Creando la sección de productos.

Basado en el curso:
Node.js - Bootcamp Desarrollo Web inc. MVC y REST APIs
Juan Pablo De la torre Valdez
https://www.udemy.com/course/nodejs-bootcamp-desarrollo-web-mvc-y-rest-apis/

Creando el Routing y Componentes

Creamos los siguientes componentes:

Por lo menos les agregamos a todos ellos la información básica.

EditarProducto.jsx

```jsx
import React from 'react';

const EditarProducto = () => {
    return (
        <>
            <h2>Editar producto</h2>
        </>
    );
};

export default EditarProducto;
```

NuevoProducto.jsx

```jsx
import React from 'react';

const NuevoProducto = () => {
    return (
        <>
            <h2>Nuevo producto</h2>
        </>
    );
};

export default NuevoProducto;
```

Producto.jsx

```
import React from 'react';

const Producto = () => {
    return (
        <>
            <h2>Producto</h2>
        </>
    );
};

export default Producto;
```

Agregamos las rutas al archivo App.jsx

```
import React, {Fragment} from 'react'
import {BrowserRouter as Router, Route, Switch} from "react-router-dom";

/* Layouts*/
import Header from "./components/layout/Header.jsx";
import Navegacion from './components/layout/Navegacion.jsx';

/* Components*/
import Clientes from "./components/clientes/Clientes.jsx";
import NuevoCliente from './components/clientes/NuevoCliente';
import EditarCliente from './components/clientes/EditarCliente';

import Pedidos from "./components/pedidos/Pedidos.jsx";

import Productos from "./components/productos/Productos.jsx";
import NuevoProducto from "./components/productos/NuevoProducto.jsx";
import EditarProducto from "./components/productos/EditarProducto.jsx";

const App = function () {

    return (
        <Router>
            <Fragment>
                <Header/>

                <div className="grid contenedor contenido-principal">
                    <Navegacion/>

                    <main className="caja-contenido col-9">
                        <Switch>
                            <Route exact path="/" component={Clientes}/>
                            <Route exact path="/clientes/nuevo" component={NuevoCliente}/>
                            <Route exact path="/clientes/editar/:id"
                                   component={EditarCliente}/>

                            <Route exact path="/pedidos" component={Pedidos}/>

                            <Route exact path="/productos" component={Productos}/>
                            <Route exact path="/productos/nuevo" component={NuevoProducto}/>
```

```
                            <Route exact path="/productos/editar/:id"
                                   component={EditarProducto}/>

                        </Switch>
                    </main>
                </div>
            </Fragment>
        </Router>

    )
}

export default App;
```

Obteniendo los Productos de la API

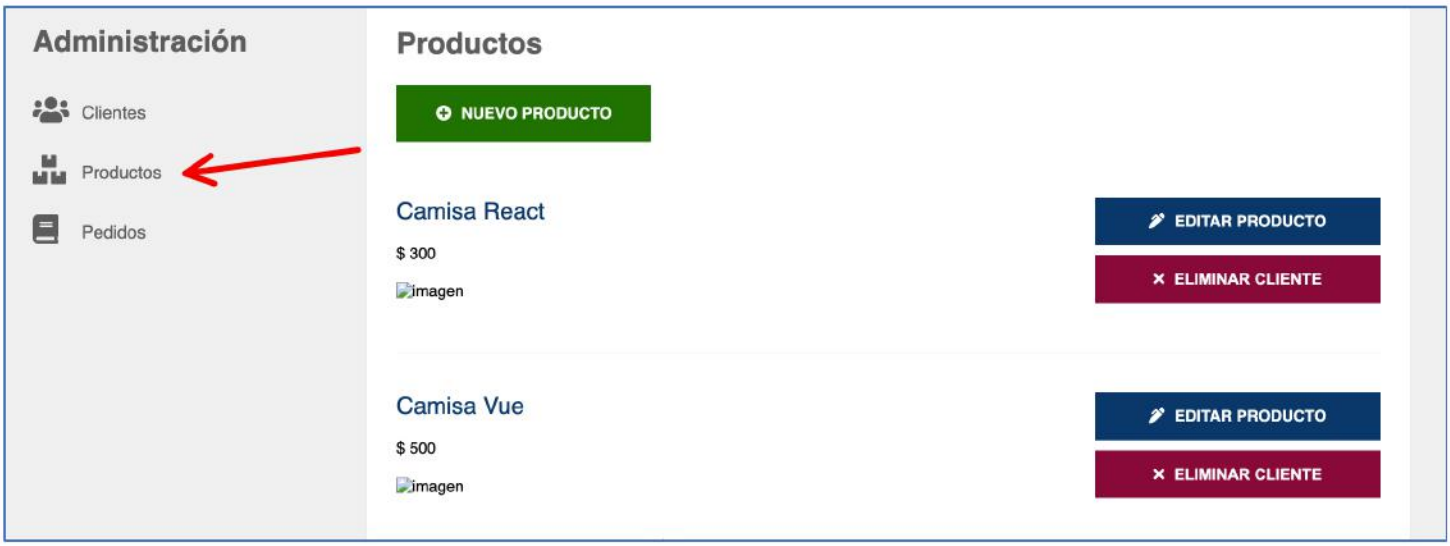
Modificamos el archivo Producto.jsx

```
import React from 'react';
import {Link} from 'react-router-dom';
import Swal from 'sweetalert2';
import clienteAxios from '../../config/axios';

const Producto = ({producto}) => {
    // elimina un producto
    const eliminarProducto = id => {

    }
    const {_id, nombre, precio, imagen} = producto;
    return (
        <>
            <li className="producto">
                <div className="info-producto">
                    <p className="nombre">{nombre}</p>
                    <p className="precio">$ {precio}</p>
                    {imagen ? (
                        <img src={`http://localhost:4000/${imagen}`} alt="imagen"/>
                    ) : null}
                </div>
                <div className="acciones">
                    <Link to={`/productos/editar/${_id}`} className="btn btn-azul">
                        <i className="fas fa-pen-alt"></i>
                        Editar Producto
                    </Link>

                    <button
                        type="button"
                        className="btn btn-rojo btn-eliminar"
                        onClick={() => eliminarProducto(_id)}
                    >
                        <i className="fas fa-times"></i>
                        Eliminar Cliente
                    </button>
                </div>
            </li>
        </>
    );
};

export default Producto;
```

