

```
// Code: App.tsx
import {NavigationContainer} from '@react-navigation/native';
import {createNativeStackNavigator} from
 '@react-navigation/native-stack';
import { HomeScreen } from './views/home/home';
import Login from './views/home/login';
import Dashboard from './views/home/dashboard';

export type RootStackParamList = {
  Home: undefined;
  Login: undefined;
  Dashboard: { userName: string };
};

export default function App() {
  const Stack = createNativeStackNavigator <RootStackParamList>();
  return (

    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home" component={HomeScreen} />
        <Stack.Screen name="Login" component={Login} />
        <Stack.Screen name="Dashboard" component={Dashboard} />
      </Stack.Navigator>
    </NavigationContainer>

  );
}
```

## home

```
// code: HomeScreen.tsx
import React from 'react'
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View, Image, Button} from 'react-native';
import { useNavigation } from '@react-navigation/native';
import { NativeStackNavigationProp } from
 '@react-navigation/native-stack';
import { RootStackParamList } from '../App';

export const HomeScreen = () => {
```

```

    const navigation =
useNavigation<NativeStackNavigationProp<RootStackParamList>>();
return (
<View style={styles.container}>
  <Image
    style={styles.imageBackground}
    source={require('../assets/1.jpg')}
  />
  <Text
    style={{
      position: 'absolute',
      top: 100,
      fontSize: 40,
      fontWeight: 'bold',
      color: 'white',
    }}
    >Home Screen</Text>

<View style={styles.button}>
  <Button
    title="Entrar"
    onPress={() => {
      navigation.navigate('Login');
    }}
  />
</View>

  <StatusBar style="auto" />
</View>
);
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  imageBackground: {
    width: "100%",
    height: "100%",
  },
},

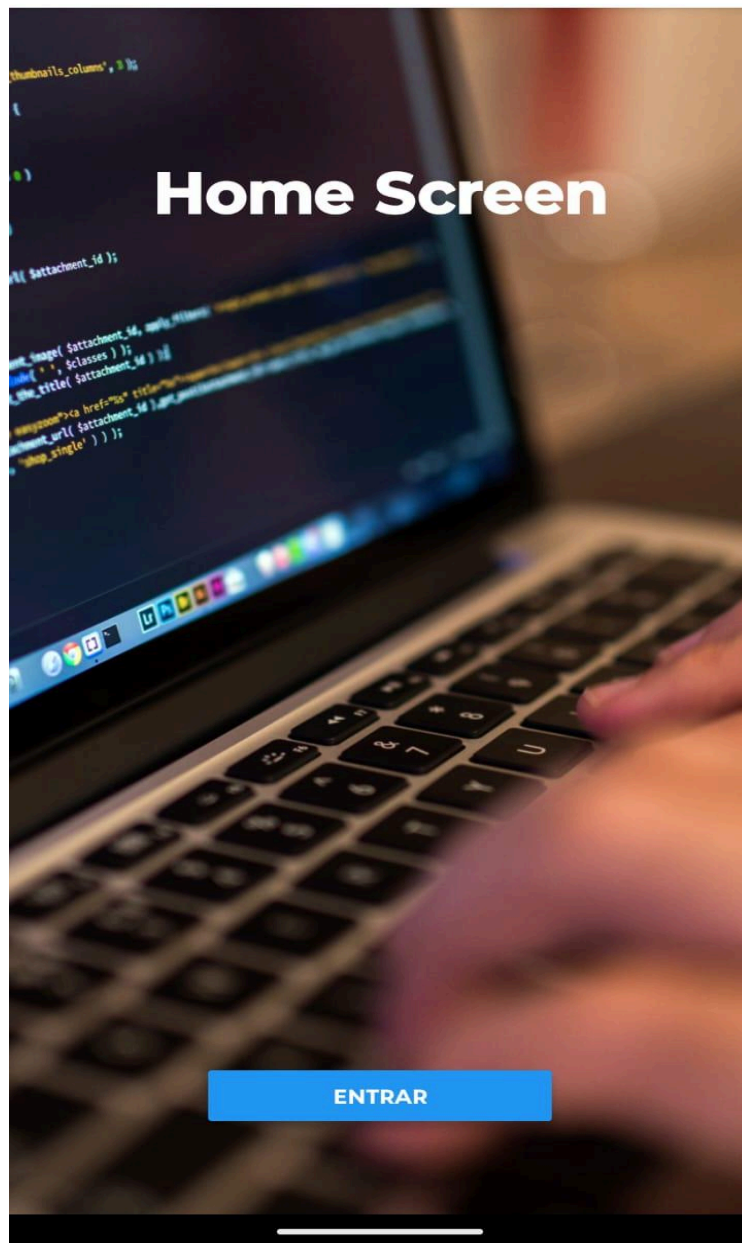
```

```
button: {  
  
    position: 'absolute',  
    bottom: 50,  
    width: 200,  
    height: 50,  
}  
});
```

