



Clase 02. React JS

# ***INSTALACIÓN Y CONFIGURACIÓN DEL ENTORNO***

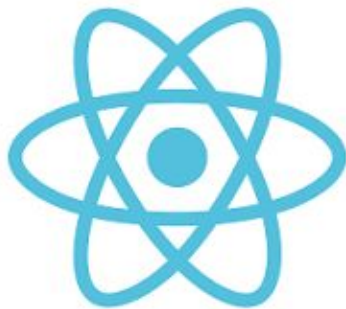


## ***OBJETIVOS DE LA CLASE***

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

# ***RECAP: HTML, DOM, JS***

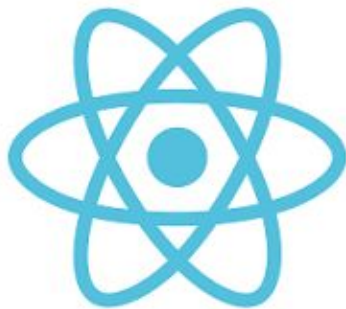
# ***FUNCIONAMIENTO DE REACT JS***



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

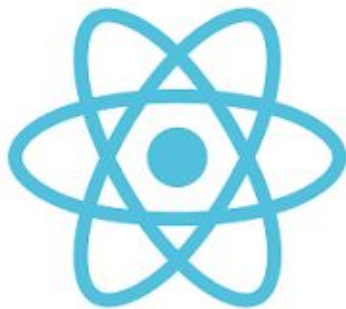
Hablemos de tres conceptos:

**Virtual DOM** vs **React Fiber** y la **Reconciliación**

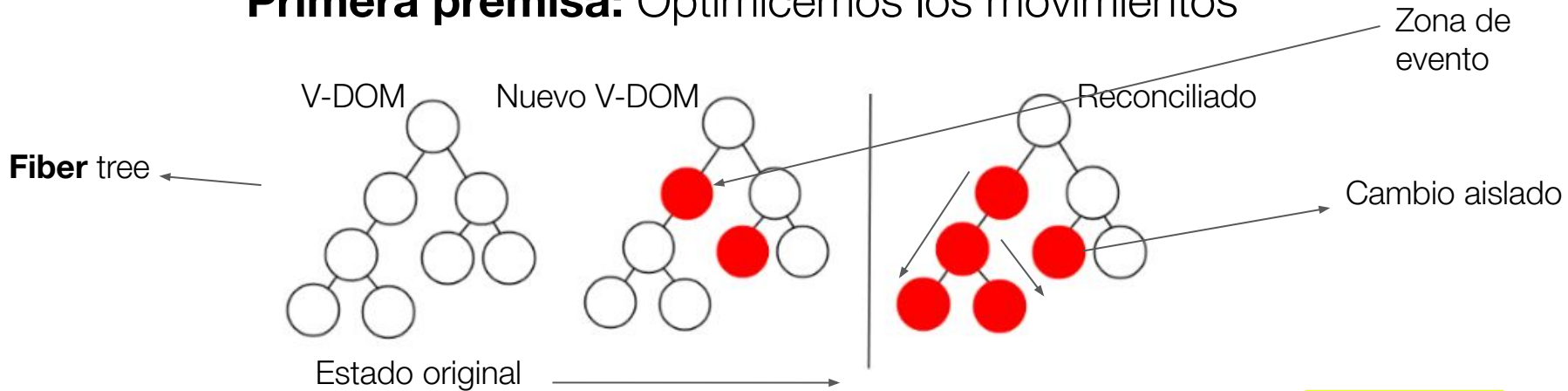


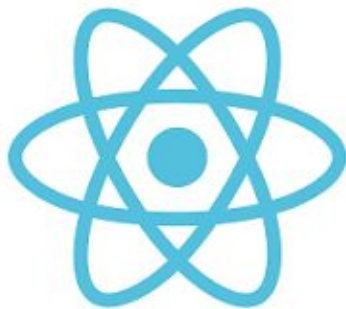
## Primera premisa

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



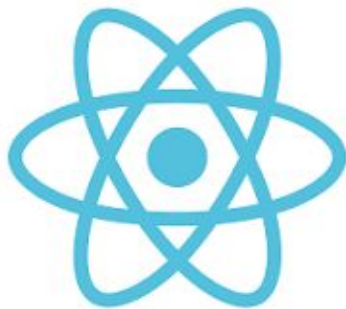
## Primera premisa: Optimicemos los movimientos





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.



