

**UNIVERSIDADE NOVE DE JULHO - UNINOVE  
DIRETORIA DOS CURSOS DE INFORMÁTICA**

**MATHEUS HENRIQUE DE MACEDO - RA 918115396**

**Auto Motors do Zé: Sistema para loja de carros**

**SÃO PAULO  
2022**

MATHEUS HENRIQUE DE MACEDO - **RA 918115396**

## **Auto Motors do Zé: Sistema para loja de carros**

Trabalho apresentado à Universidade Nove de Julho, UNINOVE, em cumprimento parcial às exigências da disciplina de Projeto em Desenvolvimento de Software, sob orientação do Prof. **Márcio Romero** e Prof. **Adriano Michelotti Schroeder**.

**SÃO PAULO  
2022**

## **Resumo**

**Palavras chave: Carros, Loja, Automóveis, Seminovos, Veículos.**

## **Abstract**

**Keywords: Cars, Store, Automobiles, Used, Vehicles .**

## **Lista de Figuras**

## **SUMÁRIO**

## **1. INTRODUÇÃO**

### **1.1. RESUMO DE NEGÓCIOS DA STAR CLOVER TECHNOLOGIES**

A Star Clover Technologies é uma empresa de desenvolvimento, que busca transformar a ideia do cliente em realidade, oferecendo as melhores inovações de mobilidade com uma metodologia diferenciada que integra estratégia de ponta a ponta, ajudando os clientes e parceiros desde a concepção da ideia a publicação e divulgação do projeto.

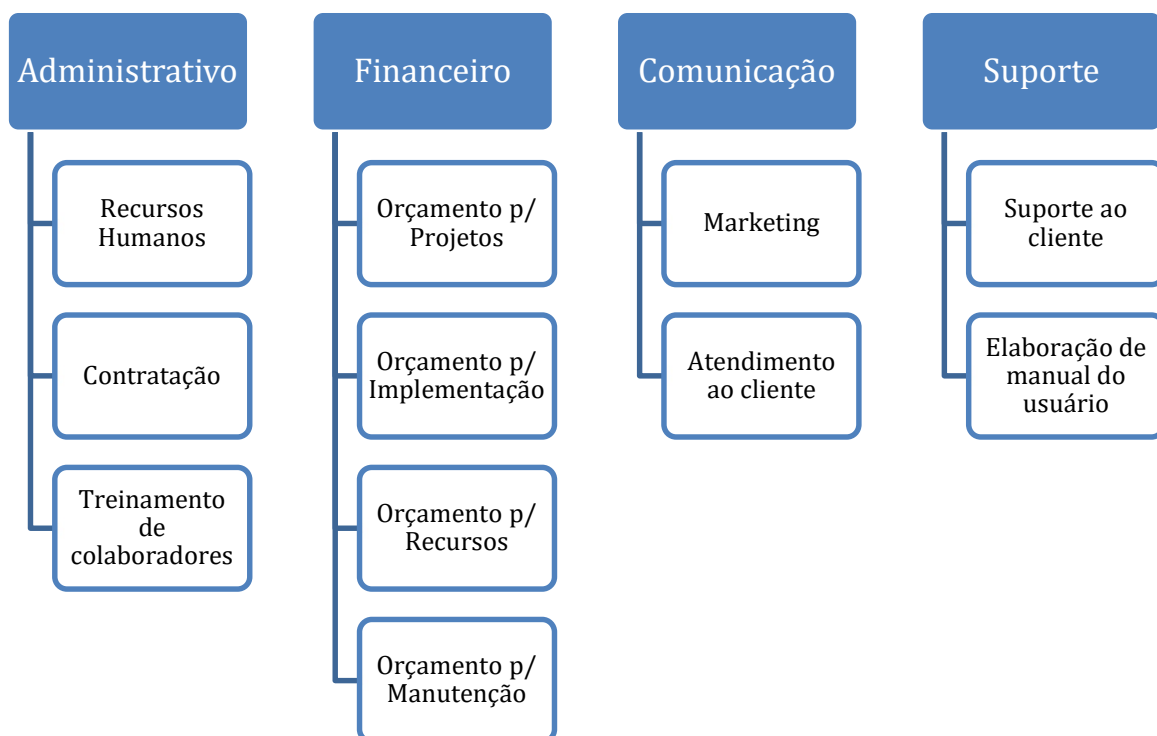
#### **1.1.1. SETORES DA STAR CLOVER TECHNOLOGIES (S.C.T.)**

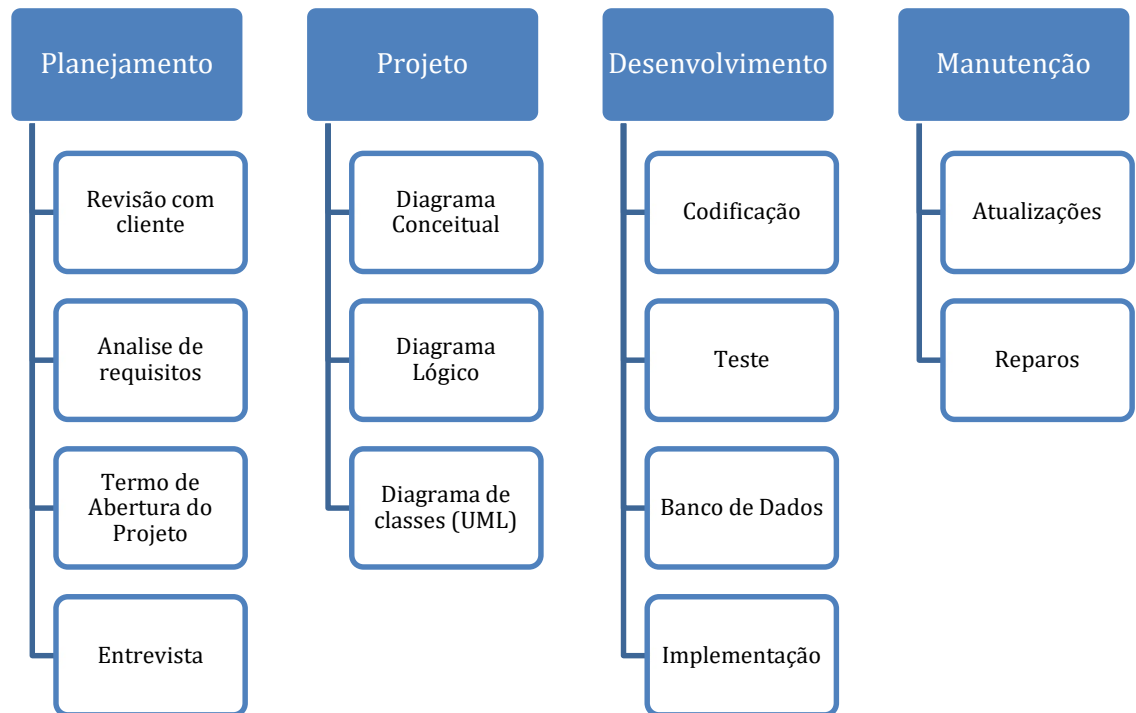
- **Administrativo/Recursos Humanos:** Este setor é responsável por tem como principal função planejar, organizar e gerenciar o uso dos recursos pessoais e financeiros de uma organização. O objetivo é elaborar estratégias para melhorar o desempenho da empresa, maximizar os lucros, evitar desperdícios e reduzir custos;
- **Financeiro:** Este setor é responsável por administrar os recursos de uma empresa. Ele faz o controle da tesouraria, dos investimentos e dos riscos, além do planejamento financeiro da companhia e da divulgação de seus resultados;
- **Comunicação:** Este setor é responsável por fazer com que os públicos interno e

externo da empresa recebam informações relevantes sobre seu trabalho;

- **Planejamento:** Este setor é responsável por fazer a iniciação dos projetos da empresa, de forma a garantir os interesses do cliente e da empresa, elaborando o esboço e a esquematização até a entrega de cada iniciativa;
- **Projetos:** Este setor é responsável por fazer a condução dos projetos da empresa de maneira integrada, fornecendo todo o suporte e desenvolvimento da estrutura inicial de recursos e serviços necessários para o desenvolvimento de cada iniciativa;
- **Desenvolvimento:** Este setor é responsável pela criação e implementação do Sistema;
- **Manutenção:** Este setor é responsável pelas atualizações e correção de bugs no sistema;
- **Suporte:** Este setor é responsável por ajudar e sanar qualquer dúvida de usuários do sistema.

#### 1.1.2. ORGANOGRAMA DOS SETORES DA S.C.T.





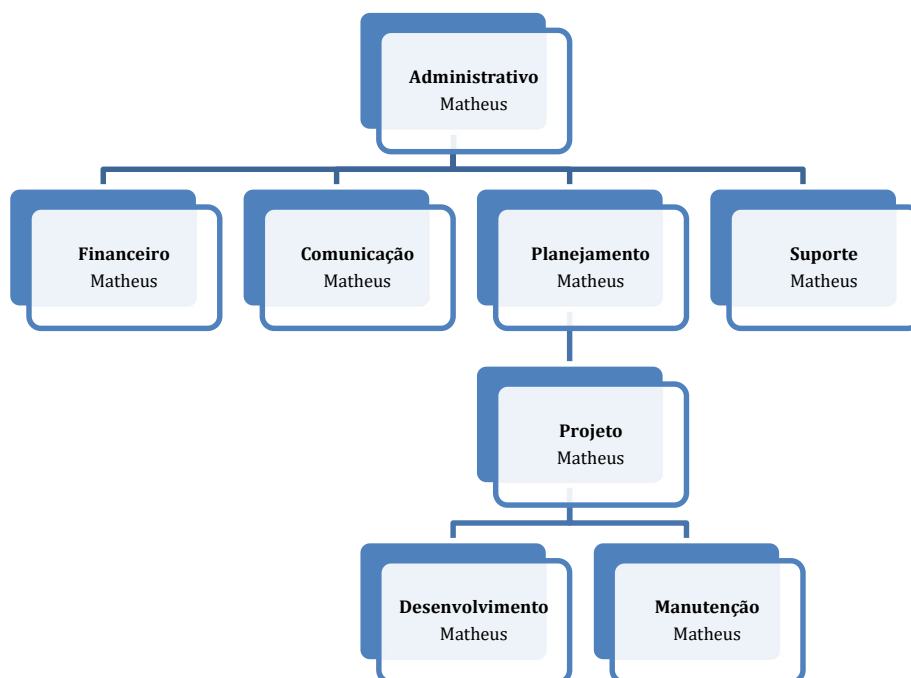
### 1.1.3. RECURSO DO PROJETO

• **Matheus Henrique de Macedo**

✓ **RA:** 918115396

✓ **E-MAIL:** [matheushmacedo16@uni9.edu.br](mailto:matheushmacedo16@uni9.edu.br)

