UNIVERSIDADE PAULISTA – UNIP EaD
Projeto Integrado Multidisciplinar
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistema

MIRIAN DE SOUZA BERNARDES SANTANA – 0550217
RICARDO SANTOS SANTANA – 0549588

CODIFICAÇÃO EM C# DO MECANISMO DE ACESSO A UM TRECHO DE BANCO DE DADOS, ASSIM COMO OS PROTÓTIPOS DE INTERFACE GRÁFICA COM O USUÁRIO EM ASP.NET E ANDROID.

São Paulo
2021

MIRIAN DE SOUZA BERNARDES SANTANA - 0550217
RICARDO SANTOS SANTANA – 0549588

CODIFICAÇÃO EM C# DO MECANISMO DE ACESSO A UM TRECHO DE BANCO DE DADOS, ASSIM COMO OS PROTÓTIPOS DE INTERFACE GRÁFICA COM O USUÁRIO EM ASP.NET E ANDROID.

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

Orientador: Prof. José Cassiano Grassi Gunji

São Paulo

2021

“Desde a minha juventude, ó Deus, tens me ensinado, e até hoje eu anuncio as tuas maravilhas. ”

(BÍBLIA, Salmos 71:17).

RESUMO

O Propósito do presente projeto é desenvolver a codificação, em linguagem de programação C#, de um mecanismo de acesso a um trecho de banco de dados predefinido, assim como os protótipos de interface gráfica em ASP.NET e Android, ambos respeitando os princípios da responsividade, garantido assim que diversos dispositivos consigam visualizar a aplicação satisfatoriamente. Todo o projeto deve se embasar nas disciplinas de Programação Orientada a Objetos II, para auxílio da codificação das classes de entidades e da lógica dos métodos da classe de entidades, acesso a dados e regras de negócio, e no desenvolvimento de métodos privados, protegidos e públicos, conforme diagramas fornecidos. A disciplina de Desenvolvimento de Software para Internet, para auxílio no desenvolvimento da interface gráfica em ASP.NET, além de uso de HTML (Linguagem de Marcação de Hipertexto), e CSS (Folhas de Estilo em Cascata) para estilização das TAGS do HTML, utilizando o Microsoft Visual Studio. A disciplina Tópicos Especiais de Programação Orientada a Objetos, para auxílio no desenvolvimento da interface gráfica em Android, utilizando como ferramenta auxiliar o Android Studio.

Palavras-Chaves: Programação Orientada a Objetos II. Desenvolvimento de Software para a Internet. Tópicos Especiais de Programação Orientada a Objetos.

ABSTRACT

The purpose of this project is to develop the coding, in C# programming language, of an access mechanism to a predefined database snippet, as well as the graphical interface prototypes in ASP.NET and Android, both respecting the principles of responsiveness, guaranteed as soon as several devices can view the application satisfactorily. The entire project must be based on the Object-Oriented Programming II disciplines, to aid in the codification of the entity classes and the logic of the entity class methods, data access and business rules, and in the development of private, protected and according to the diagrams provided. The discipline of Software Development for the Internet, to aid in the development of the graphical interface in ASP.NET, in addition to the use of HTML (Hypertext Markup Language) and CSS (Cascade Style Sheets) for styling HTML TAGS, using Microsoft Visual Studio. The course Special Topics in Object-Oriented Programming, to aid in the development of the graphical interface in Android, using the Android Studio as an auxiliary tool.

Keywords: Object Oriented Programming II. Internet Software Development. Special Topics in Oriented Programming to Objects.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 INTRODUÇÃO

Para o desenvolvimento desse projeto, levamos como premissa, fazermos parte de uma empresa fictícia de desenvolvimento de software, que recebeu como tarefa o desenvolvimento de alguns aspectos de um sistema que já está em desenvolvimento, onde será necessário criar o mecanismo de acesso ao banco de dados em linguagem C#, esse mecanismo deve ser responsável por oferecer acesso a um trecho do banco de dados por parte do resto do sistema. Sempre que um trecho do sistema precisar acessar esse trecho do banco de dados, deverá fazê-lo por meio desse mecanismo desenvolvido por nossa equipe. Será também necessário fazer a entrega do protótipo de interface gráfica do usuário em ASP.NET, que permita que o usuário interaja com os dados modelados por esse trecho do banco de dados supracitado, além de um protótipo de interface gráfica com o usuário em Android que permita que o usuário interaja com os dados modelados pelo mesmo trecho do banco de dados. O modelo do trecho do banco de dados foi fornecido por nosso cliente fictício, sendo dois diagramas, um DER (Diagrama Entidade-Relacionamento) e o diagrama de Classe.

As classes devem ser desenvolvidas exatamente conforme definidas no diagrama de classe, a classe PessoaDAO definida no diagrama de classe deve conter exatamente os métodos públicos que foram explicitamente declarados, entretanto, se for necessário é permitido criar métodos auxiliares, como privados ou protegidos, porém não públicos. Embora seja declarado aqui como apenas um projeto, a solução do mesmo será dividida em três partes, sendo um projeto a codificação do PessoaDAO e as classes de entidades e acesso a dados, um projeto o código em ASP.NET e outro projeto em Android Studio com o uso da linguagem JAVA.

2 PROGRAMAÇÃO ORIENTADA A OBJETOS II

Segundo AGUILAR (2011), o paradigma orientado a objetos foi criado em 1969 pelo norueguês Kristin Nygaard que ao tentar escrever um programa de computador que descrevesse o movimento dos barcos em um fiorde, descobriu ser muito difícil simular as marés, os movimentos dos barcos e as formas da linha da costa com os métodos de programação existentes naquele momento. Ele descobriu que os elementos do ambiente que pretendia modelar (barcos, marés e linhas da costa dos fiordes) e as ações que cada elemento podia executar constituíam relações mais fáceis de serem manuseadas.

Ainda segundo AGUILAR (2011), a programação orientada a objetos é um conjunto importante de técnicas que podem ser utilizadas para realizar o desenvolvimento de programas dando eficiência e melhora a confiabilidade de programas de computação. Na programação orientada a objetos, os objetos são os elementos principais de construção. Entretanto, a simples compreensão do que é um objeto, ou o uso de objetos em um programa, não significa que estamos programando de uma maneira orientada a objetos. O que conta é o sistema no qual os objetos se interconectam e se comunicam entre si.

Segundo SANTOS, SARAIVA, GONÇALVEZ (2018), a programação orientada a objetos é um dos paradigmas mais utilizados na atualidade no desenvolvimento de sistemas, onde é possível aos desenvolvedores e analistas construírem classes e implementar métodos que representam o mundo real, por meio de características, estados e comportamentos.

Conforme SINTES (2002), a programação orientada a objetos (POO) é apoiada em três pilares: encapsulamento, herança e polimorfismo.

2.1 Objetos

Segundo BARNES, KÖLLING (2009), os objetos representam instanciações individuais da classe, ou seja, um objeto está para um produto assim como uma classe está para uma receita, pois de uma receita podemos criar vários produtos do mesmo. É possível se comunicar com os objetos invocando seus métodos.

2.2 Classes

Segundo SINTES (2002), uma classe define os atributos e comportamentos comuns compartilhados por um tipo de objeto. Os objetos de certo tipo ou classificação compartilham os mesmos comportamentos e atributos. As classes atuam de forma muito parecida com um cortador de molde ou biscoito, no sentido de que você usa uma classe para criar ou instanciar objetos, ou seja, a receita para a construção de objetos.

Segundo BARNES, KÖLLING (2009), uma classe é um modelo de construção de objetos, uma maneira abstrata de descrever os objetos de um tipo particular.

2.3 Herança

Segundo BARNES, KÖLLING (2009), a herança é uma construção poderosa que pode ser utilizada para criar soluções variadas para diferentes problemas. A Herança permite as classes compartilharem métodos e atributos, o que auxilia sobremaneira o reaproveitamento de código.

Conforme SANTOS, SARAIVA, GONÇALVEZ (2018), a herança é um dos principais pontos da programação orientada a objetos, promovendo a extensibilidade do código, a reutilização e uma maior coerência lógica no modelo de implementação. Os métodos herdados são usados da mesma forma que os métodos não herdados, nenhum trecho de código é necessário para mencionar que os métodos serão herdados, o uso do recurso em uma classe não requer saber se ele foi ou não herdado, o que gera flexibilidade para o programador. Entre as vantagens oferecidas pela herança, podemos citar a manutenção de bibliotecas, para uso futuro de determinados recursos que sejam utilizados com frequência.

2.4 Polimorfismo

Segundo SANTOS, SARAIVA, GONÇALVEZ (2018), o polimorfismo é um dos conceitos mais utilizados em programação orientada ao objeto, pois promove a reutilização contínua dos códigos, apresentando a maneira como um método pode assumir formas diferentes das quais foram implementadas inicialmente e agir de modo que possa ser reutilizado, inclusive em outra classe. O polimorfismo cria variações de

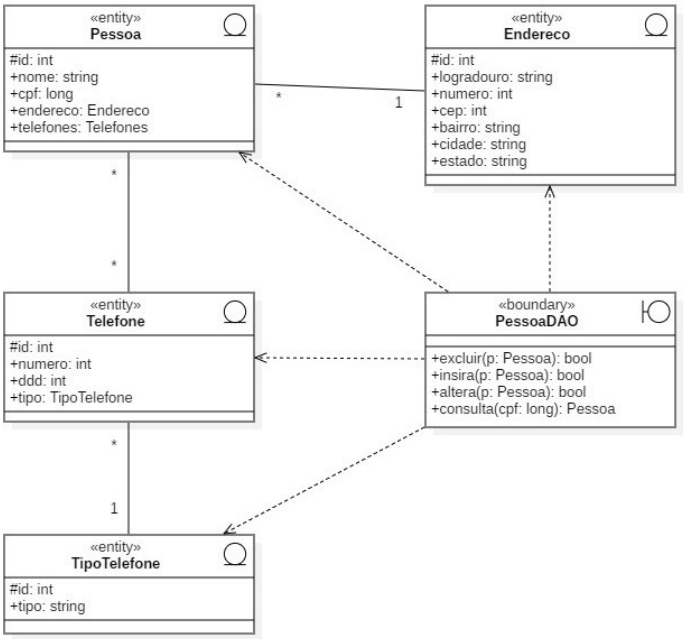
métodos com nomes totalmente iguais em uma classe, contendo lista de argumentos diferentes para que seja feita a separação deles.

2.5 Encapsulamento

Segundo AGUILAR (2011) é a programação orientada a objetos que proporciona a forma adequada para a reutilização das classes e conceitos de encapsulamento e herança formam a base que facilitam a reutilização. Um programador pode acrescentar novas funcionalidades e características em uma classe sem modifica-la, apenas utilizando como base uma classe existente. Sendo que uma classe pode herdar as propriedades de uma classe antiga, e ainda somar suas propriedades. AGUILAR (2011) exemplifica que supondo que escrevendo (ou comprando) uma classe menu que cria um sistema de menu (barras de ferramentas, quadros de diálogos, botões etc.); com o tempo, ainda que a classe funcione bem, observamos que seria interessante que as legendas das opções dos menus mudassem de cor. Para realizar essa tarefa, projetamos uma classe derivada de menu que agregue as novas propriedades de mudança de cor.

