

Universidad Mariano Gálvez

Ingeniera en Sistemas

Ing. José Miguel Villatoro Hidalgo

Desarrollo Web

Manual Técnico de Tienda en línea con React js, Typescript y SASS



Giancarlo André López Silva

9490-18-3141

Elmer Andrée Belloso Nimatuj

9490-17-12488

Índice

Introducción.....	3
Manual Técnico - Tienda en Línea (E-commerce)	4
Requisitos del Sistema.....	4
Estructura del Proyecto	4
Configuración del Proyecto	6
Desarrollo Frontend.....	6
Componentes	6
Estilos	8
Rutas.....	9
AppShopping.tsx	10
AppAdmin.tsx.....	11
Conexión a la API.....	11
Autenticación	12
Carrito de Compras	13
Panel de Administrador.....	13
Dashboard.tsx.....	13
Products.tsx.....	15
Users.tsx	17
LoginAdmin.tsx	19
Seguridad	21
Pruebas.....	21
Documentación	21
Url Repositorio: https://github.com/glopezs18/Proyecto2_9490-18-3141.git.....	21
Render	22
Netlify	22
Credenciales Web App - Netlify.....	23
Conclusión.....	24

Introducción

Este manual técnico proporciona una visión general de la implementación de la tienda en línea (E-commerce) desarrollada con tecnologías web, incluyendo React, SASS/SCSS y TypeScript. El proyecto incluye características de registro, inicio de sesión, catálogo de productos, carrito de compras, perfiles de usuario y compra segura. Además, se presenta un panel de administrador que permite la gestión de productos, usuarios, visualización de clientes y sus compras, así como un resumen detallado de las transacciones.

Manual Técnico - Tienda en Línea (E-commerce)

Requisitos del Sistema

Antes de comenzar a trabajar en el proyecto, asegúrese de contar con los siguientes requisitos del sistema:

- Node.js y npm instalados.
- Acceso a una API previamente desarrollada en Node.js.
 - <https://final-project-api-ibbn.onrender.com/api/test>
- Un entorno de desarrollo React configurado.

Estructura del Proyecto

La estructura del proyecto se organiza de la siguiente manera:

```
Proyecto2_9490-18-3141/  
├─ public/  
│  ├─ index.html  
│  └─ ...  
├─ src/  
│  ├─ admin/  
│  │  ├─ assets/  
│  │  │  ├─ logo-mariano-galvez.png  
│  │  │  └─ mdb-react.png  
│  │  └─ auth/  
│  │     ├─ PrivateRoute.tsx  
│  │     ├─ RestrictedBtn.tsx  
│  │     └─ RestrictedRoute.tsx  
│  │  └─ components/  
│  │     └─ DatatablePage.tsx  
│  │  └─ shared/  
│  │     ├─ HeaderAdmin.tsx  
│  │     └─ Sidebar.tsx  
│  │  └─ styles/  
│  │     ├─ app.admin.scss  
│  │     ├─ dashboard.scss  
│  │     ├─ header-admin.scss  
│  │     ├─ pages.scss  
│  │     ├─ sidebar.scss  
│  │     └─ users.scss
```



```

|   |   | styles/
|   |   |   |─ carrito.scss
|   |   |   |─ footer.scss
|   |   |   |─ header.scss
|   |   |   |─ home.scss
|   |   |   |─ pages.scss
|   |   |   |─ productos.scss
|   |   |   |─ profile.scss
|   |   |   |─ shopping-cart.scss
|   |   |   |─ variables.scss
|   |   | types/
|   |   |   |─ user.type.ts
|   | App.scss
|   | App.tsx
|   | index.scss
|   | index.tsx
|   | react-app-env.ts
|   | reportWebVitals.ts
|─ package-lock.json
|─ package.json
|─ tsconfig.json
|─ README.md

```

Configuración del Proyecto

Antes de comenzar, asegúrese de configurar las variables de entorno en un `EnvConfig.ts`. Aquí es donde almacenaremos información sensible, como la URL de la API y las credenciales necesarias.

```

export default function EnvConfig() {
  const env: any = "prod";

  if (env == "prod") {
    return { BASE_URL: "https://final-project-api-ibbn.onrender.com/api", IMAGE_URL: "https://final-project-api-ibbn.onrender.com/images/" };
  } else {
    return { BASE_URL: "http://localhost:3000/api", IMAGE_URL: "http://localhost:3000/images/" };
  }
}

```


Desarrollo Frontend

Componentes

Los componentes son la base de la aplicación. Se han desarrollado componentes reutilizables para diversas partes de la aplicación, como la lista de productos, el carrito de compras, el perfil de usuario, entre otros.

