ECOSSISTEMA ÂNIMA EDUCAÇÃO

Fabiano Figueredo Chaves

RA: 324118586

Renato Comarú Matos

RA: 1292314966

Viviane da Costa Ferreira

RA: 1262429956

Lívia Silva Lima

RA: 824157723

SEGUNDA ENTREGA DO TRABALHO FINAL DA TURMA DE USABILIDADE, WEB, MOBILE E GAMES

Novembro de 2024

ECOSSISTEMA ÂNIMA EDUCAÇÃO

Professores:
Alexandre de Oliveira
Stella Marys Dornelas

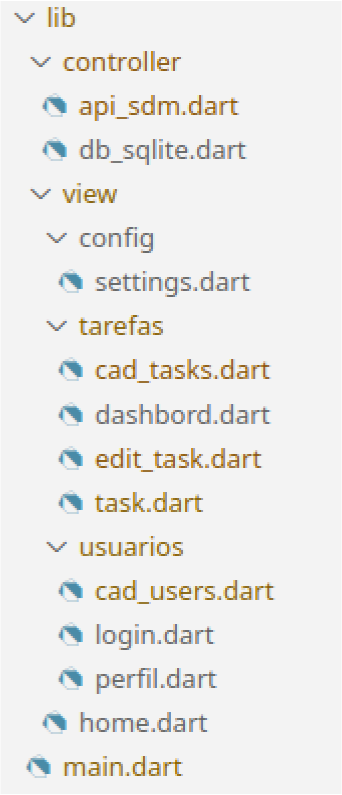
Alunos:
Fabiano Figueredo Chaves
RA: 324118586
Renato Comarú Matos
RA: 1292314966
Viviane da Costa Ferreira
RA: 1262429956
Lívia Silva Lima
RA: 824157723

Trabalho apresentado à disciplina Usabilidade,
Desenvolvimento Web, Mobile e Jogos.

Novembro de 2024

Neste trabalho apresentamos o projeto Taskify, um aplicativo para controle de To-Do List e lembretes, construído com o Flutter.

O diretório lib possui a seguinte organização, conforme a imagem abaixo:



```
lib
  controller
    api_sdm.dart
    db_sqlite.dart
  view
    config
      settings.dart
    tarefas
      cad_tasks.dart
      dashbord.dart
      edit_task.dart
      task.dart
    usuarios
      cad_users.dart
      login.dart
      perfil.dart
    home.dart
  main.dart
```

A seguir, o caminho completo do arquivo no diretório lib, a imagem da tela e a respectiva classe que a construiu, são descritas a seguir:

- lib > main.dart

```
import 'package:a3_udwmj/view/home.dart';
import 'package:device_preview/device_preview.dart';
import 'package:flutter/material.dart';
import 'package:a3_udwmj/controller/db_sqlite.dart';
import 'package:sqflite_common_ffi/sqflite_ffi.dart'
    show databaseFactory, databaseFactoryFfi, sqfliteFfiInit;
import 'package:provider/provider.dart';

void main() async {
 sqfliteFfiInit();
 databaseFactory = databaseFactoryFfi;
 db_sqlite().openMyDatabase();
 runApp(
   ChangeNotifierProvider(
     create: (_) => ThemeProvider(),
     child: MainApp(),
   ),
 );
}

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

 ThemeData temaClaro = ThemeData(
   brightness: Brightness.light,
 );

 ThemeData temaEscuro = ThemeData(
   brightness: Brightness.dark,
 );

 bool temaEscuroAtivo = false;

 @override
 Widget build(BuildContext context) {
   final themeProvider = Provider.of<ThemeProvider>(context);
   return MaterialApp(
     useInheritedMediaQuery: true,
     locale: DevicePreview.locale(context),
     builder: DevicePreview.appBuilder,
     theme: ThemeData.light(),
     darkTheme: ThemeData.dark(),
     themeMode: themeProvider.themeMode,
     home: const Home(),
   );
 }
}

void mostrarErro(BuildContext context, String mensagem) {
 showDialog(
   context: context,
   builder: (BuildContext context) {
     return AlertDialog(
       title: const Text("Erro"),
       content: Text(mensagem),
       actions: [
         TextButton(
           onPressed: () {
             Navigator.of(context).pop();
           },
           child: const Text("OK"),
         ),
       ],
     );
   },
 );
}

void mostrarSucesso(BuildContext context, String mensagem) {
 showDialog(
   context: context,
   builder: (BuildContext context) {
     return AlertDialog(
       title: const Text("Sucesso"),
       content: Text(mensagem),
       actions: [
         TextButton(
           onPressed: () {
             Navigator.of(context).pop();
           },
           child: const Text("OK"),
         ),
       ],
     );
   },
 );
}

class ThemeProvider with ChangeNotifier {
 ThemeMode _themeMode = ThemeMode.light;

 ThemeMode get themeMode => _themeMode;

 void toggleTheme() {
   _themeMode =
       _themeMode == ThemeMode.light ? ThemeMode.dark : ThemeMode.light;
   // notifyListeners();
 }
}
```

- lib > controller > db_sqlite.dart

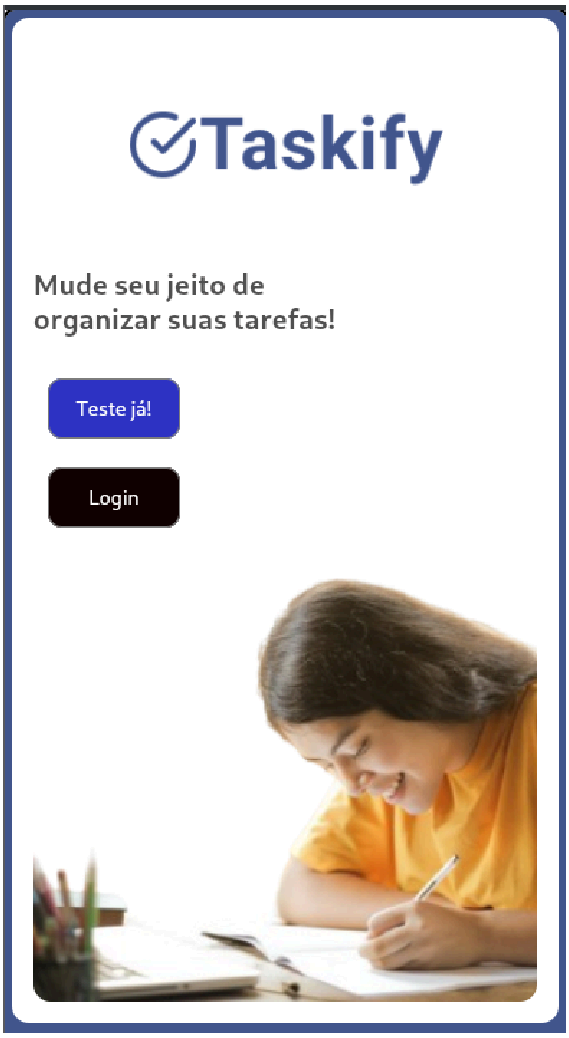
```
/*
CRUD
*/

import 'package:a3_udwmj/controller/api_sdm.dart';
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';
import 'package:sqflite_common_ffi/sqflite_ffi.dart'
   show ConflictAlgorithm, Database, getDatabasesPath, openDatabase;

class db_sqlite {
 Future<Database> openMyDatabase() async {
   return await openDatabase(join(await getDatabasesPath(), 'taskify.db'),
       version: 1, onCreate: (db, version) async {
     return db.execute('''
           PRAGMA foreign_keys = ON;
           CREATE TABLE IF NOT EXISTS usuario (
             id INTEGER PRIMARY KEY AUTOINCREMENT,
             apiId TEXT,
             nome TEXT NOT NULL,
             email TEXT NOT NULL,
             senha TEXT NOT NULL
           );
           CREATE TABLE IF NOT EXISTS tarefa (
             id INTEGER PRIMARY KEY AUTOINCREMENT,
             usuarioId INTEGER,
             apiId TEXT,
             titulo TEXT NOT NULL,
             descricao TEXT,
             data_inicial TEXT,
             data_final TEXT,
             time TEXT,
             categoria TEXT,
             status TEXT,
                  FOREIGN KEY (usuarioId) REFERENCES usuario (id) ON DELETE
CASCADE
```

```
         );
         ''');
  });
}

Future<void> insertUser(String nome, String email, String senha) async {
  final db = await openMyDatabase();

  db.insert(
      'usuario',
      {
        'nome': nome,
        'email': email,
        'senha': senha,
      },
      conflictAlgorithm: ConflictAlgorithm.replace);

  cadUser(email, senha, nome);

}

Future<void> deleteUser(int id) async {
  final db = await openMyDatabase();

  db.delete('usuario', where: 'id = ?', whereArgs: [id]);
}

Future<void> updateUser(int id, String nome, String email) async {
  final db = await openMyDatabase();

  db.update(
      'usuario',
      {
        'nome': nome,
        'email': email,
      },
      where: 'id = ?',
      whereArgs: [id]);
}

Future<String> searchUserByEmail(String email) async {
  final db = await openMyDatabase();
```

```
   List<Map<String, Object?>> usr = (await db.query('usuario',
                  columns: ['id', 'senha'], where: 'email = ?', whereArgs:
[email])); // as Future<List<Map<String, Object?>>>;

   var senhaRet = usr.firstWhere(
     (mapa) => mapa.containsKey('senha'),
     orElse: () => {},
   );

   Object? valor = senhaRet['senha'];

   return valor?.toString() ?? '';
 }

 Future<Map<String, Object?>> searchUserByEmail2(String email) async {
   final db = await openMyDatabase();

   List<Map<String, Object?>> usr = (await db.query('usuario',
                  columns: ['id', 'senha'], where: 'email = ?', whereArgs:
[email])); // as Future<List<Map<String, Object?>>>;

   return usr[0];
 }


 // CRUD para Tarefas

  Future<void> insertTask(int usuarioId, String titulo, String descricao,
String dataInicial, String dataFinal, String categoria, String status) async
{

   final db = await openMyDatabase();

   await db.insert(
        'tarefa',
        {
          'usuarioId': usuarioId,
          'titulo': titulo,
          'descricao': descricao,
          'time': DateTime.now().toString(),
          'data_inicial': dataInicial,
          'data_final': dataFinal,
          'categoria': categoria,
          'status': status,
        },
        conflictAlgorithm: ConflictAlgorithm.replace);
 }

 Future<void> deleteTask(int id) async {
   final db = await openMyDatabase();

   db.delete('tarefa', where: 'id = ?', whereArgs: [id]);
 }

 Future<void> updateTask(int id, int usuarioId, String titulo, String
descricao,
      String dataInicial, String dataFinal, String categoria, String status)
async {
   final db = await openMyDatabase();
   print("sqlite update:");
   print(
   db.update(
        'tarefa',
        {
          'usuarioId': usuarioId,
          'titulo': titulo,
          'descricao': descricao,
          'data_inicial': dataInicial,
          'data_final': dataFinal,
          'categoria': categoria,
          'status': status,
        },
        where: 'id = ?',
        whereArgs: [id]
   )//;
   );

 }

 Future<List<Map<String, dynamic>>> getTasksByIdUser(int usuarioId) async {
   final db = await openMyDatabase();
   return await db.query('tarefa',
        columns: ['id', 'titulo', 'descricao', 'data_inicial', 'data_final',
'categoria', 'status'],
        where: 'usuarioId = ?',
        whereArgs: [usuarioId]);
```

```
 }

 Future<Map<String, Object?>> getTaskByIdTask(int id) async {
   final db = await openMyDatabase();

   List<Map<String, Object?>> tsk = await db.query('tarefa',
         columns: ['id', 'usuarioId', 'titulo', 'descricao', 'data_inicial',
'data_final', 'categoria', 'status'],
       where: 'id = ?',
       whereArgs: [id]);
   return tsk.firstWhere(
     (mapa) => mapa.containsKey('id'),
     orElse: () => {},
   );
 }

}
```

