

Glosario





API (APPLICATION PROGRAMMING INTERFACE)

Conjunto de definiciones y protocolos que permiten que dos aplicaciones se comuniquen entre sí.

Ejemplo: utilizar Fetch o Axios para hacer una solicitud a una API REST.

AXIOS

Una biblioteca de terceros basada en promesas que facilita la realización de solicitudes HTTP.

Ejemplo: `axios.get('https://api.example.com/data').then(response => console.log(response.data));`



COMPONENTES

Bloques de construcción reutilizables que encapsulan lógica y presentación en React.

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



ESTADO (STATE)

Un objeto que representa la información local de un componente que puede cambiar con el tiempo y afectar la renderización.

Ejemplo: `const [cuenta, setCuenta] = useState(0);`



FETCH

Una API nativa de JavaScript para realizar solicitudes HTTP basada en promesas.

Ejemplo: `fetch('https://api.example.com/data').then(response => response.json()).then(data => console.log(data));`



HOOKS

Funciones especiales en React que permiten usar estado y otras características de React en componentes funcionales.

Ejemplo: `useState, useEffect.`



JSX (JAVASCRIPT XML)

Una extensión de JavaScript que permite escribir código que se parece a HTML dentro de JavaScript.

Ejemplo: `<h1>Hola, {props.nombre}</h1>`



LOCALSTORAGE

Una API de almacenamiento web que permite almacenar datos de manera persistente en el navegador del usuario.

Ejemplo: `localStorage.setItem('clave', 'valor');`



PROPS (PROPIEDADES)

Argumentos que se pasan a los componentes para personalizarlos. Son de solo lectura.

Ejemplo: `<Saludo nombre="Mundo" />`

PROMESAS (PROMISES)

Un objeto que representa la eventual finalización (o falla) de una operación asíncrona y su valor resultante.

Ejemplo: `fetch('https://api.example.com/data').then(response => response.json());`



REACT

Una biblioteca de JavaScript para construir interfaces de usuario interactivas desarrollada por Facebook.

Ejemplo: Crear una aplicación SPA utilizando componentes de React.



USEEFFECT

Un *hook* que permite realizar efectos secundarios en componentes funcionales, similar a los métodos de ciclo de vida en componentes de clase.

Ejemplo: `useEffect(() => { document.title = Cuenta: ${cuenta}; }, [cuenta]);`

USESTATE

Un *hook* que permite agregar estado a componentes funcionales en React.

Ejemplo: `const [cuenta, setCuenta] = useState(0);`

Abramov, D. y Clark, A. (2020). *React: Up & Running: Building Web Applications*. O'Reilly Media.

Axios GitHub Repository. (2023). *Axios*.
<https://github.com/axios/axios>

Banks, A. y Porcello, E. (2017). *Learning React: Functional Web Development with React and Redux*. O'Reilly Media.

Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.

Mozilla Developer Network (MDN). (2023). *JavaScript documentation*.
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>