Universidad Tecnológica de Tijuana

TSU en Tecnologías de la Información Área Desarrollo de Software Multiplataforma

Nombres:

Dominguez Diaz Alma Juanita.

Materia:

Desarrollo Móvil Multiplataforma.

Examen 2

Cuatrimestre: Quinto. ***Grupo:*** B.

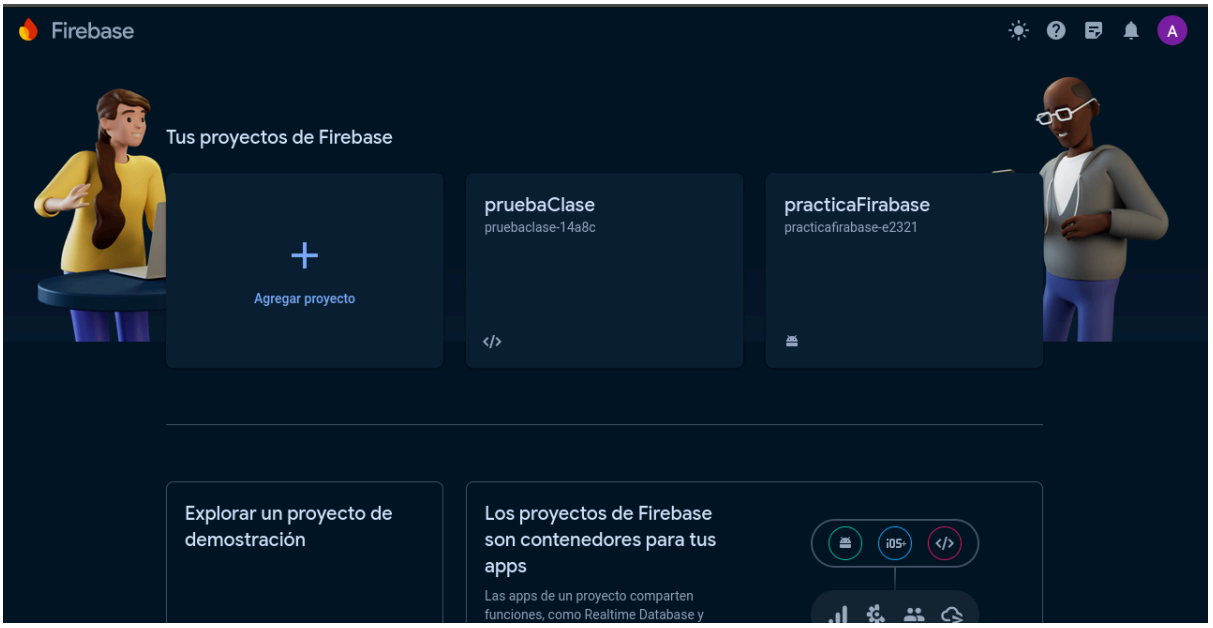
Profesor:

Dr. Ray Brunet Galaviz Parra.

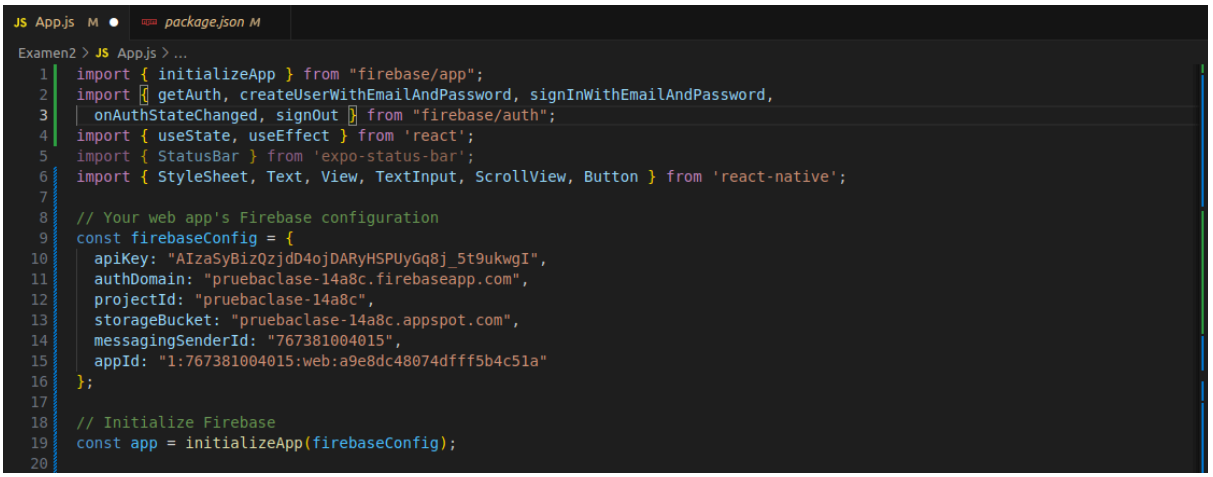
Fecha de realización:

06 de Julio del 2024.

