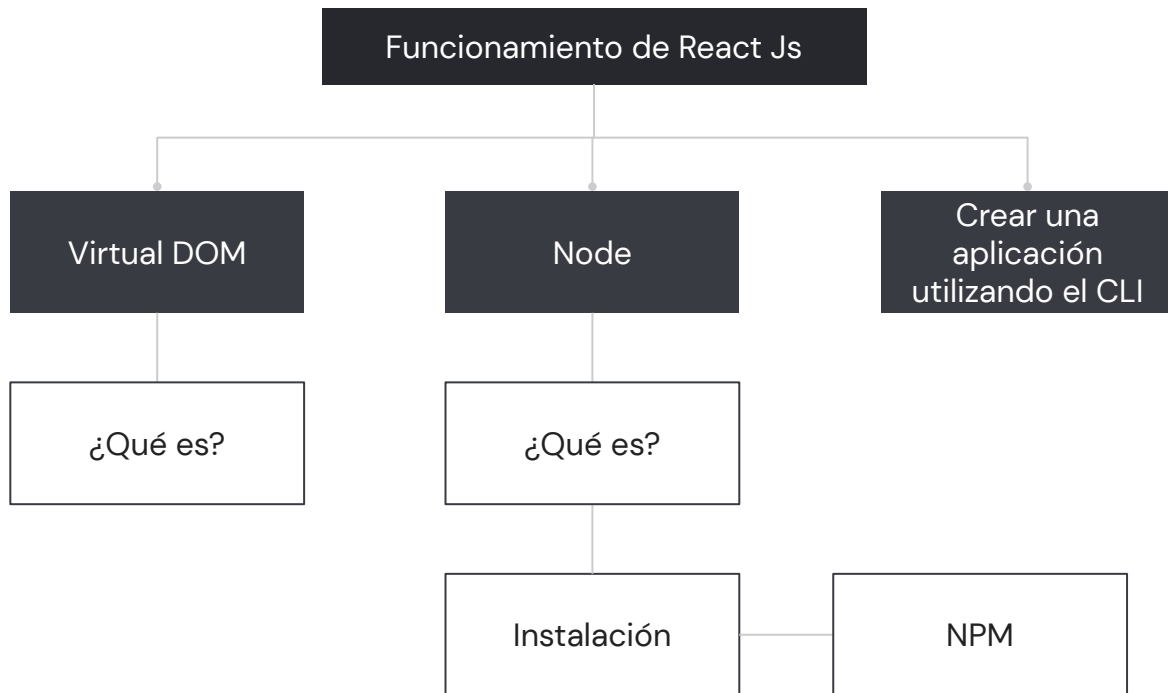
Javascript: es el lenguaje de programación web por excelencia. Decimos que se trata de un lenguaje de programación interpretado.

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.

.



MAPA DE CONCEPTOS



RECAP: HTML, DOM, JS

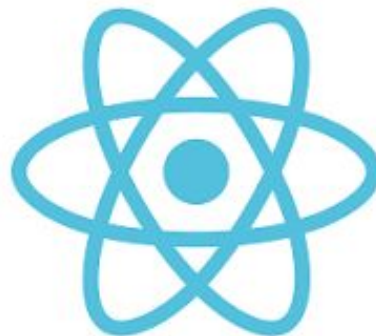
Funcionamiento de **REACT JS**

React

¿Cómo llega React a la performance que tanta fama le trae?

Hablemos de tres conceptos:

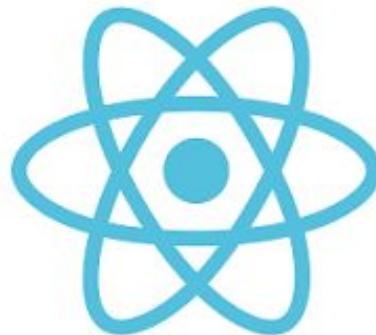
Virtual DOM vs React Fiber y la Reconciliación.



