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

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 veiculos, pecas, servicos, 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 autentiquem no sistema e manipularem 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.

3 SOMMERVILLE, I. Engenharia de Software. Pearson/Prentice Hall.

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

O sistema deve gerenciar, empregados, vendas, estoque e serviços de um concessionaria.

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 Prioridade

- 1: Prioridade alta.
- 2: Prioridade media.
- 3: Prioridade baixa.

1.7.2 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.: 3):

O sistema deve permitir ao funcionário inativar cadastro de clientes.

RF4 Atualizar Cliente (Pr.: 2):

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.: 2):

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

RF8 Inativar Funcionário (Pr.: 3):

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.: 2):

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.: 2):

O sistema deve permitir ao gerente atualizar fornecedores.

RF16 Inativar Fornecedor (Pr.: 3):

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.: 2):

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.: 3):

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.: 2):

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

RF34 Gerar Comprovante de Pagamento da venda de veículos (Pr.: 2):

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 gerar um comprovante de pagamento pela contratação de serviços.

RF36 Verificar disponibilidade (Pr.: 1):

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

RF37 Atualizar Estoque de peças (Pr.: 1):

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

RF38 Atualizar Estoque de veículos (Pr.: 1):

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

RF39 Relatório de Vendas (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.

1.7.3 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

Dividiu-se o sistema em 6 pacotes principais, com as funcionalidades do sistema distribuídas entre eles, cada dupla ficaria responsável em desenvolver um pacote, podendo ser ajudados por outros componente de acordo com a demanda.

2.1 Motivação

Motivação para o projeto:

- Praticar desenvolvimento em equipe.
- Aprender a documentar um software.
- Desenvolver seguindo a documentação.
- Vivenciar o processo de desenvolvimento de software.

3 Casos de Uso

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:

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 - Seleciona cliente	
9 - Seleciona opção Continuar.	
	10 - Atualiza Quantidade do Veículo no Estoque
	11 - Exibe documento de comprovação de venda.
	12 - Gera cobrança.
13 - Seleciona opção imprimir comprovante de venda.	
14 - Seleciona 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 Códigos

4.1 Pacote Cliente

4.1.1 Model

```java
package opencarshop.cliente.model;

import java.time.LocalDate;

public class Cliente {

    private String cpf;
    private String nome;
    private LocalDate dataNascimento;
    private String email;
    private String telefone1;
    private String telefone2;
    private Boolean ativo;

    public Cliente() {

    }

    public String getCpf() {
        return cpf;
    }

    public void setCpf(String cpf) {
        this.cpf = cpf;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public LocalDate getDataNascimento() {
        return dataNascimento;
    }

    public void setDataNascimento(LocalDate dataNascimento) {
        this.dataNascimento = dataNascimento;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getTelefone1() {
        return telefone1;
    }

    public void setTelefone1(String telefone1) {
        this.telefone1 = telefone1;
    }

    public String getTelefone2() {
        return telefone2;
    }

    public void setTelefone2(String telefone2) {
        this.telefone2 = telefone2;
    }

    public Boolean getAtivo() {
        return ativo;
    }

    public void setAtivo(Boolean ativo) {
        this.ativo = ativo;
    }

    @Override
    public String toString() {
        return this.nome;
    }

}
```