Ahora modificamos el archivo Productos.jsx

```
import React, {useEffect, useState} from 'react';
import {Link} from 'react-router-dom';
// importar cliente axios
import clienteAxios from '../../config/axios';
import Producto from './Producto';

const Productos = (props) => {
    // productos = state, guardarproductos = funcion para guardar el state
    const [productos, guardarProductos] = useState([]);
    // useEffect para consultar api cuando cargue
    useEffect(() => {
        // Query a la API
        const consultarAPI = async () => {
            try {
                const productosConsulta = await clienteAxios.get('/productos');
                guardarProductos(productosConsulta.data);
            } catch (error) {

            }
        }
        // llamado a la api
        consultarAPI();

    }, [productos]);
    return (
        <>
            <h2>Productos</h2>
            <Link to={'/productos/nuevo'} className="btn btn-verde nvo-cliente">
                <i className="fas fa-plus-circle"></i>
                Nuevo Producto
            </Link>

            <ul className="listado-productos">
                {productos.map(producto => (
                    <Producto
                        key={producto._id}
                        producto={producto}
                    />
                ))}
            </ul>
        </>
    );
};

export default Productos;
```

Si visualizamos la página en productos nos mostrará lo siguiente:



Listando los Productos y sus Imágenes

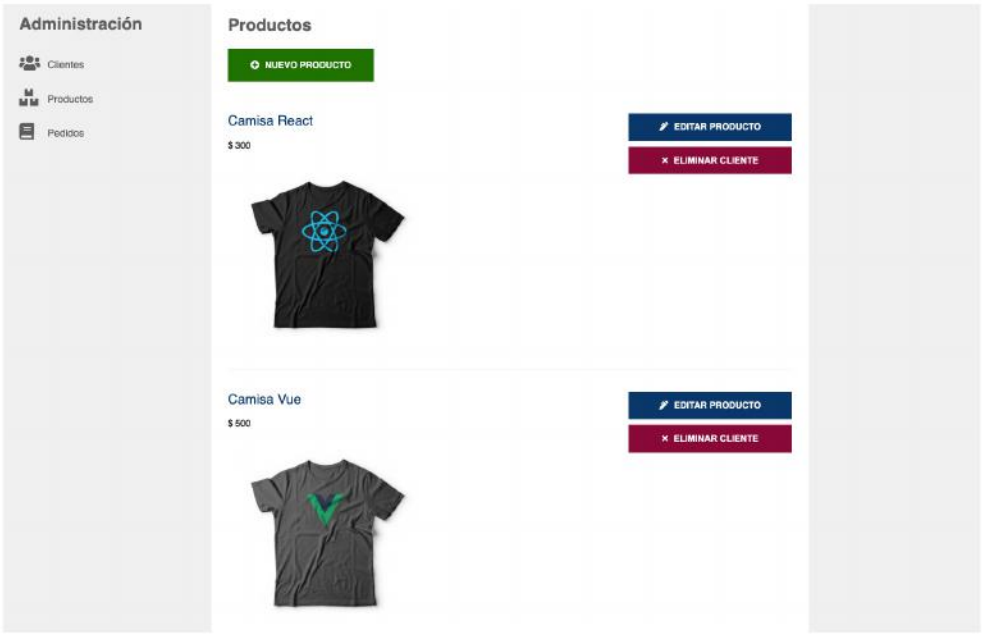
Para poder ver las imágenes tenemos que hacer un cambio en el `servidor`. Cuando se cargan las imágenes se guardan en la siguiente carpeta:

Por lo que hay que dar de alta esa carpeta en main.js

```
const express = require("express");
const routes = require("./routes");
const mongoose = require("mongoose");
const bodyParser = require('body-parser');

// Cors permite que un cliente se conecte a otro servidor para el intercambio de recursos
const cors = require('cors');

// Conectar mongo
mongoose.Promise = global.Promise;
mongoose.connect("mongodb://localhost:27017/restapis", {
    useNewUrlParser: true
});

// Crear el servidor
const app = express();

// habilitar bodyparser
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));

// Habilitar cors
app.use(cors());
```

```
// Rutas de la app
app.use("/", routes());

// Carpeta pública
app.use(express.static("uploads"));


// puerto
app.listen(4000, function () {
    console.log("Servidor corriendo en puerto 4000");
})
```

Reiniciamos el servidor y actualizamos la página del cliente.



Eliminando Productos

Para eliminar el producto le anexamos el siguiente código en `Producto.jsx`:

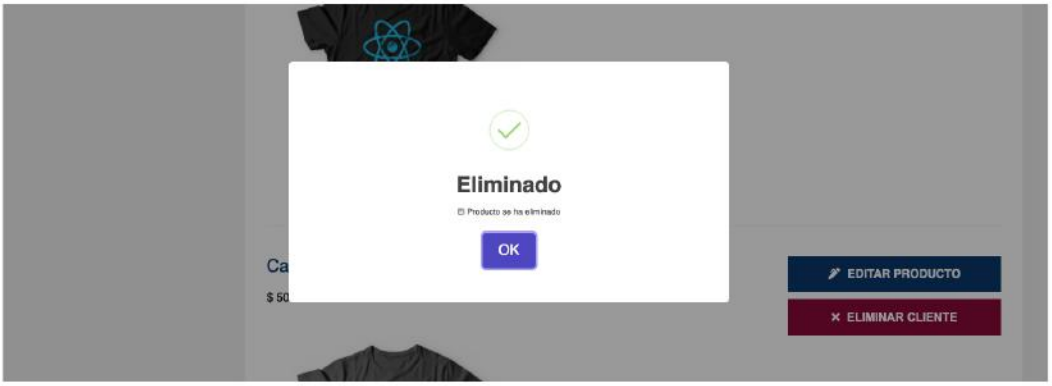
```
import React from 'react';
import {Link} from 'react-router-dom';
import Swal from 'sweetalert2';
import clienteAxios from '../../config/axios';

const Producto = ({producto}) => {
    // elimina un producto
    const eliminarProducto = id => {
        Swal.fire({
            title: '¿Estás seguro?',
            text: "Un producto eliminado no se puede recuperar",
            type: 'warning',
            showCancelButton: true,
            confirmButtonColor: '#3085d6',
            cancelButtonColor: '#d33',
            confirmButtonText: 'Si, Eliminar',
            cancelButtonText: 'No, Cancelar'
        }).then((result) => {
            if (result.value) {
                // eliminar en la rest api
                clienteAxios.delete(`/productos/${id}`)
                    .then(res => {
                        if (res.status === 200) {
                            Swal.fire(
                                'Eliminado',
                                res.data.mensaje,
                                'success'
                            )
                        }
                    })
            }
        })
    }
    const {_id, nombre, precio, imagen} = producto;
    return (
        <>
            <li className="producto">
                <div className="info-producto">
                    <p className="nombre">{nombre}</p>
                    <p className="precio">$ {precio}</p>
                    {imagen ? (
                        <img src={`http://localhost:4000/${imagen}`} alt="imagen"/>
                    ) : null}
                </div>
                <div className="acciones">
                    <Link to={`/productos/editar/${_id}`} className="btn btn-azul">
                        <i className="fas fa-pen-alt"></i>
                        Editar Producto
                    </Link>

                    <button
                        type="button"
                        className="btn btn-rojo btn-eliminar"
                        onClick={() => eliminarProducto(_id)}
                    >
                        <i className="fas fa-times"></i>
                        Eliminar Cliente
                    </button>
```

```
                </div>
            </li>
        </>
    );
};

export default Producto;
```

