

Desenvolvimento de Tela de Login em Flutter com Integração a API

Aluno: Lucas Eduardo Dias Curso: Ciências da Computação RA: 123117336

1. Objetivo O objetivo deste trabalho foi desenvolver uma tela de login utilizando Flutter, com integração a uma API REST. A ideia foi criar uma interface funcional que pudesse se comunicar com um backend para autenticação de usuários, simulando o login em serviços de nuvem.
-

1. Ferramentas Utilizadas

2. Flutter: Framework para desenvolvimento de interfaces multiplataforma.
 3. Dart: Linguagem utilizada pelo Flutter.
 4. VS Code: Ambiente de desenvolvimento integrado (IDE).
 5. API REST (mock): Simulação de backend para testes de login.
 6. Pacote http: Utilizado para enviar requisições HTTP para a API.
-

1. Estrutura do Projeto O projeto foi estruturado da seguinte forma:

2. lib/main.dart: Arquivo principal que inicializa o aplicativo.
 3. lib/pages/login_page.dart: Tela de login do usuário.
 4. lib/services/api_service.dart: Serviço responsável por enviar requisições para a API.
-

1. Desenvolvimento

4.1 Tela de Login A tela de login foi criada em login_page.dart com os seguintes componentes: - Campos de texto para email e senha. - Botão Entrar, que envia os dados para a API. - Botão alternativo de login com Facebook (simbolicamente). - Mensagens de feedback (sucesso ou erro) exibidas via Snackbar.

O layout foi feito usando Column, TextField e ElevatedButton do Flutter, com estilo semelhante à tela de login do Instagram.

4.2 Serviço de API O arquivo api_service.dart contém a classe ApiService, que realiza requisições HTTP. Exemplo da função de login:

```
import 'dart:convert';
import 'package:http/http.dart' as http;

class ApiService {
  final String baseUrl = "https://sua-api-aqui.com";

  Future<bool> login(String email, String senha) async {
    final url = Uri.parse('\$baseUrl/login');
    final response = await http.post(
      url,
      headers: {"Content-Type": "application/json"},
      body: jsonEncode({"email": email, "senha": senha}),
    );
  }
}
```

```
);  
  
if (response.statusCode == 200) {  
  return true; // login bem-sucedido  
} else {  
  return false; // login falhou  
}  
}  
}
```

Observação: Para fins de demonstração, a URL da API foi simulada. O comportamento do app foi testado com respostas fictícias de sucesso e erro.

4.3 Funcionamento 1. O usuário insere email e senha. 2. Ao clicar em "Entrar", o app chama `ApiService.login()`. 3. Com base na resposta da API, é exibida uma mensagem de sucesso ou falha.

-
1. Testes Realizados
 2. A interface abriu corretamente no navegador.
 3. Os campos de email e senha aceitaram entrada de dados.
 4. O botão "Entrar" acionou a função de login corretamente, retornando mensagens simuladas.
 5. O layout se manteve responsivo e semelhante à tela do Instagram.
-

1. Considerações Finais O projeto demonstra:
2. Criação de interface Flutter funcional.
3. Estruturação de chamadas a API usando Dart e http.
4. Facilidade de integração futura com qualquer backend real.

Mesmo sem uma API real em execução, o app já mostra o fluxo completo de login, permitindo expansão futura.