Código 1: Cliente.java

```java
package opencarshop.cliente.model;

import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import opencarshop.Endereco;
import opencarshop.cliente.model.Cliente;
import opencarshop.util.ConexaoMySQL;
import opencarshop.util.Utilidades;

public class ClienteDAO {

    private Connection conn;
    private final ConexaoMySQL c = new ConexaoMySQL();

    public boolean cadastraCliente(Cliente cli, Endereco end) {
        Connection conn = null;

        PreparedStatement stmtEnd = null;
        PreparedStatement stmtCli = null;

        String queryEnd = "INSERT INTO Endereco (cep, estado, cidade, bairro, rua, numero, complemento, tipo) VALUES (?,?,?,?,?,?,?,?)";
        String queryFun = "INSERT INTO Cliente (cpf, nome, dataNascimento, email, telefone1, telefone2, endereco, ativo) VALUES (?,?,?,?,?,?,(select LAST_INSERT_ID()),?)";

        try {
            conn = c.conectar();
            conn.setAutoCommit(false);

            stmtEnd = conn.prepareStatement(queryEnd);
            stmtCli = conn.prepareStatement(queryFun);

            stmtEnd.setString(1, end.getCEP());
            stmtEnd.setString(2, end.getEstado());
            stmtEnd.setString(3, end.getCidade());
            stmtEnd.setString(4, end.getBairro());
            stmtEnd.setString(5, end.getRua());
            stmtEnd.setInt(6, end.getNumero());
            stmtEnd.setString(7, end.getComplemento());
            stmtEnd.setString(8, Character.toString(end.getTipo()));

            stmtCli.setString(1, cli.getCpf());
            stmtCli.setString(2, cli.getNome());
            stmtCli.setDate(3, Date.valueOf(cli.getDataNascimento()));
            stmtCli.setString(4, cli.getEmail());
            stmtCli.setString(5, cli.getTelefone1());
            stmtCli.setString(6, cli.getTelefone2());
            stmtCli.setBoolean(7, true);

            stmtEnd.execute();
            stmtCli.execute();

            conn.commit();

            conn.close();
            return true;
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }



    public List<Cliente> getAllCliente() throws Exception
    {
        String query = "SELECT * FROM Cliente";
        List<Cliente> retorno = new ArrayList<>();
        Utilidades u = new Utilidades();
        conn = c.conectar();
        try {
            PreparedStatement stmt = conn.prepareStatement(query);
            ResultSet resultado = stmt.executeQuery();
            while (resultado.next()) {
                System.out.println(resultado.getString("nome"));
                Cliente cliente = new Cliente();
                cliente.setCpf(resultado.getString("cpf"));
                cliente.setNome(resultado.getString("nome"));
                cliente.setDataNascimento(u.toLocalDate(resultado.getDate("dataNascimento")));
                cliente.setEmail(resultado.getString("email"));
                cliente.setTelefone1(resultado.getString("telefone1"));
                cliente.setTelefone2(resultado.getString("telefone2"));
                cliente.setAtivo(resultado.getBoolean("ativo"));

                retorno.add(cliente);
            }
```

```
92          } catch (SQLException ex) {
93              Logger.getLogger(ClienteDAO.class.getName()).log(Level.SEVERE, null, ex);
94          }
95          conn.close();
96          return retorno;
97      }
98
99      public Cliente buscar(Cliente cliente) {
100         String sql = "SELECT␣*␣FROM␣Cliente␣WHERE␣cpf=?";
101         Cliente retorno = new Cliente();
102         Utilidades u = new Utilidades();
103         try {
104             PreparedStatement stmt = conn.prepareStatement(sql);
105             stmt.setString(1, cliente.getCpf());
106             ResultSet resultado = stmt.executeQuery();
107             if (resultado.next()) {
108                 cliente.setCpf(resultado.getString("cpf"));
109                 cliente.setNome(resultado.getString("nome"));
110                 cliente.setDataNascimento(u.toLocalDate(resultado.getDate("dataNascimento")));
111                 cliente.setEmail(resultado.getString("email"));
112                 cliente.setTelefone1(resultado.getString("telefone1"));
113                 cliente.setTelefone2(resultado.getString("telefone2"));
114                 cliente.setAtivo(resultado.getBoolean("ativo"));
115                 retorno = cliente;
116             }
117         } catch (SQLException ex) {
118             Logger.getLogger(ClienteDAO.class.getName()).log(Level.SEVERE, null, ex);
119         }
120         return retorno;
121     }
122
123     public Boolean alteraCliente(Cliente cli) throws SQLException {
124         String query = "UPDATE␣Cliente␣SET␣nome=?,␣dataNascimento=?,␣email=?,␣telefone1=?,␣telefone2=?,␣ativo=?␣WHERE␣
      cpf=?";
125
126         try {
127             conn = c.conectar();
128             PreparedStatement stmt = conn.prepareStatement(query);
129
130             stmt.setString(1, cli.getNome());
131             stmt.setDate(2, Date.valueOf(cli.getDataNascimento()));
132             stmt.setString(3, cli.getEmail());
133             stmt.setString(4, cli.getTelefone1());
134             stmt.setString(5, cli.getTelefone2());
135             stmt.setBoolean(6, cli.getAtivo());
136             stmt.setString(7, cli.getCpf());
137             stmt.execute();
138             conn.close();
139             return true;
140         } catch (Exception ex) {
141
142             Logger.getLogger(ClienteDAO.class.getName()).log(Level.SEVERE, null, ex);
143             return false;
144         }
145     }
146
147     public Connection getConnection() {
148         return conn;
149     }
150
151     public void setConnection(Connection connection) {
152         this.conn = connection;
153     }
154 }
```

Código 2: ClienteDAO.java

4.1.2 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"
       fx:controller="opencarshop.cliente.controller.ClienteController">
14     <children>
15         <TabPane prefHeight="494.0" prefWidth="1024.0" tabClosingPolicy="UNAVAILABLE">
16             <tabs>
17                 <Tab text="Identificacao">
18                     <content>
```

