



Universidade Federal de Sergipe
Centro de Ciências Exatas e Tecnologia
DEPARTAMENTO DE COMPUTAÇÃO - DCOMP
CIÊNCIA DA COMPUTAÇÃO

Documentos Desenvolvimento de Software - Concessionária

Prof. Michel dos Santos Soares

São Cristóvão, Sergipe
Outubro – 2016

Autores:

GRUPO 03

DIMITRI CARVALHO MENEZES – 201120000786
JOÃO MATEUS SANTANA DA CUNHA – 201110007166
KEOMAS DA SILVA SANTOS – 201220001370
LUCAS RENATO ARAGÃO SILVA – 201220001325
RAABE NOA SANTOS CORREIA – 201110008850
THARLYSSON BRENO LIMA DE MENEZES – 201220002117

GRUPO 04

JOMAR GONÇALVES RAMOS – 201020000940
LUCAS DE OLIVEIRA MACÊDO – 201500018252

Conteúdo

1	Levantamento de Requisitos	5
1.1	Propósito do Documento	5
1.2	Escopo do Produto	5
1.3	Definições e Abreviações	5
1.3.1	Definições	5
1.3.2	Abreviações	5
1.4	Referências	5
1.5	Visão Geral do Restante do Documento	6
1.6	Descrição Geral	6
1.6.1	Perspectiva do Produto	6
1.6.2	Funções do Produto	6
1.6.3	Características do Usuário	6
1.6.4	Restrições Gerais	6
1.6.5	Suposições e Dependências	7
1.7	Requisitos específicos	7
1.7.1	Requisitos Funcionais	7
1.7.2	Requisitos Não Funcionais	10
2	Plano de Projeto	11
2.1	Motivação	11
3	Casos de Uso	12
4	Arquitetura	15
5	Códigos	16
5.1	Pacote Cliente	16
5.1.1	Model	16
5.1.2	View	18
5.1.3	Controller	20
5.2	Pacote Fornecedor	22
5.2.1	Model	22
5.2.2	View	24
5.2.3	Controller	25
5.3	Pacote Funcionário	27
5.3.1	Model	27
5.3.2	View	31
5.3.3	Controller	32
5.4	Pacote Peça	35
5.4.1	Model	35
5.4.2	View	44
5.4.3	Controller	47
5.5	Pacote Serviço	55
5.5.1	Model	55

5.5.2	View	57
5.5.3	Controller	58
5.6	Pacote Veiculo	60
5.6.1	Model	60
5.6.2	View	62
5.6.3	Controller	63
5.7	Pacote Utilidades	65
5.8	Outros	66
5.9	SQL	71
6	Diagramas	77
6.1	Diagrama de Casos de Uso	77
6.2	Diagrama de Classes - Analise	78
6.3	Diagrama de Classes - Projeto	79
6.4	Diagrama Entidade Relacionamento	80

1 Levantamento de Requisitos

1.1 Propósito do Documento

O objetivo deste documento é detalhar a descrição de requisitos do software OpenCarShop, deixar claro a motivação do desenvolvimento do sistema, bem como funcionalidades, interfaces, componentes, interações e restrições que o software contém. Este documento, deve ser aprovado pelos stakeholders, e assim, servir de referência para o time de desenvolvimento, auxiliando na evolução do software.

1.2 Escopo do Produto

O OpenCarShop é um sistema de gestão que controlará os setores de venda de veículos, estoque, realização de orçamentos de serviços de uma concessionária de veículos de única marca.

Uma base de dados de veículos, peças, serviços, clientes e funcionários deve ser produzida e atualizada a medida que os usuários do sistema, os funcionários, alterem e adicionem tais dados durante a utilização do sistema.

Os funcionários que irão interagir com o software o farão através de seus desktops. O software necessita de conexão com o servidor de dados para que os funcionários se autenticuem no sistema e manipulem os dados.

1.3 Definições e Abreviações

1.3.1 Definições

- Funcionário: Ator principal do sistema.
- Orçamento: Levantamento de preços de serviços e peças atreladas a esses serviços.
- Peça: Peça mecânica ou acessório veicular.
- Serviço: Serviço veicular, reparo, manutenção, instalação de peça.

1.3.2 Abreviações

- RF: Requisito Funcional.
- RNF: Requisito Não Funcional.
- CDU: Caso de Uso.

1.4 Referências

- 1 Material usado nas aulas da disciplina Desenvolvimento de Software II ministrada pelo professor Michel dos Santos Soares disponibilizado em www.sigaa.ufs.br
- 2 Pressman, Roger. Engenharia de Software: Uma abordagem profissional. Porta Alegre: AMGH, 2011.

1.5 Visão Geral do Restante do Documento

O restante deste documento inclui dois capítulos e um apêndice. O Segundo capítulo apresenta uma descrição geral do sistema, ou seja, uma perspectiva funcional e objetivos do mesmo, descrição de seus usuários, restrições e dependências para utilização e desenvolvimento do sistema.

O Terceiro capítulo detalha os requisitos: especifica todos os requisitos funcionais e não funcionais que devem ser implementados. ser implementados.

1.6 Descrição Geral

1.6.1 Perspectiva do Produto

O sistema consistirá em uma aplicação desktop. A aplicação será usada para gerenciar vendas de peças e veículos, orçamentos de diversos serviços, controlar estoque de peças e veículos, gerir clientes e funcionários, e exibir relatórios. As funcionalidades devem estar disponíveis em uma interface gráfica, responsável pela intermediação do funcionário com a manipulação dos dados.

Os dados devem ser persistidos, em um banco de dados. Isso quer dizer que o sistema será capaz de salvar dados e recuperar dados do banco de dados. Os usuários devem ter um desktop conectado ao servidor de dados local.

1.6.2 Funções do Produto

A seguir, detalha-se cinco das mais importantes funcionalidades do sistema. É apresentado os casos de usos de interação do ator principal, o Funcionário, e os fluxos principais e alternativos. Os demais casos de usos se encontram na seção de diagramas:

1.6.3 Características do Usuário

Gerente: Responsável pela gestão da concessionária, tem acesso as todas funcionalidades do sistema Open Car Shop.

Funcionário: Responsável pelo atendimento ao cliente, geração de orçamento de serviços e vendas.

1.6.4 Restrições Gerais

O sistema deve ter no mínimo conexão com o banco de dados para que o funcionário se autenticar e poder utilizar os recursos do sistema.

Somente o gerente pode realizar o cadastro, atualização e solicitar listagem de funcionários e também realizar cadastro de fornecedor.

Apenas funcionários com contratos ativos podem ter acesso às funcionalidades do sistema.

1.6.5 Suposições e Dependências

- Ao gerar uma ordem de serviço, supõe-se que sempre há algum funcionário mecânico disponível para fazer o serviço.
- Existe dependência que uma venda possui em relação a quantidade de peças solicitadas.

1.7 Requisitos específicos

1.7.1 Requisitos Funcionais

...

Os requisitos listados abaixo, são funcionalidades que o funcionário pode interagir com o sistema.

RF1 Autenticar Funcionário (Pr.: 1):

Descrição: O sistema deve autenticar os funcionários, por meio de cpf e senha, de forma a não permitir acesso não autorizado.

RF2 Cadastrar Cliente (Pr.: 1):

O sistema deve permitir ao funcionário cadastrar clientes.

RF3 Inativar Cliente (Pr.: 1):

O sistema deve permitir ao funcionário inativar cadastro de clientes.

RF4 Atualizar Cliente (Pr.: 1):

O sistema deve permitir ao funcionário atualizar cadastro de clientes.

RF5 Listar Cliente (Pr.: 1):

O sistema deve listar os clientes para o funcionário.

RF6 Cadastrar Funcionário (Pr.: 1):

Descrição: O sistema deve permitir ao gerente cadastrar funcionários.

RF7 Atualizar Funcionário (Pr.: 1):

O sistema deve permitir ao gerente atualizar os dados dos funcionários.

RF8 Inativar Funcionário (Pr.: 1):

O sistema deve permitir ao gerente a inativação de funcionários..

RF9 Listar Funcionário. (Pr.: 1):

Descrição: O sistema deve permitir listar os funcionários pelo gerente.

RF10 Cadastrar Serviço (Pr.: 1):

O sistema deve permitir ao funcionário cadastrar serviços.

RF11 Atualizar Serviço (Pr.: 1):

O sistema deve permitir ao funcionário atualizar serviços.

RF12 Listar Serviço (Pr.: 1):

O sistema deve listar os Serviços para o funcionário.

RF13 Finalizar Ordem de Serviço (Pr.: 1):

O sistema deve permitir ao funcionário finalizar ordens de serviços.

RF14 Cadastrar Fornecedor (Pr.: 1):

O sistema deve permitir ao gerente cadastrar fornecedores.

RF15 Atualizar Fornecedor (Pr.: 1):

O sistema deve permitir ao gerente atualizar fornecedores.

RF16 Inativar Fornecedor (Pr.: 1):

O sistema deve permitir ao gerente inativar fornecedores.

RF17 Listar Fornecedor (Pr.: 1):

O sistema deve listar os Fornecedores para o gerente.

RF18 Cadastrar Veículo (Pr.: 1):

Descrição: O sistema deve permitir ao funcionário cadastrar veículos.

RF19 Listar Veículos (Pr.: 1):

O sistema deve permitir ao funcionário listar os veículos.

RF20 Atualizar Estoque de Veículos (Pr.: 1):

O sistema deve permitir ao funcionário atualizar a quantidade de itens de uma determinada peça no estoque.

RF21 Cadastrar Peça (Pr.: 1):

O sistema deve permitir ao funcionário cadastrar peças.

RF22 Atualizar Peça (Pr.: 1):

O sistema deve permitir ao funcionário atualizar dados da peças.

RF23 Atualizar Estoque de Peça (Pr.: 1):

O sistema deve permitir ao funcionário atualizar a quantidade de itens de uma determinada peça no estoque.

RF24 Inativar Peça (Pr.: 1):

O sistema deve permitir ao funcionário a inativação de peças.

RF25 Listar Peças (Pr.: 1):

O sistema deve permitir ao funcionário listar as peças..

RF26 Orçar serviços (Pr.: 1):

O sistema deve permitir ao funcionário gerar um orçamento de serviços solicitado pelo cliente.

RF27 Vender Peça (Pr.: 1):

O sistema deve permitir ao funcionário realizar a venda de itens de peça para um cliente.

RF28 Vender Veículo (Pr.: 1):

O sistema deve permitir ao funcionário realizar a venda de veículos para um cliente.

RF29 Autorizar Serviço (Pr.: 1):

O sistema deve permitir ao funcionário registrar a contratação de serviços a partir de um orçamento de serviços válido.

RF30 Pagamento de Venda de Veículos (Pr.: 1):

O sistema deve armazenar os pagamentos das vendas de veículos.

RF31 Pagamento de Venda de Peças (Pr.: 1):

O sistema deve armazenar os pagamentos das vendas de peças.

RF32 Pagamento de Contratação de Serviços (Pr.: 1):

O sistema deve armazenar o pagamento da contratação de serviços. .

RF33 Gerar Comprovante de Pagamento da venda de peças (Pr.: 1):

O sistema deve gerar um comprovante de pagamento pela venda de peças.

RF34 Gerar Comprovante de Pagamento da venda de veículos (Pr.: 1):

O sistema deve gerar um comprovante de pagamento pela venda de veículos.

RF35 Gerar Comprovante de Pagamento de Contratação de serviço (Pr.: 1):

O sistema deve efetuar a exclusão de fornecedores.

RF36 Gerar Comprovante de Pagamento de Contratação de serviço (Pr.: 1):

O sistema deve gerar um comprovante de pagamento pela venda de peças, veículos ou contratação de serviços.

RF37 Verificar disponibilidade (Pr.: 1):

O sistema deve verificar se a peça está disponível no estoque antes da venda.

RF38 Atualizar Estoque (Pr.: 1):

O sistema deve atualizar a quantidade de peças e de veículos após concretizar vendas.

RF39 Relatório de Faturamento (Pr.: 1):

O sistema deve exibir relatório de quantidade de vendas solicitado pelo gerente.

RF40 Relatório de Cliente (Pr.: 1):

O sistema deve exibir histórico de compra de clientes solicitado pelo funcionário.

1.7.2 Requisitos Não Funcionais

RNF1 Integridade (Pr.: 1):

O sistema deve permitir apenas usuários com privilégios de gerente visualizar informações de contrato dos funcionários.

RNF2 Tempo de Resposta (Pr.: 1):

O tempo de processamento para todas as requisições devem ser de 2 segundos para 90

RNF3 Usuários Simultâneos (Pr.: 1):

O sistema deverá suportar processamento multiusuários, até 50 usuários poderão utilizar o sistema simultaneamente.

RNF4 Interface gráfica (Pr.: 1):

Para um teste com 20 usuários, o tempo para o 90

RNF5 Portabilidade (Pr.: 1):

O sistema deverá ser independente de plataforma de sistema operacional.

2 Plano de Projeto

Descrever Plano de Projeto

2.1 Motivação

Motivação para o projeto:

- Motivação 1
- Motivação 2
- Motivação 3

3 Casos de Uso

Nome: Autenticar Funcionário.

Descrição: Autenticação dos funcionários para uso do sistema.

Identificador: CDU01.

Ator Primário: Funcionário.

Fluxo principal

Funcionário	Sistema
1 - Inserir cpf e senha	
	2 - Valida dados inseridos
	3 - Exibe opções disponíveis

Fluxo Alternativo(Login ou senha incorreto, funcionário inexistente ou inativado)

Funcionário	Sistema
1 - Inserir cpf e senha	
	2 - Valida dados inseridos
	3 - Exibe mensagem de falha

Nome: Cadastrar Cliente.

Descrição: Funcionário cadastra dados do cliente no sistema.

Identificador: CDU02.

Ator Primário: Funcionário

Precondição: Funcionário deve estar autenticado no sistema.

Fluxo principal

Funcionário	Sistema
1 - Selecionar opção de cadastrar cliente	
	2 - Exibir formulário de cadastro
3 - Preencher dados de cadastro	
4 - Selecionar opção de confirmar cadastro.	
	5 - Salvar cadastro.
	6 - Exibir Mensagem “Cadastro Realizado”.

Fluxo Alternativo(Cliente já cadastrado)

Funcionário	Sistema
1 - Selecionar opção de cadastrar cliente	
	2 - Exibir formulário de cadastro
3 - Preencher dados de cadastro	
4 - Selecionar opção de confirmar cadastro.	
	5 - Retornar Mensagem “Usuário já cadastrado.”
	6 - Exibir opção de atualizar dados ou Sair.

Nome: Orçar Serviços.

Descrição: Gerar orçamento de um serviço.

Identificador: CDU03

Ator Primário: Funcionário

Precondição: Funcionário deve estar autenticado no sistema.

Fluxo principal

Funcionário	Sistema
1 - Selecionar opção “Serviços”.	
	2 - Exibir tela menu de Serviços.
3 - Selecionar a opção “Orçar Serviço”	
	4 - Exibir lista de serviços cadastrados.
5 - Selecionar um ou mais serviços.	
6 - Selecionar opção “Continuar”	
	7 - Exibir tela com preços para cada serviço com opção de alterar para serviços sem preço fixo.
8 - Alterar os preços de serviço que possuam opção de alterar preço.	
9 - Selecionar opção “Adicionar peça”	
	10 - Exibir tela para seleção de peças.
11 - Selecionar uma ou mais peças.	
12 - Selecionar opção “Continuar”.	
	13 - Exibir tela de resumo com serviço(s) e peça(s) selecionados .
14 - Inserir placa do veículo do cliente.	
15 - Selecionar opção “Confirmar orçamento”.	
	16 - Exibir Documento de Orçamento com descrição dos serviços, placa do veículo, código de identificação e custo total.
17 - Selecionar opção “imprimir”.	
	18 - Encaminhar documento para impressão.

Nome: Vender Veículo

Descrição: Realizar venda de veículo

Identificador: CDU04

Ator Primário: Funcionário

Precondição: Funcionário deve estar autenticado no sistema.

Fluxo principal

Funcionário	Sistema
1 - Selecionar opção de veículos.	
	2 - Exibir tela menu de veículos.
3 - Selecionar opção de “Vender Veículo”	
	4 - Exibir veículos para venda.
5 - Selecionar veículo para venda.	
6 - Selecionar opção “Continuar.	
	7 - Exibir tela para seleção de cliente
8 - Selecciona cliente	
9 - Selecciona opção “Continuar”.	
	10 - Atualiza Quantidade do Veículo no Estoque
	11 - Exibe documento de comprovação de venda.
	12 - Gera cobrança.
13 - Selecciona opção “imprimir comprovante de venda”.	
14 - Selecciona opção “imprimir cobrança”.	

Nome: Autenticar Funcionário.

Descrição: Atualizar quantidade de peças no estoque.

Identificador: CDU05

Ator Primário: Funcionário.

Precondição: Funcionário deve estar autenticado no sistema.

Fluxo principal

Funcionário	Funcionário
1 - Selecionar opção de peças.	
	2 - Exibir menu de peças
3 - Selecionar opção de gerenciar	
	4 - Exibir listagem de peças.
5 - Clicar em uma Peça.	
6 - Atualizar Quantidade de Peças.	
7 - Confirmar alteração.	
	8 - Salvar alterações.

4 Arquitetura

5 Códigos

5.1 Pacote Cliente

5.1.1 Model

```
1 package opencarshop.cliente.model;
2
3 import java.time.LocalDate;
4
5 public class Cliente {
6
7     private String cpf;
8     private String nome;
9     private LocalDate dataNascimento;
10    private String email;
11    private String telefone1;
12    private String telefone2;
13    private Boolean ativo;
14
15    public String getCpf() {
16        return cpf;
17    }
18
19    public void setCpf(String cpf) {
20        this.cpf = cpf;
21    }
22
23    public String getNome() {
24        return nome;
25    }
26
27    public void setNome(String nome) {
28        this.nome = nome;
29    }
30
31    public LocalDate getDataNascimento() {
32        return dataNascimento;
33    }
34
35    public void setDataNascimento(LocalDate dataNascimento) {
36        this.dataNascimento = dataNascimento;
37    }
38
39    public String getEmail() {
40        return email;
41    }
42
43    public void setEmail(String email) {
44        this.email = email;
45    }
46
47    public String getTelefone1() {
48        return telefone1;
49    }
50
51    public void setTelefone1(String telefone1) {
52        this.telefone1 = telefone1;
53    }
54
55    public String getTelefone2() {
56        return telefone2;
57    }
58
59    public void setTelefone2(String telefone2) {
60        this.telefone2 = telefone2;
61    }
62
63    public Boolean getAtivo() {
64        return ativo;
65    }
66
67    public void setAtivo(Boolean ativo) {
68        this.ativo = ativo;
69    }
70 }
```

Código 1: Cliente.java

```
1 package opencarshop.cliente.model;
2
3 import java.sql.Connection;
4 import java.sql.Date;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.util.ArrayList;
```



```

9  import java.util.List;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12 import opencarshop.Endereco;
13 import opencarshop.cliente.model.Cliente;
14 import opencarshop.util.ConexaoMySQL;
15 import opencarshop.util.Utilidades;
16
17 public class ClienteDAO {
18
19     private Connection conn;
20     private final ConexaoMySQL c = new ConexaoMySQL();
21
22     public boolean cadastraCliente(Cliente cli, Endereco end) {
23         Connection conn = null;
24
25         PreparedStatement stmtEnd = null;
26         PreparedStatement stmtCli = null;
27
28         String queryEnd = "INSERT INTO Endereco(cep,estado,cidade,bairro,rua,numero,complemento,tipo)VALUES(
29             (?,?,?,?,?,?,?))";
30         String queryFun = "INSERT INTO Cliente(cpf,nome,dataNascimento,email,telefone1,telefone2,endereco,
31             ativo)VALUES(?,?,?,?,?,(select LAST_INSERT_ID()),?)";
32
33         try {
34             conn = c.conectar();
35             conn.setAutoCommit(false);
36
37             stmtEnd = conn.prepareStatement(queryEnd);
38             stmtCli = conn.prepareStatement(queryFun);
39
40             stmtEnd.setString(1, end.getCEP());
41             stmtEnd.setString(2, end.getEstado());
42             stmtEnd.setString(3, end.getCidade());
43             stmtEnd.setString(4, end.getBairro());
44             stmtEnd.setString(5, end.getRua());
45             stmtEnd.setInt(6, end.getNumero());
46             stmtEnd.setString(7, end.getComplemento());
47             stmtEnd.setString(8, Character.toString(end.getTipo()));
48
49             stmtCli.setString(1, cli.getCpf());
50             stmtCli.setString(2, cli.getNome());
51             stmtCli.setDate(3, Date.valueOf(cli.getDataNascimento()));
52             stmtCli.setString(4, cli.getEmail());
53             stmtCli.setString(5, cli.getTelefone1());
54             stmtCli.setString(6, cli.getTelefone2());
55             stmtCli.setBoolean(7, true);
56
57             stmtEnd.execute();
58             stmtCli.execute();
59
60             conn.commit();
61
62             conn.close();
63             return true;
64         } catch (Exception e) {
65             e.printStackTrace();
66             return false;
67         }
68     }
69
70     public List<Cliente> getAllCliente() throws Exception {
71         String query = "SELECT * FROM Cliente";
72         List<Cliente> retorno = new ArrayList<>();
73         Utilidades u = new Utilidades();
74         ConexaoMySQL c = new ConexaoMySQL();
75         Connection conn = null;
76         conn = c.conectar();
77
78         try {
79             PreparedStatement stmt = conn.prepareStatement(query);
80             ResultSet resultado = stmt.executeQuery();
81             while (resultado.next()) {
82                 Cliente cliente = new Cliente();
83                 cliente.setCpf(resultado.getString("cpf"));
84                 cliente.setNome(resultado.getString("nome"));
85                 cliente.setDataNascimento(u.toLocalDate(resultado.getDate("dataNascimento")));
86                 cliente.setEmail(resultado.getString("email"));
87                 cliente.setTelefone1(resultado.getString("telefone1"));
88                 cliente.setTelefone2(resultado.getString("telefone2"));
89                 cliente.setAtivo(resultado.getBoolean("ativo"));
90
91                 retorno.add(cliente);
92             }
93         } catch (Exception e) {
94             e.printStackTrace();
95         }
96         conn.close();
97         return retorno;
98     }
99
100     public Cliente buscar(Cliente cliente) {
101         String sql = "SELECT * FROM Cliente WHERE cpf=?";
102         Cliente retorno = new Cliente();
103         Utilidades u = new Utilidades();

```

```

101     try {
102         PreparedStatement stmt = conn.prepareStatement(sql);
103         stmt.setString(1, cliente.getCpf());
104         ResultSet resultado = stmt.executeQuery();
105         if (resultado.next()) {
106             cliente.setCpf(resultado.getString("cpf"));
107             cliente.setNome(resultado.getString("nome"));
108             cliente.setDataNascimento(u.toLocalDate(resultado.getDate("dataNascimento")));
109             cliente.setEmail(resultado.getString("email"));
110             cliente.setTelefone1(resultado.getString("telefone1"));
111             cliente.setTelefone2(resultado.getString("telefone2"));
112             cliente.setAtivo(resultado.getBoolean("ativo"));
113             retorno = cliente;
114         }
115     } catch (SQLException ex) {
116         Logger.getLogger(ClienteDAO.class.getName()).log(Level.SEVERE, null, ex);
117     }
118     return retorno;
119 }
120
121 public Boolean alteraCliente(Cliente cli) throws SQLException {
122     String query = "UPDATE Cliente SET nome=?, dataNascimento=?, email=?, telefone1=?, telefone2=?, ativo=? WHERE cpf=?";
123
124     try {
125         conn = c.conectar();
126         PreparedStatement stmt = conn.prepareStatement(query);
127
128         stmt.setString(1, cli.getNome());
129         stmt.setDate(2, Date.valueOf(cli.getDataNascimento()));
130         stmt.setString(3, cli.getEmail());
131         stmt.setString(4, cli.getTelefone1());
132         stmt.setString(5, cli.getTelefone2());
133         stmt.setBoolean(6, cli.getAtivo());
134         stmt.setString(7, cli.getCpf());
135         stmt.execute();
136         conn.close();
137         return true;
138     } catch (Exception ex) {
139
140         Logger.getLogger(ClienteDAO.class.getName()).log(Level.SEVERE, null, ex);
141         return false;
142     }
143 }
144
145 public Connection getConnection() {
146     return conn;
147 }
148
149 public void setConnection(Connection connection) {
150     this.conn = connection;
151 }
152 }

```

Código 2: ClienteDAO.java

5.1.2 View

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.chart.*>
4  <?import javafx.scene.text.*>
5  <?import java.lang.*>
6  <?import java.util.*>
7  <?import javafx.scene.*>
8  <?import javafx.scene.control.*>
9  <?import javafx.scene.layout.*>
10
11 <?import javafx.collections.*>
12
13 <AnchorPane id="AnchorPane" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
14   fx:controller="opencarshop.cliente.controller.ClienteController">
15     <children>
16       <TabPane prefHeight="494.0" prefWidth="1024.0" tabClosingPolicy="UNAVAILABLE">
17         <tabs>
18           <Tab text="Identificacao">
19             <content>
20               <AnchorPane minHeight="0.0" minWidth="0.0" prefWidth="1024.0">
21                 <children>
22                   <TextField fx:id="tf_cpfCadastro" layoutX="194.0" layoutY="71.0" prefHeight="25.0"
23                     prefWidth="170.0" promptText="CPF" />
24                   <TextField fx:id="tf_nomeCadastro" layoutX="14.0" layoutY="25.0" prefHeight="25.0"
25                     prefWidth="350.0" promptText="Nome completo" />
26                   <DatePicker fx:id="dp_dataNascimentoCadastro" layoutX="14.0" layoutY="71.0" prefHeight=
27                     "25.0" prefWidth="170.0" promptText="Data de Nascimento" />
28                   <Label layoutX="14.0" layoutY="6.0" text="Nome:" />
29                   <Label layoutX="14.0" layoutY="50.0" text="Data de Nascimento:" />
30                   <Label layoutX="194.0" layoutY="50.0" text="CPF:" />

```

