🛍️ Documentación Técnica — Proyecto VCA Shop

📌 Descripción General

Aplicación web de comercio electrónico desarrollada con React, Node.js y WebSocket. Permite a los usuarios registrarse, iniciar sesión, explorar productos, añadir al carrito y realizar pagos.

🚀 Tecnologías Utilizadas

- React + Context API
- Material UI + Bootstrap
- Node.js + Express
- WebSocket (socket.io)
- Redsys (pasarela de pago)

🎁 Estructura del Proyecto

src/

├── components/ → Componentes visuales reutilizables

├── context/ → Contextos globales (cliente, carrito, productos...)

├── servicios/ → Funciones que interactúan con el backend

├── pages/ → Vistas principales (Home, Validación, Registro...)

├── utils/ → Funciones auxiliares

├── App.jsx → Enrutador principal

├── index.js → Punto de entrada

📄 Componentes Principales

- Validacion.jsx

Formulario de login con email/contraseña y Google. Carga cliente y carrito.

- Registro.jsx

Formulario de registro con verificación por código.

- ProductoMin.jsx

Tarjeta compacta para mostrar productos en el catálogo.

- ProductoMax.jsx

Vista detallada de un producto con opción de añadir al carrito.

- CarroComp.jsx

Resumen del carrito con opción de modificar cantidades.

- Compra.jsx

Pantalla de confirmación de compra y generación de pago.

- RutaProtegida.jsx

Protege rutas que requieren autenticación.

- Temporizador.jsx

Cuenta regresiva visual con acción al finalizar.

- PagoOk.jsx / PagoError.jsx

Pantallas de resultado tras el intento de pago.

🔧 Servicios

- actualizarCliente(cliente)

Envía los datos del cliente al backend para actualizar su perfil.

- actualizarProductos(productos, producto)

Reemplaza un producto dentro del array por su versión actualizada.

- productoEnCarro(producto, carro)

Verifica si un producto ya está en el carrito.

- actualizarStock(productos, producto, cantidad)

Reduce el stock de un producto en la lista.

- generarPago(cliente, carro)

Genera el formulario de pago para Redsys.

- verificarCodigo(codigo, email)

Verifica el código enviado por correo para registro o recuperación.

🌐 Contextos

- ClienteContext

Gestiona el estado del cliente autenticado. Se sincroniza con .

- CarroContext

Gestiona el contenido del carrito. Permite persistencia entre recargas.

- productosContext

Contiene la lista de productos disponibles en la tienda.

- productoContext

Guarda el producto seleccionado para mostrar en detalle.

- CategoriaContext

Permite filtrar productos por categoría seleccionada.

- SocketContext

Establece y gestiona la conexión WebSocket con el servidor.

 Autor

Vicente Contreras Alcuña

Desarrollador del proyecto VCA Shop

Ubicación: Huelva, España