

ETEC – Ermelinda Giannini Teixeira

PLANO DE GERENCIAMENTO DE PROJETO

PROJETO SISTEMA LOCADORA

Versão: **2.0**
Data: **01/12/2023**

Responsável:

Nome: Eduarda Maciel Leite

RM 22048

Aprovado por:

André Linares e Cristiano Teixeira

**SANTANA DE PARNAÍBA
2023**

Sumário

Sumário

1. Resumo geral da proposta	3
2. Aplicações desenvolvidas	4
3. Cronograma	4
4. Diário	6
5. Desenvolvimento do Projeto.....	7
5.1 Passo a passo:.....	7
5.2 Design do Projeto	9
5.3 Código do programa.....	17
6. Resultado Obtido e considerações finais	103
7. Referências	104

1. Resumo geral da proposta

O Projeto Locadora tem como principal intuito incentivar a tarefa de programar aplicações cotidianas, além do uso exclusivo das linguagens C# e MySQL. Promovendo-nos conhecimento na área de Desenvolvimento de Sistemas, a tarefa utiliza da lógica, criatividade e raciocínio para operar um sistema funcional de uma Locadora de Carros.

A programação neste âmbito nos atribui habilidades de resolução de problemas e trabalho em equipe, visando uma proposta didática em que a turma se propõe a buscar as possíveis correções para falhas de um Sistema de Informação, novos utilitários e o manuseio de uma interface Windows Form.

De um modo geral, o projeto dispõe cadastros completos de entidades constituintes de uma empresa locadora, realização de alterações de registros, bem como suas exclusões, atribuindo um cunho administrativo em relação à gestão e operação de uma equipe.

Palavras-chave: Desenvolvimento de Sistemas, Lógica de Programação, C#, MySQL Workbench, Sistema de Informação, Locadora de Carros.

2. Aplicações desenvolvidas

Foi aplicado no Sistema Locadora em sua versão inicial e ao longo de sua execução, a princípio, diagramas de planejamento e métodos para sua análise e, posteriormente, a programação em linguagens C# e MySQL, enquanto necessária a integração de um Banco de Dados relacional.

O software consiste em um gerenciamento geral de uma empresa locadora de veículos, considerando os principais integrantes responsáveis por sua operação, tais como: Clientes, Funcionários, Automóveis, Cargos, Modelos, Marcas e suas próprias Locações.

É de suma importância a interação do Usuário com o Sistema de Gestão, visando uma automatização das tarefas cotidianas que atuam como meio de registro, mantendo assim uma comunicação benéfica entre Empresa e Cliente.

Portanto, foram adotados os artifícios Create (criar), Read (ler), Update (atualizar) e Delete (apagar), e, por conseguinte, abrangendo formulários de buscas eficientes, uma vez que facilitam a manipulação dos Dados inseridos no contexto. Além disso, possui mecanismos que mantêm explícito o quanto proveitoso é o uso da tecnologia nesse setor e seus benefícios no ambiente corporativo, como por exemplo, o manuseamento de caixas de seleção, que permitem poupar o tempo destinado à digitação do Cliente e o processo tardio das informações.

Entretanto, o Projeto Locadora nos promove a visualização de um modelo de plataforma dinâmica, desempenhando um papel fundamental ao Usuário em que se aplica.

3. Cronograma

Nome da atividade: Início e Planejamento do Projeto;

Descrição: Início do Projeto Locadora;

Data: 29/07/2023

Objetivo previsto a ser alcançado: Elaboração do Projeto Locadora e suas vertentes, tais como o Diagrama de Casos de Uso, Questionário de Planejamento e Projeção destinado ao usuário, Banco de Dados e Requisitos de Usuário/Sistema.

Status: Finalizado.

Nome da atividade: Entrega Parcial do Projeto para Avaliação;

Descrição: Entrega dos formulários de Gerenciamento de cadastro e Pesquisa de cargo.

Data Prevista: 29/09/2023

Objetivo previsto a ser alcançado: A entrega parcial, tendo sido efetivados os principais formulários de Cadastro, entretanto ainda em processo de desenvolvimento.

Status: Finalizado.

Nome da atividade: Entrega Integral do Projeto para Avaliação;

Descrição: Entrega dos formulários de Gerenciamento de cadastro e Pesquisa das Entidades presentes na Locadora.

Data Prevista: 01/12/2023

Objetivo previsto a ser alcançado: A entrega integral, tendo sido efetivados os formulários de Cadastro e Pesquisa, contudo, o projeto finalizado.

Status: Finalizado.

4. Diário

Data: Julho/2023

Atividades desenvolvidas: Elaboração do Questionário de Planejamento de projeto, visando aprimorar as metas que serão atingidas ao longo do processo de desenvolvimento, tendo em vista requisitos solicitados pelo usuário.

Integrante: Eduarda Maciel Leite; **Atividades desenvolvida:** Tarefa de Análise e Projeto de Sistemas;

Data: Agosto/2023

Atividades desenvolvidas: Desenvolvimento dos Requisitos de Usuário e Sistema, do mesmo modo que o Modelo de Diagrama de Casos de Uso (UML).

Integrante: Eduarda Maciel Leite; **Atividades desenvolvida:** Tarefa de Análise e Projeto de Sistemas;

Data: Agosto/2023

Atividades desenvolvida: Com base no Modelo de Entidade e Relacionamento (MER) requerido, foi feita a construção do Banco e Inserção de Dados na plataforma MySQL Workbench.

Integrante: Eduarda Maciel Leite; **Atividades desenvolvida:** Banco de Dados;

Data: Agosto/2023

Atividades desenvolvida: Escolha da Paleta de Cores e realização do Design da tela inicial do Sistema na plataforma Canva.

Integrante: Eduarda Maciel Leite; **Atividades desenvolvida:** Design Digital;

Data: Agosto-Setembro/2023

Atividades desenvolvida: Início do desenvolvimento dos formulários de formato Windows Forms no Visual Studio 2019, com o foco voltado na finalização do Design integrado nas diversas páginas.

Integrante: Eduarda Maciel Leite; **Atividades desenvolvida:** Design Digital – Visual Studio 2019;

Data: Setembro/2023

Atividades desenvolvida: Início da programação de conexão entre o Banco de Dados e os formulários, bem como a realização das operações Create, Read, Update e Delete.

Integrante: Eduarda Maciel Leite; **Atividades desenvolvida:** Desenvolvimento do Sistema Locadora – Visual Studio 2019;

Data: 29 de Setembro/2023

Atividades desenvolvida: Realização da documentação para avaliação parcial do Projeto Locadora.

Integrante: Eduarda Maciel Leite; **Atividades desenvolvida:** Documentação do Projeto Locadora;

Data: 02 de Outubro/2023

Atividades desenvolvida: Entrega Parcial

Integrante: Eduarda Maciel Leite;

Data: 01 de Dezembro/2023

Atividades desenvolvida: Entrega Integral

Integrante: Eduarda Maciel Leite;

5. Desenvolvimento do Projeto

5.1 Passo a passo:

1. Criação do Questionário de Planejamento do Projeto Locadora

O questionário é uma das ferramentas iniciais na criação de um Projeto de Sistema. É nele que se são estabelecidas as primeiras regras de negócios, definidas, a princípio, para que as expectativas do cliente sejam sanadas de acordo com suas necessidades.

Para isso, é fundamental que o usuário seja entrevistado seguindo os parâmetros de como, o quê, quando e por que deverá ser realizado um software.

2. Definição de Requisitos de Usuário e de Sistema

É de suma importância estabelecer de forma concreta as funcionalidades que se farão presentes, as quais são denominadas Requisitos de Usuário e de Sistema. Os Requisitos de Usuário são os solicitados pelo cliente, enquanto os de Sistema são aqueles quais só serão possíveis se estabelecidos pelo Desenvolvedor, bem como examinados pelo profissional Analista de Sistemas.

3. Desenvolvimento do Diagrama de Casos de Uso (UML)

O Diagrama de Casos de Uso dispõe uma visão mais clara da relação entre o Ator (Usuário que executará ações no software) e as funções desempenhadas pelo sistema, de forma que ilustra detalhadamente pelos métodos de *include*, *extend*, *generalização*, dentre outros presentes nesta tarefa.

Esclarecer as metas de um software por meios visuais implicam o conhecimento destinado a públicos mais abrangentes, que interpretando o diagrama, serão possibilitados de determinar o que foi requestado.

4. Criação do Banco de Dados

Para criar um Banco de Dados será requerido a definição das entidades presentes no Sistema Locadora, deste modo, identificando quais os objetos/pessoas que sofrerão atualizações mediante a ele.

Nesse caso, as entidades estipuladas são: Cliente, Funcionário, Automóvel, Cargo, Marca, Modelo e Locação; estes serão cadastrados na Base de Dados depois que formulada na plataforma MySQL.

5. Definição da Paleta de Cores e Design Digital do Sistema

A Paleta de Cores é a parte crucial de layout de um sistema de informação, é nela que a precisão da identidade visual será obtida. Portanto, as cores selecionadas para o Projeto Locadora foram o preto e vermelho, visto que trazem a sensação de ambiente automotivo e que, juntas, formam um contraste um tanto quanto chamativo.

O restante do design foi designado pela aplicação Visual Studio 2019, com a escolha voltada a fontes claras e visualmente não poluídas. Além disso, os botões e destaques presentes nas telas não deixam de ser expostos com as cores alcançadas na paleta, para uma combinação otimizada e aspecto límpido.

6. Programação e Desenvolvimento das Conexões com a Base de Dados e Windows Forms

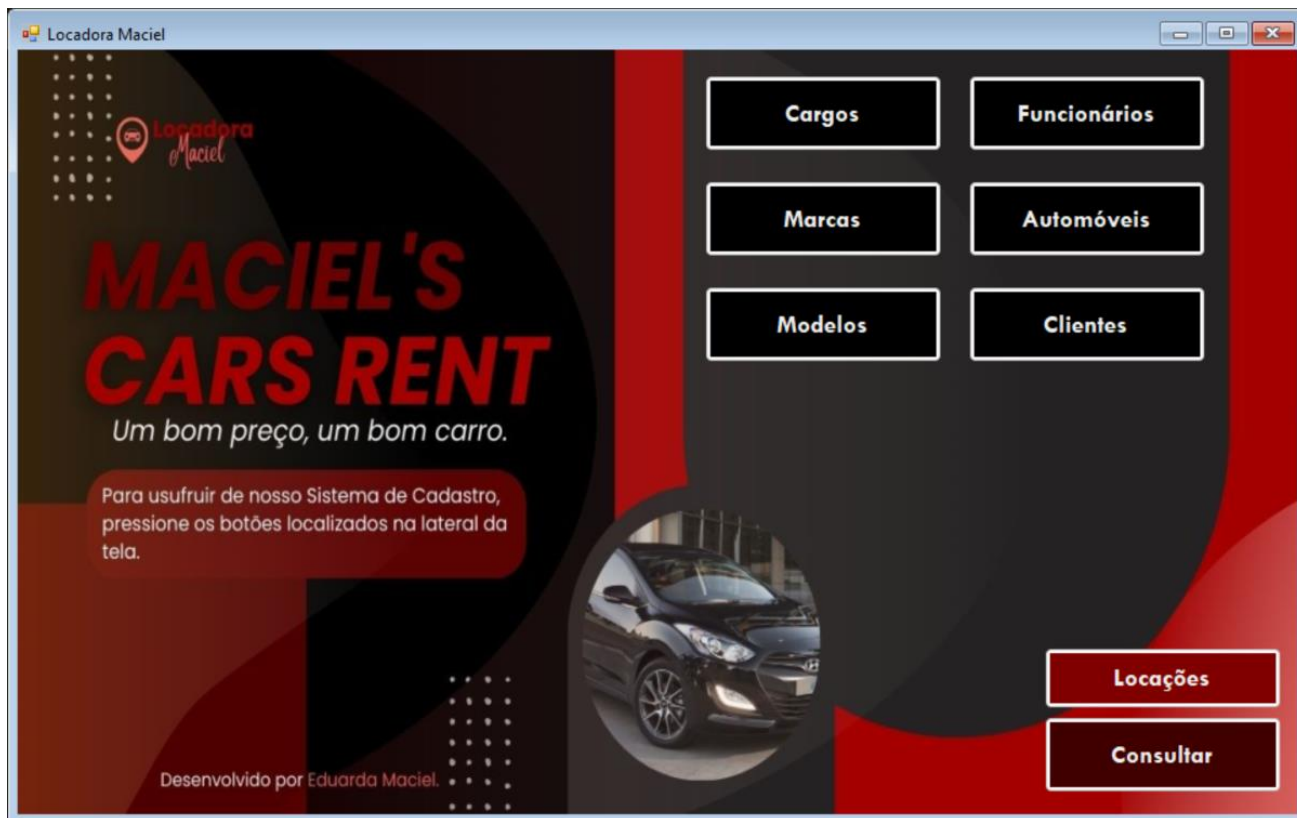
A escolha da linguagem de programação é obtida a partir das necessidades de um software, visando praticidade tanto para o desenvolvedor quanto para o utilizador da aplicação.

O Sistema Locadora foi integralmente produzido em linguagem C# através da plataforma Visual Studio 2019, destacada pela otimização de tempo voltado à

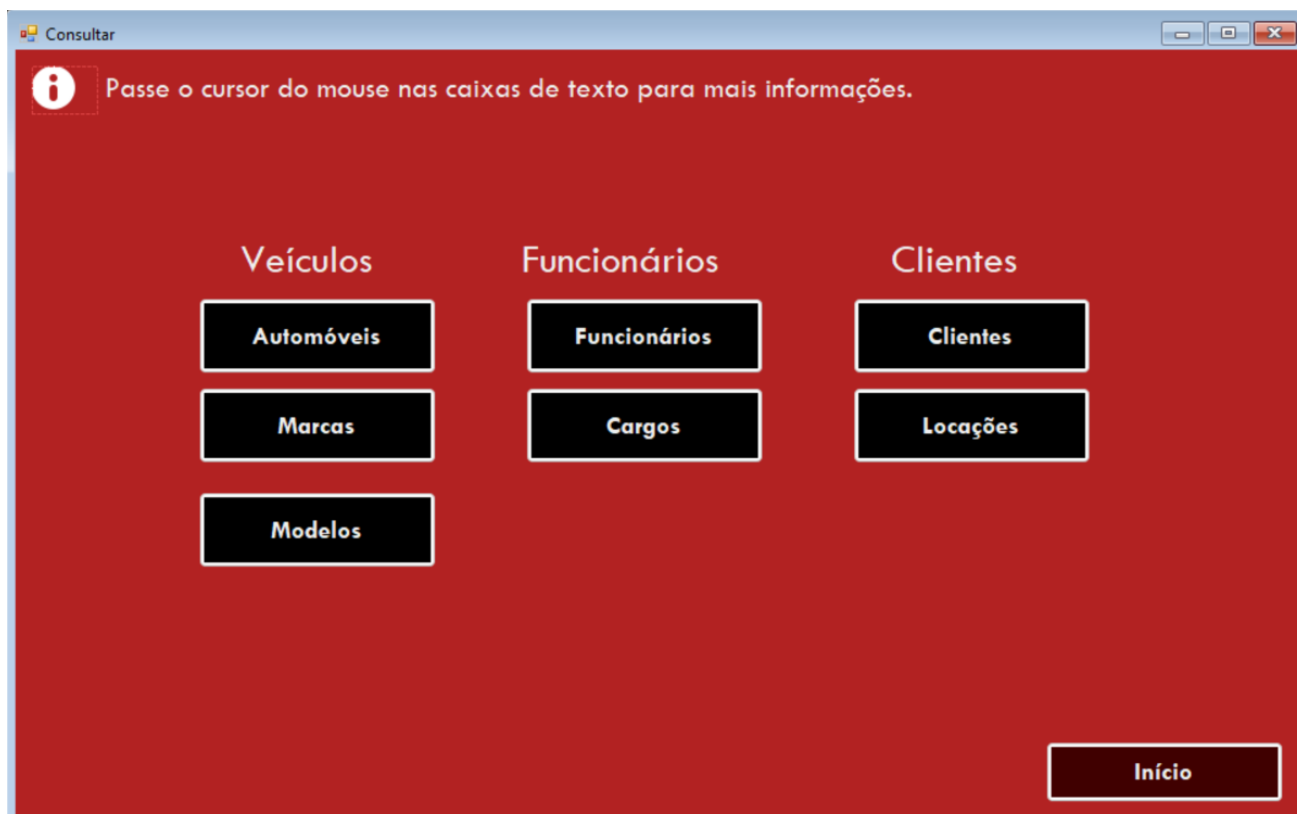
interface, justamente pela possibilidade de ajuste aos elementos presentes na tela, uma vastidão de Eventos programáveis, entre outras configurações exequíveis.