### 1.1.4. ORGANOGRAMA FUNCIONAL



#### 1.1.5. DESCRIÇÃO DAS FUNÇÕES DA EMPRESA

✓ **Matheus – Analista de Sistemas:** Análise de sistemas é a atividade que tem como finalidade a realização de estudos de processos a fim de encontrar o melhor caminho racional para que a informação possa ser processada. Os analistas de sistemas estudam os diversos sistemas existentes entre hardwares, softwares e o usuário final;

✓ **Matheus – Administrador de Banco de Dados:** Administrador de banco de dados, comumente chamado de DBA, é o profissional responsável por gerenciar, instalar, configurar, atualizar e monitorar um banco de dados ou sistemas de bancos de dados;

✓ **Matheus – Programador Sênior:** Sua missão é desenvolver sistemas e aplicações, determinando interface gráfica, critérios ergonômicos de navegação, montagem da estrutura de banco de dados e codificação de programas; projetar, implantar e realizam manutenção de sistemas e aplicações; selecionam recursos de trabalho, tais como metodologias de desenvolvimento de sistemas, linguagem de programação e ferramentas de desenvolvimento. Planejam etapas e ações de trabalho;

✓ **Matheus – Analista de Comunicações:** O Analista de Comunicação é o profissional responsável por atuar junto à equipe de comunicação e marketing,

elaborando análises e planos de ação para a área. Um Analista de Comunicação cuida do sistema de comunicação da empresa é o Analista de Comunicação quem administra os veículos internos de comunicação;

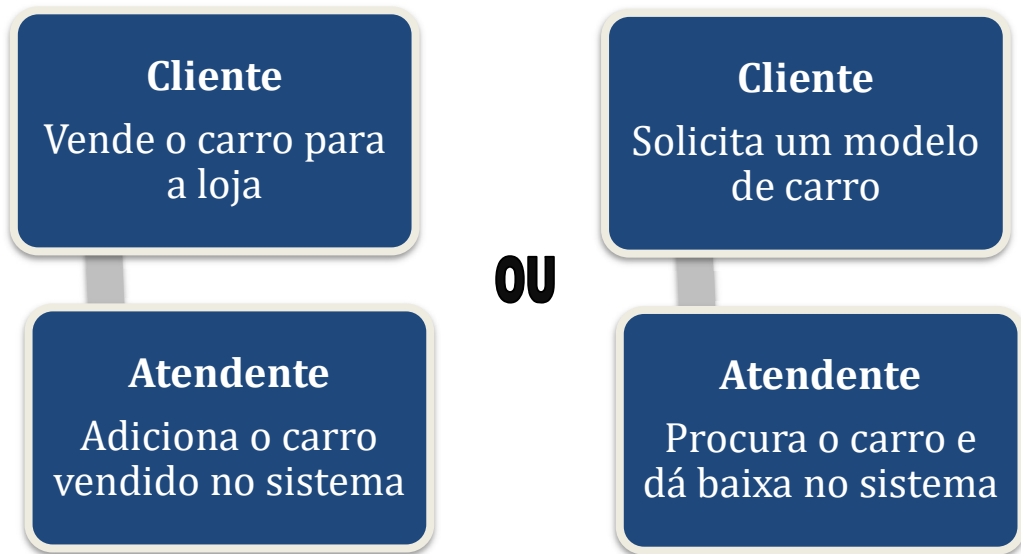
✓ **Matheus – Contador:** Sua missão é administrar os tributos, registrar atos e fatos contábeis; controlar o ativo permanente; gerenciar custos; preparar obrigações acessórias, tais como: declarações acessórias ao fisco, órgãos competentes e contribuintes e administrar o registro dos livros nos órgãos apropriados; elaborar demonstrações contábeis; prestar consultoria e informações gerenciais; realizar auditoria interna e externa; atender solicitações de órgãos fiscalizadores e realizar perícia. Assessorar nas atividades de ensino, pesquisa e extensão;

✓ **Matheus – Gerente de Projetos:** Gerencia projetos, planeja sua execução e acompanha escopo estabelecido e o progresso das rotinas, a fim de cumprir metas, prazos e custos estabelecidos. Identifica os riscos para estudar formas de mitigar impactos e corrigir ações;

✓ **Matheus – Técnico especialista:** Elaborar manuais e procedimentos, prestando atendimento em qualquer dúvida que o usuário possa ter;

✓ **Matheus – Administrador:** Planejar, organizar, controlar e assessorar as organizações nas áreas de recursos humanos, patrimônio, materiais, informações, financeira, tecnológica, entre outras; implementar programas e projetos; elaborar planejamento organizacional; promover estudos de racionalização e controlar o desempenho organizacional.

## 1.2. RESUMO DE NEGÓCIOS DO ESTACIONAMENTO DO ZÉ



### 1.2.1. PROBLEMA A SER RESOLVIDO

A Auto Motors do Zé, solicita um sistema para que o gerenciamento de veículos seja feito com mais facilidade.

### 1.2.2. SOLUÇÃO

Desenvolver um sistema de cadastro que além de auxiliar os atendentes durante o atendimento, possa ajudar no controle dos automóveis adquiridos pela loja.

## 1.3. TERMO DE ABERTURA DO PROJETO

<b>Nome do Projeto:</b> Auto Motors do Zé: Sistema para loja de carros
<b>Responsável pela Autorização do Projeto:</b>  <b>Nome:</b> Sr. José <b>Cargo:</b> Dono da loja
<b>Gerente do Projeto:</b>  <b>Nome:</b> Matheus Henrique de Macedo <b>Cargo:</b> Gerente do Projeto
<b>Justificativa do projeto:</b>  Para que se tenha um controle melhor dos carros disponíveis e facilitar a localização dos mesmos, foi solicitado um software que cadastre os veículos



que a loja possui, indicando alguns detalhes do mesmo.

### **Finalidade do Projeto:**

Desenvolver um software que seja capaz de cadastrar e facilitar a consulta dos carros da loja.

### **Objetivo(s) do Projeto:**

- Cadastrar todos os veículos no banco de dados;
- Listar todos os veículos do banco de dados;
- Alterar o cadastro de um veículo no banco de dados;
- Excluir um veículo cadastrado do banco de dados.

### **Resultados Principais (Descrição do Produto):**

A Auto Motors do Zé, contará com um software de gerenciamento de ativos da loja, facilitando a organização de compra e venda dos mesmos.

### **Stakeholders do Projeto:**

- Cliente do projeto – seu Zé
- Patrocinador – seu Joaquim
- Gerente de projeto – Matheus Macedo
- Equipe do projeto – Matheus Macedo
- Professores – Marcio Romero e Adriano Schroeder

### **Restrições do Projeto (Descrição):**

- Orçamento limite;
- Tempo de projeto de 2 meses;
- Quantidade de pessoal reduzido.

### **Premissas (Descrição):**

- Menor tempo médio de atendimento;
- Facilidade e comodidade durante a compra;
- Melhor gerenciamento dos ativos do cliente.

### **Riscos:**

- Financeiro, ultrapassar orçamento limite;
- Tempo de projeto ultrapassar prazo de entrega;
- Quantidade de pessoal ser insuficiente.

<b>Desafios:</b> <ul style="list-style-type: none"> <li>Trabalhar com datas de compra e venda <b>X</b>.</li> </ul>	
<b>Estimativas de recursos (Descrição):</b> <ul style="list-style-type: none"> <li>Equipe de Desenvolvimento;</li> <li>Infraestrutura para desenvolvimento;</li> <li>Novas ferramentas;</li> <li>Apoio dos colaboradores da pizzeria.</li> </ul>	
<b>Fases, Marcos (Descrição):</b> <ul style="list-style-type: none"> <li>Análise de requisitos;</li> <li>Desenvolvimento;</li> <li>Implementação do sistema;</li> <li>Aceitação do projeto pelos usuários.</li> </ul>	
Data: 26/05/2022	Assinatura autorizador: <b>Sr. José</b>

## 2. DESCRIÇÃO DO SISTEMA WEB

Descrição do Sistema

Segue abaixo as funcionalidades do protótipo do sistema:

1. Ao abrir o sistema, será mostrado todos os veículos disponíveis para a compra;
2. Haverá um botão para cadastrar um novo veículo no sistema;
3. Haverá um botão para alterar um veículo do sistema;
4. Haverá um botão para excluir um veículo do sistema.

## 2.1. TELAS DO SISTEMA

### 2.1.1. PÁGINA INICIAL

**AUTO MOTORS DO ZÉ**

Modelo:  Marca:

Placa:  Versão:  Ano:

Cor:  Portas:

Preço:  Combustível:

Câmbio:  Quilometragem:

Código	Modelo	Marca	Ano	Placa	Cor	Preço	Combustível	Câmbio	Portas	Quilometrag...	Versão
--------	--------	-------	-----	-------	-----	-------	-------------	--------	--------	----------------	--------

Figura 01 — Pagina principal → Auto Motors do Zé  
Fonte: Os Autores.

