

UNIVERSIDADE PAULISTA – UNIP EaD

Projeto Integrado Multidisciplinar

**Curso Superior de Tecnologia em
Análise e Desenvolvimento de Sistemas**

RODRIGO LANDIM CARNEIRO - RA 1916376

MEDCONNECT - SISTEMA DE TELEATENDIMENTO MÉDICO VIA APP / WEB

Desenvolvimento e criação de interface para plataforma de teleatendimento médico via
APP para a startup MedTech Solutions

UNIP POLO ANA NORTE, BRASÍLIA

2024

RODRIGO LANDIM CARNEIRO - RA 1916376

MEDCONNECT - SISTEMA DE TELEATENDIMENTO MÉDICO VIA APP / WEB

Desenvolvimento e criação de interface para plataforma de teleatendimento médico via APP para a startup MedTech Solutions

Projeto Integrado Multidisciplinar em
Análise e Desenvolvimento de Sistemas

Projeto Integrado Multidisciplinar para obtenção do título de tecnólogo em Análise e Desenvolvimento de Sistemas, apresentado à Universidade Paulista – UNIP EaD.

Orientador(a): Dra. Sandra Bozolan

UNIP POLO ANA NORTE, BRASÍLIA

2024

RESUMO

O objetivo principal deste trabalho é criar um guia abrangente que discorre sobre o desenvolvimento de um sistema de controle utilizando a linguagem C# para acessar uma seção de um banco de dados. Além disso, estão incluídos protótipos de interfaces gráficas de usuário em ASP.Net e Android. Este guia aborda as melhores práticas de adaptação de custos, engenharia de software, economia de mercado, telas de interface e o uso de Linguagem Orientada a Objetos para a implementação de um projeto de sistema. O objetivo final é entregar um pacote de software completo ao cliente final, dentro de um ambiente de intranet que é fornecido por um servidor mainframe interno. O sistema foi projetado para fornecer acesso autorizado apenas a um grupo seletivo, garantindo a preservação de ativos valiosos. Em última análise, este trabalho visa melhorar os métodos de avaliação usados para medir os resultados do projeto.

Palavras-chave: Desenvolvimento, Sistema, Controle, Linguagem C#, Banco de dados, Interfaces, Projeto, Software.

ABSTRACT

The main objective of this work is to create a comprehensive guide that discusses the development of a control system using the C# language to access a section of a database. In addition, prototypes of graphical user interfaces in ASP.Net and Android are included. This guide addresses the best practices for cost adaptation, software engineering, market economics, interface screens and the use of Object Oriented Language for the implementation of a system project. The ultimate goal is to deliver a complete software package to the end customer, within an intranet environment that is provided by an internal mainframe server. The system is designed to only provide authorized access to a select few, ensuring the preservation of valuable assets. Ultimately, this work aims to improve the evaluation methods used to measure project results.

Keywords: Development, System, Control, C# Language, Database, Interfaces, Project, Software.

SUMÁRIO

1. INTRODUÇÃO	04
2. CODIFICAÇÃO	06
2.1. Pessoa.cs	06
2.2. Endereco.cs	06
2.3. Telefone.cs	07
2.4. TipoTelefone.cs	07
2.5. PessoaDAO.cs	07
2.6. Protótipo funcional	13
3. LAYOUT	13
3.1 Layout Asp.NET	13
3.1.1. Telas Responsivas Asp.NET	14
3.1.2. Codificação arquivo _Layout.cshtml usando RAZOR	15
3.1.3. Codificação arquivo index.cshtml usando RAZOR	16
3.2 Layout Android	18
3.2.1. Tela Android	19
3.2.2. Codificação arquivo activity_main.xml	19
4. CONCLUSÃO	22
5. REFERENCIAS BIBLIOGRÁFICAS	23