```

27         </children>
28     </AnchorPane>
29 </content>
30 </Tab>
31 <Tab text="Contato">
32     <content>
33         <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="258.0" prefWidth="1024.0">
34             <children>
35                 <TextField fx:id="tf_emailCadastro" layoutX="14.0" layoutY="26.0" prefHeight="25.0"
36 prefWidth="350.0" promptText="Email" />
37                 <TextField fx:id="tf_telefone1Cadastro" layoutX="14.0" layoutY="70.0" prefWidth="170.0"
38 promptText="Telefone_1" />
39                 <TextField fx:id="tf_telefone2Cadastro" layoutX="194.0" layoutY="70.0" prefWidth="170.0"
40 promptText="Telefone_2" />
41                 <TextField fx:id="tf_ruaCadastro" layoutX="14.0" layoutY="160.0" prefWidth="350.0"
42 promptText="Rua" />
43                 <TextField fx:id="tf_cidadeCadastro" layoutX="14.0" layoutY="249.0" prefWidth="350.0"
44 promptText="Cidade" />
45                 <TextField fx:id="tf_estadoCadastro" layoutX="14.0" layoutY="293.0" prefWidth="170.0"
46 promptText="UF" />
47                 <TextField fx:id="tf_bairroCadastro" layoutX="194.0" layoutY="293.0" prefWidth="170.0"
48 promptText="Bairro" />
49                 <TextField fx:id="tf_cepCadastro" layoutX="14.0" layoutY="336.0" prefWidth="170.0"
50 promptText="CEP" />
51                 <TextField fx:id="tf_numeroCadastro" layoutX="194.0" layoutY="336.0" prefWidth="170.0"
52 promptText="Número" />
53                 <TextField fx:id="tf_complementoCadastro" layoutX="14.0" layoutY="204.0" prefWidth="
54 350.0" promptText="Complemento" />
55                 <Button id="btn_cadastrar" layoutX="151.0" layoutY="414.0" mnemonicParsing="false"
56 onAction="#cadastrar" text="Cadastrar" />
57                 <ComboBox fx:id="cb_tipoCadastro" layoutX="14.0" layoutY="115.0" prefWidth="350.0"
58 promptText="Tipo de Endereço">
59                     <items>
60                         <FXCollections fx:factory="observableArrayList">
61                             <String fx:value="Residencial" />
62                             <String fx:value="Comercial" />
63                         </FXCollections>
64                     </items>
65                 </ComboBox>
66                 <Label layoutX="14.0" layoutY="6.0" text="Email:" />
67                 <Label layoutX="16.0" layoutY="51.0" text="Telefone_1:" />
68                 <Label layoutX="194.0" layoutY="51.0" text="Telefone_2:" />
69                 <Label layoutX="16.0" layoutY="95.0" text="Endereço:" />
70                 <Label layoutX="16.0" layoutY="140.0" text="Rua:" />
71                 <Label layoutX="16.0" layoutY="185.0" text="Complemento:" />
72                 <Label layoutX="16.0" layoutY="233.0" text="Cidade:" />
73                 <Label layoutX="16.0" layoutY="274.0" text="UF-Estado:" />
74                 <Label layoutX="194.0" layoutY="274.0" text="Bairro:" />
75                 <Label layoutX="16.0" layoutY="318.0" text="CEP:" />
76                 <Label layoutX="197.0" layoutY="318.0" text="Número:" />
77                 <Label fx:id="resultadoCadastro" alignment="CENTER" layoutX="80.0" layoutY="384.0"
78 prefHeight="17.0" prefWidth="218.0" />
79             </children>
80         </AnchorPane>
81     </content>
82 </Tab>
83 </tabs>
84 </TabPane>
85 </children>
86 </AnchorPane>

```

Código 3: Cadastrar.fxml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import java.lang.*?>
4 <?import java.util.*?>
5 <?import javafx.scene.*?>
6 <?import javafx.scene.control.*?>
7 <?import javafx.scene.layout.*?>
8
9 <AnchorPane id="AnchorPane" prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
10 javafx.com/fxml/1" fx:controller="opencarshop.cliente.controller.ClienteController">
11     <children>
12         <TableView fx:id="tbl_cliente" prefHeight="650.0" prefWidth="650.0">
13             <columns>
14                 <TableColumn fx:id="col_cpf" prefWidth="100.0" text="CPF" />
15                 <TableColumn fx:id="col_nome" prefWidth="155.0" text="Nome" />
16                 <TableColumn id="col_telefone" fx:id="col_telefone1" minWidth="0.0" prefWidth="103.0" text="Telefone_1"
17 />
18                 <TableColumn id="col_telefone" fx:id="col_telefone2" prefWidth="98.0" text="Telefone_2" />
19                 <TableColumn id="col_email" fx:id="col_email" prefWidth="193.0" text="Email" />
20             </columns>
21         </TableView>
22         <TextField fx:id="tf_nomeCadastro" layoutX="846.0" layoutY="95.0" prefWidth="150.0" promptText="Nome" />
23         <TextField fx:id="tf_emailCadastro" layoutX="846.0" layoutY="151.0" promptText="Email" />
24         <TextField fx:id="tf_telefone1Cadastro" layoutX="675.0" layoutY="208.0" promptText="Telefone_1" />
25         <TextField fx:id="tf_telefone2Cadastro" layoutX="846.0" layoutY="208.0" promptText="Telefone_2" />
26         <DatePicker fx:id="dp_dataNascimentoCadastro" layoutX="675.0" layoutY="151.0" prefWidth="150.0" promptText="
27 Data de Nascimento" />
28         <Label layoutX="844.0" layoutY="78.0" text="Nome:" />
29         <Label layoutX="846.0" layoutY="134.0" text="Email:" />

```

```

27     <Label layoutX="678.0" layoutY="191.0" text="Telefone_1:" />
28     <Label layoutX="847.0" layoutY="191.0" text="Telefone_2:" />
29     <Label layoutX="678.0" layoutY="134.0" text="Data_de_Nascimento:" />
30     <CheckBox fx:id="cb_ativo" layoutX="674.0" layoutY="263.0" mnemonicParsing="false" text="Ativo" />
31     <Button fx:id="btn_alterarCadastro" layoutX="782.0" layoutY="359.0" mnemonicParsing="false" onAction="#
    alterarCadastro" text="Salvar_Alterar" />
32     <TextField fx:id="tf_cpfCadastro" editable="false" layoutX="675.0" layoutY="95.0" />
33     <Label layoutX="675.0" layoutY="78.0" text="CPF:" />
34     <Label fx:id="confirmaAlteracao" alignment="CENTER" contentDisplay="CENTER" layoutX="736.0" layoutY="325.0"
    prefHeight="17.0" prefWidth="199.0" textAlignment="CENTER" />
35 </children>
36 </AnchorPane>

```

Código 4: Buscar.fxml

5.1.3 Controller

```

1 package opencarshop.cliente.controller;
2
3 import java.net.URL;
4 import java.text.DecimalFormat;
5 import java.text.ParseException;
6 import java.util.List;
7 import java.util.ResourceBundle;
8 import javafx.collections.FXCollections;
9 import javafx.collections.ObservableList;
10 import javafx.event.ActionEvent;
11 import javafx.fxml.FXML;
12 import javafx.fxml.Initializable;
13 import javafx.scene.control.CheckBox;
14 import javafx.scene.control.ComboBox;
15 import javafx.scene.control.DatePicker;
16 import javafx.scene.control.Label;
17 import javafx.scene.control.PasswordField;
18 import javafx.scene.control.TableColumn;
19 import javafx.scene.control.TableView;
20 import javafx.scene.control.TextField;
21 import javafx.scene.control.cell.PropertyValueFactory;
22 import opencarshop.Endereco;
23 import opencarshop.cliente.model.Cliente;
24 import opencarshop.cliente.model.ClienteDAO;
25 import opencarshop.util.Utilidades;
26
27 public class ClienteController implements Initializable {
28
29     // TELA DE CADASTRO
30     @FXML
31     private TextField tf_cpfCadastro;
32     @FXML
33     private TextField tf_nomeCadastro;
34     @FXML
35     private DatePicker dp_dataNascimentoCadastro;
36     @FXML
37     private TextField tf_emailCadastro;
38     @FXML
39     private TextField tf_telefone1Cadastro;
40     @FXML
41     private TextField tf_telefone2Cadastro;
42
43     @FXML
44     private ComboBox<String> cb_tipoCadastro;
45     @FXML
46     private TextField tf_ruaCadastro;
47     @FXML
48     private TextField tf_cidadeCadastro;
49     @FXML
50     private TextField tf_estadoCadastro;
51     @FXML
52     private TextField tf_bairroCadastro;
53     @FXML
54     private TextField tf_cepCadastro;
55     @FXML
56     private TextField tf_numeroCadastro;
57     @FXML
58     private TextField tf_complementoCadastro;
59
60     @FXML
61     private Label resultadoCadastro;
62
63     // TABELA CLIENTE
64     @FXML
65     private TableColumn<Cliente, String> col_nome;
66     @FXML
67     private TableColumn<Cliente, String> col_cpf;
68     @FXML
69     private TableColumn<Cliente, String> col_telefone1;
70     @FXML
71     private TableColumn<Cliente, String> col_telefone2;

```

```

72 @FXML
73 private TableColumn<Cliente, String> col_email;
74
75 @FXML
76 private TableView<Cliente> tbl_cliente;
77
78 @FXML
79 private CheckBox cb_ativo;
80
81 @FXML
82 private Label confirmaAlteracao;
83
84 @FXML
85 private void cadastrar(ActionEvent event) throws ParseException {
86     //cb_cargoCadaastro.setItems(cargos);
87     Cliente cli = new Cliente();
88     Endereco end = new Endereco();
89     ClienteDAO c = new ClienteDAO();
90
91     // OBJETO FUNCIONARIO
92     cli.setCpf(tf_cpfCadaastro.getText());
93     cli.setNome(tf_nomeCadaastro.getText());
94     cli.setDataNascimento(dp_dataNascimentoCadaastro.getValue());
95     cli.setEmail(tf_emailCadaastro.getText());
96     cli.setTelefone1(tf_telefone1Cadaastro.getText());
97     cli.setTelefone2(tf_telefone2Cadaastro.getText());
98     cli.setAtivo(true);
99
100     // OBJETO ENDEREÇO
101     end.setCEP(tf_cepCadaastro.getText());
102     end.setEstado(tf_estadoCadaastro.getText());
103     end.setCidade(tf_cidadeCadaastro.getText());
104     end.setBairro(tf_bairroCadaastro.getText());
105     end.setRua(tf_ruaCadaastro.getText());
106     end.setNumero(Integer.parseInt(tf_numeroCadaastro.getText()));
107     end.setComplemento(tf_complementoCadaastro.getText());
108     end.setTipo(cb_tipoCadaastro.getValue().charAt(0));
109
110     if (c.cadastraCliente(cli, end)) {
111         resultadoCadaastro.setText("Cadastrado com sucesso!!");
112     } else {
113         resultadoCadaastro.setText("Erro ao cadastrar!! Tente novamente.");
114     }
115 }
116
117 @FXML
118 private void alterarCadaastro(ActionEvent event) throws Exception {
119     Cliente cli = new Cliente();
120     cli.setCpf(tf_cpfCadaastro.getText());
121     cli.setNome(tf_nomeCadaastro.getText());
122     cli.setDataNascimento(dp_dataNascimentoCadaastro.getValue());
123     cli.setEmail(tf_emailCadaastro.getText());
124     cli.setTelefone1(tf_telefone1Cadaastro.getText());
125     cli.setTelefone2(tf_telefone2Cadaastro.getText());
126     cli.setAtivo(cb_ativo.isSelected());
127
128     ClienteDAO f = new ClienteDAO();
129     if (f.alteraCliente(cli)) {
130         confirmaAlteracao.setText("Alteração realizada com sucesso!!");
131     } else {
132         confirmaAlteracao.setText("Erro ao realizar alteração!!");
133     }
134 }
135
136 private void carregaTabelaCliente() throws Exception {
137     col_nome.setCellValueFactory(new PropertyValueFactory<>("nome"));
138     col_cpf.setCellValueFactory(new PropertyValueFactory<>("cpf"));
139     col_telefone1.setCellValueFactory(new PropertyValueFactory<>("telefone1"));
140     col_telefone2.setCellValueFactory(new PropertyValueFactory<>("telefone2"));
141     col_email.setCellValueFactory(new PropertyValueFactory<>("email"));
142
143     ClienteDAO f = new ClienteDAO();
144     List<Cliente> listaCliente = f.getAllCliente();
145     ObservableList<Cliente> observableListFuncionario;
146
147     observableListFuncionario = FXCollections.observableArrayList(listaCliente);
148     tbl_cliente.setItems(observableListFuncionario);
149 }
150
151 public void selecionarItemTabelaCliente(Cliente cliente) {
152     if (cliente.getCpf() != null) {
153         tf_cpfCadaastro.setText(cliente.getCpf());
154         tf_nomeCadaastro.setText(cliente.getNome());
155         tf_emailCadaastro.setText(cliente.getEmail());
156         tf_telefone1Cadaastro.setText(cliente.getTelefone1());
157         tf_telefone2Cadaastro.setText(cliente.getTelefone2());
158         dp_dataNascimentoCadaastro.setValue(cliente.getDataNascimento());
159         cb_ativo.setSelected(cliente.getAtivo());
160     }
161 }
162
163 @Override
164 public void initialize(URL url, ResourceBundle rb) {

```

```

166     try {
167         carregaTabelaCliente();
168         tbl_cliente.getSelectionModel().selectedItemProperty().addListener(
169             (observable, oldValue, newValue) -> selecionarItemTabelaCliente(newValue));
170     } catch (Exception ex) {
171         //Logger.getLogger(ClienteController.class.getName()).log(Level.SEVERE, null, ex);
172     }
173 }
174
175 }

```

Código 5: ClienteController.java

5.2 Pacote Fornecedor

5.2.1 Model

```

1  package opencarshop.fornecedor.model;
2
3  public class Fornecedor {
4
5      private String cnpj;
6      private String razaoSocial;
7      private String email;
8      private String telefone1;
9      private String telefone2;
10     private String descricao;
11
12     public String getCnpj() {
13         return cnpj;
14     }
15
16     public void setCnpj(String cnpj) {
17         this.cnpj = cnpj;
18     }
19
20     public String getRazaoSocial() {
21         return razaoSocial;
22     }
23
24     public void setRazaoSocial(String razaoSocial) {
25         this.razaoSocial = razaoSocial;
26     }
27
28     public String getEmail() {
29         return email;
30     }
31
32     public void setEmail(String email) {
33         this.email = email;
34     }
35
36     public String getTelefone1() {
37         return telefone1;
38     }
39
40     public void setTelefone1(String telefone1) {
41         this.telefone1 = telefone1;
42     }
43
44     public String getTelefone2() {
45         return telefone2;
46     }
47
48     public void setTelefone2(String telefone2) {
49         this.telefone2 = telefone2;
50     }
51
52     public String getDescricao() {
53         return descricao;
54     }
55
56     public void setDescricao(String descricao) {
57         this.descricao = descricao;
58     }
59 }

```

Código 6: Fornecedor.java

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */

```

```

6 package opencarshop.fornecedor.model;
7
8 import java.sql.Connection;
9 import java.sql.PreparedStatement;
10 import java.sql.ResultSet;
11 import java.util.ArrayList;
12 import java.util.List;
13 import opencarshop.Endereco;
14 import opencarshop.util.ConexaoMySQL;
15 import opencarshop.util.Utilidades;
16
17 /**
18  *
19  * @author JomarR
20  */
21 public class FornecedorDAO {
22
23     public Fornecedor getFornecedor(String cnpj) {
24         ConexaoMySQL c = new ConexaoMySQL();
25         Connection conn = null;
26         PreparedStatement stmt = null;
27         Utilidades u = new Utilidades();
28
29         String query = "SELECT * FROM Fornecedor WHERE cnpj=?";
30
31         Fornecedor fornecedor = new Fornecedor();
32         try {
33             conn = c.conectar();
34             stmt = conn.prepareStatement(query);
35             stmt.setString(1, cnpj);
36
37             ResultSet resultado = stmt.executeQuery();
38
39             if (resultado.next()) {
40                 fornecedor.setCnpj(resultado.getString("cnpj"));
41                 fornecedor.setDescricao(resultado.getString("razaoSocial"));
42                 fornecedor.setEmail(resultado.getString("email"));
43                 fornecedor.setTelefone1(resultado.getString("telefone1"));
44                 fornecedor.setTelefone2(resultado.getString("telefone2"));
45                 fornecedor.setDescricao(resultado.getString("descricao"));
46             }
47
48             conn.close();
49         } catch (Exception e) {
50             e.printStackTrace();
51         }
52         return fornecedor;
53     }
54
55     public boolean cadastraFornecedor(Fornecedor fornecedor, Endereco end) {
56         ConexaoMySQL c = new ConexaoMySQL();
57         Connection conn = null;
58
59         PreparedStatement stmtEnd = null;
60
61         PreparedStatement stmtFornecedor = null;
62
63         String queryEnd = "INSERT INTO Endereco (cep, estado, cidade, bairro, rua, numero, complemento, tipo) VALUES (?,?,?,?,?,?,?)";
64         String queryFornecedor = "INSERT INTO Fornecedor (cnnpj, razaoSocial, email, telefone1, telefone2, descricao, endereco) VALUES (?,?,?,?,?,?(select LAST_INSERT_ID()))";
65
66         try {
67             conn = c.conectar();
68             conn.setAutoCommit(false);
69
70             stmtEnd = conn.prepareStatement(queryEnd);
71             stmtFornecedor = conn.prepareStatement(queryFornecedor);
72
73             stmtEnd.setString(1, end.getCEP());
74             stmtEnd.setString(2, end.getEstado());
75             stmtEnd.setString(3, end.getCidade());
76             stmtEnd.setString(4, end.getBairro());
77             stmtEnd.setString(5, end.getRua());
78             stmtEnd.setInt(6, end.getNumero());
79             stmtEnd.setString(7, end.getComplemento());
80             stmtEnd.setString(8, Character.toString(end.getTipo()));
81
82             stmtFornecedor.setString(1, fornecedor.getCnpj());
83             stmtFornecedor.setString(2, fornecedor.getRazaoSocial());
84             stmtFornecedor.setString(3, fornecedor.getEmail());
85             stmtFornecedor.setString(4, fornecedor.getTelefone1());
86             stmtFornecedor.setString(5, fornecedor.getTelefone2());
87             stmtFornecedor.setString(6, fornecedor.getDescricao());
88
89             stmtEnd.execute();
90             stmtFornecedor.execute();
91
92             conn.commit();
93
94             conn.close();
95             return true;
96         } catch (Exception e) {
97             e.printStackTrace();

```

```

98         return false;
99     }
100 }
101
102 public List<Fornecedor> getAllFornecedor() throws Exception {
103     String query = "SELECT * FROM Fornecedor";
104     List<Fornecedor> retorno = new ArrayList<>();
105     Utilidades u = new Utilidades();
106     ConexaoMySQL c = new ConexaoMySQL();
107     Connection conn = null;
108     conn = c.conectar();
109     try {
110         PreparedStatement stmt = conn.prepareStatement(query);
111         ResultSet resultado = stmt.executeQuery();
112         while (resultado.next()) {
113             Fornecedor fornecedor = new Fornecedor();
114             fornecedor.setCnpj(resultado.getString("cnpj"));
115             fornecedor.setRazaoSocial(resultado.getString("razaoSocial"));
116             fornecedor.setEmail(resultado.getString("email"));
117             fornecedor.setTelefone1(resultado.getString("telefone1"));
118             fornecedor.setTelefone2(resultado.getString("telefone2"));
119             fornecedor.setDescricao(resultado.getString("descricao"));
120
121             retorno.add(fornecedor);
122         }
123     } catch (Exception e) {
124         e.printStackTrace();
125     }
126     conn.close();
127     return retorno;
128 }
129 }

```

Código 7: FornecedorDAO.java

5.2.2 View

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.chart.*?>
4  <?import javafx.scene.text.*?>
5  <?import java.lang.*?>
6  <?import java.util.*?>
7  <?import javafx.scene.*?>
8  <?import javafx.scene.control.*?>
9  <?import javafx.scene.layout.*?>
10 <?import javafx.collections.*?>
11
12 <AnchorPane id="AnchorPane" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
13     fx:controller="opencarshop.fornecedor.controller.FornecedorController">
14     <children>
15         <TabPane prefHeight="328.0" prefWidth="1024.0" tabClosingPolicy="UNAVAILABLE">
16             <tabs>
17                 <Tab text="Identificação">
18                     <content>
19                         <AnchorPane minHeight="0.0" minWidth="0.0" prefWidth="1024.0">
20                             <children>
21                                 <TextField fx:id="tf_cnpjCadastro" layoutX="14.0" layoutY="87.0" prefHeight="25.0"
22                                     prefWidth="170.0" promptText="CNPJ" />
23                                 <TextField fx:id="tf_descricaoCadastro" layoutX="15.0" layoutY="142.0" prefHeight="73.0
24                                     prefWidth="170.0" promptText="Descrição" />
25                                 <TextField fx:id="tf_razaoCadastro" layoutX="14.0" layoutY="31.0" prefHeight="25.0"
26                                     prefWidth="350.0" promptText="Razão Social" />
27                                 <Label layoutX="14.0" layoutY="6.0" text="Razão Social" />
28                                 <Label layoutX="15.0" layoutY="70.0" text="CNPJ" />
29                                 <Label layoutX="15.0" layoutY="122.0" text="Descrição" />
30                             </children>
31                         </AnchorPane>
32                     </content>
33                 </Tab>
34                 <Tab text="Contato">
35                     <content>
36                         <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="316.0" prefWidth="1024.0">
37                             <children>
38                                 <TextField fx:id="tf_emailCadastro" layoutX="14.0" layoutY="14.0" prefHeight="25.0"
39                                     prefWidth="350.0" promptText="Email" />
40                                 <TextField fx:id="tf_telefone1Cadastro" layoutX="14.0" layoutY="50.0" prefWidth="170.0"
41                                     promptText="Telefone 1" />
42                                 <TextField fx:id="tf_telefone2Cadastro" layoutX="194.0" layoutY="50.0" prefWidth="170.0"
43                                     promptText="Telefone 2" />
44                                 <TextField fx:id="tf_ruaCadastro" layoutX="14.0" layoutY="116.0" prefWidth="350.0"
45                                     promptText="Rua" />
46                                 <TextField fx:id="tf_cidadeCadastro" layoutX="14.0" layoutY="187.0" prefWidth="350.0"
47                                     promptText="Cidade" />
48                                 <TextField fx:id="tf_estadoCadastro" layoutX="14.0" layoutY="222.0" prefWidth="170.0"
49                                     promptText="UF" />
50                                 <TextField fx:id="tf_bairroCadastro" layoutX="194.0" layoutY="222.0" prefWidth="170.0"
51                                     promptText="Bairro" />

```



```

41         <TextField fx:id="tf_cepCadastro" layoutX="14.0" layoutY="256.0" prefWidth="170.0"
promptText="CEP" />
42         <TextField fx:id="tf_numeroCadastro" layoutX="194.0" layoutY="256.0" prefWidth="170.0"
promptText="Número" />
43         <TextField fx:id="tf_complementoCadastro" layoutX="14.0" layoutY="151.0" prefWidth="
350.0" promptText="Complemento" />
44         <ComboBox fx:id="cb_tipoCadastro" layoutX="14.0" layoutY="83.0" prefWidth="350.0"
promptText="Tipo de Endereço">
45             <items>
46                 <FXCollections fx:factory="observableArrayList">
47                     <String fx:value="Residencial" />
48                     <String fx:value="Comercial" />
49                 </FXCollections>
50             </items>
51         </ComboBox>
52     </children>
53 </AnchorPane>
54 </content>
55 </Tab>
56 </tabs>
57 </TabPane>
58 <Button id="btn_cadastrar" layoutX="151.0" layoutY="367.0" mnemonicParsing="false" onAction="#cadastrar" text="
Cadastrar" />
59 <Label fx:id="resultadoCadastro" alignment="CENTER" contentDisplay="CENTER" layoutX="16.0" layoutY="328.0"
prefHeight="17.0" prefWidth="346.0" textAlignment="CENTER" />
60 </children>
61 </AnchorPane>

```

Código 8: Cadastrar.fxml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.chart.*?>
4  <?import javafx.scene.text.*?>
5  <?import java.lang.*?>
6  <?import java.util.*?>
7  <?import javafx.scene.*?>
8  <?import javafx.scene.control.*?>
9  <?import javafx.scene.layout.*?>
10 <?import javafx.collections.*?>
11
12 <AnchorPane id="AnchorPane" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
javafx.com/fxml/1" fx:controller="opencarshop.fornecedor.controller.FornecedorController">
13     <children>
14         <Button layoutX="504.0" layoutY="159.0" mnemonicParsing="false" text="Cadastrar" />
15         <Button layoutX="506.0" layoutY="210.0" mnemonicParsing="false" text="Atualizar" />
16         <Button layoutX="510.0" layoutY="261.0" mnemonicParsing="false" text="Inativar" />
17         <TableView prefHeight="343.0" prefWidth="448.0">
18             <columns>
19                 <TableColumn prefWidth="75.0" text="CNPJ" />
20                 <TableColumn prefWidth="75.0" text="Razão Social" />
21                 <TableColumn prefWidth="75.0" text="Email" />
22                 <TableColumn prefWidth="75.0" text="Telefone" />
23                 <TableColumn prefWidth="75.0" text="Telefone 2" />
24                 <TableColumn prefWidth="75.0" text="Descrição" />
25             </columns>
26         </TableView>
27     </children>
28 </AnchorPane>

```

Código 9: Buscar.fxml

5.2.3 Controller

```

1 package opencarshop.fornecedor.controller;
2
3 import java.net.URL;
4 import java.text.ParseException;
5 import java.util.List;
6 import java.util.ResourceBundle;
7 import javafx.collections.FXCollections;
8 import javafx.collections.ObservableList;
9 import javafx.event.ActionEvent;
10 import javafx.fxml.FXML;
11 import javafx.fxml.Initializable;
12 import javafx.scene.control.ComboBox;
13 import javafx.scene.control.Hyperlink;
14 import javafx.scene.control.Label;
15 import javafx.scene.control.TableColumn;
16 import javafx.scene.control.TableView;
17 import javafx.scene.control.TextField;
18 import javafx.scene.control.cell.PropertyValueFactory;
19 import opencarshop.Endereco;
20 import opencarshop.fornecedor.model.Fornecedor;
21 import opencarshop.fornecedor.model.FornecedorDAO;
22 import opencarshop.util.Utilidades;

```

```

23
24 public class FornecedorController implements Initializable {
25
26     /**
27      * Initializes the controller class.
28      */
29     // TELA DE AUTENTICACAO
30     @FXML
31     private Label labelErro;
32     @FXML
33     private TextField tf_cpf;
34     @FXML
35     private Hyperlink cadastroLink;
36
37     // TELA DE CADASTRO
38     @FXML
39     private TextField tf_cnpjCadastro;
40     @FXML
41     private TextField tf_razaoCadastro;
42     @FXML
43     private TextField tf_descricaoCadastro;
44
45     @FXML
46     private ComboBox<String> cb_tipoCadastro;
47     @FXML
48     private TextField tf_emailCadastro;
49     @FXML
50     private TextField tf_telefone1Cadastro;
51     @FXML
52     private TextField tf_telefone2Cadastro;
53     @FXML
54     private TextField tf_ruaCadastro;
55     @FXML
56     private TextField tf_cidadeCadastro;
57     @FXML
58     private TextField tf_estadoCadastro;
59     @FXML
60     private TextField tf_bairroCadastro;
61     @FXML
62     private TextField tf_cepCadastro;
63     @FXML
64     private TextField tf_numeroCadastro;
65     @FXML
66     private TextField tf_complementoCadastro;
67
68     @FXML
69     private Label resultadoCadastro;
70
71     // TABELA Fornecedor
72     @FXML
73     private TableColumn<Fornecedor, String> col_cnpj;
74     @FXML
75     private TableColumn<Fornecedor, String> col_razaoSocial;
76     @FXML
77     private TableColumn<Fornecedor, String> col_email;
78     @FXML
79     private TableColumn<Fornecedor, String> col_telefone1;
80     @FXML
81     private TableColumn<Fornecedor, String> col_telefone2;
82     @FXML
83     private TableColumn<Fornecedor, String> col_descricao;
84     @FXML
85     private TableView<Fornecedor> tbl_fornecedor;
86
87     /**
88     @FXML
89     private void autenticar(ActionEvent event)
90     {
91         Funcionario funcionario;
92         FuncionarioDAO func = new FuncionarioDAO();
93         funcionario = func.getFuncionario(tf_cpf.getText());
94
95         if(funcionario.getCpf() != null)
96         {
97             if(funcionario.getSenha().equals(pf_senha.getText()))
98             {
99                 Parent root = null;
100                 try
101                 {
102                     root = FXMLLoader.load(getClass().getResource("/opencarshop/TelaPrincipal.fxml"));
103                     Scene scene = new Scene(root);
104                     Stage nStage = new Stage();
105                     nStage.setScene(scene);
106                     //nStage.setMaximized(true);
107                     nStage.setMaxHeight(768);
108                     nStage.setMaxWidth(1024);
109                     nStage.setTitle("OpenCarShop");
110                     nStage.setResizable(false);
111                     nStage.show();
112                     Stage stage = (Stage) cadastroLink.getScene().getWindow();
113                     stage.close();
114                 }
115                 catch (IOException e)
116                 {

```

