

***RECUERDA PONER A GRABAR LA  
CLASE***





***¿DUDAS DEL ON-BOARDING?***

**MIRALO AQUI**



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.

# GLOSARIO:

## Clase 1

**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.

**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).

**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.

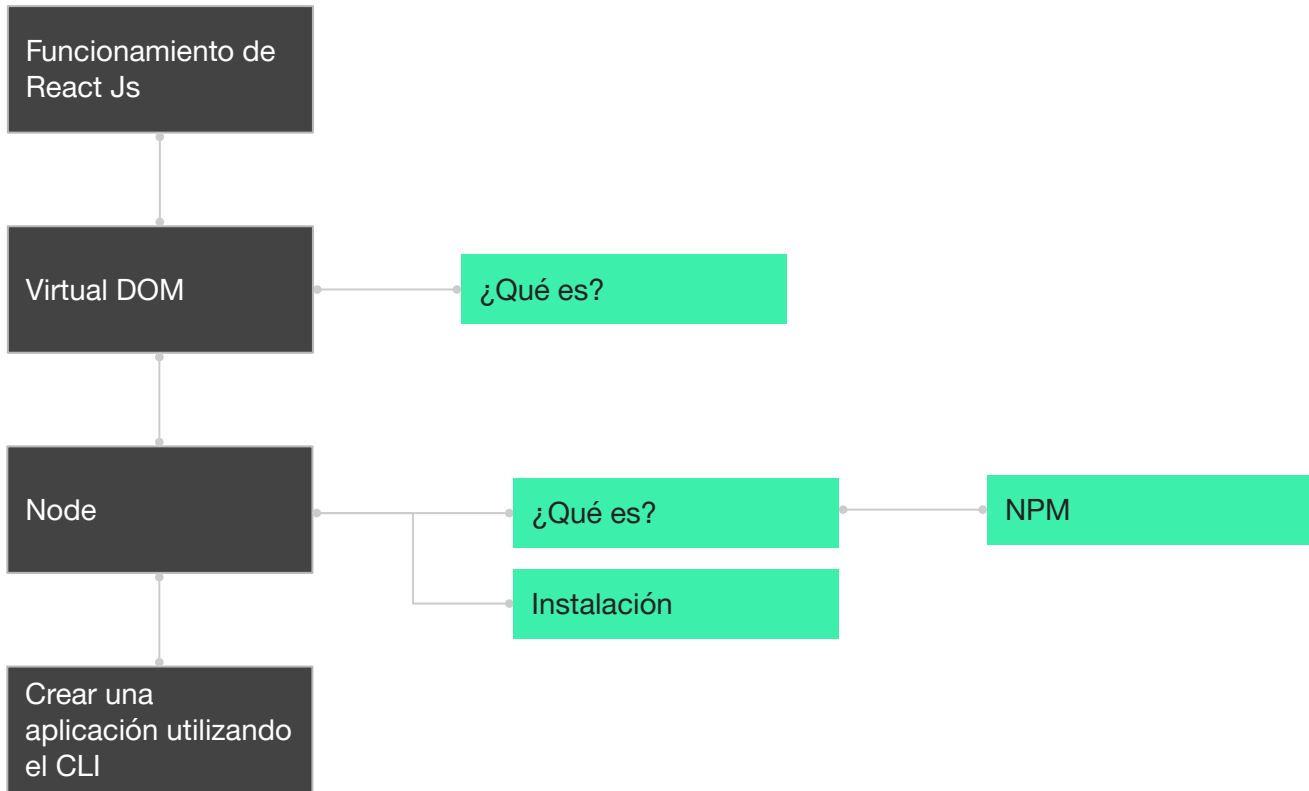
**CSS (cascading style sheets - hojas de estilo en cascada):** es un lenguaje de diseño gráfico con el cual podremos dar estilos (diseño, colores, márgenes) a nuestras webs desarrolladas con HTML.

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

# ***MAPA DE CONCEPTOS***

# MAPA DE CONCEPTOS CLASE 2

¡Para  
recordar!