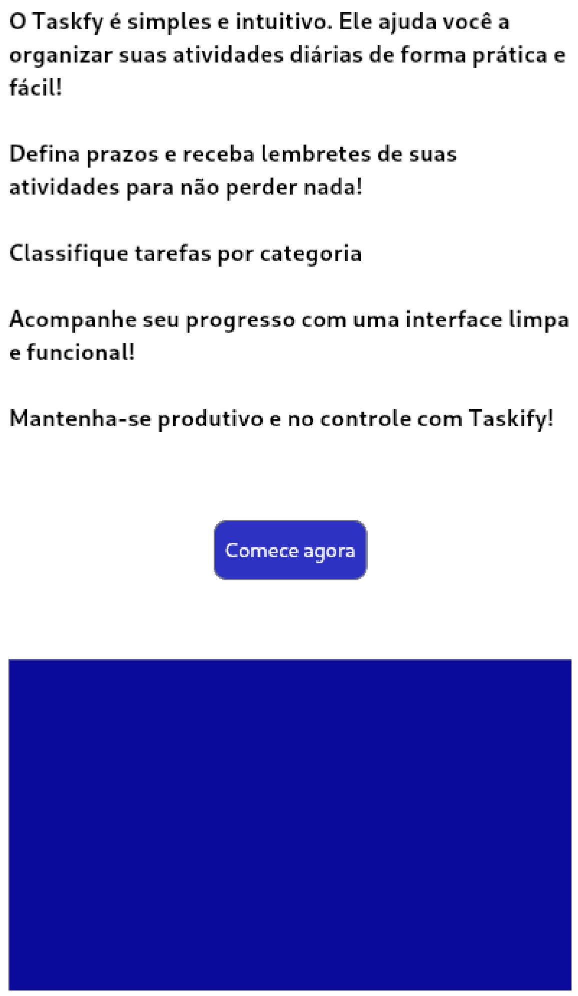
- lib > view > home.dart:



Taskify

Mude seu jeito de organizar suas tarefas!

```
import 'package:a3_udwmj/view/usuarios/cad_users.dart';
import 'package:a3_udwmj/view/usuarios/login.dart';
import 'package:flutter/material.dart';

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

 @override
 Widget build(BuildContext context) {
   return Scaffold(
     body: Container(
       width: double.infinity,
       height: double.infinity,
       padding: const EdgeInsets.only(right: 1, bottom: 20),
       decoration: ShapeDecoration(
  color: const Color(0xFF41558F),
  shape: RoundedRectangleBorder(
    side: const BorderSide(
      width: 2,
      strokeAlign: BorderSide.strokeAlignOutside,
    ),
    borderRadius: BorderRadius.circular(8),
  ),
),
child: Center(
  child: Container(
    width: 380,
    height: 700,
    clipBehavior: Clip.antiAlias,
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius: BorderRadius.circular(12),
    ),
    child: Padding(
      padding: const EdgeInsets.all(15.0),
      child: Column(
        mainAxisSize: MainAxisSize.max,
        mainAxisAlignment: MainAxisAlignment.start,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          const Image(
            image: AssetImage('assets/Taskify.png'),
            height: 150,
            width: 220,
            fit: BoxFit.contain,
          ),
          const SizedBox(height: 8),
          const Padding(
            padding: EdgeInsets.only(bottom: 20),
            child: Align(
              alignment: Alignment.centerLeft,
              child: SizedBox(
                width: double.infinity,
                child: Text(
                  'Mude seu jeito de \norganizar suas tarefas!',
                  style: TextStyle(
                    color: Color(0xFF535252),
                    fontSize: 20,
          fontFamily: 'Poppins',
          fontWeight: FontWeight.w700,
          height: 1.2,
          letterSpacing: 0.10,
        ),
      ),
    ),
  ),
),
// Botões alinhados abaixo do texto
Align(
  alignment: Alignment.centerLeft,
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Padding(
        padding: const EdgeInsets.all(10),
        child: Align(
          alignment: const Alignment(-1.0, 0.0),
          child: MaterialButton(
            onPressed: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context) => const Descr()),
              );
            },
            color: const Color(0xff2f34c5),
            elevation: 0,
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(10.0),
              side: const BorderSide(
                  color: Color(0xff808080), width: 1),
            ),
            padding: const EdgeInsets.all(16),
            textColor: const Color(0xffffffff),
            height: 50,
            minWidth: 100,
            child: const Text(
              "Teste já!",
              style: TextStyle(
                fontSize: 14,
                fontWeight: FontWeight.w400,
                fontStyle: FontStyle.normal,
              ),
            ),
          ),
        ),
      ),
      Padding(
        padding: const EdgeInsets.all(10),
        child: Align(
          alignment: const Alignment(-1.0, 0.0),
          child: MaterialButton(
            onPressed: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context) => const Login()),
              );
            },
            color: const Color(0xff100202),
            elevation: 0,
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(10.0),
              side: const BorderSide(
                  color: Color(0xff808080), width: 1),
            ),
            padding: const EdgeInsets.all(16),
            textColor: const Color(0xffffffff),
            height: 50,
            minWidth: 100,
            child: const Text(
              "Login",
              style: TextStyle(
                fontSize: 14,
                fontWeight: FontWeight.w400,
                fontStyle: FontStyle.normal,
              ),
            ),
          ),
        ),
      ),
    ],
  ),
),
                 // Imagem de fundo ajustada abaixo dos botões
                 Expanded(
                   child: Container(
                     width: double.infinity,
                     decoration: BoxDecoration(
                       image: const DecorationImage(
                         image: AssetImage('assets/Estudante1.png'),
                         fit: BoxFit.cover,
                       ),
                       borderRadius: BorderRadius.circular(12),
                     ),
                   ),
                 ),
               ],
             ),
           ),
         ),
       ),
     ),
   );
 }
}
```



O Taskfy é simples e intuitivo. Ele ajuda você a organizar suas atividades diárias de forma prática e fácil!

Defina prazos e receba lembretes de suas atividades para não perder nada!

Classifique tarefas por categoria

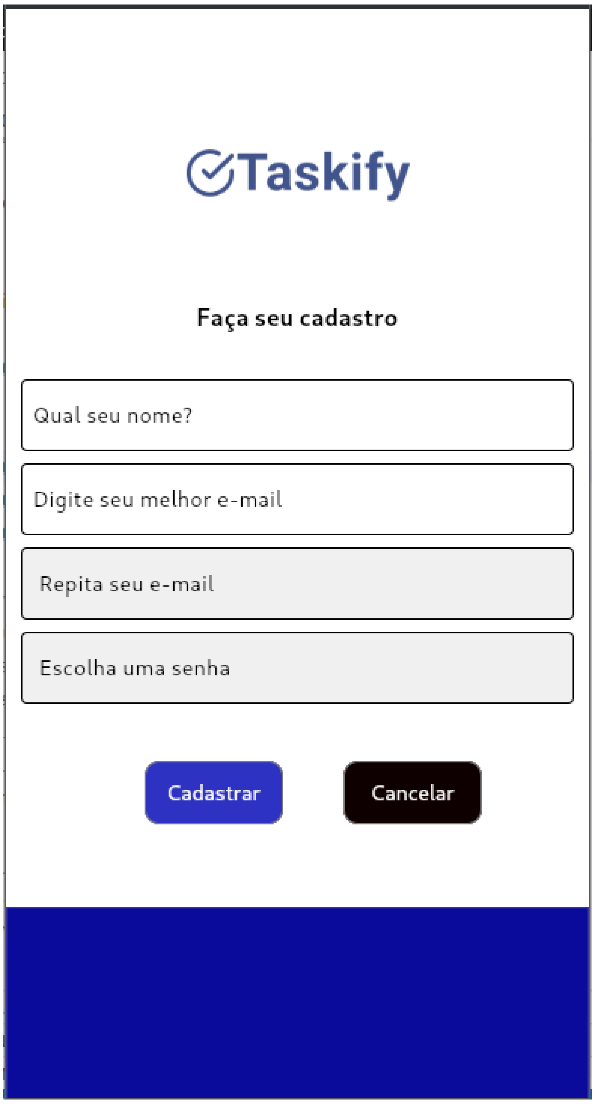
Acompanhe seu progresso com uma interface limpa e funcional!

Mantenha-se produtivo e no controle com Taskify!

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

 @override
 Widget build(BuildContext context) {
   return Scaffold(
     backgroundColor: const Color(0xffffffff),
     body: Align(
       alignment: const Alignment(0.0, 0.2),
       child: Padding(
         padding: const EdgeInsets.fromLTRB(0, 35, 0, 0),
         child: SingleChildScrollView(
           child: Column(
             mainAxisAlignment: MainAxisAlignment.center,
             crossAxisAlignment: CrossAxisAlignment.center,
             mainAxisSize: MainAxisSize.max,
             children: [
               const Padding(
                 padding: EdgeInsets.fromLTRB(0, 8, 0, 30),
                 child: Text(
                   "O Taskfy é simples e intuitivo. Ele ajuda você a 
organizar suas atividades diárias de forma prática e fácil!\n\nDefina prazos 
e receba lembretes de suas atividades para não perder nada!\n\nClassifique 
tarefas por categoria\n\nAcompanhe seu progresso com uma interface limpa e 
funcional!\n\nMantenha-se produtivo e no controle com Taskify!",
                   textAlign: TextAlign.left,
                   overflow: TextOverflow.clip,
                   style: TextStyle(
                     fontWeight: FontWeight.w700,
                     fontStyle: FontStyle.normal,
                     fontSize: 16,
                     color: Color(0xff000000),
                   ),
                 ),
               ),
               Align(
                 alignment: const Alignment(0.0, 0.0),
                 child: SingleChildScrollView(
                   scrollDirection: Axis.horizontal,
                   child: Row(
                     mainAxisAlignment: MainAxisAlignment.start,
                     crossAxisAlignment: CrossAxisAlignment.center,
                     mainAxisSize: MainAxisSize.max,
                     children: [
                       Padding(
                         padding: const EdgeInsets.all(30),
                         child: Align(
                           alignment: const Alignment(0.0, 0.0),
                           child: MaterialButton(
                             onPressed: () {
                               Navigator.push(
                                 context,
                                 MaterialPageRoute(
                                     builder: (context) => const 
Cadastro()),
                               );
                             },
                             color: const Color(0xff2f34c5),
                             elevation: 0,
                             shape: RoundedRectangleBorder(
                               borderRadius: BorderRadius.circular(10.0),
                               side: const BorderSide(
                                   color: Color(0xff808080), width: 1),
                             ),
                             padding: const EdgeInsets.all(16),
                             textColor: const Color(0xffffffff),
                             height: 50,
                             minWidth: 100,
                             child: const Text(
                               "Comece agora",
                               style: TextStyle(
                                 fontSize: 14,
                                 fontWeight: FontWeight.w400,
                                 fontStyle: FontStyle.normal,
                               ),
                             ),
                           ),
                         ),
                       ),
                     ],
                   ),
                 ),
               ),
               Container(
                 margin: const EdgeInsets.fromLTRB(0, 25, 0, 0),
                 padding: const EdgeInsets.all(0),
                 width: MediaQuery.of(context).size.width,
                 height: MediaQuery.of(context).size.width,
                 decoration: BoxDecoration(
                   color: const Color(0xff0e0e9b),
                   shape: BoxShape.rectangle,
                   borderRadius: BorderRadius.zero,
                   border:
                       Border.all(color: const Color(0x4d9e9e9e), width: 1),
                 ),
               )
             ],
           ),
         ),
       ),
     ),
   );
```