1. INTRODUÇÃO

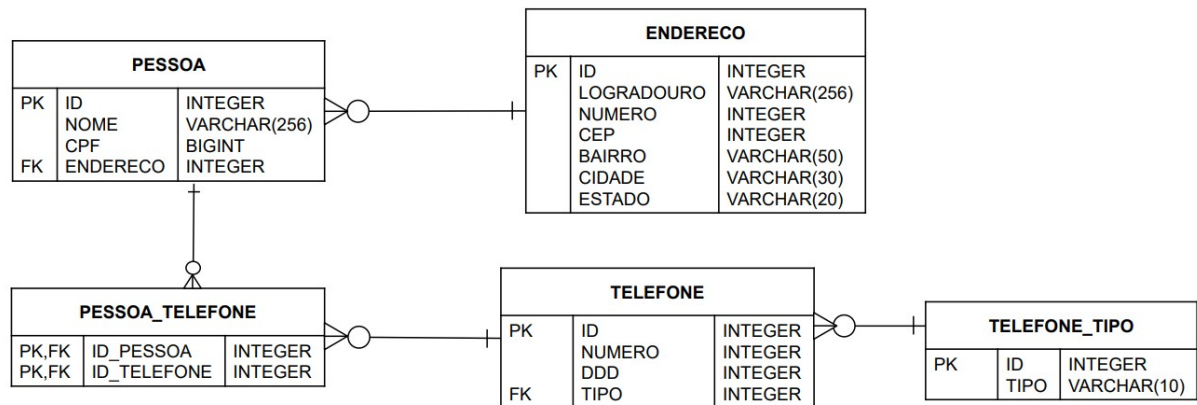
Para cumprir os requisitos do curso PIM, os alunos ou grupos devem mostrar seus conhecimentos de Programação Orientada a Objetos II, Desenvolvimento de Software para a Internet e Tópicos Especiais de Programação Orientada a Objetos, apresentando um mecanismo codificado em C# para acesso ao banco de dados e interface em ASP.Net e Android.

As metas são as seguintes:

- Para fornecer acesso a uma seção específica do banco de dados para o restante do sistema, é necessário criar um mecanismo de acesso ao banco de dados usando a linguagem C#. Este mecanismo será responsável por facilitar o acesso sempre que qualquer parte do sistema o requeira;
- O mecanismo desenvolvido deve permitir fácil acesso ao banco de dados, incluindo o trecho a seguir;
- Foi desenvolvido um protótipo de interface gráfica do usuário em ASP.Net, que possibilite ao usuário interagir com os dados modelados nesta seção específica do banco de dados.
- Foi desenvolvido um protótipo de interface gráfica do usuário Android que permite ao usuário interagir com os dados modelados por esse trecho de banco de dados.

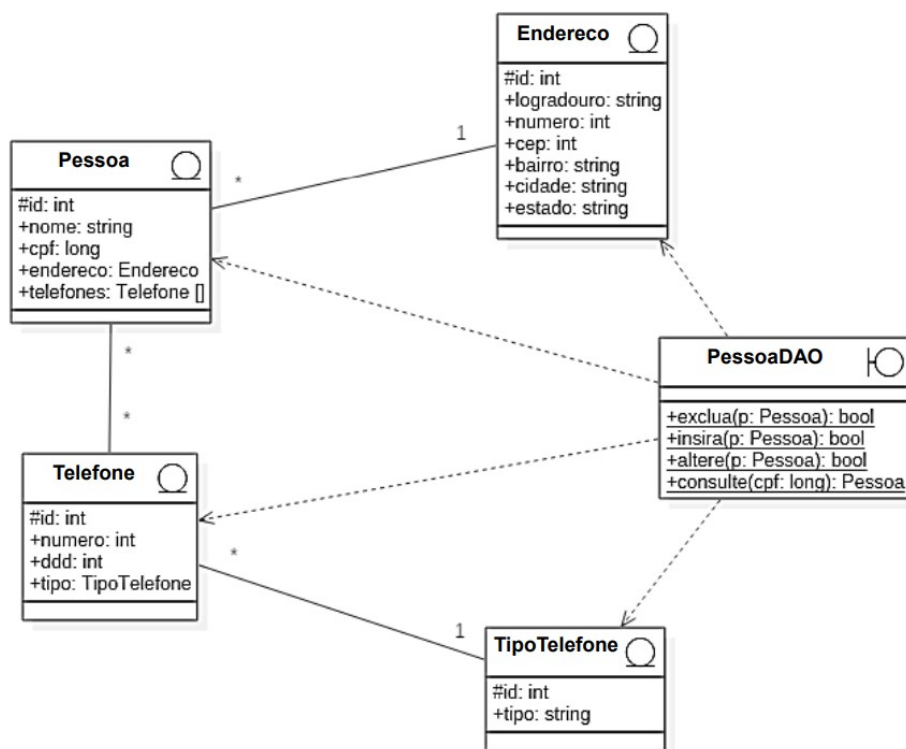
O modelo do trecho do banco de dados que será usado pela equipe no PIM VIII está representado pelo seguinte Diagrama Entidade-Relacionamento (DER):

Figura 1 - Diagrama Entidade-Relacionamento (DER)



Fonte: Manual PIM VIII

Figura 2 – Diagrama de Classe



Fonte: Manual PIM VIII

Para garantir o acesso adequado ao banco de dados, é recomendável que sua equipe crie um mecanismo de acesso que se alinhe com o diagrama de classe fornecido abaixo. Este diagrama descreve quatro entidades: **Pessoa**, **Endereco**, **Telefone** e **TipoTelefone**, todas correspondendo às tabelas primárias no banco de dados. A única exceção é a tabela associativa **PESSOA_TELEFONE**, que não é pertinente ao sistema, mas ainda é relevante para o mecanismo de persistência dos dados do banco de dados e, portanto, deve ser abordada pelo mecanismo de acesso.