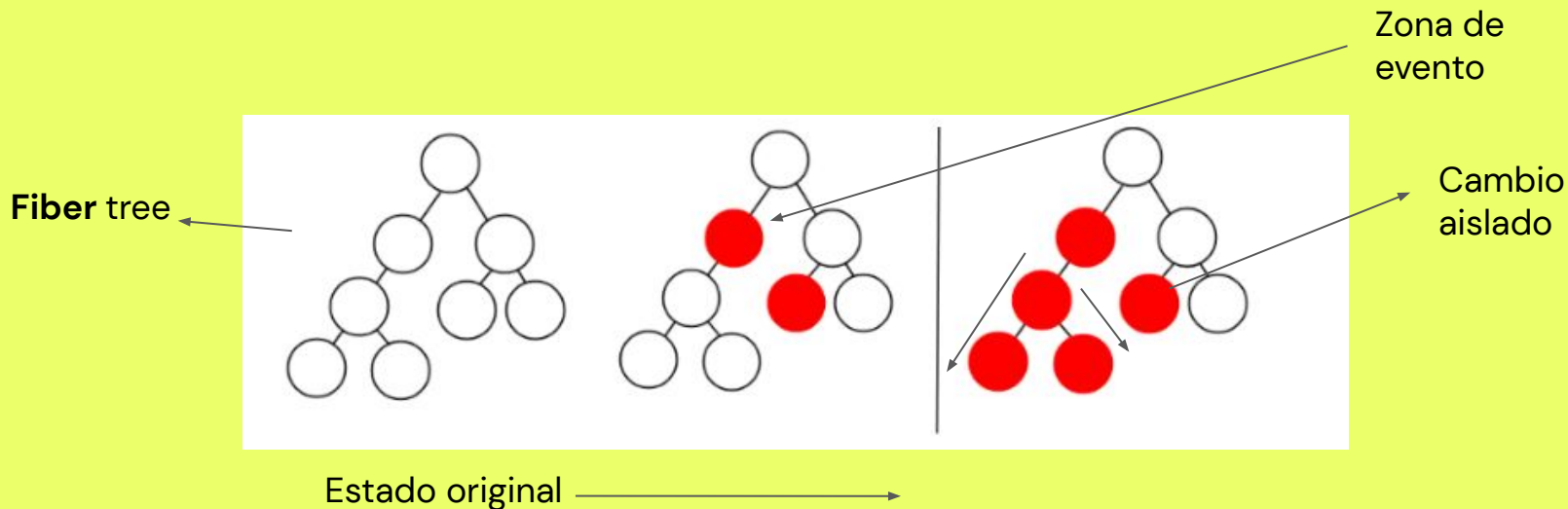
React

Primera premisa

El acceso indiscriminado al DOM **es caro**, entonces se requirió encontrar una manera de realizarlo de la manera **más óptima** posible.

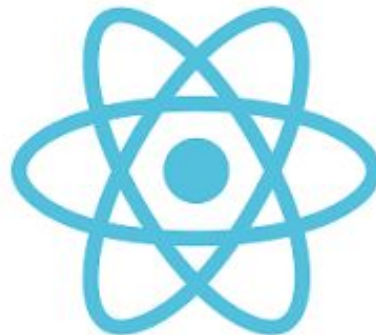


Primera premisa: Optimicemos los movimientos



React

En vez de aplicar **uno a uno** los cambios en los cinco nodos, **React** procesa este resultado en una memoria. Calcula el área de impacto y determina la menor cantidad de movimientos de modo **heurístico**, por lo que también sabe donde **no pueden haber ocurrido** cambios.

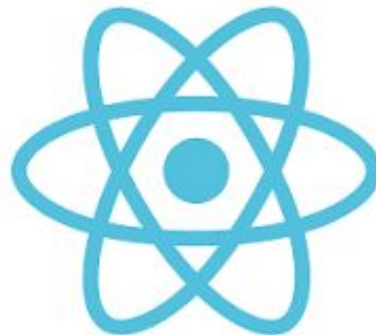


React

Segunda premisa: Flujo de datos unidireccional

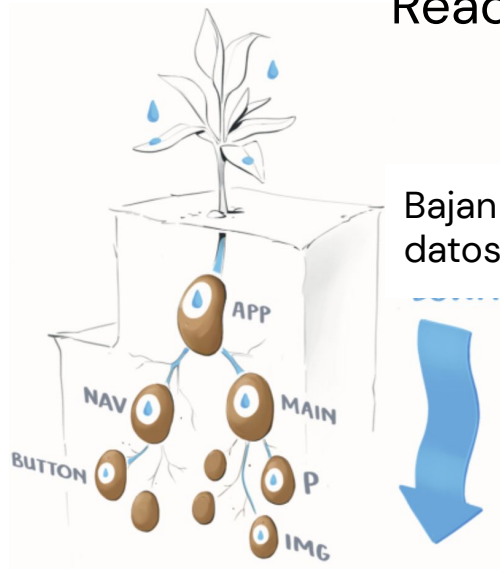
Para establecer esa seguridad, requiere que los datos y los cambios idealmente se provoquen de una manera específica con dos características:

Unidireccionalidad / De arriba hacia abajo



Flujo de datos

Reacción



Acción



Resumiendo:

¿Qué es virtual DOM?

Virtual DOM

Es un patrón de comportamiento, y React lo implementa con una tecnología llamada "Fiber".

En sí resulta ser todo lo que React sabe de tu aplicación y cada nodo o fibra.

Esto es básicamente lo que React hace con el Virtual DOM: una representación virtual de la UI que se mantiene en memoria y en sincronía "reconciliado" con el DOM "real".

Virtual DOM

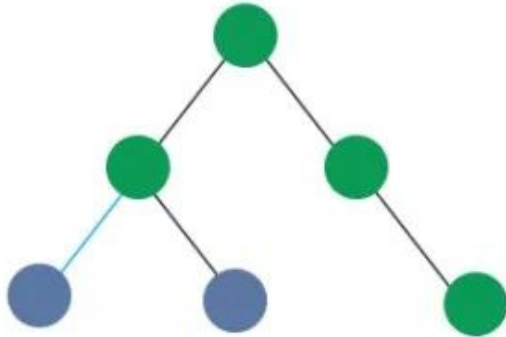
Resumiendo el proceso

- ✓ En primer lugar, React ejecuta un algoritmo de “diffing” que identifica lo que ha cambiado.
- ✓ El segundo paso es la reconciliación, donde se actualiza el DOM con los resultados de diff.

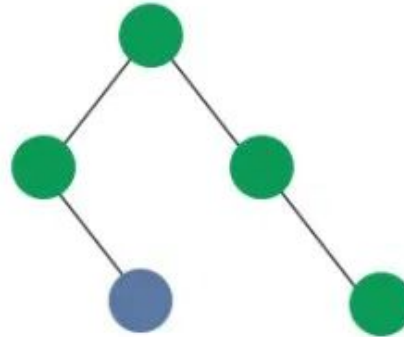
React se encarga de todo esto, nosotros solo aprenderemos a ayudarlo

Virtual DOM

Virtual DOM



Real DOM



¿Preguntas?

¿Qué es NODE?

NODE

Node.js es un entorno de ejecución de JavaScript, que le permite al código en js ser ejecutado en nuestra computadora.

Podemos darle a node un archivo de js y éste puede ejecutarlo, y darle acceso a recursos de nuestra computadora (IO, Network, Etc).

El “Node” de java es conocido como la JVM



¡Coder news!

Recientemente ha nacido un nuevo entorno de ejecución que no es node sino **Deno** (no-de = de-no)

Creado por uno de los **padres** de node.js -Ryan Dahl- también permite ejecución de código **js** y **typescript** pero con una nueva perspectiva en **seguridad**.

<https://deno.land/>



CODERHOUSE

NODE JS

¿Qué es NPM?

Node Packager Manager / NPM

Cuando usamos Node.js, rápidamente tenemos que instalar módulos nuevos (librerías), ya que al ser un sistema fuertemente modular viene prácticamente “vacío”.

Por lo tanto, para utilizar una funcionalidad de alguna librería publicada, deberemos instalar módulos adicionales.

Esta operación se realiza de forma muy sencilla con la herramienta npm (Node Package Manager).



Esta herramienta funciona de dos formas:

- ✓ Como un repositorio ampliamente utilizado para la publicación de proyectos Node.js de código abierto.
- ✓ Como una herramienta de línea de comandos. Esta utilidad ayuda a instalar y desinstalar paquetes, gestionar versiones y gestionar dependencias necesarias para ejecutar un proyecto.



Instalación de NODE JS

Para poder instalar una aplicación de React JS desde el CLI, debemos previamente instalar Node.js.