```
  }
}
```

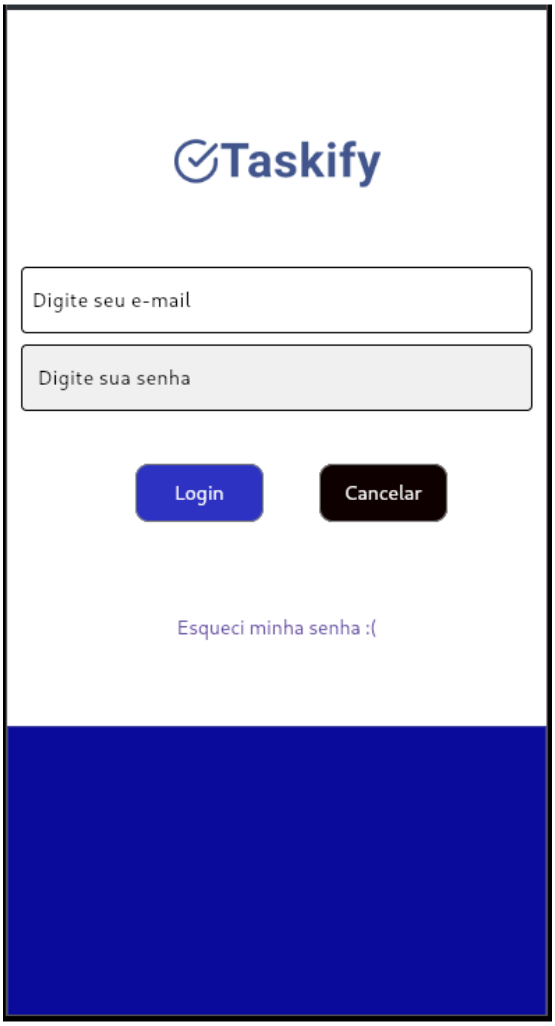
- lib > view > usuarios > cad_users.dart:



```
import 'package:a3_udwmj/view/home.dart';
import 'package:a3_udwmj/view/usuarios/login.dart';
import 'package:a3_udwmj/controller/db_sqlite.dart';

import 'package:flutter/material.dart';

late final email_pub, senha_pub;
```

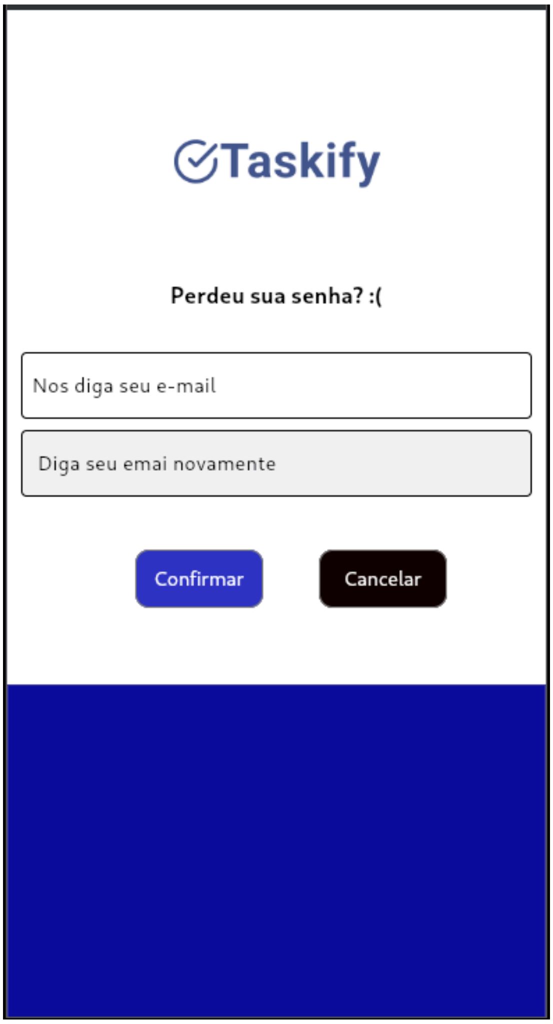
```
class Cadastro extends StatelessWidget {
 const Cadastro({super.key});

 @override
 Widget build(BuildContext context) {
   final TextEditingController nome, email1, email2, senha;

   db_sqlite sqfliteInst = db_sqlite();

   return Scaffold(
     backgroundColor: const Color(0xffffffff),
     body: Align(
       alignment: const Alignment(0.0, 0.2),
       child: Padding(
         padding: const EdgeInsets.fromLTRB(0, 35, 0, 0),
         child: SingleChildScrollView(
           child: Column(
             mainAxisAlignment: MainAxisAlignment.center,
             crossAxisAlignment: CrossAxisAlignment.center,
             mainAxisSize: MainAxisSize.max,
             children: [
               const Image(
                 image: AssetImage('assets/Taskify.png'),
                 height: 150,
                 width: 150,
                 fit: BoxFit.contain,
               ),
               const Padding(
                 padding: EdgeInsets.fromLTRB(0, 8, 0, 30),
                 child: Text(
                   "Faça seu cadastro",
                   textAlign: TextAlign.center,
                   overflow: TextOverflow.clip,
                   style: TextStyle(
                     fontWeight: FontWeight.w700,
                     fontStyle: FontStyle.normal,
                     fontSize: 16,
                     color: Color(0xff000000),
                   ),
                 ),
               ),
               Padding(
                 padding: const EdgeInsets.fromLTRB(10, 0, 10, 8),
                 child: TextField(
                   controller: nome = TextEditingController(),
                   obscureText: false,
                   textAlign: TextAlign.start,
                   maxLines: 1,
                   style: const TextStyle(
                     fontWeight: FontWeight.w400,
                     fontStyle: FontStyle.normal,
                     fontSize: 12,
                     color: Color(0xff000000),
                   ),
                   decoration: InputDecoration(
                     disabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     focusedBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     enabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     labelText: "Qual seu nome?",
                     labelStyle: const TextStyle(
                       fontWeight: FontWeight.w400,
                       fontStyle: FontStyle.normal,
                       fontSize: 14,
                       color: Color(0xff000000),
                     ),
                     filled: true,
                     fillColor: const Color(0xffffffff),
                     isDense: false,
                     contentPadding: const EdgeInsets.all(8),
                   ),
                 ),
               ),
               Padding(
                 padding: const EdgeInsets.fromLTRB(10, 0, 10, 8),
                 child: TextField(
                   controller: email1 = TextEditingController(),
                   obscureText: false,
                   textAlign: TextAlign.start,
                   maxLines: 1,
                   style: const TextStyle(
                     fontWeight: FontWeight.w400,
                     fontStyle: FontStyle.normal,
                     fontSize: 12,
                     color: Color(0xff000000),
                   ),
                   decoration: InputDecoration(
                     disabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     focusedBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     enabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     labelText: "Digite seu melhor e-mail",
                     labelStyle: const TextStyle(
                       fontWeight: FontWeight.w400,
                       fontStyle: FontStyle.normal,
                       fontSize: 14,
                       color: Color(0xff000000),
                     ),
                     filled: true,
                     fillColor: const Color(0xffffffff),
                     isDense: false,
                     contentPadding: const EdgeInsets.all(8),
                   ),
                 ),
               ),
               Padding(
                 padding: const EdgeInsets.fromLTRB(10, 0, 10, 8),
                 child: TextField(
                   controller: email2 = TextEditingController(),
                   obscureText: false,
                   textAlign: TextAlign.start,
                   maxLines: 1,
                   style: const TextStyle(
                     fontWeight: FontWeight.w400,
                     fontStyle: FontStyle.normal,
                     fontSize: 14,
                     color: Color(0xff000000),
                   ),
                   decoration: InputDecoration(
                     disabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     focusedBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     enabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     labelText: "Repita seu e-mail",
                     labelStyle: const TextStyle(
                       fontWeight: FontWeight.w400,
                       fontStyle: FontStyle.normal,
                       fontSize: 14,
                       color: Color(0xff000000),
                     ),
                     filled: true,
                     fillColor: const Color(0xfff2f2f3),
                     isDense: false,
                     contentPadding: const EdgeInsets.symmetric(
                         vertical: 8, horizontal: 12),
                   ),
                 ),
               ),
               Padding(
                 padding: const EdgeInsets.fromLTRB(10, 0, 10, 8),
                 child: TextField(
                   controller: senha = TextEditingController(),
                   obscureText: true,
                   textAlign: TextAlign.start,
                   maxLines: 1,
                   style: const TextStyle(
                     fontWeight: FontWeight.w400,
                     fontStyle: FontStyle.normal,
                     fontSize: 14,
                     color: Color(0xff000000),
                   ),
                   decoration: InputDecoration(
                     disabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     focusedBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     enabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     labelText: "Escolha uma senha",
                     labelStyle: const TextStyle(
                       fontWeight: FontWeight.w400,
                       fontStyle: FontStyle.normal,
                       fontSize: 14,
                       color: Color(0xff000000),
                     ),
                     filled: true,
                     fillColor: const Color(0xfff2f2f3),
                     isDense: false,
                     contentPadding: const EdgeInsets.symmetric(
                         vertical: 8, horizontal: 12),
                   ),
                 ),
               ),
               Align(
                 alignment: const Alignment(0.0, 0.0),
                 child: SingleChildScrollView(
                   scrollDirection: Axis.horizontal,
                   child: Row(
                     mainAxisAlignment: MainAxisAlignment.start,
                     crossAxisAlignment: CrossAxisAlignment.center,
                     mainAxisSize: MainAxisSize.max,
                     children: [
                       Padding(
                         padding: const EdgeInsets.all(30),
                         child: Align(
                           alignment: const Alignment(0.0, 0.0),
                           child: MaterialButton(
                             onPressed: () {
                               if (email1.text == email2.text) {
                                 email_pub = email1.text;
                                 // senha_pub = senha.text;
                                 // print("Cadastro: email_pub: ");
                                 // print(email_pub);
                                 sqfliteInst.insertUser(
                                     nome.text, email1.text, senha.text);

                                         // cadUser(email1.text, senha.text, nome.text);

                                 Navigator.push(
                                   context,
                                   MaterialPageRoute(
                                       builder: (context) => const Login()),
                                 );
                               }
                             },
                             color: const Color(0xff2f34c5),
                             elevation: 0,
                             shape: RoundedRectangleBorder(
                               borderRadius: BorderRadius.circular(10.0),
                               side: const BorderSide(
                                   color: Color(0xff808080), width: 1),
                             ),
                             padding: const EdgeInsets.all(16),
                             textColor: const Color(0xffffffff),
                             height: 50,
                     minWidth: 100,
                     child: const Text(
                       "Cadastrar",
                       style: TextStyle(
                         fontSize: 14,
                         fontWeight: FontWeight.w400,
                         fontStyle: FontStyle.normal,
                       ),
                     ),
                   ),
                 ),
               ),
               Padding(
                 padding: const EdgeInsets.all(10),
                 child: Align(
                   alignment: const Alignment(0.4, 0.0),
                   child: MaterialButton(
                     onPressed: () {
                       Navigator.push(
                         context,
                         MaterialPageRoute(
                             builder: (context) => const Home()),
                       );
                     },
                     color: const Color(0xff100202),
                     elevation: 0,
                     shape: RoundedRectangleBorder(
                       borderRadius: BorderRadius.circular(10.0),
                       side: const BorderSide(
                           color: Color(0xff808080), width: 1),
                     ),
                     padding: const EdgeInsets.all(16),
                     textColor: const Color(0xffffffff),
                     height: 50,
                     minWidth: 100,
                     child: const Text(
                       "Cancelar",
                       style: TextStyle(
                         fontSize: 14,
                         fontWeight: FontWeight.w400,
                         fontStyle: FontStyle.normal,
                       ),
                     ),
                           ),
                         ),
                       ),
                     ],
                   ),
                 ),
               ),
               Container(
                 margin: const EdgeInsets.fromLTRB(0, 25, 0, 0),
                 padding: const EdgeInsets.all(0),
                 width: MediaQuery.of(context).size.width,
                 height: MediaQuery.of(context).size.width,
                 decoration: BoxDecoration(
                   color: const Color(0xff0e0e9b),
                   shape: BoxShape.rectangle,
                   borderRadius: BorderRadius.zero,
                   border:
                       Border.all(color: const Color(0x4d9e9e9e), width: 1),
                 ),
               )
             ],
           ),
         ),
       ),
     ),
   );
 }
}