```
                        <AnchorPane minHeight="0.0" minWidth="0.0" prefWidth="1024.0">
                            <children>
                                <TextField fx:id="tf_cpfCadastro" layoutX="194.0" layoutY="71.0" prefHeight="25.0" prefWidth="170.0" promptText="CPF" />
                                <TextField fx:id="tf_nomeCadastro" layoutX="14.0" layoutY="25.0" prefHeight="25.0" prefWidth="350.0" promptText="Nome completo" />
                                <DatePicker fx:id="dp_dataNascimentoCadastro" layoutX="14.0" layoutY="71.0" prefHeight="25.0" prefWidth="170.0" promptText="Data Nascimento" />
                                <Label layoutX="14.0" layoutY="6.0" text="Nome:" />
                                <Label layoutX="14.0" layoutY="50.0" text="Data de Nascimento:" />
                                <Label layoutX="194.0" layoutY="50.0" text="CPF:" />
                            </children>
                        </AnchorPane>
                    </content>
                </Tab>
                <Tab text="Contato">
                    <content>
                        <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="258.0" prefWidth="1024.0">
                            <children>
                                <TextField fx:id="tf_emailCadastro" layoutX="14.0" layoutY="26.0" prefHeight="25.0" prefWidth="350.0" promptText="Email" />
                                <TextField fx:id="tf_telefone1Cadastro" layoutX="14.0" layoutY="70.0" prefWidth="170.0" promptText="Telefone 1" />
                                <TextField fx:id="tf_telefone2Cadastro" layoutX="194.0" layoutY="70.0" prefWidth="170.0" promptText="Telefone 2" />
                                <TextField fx:id="tf_ruaCadastro" layoutX="14.0" layoutY="160.0" prefWidth="350.0" promptText="Rua" />
                                <TextField fx:id="tf_cidadeCadastro" layoutX="14.0" layoutY="249.0" prefWidth="350.0" promptText="Cidade" />
                                <TextField fx:id="tf_estadoCadastro" layoutX="14.0" layoutY="293.0" prefWidth="170.0" promptText="UF" />
                                <TextField fx:id="tf_bairroCadastro" layoutX="194.0" layoutY="293.0" prefWidth="170.0" promptText="Bairro" />
                                <TextField fx:id="tf_cepCadastro" layoutX="14.0" layoutY="336.0" prefWidth="170.0" promptText="CEP" />
                                <TextField fx:id="tf_numeroCadastro" layoutX="194.0" layoutY="336.0" prefWidth="170.0" promptText="Número" />
                                <TextField fx:id="tf_complementoCadastro" layoutX="14.0" layoutY="204.0" prefWidth="350.0" promptText="Complemento" />
                                <Button id="btn_cadastrar" layoutX="151.0" layoutY="414.0" mnemonicParsing="false" onAction="#cadastrar" text="Cadastrar" />
                                <ComboBox fx:id="cb_tipoCadastro" layoutX="14.0" layoutY="115.0" prefWidth="350.0" promptText="Tipo de Endereço">
                                    <items>
                                        <FXCollections fx:factory="observableArrayList">
                                            <String fx:value="Residencial" />
                                            <String fx:value="Comercial" />
                                        </FXCollections>
                                    </items>
                                </ComboBox>
                                <Label layoutX="14.0" layoutY="6.0" text="Email:" />
                                <Label layoutX="16.0" layoutY="51.0" text="Telefone 1:" />
                                <Label layoutX="194.0" layoutY="51.0" text="Telefone 2" />
                                <Label layoutX="16.0" layoutY="95.0" text="Endereço:" />
                                <Label layoutX="16.0" layoutY="140.0" text="Rua:" />
                                <Label layoutX="16.0" layoutY="185.0" text="Complemento:" />
                                <Label layoutX="16.0" layoutY="233.0" text="Cidade:" />
                                <Label layoutX="16.0" layoutY="274.0" text="UF - Estado:" />
                                <Label layoutX="194.0" layoutY="274.0" text="Bairro:" />
                                <Label layoutX="16.0" layoutY="318.0" text="CEP:" />
                                <Label layoutX="197.0" layoutY="318.0" text="Número:" />
