



# Anáhuac Mayab

Universidad Anáhuac Mayab

**Actividad:**

**Proyecto Aplicativo**

**Asignatura:**

**Programación para dispositivos móviles**

**Alumno:**

Mauricio Cifuentes Alvarez

Mérida, Yucatán

Esta herramienta ha sido desarrollada para ayudar a equipos y profesionales a gestionar sus objetivos clave y resultados de manera eficiente.

La aplicación Gestor de Objetivos basados en acciones clave es una plataforma intuitiva que permite a los usuarios establecer, organizar y realizar un seguimiento de sus objetivos estratégicos, todo desde su dispositivo móvil.

El objetivo principal es facilitar la planificación y ejecución de metas, mejorar la productividad y fomentar el trabajo colaborativo mediante una herramienta sencilla pero poderosa. Esta aplicación está dirigida a empresas, equipos de trabajo y profesionales independientes que buscan una forma organizada y efectiva de gestionar sus objetivos."

Esta aplicación utiliza MongoDB como base de datos debido a su naturaleza NoSQL, que permite almacenar datos de forma flexible y escalable. MongoDB es ideal para manejar información estructurada y semiestructurada como objetivos y usuarios.

Esquema de la base de datos:

1. Colección 'users': contiene información de los usuarios (correo, contraseña, nombre).
2. Colección 'goals': almacena los objetivos, incluyendo título, descripción, métricas clave y fechas.

El backend está desplegado en Render y se conecta a MongoDB Atlas para asegurar alta disponibilidad y escalabilidad.

Links:

