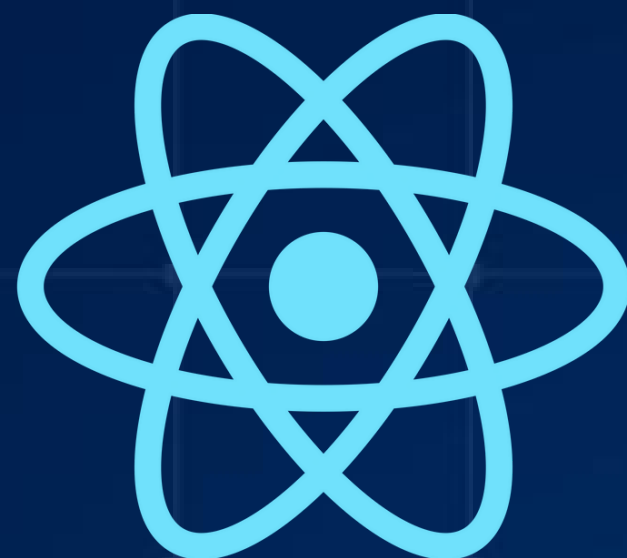


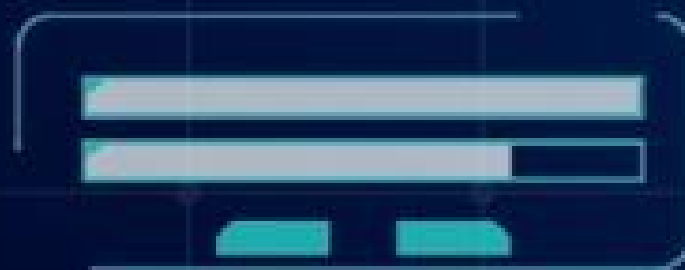


Escuela
Nacional de
Estudios
Superiores

IECAGto[®]
Instituto Estatal de Capacitación



DESARROLLO WEB CON REACT



Temario

Fundamentos de desarrollo web

Javascript

Aplicaciones con React

Hooks y navegabilidad

Consumo de Web APIs

Herramientas en la nube

Sitios estáticos y SSR

Aplicaciones en tiempo real



Hooks y navegabilidad

React Router

React Router es una herramienta que habilita la navegabilidad del lado del cliente, permitiendo crear sitios de "múltiples páginas" a través de su navegación por URL.

React Router permite generar navegación a través de links, vincular componentes de React a vistas específicas y generar navegación anidada.



Instalación de React Router

En una app creada con **Create React App**, para añadir la biblioteca de React Router basta con ejecutar el comando:

`npm install react-router-dom`

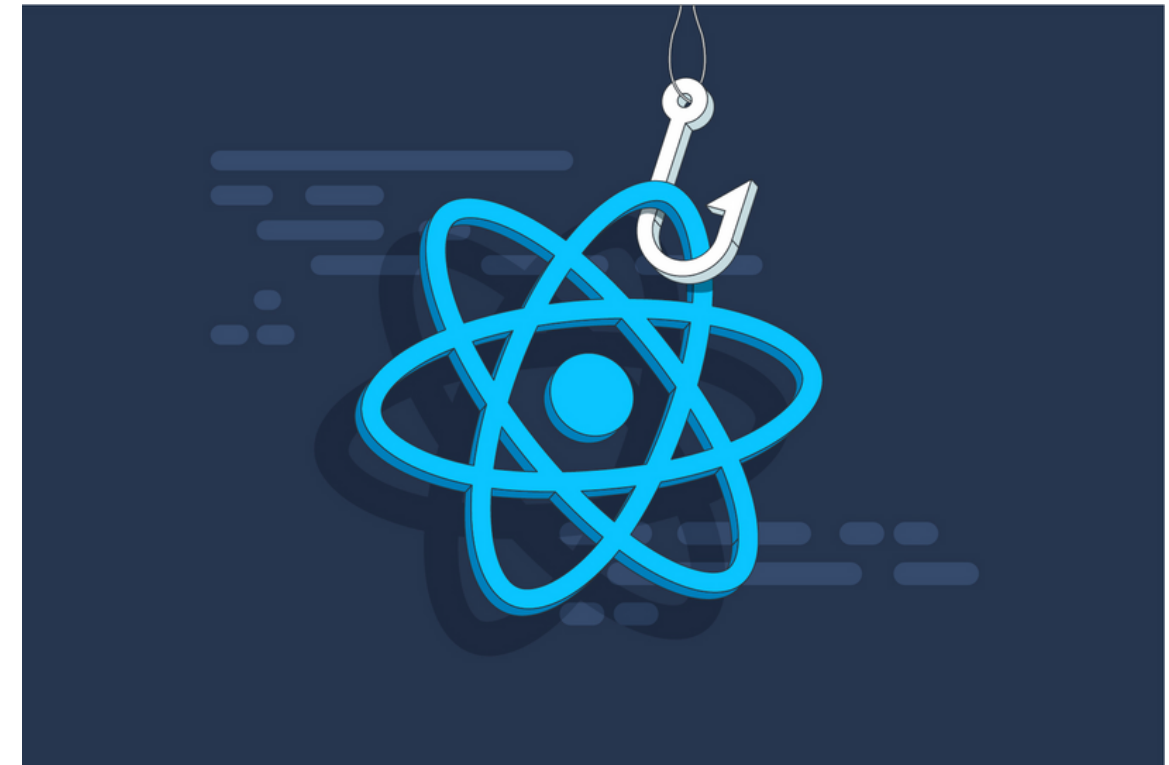
```
const App = () => (  
  <Router>  
    <div>  
      <nav>  
        <ul>  
          <li>  
            <Link to="/">Inicio</Link>  
          </li>  
          <li>  
            <Link to="/about">Acerca de</Link>  
          </li>  
        </ul>  
      </nav>  
  
      <Route path="/" exact component={Home} />  
      <Route path="/about" component={About} />  
    </div>  
  </Router>  
)
```