5.2 Design do Projeto

- FrmLogin.cs



- Consulta.cs



- FrmAutomovel.cs

Gerenciamento de Automóveis

Gerenciamento de Automóveis

Dados do Automóvel

Código:

Nome:

Ano Fabricação:



Cor:

KM:

Status:

Valor:

Marca **Modelo**

Código:  Código: 

Cadastrar **Atualizar** **Excluir** **Limpar**

Automóveis e Status

Início

Pesquisar

- FrmCargo.cs

Gerenciamento de Cargos

Gerenciamento de Cargos

Cadastro

Código:

Nome da Ocupação:

Cadastrar **Atualizar** **Excluir** **Limpar**

Cargos Disponíveis

Início

Pesquisar

- FrmCliente.cs

Gerenciamento de Clientes

Gerenciamento de Clientes

Informações Pessoais

Código:

Nome:

Telefone:

Sexo:

Email:

Data Nascimento:

Data Cadastro:

Dados Residenciais

Endereço:

Complemento: Bairro:

Cidade: UF:

Cadastrar **Atualizar** **Excluir** **Limpar**

Gerar Senha

Senha:

Gerar **Pesquisar**

Início

- FrmFuncionario.cs

Gerenciamento de Funcionários

Gerenciamento de Funcionário

Informações Pessoais

Código:

Nome:

Telefone:

Cargo

Código do Cargo:

Data Contratação:

Cadastrar **Atualizar** **Excluir** **Limpar**

Pesquisar **Início**

- FrmLocacao.cs

Gerenciamento de Locações

Gerenciamento de Locações

Cadastro

Código:

Tipo:

Valor:

Período

Data Início:

Data Fim:

Cliente **Funcionário** **Automóvel**

Código:

Código:

Código:

Cadastrar **Atualizar** **Excluir** **Limpar** **Pesquisar** **Início**

- FrmMarca.cs

Gerenciamento de Marcas

Gerenciamento de Marcas

Cadastro

Código:

Nome da Marca:

Cadastrar **Atualizar** **Excluir** **Limpar**

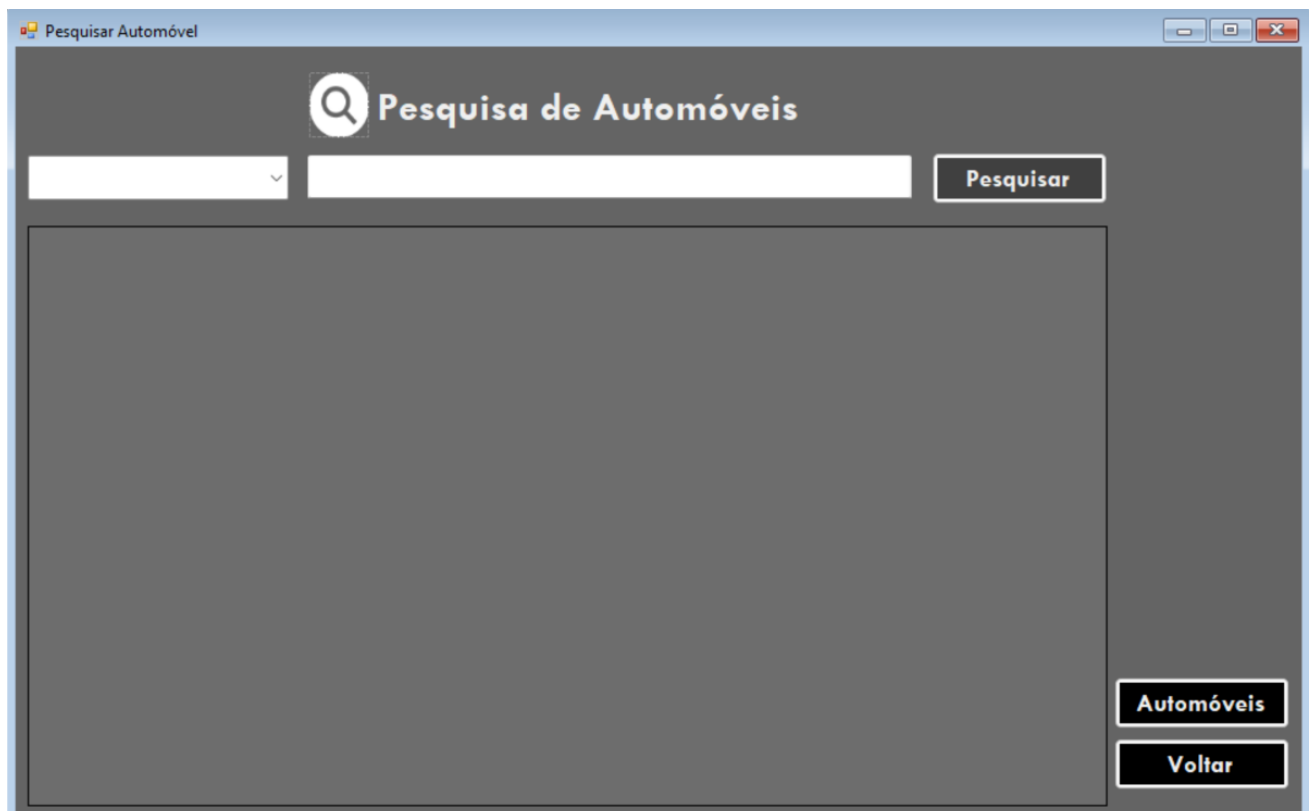
Marcas Disponíveis

Início **Pesquisar**

- FrmModelo.cs



- PesquisaAutomovel.cs



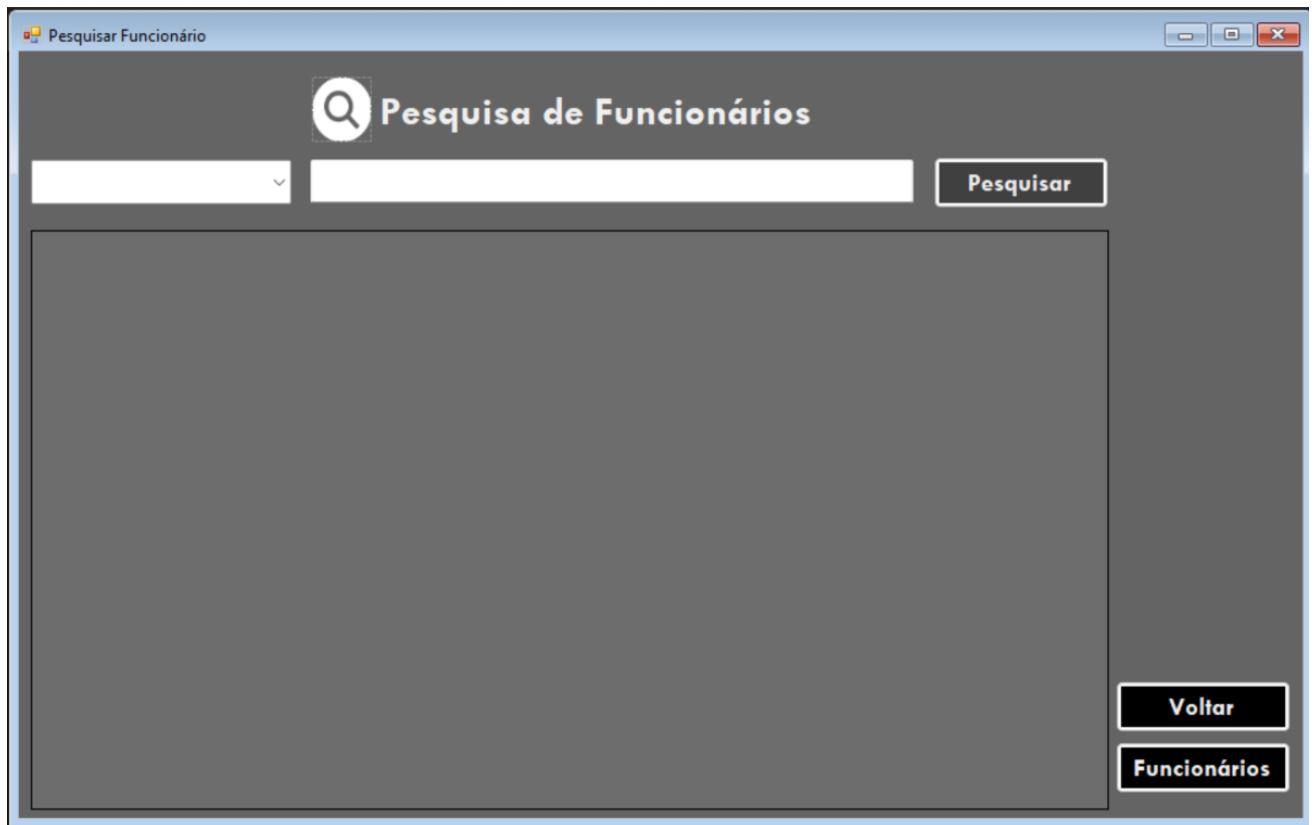
- PesquisaCargo.cs



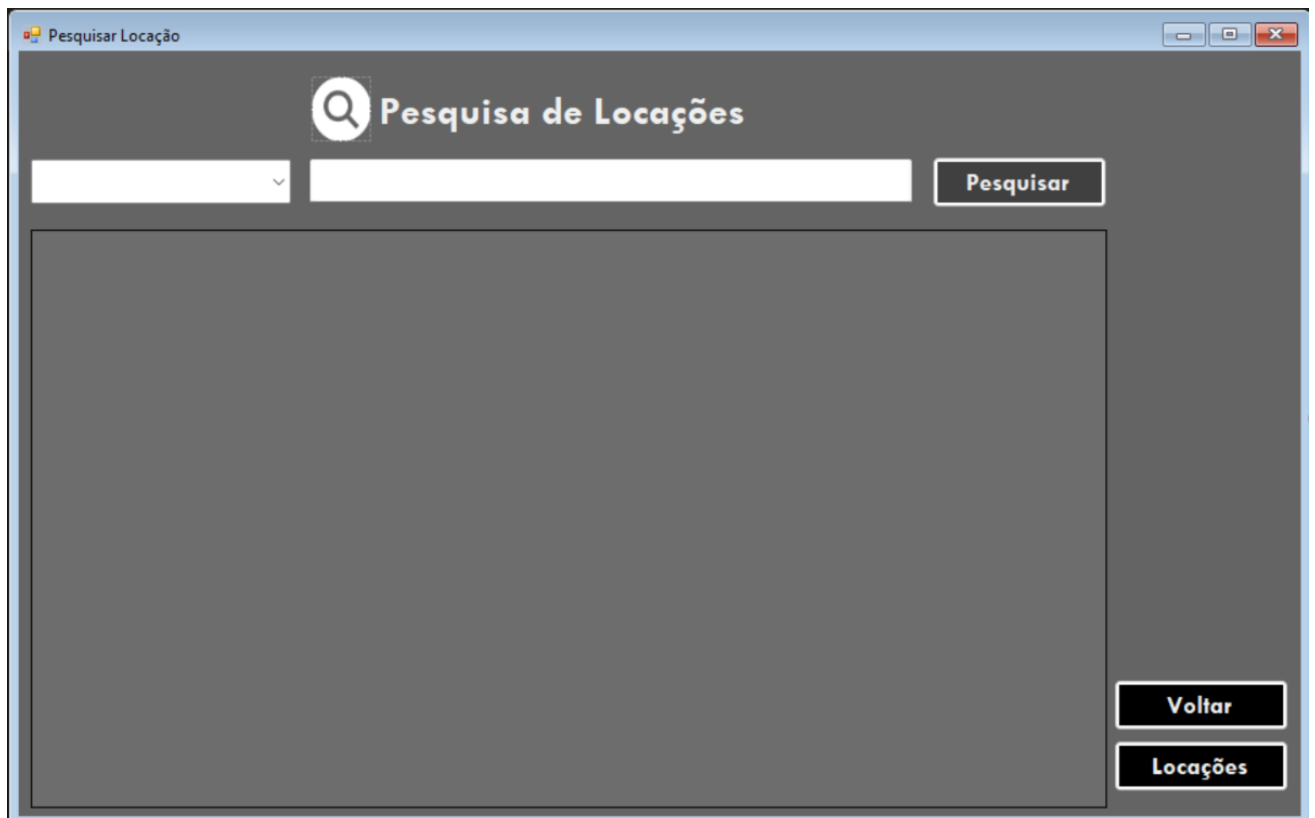
- PesquisaCliente.cs



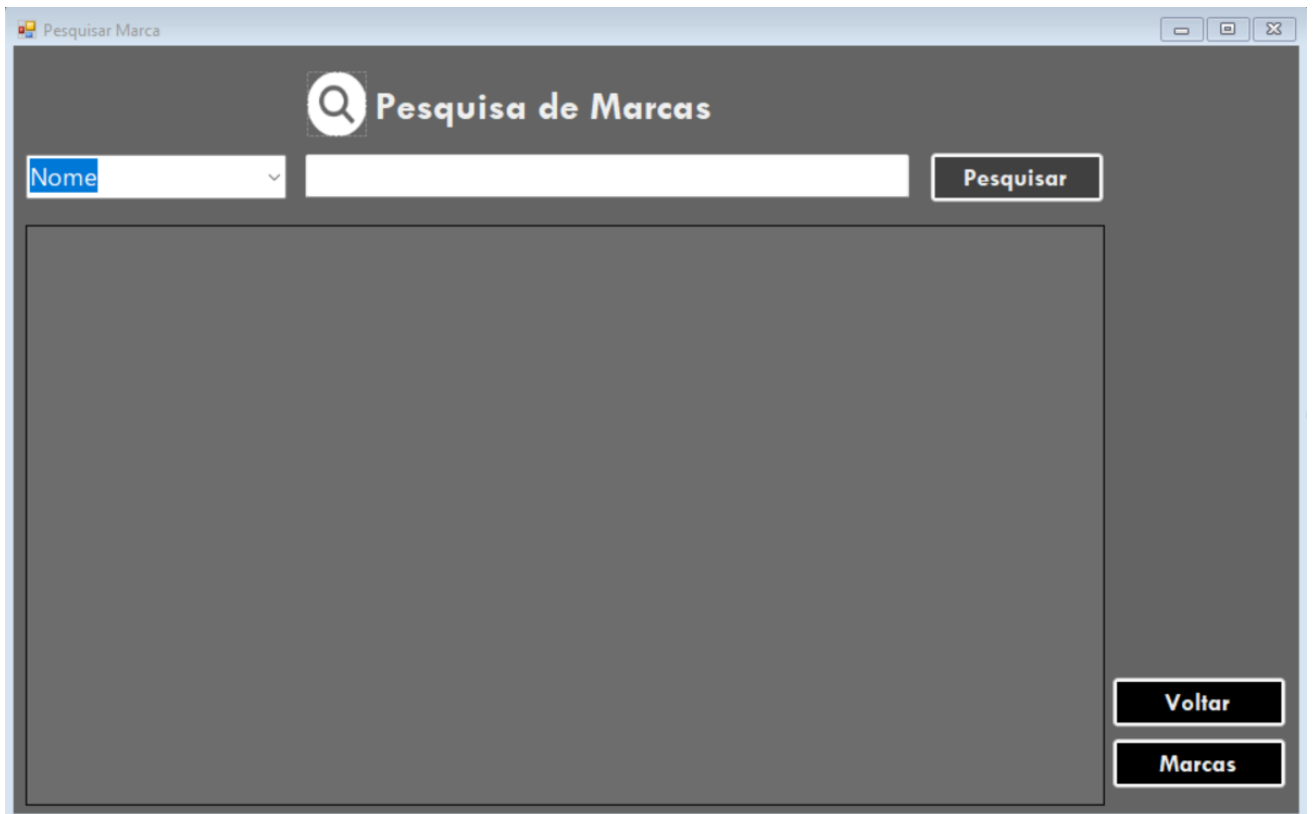
- PesquisaFuncionario.cs



- PesquisaLocacao.cs



- PesquisaMarca.cs



- PesquisaModelo.cs



5.3 Código do programa

- **App.Config**

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>

  <connectionStrings>

    <add name ="BD_LOCADORA_2MTEC_2023"
      connectionString="server=localhost;
user ID = root;
password = root;
database = BD_LOCADORA_2MTEC_2023"

      providerName="MySql.Data.MySqlClient" />

  </connectionStrings>
</configuration>
```

- **Program.cs**

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>

  <connectionStrings>

    <add name ="BD_LOCADORA_2MTEC_2023"
      connectionString="server=localhost;
user ID = root;
password = root;
database = BD_LOCADORA_2MTEC_2023"

      providerName="MySql.Data.MySqlClient" />

  </connectionStrings>
</configuration>
```

- **Consultar.cs**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ProjetoLocadora
{
    public partial class Consultar : Form
    {
        public Consultar()
        {
            InitializeComponent();
        }

        private void btncargos_Click(object sender, EventArgs e)
        {
            this.Hide();
            PesquisaCargo pesquisaCargo = new PesquisaCargo();
            pesquisaCargo.FormClosed += (s, args) => this.Close();
            pesquisaCargo.ShowDialog();
        }

        private void btnmarcas_Click(object sender, EventArgs e)
        {
            this.Hide();
            PesquisaMarca pesquisaMarca = new PesquisaMarca();
            pesquisaMarca.FormClosed += (s, args) => this.Close();
            pesquisaMarca.ShowDialog();
        }

        private void btnmodelos_Click(object sender, EventArgs e)
        {
            this.Hide();
            PesquisaModelo pesquisaModelo = new PesquisaModelo();
            pesquisaModelo.FormClosed += (s, args) => this.Close();
            pesquisaModelo.ShowDialog();
        }

        private void btnclientes_Click(object sender, EventArgs e)
        {
            this.Hide();
            PesquisaCliente pesquisaCliente = new PesquisaCliente();
            pesquisaCliente.FormClosed += (s, args) => this.Close();
            pesquisaCliente.ShowDialog();
        }

        private void btnfuncionarios_Click(object sender, EventArgs e)
        {

```

```

{
    this.Hide();
    PesquisaFuncionario pesquisaFuncionario = new PesquisaFuncionario();
    pesquisaFuncionario.FormClosed += (s, args) => this.Close();
    pesquisaFuncionario.ShowDialog();
}

```

```

private void btnautomoveis_Click(object sender, EventArgs e)
{
    this.Hide();
    PesquisaAutomovel pesquisaAutomovel = new PesquisaAutomovel();
    pesquisaAutomovel.FormClosed += (s, args) => this.Close();
    pesquisaAutomovel.ShowDialog();
}

```

```

private void btnlocacoes_Click(object sender, EventArgs e)
{
    this.Hide();
    PesquisaLocacao pesquisaLocacao = new PesquisaLocacao();
    pesquisaLocacao.FormClosed += (s, args) => this.Close();
    pesquisaLocacao.ShowDialog();
}

```

```

private void btnvoltar_Click(object sender, EventArgs e)
{
    this.Hide();
    FrmLogin login = new FrmLogin();
    login.FormClosed += (s, args) => this.Close();
    login.ShowDialog();
}

```

```

//Configurações TOOLTIP label "Automóveis"
private void label16_MouseHover(object sender, EventArgs e)
{
    toolTip1.SetToolTip(label16, "Automóveis disponíveis para locação.");
}

```

```

//Configurações TOOLTIP label "Funcionários"
private void label1_MouseHover(object sender, EventArgs e)
{
    toolTip1.SetToolTip(label1, "Funcionários e cargos disponíveis.");
}

```

```

//Configurações TOOLTIP label "Clientes"
private void label2_MouseHover(object sender, EventArgs e)
{
    toolTip1.SetToolTip(label2, "Clientes que já possuem cadastro.");
}

```

```

//Configurações TOOLTIP label "Clientes"
private void pictureBox1_MouseHover(object sender, EventArgs e)
{
    toolTip1.SetToolTip(pictureBox1, "As informações aparecerão desta forma!");
}

```

```
}  
  
}  
}
```

- **FrmAutomovel.cs**

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Configuration;  
using MySql.Data;  
using MySql.Data.MySqlClient;  
  
namespace ProjetoLocadora  
{  
    public partial class FrmAutomovel : Form  
    {  
  
        // Criar variável conexão  
        string conexao =  
ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionStr  
ing;  
  
        public FrmAutomovel()  
        {  
            InitializeComponent();  
        }  
  
  
        //Configurações dos botões exibidos no Formulário  
        private void btnvoltar_Click_1(object sender, EventArgs e)  
        {  
            this.Hide();  
            FrmLogin frmLogin = new FrmLogin();  
            frmLogin.ShowDialog();  
        }  
  
        private void btnpesquisar_Click(object sender, EventArgs e)  
        {  
            PesquisaAutomovel automovel = new PesquisaAutomovel();  
            automovel.ShowDialog();  
        }  
    }  
}
```

```

private void pbmarca_Click(object sender, EventArgs e)
{
    PesquisaMarca pesquisamarca = new PesquisaMarca();
    pesquisamarca.ShowDialog();
}

private void pbmodelo_Click(object sender, EventArgs e)
{
    PesquisaModelo pesquisamodelo = new PesquisaModelo();
    pesquisamodelo.ShowDialog();
}

// Configurações do botão CADASTRAR
private void btncadastrar_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    string nome;
    string cor;
    string status;

    int anofab;
    double valor;
    double km;

    string marca;
    string modelo;

    nome = txtnome.Text;
    cor = txtcor.Text;
    status = cmbstatus.Text;

    anofab = int.Parse(txtanofab.Text);
    valor = double.Parse(txtvalor.Text);
    km = double.Parse(txtkm.Text);
    marca = txtmarca.Text;
    modelo = txtmodelo.Text;

    if (nome == string.Empty || cor == string.Empty || marca == string.Empty || modelo
== string.Empty)
    {
        MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o corretamente.");
    }

    else
    {

