

Programação para Dispositivos Móveis Projeto Prático

Amanda Macedo Jaria RA:2840481823004

Lummy Maria Oliveira Monteiro RA:2840481723040

endereço do GitHub (repositório privado): <https://github.com/lummymaria/ifisio>

dados para acesso (tela de login): email -> plotze@hotmail.com

senha -> senha123

Código Fonte e Imagem das Telas:

```
import 'dart:async';
import 'package:flutter/rendering.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

class ProcedimentosF {
  //Atributos
  String _id;
  String _codigoProcedimento;
  String _descricaoProcedimentos;
  String _numSeries;
  String _numRepeticoes;

  //Construtor
  ProcedimentosF(this._id, this._codigoProcedimento,
    this._descricaoProcedimentos, this._numSeries,
    this._numRepeticoes);

  //Getters
  String get id => _id;
  String get codigoProcedimento => _codigoProcedimento;
  String get descricaoProcedimentos => _descricaoProcedimentos;
  String get numSeries => _numSeries;
  String get numRepeticoes => _numRepeticoes;

  ProcedimentosF.map(dynamic obj) {
    this._id = obj['id'];
    this._codigoProcedimento = obj['codigoProcedimentos'];
    this._descricaoProcedimentos = obj['descricaoProcedimentos'];
    this._numSeries = obj['numSeries'];
    this._numRepeticoes = obj['numRepeticoes'];
  }

  //Converter os dados para um Mapa
  Map<String, dynamic> toMap() {
    var map = Map<String, dynamic>();
    if (_id != null) {
      map["id"] = _id;
    }
    map["codigoProcedimento"] = _codigoProcedimento;
    map["descricaoProcedimento"] = _descricaoProcedimentos;
    map["numSeries"] = _numSeries;
  }
}
```

```

        map["numRepeticoes"] = _numRepeticoes;

        return map;
    }

    //Converter um Mapa para o modelo de dados
    ProcedimentosF.fromMap(Map<String, dynamic> map, String id) {
        //Atribuir id ao this._id, somente se id não for
        //nulo, caso contrário atribui '' (vazio).
        this._id = id ?? '';
        this._codigoProcedimento = map["codigoProcediment"];
        this._descricaoProcedimentos = map["descricaoProcedimentos"];
        this._numSeries = map["numSeries"];
        this._numRepeticoes = map["numRepeticoes"];
    }
}

class AtendimentoF {
    //Atributos
    String _id;
    String _codigoAtendimento;
    String _dataAtendimento;
    String _procedimentosRealizados;

    //Construtor
    AtendimentoF(this._id, this._codigoAtendimento,
        this._dataAtendimento,
        this._procedimentosRealizados);

    //Getters
    String get id => _id;
    String get codigoAtendimento => _codigoAtendimento;
    String get dataAtendimento => _dataAtendimento;
    String get procedimentosRealizados => _procedimentosRealizados;

    AtendimentoF.map(dynamic obj) {
        this._id = obj['id'];
        this._codigoAtendimento = obj['codigoAtendimento'];
        this._dataAtendimento = obj['dataAtendimento'];
        this._procedimentosRealizados = obj['procedimentosRealizados'];
    }

    //Converter os dados para um Mapa
    Map<String, dynamic> toMap() {
        var map = Map<String, dynamic>();
        if (_id != null) {
            map["id"] = _id;
        }
        map["codigoAtendimento"] = _codigoAtendimento;
        map["dataAtendimento"] = _dataAtendimento;
        map["procedimentosRealizados"] = _procedimentosRealizados;
        return map;
    }

    //Converter um Mapa para o modelo de dados
    AtendimentoF.fromMap(Map<String, dynamic> map, String id) {
        //Atribuir id ao this._id, somente se id não for
        //nulo, caso contrário atribui '' (vazio).
        this._id = id ?? '';
        this._codigoAtendimento = map["codigoAtendimento"];
        this._dataAtendimento = map["dataAtendimento"];
    }
}

```

```

        this._procedimentosRealizados = map["procedimentosRealizados"];
    }
}

class Pessoa {
    int codigoPessoa;
    String nome;
    String dataNascimento;
    int cep;
    int numeroCasa;
    String complemento;
    int telContato;
}

class Fisioterapeuta extends Pessoa {
    String crefito;
    String email;
    String cpf;
}

class Paciente extends Pessoa {
    String diagnostico;
    String informacaoExtra;
    String responsavel;
}

class Responsavel extends Pessoa {
    String email;
    int celular;
    String tutorado;
}

class Atendimento {
    int codigoAtendimento;
    String dataAtendimento;
    String horaInicio;
    String horaFim;
    String procedimentosRealizados;
    int codigoPaciente;
    int codigoFisioterapeuta;
}

class Procedimento {
    int codigoProcedimento;
    String descricaoProcedimento;
    int numSeries;
    int numRepeticoes;
}

class Agendamento {
    String nomePaciente;
    String nomeResponsavel;
    String dataConsulta;
    String horaConsulta;
    String localConsulta;
    String fisioResponsavel;
}

class Item {
    var icone;
    var foto;
}

```

```

    int codigoPessoa;
    String nome;
    String dataNascimento;
    int cep;
    int numeroCasa;
    String complemento;
    int telContato;
    String diagnostico;
    String informacaoExtra;
    String responsavel;

    Item(
        this.icone,
        this.nome,
        this.codigoPessoa,
        this.foto,
        this.dataNascimento,
        this.cep,
        this.numeroCasa,
        this.complemento,
        this.telContato,
        this.diagnostico,
        this.informacaoExtra,
        this.responsavel);
}

// Definir as informações que serão apresentadas em cada item da Lista
class ItemResponsaveis {
    var icone;
    var foto;
    int codigoPessoa;
    String nome;
    String dataNascimento;
    int cep;
    int numeroCasa;
    String complemento;
    int telContato;
    String tutorado;
    String email;

    ItemResponsaveis(
        this.icone,
        this.nome,
        this.codigoPessoa,
        this.foto,
        this.dataNascimento,
        this.cep,
        this.numeroCasa,
        this.complemento,
        this.telContato,
        this.tutorado,
        this.email);
}

// Definir as informações que serão apresentadas em cada item da Lista
class ItemFisioterapeutas {
    var icone;
    var foto;
    int codigoPessoa;
    String nome;
    String dataNascimento;

```

```

int cep;
int numeroCasa;
String complemento;
int telContato;
String crefito;
String email;

ItemFisioterapeutas(
    this.icone,
    this.nome,
    this.codigoPessoa,
    this.foto,
    this.dataNascimento,
    this.cep,
    this.numeroCasa,
    this.complemento,
    this.telContato,
    this.crefito,
    this.email);
}

// Definir as informações que serão apresentadas em cada item da Lista
class ItemAtendimento {
    var icone;
    var foto;
    int codigoAtendimento;
    String dataAtendimento;
    String horaInicio;
    String horaFim;
    String procedimentosRealizados;
    int codigoPaciente;
    int codigoFisioterapeuta;

    ItemAtendimento(
        this.icone,
        this.foto,
        this.codigoAtendimento,
        this.dataAtendimento,
        this.horaInicio,
        this.horaFim,
        this.procedimentosRealizados,
        this.codigoPaciente,
        this.codigoFisioterapeuta,
    );
}

// Definir as informações que serão apresentadas em cada item da Lista
class ItemProcedimento {
    var icone;
    var foto;
    int codigoProcedimento;
    String descricaoProcedimento;
    int numSeries;
    int numRepeticoes;

    ItemProcedimento(
        this.icone,
        this.foto,
        this.codigoProcedimento,
        this.descricaoProcedimento,
        this.numSeries,
    );
}

```

```

        this.numRepeticoes,
    );
}

Paciente o_paciente = new Paciente();
Fisioterapeuta o_fisioterapeuta = new Fisioterapeuta();
Responsavel o_responsavel = new Responsavel();
Atendimento o_atendimento = new Atendimento();
Procedimento o_procedimento = new Procedimento();
Agendamento o_agendamento = new Agendamento();

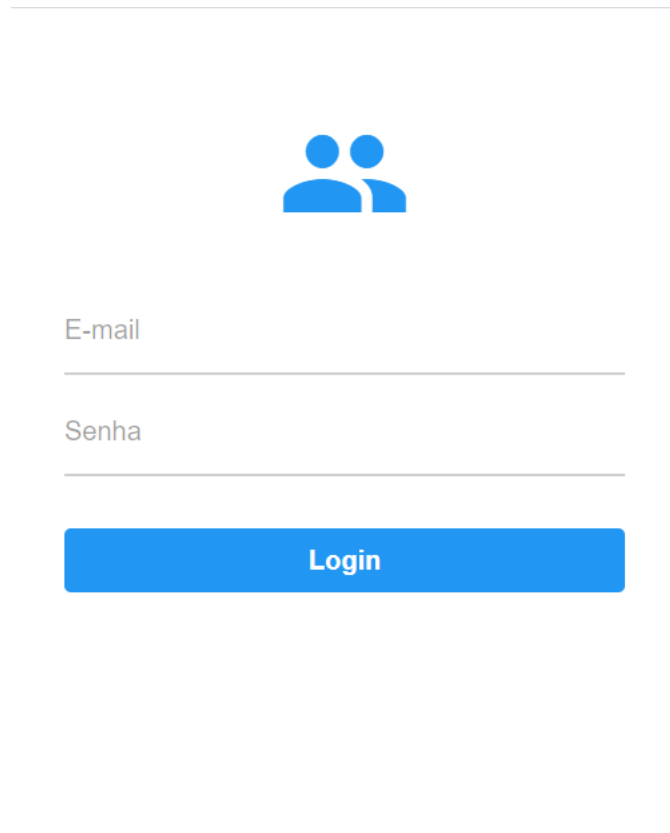
void main() {
    runApp(
        MaterialApp(
            debugShowCheckedModeBanner: false,
            title: 'Navegação',

            //home: MyApp(),

            // ROTAS de Navegação
            initialRoute: '/pagina17',
            routes: {
                '/pagina17': (context) => Inicio(),
                '/pagina1': (context) => Login(),
                '/pagina2': (context) => ExplicacaoPagina(),
                '/pagina3': (context) => MenuPrincipal(),
                '/pagina4': (context) => MenuCadastroPagina(),
                '/pagina5': (context) => CadastroPacientes(),
                '/pagina6': (context) => PacientesCadastradosPagina(),
                '/pagina7': (context) => CadastroResponsavel(),
                '/pagina8': (context) => ResponsaveisCadastradosPagina(),
                '/pagina9': (context) => CadastroFisioterapeuta(),
                '/pagina10': (context) =>
FisioterapeutasCadastradosPagina(),
                '/pagina11': (context) => CadastroAtendimento(),
                '/pagina12': (context) => AtendimentosCadastrados(),
                '/pagina13': (context) => CadastroProcedimento(),
                '/pagina14': (context) => ProcedimentosCadastrados(),
                '/pagina15': (context) => AgendamentoPagina(),
                '/pagina16': (context) => MenuListasPagina()
            },
        ),
    );
}

```

```
//  
// PRIMEIRA PÁGINA - LOGIN  
//
```



The image shows a login form within a light gray rounded rectangle. At the top center is a blue icon of two people. Below it are two text input fields: the first is labeled 'E-mail' and the second is labeled 'Senha'. At the bottom center is a blue button with the text 'Login' in white.

```
class MyApp extends StatefulWidget {  
  _MyAppState createState() => _MyAppState();  
}  
  
class _MyAppState extends State<MyApp> {  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      debugShowCheckedModeBanner: false,  
      title: "iFisio",  
      home: Login(),  
    );  
  }  
}  
  
class Login extends StatefulWidget {  
  @override  
  _LoginState createState() => _LoginState();  
}  
  
final FirebaseAuth mAuth = FirebaseAuth.instance;  
  
class _LoginState extends State<Login> {  
  //Widget appBarTitle = Text('iFisio');  
  TextEditingController emailController = new TextEditingController();  
  TextEditingController passwordController = new  
  TextEditingController();
```

```

String _email, _password;
// ignore: unused_element

final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      padding: EdgeInsets.only(top: 60, left: 40, right: 40),
      color: Colors.white,
      child: ListView(
        children: <Widget>[
          SizedBox(
            width: 128,
            height: 128,
            child: Icon(Icons.people, size: 100, color:
Colors.blue),
          ),
          SizedBox(
            height: 20,
          ),
          TextFormField(
            // autofocus: true,
            keyboardType: TextInputType.emailAddress,
            onChanged: (input) => _email = input,
            decoration: InputDecoration(
              labelText: "E-mail",
              labelStyle: TextStyle(
                color: Colors.black38,
                fontWeight: FontWeight.w400,
                fontSize: 20,
              ),
            ),
            style: TextStyle(fontSize: 20),
          ),
          SizedBox(
            height: 10,
          ),
          TextFormField(
            // autofocus: true,
            keyboardType: TextInputType.text,
            onChanged: (input) => _password = input,
            obscureText: true,
            decoration: InputDecoration(
              labelText: "Senha",
              labelStyle: TextStyle(
                color: Colors.black38,
                fontWeight: FontWeight.w400,
                fontSize: 20,
              ),
            ),
            style: TextStyle(fontSize: 20),
          ),
          SizedBox(
            height: 40,
          ),
          Container(
            //height: 60,
            alignment: Alignment.center,
            decoration: BoxDecoration(
              color: Colors.blue,

