Dating App

Flutter Aplication

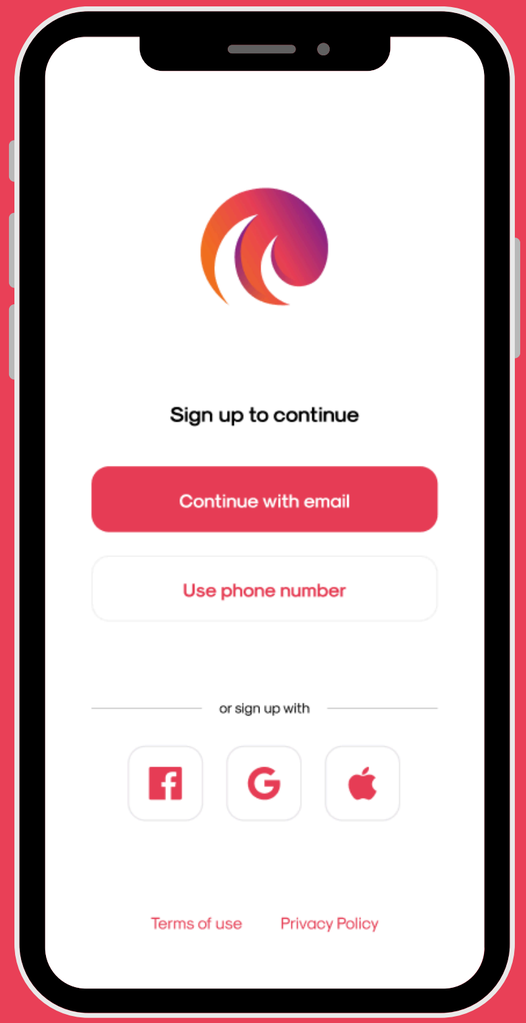
Dating App

Nosso projeto consiste em um aplicativo de namoro, propondo a conexão entre as pessoas através de um sistema de deslize de perfil, permitindo que os usuários encontrem e interajam com outros que também se interessam mutuamente.

Visamos facilitar encontros, amizades e até mesmo relacionamentos amorosos, utilizando interesses mútuos cadastrados no perfil e depois utlizando a seleção por meio de “matchs”