Examen del segundo parcial
Para realizar el examen se necesito crear un proyecto en firebase y registrar tu app y configurar el app para que la autenticación sea con correo electrónico y contraseña.



Al registrar tu app se te dará el código para agregar el SDK de Firebase a nuestra aplicación.
Se requiere descargar las siguientes dependencias a la aplicación de expo:
firebase
@react-native-firebase/app
@react-native-firebase/auth



```
import { initializeApp } from "firebase/app";
import { getAuth, createUserWithEmailAndPassword, signInWithEmailAndPassword,
  onAuthStateChanged, signOut } from "firebase/auth";
import { useState, useEffect } from 'react';
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View, TextInput, ScrollView, Button } from 'react-native';

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyBizQzjdD4ojDARyHSPUyGq8j_5t9ukwgI",
  authDomain: "pruebaclase-14a8c.firebaseapp.com",
  projectId: "pruebaclase-14a8c",
  storageBucket: "pruebaclase-14a8c.appspot.com",
  messagingSenderId: "767381004015",
  appId: "1:767381004015:web:a9e8dc48074dfff5b4c51a"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

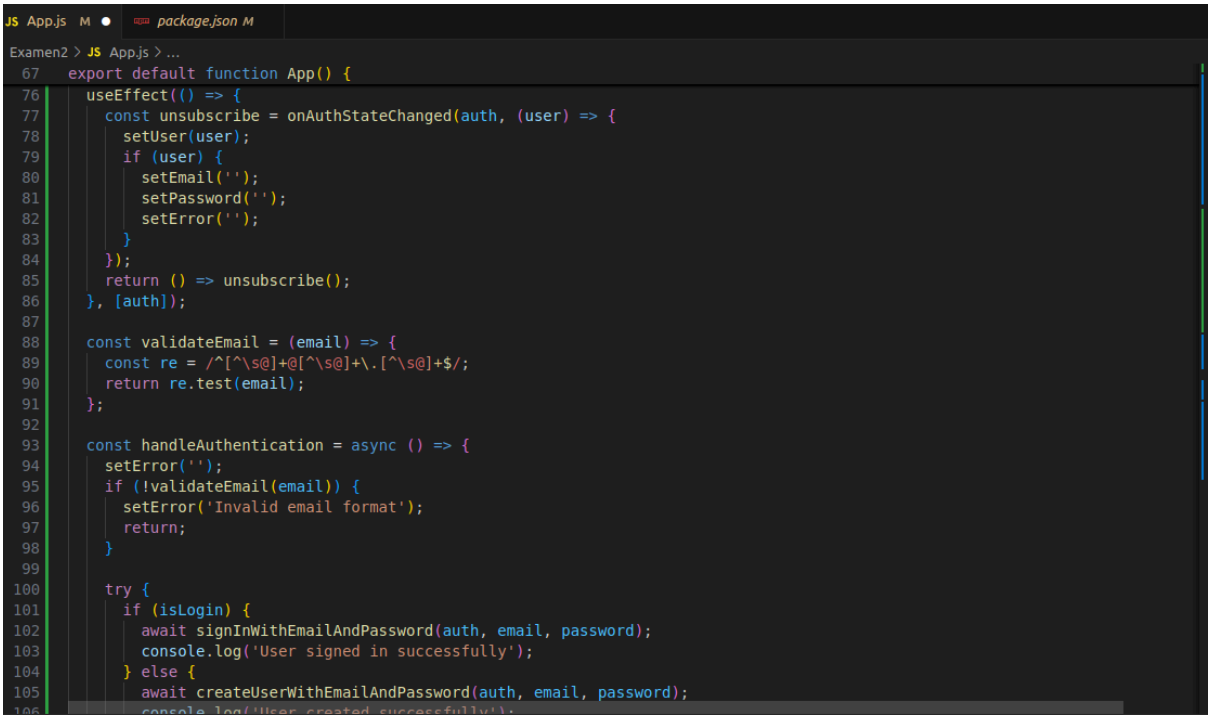
El AuthScreen que maneja la autenticación de usuarios. Dependiendo si el usuario quiere iniciar sesion o registrarse, muestra un formulario con campos de correo electrónico y contraseña.