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

 @override
 Widget build(BuildContext context) {
   final TextEditingController credencial;
   String email;
   // int teste;

   return Scaffold(
     backgroundColor: const Color(0xffffffff),
     body: Align(
       alignment: const Alignment(0.0, 0.2),
       child: Padding(
         padding: const EdgeInsets.fromLTRB(0, 35, 0, 0),
         child: SingleChildScrollView(
           child: Column(
             mainAxisAlignment: MainAxisAlignment.center,
             crossAxisAlignment: CrossAxisAlignment.center,
             mainAxisSize: MainAxisSize.max,
             children: [
               ///***If you have exported images you must have to copy those
images in assets/images directory.
               const Image(
                 image: AssetImage('assets/Taskify.png'),
                 height: 150,
                 width: 150,
                 fit: BoxFit.contain,
               ),
               const Padding(
                 padding: EdgeInsets.fromLTRB(0, 8, 0, 30),
                 child: Text(
                   "Seu cadastro está completo!",
                   textAlign: TextAlign.center,
                   overflow: TextOverflow.clip,
                   style: TextStyle(
                     fontWeight: FontWeight.w700,
                     fontStyle: FontStyle.normal,
                     fontSize: 16,
                     color: Color(0xff000000),
                   ),
                 ),
               ),
               const Padding(
                 padding: EdgeInsets.fromLTRB(0, 8, 0, 30),
                 child: Text(
                     "Falta pouco!\nApenas verifique seu e-mail e digite o
número que você recebeu!",
                   textAlign: TextAlign.center,
                   overflow: TextOverflow.clip,
                   style: TextStyle(
                     fontWeight: FontWeight.w700,
                     fontStyle: FontStyle.normal,
                     fontSize: 14,
                     color: Color(0xff000000),
                   ),
                 ),
               ),

               Padding(
                 padding: const EdgeInsets.fromLTRB(10, 0, 10, 8),
                 child: TextField(
                   controller: credencial = TextEditingController(),
                   obscureText: false,
                   textAlign: TextAlign.start,
                   maxLines: 1,
                   style: const TextStyle(
                     fontWeight: FontWeight.w400,
                     fontStyle: FontStyle.normal,
                     fontSize: 12,
                     color: Color(0xff000000),
                   ),
                   decoration: InputDecoration(
                     disabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     focusedBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     enabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     labelText: "Digite o numero recebido",
                     labelStyle: const TextStyle(
                       fontWeight: FontWeight.w400,
                       fontStyle: FontStyle.normal,
                       fontSize: 14,
                       color: Color(0xff000000),
                     ),
                     filled: true,
                     fillColor: const Color(0xffffffff),
                     isDense: false,
                     contentPadding: const EdgeInsets.all(8),
                   ),
                 ),
               ),
               Align(
                 alignment: const Alignment(0.0, 0.0),
                 child: SingleChildScrollView(
                   scrollDirection: Axis.horizontal,
                   child: Row(
                     mainAxisAlignment: MainAxisAlignment.start,
                     crossAxisAlignment: CrossAxisAlignment.center,
                     mainAxisSize: MainAxisSize.max,
                     children: [
                       Padding(
                         padding: const EdgeInsets.all(30),
                         child: Align(
                           alignment: const Alignment(0.0, 0.0),
                           child: MaterialButton(
                             onPressed: () async {
                               email = email_pub;
                               Navigator.push(
                                 context,
                                 MaterialPageRoute(
                                     builder: (context) => const Login()),
                               );
                               // }
                             },
                             color: const Color(0xff2f34c5),
                             elevation: 0,
                             shape: RoundedRectangleBorder(
                               borderRadius: BorderRadius.circular(10.0),
                               side: const BorderSide(
                                   color: Color(0xff808080), width: 1),
                             ),
                             padding: const EdgeInsets.all(16),
                             textColor: const Color(0xffffffff),
                             height: 50,
                             minWidth: 100,
                             child: const Text(
                               "Confirmar",
                               style: TextStyle(
                                 fontSize: 14,
                                 fontWeight: FontWeight.w400,
                                 fontStyle: FontStyle.normal,
                               ),
                                             ),
                                           ),
                                         ),
                                       ),
                                     ],
                                   ),
                                 ),
                               ),
                               Container(
                                 margin: const EdgeInsets.fromLTRB(0, 25, 0, 0),
                                 padding: const EdgeInsets.all(0),
                                 width: MediaQuery.of(context).size.width,
                                 height: MediaQuery.of(context).size.width,
                                 decoration: BoxDecoration(
                                   color: const Color(0xff0e0e9b),
                                   shape: BoxShape.rectangle,
                                   borderRadius: BorderRadius.zero,
                                   border:
                                       Border.all(color: const Color(0x4d9e9e9e), width: 1),
                                 ),
                               )
                             ],
                           ),
                         ),
                       ),
                     ),
                   );
                 }
               }
```

- lib > view > usuarios > login.dart:



```
import 'package:a3_udwmj/controller/api_sdm.dart';
import 'package:a3_udwmj/view/home.dart';
import 'package:a3_udwmj/view/tarefas/dashbord.dart';
import 'package:a3_udwmj/controller/db_sqlite.dart';

import 'package:flutter/material.dart';

class user_Pub {
 static int userOn = 1;
}

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

 Object? getItemFromMap(Map<String, Object?> map, String key) {
  return map[key];
}

@override
Widget build(BuildContext context) {
  final TextEditingController email, senha;
  //var psw;
  db_sqlite sqfliteInst = db_sqlite();

  return Scaffold(
    backgroundColor: const Color(0xffffffff),
    body: Align(
      alignment: const Alignment(0.0, 0.2),
      child: Padding(
        padding: const EdgeInsets.fromLTRB(0, 35, 0, 0),
        child: SingleChildScrollView(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.center,
            mainAxisSize: MainAxisSize.max,
            children: [
              const Image(
                image: AssetImage('assets/Taskify.png'),
                height: 150,
                width: 150,
                fit: BoxFit.contain,
              ),
              Padding(
                padding: const EdgeInsets.fromLTRB(10, 0, 10, 8),
                child: TextField(
                  controller: email = TextEditingController(),
                  obscureText: false,
                  textAlign: TextAlign.start,
                  maxLines: 1,
                  style: const TextStyle(
                    fontWeight: FontWeight.w400,
                    fontStyle: FontStyle.normal,
                    fontSize: 12,
                    color: Color(0xff000000),
                  ),
                  decoration: InputDecoration(
                    disabledBorder: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(4.0),
        borderSide: const BorderSide(
            color: Color(0xff000000), width: 1),
      ),
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(4.0),
        borderSide: const BorderSide(
            color: Color(0xff000000), width: 1),
      ),
      enabledBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(4.0),
        borderSide: const BorderSide(
            color: Color(0xff000000), width: 1),
      ),
      labelText: "Digite seu e-mail",
      labelStyle: const TextStyle(
        fontWeight: FontWeight.w400,
        fontStyle: FontStyle.normal,
        fontSize: 14,
        color: Color(0xff000000),
      ),
      filled: true,
      fillColor: const Color(0xffffffff),
      isDense: false,
      contentPadding: const EdgeInsets.all(8),
    ),
  ),
),
Padding(
  padding: const EdgeInsets.fromLTRB(10, 0, 10, 8),
  child: TextField(
    controller: senha = TextEditingController(),
    obscureText: true,
    textAlign: TextAlign.start,
    maxLines: 1,
    style: const TextStyle(
      fontWeight: FontWeight.w400,
      fontStyle: FontStyle.normal,
      fontSize: 14,
      color: Color(0xff000000),
    ),
    decoration: InputDecoration(
      disabledBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(4.0),
        borderSide: const BorderSide(
            color: Color(0xff000000), width: 1),
      ),
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(4.0),
        borderSide: const BorderSide(
            color: Color(0xff000000), width: 1),
      ),
      enabledBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(4.0),
        borderSide: const BorderSide(
            color: Color(0xff000000), width: 1),
      ),
      labelText: "Digite sua senha",
      labelStyle: const TextStyle(
        fontWeight: FontWeight.w400,
        fontStyle: FontStyle.normal,
        fontSize: 14,
        color: Color(0xff000000),
      ),
      filled: true,
      fillColor: const Color(0xfff2f2f3),
      isDense: false,
      contentPadding: const EdgeInsets.symmetric(
          vertical: 8, horizontal: 12),
    ),
  ),
),
Align(
  alignment: const Alignment(0.0, 0.0),
  child: SingleChildScrollView(
    scrollDirection: Axis.horizontal,
    child: Row(
      mainAxisAlignment: MainAxisAlignment.start,
      crossAxisAlignment: CrossAxisAlignment.center,
      mainAxisSize: MainAxisSize.max,
      children: [
        Padding(
          padding: const EdgeInsets.all(30),
          child: Align(
            alignment: const Alignment(0.0, 0.0),
            child: MaterialButton(
              onPressed: () async {
```

```
        var psw = await sqfliteInst
            .searchUserByEmail2(email.text);

        if (email.text != "" && senha.text != "") {
          if (getItemFromMap(psw, 'senha') ==
              (senha.text)) {
            user_Pub.userOn = int.parse(
                                  getItemFromMap(psw,
'id').toString());

            Navigator.push(
              context,
              MaterialPageRoute(
                  builder: (context) =>
                      const Dashboard()),
            );
          }
        }
      },
      color: const Color(0xff2f34c5),
      elevation: 0,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10.0),
        side: const BorderSide(
            color: Color(0xff808080), width: 1),
      ),
      padding: const EdgeInsets.all(16),
      textColor: const Color(0xffffffff),
      height: 50,
      minWidth: 100,
      child: const Text(
        "Login",
        style: TextStyle(
          fontSize: 14,
          fontWeight: FontWeight.w400,
          fontStyle: FontStyle.normal,
        ),
      ),
    ),
  ),
),
Padding(
  padding: const EdgeInsets.all(10),
          child: Align(
            alignment: const Alignment(0.4, 0.0),
            child: MaterialButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                      builder: (context) => const Home()),
                );
              },
              color: const Color(0xff100202),
              elevation: 0,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10.0),
                side: const BorderSide(
                    color: Color(0xff808080), width: 1),
              ),
              padding: const EdgeInsets.all(16),
              textColor: const Color(0xffffffff),
              height: 50,
              minWidth: 100,
              child: const Text(
                "Cancelar",
                style: TextStyle(
                  fontSize: 14,
                  fontWeight: FontWeight.w400,
                  fontStyle: FontStyle.normal,
                ),
              ),
            ),
          ),
        ),
      ],
    ),
  ),
),
Padding(
  padding: const EdgeInsets.all(30),
  child: Align(
    alignment: const Alignment(0.0, 0.0),
    child: TextButton(
      onPressed: () {
        Navigator.push(
                          context,
                          MaterialPageRoute(
                              builder: (context) => const PerdeuSenha()),
                        );
                      },
                      child: const Text(
                        "Esqueci minha senha :(",
                        style: TextStyle(
                          fontSize: 14,
                          fontWeight: FontWeight.w400,
                          fontStyle: FontStyle.normal,
                        ),
                      ),
                    ),
                  ),
                ),
                Container(
                  margin: const EdgeInsets.fromLTRB(0, 25, 0, 0),
                  padding: const EdgeInsets.all(0),
                  width: MediaQuery.of(context).size.width,
                  height: MediaQuery.of(context).size.width,
                  decoration: BoxDecoration(
                    color: const Color(0xff0e0e9b),
                    shape: BoxShape.rectangle,
                    borderRadius: BorderRadius.zero,
                    border:
                        Border.all(color: const Color(0x4d9e9e9e), width: 1),
                  ),
                )
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```