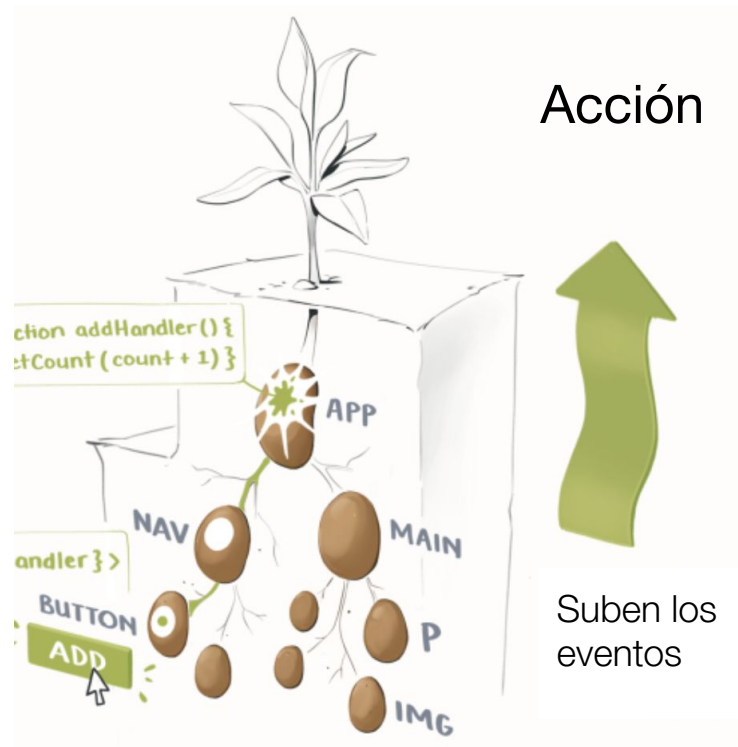
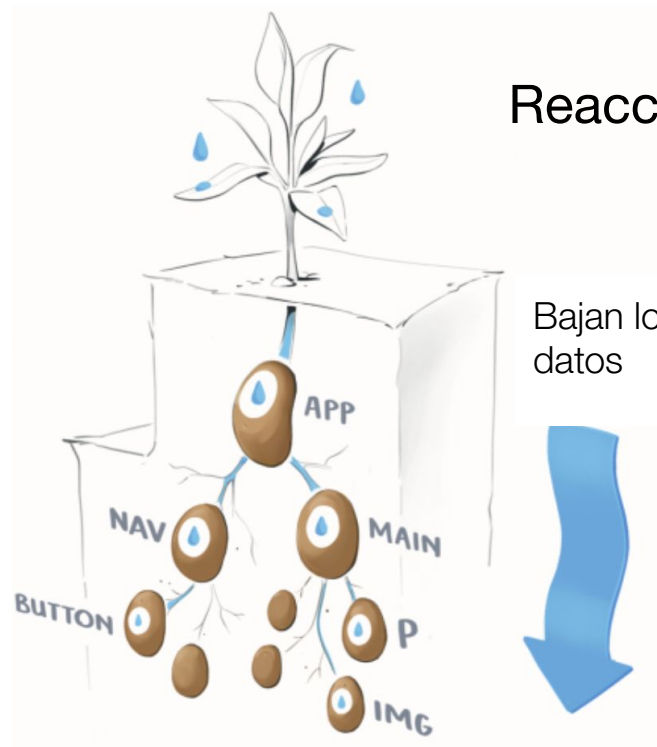


**Segunda premisa:** Flujo de datos unidireccional

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



***RESUMIENDO:***  
***¿QUÉ ES EL 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 IU que se mantiene en memoria y en sincronía “reconciliado” con el DOM “real”.**