La linea que manda llamar a la REST API es:
`clienteAxios.delete(`/productos/${id}`)`

Todo lo demás manda llamar al cuadro de confirmación y el mensaje de que el producto ha sido borrado.

Visualizamos la página y borramos.



Ya se borró de la página

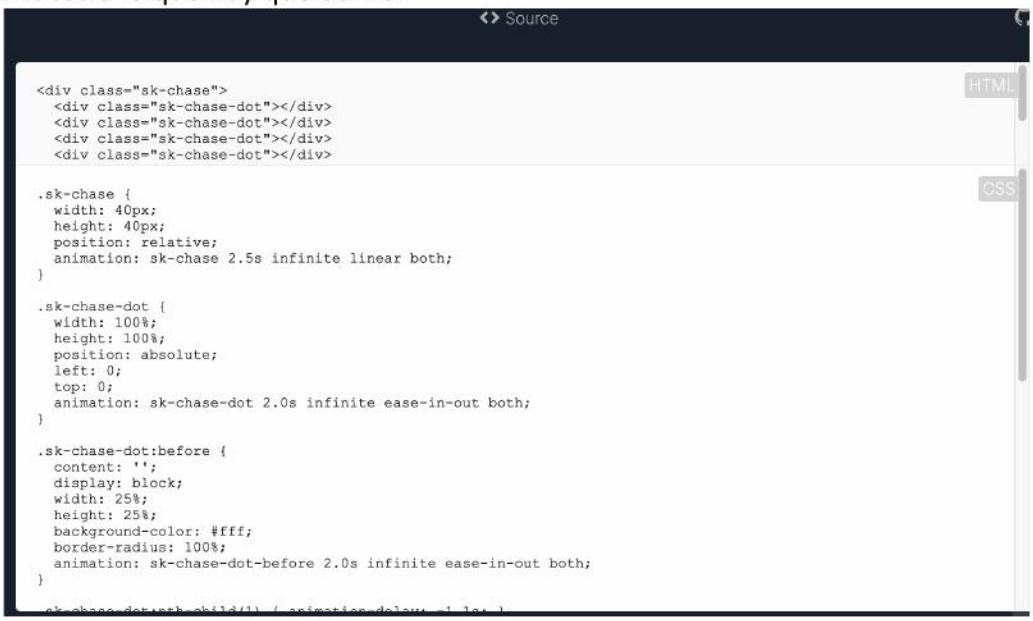
Agregando un spinner de carga

Muchas veces el contenido tarda en cargar. Para que no dé la apariencia de que la página se ha congelado se puede agregar un spinner.

Se pueden encontrar algunos de ellos en la siguiente dirección: https://tobiasahlin.com/spinkit/

El código para generar el Spinner se obtiene haciendo clic en el enlace **Source**

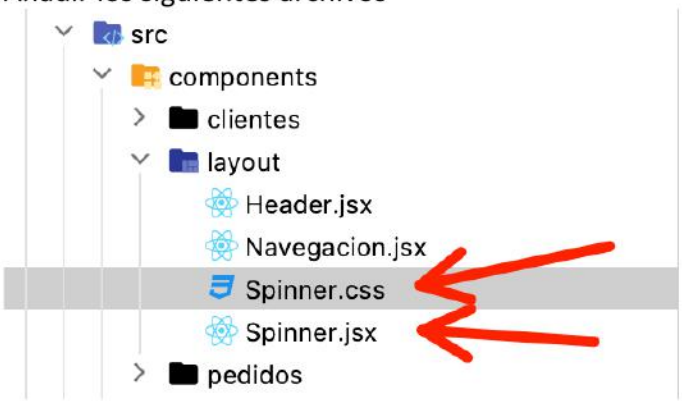
Muestra lo que hay que utilizar




```
<div class="sk-chase">
  <div class="sk-chase-dot"></div>
  <div class="sk-chase-dot"></div>
  <div class="sk-chase-dot"></div>
  <div class="sk-chase-dot"></div>
```

```
.sk-chase {
  width: 40px;
  height: 40px;
  position: relative;
  animation: sk-chase 2.5s infinite linear both;
}

.sk-chase-dot {
  width: 100%;
  height: 100%;
  position: absolute;
  left: 0;
  top: 0;
  animation: sk-chase-dot 2.0s infinite ease-in-out both;
}

.sk-chase-dot:before {
  content: '';
  display: block;
  width: 25%;
  height: 25%;
  background-color: #fff;
  border-radius: 100%;
  animation: sk-chase-dot-before 2.0s infinite ease-in-out both;
}
```