```

117         e.printStackTrace();
118     }
119 }
120 else
121 {
122     labelErro.setText("Login ou senha errado!!!");
123 }
124 }
125 else
126 {
127     labelErro.setText("Login ou senha errado!!!");
128 }
129 }
130 */
131 @FXML
132 private void cadastrar(ActionEvent event) throws ParseException {
133
134     Fornecedor fornecedor = new Fornecedor();
135     Endereco end = new Endereco();
136
137     FornecedorDAO f = new FornecedorDAO();
138     Utilidades u = new Utilidades();
139
140     // OBJETO FORNECEDOR
141     fornecedor.setCnpj(tf_cnpjCadastro.getText());
142     fornecedor.setRazaoSocial(tf_razaoCadastro.getText());
143     fornecedor.setEmail(tf_emailCadastro.getText());
144     fornecedor.setTelefone1(tf_telefone1Cadastro.getText());
145     fornecedor.setTelefone2(tf_telefone2Cadastro.getText());
146     fornecedor.setDescricao(tf_descricaoCadastro.getText());
147
148     // OBJETO ENDEREÇO
149     end.setCEP(tf_cepCadastro.getText());
150     end.setEstado(tf_estadoCadastro.getText());
151     end.setCidade(tf_cidadeCadastro.getText());
152     end.setBairro(tf_bairroCadastro.getText());
153     end.setRua(tf_ruaCadastro.getText());
154     end.setNumero(Integer.parseInt(tf_numeroCadastro.getText()));
155     end.setComplemento(tf_complementoCadastro.getText());
156     end.setTipo(cb_tipoCadastro.getValue().charAt(0));
157
158     if (f.cadastraFornecedor(fornecedor, end)) {
159         resultadoCadastro.setText("Cadastrado com sucesso!!");
160     } else {
161         resultadoCadastro.setText("Erro ao cadastrar!! Tente novamente.");
162     }
163 }
164
165 @FXML
166 private void buscar(ActionEvent event) throws Exception {
167
168 }
169
170 private void carregaTabelaFornecedor() throws Exception {
171     col_cnpj.setCellValueFactory(new PropertyValueFactory<>("cnpj"));
172     col_razaoSocial.setCellValueFactory(new PropertyValueFactory<>("razaoSocial"));
173     col_email.setCellValueFactory(new PropertyValueFactory<>("email"));
174     col_telefone1.setCellValueFactory(new PropertyValueFactory<>("telefone1"));
175     col_telefone2.setCellValueFactory(new PropertyValueFactory<>("telefone2"));
176     col_descricao.setCellValueFactory(new PropertyValueFactory<>("descricao"));
177     FornecedorDAO f = new FornecedorDAO();
178     List<Fornecedor> listaFornecedor = f.getAllFornecedor();
179     ObservableList<Fornecedor> observableListFornecedor;
180
181     observableListFornecedor = FXCollections.observableArrayList(listaFornecedor);
182     tbl_fornecedor.setItems(observableListFornecedor);
183 }
184
185 @Override
186 public void initialize(URL url, ResourceBundle rb) {
187     try {
188         carregaTabelaFornecedor();
189     } catch (Exception ex) {
190         //Logger.getLogger(FuncionarioController.class.getName()).log(Level.SEVERE, null, ex);
191     }
192 }
193
194 }
195 }

```

Código 10: FornecedorController.java

5.3 Pacote Funcionário

5.3.1 Model

```

1 package opencarshop.funcionario.model;
2
3 import java.time.LocalDate;
4 import java.util.Date;
5
6 public class Contrato {
7
8     private Character cargo;
9     private Double salario;
10    private LocalDate dataInicio;
11    private LocalDate dataTermino;
12
13    public Character getCargo() {
14        return cargo;
15    }
16
17    public void setCargo(Character cargo) {
18        this.cargo = cargo;
19    }
20
21    public Double getSalario() {
22        return salario;
23    }
24
25    public void setSalario(Double salario) {
26        this.salario = salario;
27    }
28
29    public LocalDate getDataInicio() {
30        return dataInicio;
31    }
32
33    public void setDataInicio(LocalDate dataInicio) {
34        this.dataInicio = dataInicio;
35    }
36
37    public LocalDate getDataTermino() {
38        return dataTermino;
39    }
40
41    public void setDataTermino(LocalDate dataTermino) {
42        this.dataTermino = dataTermino;
43    }
44
45 }

```

Código 11: Contrato.java

```

1 package opencarshop.funcionario.model;
2
3 import java.time.LocalDate;
4
5 public class Funcionario {
6
7     private String cpf;
8     private String nome;
9     private String senha;
10    private LocalDate dataNascimento;
11    private String email;
12    private String telefone1;
13    private String telefone2;
14    private Boolean ativo;
15
16    public Boolean getAtivo() {
17        return ativo;
18    }
19
20    public void setAtivo(Boolean ativo) {
21        this.ativo = ativo;
22    }
23
24    public String getCpf() {
25        return cpf;
26    }
27
28    public void setCpf(String cpf) {
29        this.cpf = cpf;
30    }
31
32    public String getNome() {
33        return nome;
34    }
35
36    public void setNome(String nome) {
37        this.nome = nome;
38    }
39
40    public String getSenha() {
41        return senha;
42    }
43
44    public void setSenha(String senha) {

```

```

45     this.senha = senha;
46 }
47
48 public LocalDate getDataNascimento() {
49     return dataNascimento;
50 }
51
52 public void setDataNascimento(LocalDate dataNascimento) {
53     this.dataNascimento = dataNascimento;
54 }
55
56 public String getEmail() {
57     return email;
58 }
59
60 public void setEmail(String email) {
61     this.email = email;
62 }
63
64 public String getTelefone1() {
65     return telefone1;
66 }
67
68 public void setTelefone1(String telefone1) {
69     this.telefone1 = telefone1;
70 }
71
72 public String getTelefone2() {
73     return telefone2;
74 }
75
76 public void setTelefone2(String telefone2) {
77     this.telefone2 = telefone2;
78 }
79 }

```

Código 12: Funcionario.java

```

1 package opencarshop.funcionario.model;
2
3 import java.sql.Connection;
4 import java.sql.Date;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.util.ArrayList;
9 import java.util.List;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12 import opencarshop.util.ConexaoMySQL;
13 import opencarshop.Endereco;
14 import opencarshop.util.Utilidades;
15
16 public class FuncionarioDAO {
17
18     public Funcionario getFuncionario(String cpf) {
19         ConexaoMySQL c = new ConexaoMySQL();
20         Connection conn = null;
21         PreparedStatement stmt = null;
22         Utilidades u = new Utilidades();
23
24         String query = "SELECT * FROM Funcionario WHERE cpf=?";
25
26         Funcionario funcionario = new Funcionario();
27         try {
28             conn = c.conectar();
29             stmt = conn.prepareStatement(query);
30             stmt.setString(1, cpf);
31
32             ResultSet resultado = stmt.executeQuery();
33
34             if (resultado.next()) {
35                 funcionario.setCpf(resultado.getString("cpf"));
36                 funcionario.setNome(resultado.getString("nome"));
37                 funcionario.setSenha(resultado.getString("senha"));
38                 funcionario.setDataNascimento(u.toLocalDate(resultado.getDate("dataNascimento")));
39                 funcionario.setEmail(resultado.getString("email"));
40                 funcionario.setTelefone1(resultado.getString("telefone1"));
41                 funcionario.setTelefone2(resultado.getString("telefone2"));
42                 funcionario.setAtivo(resultado.getBoolean("ativo"));
43             }
44
45             conn.close();
46         } catch (Exception e) {
47             e.printStackTrace();
48         }
49         return funcionario;
50     }
51
52     public boolean cadastraFuncionario(Funcionario func, Endereco end, Contrato cont) {
53         ConexaoMySQL c = new ConexaoMySQL();
54         Connection conn = null;

```

```

55
56     PreparedStatement stmtEnd = null;
57     PreparedStatement stmtCon = null;
58     PreparedStatement stmtFun = null;
59
60     String queryEnd = "INSERT INTO Endereco(cep,estado,cidade,bairro,rua,numero,complemento,tipo) VALUES
61     (?,?,?,?,?,?,?,?)";
62     String queryFun = "INSERT INTO Funcionario(cpf,nome,senha,dataNascimento,email,telefone1,telefone2,
63     endereco,ativo) VALUES(?,?,?,?,?,(select LAST_INSERT_ID()),?)";
64     String queryCon = "INSERT INTO Contrato(cargo,salario,dataInicio,dataTermino,funcionario) VALUES
65     (?,?,?,?,?)";
66
67     try {
68         conn = c.conectar();
69         conn.setAutoCommit(false);
70
71         stmtEnd = conn.prepareStatement(queryEnd);
72         stmtFun = conn.prepareStatement(queryFun);
73         stmtCon = conn.prepareStatement(queryCon);
74
75         stmtEnd.setString(1, end.getCep());
76         stmtEnd.setString(2, end.getEstado());
77         stmtEnd.setString(3, end.getCidade());
78         stmtEnd.setString(4, end.getBairro());
79         stmtEnd.setString(5, end.getRua());
80         stmtEnd.setInt(6, end.getNumero());
81         stmtEnd.setString(7, end.getComplemento());
82         stmtEnd.setString(8, Character.toString(end.getTipo()));
83
84         stmtFun.setString(1, func.getCpf());
85         stmtFun.setString(2, func.getNome());
86         stmtFun.setString(3, func.getSenha());
87         stmtFun.setDate(4, Date.valueOf(func.getDataNascimento()));
88         stmtFun.setString(5, func.getEmail());
89         stmtFun.setString(6, func.getTelefone1());
90         stmtFun.setString(7, func.getTelefone2());
91         stmtFun.setBoolean(8, true);
92
93         stmtCon.setString(1, Character.toString(cont.getCargo()));
94         stmtCon.setDouble(2, cont.getSalario());
95         stmtCon.setDate(3, Date.valueOf(cont.getDataInicio()));
96         stmtCon.setDate(4, Date.valueOf(cont.getDataInicio()));
97         stmtCon.setString(5, func.getCpf());
98
99         stmtEnd.execute();
100        stmtFun.execute();
101        stmtCon.execute();
102
103        conn.commit();
104
105        conn.close();
106        return true;
107    } catch (Exception e) {
108        e.printStackTrace();
109        return false;
110    }
111
112    public List<Funcionario> getAllFuncionario() throws Exception {
113        String query = "SELECT * FROM Funcionario";
114        List<Funcionario> retorno = new ArrayList<>();
115        Utilidades u = new Utilidades();
116        ConexaoMySQL c = new ConexaoMySQL();
117        Connection conn = null;
118        conn = c.conectar();
119
120        try {
121            PreparedStatement stmt = conn.prepareStatement(query);
122            ResultSet resultado = stmt.executeQuery();
123            while (resultado.next()) {
124                Funcionario funcionario = new Funcionario();
125                funcionario.setCpf(resultado.getString("cpf"));
126                funcionario.setNome(resultado.getString("nome"));
127                funcionario.setSenha(resultado.getString("senha"));
128                funcionario.setDataNascimento(u.toLocalDate(resultado.getDate("dataNascimento")));
129                funcionario.setEmail(resultado.getString("email"));
130                funcionario.setTelefone1(resultado.getString("telefone1"));
131                funcionario.setTelefone2(resultado.getString("telefone2"));
132                funcionario.setAtivo(resultado.getBoolean("ativo"));
133
134                retorno.add(funcionario);
135            }
136        } catch (Exception e) {
137            e.printStackTrace();
138        }
139        conn.close();
140        return retorno;
141    }
142
143    public Boolean alteraFuncionario(Funcionario func) throws SQLException {
144        String query = "UPDATE Funcionario SET nome=?, senha=?, dataNascimento=?, email=?, telefone1=?, telefone2=?,
145        ativo=? WHERE cpf=?";
146
147        ConexaoMySQL c = new ConexaoMySQL();
148        Connection conn = null;

```

```

145     try {
146         conn = c.conectar();
147         PreparedStatement stmt = conn.prepareStatement(query);
148
149         stmt.setString(1, func.getNome());
150         stmt.setString(2, func.getSenha());
151         stmt.setDate(3, Date.valueOf(func.getDataNascimento()));
152         stmt.setString(4, func.getEmail());
153         stmt.setString(5, func.getTelefone1());
154         stmt.setString(6, func.getTelefone2());
155         stmt.setBoolean(7, func.getAtivo());
156         stmt.setString(8, func.getCpf());
157         stmt.execute();
158         conn.close();
159         return true;
160     } catch (Exception ex) {
161
162         Logger.getLogger(FuncionarioDAO.class.getName()).log(Level.SEVERE, null, ex);
163         return false;
164     }
165 }
166 }

```

Código 13: FuncionarioDAO.java

5.3.2 View

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.chart.*?>
4  <?import javafx.scene.text.*?>
5  <?import java.lang.*?>
6  <?import java.util.*?>
7  <?import javafx.scene.*?>
8  <?import javafx.scene.control.*?>
9  <?import javafx.scene.layout.*?>
10 <?import javafx.collections.*?>
11
12 <AnchorPane id="AnchorPane" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
13   fx:controller="opencarshop.fornecedor.controller.FornecedorController">
14   <children>
15     <TabPane prefHeight="328.0" prefWidth="1024.0" tabClosingPolicy="UNAVAILABLE">
16       <tabs>
17         <Tab text="Identifica o">
18           <content>
19             <AnchorPane minHeight="0.0" minWidth="0.0" prefWidth="1024.0">
20               <children>
21                 <TextField fx:id="tf_cnpjCadastro" layoutX="14.0" layoutY="87.0" prefHeight="25.0"
22                   prefWidth="170.0" promptText="CNPJ" />
23                 <TextField fx:id="tf_descricaoCadastro" layoutX="15.0" layoutY="142.0" prefHeight="73.0"
24                   prefWidth="170.0" promptText="Descri o" />
25                 <TextField fx:id="tf_razaoCadastro" layoutX="14.0" layoutY="31.0" prefHeight="25.0"
26                   prefWidth="350.0" promptText="Raz o Social" />
27                 <Label layoutX="14.0" layoutY="6.0" text="Raz o Social" />
28                 <Label layoutX="15.0" layoutY="70.0" text="CNPJ" />
29                 <Label layoutX="15.0" layoutY="122.0" text="Descri o" />
30               </children>
31             </AnchorPane>
32           </content>
33         </Tab>
34         <Tab text="Contato">
35           <content>
36             <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="316.0" prefWidth="1024.0">
37               <children>
38                 <TextField fx:id="tf_emailCadastro" layoutX="14.0" layoutY="14.0" prefHeight="25.0"
39                   prefWidth="350.0" promptText="Email" />
40                 <TextField fx:id="tf_telefone1Cadastro" layoutX="14.0" layoutY="50.0" prefWidth="170.0"
41                   promptText="Telefone 1" />
42                 <TextField fx:id="tf_telefone2Cadastro" layoutX="194.0" layoutY="50.0" prefWidth="170.0"
43                   promptText="Telefone 2" />
44                 <TextField fx:id="tf_ruacadastro" layoutX="14.0" layoutY="116.0" prefWidth="350.0"
45                   promptText="Rua" />
46                 <TextField fx:id="tf_cidadeCadastro" layoutX="14.0" layoutY="187.0" prefWidth="350.0"
47                   promptText="Cidade" />
48                 <TextField fx:id="tf_estadoCadastro" layoutX="14.0" layoutY="222.0" prefWidth="170.0"
49                   promptText="UF" />
50                 <TextField fx:id="tf_bairroCadastro" layoutX="194.0" layoutY="222.0" prefWidth="170.0"
51                   promptText="Bairro" />
52                 <TextField fx:id="tf_cepCadastro" layoutX="14.0" layoutY="256.0" prefWidth="170.0"
53                   promptText="CEP" />
54                 <TextField fx:id="tf_numeroCadastro" layoutX="194.0" layoutY="256.0" prefWidth="170.0"
55                   promptText="Número" />
56                 <TextField fx:id="tf_complementoCadastro" layoutX="14.0" layoutY="151.0" prefWidth="350.0"
57                   promptText="Complemento" />
58                 <ComboBox fx:id="cb_tipoCadastro" layoutX="14.0" layoutY="83.0" prefWidth="350.0"
59                   promptText="Tipo de Endereço">
60                   <items>
61                     <FXCollections fx:factory="observableArrayList">

```

```

47         <String fx:value="Residencial" />
48         <String fx:value="Comercial" />
49     </FXCollections>
50 </items>
51 </ComboBox>
52 </children>
53 </AnchorPane>
54 </content>
55 </Tab>
56 </tabs>
57 </TabPane>
58 <Button id="btn_cadastrar" layoutX="151.0" layoutY="367.0" mnemonicParsing="false" onAction="#cadastrar" text="
Cadastrar" />
59 <Label fx:id="resultadoCadastro" alignment="CENTER" contentDisplay="CENTER" layoutX="16.0" layoutY="328.0"
prefHeight="17.0" prefWidth="346.0" textAlignment="CENTER" />
60 </children>
61 </AnchorPane>

```

Código 14: Cadastrar.fxml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import java.lang.*?>
4  <?import java.util.*?>
5  <?import javafx.scene.*?>
6  <?import javafx.scene.control.*?>
7  <?import javafx.scene.layout.*?>
8
9  <AnchorPane id="AnchorPane" prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
javafx.com/fxml/1" fx:controller="opencarshop.funcionario.controller.FuncionarioController">
10    <children>
11      <TableView fx:id="tbl_funcionario" prefHeight="650.0" prefWidth="650.0">
12        <columns>
13          <TableColumn fx:id="col_cpf" prefWidth="100.0" text="CPF" />
14          <TableColumn fx:id="col_nome" prefWidth="155.0" text="Nome" />
15          <TableColumn id="col_telefone" fx:id="col_telefone1" minWidth="0.0" prefWidth="103.0" text="Telefone_1" />
16          <TableColumn id="col_telefone" fx:id="col_telefone2" prefWidth="98.0" text="Telefone_2" />
17          <TableColumn id="col_email" fx:id="col_email" prefWidth="193.0" text="Email" />
18        </columns>
19      </TableView>
20      <TextField fx:id="tf_nomeCadastro" layoutX="846.0" layoutY="95.0" prefWidth="150.0" promptText="Nome" />
21      <TextField fx:id="tf_emailCadastro" layoutX="846.0" layoutY="151.0" promptText="Email" />
22      <TextField fx:id="tf_telefone1Cadastro" layoutX="675.0" layoutY="208.0" promptText="Telefone_1" />
23      <TextField fx:id="tf_telefone2Cadastro" layoutX="846.0" layoutY="208.0" promptText="Telefone_2" />
24      <DatePicker fx:id="dp_dataNascimentoCadastro" layoutX="675.0" layoutY="262.0" promptText="Data_de_nascimento" />
25      <Label layoutX="844.0" layoutY="78.0" text="Nome:" />
26      <Label layoutX="675.0" layoutY="134.0" text="Senha:" />
27      <PasswordField fx:id="pf_senhaCadastro" layoutX="675.0" layoutY="151.0" promptText="Senha" />
28      <Label layoutX="846.0" layoutY="134.0" text="Email:" />
29      <Label layoutX="678.0" layoutY="191.0" text="Telefone_1:" />
30      <Label layoutX="847.0" layoutY="191.0" text="Telefone_2:" />
31      <Label layoutX="678.0" layoutY="245.0" text="Data_de_nascimento:" />
32      <CheckBox fx:id="cb_ativo" layoutX="903.0" layoutY="266.0" mnemonicParsing="false" text="Ativo" />
33      <Button fx:id="btn_alterarCadastro" layoutX="782.0" layoutY="359.0" mnemonicParsing="false" onAction="#
alterarCadastro" text="Salvar Altera es" />
34      <TextField fx:id="tf_cpfCadastro" editable="false" layoutX="675.0" layoutY="95.0" />
35      <Label layoutX="675.0" layoutY="78.0" text="CPF:" />
36      <Label fx:id="confirmaAlteracao" alignment="CENTER" contentDisplay="CENTER" layoutX="736.0" layoutY="325.0"
prefHeight="17.0" prefWidth="199.0" textAlignment="CENTER" />
37    </children>
38  </AnchorPane>

```

Código 15: Buscar.fxml

5.3.3 Controller

```

1  package opencarshop.funcionario.controller;
2
3  import java.io.IOException;
4  import java.net.URL;
5  import java.text.DecimalFormat;
6  import java.text.ParseException;
7  import java.util.List;
8  import java.util.ResourceBundle;
9  import java.util.logging.Level;
10 import java.util.logging.Logger;
11 import javafx.collections.FXCollections;
12 import javafx.collections.ObservableList;
13 import javafx.event.ActionEvent;
14 import javafx.fxml.FXML;
15 import javafx.fxml.FXMLLoader;
16 import javafx.fxml.Initializable;
17 import javafx.scene.Parent;
18 import javafx.scene.Scene;

```



```

19 import javafx.scene.control.CheckBox;
20 import javafx.scene.control.ComboBox;
21 import javafx.scene.control.DatePicker;
22 import javafx.scene.control.Hyperlink;
23 import javafx.scene.control.Label;
24 import javafx.scene.control.PasswordField;
25 import javafx.scene.control.TableColumn;
26 import javafx.scene.control.TableView;
27 import javafx.scene.control.TextField;
28 import javafx.scene.control.cell.PropertyValueFactory;
29 import javafx.stage.Stage;
30 import opencarshop.funcionario.model.Contrato;
31 import opencarshop.funcionario.model.Funcionario;
32 import opencarshop.funcionario.model.FuncionarioDAO;
33 import opencarshop.Endereco;
34 import opencarshop.util.Utilidades;
35
36 public class FuncionarioController implements Initializable {
37
38     /**
39      * Initializes the controller class.
40      */
41     // TELA DE AUTENTICACAO
42     @FXML
43     private Label labelErro;
44     @FXML
45     private TextField tf_cpf;
46     @FXML
47     private PasswordField pf_senha;
48     @FXML
49     private Hyperlink cadastroLink;
50
51     // TELA DE CADASTRO
52     @FXML
53     private TextField tf_cpfCadastro;
54     @FXML
55     private PasswordField pf_senhaCadastro;
56     @FXML
57     private TextField tf_nomeCadastro;
58     @FXML
59     private DatePicker dp_dataNascimentoCadastro;
60
61     @FXML
62     private ComboBox<String> cb_tipoCadastro;
63     @FXML
64     private TextField tf_emailCadastro;
65     @FXML
66     private TextField tf_telefone1Cadastro;
67     @FXML
68     private TextField tf_telefone2Cadastro;
69     @FXML
70     private TextField tf_ruaCadastro;
71     @FXML
72     private TextField tf_cidadeCadastro;
73     @FXML
74     private TextField tf_estadoCadastro;
75     @FXML
76     private TextField tf_bairroCadastro;
77     @FXML
78     private TextField tf_cepCadastro;
79     @FXML
80     private TextField tf_numeroCadastro;
81     @FXML
82     private TextField tf_complementoCadastro;
83
84     @FXML
85     private TextField tf_salarioCadastro;
86     @FXML
87     private ComboBox<String> cb_cargoCadastro;
88     @FXML
89     private DatePicker dp_dataInicioCadastro;
90     @FXML
91     private DatePicker dp_dataTerminoCadastro;
92
93     @FXML
94     private Label resultadoCadastro;
95
96     // TABELA FUNCIONARIO
97     @FXML
98     private TableColumn<Funcionario, String> col_nome;
99     @FXML
100    private TableColumn<Funcionario, String> col_cpf;
101    @FXML
102    private TableColumn<Funcionario, String> col_telefone1;
103    @FXML
104    private TableColumn<Funcionario, String> col_telefone2;
105    @FXML
106    private TableColumn<Funcionario, String> col_email;
107
108    @FXML
109    private TableView<Funcionario> tbl_funcionario;
110
111    @FXML
112    private Label label_nome;

```

```

113
114 @FXML
115 private CheckBox cb_ativo;
116
117 @FXML
118 private Label confirmaAlteracao;
119
120 @FXML
121 private void autenticar(ActionEvent event) {
122     Funcionario funcionario;
123     FuncionarioDAO func = new FuncionarioDAO();
124     funcionario = func.getFuncionario(tf_cpf.getText());
125
126     if (funcionario.getCpf() != null) {
127         if (funcionario.getSenha().equals(pf_senha.getText())) {
128             Parent root = null;
129             try {
130                 root = FXMLLoader.load(getClass().getResource("/opencarshop/TelaPrincipal.fxml"));
131                 Scene scene = new Scene(root);
132                 Stage nStage = new Stage();
133                 nStage.setScene(scene);
134                 //nStage.setMaximized(true);
135                 nStage.setMaxHeight(768);
136                 nStage.setMaxWidth(1024);
137                 nStage.setTitle("OpenCarShop");
138                 nStage.setResizable(false);
139                 nStage.show();
140                 Stage stage = (Stage) cadastroLink.getScene().getWindow();
141                 stage.close();
142             } catch (IOException e) {
143                 e.printStackTrace();
144             }
145         } else {
146             labelErro.setText("Login ou senha errado!!!");
147         }
148     } else {
149         labelErro.setText("Login ou senha errado!!!");
150     }
151 }
152
153 @FXML
154 private void cadastrar(ActionEvent event) throws ParseException {
155     //cb_cargoCadaastro.setItems(cargos);
156     Funcionario func = new Funcionario();
157     Endereco end = new Endereco();
158     Contrato contr = new Contrato();
159     FuncionarioDAO f = new FuncionarioDAO();
160     Utilidades u = new Utilidades();
161
162     // OBJETO FUNCIONARIO
163     func.setCpf(tf_cpfCadaastro.getText());
164     func.setNome(tf_nomeCadaastro.getText());
165     func.setSenha(pf_senhaCadaastro.getText());
166     func.setDataNascimento(dp_dataNascimentoCadaastro.getValue());
167     func.setEmail(tf_emailCadaastro.getText());
168     func.setTelefone1(tf_telefone1Cadaastro.getText());
169     func.setTelefone2(tf_telefone2Cadaastro.getText());
170     func.setAtivo(true);
171
172     // OBJETO ENDEREÇO
173     end.setCEP(tf_cepCadaastro.getText());
174     end.setEstado(tf_estadoCadaastro.getText());
175     end.setCidade(tf_cidadeCadaastro.getText());
176     end.setBairro(tf_bairroCadaastro.getText());
177     end.setRua(tf_ruaCadaastro.getText());
178     end.setNumero(Integer.parseInt(tf_numeroCadaastro.getText()));
179     end.setComplemento(tf_complementoCadaastro.getText());
180     end.setTipo(cb_tipoCadaastro.getValue().charAt(0));
181
182     // OBJETO CONTRATO
183     contr.setCargo(cb_cargoCadaastro.getValue().charAt(0));
184     contr.setSalario(DecimalFormat.getInstance().parse(tf_salarioCadaastro.getText()).doubleValue());
185     contr.setDataInicio(dp_dataInicioCadaastro.getValue());
186     contr.setDataTermino(dp_dataTerminoCadaastro.getValue());
187
188     if (f.cadastraFuncionario(func, end, contr)) {
189         resultadoCadaastro.setText("Cadastrado com sucesso!!!");
190     } else {
191         resultadoCadaastro.setText("Erro ao cadastrar!! Tente novamente.");
192     }
193 }
194
195 @FXML
196 private void alterarCadaastro(ActionEvent event) throws Exception {
197     Funcionario func = new Funcionario();
198     func.setCpf(tf_cpfCadaastro.getText());
199     func.setNome(tf_nomeCadaastro.getText());
200     func.setSenha(pf_senhaCadaastro.getText());
201     func.setDataNascimento(dp_dataNascimentoCadaastro.getValue());
202     func.setEmail(tf_emailCadaastro.getText());
203     func.setTelefone1(tf_telefone1Cadaastro.getText());
204     func.setTelefone2(tf_telefone2Cadaastro.getText());
205     func.setAtivo(cb_ativo.isSelected());
206 }

```

```

207     FuncionarioDAO f = new FuncionarioDAO();
208     if (f.alteraFuncionario(func)) {
209         confirmaAlteracao.setText("Alteração realizada com sucesso!!");
210     } else {
211         confirmaAlteracao.setText("Erro ao realizar alteração!!");
212     }
213 }
214
215 private void carregaTabelaFuncionario() throws Exception {
216     col_nome.setCellValueFactory(new PropertyValueFactory<>("nome"));
217     col_cpf.setCellValueFactory(new PropertyValueFactory<>("cpf"));
218     col_telefone1.setCellValueFactory(new PropertyValueFactory<>("telefone1"));
219     col_telefone2.setCellValueFactory(new PropertyValueFactory<>("telefone2"));
220     col_email.setCellValueFactory(new PropertyValueFactory<>("email"));
221
222     FuncionarioDAO f = new FuncionarioDAO();
223     List<Funcionario> listaFuncionario = f.getAllFuncionario();
224     ObservableList<Funcionario> observableListFuncionario;
225
226     observableListFuncionario = FXCollections.observableArrayList(listaFuncionario);
227     tbl_funcionario.setItems(observableListFuncionario);
228 }
229
230 public void selecionarItemTabelaFuncionario(Funcionario funcionario) {
231     if (funcionario.getCpf() != null) {
232         tf_cpfCadastro.setText(funcionario.getCpf());
233         tf_nomeCadastro.setText(funcionario.getNome());
234         pf_senhaCadastro.setText(funcionario.getSenha());
235         tf_emailCadastro.setText(funcionario.getEmail());
236         tf_telefone1Cadastro.setText(funcionario.getTelefone1());
237         tf_telefone2Cadastro.setText(funcionario.getTelefone2());
238         dp_dataNascimentoCadastro.setValue(funcionario.getDataNascimento());
239         cb_ativo.setSelected(funcionario.getAtivo());
240     }
241 }
242
243
244 @Override
245 public void initialize(URL url, ResourceBundle rb) {
246     try {
247         carregaTabelaFuncionario();
248         tbl_funcionario.getSelectionModel().selectedItemProperty().addListener(
249             (observable, oldValue, newValue) -> selecionarItemTabelaFuncionario(newValue));
250     } catch (Exception ex) {
251         //Logger.getLogger(FuncionarioController.class.getName()).log(Level.SEVERE, null, ex);
252     }
253 }
254
255 }

```

Código 16: FuncionarioController.java

5.4 Pacote Peça

5.4.1 Model

