




Parte 3




REACT



Keys - ¿Qué? ¿Por qué?

Una "**key**" es un atributo especial que se debe incluir al crear listas de elementos.



Las **keys** ayudan a React a identificar qué elementos han cambiado, se han agregado o se han eliminado. Las **keys** se deben dar a los elementos dentro de un **array** para que tengan una identidad estable.


Implementación

```
const todoItems = todos.map((todo, index) =>  
  <li key={index}> {todo.text} </li>  
);
```

En el caso de omitir la key, se mostrará un warning por consola.



Eventos

- El manejo de eventos en React es similar al DOM.
 - Los eventos React se nombran usando camelCase, en lugar de minúsculas.
 - Con JSX pasas una función como el controlador de eventos, en lugar de una cadena.
- 



Implementación

HTML

```
<button onclick="activateLasers()"> Activate Lasers </button>
```

REACT


```
<button onClick={activateLasers}> Activate Lasers </button>
```





Listener

Cuando se usa React, generalmente no necesita llamar a `addEventListener` para agregar listeners a un elemento DOM después de haber sido creado. En su lugar, solo se proporciona uno cuando el elemento se representa inicialmente.



Ejemplo

```
class Toggle extends React.Component {
  constructor(props) {
    super(props);
    this.state = {isToggleOn: true};
    this.handleClick = this.handleClick.bind(this);
  }

  handleClick() {
    this.setState(prevState => ({ isToggleOn: !prevState.isToggleOn }));
  }
  render() { return (
    <button onClick={this.handleClick}> {this.state.isToggleOn ? 'ON' : 'OFF'}
    </button> );
  }
}

ReactDOM.render( <Toggle />, document.getElementById('root') );
```