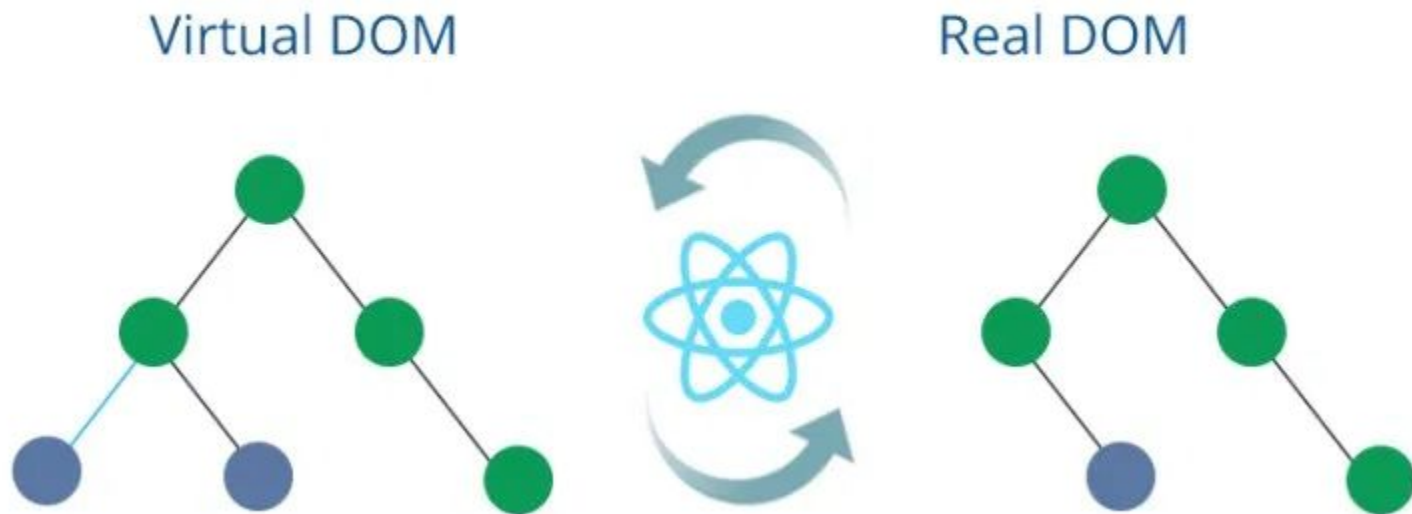
# ***Virtual DOM***

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***



***¿PREGUNTAS?***



***¿QUÉ ES 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/>



# ***NODE JS ¿QUÉ ES NPM?***

# ***Node Package 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).



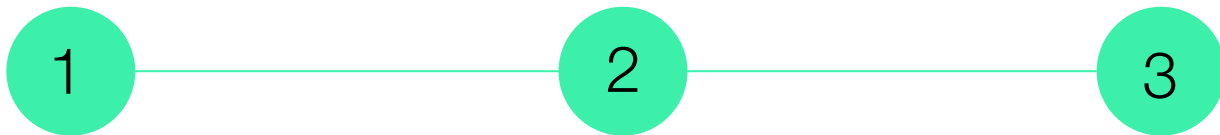
# ***Esta herramienta funciona de dos formas***

- 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.



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

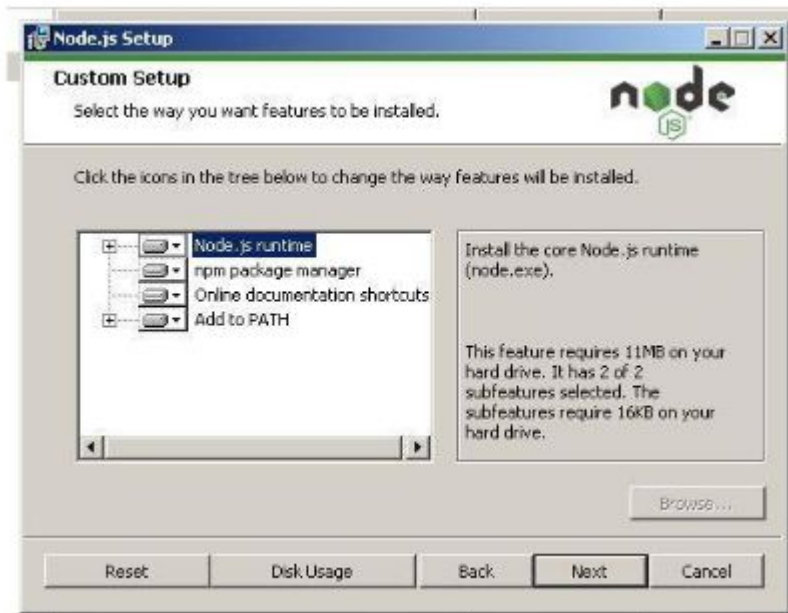
Descargar la última  
versión de Node.js