```
JS App.js M ● package.json M
Examen2 > JS App.js > ...
20  const AuthScreen = ({ email, setEmail, password, setPassword, isLogin, setIsLogin, handleAuthentication, error }) => {
21    return (
22      <View style={styles.authContainer}>
23        <Text style={styles.title}>{isLogin ? 'Sign In' : 'Sign Up'}</Text>
24        {error && <Text style={styles.errorText}>{error}</Text>}
25        <TextInput style={styles.input}
26          value={email}
27          placeholder="Email"
28          onChangeText={setEmail}
29          autoCapitalize="none"
30          keyboardType="email-address"
31        />
32        <TextInput style={styles.input}
33          value={password}
34          placeholder="Password"
35          onChangeText={setPassword}
36          secureTextEntry
37        />
38        <View style={styles.buttonContainer}>
39          <Button title={isLogin ? 'Sign In' : 'Sign Up'}
40            color="#0c870c"
41            onPress={handleAuthentication}
42          />
43        </View >
44        <View style={styles.bottomContainer}>
45          <Text style={styles.toggleText} onPress={() => setIsLogin(!isLogin)}>
46            {isLogin ? 'Need an account? Sign up' : 'Already have an account? Sign In'}
47          </Text>
48        </View>
49      </View>
50    );
```

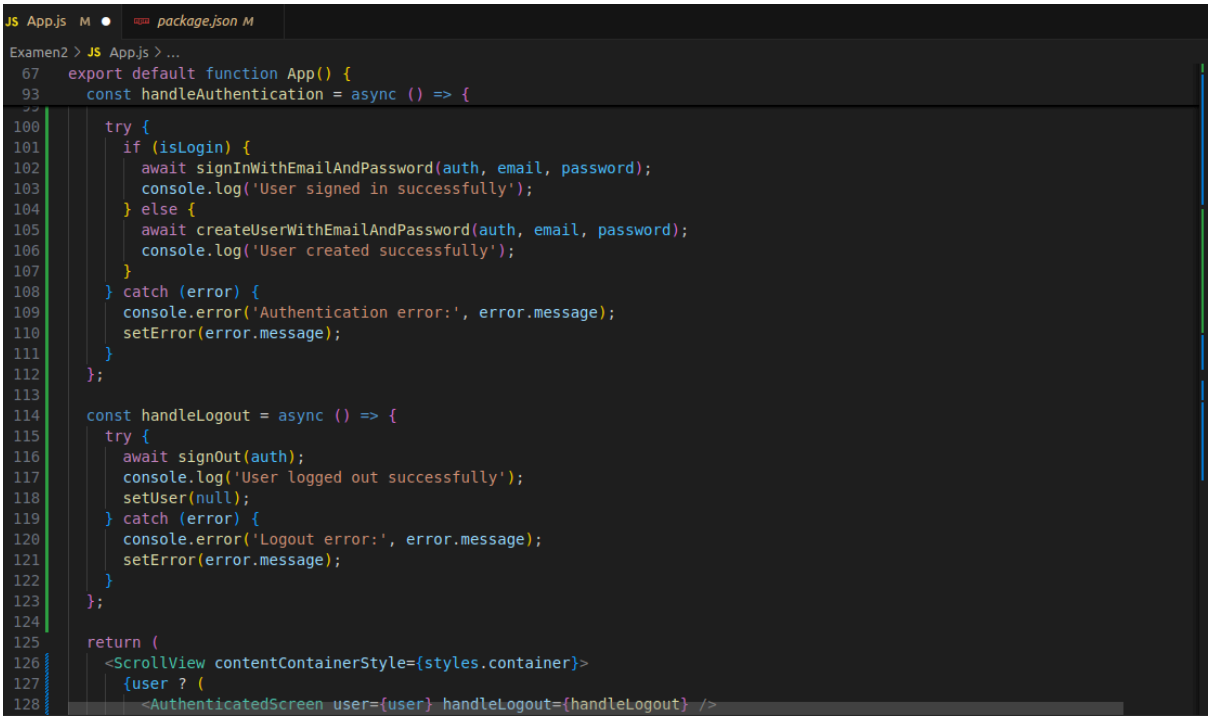
En AuthenticatedScreen es la vista cuando se inicia sesión y al registrarse con éxito; incluye un botón para hacer logout.
En la function App se almacena en una constante el email y contraseña, también se almacena los mensajes de error, y determina si el usuario está logueado o no.

