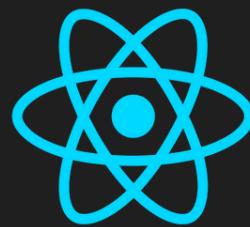


COMPONENTES Y CICLO DE VIDA EN REACT



Por:

Keyla Dayana Arboleda Mina
Carlos Andres Cifuentes Montaño
Jose Fernando Sinisterra Ibargüen
Darío Restrepo Landázury



COMPONENTES

Son **piezas de código que renderizan una parte de la interfaz**. Los componentes pueden ser parametrizados, reutilizados y pueden contener su propio estado.

Cabe resaltar que los componentes **son funciones regulares de JavaScript**, pero sus nombres deben comenzar con letra mayúscula o no funcionarán

COMPONENTES

El ejemplo de código implementa un componente **Footer**(*pie de pagina*).



```
1 export default function Footer (){
2     return(
3         <footer>
4             <p>
5                 © {new Date().getFullYear()} Todos los derechos reservados.
6             </p>
7         </footer>
8     )
9 }
```

La estructura básica de siempre debe empezar con **export default** esto es una palabra reservada que lo que hace es el componente pueda ser exportado por otros componente; **function** que le indica que es una función JavaScript.



PROPS

En **React**, son una forma de pasar datos de un componente padre a un componente hijo. Se ven y funcionan como un parámetro de una función, permitiendo que **un componente reciba información externa y la utilice para renderizar contenido dinámico**. Algunas de sus características son las siguientes:

- **Son inmutables**, es decir que no se puede modificar dentro del componente que lo recibe.
- **Son Unidireccionales** y los datos fluyen de arriba a abajo(padre a hijo).
- **Son personalizables** y permiten reutilizar componentes con diferentes datos.

PROPS

El código implementa los **props** en un componente **Avatar(person,size)** debemos asegurarnos de incluir el par de llaves `{ }` dentro del `()` al declarar las props, dentro del componente tenemos atributos a los cuales podemos definir diferentes datos en este componente.



```
1  export default function Avatar({ person, size }){  
2      return(  
3          <img className="Avatar"  
4              src={person.imageUrl}  
5              alt={person.name}  
6              width={size}  
7              height={size}  
8          />  
9      )  
10 }
```



EVENTOS

React te permite añadir **controladores de eventos** a tu **JSX**. Los controladores de eventos **son tus propias funciones que se ejecutarán en respuesta a interacciones** como hacer clic, hover, enfocar inputs en formularios, entre otras.

EVENTOS



El siguiente ejemplo muestra como se implementa los **eventos** en un componente

```
1 export default function Button({ children, onClick,  
2   disabled = false }) {  
3   return (  
4     <button onClick={onClick} disabled={disabled}>  
5       {children}  
6     </button>  
7   );  
8 }
```

Se muestra cómo se utiliza un controlador de eventos para un botón. Este botón, al principio, está inactivo hasta que se hace clic en él, como se indica en la línea que menciona onClick y disabled.

CICLO DE VIDA

Este se divide en tres fases:

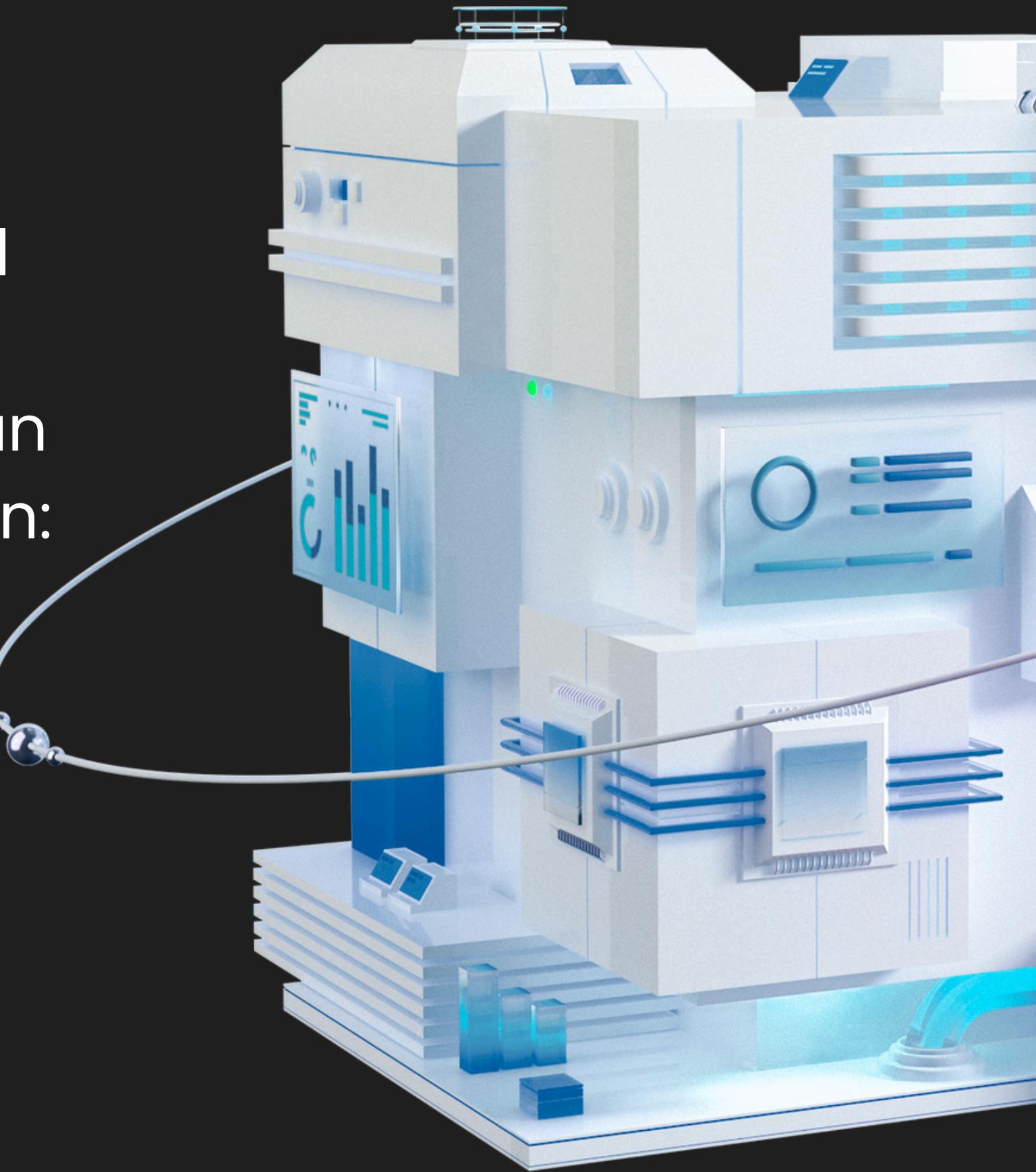
- **Montaje:** cuando el componente se añade al DOM.
- **Actualización:** cuando el componente se actualiza.
- **Desmontaje:** cuando el componente se elimina del DOM.



CICLO DE VIDA

Dentro de este ciclo de vida, existe un conjunto de métodos que se ejecutan en el componente. Los métodos mencionados anteriormente definen la clase y se ejecutan en el orden que se muestran a continuación:

- **constructor**
- **render**
- **componentDidMount**
- **componentDidUpdate**
- **componentWillUnmount**



MUCHAS GRACIAS