```

```
MessageBox.Show("Cadastro efetuado com sucesso!");
```

```
txtcodigo.Text = String.Empty;  
txtnome.Text = String.Empty;  
txtcor.Text = String.Empty;  
cmbstatus.Text = String.Empty;  
txtanofab.Text = String.Empty;  
txtvalor.Text = String.Empty;  
txtkm.Text = String.Empty;  
txtmarca.Text = String.Empty;  
txtmodelo.Text = String.Empty;
```

```
string sql_insert = @"insert into TB_AUTOMOVEL
```

```
(TB_AUTOMOVEL_NOME,  
TB_AUTOMOVEL_ANO_FAB,  
TB_AUTOMOVEL_COR,  
TB_AUTOMOVEL_KM,  
TB_AUTOMOVEL_VALOR_D,  
TB_AUTOMOVEL_STATUS,  
TB_MARCA_ID,  
TB_MODELO_ID)
```

```
values (@AUTOMOVEL_NOME,  
@AUTOMOVEL_ANO_FAB,  
@AUTOMOVEL_COR,  
@AUTOMOVEL_KM,  
@AUTOMOVEL_VALOR_D,  
@AUTOMOVEL_STATUS,  
@MARCA_ID,  
@MODELO_ID);
```

```
MySqlCommand executacmdMySql_insert = new MySqlCommand(sql_insert,  
con);  
  
executacmdMySql_insert.Parameters.AddWithValue("@AUTOMOVEL_NOME",  
nome);  
  
executacmdMySql_insert.Parameters.AddWithValue("@AUTOMOVEL_ANO_FAB",  
anofab);  
executacmdMySql_insert.Parameters.AddWithValue("@AUTOMOVEL_COR",  
cor);  
executacmdMySql_insert.Parameters.AddWithValue("@AUTOMOVEL_KM", km);  
  
executacmdMySql_insert.Parameters.AddWithValue("@AUTOMOVEL_VALOR_D", valor);  
  
executacmdMySql_insert.Parameters.AddWithValue("@AUTOMOVEL_STATUS", status);  
executacmdMySql_insert.Parameters.AddWithValue("@MARCA_ID", marca);  
executacmdMySql_insert.Parameters.AddWithValue("@MODELO_ID", modelo);
```

```

con.Open();
executacmdMySql_insert.ExecuteNonQuery();

string sql_select_automovel = "select * from TB_AUTOMOVEL";

MySqlCommand executacmdMySql_select_automovel = new
MySqlCommand(sql_select_automovel, con);
executacmdMySql_select_automovel.ExecuteNonQuery();

DataTable tabela_automovel = new DataTable();

MySqlDataAdapter da_automovel = new
MySqlDataAdapter(executacmdMySql_select_automovel);
da_automovel.Fill(tabela_automovel);

dgvautomovel.DataSource = tabela_automovel;

con.Close();
}

```

```

}

```

```

//Configuração botão ATUALIZAR

```

```

private void btnatualizar_Click(object sender, EventArgs e)
{

```

```

    btncadastrar.Enabled = true;

```

```

    int codigo;
    string nome;
    string cor;
    string status;

```

```

    int anofab;
    double valor;
    double km;

```

```

    string marca;
    string modelo;

```

```

    codigo = int.Parse(txtcodigo.Text);
    nome = txtnome.Text;
    cor = txtcor.Text;
    status = cmbstatus.Text;

```

```

    anofab = int.Parse(txtanofab.Text);
    valor = double.Parse(txtvalor.Text);

```

```

km = double.Parse(txtkm.Text);
marca = txtmarca.Text;
modelo = txtmodelo.Text;

if (nome == string.Empty || cor == string.Empty || marca == string.Empty || modelo
== string.Empty)
{
    MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o corretamente.");
}

else
{
    DialogResult respostaeditar;

    respostaeditar = MessageBox.Show("Deseja editar o registro selecionado? Os
dados serão permanentemente alterados e não será possível recuperá-los!", "",
    MessageBoxButtons.YesNo, MessageBoxIcon.Information,
    MessageBoxDefaultButton.Button2);

    if (respostaeditar == DialogResult.Yes)
    {

        MySqlConnection con = new MySqlConnection(conexao);

        string sql_update_automovel = @"update TB_AUTOMOVEL
set TB_AUTOMOVEL_NOME = @AUTOMOVEL_NOME,
    TB_AUTOMOVEL_ANO_FAB = @AUTOMOVEL_ANO_FAB,
    TB_AUTOMOVEL_COR = @AUTOMOVEL_COR,
    TB_AUTOMOVEL_KM= @AUTOMOVEL_KM,
    TB_AUTOMOVEL_VALOR_D = @AUTOMOVEL_VALOR_D,
    TB_AUTOMOVEL_STATUS= @AUTOMOVEL_STATUS,
    TB_MARCA_ID = @MARCA_ID,
    TB_MODELO_ID = @MODELO_ID
    where TB_AUTOMOVEL_ID = @id";

        con.Open();

        MySqlCommand executacmdMySql_update_automovel = new
        MySqlCommand(sql_update_automovel, con);

        executacmdMySql_update_automovel.Parameters.AddWithValue("@id",
        codigo);

        executacmdMySql_update_automovel.Parameters.AddWithValue("@AUTOMOVEL_NOME
", nome);

        executacmdMySql_update_automovel.Parameters.AddWithValue("@AUTOMOVEL_ANO_
FAB", anofab);

        executacmdMySql_update_automovel.Parameters.AddWithValue("@AUTOMOVEL_COR",
        cor);

```



```

executacmdMySql_update_automovel.Parameters.AddWithValue("@AUTOMOVEL_KM",
km);

executacmdMySql_update_automovel.Parameters.AddWithValue("@AUTOMOVEL_VALOR_D", valor);

executacmdMySql_update_automovel.Parameters.AddWithValue("@AUTOMOVEL_STATUS", status);

executacmdMySql_update_automovel.Parameters.AddWithValue("@MARCA_ID", marca);

executacmdMySql_update_automovel.Parameters.AddWithValue("@MODELO_ID",
modelo);

        executacmdMySql_update_automovel.ExecuteNonQuery();

        MessageBox.Show("Seus dados foram alterados com sucesso!");

        txtcodigo.Text = String.Empty;
        txtnome.Text = String.Empty;
        txtcor.Text = String.Empty;
        cmbstatus.Text = String.Empty;
        txtanofab.Text = String.Empty;
        txtvalor.Text = String.Empty;
        txtkm.Text = String.Empty;
        txtmarca.Text = String.Empty;
        txtmodelo.Text = String.Empty;

        string sqlstm = "Select * from TB_AUTOMOVEL";
        MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
        DataSet DS = new System.Data.DataSet();
        SDA.Fill(DS, "TB_AUTOMOVEL");
        dgvautomovel.DataSource = DS.Tables[0];

        btncadastrar.Enabled = true;

        con.Close();
    }
}

}

```

//Configurações do Botão LIMPAR

```

private void btnlimpar_Click(object sender, EventArgs e)
{
    txtcodigo.Text = String.Empty;
    txtnome.Text = String.Empty;
    txtcor.Text = String.Empty;
    cmbstatus.Text = String.Empty;
    txtanofab.Text = String.Empty;
    txtvalor.Text = String.Empty;
    txtkm.Text = String.Empty;
    txtmarca.Text = String.Empty;
    txtmodelo.Text = String.Empty;

    btncadastrar.Enabled = true;
}

//Configurações de LOAD FRMAUTOMOVEL
private void FrmAutomovel_Load(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_automovel = "select * from TB_AUTOMOVEL ORDER BY
TB_AUTOMOVEL_ANO_FAB DESC";

    MySqlCommand executacmdMySql_select_automovel = new
MySqlCommand(sql_select_automovel, con);
    executacmdMySql_select_automovel.ExecuteNonQuery();

    DataTable tabela_automovel = new DataTable();

    MySqlDataAdapter da_automovel = new
MySqlDataAdapter(executacmdMySql_select_automovel);
    da_automovel.Fill(tabela_automovel);

    dgvautomovel.DataSource = tabela_automovel;

    //NOME DAS COLUNAS
    dgvautomovel.Columns["TB_AUTOMOVEL_ID"].HeaderText = "ID";
    dgvautomovel.Columns["TB_AUTOMOVEL_NOME"].HeaderText = "NOME";
    dgvautomovel.Columns["TB_AUTOMOVEL_ANO_FAB"].HeaderText = "ANO DE
FABRICAÇÃO";
    dgvautomovel.Columns["TB_AUTOMOVEL_COR"].HeaderText = "COR";
    dgvautomovel.Columns["TB_AUTOMOVEL_KM"].HeaderText = "KM";
    dgvautomovel.Columns["TB_AUTOMOVEL_VALOR_D"].HeaderText = "VALOR";
    dgvautomovel.Columns["TB_AUTOMOVEL_STATUS"].HeaderText = "STATUS";
    dgvautomovel.Columns["TB_MARCA_ID"].HeaderText = "MARCA";
    dgvautomovel.Columns["TB_MODELO_ID"].HeaderText = "MODELO";

```

```

    con.Close();
}

//Configurações Clique de Célula DGVAUTOMOVEL
private void dgvautomovel_CellClick(object sender, DataGridViewCellEventArgs e)
{
    txtcodigo.Text = dgvautomovel.CurrentRow.Cells[0].Value.ToString();
    txtnome.Text = dgvautomovel.CurrentRow.Cells[1].Value.ToString();
    txtanofab.Text = dgvautomovel.CurrentRow.Cells[2].Value.ToString();
    txtcor.Text = dgvautomovel.CurrentRow.Cells[3].Value.ToString();
    txtkm.Text = dgvautomovel.CurrentRow.Cells[4].Value.ToString();
    txtvalor.Text = dgvautomovel.CurrentRow.Cells[5].Value.ToString();
    cmbstatus.Text = dgvautomovel.CurrentRow.Cells[6].Value.ToString();
    txtmarca.Text = dgvautomovel.CurrentRow.Cells[7].Value.ToString();
    txtmodelo.Text = dgvautomovel.CurrentRow.Cells[8].Value.ToString();

    btncadastrar.Enabled = false;
}

// Configurações do botão EXCLUIR

private void btnexcluir_Click(object sender, EventArgs e)
{
    try
    {
        MySqlConnection con = new MySqlConnection(conexao);

        int codigo;
        codigo = int.Parse(txtcodigo.Text);
        string nome;
        string cor;
        string status;

        int anofab;
        double valor;
        double km;

        string marca;
        string modelo;

        con.Open();

        DialogResult respostadeletar;

```

```

        respostadeletar = MessageBox.Show("Deseja excluir o registro selecionado? Os
dados serão permanentemente apagado e não será possível recuperá-los!", "",
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2);

        if (respostadeletar == DialogResult.Yes)
        {

            string sql_delete_automovel = @"delete from TB_AUTOMOVEL where
TB_AUTOMOVEL_ID = @codigo";

            MySqlCommand executarcmdMySql_delete_automovel = new
MySqlCommand(sql_delete_automovel, con);
            executarcmdMySql_delete_automovel.Parameters.AddWithValue("@codigo",
codigo);

            executarcmdMySql_delete_automovel.ExecuteNonQuery();

            MessageBox.Show("Registro deletado com sucesso");

            txtcodigo.Text = String.Empty;
            txtnome.Text = String.Empty;
            txtcor.Text = String.Empty;
            cmbstatus.Text = String.Empty;
            txtanofab.Text = String.Empty;
            txtvalor.Text = String.Empty;
            txtkm.Text = String.Empty;
            txtmarca.Text = String.Empty;
            txtmodelo.Text = String.Empty;

            btnexcluir.Enabled = true;

            string sqlstm = "Select * from TB_AUTOMOVEL";
            MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
            DataSet DS = new System.Data.DataSet();
            SDA.Fill(DS, "TB_AUTOMOVEL");
            dgvautomovel.DataSource = DS.Tables[0];

        }

        con.Close();
    }
    catch (Exception)
    {
        MessageBox.Show("Os dados não podem ser excluídos pois apresentam valores
nulos ou que estão sendo referenciados em outra tabela. \r\nTente novamente
selecionando uma das células.");
    }
}

```

```
}
```

```
//Configurações textbox FK MARCA  
private void txtmarca_KeyPress(object sender, KeyPressEventArgs e)  
{  
    if (!Char.IsDigit(e.KeyChar) && e.KeyChar != (char)8)  
    {  
        e.Handled = true;  
    }  
}  
//Configurações textbox FK MODELO  
private void txtmodelo_KeyPress(object sender, KeyPressEventArgs e)  
{  
    if (!Char.IsDigit(e.KeyChar) && e.KeyChar != (char)8)  
    {  
        e.Handled = true;  
    }  
}
```

```
//Configurações KEYPRESS para ERRO DE CARACTERES DECIMAIS  
  
private void txtkm_KeyPress(object sender, KeyPressEventArgs e)  
{  
    if (e.KeyChar == ',' || e.KeyChar == '.')  
    {  
        e.KeyChar = '.';  
  
        //Verifica se já existe algum ponto digitada no TextBox  
  
        if (txtvalor.Text.Contains("."))  
            e.Handled = true; //Caso exista, aborte  
    }  
  
    //Permite apenas a digitação de números e da tecla backspace  
  
    else if (!Char.IsDigit(e.KeyChar) && e.KeyChar != (char)8)  
    {  
        e.Handled = true;  
    }  
}
```

```
//Configurações textbox ANO FABRICAÇÃO  
private void txtanofab_KeyPress(object sender, KeyPressEventArgs e)  
{
```



```

    {

    // Criar variável conexão
    string conexao =
    ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionStr
ing;

        public FrmCargo()
        {
            InitializeComponent();
        }

        //Configurações dos botões exibidos no Formulário
        private void btnvoltar_Click_1(object sender, EventArgs e)
        {
            this.Hide();
            FrmLogin frmLogin = new FrmLogin();
            frmLogin.ShowDialog();
        }

        private void btnpesquisar_Click(object sender, EventArgs e)
        {
            PesquisaCargo cargo = new PesquisaCargo();
            cargo.ShowDialog();
        }

        //Configurações do botão CADASTRAR
        private void btncadastrar_Click(object sender, EventArgs e)
        {
            try
            {

                MySqlConnection con = new MySqlConnection(conexao);

                string nome;
                nome = txtnome.Text;

                if (nome == string.Empty)
                {

                    MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o
corretamente.");

                }

                else
                {

                    MessageBox.Show("Cadastro efetuado com sucesso!");
                }
            }
            catch { }
        }
    }
}

```

```

txtcodigo.Text = String.Empty;
txtnome.Text = String.Empty;

string sql_insert = @"insert into TB_CARGO
    (TB_CARGO_NOME)
values (@CARGO_NOME)";

    MySqlCommand executacmdMySql_insert = new MySqlCommand(sql_insert,
con);

    executacmdMySql_insert.Parameters.AddWithValue("@CARGO_NOME",
nome);

    con.Open();
    executacmdMySql_insert.ExecuteNonQuery();

    string sql_select_cargo = "select * from TB_CARGO";

    MySqlCommand executacmdMySql_select_cargo = new
MySqlCommand(sql_select_cargo, con);
    executacmdMySql_select_cargo.ExecuteNonQuery();

    DataTable tabela_cargo = new DataTable();

    MySqlDataAdapter da_cargo = new
MySqlDataAdapter(executacmdMySql_select_cargo);
    da_cargo.Fill(tabela_cargo);

    dgvcargo.DataSource = tabela_cargo;

    con.Close();
}

}

catch (Exception)
{
    MessageBox.Show("Ops! Ocorreu um erro, esses dados já foram cadastrados!
\r\nTente novamente.");
}

}

}

//Configurações botão ATUALIZAR

```



```

private void btnatualizar_Click(object sender, EventArgs e)
{
    try
    {
        btncadastrar.Enabled = true;

        int codigo;
        string nome;

        codigo = int.Parse(txtcodigo.Text);
        nome = txtnome.Text;

        if (nome == string.Empty)
        {
            MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o
corretamente.");
        }

        else
        {
            DialogResult respostaeditar;

            respostaeditar = MessageBox.Show("Deseja editar o registro selecionado? Os
dados serão permanentemente alterados e não será possível recuperá-los!", "",
            MessageBoxButtons.YesNo, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button2);

            if (respostaeditar == DialogResult.Yes)
            {

                MySqlConnection con = new MySqlConnection(conexao);

                string sql_update_cargo = @"update TB_CARGO
set TB_CARGO_NOME = @CARGO_NOME
where TB_CARGO_ID = @id";

                con.Open();

                MySqlCommand executacmdMySql_update_cargo = new
MySqlCommand(sql_update_cargo, con);

                executacmdMySql_update_cargo.Parameters.AddWithValue("@id", codigo);
executacmdMySql_update_cargo.Parameters.AddWithValue("@CARGO_NOME", nome);

                executacmdMySql_update_cargo.ExecuteNonQuery();

                MessageBox.Show("Seus dados foram alterados com sucesso!");

                txtcodigo.Text = String.Empty;
                txtnome.Text = String.Empty;
            }
        }
    }
}

```

```

string sqlstm = "Select * from TB_CARGO";
MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
DataSet DS = new System.Data.DataSet();
SDA.Fill(DS, "TB_CARGO");
dgvcargo.DataSource = DS.Tables[0];

btncadastrar.Enabled = true;

        con.Close();
    }
}
}

catch
{
    MessageBox.Show("Esses dados já foram cadastrados! \r\nTente novamente.");
}
}

//Configurações do botão LIMPAR

private void btnlimpar_Click(object sender, EventArgs e)
{
    txtcodigo.Text = String.Empty;
    txtnome.Text = String.Empty;

    btncadastrar.Enabled = true;

}

//Configurações de Load FMRCARGO
private void FrmCargo_Load(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_cargo = "select * from TB_CARGO ORDER BY TB_CARGO_ID";

    MySqlCommand executacmdMySql_select_cargo = new
MySqlCommand(sql_select_cargo, con);
    executacmdMySql_select_cargo.ExecuteNonQuery();

    DataTable tabela_cargo = new DataTable();

    MySqlDataAdapter da_cargo = new
MySqlDataAdapter(executacmdMySql_select_cargo);