Proyecto Front con axios

<https://github.com/MauricioCifuentes2000/FullApp>

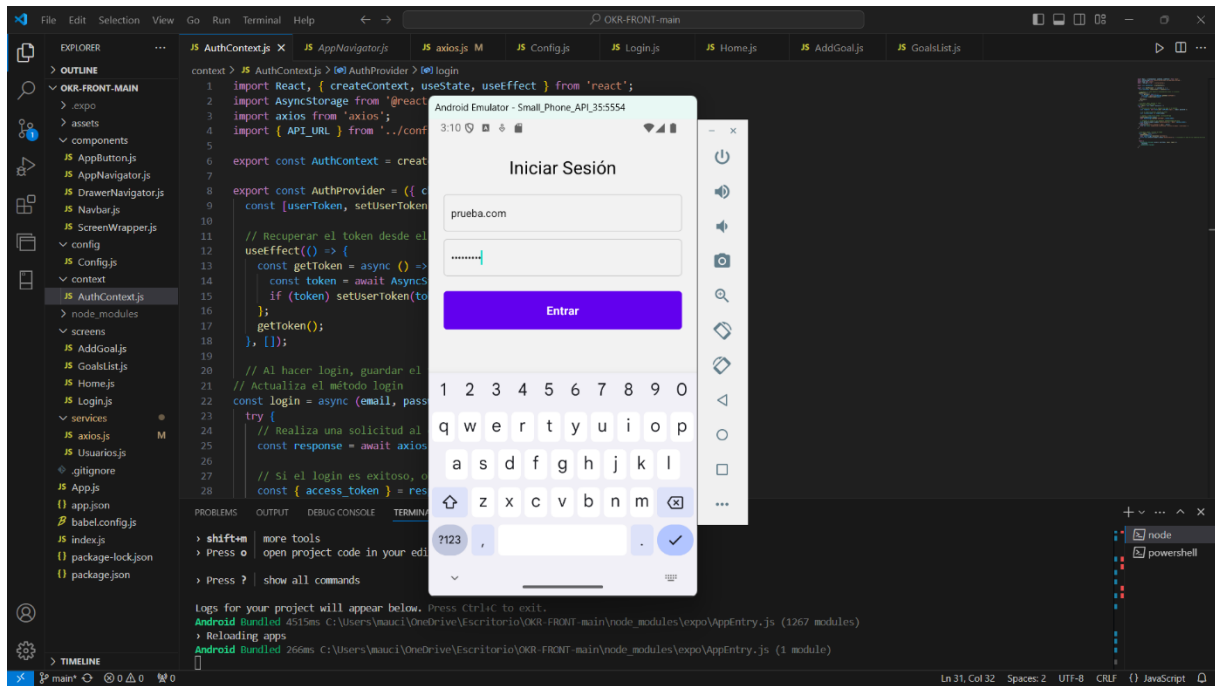
Videos de youtube

<https://youtu.be/qITYr3dkgMY>

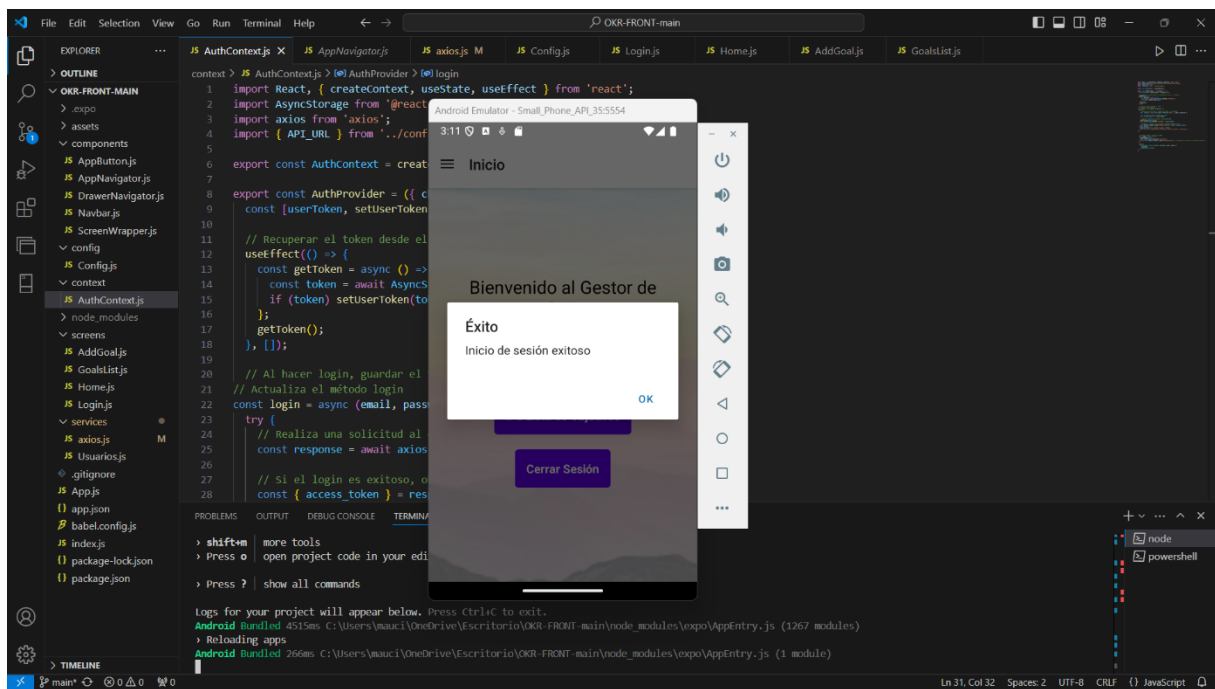
<https://youtu.be/rheL1c5NPio>

Backend

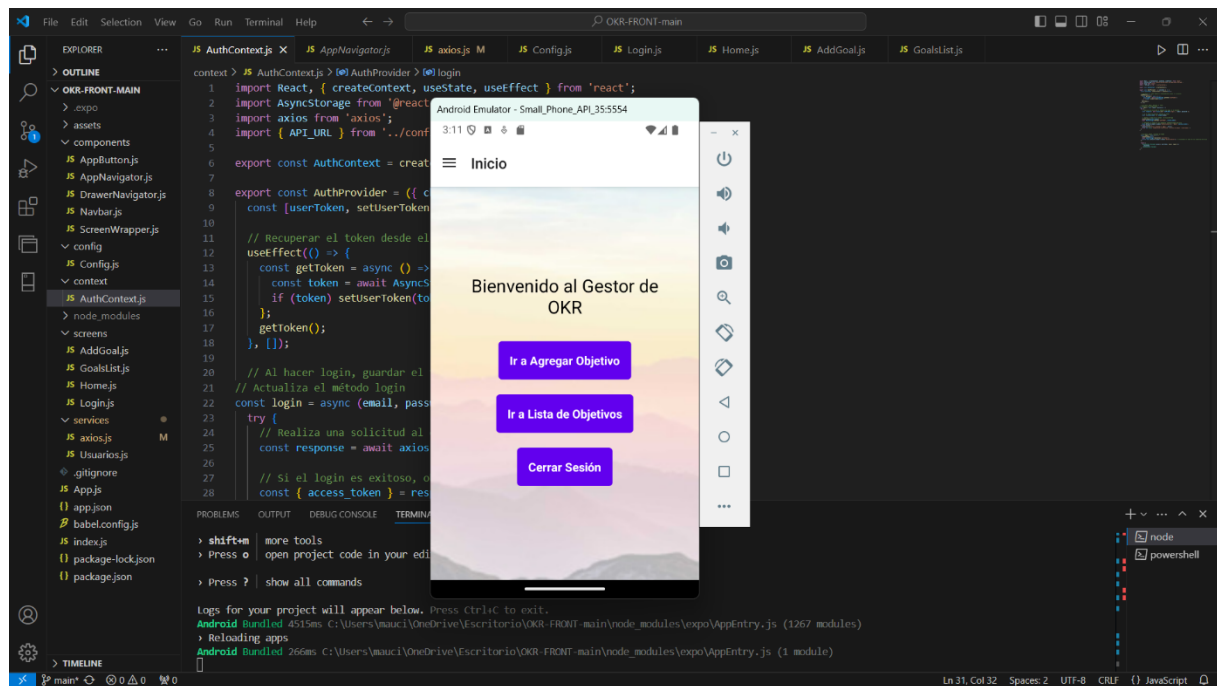
<https://github.com/MauricioCifuentes2000/absoluteCopy>



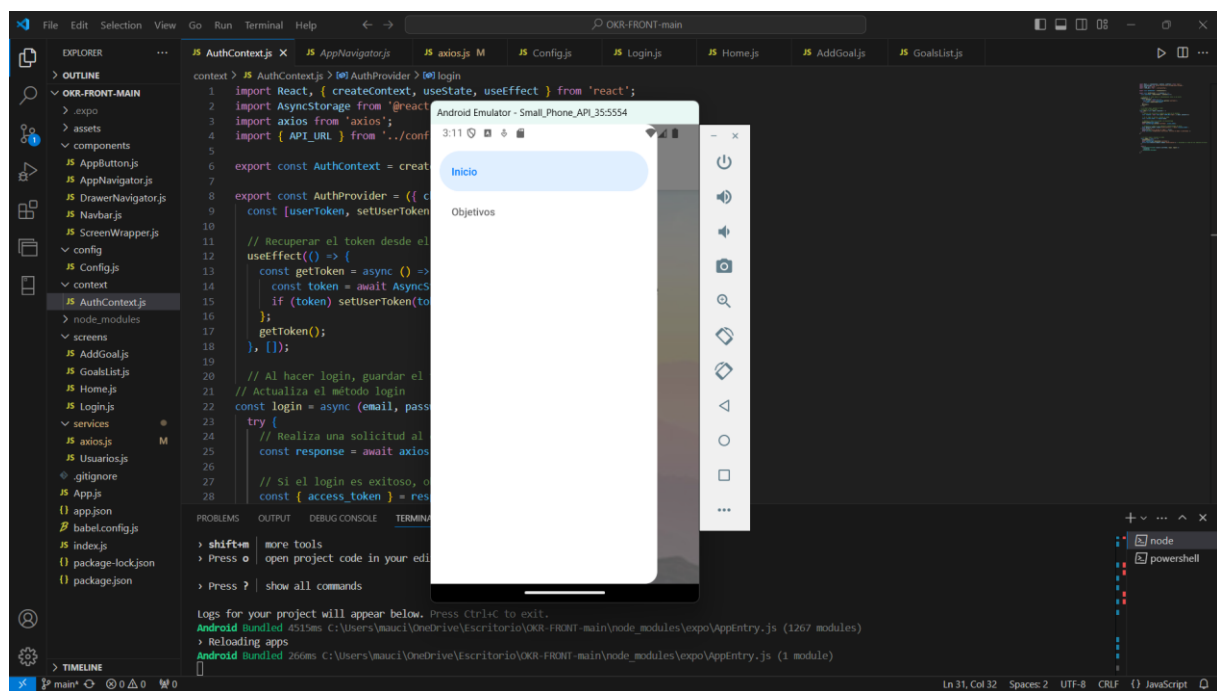
La pantalla principal es Login donde se esperan las credenciales de los usuarios, las contraseñas están hasheadas al hacer registro.