```

1 package opencarshop.peca.model;
2
3 import java.io.Serializable;
4
5 public class ItemPeca implements Serializable {
6
7     private int id;
8     private int quantidadeVendida;
9     private double valor;
10    private Peca peca;
11    private VendaPeca vendaPeca;
12
13    public ItemPeca() {
14    }
15
16    /**
17     * @return the id
18     */
19    public int getId() {
20        return id;
21    }
22
23    /**
24     * @param id the id to set
25     */
26    public void setId(int id) {
27        this.id = id;
28    }
29
30 }

```

```

31  /**
32   * @return the quantidadeVendida
33   */
34  public int getQuantidadeVendida() {
35      return quantidadeVendida;
36  }
37
38  /**
39   * @param quantidadeVendida the quantidadeVendida to set
40   */
41  public void setQuantidadeVendida(int quantidadeVendida) {
42      this.quantidadeVendida = quantidadeVendida;
43  }
44
45  /**
46   * @return the valor
47   */
48  public double getValor() {
49      return valor;
50  }
51
52  /**
53   * @param valor the valor to set
54   */
55  public void setValor(double valor) {
56      this.valor = valor;
57  }
58
59  /**
60   * @return the peca
61   */
62  public Peca getPeca() {
63      return peca;
64  }
65
66  /**
67   * @param peca the peca to set
68   */
69  public void setPeca(Peca peca) {
70      this.pecas = peca;
71  }
72
73  /**
74   * @return the vendaPeca
75   */
76  public VendaPeca getVendaPeca() {
77      return vendaPeca;
78  }
79
80  /**
81   * @param vendaPeca the vendaPeca to set
82   */
83  public void setVendaPeca(VendaPeca vendaPeca) {
84      this.vendaPeca = vendaPeca;
85  }
86
87  }

```

Código 17: ItemPeca.java

```

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package opencarshop.pecas.model;
7
8  import java.sql.Connection;
9  import java.sql.Date;
10 import java.sql.PreparedStatement;
11 import java.sql.ResultSet;
12 import java.sql.SQLException;
13 import java.util.ArrayList;
14 import java.util.List;
15 import java.util.logging.Level;
16 import java.util.logging.Logger;
17 import opencarshop.cliente.model.ClienteDAO;
18 import opencarshop.util.ConexaoMySQL;
19
20 public class ItemPecaDAO {
21
22     private Connection connection;
23     private final ConexaoMySQL database = new ConexaoMySQL();
24
25     public ItemPecaDAO() {
26         try {
27             connection = database.conectar();
28         } catch (Exception ex) {
29             Logger.getLogger(PecaDAO.class.getName()).log(Level.SEVERE, null, ex);
30         }
31     }
32

```

```

33 public Connection getConnection() {
34     return connection;
35 }
36
37 public void setConnection(Connection connection) {
38     this.connection = connection;
39 }
40
41 public boolean inserir(ItemPeca item) {
42     String sql = "INSERT INTO ItemPeca(valor, quantidadeVendida, idPeca, idVenda) "
43         + "VALUES(?, ?, ?, ?)";
44     try {
45         PreparedStatement stmt = connection.prepareStatement(sql);
46
47         stmt.setDouble(1, item.getValor());
48         stmt.setInt(2, item.getQuantidadeVendida());
49         stmt.setInt(3, item.getPeca().getId());
50         stmt.setInt(4, item.getVendaPeca().getId());
51
52         stmt.execute();
53
54         return true;
55     } catch (SQLException ex) {
56         Logger.getLogger(ItemPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
57         return false;
58     }
59 }
60
61 }
62
63 public ItemPeca buscar(ItemPeca itemPeca) {
64     String sql = "SELECT * FROM itensDeVenda WHERE id=?";
65     ItemPeca retorno = new ItemPeca();
66
67     try {
68         PreparedStatement stmt = connection.prepareStatement(sql);
69         stmt.setInt(1, itemPeca.getId());
70         ResultSet resultado = stmt.executeQuery();
71         if (resultado.next()) {
72             Peca peca = new Peca();
73             VendaPeca venda = new VendaPeca();
74             itemPeca.setId(resultado.getInt("id"));
75             itemPeca.setQuantidadeVendida(resultado.getInt("quantidadeVendida"));
76             itemPeca.setValor(resultado.getDouble("valor"));
77
78             peca.setId(resultado.getInt("id"));
79             venda.setId(resultado.getInt("id"));
80
81             //Obtendo os dados completos do Cliente associado Venda
82             PecaDAO pecaDAO = new PecaDAO();
83             pecaDAO.setConnection(connection);
84             peca = pecaDAO.buscar(peca);
85
86             VendaPecaDAO vendaDAO = new VendaPecaDAO();
87             vendaDAO.setConnection(connection);
88             venda = vendaDAO.buscar(venda);
89
90             itemPeca.setPeca(peca);
91             itemPeca.setVendaPeca(venda);
92
93             retorno = itemPeca;
94         }
95     } catch (SQLException ex) {
96         Logger.getLogger(ItemPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
97     }
98     return retorno;
99 }
100
101 }
102
103 public List<ItemPeca> listar() {
104     String sql = "SELECT * FROM ItemPeca";
105     List<ItemPeca> retorno = new ArrayList<>();
106     try {
107         PreparedStatement stmt = connection.prepareStatement(sql);
108         ResultSet resultado = stmt.executeQuery();
109         while (resultado.next()) {
110             Peca peca = new Peca();
111             ItemPeca item = new ItemPeca();
112             VendaPeca venda = new VendaPeca();
113
114             item.setId(resultado.getInt("id"));
115             item.setValor(resultado.getDouble("valor"));
116             item.setQuantidadeVendida(resultado.getInt("quantidadeVendida"));
117
118             peca.setId(resultado.getInt("id"));
119             venda.setId(resultado.getInt("id"));
120
121             PecaDAO pecaDAO = new PecaDAO();
122             pecaDAO.setConnection(connection);
123             peca = pecaDAO.buscar(peca);
124
125             VendaPecaDAO vendaDAO = new VendaPecaDAO();
126             vendaDAO.setConnection(connection);
127             venda = vendaDAO.buscar(venda);

```

```

127         item.setPeca(peca);
128         item.setVendaPeca(venda);
129
130         retorno.add(item);
131     }
132 } catch (SQLException ex) {
133     Logger.getLogger(ItemPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
134 }
135 return retorno;
136 }
137
138 public List<ItemPeca> listarPorVenda(VendaPeca venda) {
139     String sql = "SELECT*FROMItemPecaWHEREid=?";
140     List<ItemPeca> retorno = new ArrayList<>();
141     try {
142         PreparedStatement stmt = connection.prepareStatement(sql);
143         stmt.setInt(1, venda.getId());
144         ResultSet resultado = stmt.executeQuery();
145         while (resultado.next()) {
146             ItemPeca item = new ItemPeca();
147             Peca peca = new Peca();
148             VendaPeca v = new VendaPeca();
149             item.setId(resultado.getInt("id"));
150             item.setQuantidadeVendida(resultado.getInt("quantidadeVendida"));
151             item.setValor(resultado.getDouble("valor"));
152
153             peca.setId(resultado.getInt("id"));
154             v.setId(resultado.getInt("id"));
155
156             //Obtendo os dados completos do Produto associado ao Item de Venda
157             PecaDAO pecaDAO = new PecaDAO();
158             pecaDAO.setConnection(connection);
159             peca = pecaDAO.buscar(peca);
160
161             item.setPeca(peca);
162             item.setVendaPeca(v);
163
164             retorno.add(item);
165         }
166     } catch (SQLException ex) {
167         Logger.getLogger(ItemPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
168     }
169     return retorno;
170 }
171
172 }

```

Código 18: ItemPecaDAO.java

```

1 package opencarshop.peca.model;
2
3 import java.io.Serializable;
4
5 public class Peca implements Serializable {
6
7     private int id;
8     private String nome;
9     private double valor;
10    private int quantidade;
11    private boolean ativa;
12
13    public Peca() {
14    }
15
16    public Peca(int id, String nome, double preco, int quantidade, boolean ativa) {
17        this.id = id;
18        this.nome = nome;
19        this.valor = preco;
20        this.quantidade = quantidade;
21        this.ativa = ativa;
22    }
23
24    public int getId() {
25        return id;
26    }
27
28    public void setId(int id) {
29        this.id = id;
30    }
31
32    public String getNome() {
33        return nome;
34    }
35
36    public void setNome(String nome) {
37        this.nome = nome;
38    }
39
40    public double getValor() {
41        return valor;
42    }
43 }

```

```

44
45     public void setValor(double valor) {
46         this.valor = valor;
47     }
48
49     public int getQuantidade() {
50         return quantidade;
51     }
52
53     public void setQuantidade(int quantidade) {
54         this.quantidade = quantidade;
55     }
56
57     @Override
58     public String toString() {
59         return this.nome;
60     }
61
62     /**
63      * @return the ativa
64      */
65     public boolean isAtiva() {
66         return ativa == true;
67     }
68
69     /**
70      * @param ativa the ativa to set
71      */
72     public void setAtiva(boolean ativa) {
73         this.ativa = ativa;
74     }
75
76 }

```

Código 19: Peca.java

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package opencarshop.peca.model;
7
8  import java.sql.Connection;
9  import java.sql.PreparedStatement;
10 import java.sql.ResultSet;
11 import java.sql.SQLException;
12 import java.util.ArrayList;
13 import java.util.List;
14 import java.util.logging.Level;
15 import java.util.logging.Logger;
16 import opencarshop.util.ConexaoMySQL;
17
18 /**
19  *
20  * @author Dimitri
21  */
22 public class PecaDAO {
23
24     private Connection connection;
25     private final ConexaoMySQL database = new ConexaoMySQL();
26
27     public PecaDAO() {
28         try {
29             connection = database.conectar();
30         } catch (Exception ex) {
31             Logger.getLogger(PecaDAO.class.getName()).log(Level.SEVERE, null, ex);
32         }
33     }
34
35     public Connection getConnection() {
36         return connection;
37     }
38
39     public void setConnection(Connection connection) {
40         this.connection = connection;
41     }
42
43     public boolean inserir(Peca peca) {
44         String sql = "INSERT INTO Peca(nome, valor, quantidade, ativa) VALUES(?,?,?,?)";
45
46         try {
47             PreparedStatement stmt = connection.prepareStatement(sql);
48             stmt.setString(1, peca.getNome());
49             stmt.setDouble(2, peca.getValor());
50             stmt.setInt(3, peca.getQuantidade());
51             stmt.setBoolean(4, true);
52             stmt.execute();
53             return true;
54         } catch (SQLException ex) {
55             Logger.getLogger(PecaDAO.class.getName()).log(Level.SEVERE, null, ex);
56             return false;

```

```

57     }
58
59 }
60
61 public boolean atualizar(Peca peca) {
62     String sql = "UPDATE Peca SET nome=?, valor=?, quantidade=?, ativa=? WHERE id=?";
63     try {
64         PreparedStatement stmt = connection.prepareStatement(sql);
65         stmt.setString(1, peca.getNome());
66         stmt.setDouble(2, peca.getValor());
67         stmt.setInt(3, peca.getQuantidade());
68         stmt.setBoolean(4, true);
69         stmt.setInt(5, peca.getId());
70
71         stmt.execute();
72         return true;
73     } catch (SQLException ex) {
74         Logger.getLogger(PecaDAO.class.getName()).log(Level.SEVERE, null, ex);
75         return false;
76     }
77 }
78
79 public boolean inativar(Peca peca) {
80     String sql = "UPDATE Peca SET nome=?, valor=?, quantidade=?, ativa=? WHERE id=?";
81     try {
82         PreparedStatement stmt = connection.prepareStatement(sql);
83         stmt.setString(1, peca.getNome());
84         stmt.setDouble(2, peca.getValor());
85         stmt.setInt(3, peca.getQuantidade());
86         stmt.setBoolean(4, false);
87         stmt.setInt(5, peca.getId());
88
89         stmt.execute();
90         return true;
91     } catch (SQLException ex) {
92         Logger.getLogger(PecaDAO.class.getName()).log(Level.SEVERE, null, ex);
93         return false;
94     }
95 }
96
97 public boolean remover(Peca peca) {
98     String sql = "DELETE FROM Peca WHERE id=?";
99     try {
100         PreparedStatement stmt = connection.prepareStatement(sql);
101         stmt.setInt(1, peca.getId());
102         stmt.execute();
103         return true;
104     } catch (SQLException ex) {
105         Logger.getLogger(PecaDAO.class.getName()).log(Level.SEVERE, null, ex);
106         return false;
107     }
108 }
109
110 public List<Peca> listar() {
111     String sql = "SELECT * FROM Peca WHERE ativa=1";
112     List<Peca> retorno = new ArrayList<>();
113     try {
114         PreparedStatement stmt = connection.prepareStatement(sql);
115         ResultSet resultado = stmt.executeQuery();
116         while (resultado.next()) {
117             Peca peca = new Peca();
118             peca.setId(resultado.getInt("id"));
119             peca.setNome(resultado.getString("nome"));
120             peca.setValor(resultado.getDouble("valor"));
121             peca.setQuantidade(resultado.getInt("quantidade"));
122             retorno.add(peca);
123         }
124     } catch (SQLException ex) {
125         Logger.getLogger(PecaDAO.class.getName()).log(Level.SEVERE, null, ex);
126     }
127     return retorno;
128 }
129
130
131 public Peca buscar(Peca peca) {
132     String sql = "SELECT * FROM Peca WHERE id=?";
133     Peca retorno = new Peca();
134     try {
135         PreparedStatement stmt = connection.prepareStatement(sql);
136         stmt.setInt(1, peca.getId());
137         ResultSet resultado = stmt.executeQuery();
138         if (resultado.next()) {
139             peca.setNome(resultado.getString("nome"));
140             peca.setValor(resultado.getDouble("valor"));
141             peca.setQuantidade(resultado.getInt("quantidade"));
142
143             retorno = peca;
144         }
145     } catch (SQLException ex) {
146         Logger.getLogger(PecaDAO.class.getName()).log(Level.SEVERE, null, ex);
147     }
148     return retorno;
149 }
150

```


Código 20: PecaDAO.java

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package opencarshop.peca.model;
7
8  import java.sql.Date;
9  import java.util.List;
10 import opencarshop.cliente.model.Cliente;
11
12 public class VendaPeca {
13
14     private int id;
15     private List<ItemPeca> itemsVendidos;
16     private Date dataVenda;
17     private double valor;
18     private boolean pago;
19     private Cliente cliente;
20
21     public VendaPeca() {
22
23     }
24
25     /**
26      * @return the id
27      */
28     public int getId() {
29         return id;
30     }
31
32     /**
33      * @param id the id to set
34      */
35     public void setId(int id) {
36         this.id = id;
37     }
38
39     /**
40      * @return the itemsVendidos
41      */
42     public List<ItemPeca> getItemsVendidos() {
43         return itemsVendidos;
44     }
45
46     /**
47      * @param itemsVendidos the itemsVendidos to set
48      */
49     public void setItemsVendidos(List<ItemPeca> itemsVendidos) {
50         this.itemsVendidos = itemsVendidos;
51     }
52
53     /**
54      * @return the dataVenda
55      */
56     public Date getDataVenda() {
57         return dataVenda;
58     }
59
60     /**
61      * @param dataVenda the dataVenda to set
62      */
63     public void setDataVenda(Date dataVenda) {
64         this.dataVenda = dataVenda;
65     }
66
67     /**
68      * @return the valor
69      */
70     public double getValor() {
71         return valor;
72     }
73
74     /**
75      * @param valor the valor to set
76      */
77     public void setValor(double valor) {
78         this.valor = valor;
79     }
80
81     /**
82      * @return the pago
83      */
84     public boolean isPago() {
85         return pago;
86     }
87
88     /**

```

```

89     * @param pago the pago to set
90     */
91     public void setPago(boolean pago) {
92         this.pago = pago;
93     }
94
95     /**
96     * @return the cliente
97     */
98     public Cliente getCliente() {
99         return cliente;
100     }
101
102     /**
103     * @param cliente the cliente to set
104     */
105     public void setCliente(Cliente cliente) {
106         this.cliente = cliente;
107     }
108 }
109

```

Código 21: VendaPeca.java

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package opencarshop.pecas.model;
7
8  import java.sql.Connection;
9  import java.sql.Date;
10 import java.sql.PreparedStatement;
11 import java.sql.ResultSet;
12 import java.sql.SQLException;
13 import java.util.ArrayList;
14 import java.util.HashMap;
15 import java.util.List;
16 import java.util.Map;
17 import java.util.logging.Level;
18 import java.util.logging.Logger;
19 import opencarshop.cliente.model.Cliente;
20 import opencarshop.cliente.model.ClienteDAO;
21 import opencarshop.util.ConexaoMySQL;
22
23 /**
24 *
25 * @author Dimitri
26 */
27 public class VendaPecaDAO {
28
29     private Connection connection;
30     private final ConexaoMySQL database = new ConexaoMySQL();
31
32     public VendaPecaDAO() {
33         try {
34             connection = database.conectar();
35         } catch (Exception ex) {
36             Logger.getLogger(VendaPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
37         }
38     }
39
40     public Connection getConnection() {
41         return connection;
42     }
43
44     public void setConnection(Connection connection) {
45         this.connection = connection;
46     }
47
48     public boolean inserir(VendaPeca vendaPeca) {
49         String sql = "INSERT INTO VendaPeca(dataVenda, valor, pago, cpfCliente) VALUES(?, ?, ?, ?)";
50         try {
51             PreparedStatement stmt = connection.prepareStatement(sql);
52             stmt.setDate(1, vendaPeca.getDataVenda());
53             stmt.setDouble(2, vendaPeca.getValor());
54             stmt.setBoolean(3, vendaPeca.isPago());
55             stmt.setString(4, vendaPeca.getCliente().getCpf());
56             stmt.execute();
57             return true;
58         } catch (SQLException ex) {
59             Logger.getLogger(VendaPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
60             return false;
61         }
62     }
63
64     public VendaPeca buscar(VendaPeca venda) {
65         String sql = "SELECT * FROM VendaPeca WHERE id=?";
66         VendaPeca retorno = new VendaPeca();
67         try {
68             PreparedStatement stmt = connection.prepareStatement(sql);

```

```

69         stmt.setInt(1, venda.getId());
70         ResultSet resultado = stmt.executeQuery();
71         if (resultado.next()) {
72             Cliente cliente = new Cliente();
73             venda.setId(resultado.getInt("id"));
74             venda.setDataVenda(resultado.getDate("dataVenda"));
75             venda.setValor(resultado.getDouble("valor"));
76             venda.setPago(resultado.getBoolean("pago"));
77             cliente.setCpf(resultado.getString("cpf"));
78             venda.setCliente(cliente);
79             retorno = venda;
80         }
81     } catch (SQLException ex) {
82         Logger.getLogger(VendaPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
83     }
84     return retorno;
85 }
86
87 public VendaPeca buscarUltimaVenda() {
88     String sql = "SELECT max(id) as maximo FROM VendaPeca";
89     VendaPeca retorno = new VendaPeca();
90     try {
91         PreparedStatement stmt = connection.prepareStatement(sql);
92         ResultSet resultado = stmt.executeQuery();
93         if (resultado.next()) {
94             retorno.setId(resultado.getInt("maximo"));
95             return retorno;
96         }
97     } catch (SQLException ex) {
98         Logger.getLogger(VendaPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
99     }
100     return retorno;
101 }
102
103 public List<VendaPeca> listar() {
104     String sql = "SELECT * FROM VendaPeca";
105     List<VendaPeca> retorno = new ArrayList<>();
106     try {
107         PreparedStatement stmt = connection.prepareStatement(sql);
108         ResultSet resultado = stmt.executeQuery();
109         while (resultado.next()) {
110             VendaPeca venda = new VendaPeca();
111             Cliente cliente = new Cliente();
112             List<ItemPeca> itemPeca = new ArrayList();
113
114             venda.setId(resultado.getInt("id"));
115             venda.setDataVenda(resultado.getDate("dataVenda"));
116             venda.setValor(resultado.getDouble("valor"));
117             venda.setPago(resultado.getBoolean("pago"));
118
119             cliente.setCpf(resultado.getString("cpfCliente"));
120
121             //Obtendo os dados completos do Cliente associado Venda
122             ClienteDAO clienteDAO = new ClienteDAO();
123             clienteDAO.setConnection(connection);
124             cliente = clienteDAO.buscar(cliente);
125
126             //Obtendo os dados completos dos Itens de Venda associados Venda
127             ItemPecaDAO itemPecaDAO = new ItemPecaDAO();
128             itemPecaDAO.setConnection(connection);
129             itemPeca = itemPecaDAO.listarPorVenda(venda);
130
131             venda.setCliente(cliente);
132             venda.setItemsVendidos(itemPeca);
133             retorno.add(venda);
134         }
135     } catch (SQLException ex) {
136         Logger.getLogger(VendaPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
137     }
138     return retorno;
139 }
140
141 public Map<Integer, ArrayList> listarQuantidadeVendasPorMes() {
142     String sql = "select count(id) as contador, extract(year from dataVenda) as ano, extract(month from dataVenda) as mes from VendaPeca group by ano, mes order by ano, mes";
143     Map<Integer, ArrayList> retorno = new HashMap();
144
145     try {
146         PreparedStatement stmt = connection.prepareStatement(sql);
147         ResultSet resultado = stmt.executeQuery();
148
149         while (resultado.next()) {
150             ArrayList linha = new ArrayList();
151             if (!retorno.containsKey(resultado.getInt("ano"))) {
152                 linha.add(resultado.getInt("mes"));
153                 linha.add(resultado.getInt("contador"));
154                 retorno.put(resultado.getInt("ano"), linha);
155             } else {
156                 ArrayList linhaNova = retorno.get(resultado.getInt("ano"));
157                 linhaNova.add(resultado.getInt("mes"));
158                 linhaNova.add(resultado.getInt("contador"));
159             }
160         }
161     }

```

```

162     } catch (SQLException ex) {
163         Logger.getLogger(VendaPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
164     }
165     return retorno;
166 }
167
168 }

```

Código 22: VendaPecaDAO.java

5.4.2 View

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.text.*?>
4  <?import java.lang.*?>
5  <?import java.util.*?>
6  <?import javafx.scene.*?>
7  <?import javafx.scene.control.*?>
8  <?import javafx.scene.layout.*?>
9
10 <AnchorPane id="AnchorPane" prefHeight="378.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
    javafx.com/fxml/1" fx:controller="opencarshop.peca.controller.CadastroPeca">
11     <children>
12         <SplitPane dividerPositions="0.40635451505016723" layoutX="92.0" layoutY="53.0" prefHeight="400.0" prefWidth="
            600.0" AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.
            topAnchor="0.0">
13             <items>
14                 <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="160.0" prefWidth="100.0">
15                     <children>
16                         <TableView fx:id="tableViewPecas" layoutY="32.0" maxWidth="500.0" prefHeight="398.0" prefWidth="
                            240.0" AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.
                            topAnchor="0.0">
17                             <columns>
18                                 <TableColumn fx:id="tableColumnPecaNome" prefWidth="75.0" text="Nome" />
19                                 <TableColumn fx:id="tableColumnPecaQuantidade" prefWidth="75.0" text="Quantidade" />
20                             </columns>
21                             <columnResizePolicy>
22                                 <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
23                             </columnResizePolicy>
24                         </TableView>
25                     </children>
26                 </AnchorPane>
27                 <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="160.0" prefWidth="100.0">
28                     <children>
29                         <Label layoutX="112.0" layoutY="14.0" text="Detalhes Da Peca">
30                             <font>
31                                 <Font name="System_Bold" size="14.0" />
32                             </font>
33                         </Label>
34                         <GridPane layoutX="54.0" layoutY="90.0" AnchorPane.leftAnchor="75.0">
35                             <columnConstraints>
36                                 <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
37                                 <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
38                             </columnConstraints>
39                             <rowConstraints>
40                                 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
41                                 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
42                                 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
43                                 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
44                             </rowConstraints>
45                             <children>
46                                 <Label text="Codigo" />
47                                 <Label text="Nome" GridPane.rowIndex="1" />
48                                 <Label text="Preco" GridPane.rowIndex="2" />
49                                 <Label text="Quantidade" GridPane.rowIndex="3" />
50                                 <Label fx:id="labelPecaCodigo" GridPane.columnIndex="1" GridPane.rowIndex="1" />
51                                 <Label fx:id="labelPecaNome" GridPane.columnIndex="1" GridPane.rowIndex="1" />
52                                 <Label fx:id="labelPecaPreco" GridPane.columnIndex="1" GridPane.rowIndex="2" />
53                                 <Label fx:id="labelPecaQuantidade" GridPane.columnIndex="1" GridPane.rowIndex="3" />
54                             </children>
55                         </GridPane>
56                         <Group AnchorPane.bottomAnchor="20.0" AnchorPane.rightAnchor="20.0">
57                             <children>
58                                 <Button fx:id="buttonInserir" layoutX="97.0" layoutY="293.0" mnemonicParsing="false"
                                    onAction="#handleButtonInserir" prefHeight="25.0" prefWidth="64.0" text="Inserir" />
59                                 <Button fx:id="buttonAlterar" layoutX="169.0" layoutY="293.0" mnemonicParsing="false"
                                    onAction="#handleButtonAlterar" prefHeight="25.0" prefWidth="64.0" text="Alterar" />
60                                 <Button fx:id="buttonInativar" layoutX="240.0" layoutY="293.0" mnemonicParsing="false"
                                    onAction="#handleButtonInativar" text="Inativar" />
61                             </children>
62                         </Group>
63                     </children>
64                 </AnchorPane>
65             </items>
66         </SplitPane>
67     </children>
68 </AnchorPane>

```

Código 23: CadastroPeca.fxml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import java.lang.*?>
4 <?import java.util.*?>
5 <?import javafx.scene.*?>
6 <?import javafx.scene.control.*?>
7 <?import javafx.scene.layout.*?>
8
9 <AnchorPane id="AnchorPane" prefHeight="183.0" prefWidth="340.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
10   javafx.com/fxml/1" fx:controller="opencarshop.pecas.controller.CadastroPecaDialog">
11   <children>
12     <GridPane layoutX="20.0" layoutY="8.0" prefHeight="120.0" prefWidth="298.0">
13       <columnConstraints>
14         <ColumnConstraints hgrow="SOMETIMES" maxWidth="168.0" minWidth="10.0" prefWidth="128.0" />
15         <ColumnConstraints hgrow="SOMETIMES" maxWidth="248.0" minWidth="10.0" prefWidth="218.0" />
16       </columnConstraints>
17       <rowConstraints>
18         <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
19         <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
20         <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
21       </rowConstraints>
22       <children>
23         <Label fx:id="labelPecaNome" text="Nome" />
24         <Label fx:id="labelPecaPreco" text="Preco" GridPane.rowIndex="1" />
25         <Label fx:id="labelPecaQuantidade" text="Quantidade" GridPane.rowIndex="2" />
26         <TextField fx:id="textFieldPecaNome" GridPane.columnIndex="1" />
27         <TextField fx:id="textFieldPecaPreco" GridPane.columnIndex="1" GridPane.rowIndex="1" />
28         <TextField fx:id="textFieldPecaQuantidade" GridPane.columnIndex="1" GridPane.rowIndex="2" />
29       </children>
30     </GridPane>
31     <Group layoutX="15.0" AnchorPane.bottomAnchor="20.0" AnchorPane.rightAnchor="20.0">
32       <children>
33         <Button fx:id="buttonConfirmar" layoutX="129.0" layoutY="137.0" mnemonicParsing="false" onAction="#"
34           handleButtonConfirmar" text="Confirmar" />
35         <Button fx:id="buttonCancelar" layoutX="208.0" layoutY="137.0" mnemonicParsing="false" onAction="#"
36           handleButtonCancelar" text="Cancelar" />
37       </children>
38     </Group>
39   </children>
40 </AnchorPane>
```