                                <Label fx:id="resultadoCadastro" alignment="CENTER" layoutX="80.0" layoutY="384.0" prefHeight="17.0" prefWidth="218.0" />
                            </children>
                        </AnchorPane>
                    </content>
                </Tab>
            </tabs>
        </TabPane>
    </children>
</AnchorPane>
```

Código 3: Cadastrar.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<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.cliente.controller.ClienteController">
    <children>
        <TableView fx:id="tbl_cliente" prefHeight="650.0" prefWidth="650.0">
            <columns>
                <TableColumn fx:id="col_cpf" prefWidth="100.0" text="CPF" />
                <TableColumn fx:id="col_nome" prefWidth="155.0" text="Nome" />
                <TableColumn id="col_telefone" fx:id="col_telefone1" minWidth="0.0" prefWidth="103.0" text="Telefone 1" />
                <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="151.0" prefWidth="150.0" promptText="
        Data␣de␣nascimento" />
25          <Label layoutX="844.0" layoutY="78.0" text="Nome:" />
26          <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␣Alterações" />
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

4.1.3 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_cargoCadastro.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_cpfCadastro.getText());
93          cli.setNome(tf_nomeCadastro.getText());
94          cli.setDataNascimento(dp_dataNascimentoCadastro.getValue());
95          cli.setEmail(tf_emailCadastro.getText());
96          cli.setTelefone1(tf_telefone1Cadastro.getText());
97          cli.setTelefone2(tf_telefone2Cadastro.getText());
98          cli.setAtivo(true);
99
100         // OBJETO ENDERECO
101         end.setCEP(tf_cepCadastro.getText());
102         end.setEstado(tf_estadoCadastro.getText());
103         end.setCidade(tf_cidadeCadastro.getText());
104         end.setBairro(tf_bairroCadastro.getText());
105         end.setRua(tf_ruaCadastro.getText());
106         end.setNumero(Integer.parseInt(tf_numeroCadastro.getText()));
107         end.setComplemento(tf_complementoCadastro.getText());
108         end.setTipo(cb_tipoCadastro.getValue().charAt(0));
109
110         if (c.cadastraCliente(cli, end)) {
111             resultadoCadastro.setText("Cadastrado␣com␣sucesso!!");
112         } else {
113             resultadoCadastro.setText("Erro␣ao␣cadastrar!!␣Tente␣novamente.");
114         }
115     }
116
117     @FXML
118     private void alterarCadastro(ActionEvent event) throws Exception {
119         Cliente cli = new Cliente();
120         cli.setCpf(tf_cpfCadastro.getText());
121         cli.setNome(tf_nomeCadastro.getText());
122         cli.setDataNascimento(dp_dataNascimentoCadastro.getValue());
123         cli.setEmail(tf_emailCadastro.getText());
124         cli.setTelefone1(tf_telefone1Cadastro.getText());
125         cli.setTelefone2(tf_telefone2Cadastro.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␣a␣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> observableListFuncionatio;
146
147         observableListFuncionatio = FXCollections.observableArrayList(listaCliente);
148         tbl_cliente.setItems(observableListFuncionatio);
149     }
150
151     public void selecionarItemTablelaCliente(Cliente cliente) {
152         if (cliente.getCpf() != null) {
153             tf_cpfCadastro.setText(cliente.getCpf());
154             tf_nomeCadastro.setText(cliente.getNome());
```

```
155             tf_emailCadastro.setText(cliente.getEmail());
156             tf_telefone1Cadastro.setText(cliente.getTelefone1());
157             tf_telefone2Cadastro.setText(cliente.getTelefone2());
158             dp_dataNascimentoCadastro.setValue(cliente.getDataNascimento());
159             cb_ativo.setSelected(cliente.getAtivo());
160 
161         }
162     }
163 
164     @Override
165     public void initialize(URL url, ResourceBundle rb) {
166         try {
167             carregaTabelaCliente();
168             tbl_cliente.getSelectionModel().selectedItemProperty().addListener(
169                     (observable, oldValue, newValue) -> selecionarItemTablelaCliente(newValue));
170         } catch (Exception ex) {
171             //Logger.getLogger(ClienteController.class.getName()).log(Level.SEVERE, null, ex);
172         }
173     }
174 
175 }
```