7:44    

      45 %

Home



## Login

```
// code : Login.tsx
import React, { useState } from 'react';
import { View, TextInput, Button, StyleSheet, Text } from
'react-native';
import { useNavigation } from '@react-navigation/native';
import { NativeStackNavigationProp } from
 '@react-navigation/native-stack';
import { RootStackParamList } from '../App';
const Login = () => {
  const navigation =
useNavigation<NativeStackNavigationProp<RootStackParamList>>();
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');

  const handleLogin = async () => {
    // Add your login logic here
    console.log('Username:', username);
    console.log('Password:', password);
    await
fetch('https://rn1nb289-9000.use2.devtnnls.ms/auth/login', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ username, password }),
    }).then((response) => response.json()).then((data) => {
      console.log('Success:', data.accessToken);
      if (data.accessToken) {
        navigation.navigate('Dashboard', { userName: username
});
      } else {
        alert(data.error);
      }
    })
    .catch((error) => {
      console.error('Error:', error);
      alert("Error: " + error);
    })
  };
};
```

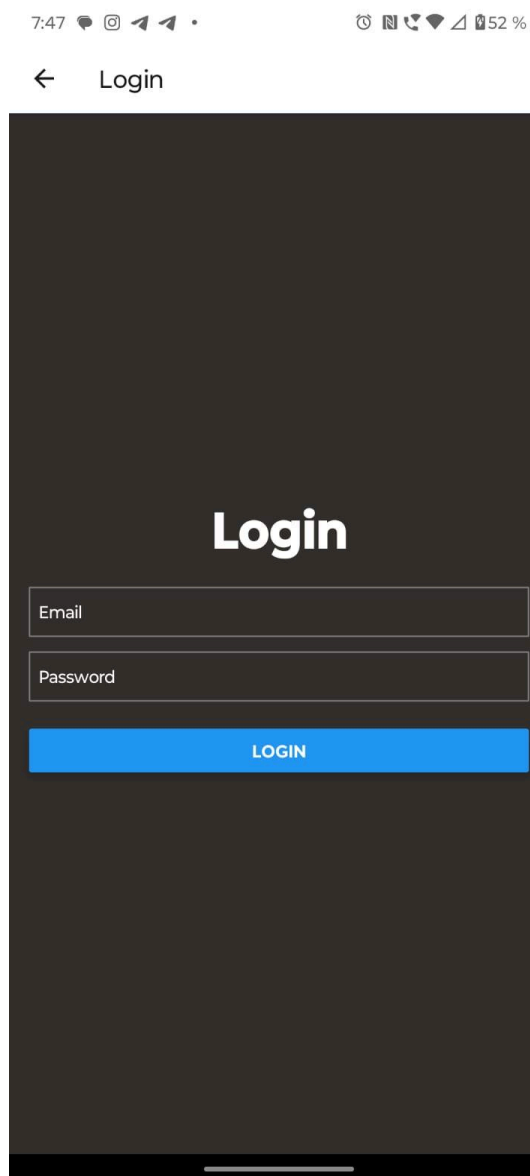
```

return (
  <View style={styles.container}>
    <Text style={styles.Title}>Login</Text>
    <TextInput
      style={styles.input}
      placeholder="Email"
      placeholderTextColor={'white'}
      value={username}
      onChangeText={setUsername}
    />
    <TextInput
      style={styles.input}
      placeholder="Password"
      placeholderTextColor={'white'}
      secureTextEntry
      value={password}
      onChangeText={setPassword}
    />
    <View style={styles.Boton}>
      <Button title="Login" onPress={handleLogin} />
    </View>
  </View>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    padding: 16,
    backgroundColor: '#332f2c',
  },
  input: {
    width: '100%',
    height: 40,
    borderColor: 'gray',
    borderWidth: 1,
    marginBottom: 12,
    paddingHorizontal: 8,
    color : 'white',
  },
});