Perdeu sua senha? :(

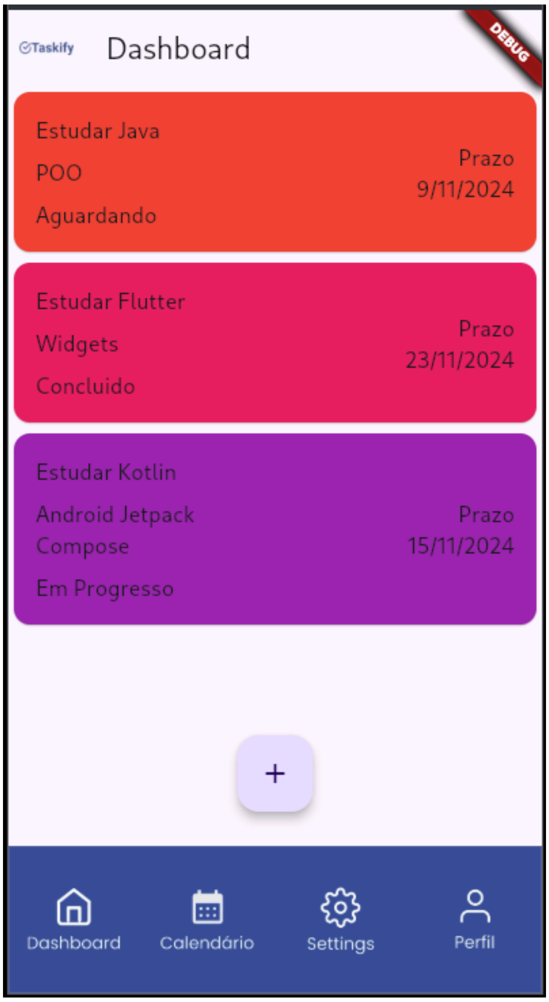
Nos diga seu e-mail

Diga seu emai novamente

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

 @override
 Widget build(BuildContext context) {
   final TextEditingController email1, email2;

   return Scaffold(
     backgroundColor: const Color(0xffffffff),
     body: Align(
       alignment: const Alignment(0.0, 0.2),
       child: Padding(
         padding: const EdgeInsets.fromLTRB(0, 35, 0, 0),
         child: SingleChildScrollView(
           child: Column(
             mainAxisAlignment: MainAxisAlignment.center,
             crossAxisAlignment: CrossAxisAlignment.center,
             mainAxisSize: MainAxisSize.max,
             children: [
               ///***If you have exported images you must have to copy those images in assets/images directory.
               const Image(
                 image: AssetImage('assets/Taskify.png'),
                 height: 150,
                 width: 150,
                 fit: BoxFit.contain,
               ),
               const Padding(
                 padding: EdgeInsets.fromLTRB(0, 8, 0, 30),
                 child: Text(
                   "Perdeu sua senha? :(",
                   textAlign: TextAlign.center,
                   overflow: TextOverflow.clip,
                   style: TextStyle(
                     fontWeight: FontWeight.w700,
                     fontStyle: FontStyle.normal,
                     fontSize: 16,
                     color: Color(0xff000000),
                   ),
                 ),
               ),

               Padding(
                 padding: const EdgeInsets.fromLTRB(10, 0, 10, 8),
                 child: TextField(
                   controller: email1 = TextEditingController(),
                   obscureText: false,
                   textAlign: TextAlign.start,
                   maxLines: 1,
                   style: const TextStyle(
                     fontWeight: FontWeight.w400,
                     fontStyle: FontStyle.normal,
                     fontSize: 12,
                     color: Color(0xff000000),
                   ),
                   decoration: InputDecoration(
                     disabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     focusedBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     enabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     labelText: "Nos diga seu e-mail",
                     labelStyle: const TextStyle(
                       fontWeight: FontWeight.w400,
                       fontStyle: FontStyle.normal,
                       fontSize: 14,
                       color: Color(0xff000000),
                     ),
                     filled: true,
                     fillColor: const Color(0xffffffff),
                     isDense: false,
                     contentPadding: const EdgeInsets.all(8),
                   ),
                 ),
               ),
               Padding(
                 padding: const EdgeInsets.fromLTRB(10, 0, 10, 8),
                 child: TextField(
                   controller: email2 = TextEditingController(),
                   obscureText: false,
                   textAlign: TextAlign.start,
                   maxLines: 1,
                   style: const TextStyle(
                     fontWeight: FontWeight.w400,
                     fontStyle: FontStyle.normal,
                     fontSize: 14,
                     color: Color(0xff000000),
                   ),
                   decoration: InputDecoration(
                     disabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     focusedBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     enabledBorder: OutlineInputBorder(
                       borderRadius: BorderRadius.circular(4.0),
                       borderSide: const BorderSide(
                           color: Color(0xff000000), width: 1),
                     ),
                     labelText: "Diga seu emai novamente",
                     labelStyle: const TextStyle(
                       fontWeight: FontWeight.w400,
                       fontStyle: FontStyle.normal,
                       fontSize: 14,
                       color: Color(0xff000000),
                     ),
                     filled: true,
                     fillColor: const Color(0xfff2f2f3),
                     isDense: false,
                     contentPadding: const EdgeInsets.symmetric(
                         vertical: 8, horizontal: 12),
                   ),
                 ),
               ),
               Align(
                 alignment: const Alignment(0.0, 0.0),
                 child: SingleChildScrollView(
                   scrollDirection: Axis.horizontal,
                   child: Row(
                     mainAxisAlignment: MainAxisAlignment.start,
                     crossAxisAlignment: CrossAxisAlignment.center,
                     mainAxisSize: MainAxisSize.max,
                     children: [
                       Padding(
                         padding: const EdgeInsets.all(30),
                         child: Align(
                           alignment: const Alignment(0.0, 0.0),
                           child: MaterialButton(
                             onPressed: () async {
                               if (email1.text == email2.text) {
```

```
                                  if (await forgotPswrd(email1.text) == 200)
{
                                   Navigator.push(
                                     context,
                                     MaterialPageRoute(
                                         builder: (context) =>
                                             const NovaSenha()),
                                   );
                                 }
                               }
                             },
                             color: const Color(0xff2f34c5),
                             elevation: 0,
                             shape: RoundedRectangleBorder(
                               borderRadius: BorderRadius.circular(10.0),
                               side: const BorderSide(
                                   color: Color(0xff808080), width: 1),
                             ),
                             padding: const EdgeInsets.all(16),
                             textColor: const Color(0xffffffff),
                             height: 50,
                             minWidth: 100,
                             child: const Text(
                               "Confirmar",
                               style: TextStyle(
                                 fontSize: 14,
                                 fontWeight: FontWeight.w400,
                                 fontStyle: FontStyle.normal,
                               ),
                             ),
                           ),
                         ),
                       ),
                       Padding(
                         padding: const EdgeInsets.all(10),
                         child: Align(
                           alignment: const Alignment(0.4, 0.0),
                           child: MaterialButton(
                             onPressed: () {
                               Navigator.push(
                                 context,
                                 MaterialPageRoute(
                                     builder: (context) => const Login()),
                    );
                  },
                  color: const Color(0xff100202),
                  elevation: 0,
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(10.0),
                    side: const BorderSide(
                        color: Color(0xff808080), width: 1),
                  ),
                  padding: const EdgeInsets.all(16),
                  textColor: const Color(0xffffffff),
                  height: 50,
                  minWidth: 100,
                  child: const Text(
                    "Cancelar",
                    style: TextStyle(
                      fontSize: 14,
                      fontWeight: FontWeight.w400,
                      fontStyle: FontStyle.normal,
                    ),
                  ),
                ),
              ),
            ),
          ],
        ),
      ),
    ),
    Container(
      margin: const EdgeInsets.fromLTRB(0, 25, 0, 0),
      padding: const EdgeInsets.all(0),
      width: MediaQuery.of(context).size.width,
      height: MediaQuery.of(context).size.width,
      decoration: BoxDecoration(
        color: const Color(0xff0e0e9b),
        shape: BoxShape.rectangle,
        borderRadius: BorderRadius.zero,
        border:
            Border.all(color: const Color(0x4d9e9e9e), width: 1),
      ),
    )
  ],
),
         ),
       ),
     ),
   );
 }
}

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

 @override
 Widget build(BuildContext context) {
   return Scaffold(
     backgroundColor: const Color(0xffffffff),
     body: Align(
       alignment: const Alignment(0.0, 0.2),
       child: Padding(
         padding: const EdgeInsets.fromLTRB(0, 35, 0, 0),
         child: SingleChildScrollView(
           child: Column(
             mainAxisAlignment: MainAxisAlignment.center,
             crossAxisAlignment: CrossAxisAlignment.center,
             mainAxisSize: MainAxisSize.max,
             children: [
               ///***If you have exported images you must have to copy those images in assets/images directory.
               const Image(
                 image: AssetImage('assets/Taskify.png'),
                 height: 150,
                 width: 150,
                 fit: BoxFit.contain,
               ),
               const Padding(
                 padding: EdgeInsets.fromLTRB(0, 8, 0, 30),
                 child: Text(
                   "Não se preocupe nós vamos te ajudar!\n\n\nUma nova senha foi enviada para o seu e-mail!",
                   textAlign: TextAlign.center,
                   overflow: TextOverflow.clip,
                   style: TextStyle(
                     fontWeight: FontWeight.w700,
                     fontStyle: FontStyle.normal,
                     fontSize: 16,
      color: Color(0xff000000),
    ),
  ),
),

Align(
  alignment: const Alignment(0.0, 0.0),
  child: SingleChildScrollView(
    scrollDirection: Axis.horizontal,
    child: Row(
      mainAxisAlignment: MainAxisAlignment.start,
      crossAxisAlignment: CrossAxisAlignment.center,
      mainAxisSize: MainAxisSize.max,
      children: [
        Padding(
          padding: const EdgeInsets.all(30),
          child: Align(
            alignment: const Alignment(0.0, 0.0),
            child: MaterialButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                      builder: (context) => const Login()),
                );
              },
              color: const Color(0xff2f34c5),
              elevation: 0,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10.0),
                side: const BorderSide(
                    color: Color(0xff808080), width: 1),
              ),
              padding: const EdgeInsets.all(16),
              textColor: const Color(0xffffffff),
              height: 50,
              minWidth: 100,
              child: const Text(
                "Retornar",
                style: TextStyle(
                  fontSize: 14,
                  fontWeight: FontWeight.w400,
                  fontStyle: FontStyle.normal,
                               ),
                             ),
                           ),
                         ),
                       ),
                     ],
                   ),
                 ),
               ),
               Container(
                 margin: const EdgeInsets.fromLTRB(0, 25, 0, 0),
                 padding: const EdgeInsets.all(0),
                 width: MediaQuery.of(context).size.width,
                 height: MediaQuery.of(context).size.width,
                 decoration: BoxDecoration(
                   color: const Color(0xff0e0e9b),
                   shape: BoxShape.rectangle,
                   borderRadius: BorderRadius.zero,
                   border:
                       Border.all(color: const Color(0x4d9e9e9e), width: 1),
                 ),
               )
             ],
           ),
         ),
       ),
     ),
   );
 }
}
```

- lib > view > tarefas > dashboard.dart:



Estudar Java
POO
Aguardando
Prazo
9/11/2024

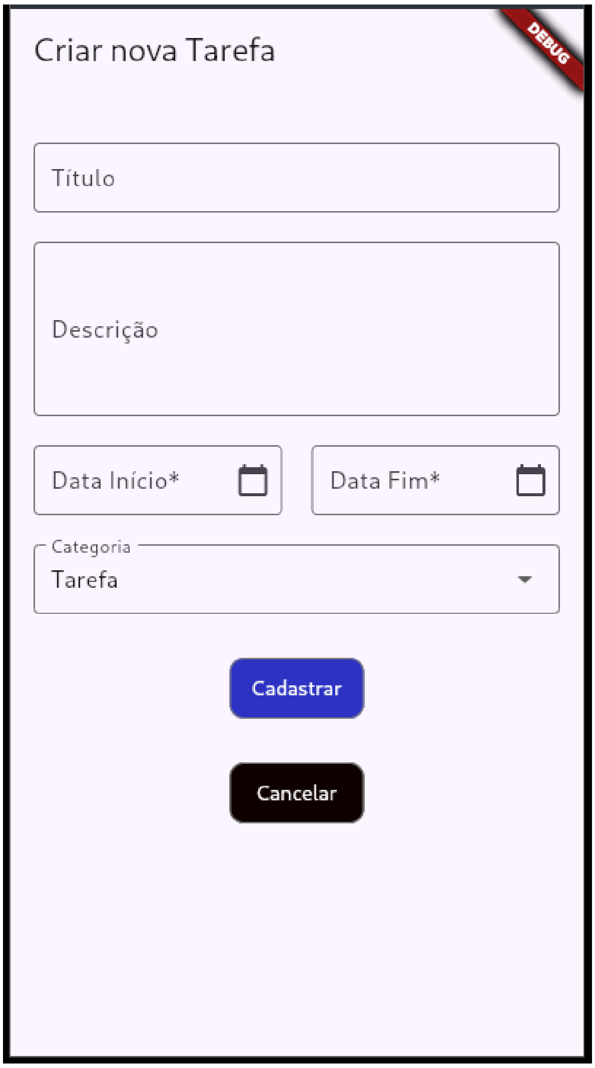
Estudar Flutter
Widgets
Concluido
Prazo
23/11/2024

Estudar Kotlin
Android Jetpack Compose
Em Progresso
Prazo
15/11/2024