Como por ejemplo `src/admin/components/DatatablePage.tsx` que se creó para su posterior reutilización en la muestra de los productos disponibles, usuarios creados y detalles de compras.

tsx

 Copy code

```
// Ejemplo de componente DatatablePage.tsx
import React, { useState, useEffect } from 'react';
import { MDBDataTableV5 } from 'mdbreact';

const DatatablePage = (props: any) => {
  const res = props.data;
  const columnsProducts = props.columns;
  const rowsProducts: Array = [];

  res.map((item: any) => {
    rowsProducts.push(item)
  })

  const newList: any = {
    columns: columnsProducts.columns,
    rows: rowsProducts
  }

  const [datatable, setDatatable]: any = useState(newList);

  return (

  );
}

export default DatatablePage;
```

Y así también los siguientes componentes.

```
| | | — shared/
| | |   | — HeaderAdmin.tsx
| | |   | — Sidebar.tsx
```

```
| | | — auth/
| | |   | — PrivateRoute.tsx
| | |   | — RestrictedBtn.tsx
| | |   | — RestrictedRoute.tsx
```

```
| | └─ auth/
| |   └─ AuthHeaders.tsx
| |   └─ PrivateRoute.tsx
| |   └─ ResctrictedBtn.tsx
| |   └─ RestrictedCartBtn.tsx
| |   └─ RestrictedRoute.tsx
```

```
| | └─ Footer.tsx
| | └─ Header.tsx
```

Estilos

La aplicación utiliza SASS/SCSS para gestionar los estilos. Se han dividido los estilos en archivos separados y se han importado en los componentes correspondientes.

```
/* Ejemplo de estilos en footer.scss */
@import "variables.scss";

.footer {
  background: $bg-footer;
  width: 100%;
  position: relative;

  .content-footer {
    position: absolute;
    width: 100%;
    top: 30px;

    h3 {
      text-align: center;
      color: #fff;
      text-transform: uppercase;
      font-weight: 200;
      font-size: 15px;
    }
  }

  ul {
    li {
      a {
        color: #fff !important;
      }
    }
  }

  p {
    color: #fff !important;
  }
}
```


Y así también los siguientes estilos.

Estilos del dashboard de administrador:

```
| | | | styles/
| | | |   app.admin.scss
| | | |   dashboard.scss
| | | |   header-admin.scss
| | | |   pages.scss
| | | |   sidebar.scss
| | | |   users.scss
```

Estilos tienda en línea:

```
| | | | styles/
| | | |   carrito.scss
| | | |   footer.scss
| | | |   header.scss
| | | |   home.scss
| | | |   pages.scss
| | | |   productos.scss
| | | |   profile.scss
| | | |   shopping-cart.scss
| | | |   variables.scss
```

Rutas

La navegación en la aplicación se gestiona utilizando **react-router-dom** para permitir el acceso a diferentes páginas, como el catálogo de productos, el carrito de compras, el perfil de usuario y la finalización de la compra.

```
import React, { Fragment, useState } from 'react';
import { BrowserRouter, Route, Routes } from 'react-router-dom';
import AppShopping from './pages/AppShopping';
import AppAdmin from './admin/AppAdmin';

import './App.scss';

function App() {
  const [loggedIn, setLoggedIn] = useState(false);
  return (
    <div className="App">
      <BrowserRouter>
        <Fragment>
          <Routes>
            <Route path="*" element={<AppShopping />} />
            <Route path="/admin/*" element={<AppAdmin />} />
          </Routes>
        </Fragment>
      </BrowserRouter>
    </div>
  );
}

export default App;
```

AppShopping.tsx

```
import React, { Fragment, useState } from 'react';
import Header from '../components/Header'
import Footer from '../components/Footer'
import { BrowserRouter, Route, Routes } from 'react-router-dom';
import Home from './Home'
import Login from './Login'
import Register from './Register'
import Profile from './Profile'
import ShoppingCart from './ShoppingCart'
import PrivateRoute from "../auth/PrivateRoute";
import RestrictedRoute from "../auth/RestrictedRoute";

import './App.scss';

function AppShopping() {
  const [loggedIn, setLoggedIn] = useState(false);
  return (
    <div className="App">
      { /* <BrowserRouter> */ }
      <Fragment>
        <Header />
        <Routes>
          <Route path="/" element={ <Home /> } />
          <Route path="/login" element={ <RestrictedRoute /> }>
            <Route path="/login" element={ <Login /> } />
          </Route>
          <Route path="/register" element={ <RestrictedRoute /> }>
            <Route path="/register" element={ <Register /> } />
          </Route>
          <Route path="/profile" element={ <PrivateRoute /> }>
            <Route path="/profile" element={ <Profile /> } />
          </Route>
          <Route path="/shopping-cart" element={ <PrivateRoute /> }>
            <Route path="/shopping-cart" element={ <ShoppingCart /> } />
          </Route>
        </Routes>
        <Footer />
      </Fragment>
    { /* </BrowserRouter> */ }
  </div>
  );
}

export default AppShopping;
```