# ***CRONOGRAMA DEL CURSO***

## Clase 1



### Nivelación



EJEMPLO EN VIVO



FORMULARIO

## Clase 2



### Instalación y configuración del entorno



EJEMPLO EN VIVO



CREAR LA APP UTILIZANDO EL CLI

## Clase 3



### JSX y Transpiling



EJEMPLOS EN VIVO

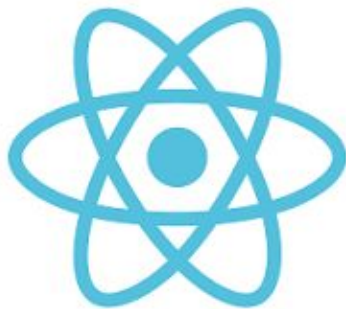


MENÚ E-COMMERCE



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

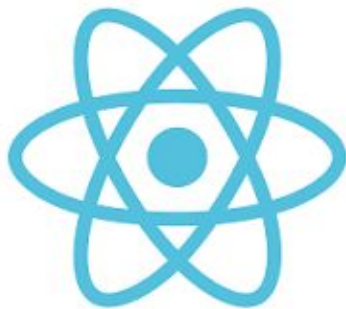
# ***FUNCIONAMIENTO DE REACT JS***



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

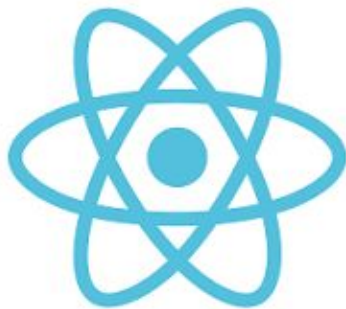
Hablemos de tres conceptos:

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

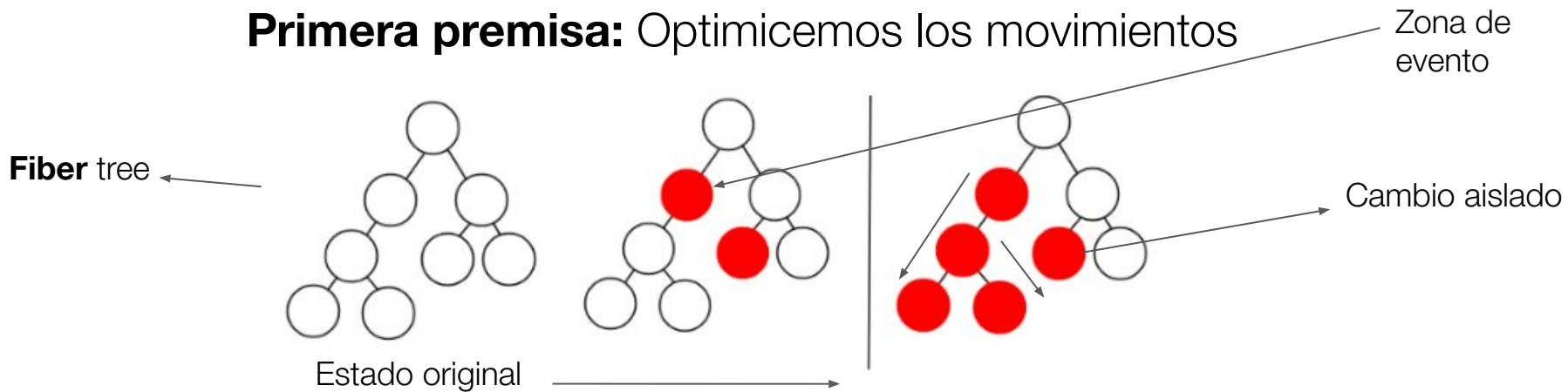


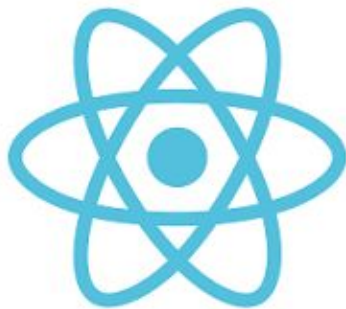
## Primera premisa

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

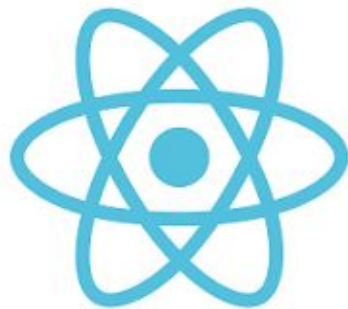


## Primera premisa: Optimicemos los movimientos





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.