```dart
import 'package:a3_udwmj/controller/db_sqlite.dart';
import 'package:a3_udwmj/view/config/settings.dart';
import 'package:a3_udwmj/view/tarefas/cad_tasks.dart';
import 'package:a3_udwmj/view/tarefas/task.dart';
import 'package:a3_udwmj/view/usuarios/login.dart';
import 'package:flutter/material.dart';

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

 @override
 _CardListScreenState createState() => _CardListScreenState();
}
```

```
class _CardListScreenState extends State<Dashboard> {
 List<Map<String, dynamic>>? jsonData;

 @override
 void initState() {
   super.initState();
   buscaTarefas();
 }

 Future<void> buscaTarefas() async {
   db_sqlite sqfliteInst = db_sqlite();
   try {
     final data = await sqfliteInst.getTasksByIdUser(user_Pub.userOn);
     setState(() {
       jsonData = data;
     });
   } catch (e) {
     print('Erro ao carregar os dados: $e');
   }
 }

 int? selectedIndex;

 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: const Text('Dashboard'),
       leading: Padding(
         padding: const EdgeInsets.all(8.0),
         child: Image.asset(
           'assets/Taskify.png',
           fit: BoxFit.contain,
         ),
       ),
     ),
     body: Column(
       children: [
         Expanded(
           child: ListView.builder(
             itemCount: jsonData!.length,
             itemBuilder: (context, index) {
                final item = jsonData!.elementAt(index);
                return Card(
                  color: Colors.primaries[index % Colors.primaries.length],
                  child: InkWell(
                    onTap: () {
                      idTask_Pub.idTsk = item["id"];

                      Navigator.push(
                        context,
                        MaterialPageRoute(
                          builder: (context) => const Task(),
                        ),
                      );
                    },
                    child: Padding(
                      padding: const EdgeInsets.all(16.0),
                      child: Row(
                        children: [
                          Expanded(
                            child: Column(
                              crossAxisAlignment: CrossAxisAlignment.start,
                              children: [
                                Row(
                                  children: [
                                    Expanded(
                                      child: Column(
                                        crossAxisAlignment:
                                            CrossAxisAlignment.start,
                                        children: [
                                          Text(
                                            '${item["titulo"]}',
                                            style:
                                                     const TextStyle(fontSize:
16),
                                          ),
                                          const SizedBox(height: 8),
                                          Text(
                                            '${item["descricao"]}',
                                            style:
                                                     const TextStyle(fontSize:
16),
                                          ),
                                          const SizedBox(height: 8),
                                          Text(
                                            '${item["status"]}',
                                            style:
                                                     const TextStyle(fontSize:
16),
                                          ),
                                        ],
                                      ),
                                    ),
                                    Expanded(
                                      child: Text(
                                        'Prazo\n${item["data_final"]}',
                                        style: const TextStyle(fontSize: 16),
                                        textAlign: TextAlign.right,
                                      ),
                                    ),
                                  ],
                                ),
                              ],
                            ),
                          ),
                        ],
                      ),
                    ),
                  ),
                );
              },
            ),
          ),
          Positioned(
            bottom: 0, // Distância da parte inferior
            right: 0, // Distância da parte direita
            child: FloatingActionButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (context) => const CadTask()),
                );
              },
              child: const Icon(Icons.add),
            ),
          ),
          RodapeApk(context),
         ],
       ),
     );
   }
}

Container RodapeApk(BuildContext context) {
   return Container(
     margin: const EdgeInsets.fromLTRB(0, 25, 0, 0),
     padding: const EdgeInsets.all(0),
     width: double.infinity, //MediaQuery.of(context).size.width,
     height: MediaQuery.of(context).size.height * 0.15,
     decoration: BoxDecoration(
       color: const Color.fromARGB(211, 14, 44, 133),
       shape: BoxShape.rectangle,
       borderRadius: BorderRadius.zero,
       border: Border.all(color: const Color(0x4d9e9e9e), width: 1),
     ),
     child: Row(
       mainAxisAlignment: MainAxisAlignment.center,
       crossAxisAlignment: CrossAxisAlignment.center,
       mainAxisSize: MainAxisSize.min,
       children: [
         Expanded(
           flex: 1,
           child: IconButton(
             icon: ClipRRect(
               borderRadius: BorderRadius.circular(8.0),
               child: Image.asset(
                 'assets/Dashboard.png',
                 width: 200,
                 height: 80,
                 fit: BoxFit.cover,
               ),
             ),
             onPressed: () {
               Navigator.push(
                 context,
                 MaterialPageRoute(builder: (context) => const Dashboard()),
               );
             },
             color: const Color.fromARGB(255, 45, 57, 127),
             iconSize: 24,
    ),
),
Expanded(
    flex: 1,
    child: IconButton(
      icon: ClipRRect(
        borderRadius: BorderRadius.circular(8.0),
        child: Image.asset(
          'assets/Calendario.png',
          width: 200,
          height: 80,
          fit: BoxFit.cover,
        ),
      ),
      onPressed: () {},
      color: const Color.fromARGB(255, 101, 110, 163),
      iconSize: 24,
    ),
),
Expanded(
    flex: 1,
    child: IconButton(
      icon: ClipRRect(
        borderRadius: BorderRadius.circular(8.0),
        child: Image.asset(
          'assets/Settings.png',
          width: 200,
          height: 80,
          fit: BoxFit.cover,
        ),
      ),
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => const Settings()),
        );
      },
      color: const Color(0xff212435),
      iconSize: 24,
    ),
),
Expanded(
    flex: 1,
          child: IconButton(
            icon: ClipRRect(
              borderRadius: BorderRadius.circular(8.0),
              child: Image.asset(
                'assets/Perfil.png',
                width: 200,
                height: 80,
                fit: BoxFit.cover,
              ),
            ),
            onPressed: () {},
            color: const Color(0xff212435),
            iconSize: 24,
          ),
        ),
      ],
    ),
  );
}
```

- lib > view > tarefas > cad_tasks.dart:


Criar nova Tarefa

Título

Descrição

Data Início*

Data Fim*

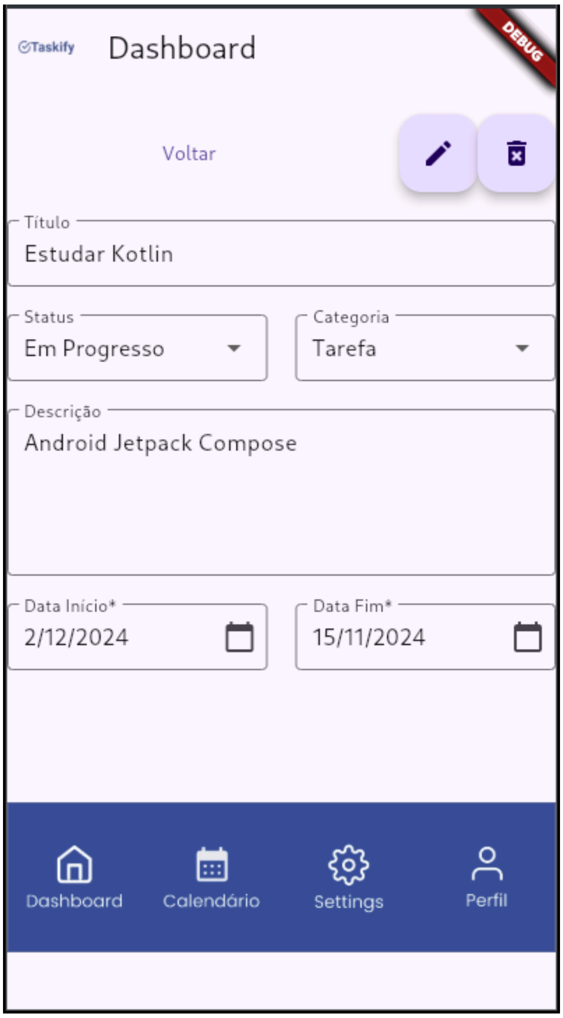
Categoria
Tarefa

```
import 'package:a3_udwmj/controller/db_sqlite.dart';
import 'package:a3_udwmj/view/tarefas/dashbord.dart';
import 'package:a3_udwmj/view/usuarios/login.dart';
import 'package:flutter/material.dart';

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

 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     home: Scaffold(
       appBar: AppBar(
         title: const Text('Criar nova Tarefa'),
       ),
       body: const Padding(
         padding: EdgeInsets.all(16.0),
         child: MyForm(),
       ),
     ),
   );
 }
}

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

 @override
 State<MyForm> createState() => _MyFormState();
}

class _MyFormState extends State<MyForm> {
 final TextEditingController dtInicio = TextEditingController();
 final TextEditingController dtFim = TextEditingController();
 final TextEditingController titulo = TextEditingController();
 final TextEditingController descricao = TextEditingController();

 Future<void> _selectDate(
     BuildContext context, TextEditingController controller) async {
   DateTime? pickedDate = await showDatePicker(
     context: context,
     initialDate: DateTime.now(),
     firstDate: DateTime(2000),
     lastDate: DateTime(2100),
   );

   if (pickedDate != null) {
     setState(() {
       controller.text =
           '${pickedDate.day}/${pickedDate.month}/${pickedDate.year}';
     });
   }
 }
```

```
@override
Widget build(BuildContext context) {
  db_sqlite sqfliteInst = db_sqlite();

  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      const SizedBox(height: 20),
      TextField(
        controller: titulo,
        maxLines: 1,
        decoration: const InputDecoration(
          labelText: 'Título',
          border: OutlineInputBorder(),
        ),
      ),
      const SizedBox(height: 20),
      TextField(
        controller: descricao,
        maxLines: 4,
        decoration: const InputDecoration(
          labelText: 'Descrição',
          border: OutlineInputBorder(),
        ),
      ),
      const SizedBox(height: 20),
      Row(
        children: [
          Expanded(
            child: TextField(
              controller: dtInicio,
              readOnly: true,
              decoration: InputDecoration(
                labelText: 'Data Início*',
                border: const OutlineInputBorder(),
                suffixIcon: IconButton(
                  icon: const Icon(Icons.calendar_today),
                  onPressed: () => _selectDate(context, dtInicio),
                ),
              ),
            ),
          ),
          const SizedBox(width: 20),
           Expanded(
             child: TextField(
               controller: dtFim,
               readOnly: true,
               decoration: InputDecoration(
                 labelText: 'Data Fim*',
                 border: const OutlineInputBorder(),
                 suffixIcon: IconButton(
                   icon: const Icon(Icons.calendar_today),
                   onPressed: () => _selectDate(context, dtFim),
                 ),
               ),
             ),
           ),
         ],
       ),
       const SizedBox(height: 20),
       const Padding(
         padding: EdgeInsets.all(0.0),
         child: DropdownField(),
       ),
       const SizedBox(height: 20),
       Padding(
         padding: const EdgeInsets.all(10),
         child: Align(
           alignment: const Alignment(0.0, 0.0),
           child: MaterialButton(
             onPressed: () async {
               if (titulo.text != "" &&
                   (_DropdownFieldState.categoria != null)) {
                 sqfliteInst.insertTask(
                     user_Pub.userOn,
                     titulo.text,
                     descricao.text,
                     dtInicio.text,
                     dtFim.text,
                     _DropdownFieldState.categoria.toString(),
                     'Em Progresso');
                 Navigator.push(
                   context,
                              MaterialPageRoute(builder: (context) => const
Dashboard()),
                 );
```