AppAdmin.tsx

```
import React, { Fragment, useState } from 'react';
import Header from '../components/Header';
import Footer from '../components/Footer';
import { BrowserRouter, Route, Routes } from 'react-router-dom';
import '@fortawesome/fontawesome-free/css/all.min.css';
import 'bootstrap-css-only/css/bootstrap.min.css';
import 'mdbreact/dist/css/mdb.css';
import DashboardApp from './DashboardApp';
import LoginAdmin from './LoginAdmin';
import PrivateRoute from './auth/PrivateRoute';
import RestrictedRoute from './auth/RestrictedRoute';

import './styles/app-admin.scss';

function AppAdmin() {

  return (
    <div>
      { /* <BrowserRouter> */ }
      <Fragment>
        <Routes>
          <Route path='/login' element={<RestrictedRoute />>
            <Route path='/login' element={<LoginAdmin /> } />
          </Route>
          <Route path='/' element={<RestrictedRoute />>
            <Route path='/' element={<LoginAdmin /> } />
          </Route>
          <Route path='/*' element={<DashboardApp /> } />
        </Routes>
      </Fragment>
      { /* </BrowserRouter> */ }
    </div>
  );
}

export default AppAdmin;
```

Conexión a la API

La aplicación se conecta a la API previamente desarrollada en Node.js. Se utilizan solicitudes HTTP para obtener información sobre productos, usuarios y procesar compras.

Partimos de nuestro archivo de environment para configurar una variable global, es aquí donde se detalla la url de la API

```
export default function EnvConfig() {
  const env: any = "prod";

  if (env == "prod") {
    return { BASE_URL: "https://final-project-api-ibbn.onrender.com/api", IMAGE_URL: "https://final-project-api-ibbn.onrender.com/images/" };
  } else {
    return { BASE_URL: "http://localhost:3000/api", IMAGE_URL: "http://localhost:3000/images/" };
  }
}
```

Y el código que se utilizó para las solicitudes de la tienda en línea a los diversos servicios de la API mencionada anteriormente.

Variable de componente, hace referencia a la variable global.

```
const BASE_URL = EnvConfig().BASE_URL;
```

Solicitud a API, se utilizó el módulo axios para realizar las solicitudes a la API.

```
axios
  .post(BASE_URL + '/login', params)
  .then(function (response) {
    // IF EMAIL ALREADY EXISTS
    if (response.data.success === false) {
      toast.error(response.data.message, {
        position: "top-right",
        autoClose: 3000,
        hideProgressBar: true,
        closeOnClick: true,
        pauseOnHover: true,
        draggable: false,
        progress: 0,
        toastId: "my_toast",
      });
    } else {
      toast.success(response.data.message, {
        position: "top-right",
        autoClose: 3000,
        hideProgressBar: true,
        closeOnClick: true,
        pauseOnHover: true,
        draggable: false,
        progress: 0,
        toastId: "my_toast",
      });
      localStorage.setItem("auth", response.data.token);
      localStorage.setItem("id_user", response.data.result._id);
      localStorage.setItem("user", response.data.result.user);
      localStorage.setItem("rol", response.data.result.rol);
      localStorage.setItem("nombre", response.data.result.nombres + " " + response.data.result.apellidos);
      localStorage.setItem("dpi", response.data.result.dpi);
      setTimeout(() => {
        navigate("/");
      }, 3000);
    }
  })
  .catch(function (error) {
    console.log(error);
  });
```

Autenticación

La autenticación de usuarios se gestiona a través de tokens JWT para mantener a los usuarios autenticados y almacenar su información en el estado de la aplicación.

Al momento de que la autenticación de inicio de sesión fuera correcta, se guardaba el token obtenido desde la API en una variable de local.storage, del lado del navegador.

```
localStorage.setItem("auth", response.data.token);
```

Para su posterior utilización.

```
export default function AuthHeaders() {
  const token = (localStorage.getItem("auth")) ? localStorage.getItem("auth") : null;

  if (token) {
    return { Authorization: 'Bearer ' + token }; // for Spring Boot back-end
    // return { 'x-access-token': user.accessToken }; // for Node.js Express back-end
  } else {
    return { Authorization: '' }; // for Spring Boot back-end
    // return { 'x-access-token': null }; // for Node Express back-end
  }
}
```

Carrito de Compras

El carrito de compras permite a los usuarios agregar y eliminar productos, así como calcular el total de la compra. Se utiliza el estado de la aplicación para mantener la información del carrito.

```
import React, { Component, useState } from 'react';
import ListGroup from 'react-bootstrap/ListGroup';
import '../styles/carrito.scss'

function Carrito(props: any) {

  const cartItems = props.items.map((item: any) => (
    <ListGroup className="lista-carrito" key={item._id}>
      <ListGroup.Item data-id={item._id}>{item.nombre} - Q {item.precio} x {item.quantity}
        <button
          className="btn btn-danger ml-2 button-borrar"
          onClick={() => props.onRemoveFromCart(item)}
        >
          Eliminar
        </button>
      </ListGroup.Item>
    </ListGroup>
  ));

  return (
    <div className="cart">
      <h2>Carrito de compras</h2>
      <ul>{cartItems}</ul>
      <h2>Total: Q {props.total}</h2>
    </div>
  );
}

export default Carrito;
```

Panel de Administrador

El panel de administrador permite la gestión de productos, usuarios, la visualización de clientes y sus compras, así como un resumen detallado de las transacciones. Asegúrese de proteger las funciones del administrador y limitar el acceso a usuarios con permisos adecuados.

