1. Crear proyecto login-app typescript blank
2. Ejecutar los pasos de la práctica anterior
3. Instalar las dependencias de navegación

npx expo install @react-navigation/native

npx expo install react-native-screens react-native-safe-area-context

npm install @react-navigation/native-stack

Si se ejecuta la web requiere:

npx expo install react-dom react-native-web

1. Crear carpeta src
2. Crear carpeta navigation, crear archivo AppNavigator.tsx

import React from 'react';

import { NavigationContainer } from '@react-navigation/native';

import { createNativeStackNavigator } from '@react-navigation/native-stack';

import LoginScreen from '../screens/LoginScreen';

import HomeScreen from '../screens/HomeScreen';

const Stack = createNativeStackNavigator();

export default function AppNavigator() {

  return (

    <NavigationContainer>

      <Stack.Navigator initialRouteName="Login">

        <Stack.Screen name="Login" component={LoginScreen} options={{ title: 'Iniciar sesión' }} />

        <Stack.Screen name="Home" component={HomeScreen} options={{ title: 'Inicio' }} />

      </Stack.Navigator>

    </NavigationContainer>

  );

}

1. Crear carpeta screens
2. Crear archivo HomeScreen.tsx

import React from 'react';

import { View, Text, StyleSheet } from 'react-native';

export default function HomeScreen() {

return (

<View style={styles.container}>

<Text testID="home-text">Bienvenido al sistema 🎉</Text>

</View>

);

}

const styles = StyleSheet.create({

container: { flex: 1, justifyContent: 'center', alignItems: 'center' },

});

1. Crear LoginScreen.tsx

import React, { useState } from 'react';

import { View, TextInput, Button, Text, StyleSheet } from 'react-native';

import { useNavigation } from '@react-navigation/native';

export default function LoginScreen() {

const navigation = useNavigation<any>();

const [email, setEmail] = useState('');

const [password, setPassword] = useState('');

const [error, setError] = useState('');

const handleLogin = () => {

if (email === 'admin@test.com' && password === '1234') {

setError('');

navigation.navigate('Home');

} else {

setError('Credenciales incorrectas');

}

};

return (

<View style={styles.container}>

<Text>Iniciar sesión</Text>

<TextInput

placeholder="Correo"

value={email}

onChangeText={setEmail}

accessibilityLabel="input-email"

style={styles.input}

/>

<TextInput

placeholder="Contraseña"

value={password}

onChangeText={setPassword}

secureTextEntry

accessibilityLabel="input-password"

style={styles.input}

/>

<Button title="Ingresar" onPress={handleLogin} />

{error ? <Text testID="error-message">{error}</Text> : null}

</View>

);

}

const styles = StyleSheet.create({

container: { flex: 1, justifyContent: 'center', alignItems: 'center', padding: 20 },

input: {

borderWidth: 1,

borderColor: '#ccc',

marginVertical: 8,

padding: 8,

width: '100%',

borderRadius: 5,

},

});

1. Crear carpeta \_\_test\_\_, crear archivo test LoginScreen.test.tsx

import React from 'react';

import { render, fireEvent, waitFor } from '@testing-library/react-native';

import { NavigationContainer } from '@react-navigation/native';

import { createNativeStackNavigator } from '@react-navigation/native-stack';

import LoginScreen from '../src/screens/LoginScreen';

import HomeScreen from '../src/screens/HomeScreen';

const Stack = createNativeStackNavigator();

const renderWithNavigation = () => {

return render(

<NavigationContainer>

<Stack.Navigator>

<Stack.Screen name="Login" component={LoginScreen} />

<Stack.Screen name="Home" component={HomeScreen} />

</Stack.Navigator>

</NavigationContainer>

);

};

describe('LoginScreen Navigation', () => {

it('redirige a Home si las credenciales son correctas', async () => {

const { getByLabelText, getByText, queryByTestId } = renderWithNavigation();

const emailInput = getByLabelText('input-email');

const passwordInput = getByLabelText('input-password');

const loginButton = getByText('Ingresar');

fireEvent.changeText(emailInput, 'admin@test.com');

fireEvent.changeText(passwordInput, '1234');

fireEvent.press(loginButton);

await waitFor(() => {

expect(queryByTestId('home-text')).toBeTruthy();

});

});

it('muestra mensaje de error si las credenciales son incorrectas', async () => {

const { getByLabelText, getByText, queryByTestId } = renderWithNavigation();

const emailInput = getByLabelText('input-email');

const passwordInput = getByLabelText('input-password');

const loginButton = getByText('Ingresar');

fireEvent.changeText(emailInput, 'usuario@test.com');

fireEvent.changeText(passwordInput, 'wrong');

fireEvent.press(loginButton);

await waitFor(() => {

expect(queryByTestId('error-message')).toBeTruthy();

expect(getByText('Credenciales incorrectas')).toBeTruthy();

});

});

});