



CICLO 4A

[FORMACIÓN POR CICLOS]

Profundización  
Web

REACT JS



# Agenda

- ¿Qué es React.js
- Tecnologías relacionadas
- Servidor de app
- Ejercicios
- Componentes

# ¿Qué es React.js

- Biblioteca UI
- ¿Requerimientos?
- Single page application
- Foco en la lógica del negocio
- Mejor rendimiento
- Virtual DOM
- Renderizado desde el navegador y desde el servidor.
- Customización de etiquetas [Reúso]
- Crear aplicaciones dinámicas
- Buen desempeño (fácil refactorización)
- Componentes reutilizables
- Fácil de aprender y usar

# React vs Angular

- **Similitudes:** open source, constante evolución, generación desde 0, componentes, dependen de js, herramientas de front, buen rendimiento. SPA
- Librería base-biblioteca vs Framework
- Facebook vs Google
- Jsx vs Tx (transpiling code)
- IONIC VS ReactNative
- Abierto vs guiado

# Tecnologías relacionadas

- React native
- React Windows
- Next.js
- GraphQL – apollo
- REDUX
- Flux
- **NodeJs /npm**
- Atom – visual studio code

# jsx

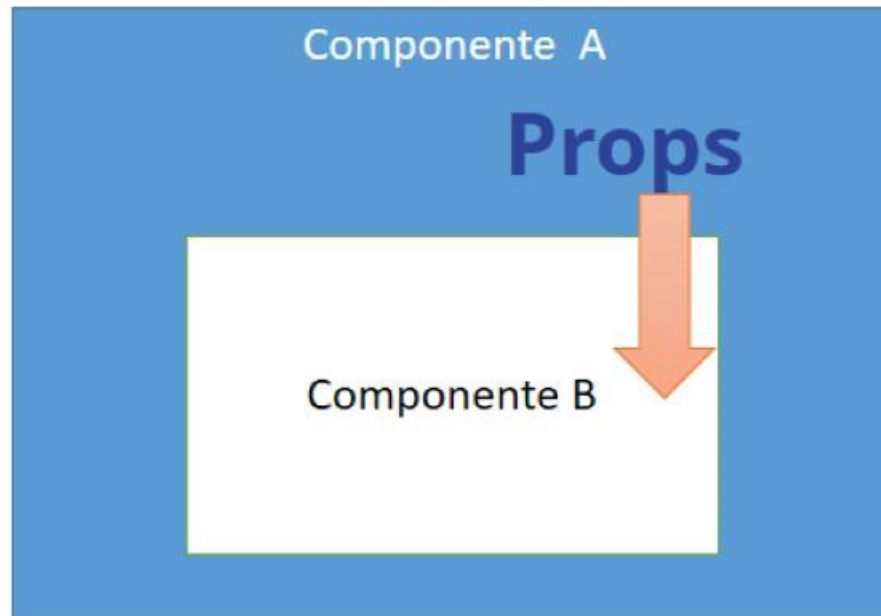
```
const element = <h1>Hello, world!</h1>;
```

- Permite describir en javaScript componentes usando estructura similar e HTML
- Más fácil de visualizar
- Errores y advertencias más útiles
- Es opcional – pero es “RECT”
- Cambios de sintaxis



# Props

- Argumentos/parámetros que puede recibir un componente de react.
- Props =objetos y sus propiedades.
- Por lo general enviados de padre a hijo



# Componentes

- **Funcionales**

- Debe retornar un elemento de React (JSX)
- Debe comenzar con una letra mayúscula
- Puede recibir (parámetros) valores si es necesario

```
function Saludo(props){  
  return <h1>Hola, {props.nombre}!</h1>;  
}
```

- **Clase**

- Clase de ES6 que retorna JSX
- Extiende de React Component
- Debe tener un método render
- Puede recibir valores si es necesario

```
class Saludo extends React.Component {  
  render(){  
    return <h1>Hola,  
    {this.props.nombre}!</h1>;  
  }  
}
```



# ¿Qué son los Componentes?

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';

function Componente({titulo, children}){
  return( <><h1>{titulo}</h1>
    <div> {children}</div></>);
}