Dashboard.tsx

Contiene el código que se utilizó para el resumen de compras dentro de la página web de la tienda en línea.

```

import React, { FC } from "react";
import { useState, useEffect } from "react";
import Modal from "react-bootstrap/Modal";
import Container from "react-bootstrap/Container";
import Row from "react-bootstrap/Row";
import Col from "react-bootstrap/Col";
import EnvConfig from "../env/EnvConfig";
import DataTablePage from "../components/DataTablePage";
import AdminCardSection from "../AdminCardSection";
import axios from "axios";
import "../styles/dashboard.scss";

const Dashboard = () => {
  const BASE_URL = EnvConfig().BASE_URL;
  const IMAGE_URL = EnvConfig().IMAGE_URL;
  const [saleId, setSaleId] = useState([] as any);
  const [sales, setSales] = useState([] as any);
  const [salesDetails, setSalesDetails] = useState([] as any);
  const [saleUser, setSaleUser] = useState([] as any);
  const [saleUserTotal, setSaleUserTotal] = useState([] as any);
  const [saleTotalInfo, setSaleTotalInfo] = useState(0);
  const [products, setProducts] = useState([] as any);
  let sumTotals = 0;
  const [show, setShow] = useState(false);

  const handleClose = () => setShow(false);
  const handleShow = () => setShow(true);
  const handleEdit = (_id: any) => {
    // console.log("sale", _id);
    setSaleId(_id);
    axios
      .get(BASE_URL + "/compra/" + _id, { headers: { Authorization: "Bearer " + localStorage.getItem("authAdmin") } })
      .then((res) => {
        if (res.data.success) {
          setSaleUser(res.data.result.nombre_usuario);
          setSaleUserTotal(res.data.result.total);
          const rowsSales: Array<any> = [];
          console.log(res);
          setShow(true);
          res.data.result.datosCarrito.productos.map((item: any) => {

```

```

const getProductsData = () => {
  axios
    .get(BASE_URL + "/productos", { headers: { Authorization: "Bearer " + localStorage.getItem("authAdmin") } })
    .then((res) => {
      setProducts(res.data);
    });
};

return (
  <div style={{ display: "flex" }}>
    <Container>
      <Row>
        <Col sm={8}>
          <h3 className="m-3 text-left fw-bold">Dashboard</h3>
        </Col>
      </Row>
    </Container>
    <div className="container">
      <div className="row d-flex justify-content-center align-items-center text-center">
        <AdminCardSection saleTotalInfo={saleTotalInfo} salesLength={sales.length} productsLength={products.length}>
          <hr />
          <DataTablePage columns={columnsSales} data={sales} />
        </AdminCardSection>
        <Modal dialogClassName="modal-90w" aria-labelledby="example-custom-modal-styling-title" show={show} onHide={handleClose}>
          <Modal.Header closeButtons>
            <Modal.Title>Detalle de compra: {saleId}</Modal.Title>
          </Modal.Header>
          <Modal.Body>
            <h5 className="title-sale-client-detail">Cliente: <strong>{saleUser}</strong></h5>
            <DataTablePage columns={columnsSalesDetails} data={salesDetails} />
            <hr />
            <div className="px-md-0 px-1 text-center" id="footer-font">
              <b className="pl-md-4">TOTAL COMPRADO:<span className="pl-md-4"> Q{saleUserTotal}</span></b>
            </div>
          </Modal.Body>
        </Modal>
      </div>
    </div>
  </div>
);
export default Dashboard;