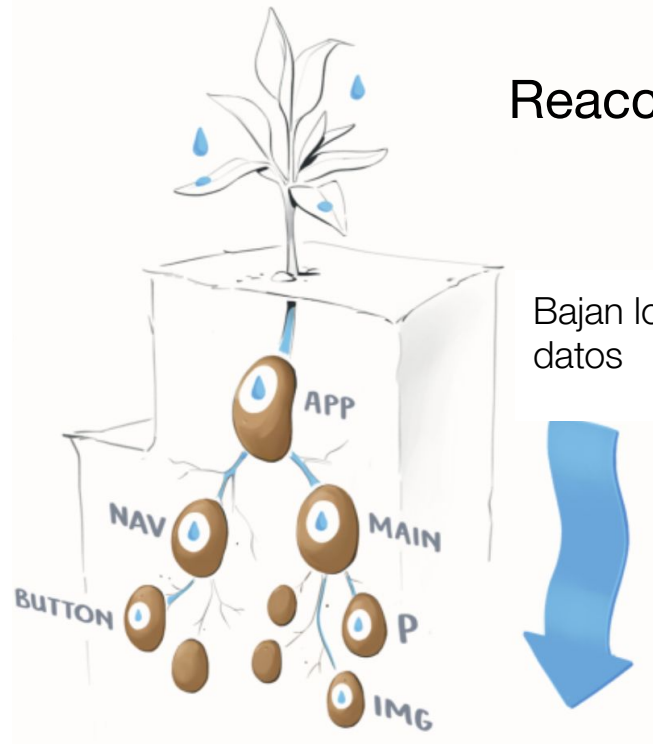


**Segunda premisa:** Flujo de datos unidireccional

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 UI que se mantiene en memoria y en sincronía “reconciliado” con el DOM “real”.**