### 2.1.2. PÁGINA DE CADASTRO/EDIÇÃO

Modelo:

Marca:  Ano:

Placa:  Versão:

Cor:  Portas:

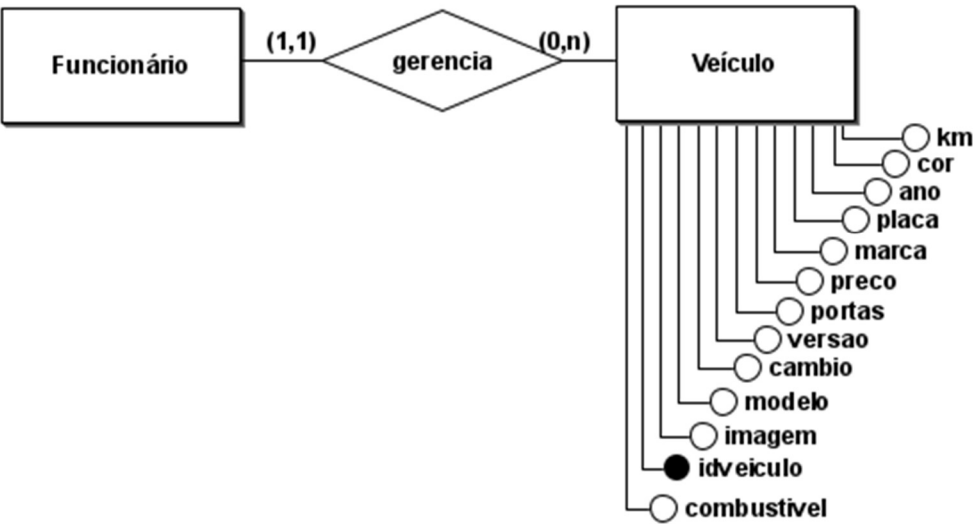
Preço:  Combustível:

Câmbio:  Quilometragem:

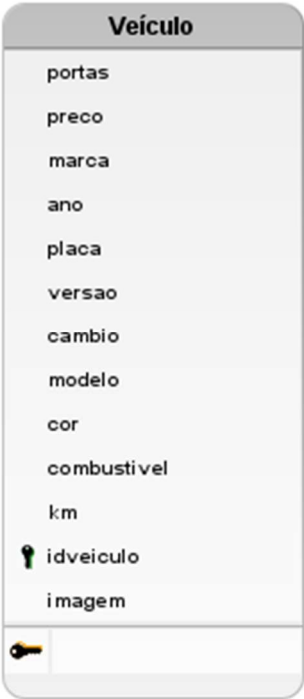
Figura 02 — Pagina de Cadastro/Edição → Auto Motors do Zé  
Fonte: Os Autores.

3. MODELAGEM DO BANCO DE DADOS

3.1. DIAGRAMA CONCEITUAL



3.2. DIAGRAMA LÓGICO



### 3.3. IMPLEMENTAÇÃO FÍSICA

```
-- phpMyAdmin SQL Dump
-- version 5.0.2
-- https://www.phpmyadmin.net/
--
```

```
-- Host: 127.0.0.1
-- Tempo de geração: 26-Maio-2022 às 20:39
-- Versão do servidor: 10.4.14-MariaDB
-- versão do PHP: 7.4.10
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8mb4 */;
-- Banco de dados: `veiculos`
--
-- Estrutura da tabela `veiculo`
```

```
CREATE TABLE `veiculo` (
  `idveiculo` int(11) NOT NULL,
  `modelo` varchar(45) NOT NULL,
  `marca` varchar(45) NOT NULL,
  `ano` int(11) DEFAULT NULL,
  `placa` varchar(8) NOT NULL,
  `imagem` longblob DEFAULT NULL,
  `cor` varchar(20) NOT NULL,
  `preco` decimal(10,2) NOT NULL,
```

```
`combustivel` varchar(20) DEFAULT NULL,  
`cambio` varchar(20) NOT NULL,  
`portas` int(1) NOT NULL,  
`km` int(11) NOT NULL,  
`versao` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Extraindo dados da tabela `veiculo`

```
INSERT INTO `veiculo` (`idveiculo`, `modelo`, `marca`, `ano`, `placa`, `imagem`, `cor`,  
`preco`, `combustivel`, `cambio`, `portas`, `km`, `versao`) VALUES  
(1, 'Uno', 'Fiat', 2002, 'ABC-1598', imgUno.jpg, 'Azul Mar', '5000.00', 'Flex', 'Manual', 4,  
21526, '1.6');  
INSERT INTO `veiculo` (`idveiculo`, `modelo`, `marca`, `ano`, `placa`, `imagem`, `cor`,  
`preco`, `combustivel`, `cambio`, `portas`, `km`, `versao`) VALUES  
(2, 'Toro', 'Fiat', 2022, 'RTG-8975', imgToro.jpg, 'Vermelho', '110000.00', 'Gasolina',  
'Manual', 4, 8000, '2.0');  
INSERT INTO `veiculo` (`idveiculo`, `modelo`, `marca`, `ano`, `placa`, `imagem`, `cor`,  
`preco`, `combustivel`, `cambio`, `portas`, `km`, `versao`) VALUES  
(3, '206', 'Peugeot', 2002, 'ABC-9876', imgPeugeot.jpg, 'Prata', '6500.00', 'Alcool',  
'Manual', 2, 18759, '1.6'),  
(5, 'Camaro', 'Chevrolet', 2006, 'CAM-8975', imgCamaro.jpg, 'Amarelo', '125000.00',  
'Gasolina', 'Automatico', 2, 55489, '2.0');
```

-- Índices para tabelas despejadas

-- Índices para tabela `veiculo`

```
ALTER TABLE `veiculo`  
ADD PRIMARY KEY (`idveiculo`);
```

-- AUTO\_INCREMENT de tabelas despejadas

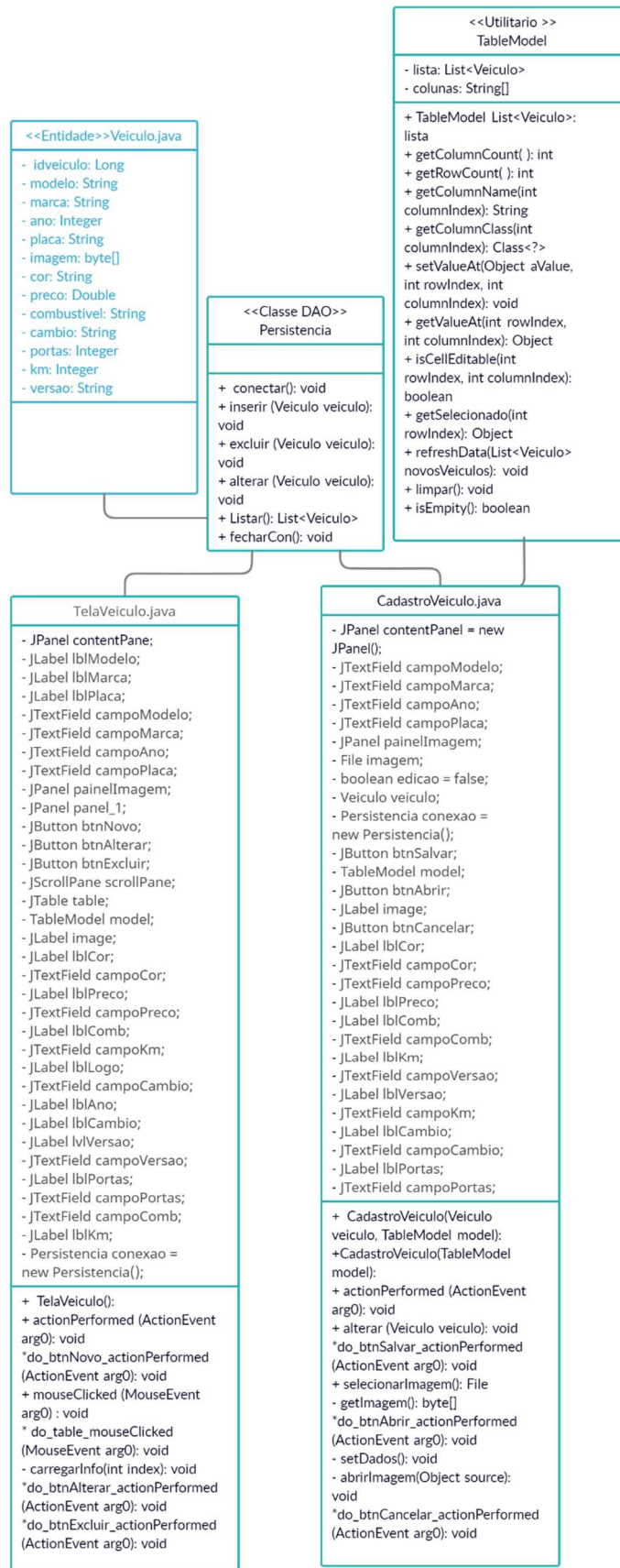
-- AUTO\_INCREMENT de tabela `veiculo`

```
ALTER TABLE `veiculo`  
  MODIFY `idveiculo` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;  
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS  
*/;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