```



```

        borderRadius: BorderRadius.all(
          Radius.circular(5),
        ),
      ),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: <Widget>[
          FlatButton(
            child: Text(
              "Login",
              style: TextStyle(
                fontWeight: FontWeight.bold,
                color: Colors.white,
                fontSize: 20,
              ),
              textAlign: TextAlign.center,
            ),
            onPressed: loginUser,
          ),
        ],
      ),
    ),
    SizedBox(
      height: 10,
    ),
  ],
),
);
}

// ignore: missing_return
Future<FirebaseUser> loginUser() async {
  try {
    await FirebaseAuth.instance
      .signInWithEmailAndPassword(email: _email, password:
_password);
    Navigator.pushNamed(context, '/pagina2');
  } catch (e) {
    popupEnvioMensagem(context);
  }
}

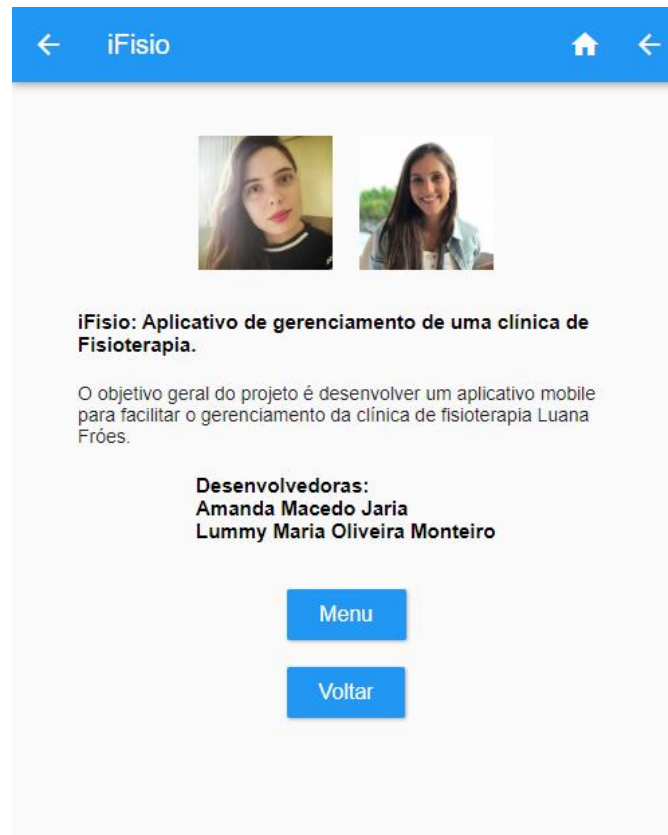
popupEnvioMensagem(BuildContext context) {
  // configura o button
  Widget okButton = FlatButton(
    child: Text("OK"),
    onPressed: () {
      Navigator.of(context).pop(context);
    },
  );
  // configura o AlertDialog
  AlertDialog alerta = AlertDialog(
    title: Text(""),
    content: Text("Usuário não cadastrado ou senha/email incorreto!"
+
    "\ne-mail: " +
    _email +
    "\nsenha: " +
    _password),

```

```
        actions: [
            okButton,
        ],
    );

    // exibe o dialog
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return alerta;
      },
    );
  }
}
```

```
//
// SEGUNDA PÁGINA - EXPLICAÇÃO DO APP
//
```



```
class ExplicacaoPagina extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("iFisio"),
        actions: <Widget>[
          new IconButton(
            icon: new Icon(Icons.home),
            onPressed: () {
              Navigator.pushNamed(context, '/pagina3');
            },
          ),
          new IconButton(
            icon: new Icon(Icons.arrow_back),
            onPressed: () {
              Navigator.pushNamed(context, '/pagina17');
            },
          ),
        ],
      ),
      body: SingleChildScrollView(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: <Widget>[
            Padding(
              padding: EdgeInsets.only(bottom: 40),
            ),
          ],
        ),
      ),
    );
  }
}
```

```

        Row(mainAxisAlignment: MainAxisAlignment.center, children:
[
    Image.network(
        "https://il.rgstatic.net/ii/profile.image/457088460693512-1485989789584_Q512/Amanda_Jaria.jpg",
        fit: BoxFit.fill,
        width: 100.0,
        alignment: Alignment.bottomRight),
    Column(children: <Widget>[
        Padding(
            padding: EdgeInsets.all(10),
        ),
    ]),
    Image.network(
        "https://loop.frontiersin.org/images/profile/380099/203",
        fit: BoxFit.fill,
        width: 100.0,
        alignment: Alignment.bottomRight),
    ]),
    Container(
        //margin: EdgeInsets.only(top: 200),
        margin: const EdgeInsets.only(top: 30, left: 50.0,
right: 50.0),
        child: Center(
            child: Text(
                'iFisio: Aplicativo de gerenciamento de uma
clínica de Fisioterapia.',
                style: TextStyle(
                    fontSize: 15,
                    color: Colors.black,
                    fontFamily: 'Open Sans',
                    fontWeight: FontWeight.bold,
                )),
        ),
    ),
    Container(
        //margin: EdgeInsets.only(top: 200),
        margin: const EdgeInsets.only(top: 20, left: 50.0,
right: 50.0),
        child: Center(
            child: Text(
                'O objetivo geral do projeto é desenvolver um
aplicativo mobile para facilitar o gerenciamento da clínica de
fisioterapia Luana Fróes.'),
        ),
    ),
    Container(
        //margin: EdgeInsets.only(top: 200),
        margin: const EdgeInsets.only(top: 20, left: 50.0,
right: 50.0),
        child: Center(
            child: Text(
                'Desenvolvedoras:' +
                "\n" +
                'Amanda Macedo Jaria' +
                "\n" +
                'Lummy Maria Oliveira Monteiro',
                style: TextStyle(

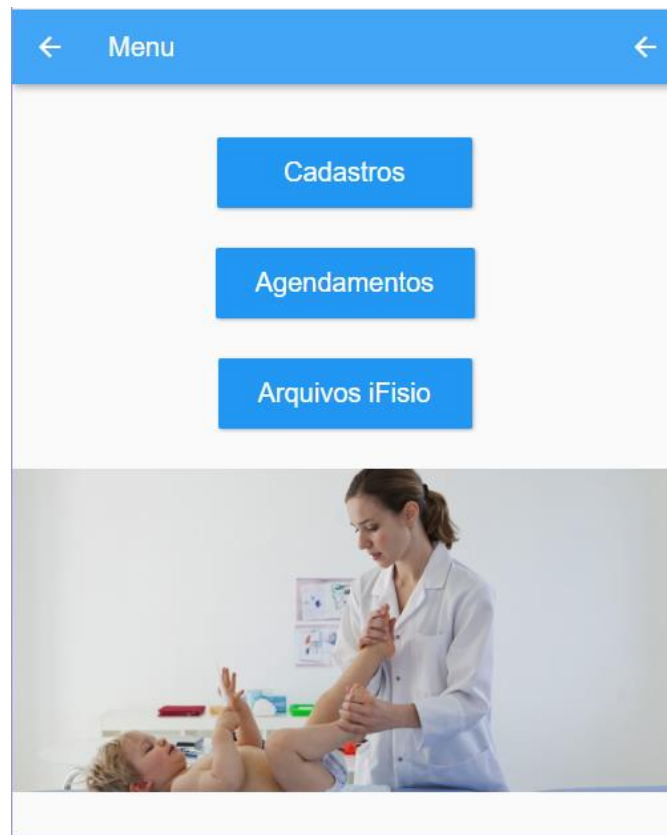
```

```

        fontSize: 15,
        color: Colors.black,
        fontFamily: 'Open Sans',
        fontWeight: FontWeight.bold,
      )),
    ),
  ),
  SizedBox(
    height: 20,
  ),
  Padding(
    padding: const EdgeInsets.symmetric(vertical: 5,
horizontal: 10),
  ),
  RaisedButton(
    padding: const EdgeInsets.symmetric(vertical: 10,
horizontal: 20),
    child: Text(
      " Menu ",
      style: TextStyle(
        color: Colors.white,
        fontSize: 16,
      ),
    ),
    color: Colors.blue[500],
    onPressed: () {
      Navigator.pushNamed(context, '/pagina3');
    },
  ),
  Padding(
    padding: const EdgeInsets.symmetric(vertical: 5,
horizontal: 10),
  ),
  RaisedButton(
    padding: const EdgeInsets.symmetric(vertical: 10,
horizontal: 20),
    child: Text(
      "Voltar",
      style: TextStyle(
        color: Colors.white,
        fontSize: 16,
      ),
    ),
    color: Colors.blue[500],
    onPressed: () {
      Navigator.pushNamed(context, '/pagina1');
    },
  ),
),
],
),
),
);
}
}

```

```
//
// TERCEIRA PÁGINA - MENU PRINCIPAL
//
```



```
class MenuPrincipal extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Menu"),
        actions: <Widget>[
          new IconButton(
            icon: new Icon(Icons.arrow_back),
            onPressed: () {
              Navigator.pushNamed(context, '/pagina2');
            },
          ),
        ],
        backgroundColor: Colors.blue[400]),
      body: Center(
        child: Column(

          children: <Widget>[
            Padding(
              padding: EdgeInsets.only(bottom: 40),
            ),

            RaisedButton(
              padding: const EdgeInsets.symmetric(vertical: 15,
horizontal: 50),
              child: Text(
                "Cadastros",
                style: TextStyle(
```

```

        color: Colors.white,
        fontSize: 20,
      ),
    ),
    color: Colors.blue[500],
    onPressed: () {
      Navigator.pushNamed(context, '/pagina4');
    },
  ),

  Padding(
    padding: const EdgeInsets.symmetric(vertical: 15,
horizontal: 10),
  ),
  RaisedButton(
    padding: const EdgeInsets.symmetric(vertical: 15,
horizontal: 30),
    child: Text(
      "Agendamentos",
      style: TextStyle(
        color: Colors.white,
        fontSize: 20,
      ),
    ),
    color: Colors.blue[500],
    onPressed: () {
      Navigator.pushNamed(context, '/pagina15');
    },
  ),

  Padding(
    padding: const EdgeInsets.symmetric(vertical: 15,
horizontal: 10),
  ),
  RaisedButton(
    padding: const EdgeInsets.symmetric(vertical: 15,
horizontal: 30),
    child: Text(
      "Arquivos iFisio",
      style: TextStyle(
        color: Colors.white,
        fontSize: 20,
      ),
    ),
    color: Colors.blue[500],
    onPressed: () {
      Navigator.pushNamed(context, '/paginal6');
    },
  ),

  Padding(
    padding: EdgeInsets.only(top: 30),
  ),

  Image.network(
    "https://interfisio.com.br/wp-
content/uploads/2017/10/Fotolia_158986386_Subscription_Monthly_M-
1170x568.jpg",
    fit: BoxFit.fill,
    width: 500.0,
  ),

```

```

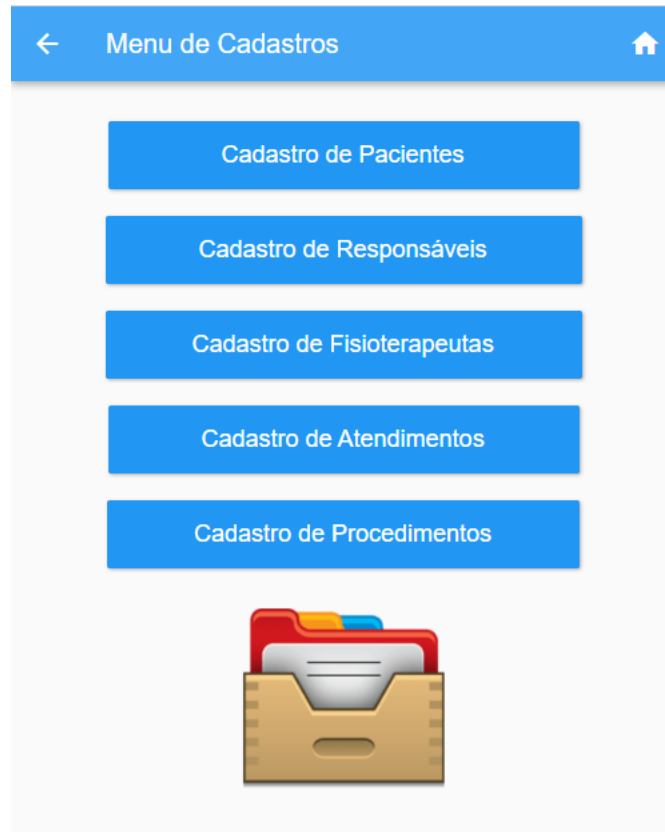
        alignment: Alignment.bottomRight),
      ],
    ),
  );
}
}

