



MATERIA:
DPS441

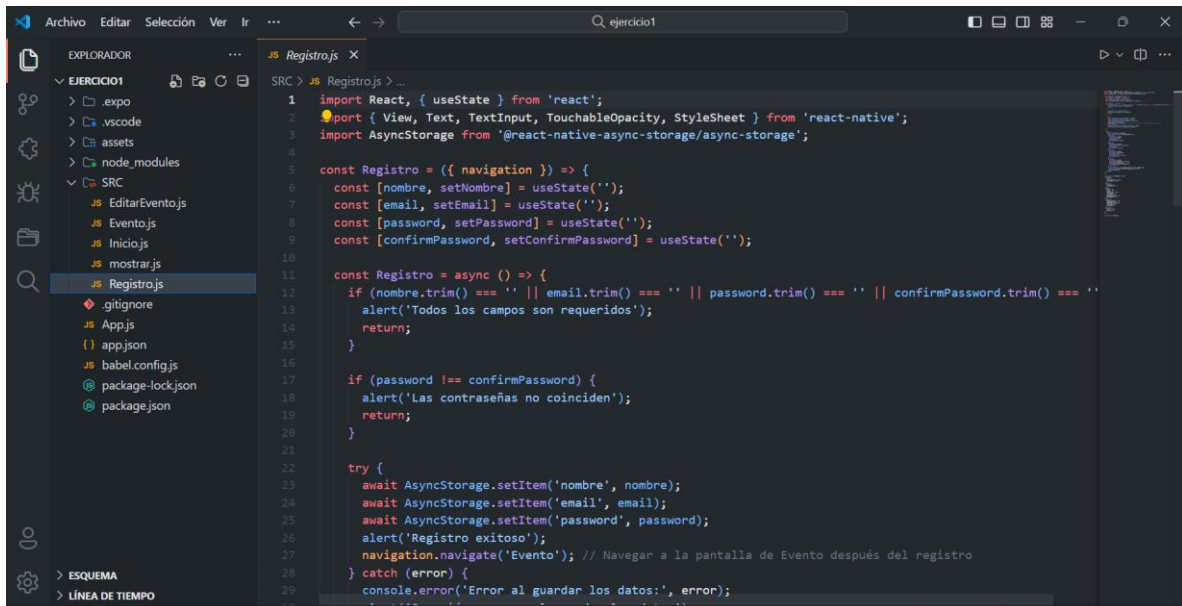
GRUPO: G2L

DOCENTE:
Karens Medrano

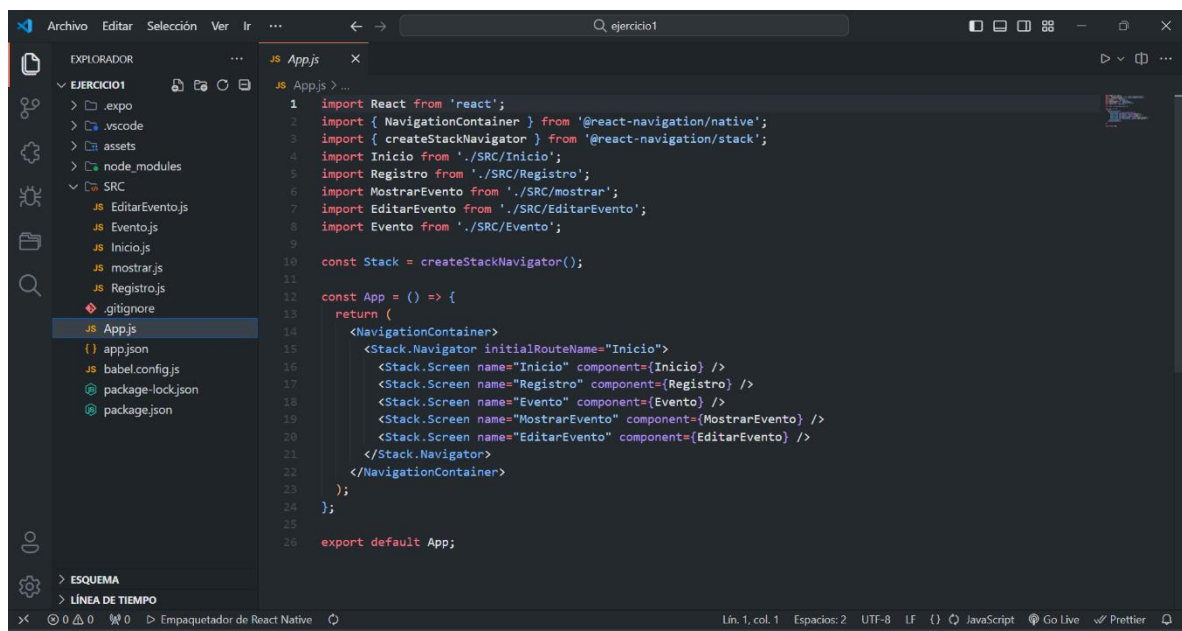
ESTUDIANTES:

Cabrera Velasco, Mario Steven-CV230385
Javier Stanley Valladares Valladares VV230315

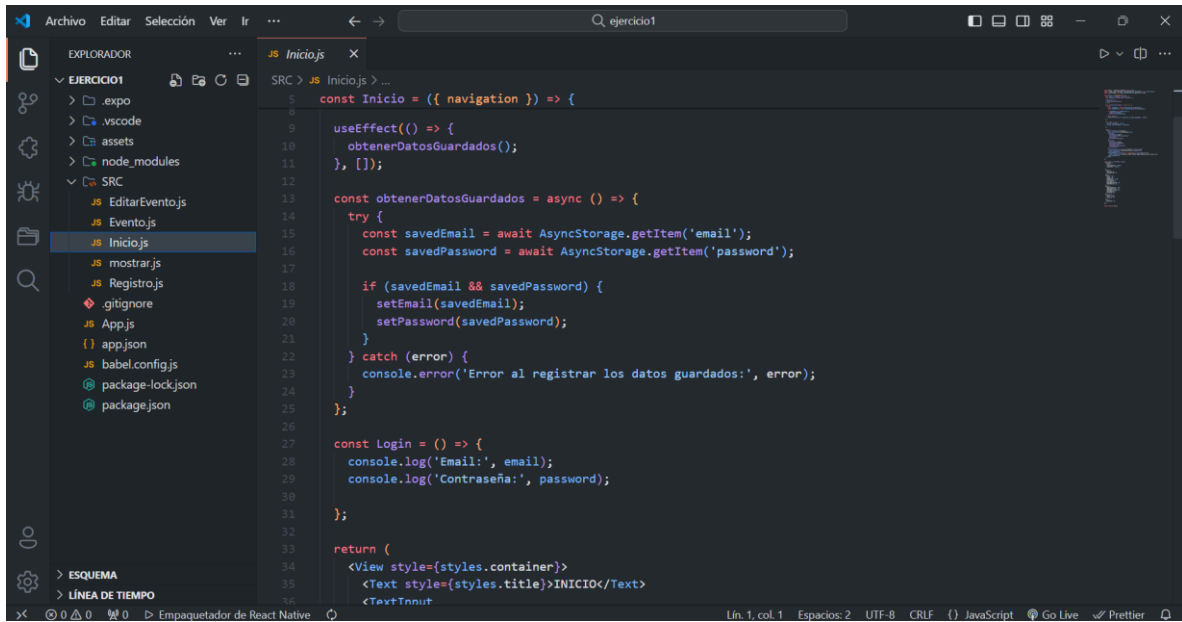
EJERCICIO 1



```
1 import React, { useState } from 'react';
2 import { View, Text, TextInput, TouchableOpacity, StyleSheet } from 'react-native';
3 import AsyncStorage from '@react-native-async-storage/async-storage';
4
5 const Registro = ({ navigation }) => {
6   const [nombre, setNombre] = useState('');
7   const [email, setEmail] = useState('');
8   const [password, setPassword] = useState('');
9   const [confirmPassword, setConfirmPassword] = useState('');
10
11   const Registro = async () => {
12     if (nombre.trim() === '' || email.trim() === '' || password.trim() === '' || confirmPassword.trim() === '') {
13       alert('Todos los campos son requeridos');
14       return;
15     }
16
17     if (password !== confirmPassword) {
18       alert('Las contraseñas no coinciden');
19       return;
20     }
21
22     try {
23       await AsyncStorage.setItem('nombre', nombre);
24       await AsyncStorage.setItem('email', email);
25       await AsyncStorage.setItem('password', password);
26       alert('Registro exitoso');
27       navigation.navigate('Evento'); // Navegar a la pantalla de Evento después del registro
28     } catch (error) {
29       console.error('Error al guardar los datos:', error);
30     }
31   }
32 }
```