Código 5: ClienteController.java

4.2 Pacote Fornecedor

4.2.1 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     private Boolean ativo;
12 
13     public Boolean getAtivo() {
14         return ativo;
15     }
16 
17     public void setAtivo(Boolean ativo) {
18         this.ativo = ativo;
19     }
20 
21     public String getCnpj() {
22         return cnpj;
23     }
24 
25     public void setCnpj(String cnpj) {
26         this.cnpj = cnpj;
27     }
28 
29     public String getRazaoSocial() {
30         return razaoSocial;
31     }
32 
33     public void setRazaoSocial(String razaoSocial) {
34         this.razaoSocial = razaoSocial;
35     }
36 
37     public String getEmail() {
38         return email;
39     }
40 
41     public void setEmail(String email) {
42         this.email = email;
43     }
44 
45     public String getTelefone1() {
46         return telefone1;
47     }
48 
49     public void setTelefone1(String telefone1) {
50         this.telefone1 = telefone1;
51     }
52 
53     public String getTelefone2() {
54         return telefone2;
55     }
56 
57     public void setTelefone2(String telefone2) {
58         this.telefone2 = telefone2;
```

```
59      }
60
61      public String getDescricao() {
62          return descricao;
63      }
64
65      public void setDescricao(String descricao) {
66          this.descricao = descricao;
67      }
68  }
```

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.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.Endereco;
17  import opencarshop.util.ConexaoMySQL;
18  import opencarshop.util.Utilidades;
19
20  /**
21   *
22   * @author JomarR
23   */
24  public class FornecedorDAO {
25
26  public Fornecedor getFornecedor(String cnpj)
27      {
28          ConexaoMySQL c = new ConexaoMySQL();
29          Connection conn = null;
30          PreparedStatement stmt = null;
31          Utilidades u = new Utilidades();
32
33          String query = "SELECT * FROM Fornecedor WHERE cnpj=?";
34
35          Fornecedor fornecedor = new Fornecedor();
36          try
37          {
38              conn = c.conectar();
39              stmt = conn.prepareStatement(query);
40              stmt.setString(1,cnpj);
41
42              ResultSet resultado = stmt.executeQuery();
43
44              if(resultado.next())
45              {
46                  fornecedor.setCnpj(resultado.getString("cnpj"));
47                  fornecedor.setDescricao(resultado.getString("razaoSocial"));
48                  fornecedor.setEmail(resultado.getString("email"));
49                  fornecedor.setTelefone1(resultado.getString("telefone1"));
50                  fornecedor.setTelefone2(resultado.getString("telefone2"));
51                   fornecedor.setDescricao(resultado.getString("descricao"));
52                   fornecedor.setAtivo(resultado.getBoolean("ativo"));
53              }
54
55              conn.close();
56          }
57          catch(Exception e)
58          {
59              e.printStackTrace();
60          }
61          return fornecedor;
62      }
63
64      public boolean cadastraFornecedor(Fornecedor fornecedor, Endereco end)
65      {
66          ConexaoMySQL c = new ConexaoMySQL();
67          Connection conn = null;
68
69          PreparedStatement stmtEnd = null;
70
71          PreparedStatement stmtFornecedor = null;
72
73          String queryEnd = "INSERT INTO Endereco (cep, estado, cidade, bairro, rua, numero, complemento, tipo) VALUES
        (?,?,?,?,?,?,?,?)";
74          String queryFornecedor = "INSERT INTO Fornecedor (cnpj, razaoSocial, email, telefone1, telefone2, descricao,
        endereco, ativo) VALUES (?,?,?,?,?,?,(select LAST_INSERT_ID()),?)";
75
76
77          try
```

```
78          {
79              conn = c.conectar();
80              conn.setAutoCommit(false);
81
82              stmtEnd = conn.prepareStatement(queryEnd);
83              stmtFornecedor = conn.prepareStatement(queryFornecedor);
84
85
86
87              stmtEnd.setString(1, end.getCEP());
88              stmtEnd.setString(2, end.getEstado());
89              stmtEnd.setString(3, end.getCidade());
90              stmtEnd.setString(4, end.getBairro());
91              stmtEnd.setString(5, end.getRua());
92              stmtEnd.setInt(6, end.getNumero());
93              stmtEnd.setString(7, end.getComplemento());
94              stmtEnd.setString(8, Character.toString(end.getTipo()));
95
96              stmtFornecedor.setString(1, fornecedor.getCnpj());
97              stmtFornecedor.setString(2, fornecedor.getRazaoSocial());
98              stmtFornecedor.setString(3, fornecedor.getEmail());
99              stmtFornecedor.setString(4, fornecedor.getTelefone1());
100             stmtFornecedor.setString(5, fornecedor.getTelefone2());
101             stmtFornecedor.setString(6, fornecedor.getDescricao());
102             stmtFornecedor.setBoolean(7, true);
103
104
105
106
107             stmtEnd.execute();
108             stmtFornecedor.execute();
109
110
111             conn.commit();
112
113             conn.close();
114             return true;
115         }
116         catch(Exception e)
117         {
118             e.printStackTrace();
119             return false;
120         }
121     }
122
123     public List<Fornecedor> getAllFornecedor() throws Exception
124     {
125         String query = "SELECT␣*␣FROM␣Fornecedor";
126         List<Fornecedor> retorno = new ArrayList<>();
127         Utilidades u = new Utilidades();
128         ConexaoMySQL c = new ConexaoMySQL();
129         Connection conn = null;
130         conn = c.conectar();
131         try {
132             PreparedStatement stmt = conn.prepareStatement(query);
133             ResultSet resultado = stmt.executeQuery();
134             while (resultado.next()) {
135                 Fornecedor fornecedor = new Fornecedor();
136                 fornecedor.setCnpj(resultado.getString("cnpj"));
137                 fornecedor.setRazaoSocial(resultado.getString("razaoSocial"));
138                 fornecedor.setEmail(resultado.getString("email"));
139                 fornecedor.setTelefone1(resultado.getString("telefone1"));
140                 fornecedor.setTelefone2(resultado.getString("telefone2"));
141                 fornecedor.setDescricao(resultado.getString("descricao"));
142                 fornecedor.setAtivo(resultado.getBoolean("ativo"));
143                 retorno.add(fornecedor);
144             }
145         } catch (Exception e) {
146             e.printStackTrace();
147         }
148         conn.close();
149         return retorno;
150     }
151
152     public Boolean alteraFornecedor(Fornecedor fornecedor) throws SQLException
153     {
154         String query = "UPDATE␣Fornecedor␣SET␣cnpj=?,␣razaoSocial=?,␣email=?,␣telefone1=?,␣telefone2=?,␣descricao␣=?,␣
     ativo=?␣WHERE␣cnpj=?";
155
156         ConexaoMySQL c = new ConexaoMySQL();
157         Connection conn = null;
158         try {
159             conn = c.conectar();
160             PreparedStatement stmt = conn.prepareStatement(query);
161
162
163             stmt.setString(1, fornecedor.getCnpj());
164             stmt.setString(2, fornecedor.getRazaoSocial());
165             stmt.setString(3, fornecedor.getEmail());
166             stmt.setString(4, fornecedor.getTelefone1());
167             stmt.setString(5, fornecedor.getTelefone2());
168             stmt.setString(6, fornecedor.getDescricao());
169             stmt.setBoolean(7, fornecedor.getAtivo());
170             stmt.setString(8, fornecedor.getCnpj());
```

```
            stmt.execute();
            conn.close();
            return true;
        } catch (Exception ex) {

            Logger.getLogger(FornecedorDAO.class.getName()).log(Level.SEVERE, null, ex);
            return false;
        }
    }
}
```