Ejecutar el archivo  
descargado y ejecutar  
los siguientes pasos

# ***Instalación de Node JS***

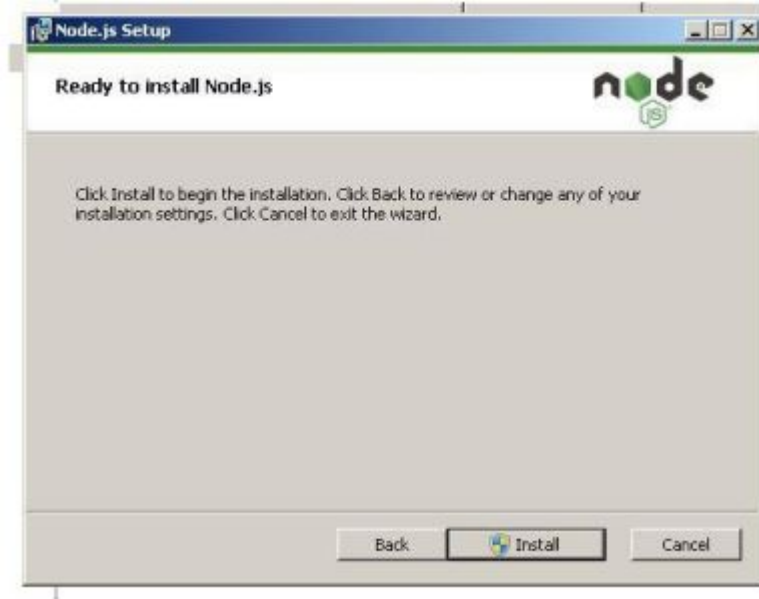


# ***Instalación de Node JS***





# ***Instalación de Node JS***



***¿PREGUNTAS?***



***BREAK***



***CODER HOUSE***

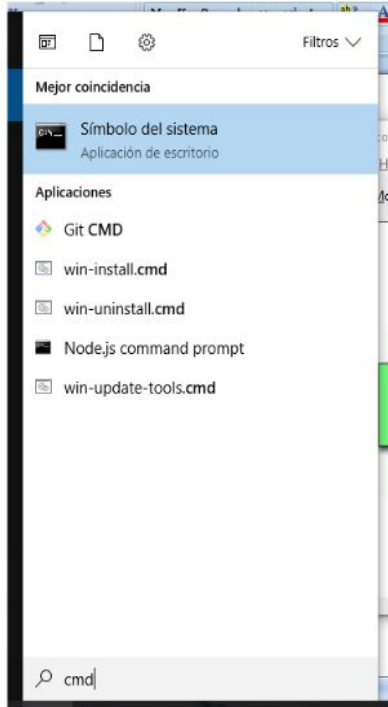
# ***CREAR UNA APLICACIÓN UTILIZANDO EL CLI***

# ***¿Qué es 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).

# ***Abrir la consola***



En Windows



En Linux

***CODER HOUSE***

***VAMOS AL CÓDIGO***



***CODER HOUSE***

## ***Ubicarnos en un directorio específico***

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

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

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



## ***Instalar React JS***

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

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

## ***Crear la aplicación***

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

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

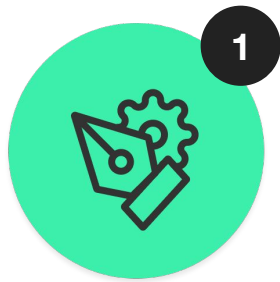
Ahora escribe la dirección obtenida (Ej:localhost:3000) en tu navegador y el resultado obtenido será el siguiente:



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

***¿PREGUNTAS?***





## ***Crear la app utilizando el CLI***

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.

Realizar captura de pantalla del proceso en la consola y del navegador para la carga del desafío (Formato JPG,PNG,JPEG)

*Tiempo: 20 minutos*

***CODER HOUSE***

***¿PREGUNTAS?***





## ***RECURSOS DE LA CLASE***

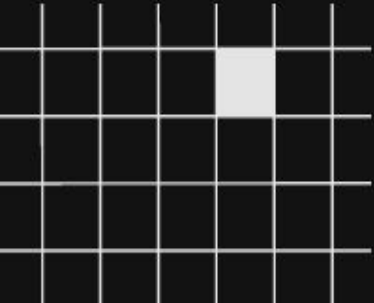
- Amler (2016). ReactJS by Example (1 ed.). EEUU, Packt.
- Stein, J. (2016). ReactJS Cookbook (1 ed.). EEUU, Packt.
- <https://reactjs.org/tutorial/tutorial.html>
- <https://reactjs.org/docs/hello-world.html>
- <https://carlosvillu.com/introduccion-a-reactjs/>





# ***¡MUCHAS GRACIAS!***

Resumen de lo visto en clase hoy:

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



***OPINÁ Y VALORÁ ESTA CLASE***