```
JS App.js M ● package.json M
Examen2 > JS App.js > ...
53  const AuthenticatedScreen = ({ user, handleLogout }) => {
54    return (
55      <View style={styles.authContainer}>
56        <Text style={styles.title}>Welcome</Text>
57        <Text style={styles.emailText}>{user.email}</Text>
58        <Button
59          title="Logout"
60          color="#0e0bd9"
61          onPress={handleLogout}
62        />
63      </View>
64    );
65  }
66
67  export default function App() {
68    const [email, setEmail] = useState('');
69    const [password, setPassword] = useState('');
70    const [user, setUser] = useState(null);
71    const [isLogin, setIsLogin] = useState(true);
72    const [error, setError] = useState('');
73
74    const auth = getAuth(app);
75
76    useEffect(() => {
77      const unsubscribe = onAuthStateChanged(auth, (user) => {
78        setUser(user);
79        if (user) {
80          setEmail('');
81          setPassword('');
82          setError('');
83        }
```

la constante validateEmail se realiza el pattern del email y en handleAuthentication se valida si el email es válido.



Se verifica que si se realizó el login o el registro correctamente. Si no se realiza correctamente manda el mensaje de error.



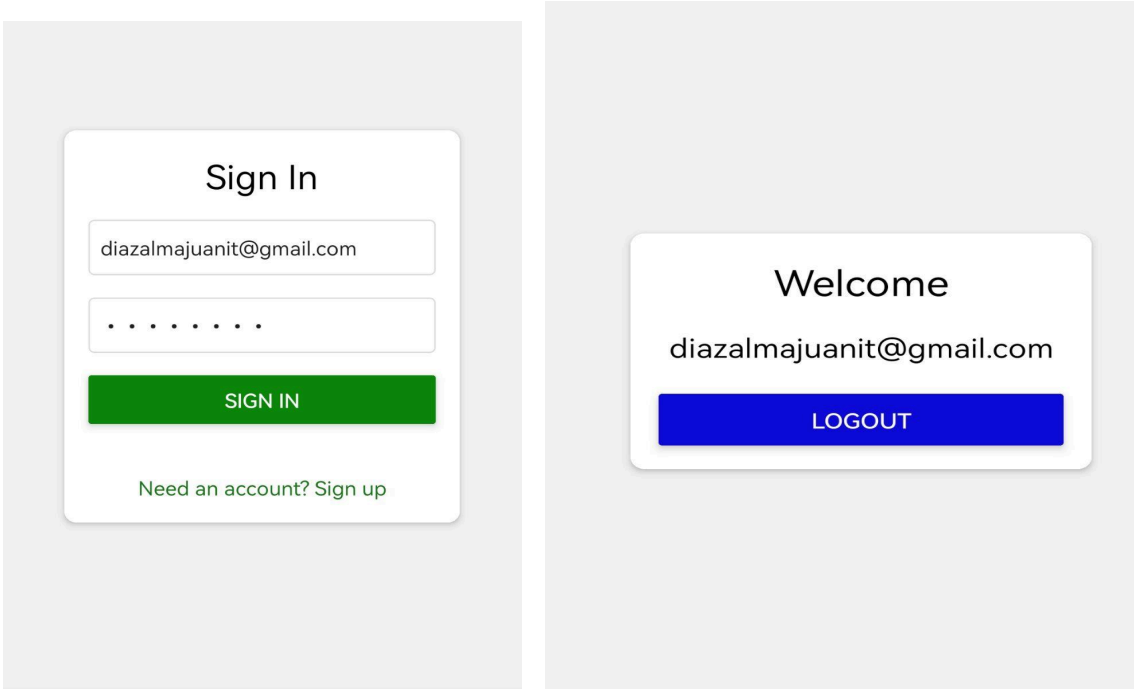
La función App devuelve un JSX que utiliza un ScrollView para contener dos posibles vistas: AuthenticatedScreen y AuthScreen. La vista que se muestra depende de si hay un usuario autenticado o no.

```
  return (
    <ScrollView contentContainerStyle={styles.container}>
      {user ? (
        <AuthenticatedScreen user={user} handleLogout={handleLogout} />
      ) : (
        <AuthScreen
          email={email}
          setEmail={setEmail}
          password={password}
          setPassword={setPassword}
          isLogin={isLogin}
          setIsLogin={setIsLogin}
          handleAuthentication={handleAuthentication}
          error={error}
        />
      )}
    </ScrollView>
  );
}
```

Estos son los estilos aplicados para el login y registro.

```
const styles = StyleSheet.create({
  container: { flexGrow: 1, justifyContent: 'center', alignItems: 'center',
    padding: 16,
    backgroundColor: '#f0f0f0',
  },
  authContainer: { width: '80%', maxWidth: 400, backgroundColor: '#fff', padding: 16,
    borderRadius: 8,
    elevation: 3,
  },
  title: { fontSize: 24, marginBottom: 16,
    textAlign: 'center',
  },
  input: { height: 40, borderColor: '#ddd', borderWidth: 1, marginBottom: 16,
    padding: 8,
    borderRadius: 4,
  },
  buttonContainer: {
    marginBottom: 16,
  },
  toggleText: { color: '#1f781e', textAlign: 'center',
  },
  bottomContainer: { marginTop: 20,
  },
  emailText: { fontSize: 18, textAlign: 'center',
    marginBottom: 20,
  },
  errorText: { color: 'red', marginBottom: 16,
    textAlign: 'center',
  },
});
```

Vista del login de un usuario ya registrado



Prueba del usuario ya registrado.

Vista del registro y evidencia que se registró.