Añadir los siguientes archivos



Spinner.css (Esta hoja de estilos me la pueden solicitar)

```css
.spinner {
    margin: 100px auto;
    width: 40px;
    height: 40px;
    position: relative;
    text-align: center;

    -webkit-animation: sk-rotate 2.0s infinite linear;
    animation: sk-rotate 2.0s infinite linear;
}

.dot1, .dot2 {
    width: 60%;
    height: 60%;
    display: inline-block;
    position: absolute;
    top: 0;
    background-color: #333;
    border-radius: 100%;

    -webkit-animation: sk-bounce 2.0s infinite ease-in-out;
    animation: sk-bounce 2.0s infinite ease-in-out;
}

.dot2 {
    top: auto;
    bottom: 0;
    -webkit-animation-delay: -1.0s;
    animation-delay: -1.0s;
}

@-webkit-keyframes sk-rotate { 100% { -webkit-transform: rotate(360deg) }}
@keyframes sk-rotate { 100% { transform: rotate(360deg); -webkit-transform: rotate(360deg) }}

@-webkit-keyframes sk-bounce {
    0%, 100% { -webkit-transform: scale(0.0) }
    50% { -webkit-transform: scale(1.0) }
}
```

```
@keyframes sk-bounce {
    0%, 100% {
        transform: scale(0.0);
        -webkit-transform: scale(0.0);
    } 50% {
          transform: scale(1.0);
          -webkit-transform: scale(1.0);
      }
}
```

Spinner.jsx

```
import React from 'react';
import './Spinner.css';

const Spinner = () => {
    return (
        <>
            <div className="spinner">
                <div className="dot1"></div>
                <div className="dot2"></div>
            </div>
        </>
    );
};

export default Spinner;
```

En Productos.jsx anexamos la carga del Spinner en caso de que no haya productos aún cargados

```
import React, {useEffect, useState} from 'react';
import {Link} from 'react-router-dom';
// importar cliente axios
import clienteAxios from '../../config/axios';
import Producto from './Producto';
import Spinner from "../layout/Spinner.jsx";

const Productos = (props) => {
    // productos = state, guardarproductos = funcion para guardar el state
    const [productos, guardarProductos] = useState([]);
    // useEffect para consultar api cuando cargue
    useEffect(() => {
        // Query a la API
        const consultarAPI = async () => {
            try {
                const productosConsulta = await clienteAxios.get('/productos');
                guardarProductos(productosConsulta.data);
            } catch (error) {

            }
        }
```

```
        // llamado a la api
        consultarAPI();

    }, [productos]);

    if (!productos.length) return <Spinner/>
    return (
        <>
            <h2>Productos</h2>
            <Link to={'/productos/nuevo'} className="btn btn-verde nvo-cliente">
                <i className="fas fa-plus-circle"></i>
                Nuevo Producto
            </Link>

            <ul className="listado-productos">
                {productos.map(producto => (
                    <Producto
                        key={producto._id}
                        producto={producto}
                    />
                ))}
            </ul>
        </>
    );
};

export default Productos;
```

Por ahora, mientras tengamos pocos productos no se va a notar el Spinner, pero cuando haya más se notará antes de mostrarlos.

En Clientes.jsx hacer lo mismo, pero ocupando la línea:

```
if (!clientes.length) return <Spinner/>
```

Creando el formulario para nuevos productos

Primero se crea el formulario en NuevoProducto.jsx

```
import React, {useState} from 'react';

const NuevoProducto = () => {
    return (
        <>
            <h2>Nuevo producto</h2>
            <form            >
                <legend>Llena todos los campos</legend>

                <div className="campo">
                    <label>Nombre:</label>
                    <input
                        type="text"
                        placeholder="Nombre Producto"
                        name="nombre"

                    />
                </div>

                <div className="campo">
                    <label>Precio:</label>
                    <input
                        type="number"
                        name="precio"
                        min="0.00"
                        step="0.01"
                        placeholder="Precio"

                    />
                </div>

                <div className="campo">
                    <label>Imagen:</label>
                    <input
                        type="file"
                        name="imagen"

                    />
                </div>

                <div className="enviar">
                    <input type="submit" className="btn btn-azul"
                           value="Agregar Producto" />
                </div>
            </form>
        </>
    );
};

export default NuevoProducto;
```

Visualizamos

Colocando la información en el State

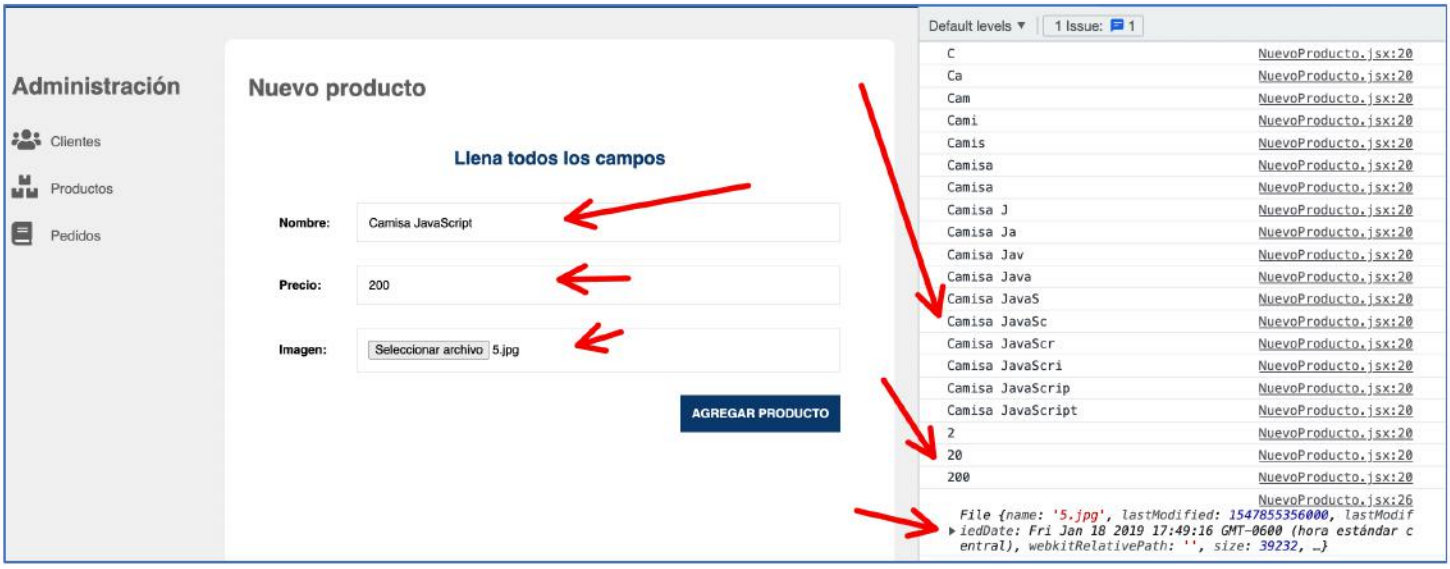
El siguiente paso es colocar los valores de las cajas en el state. En el caso de la imagen se anexa un State por separado.