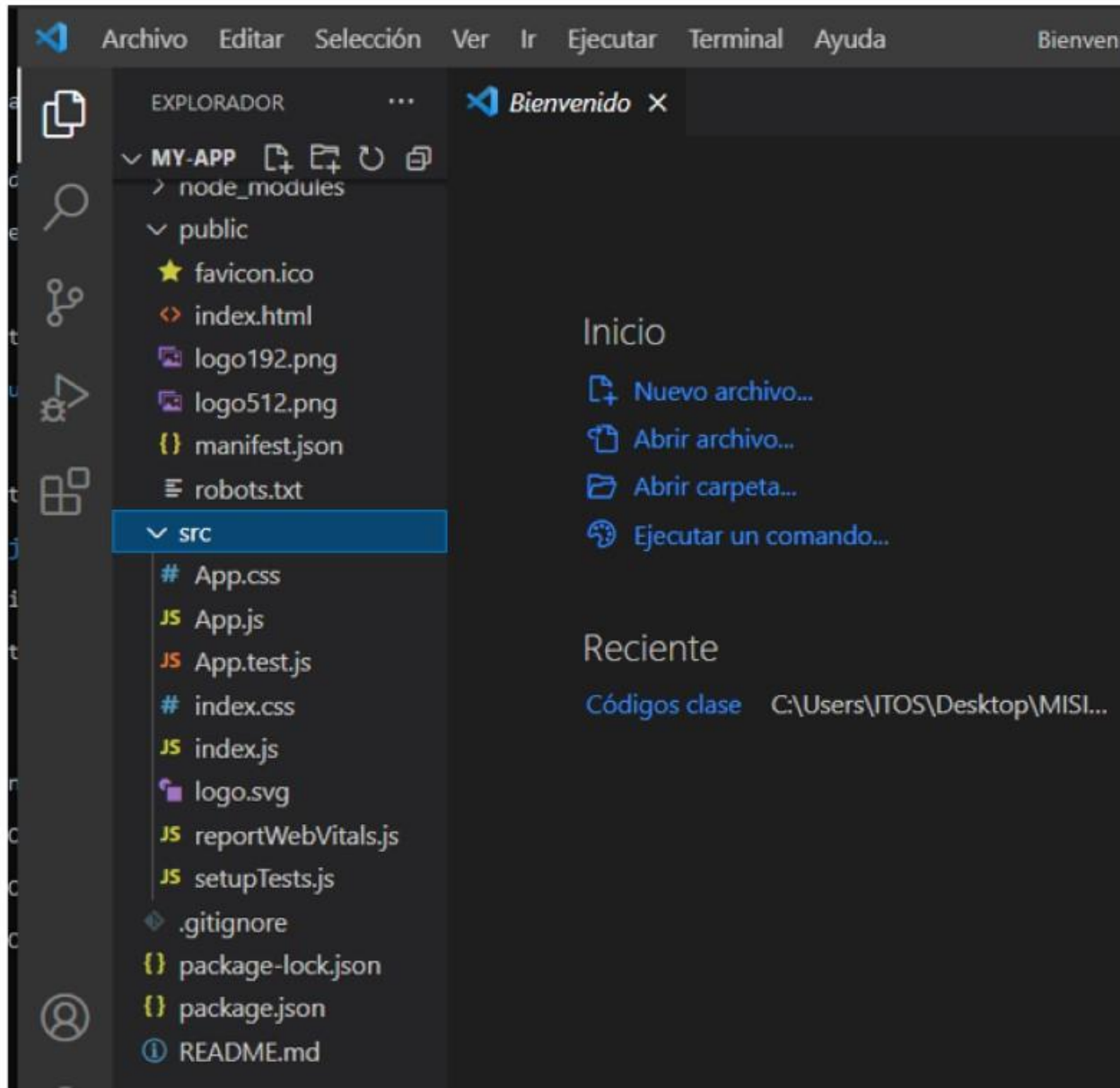
ReactDOM.render(
  <Componente titulo= "Titulo 1">Hola desde mi propio componente</Componente>,
  document.getElementById('root')
);
```

# ELEMENTOS

- Las unidades más pequeñas
- Define lo que se ve en pantalla
- Los componentes están hechos de elementos.

`<>xxx</>`

Componentes en mayúsculas, elementos en minúscula



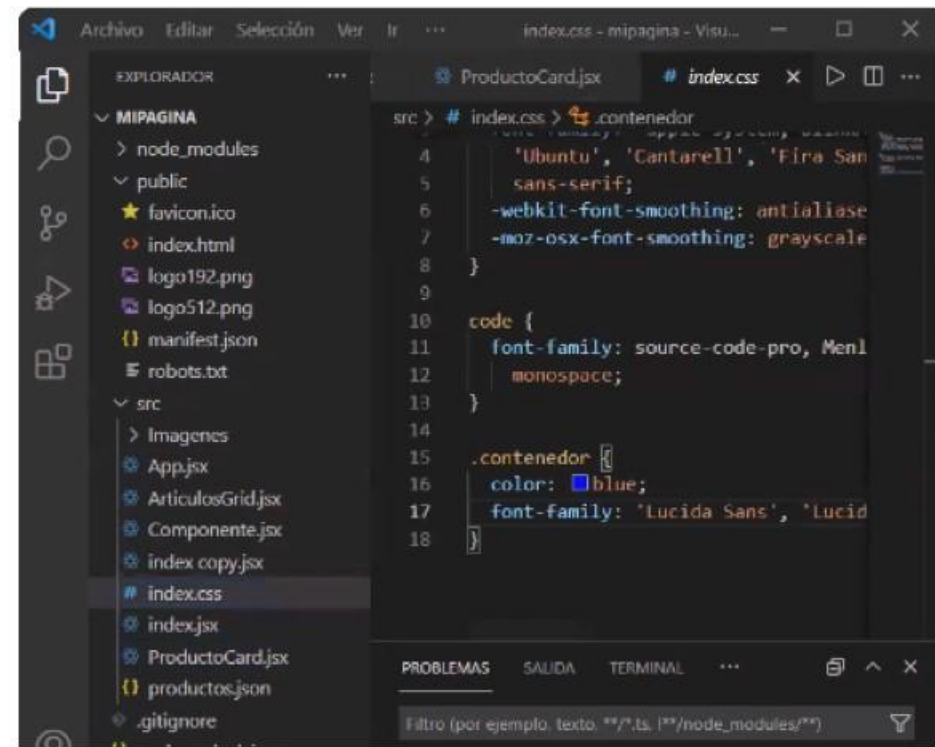
# Servidor de app

- Create-react-app myfirstapp
- npm start

# Estilos para los componentes

