

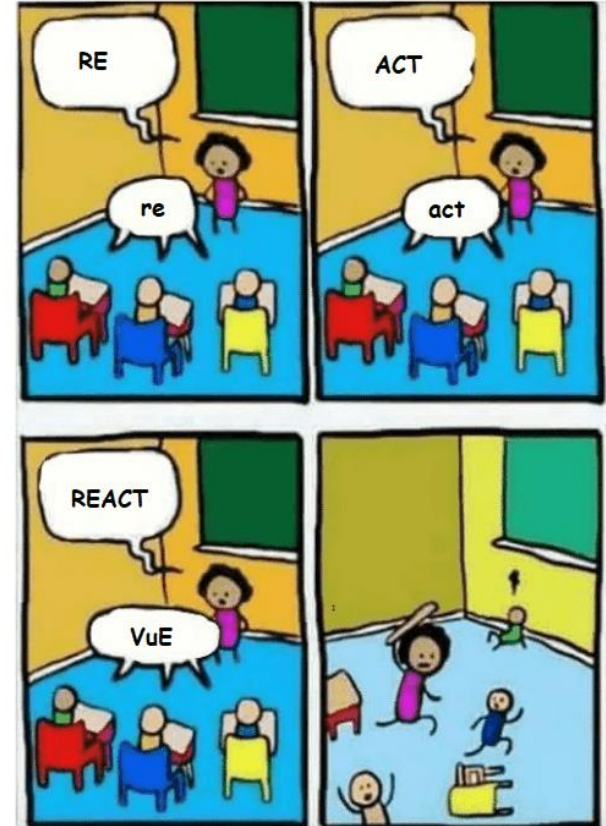


# React

## Introducción a ReactJS

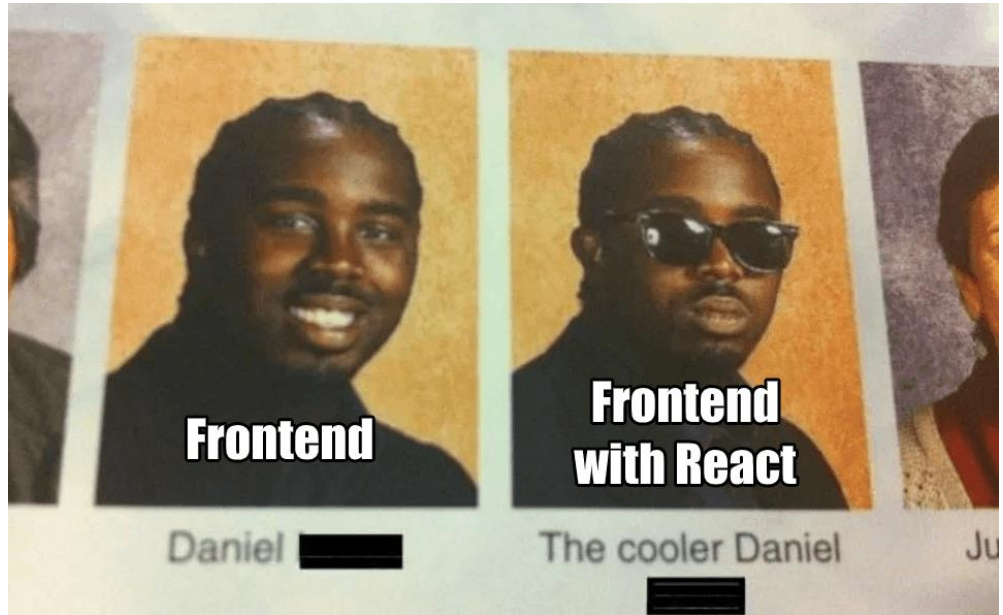
# Introducción a ReactJS

- Introducción a ReactJS
- Primeros pasos
- React ES6
- React render HTML
- Componentes en ReactJS
- JSX
- Ejercicio práctico
  - Creación de un componente simple



# Introducción

¿ Qué es React?