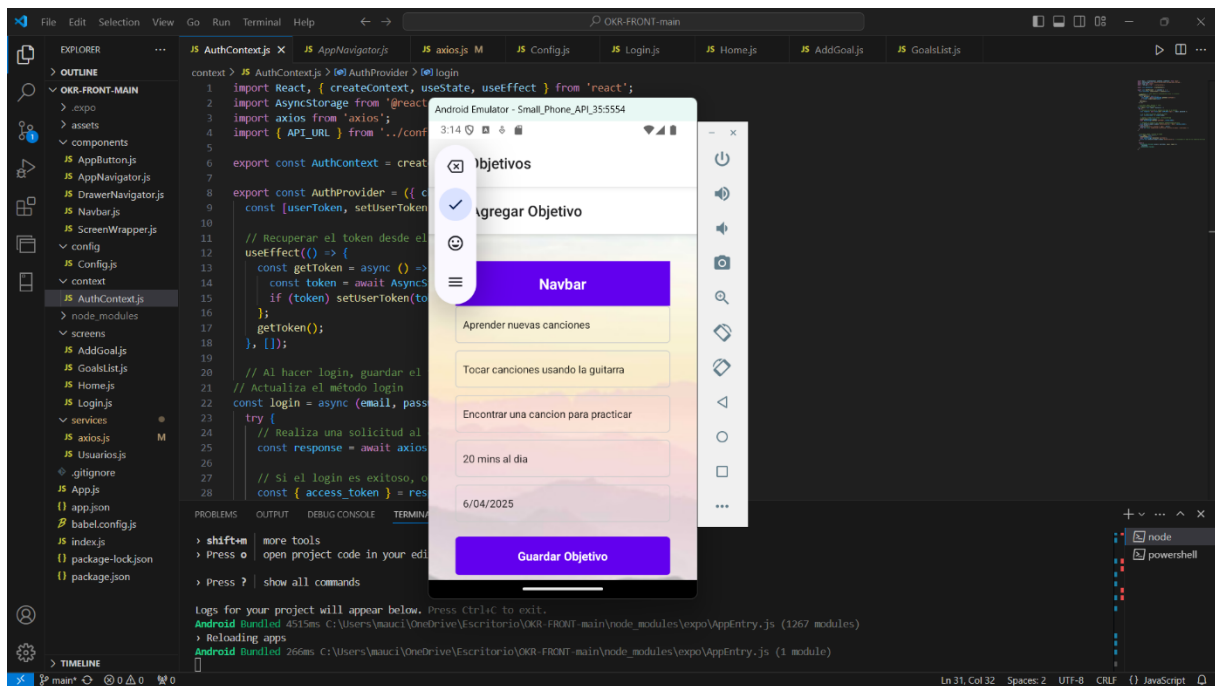
Al introducir las credenciales adecuadas se muestra una ventana emergente con el mensaje de éxito.



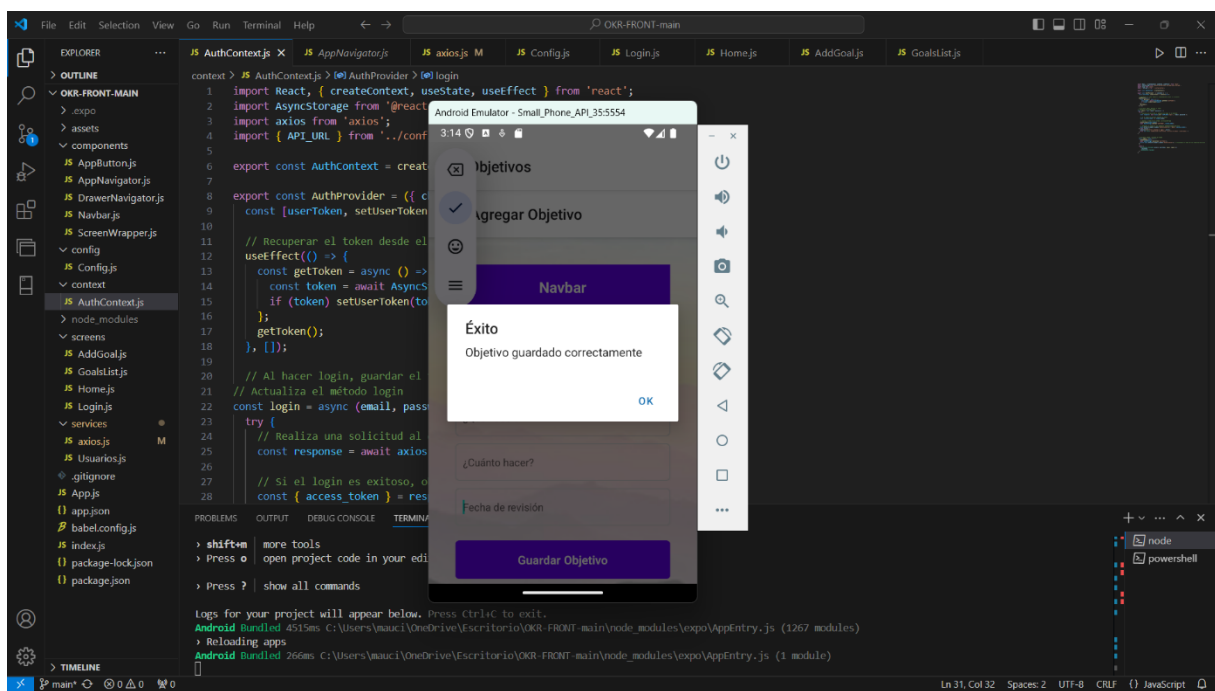
La pantalla home muestra las tres opciones de la app, agregar objetivos, lista de objetivos y cerrar sesión.