## 4. MODELAGEM UML

### 4.1. DIAGRAMA DE CLASSES





## 5. ARQUITETURA DE SOFTWARE

### 5.1. DESENVOLVIMENTO

#### Veiculo.java (Entidade)

```
1 package entidade;
2
3 public class Veiculo {
4
5     private Long idveiculo;
6     private String modelo;
7     private String marca;
8     private Integer ano;
9     private String placa;
10    private byte[] imagem;
11    private String cor;
12    private Double preco;
13    private String combustivel;
14    private String cambio;
15    private Integer portas;
16    private Integer km;
17    private String versao;
18
19    public Long getIdveiculo() {
20        return idveiculo;
21    }
22    public void setIdveiculo(Long idveiculo) {
23        this.idveiculo = idveiculo;
24    }
25    public String getModelo() {
26        return modelo;
27    }
28    public void setModelo(String modelo) {
29        this.modelo = modelo;
30    }
31    public String getMarca() {
32        return marca;
33    }
34    public void setMarca(String marca) {
35        this.marca = marca;
36    }
37    public Integer getAno() {
38        return ano;
39    }
40    public void setAno(Integer ano) {
41        this.ano = ano;
42    }
43    public String getPlaca() {
44        return placa;
45    }
```

```
46● public void setPlaca(String placa) {
47     this.placa = placa;
48 }
49● public byte[] getImagem() {
50     return imagem;
51 }
52● public void setImagem(byte[] imagem) {
53     this.imagem = imagem;
54 }
55● public String getCor() {
56     return cor;
57 }
58● public void setCor(String cor) {}
59     this.cor = cor;
60 }
61● public Double getPreco() {
62     return preco;
63 }
64● public void setPreco(Double preco) {
65     this.preco = preco;
66 }
67● public String getCombustivel() {
68     return combustivel;
69 }
70● public void setCombustivel(String combustivel) {
71     this.combustivel = combustivel;
72 }
73● public String getCambio() {
74     return cambio;
75 }
76● public void setCambio(String cambio) {
77     this.cambio = cambio;
78 }
79● public int getPortas() {
80     return portas;
81 }
82● public void setPortas(Integer portas) {
83     this.portas = portas;
84 }
85● public int getKm() {
86     return km;
87 }
88● public void setKm(Integer km) {
89     this.km = km;
90 }
```

```

91 public String getVersao() {
92     return versao;
93 }
94 public void setVersao(String versao) {
95     this.versao = versao;
96 }
97
98 }

```

### Persistencia.java (Interação com o Banco de Dados)

```

1 package dao;
2
3 import java.sql.Connection;
12
13 public class Persistencia {
14
15     private Connection con;
16     private PreparedStatement stmt;
17
18     private final String USER = "root";
19     private final String PASS = "";
20     private final String URL = "jdbc:mysql://localhost:3306/veiculos";
21
22
23 public void conectar(){
24     try {
25
26         Class.forName("com.mysql.jdbc.Driver");
27         con = DriverManager.getConnection(URL,USER,PASS);
28
29
30     } catch (ClassNotFoundException | SQLException e) {
31         e.printStackTrace();
32     }
33 }
34
35
36 public void inserir(Veiculo veiculo){
37     String sql = "INSERT INTO veiculo(modelo,"
38         + "marca,"
39         + "ano,"
40         + "placa,"
41         + "imagem,"
42         + "cor,"
43         + "preco,"
44         + "combustivel,"
45         + "cambio, "
46         + "portas, "
47         + "km, "
48         + "versao) "
49         + "VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?)";
50
51     try {
52

```

```

53         stmt = con.prepareStatement(sql);
54         stmt.setString(1, veiculo.getModelo());
55         stmt.setString(2, veiculo.getMarca());
56         stmt.setInt(3, veiculo.getAno());
57         stmt.setString(4, veiculo.getPlaca());
58         stmt.setBytes(5, veiculo.getImagem());
59         stmt.setString(6, veiculo.getCor());
60         stmt.setDouble(7, veiculo.getPreco());
61         stmt.setString(8, veiculo.getCombustivel());
62         stmt.setString(9, veiculo.getCambio());
63         stmt.setInt(10, veiculo.getPortas());
64         stmt.setInt(11, veiculo.getKm());
65         stmt.setString(12, veiculo.getVersao());
66         stmt.execute();
67
68
69     } catch (SQLException e) {
70         e.printStackTrace();
71     }
72 }
73
74 public void excluir(Veiculo veiculo){
75     String sql = "DELETE FROM veiculo WHERE idveiculo = ?";
76
77     try {
78
79         stmt = con.prepareStatement(sql);
80         stmt.setLong(1, veiculo.getIdveiculo());
81         stmt.execute();
82
83
84     } catch (SQLException e) {
85         e.printStackTrace();
86     }
87
88 }
89
90 public void alterar(Veiculo veiculo){
91     String sql = "UPDATE veiculo SET "
92         + "modelo = ?, "
93         + "marca = ?, "
94         + "ano = ?, "
95         + "placa = ?, "
96         + "imagem = ?, "

```



```

98         + "preco = ?, "
99         + "combustivel = ?, "
100        + "cambio = ?, "
101        + "portas = ?, "
102        + "km = ?, "
103        + "versao = ? "
104        + "WHERE idveiculo = ?";
105
106        try {
107
108            stmt = con.prepareStatement(sql);
109            stmt.setString(1, veiculo.getModelo());
110            stmt.setString(2, veiculo.getMarca());
111            stmt.setInt(3, veiculo.getAno());
112            stmt.setString(4, veiculo.getPlaca());
113            stmt.setBytes(5, veiculo.getImage());
114            stmt.setString(6, veiculo.getCor());
115            stmt.setDouble(7, veiculo.getPreco());
116            stmt.setString(8, veiculo.getCombustivel());
117            stmt.setString(9, veiculo.getCambio());
118            stmt.setInt(10, veiculo.getPortas());
119            stmt.setInt(11, veiculo.getKm());
120            stmt.setString(12, veiculo.getVersao());
121            stmt.setLong(13, veiculo.getIdveiculo());
122
123
124            stmt.execute();
125
126        } catch (SQLException e) {
127            e.printStackTrace();
128        }
129    }
130
131    public List<Veiculo> listar(){
132        String sql = "SELECT * FROM veiculo";
133        List<Veiculo> lista = new ArrayList<Veiculo>();
134
135        try {
136
137            stmt = con.prepareStatement(sql);
138            ResultSet res = stmt.executeQuery();
139
140            while(res.next()){
141                Veiculo veiculo = new Veiculo();
142                veiculo.setIdveiculo(res.getLong("idveiculo"));

```

```

143         veiculo.setModelo(res.getString("modelo"));
144         veiculo.setMarca(res.getString("marca"));
145         veiculo.setAno(res.getInt("ano"));
146         veiculo.setPlaca(res.getString("placa"));
147         veiculo.setImagem(res.getBytes("imagem"));
148         veiculo.setCor(res.getString("cor"));
149         veiculo.setPreco(res.getDouble("preco"));
150         veiculo.setCombustivel(res.getString("combustivel"));
151         veiculo.setCambio(res.getString("cambio"));
152         veiculo.setPortas(res.getInt("portas"));
153         veiculo.setKm(res.getInt("km"));
154         veiculo.setVersao(res.getString("versao"));
155
156         lista.add(veiculo);
157     }
158
159     } catch (SQLException e) {
160         e.printStackTrace();
161     }
162
163     return lista;
164 }
165
166
167 public void fecharCon(){
168     try {
169         con.close();
170     } catch (SQLException e) {
171         e.printStackTrace();
172     }
173 }
174
175
176 }

```

**CadastroPrincipal.java (Abre um 'modal' na TelaVeiculo.java relativo a CadastroVeiculo.java)**

```

1 package sistema;
2
3 import gui.TelaVeiculo;
4
5 public class CadastroPrincipal {
6
7     public static void main(String[] args) {
8         TelaVeiculo gui = new TelaVeiculo();
9         gui.setVisible(true);
10        gui.setLocationRelativeTo(null);
11    }
12
13 }
14

```

## CadastroVeiculo.java

```
1 package gui;
2
3 import java.awt.AlphaComposite;
38
39 public class CadastroVeiculo extends JDialog implements ActionListener {
40
41     private final JPanel contentPanel = new JPanel();
42     private JTextField campoModelo;
43     private JTextField campoMarca;
44     private JTextField campoAno;
45     private JTextField campoPlaca;
46     private JPanel painelImagem;
47     private File imagem;
48     private boolean edicao = false;
49     private Veiculo veiculo;
50     private Persistencia conexao = new Persistencia();
51     private JButton btnSalvar;
52     private TableModel model;
53     private JButton btnAbrir;
54     private JLabel image;
55     private JButton btnCancelar;
56     private JLabel lblCor;
57     private JTextField campoCor;
58     private JTextField campoPreco;
59     private JLabel lblPreco;
60     private JLabel lblComb;
61     private JTextField campoComb;
62     private JLabel lblKm;
63     private JTextField campoVersao;
64     private JLabel lblVersao;
65     private JTextField campoKm;
66     private JLabel lblCambio;
67     private JTextField campoCambio;
68     private JLabel lblPortas;
69     private JTextField campoPortas;
70
71     /**
72      * Launch the application.
73      */
74     public static void main(String[] args, TableModel model) {
75         try {
76             CadastroVeiculo dialog = new CadastroVeiculo(model);
77             dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
```



```

78         dialog.setVisible(true);
79     } catch (Exception e) {
80         e.printStackTrace();
81     }
82 }
83
84 public CadastroVeiculo(Veiculo veiculo, TableModel model) {
85     this(model);
86     this.model = model;
87     this.edicao = true;
88     this.veiculo = veiculo;
89     setDados();
90 }
91
92 /**
93  * Create the dialog.
94  * @wbp.parser.constructor
95  */
96 public CadastroVeiculo(TableModel model) {
97     this.model = model;
98     setBounds(100, 100, 806, 395);
99     getContentPane().setLayout(new BorderLayout());
100    contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
101    getContentPane().add(contentPanel, BorderLayout.NORTH);
102    JLabel lblModelo = new JLabel("Modelo:");
103    lblModelo.setFont(new Font("SansSerif", Font.BOLD, 12));
104    JLabel lblMarca = new JLabel("Marca:");
105    lblMarca.setFont(new Font("SansSerif", Font.BOLD, 12));
106    JLabel lblAno = new JLabel("Ano:");
107    lblAno.setFont(new Font("SansSerif", Font.BOLD, 12));
108    JLabel lblPlaca = new JLabel("Placa:");
109    lblPlaca.setFont(new Font("SansSerif", Font.BOLD, 12));
110    campoModelo = new JTextField();
111    campoModelo.setColumns(10);
112    campoMarca = new JTextField();
113    campoMarca.setColumns(10);
114    campoAno = new JTextField();
115    campoAno.setColumns(10);
116    campoPlaca = new JTextField();
117    campoPlaca.setColumns(10);
118    JPanel panel = new JPanel();
119    FlowLayout flowLayout = (FlowLayout) panel.getLayout();
120    flowLayout.setVgap(10);