Código 24: CadastroPecaDialog.fxml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.chart.*?>
4 <?import java.lang.*?>
5 <?import java.util.*?>
6 <?import javafx.scene.*?>
7 <?import javafx.scene.control.*?>
8 <?import javafx.scene.layout.*?>
9
10
11 <AnchorPane id="AnchorPane" prefHeight="400.0" prefWidth="600.0" xmlns:fx="http://javafx.com/fxml/1" xmlns="http://
12   javafx.com/javafx/8" fx:controller="opencarshop.pecas.controller.GraficosVendasPorMesController">
13   <children>
14     <BarChart fx:id="barChart" layoutX="37.0" layoutY="18.0" prefHeight="364.0" prefWidth="500.0" title="Vendas por
15       M s">
16       <xAxis>
17         <CategoryAxis fx:id="categoryAxis" side="BOTTOM" />
18       </xAxis>
19       <yAxis>
20         <NumberAxis fx:id="numberAxis" side="LEFT" />
21       </yAxis>
22     </BarChart>
23   </children>
24 </AnchorPane>
```

Código 25: GraficosVendasPorMes.fxml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.text.*?>
4 <?import java.lang.*?>
5 <?import java.util.*?>
6 <?import javafx.scene.*?>
7 <?import javafx.scene.control.*?>
8 <?import javafx.scene.layout.*?>
9
10 <AnchorPane id="AnchorPane" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
11   javafx.com/fxml/1" fx:controller="opencarshop.pecas.controller.VendaPecasController">
12   <children>
```

```

12     <SplitPane dividerPositions="0.46153846153846156" layoutX="73.0" layoutY="21.0" prefHeight="400.0" prefWidth="
13     600.0" AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.
14     topAnchor="0.0">
15         <items>
16             <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="398.0" prefWidth="269.0">
17                 <children>
18                     <TableView fx:id="tableViewVendas" layoutX="28.0" layoutY="29.0" prefHeight="398.0" prefWidth="
19                     275.0" AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.
20                     topAnchor="0.0">
21                         <columns>
22                             <TableColumn fx:id="tableColumnVendaCodigo" prefWidth="79.0" text="Código" />
23                             <TableColumn fx:id="tableColumnVendaData" prefWidth="82.0" text="Data" />
24                             <TableColumn fx:id="tableColumnVendaPrecoTotal" prefWidth="111.0" text="Preço Total" />
25                         </columns>
26                     </TableView>
27                 </children>
28             </AnchorPane>
29             <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="160.0" prefWidth="100.0">
30                 <children>
31                     <Label layoutX="100.0" layoutY="14.0" text="Detalhes da Venda">
32                         <font>
33                             <Font name="System_Bold" size="14.0" />
34                         </font>
35                     </Label>
36                     <GridPane layoutX="25.0" layoutY="62.0" prefHeight="146.0" prefWidth="251.0">
37                         <columnConstraints>
38                             <ColumnConstraints hgrow="SOMETIMES" maxWidth="113.0" minWidth="10.0" prefWidth="84.0"
39                             />
40                             <ColumnConstraints hgrow="SOMETIMES" maxWidth="202.0" minWidth="10.0" prefWidth="180.0"
41                             />
42                         </columnConstraints>
43                         <rowConstraints>
44                             <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
45                             <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
46                             <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
47                             <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
48                             <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
49                         </rowConstraints>
50                         <children>
51                             <Label text="Código" />
52                             <Label text="Data" GridPane.rowIndex="1" />
53                             <Label text="Valor" GridPane.rowIndex="2" />
54                             <Label text="Pago" GridPane.rowIndex="3" />
55                             <Label text="Cliente" GridPane.rowIndex="4" />
56                             <Label fx:id="labelVendaCodigo" GridPane.columnIndex="1" />
57                             <Label fx:id="labelVendaData" GridPane.columnIndex="1" GridPane.rowIndex="1" />
58                             <Label fx:id="labelVendaValor" GridPane.columnIndex="1" GridPane.rowIndex="2" />
59                             <Label fx:id="labelVendaPago" GridPane.columnIndex="1" GridPane.rowIndex="3" />
60                             <Label fx:id="labelVendaCliente" GridPane.columnIndex="1" GridPane.rowIndex="4" />
61                         </children>
62                     </GridPane>
63                     <Group AnchorPane.bottomAnchor="20.0" AnchorPane.rightAnchor="20.0" />
64                     <Button fx:id="buttonInserir" layoutX="100.0" layoutY="342.0" mnemonicParsing="false" onAction=
65                     "#handleButtonInserir" prefHeight="25.0" prefWidth="65.0" text="Inserir" />
66                     <TableView layoutX="22.0" layoutY="219.0" prefHeight="102.0" prefWidth="257.0">
67                         <columns>
68                             <TableColumn fx:id="tableColumnNomePeca" prefWidth="75.0" text="Peca" />
69                             <TableColumn fx:id="tableColumnQuantidadePeca" prefWidth="75.0" text="Quantidade" />
70                             <TableColumn fx:id="tableColumnPrecoPeca" prefWidth="75.0" text="Preco" />
71                         </columns>
72                     </TableView>
73                 </children>
74             </AnchorPane>
75         </items>
76     </SplitPane>
77 </children>
78 </AnchorPane>

```

Código 26: VendaPecas.fxml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import java.lang.*?>
4  <?import java.util.*?>
5  <?import javafx.scene.*?>
6  <?import javafx.scene.control.*?>
7  <?import javafx.scene.layout.*?>
8
9  <AnchorPane id="AnchorPane" prefHeight="344.0" prefWidth="443.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
10  javafx.com/fxml/1" fx:controller="opencarshop.peca.controller.VendaPecasDialogController">
11     <children>
12         <GridPane layoutX="81.0" layoutY="33.0" prefHeight="251.0" prefWidth="324.0">
13             <columnConstraints>
14                 <ColumnConstraints hgrow="SOMETIMES" maxWidth="95.0" minWidth="7.0" prefWidth="60.0" />
15                 <ColumnConstraints hgrow="SOMETIMES" maxWidth="267.0" minWidth="10.0" prefWidth="264.0" />
16             </columnConstraints>
17             <rowConstraints>
18                 <RowConstraints maxHeight="37.0" minHeight="10.0" prefHeight="25.0" vgrow="SOMETIMES" />
19                 <RowConstraints maxHeight="64.0" minHeight="10.0" prefHeight="26.0" vgrow="SOMETIMES" />
20                 <RowConstraints maxHeight="77.0" minHeight="10.0" prefHeight="23.0" vgrow="SOMETIMES" />
21                 <RowConstraints maxHeight="91.0" minHeight="10.0" prefHeight="27.0" vgrow="SOMETIMES" />
22                 <RowConstraints maxHeight="112.0" minHeight="10.0" prefHeight="112.0" vgrow="SOMETIMES" />

```

```

22         <RowConstraints maxHeight="29.0" minHeight="10.0" prefHeight="24.0" vgrow="SOMETIMES" />
23     </rowConstraints>
24     <children>
25         <Label text="Cliente" />
26         <Label text="Data" GridPane.rowIndex="1" />
27         <Label text="Pago" GridPane.rowIndex="2" />
28         <Label text="Peca" GridPane.rowIndex="3" />
29         <Label text="Itens" GridPane.rowIndex="4" />
30         <Label text="Valor" GridPane.rowIndex="5" />
31         <ComboBox fx:id="comboBoxVendaCliente" prefHeight="25.0" prefWidth="173.0" promptText="Selecione o cliente..." GridPane.columnIndex="1" />
32         <DatePicker fx:id="datePickerVendaData" GridPane.columnIndex="1" GridPane.rowIndex="1" />
33         <CheckBox fx:id="checkBoxVendaPago" mnemonicParsing="false" GridPane.columnIndex="1" GridPane.rowIndex="2" />
34         <TableView fx:id="tableViewItensDeVenda" prefHeight="200.0" prefWidth="200.0" GridPane.columnIndex="1" GridPane.rowIndex="4">
35             <columns>
36                 <TableColumn fx:id="tableColumnItemDeVendaPeca" prefWidth="127.0" text="Peca" />
37                 <TableColumn fx:id="tableColumnItemDeVendaQuantidade" prefWidth="75.0" text="Quantidade" />
38                 <TableColumn fx:id="tableColumnItemDeVendaValor" prefWidth="60.0" text="Valor" />
39             </columns>
40         </TableView>
41         <TextField fx:id="textFieldVendaValor" editable="false" GridPane.columnIndex="1" GridPane.rowIndex="5" />
42         <HBox prefHeight="100.0" prefWidth="200.0" GridPane.columnIndex="1" GridPane.rowIndex="3">
43             <children>
44                 <ComboBox fx:id="comboBoxVendaPeca" prefHeight="25.0" prefWidth="179.0" promptText="Selecione a peca..." />
45                 <TextField fx:id="textFieldVendaQuantidade" prefHeight="25.0" prefWidth="40.0" />
46                 <Button fx:id="buttonAdicionar" mnemonicParsing="false" onAction="#handleButtonAdicionar" prefHeight="25.0" prefWidth="68.0" text="+" />
47             </children>
48         </HBox>
49     </children>
50 </GridPane>
51 <Button fx:id="buttonConfirmar" layoutX="252.0" layoutY="297.0" mnemonicParsing="false" onAction="#handleButtonConfirmar" text="Confirmar" />
52 <Button fx:id="buttonCancelar" layoutX="343.0" layoutY="297.0" mnemonicParsing="false" onAction="#handleButtonCancelar" text="Cancelar" />
53 </children>
54 </AnchorPane>

```

Código 27: VendaPecasDialog.fxml

5.4.3 Controller

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package opencarshop.peca.controller;
7
8  import java.io.IOException;
9  import java.net.URL;
10 import java.sql.Connection;
11 import java.util.List;
12 import java.util.ResourceBundle;
13 import java.util.logging.Level;
14 import java.util.logging.Logger;
15 import javafx.collections.FXCollections;
16 import javafx.collections.ObservableList;
17 import javafx.fxml.FXML;
18 import javafx.fxml.FXMLLoader;
19 import javafx.fxml.Initializable;
20 import javafx.scene.Scene;
21 import javafx.scene.control.Alert;
22 import javafx.scene.control.Button;
23 import javafx.scene.control.Label;
24 import javafx.scene.control.TableColumn;
25 import javafx.scene.control.TableView;
26 import javafx.scene.control.cell.PropertyValueFactory;
27 import javafx.scene.layout.AnchorPane;
28 import javafx.stage.Stage;
29 import opencarshop.peca.model.PecaDAO;
30 import opencarshop.util.ConexaoMySQL;
31 import opencarshop.peca.model.Peca;
32
33 /**
34 *
35 * @author Dimitri
36 */
37 public class CadastroPeca implements Initializable {
38
39     @FXML
40     private TableView<Peca> tableViewPecas;
41     @FXML
42     private TableColumn<Peca, String> tableColumnPecaNome;

```

```

43  @FXML
44  private TableColumn<Peca, String> tableColumnPecaQuantidade;
45  @FXML
46  private Label labelPecaCodigo;
47  @FXML
48  private Label labelPecaNome;
49  @FXML
50  private Label labelPecaPreco;
51  @FXML
52  private Label labelPecaQuantidade;
53  @FXML
54  private Button buttonInserir;
55  @FXML
56  private Button buttonAlterar;
57  @FXML
58  private Button buttonInativar;
59
60  private List<Peca> listPecas;
61  private ObservableList<Peca> observableListPecas;
62
63  //Atributos para manipula o de Banco de Dados
64  private final PecaDAO pecaDAO = new PecaDAO();
65
66  @Override
67  public void initialize(URL url, ResourceBundle rb) {
68
69      carregarTableViewPeca();
70      // Listen acionado diante de quaisquer altera es na sele o de itens do TableView
71      tableViewPecas.getSelectionModel().selectedItemProperty().addListener(
72          (observable, oldValue, newValue) -> selecionarItemTableViewPecas(newValue));
73  }
74
75  public void carregarTableViewPeca() {
76      tableColumnPecaNome.setCellValueFactory(new PropertyValueFactory<>("nome"));
77      tableColumnPecaQuantidade.setCellValueFactory(new PropertyValueFactory<>("quantidade"));
78
79      listPecas = pecaDAO.listar();
80
81      observableListPecas = FXCollections.observableArrayList(listPecas);
82      tableViewPecas.setItems(observableListPecas);
83  }
84
85  public void selecionarItemTableViewPecas(Peca peca) {
86      if (peca != null) {
87          String cdPeca = String.valueOf(peca.getId());
88          labelPecaCodigo.setText(cdPeca);
89          labelPecaNome.setText(peca.getNome());
90          String preco = String.valueOf(peca.getValor());
91          labelPecaPreco.setText(preco);
92          String quantidade = String.valueOf(peca.getQuantidade());
93          labelPecaQuantidade.setText(quantidade);
94      }
95      else {
96          labelPecaCodigo.setText("");
97          labelPecaNome.setText("");
98          labelPecaPreco.setText("");
99          labelPecaQuantidade.setText("");
100     }
101 }
102
103 @FXML
104 public void handleButtonInserir() throws IOException {
105     Peca peca = new Peca();
106     boolean buttonConfirmarClicado = showCadastroPecaDialog(peca);
107     System.out.println(peca.getNome());
108     if (buttonConfirmarClicado) {
109         pecaDAO.inserir(peca);
110         carregarTableViewPeca();
111     }
112 }
113
114 @FXML
115 public void handleButtonAlterar() throws IOException {
116     Peca peca = tableViewPecas.getSelectionModel().getSelectedItem();
117     if (peca != null) {
118         boolean buttonConfirmarClicado = showCadastroPecaDialog(peca);
119         if (buttonConfirmarClicado) {
120             pecaDAO.atualizar(peca);
121             carregarTableViewPeca();
122         }
123     }
124     else {
125         Alert alert = new Alert(Alert.AlertType.ERROR);
126         alert.setContentText("Por favor, escolha uma peca na Tabela!");
127         alert.show();
128     }
129 }
130
131 @FXML
132 public void handleButtonInativar() throws IOException {
133     Peca peca = tableViewPecas.getSelectionModel().getSelectedItem();
134     if (peca != null) {
135         pecaDAO.inativar(peca);
136         carregarTableViewPeca();
137     }
138     else {

```



```

137         Alert alert = new Alert(Alert.AlertType.ERROR);
138         alert.setContentText("Por favor, escolha uma peça na Tabela!");
139         alert.show();
140     }
141 }
142
143 public boolean showCadastroPecaDialog(Peca peca) throws IOException {
144     FXMLLoader loader = new FXMLLoader();
145     loader.setLocation(CadastroPecaDialog.class.getResource("/opencarshop/peca/view/CadastroPecaDialog.fxml"));
146     AnchorPane page = (AnchorPane) loader.load();
147
148     // Criando um Estgio de Dialogo (Stage Dialog)
149     Stage dialogStage = new Stage();
150     dialogStage.setTitle("Cadastro de Peças");
151     Scene scene = new Scene(page);
152     dialogStage.setScene(scene);
153
154     // Setando o peça no Controller.
155     CadastroPecaDialog controller = loader.getController();
156     controller.setDialogStage(dialogStage);
157     controller.setPeca(peca);
158
159     // Mostra o Dialog e espera até que o usuário o feche
160     dialogStage.showAndWait();
161
162     return controller.isButtonConfirmarClickado();
163 }
164 }
165 }
166 }

```

Código 28: CadastroPeca.java

```

1 package opencarshop.peca.controller;
2
3 import java.net.URL;
4 import java.util.ResourceBundle;
5 import javafx.fxml.FXML;
6 import javafx.fxml.Initializable;
7 import javafx.scene.control.Button;
8 import javafx.scene.control.Label;
9 import javafx.scene.control.TextField;
10 import javafx.stage.Stage;
11 import opencarshop.peca.model.Peca;
12
13 public class CadastroPecaDialog implements Initializable {
14
15     @FXML
16     private Label labelPecaNome;
17     @FXML
18     private Label labelPecaPreco;
19     @FXML
20     private Label labelPecaQuantidade;
21     @FXML
22     private TextField textFieldPecaNome;
23     @FXML
24     private TextField textFieldPecaPreco;
25     @FXML
26     private TextField textFieldPecaQuantidade;
27
28     @FXML
29     private Button buttonConfirmar;
30     @FXML
31     private Button buttonCancelar;
32
33     private Stage dialogStage;
34     private boolean buttonConfirmarClickado = false;
35     private Peca peca;
36
37     @Override
38     public void initialize(URL url, ResourceBundle rb) {
39         // TODO
40     }
41
42     public Stage getDialogStage() {
43         return dialogStage;
44     }
45
46     public void setDialogStage(Stage dialogStage) {
47         this.dialogStage = dialogStage;
48     }
49
50     public boolean isButtonConfirmarClickado() {
51         return buttonConfirmarClickado;
52     }
53
54     public void setButtonConfirmarClickado(boolean buttonConfirmarClickado) {
55         this.buttonConfirmarClickado = buttonConfirmarClickado;
56     }
57
58     public Peca getPeca() {
59         return peca;
60     }
61 }

```

```

60     }
61
62     public void setPeca(Peca peca) {
63         this.pecas = peca;
64         this.textFieldPecaNome.setText(pecas.getNome());
65         String preco = String.valueOf(pecas.getValor());
66         this.textFieldPecaPreco.setText(preco);
67         String quantidade = String.valueOf(pecas.getQuantidade());
68         this.textFieldPecaQuantidade.setText(quantidade);
69     }
70
71     @FXML
72     public void handleButtonConfirmar() {
73         peca.setNome(textFieldPecaNome.getText());
74         Double preco = Double.parseDouble(textFieldPecaPreco.getText());
75         peca.setValor(preco);
76         int quantidade = Integer.parseInt(textFieldPecaQuantidade.getText());
77         peca.setQuantidade(quantidade);
78
79         buttonConfirmarClicked = true;
80         dialogStage.close();
81     }
82
83     @FXML
84     public void handleButtonCancelar() {
85         dialogStage.close();
86     }
87 }

```

Código 29: CadastroPecaDialog.java

```

1  package opencarshop.pecas.controller;
2
3  import java.net.URL;
4  import java.sql.Connection;
5  import java.util.ArrayList;
6  import java.util.Arrays;
7  import java.util.Map;
8  import java.util.ResourceBundle;
9  import javafx.collections.FXCollections;
10 import javafx.collections.ObservableList;
11 import javafx.fxml.FXML;
12 import javafx.fxml.Initializable;
13 import javafx.scene.chart.BarChart;
14 import javafx.scene.chart.CategoryAxis;
15 import javafx.scene.chart.NumberAxis;
16 import javafx.scene.chart.XYChart;
17 import opencarshop.pecas.modelo.VendaPecaDAO;
18
19 public class GraficosVendasPorMesController implements Initializable {
20
21     @FXML
22     private BarChart<String, Integer> barChart;
23     @FXML
24     private CategoryAxis categoryAxis;
25     @FXML
26     private NumberAxis numberAxis;
27
28     private ObservableList<String> observableListMeses = FXCollections.observableArrayList();
29     //Atributos para manipular o de Banco de Dados
30
31     private final VendaPecaDAO vendaDAO = new VendaPecaDAO();
32
33     @Override
34     public void initialize(URL url, ResourceBundle rb) {
35         // Obtém um array com nomes dos meses em inglês.
36         String[] arrayMeses = {"Jan", "Fev", "Mar", "Abr", "Mai", "Jun", "Jul", "Ago", "Set", "Out", "Nov", "Dez"};
37         // Converte o array em uma lista e adiciona em nossa ObservableList de meses.
38         observableListMeses.addAll(Arrays.asList(arrayMeses));
39         // Associa os nomes de meses como categorias para o eixo horizontal.
40         categoryAxis.setCategories(observableListMeses);
41
42         Map<Integer, ArrayList> dados = vendaDAO.listarQuantidadeVendasPorMes();
43         for (Map.Entry<Integer, ArrayList> dadosItem : dados.entrySet()) {
44             XYChart.Series<String, Integer> series = new XYChart.Series<>();
45             series.setName(dadosItem.getKey().toString());
46             for (int i = 0; i < dadosItem.getValue().size(); i = i + 2) {
47                 String mes;
48                 Integer quantidade;
49                 mes = retornaNomeMes((int) dadosItem.getValue().get(i));
50                 quantidade = (Integer) dadosItem.getValue().get(i + 1);
51                 series.getData().add(new XYChart.Data<>(mes, quantidade));
52             }
53             barChart.getData().add(series);
54         }
55     }
56
57     public String retornaNomeMes(int mes) {
58         switch (mes) {
59             case 1:
60                 return "Jan";
61             case 2:

```

```

62         return "Fev";
63     case 3:
64         return "Mar";
65     case 4:
66         return "Abr";
67     case 5:
68         return "Mai";
69     case 6:
70         return "Jun";
71     case 7:
72         return "Jul";
73     case 8:
74         return "Ago";
75     case 9:
76         return "Set";
77     case 10:
78         return "Out";
79     case 11:
80         return "Nov";
81     case 12:
82         return "Dez";
83     default:
84         return "";
85     }
86 }
87
88 }

```

Código 30: GraficosVendasPorMesController.java

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package opencarshop.peca.controller;
7
8  import java.io.IOException;
9  import java.net.URL;
10 import java.sql.Connection;
11 import java.sql.SQLException;
12 import java.time.LocalDate;
13 import java.util.ArrayList;
14 import java.util.List;
15 import java.util.ResourceBundle;
16 import java.util.logging.Level;
17 import java.util.logging.Logger;
18 import javafx.collections.FXCollections;
19 import javafx.collections.ObservableList;
20 import javafx.fxml.FXML;
21 import javafx.fxml.FXMLLoader;
22 import javafx.fxml.Initializable;
23 import javafx.scene.Scene;
24 import javafx.scene.control.Button;
25 import javafx.scene.control.Label;
26 import javafx.scene.control.TableColumn;
27 import javafx.scene.control.TableView;
28 import javafx.scene.control.cell.PropertyValueFactory;
29 import javafx.scene.layout.AnchorPane;
30 import javafx.stage.Stage;
31 import opencarshop.peca.model.PecaDAO;
32 import opencarshop.peca.model.ItemPecaDAO;
33 import opencarshop.peca.model.ItemPeca;
34 import opencarshop.peca.model.Peca;
35 import opencarshop.peca.model.VendaPeca;
36 import opencarshop.peca.model.VendaPecaDAO;
37 import opencarshop.util.ConexaoMySQL;
38
39 public class VendaPecasController implements Initializable {
40
41     @FXML
42     private TableView<VendaPeca> tableViewVendas;
43     @FXML
44     private TableColumn<VendaPeca, Integer> tableColumnVendaCodigo;
45     @FXML
46     private TableColumn<VendaPeca, LocalDate> tableColumnVendaData;
47     @FXML
48     private TableColumn<VendaPeca, String> tableColumnVendaPrecoTotal;
49
50     @FXML
51     private TableView<ItemPeca> tableViewPecasVendidas;
52     @FXML
53     private TableColumn<ItemPeca, String> tableColumnNomePeca;
54     @FXML
55     private TableColumn<ItemPeca, Double> tableColumnPrecoPeca;
56     @FXML
57     private TableColumn<ItemPeca, Integer> tableColumnQuantidadePeca;
58
59     @FXML
60     private Button buttonInserir;
61     @FXML
62     private Label labelVendaCodigo;

```

```

63  @FXML
64  private Label labelVendaData;
65  @FXML
66  private Label labelVendaValor;
67  @FXML
68  private Label labelVendaPago;
69  @FXML
70  private Label labelVendaCliente;
71
72  private List<VendaPeca> listVendas;
73  private ObservableList<VendaPeca> observableListVendas;
74
75  private final ConexaoMySQL database = new ConexaoMySQL();
76  private Connection connection;
77  private final VendaPecaDAO vendaDAO = new VendaPecaDAO();
78  private final ItemPecaDAO itemDeVendaDAO = new ItemPecaDAO();
79  private final PecaDAO pecaDAO = new PecaDAO();
80
81  @Override
82  public void initialize(URL location, ResourceBundle resources) {
83      try {
84          connection = database.conectar();
85          vendaDAO.setConnection(connection);
86      } catch (Exception ex) {
87          Logger.getLogger(VendaPecasController.class.getName()).log(Level.SEVERE, null, ex);
88      }
89      carregarTableViewVendas();
90      selecionarItemTableViewVendas(null);
91      // Listen acionado diante de quaisquer altera es na sele o de itens do TableView
92      tableViewVendas.getSelectionModel().selectedItemProperty().addListener(
93          (observable, oldValue, newValue) -> selecionarItemTableViewVendas(newValue));
94  }
95
96  public void mostrarItemTable(VendaPeca venda) {
97      for (ItemPeca i : venda.getItemsVendidos()) {
98          tableColumnNomePeca.setCellValueFactory(new PropertyValueFactory<>("nome"));
99          tableColumnPrecoPeca.setCellValueFactory(new PropertyValueFactory<>("valor"));
100          tableColumnQuantidadePeca.setCellValueFactory(new PropertyValueFactory<>("quantidadeVenda"));
101          System.out.println(i.getPeca().getNome());
102      }
103  }
104
105  }
106
107  public void selecionarItemTableViewVendas(VendaPeca venda) {
108      if (venda != null) {
109          labelVendaCodigo.setText(String.valueOf(venda.getId()));
110          labelVendaData.setText(String.valueOf(venda.getDataVenda()));
111          labelVendaValor.setText(String.format("%.2f", venda.getValor()));
112          String pago;
113          if (venda.isPago() == true) {
114              pago = "Sim";
115          } else {
116              pago = "Nao";
117          }
118          labelVendaPago.setText(pago);
119          labelVendaCliente.setText(venda.getCliente().getNome());
120
121          mostrarItemTable(venda);
122      } else {
123          labelVendaCodigo.setText("");
124          labelVendaData.setText("");
125          labelVendaValor.setText("");
126          labelVendaPago.setText("");
127          labelVendaCliente.setText("");
128      }
129  }
130
131  }
132
133  public void carregarTableViewVendas() {
134      tableColumnVendaCodigo.setCellValueFactory(new PropertyValueFactory<>("id"));
135      tableColumnVendaData.setCellValueFactory(new PropertyValueFactory<>("dataVenda"));
136      tableColumnVendaPrecoTotal.setCellValueFactory(new PropertyValueFactory<>("valor"));
137
138      listVendas = vendaDAO.listar();
139
140      observableListVendas = FXCollections.observableArrayList(listVendas);
141      tableViewVendas.setItems(observableListVendas);
142  }
143
144  @FXML
145  public void handleButtonInserir() throws IOException, SQLException {
146      VendaPeca venda = new VendaPeca();
147      List<ItemPeca> listItensDeVenda = new ArrayList<>();
148      venda.setItemsVendidos(listItensDeVenda);
149      boolean buttonConfirmarClickado = showVendaPecasDialog(venda);
150
151      if (buttonConfirmarClickado) {
152          try {
153              vendaDAO.inserir(venda);
154
155              connection.setAutoCommit(false);
156              vendaDAO.setConnection(connection);
157              itemDeVendaDAO.setConnection(connection);

```