A implementação de um mecanismo de acesso ao banco de dados é de responsabilidade da classe **PessoaDAO**. Essa classe oferece quatro métodos estáticos que são acessíveis em todo o sistema e permitem que operações CRUD sejam executadas no banco de dados. É importante observar que os dados que entram e saem da classe **PersonDAO** devem ser filtrados por meio de classes de entidade (como Pessoa, Endereço, Telefone e Tipo de Telefone) ou parâmetros de tipo primitivo (como CPF do tipo long).

2. CODIFICAÇÃO

2.1. Pessoa.cs

```
using System.Collections.Generic;

namespace PessoaDAO
{
    public class Pessoa
    {
        public int id { get; set; }
        public string nome { get; set; }
        public long cpf { get; set; }
        public Endereco endereco { get; set; }
        public IList<Telefone> telefones { get; set; }
    }
}
```

2.2. Endereco.cs

```
namespace PessoaDAO
{
    public class Endereco
    {

```



```

        public int id { get; set; }
        public string logradouro { get; set; }
        public int numero { get; set; }
        public int cep { get; set; }
        public string bairro { get; set; }
        public string cidade { get; set; }
        public string estado { get; set; }
    }
}

```

2.3. Telefone.cs

```

namespace PessoaDAO
{
    public class Telefone
    {
        public int id { get; set; }
        public int numero { get; set; }
        public int ddd { get; set; }
        public TipoTelefone tipo { get; set; }
    }
}

```

2.4. TipoTelefone.cs

```

namespace PessoaDAO
{
    public class TipoTelefone
    {
        public int id { get; set; }
        public string tipo { get; set; }
    }
}

```

2.5. PessoaDAO.cs

```

using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PessoaDAO
{
    public class PessoaDAO
    {
        private string _cnnStr = Properties.Settings.Default.PessoaDAOConnectionString;

        private IList<Telefone> listarTelefonePorPessoa(SqlConnection cnn, int idPessoa)
        {
            var t = new List<Telefone>();
            var queryString = @"
                SELECT
                    T.ID,
                    T.NUMERO,

```

```

        T.DDD,
        TT.ID TIPO_ID,
        TT.TIPO
    FROM PESSOA_TELEFONE PT
    JOIN TELEFONE T ON T.ID = PT.ID_TELEFONE
    JOIN TELEFONE_TIPO TT ON TT.ID = T.TIPO
    WHERE PT.ID_PESSOA = @ID
";
var cmd = new SqlCommand(queryString, cnn);
cmd.Parameters.Add("@ID", System.Data.SqlDbType.Int).Value = idPessoa;
using (var row = cmd.ExecuteReader())
{
    while (row.Read())
    {
        t.Add(new Telefone
        {
            id = row.GetInt32(0),
            numero = row.GetInt32(1),
            ddd = row.GetInt32(2),
            tipo = new TipoTelefone
            {
                id = row.GetInt32(3),
                tipo = row.GetString(4),
            }
        });
    }
}
return t;
}

private Pessoa pegarPessoaPorCpf(SqlConnection cnn, long cpf)
{
    var queryString = @"
    SELECT
        P.ID,
        P.NOME,
        P.CPF,
        P.ENDERECO AS ENDERECO_ID,
        E.LOGRADOURO AS ENDERECO_LOGRADOURO,
        E.NUMERO AS ENDERECO_NUMERO,
        E.CEP ENDERECO_CEP,
        E.BAIRRO ENDERECO_BAIRRO,
        E.CIDADE ENDERECO_CIDADE,
        E.ESTADO ENDERECO_ESTADO
    FROM PESSOA P
    LEFT JOIN ENDERECO E ON E.ID = P.ENDERECO
    WHERE P.CPF = @CPF
";
    var cmd = new SqlCommand(queryString, cnn);
    cmd.Parameters.Add("@CPF", System.Data.SqlDbType.BigInt).Value = cpf;
    using (var row = cmd.ExecuteReader())
    {
        if (row.Read())
        {
            var p = new Pessoa
            {
                id = row.GetInt32(0),
                nome = row.GetString(1),
                cpf = row.GetInt64(2)
            };
            if (!row.IsDBNull(3))
            {
                p.endereco = new Endereco
                {
                    id = row.GetInt32(3),
                    logradouro = row.GetString(4),
                    numero = row.GetInt32(5),
                    cep = row.GetInt32(6),
                }
            }
        }
    }
}