```

```
    },  
    Title: {  
      fontSize: 40,  
      fontWeight: 'bold',  
      color: 'white',  
      marginBottom: 20,  
    },  
    Boton: {  
      width: '100%',  
      marginTop: 10,  
    },  
  },  
));  
  
export default Login;
```



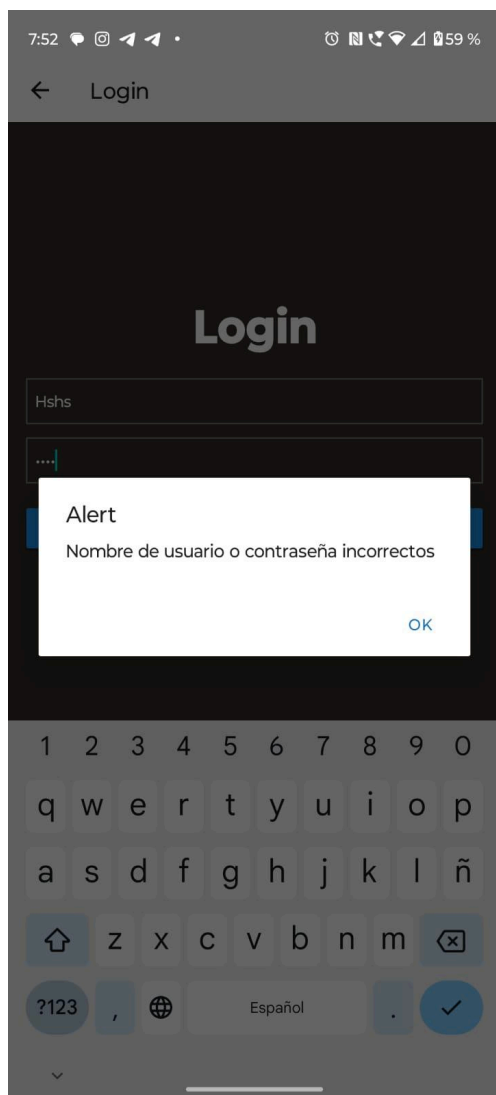
LOG Username: Hshs  
LOG Password: Bsjs  
LOG Success: undefined  
LOG Username: Hshs  
LOG Password: Bsjs  
LOG Success: {"error": "Nombre de usuario o contraseña incorrectos"}  
LOG Username: Hshs  
LOG Password: Bsjs  
LOG Success: {"error": "Nombre de usuario o contraseña incorrectos"}

```
estoy logeando { username: 'Vajdj', password  
: 'Hshsia' }  
Usuario encontrado: []  
usuario no encontrado []  
estoy logeando { username: 'Hshs', password:  
'Bsjs' }  
Usuario encontrado: []  
usuario no encontrado []  
estoy logeando { username: 'Hshs', password:  
'Bsjs' }  
Usuario encontrado: []  
usuario no encontrado []  
estoy logeando { username: 'Hshs', password:  
'Bsjs' }  
Usuario encontrado: []  
usuario no encontrado []
```

JS invoicing.js  
JS persroutes.js  
JS proveedor.js  
JS rolroutes.js  
JS routes.js  
JS index.js