```

```

//
// QUARTA PÁGINA - MENU DE CADASTROS
//

```



```

class MenuCadastroPagina extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Menu de Cadastros"),
        actions: <Widget>[
          new IconButton(
            icon: new Icon(Icons.home),
            onPressed: () {
              /*
                Navigator.push(context,
                  MaterialPageRoute(
                    builder: (context) => MenuPrincipal())
                );
              */
              Navigator.pushNamed(context, '/pagina3');
            },
          ),
        ],
      ),
    );
  }
}

```



```

        backgroundColor: Colors.blue[400]),
body: SingleChildScrollView(
  child: Center(
    child: Column(
      //mainAxisAlignment: MainAxisAlignment.center,

  children: <Widget>[
    Padding(
      padding: EdgeInsets.only(top: 30),
    ),
    RaisedButton(
      padding:
        const EdgeInsets.symmetric(vertical: 15,
horizontal: 85),
      child: Text(
        "Cadastro de Pacientes",
        style: TextStyle(
          color: Colors.white,
          fontSize: 18,
        ),
      ),
      color: Colors.blue[500],
      onPressed: () {
        /*
        Navigator.push(context,
          MaterialPageRoute(
            builder: (context) => CadastroPacientes())
        );
        */
        Navigator.pushNamed(context, '/pagina5');
      },
    ),
    Padding(
      padding: EdgeInsets.only(top: 20),
    ),
    RaisedButton(
      padding:
        const EdgeInsets.symmetric(vertical: 15,
horizontal: 70),
      child: Text(
        "Cadastro de Responsáveis",
        style: TextStyle(
          color: Colors.white,
          fontSize: 18,
        ),
      ),
      color: Colors.blue[500],
      onPressed: () {
        /*
        Navigator.push(context,
          MaterialPageRoute(
            builder: (context) => CadastroResponsavel())
        );
        */
        Navigator.pushNamed(context, '/pagina7');
      },
    ),
    Padding(
      padding: EdgeInsets.only(top: 20),
    ),
    RaisedButton(

```

```

        padding:
            const EdgeInsets.symmetric(vertical: 15,
horizontal: 65),
        child: Text(
            "Cadastro de Fisioterapeutas",
            style: TextStyle(
                color: Colors.white,
                fontSize: 18,
            ),
        ),
        color: Colors.blue[500],
        onPressed: () {
            /*
Navigator.push(context,
    MaterialPageRoute(
        builder: (context) => CadastroFisioterapeuta()
    );
*/
            Navigator.pushNamed(context, '/pagina9');
        },
    ),
    Padding(
        padding: EdgeInsets.only(top: 20),
    ),
    RaisedButton(
        padding:
            const EdgeInsets.symmetric(vertical: 15,
horizontal: 70),
        child: Text(
            "Cadastro de Atendimentos",
            style: TextStyle(
                color: Colors.white,
                fontSize: 18,
            ),
        ),
        color: Colors.blue[500],
        onPressed: () {
            /*
Navigator.push(context,
    MaterialPageRoute(
        builder: (context) => CadastroAtendimento()
    );
*/
            Navigator.pushNamed(context, '/pagina11');
        },
    ),
    Padding(
        padding: EdgeInsets.only(top: 20),
    ),
    RaisedButton(
        padding:
            const EdgeInsets.symmetric(vertical: 15,
horizontal: 65),
        child: Text(
            "Cadastro de Procedimentos",
            style: TextStyle(
                color: Colors.white,
                fontSize: 18,
            ),
        ),
        color: Colors.blue[500],
    ),

```



```

class CadastroPacientes extends StatefulWidget {
  @override
  _CadastroPacientesState createState() => _CadastroPacientesState();
}

class _CadastroPacientesState extends State<CadastroPacientes> {
  GlobalKey<FormState> _formkey = GlobalKey<FormState>();

  TextEditingController txtCodigo = TextEditingController();
  TextEditingController txtNome = TextEditingController();
  TextEditingController txtDataNascimento = TextEditingController();
  TextEditingController txtCEP = TextEditingController();
  TextEditingController txtNumCasa = TextEditingController();
  TextEditingController txtComplemento = TextEditingController();
  TextEditingController txtTelContato = TextEditingController();
  TextEditingController txtDiagnostico = TextEditingController();
  TextEditingController txtInfoExtra = TextEditingController();
  TextEditingController txtResponsavel = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Cadastro de Paciente"),
        actions: <Widget>[
          new IconButton(icon: new Icon(Icons.backspace), onPressed:
_reset),
          new IconButton(
            icon: new Icon(Icons.arrow_back),
            onPressed: () {
              /*
              Navigator.push(context,
                MaterialPageRoute(
                  builder: (context) => MenuCadastroPagina()
                );
              */
              Navigator.pushNamed(context, '/pagina4');
            },
          ),
          new IconButton(
            icon: new Icon(Icons.home),
            onPressed: () {
              /*
              Navigator.push(context,
                MaterialPageRoute(
                  builder: (context) => MenuPrincipal()
                );
              */
              Navigator.pushNamed(context, '/pagina3');
            },
          ),
        ],
        backgroundColor: Colors.blue[400]),
    body: SingleChildScrollView(
      padding: const EdgeInsets.all(20.0),
      child: Form(
        key: _formkey,
        child: Column(crossAxisAlignment:
CrossAxisAlignment.stretch,
          //mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: <Widget>[

```

```

        Image.network(
            "https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AANd9GcR3ThzzjvfkQW7zOErUfmC_ouJ6pDWayG
8NNvI6wMg9jmS5mUYc&usqp=CAU",
            width: 80,
            height: 80),
        campoTexto("Código do paciente:", txtCodigo),
        campoTexto("Nome do paciente:", txtNome),
        campoTexto("Data de Nascimento:", txtDataNascimento),
        campoTexto("CEP:", txtCEP),
        campoTexto("Número da residência:", txtNumCasa),
        campoTexto("Complemento:", txtComplemento),
        campoTexto("Telefone de Contato:", txtTelContato),
        campoTexto("Diagnóstico do paciente:",
txtDiagnostico),
        campoTexto("Informações Adicionais:", txtInfoExtra),
        campoTexto("Responsável:", txtResponsavel),
        botaoCadastrar(context),
        botaoLimpar(context),
        Padding(
            padding: const EdgeInsets.only(top: 20),
        ),
        RaisedButton(
            child: Text(
                "Visualizar Pacientes Cadastrados",
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 18,
                ),
            ),
            color: Colors.grey[500],
            onPressed: () {
                /*
                Navigator.push(context,
                    MaterialPageRoute(
                        builder: (context) =>
PacientesCadastradosPagina())
                );
                */
                Navigator.pushNamed(context, '/pagina6');
            },
        ),
    ]),
),
),
backgroundColor: Colors.white,
);
}

campoTexto(rotulo, controle) {
    return Container(
        padding: const EdgeInsets.symmetric(vertical: 5),
        child: TextFormField(
            keyboardType: TextInputType.text,
            style: TextStyle(color: Colors.black54, fontSize: 20),
            decoration: InputDecoration(
                labelText: rotulo,
                labelStyle: TextStyle(
                    color: Colors.blue[700],
                    fontSize: 15,
                ),
            ),
        ),
    ),
}

```

```

    ),
    controller: controle,
    validator: (value) {
        return (value.isEmpty) ? "Informe o valor" : null;
    },
    ),
    );
}

```

```

botaoCadastrar(BuildContext context) {
    return Container(
        padding: const EdgeInsets.only(top: 20),
        child: RaisedButton(
            child: Text(
                "Finalizar Cadastro",
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 18,
                ),
            ),
            color: Colors.grey[500],
            onPressed: () {
                if (_formkey.currentState.validate()) {
                    _cadastrar();
                }
            },
        ));
}

```

```

botaoLimpar(BuildContext context) {
    return Container(
        padding: const EdgeInsets.only(top: 20),
        child: RaisedButton(
            child: Text(
                "Limpar",
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 18,
                ),
            ),
            color: Colors.grey[500],
            onPressed: () {
                {
                    _reset();
                }
            },
        ));
}

```

```

void _cadastrar() {
    setState(() {
        o_paciente.codigoPessoa = int.parse(txtCodigo.text);
        o_paciente.nome = (txtNome.text);
        o_paciente.dataNascimento = (txtDataNascimento.text);
        o_paciente.cep = int.parse(txtCEP.text);
        o_paciente.numeroCasa = int.parse(txtNumCasa.text);
        o_paciente.complemento = (txtComplemento.text);
        o_paciente.telContato = int.parse(txtTelContato.text);
        o_paciente.diagnostico = (txtDiagnostico.text);
        o_paciente.informacaoExtra = (txtInfoExtra.text);
        o_paciente.responsavel = (txtResponsavel.text);
    });
}

```

```

        _cadastroDialog();
    });
}

void _cadastroDialog() {
    showDialog(
      context: context,
      builder: (BuildContext context) {
        // retorna um objeto do tipo Dialog
        return AlertDialog(
          //title: new Text("Alert Dialog titulo"),
          content: new Text("Cadastro efetuado com sucesso!"),
          actions: <Widget>[
            // define os botões na base do dialogo
            new FlatButton(
              child: new Text("OK"),
              onPressed: () {
                Navigator.pop(context);
              },
            ),
          ],
        );
      },
    );
}

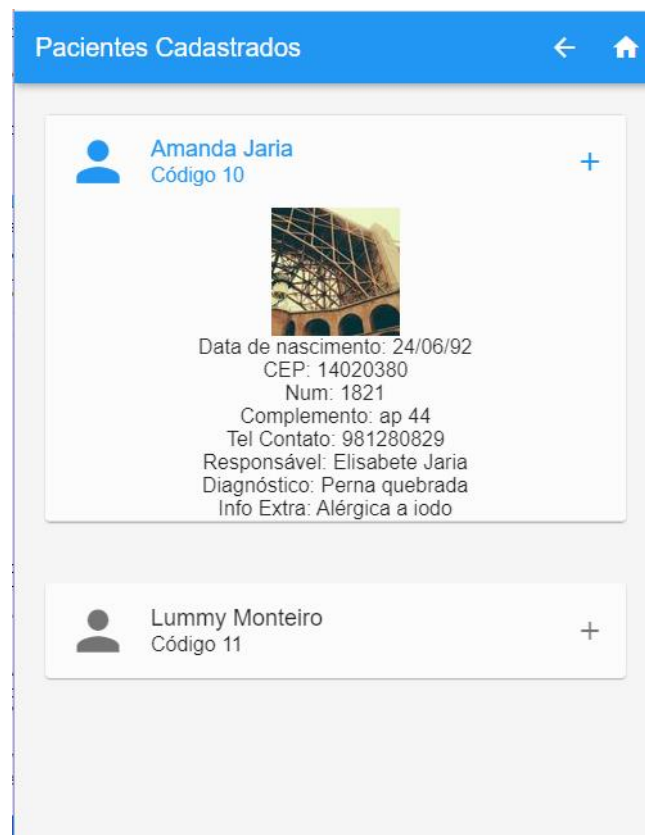
void _voltar() {
    Navigator.pop(context);
}

void _home() {
    Navigator.pop(context);
}

void _reset() {
    setState(() {
      txtCodigo.text = "";
      txtNome.text = "";
      txtDataNascimento.text = "";
      txtCEP.text = "";
      txtNumCasa.text = "";
      txtComplemento.text = "";
      txtTelContato.text = "";
      txtDiagnostico.text = "";
      txtInfoExtra.text = "";
      txtResponsavel.text = "";
    });
}
}