2.6 Diagrama de Classes

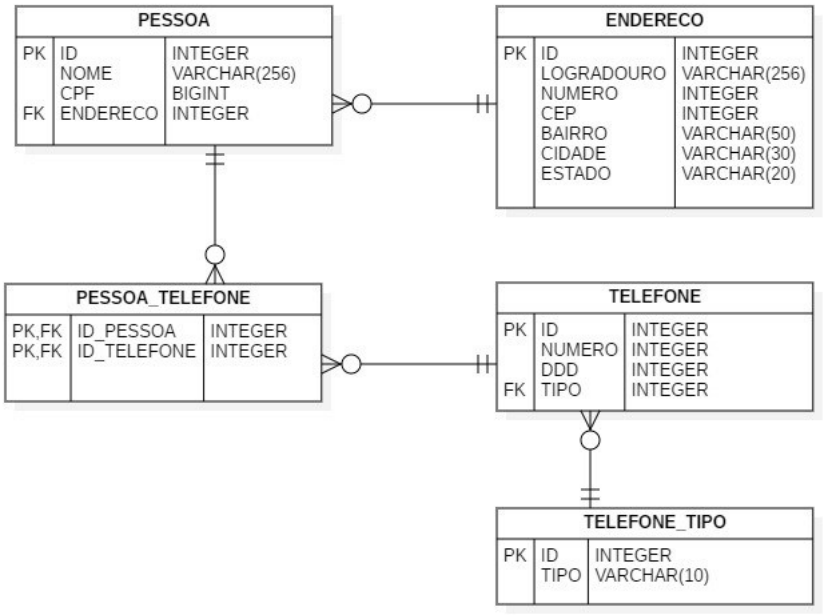
Imagem 1 - Diagrama de Classes



Fonte: O Autor (2021)

2.7 Diagrama de Entidade Relacional

Imagem 2 - Diagrama de Entidade Relacional



Fonte: O Autor (2021)

2.8 Codificação da classe PessoaDAO

Segue no APÊNDICE A a estrutura da classe PessoaDAO

2.9 Codificação das classes de entidades

Segue no APÊNDICE B as classes de entidades pertencente a classe PessoaDAO, baseados do diagrama de classe e entidade relacional.

2.10 Lógica dos métodos públicos da classe PessoaDAO

Segue no APÊNDICE A os métodos públicos da classe PessoaDAO baseados no diagrama de classe fornecido para a solução.

2.11 Métodos Privados e Atributos da Classe PessoaDAO

Segue no APÊNDICE A os métodos privados da classe PessoaDAO, assim como os atributos privados, conforme solicitado pelos diagramas de classe fornecido para a solução.

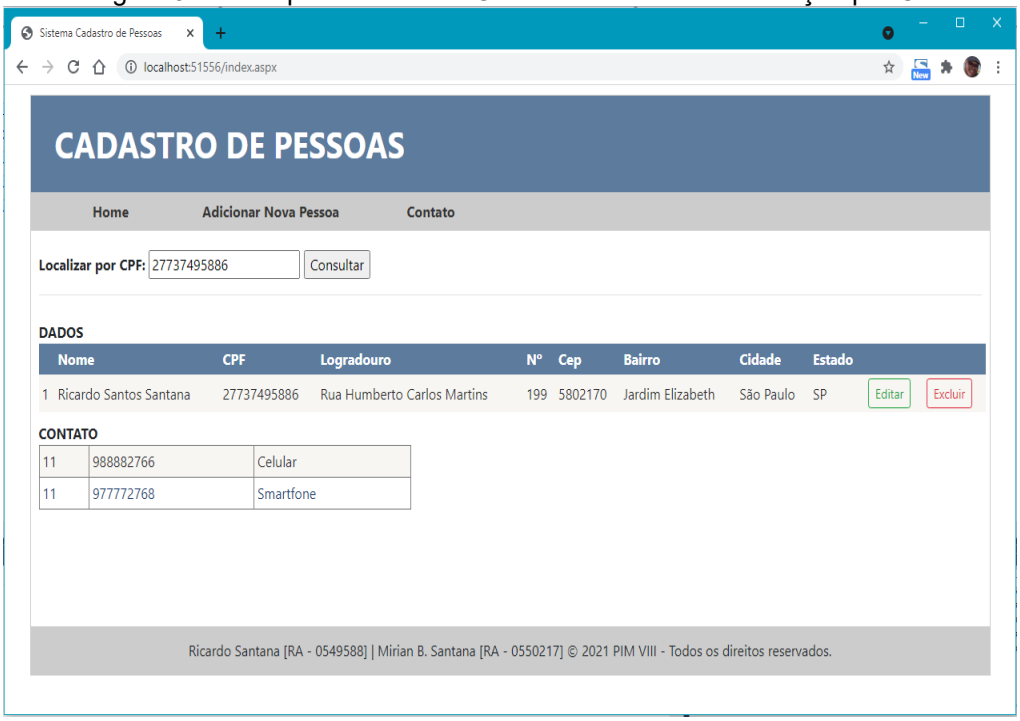
3 DESENVOLVIMENTO DE SOFTWARE PARA A INTERNET

Segundo MILETTO, BERTAGNOLLI (2014), o HTML é a base para criar uma página e exibir em um navegador Web. O HTML é formado por um conjunto de TAGs que possibilita exibir o conteúdo e utilizar recursos hipermídia – links, imagens, tabelas, vídeos. Porém, seus recursos de formatação visual são muito restritos e simples. Para resolver os problemas do HTML puro, existe o CSS, ou folhas de estilo, que permite diferentes tipos de formatações, como bordas, cores, fundo, elementos textuais estilizados e layouts diferenciados.

Ainda conforme MILETTO, BERTAGNOLLI (2014), para projetar um site, é necessário reconhecer que as aplicações Web são um tipo de software que utiliza a Internet como um ambiente de execução. Essa aplicação amplia o conceito de Web site ao adicionar funcionalidades ao sistema. E, para compreender esse tipo de sistema, devemos entender os requisitos e mapear suas principais funcionalidades. Para projetar um site é necessário seguir alguns passos, que devem ser mapeados em diagramas e modelos, de modo a facilitar a comunicação entre as pessoas envolvidas.

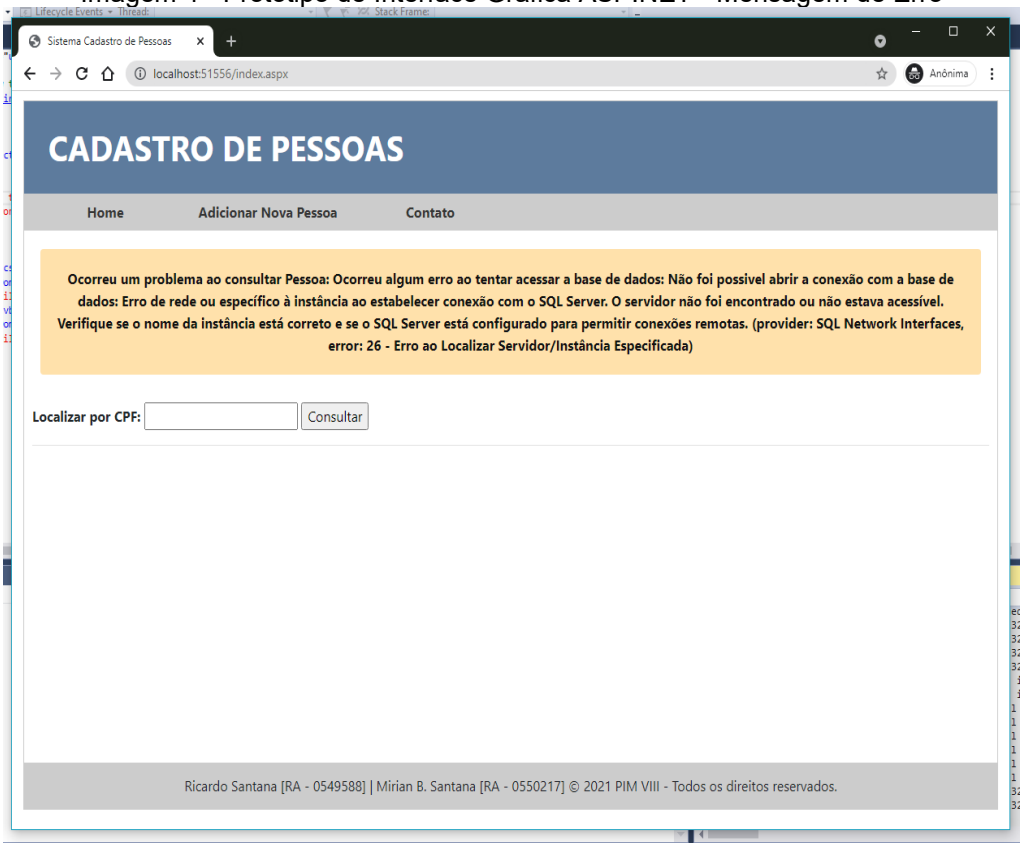
3.1 Protótipo de Interface Gráfica em ASP.NET

Imagem 3 - Protótipo de Interface Gráfica ASP.NET - Localização por CPF



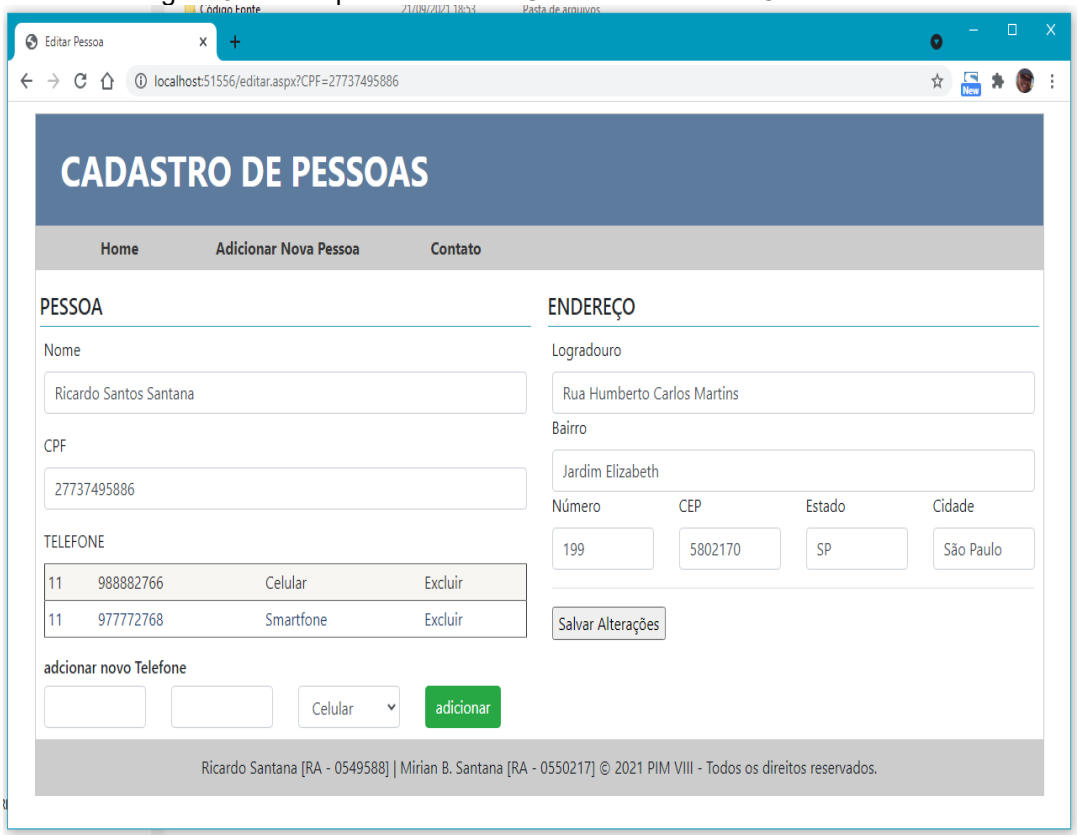
Fonte: O Autor (2021)

Imagem 4 - Protótipo de Interface Gráfica ASP.NET - Mensagem de Erro



Fonte: O Autor (2021)