```

157         pecaDAO.setConnection(connection);
158
159         for (ItemPeca itemPeca : venda.getItemsVendidos()) {
160             Peca peca = itemPeca.getPeca();
161             itemPeca.setVendaPeca(vendaDAO.buscarUltimaVenda());
162             itemDeVendaDAO.inserir(itemPeca);
163             peca.setQuantidade(pecas.getQuantidade() - itemPeca.getQuantidadeVendida());
164             pecaDAO.atualizar(pecas);
165         }
166         connection.commit();
167         carregarTableViewVendas();
168     } catch (SQLException ex) {
169         try {
170             connection.rollback();
171         } catch (SQLException ex1) {
172             Logger.getLogger(VendaPecasController.class.getName()).log(Level.SEVERE, null, ex1);
173         }
174         Logger.getLogger(VendaPecasController.class.getName()).log(Level.SEVERE, null, ex);
175     }
176 }
177
178 }
179
180 public boolean showVendaPecasDialog(VendaPeca venda) throws IOException {
181
182     FXMLLoader loader = new FXMLLoader();
183     loader.setLocation(VendaPecasDialogController.class.getResource("/opencarshop/peca/view/VendaPecasDialog.fxml"));
184     AnchorPane page = (AnchorPane) loader.load();
185
186     // Criando um Estgio de Di logo (Stage Dialog)
187     Stage dialogStage = new Stage();
188     dialogStage.setTitle("Registro de Vendas");
189     Scene scene = new Scene(page);
190     dialogStage.setScene(scene);
191     // Setando a Venda no Controller.
192     VendaPecasDialogController controller = loader.getController();
193     controller.setDialogStage(dialogStage);
194     controller.setVenda(venda);
195     // Mostra o Dialog e espera at que o usu rio o feche
196     dialogStage.showAndWait();
197     return controller.isButtonConfirmarClicked();
198 }
199
200 }

```

Código 31: VendaPecasController.java

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package opencarshop.peca.controller;
7
8  import java.net.URL;
9  import java.sql.Date;
10 import java.util.List;
11 import java.util.ResourceBundle;
12 import java.util.logging.Level;
13 import java.util.logging.Logger;
14 import javafx.collections.FXCollections;
15 import javafx.collections.ObservableList;
16 import javafx.fxml.FXML;
17 import javafx.fxml.Initializable;
18 import javafx.scene.control.Alert;
19 import javafx.scene.control.Button;
20 import javafx.scene.control.CheckBox;
21 import javafx.scene.control.ComboBox;
22 import javafx.scene.control.DatePicker;
23 import javafx.scene.control.TableColumn;
24 import javafx.scene.control.TableView;
25 import javafx.scene.control.TextField;
26 import javafx.scene.control.cell.PropertyValueFactory;
27 import javafx.stage.Stage;
28 import opencarshop.cliente.model.Cliente;
29 import opencarshop.peca.model.ItemPeca;
30 import opencarshop.peca.model.Peca;
31 import opencarshop.peca.model.PecaDAO;
32 import opencarshop.cliente.model.ClienteDAO;
33 import opencarshop.peca.model.VendaPeca;
34
35 public class VendaPecasDialogController implements Initializable {
36
37     @FXML
38     private ComboBox comboBoxVendaCliente;
39     @FXML
40     private DatePicker datePickerVendaData;
41     @FXML
42     private CheckBox checkBoxVendaPago;
43     @FXML
44     private ComboBox comboBoxVendaPeca;

```

```

45  @FXML
46  private TableView<ItemPeca> tableViewItensDeVenda;
47  @FXML
48  private TableColumn<ItemPeca, Peca> tableColumnItemDeVendaPeca;
49  @FXML
50  private TableColumn<ItemPeca, Integer> tableColumnItemDeVendaQuantidade;
51  @FXML
52  private TableColumn<ItemPeca, Double> tableColumnItemDeVendaValor;
53  @FXML
54  private TextField textFieldVendaValor;
55  @FXML
56  private TextField textFieldVendaQuantidade;
57  @FXML
58  private Button buttonConfirmar;
59  @FXML
60  private Button buttonCancelar;
61  @FXML
62  private Button buttonAdicionar;
63
64  private List<Cliente> listClientes;
65  private List<Peca> listPecas;
66  private ObservableList<Cliente> observableListClientes;
67  private ObservableList<Peca> observableListPecas;
68  private ObservableList<ItemPeca> observableListItensDeVenda;
69
70  private final PecaDAO pecaDAO = new PecaDAO();
71  private final ClienteDAO clienteDAO = new ClienteDAO();
72
73  private Stage dialogStage;
74  private boolean buttonConfirmarClicked = false;
75  private VendaPeca venda;
76
77  @Override
78  public void initialize(URL url, ResourceBundle rb) {
79
80      carregarComboBoxClientes();
81      carregarComboBoxPecas();
82      tableColumnItemDeVendaPeca.setCellValueFactory(new PropertyValueFactory<>("nome"));
83      tableColumnItemDeVendaQuantidade.setCellValueFactory(new PropertyValueFactory<>("quantidadeVendida"));
84      tableColumnItemDeVendaValor.setCellValueFactory(new PropertyValueFactory<>("valor"));
85  }
86
87  public void carregarComboBoxClientes() {
88
89      try {
90          listClientes = clienteDAO.getAllCliente();
91      } catch (Exception ex) {
92          Logger.getLogger(VendaPecasDialogController.class.getName()).log(Level.SEVERE, null, ex);
93      }
94      System.out.println(listClientes.size());
95      observableListClientes = FXCollections.observableArrayList(listClientes);
96      comboBoxVendaCliente.setItems(observableListClientes);
97      System.out.println(comboBoxVendaCliente.getItems());
98  }
99
100  public void carregarComboBoxPecas() {
101      listPecas = pecaDAO.listar();
102      observableListPecas = FXCollections.observableArrayList(listPecas);
103      comboBoxVendaPeca.setItems(observableListPecas);
104  }
105
106  public Stage getDialogStage() {
107      return dialogStage;
108  }
109
110  public void setDialogStage(Stage dialogStage) {
111      this.dialogStage = dialogStage;
112  }
113
114  public boolean isButtonConfirmarClicked() {
115      return buttonConfirmarClicked;
116  }
117
118  public VendaPeca getVenda() {
119      return this.venda;
120  }
121
122  public void setVenda(VendaPeca venda) {
123      this.venda = venda;
124  }
125
126  @FXML
127  public void handleButtonAdicionar() {
128      Peca peca;
129      ItemPeca itemPeca = new ItemPeca();
130
131      if (comboBoxVendaPeca.getSelectionModel().getSelectedItem() != null) {
132          peca = (Peca) comboBoxVendaPeca.getSelectionModel().getSelectedItem();
133          int quantidade = Integer.parseInt(textFieldVendaQuantidade.getText());
134          if (peca.getQuantidade() >= quantidade) {
135              itemPeca.setPeca(peca);
136              itemPeca.setValor(peca.getValor() * quantidade);
137              itemPeca.setQuantidadeVendida(quantidade);
138

```

```

139         venda.getItemsVendidos().add(itemPeca);
140         venda.setValor(venda.getValor() + itemPeca.getValor());
141
142         observableListItensDeVenda = FXCollections.observableArrayList(itemPeca);
143         tableViewItensDeVenda.getItems().add(itemPeca);
144
145         textFieldVendaValor.setText(String.format("%.2f", venda.getValor()));
146     } else {
147         Alert alert = new Alert(Alert.AlertType.ERROR);
148         alert.setHeaderText("Problemas na escolha do produto!");
149         alert.setContentText("N o existe a quantidade de produtos dispon veis no estoque!");
150         alert.show();
151     }
152 }
153 }
154
155 public void handleButtonConfirmar() {
156
157     Cliente c = (Cliente) comboBoxVendaCliente.getSelectionModel().getSelectedItem();
158     venda.setCliente(c);
159     venda.setPago(checkBoxVendaPago.isSelected());
160
161     Date date = java.sql.Date.valueOf(datePickerVendaData.getValue());
162     venda.setDataVenda(date);
163
164     buttonConfirmarClicked = true;
165     dialogStage.close();
166 }
167
168 @FXML
169 public void handleButtonCancelar() {
170     getDialogStage().close();
171 }
172
173 }

```

Código 32: VendaPecasDialogController.java

5.5 Pacote Serviço

5.5.1 Model

```

1 package opencarshop.servico.model;
2
3 public class Servico {
4
5     private String descricao;
6     private double valorPadrao;
7     private boolean valorFixo;
8     private int id;
9
10    private String valorF;
11    private String valorP;
12
13    public Servico() {
14    }
15
16    public Servico(String desc, double valor, boolean bfixo) {
17
18        this.descricao = desc;
19        this.valorPadrao = valor;
20        this.valorFixo = bfixo;
21    }
22
23
24    public String getDescricao() {
25        return descricao;
26    }
27
28    public void setDescricao(String descricao) {
29        this.descricao = descricao;
30    }
31
32    public double getValorPadrao() {
33        return valorPadrao;
34    }
35
36    public void setValorPadrao(double valorPadrao) {
37        this.valorPadrao = valorPadrao;
38    }
39
40    public boolean getValorFixo() {
41        return valorFixo;
42    }
43
44    public void setValorFixo(boolean valorFixo) {

```

```

45     this.valorFixo = valorFixo;
46 }
47
48 public int getId() {
49     return id;
50 }
51
52 public void setId(int id) {
53     this.id = id;
54 }
55
56 public String getValorP() {
57     return valorP;
58 }
59
60 public void setValorP(double v) {
61     String v2 = Double.toString(v).replace(".", ",");
62     this.valorP = "R$␣" + v2;
63 }
64
65 public String getValorF() {
66     return valorF;
67 }
68
69 public void setValorF(boolean t) {
70     if (t) {
71         this.valorF = "Sim";
72     } else {
73         this.valorF = "N o";
74     }
75 }
76 }

```

Código 33: Servico.java

```

1 package opencarshop.servico.model;
2
3 import java.sql.Connection;
4 import java.sql.Date;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.util.ArrayList;
9 import java.util.List;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12 import opencarshop.util.ConexaoMySQL;
13 import opencarshop.util.Utilidades;
14
15 public class ServicoDAO {
16
17     public boolean insertServico(Servico servico) {
18
19         ConexaoMySQL c = new ConexaoMySQL();
20         Connection conn = null;
21         PreparedStatement stmt = null;
22         boolean retorno = true;
23         String query = "INSERT␣INTO␣Servico␣(descri o ,␣valorPadr o ,␣valorFixo)␣VALUES␣(?,␣?,␣?)";
24
25         try {
26             conn = c.conectar();
27             stmt = conn.prepareStatement(query);
28             stmt.setString(1, servico.getDescricao());
29             stmt.setDouble(2, servico.getValorPadrao());
30             stmt.setBoolean(3, servico.getValorFixo());
31             retorno = stmt.execute();
32         } catch (Exception e) {
33             e.printStackTrace();
34         }
35
36         return retorno;
37     }
38
39     public List<Servico> getAllServicos() throws Exception {
40
41         String query = "SELECT␣*␣FROM␣Servico";
42         List<Servico> retorno = new ArrayList<>();
43         Utilidades u = new Utilidades();
44         ConexaoMySQL c = new ConexaoMySQL();
45         Connection conn = null;
46         conn = c.conectar();
47
48         try {
49             PreparedStatement stmt = conn.prepareStatement(query);
50             ResultSet resultado = stmt.executeQuery();
51             while (resultado.next()) {
52                 Servico servico = new Servico();
53                 servico.setDescricao(resultado.getString("descri o"));
54                 servico.setValorPadrao(resultado.getDouble("valorPadr o"));
55                 servico.setValorFixo(resultado.getBoolean("valorFixo"));
56                 servico.setValorF(resultado.getBoolean("valorFixo"));
57                 servico.setValorP(resultado.getDouble("valorPadr o"));

```



```

58         servico.setId(resultado.getInt("id"));
59         retorno.add(servico);
60     }
61     } catch (Exception e) {
62         e.printStackTrace();
63     }
64 }
65 conn.close();
66 return retorno;
67 }
68
69 public Boolean alteraServico(Servico srv) throws SQLException {
70     String query = "UPDATE Servico SET descricao = ?, valorPadr o = ?, valorFixo = ? WHERE id = ?";
71
72     ConexaoMySQL c = new ConexaoMySQL();
73     Connection conn = null;
74     try {
75         conn = c.conectar();
76         PreparedStatement stmt = conn.prepareStatement(query);
77
78         stmt.setString(1, srv.getDescricao());
79         stmt.setDouble(2, srv.getValorPadrao());
80         stmt.setBoolean(3, srv.getValorFixo());
81         stmt.setInt(4, srv.getId());
82
83         stmt.execute();
84         conn.close();
85         return true;
86     } catch (Exception ex) {
87
88         Logger.getLogger(ServicoDAO.class.getName()).log(Level.SEVERE, null, ex);
89         return false;
90     }
91 }
92
93 }

```

Código 34: FuncionarioDAO.java

5.5.2 View

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import java.lang.*?>
4  <?import java.util.*?>
5  <?import javafx.scene.*?>
6  <?import javafx.scene.control.*?>
7  <?import javafx.scene.layout.*?>
8  <?import javafx.scene.text.*?>
9  <?import javafx.scene.image.*?>
10 <?import javafx.geometry.*?>
11
12 <AnchorPane id="AnchorPane" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
13     javafx.com/fxml/1" fx:controller="opencarshop.servico.controller.ServicoController">
14     <children>
15         <AnchorPane maxHeight="383.0" maxWidth="1000.0" minHeight="200.0" minWidth="350.0" prefHeight="383.0" prefWidth
16             ="457.0">
17             <children>
18                 <TextField fx:id="tfdescricao" alignment="TOP_LEFT" depthTest="ENABLE" layoutX="14.0" layoutY="31.0"
19                     nodeOrientation="LEFT_TO_RIGHT" promptText="Descrição do serviço">
20                     <tooltip>
21                         <Tooltip text="Usuário não precisa fazer login" />
22                     </tooltip>
23                     <font>
24                         <Font size="11.0" />
25                     </font>
26                     <cursor>
27                         <Cursor fx:constant="DEFAULT" />
28                     </cursor>
29                 </TextField>
30                 <TextField fx:id="tfvalor" layoutX="165.0" layoutY="31.0" promptText="ex: 50,00">
31                     <font>
32                         <Font size="11.0" />
33                     </font>
34                 </TextField>
35                 <Button fx:id="btnSignUp" layoutX="316.0" layoutY="74.0" mnemonicParsing="false" onAction="#
36                     cadastraServico" text="Cadastrar" />
37                 <Label layoutX="14.0" layoutY="14.0" text="Descrição: " />
38                 <Label layoutX="165.0" layoutY="14.0" text="Valor: " />
39                 <CheckBox fx:id="chkValue" layoutX="312.0" layoutY="35.0" mnemonicParsing="false" text="Valor Fixo" />
40                 <Label fx:id="labelErroServ" layoutX="106.0" layoutY="176.0" textFill="#ee0303">
41                     <font>
42                         <Font size="11.0" />
43                     </font>
44                 </Label>
45             </children>
46     </AnchorPane>
47 </children>

```

```

44
45
46 </AnchorPane>

```

Código 35: Cadastrar.fxml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.text.*?>
4  <?import java.lang.*?>
5  <?import java.util.*?>
6  <?import javafx.scene.*?>
7  <?import javafx.scene.control.*?>
8  <?import javafx.scene.layout.*?>
9
10 <AnchorPane id="AnchorPane" prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
    javafx.com/fxml/1" fx:controller="opencarshop.servico.controller.ServicoController">
11     <children>
12         <TableView fx:id="tbl_servico" prefHeight="650.0" prefWidth="446.0">
13             <columns>
14                 <TableColumn fx:id="col_descricao" minWidth="0.0" prefWidth="250.0" text="Descrição" />
15                 <TableColumn fx:id="col_valor" minWidth="0.0" prefWidth="110.0" text="Valor" />
16                 <TableColumn fx:id="col_tpvalor" minWidth="0.0" prefWidth="85.0" text="Valor fixo" />
17             </columns>
18         </TableView>
19         <TextField fx:id="tfdescricao" layoutX="632.0" layoutY="255.0" prefWidth="200.0" promptText="Descrição" />
20         <TextField fx:id="tfvalor" layoutX="632.0" layoutY="301.0" prefWidth="200.0" promptText="Valor" />
21         <CheckBox fx:id="chkValue" layoutX="632.0" layoutY="342.0" mnemonicParsing="false" text="Valor fixo" />
22         <Button layoutX="688.0" layoutY="399.0" mnemonicParsing="false" onAction="#alterarServico" text="Alterar" />
23         <TextField fx:id="tfid" editable="false" layoutX="632.0" layoutY="209.0" prefWidth="200.0" promptText="ID" />
24         <Label fx:id="confirmaAtualizacao" alignment="CENTER" layoutX="634.0" layoutY="365.0" prefHeight="17.0"
            prefWidth="200.0" />
25         <Label layoutX="634.0" layoutY="186.0" text="ID do serviço" />
26         <Label layoutX="634.0" layoutY="234.0" text="Descrição do serviço" />
27         <Label layoutX="634.0" layoutY="280.0" text="Valor do serviço" />
28     </children>
29 </AnchorPane>

```

Código 36: Buscar.fxml

5.5.3 Controller

```

1  package opencarshop.servico.controller;
2
3  import java.net.URL;
4  import java.sql.SQLException;
5  import java.util.List;
6  import java.util.ResourceBundle;
7  import java.util.regex.Pattern;
8  import javafx.collections.FXCollections;
9  import javafx.collections.ObservableList;
10 import javafx.event.ActionEvent;
11 import javafx.fxml.FXML;
12 import javafx.fxml.Initializable;
13 import javafx.scene.control.CheckBox;
14 import javafx.scene.control.Label;
15 import javafx.scene.control.TableColumn;
16 import javafx.scene.control.TableView;
17 import javafx.scene.control.TextField;
18 import javafx.scene.control.cell.PropertyValueFactory;
19 import opencarshop.servico.model.Servico;
20 import opencarshop.servico.model.ServicoDAO;
21
22 public class ServicoController implements Initializable {
23
24     /**
25      * Initializes the controller class.
26      */
27     //Variáveis
28     @FXML
29     private TextField tfdescricao;
30
31     @FXML
32     private TextField tfvalor;
33
34     @FXML
35     private CheckBox chkValue;
36
37     @FXML
38     private TextField tfid;
39
40     @FXML
41     private Label labelErroServ;
42

```

```

43     @FXML
44     private Label confirmaAtualizacao;
45
46     //Colunas da tabela listar servi os
47     @FXML
48     private TableColumn<Servico, String> col_descricao;
49     @FXML
50     private TableColumn<Servico, Double> col_valor;
51     @FXML
52     private TableColumn<Servico, String> col_tpvalor;
53     @FXML
54     private TableView<Servico> tbl_servico;
55
56     //metodos
57     @FXML
58     public void cadastraServico(ActionEvent event) {
59         //instanciando objeto servico para configuracao de atributos
60         Servico servico;
61         //instanciando objeto para insercao de objeto cadastrado no banco
62         ServicoDAO servDao = new ServicoDAO();
63
64         String descri = tfdescricao.getText();
65         String valor = tfvalor.getText();
66
67         //Testando valores do cadastro
68         if ((descri.length() > 45) || (!descri.equals(descri))) {
69             labelErroServ.setText("Descricao deve ter at 45 caracteres");
70         }
71         if (Pattern.matches("[a-zA-Z]+", valor) == true) {
72             labelErroServ.setText("por favor insira apenas numeros e virgula/ponto");
73         }
74
75         //passando para double
76         valor = valor.replace(",", ".");
77         Double valorDouble = Double.parseDouble(valor);
78
79         //Criando objeto servico
80         servico = new Servico(descri, valorDouble, chkValue.isSelected());
81
82         if (!servDao.insertServico(servico)) {
83             labelErroServ.setText("Servico cadastrado... retornando");
84         }
85     }
86
87 }
88
89
90 @FXML
91 private void carregaTabelaServico() throws Exception {
92
93     col_descricao.setCellValueFactory(new PropertyValueFactory<>("descricao"));
94     col_valor.setCellValueFactory(new PropertyValueFactory<>("valorP"));
95     col_tpvalor.setCellValueFactory(new PropertyValueFactory<>("valorF"));
96
97     ServicoDAO serv = new ServicoDAO();
98     List<Servico> listaServico = serv.getAllServicos();
99     ObservableList<Servico> observableListServico;
100
101     observableListServico = FXCollections.observableArrayList(listaServico);
102     tbl_servico.setItems(observableListServico);
103 }
104
105 @FXML
106 public void alterarServico(ActionEvent event) throws SQLException {
107     Servico srv = new Servico();
108
109     srv.setId(Integer.valueOf(tfid.getText()));
110     srv.setDescricao(tfdescricao.getText());
111     srv.setValorPadrao(Double.valueOf(tfvalor.getText()));
112     srv.setValorFixo(chkValue.isSelected());
113
114     ServicoDAO s = new ServicoDAO();
115
116     if (s.alteraServico(srv)) {
117         confirmaAtualizacao.setText("Alteracao realizada com sucesso!!");
118     } else {
119         confirmaAtualizacao.setText("Erro ao realizar a alteracao!!");
120     }
121 }
122
123
124 public void selecionarItemTabelaServico(Servico servico) {
125     if (servico.getDescricao() != null) {
126         tfdescricao.setText(servico.getDescricao());
127         tfvalor.setText(String.valueOf(servico.getValorPadrao()));
128         chkValue.setSelected(servico.getValorFixo());
129         tfid.setText(String.valueOf(servico.getId()));
130     }
131 }
132
133 @Override
134 public void initialize(URL url, ResourceBundle rb) {
135     try {
136         //System.out.println("Chamou");

```

```

137         carregaTabelaServico();
138         tbl_servico.getSelectionModel().selectedItemProperty().addListener(
139             (observable, oldValue, newValue) -> selecionarItemTabelaServico(newValue));
140     } catch (Exception ex) {
141     }
142 }
143 }
144 }
145 }
146 }

```

Código 37: ServicoController.java

5.6 Pacote Veiculo

5.6.1 Model

```

1 package opencarshop.veiculo.model;
2
3 public class Veiculo {
4
5     private String modelo;
6     private String versao;
7     private int ano;
8     private int quantidade;
9     private double valor;
10    private boolean opcionalVidrosEletricos;
11    private boolean opcionalTravasEletricas;
12    private boolean opcionalAr;
13    private boolean opcionalFarolNeblina;
14    private boolean opcionalAltoFalantes;
15
16    public Veiculo() {
17    }
18
19    public String getModelo() {
20        return modelo;
21    }
22
23    public void setModelo(String modelo) {
24        this.modelo = modelo;
25    }
26
27    public String getVersao() {
28        return versao;
29    }
30
31    public void setVersao(String versao) {
32        this.versao = versao;
33    }
34
35    public int getAno() {
36        return ano;
37    }
38
39    public void setAno(int ano) {
40        this.ano = ano;
41    }
42
43    public int getQuantidade() {
44        return quantidade;
45    }
46
47    public void setQuantidade(int quantidade) {
48        this.quantidade = quantidade;
49    }
50
51    public double getValor() {
52        return valor;
53    }
54
55    public void setValor(double valor) {
56        this.valor = valor;
57    }
58
59    public boolean isOpcionalVidrosEletricos() {
60        return opcionalVidrosEletricos;
61    }
62
63    public void setOpcionalVidrosEletricos(boolean opcionalVidrosEletricos) {
64        this.opcionalVidrosEletricos = opcionalVidrosEletricos;
65    }
66
67    public boolean isOpcionalTravasEletricas() {
68        return opcionalTravasEletricas;
69    }

```

```

70     }
71
72     public void setOpcionalTravasEletricas(boolean opcionalTravasEletricas) {
73         this.opcionalTravasEletricas = opcionalTravasEletricas;
74     }
75
76     public boolean isOpcionalAr() {
77         return opcionalAr;
78     }
79
80     public void setOpcionalAr(boolean opcionalAr) {
81         this.opcionalAr = opcionalAr;
82     }
83
84     public boolean isOpcionalFarolNeblina() {
85         return opcionalFarolNeblina;
86     }
87
88     public void setOpcionalFarolNeblina(boolean opcionalFarolNeblina) {
89         this.opcionalFarolNeblina = opcionalFarolNeblina;
90     }
91
92     public boolean isOpcionalAltoFalantes() {
93         return opcionalAltoFalantes;
94     }
95
96     public void setOpcionalAltoFalantes(boolean opcionalAltoFalantes) {
97         this.opcionalAltoFalantes = opcionalAltoFalantes;
98     }
99
100 }

```

Código 38: Veiculo.java

```

1  package opencarshop.veiculo.model;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.ResultSet;
6  import java.util.ArrayList;
7  import java.util.List;
8  import opencarshop.util.ConexaoMySQL;
9
10 public class VeiculoDAO {
11
12     public List<Veiculo> getAllVeiculo() throws Exception {
13         String query = "SELECT * FROM Veiculo";
14         List<Veiculo> retorno = new ArrayList<>();
15         ConexaoMySQL c = new ConexaoMySQL();
16         Connection conn = null;
17         conn = c.conectar();
18         try {
19             PreparedStatement stmt = conn.prepareStatement(query);
20             ResultSet resultado = stmt.executeQuery();
21             while (resultado.next()) {
22
23                 Veiculo veiculo = new Veiculo();
24                 veiculo.setModelo(resultado.getString("modelo"));
25                 veiculo.setAno(resultado.getInt("ano"));
26                 veiculo.setVersao(resultado.getString("versao"));
27                 veiculo.setQuantidade(resultado.getInt("qntd"));
28                 veiculo.setValor(resultado.getDouble("valor"));
29                 veiculo.setOpcionalAltoFalantes(resultado.getBoolean("altoFalantes"));
30                 veiculo.setOpcionalAr(resultado.getBoolean("ar"));
31                 veiculo.setOpcionalFarolNeblina(resultado.getBoolean("farolNeblina"));
32                 veiculo.setOpcionalTravasEletricas(resultado.getBoolean("travasEletricas"));
33                 veiculo.setOpcionalVidrosEletricos(resultado.getBoolean("vidrosEletricos"));
34
35                 retorno.add(veiculo);
36             }
37         } catch (Exception e) {
38             e.printStackTrace();
39         }
40         conn.close();
41         return retorno;
42     }
43
44     /*
45     //public Veiculo buscarVeiculo(String modelo)
46     public Veiculo buscarVeiculo()
47     {
48         ConexaoMySQL c = new ConexaoMySQL();
49         Connection conn = null;
50         PreparedStatement stmt = null;
51
52         // String query = "SELECT * FROM opencarshop.Veiculo Where modelo like ?";
53         String query = "SELECT * FROM opencarshop.Veiculo";
54         Veiculo veiculo = new Veiculo();
55
56         try
57         {
58             conn = c.conectar();

```

```

59         stmt = conn.prepareStatement(query);
60         // stmt.setString(1,modelo);
61
62         ResultSet resultado = stmt.executeQuery();
63
64         if(resultado.next())
65         {
66             veiculo.setModelo(resultado.getString("modelo"));
67             veiculo.setAno(resultado.getInt("ano"));
68             veiculo.setVersao(resultado.getString("versao"));
69             veiculo.setQuantidade(resultado.getInt("qntd"));
70             veiculo.setValor(resultado.getDouble("valor"));
71             veiculo.setOpcionalAltoFalantes(resultado.getBoolean("altoFalantes"));
72             veiculo.setOpcionalAr(resultado.getBoolean("ar"));
73             veiculo.setOpcionalFarolNeblina(resultado.getBoolean("farolNeblina"));
74             veiculo.setOpcionalTravasEletricas(resultado.getBoolean("travasEletricas"));
75             veiculo.setOpcionalVidrosEletricos(resultado.getBoolean("vidrosEletricos"));
76         }
77
78         conn.close();
79     }
80     catch(Exception e)
81     {
82         e.printStackTrace();
83     }
84     return veiculo;
85 }*/
86 public boolean insertVeiculo(Veiculo veiculo) {
87
88     ConexaoMySQL c = new ConexaoMySQL();
89     Connection conn = null;
90     PreparedStatement stmt = null;
91
92     //String query = "INSERT INTO teste(modelo) VALUES(?)";
93     String query = "INSERT INTO opencarshop.Veiculo(modelo,ano,versao,quantidade,valor,"
94         + "opcionalVidrosEletricos,opcionalTravasEletricas,opcionalAr,opcionalFarolNeblina,
95         opcionalAltoFalantes)"
96         + "VALUES(?,?,?,?,?,?,?,?,?)";
97
98     try {
99         conn = c.conectar();
100         conn.setAutoCommit(false);
101
102         stmt = conn.prepareStatement(query);
103
104         stmt.setString(1, veiculo.getModelo());
105         stmt.setInt(2, veiculo.getAno());
106         stmt.setString(3, veiculo.getVersao());
107         stmt.setInt(4, veiculo.getQuantidade());
108         stmt.setDouble(5, veiculo.getValor());
109         stmt.setBoolean(6, veiculo.isOpcionalVidrosEletricos());
110         stmt.setBoolean(7, veiculo.isOpcionalTravasEletricas());
111         stmt.setBoolean(8, veiculo.isOpcionalAr());
112         stmt.setBoolean(9, veiculo.isOpcionalFarolNeblina());
113         stmt.setBoolean(10, veiculo.isOpcionalAltoFalantes());
114
115         stmt.execute();
116         conn.commit();
117
118         conn.close();
119         return true;
120     } catch (Exception e) {
121         e.printStackTrace();
122         return false;
123     }
124 }