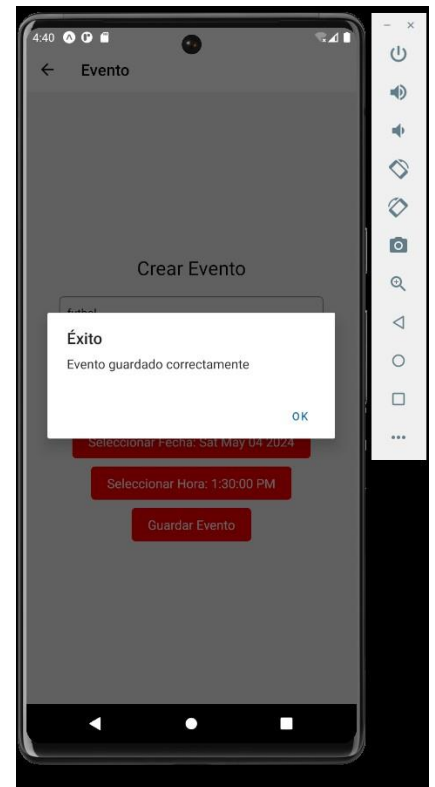
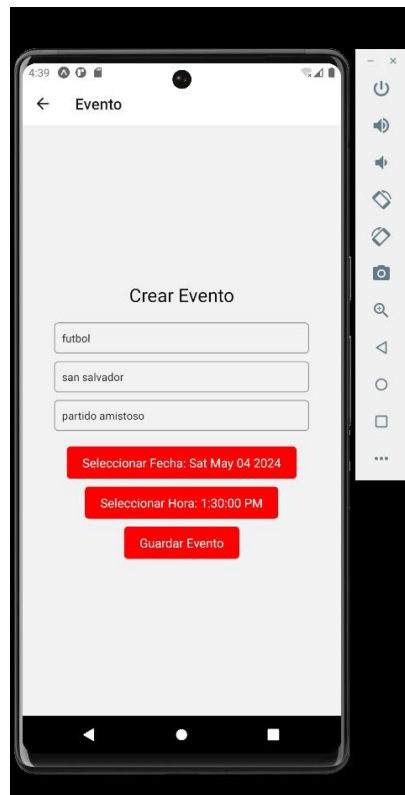
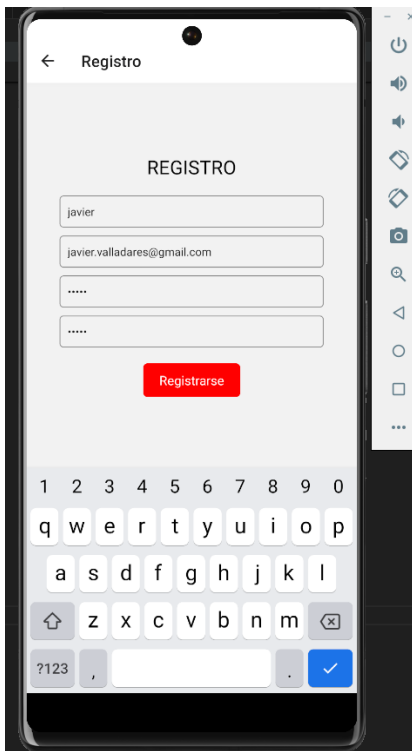


```
1 import React from 'react';
2 import { NavigationContainer } from '@react-navigation/native';
3 import { createStackNavigator } from '@react-navigation/stack';
4 import Inicio from './SRC/Inicio';
5 import Registro from './SRC/Registro';
6 import MostrarEvento from './SRC/mostrar';
7 import EditarEvento from './SRC/EditarEvento';
8 import Evento from './SRC/Evento';
9
10 const Stack = createStackNavigator();
11
12 const App = () => {
13   return (
14     <NavigationContainer>
15       <Stack.Navigator initialRouteName="Inicio">
16         <Stack.Screen name="Inicio" component={Inicio} />
17         <Stack.Screen name="Registro" component={Registro} />
18         <Stack.Screen name="Evento" component={Evento} />
19         <Stack.Screen name="MostrarEvento" component={MostrarEvento} />
20         <Stack.Screen name="EditarEvento" component={EditarEvento} />
21       </Stack.Navigator>
22     </NavigationContainer>
23   );
24 };
25
26 export default App;
```