{ } package-lock.json  
{ } package.json  
≡ request.http

> ESQUEMA  
> LÍNEA DE TIEMPO  
> SERVERS



Username: maqui2807

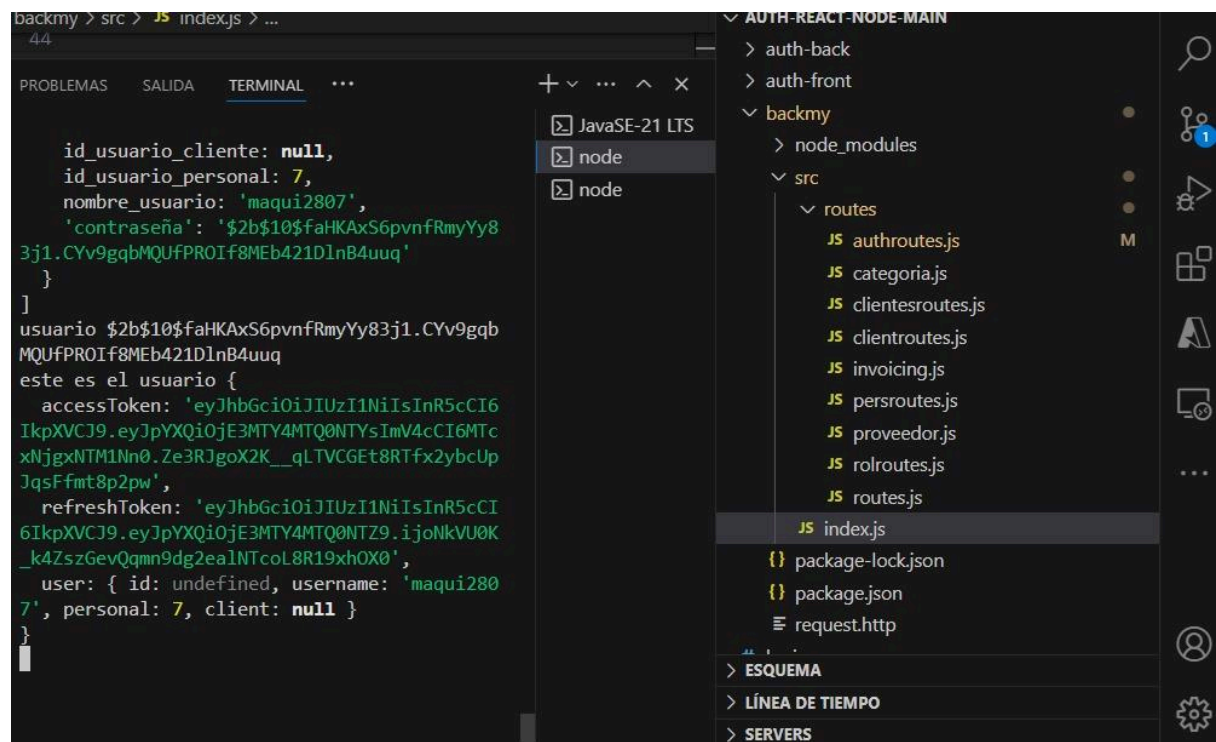
LOG Password: asdf12234

LOG Success: {"accessToken":

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOiJlMTY4MTQ0NTYsImV4cCI6MTcxNjgxNTM1Nn0.Ze3RJgoX2K\_\_qLTVCGEt8RTfx2ybcUpJqsFfmt8p2pw", "refreshToken":

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOiJlMTY4MTQ0NTZ9.ijoNkVU0K\_k4ZszGevQqmn9dg2ea1NTcoL8R19xhOX0", "user": {"client": null, "personal": 7,