```

Products.tsx

Contiene el código utilizado para la gestión de los productos.

```
import React, { FC } from "react";
import { useState, useEffect } from "react";
import Button from "react-bootstrap/Button";
import Form from "react-bootstrap/Form";
import Modal from "react-bootstrap/Modal";
import Container from "react-bootstrap/Container";
import Row from "react-bootstrap/Row";
import Col from "react-bootstrap/Col";
import DatatablePage from "../components/DatatablePage";
import AuthHeaders from "../auth/AuthHeaders";
import EnvConfig from "../env/EnvConfig";
import { ToastContainer, toast, Flip } from "react-toastify";

import "react-toastify/dist/ReactToastify.min.css";
import "../styles/pages.scss";
import axios from "axios";

const Products = () => {
  const [edit, setEdit] = useState(false);
  const [show, setShow] = useState(false);
  const [products, setProducts] = useState([] as any);
  const [validated, setValidated] = useState(false);
  const [file, setFile]: any = useState();

  const [idProducto, setIdProducto]: any = useState();
  const [nombreProducto, setNombreProducto]: any = useState();
  const [marcaProducto, setMarcaProducto]: any = useState();
  const [dispProducto, setDispProducto]: any = useState();
  const [descProducto, setDescProducto]: any = useState();
  const [precioProducto, setPrecioProducto]: any = useState();
  const [precioDescProducto, setPrecioDescProducto]: any = useState();
  const [imagenProducto, setImagenProducto]: any = useState();
  const [descripcionProducto, setDescripcionProducto]: any = useState();
  const [categoriaProducto, setCategoriaProducto]: any = useState();

  const BASE_URL = EnvConfig().BASE_URL;
  const IMAGE_URL = EnvConfig().IMAGE_URL;

  const handleClose = () => {
    setShow(false);
    clearForm();
    setEdit(false);
  };
  const handleShow = () => setShow(true);
```

```

const handleEdit = (_id: any) => {
  console.log("product", _id);
  axios
    .get(BASE_URL + '/producto/' + _id, { headers: { Authorization: 'Bearer ' + localStorage.getItem("authAdmin") } })
    .then((res) => {
      // console.log(res);
      if (res.data.success) {
        setEdit(true);
        setShow(true);
        setIdProducto(_id);
        setNombreProducto(res.data.result.nombre);
        setMarcaProducto(res.data.result.marca);
        setDispProducto(res.data.result.disponibilidad);
        setDescProducto(res.data.result.descuento);
        setPrecioProducto(res.data.result.precio);
        setPrecioDescProducto(res.data.result.precioDescuento);
        setImagenProducto(res.data.result.imagen);
        setDescriptionProducto(res.data.result.descripcion);
        setCategoriaProducto(res.data.result.categorias);
      }
    })
    .catch(er => console.log(er));
};

const handleDelete = (_id: any) => {
  console.log("delete product", _id);
  axios
    .delete(BASE_URL + '/producto/' + _id, { headers: { Authorization: 'Bearer ' + localStorage.getItem("authAdmin") } })
    .then((res) => {
      setShow(false);
      if (res.data.success) {
        toast.success(res.data.message, {
          position: "top-right",
          autoClose: 3000,
          hideProgressBar: true,
          closeOnClick: true,
          pauseOnHover: true,
          draggable: false,
          progress: 0,
          toastId: "my_toast",
        });
      }
    })
    .catch(er => console.log(er));
};

```

```

return (
  <>
    <div style={{ display: "flex" }}>
      <Container>
        <Row>
          <Col sm={8}>
            <h3 className="m-3 text-left fw-bold">Productos</h3>
          </Col>
          <Col sm={4}>
            <Button className="float-right" variant="primary" onClick={handleShow}>
              Agregar nuevo producto
            </Button>
          </Col>
        </Row>
      </Container>
    </div>
    <div className="container">
      <div className="row d-flex justify-content-center align-items-center text-center">
        <hr />
        <DatatablePage columns={columnsProducts} data={products} />
        <Modal show={show} onHide={handleClose}>
          <Modal.Header closeButton>
            <Modal.Title>{(edit) ? 'Editar producto' : 'Agregar Nuevo Producto'}</Modal.Title>
          </Modal.Header>
          <Modal.Body>
            <Form noValidate validated={validated} onSubmit={handleSubmit}>
              <Form.Group as={Col} md="12" className="mb-3" controlId="validationCustom01">
                <Form.Label>Producto</Form.Label>
                <Form.Control
                  required
                  type="text"
                  name="nombre"
                  placeholder="Nombre de Producto"
                  autoFocus
                  defaultValue={nombreProducto}
                />
                <Form.Control.Feedback type="invalid">Nombre de producto requerido</Form.Control.Feedback>
              </Form.Group>
              <Form.Group as={Col} md="12" className="mb-3" controlId="validationCustom02">
                <Form.Label>Marca</Form.Label>
                <Form.Control
                  required
                  type="text"
                  name="marca"
                  placeholder="Nombre de Marca"
                />
              </Form.Group>
            </Form>
          </Modal.Body>
        </Modal>
      </div>
    </div>
  </>
)

```



```

        <Form.Control required name="descripcion" as="textarea" defaultValue={descripcionProducto} rows={3} />
        <Form.Control.Feedback type="invalid">Descripción de producto requerido!</Form.Control.Feedback>
      </Form.Group>
      <Form.Group as={Col} md="12" className="mb-3" controlId="validationCustom08">
        <Form.Label>Categoría</Form.Label>
        <Form.Control
          required
          type="text"
          name="categorias"
          placeholder="Categoría de producto"
          defaultValue={categoriaProducto}
          autoFocus
        />
        <Form.Control.Feedback type="invalid">Categoría de producto requerido!</Form.Control.Feedback>
      </Form.Group>
      <Modal.Footer style={{ display: "block" }}>
        {
          (edit) ? <Button variant="danger" onClick={() => handleDelete(idProducto)}> Eliminar </Button> : <Button variant="blue-grey" onClick={handleClose}> Cerrar </Button>
        }
        <Button style={{ float: "right" }} variant="primary" type="submit">
          Guardar
        </Button>
      </Modal.Footer>
    </Form>
  </Modal.Body>
</Modal>
</div>
</div>
<ToastContainer
  position="top-right"
  autoClose={5000}
  hideProgressBar
  closeOnClick
  rtl={false}
  pauseOnFocusLoss={false}
  draggable={false}
  pauseOnHover
  limit={1}
  transition={Flip}
/>
</>
);
};
export default Products;