```

```
//
// SEXTA PÁGINA - listview de paciente
//
```



```
class PacientesCadastradosPagina extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(title: Text("Pacientes Cadastrados"), actions:
<Widget>[
          new IconButton(
            icon: new Icon(Icons.arrow_back),
            onPressed: () {
              /*
                Navigator.push(context,
                  MaterialPageRoute(
                    builder: (context) => MenuPrincipal()
                  );
              */
              Navigator.pushNamed(context, '/pagina3');
            },
          ),
          new IconButton(
            icon: new Icon(Icons.home),
            onPressed: () {
              /*
                Navigator.push(context,
                  MaterialPageRoute(
                    builder: (context) => MenuPrincipal()
                  );
              */
            },
          ),
        ],
      ),
    );
  }
}
```



```

        );
        */
        Navigator.pushNamed(context, '/pagina3');
    },
    ),
  ]),
  body: PacientesCadastrados(),
),
);
}
}

class PacientesCadastrados extends StatefulWidget {
  @override
  _PacientesCadastradosPaginaState createState() =>
    _PacientesCadastradosPaginaState();
}

class _PacientesCadastradosPaginaState extends
State<PacientesCadastrados> {
  //Lista Dinâmica de Itens contendo: Ícone, título e subtítulo
  List<Item> lista = [];

  //Aparência dos itens do ListView
  Widget _itemLista(context, index) {
    return Container(
      width: MediaQuery.of(context).size.width * 0.40,
      //height: MediaQuery.of(context).size.height*0.40,
      padding: EdgeInsets.all(20),
      child: Card(
        color: Colors.white70,
        child: Column(
          children: <Widget>[
            //SizedBox(height: 15,),
            ExpansionTile(
              leading: Icon(
                lista[index].icone,
                size: 50,
              ),
              title: Text(
                lista[index].nome,
                style: TextStyle(
                  fontSize: 18,
                ),
              ),
              subtitle: Text(
                "Código " + lista[index].codigoPessoa.toString(),
                style: TextStyle(
                  fontSize: 16,
                ),
              ),
            ),
            trailing: Icon(Icons.add),
            children: <Widget>[
              Image.network(
                "https://picsum.photos/seed/${index +
1}/100/100"),
              Text(
                "Data de nascimento: " +
                lista[index].dataNascimento.toString(),
                style: TextStyle(fontSize: 16),
              ),
            ],
          ],
        ),
      ),
    );
  }
}

```

```

        Text(
            "CEP: " + lista[index].cep.toString(),
            style: TextStyle(fontSize: 16),
        ),
        Text(
            "Num: " + lista[index].numeroCasa.toString(),
            style: TextStyle(fontSize: 16),
        ),
        Text(
            "Complemento: " +
lista[index].complemento.toString(),
            style: TextStyle(fontSize: 16),
        ),
        Text(
            "Tel Contato: " +
lista[index].telContato.toString(),
            style: TextStyle(fontSize: 16),
        ),
        Text(
            "Responsável: " +
lista[index].responsavel.toString(),
            style: TextStyle(fontSize: 16),
        ),
        Text(
            "Diagnóstico: " +
lista[index].diagnostico.toString(),
            style: TextStyle(fontSize: 16),
        ),
        Text(
            "Info Extra: " +
lista[index].informacaoExtra.toString(),
            style: TextStyle(fontSize: 16),
        ),
    ],
),
),
),
);
}

@override
Widget build(BuildContext context) {
  //gerar dados para Lista Din,mica
  lista.add(Item(
    Icons.person,
    "Amanda Jaria",
    10,
    Image.network(
      "https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AAND9GcR3ThzzjvfKQW7zOErUfmc_ouJ6pDWayG
8NNvI6wMg9jms5mUYc&usqp=CAU",
      width: 80,
      height: 80),
    "24/06/92",
    14020380,
    1821,
    "ap 44",
    981280829,
    "Perna quebrada",
    "Alérgica a iodo",
  ));
}

```

```

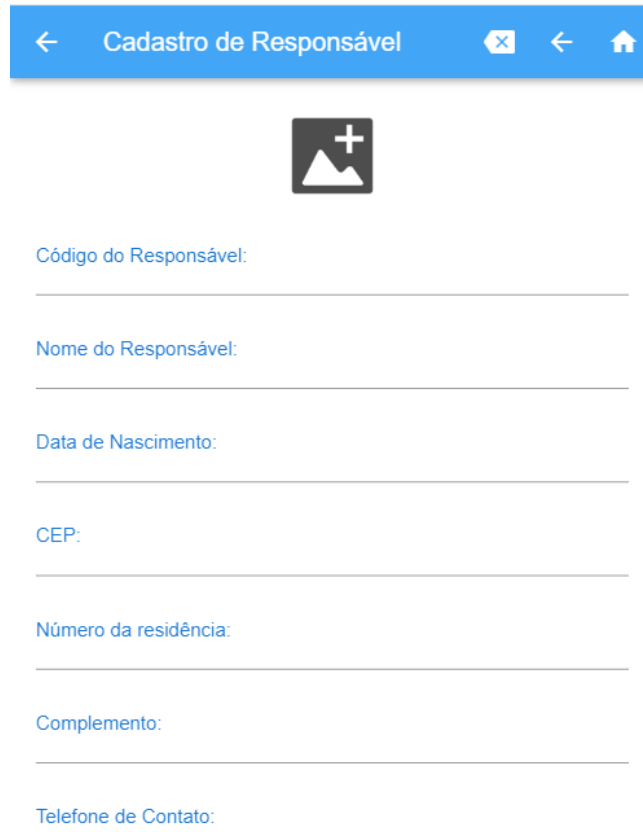
        "Elisabete Jaria"));
    lista.add(Item(
        Icons.person,
        "Lummy Monteiro",
        11,
        Image.network(
            "https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AAND9GcR3ThzzjvfKQW7zOErUfmC_ouJ6pDWayG
8NNvI6wMg9jmS5mUYc&usqp=CAU",
            width: 80,
            height: 80),
        "24/06/92",
        14020380,
        1821,
        "ap 44",
        981280829,
        "Perna quebrada",
        "Alérgica a iodo",
        "Elisabete Jaria"));

    return Container(
        color: Colors.grey[100],
        height: MediaQuery.of(context).size.height * 1.0,
        child: ListView.builder(
            scrollDirection: Axis.vertical,
            itemCount: lista.length, // total de itens da lista
            itemBuilder: _itemLista, // especificar a aparência dos itens
            da lista
        ),
    );
}

void _home() {}

```

```
//
// SETIMA PÁGINA - CADASTRO DE RESPONSÁVEIS
//
```



```
class CadastroResponsavel extends StatefulWidget {
  @override
  _CadastroResponsavelState createState() =>
    _CadastroResponsavelState();
}

class _CadastroResponsavelState extends State<CadastroResponsavel> {
  GlobalKey<FormState> _formkey = GlobalKey<FormState>();

  TextEditingController txtCodigo = TextEditingController();
  TextEditingController txtNome = TextEditingController();
  TextEditingController txtDataNascimento = TextEditingController();
  TextEditingController txtCEP = TextEditingController();
  TextEditingController txtNumCasa = TextEditingController();
  TextEditingController txtComplemento = TextEditingController();
  TextEditingController txtTelContato = TextEditingController();
  //fim heranca
  TextEditingController txtTutorado = TextEditingController();
  TextEditingController txtEmail = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Cadastro de Responsável"),
        actions: <Widget>[
```

```

        new IconButton(icon: new Icon(Icons.backspace), onPressed:
_reset),
        new IconButton(
            icon: new Icon(Icons.arrow_back),
            onPressed: () {
                /*
                Navigator.push(context,
                    MaterialPageRoute(
                        builder: (context) => MenuCadastroPagina())
                );
                */
                Navigator.pushNamed(context, '/pagina4');
            },
        ),
        new IconButton(
            icon: new Icon(Icons.home),
            onPressed: () {
                /*
                Navigator.push(context,
                    MaterialPageRoute(
                        builder: (context) => MenuPrincipal())
                );
                */
                Navigator.pushNamed(context, '/pagina3');
            },
        ),
    ],
    backgroundColor: Colors.blue[400]),
    body: SingleChildScrollView(
        padding: const EdgeInsets.all(20.0),
        child: Form(
            key: _formkey,
            child: Column(crossAxisAlignment:
CrossAxisAlignment.stretch,
                //mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget>[
                    Image.network(
                        "https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AANd9GcR3ThzzjvfKQW7zOErUfmC_ouJ6pDWayG
8NNvI6wMg9jmS5mUYc&usqp=CAU",
                        width: 80,
                        height: 80),
                    campoTexto("Código do Responsável:", txtCodigo),
                    campoTexto("Nome do Responsável:", txtNome),
                    campoTexto("Data de Nascimento:", txtDataNascimento),
                    campoTexto("CEP:", txtCEP),
                    campoTexto("Número da residência:", txtNumCasa),
                    campoTexto("Complemento:", txtComplemento),
                    campoTexto("Telefone de Contato:", txtTelContato),
                    //fim da heranca
                    campoTexto("Tutorado:", txtTutorado),
                    campoTexto("Email:", txtEmail),

                    botaoCadastrar(context),
                    botaoLimpar(context),

                    Padding(
                        padding: const EdgeInsets.only(top: 20),
                    ),
                    RaisedButton(
                        child: Text(

```

```

        "Visualizar Responsáveis Cadastrados",
        style: TextStyle(
          color: Colors.white,
          fontSize: 18,
        ),
      ),
    ),
    color: Colors.grey[500],
    onPressed: () {
      /*
        Navigator.push(context,
          MaterialPageRoute(
            builder: (context) =>
ResponsaveisCadastradosPagina())
          );
      */
      Navigator.pushNamed(context, '/pagina8');
    },
  ),
]),
),
),
backgroundColor: Colors.white,
);
}

campoTexto(rotulo, controle) {
  return Container(
    padding: const EdgeInsets.symmetric(vertical: 5),
    child: TextFormField(
      keyboardType: TextInputType.text,
      style: TextStyle(color: Colors.black54, fontSize: 20),
      decoration: InputDecoration(
        labelText: rotulo,
        labelStyle: TextStyle(
          color: Colors.blue[700],
          fontSize: 15,
        ),
      ),
    ),
    controller: controle,
    validator: (value) {
      return (value.isEmpty) ? "Informe o valor" : null;
    },
  ),
);
}

botaoCadastrar(BuildContext context) {
  return Container(
    padding: const EdgeInsets.only(top: 20),
    child: RaisedButton(
      child: Text(
        "Cadastrar",
        style: TextStyle(
          color: Colors.white,
          fontSize: 18,
        ),
      ),
    ),
    color: Colors.grey[500],
    onPressed: () {
      if (_formkey.currentState.validate()) {
        _cadastrar();
      }
    },
  );
}

```

```

        }
    },
));
}

botaoLimpar(BuildContext context) {
    return Container(
        padding: const EdgeInsets.only(top: 20),
        child: RaisedButton(
            child: Text(
                "Limpar",
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 18,
                ),
            ),
            color: Colors.grey[500],
            onPressed: () {
                {
                    _reset();
                }
            },
        ),
    );
}

void _cadastrar() {
    setState(() {
        o_responsavel.codigoPessoa = int.parse(txtCodigo.text);
        o_responsavel.nome = txtNome.text;
        o_responsavel.dataNascimento = txtDataNascimento.text;
        o_responsavel.cep = int.parse(txtCEP.text);
        o_responsavel.numeroCasa = int.parse(txtNumCasa.text);
        o_responsavel.complemento = txtComplemento.text;
        o_responsavel.telContato = int.parse(txtTelContato.text);
        //fim da heranca
        o_responsavel.tutorado = txtTutorado.text;
        o_responsavel.email = txtEmail.text;
        _cadastroDialog();
    });
}

void _cadastroDialog() {
    showDialog(
        context: context,
        builder: (BuildContext context) {
            // retorna um objeto do tipo Dialog
            return AlertDialog(
                //title: new Text("Alert Dialog titulo"),
                content: new Text("Cadastro efetuado com sucesso!"),
                actions: <Widget>[
                    // define os botões na base do dialogo
                    new FlatButton(
                        child: new Text("OK"),
                        onPressed: () {
                            Navigator.pop(context);
                        },
                    ),
                ],
            );
        },
    );
}

