import 'package:gerenciadorsenhas/Data/senha\_entity.dart';

import 'package:path/path.dart';

import 'package:sqflite/sqflite.dart';

import 'data\_container.dart';

class senhaSQLiteDatasource {

  Future<Database> \_getDatabase() async {

    // await deleteDatabase(

    //   join(await getDatabasesPath(),DATABASE\_NAME),

    // );

    return openDatabase(

      join(await getDatabasesPath(), DATABASE\_NAME),

      onCreate: (db, version) async {

        await db.execute(CREATE\_SENHA\_TABLE\_SCRIPT);

       /\* await db.rawInsert('''insert into $SENHA\_TABLE\_NAME(

              $SENHA\_COLUMN\_DESCRICAO,

              $SENHA\_COLUMN\_LOGIN,

              $SENHA\_COLUMN\_SENHA)

              values(

                'admin','admin@admin','123')

              ''');\*/

      },

      version: 1,

    );

  }

  Future inserirSenha(descricao, login, senha) async {

    try {

      final Database db = await \_getDatabase();

      senha.senhaID = await db.rawInsert('''insert into $SENHA\_TABLE\_NAME(

              $SENHA\_COLUMN\_DESCRICAO,

              $SENHA\_COLUMN\_LOGIN,

              $SENHA\_COLUMN\_SENHA)

              values(

                '${senha.descricao}','${senha.login}','${senha.senha}')

              ''');

      queryAllRows();

    } catch (ex) {

      return;

    }

  }

  Future<List<Map<String, dynamic>>> queryAllRows() async {

    Database db = await \_getDatabase();

    return await db.query(SENHA\_TABLE\_NAME);

  }

  Future<List<SenhaEntity>> getAllSenha() async {

    Database db = await \_getDatabase();

    List<Map> dbResult = await db.rawQuery('SELECT \* from $SENHA\_TABLE\_NAME');

    List<SenhaEntity> senhas = [];

    for (var row in dbResult) {

      SenhaEntity senha = SenhaEntity();

      senha.senhaID = row['senhaID'];

      senha.descricao = row['descricao'];

      senha.login = row['login'];

      senha.senha = row['senha'];

      senhas.add(senha);

    }

    return senhas;

  }

  Future<void> atualizarSenha(SenhaEntity senha) async {

    Database db = await \_getDatabase();

    await db.transaction((txn) async {

      await txn.rawUpdate('update $SENHA\_TABLE\_NAME set $SENHA\_COLUMN\_DESCRICAO = ?, $SENHA\_COLUMN\_LOGIN = ?, $SENHA\_COLUMN\_SENHA = ? where id = ?', [senha.descricao, senha.login, senha.senha]);

    });

  }

  Future<void> deletarSenha(SenhaEntity senha) async {

    Database db = await \_getDatabase();

    await db.transaction((txn) async {

      await txn.rawUpdate('delete from $SENHA\_TABLE\_NAME where id = ?', [senha.senhaID]);

    });

  }

  Future<List<SenhaEntity>> pesquisarSenha(String filtro) async {

  List<SenhaEntity> senhas = [];

  Database db = await \_getDatabase();

  List<Map> dbResult = await db.rawQuery('SELECT \* from $SENHA\_TABLE\_NAME where $SENHA\_COLUMN\_DESCRICAO like ?', ['%$filtro%']);

  for (var row in dbResult) {

    SenhaEntity senha = SenhaEntity();

    senha.senhaID = row['senhaID'];

    senha.descricao = row['descricao'];

    senha.login = row['login'];

    senha.senha = row['senha'];

    senhas.add(senha);

  }

    return senhas;

  }

}

// ignore\_for\_file: prefer\_const\_constructors

import 'package:flutter/material.dart';

import 'Data/senha\_entity.dart';

import 'Data/senha\_sqlite\_datasource..dart';

class cadsenhas extends StatefulWidget {

  const cadsenhas({Key? key}) : super(key: key);

  @override

  \_cadsenhasState createState() {

    return \_cadsenhasState();

  }

}

class \_cadsenhasState extends State<cadsenhas> {

  TextEditingController descricaoController = TextEditingController();

  TextEditingController loginController = TextEditingController();

  TextEditingController senhaController = TextEditingController();

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

        appBar: AppBar(

          title: const Text('Cadastro de senhas'),

          leading: IconButton(

            icon: Icon(Icons.arrow\_back),

            color: Colors.black38,

            onPressed: () => Navigator.pop(context, false),

          ),

        ),

        body: Stack(

          children: <Widget>[

            ListView(children: <Widget>[

              fieldDescricao(),

              fieldLogin(),

              fieldSenha(),

            ])

          ],

        ),

        floatingActionButton: FloatingActionButton(

          backgroundColor: Colors.green,

          foregroundColor: Colors.black,

          onPressed: () {

            senhaSQLiteDatasource().inserirSenha(descricaoController, loginController, senhaController);

          },

          child: Icon(Icons.add),

        ),

      ),

    );

  }

  Widget fieldDescricao() {

    return Container(

      padding: const EdgeInsets.all(10),

      child: TextField(

        controller: descricaoController,

        keyboardType: TextInputType.text,

        decoration: const InputDecoration(

          border: OutlineInputBorder(),

          labelText: 'Informe a descrição',

        ),

      ),

    );

  }

  Widget fieldLogin() {

    return Container(

      padding: const EdgeInsets.all(10),

      child: TextField(

        controller: loginController,

        keyboardType: TextInputType.text,

        decoration: const InputDecoration(

          border: OutlineInputBorder(),

          labelText: 'Login',

        ),

      ),

    );

  }

  Widget fieldSenha() {

    return Container(

      padding: const EdgeInsets.all(10),

      child: TextField(

        controller: senhaController,

        keyboardType: TextInputType.text,

        decoration: const InputDecoration(

          border: OutlineInputBorder(),

          labelText: 'Senha',

        ),

      ),

    );

  }

}