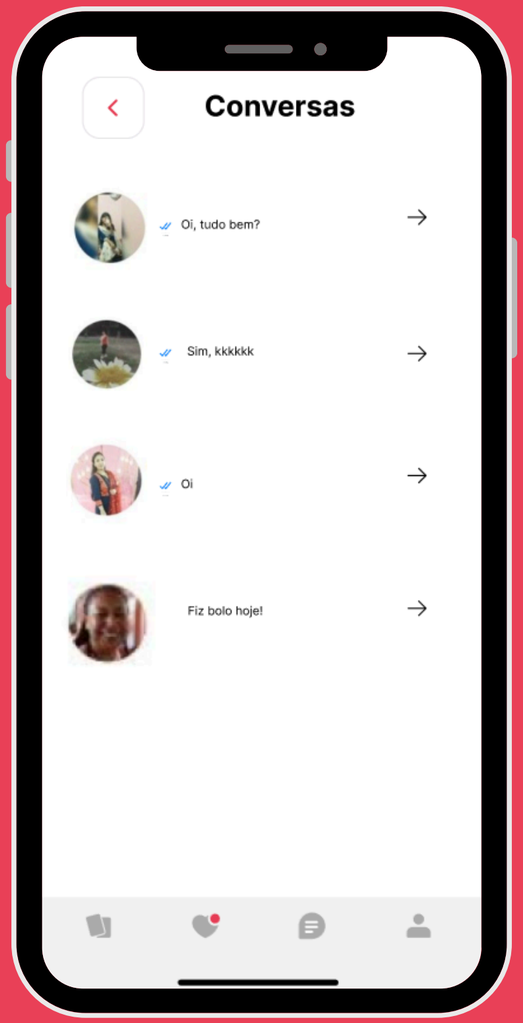
Dating App

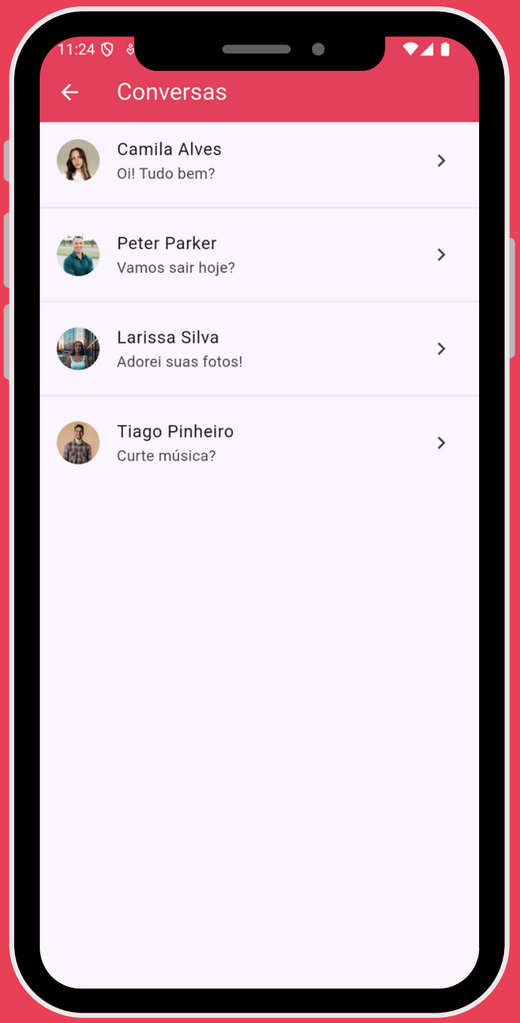
Tela de boas-vindas



Dating App

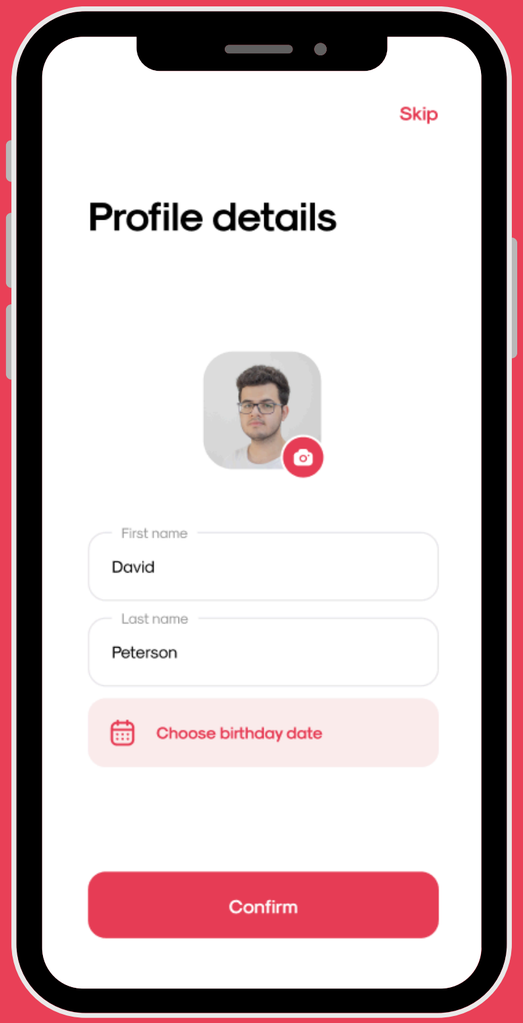
Tela que se adequa ao uso de ListView

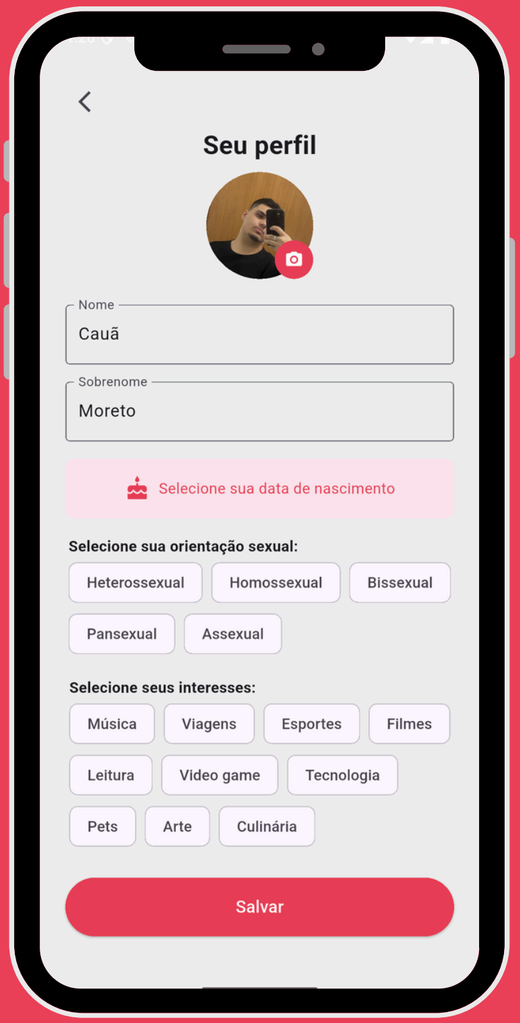




Dating App

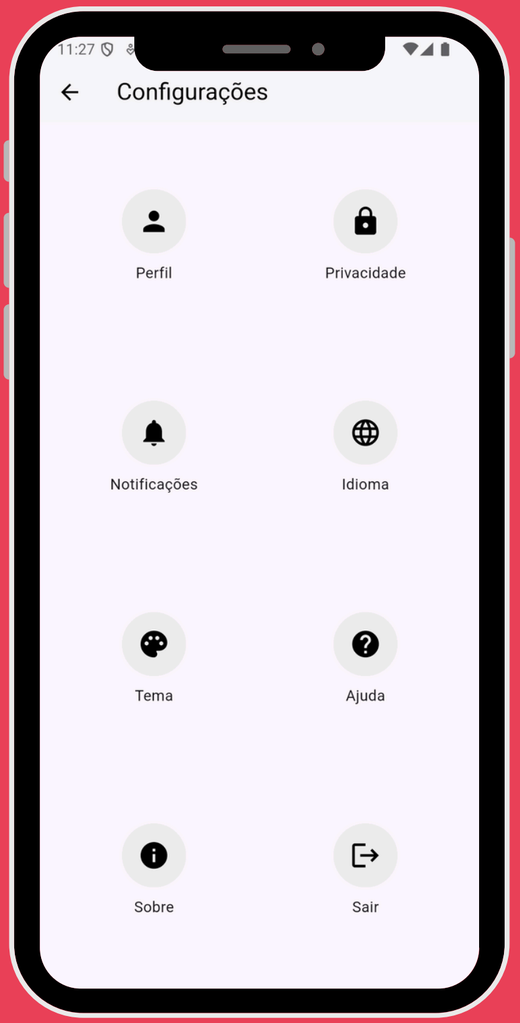
Tela que se adequa ao uso de formulário





Dating App

Tela que se adequa ao uso de GridView

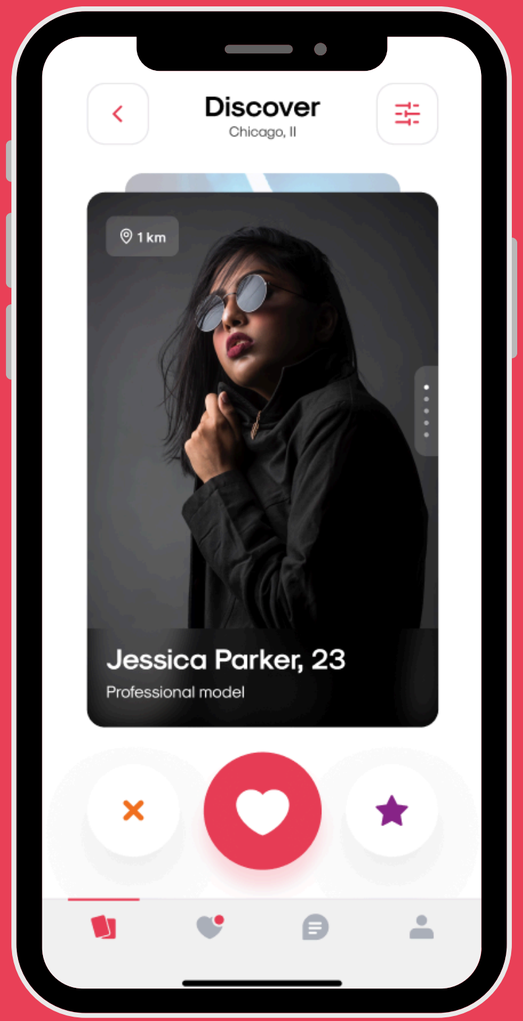


Dating App

Tela que apresenta um item selecionado

Dating App

Uso de Widgets Material (AppBar, Drawer,ButtonNavigation Bar, FloatActionButton) sem funcionalidades

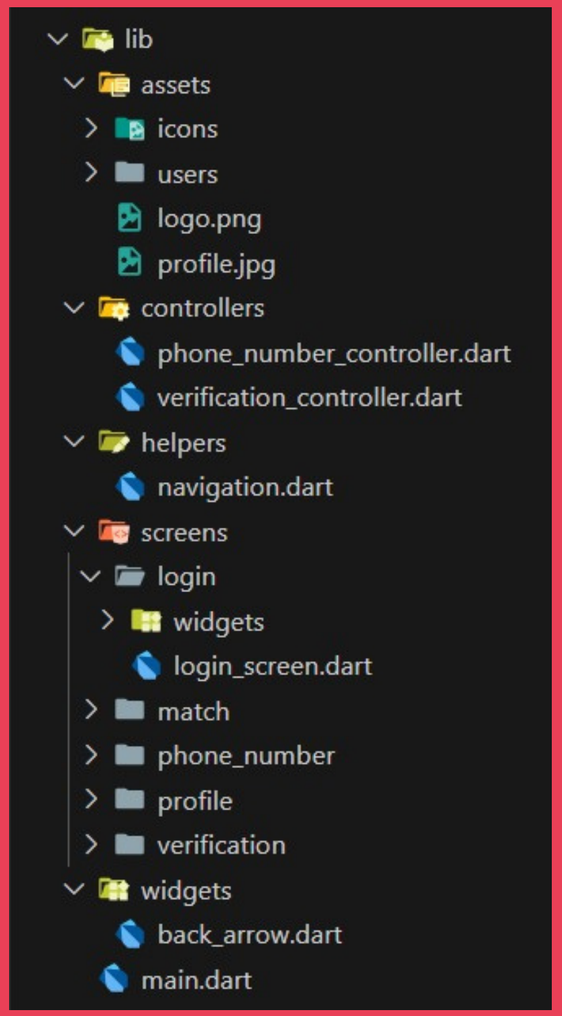


Dating App

Recurso 01