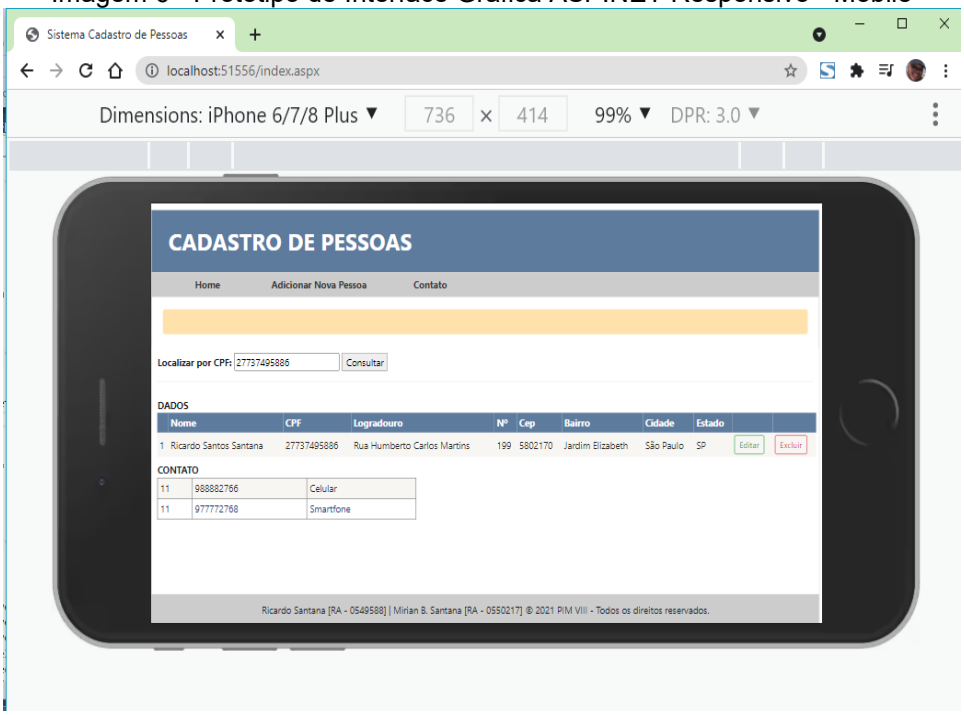
Imagem 5 - Protótipo de Interface Gráfica ASP.NET – Cadastro Pessoa



Fonte: O Autor (2021)

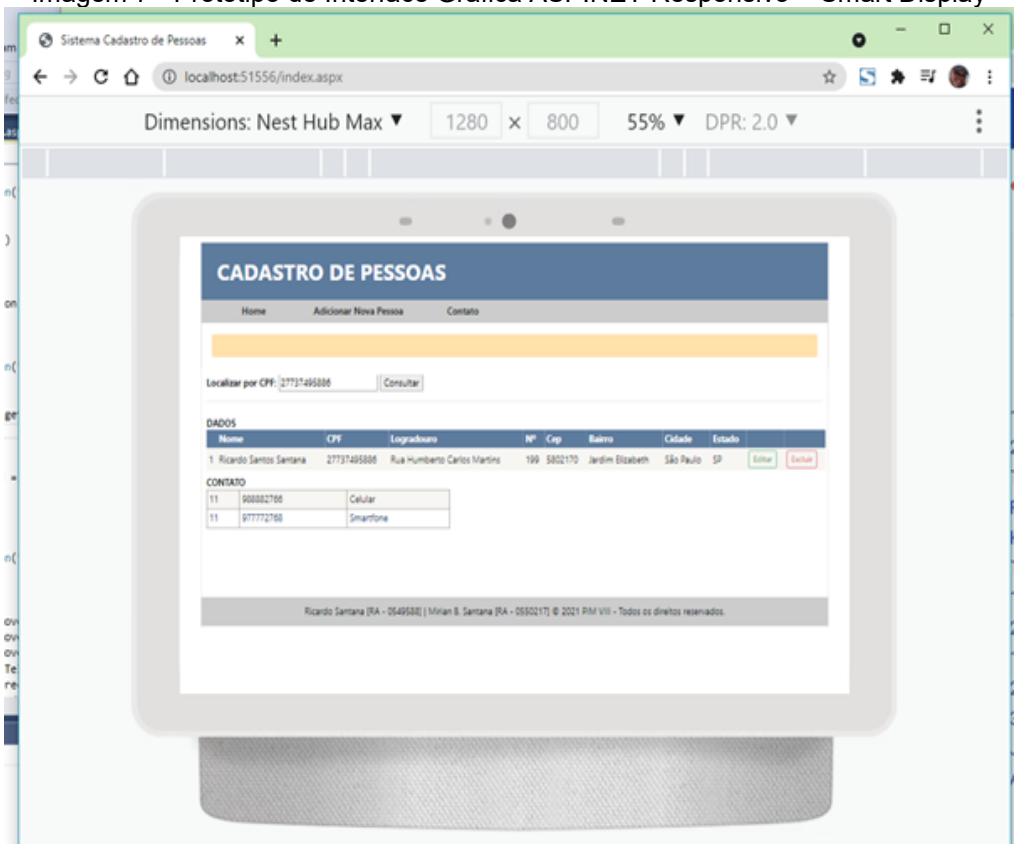
3.2 Protótipo de Interface Gráfica em ASP.NET Responsivo

Imagem 6 - Protótipo de Interface Gráfica ASP.NET Responsivo - Mobile



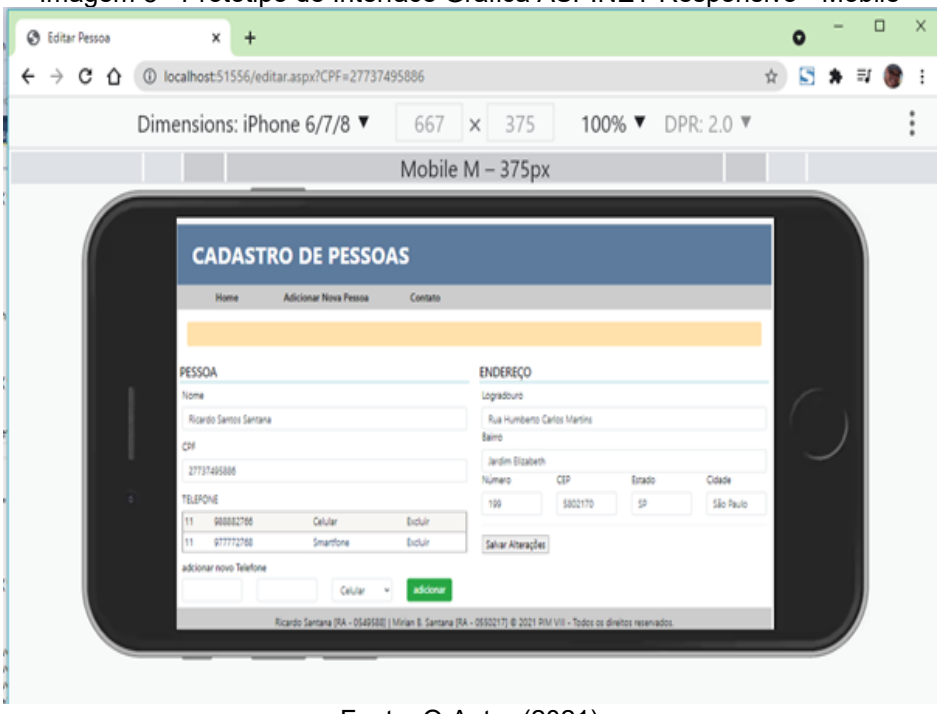
Fonte: O Autor (2021)

Imagem 7 - Protótipo de Interface Gráfica ASP.NET Responsivo – Smart Display



Fonte: O Autor (2021)

Imagem 8 - Protótipo de Interface Gráfica ASP.NET Responsivo - Mobile



Fonte: O Autor (2021)

Imagem 9 - Protótipo de Interface Gráfica ASP.NET Responsivo – Smart Display

Fonte: O Autor (2021)

3.3 Código ASPX do Layout dos Formulários

O código do index.aspx segue no APÊNDICE C, index.aspx.cs segue no APÊNDICE D, o editar.aspx segue no APÊNDICE E, o editar.aspx.cs segue no APÊNDICE F.

4 TÓPICOS ESPECIAIS DE PROGRAMAÇÃO ORIENTADA A OBJETOS

> O conhecimento tecnológico dos homens propiciou a criação de diversos mecanismos capazes de realizar tarefas. Há séculos existem aparelhos mecânicos capazes de realizar tarefas dos mais diversos níveis de complexidade. Por exemplo, existiram teares capazes de tecer padrões complexos em tecidos de maneira automática e máquinas capazes de realizar cálculos matemáticos e até mesmo codificar e decodificar mensagens de rádio. A tecnologia desses mecanismos podia ser mecânica, elétrica e, mais recentemente, eletrônica. Algo que esses dispositivos tinham em comum era algum método pelo qual eles poderiam ser programados. (GUNJI, 2020, p. 7)

Segundo FURGERI (2012), a linguagem JAVA possui um nome relativamente estranho para uma linguagem de programação se comparado a Pascal, Cobol, Basic e diversos outros nomes de linguagem. O nome JAVA surgiu com acaso, quando a equipe de engenheiros da Sun, a criadora do Java, foi tomar café na esquina. Como numa reunião de amigos, começaram a chamar a linguagem de Java, inspirados pelo nome da cidade de onde provinha o café que tomavam, a equipe nem imaginaria que o nome seria conhecido mundialmente.

Ainda segundo FURGERI (2012), originalmente a linguagem foi concebida para a utilização em pequenos dispositivos eletrônicos inteligentes; entretanto, com as dificuldades de financiamento desse setor na época e, principalmente, devido ao surgimento da internet a partir de 1993, novas oportunidades apareceram e a Sun segundo ele “entrou de cabeça” nessa área.

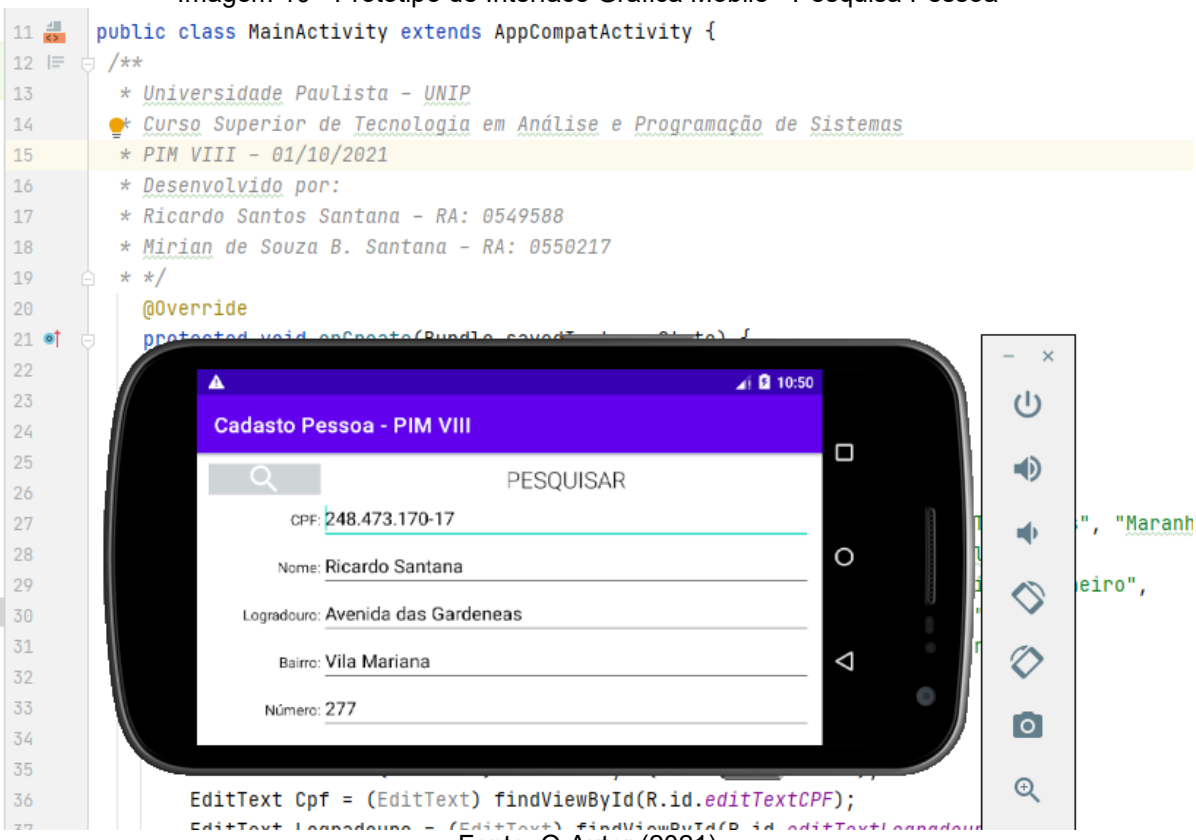
Conforme SCHILDT (2015), o avanço da Word Wide Web reformulou a computação. Antes da Web, o mundo cibernético ou o panorama como ele mesmo diz, era dominado por PCs isolados, entretanto, hoje, quase todos os computadores estão conectados à Internet. A própria Internet foi transformada, originalmente, oferecia uma maneira conveniente de compartilhar arquivos e informações; hoje é um universo de computação vasto e distribuído. Com essas mudanças surgiu uma nova forma de programar: Java.