Código 7: FornecedorDAO.java

4.2.2 View

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.chart.*?>
<?import javafx.scene.text.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<?import javafx.collections.*?>

<AnchorPane id="AnchorPane" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
     fx:controller="opencarshop.fornecedor.controller.FornecedorController">
   <children>
      <TabPane prefHeight="328.0" prefWidth="1024.0" tabClosingPolicy="UNAVAILABLE">
        <tabs>
          <Tab text="Identificação">
            <content>
              <AnchorPane minHeight="0.0" minWidth="0.0" prefWidth="1024.0">
                     <children>
                        <TextField fx:id="tf_cnpjCadastro" layoutX="14.0" layoutY="87.0" prefHeight="25.0" prefWidth="
     170.0" promptText="CNPJ" />
                        <TextField fx:id="tf_descricaoCadastro" layoutX="15.0" layoutY="142.0" prefHeight="73.0"
     prefWidth="170.0" promptText="Descrição" />
                        <TextField fx:id="tf_razaoCadastro" layoutX="14.0" layoutY="31.0" prefHeight="25.0" prefWidth="
     350.0" promptText="Razão Social" />
                        <Label layoutX="14.0" layoutY="6.0" text="Razão Social" />
                        <Label layoutX="15.0" layoutY="70.0" text="CNPJ" />
                        <Label layoutX="15.0" layoutY="122.0" text="Descricão" />
                     </children>
                  </AnchorPane>
            </content>
          </Tab>
          <Tab text="Contato">
            <content>
              <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="316.0" prefWidth="1024.0">
                     <children>
                        <TextField fx:id="tf_emailCadastro" layoutX="14.0" layoutY="14.0" prefHeight="25.0" prefWidth="
     350.0" promptText="Email" />
                        <TextField fx:id="tf_telefone1Cadastro" layoutX="14.0" layoutY="50.0" prefWidth="170.0"
     promptText="Telefone 1" />
                        <TextField fx:id="tf_telefone2Cadastro" layoutX="194.0" layoutY="50.0" prefWidth="170.0"
     promptText="Telefone 2" />
                        <TextField fx:id="tf_ruaCadastro" layoutX="14.0" layoutY="116.0" prefWidth="350.0" promptText="
     Rua" />
                        <TextField fx:id="tf_cidadeCadastro" layoutX="14.0" layoutY="187.0" prefWidth="350.0"
     promptText="Cidade" />
                        <TextField fx:id="tf_estadoCadastro" layoutX="14.0" layoutY="222.0" prefWidth="170.0"
     promptText="UF" />
                        <TextField fx:id="tf_bairroCadastro" layoutX="194.0" layoutY="222.0" prefWidth="170.0"
     promptText="Bairro" />
                        <TextField fx:id="tf_cepCadastro" layoutX="14.0" layoutY="256.0" prefWidth="170.0" promptText="
     CEP" />
                        <TextField fx:id="tf_numeroCadastro" layoutX="194.0" layoutY="256.0" prefWidth="170.0"
     promptText="Número" />
                        <TextField fx:id="tf_complementoCadastro" layoutX="14.0" layoutY="151.0" prefWidth="350.0"
     promptText="Complemento" />
                        <ComboBox fx:id="cb_tipoCadastro" layoutX="14.0" layoutY="83.0" prefWidth="350.0" promptText="
     Tipo de Endereço">
                            <items>
                            <FXCollections fx:factory="observableArrayList">
                                <String fx:value="Residencial" />
                                <String fx:value="Comercial" />
                            </FXCollections>
                          </items>
                          </ComboBox>
                     </children>
                  </AnchorPane>
            </content>
          </Tab>
        </tabs>
      </TabPane>
      <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 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.fornecedor.controller.FornecedorController">
10    <children>
11       <TableView fx:id="tbl_fornecedor" prefHeight="650.0" prefWidth="650.0">
12         <columns>
13          <TableColumn fx:id="col_cnpj" prefWidth="75.0" text="CNPJ" />
14           <TableColumn fx:id="col_razaoSocial" prefWidth="75.0" text="Razao␣Social" />
15             <TableColumn fx:id="col_email" prefWidth="75.0" text="Email" />
16             <TableColumn fx:id="col_telefone1" prefWidth="75.0" text="Telefone" />
17             <TableColumn fx:id="col_telefone2" prefWidth="75.0" text="Telefone␣2" />
18             <TableColumn fx:id="col_descricao" prefWidth="75.0" text="Descrição" />
19         </columns>
20       </TableView>
21       <TextField fx:id="tf_razaoCadastro" layoutX="846.0" layoutY="95.0" prefWidth="150.0" promptText="Razão␣Social" />
22       <TextField fx:id="tf_emailCadastro" layoutX="846.0" layoutY="151.0" promptText="Email" />
23       <TextField fx:id="tf_telefone1Cadastro" layoutX="675.0" layoutY="208.0" promptText="Telefone␣1" />
24       <TextField fx:id="tf_telefone2Cadastro" layoutX="846.0" layoutY="208.0" promptText="Telefone␣2" />
25
26       <Label layoutX="844.0" layoutY="78.0" text="Razão␣Social:" />
27       <Label layoutX="675.0" layoutY="134.0" text="Descrição:" />
28       <TextField fx:id="tf_descricaoCadastro" layoutX="675.0" layoutY="151.0" promptText="Descrição" />
29       <Label layoutX="846.0" layoutY="134.0" text="Email:" />
30       <Label layoutX="678.0" layoutY="191.0" text="Telefone␣1:" />
31       <Label layoutX="847.0" layoutY="191.0" text="Telefone␣2:" />
32
33       <CheckBox fx:id="cb_ativo" layoutX="903.0" layoutY="266.0" mnemonicParsing="false" text="Ativo" />
34       <Button fx:id="btn_alterarCadastro" layoutX="782.0" layoutY="359.0" mnemonicParsing="false" onAction="#
       alterarCadastro" text="Salvar␣Alterações" />
35       <TextField fx:id="tf_cnpjCadastro" editable="false" layoutX="675.0" layoutY="95.0" promptText="CNPJ"/>
36       <Label layoutX="675.0" layoutY="78.0" text="CNPJ:" />
37       <Label fx:id="confirmaAlteracao" alignment="CENTER" contentDisplay="CENTER" layoutX="736.0" layoutY="325.0"
          prefHeight="17.0" prefWidth="199.0" textAlignment="CENTER" />
38    </children>
39 </AnchorPane>
```

Código 9: Buscar.fxml

4.2.3 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.CheckBox;
13 import javafx.scene.control.ComboBox;
14 import javafx.scene.control.Hyperlink;
15 import javafx.scene.control.Label;
16 import javafx.scene.control.TableColumn;
17 import javafx.scene.control.TableView;