```
        }
      },
      color: const Color(0xff2f34c5),
      elevation: 0,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10.0),
        side: const BorderSide(color: Color(0xff808080), width: 1),
      ),
      padding: const EdgeInsets.all(16),
      textColor: const Color(0xffffffff),
      height: 50,
      minWidth: 100,
      child: const Text(
        "Cadastrar",
        style: TextStyle(
          fontSize: 14,
          fontWeight: FontWeight.w400,
          fontStyle: FontStyle.normal,
        ),
      ),
    ),
  ),
),
const SizedBox(height: 20),
Padding(
  padding: const EdgeInsets.all(0),
  child: Align(
    alignment: const Alignment(0, 0.0),
    child: MaterialButton(
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => const Dashboard()),
        );
      },
      color: const Color(0xff100202),
      elevation: 0,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10.0),
        side: const BorderSide(color: Color(0xff808080), width: 1),
      ),
      padding: const EdgeInsets.all(16),
      textColor: const Color(0xffffffff),
             height: 50,
             minWidth: 100,
             child: const Text(
               "Cancelar",
               style: TextStyle(
                 fontSize: 14,
                 fontWeight: FontWeight.w400,
                 fontStyle: FontStyle.normal,
               ),
             ),
           ),
         ),
       ),
     ],
   );
 }
}

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

 @override
 State<DropdownField> createState() => _DropdownFieldState();
}

class _DropdownFieldState extends State<DropdownField> {
 static String? categoria;

 final List<String> options = [
   'Tarefa',
   'Lembrete',
 ];

 @override
 Widget build(BuildContext context) {
   return DropdownButtonFormField<String>(
     decoration: const InputDecoration(
       labelText: 'Categoria',
       border: OutlineInputBorder(),
     ),
     value: categoria,
     isExpanded: true,
     items: options.map((String option) {
       return DropdownMenuItem<String>(
         value: option,
         child: Text(option),
       );
     }).toList(),
     onChanged: (String? newValue) {
       setState(() {
         categoria = newValue;
       });
     },
   );
 }
}
```

- lib > view > tarefas > tasks.dart:



Titulo: Estudar Kotlin

Status: Em Progresso

Categoria: Tarefa

Descrição: Android Jetpack Compose

Data Início*: 2/12/2024

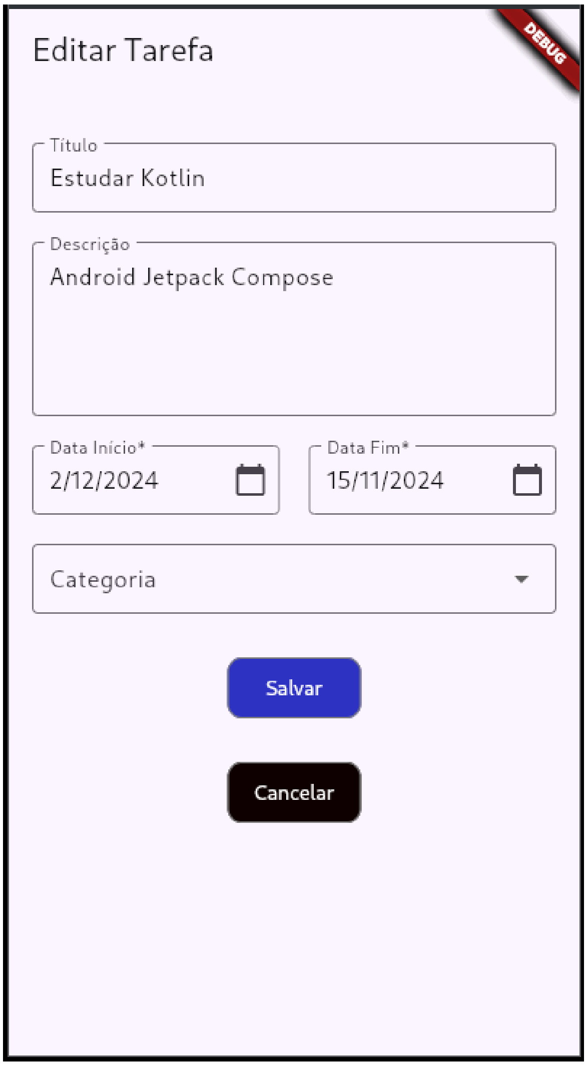
Data Fim*: 15/11/2024

```
import 'package:a3_udwmj/controller/db_sqlite.dart';
import 'package:a3_udwmj/view/tarefas/dashbord.dart';
import 'package:a3_udwmj/view/tarefas/edit_task.dart';
import 'package:a3_udwmj/view/usuarios/login.dart';
import 'package:flutter/material.dart';

class idTask_Pub {
 static int idTsk = 0;
}

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

 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     home: Scaffold(
       appBar: AppBar(
         title: const Text('Dashboard'),
         leading: Padding(
           padding: const EdgeInsets.all(8.0),
           child: Image.asset(
             'assets/Taskify.png',
             fit: BoxFit.contain,
           ),
         ),
       ),
       body: const Padding(
         padding: EdgeInsets.all(0.0),
         child: MyTask(),
       ),
     ),
   );
 }
}

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

 @override
 State<MyTask> createState() => _MyFormState();
}
```

```
class _MyFormState extends State<MyTask> {
 final TextEditingController dtInicio = TextEditingController();
 final TextEditingController dtFim = TextEditingController();
 final TextEditingController titulo = TextEditingController();
 final TextEditingController descricao = TextEditingController();
 final TextEditingController categoria = TextEditingController();

 @override
 void initState() {
   super.initState();
   loadTask();
 }

 Future<void> loadTask() async {
   Map<String, Object?> dados = await getTask(idTask_Pub.idTsk);
   setState(() {
     titulo.text = dados['titulo']!.toString();
     descricao.text = dados['descricao']!.toString();
     dtInicio.text = dados['data_inicial']!.toString();
     dtFim.text = dados['data_final']!.toString();
     DropdownUtils.StatusDropdownValue = dados['status']!.toString();
     DropdownUtils.CategoriaDropdownValue = dados['categoria']!.toString();
   });
 }

 Future<Map<String, Object?>> getTask(int id) async {
   db_sqlite sqfliteInst = db_sqlite();
   return await sqfliteInst.getTaskByIdTask(idTask_Pub.idTsk);
 }

 Future<void> _selectDate(
     BuildContext context, TextEditingController controller) async {
   DateTime? pickedDate = await showDatePicker(
     context: context,
     initialDate: DateTime.now(),
     firstDate: DateTime(2000),
     lastDate: DateTime(2100),
   );

   if (pickedDate != null) {
     setState(() {
       controller.text =
           '${pickedDate.day}/${pickedDate.month}/${pickedDate.year}';
     });
   }
 }

 @override
 Widget build(BuildContext context) {
   db_sqlite sqfliteInst = db_sqlite();

   var value;
   return Scaffold(
       body: Column(
     crossAxisAlignment: CrossAxisAlignment.start,
     children: [
       const SizedBox(height: 20),
       Row(
         children: [
           Expanded(
             child: TextButton(
               onPressed: () {
                 if (titulo.text != "" && (descricao.text != "")) {
                   sqfliteInst.updateTask(
                       idTask_Pub.idTsk,
                       user_Pub.userOn,
                       titulo.text,
                       descricao.text,
                       dtInicio.text,
                       dtFim.text,
                       DropdownUtils.CategoriaDropdownValue.toString(),
                       DropdownUtils.StatusDropdownValue.toString());

                   Navigator.push(
                     context,
                     MaterialPageRoute(
                         builder: (context) => const Dashboard()),
                   );
                 }
               },
               child: const Text(
                 "Voltar",
                 style: TextStyle(
                   fontSize: 14,
                   fontWeight: FontWeight.w400,
                    fontStyle: FontStyle.normal,
                  ),
                ),
              ),
            ),
            const SizedBox(width: 20),
            Positioned(
              bottom: 0,
              right: 0,
              child: FloatingActionButton(
                onPressed: () {
                  Navigator.push(
                    context,
                               MaterialPageRoute(builder:  (context)  =>  const
 EditTask()),
                  );
                },
                child: const Icon(Icons.edit),
              ),
            ),
            Positioned(
              bottom: 0,
              right: 0,
              child: FloatingActionButton(
                onPressed: () {
                  sqfliteInst.deleteTask(idTask_Pub.idTsk);

                  Navigator.push(
                    context,
                               MaterialPageRoute(builder:  (context)  =>  const
 Dashboard()),
                  );
                },
                child: const Icon(Icons.delete_forever),
              ),
            ),
          ],
        ),
        const SizedBox(height: 20),
        TextField(
          // enabled: false,
          readOnly: true,
          controller: titulo,
   maxLines: 1,
   decoration: const InputDecoration(
     labelText: 'Título',
     border: OutlineInputBorder(),
   ),
 ),
 const SizedBox(height: 20),
 Row(
   children: [
     Expanded(
       child: DropdownUtils.StatusDropdown(
         value: DropdownUtils.StatusDropdownValue, //"Atrasado"
         onChanged: (newValue) {
           setState(() {
             DropdownUtils.StatusDropdownValue = newValue;
           });
         },
       ),
     ),
     const SizedBox(width: 20),
     Expanded(
       child: DropdownUtils.CategoriaDropdown(
         value: DropdownUtils.CategoriaDropdownValue, //"Tarefa"
         onChanged: (newValue) {
           setState(() {
             DropdownUtils.CategoriaDropdownValue = newValue;
           });
         },
       ),
     ),
   ],
 ),
 const SizedBox(height: 20),
 TextField(
   // enabled: false,
   readOnly: true,
   controller: descricao,
   maxLines: 4,
   decoration: const InputDecoration(
     labelText: 'Descrição',
     border: OutlineInputBorder(),
   ),
 ),
    const SizedBox(height: 20),
    Row(
      // Data Inicio e Fim
      children: [
        Expanded(
          child: TextField(
            //  enabled: false,

            controller: dtInicio,
            readOnly: true,
            decoration: InputDecoration(
              labelText: 'Data Início*',
              border: const OutlineInputBorder(),
              suffixIcon: IconButton(
                icon: const Icon(Icons.calendar_today),
                onPressed: () => _selectDate(context, dtInicio),
              ),
            ),
          ),
        ),
        const SizedBox(width: 20),
        Expanded(
          child: TextField(
            //  enabled: false,
            controller: dtFim,
            readOnly: true,
            decoration: InputDecoration(
              labelText: 'Data Fim*',
              border: const OutlineInputBorder(),
              suffixIcon: IconButton(
                icon: const Icon(Icons.calendar_today),
                onPressed: () => _selectDate(context, dtFim),
              ),
            ),
          ),
        ),
      ],
    ),
    const SizedBox(height: 20),
    const SizedBox(height: 50),
    RodapeApk(context),
  ],
));
```