Java é muito mais que simplesmente a principal linguagem da Internet, de acordo com SCHILDT (2015), “Ela revolucionou a programação, mudando a maneira de pensarmos tanto sobre a forma quanto sobre a função de um programa. Atualmente, ser um programador profissional exige a habilidade de programar em Java”.

De acordo com DEITEL (2015), a primeira geração de telefones com Android foi lançada em outubro de 2013, e o Android tinha 81,3% da fatia de mercado global de smartphones, comparados a 13,4% da Apple, 4,1% da Microsoft e 1% Blackberry. Em abril de 2013, mais de 1,5 milhões de aparelhos Android estava sendo ativado diariamente. Atualmente, dispositivos com Android incluem smartphones, tablets, leitores digitais, robôs, motores a jato, satélites da NASA, consoles de jogos, geladeiras, televisões, câmeras, equipamentos voltados à saúde, relógios inteligentes sistemas automotivos de bordo (para controlar rádio, GPS, ligações telefônicas, termostato, etc.) e muitos outros.

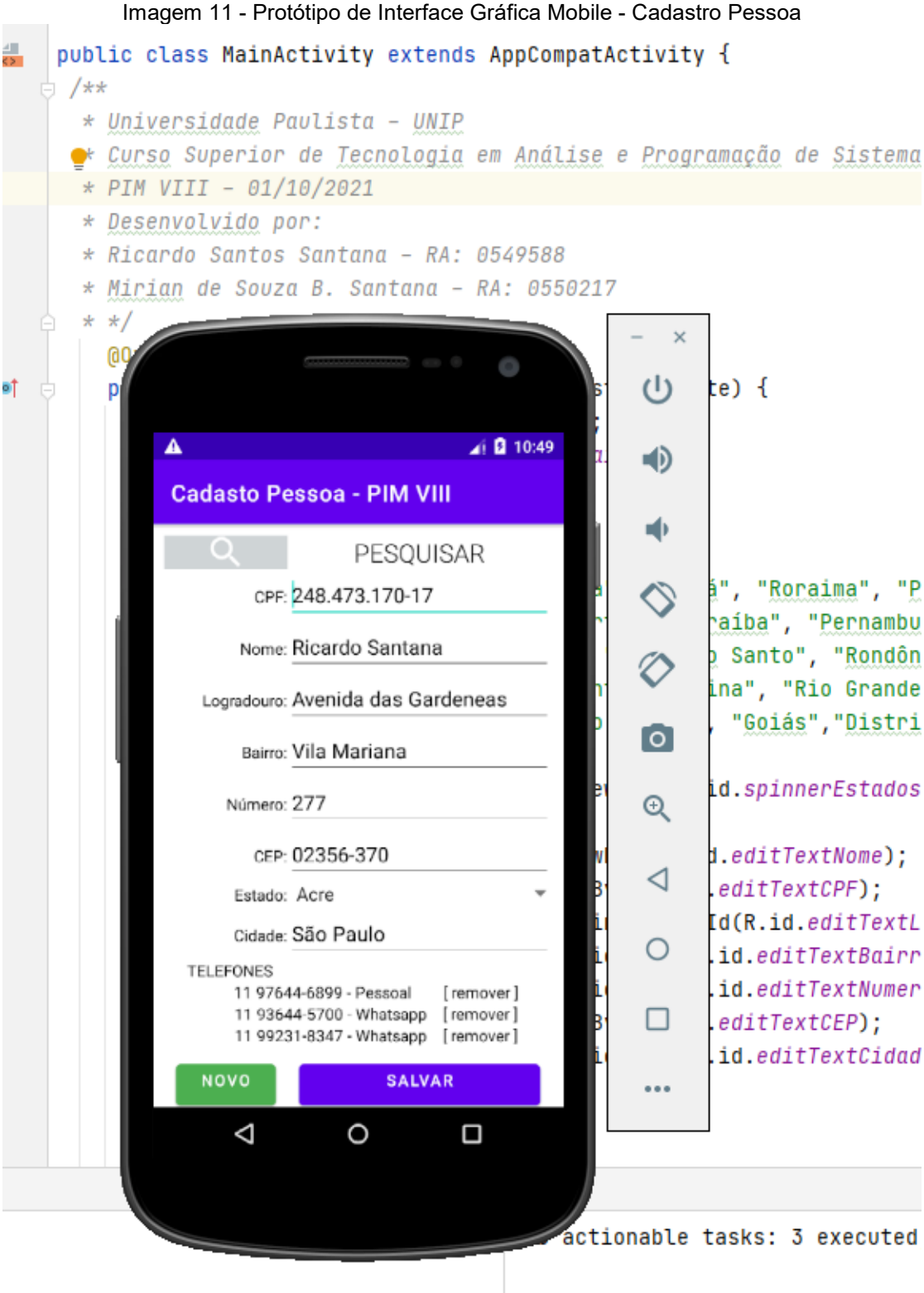
Segundo DEITEL (2015), os aplicativos Android são desenvolvidos com Java – uma das linguagens de programação mais usadas do mundo. Essa linguagem foi uma escolha lógica para a plataforma Android, pois é poderosa, gratuita, de código-fonte aberto e milhões de desenvolvedores já a conhecem. Os programadores Java experientes podem se aprofundar rapidamente no desenvolvimento com Android, usando as APIs (interfaces de programação de aplicativo) Android do Google e de outros.

Imagem 10 - Protótipo de Interface Gráfica Mobile - Pesquisa Pessoa



Fonte: O Autor (2021)

Imagem 11 - Protótipo de Interface Gráfica Mobile - Cadastro Pessoa



Fonte: O Autor (2021)

Segue o código XML do layout da Activity no APÊNDICE G.

CONCLUSÃO

Esse projeto resumiu tudo o que aprendemos até o momento, Análise de sistemas e Programação de Sistemas, e todas as disciplinas que envolvem o tema, o desafio de desenvolver um software do zero, seja uma aplicação WEB, Desktop ou Android, seria impossível sem conhecimento prévio da multidisciplinaridade que envolve a solução. O curso de Análise e Desenvolvimento de Sistemas, paulatinamente foi nos incutindo de conhecimento teórico e com os projetos multidisciplinares fomos desenvolvendo o conhecimento empírico, e aqui no PIM VIII tudo fez muito sentido, quando tivemos que ordenar todo o conhecimento de forma sequencial para desenvolver o projeto.

À Medida que íamos lendo o enunciado, íamos também visualizando inúmeras possibilidades de soluções, embora nesse caso específico fomos tolhidos para objetivar apenas uma solução específica, que é a solução proposta.

Nesse projeto demonstramos a aplicação prática do conhecimento adquirido nas disciplinas de Programação Orientada a Objetos II, Desenvolvimento de Software para a Internet e Tópicos Especiais de Programação Orientada a Objetos. De forma que desenvolvemos o projeto em ASP.NET com linguagem de programação C# no Visual Studio da Microsoft e Aplicativo Android com linguagem de programação JAVA no Android Studio.

Esse projeto nos deu a oportunidade de desenvolver e identificar necessidades sistêmicas para que propuséssemos soluções técnicas, tanto no desenvolvido das soluções propriamente ditas como na prática da leitura e no entendimento dos diagramas, seus símbolos e significados.

REFERÊNCIAS

AGUILAR, Luis Joyanes. **Fundamentos de Programação: Algoritmos, estruturas de dados e objetos**. Tradução Paulo Heraldo Costa do Valle. 3. ed. Porto Alegre: AMGH, 2011.

BARNES, David J.; KÖLLING, Michael. **Programação orientada a objetos com Java uma introdução prática usando o BlueJ**. 4. Ed. Tradução Edson Furmankiewicz. São Paulo: Pearson Education do Brasil, 2009.

DEITEL, Paul. **Android: Como Programar**. 2. ed. Tradução João Eduardo Nóbrega Tortello. Porto Alegre: Bookman, 2015.

FURGERI, Sergio. **Java 7: Ensino Didático**. 2. ed. São Paulo: Érica, 2012.

GUNJI, José Cassiano Grassi. **Tópicos Especiais de Programação Orientada a Objetos**. São Paulo: Editora Sol, 2020.

MILETTO, Evandro Marana; BERTAGNOLLI, Silvia de Castro. **Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP**. Porto Alegre: Grupo A, 2014.

SANTOS, Marcela Gonçalves dos; SARAIVA, Maurício de Oliveira; GONÇALVES, Priscila de Fátima. **Linguagem de Programação**. Porto Alegre: Editora A, 2018.

SCHILDT, Herbert. **Java para Iniciantes: Crei, Compile e Execute Programas Java Rapidamente**. 6. ed. Tradução Aldir José Coelho Corrêa Silva. Porto Alegre: Bookman, 2015.

SINTES, Anthony. **Aprenda orientação a objetos em 21 dias**. São Paulo: Pearson, 2002.

APÊNDICE A – CODIFICAÇÃO DAS CLASSE PESSOADAO

Classe PessoaDAO

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using ClassPessoa.Modal;
using System.Configuration;
namespace ClassPessoa.Controller {
	public static class PessoaDAO {
		//Atributos

		//Métodos privados

		//Métodos públicos

	}
}
```

Atributos

```
	private static SqlConnection con;
```

Métodos Privados

```
private static SqlConnection Open(){
	try{
		String strConexao = ConfigurationManager
			.ConnectionStrings["dbPessoasConnectionString"].ConnectionString;
		  if (con == null){
		      con = new SqlConnection(strConexao);
		      con.Open();
		  }
		  return con;
	}
	catch (Exception ex) {
		throw new Exception("Não foi possivel abrir a conexão com a base de
			dados: " + ex.Message);
	}
}

private static void Close() {
	try {
		if (con != null) con.Close();
	}
	catch (Exception ex) {
	    throw new Exception("Ocorreu um problema ao fechar a conexão com a
		base de dados: " + ex.Message);
	}
}
```

```
private static SqlCommand getData(String QuerySQL) {
	try {
		SqlCommand command = new SqlCommand(QuerySQL, Open());
		  return command;
	}
	catch (Exception ex) {
		throw new Exception("Ocorreu algum erro ao tentar acessar a base de
			dados: " + ex.Message);
	}
}

private static Boolean removeTelefone(Pessoa p) {
	try {
		string QueryTelefone = "";
		string QueryPessoaTelefone = "";
		List<Telefone> telefone = p.Telefone;

		foreach (Telefone item in telefone) {
			QueryTelefone = "DELETE FROM TELEFONE WHERE ID = @id";
			SqlCommand command = getData(QueryTelefone);
			command.Parameters.AddWithValue("@id", item.Id);
			command.ExecuteNonQuery();