Especifique uma estrutura de pastas (arquitetura) para o projeto do aplicativo, semelhante ao que foi apresentado na Aula 3 - Rotas e Navegação. Busque respeitar a arquitetura definida, alocando os arquivos de código fonte entre eles.

- "Assets" - Todas as imagens utilizadas separadas por contexto;
- "Controllers" - Controllers utilizado na validação do número de telefone e código de verificação;
- "Helpers" - Funções genéricas e reutilizaveis;
- "Screens" - Código fonte das telas e widgets específicos;
- "Widgets" - Widgets genéricos e reutilizaveis.



Dating App

Recurso 02

Utilize Rotas e Navegação, aplicando a técnica com MaterialApp apresentada em Aula 3 - Rotas e Navegação.

navigation.dart | back_arrow.dart

lib > helpers > navigation.dart > ...

```dart
import 'package:flutter/material.dart';

void navigateTo(BuildContext context, Widget screen) {
  Navigator.push(
    context,
    MaterialPageRoute(builder: (_) => screen),
  );
}
```

Criada uma função genérica "navigateTo" onde todas as telas utilizam para chamar a próxima tela utilizando Navigator.push

```dart
class BackArrow extends StatelessWidget implements PreferredSizeWidget {
  const BackArrow({super.key});

  @override
  Widget build(BuildContext context) {
    return AppBar(
      backgroundColor: Colors.white,
      elevation: 0,
      leading: IconButton(
        icon: const Icon(Icons.arrow_back_ios, color: Colors.black87),
        onPressed: () => Navigator.pop(context),
      ), // IconButton
    ); // AppBar
  }

  @override
  Size get preferredSize => const Size.fromHeight(kToolbarHeight);
}
```