```

```

    }

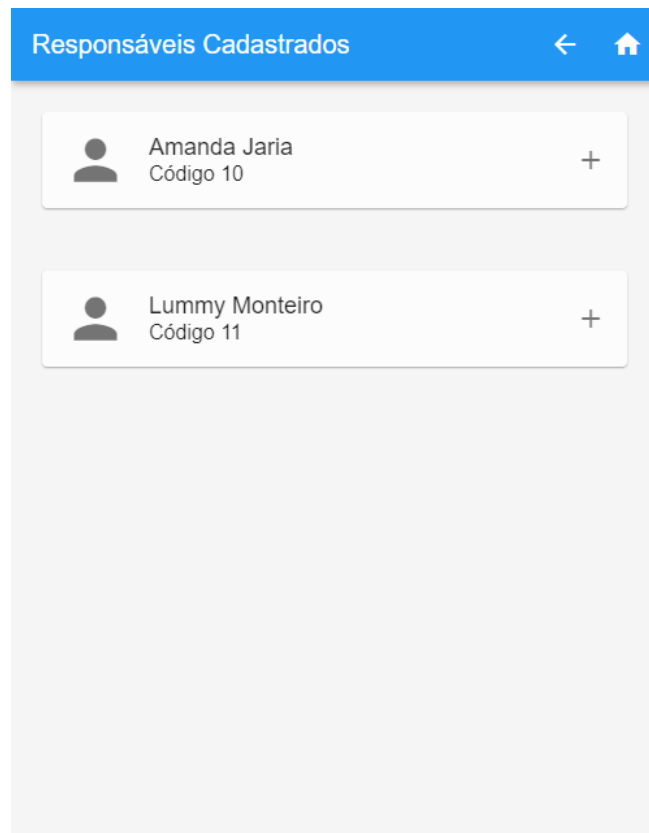
    void _voltar() {
        Navigator.pop(context);
    }

    void _home() {
        Navigator.pop(context);
    }

    void _reset() {
        setState(() {
            txtCodigo.text = "";
            txtNome.text = "";
            txtDataNascimento.text = "";
            txtCEP.text = "";
            txtNumCasa.text = "";
            txtComplemento.text = "";
            txtTelContato.text = "";
            //fim da heranca
            txtTutorado.text = "";
            txtEmail.text = "";
        });
    }
}

//
// OITAVA PÁGINA - LISTVIEW DE RESPONSÁVEIS
//

```




```

class ResponsaveisCadastradosPagina extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar:
          AppBar(title: Text("Responsáveis Cadastrados"), actions:
<Widget>[
            new IconButton(
              icon: new Icon(Icons.arrow_back),
              onPressed: () {
                /*
                  Navigator.push(context,
                    MaterialPageRoute(
                      builder: (context) => MenuPrincipal())
                );
                */
                Navigator.pushNamed(context, '/pagina3');
              },
            ),
            new IconButton(
              icon: new Icon(Icons.home),
              onPressed: () {
                /*
                  Navigator.push(context,
                    MaterialPageRoute(
                      builder: (context) => MenuPrincipal())
                );
                */
                Navigator.pushNamed(context, '/pagina3');
              },
            ),
          ]),
        body: ResponsaveisCadastrados(),
      ),
    );
  }
}

```

```

class ResponsaveisCadastrados extends StatefulWidget {
  @override
  _ResponsaveisCadastradosState createState() =>
    _ResponsaveisCadastradosState();
}

```

```

class _ResponsaveisCadastradosState extends
State<ResponsaveisCadastrados> {
  //Lista Din,mica de Itens contendo: Ícone, título e subtítulo
  List<ItemResponsaveis> lista = [];

  //Aparência dos itens do ListView
  Widget _itemLista(context, index) {
    return Container(
      width: MediaQuery.of(context).size.width * 0.40,
      //height: MediaQuery.of(context).size.height*0.40,
      padding: EdgeInsets.all(20),
      child: Card(
        color: Colors.white70,
        child: Column(
          children: <Widget>[

```

```

//SizedBox(height: 15,),
ExpansionTile(
  leading: Icon(
    lista[index].icone,
    size: 50,
  ),
  title: Text(
    lista[index].nome,
    style: TextStyle(
      fontSize: 18,
    ),
  ),
  subtitle: Text(
    "Código " + lista[index].codigoPessoa.toString(),
    style: TextStyle(
      fontSize: 16,
    ),
  ),
  trailing: Icon(Icons.add),
  children: <Widget>[
    Image.network(
      "https://picsum.photos/seed/${index +
1}/100/100"),
    Text(
      "Data de nascimento: " +
        lista[index].dataNascimento.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "CEP: " + lista[index].cep.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "Num: " + lista[index].numeroCasa.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "Complemento: " +
lista[index].complemento.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "Tel Contato: " +
lista[index].telContato.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "Email: " + lista[index].email.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "Tutorado: " + lista[index].tutorado.toString(),
      style: TextStyle(fontSize: 16),
    ),
  ],
),
],
),
),
);
}

```

```

@override
Widget build(BuildContext context) {
  //gerar dados para Lista Dinamica
  lista.add(ItemResponsaveis(
    Icons.person,
    "Amanda Jaria",
    10,
    Image.network(
      "https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AAND9GcR3ThzzjvfKQW7zOErUfmC_ouJ6pDWayG
8NNvI6wMg9jmS5mUYc&usqp=CAU",
      width: 80,
      height: 80),
    "24/06/92",
    14020380,
    1821,
    "ap 44",
    981280829,
    "Lummy Monteiro",
    "amanda@gmail.com"));
  lista.add(ItemResponsaveis(
    Icons.person,
    "Lummy Monteiro",
    11,
    Image.network(
      "https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AAND9GcR3ThzzjvfKQW7zOErUfmC_ouJ6pDWayG
8NNvI6wMg9jmS5mUYc&usqp=CAU",
      width: 80,
      height: 80),
    "24/06/92",
    14020380,
    1821,
    "ap 44",
    981280829,
    "Amanda Jaria",
    "lummy@gmail.com"));

  return Container(
    color: Colors.grey[100],
    height: MediaQuery.of(context).size.height * 1.0,
    child: ListView.builder(
      scrollDirection: Axis.vertical,
      itemCount: lista.length, // total de itens da lista
      itemBuilder: _itemLista, // especificar a apar ncia dos itens
da lista
    ),
  );
}

```

```
//
// NONA PÁGINA - CADASTRO DE FISIOTERAPEUTAS
//
```

```
class CadastroFisioterapeuta extends StatefulWidget {
  @override
  _CadastroFisioterapeutaState createState() =>
    _CadastroFisioterapeutaState();
}

class _CadastroFisioterapeutaState extends
  State<CadastroFisioterapeuta> {
  GlobalKey<FormState> _formkey = GlobalKey<FormState>();

  TextEditingController txtCodigo = TextEditingController();
  TextEditingController txtNome = TextEditingController();
  TextEditingController txtDataNascimento = TextEditingController();
  TextEditingController txtCEP = TextEditingController();
  TextEditingController txtNumCasa = TextEditingController();
  TextEditingController txtComplemento = TextEditingController();
  TextEditingController txtTelContato = TextEditingController();
  //fim heranca
  TextEditingController txtCredito = TextEditingController();
  TextEditingController txtEmail = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Cadastro de Fisioterapeuta"),
        actions: <Widget>[
```

```

        new IconButton(icon: new Icon(Icons.backspace), onPressed:
_reset),
        new IconButton(
            icon: new Icon(Icons.arrow_back),
            onPressed: () {
                /*
                Navigator.push(context,
                    MaterialPageRoute(
                        builder: (context) => MenuPrincipal())
                );
                */
                Navigator.pushNamed(context, '/pagina3');
            },
        ),
        new IconButton(
            icon: new Icon(Icons.home),
            onPressed: () {
                /*
                Navigator.push(context,
                    MaterialPageRoute(
                        builder: (context) => MenuPrincipal())
                );
                */
                Navigator.pushNamed(context, '/pagina3');
            },
        ),
    ],
    backgroundColor: Colors.blue[400]),
    body: SingleChildScrollView(
        padding: const EdgeInsets.all(20.0),
        child: Form(
            key: _formkey,
            child: Column(crossAxisAlignment:
CrossAxisAlignment.stretch,
                //mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget>[
                    Image.network(
                        "https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AANd9GcR3ThzzjvfKQW7zOErUfmC_ouJ6pDWayG
8NNvI6wMg9jmS5mUYc&usqp=CAU",
                        width: 80,
                        height: 80),
                    campoTexto("Código do fisioterapeuta:", txtCodigo),
                    campoTexto("Nome do fisioterapeuta:", txtNome),
                    campoTexto("Data de Nascimento:", txtDataNascimento),
                    campoTexto("CEP:", txtCEP),
                    campoTexto("Número da residência:", txtNumCasa),
                    campoTexto("Complemento:", txtComplemento),
                    campoTexto("Telefone de Contato:", txtTelContato),
                    //fim da heranca
                    campoTexto("CREFITO:", txtCrefito),
                    campoTexto("email:", txtEmail),

                    botaoCadastrar(context),
                    botaoLimpar(context),

                    Padding(
                        padding: const EdgeInsets.only(top: 20),
                    ),
                    RaisedButton(
                        child: Text(

```

```

        "Visualizar Fisioterapeutas Cadastrados",
        style: TextStyle(
          color: Colors.white,
          fontSize: 18,
        ),
      ),
    ),
    color: Colors.grey[500],
    onPressed: () {
      /*
        Navigator.push(context,
          MaterialPageRoute(
            builder: (context) =>
FisioterapeutasCadastradosPagina())
        );
      */
      Navigator.pushNamed(context, '/pagina10');
    },
  ),
],
),
),
backgroundColor: Colors.white,
);
}

```

```

campoTexto(rotulo, controle) {
  return Container(
    padding: const EdgeInsets.symmetric(vertical: 5),
    child: TextFormField(
      keyboardType: TextInputType.text,
      style: TextStyle(color: Colors.black54, fontSize: 20),
      decoration: InputDecoration(
        labelText: rotulo,
        labelStyle: TextStyle(
          color: Colors.blue[700],
          fontSize: 15,
        ),
      ),
    ),
    controller: controle,
    validator: (value) {
      return (value.isEmpty) ? "Informe o valor" : null;
    },
  ),
);
}

```

```

botaoCadastrar(BuildContext context) {
  return Container(
    padding: const EdgeInsets.only(top: 20),
    child: RaisedButton(
      child: Text(
        "Cadastrar",
        style: TextStyle(
          color: Colors.white,
          fontSize: 18,
        ),
      ),
    ),
    color: Colors.grey[500],
    onPressed: () {
      if (_formkey.currentState.validate()) {
        _cadastrar();
      }
    },
  );
}

```

```

        }
    },
));
}

botaoLimpar(BuildContext context) {
    return Container(
        padding: const EdgeInsets.only(top: 20),
        child: RaisedButton(
            child: Text(
                "Limpar",
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 18,
                ),
            ),
            color: Colors.grey[500],
            onPressed: () {
                {
                    _reset();
                }
            },
        ),
    );
}

void _cadastrar() {
    setState(() {
        o_fisioterapeuta.codigoPessoa = int.parse(txtCodigo.text);
        o_fisioterapeuta.nome = (txtNome.text);
        o_fisioterapeuta.dataNascimento = (txtDataNascimento.text);
        o_fisioterapeuta.cep = int.parse(txtCEP.text);
        o_fisioterapeuta.numeroCasa = int.parse(txtNumCasa.text);
        o_fisioterapeuta.complemento = (txtComplemento.text);
        o_fisioterapeuta.telContato = int.parse(txtTelContato.text);
        //fim da heranca
        o_fisioterapeuta.crefito = (txtCrefito.text);
        o_fisioterapeuta.email = (txtEmail.text);
        _cadastroDialog();
    });
}

void _cadastroDialog() {
    showDialog(
        context: context,
        builder: (BuildContext context) {
            // retorna um objeto do tipo Dialog
            return AlertDialog(
                //title: new Text("Alert Dialog titulo"),
                content: new Text("Cadastro efetuado com sucesso!"),
                actions: <Widget>[
                    // define os botões na base do dialogo
                    new FlatButton(
                        child: new Text("OK"),
                        onPressed: () {
                            Navigator.pop(context);
                        },
                    ),
                ],
            );
        },
    );
}