1

Ingresa a:
<https://nodejs.org/en/>

2

Descargar la última
versión de Node.js

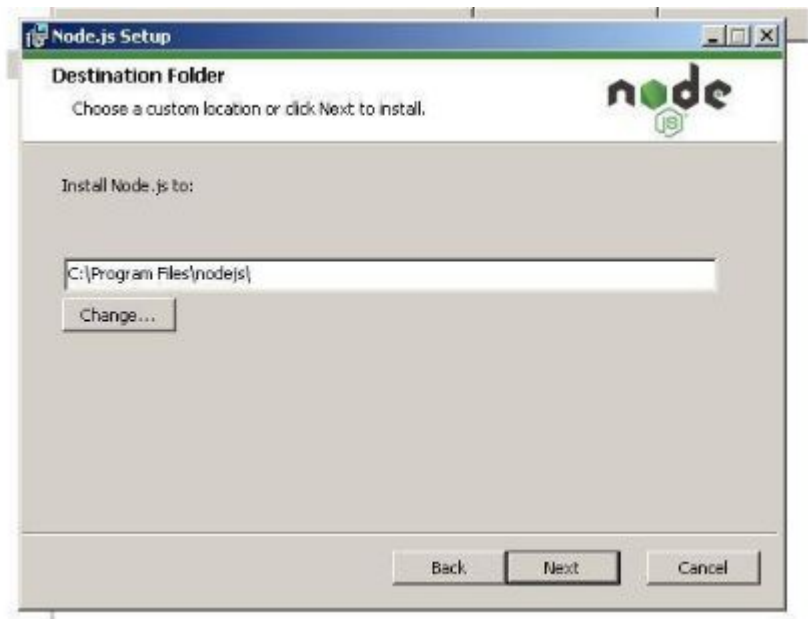
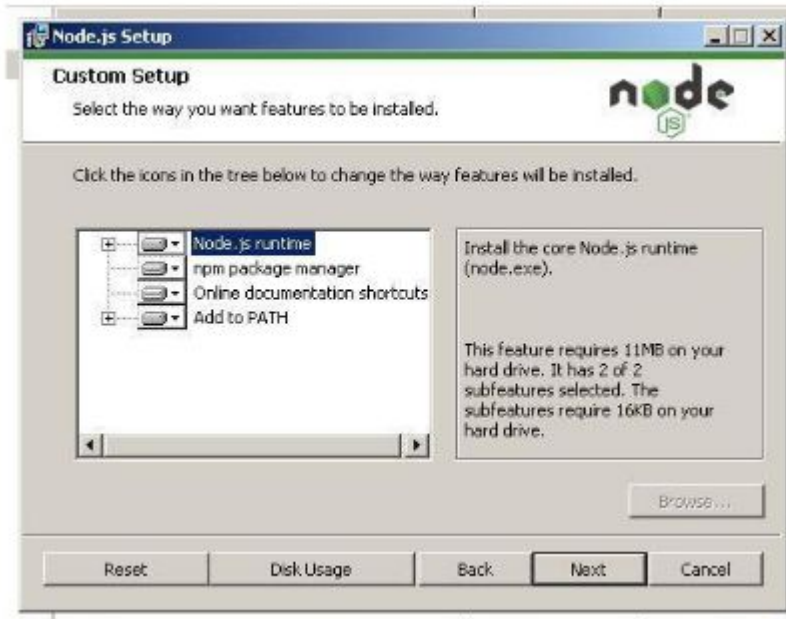
3

Ejecutar el archivo
descargado y
ejecutar los
siguientes pasos

Instalación de NODE JS



Instalación de NODE JS



Instalación de NODE JS





Break

¡10 minutos y volvemos!

Crear una aplicación utilizando el CLI

¿Qué es el CLI?

La **interfaz de línea de comandos** o interfaz de línea de órdenes, es un método que permite a los usuarios **dar instrucciones** a algún programa informático **por medio de una línea de texto** simple.

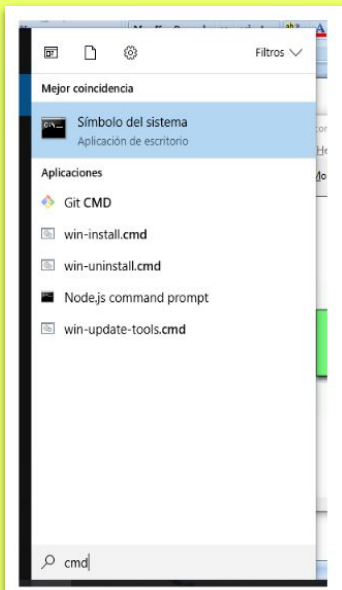
React tiene su propia CLI, pero actualmente solo admite la creación de una aplicación (create-react-app).