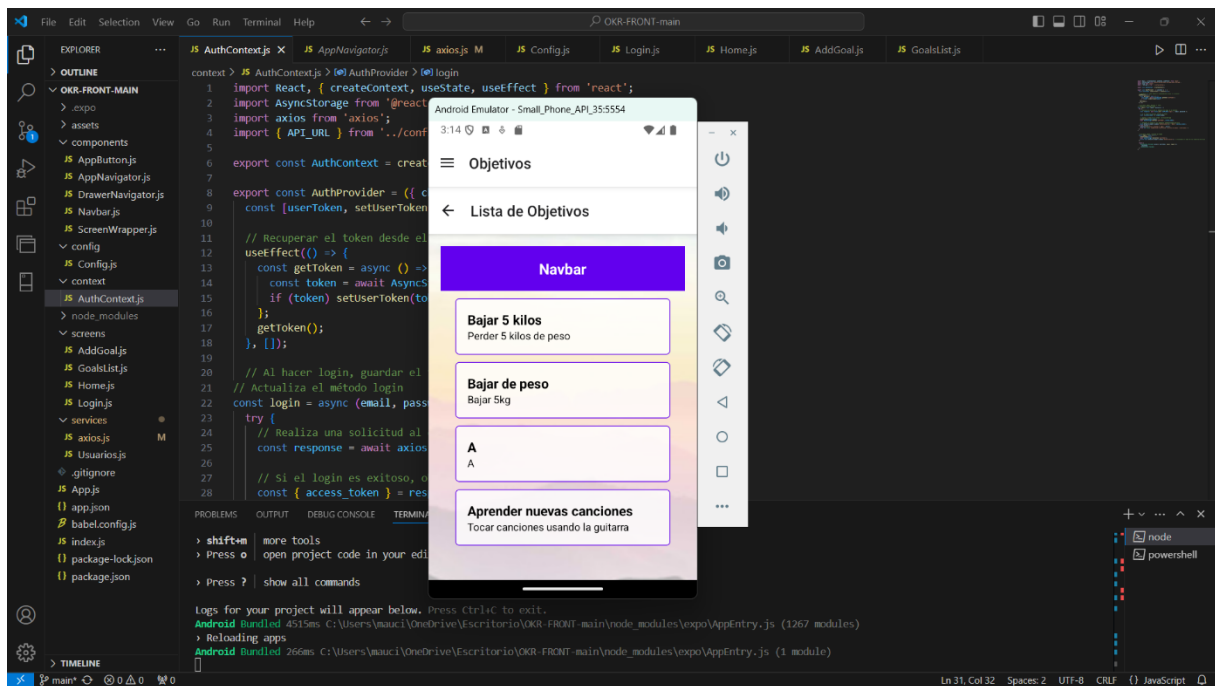
Se tiene la disposición de una sidebar para acceder a las otras ventanas



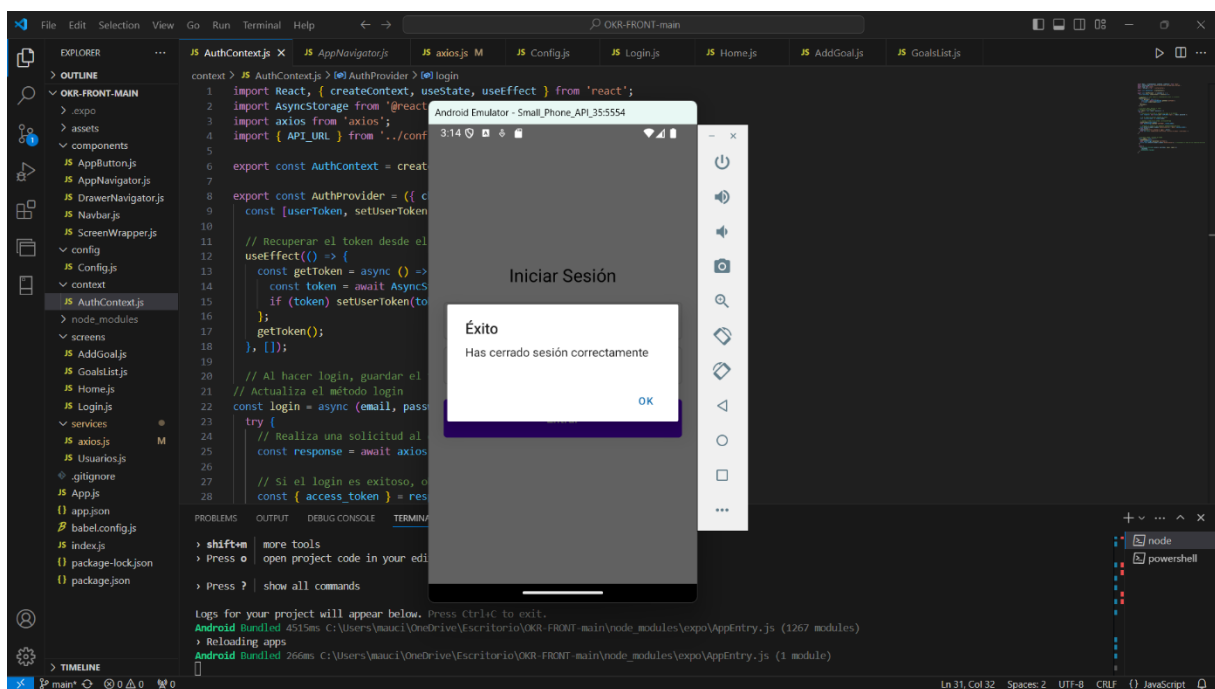
La pestaña de agregar objetivos presenta al usuario con bloques de texto los cuales puede llenar y serán presentados la lista de objetivos