```



```
121     panel.setBorder(new LineBorder(new Color(0, 0, 0)));
122     painelImagem = new JPanel();
123     painelImagem.setBorder(new LineBorder(new Color(0, 0, 0)));
124     btnAbrir = new JButton("Adicionar Imagem");
125     btnAbrir.addActionListener(this);
126
127     lblCor = new JLabel("Cor:");
128     lblCor.setFont(new Font("SansSerif", Font.BOLD, 12));
129
130     campoCor = new JTextField();
131     campoCor.setColumns(10);
132
133     campoPreco = new JTextField();
134     campoPreco.setColumns(10);
135
136     lblPreco = new JLabel("Pre\u00E7o:");
137     lblPreco.setFont(new Font("SansSerif", Font.BOLD, 12));
138
139     lblComb = new JLabel("Combustivel:");
140     lblComb.setFont(new Font("SansSerif", Font.BOLD, 12));
141
142     campoComb = new JTextField();
143     campoComb.setColumns(10);
144
145     lblKm = new JLabel("Quilometragem:");
146     lblKm.setFont(new Font("SansSerif", Font.BOLD, 12));
147
148     campoVersao = new JTextField();
149     campoVersao.setColumns(10);
150
151     lblVersao = new JLabel("Vers\u00E3o:");
152     lblVersao.setFont(new Font("SansSerif", Font.BOLD, 12));
153
154     campoKm = new JTextField();
155     campoKm.setColumns(10);
156
157     lblCambio = new JLabel("C\u00E2mbio:");
158     lblCambio.setFont(new Font("SansSerif", Font.BOLD, 12));
159
160     campoCambio = new JTextField();
161     campoCambio.setColumns(10);
162
163     lblPortas = new JLabel("Portas:");
```

```

164         lblPortas.setFont(new Font("SansSerif", Font.BOLD, 12));
165
166         campoPortas = new JTextField();
167         campoPortas.setColumns(10);
168         GroupLayout gl_contentPanel = new GroupLayout(contentPanel);
169         gl_contentPanel.setHorizontalGroup(
170             gl_contentPanel.createParallelGroup(Alignment.TRAILING)
171                 .addGroup(gl_contentPanel.createSequentialGroup()
172                     .addGroup(gl_contentPanel.createParallelGroup(Alignment.TRAILING)
173                         .addGroup(gl_contentPanel.createSequentialGroup()
174                             .addContainerGap()
175                             .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING)
176                                 .addComponent(lblCambio, Alignment.TRAILING, GroupLayout.PREFERRED_SIZE, 46, GroupLayout.PREFERRED_SIZE)
177                                 .addComponent(lblPreco, Alignment.TRAILING)
178                                 .addComponent(lblCor, Alignment.TRAILING)
179                                 .addComponent(lblPlaca, Alignment.TRAILING)
180                                 .addComponent(lblMarca, Alignment.TRAILING)
181                                 .addComponent(lblModelo, Alignment.TRAILING))
182                             .addGap(27)
183                             .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING)
184                                 .addGroup(gl_contentPanel.createSequentialGroup()
185                                     .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING, false)
186                                         .addComponent(campoPreco)
187                                         .addComponent(campoMarca)
188                                         .addComponent(campoCor)
189                                         .addComponent(campoCambio)
190                                         .addComponent(campoPlaca))
191                                     .addGap(6)
192                                     .addGroup(gl_contentPanel.createParallelGroup(Alignment.TRAILING)
193                                         .addComponent(lblKm)
194                                         .addComponent(lblComb)
195                                         .addComponent(lblVersao)
196                                         .addComponent(lblAno)
197                                         .addComponent(lblPortas))
198                                     .addGap(18)
199                                     .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING, false)
200                                         .addComponent(campoKm)
201                                         .addComponent(campoComb)
202                                         .addComponent(campoPortas)
203                                         .addComponent(campoVersao)
204                                         .addComponent(campoAno)))
205                                     .addComponent(campoModelo, GroupLayout.PREFERRED_SIZE, 162, GroupLayout.PREFERRED_SIZE))
206                                 .addPreferredGap(ComponentPlacement.RELATED, 26, Short.MAX_VALUE)
207                                     .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING)
208                                         .addComponent(btnAbrir)
209                                         .addComponent(painelImagem, GroupLayout.PREFERRED_SIZE, 333, GroupLayout.PREFERRED_SIZE)))
210                                 .addGroup(gl_contentPanel.createSequentialGroup()
211                                     .addGap(1)
212                                     .addComponent(panel, GroupLayout.DEFAULT_SIZE, 775, Short.MAX_VALUE))
213                                 .addContainerGap())
214         );
215         gl_contentPanel.setVerticalGroup(
216             gl_contentPanel.createParallelGroup(Alignment.LEADING)
217                 .addGroup(gl_contentPanel.createSequentialGroup()
218                     .addContainerGap()
219                     .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING)
220                         .addGroup(gl_contentPanel.createSequentialGroup()
221                             .addGroup(gl_contentPanel.createParallelGroup(Alignment.BASELINE)
222                                 .addComponent(lblModelo)
223                                 .addComponent(campoModelo, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
224                             .addGap(18)
225                             .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING)
226                                 .addGroup(gl_contentPanel.createParallelGroup(Alignment.BASELINE)
227                                     .addComponent(campoMarca, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
228                                     .addComponent(lblMarca))
229                                 .addGroup(gl_contentPanel.createParallelGroup(Alignment.BASELINE)
230                                     .addComponent(campoAno, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
231                                     .addComponent(lblAno)))
232                             .addGap(18)
233                             .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING)
234                                 .addGroup(gl_contentPanel.createParallelGroup(Alignment.BASELINE)
235                                     .addComponent(campoPlaca, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
236                                     .addComponent(lblPlaca))
237                                 .addGroup(gl_contentPanel.createParallelGroup(Alignment.BASELINE)
238                                     .addComponent(campoVersao, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
239                                     .addComponent(lblVersao)))
240                             .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING)
241                                 .addGroup(gl_contentPanel.createSequentialGroup()
242                                     .addPreferredGap(ComponentPlacement.UNRELATED)
243                                     .addGroup(gl_contentPanel.createParallelGroup(Alignment.BASELINE)
244                                         .addComponent(campoCor, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
245                                         .addComponent(lblCor))
246                                     .addPreferredGap(ComponentPlacement.RELATED))
247                                 .addGroup(gl_contentPanel.createSequentialGroup()
248                                     .addPreferredGap(ComponentPlacement.UNRELATED)
249                                     .addGroup(gl_contentPanel.createParallelGroup(Alignment.BASELINE)

```

```

250         .addComponent(campoPortas, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
251         .addComponent(lblPortas))
252         .addPreferredGap(ComponentPlacement.RELATED)))
253     .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING)
254     .addGroup(gl_contentPanel.createSequentialGroup()
255     .addPreferredGap(ComponentPlacement.RELATED)
256     .addGroup(gl_contentPanel.createParallelGroup(Alignment.BASELINE)
257     .addComponent(campoComb, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
258     .addComponent(lblComb)))
259     .addGroup(gl_contentPanel.createSequentialGroup()
260     .addGap(6)
261     .addGroup(gl_contentPanel.createParallelGroup(Alignment.BASELINE)
262     .addComponent(campoPreco, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
263     .addComponent(lblPreco))))
264     .addGap(12)
265     .addGroup(gl_contentPanel.createParallelGroup(Alignment.LEADING)
266     .addComponent(campoKm, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
267     .addGroup(gl_contentPanel.createParallelGroup(Alignment.BASELINE)
268     .addComponent(campoCambio, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
269     .addComponent(lblCambio)
270     .addComponent(lblKm))))
271     .addGroup(gl_contentPanel.createSequentialGroup()
272     .addComponent(painelImagem, GroupLayout.PREFERRED_SIZE, 169, GroupLayout.PREFERRED_SIZE)
273     .addPreferredGap(ComponentPlacement.RELATED)
274     .addComponent(btnAbrir))
275     .addGap(36)
276     .addComponent(panel, GroupLayout.PREFERRED_SIZE, 56, GroupLayout.PREFERRED_SIZE)
277     .addGap(95))
278 );
279 gl_contentPanel.linkSize(SwingConstants.VERTICAL, new Component[] {campoModelo, campoMarca});
280 {
281     image = new JLabel("");
282     painelImagem.add(image);
283 }
284 {
285     btnSalvar = new JButton("Salvar");
286     btnSalvar.addActionListener(this);
287     panel.add(btnSalvar);
288 }
289 {
290     btnCancelar = new JButton("Cancelar");
291     btnCancelar.addActionListener(this);
292     panel.add(btnCancelar);

```



```

293     }
294     contentPanel.setLayout(gl_contentPanel);
295 }
296
297 public void actionPerformed(ActionEvent arg0) {
298     if (arg0.getSource() == btnCancelar) {
299         do_btnCancelar_actionPerformed(arg0);
300     }
301     if (arg0.getSource() == btnAbrir) {
302         do_btnAbrir_actionPerformed(arg0);
303     }
304     if (arg0.getSource() == btnSalvar) {
305         do_btnSalvar_actionPerformed(arg0);
306     }
307 }
308 protected void do_btnSalvar_actionPerformed(ActionEvent arg0) {
309     conexao.conectar();
310     if(edicao){
311         veiculo.setModelo(campoModelo.getText());
312         veiculo.setMarca(campoMarca.getText());
313         veiculo.setAno(Integer.parseInt(campoAno.getText()));
314         veiculo.setPlaca(campoPlaca.getText());
315         veiculo.setCor(campoCor.getText());
316         veiculo.setPreco(Double.parseDouble(campoPreco.getText()));
317         veiculo.setCombustivel(campoComb.getText());
318         veiculo.setCambio(campoCambio.getText());
319         veiculo.setPortas(Integer.parseInt(campoPortas.getText()));
320         veiculo.setKm(Integer.parseInt(campoKm.getText()));
321         veiculo.setVersao(campoVersao.getText());
322
323         if(imagem != null){
324             veiculo.setImagem(getImagem());
325         }
326         conexao.alterar(veiculo);
327         model.refreshData(conexao.listar());
328         JOptionPane.showMessageDialog(this, "Registro alterado com sucesso!");
329         conexao.fecharCon();
330         dispose();
331
332     }else{
333         Veiculo veiculo = new Veiculo();
334         veiculo.setModelo(campoModelo.getText());
335         veiculo.setMarca(campoMarca.getText());