```

```
da_cargo.Fill(tabela_cargo);  
  
dgvcargo.DataSource = tabela_cargo;
```

```
//NOME DAS COLUNAS  
dgvcargo.Columns["TB_CARGO_ID"].HeaderText = "ID";  
dgvcargo.Columns["TB_CARGO_NOME"].HeaderText = "NOME";
```

```
con.Close();  
}
```

```
//Configurações Clique de célula DGVCARGO  
private void dgvcargo_CellClick(object sender, DataGridViewCellEventArgs e)  
{  
    txtcodigo.Text = dgvcargo.CurrentRow.Cells[0].Value.ToString();  
    txtnome.Text = dgvcargo.CurrentRow.Cells[1].Value.ToString();  
  
    btncadastrar.Enabled = false;  
}
```

```
//Configurações do botão EXCLUIR  
private void btnexcluir_Click(object sender, EventArgs e)  
{  
    try  
    {  
  
        MySqlConnection con = new MySqlConnection(conexao);  
  
        int codigo;  
        codigo = int.Parse(txtcodigo.Text);  
  
        con.Open();  
  
        DialogResult respostadeletar;  
  
        respostadeletar = MessageBox.Show("Deseja excluir o registro selecionado? Os  
dados serão permanentemente apagado e não será possível recuperá-los!", "",  
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,  
        MessageBoxDefaultButton.Button2);
```

```

        if (respostadeletar == DialogResult.Yes)
        {
            string sql_delete_cargo = @"delete from TB_CARGO where TB_CARGO_ID =
@codigo";

            MySqlCommand executarcmdMySql_delete_cargo = new
MySqlCommand(sql_delete_cargo, con);
            executarcmdMySql_delete_cargo.Parameters.AddWithValue("@codigo",
codigo);

            executarcmdMySql_delete_cargo.ExecuteNonQuery();

            MessageBox.Show("Registro deletado com sucesso");

            txtcodigo.Text = String.Empty;
            txtnome.Text = String.Empty;

            btnexcluir.Enabled = true;

            string sqlstm = "Select * from TB_CARGO";
            MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
            DataSet DS = new System.Data.DataSet();
            SDA.Fill(DS, "TB_CARGO");
            dgvcargo.DataSource = DS.Tables[0];
        }

        con.Close();
    }
    catch (Exception)
    {
        MessageBox.Show("Os dados não podem ser excluídos pois apresentam valores
nulos ou que estão sendo referenciados em outra tabela. \r\nTente novamente
selecionando uma das células.");
    }
}

private void dgvcargo_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}
}
}
}

```

- **FrmCliente.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace ProjetoLocadora
{
    public partial class FrmCliente : Form
    {

        // Criar variável conexão
        string conexao =
ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionStr
ing;

        public FrmCliente()
        {
            InitializeComponent();

            //Configurações dos botões exibidos no Formulário
            private void btnvoltar_Click(object sender, EventArgs e)
            {
                this.Hide();
                FrmLogin frmLogin = new FrmLogin();
                frmLogin.ShowDialog();
            }

            private void btnpesquisar_Click(object sender, EventArgs e)
            {
                PesquisaCliente cliente = new PesquisaCliente();
                cliente.ShowDialog();
            }

            //Configurações do botão CADASTRAR
            private void btncadastrar_Click(object sender, EventArgs e)
            {
                try

```

```

{
    MySqlConnection con = new MySqlConnection(conexao);

    string nome;
    string telefone;
    string sexo;
    string email;
    string senha;
    string endereco;
    string complemento;
    string bairro;
    string cidade;
    string uf;
    string datanasc;
    string datacad;

    nome = txtnome.Text;
    telefone = txttelefone.Text;
    sexo = cmbsexo.Text;
    email = txtemail.Text;
    senha = txtsenha.Text;
    endereco = txtendereco.Text;
    complemento = txtcomplemento.Text;
    bairro = txtbairro.Text;
    cidade = txtcidade.Text;
    uf = cmbuf.Text;
    datanasc = datenasc.Text;
    datacad = datecad.Text;

    if (nome == string.Empty || telefone == string.Empty || email == string.Empty ||
senha == string.Empty
        || endereco == string.Empty || bairro == string.Empty || cidade == string.Empty
|| uf == string.Empty
        || datanasc == string.Empty || datacad == string.Empty)
    {
        MessageBox.Show("Todos os campos são OBRIGATÓRIOS. \r\nPreencha-os
corretamente.");
    }

    else
    {
        MessageBox.Show("Cadastro efetuado com sucesso!");

        txtcodigo.Text = String.Empty;
        txtnome.Text = String.Empty;
        txttelefone.Text = String.Empty;
        cmbsexo.Text = String.Empty;
        txtemail.Text = String.Empty;
    }
}

```

```

txtsenha.Text = String.Empty;
txtendereco.Text = String.Empty;
txtcomplemento.Text = String.Empty;
txtbairro.Text = String.Empty;
txtcidade.Text = String.Empty;
cmbuf.Text = String.Empty;
datenasc.Text = String.Empty;
datecad.Text = String.Empty;

```

```

string sql_insert = @"insert into TB_CLIENTE
(
TB_CLIENTE_NOME,
TB_CLIENTE_TEL,
TB_CLIENTE_SEXO,
TB_CLIENTE_EMAIL,
TB_CLIENTE_SENHA,
TB_CLIENTE_ENDERECO,
TB_CLIENTE_COMPLEMENTO,
TB_CLIENTE_BAIRRO,
TB_CLIENTE_CIDADE,
TB_CLIENTE_UF,
TB_CLIENTE_DT_NASC,
TB_CLIENTE_DT_CAD)

```

```

values (

```

```

@CLIENTE_NOME,
@CLIENTE_TEL,
@CLIENTE_SEXO,
@CLIENTE_EMAIL,
@CLIENTE_SENHA,
@CLIENTE_ENDERECO,
@CLIENTE_COMPLEMENTO,
@CLIENTE_BAIRRO,
@CLIENTE_CIDADE,
@CLIENTE_UF,
@CLIENTE_DT_NASC,
@CLIENTE_DT_CAD

```

```

);

```

```

MySqlCommand executacmdMySql_insert = new MySqlCommand(sql_insert,
con);

```

```

executacmdMySql_insert.Parameters.AddWithValue("@CLIENTE_NOME",
nome);

```

```

executacmdMySql_insert.Parameters.AddWithValue("@CLIENTE_TEL",
telefone);

```

```

executacmdMySql_insert.Parameters.AddWithValue("@CLIENTE_SEXO",
sexo);

```

```

executacmdMySql_insert.Parameters.AddWithValue("@CLIENTE_EMAIL",
email);

```

```

executacmdMySql_insert.Parameters.AddWithValue("@CLIENTE_SENHA",
senha);

```

```

executacmdMySQL_insert.Parameters.AddWithValue("@CLIENTE_ENDERECO",
endereco);

executacmdMySQL_insert.Parameters.AddWithValue("@CLIENTE_COMPLEMENTO",
complemento);
    executacmdMySQL_insert.Parameters.AddWithValue("@CLIENTE_BAIRRO",
bairro);
    executacmdMySQL_insert.Parameters.AddWithValue("@CLIENTE_CIDADE",
cidade);
    executacmdMySQL_insert.Parameters.AddWithValue("@CLIENTE_UF", uf);
    executacmdMySQL_insert.Parameters.AddWithValue("@CLIENTE_DT_NASC",
datenasc.Value);
    executacmdMySQL_insert.Parameters.AddWithValue("@CLIENTE_DT_CAD",
datecad.Value);

    con.Open();
    executacmdMySQL_insert.ExecuteNonQuery();

    string sql_select_cliente = "select * from TB_CLIENTE";

    MySqlCommand executacmdMySQL_select_cliente = new
MySqlCommand(sql_select_cliente, con);
    executacmdMySQL_select_cliente.ExecuteNonQuery();

    DataTable tabela_cliente = new DataTable();

    MySqlDataAdapter da_cliente = new
MySqlDataAdapter(executacmdMySQL_select_cliente);
    da_cliente.Fill(tabela_cliente);

    dgvcliente.DataSource = tabela_cliente;

    con.Close();
}

}

catch (Exception)
{
    MessageBox.Show("Ops! Ocorreu um erro, esses dados já foram cadastrados!
\r\nTente novamente.");
}

}

```



```

//Configurações do DateTimePicker
private void datenasc_ValueChanged(object sender, EventArgs e)
{
    datenasc.CustomFormat = "yyyy/MM/dd";
}

private void datecad_ValueChanged(object sender, EventArgs e)
{
    datecad.CustomFormat = "yyyy/MM/dd";
}

private void datenasc_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Back)
    {
        datenasc.CustomFormat = " ";
    }
}

private void datecad_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Back)
    {
        datecad.CustomFormat = " ";
    }
}

// Configurações do botão ATUALIZAR
private void btnatualizar_Click(object sender, EventArgs e)
{
    try
    {
        btncadastrar.Enabled = true;

        int codigo;
        string nome;
        string telefone;
        string sexo;
        string email;
        string senha;
        string endereco;
        string complemento;
        string bairro;
    }
}

```

```
string cidade;  
string uf;  
string datanasc;  
string datacad;
```

```
codigo = int.Parse(txtcodigo.Text);  
nome = txtnome.Text;  
telefone = txttelefone.Text;  
sexo = cmbsexo.Text;  
email = txtemail.Text;  
senha = txtsenha.Text;  
endereco = txtendereco.Text;  
complemento = txtcomplemento.Text;  
bairro = txtbairro.Text;  
cidade = txtcidade.Text;  
uf = cmbuf.Text;  
datanasc = datenasc.Text;  
datacad = datecad.Text;
```

```
if (nome == string.Empty || telefone == string.Empty || email == string.Empty ||  
senha == string.Empty  
    || endereco == string.Empty || bairro == string.Empty || cidade == string.Empty  
|| uf == string.Empty  
    || datanasc == string.Empty || datacad == string.Empty)  
{  
  
    MessageBox.Show("Todos os campos são OBRIGATÓRIOS. \r\nPreencha-os  
corretamente.");  
  
}
```

```
else  
{  
    DialogResult respostaeditar;  
  
    respostaeditar = MessageBox.Show("Deseja editar o registro selecionado? Os  
dados serão permanentemente alterados e não será possível recuperá-los!", "",  
        MessageBoxButtons.YesNo, MessageBoxIcon.Information,  
        MessageBoxDefaultButton.Button2);
```

```
if (respostaeditar == DialogResult.Yes)  
{
```

```
    MySqlConnection con = new MySqlConnection(conexao);
```

```
    string sql_update_cliente = @"update TB_CLIENTE  
set TB_CLIENTE_NOME = @CLIENTE_NOME,  
TB_CLIENTE_TEL = @CLIENTE_TEL,  
TB_CLIENTE_SEXO = @CLIENTE_SEXO,
```

```
TB_CLIENTE_EMAIL = @CLIENTE_EMAIL,  
TB_CLIENTE_SENHA = @CLIENTE_SENHA,  
TB_CLIENTE_ENDERECO = @CLIENTE_ENDERECO,  
TB_CLIENTE_COMPLEMENTO = @CLIENTE_COMPLEMENTO,  
TB_CLIENTE_BAIRRO = @CLIENTE_BAIRRO,  
TB_CLIENTE_CIDADE = @CLIENTE_CIDADE,  
TB_CLIENTE_UF = @CLIENTE_UF,  
TB_CLIENTE_DT_NASC = @CLIENTE_DT_NASC,  
TB_CLIENTE_DT_CAD = @CLIENTE_DT_CAD  
where TB_CLIENTE_ID = @id";
```

```
con.Open();
```

```
MySqlCommand executacmdMySQL_update_cliente = new  
MySqlCommand(sql_update_cliente, con);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@id",  
codigo);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_NOME",  
nome);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_TEL",  
telefone);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_SEXO", sexo);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_EMAIL",  
email);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_SENHA",  
senha);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_ENDERECO",  
endereco);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_COMPLEMEN  
TO", complemento);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_BAIRRO",  
bairro);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_CIDADE",  
cidade);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_UF", uf);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_DT_NASC",  
datanasc);
```

```
executacmdMySQL_update_cliente.Parameters.AddWithValue("@CLIENTE_DT_CAD",  
datacad);
```

```
executacmdMySql_update_cliente.ExecuteNonQuery();
```

```
MessageBox.Show("Seus dados foram alterados com sucesso!");
```

```
txtcodigo.Text = String.Empty;  
txtnome.Text = String.Empty;  
txttelefone.Text = String.Empty;  
cmbsexo.Text = String.Empty;  
txtemail.Text = String.Empty;  
txtsenha.Text = String.Empty;  
txtendereco.Text = String.Empty;  
txtcomplemento.Text = String.Empty;  
txtbairro.Text = String.Empty;  
txtcidade.Text = String.Empty;  
cmbuf.Text = String.Empty;  
datenasc.Text = String.Empty;
```

```
string sqlstm = "Select * from TB_CLIENTE";  
MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);  
DataSet DS = new System.Data.DataSet();  
SDA.Fill(DS, "TB_CLIENTE");  
dgvcliente.DataSource = DS.Tables[0];
```

```
btncadastrar.Enabled = true;
```

```
        con.Close();  
    }  
}  
}  
  
catch  
{  
    MessageBox.Show("Esses dados já foram cadastrados! \r\nTente novamente.");  
}  
  
}
```

```
//Configurações botão LIMPAR  
private void btnlimpar_Click(object sender, EventArgs e)  
{  
    txtcodigo.Text = String.Empty;  
    txtnome.Text = String.Empty;  
    txttelefone.Text = String.Empty;  
    cmbsexo.Text = String.Empty;
```

```

txtemail.Text = String.Empty;
txtsenha.Text = String.Empty;
txtendereco.Text = String.Empty;
txtcomplemento.Text = String.Empty;
txtbairro.Text = String.Empty;
txtcidade.Text = String.Empty;
cmbuf.Text = String.Empty;
datenasc.Text = String.Empty;
datecad.Text = String.Empty;

btncadastrar.Enabled = true;

}

//Configurações Load FRMCLIENTE

private void FrmCliente_Load(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_cliente = "select * from TB_CLIENTE ORDER BY
TB_CLIENTE_NOME";

    MySqlCommand executacmdMySQL_select_cliente = new
MySqlCommand(sql_select_cliente, con);
    executacmdMySQL_select_cliente.ExecuteNonQuery();

    DataTable tabela_cliente = new DataTable();

    MySqlDataAdapter da_cliente = new
MySqlDataAdapter(executacmdMySQL_select_cliente);
    da_cliente.Fill(tabela_cliente);

    dgvcliente.DataSource = tabela_cliente;

//NOME DAS COLUNA
dgvcliente.Columns["TB_CLIENTE_ID"].HeaderText = "ID";
dgvcliente.Columns["TB_CLIENTE_NOME"].HeaderText = "NOME";
dgvcliente.Columns["TB_CLIENTE_TEL"].HeaderText = "TELEFONE";
dgvcliente.Columns["TB_CLIENTE_SEXO"].HeaderText = "SEXO";
dgvcliente.Columns["TB_CLIENTE_EMAIL"].HeaderText = "EMAIL";
dgvcliente.Columns["TB_CLIENTE_SENHA"].HeaderText = "SENHA";
dgvcliente.Columns["TB_CLIENTE_ENDERECO"].HeaderText = "ENDERECO";
dgvcliente.Columns["TB_CLIENTE_COMPLEMENTO"].HeaderText =
"COMPLEMENTO";

```

```
    dgvcliente.Columns["TB_CLIENTE_BAIRRO"].HeaderText = "BAIRRO";
    dgvcliente.Columns["TB_CLIENTE_CIDADE"].HeaderText = "CIDADE";
    dgvcliente.Columns["TB_CLIENTE_UF"].HeaderText = "UNIDADE FEDERATIVA";
    dgvcliente.Columns["TB_CLIENTE_DT_NASC"].HeaderText = "DATA
NASCIMENTO";
    dgvcliente.Columns["TB_CLIENTE_DT_CAD"].HeaderText = "DATA CADASTRO";
```

```
    con.Close();
}
```

```
//Configurações Clique de célula DGVCLIENTE
```

```
private void dgvcliente_CellClick(object sender, DataGridViewCellEventArgs e)
{
    txtcodigo.Text = dgvcliente.CurrentRow.Cells[0].Value.ToString();
    txtnome.Text = dgvcliente.CurrentRow.Cells[1].Value.ToString();
    txttelefone.Text = dgvcliente.CurrentRow.Cells[2].Value.ToString();
    cmbsexo.Text = dgvcliente.CurrentRow.Cells[3].Value.ToString();
    txtemail.Text = dgvcliente.CurrentRow.Cells[4].Value.ToString();
    txtsenha.Text = dgvcliente.CurrentRow.Cells[5].Value.ToString();
    txtendereco.Text = dgvcliente.CurrentRow.Cells[6].Value.ToString();
    txtcomplemento.Text = dgvcliente.CurrentRow.Cells[7].Value.ToString();
    txtbairro.Text = dgvcliente.CurrentRow.Cells[8].Value.ToString();
    txtcidade.Text = dgvcliente.CurrentRow.Cells[9].Value.ToString();
    cmbuf.Text = dgvcliente.CurrentRow.Cells[10].Value.ToString();
    datenasc.Text = dgvcliente.CurrentRow.Cells[11].Value.ToString();
    datecad.Text = dgvcliente.CurrentRow.Cells[12].Value.ToString();

    btncadastrar.Enabled = false;
}
```

```
//Configurações botão EXCLUIR
```

```
private void btnexcluir_Click(object sender, EventArgs e)
{
```

```
    try
    {
```

```
        MySqlConnection con = new MySqlConnection(conexao);
```

```

int codigo;
codigo = int.Parse(txtcodigo.Text);

con.Open();

DialogResult respostadeletar;

respostadeletar = MessageBox.Show("Deseja excluir o registro selecionado? Os
dados serão permanentemente apagado e não será possível recuperá-los!", "",
    MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
    MessageBoxDefaultButton.Button2);

if (respostadeletar == DialogResult.Yes)
{
    string sql_delete_cliente = @"delete from TB_CLIENTE where
TB_CLIENTE_ID = @codigo";

    MySqlCommand executarcmdMySql_delete_cliente = new
MySqlCommand(sql_delete_cliente, con);
    executarcmdMySql_delete_cliente.Parameters.AddWithValue("@codigo",
codigo);

    executarcmdMySql_delete_cliente.ExecuteNonQuery();

    MessageBox.Show("Registro deletado com sucesso");

    txtcodigo.Text = String.Empty;
    txtnome.Text = String.Empty;
    txttelefone.Text = String.Empty;
    cmbsexo.Text = String.Empty;
    txtemail.Text = String.Empty;
    txtsenha.Text = String.Empty;
    txtendereco.Text = String.Empty;
    txtcomplemento.Text = String.Empty;
    txtbairro.Text = String.Empty;
    txtcidade.Text = String.Empty;
    cmbuf.Text = String.Empty;
    datenasc.Text = String.Empty;
    datecad.Text = String.Empty;

    btnexcluir.Enabled = true;

    string sqlstm = "Select * from TB_CLIENTE";
    MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
    DataSet DS = new System.Data.DataSet();
    SDA.Fill(DS, "TB_CLIENTE");
    dgvcliente.DataSource = DS.Tables[0];
}

```