```
src > App.jsx > App
1 export function App() {
2   return (
3     <<div className="contenedor">Hola mundo desde un componente
4       <header>
5         <h1> Esta es mi tienda en linea</h1>
6       </header>
7       <main>
8         <ul>
9           <li>Artículo 1 de la tienda</li>
10          <li>Artículo 2 de la tienda</li>

```



```
src > ProductoCard.jsx > ProductoCard
1 import styles from "../ProductoCard.module.css";
2 export function ProductoCard({producto}) //paso el valor al compo
3 {
4   const imageURL = "https://raw.githubusercontent.com/Scalapp/MiTienda/refs/heads/main/public/logo192.png";
5   return (
6     <li className={styles.productCard}>
7       <img src={imageURL} alt={producto.nombre} />
8       {producto.nombre}
9     </li> //no se pasa por acá el key key={producto.id}
10   );
11 }
```

```
.productGrid{
  display: grid;
  grid-template-columns: repeat(auto-fill, 230px);
  gap: 40px;
}
```

# Estados

- Representación en JavaScript del conjunto de propiedades de un componente y sus valores actuales.
- Mayormente usado en los componentes de clase.
- **Anteriores a 16.8**



# ¿Qué son los hooks?

- Proveen un estado
- Mantener variables dentro de componentes

```
JS SimpleState.js | SimpleState.jsx | App.jsx
src > SimpleState.jsx > SimpleState > carrito
1 import {useState} from "react";
2
3 export function SimpleState(){
4   const [carrito, setCarrito] =useState(0);
5   return (
6     <div>
7       Su compra es: {carrito}
8       <button onClick={()=>setCarrito(carrito+1)} >
9         Imagen de producto
10      </button>
11    </div>
12  );
13 }
```