```

```

336         veiculo.setAno(Integer.parseInt(campoAno.getText()));
337         veiculo.setPlaca(campoPlaca.getText());
338         veiculo.setCor(campoCor.getText());
339         veiculo.setPreco(Double.parseDouble(campoPreco.getText()));
340         veiculo.setCombustivel(campoComb.getText());
341         veiculo.setCambio(campoCambio.getText());
342         veiculo.setPortas(Integer.parseInt(campoPortas.getText()));
343         veiculo.setKm(Integer.parseInt(campoKm.getText()));
344         veiculo.setVersao(campoVersao.getText());
345
346         veiculo.setImagem(getImagem());
347         conexao.inserir(veiculo);
348         model.refreshData(conexao.listar());
349         JOptionPane.showMessageDialog(this, "O registro foi salvo com sucesso!");
350         conexao.fecharCon();
351         dispose();
352     }
353 }
354
355 public File selecionarImagem(){
356     JFileChooser fileChooser = new JFileChooser();
357     FileNameExtensionFilter filtro = new FileNameExtensionFilter("Imagens em JPEG e PNG", "jpg", "png");
358     fileChooser.addChoosableFileFilter(filtro);
359     fileChooser.setAcceptAllFileFilterUsed(false);
360     fileChooser.setDialogType(JFileChooser.OPEN_DIALOG);
361     fileChooser.setCurrentDirectory(new File("/"));
362     fileChooser.showOpenDialog(this);
363
364     return fileChooser.getSelectedFile();
365 }
366
367 private byte[] getImagem(){
368     boolean isPng = false;
369
370     if(imagem != null){
371         isPng = imagem.getName().endsWith("png");
372
373         try {
374
375             BufferedImage image = ImageIO.read(imagem);
376             ByteArrayOutputStream out = new ByteArrayOutputStream();
377             int type = BufferedImage.TYPE_INT_RGB;
378

```

```

379         if(isPng){
380             type = BufferedImage.BITMASK;
381         }
382
383         BufferedImage novaImagem = new BufferedImage(painelImagem.getWidth() - 5, painelImagem.getHeight() - 10, type);
384         Graphics2D g = novaImagem.createGraphics();
385         g.setComposite(AlphaComposite.Src);
386         g.drawImage(image, 0,0, painelImagem.getWidth() -5, painelImagem.getHeight() -10, null);
387
388         if(isPng){
389             ImageIO.write(novaImagem, "png", out);
390         }else{
391             ImageIO.write(novaImagem, "jpg", out);
392         }
393
394         out.flush();
395         byte[] byteArray = out.toByteArray();
396         out.close();
397
398         return byteArray;
399     }
400     } catch (IOException e) {
401         e.printStackTrace();
402     }
403 }
404
405     return null;
406 }
407 ● protected void do_btnAbrir_actionPerformed(ActionEvent arg0) {
408     imagem = selecionarImagem();
409     abrirImagem(imagem);
410 }
411
412 ● private void setDados(){
413     campoModelo.setText(veiculo.getModelo());
414     campoMarca.setText(veiculo.getMarca());
415     campoAno.setText(String.valueOf(veiculo.getAno()));
416     campoPlaca.setText(veiculo.getPlaca());
417     campoCor.setText(veiculo.getCor());
418     campoPreco.setText(String.valueOf(veiculo.getPreco()));
419     campoComb.setText(veiculo.getCombustivel());
420     campoCambio.setText(veiculo.getCambio());
421     campoPortas.setText(String.valueOf(veiculo.getPortas()));
422     campoKm.setText(String.valueOf(veiculo.getKm()));
423     campoVersao.setText(veiculo.getVersao());
424     abrirImagem(veiculo.getImagem());
425 }
426
427 ● private void abrirImagem(Object source){
428     if(source instanceof File){
429         ImageIcon icon = new ImageIcon(imagem.getAbsolutePath());
430         icon.setImage(icon.getImage().getScaledInstance(painelImagem.getWidth()-5, painelImagem.getHeight() -10, 100));
431         image.setIcon(icon);
432     }else if(source instanceof byte[]){
433         ImageIcon icon = new ImageIcon(veiculo.getImagem());
434         icon.setImage(icon.getImage().getScaledInstance(painelImagem.getWidth()-5, painelImagem.getHeight() -10, 100));
435         image.setIcon(icon);
436     }
437 }
438 ● protected void do_btnCancelar_actionPerformed(ActionEvent arg0) {
439     dispose();
440 }
441 }

```

## TelaVeiculo.java



```

1 package gui;
2
3 import java.awt.BorderLayout;
35
36 public class TelaVeiculo extends JFrame implements ActionListener, MouseListener {
37
38     private JPanel contentPane;
39     private JLabel lblModelo;
40     private JLabel lblMarca;
41     private JLabel lblPlaca;
42     private JTextField campoModelo;
43     private JTextField campoMarca;
44     private JTextField campoAno;
45     private JTextField campoPlaca;
46     private JPanel painelImagem;
47     private JPanel panel_1;
48     private JButton btnNovo;
49     private JButton btnAlterar;
50     private JButton btnExcluir;
51     private JScrollPane scrollPane;
52     private JTable table;
53     private TableModel model;
54     Persistencia conexao = new Persistencia();
55     private JLabel image;
56     private JLabel lblCor;
57     private JTextField campoCor;
58     private JLabel lblPreco;
59     private JTextField campoPreco;
60     private JLabel lblComb;
61     private JTextField campoKm;
62     private JLabel lblLogo;
63     private JTextField campoCambio;
64     private JLabel lblAno;
65     private JLabel lblCambio;
66     private JLabel lblVersao;
67     private JTextField campoVersao;
68     private JLabel lblPortas;
69     private JTextField campoPortas;
70     private JTextField campoComb;
71     private JLabel lblKm;
72 /**
73      * Launch the application.
74      */

```

```

75● public static void main(String[] args) {
76●     EventQueue.invokeLater(new Runnable() {
77●         public void run() {
78             try {
79                 TelaVeiculo frame = new TelaVeiculo();
80                 frame.setVisible(true);
81             } catch (Exception e) {
82                 e.printStackTrace();
83             }
84         }
85     });
86 }
87
88● /**
89  * Create the frame.
90  */
91● public TelaVeiculo() {
92     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
93     setBounds(100, 100, 1086, 779);
94     contentPane = new JPanel();
95     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
96     setContentPane(contentPane);
97
98     lblModelo = new JLabel("Modelo:");
99     lblModelo.setFont(new Font("SansSerif", Font.BOLD, 12));
100
101     lblMarca = new JLabel("Marca:");
102     lblMarca.setFont(new Font("SansSerif", Font.BOLD, 12));
103
104     lblPlaca = new JLabel("Placa:");
105     lblPlaca.setFont(new Font("SansSerif", Font.BOLD, 12));
106
107     campoModelo = new JTextField();
108     campoModelo.setEditable(false);
109     campoModelo.setColumns(10);
110
111     campoMarca = new JTextField();
112     campoMarca.setEditable(false);
113     campoMarca.setColumns(10);
114
115     campoAno = new JTextField();
116     campoAno.setEditable(false);
117     campoAno.setColumns(10);

```



```
118
119     campoPlaca = new JTextField();
120     campoPlaca.setEditable(false);
121     campoPlaca.setColumns(10);
122
123     painelImagem = new JPanel();
124     painelImagem.setBorder(new LineBorder(new Color(0, 0, 0)));
125
126     panel_1 = new JPanel();
127     FlowLayout flowLayout = (FlowLayout) panel_1.getLayout();
128     flowLayout.setVgap(10);
129     panel_1.setBorder(new LineBorder(new Color(0, 0, 0)));
130
131     scrollPane = new JScrollPane();
132     scrollPane.addMouseListener(this);
133
134     lblCor = new JLabel("Cor:");
135     lblCor.setFont(new Font("SansSerif", Font.BOLD, 12));
136
137     campoCor = new JTextField();
138     campoCor.setEditable(false);
139     campoCor.setColumns(10);
140
141     lblPreco = new JLabel("Pre\u00E7o:");
142     lblPreco.setFont(new Font("SansSerif", Font.BOLD, 12));
143
144     campoPreco = new JTextField();
145     campoPreco.setEditable(false);
146     campoPreco.setColumns(10);
147
148     lblComb = new JLabel("Combust\u00EDvel:");
149     lblComb.setFont(new Font("SansSerif", Font.BOLD, 12));
150
151     campoKm = new JTextField();
152     campoKm.setEditable(false);
153     campoKm.setColumns(10);
154
155     lblLogo = new JLabel("AUTO MOTORS DO Z\u00C9");
156     lblLogo.setFont(new Font("Tahoma", Font.BOLD, 32));
157
158     campoCambio = new JTextField();
159     campoCambio.setEditable(false);
160     campoCambio.setColumns(10);
```

```

161
162     lblAno = new JLabel("Ano:");
163     lblAno.setFont(new Font("SansSerif", Font.BOLD, 12));
164
165     lblCambio = new JLabel("C\u00E2mbio:");
166     lblCambio.setFont(new Font("SansSerif", Font.BOLD, 12));
167
168     lvlVersao = new JLabel("Vers\u00E3o:");
169     lvlVersao.setFont(new Font("SansSerif", Font.BOLD, 12));
170
171     campoVersao = new JTextField();
172     campoVersao.setEditable(false);
173     campoVersao.setColumns(10);
174
175     lblPortas = new JLabel("Portas:");
176     lblPortas.setFont(new Font("SansSerif", Font.BOLD, 12));
177
178     campoPortas = new JTextField();
179     campoPortas.setEditable(false);
180     campoPortas.setColumns(10);
181
182     campoComb = new JTextField();
183     campoComb.setEditable(false);
184     campoComb.setColumns(10);
185
186     lblKm = new JLabel("Quilometragem:");
187     lblKm.setFont(new Font("SansSerif", Font.BOLD, 12));
188     GroupLayout gl_contentPane = new GroupLayout(contentPane);
189     gl_contentPane.setHorizontalGroup(
190         gl_contentPane.createParallelGroup(Alignment.TRAILING)
191             .addGroup(gl_contentPane.createSequentialGroup()
192                 .addContainerGap()
193                 .addComponent(panel_1, GroupLayout.DEFAULT_SIZE, 1050, Short.MAX_VALUE)
194                 .addContainerGap())
195             .addGroup(gl_contentPane.createSequentialGroup()
196                 .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 1056, Short.MAX_VALUE)
197                 .addContainerGap())
198             .addGroup(gl_contentPane.createSequentialGroup()
199                 .addContainerGap(59, Short.MAX_VALUE)
200                 .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
201                     .addComponent(lblModelo, Alignment.TRAILING)
202                     .addComponent(lblPlaca, Alignment.TRAILING)