```

```

        bairro = row.GetString(7),
        cidade = row.GetString(8),
        estado = row.GetString(9)
    };
}
return p;
}
}
return null;
}

private int insiraEndereco(SqlConnection cnn, SqlTransaction trans, Endereco endereco)
{
    var queryString = @"
        INSERT INTO ENDERECO (
            LOGRADOURO, NUMERO, CEP, BAIRRO, CIDADE, ESTADO
        ) VALUES (
            @LOGRADOURO, @NUMERO, @CEP, @BAIRRO, @CIDADE, @ESTADO
        )
        SELECT SCOPE_IDENTITY()
    ";
    var cmd = new SqlCommand(queryString, cnn, trans);
    cmd.Parameters.Add("@LOGRADOURO", System.Data.SqlDbType.VarChar, 256).Value =
endereco.logradouro;
    cmd.Parameters.Add("@NUMERO", System.Data.SqlDbType.Int).Value = endereco.numero;
    cmd.Parameters.Add("@CEP", System.Data.SqlDbType.Int).Value = endereco.cep;
    cmd.Parameters.Add("@BAIRRO", System.Data.SqlDbType.VarChar, 50).Value = endereco.bairro;
    cmd.Parameters.Add("@CIDADE", System.Data.SqlDbType.VarChar, 30).Value = endereco.cidade;
    cmd.Parameters.Add("@ESTADO", System.Data.SqlDbType.VarChar, 20).Value = endereco.estado;
    return Convert.ToInt32(cmd.ExecuteScalar());
}

private void insiraTelefone(SqlConnection cnn, SqlTransaction trans, int idPessoa,
IList<Telefone> telefones)
{
    string queryString = null;
    SqlCommand cmd = null;
    int idTelefone = 0, idTipoTelefone = 0;

    foreach (var telefone in telefones)
    {
        idTipoTelefone = 0;
        if (telefone.tipo != null)
        {
            if (telefone.tipo.id > 0)
            {
                idTipoTelefone = telefone.tipo.id;
            }
            else
            {
                queryString = "INSERT INTO TELEFONE_TIPO (TIPO) VALUES (@TIPO) SELECT
SCOPE_IDENTITY()";
                cmd = new SqlCommand(queryString, cnn, trans);
                cmd.Parameters.Add("@TIPO", System.Data.SqlDbType.VarChar, 10).Value =
telefone.tipo.tipo;
                idTipoTelefone = Convert.ToInt32(cmd.ExecuteScalar());
            }
        }
        queryString = @"
            INSERT INTO TELEFONE (
                NUMERO, DDD, TIPO
            ) VALUES (
                @NUMERO, @DDD, @TIPO
            )
            SELECT SCOPE_IDENTITY()
        ";
        cmd = new SqlCommand(queryString, cnn, trans);
        cmd.Parameters.Add("@NUMERO", System.Data.SqlDbType.Int).Value = telefone.numero;

