

Taller sobre componentes frontend

Danny Julián Perilla Mikán

Marzo, 2025

1 Introducción

En el desarrollo web moderno, los componentes frontend juegan un papel fundamental en la construcción de interfaces dinámicas y modulares. Estos elementos encapsulan lógica y presentación, permitiendo la reutilización de código y facilitando el mantenimiento de las aplicaciones. La idea de este trabajo es proporcionar una comprensión clara de su funcionamiento, explorando desde conceptos básicos hasta su implementación en frameworks populares como React.

2 Diferencia entre React y JSX

React es una biblioteca de JavaScript creada por Facebook para construir interfaces de usuario interactivas y dinámicas. Permite desarrollar aplicaciones de una sola página (SPA) y gestionar la actualización eficiente del DOM. **JSX** es una extensión de JavaScript que permite escribir código similar a HTML dentro de archivos JavaScript. JSX facilita la creación de interfaces en React al permitir la combinación de HTML y JavaScript en un mismo archivo, manteniendo la sintaxis XML. Un ejemplo de JSX es el siguiente:

```
1 const elemento = <h1>Hola, mundo</h1>;
```

3 ¿Qué son clases en React?

Las clases en React son una forma de definir componentes con estado. A diferencia de los componentes funcionales, los componentes de clase pueden almacenar y gestionar estados internos y utilizar métodos del ciclo de vida de React. Las características de los componentes de clase son:

Características de las Clases en React

1. Se definen usando la sintaxis de clases de ES6 (`class`).
 2. Utilizan el método `render()` para devolver la interfaz de usuario.
 3. Pueden manejar estados con `this.state`.
 4. Se usa `this.setState()` para actualizar el estado.
 5. Requieren la función `super()` dentro del constructor.
-

Ejemplo de un componente de clase en React:

```
1 class Bienvenido extends React.Component {  
2   constructor(props) {
```

```
3     super(props);
4     this.state = { mensaje: "Hola, usuario" };
5   }
6
7   render() {
8     return <h1>{this.state.mensaje}</h1>;
9   }
10 }
```

—

4 Principales eventos de React

Los eventos en React funcionan de manera similar a los eventos en JavaScript, pero se escriben en **camel-Case** en lugar de minúsculas, y en lugar de cadenas de texto, se pasan funciones como manejadores de eventos.

Algunos eventos más utilizados en React:

Evento	Descripción
onClick	Se activa cuando el usuario hace clic en un elemento.
onChange	Se activa cuando el valor de un campo de entrada cambia.
onSubmit	Se activa cuando un formulario es enviado.
onMouseOver	Se activa cuando el cursor pasa sobre un elemento.
onKeyDown	Se activa cuando se presiona una tecla.

Ejemplo de evento `onClick` en React:

```
1 class Boton extends React.Component {
2   manejarClick() {
3     alert("Boton clickeado!");
4   }
5
6   render() {
7     return <button onClick={this.manejarClick}>Haz clic</button>;
8   }
9 }
```

—

5 Mapa conceptual de React

A continuación, se presenta un esquema conceptual de React:

```
1 React JS
2 - Componentes
3   - Funcionales
4   - De Clase
5   - Propiedades (Props)
6   - Estado (State)
7
8 - JSX
9   - Sintaxis de JavaScript + HTML
10  - Conversion a JavaScript puro
11
12 - Eventos
13   - onClick
14   - onChange
15   - onSubmit
16
17 - Hooks (para componentes funcionales)
18   - useState
19   - useEffect
20   - useContext
21
22 - Ciclo de vida del componente (para componentes de clase)
23   - componentDidMount
24   - componentDidUpdate
25   - componentWillUnmount
26
27 - Listas y Keys
28   - Metodos como map()
29   - Claves unicas en listas (key)
30
31 - Formularios
32   - Manejo con estados
33   - Eventos de entrada y envio
34
35 - Instalacion y Ejecucion
36   - npx create-react-app my-app
37   - npm start (modo desarrollo)
38   - npm run build (modo produccion)
```

6 Conclusiones

En este taller se exploraron los fundamentos del desarrollo frontend con React, destacando la importancia de los componentes como base de la estructura modular y reutilizable de las aplicaciones. Se abordó el uso de JSX, que permite combinar JavaScript con una sintaxis similar a HTML para facilitar la creación de interfaces. Además, se analizaron los eventos en React, esenciales para la interacción con el usuario, y la diferencia entre componentes funcionales y de clase, resaltando la evolución hacia los hooks como `useState` y `useEffect` para la gestión eficiente del estado. La correcta administración del estado, ya sea mediante `this.state` en componentes de clase o con hooks en componentes funcionales, es clave para el desarrollo de aplicaciones dinámicas y reactivas. En conclusión, React proporciona una metodología estructurada y eficiente para construir interfaces modernas y escalables, siendo una herramienta esencial en el desarrollo web actual.