Modificar `NuevoProducto.jsx`

```
import React, {useState} from 'react';

const NuevoProducto = () => {

    //producto = state, guardarProducto = setstate
    const [producto, guardarProducto] = useState({
        nombre: '',
        precio: ''
    });

    // archivo = state, guardarArchivo = setState
    const [archivo, guardarArchivo] = useState('');

    // leer los datos del formulario
    const leerInformacionProducto = e => {
        guardarProducto({
            // obtener una copia del state y agregar el nuevo
            ...producto,
            [e.target.name]: e.target.value
        })
        console.log(e.target.value)
    }

    // coloca la imagen en el state
    const leerArchivo = e => {
        guardarArchivo(e.target.files[0]);
        console.log(e.target.files[0]);
    }

    return (
        <>
            <h2>Nuevo producto</h2>
            <form>
                <legend>Llena todos los campos</legend>
```

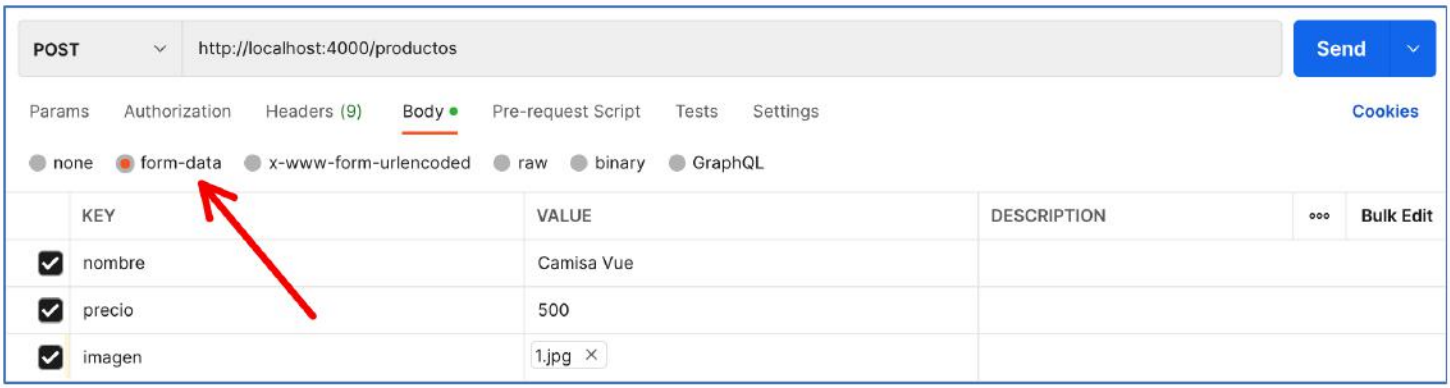
```
                <div className="campo">
                    <label>Nombre:</label>
                    <input
                        type="text"
                        placeholder="Nombre Producto"
                        name="nombre"
                        onChange={leerInformacionProducto}
                    />
                </div>

                <div className="campo">
                    <label>Precio:</label>
                    <input
                        type="number"
                        name="precio"
                        min="0.00"
                        step="0.01"
                        placeholder="Precio"
                        onChange={leerInformacionProducto}
                    />
                </div>

                <div className="campo">
                    <label>Imagen:</label>
                    <input
                        type="file"
                        name="imagen"
                        onChange={leerArchivo}
                    />
                </div>

                <div className="enviar">
                    <input type="submit" className="btn btn-azul"
                           value="Agregar Producto"/>
                </div>
            </form>
        </>
    );
};

export default NuevoProducto;
```

Ahora podemos rellenar nuestro formulario y cargar una imagen. Se agregaron salidas a la consola para ver el comportamiento de los states pero posteriormente habrá que quitarlos.



Registrar nuevos Productos y Subiendo Archivos con React

Como se estará subiendo una imagen, el código será diferente de cuando se agrega un cliente.
Si recordamos cuando se registra un nuevo producto se ocupa un `form-data`



De esa forma se ocupará en React para hacer el registro.

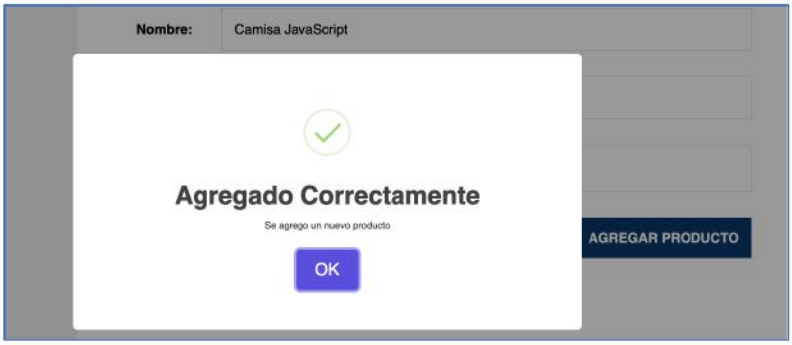
Modificar `NuevoProducto.jsx`

```
import React, {useState} from 'react';
import Swal from 'sweetalert2';
import clienteAxios from '../../config/axios';

const NuevoProducto = () => {

    //producto = state, guardarProducto = setstate
    const [producto, guardarProducto] = useState({
        nombre: '',
        precio: ''
    });
```