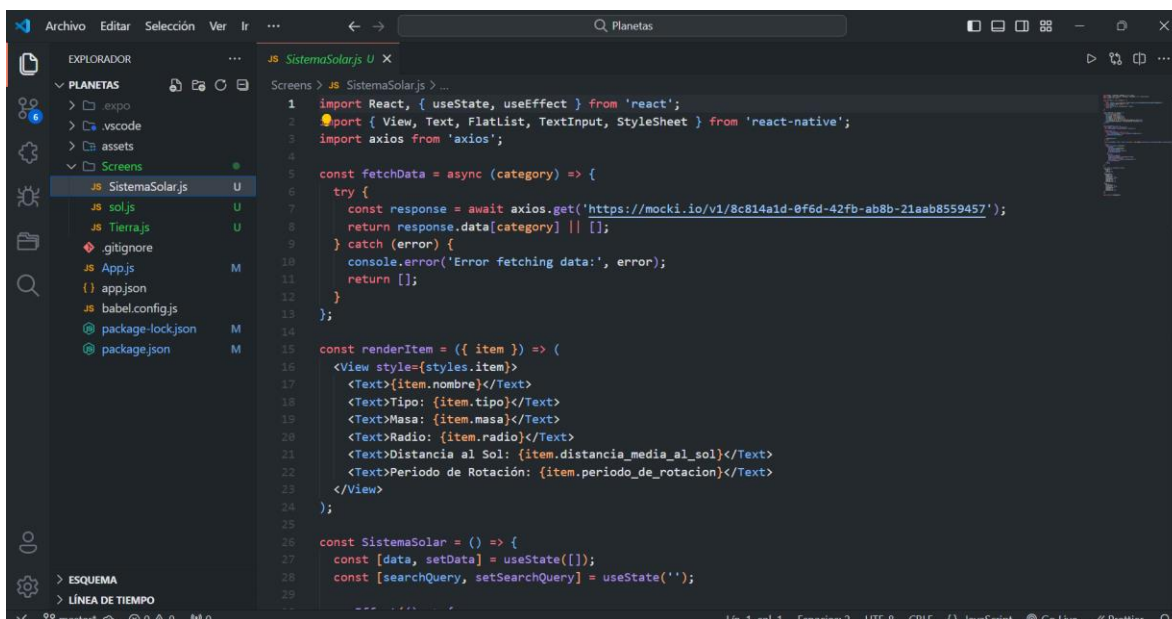


```
5 const Inicio = ({ navigation }) => {
6
7   useEffect(() => {
8     obtenerDatosGuardados();
9   }, []);
10
11   const obtenerDatosGuardados = async () => {
12     try {
13       const savedEmail = await AsyncStorage.getItem('email');
14       const savedPassword = await AsyncStorage.getItem('password');
15
16       if (savedEmail && savedPassword) {
17         setEmail(savedEmail);
18         setPassword(savedPassword);
19       }
20     } catch (error) {
21       console.error('Error al registrar los datos guardados:', error);
22     }
23   };
24
25   const Login = () => {
26     console.log('Email:', email);
27     console.log('Contraseña:', password);
28   };
29
30   return (
31     <View style={styles.container}>
32       <Text style={styles.title}>INICIO</Text>
33     </View>
34   );
35 }
```