18 import javafx.scene.control.TextField;
19 import javafx.scene.control.cell.PropertyValueFactory;
20 import opencarshop.Endereco;
21 import opencarshop.fornecedor.model.Fornecedor;
22 import opencarshop.fornecedor.model.FornecedorDAO;
23 import opencarshop.util.Utilidades;
24
25 public class FornecedorController implements Initializable {
26
27     /**
28      * Initializes the controller class.
29      */
30     // TELA DE CADASTRO
31     @FXML
32     private TextField tf_cnpjCadastro;
```

```
33      @FXML
34      private TextField tf_razaoCadastro;
35      @FXML
36      private TextField tf_descricaoCadastro;
37
38      @FXML
39      private ComboBox<String> cb_tipoCadastro;
40      @FXML
41      private TextField tf_emailCadastro;
42      @FXML
43      private TextField tf_telefone1Cadastro;
44      @FXML
45      private TextField tf_telefone2Cadastro;
46      @FXML
47      private TextField tf_ruaCadastro;
48      @FXML
49      private TextField tf_cidadeCadastro;
50      @FXML
51      private TextField tf_estadoCadastro;
52      @FXML
53      private TextField tf_bairroCadastro;
54      @FXML
55      private TextField tf_cepCadastro;
56      @FXML
57      private TextField tf_numeroCadastro;
58      @FXML
59      private TextField tf_complementoCadastro;
60      @FXML
61      private CheckBox cb_ativo;
62      @FXML
63      private Label confirmaAlteracao;
64      @FXML
65      private Label resultadoCadastro;
66
67      // TABELA Fornecedor
68      @FXML
69      private TableColumn<Fornecedor, String> col_cnpj;
70      @FXML
71      private TableColumn<Fornecedor, String> col_razaoSocial;
72      @FXML
73      private TableColumn<Fornecedor, String> col_email;
74      @FXML
75      private TableColumn<Fornecedor, String> col_telefone1;
76      @FXML
77      private TableColumn<Fornecedor, String> col_telefone2;
78      @FXML
79      private TableColumn<Fornecedor, String> col_descricao;
80
81      @FXML
82      private TableView<Fornecedor> tbl_fornecedor;
83
84      @FXML
85      private void cadastrar(ActionEvent event) throws ParseException {
86
87          Fornecedor fornecedor = new Fornecedor();
88          Endereco end = new Endereco();
89          FornecedorDAO f = new FornecedorDAO();
90          Utilidades u = new Utilidades();
91
92          // OBJETO FORNECEDOR
93          fornecedor.setCnpj(tf_cnpjCadastro.getText());
94          fornecedor.setRazaoSocial(tf_razaoCadastro.getText());
95          fornecedor.setEmail(tf_emailCadastro.getText());
96          fornecedor.setTelefone1(tf_telefone1Cadastro.getText());
97          fornecedor.setTelefone2(tf_telefone2Cadastro.getText());
98          fornecedor.setDescricao(tf_descricaoCadastro.getText());
99          fornecedor.setAtivo(true);
100         // OBJETO ENDERECO
101         end.setCEP(tf_cepCadastro.getText());
102         end.setEstado(tf_estadoCadastro.getText());
103         end.setCidade(tf_cidadeCadastro.getText());
104         end.setBairro(tf_bairroCadastro.getText());
105         end.setRua(tf_ruaCadastro.getText());
106         end.setNumero(Integer.parseInt(tf_numeroCadastro.getText()));
107         end.setComplemento(tf_complementoCadastro.getText());
108         end.setTipo(cb_tipoCadastro.getValue().charAt(0));
109
110         if (f.cadastraFornecedor(fornecedor, end)) {
111             resultadoCadastro.setText("Cadastrado com sucesso!!");
112         } else {
113             resultadoCadastro.setText("Erro ao cadastrar!! Tente novamente.");
114         }
115     }
116
117     private void carregaTabelaFornecedor() throws Exception {
118         col_cnpj.setCellValueFactory(new PropertyValueFactory<>("cnpj"));
119         col_razaoSocial.setCellValueFactory(new PropertyValueFactory<>("razaoSocial"));
120         col_email.setCellValueFactory(new PropertyValueFactory<>("email"));
121         col_telefone1.setCellValueFactory(new PropertyValueFactory<>("telefone1"));
122         col_telefone2.setCellValueFactory(new PropertyValueFactory<>("telefone2"));
123         col_descricao.setCellValueFactory(new PropertyValueFactory<>("descricao"));
124         FornecedorDAO f = new FornecedorDAO();
125         List<Fornecedor> listaFornecedor = f.getAllFornecedor();
126         ObservableList<Fornecedor> observableListFornecedor;
```

```
127
128         observableListFornecedor = FXCollections.observableArrayList(listaFornecedor);
129         tbl_fornecedor.setItems(observableListFornecedor);
130     }
131
132     @FXML
133     private void alterarCadastro(ActionEvent event) throws Exception {
134         Fornecedor fornecedor = new Fornecedor();
135         fornecedor.setCnpj(tf_cnpjCadastro.getText());
136         fornecedor.setRazaoSocial(tf_razaoCadastro.getText());
137         fornecedor.setEmail(tf_emailCadastro.getText());
138         fornecedor.setTelefone1(tf_telefone1Cadastro.getText());
139         fornecedor.setTelefone2(tf_telefone2Cadastro.getText());
140         fornecedor.setDescricao(tf_descricaoCadastro.getText());
141         fornecedor.setAtivo(cb_ativo.isSelected());
142
143         FornecedorDAO f = new FornecedorDAO();
144         if (f.alteraFornecedor(fornecedor)) {
145             confirmaAlteracao.setText("Alteração␣realizada␣com␣sucesso!!");
146         } else {
147             confirmaAlteracao.setText("Erro␣ao␣realizar␣a␣alteração!!");
148         }
149     }
150
151     public void selecionarItemTablelaFornecedor(Fornecedor fornecedor) {
152         if (fornecedor.getCnpj() != null) {
153             tf_cnpjCadastro.setText(fornecedor.getCnpj());
154             tf_razaoCadastro.setText(fornecedor.getRazaoSocial());
155             tf_emailCadastro.setText(fornecedor.getEmail());
156             tf_telefone1Cadastro.setText(fornecedor.getTelefone1());
157             tf_telefone2Cadastro.setText(fornecedor.getTelefone2());
158             tf_descricaoCadastro.setText(fornecedor.getDescricao());
159             cb_ativo.setSelected(fornecedor.getAtivo());
160
161         }
162     }
163
164     @Override
165     public void initialize(URL url, ResourceBundle rb) {
166         try {
167             carregaTabelaFornecedor();
168             tbl_fornecedor.getSelectionModel().selectedItemProperty().addListener(
169                     (observable, oldValue, newValue) -> selecionarItemTablelaFornecedor(newValue));
170         } catch (Exception ex) {
171
172         }
173
174     }
175
176 }
```