```

Users.tsx

Contiene el código utilizado para la gestión de usuarios y roles.

```

import React, { FC } from "react";
import { useState, useEffect } from "react";
import Button from "react-bootstrap/Button";
import Form from "react-bootstrap/Form";
import Modal from "react-bootstrap/Modal";
import Container from "react-bootstrap/Container";
import Row from "react-bootstrap/Row";
import Col from "react-bootstrap/Col";
import DatatablePage from "../components/DatatablePage";
import AuthHeaders from "../auth/AuthHeaders";
import EnvConfig from "../env/EnvConfig";
import axios from "axios";
import { ToastContainer, toast, Flip } from "react-toastify";

import "react-toastify/dist/ReactToastify.min.css";
import "../styles/pages.scss";
import "../styles/users.scss";

const Users = () => {
  const [edit, setEdit] = useState(false);
  const [show, setShow] = useState(false);
  const [users, setUsers] = useState([] as any);
  const [validated, setValidated] = useState(false);

  const [idUserio, setIdUsuario]: any = useState();
  const [nombreUsuario, setNombreUsuario]: any = useState();
  const [apellidosUsuario, setApellidosUsuario]: any = useState();
  const [dpiUsuario, setDpiUsuario]: any = useState();
  const [fechaNacimientoUsuario, setFechaNacimientoUsuario]: any = useState();
  const [claveUsuario, setClaveUsuario]: any = useState();
  const [validacionClaveUsuario, setValidacionClaveUsuario]: any = useState();
  const [nitUsuario, setNitUsuario]: any = useState();
  const [numeroTelefonicoUsuario, setNumeroTelefonicoUsuario]: any = useState();
  const [correoElectronicoUsuario, setCorreoElectronicoUsuario]: any = useState();
  const [rolUsuario, setRolUsuario]: any = useState();
  const [userName, setUserName]: any = useState();

  const BASE_URL = EnvConfig().BASE_URL;

  const handleClose = () => {
    setShow(false);
    clearForm();
    setEdit(false);
  };
};

```

```

const handleShow = () => setShow(true);
const handleEdit = (_dpi: any) => {
  console.log("user", _dpi);
  axios
    .get(BASE_URL + '/perfil/' + _dpi, { headers: { Authorization: 'Bearer ' + localStorage.getItem("authAdmin") } })
    .then((res) => {
      // console.log(res);

      if (res.data.success) {
        setEdit(true);
        setShow(true);
        setIdUsuario(_dpi);
        setNombreUsuario(res.data.result.nombres);
        setApellidosUsuario(res.data.result.apellidos);
        setDpiUsuario(res.data.result.dpi);
        setFechaNacimientoUsuario(new Date(res.data.result.fechaNacimiento).toISOString().substr(0, 10));
        setClaveUsuario(res.data.result.clave);
        setValidacionClaveUsuario(res.data.result.validacionClave);
        setNitUsuario(res.data.result.nit);
        setNumeroTelefonicoUsuario(res.data.result.numeroTelefonico);
        setCorreoElectronicoUsuario(res.data.result.correoElectronico);
        setRolUsuario(res.data.result.rol);
        setUsername(res.data.result.user);
      }
    })
    .catch(er => console.log(er));
};

const handleDelete = (_dpi: any) => {
  console.log("delete user", _dpi);
  axios
    .delete(BASE_URL + '/perfil/' + _dpi, { headers: { Authorization: 'Bearer ' + localStorage.getItem("authAdmin") } })
    .then((res) => {
      setShow(false);
      if (res.data.success) {
        toast.success(res.data.message, {
          position: "top-right",
        });
      }
    });
};