```

```

        cmd.Parameters.Add("@DDD", System.Data.SqlDbType.Int).Value = telefone.ddd;
        cmd.Parameters.Add("@TIPO", System.Data.SqlDbType.Int).Value = idTipoTelefone > 0 ?
(object)idTipoTelefone : DBNull.Value;
        idTelefone = Convert.ToInt32(cmd.ExecuteScalar());

        queryString = @"
            INSERT INTO PESSOA_TELEFONE (
                ID_PESSOA, ID_TELEFONE
            ) VALUES (
                @ID_PESSOA, @ID_TELEFONE
            )
        ";
        cmd = new SqlCommand(queryString, cnn, trans);
        cmd.Parameters.Add("@ID_PESSOA", System.Data.SqlDbType.Int).Value = idPessoa;
        cmd.Parameters.Add("@ID_TELEFONE", System.Data.SqlDbType.Int).Value = idTelefone;
        cmd.ExecuteNonQuery();
    }

    private int insiraPessoa(SqlConnection cnn, SqlTransaction trans, Pessoa p, int idEndereco)
    {
        var queryString = @"
            INSERT INTO PESSOA (
                NOME, CPF, ENDERECO
            ) VALUES (
                @NOME, @CPF, @ENDERECO
            )
            SELECT SCOPE_IDENTITY()
        ";
        var cmd = new SqlCommand(queryString, cnn, trans);
        cmd.Parameters.Add("@NOME", System.Data.SqlDbType.VarChar, 256).Value = p.nome;
        cmd.Parameters.Add("@CPF", System.Data.SqlDbType.BigInt).Value = p.cpf;
        cmd.Parameters.Add("@ENDERECO", System.Data.SqlDbType.Int).Value = idEndereco > 0 ?
(object)idEndereco : DBNull.Value;
        return Convert.ToInt32(cmd.ExecuteScalar());
    }

    private void alteraEndereco(SqlConnection cnn, SqlTransaction trans, Endereco endereco)
    {
        var queryString = @"
            UPDATE ENDERECO SET
                LOGRADOURO = @LOGRADOURO,
                NUMERO = @NUMERO,
                CEP = @CEP,
                BAIRRO = @BAIRRO,
                CIDADE = @CIDADE,
                ESTADO = @ESTADO
            WHERE ID = @ID
        ";
        var cmd = new SqlCommand(queryString, cnn, trans);
        cmd.Parameters.Add("@ID", System.Data.SqlDbType.Int).Value = endereco.id;
        cmd.Parameters.Add("@LOGRADOURO", System.Data.SqlDbType.VarChar, 256).Value =
endereco.logradouro;
        cmd.Parameters.Add("@NUMERO", System.Data.SqlDbType.Int).Value = endereco.numero;
        cmd.Parameters.Add("@CEP", System.Data.SqlDbType.Int).Value = endereco.cep;
        cmd.Parameters.Add("@BAIRRO", System.Data.SqlDbType.VarChar, 50).Value = endereco.bairro;
        cmd.Parameters.Add("@CIDADE", System.Data.SqlDbType.VarChar, 30).Value = endereco.cidade;
        cmd.Parameters.Add("@ESTADO", System.Data.SqlDbType.VarChar, 20).Value = endereco.estado;
        cmd.ExecuteNonQuery();
    }

    private void alterePessoa(SqlConnection cnn, SqlTransaction trans, Pessoa p)
    {
        var queryString = @"
            UPDATE PESSOA SET
                NOME = @NOME,
                CPF = @CPF,
                ENDERECO = @ENDERECO
        ";

```

```

        WHERE ID = @ID
";
var cmd = new SqlCommand(queryString, cnn, trans);
cmd.Parameters.Add("@ID", System.Data.SqlDbType.Int).Value = p.id;
cmd.Parameters.Add("@NOME", System.Data.SqlDbType.VarChar, 256).Value = p.nome;
cmd.Parameters.Add("@CPF", System.Data.SqlDbType.BigInt).Value = p.cpf;
cmd.Parameters.Add("@ENDERECO", System.Data.SqlDbType.Int).Value = (p.endereco != null) ?
(object)p.endereco.id : DBNull.Value;
cmd.ExecuteNonQuery();
}

public Pessoa consulte(long cpf)
{
    using (var cnn = new SqlConnection(_cnnStr))
    {
        cnn.Open();
        var p = pegarPessoaPorCpf(cnn, cpf);
        if (p != null)
        {
            p.telefones = listarTelefonePorPessoa(cnn, p.id);
        }
        return p;
    }
}

public bool insira(Pessoa p)
{
    using (var cnn = new SqlConnection(_cnnStr))
    {
        int idPessoa = 0, idEndereco = 0;
        cnn.Open();
        var trans = cnn.BeginTransaction();
        try
        {
            if (p.endereco != null)
            {
                idEndereco = insiraEndereco(cnn, trans, p.endereco);
            }

            idPessoa = insiraPessoa(cnn, trans, p, idEndereco);

            insiraTelefone(cnn, trans, idPessoa, p.telefones);

            trans.Commit();

            return true;
        }
        catch (Exception ex)
        {
            trans.Rollback();
            throw ex;
        }
    }
}

public bool exclua(Pessoa p)
{
    string queryString = null;
    SqlCommand cmd = null;
    using (var cnn = new SqlConnection(_cnnStr))
    {
        cnn.Open();
        var trans = cnn.BeginTransaction();
        try
        {
            queryString = "DELETE FROM PESSOA_TELEFONE WHERE ID_PESSOA = @ID";
            cmd = new SqlCommand(queryString, cnn, trans);
            cmd.Parameters.Add("@ID", System.Data.SqlDbType.Int).Value = p.id;
            cmd.ExecuteNonQuery();
        }
    }
}