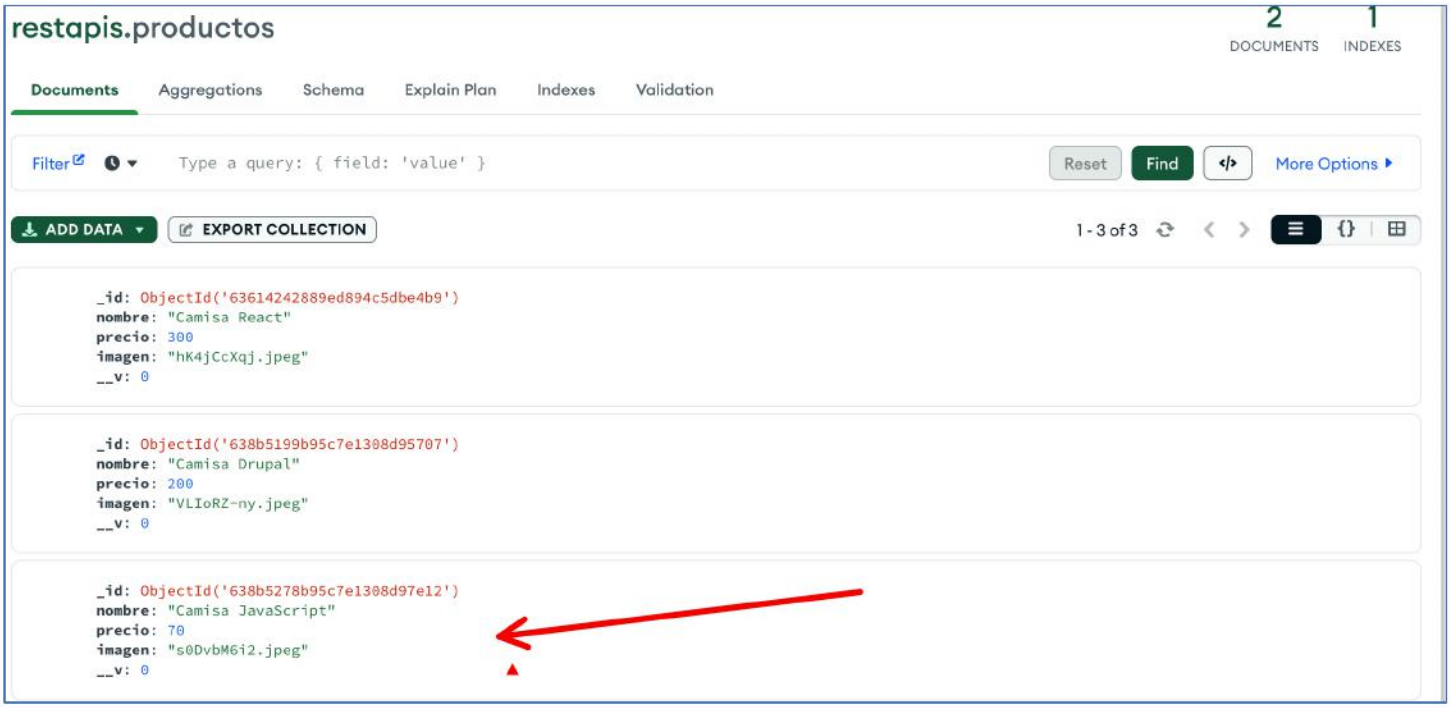
```
    // archivo = state, guardarArchivo = setState
    const [archivo, guardarArchivo] = useState('');

    // leer los datos del formulario
    const leerInformacionProducto = e => {
        guardarProducto({
            // obtener una copia del state y agregar el nuevo
            ...producto,
            [e.target.name]: e.target.value
        })
        console.log(e.target.value)
    }

    // coloca la imagen en el state
    const leerArchivo = e => {
        guardarArchivo(e.target.files[0]);
        console.log(e.target.files[0]);
    }

    // almacena el nuevo producto en la base de datos.
    const agregarProducto = async e => {
        e.preventDefault();

        // crear un formdata
        const formData = new FormData();
        formData.append('nombre', producto.nombre);
        formData.append('precio', producto.precio);
        formData.append('imagen', archivo);

        // almacenarlo en la BD
        try {
            const res = await clienteAxios.post('/productos', formData, {
                headers: {
                    'Content-Type': 'multipart/form-data'
                }
            });

            // Lanzar una alerta
            if (res.status === 200) {
                Swal.fire(
                    'Agregado Correctamente',
                    res.data.mensaje,
                    'success'
                )
            }
        } catch (error) {
            console.log(error);
            // lanzar alerta
            Swal.fire({
                type: 'error',
                title: 'Hubo un error',
                text: 'Vuelva a intentarlo'
            })
        }
    }

    return (
        <>
            <h2>Nuevo producto</h2>
            <form onSubmit={agregarProducto}>
                <legend>Llena todos los campos</legend>
```

```
                <div className="campo">
                    <label>Nombre:</label>
                    <input
                        type="text"
                        placeholder="Nombre Producto"
                        name="nombre"
                        onChange={leerInformacionProducto}
                    />
                </div>

                <div className="campo">
                    <label>Precio:</label>
                    <input
                        type="number"
                        name="precio"
                        min="0.00"
                        step="0.01"
                        placeholder="Precio"
                        onChange={leerInformacionProducto}
                    />
                </div>

                <div className="campo">
                    <label>Imagen:</label>
                    <input
                        type="file"
                        name="imagen"
                        onChange={leerArchivo}
                    />
                </div>

                <div className="enviar">
                    <input type="submit" className="btn btn-azul"
                           value="Agregar Producto"/>
                </div>
            </form>
        </>
    );
};

export default NuevoProducto;
```

Agregar el producto



Verificar en la base de datos que se haya agregado.



