

Manual Técnico de la Aplicación Web

1. Introducción

Este manual técnico describe la arquitectura, estructura, funcionamiento interno y decisiones técnicas de la aplicación web desarrollada con **React**, orientada a comercio electrónico con blog, carrito de compras y autenticación de usuarios.

La aplicación está diseñada bajo una arquitectura **SPA (Single Page Application)** utilizando **React Router**, consumo de API REST con **Axios** y manejo de estado local mediante **Hooks**.

2. Tecnologías Utilizadas

Frontend

- **React 18**
- **React Router DOM v6**
- **Axios**
- **CSS modularizado**

Backend (consumido)

- **Node.js + Express**
 - **API REST**
 - **Base de datos MySQL**
 - **Autenticación con JWT**
-

3. Arquitectura General

La aplicación sigue una arquitectura basada en componentes:

- **App.jsx**: componente raíz y controlador principal
- **Components/**: componentes reutilizables y vistas
- **Comunicación con backend** mediante peticiones HTTP

El flujo general es:

Usuario → React Router → Componente → Axios → API → Respuesta → UI

4. Estructura del Proyecto

```
src/
├── components/
│   ├── Header.jsx
│   ├── Footer.jsx
│   ├── Home.jsx
│   ├── Products.jsx
│   ├── Services.jsx
│   ├── About.jsx
│   ├── Blog.jsx
│   ├── Contact.jsx
│   ├── Policies.jsx
│   ├── Cart.jsx
│   ├── AccountModal.jsx
│   ├── Profile.jsx
│   └── Checkout.jsx
├── App.jsx
├── main.jsx
└── index.css
```

5. Componente Principal: App.jsx

App.jsx es el núcleo de la aplicación y se encarga de:

- Configurar **React Router**
- Gestionar estados globales
- Mostrar/ocultar modales
- Controlar sesión de usuario
- Administrar carrito de compras

Estados Principales

Estado	Descripción
currentSection	Controla la sección activa en Home
cart	Productos agregados al carrito
user	Usuario autenticado
showCart	Controla el modal del carrito
showAccountModal	Controla el modal de login/registro
blogPosts	Posts del blog

6. Navegación y Rutas

Se utiliza **BrowserRouter** con las siguientes rutas:

Ruta	Descripción
/	Página principal (Home + secciones)
/checkout	Proceso de pago (protegido)

La navegación interna del Home se controla mediante el estado `currentSection`.

7. Manejo de Autenticación

Flujo de Login

1. Usuario abre `AccountModal`
2. Ingresa credenciales
3. Backend valida usuario
4. Se guarda el token en `localStorage`
5. Se almacena el usuario en el estado `user`

Logout

- Se elimina el token del `localStorage`
 - Se limpia el estado `user`
 - Se redirige al inicio
-

8. Carrito de Compras

El carrito se maneja como un arreglo de objetos:

- Cada producto incluye `quantity`
- Se evita duplicar productos
- Se incrementa cantidad si el producto ya existe

Funciones Clave

- `addToCart(product)`
 - `setCart()`
-

9. Blog

El blog consume datos desde la API:

- Endpoint: /blog-posts
- Petición: GET

Estados usados:

- loadingBlog
 - errorBlog
 - blogPosts
-

10. Manejo de Modales

Modal de Cuenta

Controlado mediante: - showAccountModal - accountTab

Permite alternar entre: - Login - Registro

Modal de Carrito

Controlado por: - showCart

11. Seguridad

- Autenticación basada en **JWT**
 - Rutas protegidas (checkout)
 - Validación de sesión desde frontend
-

12. Buenas Prácticas Implementadas

- Separación de responsabilidades
 - Componentes reutilizables
 - Hooks de React
 - Manejo de errores y loading
 - Navegación controlada
-

- Tests con Jest y React Testing Library
-