```
 }
}

class DropdownUtils {
 static String? StatusDropdownValue, CategoriaDropdownValue;

 static Widget StatusDropdown({
   required String? value,
   required ValueChanged<String?> onChanged,
 }) {
   // value = StatusDropdownValue;
   return DropdownButtonFormField<String>(
     decoration: const InputDecoration(
       labelText: 'Status',
       border: OutlineInputBorder(),
     ),
     value: value,
     onChanged: onChanged,
     items: <String>['Em Progresso', 'Aguardando', 'Atrasado', 'Concluido']
         .map<DropdownMenuItem<String>>((String value) {
       return DropdownMenuItem<String>(
         value: value,
         child: Text(value),
       );
     }).toList(),
   );
 }

 static Widget CategoriaDropdown({
   required String? value,
   required ValueChanged<String?> onChanged,
 }) {
   //  value = CategoriaDropdownValue;
   return DropdownButtonFormField<String>(
     decoration: const InputDecoration(
       labelText: 'Categoria',
       border: OutlineInputBorder(),
     ),
     value: value,
     onChanged: onChanged,
     items: <String>['Tarefa', 'Lembrete']
         .map<DropdownMenuItem<String>>((String value) {
       return DropdownMenuItem<String>(
         value: value,
         child: Text(value),
       );
     }).toList(),
   );
 }
}
```

- lib > view > tarefas > edit_tasks.dart:



Editar Tarefa

Título: Estudar Kotlin

Descrição: Android Jetpack Compose

Data Início*: 2/12/2024

Data Fim*: 15/11/2024

Categoria

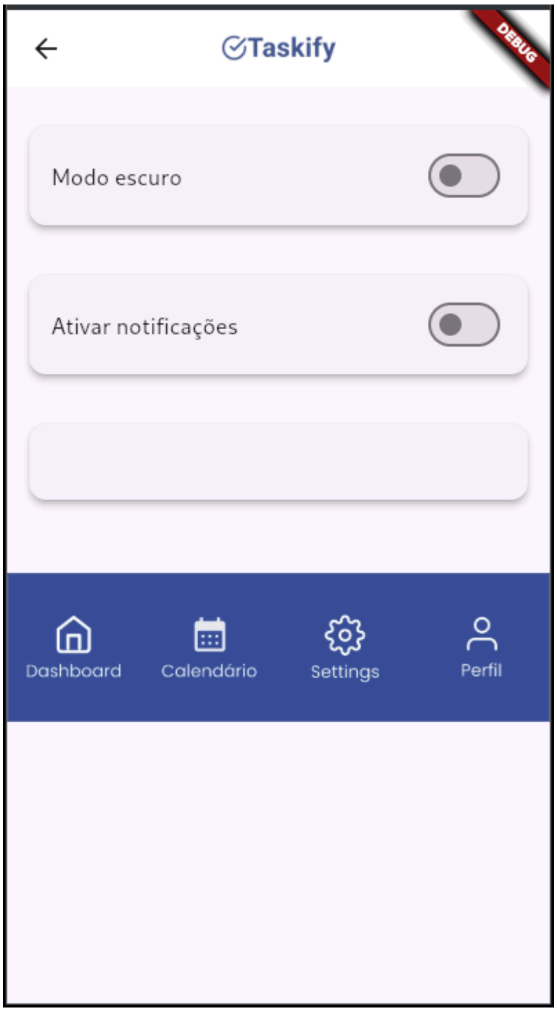
```
import 'package:a3_udwmj/controller/db_sqlite.dart';
import 'package:a3_udwmj/view/tarefas/task.dart';
import 'package:a3_udwmj/view/usuarios/login.dart';
import 'package:flutter/material.dart';
```

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

 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     home: Scaffold(
       appBar: AppBar(
         title: const Text('Editar Tarefa'),
       ),
       body: const Padding(
         padding: EdgeInsets.all(16.0),
         child: MyTask(),
       ),
     ),
   );
 }
}

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

 @override
 State<MyTask> createState() => _MyFormState();
}

class _MyFormState extends State<MyTask> {
 final TextEditingController dtInicio = TextEditingController();
 final TextEditingController dtFim = TextEditingController();
 final TextEditingController titulo = TextEditingController();
 final TextEditingController descricao = TextEditingController();

 @override
 void initState() {
   super.initState();
   loadTask();
 }

 Future<void> loadTask() async {
   Map<String, Object?> dados = await getTask(idTask_Pub.idTsk);
   setState(() {
     titulo.text = dados['titulo']!.toString();
```

```
    descricao.text = dados['descricao']!.toString();
    dtInicio.text = dados['data_inicial']!.toString();
    dtFim.text = dados['data_final']!.toString();
    DropdownUtils.StatusDropdownValue = dados['status']!.toString();
    _DropdownFieldState.categoria = dados['categoria']!.toString();
  });
}

Future<Map<String, Object?>> getTask(int id) async {
  db_sqlite sqfliteInst = db_sqlite();
  return await sqfliteInst.getTaskByIdTask(idTask_Pub.idTsk);
}

Future<void> _selectDate(
    BuildContext context, TextEditingController controller) async {
  DateTime? pickedDate = await showDatePicker(
    context: context,
    initialDate: DateTime.now(),
    firstDate: DateTime(2000),
    lastDate: DateTime(2100),
  );

  if (pickedDate != null) {
    setState(() {
      controller.text =
          '${pickedDate.day}/${pickedDate.month}/${pickedDate.year}';
    });
  }
}

@override
Widget build(BuildContext context) {
  db_sqlite sqfliteInst = db_sqlite();

  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      const SizedBox(height: 20),
      TextField(
        controller: titulo,
        maxLines: 1,
        decoration: const InputDecoration(
          labelText: 'Título',
    border: OutlineInputBorder(),
  ),
),
const SizedBox(height: 20),
TextField(
  controller: descricao,
  maxLines: 4,
  decoration: const InputDecoration(
    labelText: 'Descrição',
    border: OutlineInputBorder(),
  ),
),
const SizedBox(height: 20),
Row(
  children: [
    Expanded(
      child: TextField(
        controller: dtInicio,
        readOnly: true,
        decoration: InputDecoration(
          labelText: 'Data Início*',
          border: const OutlineInputBorder(),
          suffixIcon: IconButton(
            icon: const Icon(Icons.calendar_today),
            onPressed: () => _selectDate(context, dtInicio),
          ),
        ),
      ),
    ),
    const SizedBox(width: 20),
    Expanded(
      child: TextField(
        controller: dtFim,
        readOnly: true,
        decoration: InputDecoration(
          labelText: 'Data Fim*',
          border: const OutlineInputBorder(),
          suffixIcon: IconButton(
            icon: const Icon(Icons.calendar_today),
            onPressed: () => _selectDate(context, dtFim),
          ),
        ),
      ),
    ),
  ],
),
const SizedBox(height: 20),
const Padding(
  // Categoria
  padding: EdgeInsets.all(0.0),
  child: DropdownField(),
),
const SizedBox(height: 20),
Padding(
  padding: const EdgeInsets.all(10),
  child: Align(
    alignment: const Alignment(0.0, 0.0),
    child: MaterialButton(
      onPressed: () async {
        if (titulo.text != "" &&
            (_DropdownFieldState.categoria != null)) {
          sqfliteInst.updateTask(
              idTask_Pub.idTsk,
              user_Pub.userOn,
              titulo.text,
              descricao.text,
              dtInicio.text,
              dtFim.text,
              _DropdownFieldState.categoria.toString(),
              DropdownUtils.StatusDropdownValue.toString());
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => const Task()),
          );
        }
      },
      color: const Color(0xff2f34c5),
      elevation: 0,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10.0),
        side: const BorderSide(color: Color(0xff808080), width: 1),
      ),
      padding: const EdgeInsets.all(16),
      textColor: const Color(0xffffffff),
      height: 50,
      minWidth: 100,
      child: const Text(
        "Salvar",
        style: TextStyle(
          fontSize: 14,
          fontWeight: FontWeight.w400,
          fontStyle: FontStyle.normal,
        ),
      ),
    ),
  ),
),
const SizedBox(height: 20),
Padding(
  padding: const EdgeInsets.all(0),
  child: Align(
    alignment: const Alignment(0, 0.0),
    child: MaterialButton(
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => const Task()),
        );
      },
      color: const Color(0xff100202),
      elevation: 0,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10.0),
        side: const BorderSide(color: Color(0xff808080), width: 1),
      ),
      padding: const EdgeInsets.all(16),
      textColor: const Color(0xffffffff),
      height: 50,
      minWidth: 100,
      child: const Text(
        "Cancelar",
        style: TextStyle(
          fontSize: 14,
          fontWeight: FontWeight.w400,
          fontStyle: FontStyle.normal,
        ),
      ),
    ),
  ),
       ),
     ],
   );
 }
}

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

 @override
 State<DropdownField> createState() => _DropdownFieldState();
}

class _DropdownFieldState extends State<DropdownField> {
 static String? categoria;

 // Lista de opções
 final List<String> options = [
   'Tarefa',
   'Lembrete',
 ];

 @override
 Widget build(BuildContext context) {
   return DropdownButtonFormField<String>(
     decoration: const InputDecoration(
       labelText: 'Categoria',
       border: OutlineInputBorder(),
     ),
     value: categoria,
     isExpanded: true,
     items: options.map((String option) {
       return DropdownMenuItem<String>(
         value: option,
         child: Text(option),
       );
     }).toList(),
     onChanged: (String? newValue) {
       setState(() {
         categoria = newValue;
       });
     },
   );
```

```
  }
}
```

- lib > view > tarefas > settings.dart:



```
import 'package:a3_udwmj/main.dart';
import 'package:a3_udwmj/view/tarefas/dashbord.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

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

 @override
 Widget build(BuildContext context) {
   return const MaterialApp(
     debugShowCheckedModeBanner: false,
     home: SettingsSfW(),
   );
 }
}

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

 @override
 State<SettingsSfW> createState() => _ExampleScreenState();
}

class _ExampleScreenState extends State<SettingsSfW> {
 bool modoEscuro = false, ativNotif = false;

 @override
 Widget build(BuildContext context) {
 final themeProvider = Provider.of<ThemeProvider>(context);
   return Scaffold(
     appBar: AppBar(
       leading: IconButton(
         icon: const Icon(Icons.arrow_back),
         onPressed: () {
           // Lógica para voltar à tela anterior
           Navigator.pop(context);
         },
       ),
       title: Image.asset(
         'assets/Taskify.png',
         height: 20,
       ),
       centerTitle: true,
       backgroundColor: const Color.fromARGB(255, 255, 255, 255),
     ),
     body: Column(
       children: [
         const SizedBox(height: 20),

         Card(
           margin: const EdgeInsets.symmetric(horizontal: 16, vertical: 8),
           elevation: 4,
  child: Padding(
    padding: const EdgeInsets.all(16),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        const Text(
          'Modo escuro',
          style: TextStyle(fontSize: 16),
        ),
        Switch(
          value: themeProvider.themeMode == ThemeMode.dark,
          onChanged: (value) {
            setState(() {
              themeProvider.toggleTheme();
            });
          },

        ),
      ],
    ),
  ),
),
const SizedBox(height: 20),
Card(
  margin: const EdgeInsets.symmetric(horizontal: 16, vertical: 8),
  elevation: 4,
  child: Padding(
    padding: const EdgeInsets.all(16),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        const Text(
          'Ativar notificações',
          style: TextStyle(fontSize: 16),
        ),
        Switch(
          value: ativNotif,
          onChanged: (value) {
            setState(() {
              ativNotif = value;
            });
          },
        ),
               ],
             ),
           ),
         ),
         const SizedBox(height: 20),
         Card(
           margin: const EdgeInsets.symmetric(horizontal: 16, vertical: 8),
           elevation: 4,
           child: InkWell(
             onTap: () {
               // Ação do botão
               ScaffoldMessenger.of(context).showSnackBar(
                 const SnackBar(content: Text('Botão pressionado')),
               );
             },
             child: const Padding(
               padding: EdgeInsets.all(16),
               child: Row(
                 mainAxisAlignment: MainAxisAlignment.center,
                 children: [
                   Text(
                     '',
                     style: TextStyle(fontSize: 16, color: Colors.blue),
                   ),
                 ],
               ),
             ),
           ),
         ),
         const SizedBox(height: 20),
         RodapeApk(context),
       ],
     ),

   );
 }
}
```