```

        con.Close();
    }
    catch (Exception)
    {
        MessageBox.Show("Os dados não podem ser excluídos pois apresentam valores
nulos ou que estão sendo referenciados em outra tabela. \r\nTente novamente
selecionando uma das células.");
    }
}

```

```
//TEXTBOX TELEFONE
```

```
//Configurações textbox TELEFONE
```

```
private void txttelefone_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!Char.IsDigit(e.KeyChar) && e.KeyChar != (char)8)
    {
        e.Handled = true;
    }
}

```

```
private void txttelefone_TextChanged(object sender, EventArgs e)
{
    txttelefone.MaxLength = 16;
}

```

```
// GERADOR DE SENHAS
```

```
//Configurações botão COPIAR SENHA
```

```
private void pbcopiar_Click(object sender, EventArgs e)
{

```

```
    try
```

```
    {
```

```
        string senha = txtsenha.Text;
```

```
        Clipboard.SetText(senha);
```

```
        MessageBox.Show("Senha copiada para Área de Transferência!");
```

```
    }
```

```
    catch (Exception)
```

```
    {
```

```
        MessageBox.Show("É preciso GERAR UMA SENHA para que a tarefa seja
```



```
relizada.");
```

```
    }  
}  
  
//Configurações botão GERAR SENHA  
private void btngerarsenha_Click(object sender, EventArgs e)  
{  
    Random senhaAleatoria = new Random();  
    int senha = senhaAleatoria.Next(100, 100000);  
    txtsenha.Text = senha.ToString();  
  
}  
  
private void txtsenha_TextChanged(object sender, EventArgs e)  
{  
    txtsenha.MaxLength = 6;  
}  
}  
}
```

- **FrmFuncionario.cs**

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Configuration;  
using MySql.Data;  
using MySql.Data.MySqlClient;  
  
namespace ProjetoLocadora  
{  
    public partial class FrmFuncionario : Form  
    {  
  
        // Criar variável conexão  
        string conexao =  
ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionStr  
ing;  
  
        public FrmFuncionario()  
        {  
            InitializeComponent();  
        }  
    }  
}
```

```

        // Configurações dos botões exibidos no Formulário
private void btnvoltar_Click_1(object sender, EventArgs e)
{
    this.Hide();
    FrmLogin frmLogin = new FrmLogin();
    frmLogin.ShowDialog();
}

private void btnpesquisar_Click(object sender, EventArgs e)
{
    PesquisaFuncionario funcionario = new PesquisaFuncionario();
    funcionario.ShowDialog();
}

//Configurações do botão CADASTRAR
private void btncadastrar_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    string nome;
    string telefone;
    string data;
    string cargo;

    nome = txtnome.Text;
    telefone = txttelefone.Text;
    data = datacont.Text;
    cargo = txtcargo.Text;

    if (nome == string.Empty || data == string.Empty || cargo == string.Empty)
    {
        MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o corretamente.");
    }

    else
    {
        MessageBox.Show("Cadastro efetuado com sucesso!");

        txtcodigo.Text = String.Empty;
        txtnome.Text = String.Empty;
        txttelefone.Text = String.Empty;
        txtcargo.Text = String.Empty;
        datacont.Text = String.Empty;

        string sql_insert = @"insert into TB_FUNCIONARIO

```

```

        (TB_FUNCIONARIO_NOME,
        TB_FUNCIONARIO_TEL,
        TB_FUNCIONARIO_DT_CONTRATO,
        TB_CARGO_ID)

        values (@FUNCIONARIO_NOME,
        @FUNCIONARIO_TEL,
        @FUNCIONARIO_DT_CONTRATO,
        @CARGO_ID);

        MySqlCommand executacmdMySql_insert = new MySqlCommand(sql_insert,
con);

        executacmdMySql_insert.Parameters.AddWithValue("@FUNCIONARIO_NOME",
nome);
        executacmdMySql_insert.Parameters.AddWithValue("@FUNCIONARIO_TEL",
telefone);

        executacmdMySql_insert.Parameters.AddWithValue("@FUNCIONARIO_DT_CONTRATO",
datacont.Value);
        executacmdMySql_insert.Parameters.AddWithValue("@CARGO_ID", cargo);

        con.Open();
        executacmdMySql_insert.ExecuteNonQuery();

        string sql_select_funcionario = "select * from TB_FUNCIONARIO";

        MySqlCommand executacmdMySql_select_funcionario= new
MySqlCommand(sql_select_funcionario, con);
        executacmdMySql_select_funcionario.ExecuteNonQuery();

        DataTable tabela_funcionario = new DataTable();

        MySqlDataAdapter da_funcionario = new
MySqlDataAdapter(executacmdMySql_select_funcionario);
        da_funcionario.Fill(tabela_funcionario);

        dgvfuncionario.DataSource = tabela_funcionario;

        con.Close();
    }

}

private void btnatualizar_Click(object sender, EventArgs e)
{
    btncadastrar.Enabled = true;

```

```

int codigo;
string nome;
string telefone;
string data;
string cargo;

codigo = int.Parse(txtcodigo.Text);
nome = txtnome.Text;
telefone = txttelefone.Text;
data = datacont.Text;
cargo = txtcargo.Text;

if (nome == string.Empty || data == string.Empty || cargo == string.Empty)
{
    MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o corretamente.");
}

else
{
    DialogResult respostaeditar;

    respostaeditar = MessageBox.Show("Deseja editar o registro selecionado? Os
dados serão permanentemente alterados e não será possível recuperá-los!", "",
    MessageBoxButtons.YesNo, MessageBoxIcon.Information,
    MessageBoxDefaultButton.Button2);

    if (respostaeditar == DialogResult.Yes)
    {
        MySqlConnection con = new MySqlConnection(conexao);

        string sql_update_funcionario = @"update TB_FUNCIONARIO
set TB_FUNCIONARIO_NOME = @FUNCIONARIO_NOME,
    TB_FUNCIONARIO_TEL = @FUNCIONARIO_TEL,
    TB_FUNCIONARIO_DT_CONTRATO =
@FUNCIONARIO_DT_CONTRATO,
    TB_CARGO_ID = @CARGO_ID
    where TB_FUNCIONARIO_ID = @id";

        con.Open();

        MySqlCommand executacmdMySQL_update_funcionario = new
        MySqlCommand(sql_update_funcionario, con);

        executacmdMySQL_update_funcionario.Parameters.AddWithValue("@id",
        codigo);

        executacmdMySQL_update_funcionario.Parameters.AddWithValue("@FUNCIONARIO_NO
ME", nome);

```

```
executacmdMySql_update_funcionario.Parameters.AddWithValue("@FUNCIONARIO_TEL", telefone);
```

```
executacmdMySql_update_funcionario.Parameters.AddWithValue("@FUNCIONARIO_DT_CONTRATO", data);
```

```
executacmdMySql_update_funcionario.Parameters.AddWithValue("@CARGO_ID", cargo);
```

```
executacmdMySql_update_funcionario.ExecuteNonQuery();
```

```
MessageBox.Show("Seus dados foram alterados com sucesso!");
```

```
txtcodigo.Text = String.Empty;
```

```
txtnome.Text = String.Empty;
```

```
string sqlstm = "Select * from TB_FUNCIONARIO";
```

```
MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
```

```
DataSet DS = new System.Data.DataSet();
```

```
SDA.Fill(DS, "TB_FUNCIONARIO");
```

```
dgvfuncionario.DataSource = DS.Tables[0];
```

```
btncadastrar.Enabled = true;
```

```
con.Close();
```

```
}
```

```
}
```

```
}
```

```
//Configurações do botão LIMPAR
```

```
private void btnlimpar_Click(object sender, EventArgs e)
```

```
{
```

```
txtcodigo.Text = String.Empty;
```

```
txtnome.Text = String.Empty;
```

```
txttelefone.Text = String.Empty;
```

```
txtcargo.Text = String.Empty;
```

```
datacont.Text = String.Empty;
```

```
}
```

```
//Configurações de Load FMRFUNCIONARIO
```

```
private void FrmFuncionario_Load(object sender, EventArgs e)
```

```
{
```

```

        MySqlConnection con = new MySqlConnection(conexao);

        con.Open();

        string sql_select_funcionario = "select * from TB_FUNCIONARIO ORDER BY
        TB_FUNCIONARIO_NOME";

        MySqlCommand executacmdMySql_select_funcionario = new
        MySqlCommand(sql_select_funcionario, con);
        executacmdMySql_select_funcionario.ExecuteNonQuery();

        DataTable tabela_funcionario = new DataTable();

        MySqlDataAdapter da_funcionario = new
        MySqlDataAdapter(executacmdMySql_select_funcionario);
        da_funcionario.Fill(tabela_funcionario);

        dgvfuncionario.DataSource = tabela_funcionario;

        //NOME DAS COLUNAS
        dgvfuncionario.Columns["TB_FUNCIONARIO_ID"].HeaderText = "ID";
        dgvfuncionario.Columns["TB_FUNCIONARIO_NOME"].HeaderText = "NOME";
        dgvfuncionario.Columns["TB_FUNCIONARIO_TEL"].HeaderText = "TELEFONE";
        dgvfuncionario.Columns["TB_FUNCIONARIO_DT_CONTRATO"].HeaderText =
"DATA CONTRATO";
        dgvfuncionario.Columns["TB_CARGO_ID"].HeaderText = "CARGO";

        con.Close();
    }

    //Configurações Clique de célula DGVFUNIONARIO
    private void dgvfuncionario_CellClick(object sender, DataGridViewCellEventArgs e)
    {
        txtcodigo.Text = dgvfuncionario.CurrentRow.Cells[0].Value.ToString();
        txtnome.Text = dgvfuncionario.CurrentRow.Cells[1].Value.ToString();
        txttelefone.Text = dgvfuncionario.CurrentRow.Cells[2].Value.ToString();
        datacont.Text = dgvfuncionario.CurrentRow.Cells[3].Value.ToString();
        txtcargo.Text = dgvfuncionario.CurrentRow.Cells[4].Value.ToString();

        btncadastrar.Enabled = false;
    }

```

```

private void btnexcluir_Click(object sender, EventArgs e)
{
    try
    {

        MySqlConnection con = new MySqlConnection(conexao);

        int codigo;
        string nome;
        string telefone;
        string data;
        string cargo;

        codigo = int.Parse(txtcodigo.Text);
        nome = txtnome.Text;
        telefone = txttelefone.Text;
        data = datacont.Text;
        cargo = txtcargo.Text;

        con.Open();

        DialogResult respostadeletar;

        respostadeletar = MessageBox.Show("Deseja excluir o registro selecionado? Os
dados serão permanentemente apagado e não será possível recuperá-los!", "",
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2);

        if (respostadeletar == DialogResult.Yes)
        {

            string sql_delete_funcionario = @"delete from TB_FUNCIONARIO where
TB_FUNCIONARIO_ID = @codigo";

            MySqlCommand executarcmdMySql_delete_funcionario = new
MySqlCommand(sql_delete_funcionario, con);
            executarcmdMySql_delete_funcionario.Parameters.AddWithValue("@codigo",
codigo);

            executarcmdMySql_delete_funcionario.ExecuteNonQuery();

            MessageBox.Show("Registro deletado com sucesso");

            txtcodigo.Text = String.Empty;
            txtnome.Text = String.Empty;
            txttelefone.Text = String.Empty;
            txtcargo.Text = String.Empty;
            datacont.Text = String.Empty;

            btnexcluir.Enabled = true;

```

```

        string sqlstm = "Select * from TB_FUNCIONARIO";
        MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
        DataSet DS = new System.Data.DataSet();
        SDA.Fill(DS, "TB_FUNCIONARIO");
        dgvfuncionario.DataSource = DS.Tables[0];
    }

    con.Close();
}
catch (Exception)
{
    MessageBox.Show("Os dados não podem ser excluídos pois apresentam valores
nulos ou que estão sendo referenciados em outra tabela. \r\nTente novamente
selecionando uma das células.");
}
}

//Configurações DATETIMEPICKER DATACONT
private void datacont_ValueChanged(object sender, EventArgs e)
{
    datacont.CustomFormat = "yyyy/MM/dd";
}

//KEYDOWN para determinar um espaço vazio caso selecionado um campo de
data e pressionado a tecla de backspace

private void datacont_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Back)
    {
        datacont.CustomFormat = " ";
    }
}

//TEXTBOX TELEFONE

//Configurações textbox TELEFONE
private void txttelefone_TextChanged(object sender, EventArgs e)
{
    txttelefone.MaxLength = 16;
}

```



```

private void txttelefone_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!Char.IsDigit(e.KeyChar) && e.KeyChar != (char)8)
    {
        e.Handled = true;
    }
}

//Configurações textbox FK CARGO
private void txtcarga_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!Char.IsDigit(e.KeyChar) && e.KeyChar != (char)8)
    {
        e.Handled = true;
    }
}
}
}
}

```

- **FrmLocacao**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace ProjetoLocadora
{
    public partial class FrmLocacao : Form
    {

        // Criar variável conexão
        string conexao =
        ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionString;

        public FrmLocacao()
        {
            InitializeComponent();
        }
    }
}

```

```

//Configurações dos botões exibidos no Formulário

private void btnvoltar_Click_1(object sender, EventArgs e)
{
    this.Hide();
    FrmLogin frmLogin = new FrmLogin();
    frmLogin.ShowDialog();
}

private void btnpesquisar_Click(object sender, EventArgs e)
{
    PesquisaLocacao locacao = new PesquisaLocacao();
    locacao.ShowDialog();
}

//Configurações do botão CADASTRAR
private void btncadastrar_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    string tipo;
    string datainicio;
    string datafim;

    string valor;

    string cliente;
    string funcionario;
    string automovel;

    tipo = cmbtipo.Text;
    datainicio = dateinicio.Text;
    datafim = dateinicio.Text;
    valor = txtvalor.Text;
    cliente = txtcliente.Text;
    funcionario = txtfuncionario.Text;
    automovel = txtautomovel.Text;

    if (tipo == string.Empty || valor == string.Empty || cliente == string.Empty ||
funcionario == string.Empty || automovel == string.Empty)
    {
        MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o corretamente.");
    }
}

```

```

else
{

    MessageBox.Show("Cadastro efetuado com sucesso!");

    txtcodigo.Text = String.Empty;
    cmbtipo.Text = String.Empty;
    txtvalor.Text = String.Empty;
    dateinicio.Text = String.Empty;
    datefim.Text = String.Empty;
    txtvalor.Text = String.Empty;
    txtcliente.Text = String.Empty;
    txtfuncionario.Text = String.Empty;
    txtautomovel.Text = String.Empty;

    string sql_insert = @"insert into TB_LOCACAO

        (TB_LOCACAO_TIPO,
        TB_LOCACAO_VALOR,
        TB_LOCACAO_DT_INICIO,
        TB_LOCACAO_DT_FIM,

        TB_CLIENTE_ID,
        TB_FUNCIONARIO_ID,
        TB_AUTOMOVEL_ID)

    values (@LOCACAO_TIPO,
        @LOCACAO_VALOR,
        @LOCACAO_DT_INICIO,
        @LOCACAO_DT_FIM,
        @CLIENTE_ID,
        @FUNCIONARIO_ID,
        @AUTOMOVEL_ID)";

    MySqlCommand executacmdMySql_insert = new MySqlCommand(sql_insert,
con);

    executacmdMySql_insert.Parameters.AddWithValue("@LOCACAO_TIPO", tipo);
    executacmdMySql_insert.Parameters.AddWithValue("@LOCACAO_VALOR",
valor);
    executacmdMySql_insert.Parameters.AddWithValue("@LOCACAO_DT_INICIO",
dateinicio.Value);
    executacmdMySql_insert.Parameters.AddWithValue("@LOCACAO_DT_FIM",
datefim.Value);
    executacmdMySql_insert.Parameters.AddWithValue("@CLIENTE_ID", cliente);
    executacmdMySql_insert.Parameters.AddWithValue("@FUNCIONARIO_ID",
funcionario);
    executacmdMySql_insert.Parameters.AddWithValue("@AUTOMOVEL_ID",
automovel);

```

```

        con.Open();
        executacmdMySQL_insert.ExecuteNonQuery();

        string sql_select_locacao = "select * from TB_LOCACAO ORDER BY
TB_LOCACAO_ID";

        MySqlCommand executacmdMySQL_select_locacao = new
        MySqlCommand(sql_select_locacao, con);
        executacmdMySQL_select_locacao.ExecuteNonQuery();

        DataTable tabela_locacao = new DataTable();

        MySqlDataAdapter da_locacao = new
        MySqlDataAdapter(executacmdMySQL_select_locacao);
        da_locacao.Fill(tabela_locacao);

        dgvlocacao.DataSource = tabela_locacao;

        con.Close();
    }
}

```

```

//Configurações do botão ATUALIZAR
private void btnatualizar_Click(object sender, EventArgs e)
{
    btncadastrar.Enabled = true;

    int codigo;
    string tipo;
    string datainicio;
    string datafim;

    string valor;

    string cliente;
    string funcionario;
    string automovel;

    codigo = int.Parse(txtcodigo.Text);
    tipo = cmbtipo.Text;
    datainicio = dateinicio.Text;
    datafim = dateinicio.Text;
    valor = txtvalor.Text;
    cliente = txtcliente.Text;
}

```

```

funcionario = txtfuncionario.Text;
automovel = txtautomovel.Text;

if (tipo == string.Empty || valor == string.Empty || cliente == string.Empty ||
funcionario == string.Empty || automovel == string.Empty)
{
    MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o corretamente.");
}

else
{
    DialogResult respostaeditar;

    respostaeditar = MessageBox.Show("Deseja editar o registro selecionado? Os
dados serão permanentemente alterados e não será possível recuperá-los!", "",
    MessageBoxButtons.YesNo, MessageBoxIcon.Information,
    MessageBoxDefaultButton.Button2);

    if (respostaeditar == DialogResult.Yes)
    {

        MySqlConnection con = new MySqlConnection(conexao);

        string sql_update_locacao = @"update TB_LOCACAO

        set TB_LOCACAO_TIPO= @LOCACAO_TIPO,
        TB_LOCACAO_VALOR = @LOCACAO_VALOR,
        TB_LOCACAO_DT_INICIO = @LOCACAO_DT_INICIO,
        TB_LOCACAO_DT_FIM = @LOCACAO_DT_FIM,
        TB_CLIENTE_ID = @CLIENTE_ID,
        TB_FUNCIONARIO_ID = @FUNCIONARIO_ID,
        TB_AUTOMOVEL_ID = @AUTOMOVEL_ID
        where TB_LOCACAO_ID = @id";

        con.Open();

        MySqlCommand executacmdMySql_update_locacao = new
        MySqlCommand(sql_update_locacao, con);

        executacmdMySql_update_locacao.Parameters.AddWithValue("@id", codigo);

        executacmdMySql_update_locacao.Parameters.AddWithValue("@LOCACAO_TIPO", tipo);

        executacmdMySql_update_locacao.Parameters.AddWithValue("@LOCACAO_VALOR",
        valor);

        executacmdMySql_update_locacao.Parameters.AddWithValue("@LOCACAO_DT_INICIO",
        datainicio);

        executacmdMySql_update_locacao.Parameters.AddWithValue("@LOCACAO_DT_FIM",
        datafim);
    }
}

```