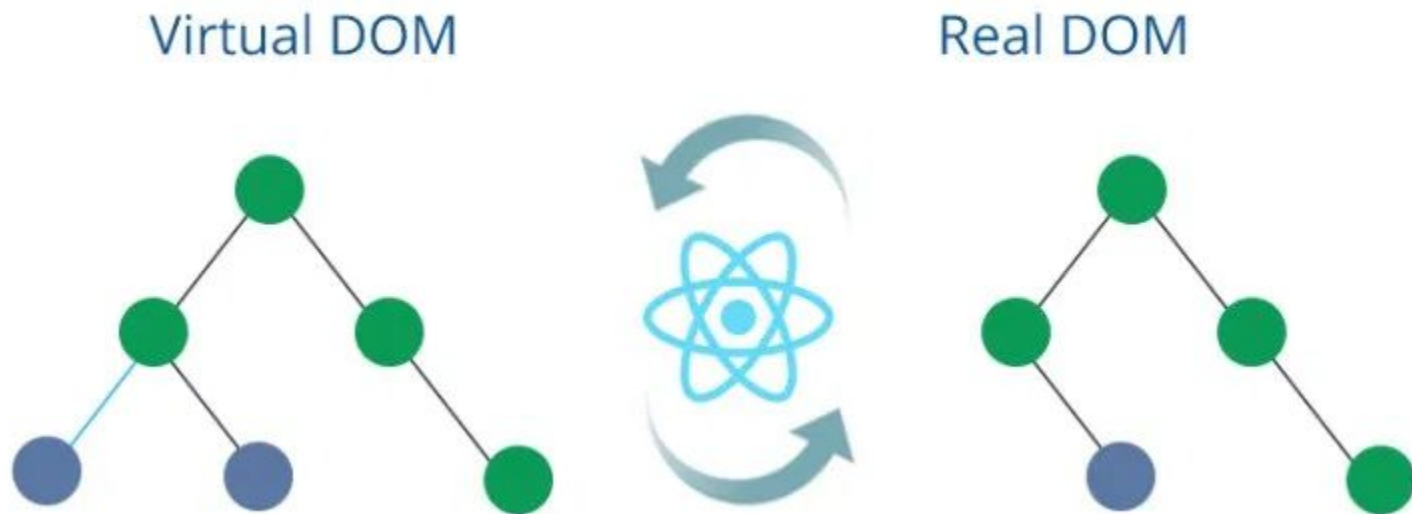
# ***VIRTUAL DOM***

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***



***¿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/>

***CODER HOUSE***



# ***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

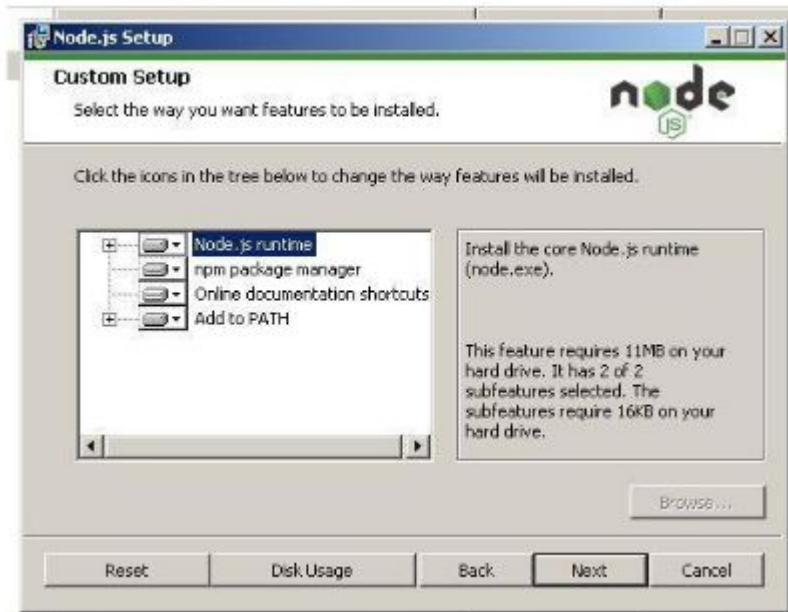
3

Ejecutar el archivo  
descargado y ejecutar  
los siguientes pasos

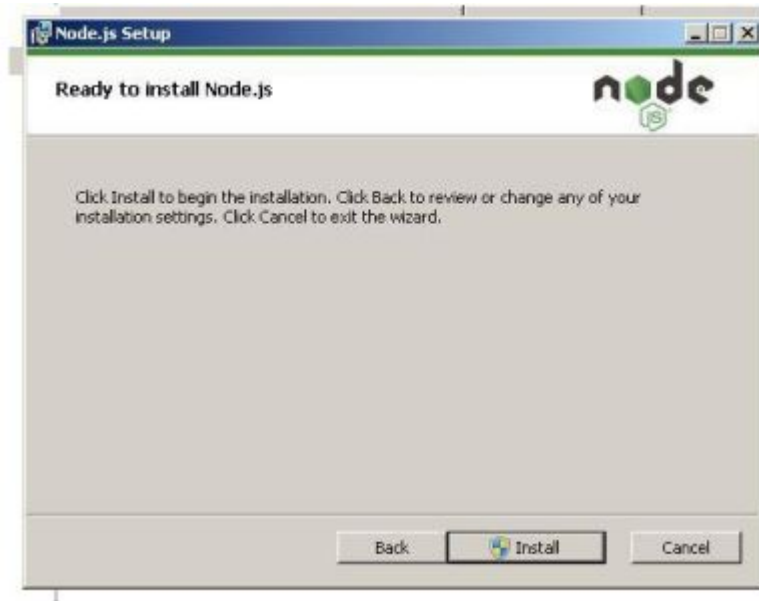
# ***INSTALACIÓN DE NODE JS***



# INSTALACIÓN DE NODE JS



# ***INSTALACIÓN DE NODE JS***





***BREAK***

**¡5/10 MINUTOS Y VOLVEMOS!**



# ***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.

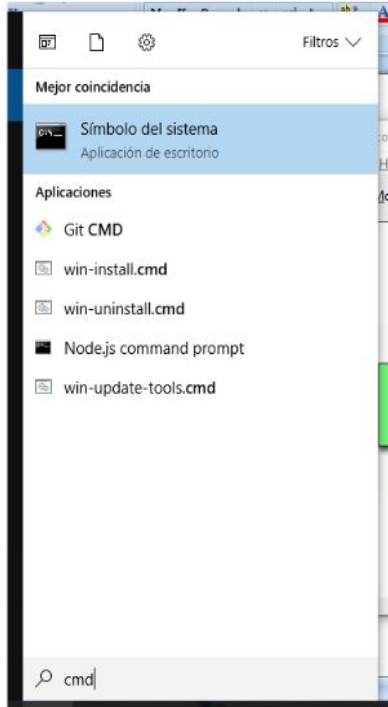
React tiene su propia CLI, pero actualmente solo admite la creación de una aplicación (create-react-app).

***VAMOS AL CÓDIGO***



***CODER HOUSE***

# ABRIR LA CONSOLA



En Windows



En Linux

**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.

Ejemplo  
en vivo



# ***¡VAMOS A PRACTICAR LO VISTO!***

***Spread operator, métodos de array y desestructuración***



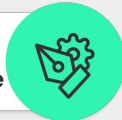
# ***CREAR LA APP UTILIZANDO EL CLI***

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

# CREAR LA APP UTILIZANDO EL CLI

**Formato:** link al repositorio de github con tu app creada con CRA debe tener el nombre "Idea+Apellido".

Desafío  
entregable



**>> Consigna:** 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

# CREAR LA APP UTILIZANDO EL CLI

**Formato:** link al repositorio de github con tu app creada con CRA debe tener el nombre "Idea+Apellido".

Desafío  
entregable



>>Ejemplo:

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?***





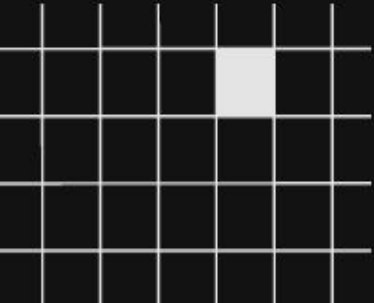
## ***RECURSOS DE LA CLASE***

- 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
- 





***OPINA Y VALORA ESTA CLASE***