"username": "maqui2807"}}}



```
backmy > src > JS index.js > ...
44

id_usuario_cliente: null,
id_usuario_personal: 7,
nombre_usuario: 'maqui2807',
'contraseña': '$2b$10$faHKAxS6pvnfRmyYy83j1.CYv9gqbMQUfPROIf8MEb421DlnB4uuq'
}
]
usuario $2b$10$faHKAxS6pvnfRmyYy83j1.CYv9gqbMQUfPROIf8MEb421DlnB4uuq
este es el usuario {
  accessToken: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOiJlMTY4MTQ0NTYsImV4cCI6MTcxNjgxNTM1Nn0.Ze3RJgoX2K__qLTVCGEt8RTfx2ybcUpJqsFfmt8p2pw',
  refreshToken: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOiJlMTY4MTQ0NTZ9.ijoNkVU0K_k4ZszGevQqmn9dg2ea1NTcoL8R19xhOX0',
  user: { id: undefined, username: 'maqui2807', personal: 7, client: null }
}

AUTH-REACT-NODE-MAIN
> auth-back
> auth-front
< backmy
  < node_modules
    < src
      < routes
        JS authroutes.js
        JS categoria.js
        JS clientesroutes.js
        JS clientroutes.js
        JS invoicing.js
        JS persroutes.js
        JS proveedor.js
        JS rolroutes.js
        JS routes.js
      JS index.js
      {} package-lock.json
      {} package.json
      request.http
> ESQUEMA
> LÍNEA DE TIEMPO
> SERVERS
```

navega al dashboard



## Dashboard

```
import React, { useEffect, useState } from 'react';
import { View, Text, FlatList, StyleSheet } from 'react-native';

interface Product {
  ID_Producto: number;
  Nombre: string;
  Precio: number;
  Cantidad: number;
  ID_categoria: number;
  Nombre_Categoria: string;
}

const Dashboard: React.FC = () => {
  const [products, setProducts] = useState<Product[]>([]);

  useEffect(() => {
    // Fetch products data from API or database
    // Replace the API_URL with your actual API endpoint
    fetch('https://rn1nb289-9000.use2.devtnnls.ms/productos')
      .then((response) => response.json())
      .then((data) => setProducts(data))
      .catch((error) => console.error(error));
  }, []);

  const renderItem = ({ item }: { item: Product }) => (
    <View style={styles.productItem}>
      <Text>{item.Nombre}</Text>
      <Text>Price: {item.Precio}</Text>
      <Text>Quantity: {item.Cantidad}</Text>
    </View>
  );
```

```

return (
  <View style={styles.container}>

    <FlatList
      data={products}
      renderItem={renderProductItem}
      keyExtractor={(item) => item.ID_Producto.toString()}
    />
  </View>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 16,
  },
  greeting: {
    fontSize: 24,
    fontWeight: 'bold',
    marginBottom: 16,
  },
  productItem: {
    marginBottom: 8,
    padding: 8,
    backgroundColor: '#f0f0f0',
    borderRadius: 4,
  },
});

export default Dashboard;

```

Unamed\minimercad2\producto\ - HeidiSQL 12.3.0.6589

Archivo Editar Buscar Consulta Herramientas Ir a Ayuda

Host: 127.0.0.1 Base de datos: minimercad2 Tabla: producto Datos Consulta

Filtro de bases de d Filtro de tablas

minimercad2 368,0 KiB

- categoria 16,0 KiB
- cliente 32,0 KiB
- factura 48,0 KiB
- personal 48,0 KiB
- producto** 32,0 KiB
- producto\_factura 48,0 KiB
- rol 16,0 KiB
- sucursal 16,0 KiB
- sucursal\_producto 48,0 KiB
- tokens 16,0 KiB
- users 48,0 KiB

minimercado  
minimercado1  
mysql  
performance\_schema  
phpmyadmin  
redsocial  
sakila

minimercad: >> Siguietes Mostrar todo Ordenación Columnas (5/5) Filtro

ID_Producto	ID_Categoria	Nombre	Precio	Cantidad
2	5	leche	2.000	48
3	6	carne de cerdo	5.500	110
4	8	jabon rey	2.000	500
5	9	tomates	500	500
6	7	arroz	2.100	300

← Dashboard

leche  
Price: 2000  
Quantity: 48

carne de cerdo  
Price: 5500  
Quantity: 110

jabon rey  
Price: 2000  
Quantity: 500

tomates  
Price: 500  
Quantity: 500

arroz  
Price: 2100  
Quantity: 300