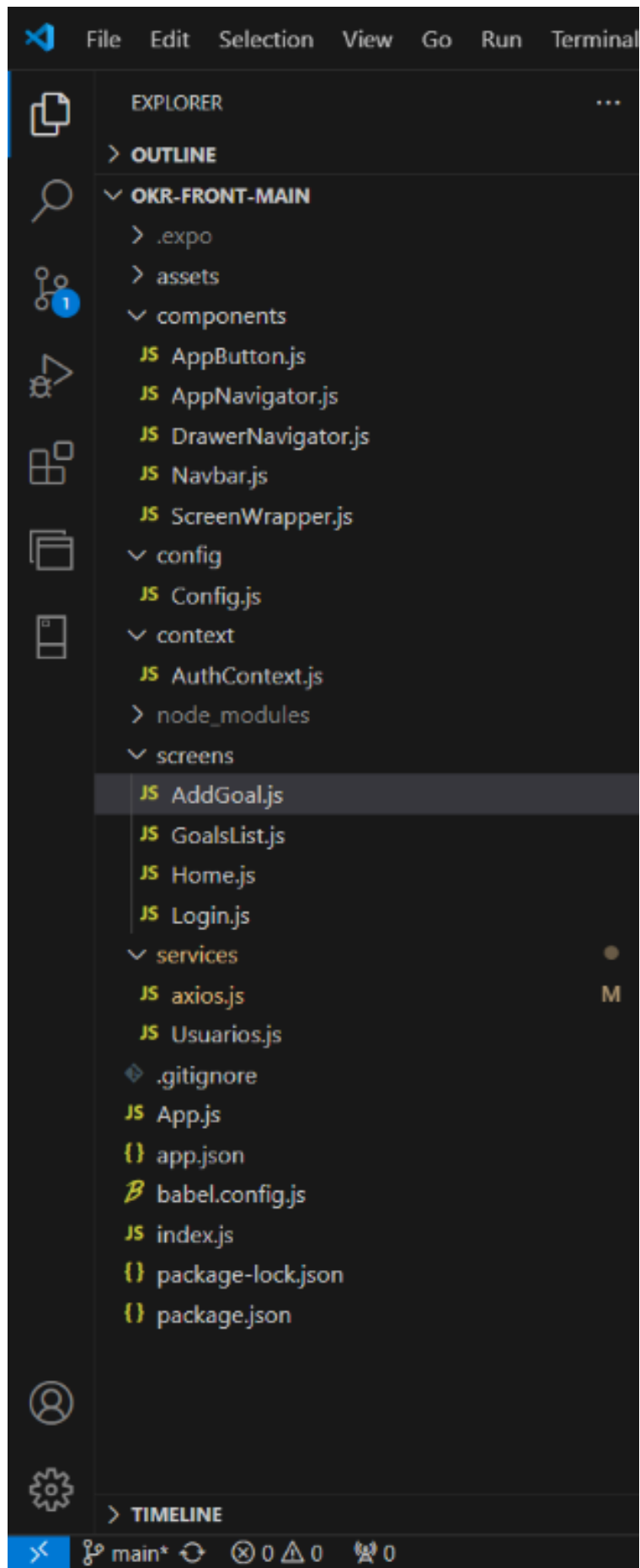
Al presionar el botón guardar objetivos se presenta una ventana emergente informando del éxito o de algún error



La ventana lista de objetivos muestra los objetivos que tiene el usuario en un estilo diferente y fácil de interpretar.



El botón cerrar sesión elimina el token guardado y por consecuencia nos envía a la pantalla de login para volver a ingresar ya sea con el mismo usuario o con otro diferente para poder acceder a las listas personales.