```

```

        queryString = "DELETE FROM PESSOA WHERE ID = @ID";
        cmd = new SqlCommand(queryString, cnn, trans);
        cmd.Parameters.Add("@ID", System.Data.SqlDbType.Int).Value = p.id;
        cmd.ExecuteNonQuery();

        if (p.endereco != null)
        {
            queryString = "DELETE FROM ENDEREÇO WHERE ID = @ID";
            cmd = new SqlCommand(queryString, cnn, trans);
            cmd.Parameters.Add("@ID", System.Data.SqlDbType.Int).Value = p.endereco.id;
            cmd.ExecuteNonQuery();
        }

        trans.Commit();

        return true;
    }
    catch (Exception ex)
    {
        trans.Rollback();
        throw ex;
    }
}

public bool altere(Pessoa p)
{
    string queryString = null;
    SqlCommand cmd = null;
    using (var cnn = new SqlConnection(_cnnStr))
    {
        cnn.Open();
        var trans = cnn.BeginTransaction();
        try
        {
            if (p.endereco != null)
            {
                if (p.endereco.id > 0)
                {
                    alteraEndereco(cnn, trans, p.endereco);
                }
                else
                {
                    p.endereco.id = insiraEndereco(cnn, trans, p.endereco);
                }
            }

            alterePessoa(cnn, trans, p);

            queryString = @"
                DELETE PESSOA_TELEFONE
                FROM PESSOA_TELEFONE
                WHERE ID_PESSOA = @ID
            ";
            cmd = new SqlCommand(queryString, cnn, trans);
            cmd.Parameters.Add("@ID", System.Data.SqlDbType.Int).Value = p.id;
            cmd.ExecuteNonQuery();

            insiraTelefone(cnn, trans, p.id, p.telefones);
            trans.Commit();

            return true;
        }
        catch (Exception ex)
        {
            trans.Rollback();
        }
    }
}

```

```

    }
    }
    }
    }
    throw ex;
}

```

2.6. Protótipo funcional

Figura 3 – Protótipo funcional em Console

```

-----
Selecione a opção:
[1] - Consultar por CPF
[2] - Inserir Pessoa
[3] - Alterar Pessoa
[4] - Excluir Pessoa
[x] - Sair
-----
Digite o comando: 2
-----
Inserir Pessoa - Preencha dos dados
Nome: John Doe
CPF: 98765432100
Deseja cadastrar o endereço?[s,n]: n
Deseja cadastrar um novo telefone?[s,n]: s
DDD: 6198765432
Erro: '6198765432' não é um número válido.
DDD: 61
Número: 987654321
Tipo: Celular
Deseja cadastrar um novo telefone?[s,n]: n
Pessoa incluída com sucesso!
-----
Selecione a opção:
[1] - Consultar por CPF
[2] - Inserir Pessoa
[3] - Alterar Pessoa
[4] - Excluir Pessoa
[x] - Sair
-----
Digite o comando: |

```

Fonte: Autor

3. LAYOUT

3.1 Layout Asp.NET

3.1.1. Telas Responsivas Asp.NET

Figura 4 – Tela em Asp.NET (Desktop)

Cadastro de Pessoa - PIM VIII

localhost:5192

PIM VIII Busca por Pessoa Incluir

Cadastro de Pessoa

@Desenvolvido por Rodrigo Landim Carneiro.

Nome: CPF:

☒ Deseja cadastrar o endereço?

CEP: Logradouro: Número:

Bairro: Cidade: Estado:

☒ Deseja cadastrar um telefone?

Telefones:

Tipo: <input type="text" value="Celular"/>	DDD: <input type="text" value="61"/>	Número: <input type="text" value="998752588"/>	<input type="button" value="Excluir"/>
Tipo: <input type="text" value="Casa"/>	DDD: <input type="text" value="DDD"/>	Número: <input type="text" value="Número"/>	<input type="button" value="Novo"/>

© 2023 - PIM VIII - [Política de Privacidade](#)

Fonte: Autor

Figura 5 – Tela em Asp.NET (Celular)

PIM VIII Busca por Pessoa Incluir

Cadastro de Pessoa

@Desenvolvido por Rodrigo Landim Carneiro.

Nome:

CPF:

☒ Deseja cadastrar o endereço?

CEP:

Logradouro:

Número:

Bairro:

Cidade:

Estado:

☒ Deseja cadastrar um telefone?

Fonte: Autor

3.1.2. Codificação arquivo _Layout.cshtml usando RAZOR

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>@ViewData["Title"] - PIM VIII</title>
  <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
  <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
  <link rel="stylesheet" href="~/TelaASPNET.styles.css" asp-append-version="true" />
</head>
<body>
  <header>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom
box-shadow mb-3">
      <div class="container">
        <a class="navbar-brand" asp-area="" asp-page="/Index">PIM VIII</a>
      </div>
    </nav>
  </header>
</body>
</html>
```

```

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target=".navbar-collapse" aria-controls="navbarSupportedContent"
        aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
            <ul class="navbar-nav flex-grow-1">
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-page="/Index">Busca por
Pessoa</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-page="/Privacy">Incluir</a>
                </li>
            </ul>
        </div>
    </nav>
</header>
<div class="container">
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>