```



```

203         .addComponent(lblCor, Alignment.TRAILING)
204         .addComponent(lblPreco, Alignment.TRAILING)
205         .addComponent(lblCambio, Alignment.TRAILING))
206     .addGap(23)
207     .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING, false)
208     .addGroup(gl_contentPane.createSequentialGroup()
209     .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING, false)
210     .addComponent(campoCor, Alignment.LEADING, GroupLayout.DEFAULT_SIZE, 197, Short.MAX_VALUE)
211     .addComponent(campoPreco, GroupLayout.DEFAULT_SIZE, 197, Short.MAX_VALUE)
212     .addComponent(campoCambio, Alignment.LEADING))
213     .addGap(12)
214     .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
215     .addComponent(lblKm, Alignment.TRAILING)
216     .addComponent(lblComb, Alignment.TRAILING)
217     .addComponent(lblPortas, Alignment.TRAILING, GroupLayout.PREFERRED_SIZE, 46, GroupLayout.PREFERRED_SIZE))
218     .addPreferredGap(ComponentPlacement.RELATED)
219     .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
220     .addComponent(campoComb, GroupLayout.DEFAULT_SIZE, 197, Short.MAX_VALUE)
221     .addComponent(campoPortas, GroupLayout.DEFAULT_SIZE, 197, Short.MAX_VALUE)
222     .addComponent(campoKm)))
223     .addGroup(gl_contentPane.createSequentialGroup()
224     .addComponent(campoPlaca, 0, 0, Short.MAX_VALUE)
225     .addPreferredGap(ComponentPlacement.UNRELATED)
226     .addComponent(lvlVersao, GroupLayout.PREFERRED_SIZE, 46, GroupLayout.PREFERRED_SIZE)
227     .addPreferredGap(ComponentPlacement.RELATED)
228     .addComponent(campoVersao, GroupLayout.PREFERRED_SIZE, 182, GroupLayout.PREFERRED_SIZE)
229     .addGap(18)
230     .addComponent(lblAno, GroupLayout.PREFERRED_SIZE, 25, GroupLayout.PREFERRED_SIZE)
231     .addPreferredGap(ComponentPlacement.UNRELATED)
232     .addComponent(campoAno, GroupLayout.PREFERRED_SIZE, 95, GroupLayout.PREFERRED_SIZE))
233     .addGroup(gl_contentPane.createSequentialGroup()
234     .addComponent(campoModelo)
235     .addPreferredGap(ComponentPlacement.RELATED)
236     .addComponent(lblMarca)
237     .addPreferredGap(ComponentPlacement.RELATED)
238     .addComponent(campoMarca, GroupLayout.PREFERRED_SIZE, 195, GroupLayout.PREFERRED_SIZE)))
239     .addGap(53)
240     .addComponent(painelImagem, GroupLayout.PREFERRED_SIZE, 318, GroupLayout.PREFERRED_SIZE)
241     .addGap(61))
242     .addGroup(gl_contentPane.createSequentialGroup()
243     .addContainerGap(289, Short.MAX_VALUE)
244     .addComponent(lblLogo, GroupLayout.PREFERRED_SIZE, 401, GroupLayout.PREFERRED_SIZE)
245     .addGap(372))
246 );
247 gl_contentPane.setVerticalGroup(
248     gl_contentPane.createParallelGroup(Alignment.LEADING)
249     .addGroup(gl_contentPane.createSequentialGroup()
250     .addComponent(lblLogo, GroupLayout.PREFERRED_SIZE, 51, GroupLayout.PREFERRED_SIZE)
251     .addGap(33)
252     .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)
253     .addGroup(gl_contentPane.createSequentialGroup()
254     .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
255     .addComponent(campoMarca, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
256     .addComponent(lblMarca)
257     .addComponent(campoModelo, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
258     .addComponent(lblModelo))
259     .addGap(18)
260     .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
261     .addComponent(lblPlaca)
262     .addComponent(lvlVersao)
263     .addComponent(campoVersao, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
264     .addComponent(lblAno)
265     .addComponent(campoAno, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
266     .addComponent(campoPlaca, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
267     .addPreferredGap(ComponentPlacement.UNRELATED)
268     .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
269     .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
270     .addComponent(campoCor, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
271     .addComponent(lblCor))
272     .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
273     .addComponent(campoPortas, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
274     .addComponent(lblPortas)))
275     .addGap(6)
276     .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
277     .addComponent(lblPreco)
278     .addComponent(campoPreco, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
279     .addComponent(lblComb)
280     .addComponent(campoComb, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
281     .addPreferredGap(ComponentPlacement.UNRELATED)
282     .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
283     .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
284     .addComponent(lblKm)
285     .addComponent(campoKm, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
286     .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)

```

```
287         .addComponent(lblCambio)
288         .addComponent(campoCambio, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)))
289     .addComponent(painelImagem, GroupLayout.PREFERRED_SIZE, 195, GroupLayout.PREFERRED_SIZE))
290     .addGap(53)
291     .addComponent(panel_1, GroupLayout.PREFERRED_SIZE, 44, GroupLayout.PREFERRED_SIZE)
292     .addGap(18)
293     .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 332, Short.MAX_VALUE)
294     .addContainerGap()
295 );
296
```

```

297         image = new JLabel("");
298         painelImagem.add(image);
299
300
301         conexao.conectar();
302         model = new TableModel(conexao.listar());
303         table = new JTable(model);
304         table.addMouseListener(this);
305         scrollPane.setViewportViewView(table);
306
307         btnNovo = new JButton("Novo");
308         btnNovo.addActionListener(this);
309         panel_1.add(btnNovo);
310
311         btnAlterar = new JButton("Alterar");
312         btnAlterar.addActionListener(this);
313         panel_1.add(btnAlterar);
314
315         btnExcluir = new JButton("Excluir");
316         btnExcluir.addActionListener(this);
317         panel_1.add(btnExcluir);
318         contentPane.setLayout(gl_contentPane);
319         carregarInfo(0);
320     }
321     public void actionPerformed(ActionEvent arg0) {
322         if (arg0.getSource() == btnExcluir) {
323             do_btnExcluir_actionPerformed(arg0);
324         }
325         if (arg0.getSource() == btnAlterar) {
326             do_btnAlterar_actionPerformed(arg0);
327         }
328         if (arg0.getSource() == btnNovo) {
329             do_btnNovo_actionPerformed(arg0);
330         }
331     }
332     protected void do_btnNovo_actionPerformed(ActionEvent arg0) {
333         CadastroVeiculo cadastro = new CadastroVeiculo(model);
334         cadastro.setVisible(true);
335         cadastro.setLocationRelativeTo(this);
336     }
337     public void mouseClicked(MouseEvent arg0) {
338         if (arg0.getSource() == table) {
339             do_table_mouseClicked(arg0);

```



```

340     }
341     if (arg0.getSource() == scrollPane) {
342         do_scrollPane_mouseClicked(arg0);
343     }
344 }
345 public void mouseEntered(MouseEvent arg0) {
346 }
347 public void mouseExited(MouseEvent arg0) {
348 }
349 public void mousePressed(MouseEvent arg0) {
350 }
351 public void mouseReleased(MouseEvent arg0) {
352 }
353 protected void do_scrollPane_mouseClicked(MouseEvent arg0) {
354 }
355 protected void do_table_mouseClicked(MouseEvent arg0) {
356     if(arg0.getClickCount() == 1){
357         carregarInfo(table.getSelectedRow());
358     }
359 }
360
361 private void carregarInfo(int index){
362     Veiculo carro = (Veiculo) model.getSelectedado(index);
363     if(carro != null){
364         campoModelo.setText(carro.getModelo());
365         campoMarca.setText(carro.getMarca());
366         campoAno.setText(String.valueOf(carro.getAno()));
367         campoPlaca.setText(carro.getPlaca());
368         campoCor.setText(carro.getCor());
369         campoPreco.setText(String.valueOf(carro.getPreco()));
370         campoComb.setText(carro.getCombustivel());
371         campoCambio.setText(carro.getCambio());
372         campoPortas.setText(String.valueOf(carro.getPortas()));
373         campoKm.setText(String.valueOf(carro.getKm()));
374         campoVersao.setText(carro.getVersao());
375
376         ImageIcon icone = new ImageIcon(carro.getImagem());
377         icone.setImage(icone.getImage().getScaledInstance(painelImagem.getWidth() -5, painelImagem.getHeight() -10, 100));
378         image.setIcon(icone);
379     }
380 }
381