			QueryPessoaTelefone = "DELETE FROM PESSOA_TELEFONE WHERE id_pessoa
			= @id_pessoa and id_telefone = @id_telefone";
			SqlCommand cmd = getData(QueryPessoaTelefone);
			cmd.Parameters.AddWithValue("@id_pessoa", p.Id);
			cmd.Parameters.AddWithValue("@id_telefone", item.Id);
			cmd.ExecuteNonQuery();

			if (con.State == System.Data.ConnectionState.Open) Close();
		}
		return true;
	}
	catch (Exception ex) {
		return false;
		throw new Exception("Ocorreu um problema ao remover telefone: " +
			ex.Message);
	}
}

private static Boolean removePessoa(Pessoa p) {
	try{
		string Query = "DELETE FROM PESSOA ID = @id";
		SqlCommand command = getData(Query);
		command.Parameters.AddWithValue("@id", p.Id);
		command.ExecuteNonQuery();

		if (con.State == System.Data.ConnectionState.Open) Close();
		return true;
	}
	catch (Exception ex) {
		return false;
		throw new Exception("Ocorreu um problema ao remover pessoa: " +
				ex.Message);
	}
}
```

```
private static Boolean removeEndereco(Pessoa p) {

	try {
		string Query = "DELETE FROM ENDERECO ID = @id";
		SqlCommand command = getData(Query);
		command.Parameters.AddWithValue("@id", p.Endereco.Id);
		command.ExecuteNonQuery();

		if (con.State == System.Data.ConnectionState.Open) Close();
		return true;
	}
	catch (Exception ex) {
		return false;
		throw new Exception("Ocorreu um problema ao remover endereço: " +
			ex.Message);
	}
}

private static Boolean addTelefone(Pessoa p) {
	try {
		string QueryTelefone = "";
		string QueryPessoaTelefone = "";
		List<Telefone> telefone = p.Telefone;

		foreach (Telefone item in telefone) {
			QueryTelefone = "INSERT INTO TELEFONE(numero, ddd, tipo)
					VALUES(@numero, @ddd, @tipo)";
			SqlCommand command = getData(QueryTelefone);
			command.Parameters.AddWithValue("@numero", item.Numero);
			command.Parameters.AddWithValue("@ddd", item.Ddd);
			command.Parameters.AddWithValue("@tipo", item.tipoTelefone.Id);
			command.ExecuteNonQuery();

			QueryPessoaTelefone = "INSERT INTO PESSOA_TELEFONE(id_pessoa,
			id_telefone) VALUES(@id_pessoa, @id_telefone)";
			SqlCommand cmd = getData(QueryPessoaTelefone);
			cmd.Parameters.AddWithValue("@id_pessoa", p.Id);
			cmd.Parameters.AddWithValue("@id_telefone", item.Id);
			cmd.ExecuteNonQuery();

			if (con.State == System.Data.ConnectionState.Open) Close();
		}
		return true;
	}
	catch (Exception ex) {
		return false;
		throw new Exception("Ocorreu um problema ao adicionar telefone: " +
			ex.Message);
	}
}
```

```
private static int addEndereco(Endereco endereco) {
	try {
		String Query = "INSERT INTO
			ENDERECO(logradouro, numero, cep, bairro, cidade, estado)
			VALUES(@logradouro, @numero, @cep, @bairro, @cidade, @estado;
			SELECT CAST(scope_identity() AS int)";

		SqlCommand Command = getData(Query);
		Command.Parameters.AddWithValue("@logradouro", endereco.Logradouro);
		Command.Parameters.AddWithValue("@numero", endereco.Numero);
		Command.Parameters.AddWithValue("@cep", endereco.Cep);
		Command.Parameters.AddWithValue("@bairro", endereco.Bairro);
		Command.Parameters.AddWithValue("@cidade", endereco.Cidade);
		Command.Parameters.AddWithValue("@estado", endereco.Estado);

		int idEndereco = (int)Command.ExecuteScalar();
		if (con.State == System.Data.ConnectionState.Open) Close();
		return idEndereco;
	}
	catch (Exception ex) {
		return 0;
		throw new Exception("Ocorreu um problema ao adicionar endereço: " +
		ex.Message);
	}
}

private static int updateEndereco(Endereco endereco) {
	try {
	   String Query = ("INSERT INTO
	       ENDERECO(logradouro,numero,cep,bairro,cidade,estado)
	       VALUES(@logradouro,@numero,@cep,@bairro,@cidade,@estado;
	       SELECT CAST(scope_identity() AS int)");

	   SqlCommand Command = getData(Query);
	   Command.Parameters.AddWithValue("@logradouro", endereco.Logradouro);
	   Command.Parameters.AddWithValue("@numero", endereco.Numero);
	   Command.Parameters.AddWithValue("@cep", endereco.Cep);
	   Command.Parameters.AddWithValue("@bairro", endereco.Bairro);
	   Command.Parameters.AddWithValue("@cidade", endereco.Cidade);
	   Command.Parameters.AddWithValue("@estado", endereco.Estado);

	   Command.Parameters.AddWithValue("@id", endereco.Id);
	   Query = ("UPDATE ENDERECO SET logradouro=@logradouro, numero=@numero,
			cep=@cep, bairro=@bairro, cidade=@cidade, estado=@estado WHERE
			id=@id");
	   Command.ExecuteNonQuery();
	   if (con.State == System.Data.ConnectionState.Open) Close();
	   return endereco.Id;
	}
	catch (Exception ex) {
		return 0;
		throw new Exception("Ocorreu um problema ao atualizar endereço: " +
			ex.Message);
	}
}
```

```
private static Boolean updatePessoa(Pessoa p) {
       try {
              String Query = ("UPDATE PESSOA SET nome = @nome, cpf=@cpf,
              endereco=@endereco WHERE id=@id");

              int idEndereco = p.Endereco.Id;

              SqlCommand Command = getData(Query);
              Command.Parameters.AddWithValue("@nome", p.Nome);
              Command.Parameters.AddWithValue("@cpf", p.CPF);
              Command.Parameters.AddWithValue("@endereco", idEndereco);
              Command.Parameters.AddWithValue("@id", p.Id);

              Command.ExecuteNonQuery();
              if (con.State == System.Data.ConnectionState.Open) Close();
              removeTelefone(p);
              addTelefone(p);
              return true;
       }
       catch (Exception ex) {
              return false;
              throw new Exception("Ocorreu um problema ao atualizar pessoa: " +
              ex.Message);
       }
}

private static Boolean addPessoa(Pessoa p) {
       try {
              String Query = ("INSERT INTO PESSOA(nome,cpf) VALUES(@nome,@cpf)");
              int idEndereco = addEndereco(p.Endereco);

              SqlCommand Command = getData(Query);
              Command.Parameters.AddWithValue("@nome", p.Nome);
              Command.Parameters.AddWithValue("@cpf", p.CPF);
              Command.Parameters.AddWithValue("@endereco", idEndereco);

              Command.ExecuteNonQuery();
              if (con.State == System.Data.ConnectionState.Open) Close();
              removeTelefone(p);
              addTelefone(p);
              return true;
       }
       catch (Exception ex) {
                 return false;
                 throw new Exception("Ocorreu um problema ao adicionar pessoa: " +
              ex.Message);
       }
}
```

Métodos Públicos

```
public static Boolean Insira(Pessoa p) {
	try {
		return addPessoa(p);
	}
	catch (Exception ex) {
		return false;
		throw new Exception("Ocorreu um problema ao inserir Pessoa: " +
			ex.Message);
	}

}

public static Boolean altera(Pessoa p) {
	try {
		return updatePessoa(p);
	}
	catch (Exception ex) {
		return false;
		throw new Exception("Ocorreu um problema ao alterar Pessoa: " +
				ex.Message);
	}
}


public static Boolean excluir(Pessoa p) {

	try {
		if (removeTelefone(p)) {
			if (removeEndereco(p)) {
			    removePessoa(p);
			}
		}
		return true;
	}
	catch (Exception ex) {
		return false;
		throw new Exception("Ocorreu um problema ao excluir Pessoa: " +
				ex.Message);
	}
}
```

```
public static Pessoa consulta(long cpf = 0) {
       try {
              string sqlQuery = "";
              sqlQuery += "SELECT ";
              sqlQuery += "P.ID, P.NOME, P.CPF , ";
              sqlQuery += " E.ID ID_ENDERECO, E.LOGRADOURO, E.NUMERO, E.CEP,
                             E.BAIRRO,E.CIDADE, E.ESTADO , ";
              sqlQuery += "T.ID ID_TELEFONE, T.NUMERO TELEFONE, T.DDD, ";
              sqlQuery += "TT.ID ID_TIPO_ENDERECO, TT.TIPO ";
              sqlQuery += "FROM PESSOA AS P ";
              sqlQuery += "INNER JOIN ENDERECO AS E ON P.ENDERECO = E.ID ";
              sqlQuery += "INNER JOIN PESSOA_TELEFONE PT ON PT.ID_PESSOA = P.ID ";
              sqlQuery += "INNER JOIN TELEFONE T ON PT.ID_TELEFONE = T.ID ";
              sqlQuery += "INNER JOIN TELEFONE_TIPO TT ON T.TIPO = TT.ID  WHERE
                      CPF=" + cpf;

              Pessoa pessoa = new Pessoa();

              Endereco endereco = new Endereco();
              List<Telefone> ListTelefone = new List<Telefone>();
              Telefone telefone;

              SqlCommand command  = getData(sqlQuery);

              int idPessoa = 0;
              using (var reader = command.ExecuteReader()) {
                     if (reader.HasRows) {
                         while (reader.Read()){
                             endereco.Id = (Int32)reader["ID_ENDERECO"];
                             endereco.Logradouro = (String)reader["LOGRADOURO"];
                             endereco.Numero = (Int32)reader["NUMERO"];
                             endereco.Cep = (Int32)reader["CEP"];
                             endereco.Bairro = (String)reader["BAIRRO"];
                             endereco.Cidade = (String)reader["CIDADE"];
                             endereco.Estado = (String)reader["ESTADO"];

                             int idTelefone = (Int32)reader["ID_TELEFONE"];
                             int numeroTelefone = (Int32)reader["TELEFONE"];
                             int numeroddd = (Int32)reader["DDD"];
                             string tipo = (string)reader["TIPO"];

                             if (idPessoa != (Int32)reader["ID"]) {
                                 ListTelefone = new List<Telefone>();
                                 telefone = new Telefone(idTelefone, numeroTelefone,
                                                numeroddd, new TipoTelefone(tipo));

                                 ListTelefone.Add(telefone);
                                 pessoa.Id = (Int32)reader["ID"];
                                 pessoa.Nome = (string)reader["NOME"];
                                 pessoa.CPF = (Int64)reader["CPF"];
                                 pessoa.Endereco = endereco;
                                 pessoa.Telefone = ListTelefone;
                             }
                             else {
                                 telefone = new Telefone(idTelefone, numeroTelefone,
                                                numeroddd, new TipoTelefone(tipo));
                                 ListTelefone.Add(telefone);
                             }
                             idPessoa = (Int32)reader["ID"];
                         }
                     }
              }
```

```
                return pessoa;
        }
        catch (Exception ex) {
                throw new Exception("Ocorreu um problema ao consultar Pessoa: " +
                        ex.Message);
        }
}
```

Classe Validação

```
using System;

namespace ClassPessoa.Controller{
        class Validacao  {
                public static Boolean isEmpty(int value) {
                        return value >= 0 ? true : false;
                }
                public static Boolean isEmpty(long value) {
                        return value >= 0 ? true : false;
                }
                public static Boolean isEmpty(string value) {
                        return value != "" ? true : false;
                }
        }
}
```

APÊNDICE B - CODIFICAÇÃO DAS CLASSES DE ENTIDADES

Classe PessoaDAO

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using ClassPessoa.Modal;
using System.Configuration;
namespace ClassPessoa.Controller {
    public static class PessoaDAO {
       //Atributos
       //Métodos privados
       //Métodos públicos
   }
}
```

