Desarrollo Web
Mayo- Agosto 2024
Ing. Ana Sol Arteaga

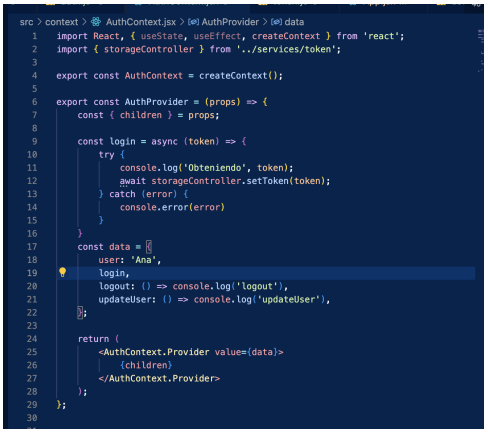
Introducción al desarrollo Web con Ant Design

Local Storage

Para crear la sesión del usuario es necesario almacenar el token en algún lado, en el caso del desarrollo web, vamos a almacenar el token en el Local Storage, Para comenzar, debemos de crear el espacio para almacenar el token, lo haremos en el directorio de src/utils/constants.js crear STORAGE y almacenar TOKEN

El segundo paso es crear las funciones para almacenar, obtener y eliminar el token, creamos un archivo en el directorio src/services/token.js

Prueba que el token se está almacenando correctamente, en **AuthContext** crea la función de login (No olvides importar **storageController**) corrige data para poder usar la función fuera del contexto.



Ahora en el componente **FormLogin** recupera por props la función **login** y dentro de la función **onFinish** reemplaza el console.log de la respuesta con la ejecución de la función **login**

const { login } = useAuthData

```
const onFinish = async (values) => {
    setLoading(true);
    setLoginError(false);
    try {
        const response = await authService.loginF(values.username, values.p
        if (response && response.data) {
            localStorage.setItem('token', response.data.token);
            login(response.data.token);
            navigate('/');
        } else {
            console.error('Error en el inicio de sesión: Respuesta inespera
            setLoginError(true);
        }
```

Ahora solo falta verificar que el token está almacenado correctamente en AsyncStorage.

En AuthContext crea la función getSession para obtener el token almacenado en el storage.

```
useEffect(() => {
    getSession();
}, [])
const getSession = async () => {
    const token = await storageController.getToken();
    console.log('Token --> :', token);
}

const login = async (token) => {
    try {
```