

**Desarrollo de aplicación SPA**

**Ingeniería en Informática**

Programación Front End

**Alejandro Segundo Carez Armijo**

Sede Virtual, 21 de abril de 2025

# 1.-Objetivo del proyecto

El proyecto consiste en el desarrollo de una aplicación web de carrito de compras donde los usuarios pueden explorar productos, agregarlos al carrito y pagar. El sistema de autenticación está implementado con Firebase, y se utilizan rutas protegidas para asegurar que solo los usuarios autenticados puedan acceder a funcionalidades como el carrito y el proceso de checkout.

# 2.-Flujo de la aplicación

**Pantalla de inicio:**

-Los usuarios ven una lista de los productos disponibles

-Los usuarios pueden hacer clik en el botón de “agregar al carrito” para incorporar productos.

-La cantidad de productos se actualiza

**Pantalla de carrito:**

-Acá, los usuarios pueden ver los productos agregados y el total de la compra.

-Se tiene la opción de proceder al proceso de pago (checkout).

**Pantalla de checkout:**

-Los usuarios deben haber iniciado sesión para poder comprar.

-La autenticación de los usuarios se realiza a través de Firebase.

# 3.-Componentes

**1.-Home.vue** **(pantalla de inicio):** acá se muestran los productos disponibles. Al realizar click en “agregar al carrito”, se disparará una mutación en el estado global (Vuex), actualizando así el carrito. Flujo:

-Muestra los productos.

-Permite agregar productos.

-Usa Vuex para actualizar el estado del carrito.

**2.-Cart.vue (pantalla de carrito):** se muestran los productos incorporados al carrito, junto con el total de la compra. Flujo:

-Muestra los productos.

-Calcula el total de la compra.

-Permite proceder al checkout si el usuario está registrado.

**3.-Checkout.vue (pantalla de checkout):** se simula el proceso de pago, mostrando el total de la compra y un botón de “proceder al pago”. Flujo:

-Verifica si el usuario está registrado.

-Muestra el total.

-Simula el pago.

**4.-Login.vue (pantalla de acceso):** permite a los usuarios iniciar sesión a través de Firebase. Si el usuario no está autenticado, no podrá usar el carrito ni pagar. Flujo:

-Permite iniciar sesión con email y contraseña.

-Redirige al usuario a la página de inicio si la autenticación es válida.

**5.-App.vue (componente principal):** es el punto de entrada en donde se incluyen las rutas y la vista principal. Flujo:

-Contiene <router view> para manejar la navegación entre distintas vistas.

-Controla las rutas y la renderización de los componentes.

**6.-router/index.js (rutas):** define las principales rutas de la aplicación. Estas están protegidas, lo cual hace que solo los usuarios autenticados puedan acceder a ciertas páginas. Flujo:

-Usa el sistema de rutas Vue para dirigir a los usuarios a las pantallas correspondientes.

-Si el usuario no está autenticado, se redirige a la pantalla de inicio.

**7.-store/index.js (Vuex store):** en este archivo se maneja el estado de la aplicación. Al agregar productos al carrito de compras, el estado se actualiza. Flujo:

-Almacena el estado del carrito.

-Permite que distintos compontentes pueda acceder al carrito y modificarlo.

**8.-firebase/firebase.js (integración con Firebase):** este archivo configura Firebase y exporta las funciones para la autenticación. Flujo:

-Inicializa Firebase y proporciona acceso al sistema de autenticación (getAuth) y la base de datos Firebase (getFirestore).

# 4.-Diseño

**-Firebase:** para la autenticación, Firebase es la opción más simple y escalable, lo cual permite que se pueda gestionar la autenticación de usuarios de forma rápida y eficaz sin configurar un backend más complicado.

**-Vuex para la gestión de estado:** se utiliza Vuex para la gestión del carrito, lo cual permite facilitar el acceso desde cualquier componente.

**-Rutas protegidas:** se implementaron rutas protegidas usando Vue Router, asegurando que sólo los usuarios autenticados puedan acceder al carrito y Checkout, garantizando así que el proceso de compra esté sólo disponible para los usuarios registrados.

# 5.-Posibles mejoras

**-Validaciones en el formulario de Login:** no se han implementado validaciones de, por ejemplo, si los campos de email o contraseña se encuentran vacíos.

**-Proceso de pago real:** el sistema de pago es simulado. En una versión futura se podría integrar un sistema de pago real como Stripe.

**-Interaccion con Firestore:** los productos se encuentran codificados en el Front end actualmente. Integrando Firestore se podría almacenar y recuperar productos de forma dinámica.