Atributos

```
private static SqlConnection con;
```

Métodos Privados

```
private static SqlConnection Open() {
      try {
             String strConexao = ConfigurationManager
               .ConnectionStrings["dbPessoasConnectionString"].ConnectionString;
          if (con == null) {
              con = new SqlConnection(strConexao);
              con.Open();
          }
          return con;
      }
      catch (Exception ex) {
          throw new Exception("Não foi possivel abrir a conexão com a base de
        dados: " + ex.Message);
      }
}

private static void Close() {
      try {
          if (con != null) con.Close();
      }
      catch (Exception ex) {
          throw new Exception("Ocorreu um problema ao fechar a conexão com a
        base de dados: " + ex.Message);
      }
 }
```

```
private static SqlCommand getData(String QuerySQL) {
      try {
          SqlCommand command = new SqlCommand(QuerySQL, Open());
          return command;
      }
      catch (Exception ex) {
          throw new Exception("Ocorreu algum erro ao tentar acessar a base de
        dados: " + ex.Message);
      }
 }

 private static Boolean removeTelefone(Pessoa p) {
      try {
          string QueryTelefone = "";
          string QueryPessoaTelefone = "";
          List<Telefone> telefone = p.Telefone;

          foreach (Telefone item in telefone) {
              QueryTelefone = "DELETE FROM TELEFONE WHERE ID = @id";
              SqlCommand command = getData(QueryTelefone);
              command.Parameters.AddWithValue("@id", item.Id);
              command.ExecuteNonQuery();

              QueryPessoaTelefone = "DELETE FROM PESSOA_TELEFONE WHERE id_pessoa
              = @id_pessoa and id_telefone = @id_telefone";
              SqlCommand cmd = getData(QueryPessoaTelefone);
              cmd.Parameters.AddWithValue("@id_pessoa", p.Id);
              cmd.Parameters.AddWithValue("@id_telefone", item.Id);
              cmd.ExecuteNonQuery();

              if (con.State == System.Data.ConnectionState.Open) Close();
          }
          return true;
      }
      catch (Exception ex) {
          return false;
          throw new Exception("Ocorreu um problema ao remover telefone: " +
        ex.Message);
      }
  }


  private static Boolean removePessoa(Pessoa p) {
      try {
          string Query = "DELETE FROM PESSOA ID = @id";
          SqlCommand command = getData(Query);
          command.Parameters.AddWithValue("@id", p.Id);
          command.ExecuteNonQuery();

          if (con.State == System.Data.ConnectionState.Open) Close();
          return true;
      }
      catch (Exception ex) {
          return false;
          throw new Exception("Ocorreu um problema ao remover pessoa: " +
              ex.Message);
      }
  }
```

```
private static Boolean removeEndereco(Pessoa p) {
    try {
        string Query = "DELETE FROM ENDERECO ID = @id";
        SqlCommand command = getData(Query);
        command.Parameters.AddWithValue("@id", p.Endereco.Id);
        command.ExecuteNonQuery();

        if (con.State == System.Data.ConnectionState.Open) Close();
        return true;
    }
    catch (Exception ex) {
        return false;
        throw new Exception("Ocorreu um problema ao remover endereço: " +
     ex.Message);
    }
}


private static Boolean addTelefone(Pessoa p)  {
    try {
        string QueryTelefone = "";
        string QueryPessoaTelefone = "";
        List<Telefone> telefone = p.Telefone;

        foreach (Telefone item in telefone) {
            QueryTelefone = "INSERT INTO TELEFONE(numero, ddd, tipo)
            VALUES(@numero, @ddd, @tipo)";
            SqlCommand command = getData(QueryTelefone);
            command.Parameters.AddWithValue("@numero", item.Numero);
            command.Parameters.AddWithValue("@ddd", item.Ddd);
            command.Parameters.AddWithValue("@tipo", item.tipoTelefone.Id);
            command.ExecuteNonQuery();

            QueryPessoaTelefone  =  "INSERT  INTO  PESSOA_TELEFONE(id_pessoa,
            id_telefone) VALUES(@id_pessoa, @id_telefone)";
            SqlCommand cmd = getData(QueryPessoaTelefone);
            cmd.Parameters.AddWithValue("@id_pessoa", p.Id);
            cmd.Parameters.AddWithValue("@id_telefone", item.Id);
            cmd.ExecuteNonQuery();

            if (con.State == System.Data.ConnectionState.Open) Close();
        }
        return true;
    }
    catch (Exception ex) {
        return false;
        throw new Exception("Ocorreu um problema ao adicionar telefone: " +
     ex.Message);
    }
}
```

```
private static int addEndereco(Endereco endereco) {
    try {
        String Query = "INSERT INTO
            ENDERECO(logradouro,numero,cep,bairro,cidade,estado)
            VALUES(@logradouro,@numero,@cep,@bairro,@cidade,@estado;SELECT
            CAST(scope_identity() AS int)";

        SqlCommand Command = getData(Query);
        Command.Parameters.AddWithValue("@logradouro", endereco.Logradouro);
        Command.Parameters.AddWithValue("@numero", endereco.Numero);
        Command.Parameters.AddWithValue("@cep", endereco.Cep);
        Command.Parameters.AddWithValue("@bairro", endereco.Bairro);
        Command.Parameters.AddWithValue("@cidade", endereco.Cidade);
        Command.Parameters.AddWithValue("@estado", endereco.Estado);

        int idEndereco = (int)Command.ExecuteScalar();
        if (con.State == System.Data.ConnectionState.Open) Close();
        return idEndereco;
    }
    catch (Exception ex) {
        return 0;
        throw new Exception("Ocorreu um problema ao adicionar endereço: " +
      ex.Message);
    }
}


private static int updateEndereco(Endereco endereco) {
    try {
        String Query = ("INSERT INTO
            ENDERECO(logradouro,numero,cep,bairro,cidade,estado)
            VALUES(@logradouro,@numero,@cep,@bairro,@cidade,@estado;SELECT
            CAST(scope_identity() AS int)");

        SqlCommand Command = getData(Query);
        Command.Parameters.AddWithValue("@logradouro", endereco.Logradouro);
        Command.Parameters.AddWithValue("@numero", endereco.Numero);
        Command.Parameters.AddWithValue("@cep", endereco.Cep);
        Command.Parameters.AddWithValue("@bairro", endereco.Bairro);
        Command.Parameters.AddWithValue("@cidade", endereco.Cidade);
        Command.Parameters.AddWithValue("@estado", endereco.Estado);

        Command.Parameters.AddWithValue("@id", endereco.Id);
        Query = ("UPDATE ENDERECO SET logradouro=@logradouro, numero=@numero,
            cep=@cep, bairro=@bairro, cidade=@cidade, estado=@estado WHERE
            id=@id");
        Command.ExecuteNonQuery();
        if (con.State == System.Data.ConnectionState.Open) Close();
        return endereco.Id;
    }
    catch (Exception ex) {
        return 0;
        throw new Exception("Ocorreu um problema ao atualizar endereço: " +
      ex.Message);
    }
}
```

```
private static Boolean updatePessoa(Pessoa p) {
    try {
        String Query = ("UPDATE PESSOA SET nome = @nome, cpf=@cpf,
            endereco=@endereco WHERE id=@id");

        int idEndereco = p.Endereco.Id;

        SqlCommand Command = getData(Query);
        Command.Parameters.AddWithValue("@nome", p.Nome);
        Command.Parameters.AddWithValue("@cpf", p.CPF);
        Command.Parameters.AddWithValue("@endereco", idEndereco);
        Command.Parameters.AddWithValue("@id", p.Id);

        Command.ExecuteNonQuery();

        if (con.State == System.Data.ConnectionState.Open) Close();
            removeTelefone(p);
            addTelefone(p);
        return true;
    }
    catch (Exception ex) {
        return false;
        throw new Exception("Ocorreu um problema ao atualizar pessoa: " +
      ex.Message);
    }
}



private static Boolean addPessoa(Pessoa p) {
    try {
        String Query = ("INSERT INTO PESSOA(nome,cpf) VALUES(@nome,@cpf)");
        int idEndereco = addEndereco(p.Endereco);

        SqlCommand Command = getData(Query);
        Command.Parameters.AddWithValue("@nome", p.Nome);
        Command.Parameters.AddWithValue("@cpf", p.CPF);
        Command.Parameters.AddWithValue("@endereco", idEndereco);

        Command.ExecuteNonQuery();
        if (con.State == System.Data.ConnectionState.Open) Close();
        removeTelefone(p);
        addTelefone(p);
        return true;
    }
    catch (Exception ex) {
        return false;
        throw new Exception("Ocorreu um problema ao adicionar pessoa: " +
      ex.Message);
    }

}
```

Métodos Públicos

```
public static Boolean Insira(Pessoa p)  {
    try {
        return addPessoa(p);
    }
    catch (Exception ex) {
        return false;
        throw new Exception("Ocorreu um problema ao inserir Pessoa: " +
        ex.Message);
    }

}


public static Boolean altera(Pessoa p) {
    try {
        return updatePessoa(p);
    }
    catch (Exception ex) {
        return false;
        throw new Exception("Ocorreu um problema ao alterar Pessoa: " +
            ex.Message);
    }
}


public static Boolean excluir(Pessoa p) {
    try {
        if (removeTelefone(p)) {
            if (removeEndereco(p))
                removePessoa(p);
        }
        return true;
    }
    catch (Exception ex) {
        return false;
        throw new Exception("Ocorreu um problema ao excluir Pessoa: " +
            ex.Message);
    }
}
```

```
public static Pessoa consulta(long cpf = 0) {
    try {
        string sqlQuery = "";
        sqlQuery += "SELECT ";
        sqlQuery += "P.ID, P.NOME, P.CPF , ";
        sqlQuery += " E.ID ID_ENDERECO, E.LOGRADOURO, E.NUMERO, E.CEP,
                     E.BAIRRO,E.CIDADE, E.ESTADO , ";
        sqlQuery += "T.ID ID_TELEFONE, T.NUMERO TELEFONE, T.DDD, ";
        sqlQuery += "TT.ID ID_TIPO_ENDERECO, TT.TIPO ";
        sqlQuery += "FROM PESSOA AS P ";
        sqlQuery += "INNER JOIN ENDERECO AS E ON P.ENDERECO = E.ID ";
        sqlQuery += "INNER JOIN PESSOA_TELEFONE PT ON PT.ID_PESSOA = P.ID ";
        sqlQuery += "INNER JOIN TELEFONE T ON PT.ID_TELEFONE = T.ID ";
        sqlQuery += "INNER JOIN TELEFONE_TIPO TT ON T.TIPO = TT.ID  WHERE
                     CPF=" + cpf;
        Pessoa pessoa = new Pessoa();
        Endereco endereco = new Endereco();
        List<Telefone> ListTelefone = new List<Telefone>();
        Telefone telefone;
        SqlCommand command  = getData(sqlQuery);

        int idPessoa = 0;
        using (var reader = command.ExecuteReader()){
            if (reader.HasRows) {
                while (reader.Read()) {
                    endereco.Id = (Int32)reader["ID_ENDERECO"];
                    endereco.Logradouro = (String)reader["LOGRADOURO"];
                    endereco.Numero = (Int32)reader["NUMERO"];
                    endereco.Cep = (Int32)reader["CEP"];
                    endereco.Bairro = (String)reader["BAIRRO"];
                    endereco.Cidade = (String)reader["CIDADE"];
                    endereco.Estado = (String)reader["ESTADO"];

                    int idTelefone = (Int32)reader["ID_TELEFONE"];
                    int numeroTelefone = (Int32)reader["TELEFONE"];
                    int numeroddd = (Int32)reader["DDD"];
                    string tipo = (string)reader["TIPO"];
                    if (idPessoa != (Int32)reader["ID"]) {
                        ListTelefone = new List<Telefone>();
                        telefone = new Telefone(idTelefone, numeroTelefone,
                                        numeroddd, new TipoTelefone(tipo));
                        ListTelefone.Add(telefone);
                        pessoa.Id = (Int32)reader["ID"];
                        pessoa.Nome = (string)reader["NOME"];
                        pessoa.CPF = (Int64)reader["CPF"];
                        pessoa.Endereco = endereco;
                        pessoa.Telefone = ListTelefone;
                    }
                    else {
                        telefone = new Telefone(idTelefone, numeroTelefone,
                                        numeroddd, new TipoTelefone(tipo));
                        ListTelefone.Add(telefone);
                    }
                    idPessoa = (Int32)reader["ID"];
                }
            }

        }

        return pessoa;
    }
     catch (Exception ex) {
```

```
            throw new Exception("Ocorreu um problema ao consultar Pessoa: " +
          ex.Message);
        }
    }
```