# Introducción

## ¿Qué es React?

- React, a veces denominado ~~framework~~ de JavaScript frontend, es una **biblioteca** de JavaScript creada por Facebook.
- React es una herramienta para **construir componentes** de interfaz de usuario.
- React es una biblioteca de JavaScript para crear **interfaces de usuario**.
- React se utiliza para crear **aplicaciones de una sola página**.
- React nos permite crear componentes de interfaz de usuario **reutilizables**.

# Introducción

- La versión actual de React.JS es V18.0.0 (abril de 2022).
- El **lanzamiento** inicial al público (V0.3.0) fue en julio de 2013.
- React.JS **se utilizó por primera vez** en 2011 para la función Newsfeed de Facebook.
- El ingeniero de software de Facebook, Jordan Walke, lo creó.
- La versión actual de **create-react-app** es v5.0.1 (abril de 2022).
- **create-react-app** incluye herramientas integradas como webpack, Babel y ESLint.

# Introducción

- ¿Cómo funciona React?
- React crea un **DOM VIRTUAL** en la memoria.
- En lugar de manipular el DOM del navegador directamente, React crea un DOM virtual en la memoria, donde realiza toda la manipulación necesaria antes de realizar los cambios en el DOM del navegador.
- ¡React solo cambia lo que necesita ser cambiado!
- React descubre qué cambios se han realizado y cambia solo lo que debe cambiarse.

# React

Primeros pasos

Entornos

# Primeros pasos

- React test. Directamente en HTML
- Entorno React



# Primeros pasos

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>
    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js" crossorigin></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  </head>
  <body>

    <div id="mydiv"></div>

    <script type="text/babel">
      function Hello() {
        return <h1>Hello World!</h1>;
      }

      const container = document.getElementById('mydiv');
      const root = ReactDOM.createRoot(container);
      root.render(<Hello />)
    </script>

  </body>
</html>
```

## Directamente en HTML

Esta forma de usar React puede estar bien para fines de prueba, pero para la producción deberá configurar un entorno React .

# Primeros pasos

## Configuración de un entorno de react

Requerimientos:

- npx

`npx -v`

- Node.js

`npm -v`

¿npx es lo mismo que npm?

# Primeros pasos

Ejecute este comando para crear una aplicación React llamada my-app:

```
npx create-react-app my-app
```

El create-react-app configurará todo lo que necesita para ejecutar una aplicación React.

# Primeros pasos

Ejecute este comando desde el terminal para moverse al directorio my-app:

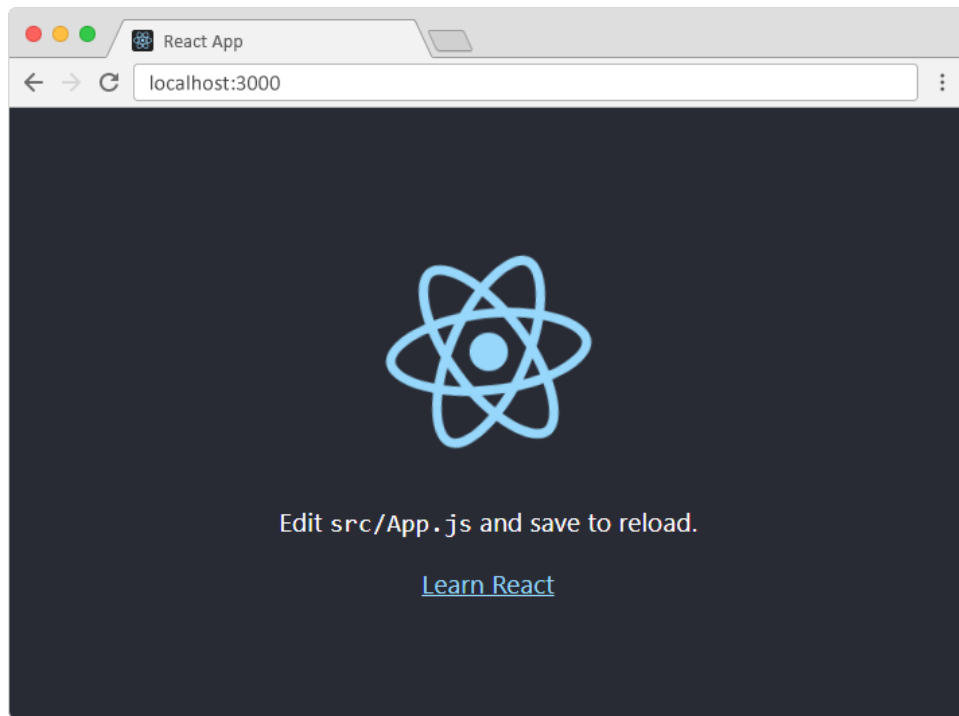
```
cd my-app
```

Ejecute este comando desde el terminal para ejecutar la aplicación React my-app:

```
npm start
```

Aparecerá una nueva ventana del navegador con su aplicación React recién creada! Si no, abra su navegador y escriba localhost:3000 en la barra de direcciones.

# Primeros pasos



# Primeros pasos

## Modificar la aplicación React

Busque en el directorio **my-app** y encontrará una carpeta **src**.

Dentro de la carpeta src hay un archivo llamado App.js,

/miApp/src/App.js:

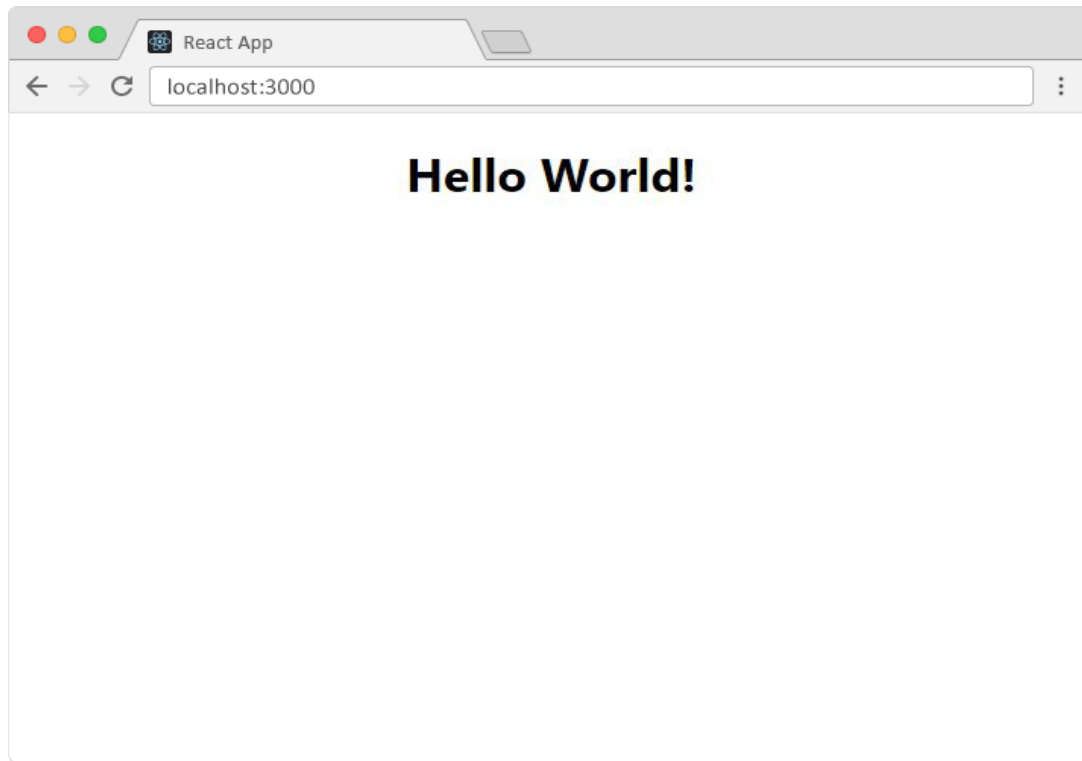
```
function App() {  
  return (  
    <div className="App">  
      <h1>Hello World!</h1>  
    </div>  
  );  
}
```