<footer class="border-top footer text-muted">
    <div class="container">
        &copy; 2023 - PIM VIII - <a asp-area="" asp-page="/Privacy">Política de Privacidade</a>
    </div>
</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/js/site.js" asp-append-version="true"></script>

    @await RenderSectionAsync("Scripts", required: false)
</body>
</html>

```

3.1.3. Codificação arquivo index.cshtml usando RAZOR

```

@page
@model IndexModel
@{
    ViewData["Title"] = "Cadastro de Pessoa";
}

<div class="text-center">
    <h1 class="display-4">Cadastro de Pessoa</h1>
    <p>@@Desenvolvido por Rodrigo Landim Carneiro</a>.</p>
</div>
<div class="container">
    <form method="post">
        <div class="row">
            <div class="col-md-8">
                <div class="form-group">
                    <label for="nome">Nome:</label>
                    <input type="text" class="form-control" id="nome" name="nome" placeholder="Digite
seu nome" required />
                </div>
            </div>
            <div class="col-md-4">
                <div class="form-group">
                    <label for="cpf">CPF:</label>
                    <input type="text" class="form-control" id="cpf" name="cpf" placeholder="Digite seu
CPF" required />
                </div>
            </div>
        </div>
    </form>

```

```

        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-12">
        <div class="form-check form-switch" style="margin: 15px auto">
            <input type="checkbox" id="temendereco" name="temendereco" class="form-check-input"
checked="checked" />
            <label for="temendereco" class="form-check-label">Deseja cadastrar o
endereço?</label>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-3">
        <div class="form-group">
            <label for="cep">CEP:</label>
            <input type="text" class="form-control" id="cep" name="cep" placeholder="Digite o
CEP" required />
        </div>
    </div>
    <div class="col-md-7">
        <div class="form-group">
            <label for="logradouro">Logradouro:</label>
            <input type="text" class="form-control" id="logradouro" name="logradouro"
placeholder="Digite seu logradouro" required />
        </div>
    </div>
    <div class="col-md-2">
        <div class="form-group">
            <label for="numero">Número:</label>
            <input type="text" class="form-control" id="numero" name="numero"
placeholder="Digite o número" required />
        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-4">
        <div class="form-group">
            <label for="bairro">Bairro:</label>
            <input type="text" class="form-control" id="bairro" name="bairro"
placeholder="Digite o bairro" required />
        </div>
    </div>
    <div class="col-md-4">
        <div class="form-group">
            <label for="cidade">Cidade:</label>
            <input type="text" class="form-control" id="cidade" name="cidade"
placeholder="Digite a cidade" required />
        </div>
    </div>
    <div class="col-md-4">
        <div class="form-group">
            <label for="estado">Estado:</label>
            <input type="text" class="form-control" id="estado" name="estado"
placeholder="Digite o estado" required />
        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-12">
        <div class="form-check form-switch" style="margin: 15px auto">
            <input type="checkbox" id="temendereco" name="temendereco" class="form-check-input"
checked="checked" />
            <label for="temendereco" class="form-check-label">Deseja cadastrar um
telefone?</label>
        </div>
    </div>
</div>

```