```

```

381
382     }
383 protected void do_btnAlterar_actionPerformed(ActionEvent arg0) {
384     int indice = table.getSelectedRow();
385     if(indice >= 0){
386         Veiculo veiculoEdicao = (Veiculo) model.getSelectedado(indice);
387         CadastroVeiculo guiEdicao = new CadastroVeiculo(veiculoEdicao, model);
388         guiEdicao.setVisible(true);
389         guiEdicao.setLocationRelativeTo(this);
390     }else{
391         JOptionPane.showMessageDialog(this, "Selecione um registro!");
392     }
393 }
394 protected void do_btnExcluir_actionPerformed(ActionEvent arg0) {
395     int indice = table.getSelectedRow();
396     if(indice >= 0){
397         Veiculo veiculoExclusao = (Veiculo) model.getSelectedado(indice);
398         if(JOptionPane.showConfirmDialog(this, "Deseja realmente excluir o registro?") == JOptionPane.YES_OPTION){
399             conexao.excluir(veiculoExclusao);
400             model.refreshData(conexao.listar());
401             JOptionPane.showMessageDialog(this, "Registro excluido com sucesso!");
402             int novoIndice = indice - 1;
403             if(novoIndice >= 0){
404                 carregarInfo(novoIndice);
405             }else{
406                 carregarInfo(novoIndice + 1);
407             }
408         }
409     }else{
410         JOptionPane.showMessageDialog(this, "Selecione um registro!");
411     }
412 }
413 }

```

## TableModel.java (Campo para listagem dos itens do banco na TelaVeiculo.java)

```
1 package util;
2
3 import java.util.List;
4
5
6
7
8
9 public class TableModel extends AbstractTableModel{
10
11
12 /**
13  *
14  */
15 private static final long serialVersionUID = -2564788613864602172L;
16
17 private List<Veiculo> lista;
18
19 private String[] colunas = {"Código", "Modelo", "Marca", "Ano", "Placa", "Cor",
20 "Preço", "Combustível", "Câmbio", "Portas", "Quilometragem", ""};
21
22
23 public TableModel(List<Veiculo> lista) {
24     this.lista = lista;
25 }
26
27
28 @Override
29 public int getColumnCount() {
30     return colunas.length;
31 }
32
33
34 @Override
35 public int getRowCount() {
36     return lista.size();
37 }
38
39 @Override
40 public String getColumnName(int columnIndex){
41     return colunas[columnIndex];
42 }
43
44 @Override
45 public Class<?> getColumnClass(int columnIndex){
46     return String.class;
47 }
48
```

```

49 public void setValueAt(Object aValue, int rowIndex, int columnIndex){
50     Veiculo veiculo = lista.get(rowIndex);
51
52     switch(columnIndex){
53     case 0:
54         veiculo.setIdveiculo(Long.parseLong(aValue.toString()));
55     case 1:
56         veiculo.setModelo(aValue.toString());
57     case 2:
58         veiculo.setMarca(aValue.toString());
59     case 3:
60         veiculo.setAno(Integer.parseInt(aValue.toString()));
61     case 4:
62         veiculo.setPlaca(aValue.toString());
63     case 5:
64         veiculo.setCor(aValue.toString());
65     case 6:
66         veiculo.setPreco(Double.parseDouble(aValue.toString()));
67     case 7:
68         veiculo.setCombustivel(aValue.toString());
69     case 8:
70         veiculo.setCambio(aValue.toString());
71     case 9:
72         veiculo.setPortas(Integer.parseInt(aValue.toString()));
73     case 10:
74         veiculo.setKm(Integer.parseInt(aValue.toString()));
75     case 11:
76         veiculo.setVersao(aValue.toString());
77     default:
78         System.err.println("Indice da coluna Ã© invalido");
79     }
80     fireTableCellUpdated(rowIndex, columnIndex);
81 }
82
83
84 @Override
85 public Object getValueAt(int rowIndex, int columnIndex) {
86     Veiculo veiculoSelecionado = lista.get(rowIndex);
87     Object valueObject = null;
88
89     switch(columnIndex){
90     case 0:
91         valueObject = veiculoSelecionado.getIdveiculo();
92         break;

```



```

93         case 1:
94             valueObject = veiculoSelecioneado.getModelo();
95             break;
96         case 2:
97             valueObject = veiculoSelecioneado.getMarca();
98             break;
99         case 3:
100             valueObject = veiculoSelecioneado.getAno();
101             break;
102         case 4:
103             valueObject = veiculoSelecioneado.getPlaca();
104             break;
105         case 5:
106             valueObject = veiculoSelecioneado.getCor();
107             break;
108         case 6:
109             valueObject = veiculoSelecioneado.getPreco();
110             break;
111         case 7:
112             valueObject = veiculoSelecioneado.getCombustivel();
113             break;
114         case 8:
115             valueObject = veiculoSelecioneado.getCambio();
116             break;
117         case 9:
118             valueObject = veiculoSelecioneado.getPortas();
119             break;
120         case 10:
121             valueObject = veiculoSelecioneado.getKm();
122             break;
123         case 11:
124             valueObject = veiculoSelecioneado.getVersao();
125             break;
126         default:
127     }
128
129     return valueObject;
130 }
131
132 @Override
133 public boolean isCellEditable(int rowIndex, int columnIndex){
134     return false;
135 }

```

```
136
137
138● public Object getSelecionado(int rowIndex){
139     return lista.get(rowIndex);
140 }
141
142● public void refreshData(List<Veiculo> novosVeiculos){
143     this.lista = novosVeiculos;
144     fireTableDataChanged();
145 }
146
147● public void limpar(){
148     lista.clear();
149     fireTableDataChanged();
150 }
151
152● public boolean isEmpty(){
153     return lista.isEmpty();
154 }
155
156 }
```

## 7. MANUAL DO USUÁRIO

Para visualizar o carro cadastrado, basta selecionar o mesmo na **tabela geral** que contém todos os carros cadastrados.

**AUTO MOTORS DO ZÉ**

Modelo:  Marca:

Placa:  Versão:  Ano:

Cor:  Portas:

Preço:  Combustível:

Câmbio:  Quilometragem:

Código	Modelo	Marca	Ano	Placa	Cor	Preço	Combustível	Câmbio	Portas	Quilometragem	Versão
1	Uno	Fiat	2002	ABC-1598	Azul Mar	5000.0	Flex	Manual	4	21526	1.6
2	Toro	Fiat	2022	RTG-8975	Vermelho	110000.0	Gasolina	Manual	4	8000	2.0
3	206	Peugeot	2002	ABC-9876	Prata	6500.0	Alcool	Manual	2	18759	1.6
5	Camaro	Chevrolet	2006	CAM-8975	Amarelo	125000.0	Gasolina	Automatico	2	55489	2.0

Para realizar o cadastro de um novo veículo, basta clicar em “**Novo**”;

Para realizar a alteração de um veículo, selecione o mesmo na **tabela geral** e clique em “**Alterar**”;

Para realizar a exclusão de um veículo, selecione o mesmo na **tabela geral** e clique em “**Excluir**”.



Quando o botão “**Novo**” é pressionado, a janela de cadastro é aberta para que seja preenchido os dados do veículo.

A screenshot of a web application window titled "Novo" (New) for vehicle registration. The window contains a form with the following fields: Modelo (Model), Marca (Brand), Ano (Year), Placa (Plate), Versão (Version), Cor (Color), Portas (Doors), Preço (Price), Combustível (Fuel), Câmbio (Transmission), and Quilometragem (Mileage). Each field is represented by a text input box. To the right of the form is a large empty rectangular area for an image, with a button labeled "Adicionar Imagem" (Add Image) below it. At the bottom of the window are two buttons: "Salvar" (Save) and "Cancelar" (Cancel).

Quando o botão “**Alterar**” é pressionado, a janela de atualização é aberta para que seja alterado os dados do veículo selecionado.

A screenshot of a web application window titled "Alterar" (Edit) for vehicle update. The window contains a form with the following fields: Modelo (Model), Marca (Brand), Ano (Year), Placa (Plate), Versão (Version), Cor (Color), Portas (Doors), Preço (Price), Combustível (Fuel), Câmbio (Transmission), and Quilometragem (Mileage). Each field is represented by a text input box. To the right of the form is a large rectangular area containing an image of a blue Fiat Uno car, with a button labeled "Adicionar Imagem" (Add Image) below it. At the bottom of the window are two buttons: "Salvar" (Save) and "Cancelar" (Cancel).

## 8. FERRAMENTAS UTILIZADAS

Para o desenvolvimento do software para a Auto Motors do Zé, foram utilizadas as seguintes ferramentas:

- **Eclipse IDE for Enterprise Java and Web Developers:** Eclipse é uma IDE para desenvolvimento Java, porém suporta várias outras linguagens a partir de plugins

como C/C++, PHP, ColdFusion, Python, Scala e Kotlin. Ele foi feito em Java e segue o modelo open source de desenvolvimento de software;

- **WindowBuilder:** possibilita a construção de telas empregando as principais APIs gráficas do Java Desktop. Como principais benefícios da utilização de plugins para o desenvolvimento de telas, temos a facilidade e a agilidade na projeção e construção das janelas;
- **XAMPP:** é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte as linguagens PHP e Perl;
- **phpMyAdmin:** é um aplicativo web livre e de código aberto desenvolvido em PHP para administração do MySQL pela Internet. A partir deste sistema é possível criar e remover bases de dados, criar, remover e alterar tabelas, inserir, remover e editar campos, executar códigos SQL e manipular campos chaves;
- **MySQL:** é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface;
- **MySQL Workbench:** é uma ferramenta visual de design de banco de dados que integra desenvolvimento, administração, design, criação e manutenção de SQL em um único ambiente de desenvolvimento integrado para o sistema de banco de dados MySQL;
- **Google Chrome:** é um navegador de internet desenvolvido pela Google;
- **GitHub:** é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.