Hooks

Los **hooks** son elementos reutilizables de código que incluyen lógica no incorporada en el árbol de componentes. Permiten **enganchar (hook up)** funcionalidad a nuestros componentes.

React contiene múltiples hooks predefinidos que se pueden utilizar directamente en los componentes con solamente importarlos.

Generalmente los hooks se definen con el prefijo **use**, por ejemplo: **useEffect**, **useState**, **useCallback**, etc.



useRef

```
import React, { useRef } from 'react';

const ExampleComponent = () => {
  const inputRef = useRef(null);

  const handleClick = () => {
    inputRef.current.focus();
  };

  return (
    <div>
      <input type="text" ref={inputRef} />
      <button onClick={handleClick}>Enfocar</button>
    </div>
  );
};

export default ExampleComponent;
```

useRef es un hook que permite referenciar elementos del DOM directamente mediante una **referencia (ref)**. En React, una referencia es un objeto que guarda valores durante todo el ciclo de vida del componente.

Un ejemplo sobre el uso de refs es para la referencia directa a componentes de un formulario.

Componentes controlados

En un componente controlado los valores de un formulario son manejados por React y no por el DOM. Requieren que no se haga uso de referencias.

Esto ayuda a facilitar la implementación de validaciones y mostrar errores.

```
import React, { useState } from 'react';

const ControlledComponent = () => {
  const [inputValue, setInputValue] = useState('');

  const handleChange = (event) => {
    setInputValue(event.target.value);
  };

  return (
    <div>
      <input type="text" value={inputValue} onChange={handleChange} />
      <p>El valor ingresado es: {inputValue}</p>
    </div>
  );
};

export default ControlledComponent;
```


Custom Hooks

Además de los hooks proporcionados por React, los hooks pueden ser desarrollados con las siguientes características:

- Acceso a otros hooks.
- Gestión de estado local (useState)
- Efectos secundarios (useEffect)

```
import React from 'react';
import useCustomHook from './useCustomHook';

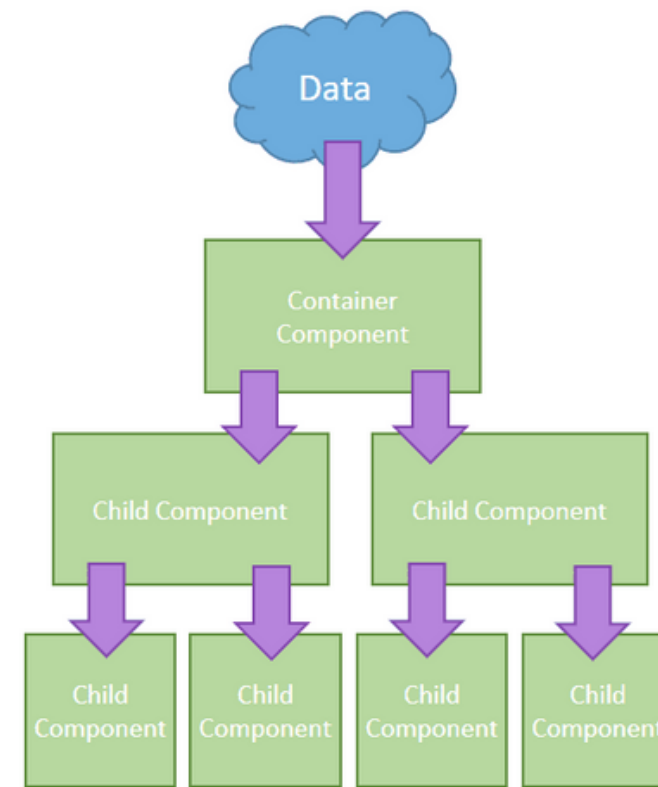
const CustomHookComponent = () => {
  const { value, increment, decrement } = useCustomHook(0);

  return (
    <div>
      <p>Valor: {value}</p>
      <button onClick={increment}>Incrementar</button>
      <button onClick={decrement}>Decrementar</button>
    </div>
  );
};

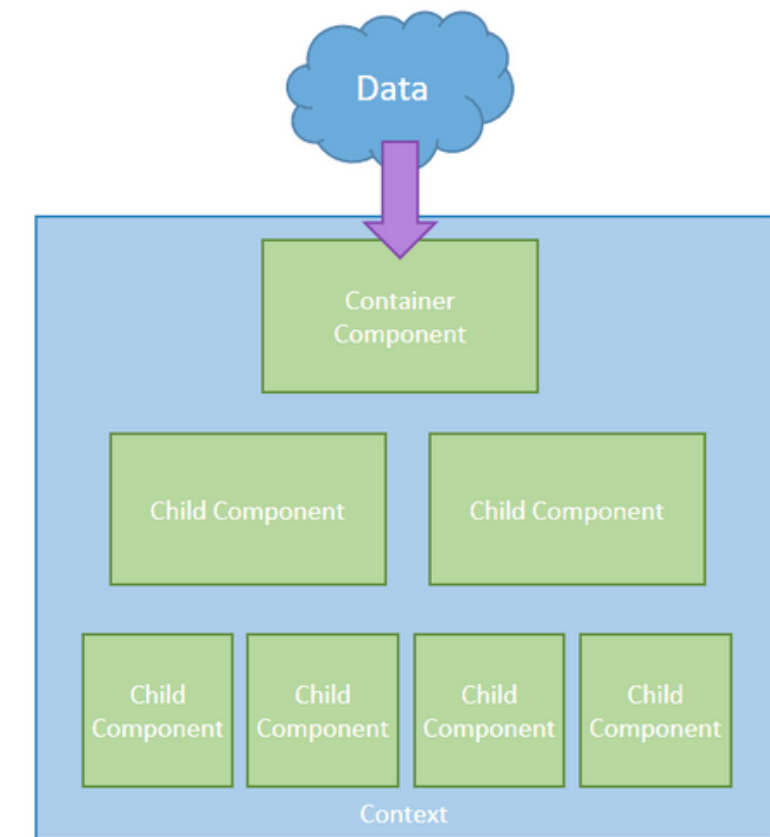
export default CustomHookComponent;
```

React Context

React Context es una característica de React que proporciona una forma de pasar datos a través de la jerarquía de componentes sin tener que pasar explícitamente las props a través de cada nivel de componente.



prop drilling



context API

React Context

El contexto de React está compuesto por tres partes principales:

1. **Proveedor (Provider):** Es un componente de React que proporciona los datos que se compartirán a través del contexto. El Proveedor envuelve a los componentes que necesitan acceder a los datos y les ofrece esos datos a través de su valor de contexto.
2. **Consumidor (Consumer):** Es un componente de React que consume los datos proporcionados por el Proveedor. Los componentes Consumidores se colocan en la jerarquía de componentes donde necesitan acceder a los datos del contexto y reciben automáticamente los datos proporcionados por el Proveedor más cercano en la jerarquía.
3. **Contexto (Context):** Es el objeto que se crea utilizando la función `createContext` de React. Representa el contexto en sí mismo y define el valor inicial que se proporcionará a los componentes Consumidores si no hay un Proveedor que proporcione datos.

React Context

```
1 import React, { createContext, useContext, useState } from "react";
2
3 // Crear el contexto
4 const MyContext = createContext();
5
6 // Crear el hook personalizado para acceder al contexto
7 const useMyContext = () => useContext(MyContext);
8
9 // Componente Proveedor
10 const MyContextProvider = ({ children }) => {
11   const [count, setCount] = useState(0);
12
13   const increment = () => {
14     setCount(count + 1);
15   };
16
17   const decrement = () => {
18     setCount(count - 1);
19   };
20
21   return (
22     <MyContext.Provider value={{ count, increment, decrement }}>
23       {children}
```

```
26   </MyContext.Provider>
27 );
28 // Componente Consumidor
29 const Counter = () => {
30   const { count, increment, decrement } = useMyContext();
31
32   return (
33     <div>
34       <p>Contador: {count}</p>
35       <button onClick={increment}>Incrementar</button>
36       <button onClick={decrement}>Decrementar</button>
37     </div>
38   );
39 };
40
41 // Componente principal
42 const App = () => {
43   return (
44     <MyContextProvider>
45       <Counter />
46     </MyContextProvider>
47   );
48 };
49
50 export default App;
51
52 // TODO: Hacer el ejercicio de Tasklist
53
```



Escuela
Nacional de
Estudios
Superiores

IECAGto[®]
Instituto Estatal de Capacitación

¡GRACIAS!