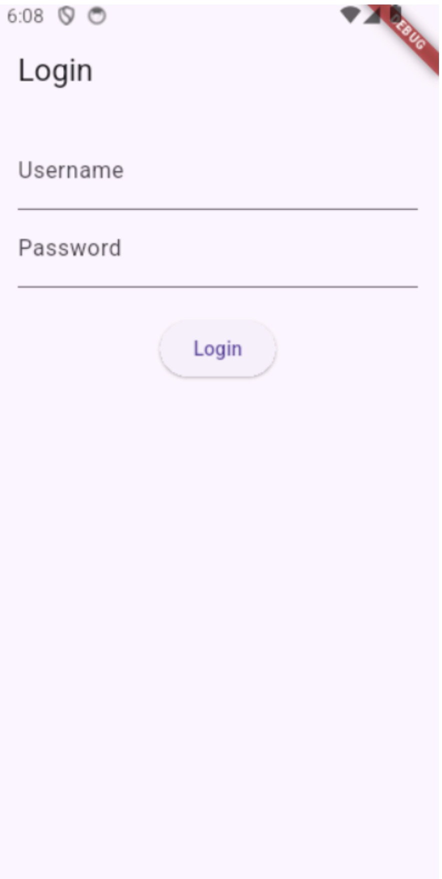
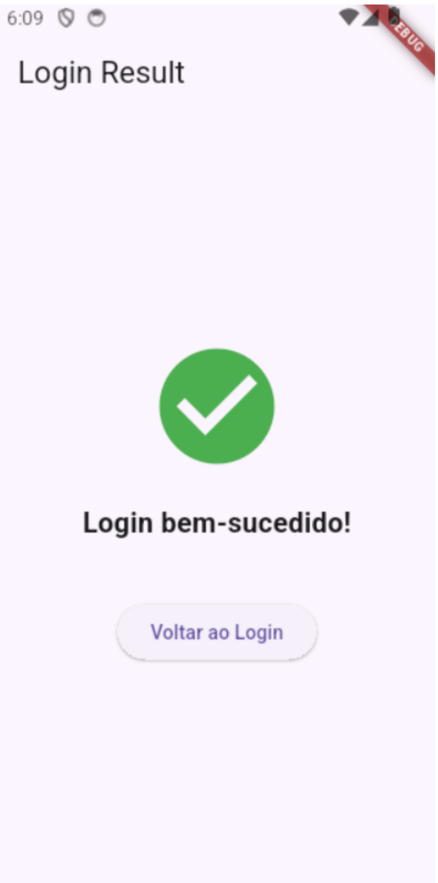
Tarefa 10 - PV26465

Código:

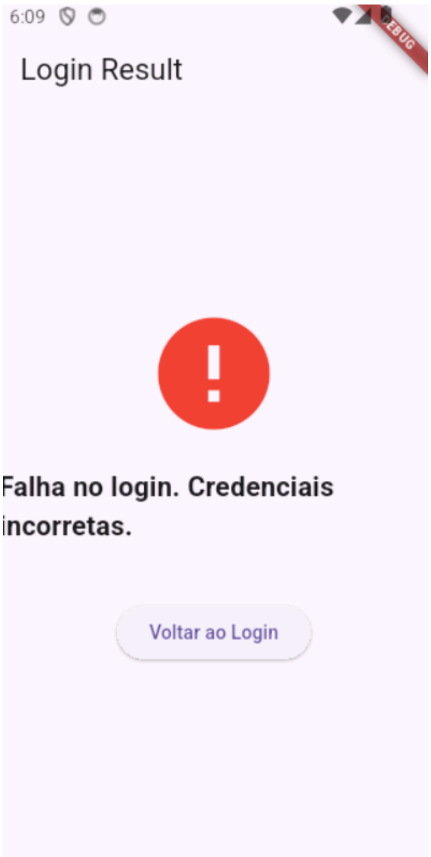
```
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:crypto/crypto.dart';
import 'dart:convert';
import 'package:go_router/go_router.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await registerUser();
  runApp(MyApp());
}

Future<void> registerUser() async {
  final prefs = await SharedPreferences.getInstance();
  const username = 'user';
  const password = 'password';
  final hashedPassword = sha256.convert(utf8.encode(password)).toString();
  await prefs.setString('username', username);
  await prefs.setString('password',
hashedPassword);
}

final _router = GoRouter(
  routes: [
    GoRoute(path: '/', builder: (context,
state) => const LoginPage()),
    GoRoute(
      path: '/success',
      builder: (context, state) => const
HomePage(loginSuccessful: true),
    ),
    GoRoute(
      path: '/failed',
      builder: (context, state) => const
HomePage(loginSuccessful: false),
    ),
  ],
);

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp.router(
      title: 'App login tarefa 10',
      routerConfig: _router,
    );
  }
}

class LoginPage extends StatefulWidget {
  const LoginPage({super.key});

  @override
  _LoginPageState createState() =>
_LoginPageState();
}

class _LoginPageState extends
State<LoginPage> {
  final _formKey = GlobalKey<FormState>();
  String _username = '';
  String _password = '';

  Future<bool> _login() async {
    final prefs = await
SharedPreferences.getInstance();
    final storedUsername =
prefs.getString('username');
    final storedPassword =
prefs.getString('password');
```

```
    final hashedPassword =
sha256.convert(utf8.encode(_password)).toStri
ng();
    return _username == storedUsername &&
hashedPassword == storedPassword;
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Login')),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
          child: Column(
            children: [
              TextFormField(
                decoration:
InputDecoration(labelText: 'Username'),
                onChanged: (value) =>
_username = value,
                validator:
                    (value) => value!.isEmpty
? 'Introduza username' : null,
              ),
              TextFormField(
                decoration:
InputDecoration(labelText: 'Password'),
                obscureText: true,
                onChanged: (value) =>
_password = value,
                validator:
                    (value) => value!.isEmpty
? 'Introduza password' : null,
              ),
              SizedBox(height: 20),
              ElevatedButton(
                onPressed: () async {
                  if
(_formKey.currentState!.validate()) {}
                  final isSucess = await
_login();
                  if (isSucess) {
                    context.go('/success');
                  } else {
                    context.go('/failed');
                  }
                },
                child: Text('Login'),
              ),
            ],
          ),
        ),
      ),
    );
  }
}

class HomePage extends StatelessWidget {
  final bool loginSuccessful;

  const HomePage({super.key, required
this.loginSuccessful});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Login Result'),
        automaticallyImplyLeading: false, //
Remove back button
      ),
      body: Center(
        child: Column(
          mainAxisAlignment:
MainAxisAlignment.center,
          children: [
            Icon(
              loginSuccessful ?
Icons.check_circle : Icons.error,
              color: loginSuccessful ?
Colors.green : Colors.red,
              size: 100,
            ),
            SizedBox(height: 20),
            Text(
              loginSuccessful
                  ? 'Login bem-sucedido!'
                  : 'Falha no login.
Credenciais incorretas.',
              style: TextStyle(fontSize: 20,
fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 40),
            ElevatedButton(
              onPressed: () =>
context.go('/'),
              child: Text('Voltar ao Login'),
            ),
          ],
        ),
      ),
    );
  }
}
```





Login bem-sucedido!



Falha no login. Credenciais incorretas.