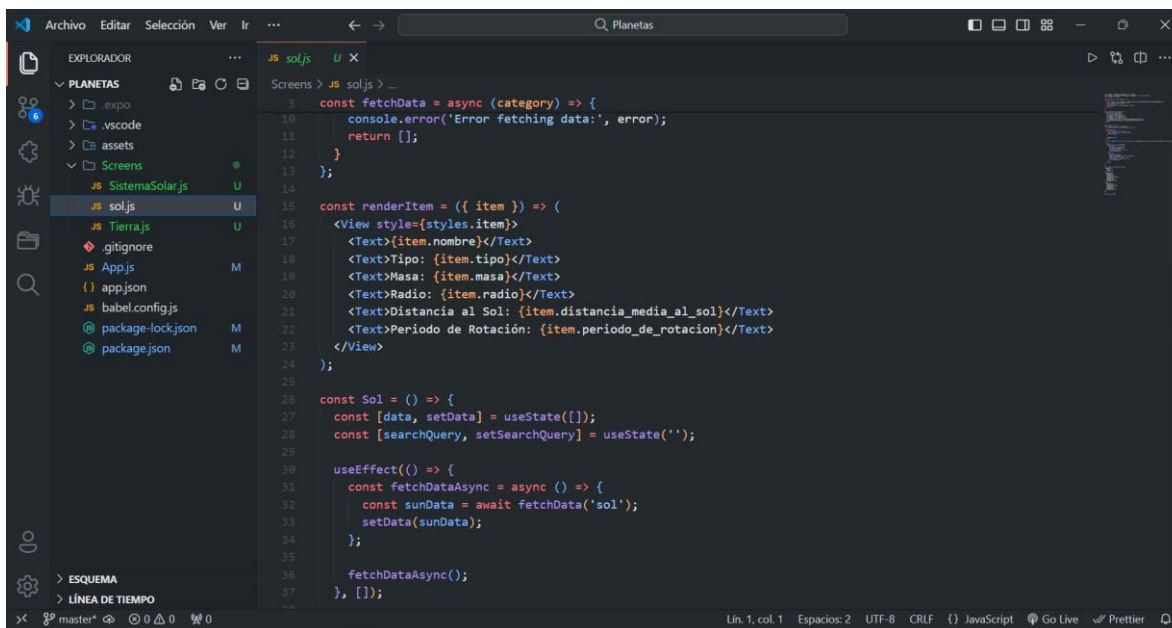
Resultado



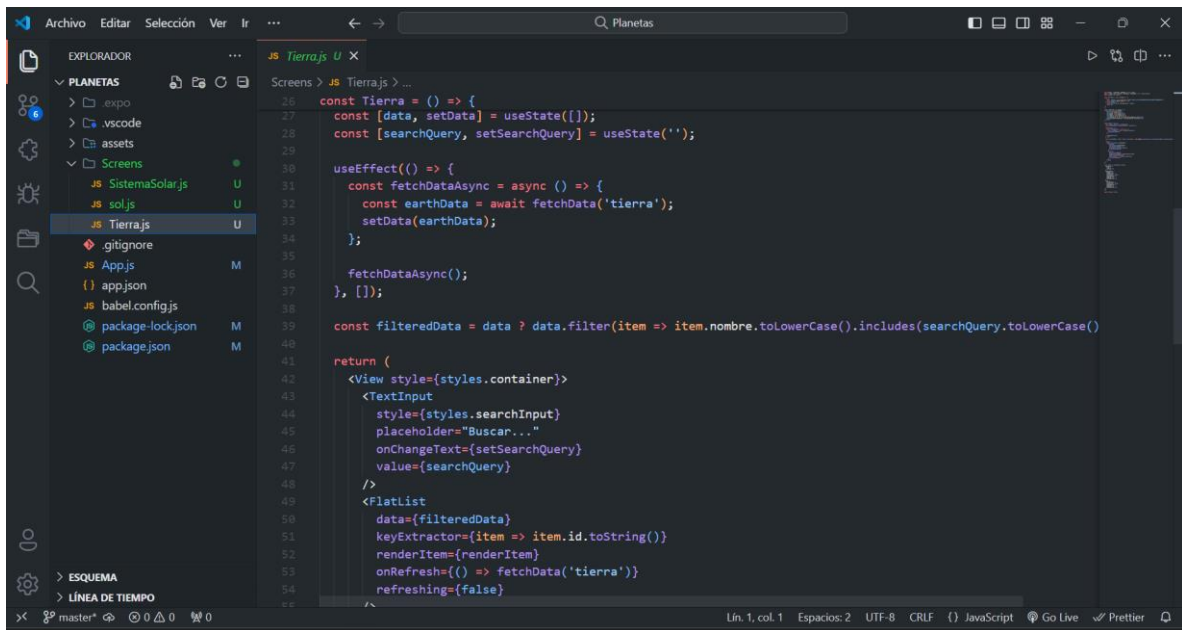
Ejercicio 2



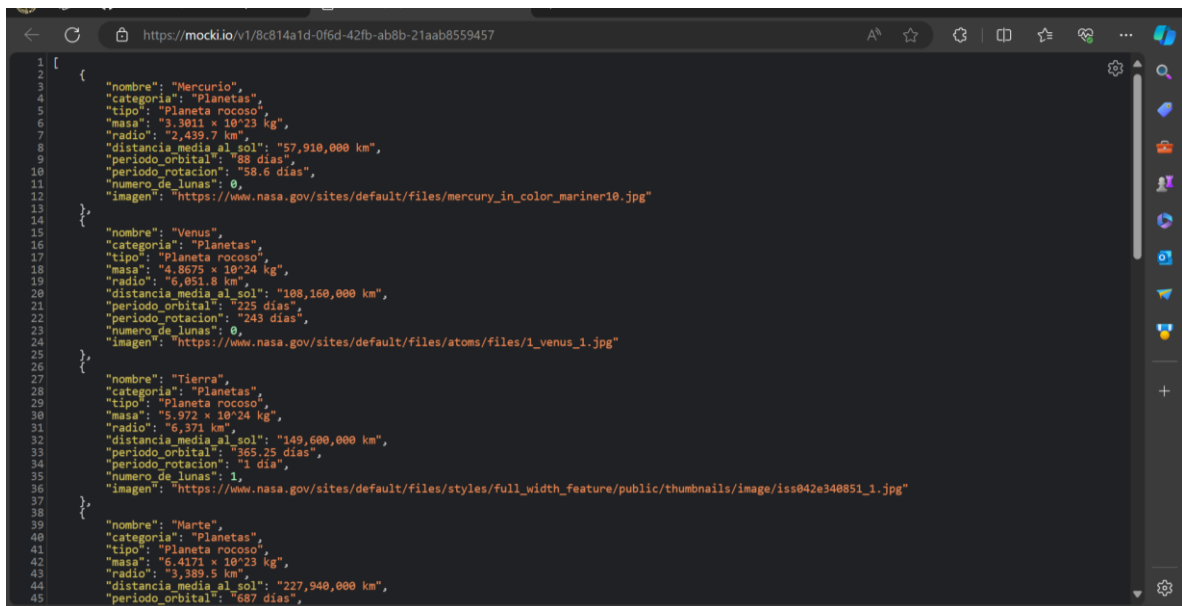
```
1 import React, { useState, useEffect } from 'react';
2 import { View, Text, FlatList, TextInput, StyleSheet } from 'react-native';
3 import axios from 'axios';
4
5 const fetchData = async (category) => {
6   try {
7     const response = await axios.get('https://mocki.io/v1/8c814a1d-0f6d-42fb-ab8b-21aab8559457');
8     return response.data[category] || [];
9   } catch (error) {
10    console.error('Error fetching data:', error);
11    return [];
12   }
13 };
14
15 const renderItem = ({ item }) => (
16   <View style={styles.item}>
17     <Text>{item.nombre}</Text>
18     <Text>Tipo: {item.tipo}</Text>
19     <Text>Masa: {item.masa}</Text>
20     <Text>Radio: {item.radio}</Text>
21     <Text>Distancia al Sol: {item.distancia_medio_al_sol}</Text>
22     <Text>Periodo de Rotación: {item.periodo_de_rotacion}</Text>
23   </View>
24 );
25
26 const SistemaSolar = () => {
27   const [data, setData] = useState([]);
28   const [searchQuery, setSearchQuery] = useState('');
29   ...
30 }
```



```