```

</div>
<div class="form-group">
  <label for="telefones">Telefones:</label>
  <div class="row">
    <div class="col-md-3">
      <label for="tipoTelefone1">Tipo:</label>
      <select class="form-select" id="tipoTelefone1" name="tipoTelefone1">
        <option value="1">Casa</option>
        <option value="2">Comercial</option>
        <option value="3" selected>Celular</option>
      </select>
    </div>
    <div class="col-md-2">
      <label for="ddd1">DDD:</label>
      <input type="text" class="form-control" id="ddd1" name="ddd1" placeholder="DDD"
value="61" />
    </div>
    <div class="col-md-5">
      <label for="numeroTelefone1">Número:</label>
      <input type="text" class="form-control" id="numeroTelefone1" name="numeroTelefone1"
placeholder="Número" value="998752588" />
    </div>
    <div class="col-md-2">
      <button type="button" class="btn btn-danger" style="margin-top:
22px">Excluir</button>
    </div>
  </div>
  <div class="row">
    <div class="col-md-3">
      <label for="tipoTelefone1">Tipo:</label>
      <select class="form-select" id="tipoTelefone1" name="tipoTelefone1">
        <option value="1">Casa</option>
        <option value="2">Comercial</option>
        <option value="3">Celular</option>
      </select>
    </div>
    <div class="col-md-2">
      <label for="ddd1">DDD:</label>
      <input type="text" class="form-control" id="ddd1" name="ddd1" placeholder="DDD" />
    </div>
    <div class="col-md-5">
      <label for="numeroTelefone1">Número:</label>
      <input type="text" class="form-control" id="numeroTelefone1" name="numeroTelefone1"
placeholder="Número" />
    </div>
    <div class="col-md-2">
      <button type="button" class="btn btn-success" style="margin-top:
22px">Novo</button>
    </div>
  </div>
  <div class="row">
    <div class="col-md-12 text-center" style="margin-top: 20px">
      <button type="submit" class="btn btn-primary">Enviar</button>
      <button type="submit" class="btn btn-danger">Cancelar</button>
    </div>
  </div>
</form>
</div>

```

3.2 Layout Android

3.2.1. Tela Android

Figura 6 – Tela Android

The image shows a smartphone screen displaying a registration form titled "Cadastro de Pessoa". The form includes the following fields: "Nome:" with a placeholder "Digite seu nome"; "CPF:" with a placeholder "Digite seu CPF"; "Endereço:" with sub-fields for "Logradouro", "Número", "CEP", "Bairro", "Cidade", and "Estado"; and "Telefones:" with sub-fields for "Tipo", "DDD", and "Número". At the bottom of the form are two purple buttons: "Adicionar Telefone" and "Enviar". The status bar at the top shows the time as 6:55 and various system icons.

Fonte: Autor

3.2.2. Codificação arquivo activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```

android:fitsSystemWindows="true"
tools:context=".MainActivity">
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".CadastroActivity">
<TextView
    android:id="@+id/textViewTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Cadastro de Pessoa"
    android:textSize="24sp"
    android:textStyle="bold" />
<TextView
    android:id="@+id/textViewNome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Nome:" />
<EditText
    android:id="@+id/editTextNome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Digite seu nome" />
<TextView
    android:id="@+id/textViewCPF"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="CPF:" />
<EditText
    android:id="@+id/editTextCPF"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Digite seu CPF" />
<TextView
    android:id="@+id/textViewEndereco"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Endereço:" />
<EditText
    android:id="@+id/editTextLogradouro"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Logradouro" />
<EditText
    android:id="@+id/editTextNumero"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Número" />
<EditText
    android:id="@+id/editTextCEP"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="CEP" />
<EditText
    android:id="@+id/editTextBairro"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Bairro" />
<EditText
    android:id="@+id/editTextCidade"
    android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:hint="Cidade" />
<EditText
    android:id="@+id/editTextEstado"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Estado" />
<TextView
    android:id="@+id/textViewTelefones"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Telefones:" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <EditText
            android:id="@+id/editTextTipo1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:hint="Tipo" />
        <EditText
            android:id="@+id/editTextDDD1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:hint="DDD" />
        <EditText
            android:id="@+id/editTextNumeroTelefone1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:hint="Número" />
    </LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:orientation="horizontal"
    android:gravity="end">
    <Button
        android:id="@+id/buttonAdicionarTelefone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Adicionar Telefone" />
    <Button
        android:id="@+id/buttonEnviar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:text="Enviar" />
    </LinearLayout>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

4. CONCLUSÃO

A análise dos requisitos para o desenvolvimento de um sistema de controle foi facilitada pelo documento de requisitos, que trazia uma descrição concisa do problema a ser tratado, bem como as funcionalidades que o sistema deveria oferecer aos seus usuários, conforme definido pelo cliente. O documento passou a apresentar requisitos funcionais e não funcionais, que determinariam as limitações de operação do sistema com base nos requisitos funcionais. Um Diagrama de Caso de Uso foi então utilizado para fornecer uma visão geral do funcionamento do software, incluindo os atores envolvidos em cada funcionalidade do sistema. Por fim, o Diagrama de Classes serviu para uma compreensão mais ampla do software a ser criado. Concluída a análise do sistema, o próximo passo é codificar as classes e fornecer exemplos de programação usando Visual Studio 2022, Android Studio em C#, CSS e XML. Para garantir que o sistema seja responsivo e adaptável a qualquer tela, foi criado um layout que funciona tanto para uso web quanto mobile. O aplicativo inclui botões para listar os usuários por CPF ou nome e, após selecionar um usuário, é possível editar, excluir ou criar novos perfis. A implementação de todos os métodos necessários é concluída e a etapa final é a revisão dos códigos apresentados.

5. REFERENCIAS BIBLIOGRÁFICAS

UNIP, Apostila de Programação Orientada a Objetos II, 2023.

UNIP, Apostila de Desenvolvimento de Software para Internet, 2023.

UNIP, Apostila de Tópicos Especiais de Programação Orientada a Objetos, 2023.

UNIP, Manual do PIM VIII, 2023.