```

```

return (
  <>
    <div style={{ display: "flex" }}>
      <Container>
        <Row>
          <Col sm={8}>
            <h3 className="m-3 text-left fw-bold">Usuarios</h3>
          </Col>
          <Col sm={4}>
            <Button className="float-right" variant="primary" onClick={handleShow}>
              Agregar nuevo usuario
            </Button>
          </Col>
        </Row>
      </Container>
    </div>
    <div className="container">
      <div className="row d-flex justify-content-center align-items-center text-center">
        <hr />
        <DatatablePage columns={columnsUsers} data={users} />
        <Modal size="lg" show={show} onHide={handleClose}>
          <Modal.Header closeButton>
            <Modal.Title>{(edit) ? 'Editar Usuario' : 'Agregar Nuevo Usuario'}</Modal.Title>
          </Modal.Header>
          <Modal.Body>
            <Form noValidate validated={validated} onSubmit={handleSubmit}>
              <Row className="mb-3">
                <Form.Group as={Col} md="6" className="mb-3" controlId="validationCustom01">
                  <Form.Label>Nombre</Form.Label>
                  <Form.Control
                    required
                    type="text"
                    name="nombres"
                    placeholder="Escribe tus nombres"
                    autoFocus
                    defaultValue={nombreUsuario}
                  />
                  <Form.Control.Feedback type="invalid">Nombre de usuario requerido!</Form.Control.Feedback>
                </Form.Group>
                <Form.Group as={Col} md="6" className="mb-3" controlId="validationCustom02">
                  <Form.Label>Apellido</Form.Label>
                  <Form.Control

```

```

        </Form.Group>
        <Form.Group as={Col} md="6" className="mb-3" controlId="validationCustom11">
          <Form.Label>Usuario</Form.Label>
          <Form.Control
            required
            type="text"
            name="user"
            placeholder="Escribe un nombre de usuario"
            autoFocus
            defaultValue={userName}
          />
          <Form.Control.Feedback type="invalid">Nombre de usuario requerido!</Form.Control.Feedback>
        </Form.Group>
        <Modal.Footer style={{ display: "block" }}>
          {
            (edit) ? <Button variant="danger" onClick={() => handleDelete(idUsuario)}> Eliminar </Button> : <Button variant="blue-grey" onClick={handleClose}> Cerrar </Button>
          }
          <Button style={{ float: "right" }} variant="primary" type="submit">
            Guardar
          </Button>
        </Modal.Footer>
      </Row>
    </Form>
  </Modal.Body>
</Modal>
</div>
</div>
</div>
<ToastContainer
  position="top-right"
  autoClose={5000}
  hideProgressBar
  closeOnClick
  rtl={false}
  pauseOnFocusLoss={false}
  draggable={false}
  pauseOnHover
  limit={1}
  transition={Flip}
/>
</>
);
};
export default Users;

```

LoginAdmin.tsx

Contiene el código utilizado para la validación de credenciales para el área de administrador.

```

import React from "react";
import { Link, useNavigate } from "react-router-dom";
import { useForm } from "react-hook-form";
import axios from "axios";
import EnvConfig from '../env/EnvConfig'
import { ToastContainer, toast, Flip } from "react-toastify";
import "react-toastify/dist/ReactToastify.min.css";
const LoginAdmin = () => {
  let navigate = useNavigate();
  const BASE_URL = EnvConfig().BASE_URL;

  const {
    register,
    handleSubmit,
    formState: { errors },
  } = useForm();

  const login = (data: any) => {
    let params = {
      username: data.user,
      clave: data.clave,
    };

    axios
      .post(BASE_URL + '/login', params)
      .then(function (response) {
        // IF EMAIL ALREADY EXISTS
        if (response.data.success === false) {
          toast.error(response.data.error, {
            position: "top-right",
            autoClose: 3000,
            hideProgressBar: true,
            closeOnClick: true,
            pauseOnHover: true,
            draggable: false,
            progress: 0,
            toastId: "my_toast",
          });
        } else {
          if (response.data.result.rol == "admin") {
            toast.success(response.data.message, {
              position: "top-right",
              autoClose: 3000,
              hideProgressBar: true,
              closeOnClick: true,
              pauseOnHover: true,
              draggable: false,
            });
          }
        }
      });
  };

```

```

return (
  <>
    <div className="container">
      <div
        className="row d-flex justify-content-center align-items-center"
        style={{ height: "100vh" }}>
        <div className="card mb-3" style={{ maxWidth: "460px", height: "380px"}}>
          <div className="col-md-12">
            <div className="card-body">
              <h3 className="card-title text-center text-secondary mt-3">
                Login Form
              </h3>
              <form autoComplete="off" onSubmit={handleSubmit(login)}>
                <div className="mb-3 mt-4">
                  <label className="form-label">User</label>
                  <input
                    type="text"
                    className="form-control shadow-none"
                    id="exampleFormControlInput1"
                    {...register("user", { required: "User is required!" })}>
                  </input>
                  {errors.user && (
                    <p className="text-danger" style={{ fontSize: 14 }}>
                      error usuario
                    </p>
                  )}
                </div>
                <div className="mb-3">
                  <label className="form-label">Contraseña</label>
                  <input
                    type="password"
                    className="form-control shadow-none"
                    id="exampleFormControlInput2"
                    {...register("clave", {
                      required: "Password is required!",
                    })}>
                  </input>
                  {errors.clave && (
                    <p className="text-danger" style={{ fontSize: 14 }}>
                      error clave
                    </p>
                  )}
                </div>
                <div className="text-center mt-4">
                  <button
                    className="btn btn-outline-primary text-center shadow-none mb-3"

