

Front End III

Métodos para la aplicación de Error Boundary

static getDerivedStateFromError()

Este método nos permitirá renderizar una interfaz alternativa cuando ocurra un error, de manera de poder dar feedback a la persona.

componentDidCatch()

Este método nos permitirá registrar información de los errores. Veamos cómo se vería este componente en nuestro código:

```
class ErrorBoundary extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { hasError: false };  
  }  
  
  static getDerivedStateFromError(error) {  
    // Actualizamos el estado para que el siguiente  
    // renderizado muestre la interfaz alternativa  
    return { hasError: true };  
  }  
  
  componentDidCatch(error, errorInfo) {
```

```

// Podemos utilizar este método para enviar el error
// a algún servicio de monitoreo
logErrorToMyService(error, userInfo);

}

render() {
  if (this.state.hasError) {
    // Cuando ocurre un error, renderizamos una interfaz
    // alternativa para dar feedback al usuario
    return <h1>Ha ocurrido un error.</h1>;
  }
  // Caso contrario retornamos los componentes hijos
  return this.props.children;
}

```

Una vez creado, podemos utilizar nuestro **ErrorBoundary** de la siguiente manera:

```

<ErrorBoundary>
  <MiComponente />
</ErrorBoundary>

```

En definitiva, un Error Boundary funciona como un bloque **catch{}** de JavaScript, pero para componentes. A su vez, la granularidad de su uso depende de cada caso en particular. Puesto que cada Error Boundary maneja los errores de sus componentes hijos, es posible tener varios de ellos en distintos niveles e, incluso, ofreciendo distintas interfaces alternativas. Por ejemplo, podríamos envolver los componentes de más alto nivel para mostrar un mensaje genérico al usuario, pero también es posible envolver componentes específicos en un **Error Boundary** para evitar que hagan fallar el resto de la aplicación. La decisión dependerá de la estrategia que se adopte en cada caso conforme las necesidades y arquitectura de la aplicación.