1 import React, { useState, useEffect } from 'react';
2 import { View, Text, FlatList, TextInput, StyleSheet } from 'react-native';
3 import axios from 'axios';
4
5 const fetchData = async (category) => {
6   try {
7     const response = await axios.get('https://mocki.io/v1/8c814a1d-0f6d-42fb-ab8b-21aab8559457');
8     return response.data[category] || [];
9   } catch (error) {
10    console.error('Error fetching data:', error);
11    return [];
12   }
13 };
14
15 const renderItem = ({ item }) => (
16   <View style={styles.item}>
17     <Text>{item.nombre}</Text>
18     <Text>Tipo: {item.tipo}</Text>
19     <Text>Masa: {item.masa}</Text>
20     <Text>Radio: {item.radio}</Text>
21     <Text>Distancia al Sol: {item.distancia_medio_al_sol}</Text>
22     <Text>Periodo de Rotación: {item.periodo_de_rotacion}</Text>
23   </View>
24 );
25
26 const Sol = () => {
27   const [data, setData] = useState([]);
28   const [searchQuery, setSearchQuery] = useState('');
29
30   useEffect(() => {
31     const fetchDataAsync = async () => {
32       const sunData = await fetchData('sol');
33       setData(sunData);
34     };
35     fetchDataAsync();
36   }, []);
37 }
```