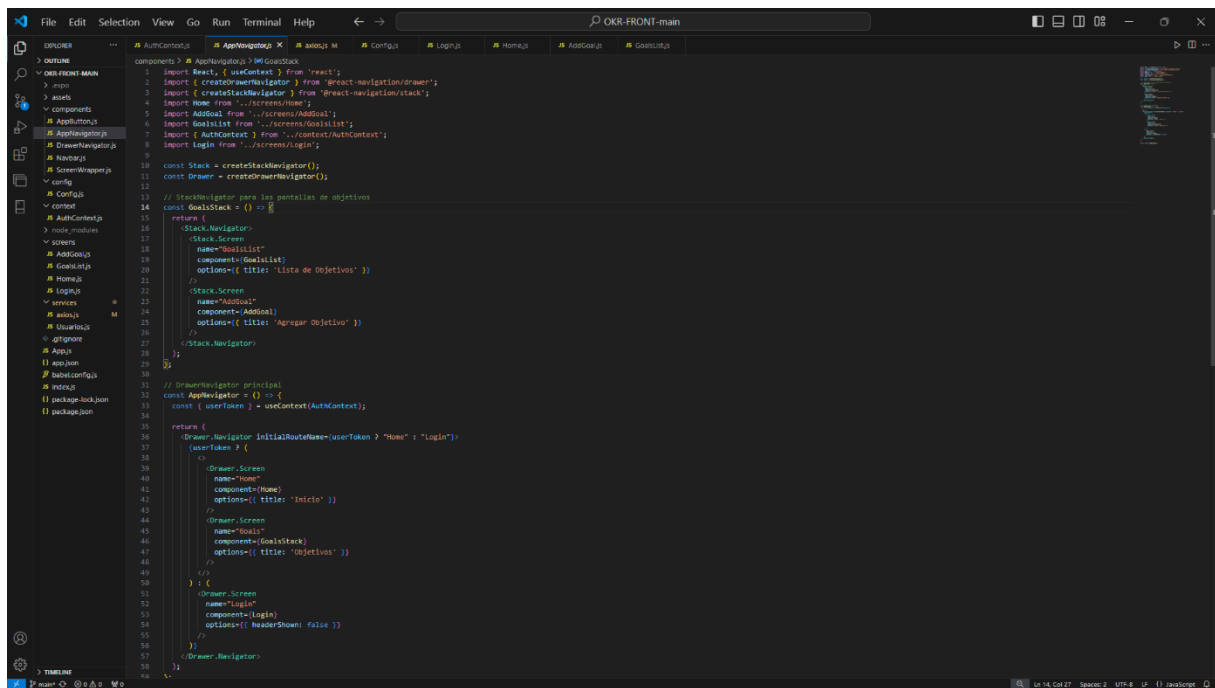


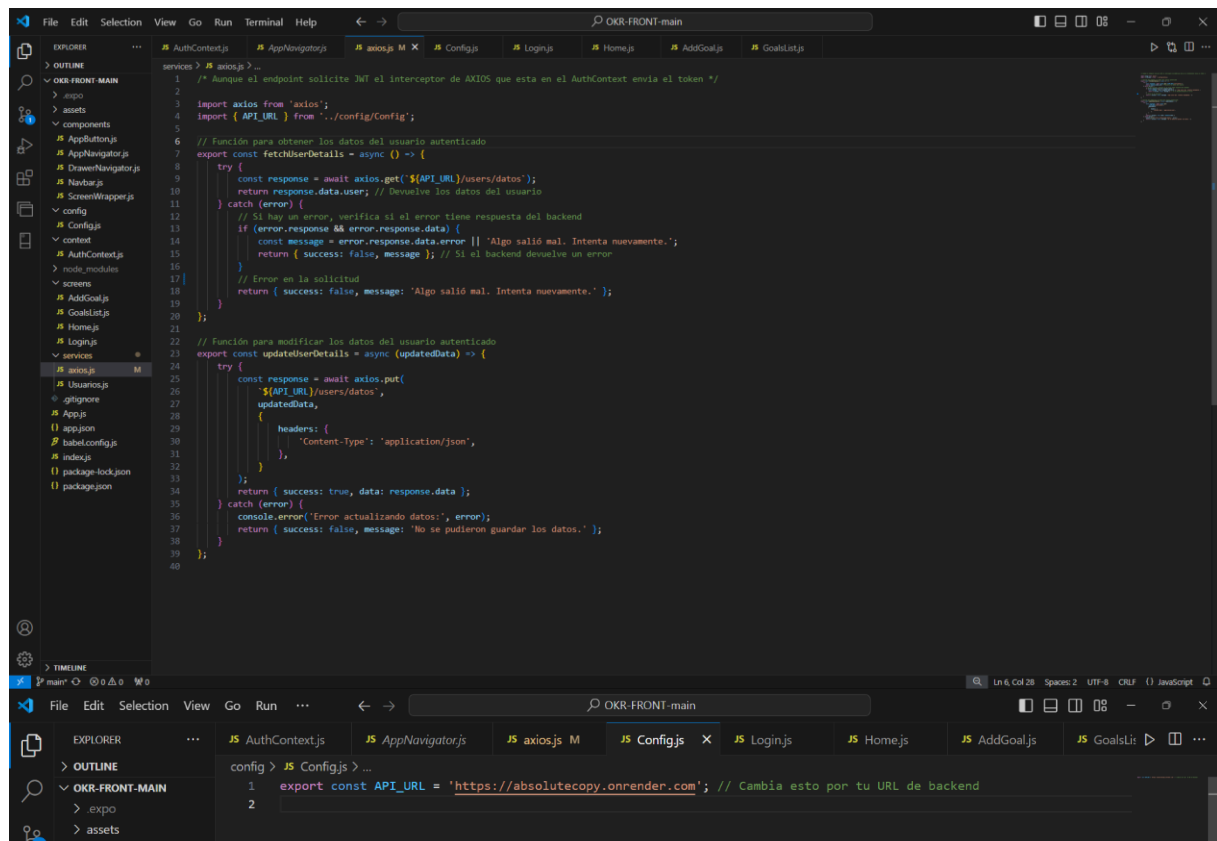
```
Go Run Terminal Help  OKR-FRONT-main  on View Go Run Terminal Help  OKR-FRONT-main

AuthContext.js x AppNavigator.js axios.js M Config.js Logins.js Home.js

context > JS AuthContext.js > login
1 import React, { createContext, useState, useEffect } from 'react';
2 import AsyncStorage from '@react-native-async-storage/async-storage';
3 import axios from 'axios';
4 import { API_URL } from '../config/Config';
5
6 export const AuthContext = createContext();
7
8 export const AuthProvider = ({ children }) => {
9   const [userToken, setUserToken] = useState(null);
10
11   // Recuperar el token desde el almacenamiento local si ya existe
12   useEffect(() => {
13     const getToken = async () => {
14       const token = await AsyncStorage.getItem('userToken');
15       if (token) setUserToken(token);
16     };
17     getToken();
18   }, []);
19
20   // Al hacer login, guardar el token
21   // Actualiza el método login
22   const login = async (email, password) => {
23     try {
24       // Realiza una solicitud al endpoint de login en el backend
25       const response = await axios.post(`${API_URL}/login`, { email, password });
26
27       // Si el login es exitoso, obtiene el token
28       const { access_token } = response.data;
29
30       // Guarda el token en el estado y en AsyncStorage
31       setUserToken(access_token);
32       await AsyncStorage.setItem('userToken', access_token);
33
34       // Establece el token en las cabeceras predeterminadas de Axios
35       axios.defaults.headers.common['Authorization'] = `Bearer ${access_token}`;
36
37     } catch (error) {
38       console.error('Error durante el login', error);
39       throw new Error('Credenciales incorrectas. Verifica tu email y contraseña.');
```







```
View Go Run Terminal Help ← → OKR-FRView Go Run Terminal Help ← →
# AuthContext.js # AppNavigator.js # axios.js M # Config.js # Logins x # Home.js # AuthContext.js # AppNavigator.js # axios.js M # Config.js # Logins

screens > # Logins > Login
1 import React, { useState, useContext } from 'react';
2 import { View, Text, TextInput, StyleSheet, Alert } from 'react-native';
3 import AppButton from '../components/AppButton';
4 import { AuthContext } from '../context/AuthContext';
5
6 const Login = () => {
7   const [email, setEmail] = useState('');
8   const [password, setPassword] = useState('');
9   const { login } = useContext(AuthContext);
10
11   const handleLogin = async () => {
12     try {
13       // llama al método login del contexto
14       await login(email, password);
15       Alert.alert('Éxito', 'Inicio de sesión exitoso');
16     } catch (error) {
17       Alert.alert('Error', error.message || 'Credenciales incorrectas');
18     }
19   };
20
21   return (
22     <View style={styles.container}>
23       <Text style={styles.title}>Iniciar Sesión</Text>
24       <TextInput
25         style={styles.input}
26         placeholder="Correo electrónico"
27         value={email}
28         onChangeText={setEmail}
29       />
30       <TextInput
31         style={styles.input}
32         placeholder="Contraseña"
33         secureTextEntry
34         value={password}
35         onChangeText={setPassword}
36       />
37       <AppButton title="Entrar" onPress={handleLogin} />
38     </View>
39   );
40 };
41
42 const styles = StyleSheet.create({
43   container: {
44     flex: 1,
45     justifyContent: 'center',
46     padding: 20,
47   },
48   title: {
49     fontSize: 24,
50     textAlign: 'center',
51     marginbottom: 20,
52   },
53   input: {
54     borderwidth: 1,
55     borderColor: '#ccc',
56     borderradius: 5,
57     padding: 10,
58     marginbottom: 10,
59   },
60 });
61
62 export default Login;
63
```

```
Go Run Terminal Help ← → Go Run Terminal Help ← → OKR
hContext.js AppNavigator.js axios.js M Config.js Login.js Home
s> # Home.js > ...
import React, { useContext } from 'react';
import { View, Text, StyleSheet, Alert } from 'react-native';
import AppButton from '../components/AppButton';