Classe Validação

```
using System;
namespace ClassPessoa.Controller {
    class Validacao {
        public static Boolean isEmpty(int value) {
            return value >= 0 ? true : false;
        }
        public static Boolean isEmpty(long value) {
            return value >= 0 ? true : false;
        }
        public static Boolean isEmpty(string value) {
            return value != "" ? true : false;
        }
    }
}
```

APÊNDICE C – CÓDIGO ASPX DO LAYOUT – INDEX.ASPX

```
<%@ Page
    Language="C#"
    AutoEventWireup="true"
    MasterPageFile="~/Site1.Master"
    CodeBehind="index.aspx.cs"
    Inherits="webform.index"
    EnableEventValidation="false"
    Title="Sistema Cadastro de Pessoas" %>

<asp:Content ID="Content1"
    ContentPlaceHolderID="ContentPlaceHolder1"
    runat="Server">
    <strong>Localizar por CPF:
    </strong>
    <asp:Label ID="lblCPF"
        runat="server" Text=" "></asp:Label>
    <asp:TextBox ID="txtCPF"
        runat="server" TextMode="Number"></asp:TextBox>
    <asp:Button ID="bntConsultar"
        runat="server" Text="Consultar"
        OnClick="click" />
    <hr />
    <div class="row">

        <div class="title">
            <asp:Label ID="lblDados"
                runat="server" Text="Dados"></asp:Label>
        </div>

        <div class="col-md-12">
            <asp:GridView ID="GridView2"
                runat="server" CellPadding="4"
                ForeColor="#333333"
                GridLines="None"
                OnRowDataBound="GridView2_RowDataBound"
                Style="margin-right: 0px"
                AutoGenerateColumns="False"
                OnSelectedIndexChanged="GridView2_SelectedIndexChanged"
                Width="1183px">
                <AlternatingRowStyle
                    BackColor="White"
                    ForeColor="#284775" />
                <Columns>
                    <asp:BoundField DataField="ID"
                        ShowHeader="False" />
                    <asp:BoundField DataField="NOME"
                        HeaderText="Nome" />
                    <asp:BoundField DataField="CPF"
                        HeaderText="CPF" />
                    <asp:BoundField DataField="Endereco.Logradouro"
                        HeaderText="Logradouro" />
                    <asp:BoundField DataField="Endereco.Numero"
                        HeaderText="Nº" />
                    <asp:BoundField DataField="Endereco.Cep"
                        HeaderText="Cep" />
                    <asp:BoundField DataField="Endereco.Bairro"
                        HeaderText="Bairro" />
                    <asp:BoundField DataField="Endereco.Cidade"
                        HeaderText="Cidade" />
```

```
                <asp:BoundField DataField="Endereco.Estado"
                    HeaderText="Estado" />
                <asp:TemplateField ShowHeader="False">
                    <ItemTemplate>
                        <asp:LinkButton ID="LinkButton1"
                            runat="server" CausesValidation="false"
                            CommandName="" Text="Editar"
                            CssClass="btn btn-outline-success btn-sm">
                        </asp:LinkButton>
                    </ItemTemplate>
                </asp:TemplateField>
                <asp:TemplateField ShowHeader="False">
                    <ItemTemplate>
                        <asp:LinkButton ID="LinkButton2"
                            runat="server" CausesValidation="false"
                            CommandName="" Text="Excluir"
                            CssClass="btn btn-outline-danger btn-sm">
                       </asp:LinkButton>
                    </ItemTemplate>
                </asp:TemplateField>
            </Columns>
            <EditRowStyle BackColor="#999999" />
            <FooterStyle BackColor="#5D7B9D"
                ForeColor="White"
                Font-Bold="True" />
            <HeaderStyle BackColor="#5D7B9D"
                Font-Bold="True"
                ForeColor="White" />
            <PagerStyle BackColor="#284775"
                ForeColor="White"
                HorizontalAlign="Center" />
            <RowStyle BackColor="#F7F6F3"
                ForeColor="#333333" />
            <SelectedRowStyle BackColor="#E2DED6"
                Font-Bold="True"
                ForeColor="#333333" />
            <SortedAscendingCellStyle
                BackColor="#E9E7E2" />
            <SortedAscendingHeaderStyle
                BackColor="#506C8C" />
            <SortedDescendingCellStyle
                BackColor="#FFFDF8" />
            <SortedDescendingHeaderStyle
                BackColor="#6F8DAE" />
        </asp:GridView>
    </div>
    <div class="title">
        <asp:Label ID="lblContato"
            runat="server" Text="Contato"></asp:Label>
    </div>

    <div class="col-md-12">
        <asp:GridView ID="GridTelefones"
            CellPadding="4" ForeColor="#333333"
            runat="server" AutoGenerateColumns="False"
            Width="465px" ShowHeader="False">
            <AlternatingRowStyle
                BackColor="White"
                ForeColor="#284775" />
            <Columns>
                <asp:BoundField DataField="DDD"
                    HeaderText="DDD" />
```

```
                    <asp:BoundField DataField="NUMERO"
                        HeaderText="Número" />
                    <asp:BoundField DataField="TIPOTELEFONE.TIPO"
                        HeaderText="TIPO" />
                </Columns>
                <EditRowStyle BackColor="#999999" />
                <FooterStyle BackColor="#5D7B9D"
                    ForeColor="White"
                    Font-Bold="True" />
                <HeaderStyle BackColor="#5D7B9D"
                    Font-Bold="True"
                    ForeColor="White" />
                <PagerStyle BackColor="#284775"
                    ForeColor="White"
                    HorizontalAlign="Center" />
                <RowStyle BackColor="#F7F6F3"
                    ForeColor="#333333" />
                <SelectedRowStyle BackColor="#E2DED6"
                    Font-Bold="True"
                    ForeColor="#333333" />
                <SortedAscendingCellStyle
                    BackColor="#E9E7E2" />
                <SortedAscendingHeaderStyle
                    BackColor="#506C8C" />
                <SortedDescendingCellStyle
                    BackColor="#FFFDF8" />
                <SortedDescendingHeaderStyle
                    BackColor="#6F8DAE" />
            </asp:GridView>
        </div>
    </div>

</asp:Content>
```

APÊNDICE D – CÓDIGO ASPX DO LAYOUT – INDEX.ASPX.CS

```
using System;
using System.Configuration;
using ClassPessoa.Controller;
using ClassPessoa.Modal;
using System.Web.UI.WebControls;
using System.Collections.Generic;

namespace webform
{
    public partial class index : System.Web.UI.Page
    {
      private void showError(String msg)
        {
            Label lbl = (Label)Master.FindControl("lblErro");
            lbl.Text = msg;
            GridView2.Visible = false;
            GridTelefones.Visible = false;
            lblDados.Visible = false;
            lblContato.Visible = false;

        }
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                if (!IsPostBack)
                {
                    getListPessoa();
                }
            }
            catch (Exception ex)
            {
                showError(ex.Message);
            }
         }

        private List<Pessoa> getListPessoa(long cpf = 0)
        {
            List<Pessoa> ListPessoa = new List<Pessoa>();

            try
            {
                Pessoa pessoa = PessoaDAO.consulta(cpf);
                ListPessoa.Add(pessoa);
                GridView2.DataSource = ListPessoa;
                GridView2.DataBind();
                GridTelefones.DataSource = pessoa.Telefone;
                GridTelefones.DataBind();
                return ListPessoa;
            }
            catch (Exception ex)
            {
                showError(ex.Message);
                return ListPessoa;
            }
        }
```

```
        protected void click(object sender, EventArgs e)
        {
            try
            {
               long cpf = txtCPF.Text.Trim() != "" ? Convert.ToInt64(txtCPF.Text) : 0;
               getListPessoa(cpf);
            }
            catch (Exception ex)
            {
               showError(ex.Message);
            }

        }

        protected void GridView2_RowDataBound(object sender,
                    System.Web.UI.WebControls.GridViewRowEventArgs e)
        {
            if (e.Row.RowType == DataControlRowType.DataRow)
            {
                e.Row.Attributes["onclick"] =
                    Page.ClientScript.GetPostBackClientHyperlink(GridView2,   "Select$"
                    + e.Row.RowIndex);
                e.Row.Attributes["style"] = "cursor:pointer";
            }
        }

        protected void GridView2_SelectedIndexChanged(object sender, EventArgs e)
        {
            string cpf = GridView2.SelectedRow.Cells[2].Text;
            Response.Redirect("editar.aspx?CPF=" + cpf);
        }

    }


}
```

APÊNDICE E – CÓDIGO ASPX DO LAYOUT – EDITAR.ASPX

```
<%@ Page Title="Editar Pessoa"
    Language="C#"
    MasterPageFile="~/Site1.Master"
    AutoEventWireup="true"
    CodeBehind="editar.aspx.cs"
    Inherits="webform.editar" %>

<asp:Content ID="Content1"
    ContentPlaceHolderID="ContentPlaceHolder1"
    runat="server">
    <asp:HiddenField ID="txtID"
        runat="server" />
    <div class="row">
        <div class="col-md-6">
            <div class="row">
                <h3>PESSOA</h3>
                <div class="col-md-12">
                    <div class="form-group">
                        <label for="txtNome">Nome</label>
                        <asp:TextBox CssClass="form-control"
                            ID="txtNome"
                            runat="server"></asp:TextBox>
                    </div>
                </div>
                <div class="col-md-12">
                    <div class="form-group">
                        <label for="txtCPF">CPF</label>
                        <asp:TextBox ID="txtCPF"
                            runat="server" CssClass="form-control"></asp:TextBox>
                    </div>
                </div>
                <div class="col-md-12">
                    <div class="form-group">
                        <label>TELEFONE</label>
                        <asp:GridView ID="gridTelefones"
                            runat="server"
                            AutoGenerateColumns="False"
                            CellPadding="4"
                            ForeColor="#333333"
                            GridLines="Horizontal"
                            Width="100%" CaptionAlign="Top"
                            ShowHeader="False">
                            <AlternatingRowStyle
                                BackColor="White"
                                ForeColor="#284775" />
                            <Columns>
                                <asp:BoundField DataField="Ddd" />
                                <asp:BoundField DataField="Numero" />
                                <asp:BoundField DataField="tipoTelefone.Tipo" />
                                <asp:ButtonField Text="Excluir" />
                            </Columns>
                            <EditRowStyle BackColor="#999999" />
                            <FooterStyle BackColor="#5D7B9D"
                                Font-Bold="True"
                                ForeColor="White" />
                            <HeaderStyle BackColor="#5D7B9D"
                                Font-Bold="True"
                                ForeColor="White" />
                            <PagerStyle BackColor="#284775"
```

```
                        ForeColor="White"
                        HorizontalAlign="Center" />
                    <RowStyle BackColor="#F7F6F3"
                        ForeColor="#333333" />
                    <SelectedRowStyle BackColor="#E2DED6"
                        Font-Bold="True"
                        ForeColor="#333333" />
                    <SortedAscendingCellStyle
                        BackColor="#E9E7E2" />
                    <SortedAscendingHeaderStyle
                        BackColor="#506C8C" />
                    <SortedDescendingCellStyle
                        BackColor="#FFFDF8" />
                    <SortedDescendingHeaderStyle
                        BackColor="#6F8DAE" />
                </asp:GridView>
            </div>
        </div>
    </div>
    <div class="row">
        <h4>adcionar novo Telefone
        </h4>
        <div class="col-md-3">
            <asp:TextBox ID="txtDDD"
                CssClass="form-control"
                runat="server"></asp:TextBox>
        </div>
        <div class="col-md-3">
            <asp:TextBox ID="txtTelefone"
                CssClass="form-control"
                runat="server"></asp:TextBox>
        </div>
        <div class="col-md-3">
            <asp:DropDownList CssClass="form-control"
                ID="DropList" runat="Server">
                <asp:ListItem Text="Celular"
                    Value="1" />
                <asp:ListItem Text="Smartfone"
                    Value="2" />
                <asp:ListItem Text="Fixo"
                    Value="3" />
                <asp:ListItem Text="WhatsApp"
                    Value="4" />
            </asp:DropDownList>
        </div>
        <div class="col-md-3">
            <button class="btn btn-success">
                adicionar</button>
        </div>
    </div>
</div>
<div class="col-md-6">
    <div class="row">
        <h3>ENDEREÇO</h3>
        <div class="col-md-12">
            <label for="txtLogradouro">
                Logradouro</label>
            <asp:TextBox ID="txtLogradouro"
                CssClass="form-control"
                runat="server"></asp:TextBox>
        </div>
        <div class="col-md-12">
```

```
                    <label for="txtBairro">Bairro</label>
                    <asp:TextBox ID="txtBairro"
                        CssClass="form-control"
                        runat="server"></asp:TextBox>
                </div>
                <div class="col-md-3">
                    <label for="txtNumero">Número</label>
                    <asp:TextBox ID="txtNumero"
                        CssClass="form-control"
                        runat="server"></asp:TextBox>
                </div>
                <div class="col-md-3">
                    <label for="txtCEP">CEP</label>
                    <asp:TextBox ID="txtCEP"
                        CssClass="form-control"
                        runat="server"></asp:TextBox>
                </div>
                <div class="col-md-3">
                    <label for="txtEstado">Estado</label>
                    <asp:TextBox ID="txtEstado"
                        CssClass="form-control"
                        runat="server"></asp:TextBox>
                </div>
                <div class="col-md-3">
                    <label for="txtCidade">Cidade</label>
                    <asp:TextBox ID="txtCidade"
                        CssClass="form-control"
                        runat="server"></asp:TextBox>
                </div>
                <div class="col-md-12">
                    <hr />
                    <p>
                        <asp:Button ID="btnSalvar"
                            runat="server"
                            Text="Salvar Alterações"
                            OnClick="Salvar" />
                    </p>
                </div>
            </div>
        </div>
    </div>
</asp:Content>
```