```

```

    }

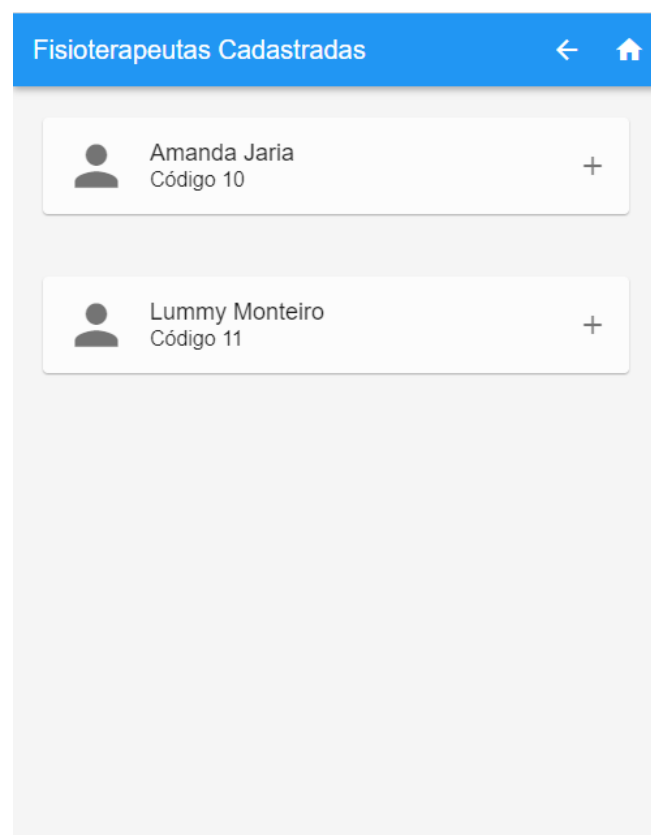
    void _voltar() {
        Navigator.pop(context);
    }

    void _home() {
        Navigator.pop(context);
    }

    void _reset() {
        setState(() {
            txtCodigo.text = "";
            txtNome.text = "";
            txtDataNascimento.text = "";
            txtCEP.text = "";
            txtNumCasa.text = "";
            txtComplemento.text = "";
            txtTelContato.text = "";
            //fim da heranca
            txtCredito.text = "";
            txtEmail.text = "";
        });
    }
}

//
// DECIMA PÁGINA - LISTVIEW DE FISIOTERAPEUTAS
//

```




```

class FisioterapeutasCadastradosPagina extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: Text("Fisioterapeutas Cadastradas"),
          actions: <Widget>[
            new IconButton(
              icon: new Icon(Icons.arrow_back),
              onPressed: () {
                /*
                Navigator.push(context,
                  MaterialPageRoute(
                    builder: (context) => MenuPrincipal())
                );
                */
                Navigator.pushNamed(context, '/pagina3');
              },
            ),
            new IconButton(
              icon: new Icon(Icons.home),
              onPressed: () {
                /*
                Navigator.push(context,
                  MaterialPageRoute(
                    builder: (context) => MenuPrincipal())
                );
                */
                Navigator.pushNamed(context, '/pagina3');
              },
            ),
          ],
        body: FisioterapeutasCadastrados(),
      ),
    );
  }
}

```

```

class FisioterapeutasCadastrados extends StatefulWidget {
  @override
  _FisioterapeutasCadastradosState createState() =>
    _FisioterapeutasCadastradosState();
}

```

```

class _FisioterapeutasCadastradosState
  extends State<FisioterapeutasCadastrados> {
  //Lista Din,mica de Itens contendo: Ícone, título e subtítulo
  List<ItemFisioterapeutas> lista = [];

  //Aparência dos itens do ListView
  Widget _itemLista(context, index) {
    return Container(
      width: MediaQuery.of(context).size.width * 0.40,
      //height: MediaQuery.of(context).size.height*0.40,
      padding: EdgeInsets.all(20),
      child: Card(
        color: Colors.white70,
        child: Column(
          children: <Widget>[

```

```

//SizedBox(height: 15,),
ExpansionTile(
  leading: Icon(
    lista[index].icone,
    size: 50,
  ),
  title: Text(
    lista[index].nome,
    style: TextStyle(
      fontSize: 18,
    ),
  ),
  subtitle: Text(
    "Código " + lista[index].codigoPessoa.toString(),
    style: TextStyle(
      fontSize: 16,
    ),
  ),
  trailing: Icon(Icons.add),
  children: <Widget>[
    Image.network(
      "https://picsum.photos/seed/${index +
1}/100/100"),
    Text(
      "Data de nascimento: " +
        lista[index].dataNascimento.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "CEP: " + lista[index].cep.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "Num: " + lista[index].numeroCasa.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "Complemento: " +
lista[index].complemento.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "Tel Contato: " +
lista[index].telContato.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "Email: " + lista[index].email.toString(),
      style: TextStyle(fontSize: 16),
    ),
    Text(
      "CREFITO: " + lista[index].crefito.toString(),
      style: TextStyle(fontSize: 16),
    ),
  ],
),
],
),
),
);
}

```

```

@Override
Widget build(BuildContext context) {
  //gerar dados para Lista Din,mica
  lista.add(ItemFisioterapeutas(
    Icons.person,
    "Amanda Jaria",
    10,
    Image.network(
      "https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AAND9GcR3ThzzjvfKQW7zOErUfmC_ouJ6pDWayG
8NNvI6wMg9jmS5mUYc&usqp=CAU",
      width: 80,
      height: 80),
    "24/06/92",
    14020380,
    1821,
    "ap 44",
    981280829,
    "F323FS",
    "amanda@gmail.com"));
  lista.add(ItemFisioterapeutas(
    Icons.person,
    "Lummy Monteiro",
    11,
    Image.network(
      "https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AAND9GcR3ThzzjvfKQW7zOErUfmC_ouJ6pDWayG
8NNvI6wMg9jmS5mUYc&usqp=CAU",
      width: 80,
      height: 80),
    "24/06/92",
    14020380,
    1821,
    "ap 44",
    981280829,
    "K76T4S",
    "lummy@gmail.com"));

  return Container(
    color: Colors.grey[100],
    height: MediaQuery.of(context).size.height * 1.0,
    child: ListView.builder(
      scrollDirection: Axis.vertical,
      itemCount: lista.length, // total de itens da lista
      itemBuilder: _itemLista, // especificar a aparência dos itens
da lista
    ),
  );
}
}

```

```
//  
// DECIMA PRIMEIRA PÁGINA - CADASTRO DE ATENDIMENTO  
//
```



Atendimentos

Codigo Atendimento

Data Atendimento

Procedimentos Realizados

Salvar

Menu

Atendimentos

```
class CadastroAtendimento extends StatefulWidget {  
  @override  
  _CadastroAtendimentoState createState() =>  
  _CadastroAtendimentoState();  
}  
  
class _CadastroAtendimentoState extends State<CadastroAtendimento> {  
  //GlobalKey<FormState> _formkey = GlobalKey<FormState>();  
  //controles para os campos de texto  
  TextEditingController txtCodigoAtendimento =  
  TextEditingController();  
  TextEditingController txtDataAtendimento = TextEditingController();  
  TextEditingController txtProcedimentosRealizados =  
  TextEditingController();  
  
  //instância do Firebase  
  var db = Firestore.instance;  
  
  //retornar dados do documento a partir do idDocumento  
  void getDocumento(String idDocumento) async {  
    //Recuperar o documento no Firestore  
    DocumentSnapshot doc =  
      await  
db.collection("atendimentos").document(idDocumento).get();  
  
    setState(() {
```



```

    ),
  ),
  SizedBox(
    height: 30,
  ),
  //CAMPO PROCEDIMENTOS
  TextField(
    controller: txtProcedimentosRealizados,
    maxLines: 5,
    style: TextStyle(
      color: Colors.teal, fontWeight:
FontWeight.w300),
    decoration: InputDecoration(
      labelText: "Procedimentos Realizados",
      hintText: "Escreva os procedimentos
realizados...",
      border: OutlineInputBorder(),
    ),
  ),
  SizedBox(
    height: 30,
  ),
  //BOTÕES
  Column(
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: <Widget>[
      Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          SizedBox(
            width: 300,
            child: RaisedButton(
              color: Colors.grey[500],
              child: Text("Salvar",
                style: TextStyle(
                  color: Colors.white, fontSize:
20)),
              onPressed: () {
                //
                // Inserir ou Atualizar
                //
                if (idDocumento == null) {
                  inserir(
                    context,
                    AtendimentoF(
                      idDocumento,
                      txtCodigoAtendimento.text,
                      txtDataAtendimento.text,
                      txtProcedimentosRealizados.text));
                } else {
                  atualizar(
                    context,
                    AtendimentoF(
                      idDocumento,
                      txtCodigoAtendimento.text,
                      txtDataAtendimento.text,
                      txtProcedimentosRealizados.text));
                }
              },
            ),
          ],
        ),
      ),
    ],
  ),

```

```

    ),
    SizedBox(
      width: 20,
    ),
    SizedBox(
      width: 300,
      child: RaisedButton(
        color: Colors.grey[500],
        child: Text("Menu",
          style: TextStyle(
            color: Colors.white, fontSize:
20)),
        onPressed: () {
          Navigator.pushNamed(context,
            '/pagina4');
        },
      ),
    ),
    SizedBox(
      width: 20,
    ),
    SizedBox(
      width: 300,
      child: RaisedButton(
        color: Colors.grey[500],
        child: Text("Atendimentos",
          style: TextStyle(
            color: Colors.white, fontSize:
20)),
        onPressed: () {
          Navigator.pushNamed(context,
            '/pagina12');
        },
      ),
    ),
  ],
),
],
),
],
)))
}
//
// ATUALIZAR
//

void atualizar(BuildContext context, AtendimentoF atendimento) async
{
  await
db.collection("atendimentos").document(atendimento.id).updateData({
  "codigoAtendimento": atendimento.codigoAtendimento,
  "dataAtendimento": atendimento.dataAtendimento,
  "procedimentosRealizados": atendimento.procedimentosRealizados
});
  Navigator.pop(context);
}

//
// INSERIR
//

```

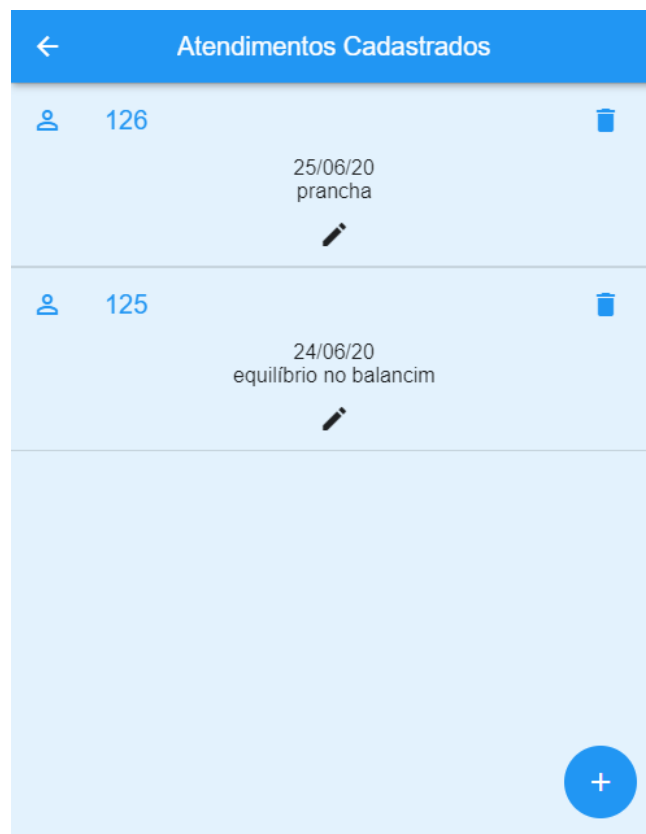
```

void inserir(BuildContext context, AtendimentoF atendimento) async {
  await db.collection("atendimentos").add({
    "codigoAtendimento": atendimento.codigoAtendimento,
    "dataAtendimento": atendimento.dataAtendimento,
    "procedimentosRealizados": atendimento.procedimentosRealizados
  });
  Navigator.pop(context);
}

}

//
// DECIMA SEGUNDA PÁGINA - LISTA DE ATENDIMENTOS
//

```



```

class AtendimentosCadastrados extends StatefulWidget {
  @override
  _AtendimentosCadastradosState createState() =>
    _AtendimentosCadastradosState();
}

class _AtendimentosCadastradosState extends
  State<AtendimentosCadastrados> {
  //Conexão Flutter+Firebase
  final db = Firestore.instance;
  final String colecao = "atendimentos";

  //Lista dinâmica para manipulação dos dados
  List<AtendimentoF> lista = List();

  //Stream para "ouvir" o Firebase
  StreamSubscription<QuerySnapshot> listen;

```



```

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

  //cancelar o listen, caso a coleção esteja vazia.
  listen?.cancel();

  //retornar dados da coleção e inserir na lista dinâmica
  listen = db.collection(colecao).snapshots().listen((res) {
    setState(() {
      lista = res.documents
        .map((doc) => AtendimentoF.fromMap(doc.data,
doc.documentID))
        .toList();
    });
  });
}

@override
void dispose() {
  listen?.cancel();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Atendimentos Cadastrados"),
      centerTitle: true,
      backgroundColor: Colors.blue,
    ),
    body: StreamBuilder<QuerySnapshot>(

      //fonte de dados
      stream: db.collection(colecao).snapshots(),

      //exibição dos dados
      builder: (context, snapshot) {
        switch (snapshot.connectionState) {
          case ConnectionState.none:
          case ConnectionState.waiting:
            return Center(child: CircularProgressIndicator());
          default:
            List<DocumentSnapshot> docs = snapshot.data.documents;
            return ListView.builder(
              itemCount: docs.length,
              itemBuilder: (context, index) {
                return ExpansionTile(
                  leading: Icon(Icons.person_outline),
                  title: Text(
                    lista[index].codigoAtendimento ?? '',
                    style: TextStyle(fontSize: 20),
                  ),
                  //subtitle: Text("(16) 3333-5678"),
                  trailing: IconButton(
                    icon: Icon(Icons.delete),
                    onPressed: () {
                      db
                        .collection(colecao)