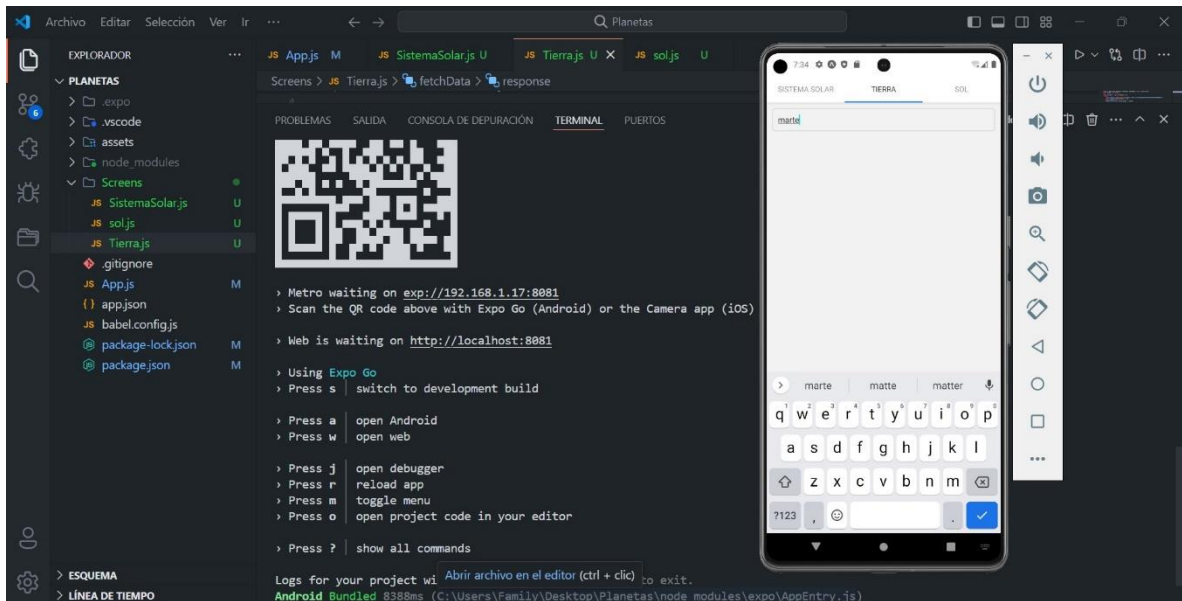


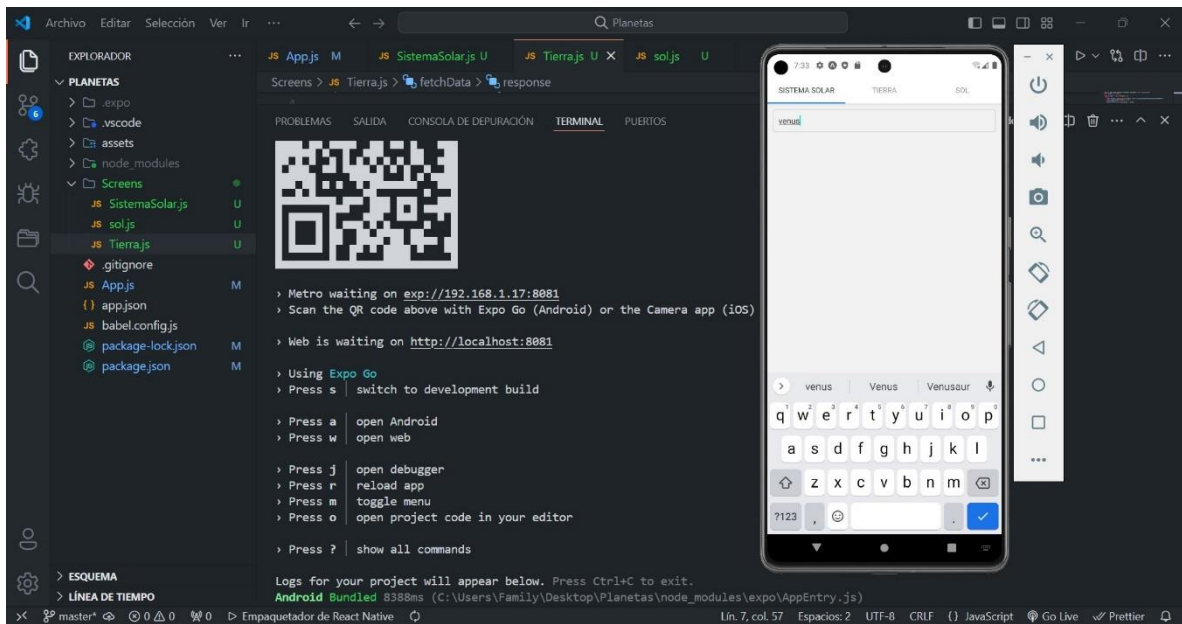
API



```
64 "nombre": "Marte",
65 "categoria": "Planetas",
66 "tipo": "Planeta gaseoso",
67 "masa": "5.6834 x 10^26 kg",
68 "radio": "50,332 km",
69 "distancia_media_al_sol": "1,429,400,000 km",
70 "periodo_orbital": "29.5 años",
71 "periodo_rotacion": "0.45 dias",
72 "numero_de_lunas": 82,
73 "imagen": "https://www.nasa.gov/sites/default/files/styles/full_width_feature/public/thumbnails/image/pia21046.jpg"
74 }
75 "nombre": "Urano",
76 "categoria": "Planetas",
77 "tipo": "Planeta gigante de hielo",
78 "masa": "8.681 x 10^25 kg",
79 "radio": "25,362 km",
80 "distancia_media_al_sol": "2,870,990,000 km",
81 "periodo_orbital": "84 años",
82 "periodo_rotacion": "0.72 dias",
83 "numero_de_lunas": 27,
84 "imagen": "https://www.nasa.gov/sites/default/files/styles/full_width_feature/public/thumbnails/image/pia18182-uranus-voyager1.png"
85 }
86 "nombre": "Neptuno",
87 "categoria": "Planetas",
88 "tipo": "Planeta gigante de hielo",
89 "masa": "1.0243 x 10^26 kg",
90 "radio": "24,622 km",
91 "distancia_media_al_sol": "4,504,300,000 km",
92 "periodo_orbital": "164.8 años",
93 "periodo_rotacion": "0.67 dias",
94 "numero_de_lunas": 14,
95 "imagen": "https://www.nasa.gov/sites/default/files/styles/full_width_feature/public/thumbnails/image/neptune.png"
96 }
97 "nombre": "Sol",
98 "categoria": "Estrella",
99 "tipo": "Estrella enana amarilla",
100 "masa": "1.989 x 10^30 kg",
101 "radio": "695,700 km",
102 "distancia_media_a_la_Via_Lactea": "27,000 años luz",
103 "periodo_rotacion": "24.47 dias",
104 "edad_aproximada": "4.6 mil millones de años",
105 "temperatura_superficie": "5,500°C",
106 "imagen": "https://www.nasa.gov/sites/default/files/styles/full_width_feature/public/thumbnails/image/eso0932a.jpg"
107 }
108 }
```

Resultados





NOTA: hubo un problema con este ejercicio, al momento de ejecutar corre todo bien pero no muestra la información de la API y el código no muestra errores, no pudimos encontrar una solución a este problema porque la API no muestra la información a pesar que la API no indica que tiene errores y que todo funciona correctamente pero al ejecutar no se comunica