

Tipo : Guía de laboratorio
Capítulo : React JS
Duración : 60 minutos

I. OBJETIVO

Crear un proyecto React para el manejo de formularios y validaciones.

II. REQUISITOS

Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Editor Visual Studio Code
- Node JS

III. EJECUCIÓN DEL LABORATORIO

- **Ejercicio 2.6: Utilizar Formik para manejo de formularios.**

1. Instalar la librería **Formik**.

```
npm install formik --save
```

2. Instalar la librería **Yup**.

```
npm install yup --save
```

3. Crear el componente **ProductForm**.

```

JS ProductForm.js
src > modules > admin > JS ProductForm.js
1  import React from 'react'
2
3  import { Formik } from 'formik';
4
5  export default ({ title, product, submit, back }) => {
6    return (
7      <div className="container">
8        <div className="columns">
9          <div className="column is-8 is-offset-2">
10             <Formik
11               initialValues={product}
12               validationSchema={productSchema}
13               onSubmit={(values) => {
14                 values.stock = parseInt(values.stock, 10);
15                 submit(values);
16               }}
17             >
18               {{{
19                 values,
20                 isValid,
21                 errors,
22                 touched,
23                 handleBlur,
24                 handleChange,
25                 handleSubmit
26               }} => (
27                 <form action="" noValidate autoComplete="off">--
118             </Formik>
119           </div>
120         </div>

```

- Utilizar la librería Yup para el manejo de las validaciones del formulario.

```

4  import * as yup from 'yup';
5
6  const productSchema = yup.object({
7    name: yup.string().required().min(6),
8    detail: yup.string().required().min(15),
9    price: yup.string().required(),
10   stock: yup.number().required().min(0)
11 });
12
13 export default ({ title, product, submit, back }) => {
14   return (
15     <div className="container">
16       <div className="columns">
17         <div className="column is-8 is-offset-2">
18           <Formik
19             initialValues={product}
20             validationSchema={productSchema}
21             onSubmit={(values) => {
22               values.stock = parseInt(values.stock, 10);
23               submit(values);
24             }}
25           >

```

- Actualizar los componentes de **NewProduct** y **EditProduct** para realizar sus respectivas acciones.

```

JS NewProduct.js ×
src > modules > admin > JS NewProduct.js > NewProduct > render
6 class NewProduct extends Component {
7   handleSubmit = product => {
8     ProductService.createProduct(product)
9     .then(response => {
10       this.goBack();
11     });
12   }
13
14   goBack = () => {
15     this.props.history.goBack();
16   }
17
18   render() {
19     const product = { name: '', detail: '', price: '', stock: '' };
20
21     return (
22       <div className="columns">
23         <div className="column is-10 is-offset-1">
24           <h2 className="subtitle is-3 has-text-centered">Crear Producto</h2>
25           <ProductForm
26             title="Crear"
27             product={product}
28             submit={this.handleSubmit}
29             back={this.goBack}
30           />
31         </div>
32       </div>
33     );
34   }
35 }

```

```

JS EditProduct.js ×
src > modules > admin > JS EditProduct.js > EditProduct > goBack
17 handleSubmit = product => {
18   ProductService.updateProduct(product)
19   .then(response => {
20     this.goBack();
21   });
22 }
23
24 goBack = () => {
25   this.props.history.goBack();
26 }
27
28 render() {
29   const { product } = this.state;
30
31   return (
32     <div className="columns">
33       <div className="column is-10 is-offset-1">
34         <h2 className="subtitle is-3 has-text-centered">Actualizar Producto</h2>
35         {
36           product.id ?
37             <ProductForm
38               title="Actualizar"
39               product={product}
40               submit={this.handleSubmit}
41               back={this.goBack}/>
42             : null
43         }
44       </div>
45     </div>
46   );

```

Actualizar Producto

Televisor 49" 4K UHD

Modelo: 49UK6200

1499.00

Stock: ej. 5

El stock es requerido, debe ser número y mayor o igual a 0.

Regresar Actualizar

Admin desarrollado en React JS © 2020

IV. EVALUACIÓN

1. ¿Qué otras librerías para manejo de Formularios y validaciones usted conocer?

El manejo de los formularios existe desde los inicios de JavaScript, por ello existen muchas librerías que nos permite trabajar con ellos, muchos de ellos basados en jQuery como dependencia principal. Pero con la llegada de los Frameworks JS, cada tecnología implementa su manejo de validaciones.