```
src > App.jsx > ...
1 //import { ArticulosGrid } from "../ArticulosGrid";
2 import SimpleState from './SimpleState'
3 export function App() {
4   return (
5     <>
6       <header>
7         <h1> My Tienda en linea</h1>
8       </header>
9       <main>
10        <SimpleState/>
11      </main>
12    </>
13  );
14 }
15
16 }
```

```
JS SimpleState.js | SimpleState > render
1 import React, { Component } from "react";
2
3 export default class SimpleState extends Component {
4   state = {
5     carrito: 0
6   };
7
8   render() {
9     return (
10      <div>
11        Su compra es: {this.state.carrito}
12        <button onClick={()=>this.setState({carrito:this.state.carrito+1})} >
13          Imagen de producto
14        </button>
15      </div>
16    );
17  }
18 }
```



# Event listener

- El manejo de eventos con elementos React es muy similar al manejo de eventos en elementos DOM. Hay algunas diferencias de sintaxis:
- Los eventos de React se nombran usando camelCase, en lugar de minúsculas.
- Con JSX, pasa una función como controlador de eventos, en lugar de una string.

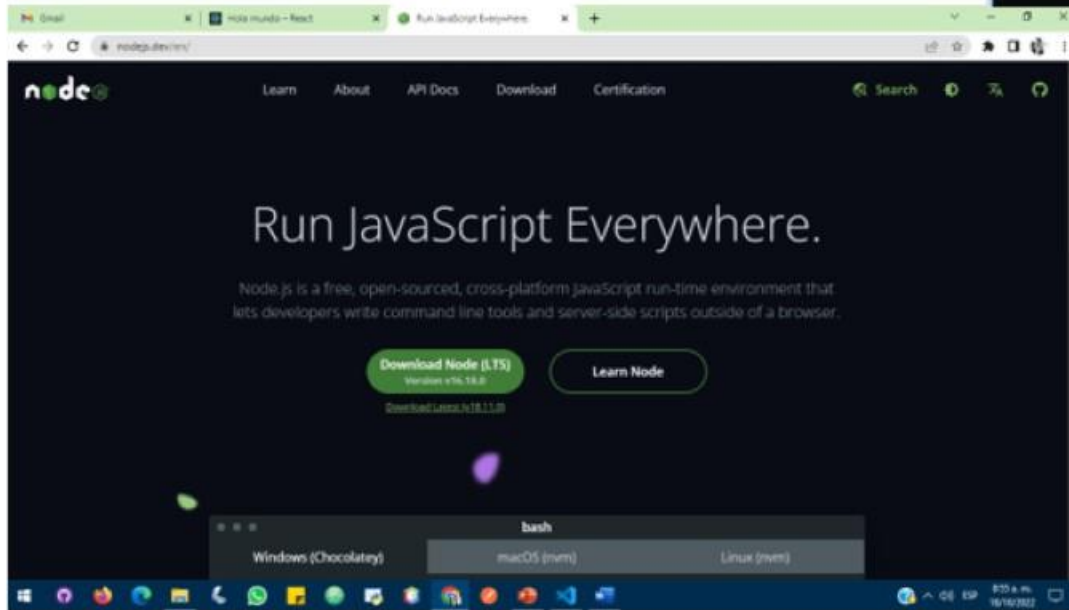
• `<button onclick="activateLasers()"> Activate Lasers </button>`

es ligeramente diferente en React:

`<button onClick={activateLasers}> Activate Lasers </button>`

# Ejercicios

- Create react app
- Descargar proyecto de ejemplo



```
Símbolo del sistema

Microsoft Windows [Versión 10.0.19042.2130]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ITOS>node -v
"node" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\ITOS>
```

```
npx create-react-app my-app
cd my-app
npm start
```

# Referencias

- <https://reactjs.org/>
- <https://reactjs.org/tutorial/tutorial.html>
- <https://git-scm.com/>
- <https://github.com/>
- <https://git-scm.com/doc>
- <https://education.github.com/git-cheat-sheet-education.pdf>
- <https://rogerdudler.github.io/git-guide/index.es.html>



Gracias