```

executacmdMySql_update_locacao.Parameters.AddWithValue("@CLIENTE_ID", cliente);

executacmdMySql_update_locacao.Parameters.AddWithValue("@FUNCIONARIO_ID",
funcionario);

executacmdMySql_update_locacao.Parameters.AddWithValue("@AUTOMOVEL_ID",
automovel);

        executacmdMySql_update_locacao.ExecuteNonQuery();

        MessageBox.Show("Seus dados foram alterados com sucesso!");

        txtcodigo.Text = String.Empty;
        cmbtipo.Text = String.Empty;
        txtvalor.Text = String.Empty;
        dateinicio.Text = String.Empty;
        datefim.Text = String.Empty;
        txtvalor.Text = String.Empty;
        txtcliente.Text = String.Empty;
        txtfuncionario.Text = String.Empty;
        txtautomovel.Text = String.Empty;

        string sqlstm = "Select * from TB_LOCACAO";
        MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
        DataSet DS = new System.Data.DataSet();
        SDA.Fill(DS, "TB_LOCACAO");
        dgvlocacao.DataSource = DS.Tables[0];

        btncadastrar.Enabled = true;

        con.Close();
    }
}
}

```

```

//Configurações do botão LIMPAR
private void btnlimpar_Click(object sender, EventArgs e)
{
    txtcodigo.Text = String.Empty;
    cmbtipo.Text = String.Empty;
    txtvalor.Text = String.Empty;
    dateinicio.Text = String.Empty;
    datefim.Text = String.Empty;
    txtvalor.Text = String.Empty;
    txtcliente.Text = String.Empty;
    txtfuncionario.Text = String.Empty;
}

```

```

txtautomovel.Text = String.Empty;

btncadastrar.Enabled = true;

}

//Configuração de Load FRMLOCACAO
private void FrmLocacao_Load(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_locacao = "select * from TB_LOCACAO ORDER BY
TB_LOCACAO_ID";

    MySqlCommand executacmdMySql_select_locacao = new
MySqlCommand(sql_select_locacao, con);
    executacmdMySql_select_locacao.ExecuteNonQuery();

    DataTable tabela_locacao = new DataTable();

    MySqlDataAdapter da_locacao = new
MySqlDataAdapter(executacmdMySql_select_locacao);
    da_locacao.Fill(tabela_locacao);

    dgvlocacao.DataSource = tabela_locacao;

//NOME DAS COLUNAS
dgvlocacao.Columns["TB_LOCACAO_ID"].HeaderText = "ID";
dgvlocacao.Columns["TB_LOCACAO_TIPO"].HeaderText = "TIPO";
dgvlocacao.Columns["TB_LOCACAO_VALOR"].HeaderText = "VALOR";
dgvlocacao.Columns["TB_LOCACAO_DT_INICIO"].HeaderText = "DATA INÍCIO";
dgvlocacao.Columns["TB_LOCACAO_DT_FIM"].HeaderText = "DATA FIM";
dgvlocacao.Columns["TB_CLIENTE_ID"].HeaderText = "CLIENTE";
dgvlocacao.Columns["TB_FUNCIONARIO_ID"].HeaderText = "FUNCIONARIO";
dgvlocacao.Columns["TB_AUTOMOVEL_ID"].HeaderText = "AUTOMOVEL";

    con.Close();
}

```

```
//Configurações Clique de célula DGVLOCACAO
private void dgvlocacao_CellClick(object sender, DataGridViewCellEventArgs e)
{
    txtcodigo.Text = dgvlocacao.CurrentRow.Cells[0].Value.ToString();
    cmbtipo.Text = dgvlocacao.CurrentRow.Cells[1].Value.ToString();
    txtvalor.Text = dgvlocacao.CurrentRow.Cells[2].Value.ToString();
    dateinicio.Text = dgvlocacao.CurrentRow.Cells[3].Value.ToString();
    datefim.Text = dgvlocacao.CurrentRow.Cells[4].Value.ToString();
    txtcliente.Text = dgvlocacao.CurrentRow.Cells[5].Value.ToString();
    txtfuncionario.Text = dgvlocacao.CurrentRow.Cells[6].Value.ToString();
    txtautomovel.Text = dgvlocacao.CurrentRow.Cells[7].Value.ToString();

    btncadastrar.Enabled = false;
}

```

```
//Configurações do botão EXCLUIR
private void btnexcluir_Click(object sender, EventArgs e)
{
    try
    {

        MySqlConnection con = new MySqlConnection(conexao);

        int codigo;
        codigo = int.Parse(txtcodigo.Text);

        con.Open();

        DialogResult respostadeletar;

        respostadeletar = MessageBox.Show("Deseja excluir o registro selecionado? Os
dados serão permanentemente apagado e não será possível recuperá-los!", "",
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2);

        if (respostadeletar == DialogResult.Yes)
        {

            string sql_delete_locacao= @"delete from TB_LOCACAO where
TB_LOCACAO_ID = @codigo";

            MySqlCommand executarcmdMySql_delete_locacao = new
MySqlCommand(sql_delete_locacao, con);
            executarcmdMySql_delete_locacao.Parameters.AddWithValue("@codigo",

```



```
codigo);
```

```
executarcmdMySql_delete_locacao.ExecuteNonQuery();
```

```
MessageBox.Show("Registro deletado com sucesso");
```

```
txtcodigo.Text = String.Empty;
```

```
cmbtipo.Text = String.Empty;
```

```
txtvalor.Text = String.Empty;
```

```
dateinicio.Text = String.Empty;
```

```
datefim.Text = String.Empty;
```

```
txtvalor.Text = String.Empty;
```

```
txtcliente.Text = String.Empty;
```

```
txtfuncionario.Text = String.Empty;
```

```
txtautomovel.Text = String.Empty;
```

```
btnexcluir.Enabled = true;
```

```
string sqlstm = "Select * from TB_LOCACAO";
```

```
MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
```

```
DataSet DS = new System.Data.DataSet();
```

```
SDA.Fill(DS, "TB_LOCACAO");
```

```
dgvlocacao.DataSource = DS.Tables[0];
```

```
}
```

```
con.Close();
```

```
}
```

```
catch (Exception)
```

```
{
```

```
    MessageBox.Show("Os dados não podem ser excluídos pois apresentam valores  
nulos ou que estão sendo referenciados em outra tabela. \r\nTente novamente  
selecionando uma das células.");
```

```
}
```

```
}
```

```
//Configurações DATETIMEPICKER DATEINICIO
```

```
private void dateinicio_ValueChanged(object sender, EventArgs e)
```

```
{
```

```
    dateinicio.CustomFormat = "yyyy/MM/dd";
```

```
}
```

```
//KEYDOWN para determinar um espaço vazio caso selecionado um campo de data e  
pressionado a tecla de backspace
```

```
private void dateinicio_KeyDown(object sender, KeyEventArgs e)
```

```
{
```

```
    if (e.KeyCode == Keys.Back)
```

```

    {
        dateinicio.CustomFormat = " ";
    }
}

```

```

//Configurações DATETIMEPICKER DATEFIM

```

```

private void datefim_ValueChanged(object sender, EventArgs e)
{
    datefim.CustomFormat = "yyyy/MM/dd";
}

```

//KEYDOWN para determinar um espaço vazio caso selecionado um campo de data e pressionado a tecla de backspace

```

private void datefim_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Back)
    {
        datefim.CustomFormat = " ";
    }
}

```

```

//Configurações KEYPRESS para ERRO DE CARACTERES DECIMAIS

```

```

private void txtvalor_KeyPress(object sender, KeyPressEventArgs e)
{
    //troca o ',' por '.'

    if (e.KeyChar == ',' || e.KeyChar == '.')
    {
        e.KeyChar = '.';

        //Verifica se já existe algum ponto digitada no TextBox

        if (txtvalor.Text.Contains("."))
            e.Handled = true; //Caso exista, aborte
    }

    //Permite apenas a digitação de números e da tecla backspace

    else if (!Char.IsDigit(e.KeyChar) && e.KeyChar != (char)8)
    {
        e.Handled = true;
    }
}

```



```

        InitializeComponent();
    }

    private void btnCargos_Click(object sender, EventArgs e)
    {
        this.Hide();
        FrmCargo frmCargo = new FrmCargo();
        frmCargo.FormClosed += (s, args) => this.Close();
        frmCargo.ShowDialog();
    }

    private void btnMarcas_Click(object sender, EventArgs e)
    {
        this.Hide();
        FrmMarca frmMarca = new FrmMarca();
        frmMarca.FormClosed += (s, args) => this.Close();
        frmMarca.ShowDialog();
    }

    private void btnModelos_Click(object sender, EventArgs e)
    {
        this.Hide();
        FrmModelo frmModelo = new FrmModelo();
        frmModelo.FormClosed += (s, args) => this.Close();
        frmModelo.ShowDialog();
    }

    private void btnClientes_Click(object sender, EventArgs e)
    {
        this.Hide();
        FrmCliente frmCliente = new FrmCliente();
        frmCliente.FormClosed += (s, args) => this.Close();
        frmCliente.ShowDialog();
    }

    private void btnFuncionarios_Click(object sender, EventArgs e)
    {
        this.Hide();
        FrmFuncionario frmFuncionario = new FrmFuncionario();
        frmFuncionario.FormClosed += (s, args) => this.Close();
        frmFuncionario.ShowDialog();
    }

    private void btnAutomoveis_Click(object sender, EventArgs e)
    {
        this.Hide();
        FrmAutomovel frmAutomovel = new FrmAutomovel();
        frmAutomovel.FormClosed += (s, args) => this.Close();
        frmAutomovel.ShowDialog();
    }

    private void btnLocacoes_Click(object sender, EventArgs e)
    {

```

```

        this.Hide();
        FrmLocacao frmLocacao = new FrmLocacao();
        frmLocacao.FormClosed += (s, args) => this.Close();
        frmLocacao.ShowDialog();
    }

    private void btnconsultar_Click(object sender, EventArgs e)
    {
        this.Hide();
        Consultar consultar = new Consultar();
        consultar.FormClosed += (s, args) => this.Close();
        consultar.ShowDialog();
    }

    private void FrmLogin_Load(object sender, EventArgs e)
    {
    }
}
}
}

```

- FrmMarca

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace ProjetoLocadora
{
    public partial class FrmMarca : Form
    {
        // Criar variável conexão
        string conexao =
        ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionString;

        public FrmMarca()
        {
            InitializeComponent();
        }
    }
}

```

```

        //Configurações do botão VOLTAR
private void btnvoltar_Click_1(object sender, EventArgs e)
{
    this.Hide();
    FrmLogin frmLogin = new FrmLogin();
    frmLogin.ShowDialog();
}

private void btnpesquisar_Click(object sender, EventArgs e)
{
    PesquisaMarca marca = new PesquisaMarca();
    marca.ShowDialog();
}

        //Configurações botão CADASTRAR
private void btncadastrar_Click(object sender, EventArgs e)
{
    try
    {
        MySqlConnection con = new MySqlConnection(conexao);

        string nome;
        nome = txtnome.Text;

        if (nome == string.Empty)
        {
            MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o
corretamente.");
        }

        else
        {
            MessageBox.Show("Cadastro efetuado com sucesso!");

            txtcodigo.Text = String.Empty;
            txtnome.Text = String.Empty;

            string sql_insert = @"insert into TB_MARCA
                (TB_MARCA_NOME)
            values (@MARCA_NOME)";

```

```

        MySqlCommand executacmdMySql_insert = new MySqlCommand(sql_insert,
con);

        executacmdMySql_insert.Parameters.AddWithValue("@MARCA_NOME",
nome);

        con.Open();
        executacmdMySql_insert.ExecuteNonQuery();

        string sql_select_marca = "select * from TB_MARCA";

        MySqlCommand executacmdMySql_select_marca = new
MySqlCommand(sql_select_marca, con);
        executacmdMySql_select_marca.ExecuteNonQuery();

        DataTable tabela_marca = new DataTable();

        MySqlDataAdapter da_marca = new
MySqlDataAdapter(executacmdMySql_select_marca);
        da_marca.Fill(tabela_marca);

        dgvmarca.DataSource = tabela_marca;

        con.Close();
    }

}

catch (Exception)
{
    MessageBox.Show("Ops! Ocorreu um erro, esses dados já foram cadastrados!
\r\nTente novamente.");
}

}

//Configurações botão ATUALIZAR
private void btnatualizar_Click(object sender, EventArgs e)
{
    try
    {
        btncadastrar.Enabled = true;

        int codigo;
        string nome;

```

```

codigo = int.Parse(txtcodigo.Text);
nome = txtnome.Text;

if (nome == string.Empty)
{
    MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o
corretamente.");
}

else
{
    DialogResult respostaeditar;

    respostaeditar = MessageBox.Show("Deseja editar o registro selecionado? Os
dados serão permanentemente alterados e não será possível recuperá-los!", "",
    MessageBoxButtons.YesNo, MessageBoxIcon.Information,
    MessageBoxDefaultButton.Button2);

    if (respostaeditar == DialogResult.Yes)
    {

        MySqlConnection con = new MySqlConnection(conexao);

        string sql_update_marca = @"update TB_MARCA
set TB_MARCA_NOME = @MARCA_NOME
where TB_MARCA_ID = @id";

        con.Open();

        MySqlCommand executacmdMySql_update_marca = new
MySqlCommand(sql_update_marca, con);

        executacmdMySql_update_marca.Parameters.AddWithValue("@id",
codigo);
        executacmdMySql_update_marca.Parameters.AddWithValue("@MARCA_NOME", nome);

        executacmdMySql_update_marca.ExecuteNonQuery();

        MessageBox.Show("Seus dados foram alterados com sucesso!");

        txtcodigo.Text = String.Empty;
        txtnome.Text = String.Empty;

        string sqlstm = "Select * from TB_MARCA";
        MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
        DataSet DS = new System.Data.DataSet();
        SDA.Fill(DS, "TB_MARCA");
        dgvmarca.DataSource = DS.Tables[0];

```



```

        btncadastrar.Enabled = true;

        con.Close();
    }
}

catch
{
    MessageBox.Show("Esses dados já foram cadastrados! \r\nTente novamente.");
}
}

//Configurações do botão LIMPAR
private void btnlimpar_Click(object sender, EventArgs e)
{
    txtcodigo.Text = String.Empty;
    txtnome.Text = String.Empty;

    btncadastrar.Enabled = true;
}

//Configurações Load FMRCARGO
private void FrmMarca_Load(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_marca = "select * from TB_MARCA ORDER BY TB_MARCA_ID";

    MySqlCommand executacmdMySql_select_marca = new
MySqlCommand(sql_select_marca, con);
    executacmdMySql_select_marca.ExecuteNonQuery();

    DataTable tabela_marca = new DataTable();

    MySqlDataAdapter da_marca = new
MySqlDataAdapter(executacmdMySql_select_marca);
    da_marca.Fill(tabela_marca);

    dgvmarca.DataSource = tabela_marca;

//NOME DAS COLUNAS
    dgvmarca.Columns["TB_MARCA_ID"].HeaderText = "ID";
    dgvmarca.Columns["TB_MARCA_NOME"].HeaderText = "NOME";
}

```

```

    con.Close();
}

//Configurações Clique de célula DGVMARCA

private void dgvmarca_CellClick(object sender, DataGridViewCellEventArgs e)
{
    txtcodigo.Text = dgvmarca.CurrentRow.Cells[0].Value.ToString();
    txtnome.Text = dgvmarca.CurrentRow.Cells[1].Value.ToString();

    btncadastrar.Enabled = false;
}

//Configurações botão EXCLUIR
private void btnexcluir_Click(object sender, EventArgs e)
{
    try
    {

        MySqlConnection con = new MySqlConnection(conexao);

        int codigo;
        codigo = int.Parse(txtcodigo.Text);

        con.Open();

        DialogResult respostadeletar;

        respostadeletar = MessageBox.Show("Deseja excluir o registro selecionado? Os
dados serão permanentemente apagado e não será possível recuperá-los!", "",
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2);

        if (respostadeletar == DialogResult.Yes)
        {

            string sql_delete_marca = @"delete from TB_MARCA where TB_MARCA_ID =
@codigo";

            MySqlCommand executarcmdMySql_delete_marca = new

```

```

MySQLCommand(sql_delete_marca, con);
        executarcmdMySQL_delete_marca.Parameters.AddWithValue("@codigo",
codigo);

        executarcmdMySQL_delete_marca.ExecuteNonQuery();

        MessageBox.Show("Registro deletado com sucesso");

        txtcodigo.Text = String.Empty;
        txtnome.Text = String.Empty;

        btnexcluir.Enabled = true;

        string sqlstm = "Select * from TB_MARCA";
        MySQLDataAdapter SDA = new MySQLDataAdapter(sqlstm, con);
        DataSet DS = new System.Data.DataSet();
        SDA.Fill(DS, "TB_MARRCA");
        dgvmarca.DataSource = DS.Tables[0];

    }

    con.Close();

}
catch (Exception)
{
    MessageBox.Show("Os dados não podem ser excluídos pois apresentam valores
nulos ou que estão sendo referenciados em outra tabela. \r\nTente novamente
selecionando uma das células.");

}

}
}
}
}
}

```

- FrmModelo

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;
using MySql.Data;
using MySql.Data.MySqlClient;

```

```

namespace ProjetoLocadora
{
    public partial class FrmModelo : Form
    {
        string conexao =
ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionStr
ing;

        public FrmModelo()
        {
            InitializeComponent();

            //Configurações dos botões exibidos no formulário

private void btnvoltar_Click(object sender, EventArgs e)
{
    this.Hide();
    FrmLogin frmLogin = new FrmLogin();
    frmLogin.ShowDialog();
}

private void btnpesquisar_Click(object sender, EventArgs e)
{
    PesquisaModelo modelo = new PesquisaModelo();
    modelo.ShowDialog();
}

            //Configurações do botão CADASTRAR
private void btncadastrar_Click(object sender, EventArgs e)
{
    try
    {

        MySqlConnection con = new MySqlConnection(conexao);

        string desc;
        desc = txtdescricao.Text;

        if (desc == string.Empty)
        {

            MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o
corretamente.");
        }

        else

```

```

{
    MessageBox.Show("Cadastro efetuado com sucesso!");

    txtcodigo.Text = String.Empty;
    txtdescricao.Text = String.Empty;

    string sql_insert = @"insert into TB_MODELO
        (TB_MODELO_DESC)
    values (@MODELO_DESC)";

    MySqlCommand executacmdMySQL_insert = new MySqlCommand(sql_insert,
con);

    executacmdMySQL_insert.Parameters.AddWithValue("@MODELO_DESC",
desc);

    con.Open();
    executacmdMySQL_insert.ExecuteNonQuery();

    string sql_select_modelo = "select * from TB_MODELO";

    MySqlCommand executacmdMySQL_select_modelo = new
MySqlCommand(sql_select_modelo, con);
    executacmdMySQL_select_modelo.ExecuteNonQuery();

    DataTable tabela_modelo = new DataTable();