```

Código 39: VeiculoDAO.java

5.6.2 View

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import java.lang.*?>
4  <?import java.util.*?>
5  <?import javafx.scene.*?>
6  <?import javafx.scene.chart.*?>
7  <?import javafx.scene.control.*?>
8  <?import javafx.scene.layout.*?>
9  <?import javafx.scene.text.*?>
10
11
12  <AnchorPane id="AnchorPane" prefHeight="312.9609375" prefWidth="1024.0" xmlns:fx="http://javafx.com/fxml/1" xmlns="
13      http://javafx.com/javafx/2.2" fx:controller="opencarshop.veiculo.controller.VeiculoController">
14      <children>
15          <AnchorPane layoutX="0.0" layoutY="0.0" minHeight="0.0" minWidth="0.0" prefHeight="263.0" prefWidth="1024.0">
16              <children>

```

```

16         <TextField fx:id="tf_ano" layoutX="14.0" layoutY="65.0" prefHeight="25.0" prefWidth="170.0" promptText=
17         "Ano" />
18         <TextField fx:id="tf_versao" layoutX="195.0" layoutY="65.0" prefHeight="25.0" prefWidth="170.0"
19         promptText="Vers o" />
20         <TextField fx:id="tf_modelo" layoutX="14.0" layoutY="14.0" prefHeight="25.0" prefWidth="
21         350.0000999999975" promptText="Modelo" />
22         <TextField fx:id="tf_qntd" layoutX="14.0" layoutY="114.0" prefHeight="25.0" prefWidth="170.0"
23         promptText="Quantidade" />
24         <TextField fx:id="tf_valor" layoutX="195.0" layoutY="114.0" prefHeight="25.0" prefWidth="170.0"
25         promptText="Valor" />
26         <CheckBox fx:id="cb_VidrosEletricos" layoutX="28.0" layoutY="161.0" mnemonicParsing="false" text="
27         Vidros El tricos" />
28         <CheckBox fx:id="cb_AltoFalante" layoutX="277.0" layoutY="161.0" mnemonicParsing="false" text="Alto
29         Palantes" />
30         <CheckBox fx:id="cb_Ar" layoutX="154.0" layoutY="161.0" mnemonicParsing="false" text="Ar Con
31         dicionado" />
32         <CheckBox fx:id="cb_FarolNeblina" layoutX="154.0" layoutY="189.0" mnemonicParsing="false" text="Farol
33         Neblina" />
34         <CheckBox fx:id="cb_TravasEletricas" layoutX="28.0" layoutY="189.0" mnemonicParsing="false" text="
35         Travas Eletricas" />
36     </children>
37 </AnchorPane>
38 <Button fx:id="btn_cadastrar" layoutX="149.0" layoutY="232.0" mnemonicParsing="false" onAction="#"
39     cadastrarVeiculo" text="Cadastrar" />
40 <Label fx:id="lb_result" labelFor="$btn_cadastrar" layoutX="69.0" layoutY="263.0" prefHeight="36.0" prefWidth="
41     220.0" text="" textFill="#ff3333">
42     <font>
43         <Font size="14.0" />
44     </font>
45 </Label>
46 </children>
47 </AnchorPane>

```

Código 40: Cadastrar.fxml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import java.lang.*?>
4 <?import java.util.*?>
5 <?import javafx.scene.*?>
6 <?import javafx.scene.control.*?>
7 <?import javafx.scene.layout.*?>
8
9
10 <AnchorPane id="AnchorPane" prefHeight="768.0" prefWidth="1201.0" xmlns:fx="http://javafx.com/fxml/1" xmlns="http://
11     javafx.com/javafx/2.2" fx:controller="opencarshop.veiculo.controller.VeiculoController">
12     <children>
13         <TableView layoutX="-11.0" layoutY="0.0" prefHeight="650.0" prefWidth="602.0">
14             <columns>
15                 <TableColumn id="col_modelo" prefWidth="75.0" text="Modelo" />
16                 <TableColumn id="col_ano" prefWidth="75.0" text="Ano" />
17                 <TableColumn id="col_versao" prefWidth="75.0" text="Vers o" />
18                 <TableColumn id="col_qntd" prefWidth="75.0" text="Quantidade" />
19                 <TableColumn id="col_valor" prefWidth="75.0" text="Valor" />
20                 <TableColumn id="col_vidrosEletricos" maxWidth="5000.0" minWidth="10.0" prefWidth="100.0" text="Vidros
21                 El tricos" />
22                 <TableColumn id="col_travasEletricas" maxWidth="5000.0" minWidth="10.0" prefWidth="100.0" text="Travas
23                 El tricos" />
24                 <TableColumn id="col_ar" maxWidth="5000.0" minWidth="10.0" prefWidth="112.0" text="Ar Con
25                 dicionado" />
26                 <TableColumn id="col_farolNeblina" maxWidth="5000.0" minWidth="10.0" prefWidth="91.0" text="Farol
27                 Neblina" />
28                 <TableColumn id="col_altoFalantes" maxWidth="5000.0" minWidth="10.0" prefWidth="84.0" text="Alto
29                 Palantes" />
29             </columns>
30         </TableView>
31         <TextField id="tf_modelo" layoutX="624.0" layoutY="29.0" prefHeight="25.0" prefWidth="286.0" promptText="Modelo
32         " />
33         <Button id="btn_buscar" layoutX="767.0" layoutY="212.0" mnemonicParsing="false" text="Buscar" />
34     </children>
35 </AnchorPane>

```

Código 41: Buscar.fxml

5.6.3 Controller

```

1 package opencarshop.veiculo.controller;
2
3 import java.net.URL;
4 import java.util.List;
5 import java.util.Properties;
6 import java.util.ResourceBundle;
7 import javafx.collections.FXCollections;
8 import javafx.collections.ObservableList;
9 import javafx.event.ActionEvent;
10 import javafx.fxml.FXML;
11 import javafx.fxml.Initializable;

```

```

12 import javafx.scene.control.CheckBox;
13 import javafx.scene.control.Label;
14 import javafx.scene.control.TableColumn;
15 import javafx.scene.control.TableView;
16 import javafx.scene.control.TextField;
17 import javafx.scene.control.cell.PropertyValueFactory;
18 import opencarshop.veiculo.model.Veiculo;
19 import opencarshop.veiculo.model.VeiculoDAO;
20
21 public class VeiculoController implements Initializable {
22
23     //Variav is
24     @FXML
25     private Label lb_result;
26
27     @FXML
28     private TextField tf_modelo;
29
30     @FXML
31     private TextField tf_ano;
32
33     @FXML
34     private TextField tf_versao;
35
36     @FXML
37     private TextField tf_qntd;
38
39     @FXML
40     private TextField tf_valor;
41
42     @FXML
43     private CheckBox cb_VidrosEletricos;
44
45     @FXML
46     private CheckBox cb_TravasEletricas;
47
48     @FXML
49     private CheckBox cb_Ar;
50
51     @FXML
52     private CheckBox cb_FarolNeblina;
53
54     @FXML
55     private CheckBox cb_AltoFalante;
56
57     //TABELA VEICULO
58     @FXML
59     private TableColumn<Veiculo, String> col_modelo;
60
61     @FXML
62     private TableColumn<Veiculo, String> col_ano;
63
64     @FXML
65     private TableColumn<Veiculo, String> col_versao;
66
67     @FXML
68     private TableColumn<Veiculo, String> col_qntd;
69
70     @FXML
71     private TableColumn<Veiculo, String> col_valor;
72
73     @FXML
74     private TableColumn<Veiculo, String> col_vidrosEletricos;
75
76     @FXML
77     private TableColumn<Veiculo, String> col_travasEletricas;
78
79     @FXML
80     private TableColumn<Veiculo, String> col_ar;
81
82     @FXML
83     private TableColumn<Veiculo, String> col_farolNeblina;
84
85     @FXML
86     private TableColumn<Veiculo, String> col_altoFalantes;
87
88     // tabela
89     @FXML
90     private TableView<Veiculo> tbl_veiculo;
91
92     @FXML
93     private void cadastrarVeiculo(ActionEvent event) {
94
95         //instanciando objeto
96         Veiculo veiculo = new Veiculo();
97         //instancia objeto para inser o de objeto cadastrado no banco
98         VeiculoDAO veiculoDao = new VeiculoDAO();
99
100         veiculo.setModelo(tf_modelo.getText());
101         veiculo.setAno(Integer.parseInt((tf_ano.getText())));
102         veiculo.setVersao(tf_versao.getText());
103         veiculo.setQuantidade(Integer.parseInt(tf_qntd.getText()));
104         veiculo.setValor(Double.parseDouble(tf_valor.getText()));
105         veiculo.setOpcionalAltoFalantes(Boolean.parseBoolean(cb_AltoFalante.getText()));

```



```

106     veiculo.setOpcionalAr( Boolean.parseBoolean(cb_Ar.getText()));
107     veiculo.setOpcionalFarolNeblina( Boolean.parseBoolean(cb_FarolNeblina.getText()));
108     veiculo.setOpcionalTravasEletricas( Boolean.parseBoolean(cb_TravasEletricas.getText()));
109     veiculo.setOpcionalVidrosEletricos( Boolean.parseBoolean(cb_VidrosEletricos.getText()));
110
111     if (veiculoDao.insertVeiculo(veiculo)) {
112         lb_result.setText("Veículo cadastrado com sucesso");
113     } else {
114         lb_result.setText("Erro ao Cadastrar!! tente novamente.");
115     }
116 }
117
118 private void carregaTabelaVeiculo() throws Exception {
119     col_modelo.setCellValueFactory(new PropertyValueFactory<>("modelo"));
120     col_versao.setCellValueFactory(new PropertyValueFactory<>("versao"));
121     col_ano.setCellValueFactory(new PropertyValueFactory<>("ano"));
122     col_qntd.setCellValueFactory(new PropertyValueFactory<>("quantidade"));
123     col_valor.setCellValueFactory(new PropertyValueFactory<>("valor"));
124     col_vidrosEletricos.setCellValueFactory(new PropertyValueFactory<>("vidrosEletricos"));
125     col_travasEletricas.setCellValueFactory(new PropertyValueFactory<>("travasEletricas"));
126     col_ar.setCellValueFactory(new PropertyValueFactory<>("ar"));
127     col_farolNeblina.setCellValueFactory(new PropertyValueFactory<>("farolNeblina"));
128     col_altoFalantes.setCellValueFactory(new PropertyValueFactory<>("altoFalantes"));
129
130     VeiculoDAO veiculoDAO = new VeiculoDAO();
131     List<Veiculo> listaVeiculo = veiculoDAO.getAllVeiculo();
132     ObservableList<Veiculo> observableListVeiculo;
133
134     observableListVeiculo = FXCollections.observableArrayList(listaVeiculo);
135     tbl_veiculo.setItems(observableListVeiculo);
136 }
137
138 public void selecionarItemTabelaVeiculo(Veiculo veiculo) {
139     if (veiculo.getModelo() != null) {
140         tf_modelo.setText(veiculo.getModelo());
141         tf_versao.setText(veiculo.getVersao());
142         tf_ano.setText(String.valueOf(veiculo.getAno()));
143         tf_valor.setText(String.valueOf(veiculo.getValor()));
144         tf_qntd.setText(String.valueOf(veiculo.getQuantidade()));
145         cb_AltoFalante.setText(String.valueOf(veiculo.isOpcionalAltoFalantes()));
146         cb_Ar.setText(String.valueOf(veiculo.isOpcionalAr()));
147         cb_FarolNeblina.setText(String.valueOf(veiculo.isOpcionalFarolNeblina()));
148         cb_TravasEletricas.setText(String.valueOf(veiculo.isOpcionalTravasEletricas()));
149         cb_VidrosEletricos.setText(String.valueOf(veiculo.isOpcionalVidrosEletricos()));
150     }
151 }
152
153 @Override
154 public void initialize(URL url, ResourceBundle rb) {
155     try {
156         carregaTabelaVeiculo();
157         tbl_veiculo.getSelectionModel().selectedItemProperty().addListener(
158             (observable, oldValue, newValue) -> selecionarItemTabelaVeiculo(newValue));
159     } catch (Exception ex) {
160         //Logger.getLogger(FuncionarioController.class.getName()).log(Level.SEVERE, null, ex);
161     }
162 }
163 ;
164 }

```

Código 42: VeiculoController.java

5.7 Pacote Utilidades

```

1 package opencarshop.util;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5
6 public class ConexaoMySQL {
7
8     //Nome do usuário do mysql
9     private final String USERNAME = "desenvolvimento";
10    //Senha do mysql
11    private final String PASSWORD = "ds123";
12    //Dados de caminho, porta e nome da base de dados que irá ser feita a conexão
13    private final String DATABASE_URL = "jdbc:mysql://tharlysson.com:3306/opencarshop";
14
15    /**
16     * Cria uma conexão com o banco de dados MySQL utilizando o nome de usuário
17     * e senha fornecidos
18     *
19     * @param username
20     * @param senha
21     * @return uma conexão com o banco de dados
22     * @throws Exception
23     */
24    public Connection conectar() throws Exception {

```

```

25     Class.forName("com.mysql.jdbc.Driver"); //Faz com que a classe seja carregada pela JVM
26     //Cria a conex o com o banco de dados
27     Connection connection = DriverManager.getConnection(DATABASE_URL, USERNAME, PASSWORD);
28     return connection;
29 }
30 }

```

Código 43: ConexaoMySQL.java

```

1 package opencarshop.util;
2
3 import java.time.Instant;
4 import java.time.LocalDate;
5 import java.time.LocalDateTime;
6 import java.time.ZoneId;
7 import java.util.Date;
8 import javax.swing.text.DatePicker;
9
10 public class Utilidades {
11
12     /**
13      * Converte LocalDate para Date
14      *
15      * @param datePicker
16      * @return date
17      */
18     public Date toDate(LocalDate datePicker) {
19         if (datePicker == null) {
20             return null;
21         }
22         LocalDate ld = datePicker;
23         Instant instant = ld.atStartOfDay().atZone(ZoneId.systemDefault()).toInstant();
24         Date date = Date.from(instant);
25
26         return date;
27     }
28
29     /**
30      * Converte Date para LocalDate
31      *
32      * @param d
33      * @return LocalDate
34      */
35     public LocalDate toLocalDate(Date d) {
36         Instant instant = Instant.ofEpochMilli(d.getTime());
37         LocalDateTime localDate = LocalDateTime.ofInstant(instant, ZoneId.systemDefault()).toLocalDate();
38         return localDate;
39     }
40 }
41 }

```

Código 44: Utilidades.java

5.8 Outros

```

1 package opencarshop;
2
3 public class Endereco {
4
5     private String CEP;
6     private String estado;
7     private String cidade;
8     private String bairro;
9     private String rua;
10    private int numero;
11    private String complemento;
12    private Character tipo;
13
14    public String getCEP() {
15        return CEP;
16    }
17
18    public Character getTipo() {
19        return tipo;
20    }
21
22    public void setTipo(Character tipo) {
23        this.tipo = tipo;
24    }
25
26    public void setCEP(String CEP) {
27        this.CEP = CEP;
28    }
29
30    public String getEstado() {

```

```

31     return estado;
32 }
33
34 public void setEstado(String estado) {
35     this.estado = estado;
36 }
37
38 public String getCidade() {
39     return cidade;
40 }
41
42 public void setCidade(String cidade) {
43     this.cidade = cidade;
44 }
45
46 public String getBairro() {
47     return bairro;
48 }
49
50 public void setBairro(String bairro) {
51     this.bairro = bairro;
52 }
53
54 public String getRua() {
55     return rua;
56 }
57
58 public void setRua(String rua) {
59     this.rua = rua;
60 }
61
62 public int getNumero() {
63     return numero;
64 }
65
66 public void setNumero(int numero) {
67     this.numero = numero;
68 }
69
70 public String getComplemento() {
71     return complemento;
72 }
73
74 public void setComplemento(String complemento) {
75     this.complemento = complemento;
76 }
77
78 }

```

Código 45: Endereco.java

```

1 package opencarshop;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Parent;
6 import javafx.scene.Scene;
7 import javafx.scene.image.Image;
8 import javafx.stage.Stage;
9
10 public class OpenCarShop extends Application {
11
12     @Override
13     public void start(Stage stage) throws Exception {
14         Parent root = FXMLLoader.load(getClass().getResource("funcionario/view/Autenticar.fxml"));
15
16         Scene scene = new Scene(root);
17
18         stage.setScene(scene);
19         stage.getIcons().add(new Image("recursos/icones/account-circle-white.png"));
20         stage.setTitle("Autentica o");
21         stage.show();
22     }
23
24     /**
25      * @param args the command line arguments
26      */
27     public static void main(String[] args) {
28         launch(args);
29     }
30
31 }

```

Código 46: OpenCarShop.java

```

1 package opencarshop;
2
3 import java.io.IOException;
4 import java.net.URL;

```

```

5 import java.util.ResourceBundle;
6 import javafx.event.ActionEvent;
7 import javafx.fxml.FXML;
8 import javafx.fxml.FXMLLoader;
9 import javafx.fxml.Initializable;
10 import javafx.scene.control.Label;
11 import javafx.scene.layout.AnchorPane;
12 import javafx.scene.layout.StackPane;
13 import javafx.scene.text.Text;
14
15 public class TelaPrincipalController implements Initializable {
16
17     @FXML
18     private StackPane acContent;
19
20     @FXML
21     private Text tx_info;
22
23     @FXML
24     private void cadastrarFuncionario(ActionEvent event) {
25         System.out.println("CadastrarFuncionario");
26         tx_info.setText("CadastrarFuncionario");
27         FXMLLoader fxmllLoader = new FXMLLoader();
28         try {
29             fxmllLoader.load(getClass().getResource("/opencarshop/funcionario/view/Cadastrar.fxml").openStream());
30         } catch (IOException e) {
31
32         }
33         AnchorPane root = fxmllLoader.getRoot();
34         acContent.getChildren().clear();
35         acContent.getChildren().add(root);
36     }
37
38     @FXML
39     private void listarFuncionario(ActionEvent event) {
40         System.out.println("BuscarFuncionario");
41         tx_info.setText("ListarFuncionario");
42         FXMLLoader fxmllLoader = new FXMLLoader();
43         try {
44             fxmllLoader.load(getClass().getResource("/opencarshop/funcionario/view/Buscar.fxml").openStream());
45         } catch (IOException e) {
46
47         }
48         AnchorPane root = fxmllLoader.getRoot();
49         acContent.getChildren().clear();
50         acContent.getChildren().add(root);
51     }
52
53     @FXML
54     private void cadastrarFornecedor(ActionEvent event) {
55         System.out.println("CadastrarFornecedor");
56         tx_info.setText("CadastrarFornecedor");
57         FXMLLoader fxmllLoader = new FXMLLoader();
58         try {
59             fxmllLoader.load(getClass().getResource("/opencarshop/fornecedor/view/Cadastrar.fxml").openStream());
60         } catch (IOException e) {
61
62         }
63         AnchorPane root = fxmllLoader.getRoot();
64         acContent.getChildren().clear();
65         acContent.getChildren().add(root);
66     }
67
68     @FXML
69     private void listarFornecedor(ActionEvent event) {
70         System.out.println("BuscarFornecedor");
71         tx_info.setText("ListarFornecedor");
72         FXMLLoader fxmllLoader = new FXMLLoader();
73         try {
74             fxmllLoader.load(getClass().getResource("/opencarshop/fornecedor/view/Buscar.fxml").openStream());
75         } catch (IOException e) {
76
77         }
78         AnchorPane root = fxmllLoader.getRoot();
79         acContent.getChildren().clear();
80         acContent.getChildren().add(root);
81     }
82
83     @FXML
84     private void cadastrarCliente(ActionEvent event) {
85         System.out.println("CadastrarCliente");
86         tx_info.setText("CadastrarCliente");
87         FXMLLoader fxmllLoader = new FXMLLoader();
88         try {
89             fxmllLoader.load(getClass().getResource("/opencarshop/cliente/view/Cadastrar.fxml").openStream());
90         } catch (IOException e) {
91
92         }
93         AnchorPane root = fxmllLoader.getRoot();
94         acContent.getChildren().clear();
95         acContent.getChildren().add(root);
96     }
97
98     @FXML

```

```

99     private void listarCliente(ActionEvent event) {
100         System.out.println("Buscar_Cliente");
101         tx_info.setText("Listar_Cliente");
102         FXMLLoader fxmlloader = new FXMLLoader();
103         try {
104             fxmlloader.load(getClass().getResource("/opencarshop/cliente/view/Buscar.fxml").openStream());
105         } catch (IOException e) {
106
107         }
108         AnchorPane root = fxmlloader.getRoot();
109         acContent.getChildren().clear();
110         acContent.getChildren().add(root);
111     }
112
113     @FXML
114     private void cadastrarServico(ActionEvent event) {
115         System.out.println("Cadastrar_Servico");
116         tx_info.setText("Cadastrar_Servico");
117         FXMLLoader fxmlloader = new FXMLLoader();
118         try {
119             fxmlloader.load(getClass().getResource("/opencarshop/servico/view/Cadastrar.fxml").openStream());
120         } catch (IOException e) {
121
122         }
123         AnchorPane root = fxmlloader.getRoot();
124         acContent.getChildren().clear();
125         acContent.getChildren().add(root);
126     }
127
128     @FXML
129     private void listarServico(ActionEvent event) {
130         System.out.println("Buscar_Servico");
131         tx_info.setText("Listar_Servico");
132         FXMLLoader fxmlloader = new FXMLLoader();
133         try {
134             fxmlloader.load(getClass().getResource("/opencarshop/servico/view/Buscar.fxml").openStream());
135         } catch (IOException e) {
136
137         }
138         AnchorPane root = fxmlloader.getRoot();
139         acContent.getChildren().clear();
140         acContent.getChildren().add(root);
141     }
142
143     @FXML
144     private void cadastrarVeiculo(ActionEvent event) {
145         System.out.println("Cadastrar_Veiculo");
146         tx_info.setText("Cadastrar_Veiculo");
147         FXMLLoader fxmlloader = new FXMLLoader();
148         try {
149             fxmlloader.load(getClass().getResource("/opencarshop/veiculo/view/Cadastrar.fxml").openStream());
150         } catch (IOException e) {
151
152         }
153         AnchorPane root = fxmlloader.getRoot();
154         acContent.getChildren().clear();
155         acContent.getChildren().add(root);
156     }
157
158     @FXML
159     private void listarVeiculo(ActionEvent event) {
160         System.out.println("Buscar_Veiculo");
161         tx_info.setText("Listar_Veiculo");
162         FXMLLoader fxmlloader = new FXMLLoader();
163         try {
164             fxmlloader.load(getClass().getResource("/opencarshop/veiculo/view/Buscar.fxml").openStream());
165         } catch (IOException e) {
166
167         }
168         AnchorPane root = fxmlloader.getRoot();
169         acContent.getChildren().clear();
170         acContent.getChildren().add(root);
171     }
172
173     @FXML
174     private void cadastrarPeca(ActionEvent event) {
175         System.out.println("Cadastrar_Peca");
176         tx_info.setText("Cadastrar_Peca");
177         FXMLLoader fxmlloader = new FXMLLoader();
178         try {
179             fxmlloader.load(getClass().getResource("/opencarshop/peca/view/CadastroPeca.fxml").openStream());
180         } catch (IOException e) {
181
182         }
183         AnchorPane root = fxmlloader.getRoot();
184         acContent.getChildren().clear();
185         acContent.getChildren().add(root);
186     }
187
188     @FXML
189     private void listarPeca(ActionEvent event) {
190         System.out.println("Buscar_Peca");
191         tx_info.setText("Listar_Peca");
192         FXMLLoader fxmlloader = new FXMLLoader();

```

```

193     try {
194         fxmllLoader.load(getClass().getResource("/opencarshop/peca/view/Buscar.fxml").openStream());
195     } catch (IOException e) {
196
197     }
198     AnchorPane root = fxmllLoader.getRoot();
199     acContent.getChildren().clear();
200     acContent.getChildren().add(root);
201 }
202
203 @FXML
204 private void venderPeca(ActionEvent event) {
205     System.out.println("Vender_␣Peca");
206     tx_info.setText("Vender_␣Pe a");
207     FXMMLoader fxmllLoader = new FXMMLoader();
208     try {
209         fxmllLoader.load(getClass().getResource("/opencarshop/peca/view/VendaPecas.fxml").openStream());
210     } catch (IOException e) {
211
212     }
213     AnchorPane root = fxmllLoader.getRoot();
214     acContent.getChildren().clear();
215     acContent.getChildren().add(root);
216
217 }
218
219 @FXML
220 private void relatorioVendaPecaMes(ActionEvent event) {
221     System.out.println("Relatorio_␣Peca/Mes");
222     tx_info.setText("Relatorio_␣Peca/Mes");
223
224     FXMMLoader fxmllLoader = new FXMMLoader();
225     try {
226         fxmllLoader.load(getClass().getResource("/opencarshop/peca/view/GraficosVendasPorMes.fxml").openStream());
227     } catch (IOException e) {
228
229     }
230     AnchorPane root = fxmllLoader.getRoot();
231     acContent.getChildren().clear();
232     acContent.getChildren().add(root);
233 }
234
235 @Override
236 public void initialize(URL url, ResourceBundle rb) {
237     // TODO
238 }
239
240 }

```

Código 47: TelaPrincipalController.java

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.effect.*?>
4  <?import javafx.scene.text.*?>
5  <?import java.lang.*?>
6  <?import java.util.*?>
7  <?import javafx.scene.*?>
8  <?import javafx.scene.control.*?>
9  <?import javafx.scene.layout.*?>
10
11  <AnchorPane id="AnchorPane" prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
12      javafx.com/fxml/1" fx:controller="opencarshop.TelaPrincipalController">
13      <children>
14          <StackPane prefHeight="740.0" prefWidth="1024.0">
15              <children>
16                  <BorderPane prefHeight="200.0" prefWidth="200.0">
17                      <top>
18                          <MenuBar prefWidth="25.0" BorderPane.alignment="CENTER">
19                              <menus>
20                                  <Menu mnemonicParsing="false" text="Cliente">
21                                      <items>
22                                          <MenuItem mnemonicParsing="false" onAction="#cadastrarCliente" text="Cadastrar"
23                                              />
24                                          <MenuItem mnemonicParsing="false" onAction="#listarCliente" text="Listar" />
25                                      </items>
26                                  </Menu>
27                                  <Menu mnemonicParsing="false" text="Servi o">
28                                      <items>
29                                          <MenuItem mnemonicParsing="false" onAction="#cadastrarServico" text="Cadastrar"
30                                              />
31                                          <MenuItem mnemonicParsing="false" onAction="#listarServico" text="Listar" />
32                                      </items>
33                                  </Menu>
34                                  <Menu mnemonicParsing="false" text="Pe a">
35                                      <items>
36                                          <MenuItem mnemonicParsing="false" onAction="#venderPeca" text="Vender" />
37                                          <SeparatorMenuItem mnemonicParsing="false" />
38                                          <MenuItem mnemonicParsing="false" onAction="#cadastrarPeca" text="Gerenciar" />
39                                          <SeparatorMenuItem mnemonicParsing="false" />
40                                          <MenuItem mnemonicParsing="false" onAction="#relatorioVendaPecaMes" text="
41      Relat rio_␣Venda/Mes" />

```

```

38         </items>
39     </Menu>
40     <Menu mnemonicParsing="false" text="Veiculo">
41         <items>
42             <MenuItem mnemonicParsing="false" text="Vender" />
43             <SeparatorMenuItem mnemonicParsing="false" />
44             <MenuItem mnemonicParsing="false" onAction="#cadastrarVeiculo" text="Cadastrar" />
45         </items>
46     </Menu>
47     <Menu mnemonicParsing="false" text="Funcion rio">
48         <items>
49             <MenuItem mnemonicParsing="false" onAction="#cadastrarFuncionario" text="Cadastrar" />
50             <MenuItem mnemonicParsing="false" onAction="#listarFuncionario" text="Listar" />
51         </items>
52     </Menu>
53     <Menu mnemonicParsing="false" text="Fornecedor">
54         <items>
55             <MenuItem mnemonicParsing="false" onAction="#cadastrarFornecedor" text="Cadastrar" />
56             <MenuItem mnemonicParsing="false" onAction="#listarFornecedor" text="Listar" />
57         </items>
58     </Menu>
59 </menus>
60 </MenuBar>
61 </top>
62 <center>
63     <StackPane fx:id="acContent" prefHeight="646.0" prefWidth="1024.0" BorderPane.alignment="CENTER" />
64 </center>
65 <bottom>
66     <ToolBar prefHeight="14.0" prefWidth="1024.0" BorderPane.alignment="CENTER">
67         <items>
68             <Text id="tx_info" fx:id="tx_info" strokeType="OUTSIDE" strokeWidth="0.0" text="OpenCarShop">
69                 <font>
70                     <Font name="System_Bold" size="18.0" />
71                 </font>
72             </Text>
73         </items>
74     </ToolBar>
75 </bottom>
76 </BorderPane>
77 </children>
78 </StackPane>
79 </children>
80 </AnchorPane>

```

Código 48: TelaPrincipal.fxml

5.9 SQL