```

```

        .document(lista[index].id)
        .delete();
    }),

    children: <Widget>[
      Text(lista[index].dataAtendimento ?? '',
        style: TextStyle(fontSize: 16)),
      Text(lista[index].procedimentosRealizados ??
'',
        style: TextStyle(fontSize: 16)),
      new IconButton(
        icon: new Icon(Icons.edit),
        onPressed: () {
          Navigator.pushNamed(context,
'/pagina11',
            arguments: lista[index].id);
        },
      ),
    ],
  );
});

}
)),
floatingActionButton: FloatingActionButton(
  foregroundColor: Colors.white,
  backgroundColor: Colors.blue,
  elevation: 0,
  child: Icon(Icons.add),
  onPressed: () {
    Navigator.pushNamed(context, '/pagina11', arguments: null);
  },
),
backgroundColor: Colors.blue[50],
);
}
}

```

```
//  
// DECIMA TERCEIRA PÁGINA - CADASTRO DE PROCEDIMENTOS  
//
```

Procedimentos

Codigo Procedimento

Descricao Procedimento

Número de Séries

Número de Repetições

Salvar

Menu

Procedimentos

```
class CadastroProcedimento extends StatefulWidget {  
  @override  
  _CadastroProcedimentoState createState() =>  
  _CadastroProcedimentoState();  
}  
  
class _CadastroProcedimentoState extends State<CadastroProcedimento> {  
  //GlobalKey<FormState> _formkey = GlobalKey<FormState>();  
  //controles para os campos de texto  
  TextEditingController txtCodigoProcedimento =  
  TextEditingController();  
  TextEditingController txtDescricaoProcedimentos =  
  TextEditingController();  
  TextEditingController txtNumRepeticoes = TextEditingController();  
  TextEditingController txtNumSeries = TextEditingController();  
  
  //instância do Firebase  
  var db = Firestore.instance;  
  
  //retornar dados do documento a partir do idDocumento  
  void getDocumento(String idDocumento) async {  
    //Recuperar o documento no Firestore  
    DocumentSnapshot doc =  
      await  
    db.collection("procedimentos").document(idDocumento).get();  
  }
```

```

        setState(() {
            txtCodigoProcedimento.text = doc.data["CodigoProcedimento"];
            txtDescricaoProcedimentos.text =
doc.data["DescricaoProcedimentos"];
            txtNumRepeticoes.text = doc.data["NumRepeticoes"];
            txtNumSeries.text = doc.data["NumSeries"];
        });
    }

    @override
    Widget build(BuildContext context) {
        // RECUPERAR o ID do Documento
        final String idDocumento =
ModalRoute.of(context).settings.arguments;

        if (idDocumento != null) {
            if (txtCodigoProcedimento.text == "" &&
                txtDescricaoProcedimentos.text == "" &&
                txtNumSeries.text == "" &&
                txtNumRepeticoes.text == "") {
                getDocumento(idDocumento);
            }
        }
        return Scaffold(
            body: SingleChildScrollView(
                padding: EdgeInsets.all(50),
                child: Column(
                    children: [
                        Text(
                            "Procedimentos",
                            style: TextStyle(
                                color: Colors.blue[500],
                                fontWeight: FontWeight.bold,
                                fontSize: 24),
                        ),
                        SizedBox(
                            height: 30,
                        ),
                        //CAMPO CODIGO
                        TextField(
                            controller: txtCodigoProcedimento,
                            style: TextStyle(
                                color: Colors.teal, fontWeight:
FontWeight.w300),
                            decoration: InputDecoration(
                                border: OutlineInputBorder(),
                                //prefixIcon: Icon(Icons.person_outline),
                                labelText: "Codigo Procedimento",
                            ),
                        ),
                        SizedBox(
                            height: 20,
                        ),
                        //CAMPO DESCRICAO
                        TextField(
                            controller: txtDescricaoProcedimentos,
                            style: TextStyle(
                                color: Colors.teal, fontWeight:
FontWeight.w300),

```

```

        decoration: InputDecoration(
          border: OutlineInputBorder(),
          //prefixIcon: Icon(Icons.person_outline),
          labelText: "Descricao Procedimento",
        ),
      ),
      SizedBox(
        height: 20,
      ),

      //CAMPO SERIES
      TextField(
        controller: txtNumSeries,
        style: TextStyle(
          color: Colors.teal, fontWeight:
FontWeight.w300),
        decoration: InputDecoration(
          //prefixIcon: Icon(Icons.email),
          labelText: "Número de Séries",
          border: OutlineInputBorder(),
        ),
      ),
      SizedBox(
        height: 30,
      ),
      //CAMPO REPETICOES
      TextField(
        controller: txtNumRepeticoes,
        //maxLines: 5,
        style: TextStyle(
          color: Colors.teal, fontWeight:
FontWeight.w300),
        decoration: InputDecoration(
          labelText: "Número de Repetições",
          //hintText: "Escreva os procedimentos
realizados...",
          border: OutlineInputBorder(),
        ),
      ),
      SizedBox(
        height: 30,
      ),
      //BOTÕES
      Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: <Widget>[
          Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              SizedBox(
                width: 300,
                child: RaisedButton(
                  color: Colors.grey[500],
                  child: Text("Salvar",
                    style: TextStyle(
                      color: Colors.white, fontSize:
20)),
                  onPressed: () {
                    //
                    // Inserir ou Atualizar
                    //

```

[illegible]

```

        ),
    ],
    ))) ;

}
//
// ATUALIZAR
//

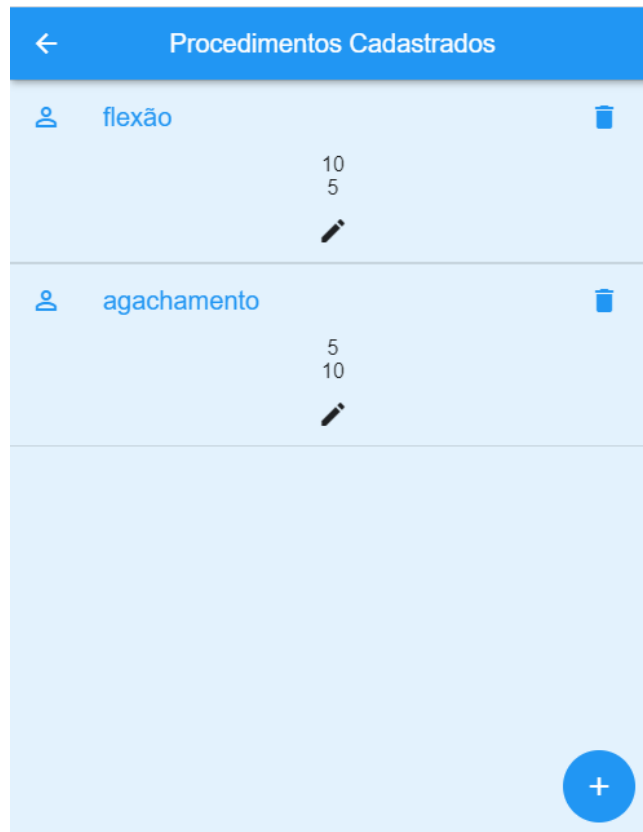
void atualizar(BuildContext context, ProcedimentosF procedimento)
async {
    await
    db.collection("procedimentos").document(procedimento.id).updateData({
        "codigoProcedimento": procedimento.codigoProcedimento,
        "descricaoProcedimentos": procedimento.descricaoProcedimentos,
        "numRepeticoes": procedimento.numRepeticoes,
        "numSeries": procedimento.numSeries
    });
    Navigator.pop(context);
}

//
// INSERIR
//
void inserir(BuildContext context, ProcedimentosF procedimento)
async {
    await db.collection("procedimentos").add({
        "codigoProcedimento": procedimento.codigoProcedimento,
        "descricaoProcedimentos": procedimento.descricaoProcedimentos,
        "numRepeticoes": procedimento.numRepeticoes,
        "numSeries": procedimento.numSeries
    });
    Navigator.pop(context);
}

}

//
// DECIMA QUARTA PÁGINA - LISTVIEW DE PROCEDIMENTOS
//

```



```
class ProcedimentosCadastrados extends StatefulWidget {
  @override
  _ProcedimentosCadastradosState createState() =>
    _ProcedimentosCadastradosState();
}

class _ProcedimentosCadastradosState extends
State<ProcedimentosCadastrados> {
  //Conexão Flutter+Firebase
  final db = Firestore.instance;
  final String colecao = "procedimentos";

  //Lista dinâmica para manipulação dos dados
  List<ProcedimentosF> lista = List();

  //Stream para "ouvir" o Firebase
  StreamSubscription<QuerySnapshot> listen;

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

    //cancelar o listen, caso a coleção esteja vazia.
    listen?.cancel();

    //retornar dados da coleção e inserir na lista dinâmica
    listen = db.collection(colecao).snapshots().listen((res) {
      setState(() {
        lista = res.documents
          .map((doc) => ProcedimentosF.fromMap(doc.data,
doc.documentID))
          .toList();
      });
    });
  }
}
```



```

    });
  });
}

@override
void dispose() {
  listen?.cancel();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Procedimentos Cadastrados"),
      centerTitle: true,
      backgroundColor: Colors.blue,
    ),
    body: StreamBuilder<QuerySnapshot>(
      //fonte de dados
      stream: db.collection(colecao).snapshots(),

      //exibição dos dados
      builder: (context, snapshot) {
        switch (snapshot.connectionState) {
          case ConnectionState.none:
          case ConnectionState.waiting:
            return Center(child: CircularProgressIndicator());
          default:
            List<DocumentSnapshot> docs = snapshot.data.documents;
            return ListView.builder(
              itemCount: docs.length,
              itemBuilder: (context, index) {
                return ExpansionTile(
                  leading: Icon(Icons.person_outline),
                  title: Text(
                    lista[index].descricaoProcedimentos ?? '',
                    style: TextStyle(fontSize: 20),
                  ),
                  //subtitle: Text("(16) 3333-5678"),
                  trailing: IconButton(
                    icon: Icon(Icons.delete),
                    onPressed: () {
                      db
                        .collection(colecao)
                        .document(lista[index].id)
                        .delete();
                    },
                  ),

                  children: <Widget>[
                    //Text(lista[index].codigoProcedimento ??
                    //
                    // style: TextStyle(fontSize: 16)),
                    Text(lista[index].numSeries ?? '',
                      style: TextStyle(fontSize: 16)),
                    Text(lista[index].numRepeticoes ?? '',
                      style: TextStyle(fontSize: 16)),
                    new IconButton(
                      icon: new Icon(Icons.edit),
                      onPressed: () {

```

```

Navigator.pushNamed(context,
'/pagina13',
arguments: lista[index].id);
    },
  ),
],
);
});
}
)),
floatingActionButton: FloatingActionButton(
  foregroundColor: Colors.white,
  backgroundColor: Colors.blue,
  elevation: 0,
  child: Icon(Icons.add),
  onPressed: () {
    Navigator.pushNamed(context, '/pagina13', arguments: null);
  },
),
backgroundColor: Colors.blue[50],
);
}
}

//
// DECIMA QUINTA PÁGINA - REALIZAR UM AGENDAMENTO
//

```

←

Agendar uma Consulta

✕

←

🏠

Back

+

Nome do Paciente:

Nome do Responsável:

Data da Consulta:

Horário da consulta:

Local da consulta:

Fisioterapeuta Responsável:

Finalizar Agendamento

```

class AgendamentoPagina extends StatefulWidget {
  @override
  _AgendamentoPaginaState createState() => _AgendamentoPaginaState();
}

class _AgendamentoPaginaState extends State<AgendamentoPagina> {
  GlobalKey<FormState> _formkey = GlobalKey<FormState>();

  TextEditingController txtNomePaciente = TextEditingController();
  TextEditingController txtNomeResponsavel = TextEditingController();
  TextEditingController txtDataAtendimento = TextEditingController();
  TextEditingController txtHoraAtendimento = TextEditingController();
  TextEditingController txtLocalAtendimento = TextEditingController();
  TextEditingController txtFisioResponsavel = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Agendar uma Consulta"),
        actions: <Widget>[
          new IconButton(icon: new Icon(Icons.backspace), onPressed:
_reset),
          new IconButton(
            icon: new Icon(Icons.arrow_back),
            onPressed: () {
              /*
              Navigator.push(context,
                MaterialPageRoute(
                  builder: (context) => MenuCadastroPagina())
                );
              */
              Navigator.pushNamed(context, '/pagina4');
            },
          ),
          new IconButton(
            icon: new Icon(Icons.home),
            onPressed: () {
              /*
              Navigator.push(context,
                MaterialPageRoute(
                  builder: (context) => MenuPrincipal())
                );
              */
              Navigator.pushNamed(context, '/pagina3');
            },
          ),
        ],
        backgroundColor: Colors.blue[400]),
      body: SingleChildScrollView(
        padding: const EdgeInsets.all(20.0),
        child: Form(
          key: _formkey,
          child: Column(crossAxisAlignment:
CrossAxisAlignment.stretch,
            //mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: <Widget>[
              Image.network(
                "https://encrypted-
tbn0.gstatic.com/images?q=tbn%3AAND9GcR3ThzzjvfKQW7zOErUfmC_ouJ6pDWayG
8NNvI6wMg9jmS5mUYc&usqp=CAU",

```

```

        width: 80,
        height: 80),
        campoTexto("Nome do Paciente:", txtNomePaciente),
        campoTexto("Nome do Responsável:",
txtNomeResponsavel),
        campoTexto("Data da Consulta:", txtDataAtendimento),
        campoTexto("Horário da consulta:",
txtHoraAtendimento),
        campoTexto("Local da consulta:", txtLocalAtendimento),
        campoTexto("Fisioterapeuta Responsável:",
txtFisioResponsavel),
        botaoAgendar(context),
        botaoLimpar(context),
    ]),
    ),
    ),
    backgroundColor: Colors.white,
);
}

```

```

campoTexto(rotulo, controle) {
  return Container(
    padding: const EdgeInsets.symmetric(vertical: 5),
    child: TextFormField(
      keyboardType: TextInputType.text,
      style: TextStyle(color: Colors.black54, fontSize: 20),
      decoration: InputDecoration(
        labelText: rotulo,
        labelStyle: TextStyle(
          color: Colors.blue[700],
          fontSize: 15,
        ),
      ),
    ),
    controller: controle,
    validator: (value) {
      return (value.isEmpty) ? "Complete o campo vazio" : null;
    },
  ),
);
}

```

//Gerar um resumo do agendamento

```

botaoAgendar(BuildContext context) {
  return Container(
    padding: const EdgeInsets.only(top: 20),
    child: RaisedButton(
      child: Text(
        "Finalizar Agendamento",
        style: TextStyle(
          color: Colors.white,
          fontSize: 18,
        ),
      ),
    ),
    color: Colors.grey[500],
    onPressed: () {
      if (_formkey.currentState.validate()) {
        _agendar();
      }
    },
  ));
}

```

```

}

botaoLimpar(BuildContext context) {
  return Container(
    padding: const EdgeInsets.only(top: 20),
    child: RaisedButton(
      child: Text(
        "Limpar",
        style: TextStyle(
          color: Colors.white,
          fontSize: 18,
        ),
      ),
      color: Colors.grey[500],
      onPressed: () {
        {
          _reset();
        }
      },
    ));
}

void _agendar() {
  setState(() {
    o_agendamento.nomePaciente = (txtNomePaciente.text);
    o_agendamento.nomeResponsavel = (txtNomeResponsavel.text);
    o_agendamento.dataConsulta = (txtDataAtendimento.text);
    o_agendamento.horaConsulta = (txtHoraAtendimento.text);
    o_agendamento.localConsulta = (txtLocalAtendimento.text);
    o_agendamento.fisioResponsavel = (txtFisioResponsavel.text);
    _cadastroDialog();
  });
}

void _cadastroDialog() {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      // retorna um objeto do tipo Dialog
      return AlertDialog(
        //title: new Text("Alert Dialog titulo"),
        title: new Text("Agendamento efetuado com sucesso!"),
        content: Text("Paciente: " +
          txtNomePaciente.text +
          "\n" +
          "Responsável: " +
          txtNomeResponsavel.text +
          "\n" +
          "Data do agendamento: " +
          txtDataAtendimento.text +
          "\n" +
          "Horario agendado: " +
          txtHoraAtendimento.text +
          "\n" +
          "Local de atendimento: " +
          txtLocalAtendimento.text +
          "\n" +
          "Fisioterapeuta Responsável: " +
          txtFisioResponsavel.text),

        actions: <Widget>[

```

```

        // define os botões na base do dialogo
        new FlatButton(
            child: new Text("OK"),
            onPressed: () {
                Navigator.pop(context);
            },
        ),
    ],
);
},
);
}

void _voltar() {
    Navigator.pop(context);
}

void _home() {
    Navigator.pop(context);
}

void _reset() {
    setState(() {
        txtNomePaciente.text = "";
        txtNomeResponsavel.text = "";
        txtDataAtendimento.text = "";
        txtHoraAtendimento.text = "";
        txtLocalAtendimento.text = "";
        txtFisioResponsavel.text = "";
    });
}
}

```

```
//
// DECIMA SEXTA PÁGINA - MENU DE LISTAS
//
```



```
class MenuListasPagina extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Arquivos iFisio"),
        actions: <Widget>[
          new IconButton(
            icon: new Icon(Icons.home),
            onPressed: () {
              /*
              Navigator.push(context,
                MaterialPageRoute(
                  builder: (context) => MenuPrincipal())
              );
              */
              Navigator.pushNamed(context, '/pagina3');
            },
          ),
        ],
        backgroundColor: Colors.blue[400]),
      body: SingleChildScrollView(
        child: Center(
          child: Column(
            //mainAxisAlignment: MainAxisAlignment.center,

            children: <Widget>[
              Padding(
```

```

        padding: EdgeInsets.only(top: 30),
      ),
      RaisedButton(
        padding:
          const EdgeInsets.symmetric(vertical: 15,
horizontal: 85),
        child: Text(
          "Pacientes Cadastrados",
          style: TextStyle(
            color: Colors.white,
            fontSize: 18,
          ),
        ),
        color: Colors.blue[500],
        onPressed: () {
          /*
          Navigator.push(context,
            MaterialPageRoute(
              builder: (context) => PacientesCadastrados()
            );
          */
          Navigator.pushNamed(context, '/pagina6');
        },
      ),
      Padding(
        padding: EdgeInsets.only(top: 20),
      ),
      RaisedButton(
        padding:
          const EdgeInsets.symmetric(vertical: 15,
horizontal: 70),
        child: Text(
          "Responsáveis Cadastrados",
          style: TextStyle(
            color: Colors.white,
            fontSize: 18,
          ),
        ),
        color: Colors.blue[500],
        onPressed: () {
          /*
          Navigator.push(context,
            MaterialPageRoute(
              builder: (context) => ResponsaveisCadastrados()
            );
          */
          Navigator.pushNamed(context, '/pagina8');
        },
      ),
      Padding(
        padding: EdgeInsets.only(top: 20),
      ),
      RaisedButton(
        padding:
          const EdgeInsets.symmetric(vertical: 15,
horizontal: 65),
        child: Text(
          "Fisioterapeutas Cadastrados",
          style: TextStyle(
            color: Colors.white,
            fontSize: 18,

```



```

        ),
        color: Colors.blue[500],
        onPressed: () {
          /*
            Navigator.push(context,
              MaterialPageRoute(
                builder: (context) =>
FisioterapeutasCadastradosPagina())
            );
          */
          Navigator.pushNamed(context, '/pagina10');
        },
      ),
      Padding(
        padding: EdgeInsets.only(top: 20),
      ),
      RaisedButton(
        padding:
          const EdgeInsets.symmetric(vertical: 15,
horizontal: 70),
        child: Text(
          "Atendimentos Cadastrados",
          style: TextStyle(
            color: Colors.white,
            fontSize: 18,
          ),
        ),
        color: Colors.blue[500],
        onPressed: () {
          /*
            Navigator.push(context,
              MaterialPageRoute(
                builder: (context) =>
AtendimentoCadastradoPagina())
            );
          */
          Navigator.pushNamed(context, '/pagina12');
        },
      ),
      Padding(
        padding: EdgeInsets.only(top: 20),
      ),
      RaisedButton(
        padding:
          const EdgeInsets.symmetric(vertical: 15,
horizontal: 65),
        child: Text(
          "Procedimentos Cadastrados",
          style: TextStyle(
            color: Colors.white,
            fontSize: 18,
          ),
        ),
        color: Colors.blue[500],
        onPressed: () {
          /*
            Navigator.push(context,
              MaterialPageRoute(
                builder: (context) =>
ProcedimentosCadastrados())
          */

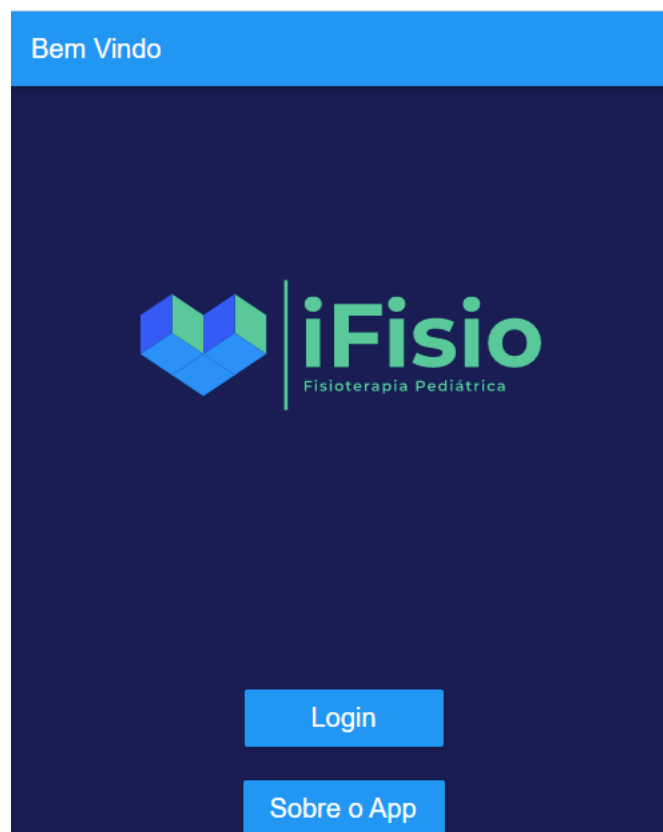
```

```

        );
        */
        Navigator.pushNamed(context, '/pagina14');
    },
),
Padding(
    padding: EdgeInsets.only(top: 30),
),
Image.network(
    "https://www.erasystems.com.br/wp-
content/uploads/2017/06/icone_cadastro-300x162.png",
    fit: BoxFit.fill,
    width: 250.0,
    alignment: Alignment.bottomRight),
],
),
),
),
);
}
}

//
// DECIMA SETIMA PÁGINA - INICIO
//

```



```

class Inicio extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Bem Vindo"), backgroundColor:
Colors.blue),
      backgroundColor: Color(int.parse('0xff1a1d53')),
      body: Center(
        child: SingleChildScrollView(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              Image.network(
                'https://i.ibb.co/drXsjWk/Captura-de-Tela-2020-06-
05-a-s-12-36-37.png',
                fit: BoxFit.cover),
              Padding(
                padding: EdgeInsets.only(top: 10),
              ),
              RaisedButton(
                padding:
                  const EdgeInsets.symmetric(vertical: 10,
horizontal: 50),
                child: Text(
                  "Login",
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 20,
                  ),
                ),
                color: Colors.blue[500],
                onPressed: () {
                  /*
Navigator.push(context,
  MaterialPageRoute(
    builder: (context) => Login())
  );
*/
Navigator.pushNamed(context, '/pagina1');
                },
              ),
              Padding(
                padding:
                  const EdgeInsets.symmetric(vertical: 10,
horizontal: 10),
              ),
              RaisedButton(
                padding:
                  const EdgeInsets.symmetric(vertical: 10,
horizontal: 20),
                child: Text(
                  "Sobre o App",
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 20,
                  ),
                ),
                color: Colors.blue[500],
                onPressed: () {
                  /*
Navigator.push(context,

```

```
MaterialPageRoute(  
  builder: (context) => InformacoesPagina()  
);  
*/  
Navigator.pushNamed(context, '/pagina2');  
},  
),  
],  
),  
),  
),  
);  
}  
}
```