Ejemplo en vivo

Vamos al código.

Abrir la consola



En Windows



En Linux

Ubicarnos en un directorio específico

```
G:\sites>cd react
```

```
G:\sites\react>
```

Con el comando `cd` podemos ingresar al directorio sobre el cual vamos a crear nuestra aplicación en react. Con `cd` volvemos al directorio anterior.

Instalar REACT JS

```
G:\sites\react>npm install -g create-react-app
```

Debemos ejecutar el comando `npm install -g create-react-app`

```
G:\sites\react>create-react-app my-app  
  
Creating a new React app in G:\sites\react\my-app.
```

Crear la aplicación

Debemos ejecutar `create-react-app my-app`

Coder “Pro Tips”

NPX – todo en un comando – eXecute

```
npx create-react-app nombre-de-app
```

```
cd coder-app
```

Ejecutar aplicación en el navegador

```
G:\sites\react\utn>npm start

> utn@0.1.0 start G:\sites\react\utn
> react-scripts start
Starting the development server...
Compiled successfully!

You can now view utn in the browser.

Local:            http://localhost:3000/
On Your Network:  http://192.168.0.11:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Para ejecutar una aplicación y poder acceder desde el navegador debemos ejecutar (dentro del directorio de la aplicación creada) `npm start`

Ejecutar aplicación en el navegador



Welcome to React

To get started, edit `src/App.js` and save to reload.

Ahora escribe la dirección obtenida (Ej:localhost:3000) en tu navegador y el resultado obtenido será el siguiente:



Crear la app utilizando el CLI

Crea una aplicación utilizando el CLI con el nombre de tu tienda.

Duración: **15 minutos**



ACTIVIDAD EN CLASE

Crear la app utilizando el CLI

Descripción de la actividad.

Crea una aplicación utilizando el CLI con el nombre de tu tienda, y ejecuta los comandos necesarios para instalar React, configurarlo y visualizarlo en tu navegador.

Aspectos a incluir en el entregable:

- ✓ Link al repositorio de github
- ✓ Dentro del repositorio, deberían encontrarse las carpetas y los archivos src, README.md y package.json, entre otros



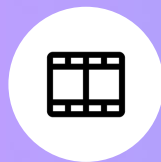
ACTIVIDAD EN CLASE

Crear la app utilizando el CLI

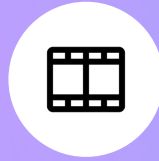
Ejemplo:

	deejay47 Update README.md	f575471 2 days ago	 92 commits
	public	Algo de SEO	2 days ago
	src	Fix urls	2 days ago
	.firebaserc	Firebase	2 days ago
	.gitignore	Exclusión carpetas	2 days ago
	Changelog.md	Version	2 days ago
	README.md	Update README.md	2 days ago
	TODO.md	Avances y pendientes del proyecto	2 days ago
	firebase.json	Firebase	2 days ago
	package-lock.json	Nuevos componentes visuales para circuito de pagos	2 days ago
	package.json	Nuevos componentes visuales para circuito de pagos	2 days ago

¿Preguntas?



¿Quieres saber más?
**Te dejamos material
ampliado de la clase**



Nuestro GITHUB
LINK



MATERIAL AMPLIADO

Recursos multimedia

- ✓ Amler (2016). ReactJS by Example (1 ed.). EEUU, Packt.
- ✓ Stein, J. (2016). ReactJS Cookbook (1 ed.). EEUU, Packt.
- ✓ <https://carlosvillu.com/introduccion-a-reactjs/> | **Carlos Villu**
- ✓ <https://reactjs.org/tutorial/tutorial.html> | **Reactjs.org**
- ✓ <https://reactjs.org/docs/hello-world.html> | **Reactjs.org**

Resumen de la clase hoy

- ✓ Virtual DOM
- ✓ Nodejs
- ✓ Creación de App con CLI

Opina y valora
esta clase

Muchas gracias.

#DemocratizandoLaEducación