Widget "BackArrow" genérico e utilizado em todas as telas para voltar a tela anterior utilizando Navigator.pop

Dating App

Recurso 03

Passe dados entre as páginas utilizando as estratégias descritas na Aula 3 - Rotas e Navegação.

```dart
class PhoneNumberController {
  String fullPhoneNumber = '';
  bool isValid = false;

  void onPhoneChanged(String number, int length) {
    fullPhoneNumber = number;
    isValid = length >= 11;
  }

  void goToVerification(BuildContext context) {
    if (isValid) {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => VerificationScreen(phoneNumber: fullPhoneNumber),
        ), // MaterialPageRoute
      );
    } else {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
          content: Text('O número está incompleto'),
          backgroundColor: Colors.red,
        ), // SnackBar
      );
    }
  }
}
```

PhoneNumberController utiliza uma função "onPhoneChanged" para capturar o número digitado pelo usuário e uma função "goToVerification" para chamar a verification_screen que recebe os valores capturados como parâmetro.

Dating App

Recurso 04

Defina pelo menos um formulário no seu aplicativo, como apresentado na Aula 4 (PDMI6) - Widgets de Input e Dialog, Buttons e Material Widgets.

```
void _submitForm() {
  if (_formKey.currentState?.validate() ?? false) {
    navigateTo(context, const MatchScreen());
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey.shade200,
    body: SafeArea(
      child: LayoutBuilder(
        builder: (context, constraints) {
          final isWide = constraints.maxWidth > 600;

          return SingleChildScrollView(
            padding: const EdgeInsets.all(24.0),
            child: Form(
              key: _formKey,
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.center,
                children: [
                  Align(
                    alignment: Alignment.centerLeft,
                    child: IconButton(
                      icon: const Icon(Icons.arrow_back_ios),
                      onPressed: () => Navigator.pop(context),
                    ), // IconButton
                  ), // Align
                  const Text("Seu perfil", style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
                  const SizedBox(height: 8),
                  const ProfilePhoto(),
                  const SizedBox(height: 24),
```

A proofile_screen utiliza o widget Form para apresentação dos campos.

Dating App

Recurso 05

Utilize a classe LayoutBuilder ou MediaQuery para tornar o aplicativo responsivo à rotação de tela, tendo como base a aula Aula 5 (PDMI6) - Criação de Layouts com Flutter.

```dart
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey.shade200,
    body: SafeArea(
      child: LayoutBuilder(
        builder: (context, constraints) {
          final isWide = constraints.maxWidth > 600;

          return SingleChildScrollView(
            padding: const EdgeInsets.all(24.0),
            child: Form(
              key: _formKey,
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.center,
                children: [
                  Align(
                    alignment: Alignment.centerLeft,
                    child: IconButton(
                      icon: const Icon(Icons.arrow_back_ios),
                      onPressed: () => Navigator.pop(context),
                    ), // IconButton
                  ), // Align
                  const Text("Seu perfil", style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
                  const SizedBox(height: 8),
                  const ProfilePhoto(),
                  const SizedBox(height: 24),
```

profile_screen utiliza o LayoutBuilder para apresentar o formulário de forma responsiva ao rotacionar a tela.

Dating App

Recurso 06

Defina o estilo do app em um ThemeData, como apresentado na Aula 5 (PDMI6) - Criação de Layouts com Flutter.

```dart
class MainApp extends StatelessWidget {
  const MainApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Dating App',
      theme: ThemeData(
        primarySwatch: Colors.pink,
      ), // ThemeData
      home: const LoginScreen(),
```

Classe MainApp utiliza o ThemeData para definir a cor primária do aplicativo

Obrigade!

Grupo 06

Angelo Zovaro, Cauã Barcellos, Gustavo de Oliveira, Luiza Nanni e Renan Breier

BRADEMO

Professor Luiz Gustavo Véras