```

1  -- MySQL Script generated by MySQL Workbench
2  -- 10/30/16 15:58:11
3  -- Model: New Model      Version: 1.0
4  -- MySQL Workbench Forward Engineering
5
6  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
7  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
8  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
9
10
11  -- Schema opencarshop
12
13  DROP SCHEMA IF EXISTS 'opencarshop' ;
14
15  -- Schema opencarshop
16
17  CREATE SCHEMA IF NOT EXISTS 'opencarshop' DEFAULT CHARACTER SET utf8 ;
18  USE 'opencarshop' ;
19
20
21  -- Table 'opencarshop`.`Endereco`
22
23  DROP TABLE IF EXISTS 'opencarshop`.`Endereco' ;
24
25  CREATE TABLE IF NOT EXISTS 'opencarshop`.`Endereco` (
26      'id' INT NOT NULL AUTO_INCREMENT,
27      'cep' VARCHAR(9) NULL DEFAULT '-----',
28      'estado' CHAR(2) NOT NULL,
29      'cidade' VARCHAR(45) NOT NULL,
30      'bairro' VARCHAR(45) NOT NULL,
31      'rua' VARCHAR(45) NULL,
32      'numero' INT NULL,
33

```

```

34      'complemento' VARCHAR(45) NULL,
35      'tipo' CHAR(1) NULL,
36      PRIMARY KEY ('id'))
37 ENGINE = InnoDB;
38
39
40 -----
41 -- Table 'opencarshop`.`Funcionario`
42 -----
43 DROP TABLE IF EXISTS `opencarshop`.`Funcionario` ;
44
45 CREATE TABLE IF NOT EXISTS `opencarshop`.`Funcionario` (
46   'cpf' VARCHAR(15) NOT NULL,
47   'nome' VARCHAR(50) NOT NULL,
48   'senha' VARCHAR(20) NOT NULL,
49   'dataNascimento' DATE NULL,
50   'email' VARCHAR(50) NULL,
51   'telefone1' VARCHAR(45) NULL,
52   'telefone2' VARCHAR(45) NULL,
53   'endereco' INT NOT NULL,
54   'ativo' TINYINT(1) NULL DEFAULT 1,
55   PRIMARY KEY ('cpf'),
56   INDEX 'fk_Funcionario_Endereco1_idx' ('endereco' ASC),
57   CONSTRAINT 'fk_Funcionario_Endereco1'
58     FOREIGN KEY ('endereco')
59     REFERENCES `opencarshop`.`Endereco` ('id')
60     ON DELETE NO ACTION
61     ON UPDATE NO ACTION)
62 ENGINE = InnoDB;
63
64
65 -----
66 -- Table 'opencarshop`.`Contrato`
67 -----
68 DROP TABLE IF EXISTS `opencarshop`.`Contrato` ;
69
70 CREATE TABLE IF NOT EXISTS `opencarshop`.`Contrato` (
71   'id' INT NOT NULL AUTO_INCREMENT,
72   'carga' CHAR(1) NULL,
73   'salario' DECIMAL(10,2) NULL,
74   'dataInicio' DATE NULL,
75   'dataTermino' DATE NULL,
76   'funcionario' VARCHAR(15) NOT NULL,
77   PRIMARY KEY ('id', 'funcionario'),
78   INDEX 'fk_Contrato_Funcionario1_idx' ('funcionario' ASC),
79   CONSTRAINT 'fk_Contrato_Funcionario1'
80     FOREIGN KEY ('funcionario')
81     REFERENCES `opencarshop`.`Funcionario` ('cpf')
82     ON DELETE NO ACTION
83     ON UPDATE NO ACTION)
84 ENGINE = InnoDB;
85
86
87 -----
88 -- Table 'opencarshop`.`Cliente`
89 -----
90 DROP TABLE IF EXISTS `opencarshop`.`Cliente` ;
91
92 CREATE TABLE IF NOT EXISTS `opencarshop`.`Cliente` (
93   'cpf' VARCHAR(15) NOT NULL,
94   'nome' VARCHAR(50) NOT NULL,
95   'dataNascimento' DATETIME NULL,
96   'email' VARCHAR(50) NULL,
97   'telefone1' VARCHAR(45) GENERATED ALWAYS AS (),
98   'telefone2' VARCHAR(45) NULL,
99   'ativo' TINYINT(1) NULL DEFAULT 1,
100  'endereco' INT NOT NULL,
101  PRIMARY KEY ('cpf'),
102  INDEX 'fk_Cliente_Endereco1_idx' ('endereco' ASC),
103  CONSTRAINT 'fk_Cliente_Endereco1'
104    FOREIGN KEY ('endereco')
105    REFERENCES `opencarshop`.`Endereco` ('id')
106    ON DELETE NO ACTION
107    ON UPDATE NO ACTION)
108 ENGINE = InnoDB;
109
110
111 -----
112 -- Table 'opencarshop`.`Fornecedor`
113 -----
114 DROP TABLE IF EXISTS `opencarshop`.`Fornecedor` ;
115
116 CREATE TABLE IF NOT EXISTS `opencarshop`.`Fornecedor` (
117   'cnpj' VARCHAR(15) NOT NULL,
118   'razaoSocial' VARCHAR(50) NOT NULL,
119   'descricao' VARCHAR(45) NULL,
120   'email' VARCHAR(50) NULL,
121   'telefone1' VARCHAR(45) GENERATED ALWAYS AS (),
122   'telefone2' VARCHAR(45) NULL,
123   'endereco' INT NOT NULL,
124   'ativo' TINYINT(1) NULL DEFAULT 1,
125   PRIMARY KEY ('cnpj'),
126   INDEX 'fk_Fornecedor_Endereco_idx' ('endereco' ASC),
127   CONSTRAINT 'fk_Fornecedor_Endereco'

```



```

128     FOREIGN KEY ('endereco')
129     REFERENCES 'opencarshop'.'.Endereco' ('id')
130     ON DELETE NO ACTION
131     ON UPDATE NO ACTION)
132 ENGINE = InnoDB;
133
134
135 -----
136 -- Table 'opencarshop'.'.Veiculo'
137 -----
138 DROP TABLE IF EXISTS 'opencarshop'.'.Veiculo' ;
139
140 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.Veiculo' (
141     'id' INT NOT NULL AUTO_INCREMENT,
142     'modelo' VARCHAR(45) NOT NULL,
143     'ano' YEAR NOT NULL,
144     'versao' VARCHAR(45) NOT NULL,
145     'opcionalVidrosEletricos' TINYINT(1) NULL DEFAULT 0,
146     'opcionalTravasEletricas' TINYINT(1) NULL DEFAULT 0,
147     'opcionalAr' TINYINT(1) NULL DEFAULT 0,
148     'opcionalFarolNebolina' TINYINT(1) NULL DEFAULT 0,
149     'opcionalAltoFalantes' TINYINT(1) NULL DEFAULT 0,
150     'quantidade' INT NULL,
151     'valor' DECIMAL(10,2) NULL,
152     PRIMARY KEY ('id'))
153 ENGINE = InnoDB;
154
155
156 -----
157 -- Table 'opencarshop'.'.Peca'
158 -----
159 DROP TABLE IF EXISTS 'opencarshop'.'.Peca' ;
160
161 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.Peca' (
162     'id' INT NOT NULL,
163     'descricao' VARCHAR(45) NULL,
164     'valor' VARCHAR(45) NULL,
165     'tipo' CHAR(1) NULL,
166     'quantidade' INT NULL,
167     PRIMARY KEY ('id'))
168 ENGINE = InnoDB;
169
170
171 -----
172 -- Table 'opencarshop'.'.Servico'
173 -----
174 DROP TABLE IF EXISTS 'opencarshop'.'.Servico' ;
175
176 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.Servico' (
177     'id' INT NOT NULL AUTO_INCREMENT,
178     'descricao' VARCHAR(45) NOT NULL,
179     'valorPadrao' DECIMAL(10,2) NOT NULL,
180     'valorFixo' TINYINT(1) NOT NULL,
181     PRIMARY KEY ('id'))
182 ENGINE = InnoDB;
183
184
185 -----
186 -- Table 'opencarshop'.'.OrcamentoServico'
187 -----
188 DROP TABLE IF EXISTS 'opencarshop'.'.OrcamentoServico' ;
189
190 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.OrcamentoServico' (
191     'id' INT NOT NULL,
192     'placa' VARCHAR(45) NOT NULL,
193     'data' DATE NULL,
194     'funcionario' VARCHAR(15) NOT NULL,
195     PRIMARY KEY ('id', 'funcionario'),
196     INDEX 'fk_OrcamentoServico_Funcionario1_idx' (('funcionario' ASC),
197     CONSTRAINT 'fk_OrcamentoServico_Funcionario1'
198     FOREIGN KEY ('funcionario')
199     REFERENCES 'opencarshop'.'.Funcionario' ('cpf')
200     ON DELETE NO ACTION
201     ON UPDATE NO ACTION)
202 ENGINE = InnoDB;
203
204
205 -----
206 -- Table 'opencarshop'.'.ServicoSelecionado'
207 -----
208 DROP TABLE IF EXISTS 'opencarshop'.'.ServicoSelecionado' ;
209
210 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.ServicoSelecionado' (
211     'orcamento' INT NOT NULL,
212     'servico' INT NOT NULL,
213     PRIMARY KEY ('orcamento', 'servico'),
214     INDEX 'fk_OrcamentoServico_has_Servico_Servico1_idx' (('servico' ASC),
215     INDEX 'fk_OrcamentoServico_has_Servico_OrcamentoServico1_idx' (('orcamento' ASC),
216     CONSTRAINT 'fk_OrcamentoServico_has_Servico_OrcamentoServico1'
217     FOREIGN KEY ('orcamento')
218     REFERENCES 'opencarshop'.'.OrcamentoServico' ('id')
219     ON DELETE NO ACTION
220     ON UPDATE NO ACTION,
221     CONSTRAINT 'fk_OrcamentoServico_has_Servico_Servico1'

```

```

222     FOREIGN KEY ('servico')
223     REFERENCES 'opencarshop'.'.Servico' ('id')
224     ON DELETE NO ACTION
225     ON UPDATE NO ACTION)
226 ENGINE = InnoDB;
227
228
229 -----
230 -- Table 'opencarshop'.'.VeiculoFornecido'
231 -----
232 DROP TABLE IF EXISTS 'opencarshop'.'.VeiculoFornecido' ;
233
234 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.VeiculoFornecido' (
235     'Fornecedor_cnpj' VARCHAR(15) NOT NULL,
236     'Veiculo_id' INT NOT NULL,
237     PRIMARY KEY ('Fornecedor_cnpj', 'Veiculo_id'),
238     INDEX 'fk_Fornecedor_has_Veiculo_Veiculo1_idx' ('Veiculo_id' ASC),
239     INDEX 'fk_Fornecedor_has_Veiculo_Fornecedor1_idx' ('Fornecedor_cnpj' ASC),
240     CONSTRAINT 'fk_Fornecedor_has_Veiculo_Fornecedor1'
241     FOREIGN KEY ('Fornecedor_cnpj')
242     REFERENCES 'opencarshop'.'.Fornecedor' ('cnpj')
243     ON DELETE NO ACTION
244     ON UPDATE NO ACTION,
245     CONSTRAINT 'fk_Fornecedor_has_Veiculo_Veiculo1'
246     FOREIGN KEY ('Veiculo_id')
247     REFERENCES 'opencarshop'.'.Veiculo' ('id')
248     ON DELETE NO ACTION
249     ON UPDATE NO ACTION)
250 ENGINE = InnoDB;
251
252
253 -----
254 -- Table 'opencarshop'.'.PecaFornecida'
255 -----
256 DROP TABLE IF EXISTS 'opencarshop'.'.PecaFornecida' ;
257
258 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.PecaFornecida' (
259     'fornecedor' VARCHAR(15) NOT NULL,
260     'peca' INT NOT NULL,
261     PRIMARY KEY ('fornecedor', 'peca'),
262     INDEX 'fk_Fornecedor_has_Peca_Peca1_idx' ('peca' ASC),
263     INDEX 'fk_Fornecedor_has_Peca_Fornecedor1_idx' ('fornecedor' ASC),
264     CONSTRAINT 'fk_Fornecedor_has_Peca_Fornecedor1'
265     FOREIGN KEY ('fornecedor')
266     REFERENCES 'opencarshop'.'.Fornecedor' ('cnpj')
267     ON DELETE NO ACTION
268     ON UPDATE NO ACTION,
269     CONSTRAINT 'fk_Fornecedor_has_Peca_Peca1'
270     FOREIGN KEY ('peca')
271     REFERENCES 'opencarshop'.'.Peca' ('id')
272     ON DELETE NO ACTION
273     ON UPDATE NO ACTION)
274 ENGINE = InnoDB;
275
276
277 -----
278 -- Table 'opencarshop'.'.PecaNecessarias'
279 -----
280 DROP TABLE IF EXISTS 'opencarshop'.'.PecaNecessarias' ;
281
282 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.PecaNecessarias' (
283     'peca' INT NOT NULL,
284     'orcamentoServico' INT NOT NULL,
285     PRIMARY KEY ('peca', 'orcamentoServico'),
286     INDEX 'fk_Peca_has_OrcamentoServico_OrcamentoServico1_idx' ('orcamentoServico' ASC),
287     INDEX 'fk_Peca_has_OrcamentoServico_Peca1_idx' ('peca' ASC),
288     CONSTRAINT 'fk_Peca_has_OrcamentoServico_Peca1'
289     FOREIGN KEY ('peca')
290     REFERENCES 'opencarshop'.'.Peca' ('id')
291     ON DELETE NO ACTION
292     ON UPDATE NO ACTION,
293     CONSTRAINT 'fk_Peca_has_OrcamentoServico_OrcamentoServico1'
294     FOREIGN KEY ('orcamentoServico')
295     REFERENCES 'opencarshop'.'.OrcamentoServico' ('id')
296     ON DELETE NO ACTION
297     ON UPDATE NO ACTION)
298 ENGINE = InnoDB;
299
300
301 -----
302 -- Table 'opencarshop'.'.OrdemServico'
303 -----
304 DROP TABLE IF EXISTS 'opencarshop'.'.OrdemServico' ;
305
306 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.OrdemServico' (
307     'id' INT NOT NULL,
308     'data' VARCHAR(45) NULL,
309     'valorFinal' DECIMAL(10,2) NULL,
310     'desconto' DECIMAL(6,2) NULL,
311     'orcamento' INT NOT NULL,
312     'cliente' VARCHAR(15) NOT NULL,
313     'funcionario' VARCHAR(15) NOT NULL,
314     PRIMARY KEY ('id', 'orcamento', 'cliente', 'funcionario'),
315     INDEX 'fk_OrdemServico_OrcamentoServico1_idx' ('orcamento' ASC),

```

```

316 INDEX 'fk_OrdemServico_Cliente1_idx' ('cliente' ASC),
317 INDEX 'fk_OrdemServico_Funcionario1_idx' ('funcionario' ASC),
318 CONSTRAINT 'fk_OrdemServico_OrcamentoServico1'
319 FOREIGN KEY ('orcamento')
320 REFERENCES 'opencarshop'.'.OrcamentoServico' ('id')
321 ON DELETE NO ACTION
322 ON UPDATE NO ACTION,
323 CONSTRAINT 'fk_OrdemServico_Cliente1'
324 FOREIGN KEY ('cliente')
325 REFERENCES 'opencarshop'.'.Cliente' ('cpf')
326 ON DELETE NO ACTION
327 ON UPDATE NO ACTION,
328 CONSTRAINT 'fk_OrdemServico_Funcionario1'
329 FOREIGN KEY ('funcionario')
330 REFERENCES 'opencarshop'.'.Funcionario' ('cpf')
331 ON DELETE NO ACTION
332 ON UPDATE NO ACTION)
333 ENGINE = InnoDB;
334
335
336 -----
337 -- Table 'opencarshop'.'.Venda'
338 -----
339 DROP TABLE IF EXISTS 'opencarshop'.'.Venda' ;
340
341 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.Venda' (
342 'codigo' INT NOT NULL AUTO_INCREMENT,
343 'funcionario' VARCHAR(15) NOT NULL,
344 'cliente' VARCHAR(15) NOT NULL,
345 PRIMARY KEY ('codigo'),
346 INDEX 'fk_Funcionario_has_Cliente_Cliente1_idx' ('cliente' ASC),
347 INDEX 'fk_Funcionario_has_Cliente_Funcionario1_idx' ('funcionario' ASC),
348 CONSTRAINT 'fk_Funcionario_has_Cliente_Funcionario1'
349 FOREIGN KEY ('funcionario')
350 REFERENCES 'opencarshop'.'.Funcionario' ('cpf')
351 ON DELETE NO ACTION
352 ON UPDATE NO ACTION,
353 CONSTRAINT 'fk_Funcionario_has_Cliente_Cliente1'
354 FOREIGN KEY ('cliente')
355 REFERENCES 'opencarshop'.'.Cliente' ('cpf')
356 ON DELETE NO ACTION
357 ON UPDATE NO ACTION)
358 ENGINE = InnoDB;
359
360
361 -----
362 -- Table 'opencarshop'.'.ItemVeiculo'
363 -----
364 DROP TABLE IF EXISTS 'opencarshop'.'.ItemVeiculo' ;
365
366 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.ItemVeiculo' (
367 'veiculo' INT NOT NULL,
368 'valorFinal' DECIMAL(10,2) NULL,
369 'chasi' VARCHAR(45) NULL,
370 'desconto' DECIMAL(6,2) NULL,
371 'venda' INT NOT NULL,
372 PRIMARY KEY ('veiculo', 'venda'),
373 INDEX 'fk_Veiculo_has_Funcionario_Veiculo1_idx' ('veiculo' ASC),
374 INDEX 'fk_ItemVeiculo_Venda1_idx' ('venda' ASC),
375 CONSTRAINT 'fk_Veiculo_has_Funcionario_Veiculo1'
376 FOREIGN KEY ('veiculo')
377 REFERENCES 'opencarshop'.'.Veiculo' ('id')
378 ON DELETE NO ACTION
379 ON UPDATE NO ACTION,
380 CONSTRAINT 'fk_ItemVeiculo_Venda1'
381 FOREIGN KEY ('venda')
382 REFERENCES 'opencarshop'.'.Venda' ('codigo')
383 ON DELETE NO ACTION
384 ON UPDATE NO ACTION)
385 ENGINE = InnoDB;
386
387
388 -----
389 -- Table 'opencarshop'.'.ItemPeca'
390 -----
391 DROP TABLE IF EXISTS 'opencarshop'.'.ItemPeca' ;
392
393 CREATE TABLE IF NOT EXISTS 'opencarshop'.'.ItemPeca' (
394 'id' INT NOT NULL AUTO_INCREMENT,
395 'valorFinal' DECIMAL(10,2) NULL,
396 'desconto' DECIMAL(6,2) NULL,
397 'peca' INT NOT NULL,
398 'venda' INT NOT NULL,
399 PRIMARY KEY ('id', 'peca', 'venda'),
400 INDEX 'fk_ItemPeca_Peca1_idx' ('peca' ASC),
401 INDEX 'fk_ItemPeca_Venda1_idx' ('venda' ASC),
402 CONSTRAINT 'fk_ItemPeca_Peca1'
403 FOREIGN KEY ('peca')
404 REFERENCES 'opencarshop'.'.Peca' ('id')
405 ON DELETE NO ACTION
406 ON UPDATE NO ACTION,
407 CONSTRAINT 'fk_ItemPeca_Venda1'
408 FOREIGN KEY ('venda')
409 REFERENCES 'opencarshop'.'.Venda' ('codigo')

```

```
410         ON DELETE NO ACTION
411         ON UPDATE NO ACTION)
412 ENGINE = InnoDB;
413
414
415 SET SQL_MODE=@OLD_SQL_MODE;
416 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
417 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Código 49: OpenCarShop.sql

6 Diagramas

6.1 Diagrama de Casos de Uso

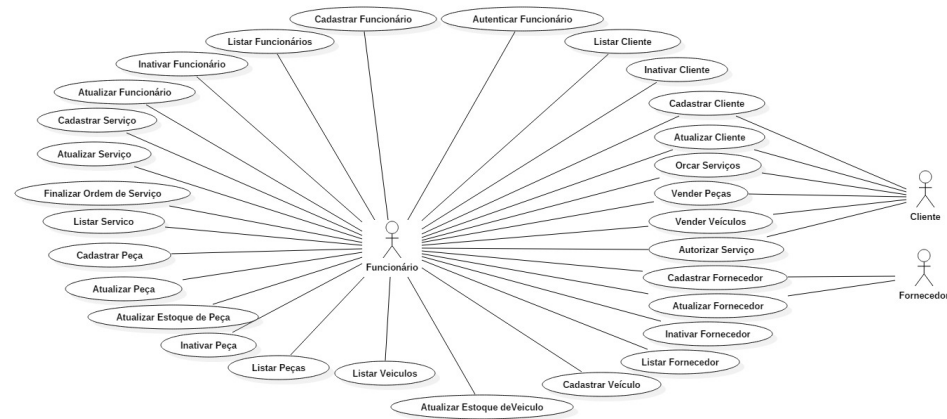


Figura 1: Diagrama de Casos de Uso

6.2 Diagrama de Classes - Analise

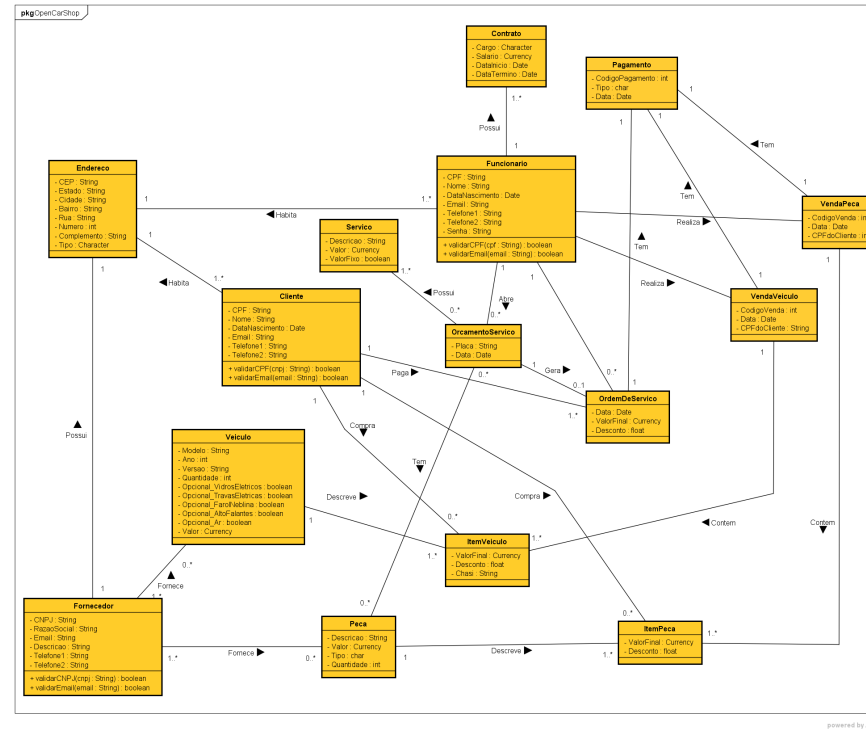


Figura 2: Diagrama de Classes - Analise

6.3 Diagrama de Classes - Projeto

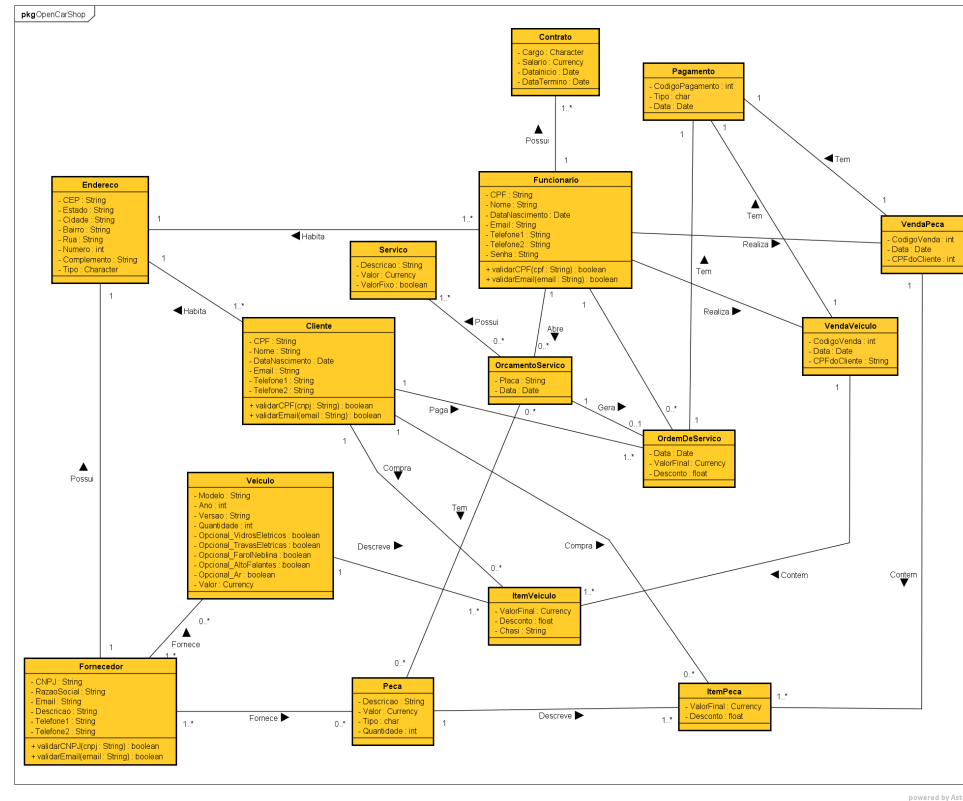


Figura 3: Diagrama de Classes - Projeto

6.4 Diagrama Entidade Relacionamento

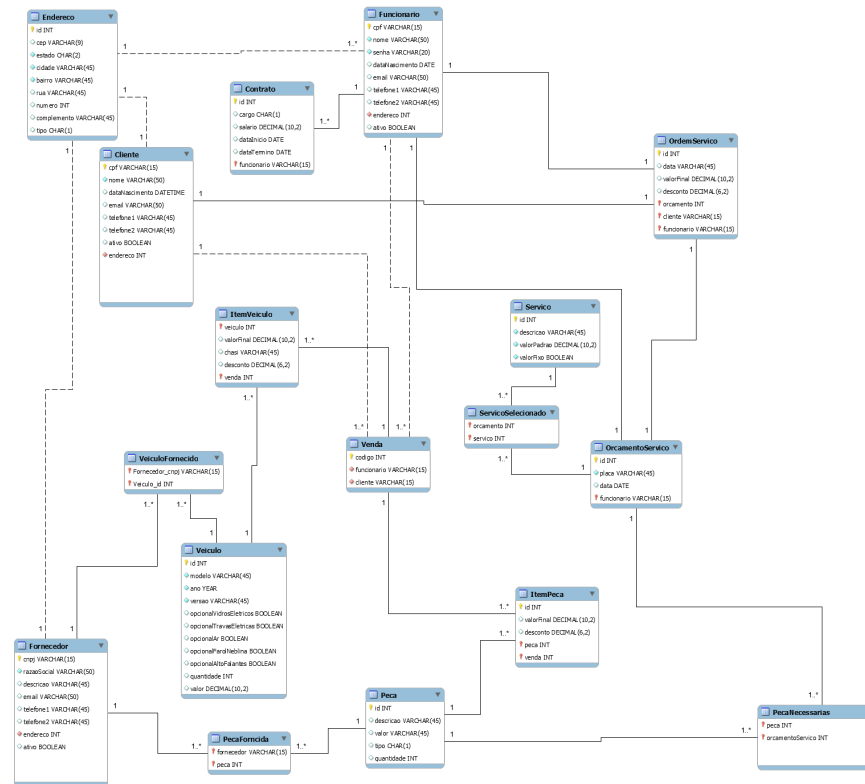


Figura 4: Diagrama de Entidade Relacionamento