```

```

                    </button>
                    <button
                      className="btn btn-outline-primary text-center shadow-none mb-3"
                      type="submit"
                    >
                      Submit
                    </button>
                  <p className="card-text pb-2">
                    Have an Account?{" "}>
                    <Link style={{ textDecoration: "none" }} to={"/register"}>
                      Sign Up
                    </Link>
                  </p>
                </div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </div>
    <ToastContainer
      position="top-right"
      autoClose={5000}
      hideProgressBar
      closeOnClick
      rtl={false}
      pauseOnFocusLoss={false}
      draggable={false}
      pauseOnHover
      limit={1}
      transition={Flip}
    />
  </div>
);
};
export default LoginAdmin;

```

Seguridad

La seguridad es una prioridad. Todas las transacciones y la información del usuario se manejan de manera segura. Se utiliza HTTPS para comunicaciones y se valida la entrada del usuario en el servidor.

Pruebas

Se han realizado pruebas unitarias y de integración para garantizar que la aplicación funcione correctamente y cumpla con los requisitos.

Documentación

A continuación, detallo el proceso de instalación y uso del proyecto mencionado anteriormente.

Url Repositorio: https://github.com/glopezs18/Proyecto2_9490-18-3141.git

Requisitos del Proyecto

- Node.js y npm instalados.

Instrucciones de Instalación

Clona el repositorio

```
git clone <URL_DEL_REPOSITORIO>
```

Navega al directorio del proyecto.

```
cd nombre_del_proyecto
```

Instala todas las dependencias necesarias que el proyecto en react necesita para poder funcionar bien. (Indicados en package.json)

```
npm install
```

Inicia la aplicación Ejecuta la aplicación en modo de desarrollo.\

```
npm start
```

Abra <http://localhost:3000> para verlo en su navegador. La página se recargará cuando realice cambios. También es posible que vea errores de pelusa en la consola.

Render

Se procedió a subir la API como bien se indicó en los requisitos del proyecto a Render, adjunto captura.

<https://final-project-api-ibbn.onrender.com>

<https://final-project-api-ibbn.onrender.com/api/test>

The screenshot shows the Render dashboard for a project named 'final_project_api'. The top navigation bar includes links for Dashboard, Blueprints, Env Groups, Docs, Community, and Help. A user profile for 'Giancarlo López' is visible in the top right. The project is currently on the 'Node' blueprint and is 'Free'. A 'Connect' button and a 'Manual Deploy' button are present. Below the project name, the repository is identified as 'glopezs18 / Proyecto1_9490-18-3141' on the 'main' branch, with the URL 'https://final-project-api-ibbn.onrender.com'. The left sidebar lists various management options: Events, Logs, Disks, Environment, Shell, Previews, Jobs, Metrics, Scaling, and Settings. The main area displays a message about free instance types spinning down with inactivity, followed by a 'Live' status for an update on October 28, 2023, at 6:41 PM. Below this is a 'Search logs' section with a search bar and a 'Maximize' button. The log output shows the deployment process: 'nodeemon' starting, watching paths and extensions, starting 'node index.js', connecting to 'mongotlas', and the server running on port 3000. Multiple 'Build successful' messages are also visible. A 'Scroll to bottom' button is at the bottom right.

Netlify

Se procedió a subir el proyecto de React (Tienda en línea) a Netlify. Adjunto captura

<https://proyecto2-react-dw.netlify.app/>

The screenshot shows the Netlify dashboard for a project named 'proyecto2-react-dw'. The top navigation bar includes links for Site overview, Site configuration, Deploys, Logs, Integrations, Metrics, Domain management, and Forms. A search bar and links for News and Support are also present. The 'Deploys' section is active, showing a deployment log with four steps: Building, Deploying, Cleanup, and Post-processing, all marked as 'Complete'. Below the log is a 'Deploy file browser' section with a search bar and a list of files. The files listed are: 'static' (25 files), '_redirects.html' (18 B), 'asset-manifest.json' (2.2 KB), 'favicon.ico' (3.8 KB), 'index.html' (649 B), 'logo192.png' (5.2 KB), 'logo512.png' (9.4 KB), and 'manifest.json' (492 B). Each file has a 'Download' link. An 'Upgrade' button is visible at the bottom left.

Credenciales Web App - Netlify

Url: <https://proyecto2-react-dw.netlify.app/admin/login>

Usuario: glopezs18

Contraseña: adminUmg

Conclusión

Este proyecto de tienda en línea (E-commerce) ha sido desarrollado con éxito, proporcionando a los usuarios una experiencia de compra en línea segura y agradable. El uso de tecnologías como React, SASS/SCSS y TypeScript ha garantizado un desarrollo eficiente y de alta calidad.

Este manual proporciona una visión general del proyecto de tienda en línea, pero debe personalizarse con imágenes y código específico según las necesidades de tu aplicación. ¡Éxito en tu proyecto!