```
export default App;
```

\*Eliminado las importaciones que no necesitamos (logo.svg y App.css).

Los cambios son visibles inmediatamente después de guardar el archivo. ¡no es necesario que vuelva a cargar el navegador!

# Primeros pasos



# Primeros pasos

- carpeta src para que solo contenga un archivo: index.js
- eliminar las líneas de código innecesarias dentro del index.js

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myFirstElement = <h1>Hello React!</h1>

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myFirstElement);

/*
You are now watching
the React file 'index.js'
through our 'Show React' tool.
*/
```



# React

## Actualizar React

# Actualizar React

## Paso 1: Instalar React 18

Para instalar la última versión, desde la carpeta de su proyecto, ejecute lo siguiente desde la terminal:

```
npm i react@latest react-dom@latest
```

# Actualizar React

## Paso 2: use la nueva API raíz

Para aprovechar las funciones simultáneas de React 18, deberá usar la nueva API raíz para la representación del cliente.

# Actualizar React

## // Before

```
import ReactDOM from 'react-dom';  
ReactDOM.render(<App />, document.getElementById('root'));
```

## // After

```
import ReactDOM from 'react-dom/client';  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(<App />);
```

Su aplicación funcionará sin usar la nueva API raíz. Si continúa usando, ReactDOM.render su aplicación se comportará como React 17.

# React

## React ES6

# React ES6

## ¿Qué es ES6?

- ES6 significa ECMAScript 6.
- ECMAScript se creó para estandarizar JavaScript
- ES6 es la sexta versión de ECMAScript (2015)

# React ES6

## ¿Por qué debo aprender ES6?

React usa ES6, y debería estar familiarizado con algunas de las nuevas características como:

- Clases
- Funciones de flecha
- Variables (let, const, var)
- Métodos de Array como .map()
- Destructuración
- Módulos
- Operador Ternario
- Operador de propagación

**¿ React ES6 es lo mismo ReactJS?**

# React

## React render HTML



# React render HTML

- objetivo de React -> representar HTML en una página web
- React representa HTML en la página web mediante el uso de una función llamada **createRoot()** y su método **render()**.

# React

## React render HTML



# React render HTML

## Function createRoot

- La función createRoot() toma un argumento, un elemento HTML.
- El propósito de la función es definir el elemento HTML donde se debe mostrar un componente React.

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myFirstElement = <h1>Hello React!</h1>

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myFirstElement);

/*
You are now watching
the React file 'index.js'
through our 'Show React' tool.
*/
```

# React render HTML

## El método de render

- render() para definir el componente React que se debe representare en el único DIV del fichero index.html que se encuentra en la carpeta public.

Tenga en cuenta que la identificación del elemento no tiene que llamarse "root", pero esta es la convención estándar.

### Ejemplo

```
//Mostrar un párrafo dentro de un elemento con el id de "root":
const container = document.getElementById('root');
const root = ReactDOM.createRoot(container);
root.render(<p>Hello</p>);

//El resultado se muestra en el elemento <div id="root">:
<body>
  <div id="root"></div>
</body>
```

# React render HTML

## Ejemplo

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myelement = (
  <table>
    <tr>
      <th>Name</th>
    </tr>
    <tr>
      <td>John</td>
    </tr>
    <tr>
      <td>Elsa</td>
    </tr>
  </table>
);

const container = document.getElementById('root');
const root = ReactDOM.createRoot(container);
root.render(myelement);
```

