

# MANEJO DE ESTADOS Y EVENTOS COMPLEJOS

- ▶ Creamos un componente con el nombre "TodoItem.js"
- ▶ Definimos los diferentes estados del componente.
- ▶ Definimos los eventos para los cambios de estos estados.

```
import React, {Component} from 'react'; 7.6K (gzipped:
```

Complexity is 9 It's time to do something...

```
export default class TodoItem extends Component { ■
```

```
  constructor(props) {  
    super(props);  
    this.state = {  
      editMode: false,  
      todo: props.children  
    }  
  }
```

```
  onEditMode () {  
    if(this.state.editMode) {  
      const {onUpdateTodo} = this.props;  
      onUpdateTodo(this.state.todo);  
    }  
    this.setState({  
      editMode: !this.state.editMode  
    });  
  }
```

```
  handleChange (evt) {  
    this.setState({todo: evt.target.value});  
  }
```

- El evento `dynamiccomponents` define la estructura html para los diferentes estados del componente

```
dynamicComponents () {  
  let input, actionButton;  
  const {editMode, todo} = this.state;  
  
  if(editMode) {  
    input = (  
      <input  
        type="text"  
        className="form-control"  
        value={todo}  
        onChange={(e) => this.handleChange(e)}  
      />  
    );  
    actionButton = (  
      <button  
        type="button"  
        className="btn btn-warning text-white"  
        onClick={this.onEditMode.bind(this)}  
      >  
        Actualizar  
      </button>  
    );  
  } else {  
    input = todo;  
    actionButton = (  
      <button  
        type="button"  
        className="btn btn-warning text-white"  
        onClick={this.onEditMode.bind(this)}  
      >  
        Editar  
      </button>  
    );  
  }  
  return {input, actionButton};  
}
```