import ScreenWrapper from '../components/ScreenWrapper';
import { AuthContext } from '../context/AuthContext';

const Home = ({ navigation }) => {
  const { logout } = useContext(AuthContext);

  const handleLogout = async () => {
    try {
      await logout(); // Llama a la función logout
      Alert.alert('Éxito', 'Has cerrado sesión correctamente');
    } catch (error) {
      Alert.alert('Error', 'Hubo un problema al cerrar sesión');
    }
  };

  return (
    <ScreenWrapper>
      <View style={styles.container}>
        <Text style={styles.title}>Bienvenido al Gestor de OKR</Text>
        <AppButton
          title="Ir a Agregar Objetivo"
          onPress={() => navigation.navigate('Goals', { screen: 'AddGoal' })}
        />
        <AppButton
          title="Ir a Lista de Objetivos"
          onPress={() => navigation.navigate('Goals', { screen: 'GoalsList' })}
        />
        <AppButton
          title="Cerrar Sesión"
          onPress={handleLogout}
        />
      </View>
    </ScreenWrapper>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    padding: 20,
  },
  title: {
    fontSize: 24,
  },
});

export default Home;
```

```

Go Run Terminal Help
OxKor-Flow Go Run Terminal Help
OxKor-Flow Go Run Terminal Help

import React, { useState, useEffect } from 'react';
import { View, TextInput, StyleSheet, Alert } from 'react-native';
import AppButton from '../components/AppButton';
import Navbar from '../components/Navbar';
import ScreenWrapper from '../components/ScreenWrapper';
import axios from 'axios';
import { API_URL } from '../config/Config';
import { AuthContext } from '../context/AuthContext';

const AddGoal = () => {
  const [formData, setFormData] = useState({
    title: '',
    description: '',
    what_to_do: '',
    how_much_to_do: '',
    check_date: ''
  });
  const { useToken } = useContext(AuthContext);

  const handleSaveGoal = async () => {
    try {
      const response = await axios.post(`${API_URL}/addGoal`, formData, {
        headers: {
          Authorization: `Bearer ${useToken}`, // Incluye el token en el encabezado
        },
      });

      if (response.status === 201) {
        Alert.alert('Éxito', 'Objetivo guardado correctamente');
        setFormData({
          title: '',
          description: '',
          what_to_do: '',
          how_much_to_do: '',
          check_date: ''
        });
      } else {
        Alert.alert('Error', 'No se pudo guardar el objetivo');
      }
      console.error('Error al guardar el objetivo:', error);
      console.error('Hubo un problema al guardar el objetivo');
    } catch (error) {
      console.error('Hubo un problema al guardar el objetivo');
    }
  };

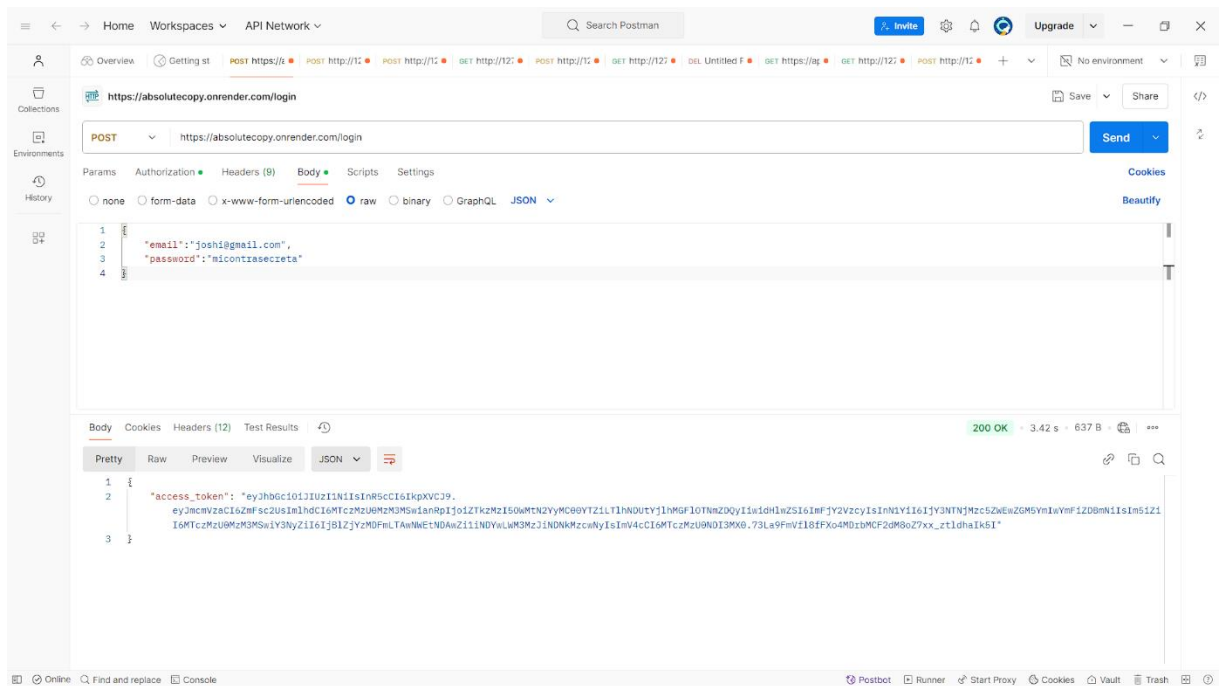
  return (
    <ScreenWrapper>
      <View style={styles.container}>
        <Navbar title="Agregar Objetivo" />
        <TextInput
          style={styles.input}
          placeholder="Título"
          value={formData.title}
          onChangeText={(text) => setFormData({ ...formData, title: text })}
        />
        <TextInput
          style={styles.input}
          placeholder="Descripción"
          value={formData.description}
          onChangeText={(text) => setFormData({ ...formData, description: text })}
        />
        <TextInput
          style={styles.input}
          placeholder="¿Qué hacer?"
          value={formData.what_to_do}
          onChangeText={(text) => setFormData({ ...formData, what_to_do: text })}
        />
        <TextInput
          style={styles.input}
          placeholder="¿Cuánto hacer?"
          value={formData.how_much_to_do}
          onChangeText={(text) => setFormData({ ...formData, how_much_to_do: text })}
        />
        <TextInput
          style={styles.input}
          placeholder="Fecha de revisión"
          value={formData.check_date}
          onChangeText={(text) => setFormData({ ...formData, check_date: text })}
        />
        <AppButton title="Guardar Objetivo" onPress={handleSaveGoal} />
      </View>
    </ScreenWrapper>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
  },
});