APÊNDICE F – CÓDIGO ASPX DO LAYOUT – EDITAR.ASPX.CS

```
using System;
using System.Configuration;
using ClassPessoa.Controller;
using ClassPessoa.Modal;
using System.Data;
using System.Collections.Generic;
using Newtonsoft.Json.Serialization;
using Newtonsoft.Json;
using System.Web.UI.WebControls;

namespace webform
{
    public partial class editar : System.Web.UI.Page
    {
        public object JsonConvert { get; private set; }

        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                if (Request.QueryString["CPF"] != null && Request.QueryString["CPF"]
                    != string.Empty)
                {
                    long _cpf = long.Parse(Request.QueryString["CPF"]);
                    Pessoa  pessoa =  PessoaDAO.consulta(_cpf);
                    fillFields(pessoa);
                }
            }
            catch (Exception ex)
            {
                throw new Exception(ex.Message);
            }
        }

        private void fillFields(Pessoa p)
        {
            try
            {
                txtID.Value = p.Id.ToString();
                txtNome.Text = p.Nome;
                txtCPF.Text = p.CPF.ToString();
                txtLogradouro.Text = p.Endereco.Logradouro;
                txtNumero.Text = p.Endereco.Numero.ToString();
                txtCEP.Text = p.Endereco.Cep.ToString();
                txtBairro.Text = p.Endereco.Bairro;
                txtCidade.Text = p.Endereco.Cidade;
                txtEstado.Text = p.Endereco.Estado;

                gridTelefones.DataSource = p.Telefone;
                gridTelefones.DataBind();
            }
            catch (Exception ex)
            {
                throw new Exception(ex.Message);
            }

        }
```

```
        private Pessoa fillPessoa()
        {
            try
            {
                Pessoa ps = new Pessoa();

                ps.Nome =  txtNome.Text;
                ps.CPF = long.Parse(txtCPF.Text);
                ps.Id = int.Parse(txtID.Value);
                ps.Endereco.Logradouro = txtLogradouro.Text;
                ps.Endereco.Numero = int.Parse(txtNumero.Text);
                ps.Endereco.Cep = long.Parse(txtCEP.Text);
                ps.Endereco.Bairro = txtBairro.Text;
                ps.Endereco.Cidade = txtCidade.Text;
                ps.Endereco.Estado = txtEstado.Text;

                List<Telefone> telefone = gridTelefones.DataSource as List<Telefone>;
                ps.Telefone = telefone;

                return ps;
            }
            catch (Exception ex)
            {
                throw new Exception(ex.Message);
            }

        }
        protected void Salvar(object sender, EventArgs e)
        {
            try
            {
                Pessoa pessoa = fillPessoa();
                if (pessoa.Id > 0)
                {
                PessoaDAO.Insira(pessoa);
                }
                else
                {
                    PessoaDAO.altera(pessoa);
                }

            }
            catch (Exception ex)
            {
                throw new Exception(ex.Message);
            }

        }

    }
}
```

APÊNDICE G – CÓDIGO XML DO LAYOUT DA ACTIVITY

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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"
    tools:context=".MainActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="0dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TableLayout
                android:id="@+id/TableLayout1"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:layout_margin="10dp"
                android:overScrollMode="always"
                android:scrollbars="horizontal|vertical"
                android:stretchColumns="1">

                <TableRow
                    android:id="@+id/tableRowTitulo"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:gravity="center_horizontal">

                    <ImageView
                        android:id="@+id/imageView"
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:background="#CFD4D6"
                        app:srcCompat="@drawable/ic_stat_name"
                        tools:ignore="VectorDrawableCompat"/>

                    <TextView
                        android:id="@+id/Titulo"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_gravity="center_horizontal"
                        android:layout_span="0"
                        android:fontFamily="sans-serif-light"
                        android:foregroundGravity="center"
                        android:gravity="center_horizontal"
                        android:text="@string/stringPesquisar"
                        android:textColor="#000000"
                        android:textSize="22sp"/>


                </TableRow>
```

```
<TableRow
    android:id="@+id/tableRowPesquisa"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>

<TableRow
    android:id="@+id/tableRow2"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textViewCPF"
        android:layout_width="91dp"
        android:layout_height="wrap_content"
        android:gravity="right"
        android:text="@string/lblCPF"
        android:textColor="#000000"/>

    <EditText
        android:id="@+id/editTextCPF"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:textColor="#000000"
        android:textSize="18sp"
        tools:ignore="SpeakableTextPresentCheck"/>
</TableRow>

<TableRow
    android:id="@+id/tableRow0"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textViewNome"
        android:layout_width="91dp"
        android:layout_height="wrap_content"
        android:gravity="right"
        android:text="@string/lblNome"
        android:textColor="#000000"/>

    <EditText
        android:id="@+id/editTextNome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:textColor="#000000"
        android:textSize="18sp"
        tools:ignore="TouchTargetSizeCheck"/>
</TableRow>

<TableRow
    android:id="@+id/tableRow3"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textViewLogradouro"
        android:layout_width="91dp"
        android:layout_height="wrap_content"
        android:gravity="right"
```

```
                android:text="@string/lblLogradouro"
                android:textColor="#000000"/>

            <EditText
                android:id="@+id/editTextLogradouro"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:inputType="textPersonName"
                android:textColor="#000000"
                android:textSize="18sp"

tools:ignore="SpeakableTextPresentCheck,TouchTargetSizeCheck"/>
        </TableRow>

        <TableRow
            android:id="@+id/tableRow4"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/textViewBairro"
                android:layout_width="91dp"
                android:layout_height="wrap_content"
                android:gravity="right"
                android:text="@string/lblBairro"
                android:textColor="#000000"/>

            <EditText
                android:id="@+id/editTextBairro"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:inputType="textPersonName"
                android:textColor="#000000"
                android:textSize="18sp"
                tools:ignore="SpeakableTextPresentCheck"/>
        </TableRow>

        <TableRow
            android:id="@+id/tableRow5"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/textViewNumero"
                android:layout_width="91dp"
                android:layout_height="wrap_content"
                android:gravity="right"
                android:text="@string/lblNumero"
                android:textColor="#000000"/>

            <EditText
                android:id="@+id/editTextNumero"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:inputType="textPersonName"
                android:textColor="#000000"
                android:textSize="18sp"
                tools:ignore="TouchTargetSizeCheck"/>
        </TableRow>
```

```
                <TableRow
                    android:id="@+id/tableRow6"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent">

                    <TextView
                        android:id="@+id/textViewCEP"
                        android:layout_width="91dp"
                        android:layout_height="wrap_content"
                        android:gravity="right"
                        android:text="@string/lblCEP"
                        android:textColor="#000000"/>

                    <EditText
                        android:id="@+id/editTextCEP"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:inputType="textPersonName"
                        android:textColor="#000000"
                        android:textSize="18sp"
                        tools:ignore="TouchTargetSizeCheck"/>
                </TableRow>

                <TableRow
                    android:id="@+id/tableRow7"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent">

                    <TextView
                        android:id="@+id/textViewEstado"
                        android:layout_width="91dp"
                        android:layout_height="wrap_content"
                        android:gravity="right"
                        android:text="@string/lblEstado"
                        android:textColor="#000000"/>

                    <Spinner
                        android:id="@+id/spinnerEstados"
                        android:layout_width="91dp"
                        android:layout_height="wrap_content"
                        android:gravity="right"
                        android:scrollbarSize="18dp"
                        android:textColor="#000000"

tools:ignore="SpeakableTextPresentCheck,TouchTargetSizeCheck"/>


                </TableRow>
```

```
<TableRow
    android:id="@+id/tableRow8"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textViewCidade"
        android:layout_width="91dp"
        android:layout_height="wrap_content"
        android:gravity="right"
        android:text="@string/lblCidade"
        android:textColor="#000000"/>

    <EditText
        android:id="@+id/editTextCidade"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:textColor="#000000"
        android:textSize="18sp"
        tools:ignore="TouchTargetSizeCheck"/>
</TableRow>

<TableRow
    android:id="@+id/tableRow9"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textViewTelefone2"
        android:layout_width="91dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="20sp"
        android:layout_marginLeft="20sp"
        android:layout_span="2"
        android:gravity="left"
        android:text="@string/lblTelefone"
        android:textColor="#000000"/>
</TableRow>

<TableRow
    android:id="@+id/tableRow10"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textViewTelefone_1"
        android:layout_width="91dp"
        android:layout_height="wrap_content"
        android:layout_span="2"
        android:gravity="center_horizontal"
        android:text="11 97644-6899 - Pessoal  [ remover ]"
        android:textColor="#000000"/>
</TableRow>
```

```
                <TableRow
                    android:id="@+id/tableRow11"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent">

                    <TextView
                        android:id="@+id/textViewTelefone_2"
                        android:layout_width="91dp"
                        android:layout_height="wrap_content"
                        android:layout_span="2"
                        android:gravity="center_horizontal"
                        android:text="11 93644-5700 - Whatsapp  [ remover ]"
                        android:textColor="#000000"/>
                </TableRow>

                <TableRow
                    android:id="@+id/tableRow12"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent">

                    <TextView
                        android:id="@+id/textViewTelefone_2"
                        android:layout_width="91dp"
                        android:layout_height="wrap_content"
                        android:layout_span="2"
                        android:gravity="center_horizontal"
                        android:text="11 99231-8347 - Whatsapp  [ remover ]"
                        android:textColor="#000000"/>

                </TableRow>

                <TableRow
                    android:id="@+id/tableRow120"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:gravity="center_horizontal">

                    <Button
                        android:id="@+id/button"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_margin="10sp"
                        android:gravity="center_horizontal"
                        android:text="Novo"
                        app:backgroundTint="#4CAF50"/>

                    <Button
                        android:id="@+id/button"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_margin="10sp"
                        android:gravity="center_horizontal"
                        android:text="Salvar"/>
                </TableRow>
            </TableLayout>

        </LinearLayout>
    </ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
```