LOGIN
Uso de Provide, Inject y Ciclo de vida de componente

Índice

Introducción

Proyecto de Manejador Global de Autenticación inspirada en el juego Final Fantasy VII, donde los usuarios pueden autenticarse como miembros de Avalancha o como aliados de Shinra.

Objetivo

Implementar autenticación de usuarios mediante un sistema de tokens, gestionando el estado del usuario y las rutas protegidas. Se utilizarán los conceptos de provide e inject, además de los elementos del ciclo de vida de los componentes.

Requisitos del Proyecto

Inicio del Proyecto

- Configura un proyecto Vue con soporte para TypeScript, ESLint y Prettier.
- Usa la estructura de componentes SFC (<script setup lang="ts">) para organizar el código.

Comprobación de formato de los datos

Tanto en el registro como en login comprobar:

- El formato correcto de los datos:
  - Si es un correo electrónico → formato de correo
  - Si es contraseña → mínimo 8 caracteres, al menos un número, al menos una letra mayúscula, al menos un carácter especial (@,#,%,&, etc)
- Longitud determinada de los capos de entrada de datos → limitarlos a rangos lógicos.

Comprobaciones en el registro y login

En el registro, tras verificar los formatos, se comprueba:

- Si es usuario ya existe en la base de datos.
  - Mostrar una ventana de mensaje indicándolo y tras cerrarla volver al registro limpiando todos los campos.
- Si el usuario no existe en la BBDD proceder a guardar los datos y mostrar una ventana de mensaje indicando si la acción fue satisfactoria o no.

En el login, tras verificar los formatos, se comprueba:

- Si es usuario ya existe en la base de datos.
  - Si es así se procede a redirigir al usuario al home.
- Si no existe mostrar una ventana de mensaje indicándolo y tras cerrarla volver al login SIN borrar datos

Creación de componente nuevo

Para comprobar el login vamos a pensar que este login está dentro de una aplicación de una tienda.

Vamos a desarrollar un nuevo componente Shop.vue accesible desde el item “Tienda” en el Navbar para acceder a la zona de exposición de productos. No hace falta que añadas productos, solo un h1 con “Tienda”.

Lo que hay que tener en cuenta es que si intentamos acceder a “Tienda” sin estar logueado no será posible y deberemos mostrar un mensaje emergente como “Debes autenticarte para acceder a la tienda”.

Autenticación con Token

- Implementa un componente AuthProvider.vue que gestione el estado de autenticación.
- Usa provide para compartir el estado global del usuario (autenticado o no) con otros componentes, como Shop.vue.
- Implementa una función para generar un token de autenticación (simulado) y guárdalo en SessionStorage. (también puedes usar servicios externos si lo deseas)

Componentes Clave

- **LoginPage.vue**: Formulario para que el usuario inicie sesión. Tras autenticarse correctamente, almacena el token y redirige al usuario a la página de inicio.
- **UserProfile.vue**: Muestra la información del usuario autenticado usando inject para acceder al estado del usuario.
- **NavBar.vue:** Implementa un menú de navegación dinámico que muestra opciones diferentes según el estado de autenticación.
- Shop.vue: no hace falta que añadas productos, solo un h1 con “Tienda”.

Rutas Protegidas

Usa vue-router para configurar rutas protegidas. Redirige a /login si el usuario intenta acceder a una ruta protegida sin estar autenticado.

Ciclo de Vida

- Usa el método onMounted en AuthProvider.vue para verificar si hay un token almacenado en SessionStorage al cargar la aplicación.
- Usa onUnmounted para limpiar cualquier dato relacionado con la sesión al cerrar sesión.

Almacenamiento de Sesión

- Almacena el token en SessionStorage.
- Usa un mecanismo de expiración del token y redirige al usuario al formulario de inicio de sesión si el token expira.

Archivos y Componentes Involucrados

- src/components/AuthProvider.vue: Implementa provide y maneja la autenticación.
- src/components/LoginPage.vue: Formulario de inicio de sesión.
- src/components/UserProfile.vue: Muestra datos del usuario autenticado.
- src/router/index.ts: Configuración de rutas y middleware para proteger rutas.
- SessionStorage: Almacenamiento del token