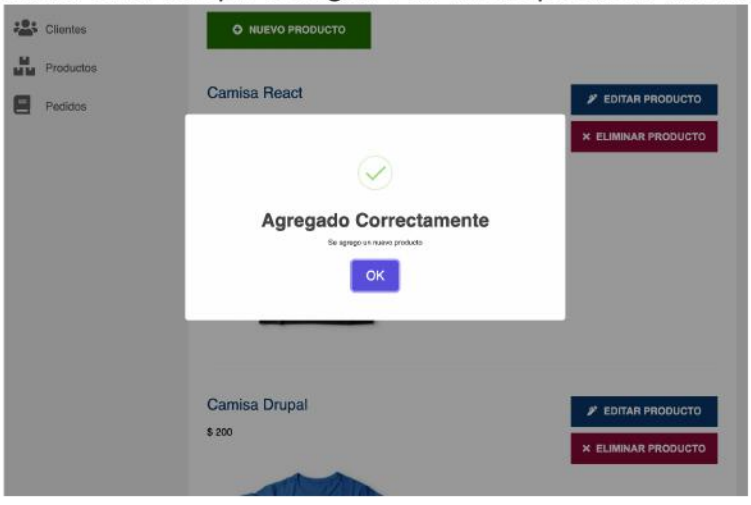
Solo falta redireccionar para que nos mande a la página de inicio (la de productos).

```
import React, {useState} from 'react';
import Swal from 'sweetalert2';
import clienteAxios from '../../config/axios';
import { withRouter } from 'react-router-dom';

const NuevoProducto = (props) => {

    //producto = state, guardarProducto = setstate
    const [producto, guardarProducto] = useState({
        nombre: '',
        precio: ''
    });

    // archivo = state, guardarArchivo = setState
    const [archivo, guardarArchivo] = useState('');
```

```
    // leer los datos del formulario
    const leerInformacionProducto = e => {
        guardarProducto({
            // obtener una copia del state y agregar el nuevo
            ...producto,
            [e.target.name]: e.target.value
        })
        console.log(e.target.value)
    }

    // coloca la imagen en el state
    const leerArchivo = e => {
        guardarArchivo(e.target.files[0]);
        console.log(e.target.files[0]);
    }

    // almacena el nuevo producto en la base de datos.
    const agregarProducto = async e => {
        e.preventDefault();

        // crear un formdata
        const formData = new FormData();
        formData.append('nombre', producto.nombre);
        formData.append('precio', producto.precio);
        formData.append('imagen', archivo);

        // almacenarlo en la BD
        try {
            const res = await clienteAxios.post('/productos', formData, {
                headers: {
                    'Content-Type': 'multipart/form-data'
                }
            });

            // Lanzar una alerta
            if (res.status === 200) {
                Swal.fire(
                    'Agregado Correctamente',
                    res.data.mensaje,
                    'success'
                )
            }
            // redireccionar
            props.history.push('/productos');
        } catch (error) {
            console.log(error);
            // lanzar alerta
            Swal.fire({
                type: 'error',
                title: 'Hubo un error',
                text: 'Vuelva a intentarlo'
            })
        }
    }

    return (
        <>
            <h2>Nuevo producto</h2>
            <form onSubmit={agregarProducto}>
                <legend>Llena todos los campos</legend>
```

```
                <div className="campo">
                    <label>Nombre:</label>
                    <input
                        type="text"
                        placeholder="Nombre Producto"
                        name="nombre"
                        onChange={leerInformacionProducto}
                    />
                </div>

                <div className="campo">
                    <label>Precio:</label>
                    <input
                        type="number"
                        name="precio"
                        min="0.00"
                        step="0.01"
                        placeholder="Precio"
                        onChange={leerInformacionProducto}
                    />
                </div>

                <div className="campo">
                    <label>Imagen:</label>
                    <input
                        type="file"
                        name="imagen"
                        onChange={leerArchivo}
                    />
                </div>

                <div className="enviar">
                    <input type="submit" className="btn btn-azul"
                           value="Agregar Producto"/>
                </div>
            </form>
        </>
    );
};

export default withRouter(NuevoProducto);
```

Ahora cada vez que se registre un nuevo producto nos mandará a la página de mostrar productos



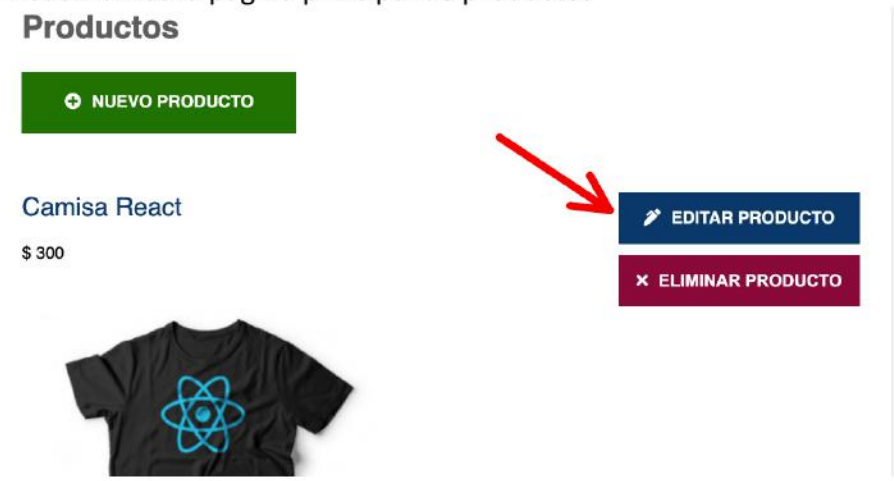
Editar Producto - Obteniendo el Producto a Editar

Cuando se presione el botón de Editar Producto nos debe devolver el producto seleccionado.

Modificar `EditarProducto.jsx`

```
import React, {useEffect, useState} from 'react';
import Swal from 'sweetalert2';
import clienteAxios from '../../config/axios';

import Spinner from '../layout/Spinner';
const EditarProducto = (props) => {
    // obtener el ID
    const {id } = props.match.params;

    // producto = state, y funcion para actualizar
    const [producto, guardarProducto ] = useState({
        nombre: '',
        precio: '',
        imagen : ''
    });

    // archivo = state, guardarArchivo = setState
    const [archivo, guardarArchivo] = useState('');

    // consultar la api para traer el producto a editar
    const consultarAPI = async () => {
        const productoConsulta = await clienteAxios.get(`/productos/${id}`);
        guardarProducto(productoConsulta.data);
        console.log(productoConsulta.data)
    }

    // cuando el componente carga
    useEffect(() => {
        consultarAPI();
    }, [])

    return (
        <>
            <h2>Editar producto</h2>
        </>
    );
};

export default EditarProducto;
```

Visualizamos la página principal de productos



Si hacemos clic en EDITAR PRODUCTO, nos mandará a la página de Editar, en la consola veremos que nos muestra los datos del producto seleccionado. Faltaría mandar esa información a un formulario en la ventana.