```

The screenshot shows a VS Code editor with a React Native project. The Explorer sidebar on the left lists the file structure, including folders like 'src', 'assets', 'components', and 'screens'. The main editor area displays the 'AddGoal.js' file, which contains a functional component for adding goals. The code includes state management for 'formData', a 'handleSaveGoal' function, and a JSX structure with three text inputs and a save button. The bottom status bar shows the file is 'AddGoal.js' in the 'src' directory.

```
1 import React, { useState, useEffect, useContext } from 'react';
2 import { View, Text, FlatList, StyleSheet, Alert } from 'react-native';
3 import Navbar from '../components/Navbar';
4 import ScreenWrapper from '../components/ScreenWrapper';
5 import axios from 'axios';
6 import { API_URL } from '../config/Config';
7 import { AuthContext } from '../context/AuthContext';
8
9 const GoalsList = () => {
10   const [goals, setGoals] = useState([]);
11   const { userToken } = useContext(AuthContext);
12
13   const fetchGoals = async () => {
14     try {
15       const response = await axios.get(`${API_URL}/getUserGoals`, {
16         headers: {
17           Authorization: `Bearer ${userToken}`, // Incluye el token en el encabezado
18         },
19       });
20
21       if (response.status === 200) {
22         setGoals(response.data.goals);
23       } else {
24         Alert.alert('Error', 'No se pudieron obtener los objetivos');
25       }
26     } catch (error) {
27       console.error('Error al obtener los objetivos:', error);
28       Alert.alert('Error', 'Hubo un problema al obtener los objetivos');
29     }
30   };
31
32   useEffect(() => {
33     fetchGoals();
34   }, []);
35
36   return (
37     <ScreenWrapper>
38       <Navbar title="Lista de Objetivos" />
39       <FlatList
40         data={goals}
41         keyExtractor={(item) => item._id}
42         renderItem={({ item }) => (
43           <View style={styles.goal}>
44             <Text style={styles.goalTitle}>{item.title}</Text>
45             <Text>{item.description}</Text>
46           </View>
47         )}
48         contentContainerStyle={styles.listContainer}
49       </FlatList>
50     </ScreenWrapper>
51   );
52 }
53
54 const styles = StyleSheet.create({
55   listContainer: {
56     padding: 20,
57   },
58   goal: {
59     borderWidth: 1,
60     borderColor: '#6200EE',
61     padding: 15,
62     margin: 10,
63     backgroundColor: 'rgba(255, 255, 255, 0.8)',
64   },
65   goalTitle: {
66     fontSize: 18,
67     fontWeight: 'bold',
68   },
69 });
70
71 export default GoalsList;
```



HomeWorkspacesAPI Network

Search Postman

Invite

Upgrade

OverviewGetting startedPOST https://tPOST http://tPOST http://tGET http://tPOST http://tGET http://tDEL Untitled FGET https://agGET http://tPOST http://t

No environment

https://absolutecopy.onrender.com/register

SaveShare

Send

POSThttps://absolutecopy.onrender.com/register

ParamsAuthorizationHeaders (9)BodyScriptsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

1 {  
2   "username": "joshua",  
3   "email": "joshi@gmail.com",  
4   "password": "micontraseceta"  
5 }

BodyCookiesHeaders (12)Test Results

201 Created - 44.02 s - 394 B

PrettyRawPreviewVisualizeJSON

1 {  
2   "msg": "Usuario Creado Correctamente"  
3 }

OnlineFind and replaceConsole

PostbotRunnerStart ProxyCookiesVaultTrash