# React render HTML

## El nodo root

- Es el elemento HTML en el que desea mostrar el resultado.
- Es como un contenedor de contenido administrado por React.
- NO tiene que ser un elemento `<div>` y NO tiene que tener `id='root'`:
- El nodo raíz se puede llamar como quieras

```
<body>  
  <header id="sandy"></header>  
</body>
```

Mostrar el resultado en el elemento `<header id="sandy">`:

```
const container = document.getElementById('sandy');  
const root = ReactDOM.createRoot(container);  
root.render(<p>Hallo</p>);
```

# React

## React JSX

# React JSX

## ¿Qué es JSX?

- JSX significa JavaScript XML.
- JSX nos permite escribir HTML en React.
- JSX facilita escribir y agregar HTML en React.

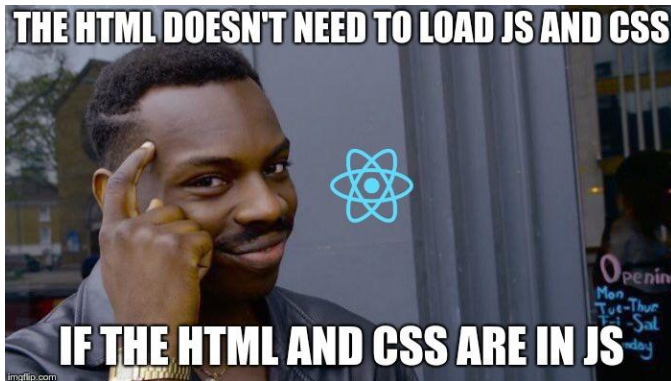




# React JSX

## Codificación JSX

- JSX nos permite escribir elementos HTML en JavaScript y colocarlos en el DOM sin ningún método `createElement()` y/o `appendChild()`.
- JSX convierte etiquetas HTML en elementos de react.
- No es necesario que use JSX, pero JSX facilita la escritura de aplicaciones React.



# React JSX

## JSX

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
  
const myElement = <h1>I Love JSX!</h1>;  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(myElement);
```

## Sin JSX

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
  
const myElement = React.createElement('h1', {}, 'I do not use JSX!');  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(myElement);
```

JSX es una extensión del lenguaje JavaScript basado en ES6 y se traduce a JavaScript normal en tiempo de ejecución.



# React JSX

## Expresiones en JSX

Con JSX puedes escribir expresiones dentro de llaves { }.

- **Ejemplo**

Ejecutar la expresión 5 + 5:

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = <h1>React is {5 + 5} times better with JSX</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

# React JSX



# React JSX

## Insertar un bloque grande de HTML

Para escribir HTML en varias líneas, coloque el HTML entre paréntesis:

### Ejemplo

Cree una lista con tres elementos de lista:

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = (
  <ul>
    <li>Apples</li>
    <li>Bananas</li>
    <li>Cherries</li>
  </ul>
);

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

# React JSX



# React JSX

## Un elemento de nivel superior

El código HTML debe estar envuelto en UN elemento de nivel superior.

## Ejemplo

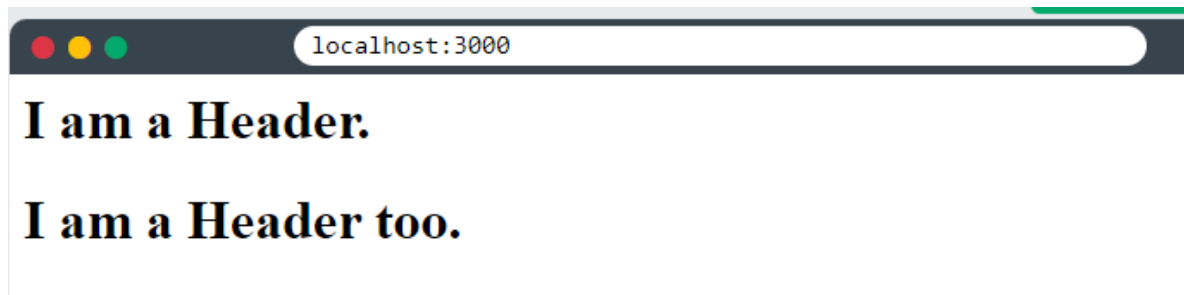
Envuelva dos párrafos dentro de un elemento DIV:

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = (
  <div>
    <h1>I am a Header.</h1>
    <h1>I am a Header too.</h1>
  </div>
);

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

# React JSX





# React JSX

## Un elemento de nivel superior

Puede usar `<></>` para envolver varias líneas.

Esto evitará agregar innecesariamente nodos adicionales al DOM.

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = (
  <>
    <p>I am a paragraph.</p>
    <p>I am a paragraph too.</p>
  </>
);

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

# React JSX

## Los elementos deben estar cerrados

- JSX sigue las reglas XML y, por lo tanto, los elementos HTML deben cerrarse correctamente.

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
const myElement = <input type="text" />;  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(myElement);
```

# React JSX

## Attribute class = className

- Utilice el atributo **className** en lugar **class**.

## Ejemplo

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = <h1 className="myclass">Hello World</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

# React JSX

## Condiciones - sentencias if

React admite declaraciones if, pero no dentro de JSX.

**Opción 1:** Escriba declaraciones if fuera del código JSX:

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const x = 5;
let text = "Goodbye";
if (x < 10) {
  text = "Hello";
}

const myElement = <h1>{text}</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

Escribe "Hola" si x es menor a 10, de lo contrario "Adiós":

# React JSX

## Condiciones - sentencias if

React admite declaraciones if, pero no dentro de JSX.

**Opción 2:** Utilice expresiones ternarias en su lugar

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const x = 5;

const myElement = <h1>{(x) < 10 ? "Hello" : "Goodbye"}</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

Tenga en cuenta que para incrustar una expresión de JavaScript dentro de JSX, el JavaScript debe estar envuelto con llaves, {}.

# React

## Componentes de React

# Componentes de React

Los componentes son como funciones que devuelven elementos HTML.

- Códig independientes y reutilizables.
- Mismo propósito que las funciones de JavaScript, pero trabajar de forma aislada y devolver HTML.
- Dos tipos, componentes de clase y componentes de función.
- Al crear un componente React, el nombre del componente DEBE comenzar con un letra mayúscula.

# Componentes de React

## Renderizando de un componente

- Tienes un componente llamado Car, que devuelve un elemento.<h2>
- Utilice una sintaxis similar a la HTML normal: <Car />

**Ejemplo** Muestre el componente en el elemento "raíz":Car

```
import React from 'react';
import ReactDOM from 'react-dom/client';

function Car() {
  return <h2>Hi, I am a Car!</h2>;
}

const root =
ReactDOM.createRoot(document.getElementById('root'));
root.render(<Car />);
```



# Componentes de React

## Componente de función

```
class Car extends React.Component {  
  render() {  
    return <h2>Hi, I am a Car!</h2>;  
  }  
}
```

# React

## Ejercicio

# Ejercicio\_1

## Enunciado del ejercicio:

- Crea una aplicación web utilizando React que consista en una plantilla con un encabezado (header), contenido principal (main) y pie de página (footer). Los componentes ya están proporcionados: Header.js, Main.js y Footer.js. Tu tarea es implementar la estructura de la aplicación y combinar los componentes para mostrar la plantilla completa en el navegador.

# Ejercicio\_1

## Encabezado de la Aplicación

### Contenido Principal

Este es el contenido principal de la aplicación.

Pie de Página de la Aplicación

# React

## Test

# Test

[https://quizizz.com/admin/quiz/64aafa900406d9001d20f6cb?source=quiz\\_share](https://quizizz.com/admin/quiz/64aafa900406d9001d20f6cb?source=quiz_share)