    MySqlDataAdapter da_modelo = new
MySqlDataAdapter(executacmdMySQL_select_modelo);
    da_modelo.Fill(tabela_modelo);

    dgvmodelo.DataSource = tabela_modelo;

    con.Close();
}

}

catch (Exception)
{
    MessageBox.Show("Ops! Ocorreu um erro, esses dados já foram cadastrados!
\r\nTente novamente.");
}

}

```

```

//Configurações botão ATUALIZAR

private void btnatualizar_Click_1(object sender, EventArgs e)
{
    try
    {
        btncadastrar.Enabled = true;

        int codigo;
        string desc;

        codigo = int.Parse(txtcodigo.Text);
        desc = txtdescricao.Text;

        if (desc == string.Empty)
        {
            MessageBox.Show("O campo é OBRIGATÓRIO. \r\nPreencha-o
corretamente.");
        }

        else
        {
            DialogResult respostaeditar;

            respostaeditar = MessageBox.Show("Deseja editar o registro selecionado? Os
dados serão permanentemente alterados e não será possível recuperá-los!", "",
            MessageBoxButtons.YesNo, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button2);

            if (respostaeditar == DialogResult.Yes)
            {
                MySqlConnection con = new MySqlConnection(conexao);

                string sql_update_modelo = @"update TB_MODELO
set TB_MODELO_DESC = @MODELO_DESC
where TB_MODELO_ID = @id";

                con.Open();

                MySqlCommand executacmdMySQL_update_modelo = new
MySqlCommand(sql_update_modelo, con);

                executacmdMySQL_update_modelo.Parameters.AddWithValue("@id",
codigo);
            }
        }
    }
}

```

```
executacmdMySql_update_modelo.Parameters.AddWithValue("@MODELO_DESC",  
desc);
```

```
    executacmdMySql_update_modelo.ExecuteNonQuery();
```

```
    MessageBox.Show("Seus dados foram alterados com sucesso!");
```

```
    txtcodigo.Text = String.Empty;  
    txtdescricao.Text = String.Empty;
```

```
    string sqlstm = "Select * from TB_MODELO";  
    MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);  
    DataSet DS = new System.Data.DataSet();  
    SDA.Fill(DS, "TB_MODELO");  
    dgvmodelo.DataSource = DS.Tables[0];
```

```
    btncadastrar.Enabled = true;
```

```
        con.Close();  
    }  
}  
}  
  
catch  
{  
    MessageBox.Show("Esses dados já foram cadastrados! \r\nTente novamente.");  
}  
}
```

```
//Configurações do botão LIMPAR  
private void btnlimpar_Click(object sender, EventArgs e)  
{  
    txtcodigo.Text = String.Empty;  
    txtdescricao.Text = String.Empty;  
  
    btncadastrar.Enabled = true;  
}
```

```
//Configurações de Load FRMMODELO  
  
private void FrmModelo_Load(object sender, EventArgs e)  
{  
    MySqlConnection con = new MySqlConnection(conexao);  
  
    con.Open();
```

```

        string sql_select_modelo = "select * from TB_MODELO ORDER BY
TB_MODELO_ID ";
        MySqlCommand executacmdMySql_select_modelo = new
MySqlCommand(sql_select_modelo, con);
        executacmdMySql_select_modelo.ExecuteNonQuery();

        DataTable tabela_modelo = new DataTable();

        MySqlDataAdapter da_modelo = new
MySqlDataAdapter(executacmdMySql_select_modelo);
        da_modelo.Fill(tabela_modelo);

        dgvmodelo.DataSource = tabela_modelo;

//NOME DAS COLUNAS
dgvmodelo.Columns["TB_MODELO_ID"].HeaderText = "ID";
dgvmodelo.Columns["TB_MODELO_DESC"].HeaderText = "DESCRIÇÃO";

        con.Close();
    }

```

```

//Configurações Clique de célula DGVMODELO
private void dgvmodelo_CellClick(object sender, DataGridViewCellEventArgs e)
{
    txtcodigo.Text = dgvmodelo.CurrentRow.Cells[0].Value.ToString();
    txtdescricao.Text = dgvmodelo.CurrentRow.Cells[1].Value.ToString();

    btncadastrar.Enabled = false;
}

```

```

//Configurações botão EXCLUIR
private void btnexcluir_Click(object sender, EventArgs e)
{
    try
    {

        MySqlConnection con = new MySqlConnection(conexao);

        int codigo;

```



```

codigo = int.Parse(txtcodigo.Text);

con.Open();

DialogResult respostadeletar;

    respostadeletar = MessageBox.Show("Deseja excluir o registro selecionado? Os
dados serão permanentemente apagado e não será possível recuperá-los!", "",
    MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
    MessageBoxDefaultButton.Button2);

    if (respostadeletar == DialogResult.Yes)
    {

        string sql_delete_modelo = @"delete from TB_MODELO where
TB_MODELO_ID = @codigo";

        MySqlCommand executarcmdMySql_delete_modelo = new
MySqlCommand(sql_delete_modelo, con);
        executarcmdMySql_delete_modelo.Parameters.AddWithValue("@codigo",
codigo);

        executarcmdMySql_delete_modelo.ExecuteNonQuery();

        MessageBox.Show("Registro deletado com sucesso");

        txtcodigo.Text = String.Empty;
        txtdescricao.Text = String.Empty;

        btnexcluir.Enabled = true;

        string sqlstm = "Select * from TB_MODELO";
        MySqlDataAdapter SDA = new MySqlDataAdapter(sqlstm, con);
        DataSet DS = new System.Data.DataSet();
        SDA.Fill(DS, "TB_CARGO");
        dgvmodelo.DataSource = DS.Tables[0];

    }

    con.Close();

}
catch (Exception)
{
    MessageBox.Show("Os dados não podem ser excluídos pois apresentam valores
nulos ou que estão sendo referenciados em outra tabela. \r\nTente novamente
selecionando uma das células.");

}
}

```

```
}  
}
```

- PesquisaAutomovel.cs

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Configuration;  
using MySql.Data;  
using MySql.Data.MySqlClient;  
  
namespace ProjetoLocadora  
{  
    public partial class PesquisaAutomovel : Form  
    {  
        // Criar variável conexão  
        string conexao =  
ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionStr  
ing;  
  
        public PesquisaAutomovel()  
        {  
            InitializeComponent();  
        }  
  
        private void btnautomoveis_Click(object sender, EventArgs e)  
        {  
            this.Hide();  
            FrmAutomovel frmautomovel = new FrmAutomovel();  
            frmautomovel.ShowDialog();  
        }  
  
        private void btnvoltar_Click(object sender, EventArgs e)  
        {  
            this.Hide();  
            Consultar consulta = new Consultar();  
            consulta.ShowDialog();  
        }  
    }  
}
```

```

private void btnpesquisar_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_automovel = "select * from TB_AUTOMOVEL ORDER BY
TB_AUTOMOVEL_ANO_FAB DESC";

    MySqlCommand executacmdMySql_select_automovel = new
MySqlCommand(sql_select_automovel, con);
    executacmdMySql_select_automovel.ExecuteNonQuery();

    DataTable tabela_automovel = new DataTable();

    MySqlDataAdapter da_automovel = new
MySqlDataAdapter(executacmdMySql_select_automovel);
    da_automovel.Fill(tabela_automovel);

    dgvpesquisaautomovel.DataSource = tabela_automovel;

    switch (cmbtipopesquisa.Text)
    {

        case "Nome":
            tabela_automovel.DefaultView.RowFilter = string.Format("[{0}] LIKE '%{1}%",
"TB_AUTOMOVEL_NOME", txtpesquisa.Text);
            txtpesquisa.Text = String.Empty;
            break;

        case "Cor":
            tabela_automovel.DefaultView.RowFilter = string.Format("[{0}] LIKE '%{1}%",
"TB_AUTOMOVEL_COR", txtpesquisa.Text);
            txtpesquisa.Text = String.Empty;
            break;

        case "Status":
            tabela_automovel.DefaultView.RowFilter = string.Format("[{0}] LIKE '%{1}%",
"TB_AUTOMOVEL_STATUS", txtpesquisa.Text);
            txtpesquisa.Text = String.Empty;
            break;

        case "Marca":
            tabela_automovel.DefaultView.RowFilter =
string.Format("CONVERT(TB_MARCA_ID, 'System.String') LIKE '{1}'", "TB_MARCA_ID",
txtpesquisa.Text);
            txtpesquisa.Text = String.Empty;
            break;

        case "Modelo":

```

```

        tabela_automovel.DefaultView.RowFilter =
string.Format("CONVERT(TB_MODELO_ID, 'System.String') LIKE '{1}'",
"TB_MODELO_ID", txtpesquisa.Text);
        txtpesquisa.Text = String.Empty;
        break;

```

```

    }
}

```

```

private void PesquisaAutomovel_Load(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_automovel = "select * from TB_AUTOMOVEL ORDER BY
TB_AUTOMOVEL_ANO_FAB DESC";

    MySqlCommand executacmdMySql_select_automovel = new
MySqlCommand(sql_select_automovel, con);
    executacmdMySql_select_automovel.ExecuteNonQuery();

    DataTable tabela_automovel = new DataTable();

    MySqlDataAdapter da_automovel = new
MySqlDataAdapter(executacmdMySql_select_automovel);
    da_automovel.Fill(tabela_automovel);

    dgvpesquisaautomovel.DataSource = tabela_automovel;

    //Nome Cor Status Marca Modelo
    dgvpesquisaautomovel.Columns["TB_AUTOMOVEL_ID"].HeaderText = "ID";
    dgvpesquisaautomovel.Columns["TB_AUTOMOVEL_NOME"].HeaderText =
"NOME";
    dgvpesquisaautomovel.Columns["TB_AUTOMOVEL_ANO_FAB"].HeaderText =
"ANO FAB";
    dgvpesquisaautomovel.Columns["TB_AUTOMOVEL_COR"].HeaderText = "COR";
    dgvpesquisaautomovel.Columns["TB_AUTOMOVEL_KM"].HeaderText = "KM";
    dgvpesquisaautomovel.Columns["TB_AUTOMOVEL_VALOR_D"].HeaderText =
"VALOR";
    dgvpesquisaautomovel.Columns["TB_AUTOMOVEL_STATUS"].HeaderText =
"STATUS";
    dgvpesquisaautomovel.Columns["TB_MARCA_ID"].HeaderText = "MARCA";
    dgvpesquisaautomovel.Columns["TB_MODELO_ID"].HeaderText = "MODELO";

}

private void dgvpesquisaautomovel_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    switch (cmbtipopesquisa.Text)

```

```

    {
        case "Nome":
            txtpesquisa.Text =
                dgvpesquisaautomovel.CurrentRow.Cells[1].Value.ToString();
            break;

        case "Cor":
            txtpesquisa.Text =
                dgvpesquisaautomovel.CurrentRow.Cells[3].Value.ToString();
            break;

        case "Status":
            txtpesquisa.Text =
                dgvpesquisaautomovel.CurrentRow.Cells[6].Value.ToString();
            break;

        case "Marca":
            txtpesquisa.Text =
                dgvpesquisaautomovel.CurrentRow.Cells[7].Value.ToString();
            break;

        case "Modelo":
            txtpesquisa.Text =
                dgvpesquisaautomovel.CurrentRow.Cells[8].Value.ToString();
            break;
    }
}
}
}
}

```

- PesquisaCargo.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;
using MySql.Data;
using MySql.Data.MySqlClient;
namespace ProjetoLocadora
{
    public partial class PesquisaCargo : Form
    {

        // Criar variável conexão

```

```

string conexao =
ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionStr
ing;

public object DefaultView { get; private set; }

public PesquisaCargo()
    {
        InitializeComponent();
    }

private void btncargos_Click(object sender, EventArgs e)
{
    this.Hide();
    FrmCargo frmcargo = new FrmCargo();
    frmcargo.ShowDialog();

}

private void btnvoltar_Click(object sender, EventArgs e)
{
    this.Hide();
    Consultar consulta = new Consultar();
    consulta.ShowDialog();

}

private void btnpesquisar_Click(object sender, EventArgs e)
{
    FiltrarCargo();
}

private void FiltrarCargo()
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_cargo = "select * from TB_CARGO ORDER BY TB_CARGO_ID";

    MySqlCommand executacmdMySql_select_cargo = new
MySqlCommand(sql_select_cargo, con);
    executacmdMySql_select_cargo.ExecuteNonQuery();

    DataTable tabela_cargo = new DataTable();

    MySqlDataAdapter da_cargo = new
MySqlDataAdapter(executacmdMySql_select_cargo);
    da_cargo.Fill(tabela_cargo);

    dgvpesquisacargo.DataSource = tabela_cargo;
}

```

```

switch (cmbtipopesquisa.Text)
{

    // PESQUISA POR VARIOS CAMPOS

    // 1. Atribuir texto utilizado pela combobox
    // 2. Atribuir o nome da tabela utilizada
    // 3. Definir o formato da cadeia de caracteres utilizada (string), se caso for inteiro
/ decimal, converter utilizando CONVERT('NOME_TABELA', 'System.String')
    // 4. [{0}] = (recebe) "TB_CARGO_NOME"    '{1}%' = (recebe) txtpesquisa.Text
    // 5. Quando TB_CARGO_NOME for igual as informações inseridas na
txtpesquisa, exibirá seu resultado no dgv

    case "Nome":
        tabela_cargo.DefaultView.RowFilter = string.Format("[{0}] LIKE '{1}%",
"TB_CARGO_NOME", txtpesquisa.Text);
        txtpesquisa.Text = String.Empty;
        break;

    case "ID":
        tabela_cargo.DefaultView.RowFilter =
string.Format("CONVERT(TB_CARGO_ID, 'System.String') LIKE '{1}'", "TB_CARGO_ID",
txtpesquisa.Text);
        txtpesquisa.Text = String.Empty;
        break;

}

}

```

```

private void PesquisaCargo_Load(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_cargo = "select * from TB_CARGO ORDER BY TB_CARGO_ID";

    MySqlCommand executacmdMySql_select_cargo = new
MySqlCommand(sql_select_cargo, con);
    executacmdMySql_select_cargo.ExecuteNonQuery();

    DataTable tabela_cargo = new DataTable();

    MySqlDataAdapter da_cargo = new
MySqlDataAdapter(executacmdMySql_select_cargo);
    da_cargo.Fill(tabela_cargo);

    dgvpesquisacargo.DataSource = tabela_cargo;
}

```

```

dgvpesquisacargo.Columns["TB_CARGO_ID"].HeaderText = "ID";
dgvpesquisacargo.Columns["TB_CARGO_NOME"].HeaderText = "NOME";

}

private void dgvpesquisacargo_CellClick(object sender, DataGridViewCellEventArgs
e)
{
    switch (cmbtipopesquisa.Text)
    {
        case "Nome":
            txtpesquisa.Text = dgvpesquisacargo.CurrentRow.Cells[1].Value.ToString();
            break;

        case "ID":
            txtpesquisa.Text = dgvpesquisacargo.CurrentRow.Cells[0].Value.ToString();
            break;
    }
}
}
}
}

```

- PesquisaCliente.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace ProjetoLocadora
{
    public partial class PesquisaCliente : Form
    {

        // Criar variável conexão
        string conexao =
ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionString;

        public PesquisaCliente()
        {

```



```

        InitializeComponent();
    }

//Configurações dos botões exibidos no Formulário
private void btnvoltar_Click_1(object sender, EventArgs e)
{
    this.Hide();
    Consultar consulta = new Consultar();
    consulta.ShowDialog();
}

private void btnclientes_Click(object sender, EventArgs e)
{
    this.Hide();
    FrmCliente frmcliente = new FrmCliente();
    frmcliente.ShowDialog();
}

//Configurações botão COLAR
private void pbcolar_Click(object sender, EventArgs e)
{
    if (Clipboard.ContainsText())
        txtpesquisa.Text = Clipboard.GetText();
}

private void btnpesquisar_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_cliente = "select * from TB_CLIENTE ORDER BY
TB_CLIENTE_NOME";

    MySqlCommand executacmdMySQL_select_cliente = new
MySqlCommand(sql_select_cliente, con);
    executacmdMySQL_select_cliente.ExecuteNonQuery();

    DataTable tabela_cliente = new DataTable();

    MySqlDataAdapter da_cliente = new
MySqlDataAdapter(executacmdMySQL_select_cliente);
    da_cliente.Fill(tabela_cliente);

    dgvpesquisacliente.DataSource = tabela_cliente;
}

```

```

switch (cmbtipopesquisa.Text)
{
    //Senha Nome Cidade Unidade Federativa

    case "Nome":
        tabela_cliente.DefaultView.RowFilter = string.Format("[{0}] LIKE '%{1}%',",
"TB_CLIENTE_NOME", txtpesquisa.Text);
        txtpesquisa.Text = String.Empty;
        break;

    case "Cidade":
        tabela_cliente.DefaultView.RowFilter = string.Format("[{0}] LIKE '%{1}%',",
"TB_CLIENTE_CIDADE", txtpesquisa.Text);
        txtpesquisa.Text = String.Empty;
        break;

    case "Unidade Federativa":
        tabela_cliente.DefaultView.RowFilter = string.Format("[{0}] LIKE '%{1}%',",
"TB_CLIENTE_UF", txtpesquisa.Text);
        txtpesquisa.Text = String.Empty;
        break;

    case "Senha":
        tabela_cliente.DefaultView.RowFilter =
string.Format("CONVERT(TB_CLIENTE_SENHA, 'System.String') LIKE '{1}'",
"TB_CARGO_ID", txtpesquisa.Text);
        txtpesquisa.Text = String.Empty;
        break;

}
}

private void PesquisaCliente_Load(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_cliente = "select * from TB_CLIENTE ORDER BY
TB_CLIENTE_NOME";

    MySqlCommand executacmdMySql_select_cliente = new
MySqlCommand(sql_select_cliente, con);
    executacmdMySql_select_cliente.ExecuteNonQuery();

    DataTable tabela_cliente = new DataTable();

    MySqlDataAdapter da_cliente = new
MySqlDataAdapter(executacmdMySql_select_cliente);
    da_cliente.Fill(tabela_cliente);

    dgvpesquisacliente.DataSource = tabela_cliente;

    dgvpesquisacliente.Columns["TB_CLIENTE_ID"].HeaderText = "ID";