Editar Producto - Mostrando la información del Producto a Editar

La siguiente parte es crear el formulario para vaciar los datos de la consulta.

Modificar `EditarProducto.jsx`

```
import React, {useEffect, useState} from 'react';
import Swal from 'sweetalert2';
import clienteAxios from '../../config/axios';
import Spinner from '../layout/Spinner';

const EditarProducto = (props) => {
    // obtener el ID
    const {id} = props.match.params;

    // producto = state, y funcion para actualizar
    const [producto, guardarProducto] = useState({
        nombre: '',
        precio: '',
        imagen: ''
    });

    // archivo = state, guardarArchivo = setState
    const [archivo, guardarArchivo] = useState('');

    // consultar la api para traer el producto a editar
    const consultarAPI = async () => {
        const productoConsulta = await clienteAxios.get(`/productos/${id}`);
```

```
        guardarProducto(productoConsulta.data);
    }

    // cuando el componente carga
    useEffect(() => {
        consultarAPI();
    }, [])

    // extraer los valores del state
    const {nombre, precio, imagen} = producto;

    if (!nombre) return <Spinner/>

    return (
        <>
            <h2>Editar producto</h2>
            <form>
                <legend>Llena todos los campos</legend>

                <div className="campo">
                    <label>Nombre:</label>
                    <input
                        type="text"
                        placeholder="Nombre Producto"
                        name="nombre"

                        defaultValue={nombre}
                    />
                </div>

                <div className="campo">
                    <label>Precio:</label>
                    <input
                        type="number"
                        name="precio"
                        min="0.00"
                        step="0.01"
                        placeholder="Precio"

                        defaultValue={precio}
                    />
                </div>

                <div className="campo">
                    <label>Imagen:</label>
                    {imagen ? (
                        <img src={`http://localhost:4000/${imagen}`} alt="imagen"
                             width="300"/>
                    ) : null}
                    <input
                        type="file"
                        name="imagen"

                    />
                </div>

                <div className="enviar">
                    <input type="submit" className="btn btn-azul"
                           value="Editar Producto"/>
                </div>
            </form>
        </>
```

```
    );
};

export default EditarProducto;
```

Visualizar la página. Ahora cuando hacemos clic sobre el botón EDITAR PRODUCTO, nos lleva a la vista del formulario donde se cargan los valores del producto a editar, incluso su imagen.

Editar Producto - Guardando la Información

Ya solo falta guardar los cambios. Se recupera la información igual que cuando se da de alta un producto, pero en lugar de hacer un POST se pone un PUT.

Modificar `EditarProducto.jsx`

```
import React, {useEffect, useState} from 'react';
import Swal from 'sweetalert2';
import clienteAxios from '../../config/axios';
import Spinner from '../layout/Spinner';
import {withRouter} from 'react-router-dom';

const EditarProducto = (props) => {
    // obtener el ID
    const {id} = props.match.params;

    // producto = state, y funcion para actualizar
    const [producto, guardarProducto] = useState({
        nombre: '',
        precio: '',
        imagen: ''
    });

    // archivo = state, guardarArchivo = setState
    const [archivo, guardarArchivo] = useState('');

    // consultar la api para traer el producto a editar
    const consultarAPI = async () => {
        const productoConsulta = await clienteAxios.get(`/productos/${id}`);
        guardarProducto(productoConsulta.data);
    }

    // cuando el componente carga
    useEffect(() => {
        consultarAPI();
    }, [])

    // extraer los valores del state
    const {nombre, precio, imagen} = producto;

    if (!nombre) return <Spinner/>

    // Edita un Producto en la base de datos
    const editarProducto = async e => {
        e.preventDefault();

        // crear un formdata
        const formData = new FormData();
        formData.append('nombre', producto.nombre);
        formData.append('precio', producto.precio);
        formData.append('imagen', archivo);

        // almacenarlo en la BD
        try {
            const res = await clienteAxios.put(`/productos/${id}`, formData, {
                headers: {
                    'Content-Type': 'multipart/form-data'
                }
            });
```

```
            // Lanzar una alerta
            if (res.status === 200) {
                Swal.fire(
                    'Editado Correctamente',
                    res.data.mensaje,
                    'success'
                )
            }
            // redireccionar
            props.history.push('/productos');

        } catch (error) {
            console.log(error);
            // lanzar alerta
            Swal.fire({
                type: 'error',
                title: 'Hubo un error',
                text: 'Vuelva a intentarlo'
            })
        }
    }

    // leer los datos del formulario
    const leerInformacionProducto = e => {
        guardarProducto({
            // obtener una copia del state y agregar el nuevo
            ...producto,
            [e.target.name]: e.target.value
        })
    }

    // coloca la imagen en el state
    const leerArchivo = e => {
        guardarArchivo(e.target.files[0]);
    }

    return (
        <>
            <h2>Editar producto</h2>
            <form onSubmit={editarProducto}>
                <legend>Llena todos los campos</legend>

                <div className="campo">
                    <label>Nombre:</label>
                    <input
                        type="text"
                        placeholder="Nombre Producto"
                        name="nombre"
                        onChange={leerInformacionProducto}
                        defaultValue={nombre}
                    />
                </div>

                <div className="campo">
                    <label>Precio:</label>
                    <input
                        type="number"
                        name="precio"
                        min="0.00"
                        step="0.01"
                        placeholder="Precio"
                        onChange={leerInformacionProducto}
```

```
                        defaultValue={precio}
                    />
                </div>

                <div className="campo">
                    <label>Imagen:</label>
                    {imagen ? (
                        <img src={`http://localhost:4000/${imagen}`} alt="imagen"
                             width="300"/>
                    ) : null}
                    <input
                        type="file"
                        name="imagen"
                        onChange={leerArchivo}
                    />
                </div>

                <div className="enviar">
                    <input type="submit" className="btn btn-azul"
                           value="Editar Producto"/>
                </div>
            </form>
        </>
    );
};

export default withRouter(EditarProducto);
```

Por ejemplo, tengo el siguiente producto:

Voy a editar nombre, precio e imagen.

Después de guardar, se visualiza:

Fin.