```

```

        dgvpesquisacliente.Columns["TB_CLIENTE_NOME"].HeaderText = "NOME";
        dgvpesquisacliente.Columns["TB_CLIENTE_TEL"].HeaderText = "TELEFONE";
        dgvpesquisacliente.Columns["TB_CLIENTE_SEXO"].HeaderText = "SEXO";
        dgvpesquisacliente.Columns["TB_CLIENTE_EMAIL"].HeaderText = "EMAIL";
        dgvpesquisacliente.Columns["TB_CLIENTE_SENHA"].HeaderText = "SENHA";
        dgvpesquisacliente.Columns["TB_CLIENTE_ENDERECO"].HeaderText =
"ENDEREÇO";
        dgvpesquisacliente.Columns["TB_CLIENTE_COMPLEMENTO"].HeaderText =
"COMPLEMENTO";
        dgvpesquisacliente.Columns["TB_CLIENTE_BAIRRO"].HeaderText = "BAIRRO";
        dgvpesquisacliente.Columns["TB_CLIENTE_CIDADE"].HeaderText = "CIDADE";
        dgvpesquisacliente.Columns["TB_CLIENTE_UF"].HeaderText = "UF";
        dgvpesquisacliente.Columns["TB_CLIENTE_DT_NASC"].HeaderText = "DATA
NASC";
        dgvpesquisacliente.Columns["TB_CLIENTE_DT_CAD"].HeaderText = "DATA
CAD";

    }

    private void dgvpesquisacliente_CellClick(object sender, DataGridViewCellEventArgs
e)
    {

        switch (cmbtipopesquisa.Text)
        {
            case "Nome":
                txtpesquisa.Text = dgvpesquisacliente.CurrentRow.Cells[1].Value.ToString();
                break;

            case "Cidade":
                txtpesquisa.Text = dgvpesquisacliente.CurrentRow.Cells[9].Value.ToString();
                break;

            case "Unidade Federativa":
                txtpesquisa.Text = dgvpesquisacliente.CurrentRow.Cells[10].Value.ToString();
                break;

            case "Senha":
                txtpesquisa.Text = dgvpesquisacliente.CurrentRow.Cells[5].Value.ToString();
                break;
        }

    }

}
}
}

```

- PesquisaFuncionario.cs

```

using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace ProjetoLocadora
{
    public partial class PesquisaFuncionario : Form
    {
        string conexao =
ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionStr
ing;

        public PesquisaFuncionario()
        {
            InitializeComponent();
        }

        private void btnvoltar_Click(object sender, EventArgs e)
        {
            this.Hide();
            Consultar consulta = new Consultar();
            consulta.ShowDialog();
        }

        private void btnfuncionarios_Click(object sender, EventArgs e)
        {
            this.Hide();
            FrmFuncionario frmfuncionario = new FrmFuncionario();
            frmfuncionario.ShowDialog();
        }

        private void btnpesquisar_Click(object sender, EventArgs e)
        {
            MySqlConnection con = new MySqlConnection(conexao);

            con.Open();

            string sql_select_funcionario = "select * from TB_FUNCIONARIO ORDER BY
TB_FUNCIONARIO_NOME";

            MySqlCommand executacmdMySql_select_funcionario = new
MySqlCommand(sql_select_funcionario, con);
            executacmdMySql_select_funcionario.ExecuteNonQuery();

            DataTable tabela_funcionario = new DataTable();

```

```

        MySqlDataAdapter da_funcionario = new
        MySqlDataAdapter(executacmdMySql_select_funcionario);
        da_funcionario.Fill(tabela_funcionario);

        dgvpesquisafuncionario.DataSource = tabela_funcionario;

        switch (cmbtipopesquisa.Text)
        {

            case "Nome":
                tabela_funcionario.DefaultView.RowFilter = string.Format("[{0}] LIKE '%{1}%' ",
                "TB_FUNCIONARIO_NOME", txtpesquisa.Text);
                txtpesquisa.Text = String.Empty;
                break;

            case "Cargo":
                tabela_funcionario.DefaultView.RowFilter =
                string.Format("CONVERT(TB_CARGO_ID, 'System.String') LIKE '{1}'", "TB_CARGO_ID",
                txtpesquisa.Text);
                txtpesquisa.Text = String.Empty;
                break;

        }
    }

    private void PesquisaFuncionario_Load(object sender, EventArgs e)
    {
        MySqlConnection con = new MySqlConnection(conexao);

        con.Open();

        string sql_select_funcionario = "select * from TB_FUNCIONARIO ORDER BY
        TB_FUNCIONARIO_NOME";

        MySqlCommand executacmdMySql_select_funcionario = new
        MySqlCommand(sql_select_funcionario, con);
        executacmdMySql_select_funcionario.ExecuteNonQuery();

        DataTable tabela_funcionario = new DataTable();

        MySqlDataAdapter da_funcionario = new
        MySqlDataAdapter(executacmdMySql_select_funcionario);
        da_funcionario.Fill(tabela_funcionario);

        dgvpesquisafuncionario.DataSource = tabela_funcionario;

        dgvpesquisafuncionario.Columns["TB_FUNCIONARIO_ID"].HeaderText = "ID";
        dgvpesquisafuncionario.Columns["TB_FUNCIONARIO_NOME"].HeaderText =
        "NOME";
        dgvpesquisafuncionario.Columns["TB_FUNCIONARIO_TEL"].HeaderText =

```

```
"TELEFONE";
```

```
dgvpesquisafuncionario.Columns["TB_FUNCIONARIO_DT_CONTRATO"].HeaderText =  
"DATA CONT";
```

```
    dgvpesquisafuncionario.Columns["TB_CARGO_ID"].HeaderText = "CARGO";
```

```
    }
```

```
    private void dgvpesquisafuncionario_CellClick(object sender,  
DataGridViewCellEventArgs e)
```

```
    {
```

```
        switch (cmbtipopesquisa.Text)
```

```
        {
```

```
            case "Nome":
```

```
                txtpesquisa.Text =
```

```
dgvpesquisafuncionario.CurrentRow.Cells[1].Value.ToString();
```

```
                break;
```

```
            case "Cargo":
```

```
                txtpesquisa.Text =
```

```
dgvpesquisafuncionario.CurrentRow.Cells[4].Value.ToString();
```

```
                break;
```

```
        }
```

```
    }
```

```
    }
```

```
}
```

- PesquisaLocacao.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel;
```

```
using System.Data;
```

```
using System.Drawing;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
using System.Windows.Forms;
```

```
using System.Configuration;
```

```
using MySql.Data;
```

```
using MySql.Data.MySqlClient;
```

```
namespace ProjetoLocadora
```

```
{
```

```
    public partial class PesquisaLocacao : Form
```

```
    {
```

```
        string conexao =
```

```
ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionString;
```

```
        public PesquisaLocacao()
```

```
        {
```

```

        InitializeComponent();
    }

    private void btnvoltar_Click(object sender, EventArgs e)
    {
        this.Hide();
        Consultar consulta = new Consultar();
        consulta.ShowDialog();
    }

    private void btnlocacoes_Click(object sender, EventArgs e)
    {
        this.Hide();
        FrmLocacao frmlocacao = new FrmLocacao();
        frmlocacao.ShowDialog();
    }

    private void btnpesquisar_Click(object sender, EventArgs e)
    {
        MySqlConnection con = new MySqlConnection(conexao);

        con.Open();

        string sql_select_locacao = "select * from TB_LOCACAO ORDER BY
TB_LOCACAO_ID";

        MySqlCommand executacmdMySQL_select_locacao = new
        MySqlCommand(sql_select_locacao, con);
        executacmdMySQL_select_locacao.ExecuteNonQuery();

        DataTable tabela_locacao = new DataTable();

        MySqlDataAdapter da_locacao = new
        MySqlDataAdapter(executacmdMySQL_select_locacao);
        da_locacao.Fill(tabela_locacao);

        dgvpesquisalocacao.DataSource = tabela_locacao;

        switch (cmbtipopesquisa.Text)
        {
            case "Tipo":
                tabela_locacao.DefaultView.RowFilter = string.Format("[{0}] LIKE '%{1}%',
                "TB_LOCACAO_TIPO", txtpesquisa.Text);
                txtpesquisa.Text = String.Empty;
                break;

            case "Cliente":

```

```

        tabela_locacao.DefaultView.RowFilter =
string.Format("CONVERT(TB_CLIENTE_ID, 'System.String') LIKE '{1}'",
"TB_CLIENTE_ID", txtpesquisa.Text);
        txtpesquisa.Text = String.Empty;
        break;

        case "Funcionario":
            tabela_locacao.DefaultView.RowFilter =
string.Format("CONVERT(TB_FUNCIONARIO_ID, 'System.String') LIKE '{1}'",
"TB_FUNCIONARIO_ID", txtpesquisa.Text);
            txtpesquisa.Text = String.Empty;
            break;

        case "Automóvel":
            tabela_locacao.DefaultView.RowFilter =
string.Format("CONVERT(TB_AUTOMOVEL_ID, 'System.String') LIKE '{1}'",
"TB_AUTOMOVEL_ID", txtpesquisa.Text);
            txtpesquisa.Text = String.Empty;
            break;
    }

}

private void PesquisaLocacao_Load(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_locacao = "select * from TB_LOCACAO ORDER BY
TB_LOCACAO_ID";

    MySqlCommand executacmdMySql_select_locacao = new
MySqlCommand(sql_select_locacao, con);
    executacmdMySql_select_locacao.ExecuteNonQuery();

    DataTable tabela_locacao = new DataTable();

    MySqlDataAdapter da_locacao = new
MySqlDataAdapter(executacmdMySql_select_locacao);
    da_locacao.Fill(tabela_locacao);

    dgvpesquisalocacao.DataSource = tabela_locacao;

    //NOME DAS COLUNAS
    dgvpesquisalocacao.Columns["TB_LOCACAO_ID"].HeaderText = "ID";
    dgvpesquisalocacao.Columns["TB_LOCACAO_TIPO"].HeaderText = "TIPO";
    dgvpesquisalocacao.Columns["TB_LOCACAO_VALOR"].HeaderText = "VALOR";
    dgvpesquisalocacao.Columns["TB_LOCACAO_DT_INICIO"].HeaderText = "DATA
INÍCIO";
}

```



```

        dgvpesquisalocacao.Columns["TB_LOCACAO_DT_FIM"].HeaderText = "DATA
FIM";
        dgvpesquisalocacao.Columns["TB_CLIENTE_ID"].HeaderText = "CLIENTE";
        dgvpesquisalocacao.Columns["TB_FUNCIONARIO_ID"].HeaderText =
"FUNCIONARIO";
        dgvpesquisalocacao.Columns["TB_AUTOMOVEL_ID"].HeaderText =
"AUTOMOVEL";

        con.Close();

    }

    private void dgvpesquisalocacao_CellClick(object sender, DataGridViewCellEventArgs
e)
    {
        switch (cmbtipopesquisa.Text)
        {
            case "Tipo":
                txtpesquisa.Text = dgvpesquisalocacao.CurrentRow.Cells[1].Value.ToString();
                break;

            case "Cliente":
                txtpesquisa.Text = dgvpesquisalocacao.CurrentRow.Cells[5].Value.ToString();
                break;

            case "Funcionário":
                txtpesquisa.Text = dgvpesquisalocacao.CurrentRow.Cells[6].Value.ToString();
                break;

            case "Automóvel":
                txtpesquisa.Text = dgvpesquisalocacao.CurrentRow.Cells[7].Value.ToString();
                break;
        }
    }
}
}
}
}
}

```

- PesquisaMarca.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace ProjetoLocadora

```

```

{
    public partial class PesquisaMarca : Form
    {
        string conexao =
ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionStr
ing;

        public PesquisaMarca()
        {
            InitializeComponent();
        }

private void btnmarcas_Click_1(object sender, EventArgs e)
{
            this.Hide();
            FrmMarca frmmarca = new FrmMarca();
            frmmarca.ShowDialog();
        }

private void btnvoltar_Click_1(object sender, EventArgs e)
{
            this.Hide();
            Consultar consulta = new Consultar();
            consulta.ShowDialog();
        }

private void btnpesquisar_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_marca = "select * from TB_MARCA ORDER BY TB_MARCA_ID";

    MySqlCommand executacmdMySql_select_marca = new
MySqlCommand(sql_select_marca, con);
    executacmdMySql_select_marca.ExecuteNonQuery();

    DataTable tabela_marca = new DataTable();

    MySqlDataAdapter da_marca = new
MySqlDataAdapter(executacmdMySql_select_marca);
    da_marca.Fill(tabela_marca);

    dgvpesquisamarca.DataSource = tabela_marca;

    switch (cmbtipopesquisa.Text)

```

```

    {
        case "Nome":
            tabela_marca.DefaultView.RowFilter = string.Format("[{0}] LIKE '%{1}%",
"TB_MARCA_NOME", txtpesquisa.Text);
            txtpesquisa.Text = String.Empty;
            break;

        case "ID":
            tabela_marca.DefaultView.RowFilter =
string.Format("CONVERT(TB_MARCA_ID, 'System.String') LIKE '{1}'", "TB_MARCA_ID",
txtpesquisa.Text);
            txtpesquisa.Text = String.Empty;
            break;
    }
}

private void PesquisaMarca_Load(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_marca = "select * from TB_MARCA ORDER BY TB_MARCA_ID";

    MySqlCommand executacmdMySql_select_marca = new
MySqlCommand(sql_select_marca, con);
    executacmdMySql_select_marca.ExecuteNonQuery();

    DataTable tabela_marca = new DataTable();

    MySqlDataAdapter da_marca = new
MySqlDataAdapter(executacmdMySql_select_marca);
    da_marca.Fill(tabela_marca);

    dgvpesquisamarca.DataSource = tabela_marca;

    dgvpesquisamarca.Columns["TB_MARCA_ID"].HeaderText = "ID";
    dgvpesquisamarca.Columns["TB_MARCA_NOME"].HeaderText = "NOME";
}

e) private void dgvpesquisamarca_CellClick(object sender, DataGridViewCellEventArgs
{
    switch (cmbtipopesquisa.Text)
    {
        case "Nome":

```

```

        txtpesquisa.Text = dgvpesquisamarca.CurrentRow.Cells[1].Value.ToString();
        break;

        case "ID":
            txtpesquisa.Text = dgvpesquisamarca.CurrentRow.Cells[0].Value.ToString();
            break;
    }
}
}
}

```

- PesquisaModelo.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace ProjetoLocadora
{
    public partial class PesquisaModelo : Form
    {
        string conexao =
        ConfigurationManager.ConnectionStrings["BD_LOCADORA_2MTEC_2023"].ConnectionString;

        public PesquisaModelo()
        {
            InitializeComponent();
        }

        private void btnmodelos_Click(object sender, EventArgs e)
        {
            this.Hide();
            FrmModelo frmmodelo = new FrmModelo();
            frmmodelo.ShowDialog();
        }

        private void btnvoltar_Click(object sender, EventArgs e)
        {
            this.Hide();
            Consultar consulta = new Consultar();
            consulta.ShowDialog();
        }
    }
}

```

```

private void btnpesquisar_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_modelo = "select * from TB_MODELO ORDER BY
TB_MODELO_ID";
    MySqlCommand executacmdMySql_select_modelo = new
MySqlCommand(sql_select_modelo, con);
    executacmdMySql_select_modelo.ExecuteNonQuery();

    DataTable tabela_modelo = new DataTable();

    MySqlDataAdapter da_modelo = new
MySqlDataAdapter(executacmdMySql_select_modelo);
    da_modelo.Fill(tabela_modelo);

    dgvpesquisamodelo.DataSource = tabela_modelo;

    switch (cmbtipopesquisa.Text)
    {

        case "Descrição":
            tabela_modelo.DefaultView.RowFilter = string.Format("[{0}] LIKE '%{1}%',
"TB_MODELO_DESC", txtpesquisa.Text);
            txtpesquisa.Text = String.Empty;
            break;

        case "ID":
            tabela_modelo.DefaultView.RowFilter =
string.Format("CONVERT(TB_MODELO_ID, 'System.String') LIKE '{1}'",
"TB_MODELO_ID", txtpesquisa.Text);
            txtpesquisa.Text = String.Empty;
            break;

    }
}

private void PesquisaModelo_Load(object sender, EventArgs e)
{

    MySqlConnection con = new MySqlConnection(conexao);

    con.Open();

    string sql_select_modelo = "select * from TB_MODELO ORDER BY
TB_MODELO_ID";
    MySqlCommand executacmdMySql_select_modelo = new
MySqlCommand(sql_select_modelo, con);
    executacmdMySql_select_modelo.ExecuteNonQuery();

```

```

        DataTable tabela_modelo = new DataTable();

        MySqlDataAdapter da_modelo = new
        MySqlDataAdapter(executacmdMySql_select_modelo);
        da_modelo.Fill(tabela_modelo);

        dgvpesquisamodelo.DataSource = tabela_modelo;

        dgvpesquisamodelo.Columns["TB_MODELO_ID"].HeaderText = "ID";
        dgvpesquisamodelo.Columns["TB_MODELO_DESC"].HeaderText =
        "DESCRIÇÃO";

    }

    private void dgvpesquisamodelo_CellClick(object sender, DataGridViewCellEventArgs
e)
    {
        switch (cmbtipopesquisa.Text)
        {
            case "Descrição":
                txtpesquisa.Text = dgvpesquisamodelo.CurrentRow.Cells[1].Value.ToString();
                break;

            case "ID":
                txtpesquisa.Text = dgvpesquisamodelo.CurrentRow.Cells[0].Value.ToString();
                break;
        }
    }
}

```

6. Resultado Obtido e considerações finais

As metas preestabelecidas no início do projeto foram integralmente realizadas conforme o planejado.

Para a avaliação de 4º bimestre concluída pelo orientador Cristiano Teixeira, a apresentação integral do Projeto Locadora dispõe formulários de Cadastro de Clientes, Funcionários, Cargos, Automóveis, Modelos, Marcas e Locações; além de proporcionar uma série de funções integradas ao software que foram programadas visando a utilização prática e otimizada do Usuário. Contudo, para o dia 01/12/2023, estão concluídos o Banco de Dados, Diagrama de Casos de Uso, Questionário de Planejamento, Requisitos de Usuário/Sistema e os Formulários de Pesquisa das entidades selecionadas.

O Projeto Locadora nos permitiu instigar a criatividade e conhecimento nas disciplinas de Banco de Dados e Desenvolvimento de Sistemas, de forma que pudemos colocar em prática tudo aquilo que fora trabalhado em sala de aula.

A tecnologia não só automatiza serviços cotidianos, como também nos disponibiliza uma maior autonomia no ambiente corporativo, de forma que se torna crucial seu estímulo e aprendizagem.

7. Referências

- **PESQUISAS ONLINE:**

1. STACK OVERFLOW
<https://pt.stackoverflow.com/>
2. DEV MEDIA:
<https://www.devmedia.com.br/>
3. ALURA:
<https://www.alura.com>
4. MICROSOFT LEARN:
<https://learn.microsoft.com/pt-br/>
5. W3 SCHOOLS:
<https://www.w3schools.com/>
6. DESENVOLVIMENTO ABERTO:
<https://desenvolvimentoaberto.org/>
7. C SHARP CORNER:
<https://www.c-sharpcorner.com/>
8. CRIANDO BITS:
<https://www.criandobits.com.br/>

- **CANAIS NO YOUTUBE:**

1. BÓSON TREINAMENTOS:
<https://www.youtube.com/@bosontreinamentos>
2. PROGRAMAÇÃO PLENA:
<https://www.youtube.com/@ProgramacaoPlena>
3. IGOR VINÍCIUS:
<https://www.youtube.com/@igorvinicius2531>
4. CBF CURSOS
<https://www.youtube.com/@cfbcursos>