Código 10: FornecedorController.java

4.3 Pacote Funcionário

4.3.1 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;
```

```
    }

    public void setTelefone2(String telefone2) {
        this.telefone2 = telefone2;
    }
}
```

Código 12: Funcionario.java

```
package opencarshop.funcionario.model;

import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import opencarshop.util.ConexaoMySQL;
import opencarshop.Endereco;
import opencarshop.util.Utilidades;

public class FuncionarioDAO {

    public Funcionario getFuncionario(String cpf) {
        ConexaoMySQL c = new ConexaoMySQL();
        Connection conn = null;
        PreparedStatement stmt = null;
        Utilidades u = new Utilidades();

        String query = "SELECT * FROM Funcionario WHERE cpf=?";

        Funcionario funcionario = new Funcionario();
        try {
            conn = c.conectar();
            stmt = conn.prepareStatement(query);
            stmt.setString(1, cpf);

            ResultSet resultado = stmt.executeQuery();

            if (resultado.next()) {
                funcionario.setCpf(resultado.getString("cpf"));
                funcionario.setNome(resultado.getString("nome"));
                funcionario.setSenha(resultado.getString("senha"));
                funcionario.setDataNascimento(u.toLocalDate(resultado.getDate("dataNascimento")));
                funcionario.setEmail(resultado.getString("email"));
                funcionario.setTelefone1(resultado.getString("telefone1"));
                funcionario.setTelefone2(resultado.getString("telefone2"));
                funcionario.setAtivo(resultado.getBoolean("ativo"));
            }

            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return funcionario;
    }

    public boolean cadastraFuncionario(Funcionario func, Endereco end, Contrato cont) {
        ConexaoMySQL c = new ConexaoMySQL();
        Connection conn = null;

        PreparedStatement stmtEnd = null;
        PreparedStatement stmtCon = null;
        PreparedStatement stmtFun = null;

        String queryEnd = "INSERT INTO Endereco (cep, estado, cidade, bairro, rua, numero, complemento, tipo) VALUES (?,?,?,?,?,?,?,?)";
        String queryFun = "INSERT INTO Funcionario (cpf, nome, senha, dataNascimento, email, telefone1, telefone2, endereco, ativo) VALUES (?,?,?,?,?,?,?,(select LAST_INSERT_ID()),?)";
        String queryCon = "INSERT INTO Contrato (cargo, salario, dataInicio, dataTermino, funcionario) VALUES (?,?,?,?,?)";

        try {
            conn = c.conectar();
            conn.setAutoCommit(false);

            stmtEnd = conn.prepareStatement(queryEnd);
            stmtFun = conn.prepareStatement(queryFun);
            stmtCon = conn.prepareStatement(queryCon);

            stmtEnd.setString(1, end.getCEP());
            stmtEnd.setString(2, end.getEstado());
            stmtEnd.setString(3, end.getCidade());
            stmtEnd.setString(4, end.getBairro());
            stmtEnd.setString(5, end.getRua());
            stmtEnd.setInt(6, end.getNumero());
            stmtEnd.setString(7, end.getComplemento());
            stmtEnd.setString(8, Character.toString(end.getTipo()));

```

```
81              stmtFun.setString(1, func.getCpf());
82              stmtFun.setString(2, func.getNome());
83              stmtFun.setString(3, func.getSenha());
84              stmtFun.setDate(4, Date.valueOf(func.getDataNascimento()));
85              stmtFun.setString(5, func.getEmail());
86              stmtFun.setString(6, func.getTelefone1());
87              stmtFun.setString(7, func.getTelefone2());
88              stmtFun.setBoolean(8, true);
89
90              stmtCon.setString(1, Character.toString(cont.getCargo()));
91              stmtCon.setDouble(2, cont.getSalario());
92              stmtCon.setDate(3, Date.valueOf(cont.getDataInicio()));
93              stmtCon.setDate(4, Date.valueOf(cont.getDataInicio()));
94              stmtCon.setString(5, func.getCpf());
95
96              stmtEnd.execute();
97              stmtFun.execute();
98              stmtCon.execute();
99
100             conn.commit();
101
102             conn.close();
103             return true;
104         } catch (Exception e) {
105             e.printStackTrace();
106             return false;
107         }
108     }
109
110     public List<Funcionario> getAllFuncionario() throws Exception {
111         String query = "SELECT␣*␣FROM␣Funcionario";
112         List<Funcionario> retorno = new ArrayList<>();
113         Utilidades u = new Utilidades();
114         ConexaoMySQL c = new ConexaoMySQL();
115         Connection conn = null;
116         conn = c.conectar();
117         try {
118             PreparedStatement stmt = conn.prepareStatement(query);
119             ResultSet resultado = stmt.executeQuery();
120             while (resultado.next()) {
121                 Funcionario funcionario = new Funcionario();
122                 funcionario.setCpf(resultado.getString("cpf"));
123                 funcionario.setNome(resultado.getString("nome"));
124                 funcionario.setSenha(resultado.getString("senha"));
125                 funcionario.setDataNascimento(u.toLocalDate(resultado.getDate("dataNascimento")));
126                 funcionario.setEmail(resultado.getString("email"));
127                 funcionario.setTelefone1(resultado.getString("telefone1"));
128                 funcionario.setTelefone2(resultado.getString("telefone2"));
129                 funcionario.setAtivo(resultado.getBoolean("ativo"));
130
131                 retorno.add(funcionario);
132             }
133         } catch (Exception e) {
134             e.printStackTrace();
135         }
136         conn.close();
137         return retorno;
138     }
139
140     public Boolean alteraFuncionario(Funcionario func) throws SQLException {
141         String query = "UPDATE␣Funcionario␣SET␣nome=?,␣senha=?,␣dataNascimento=?,␣email=?,␣telefone1=?,␣telefone2=?,␣
     ativo=?␣WHERE␣cpf=?";
142
143         ConexaoMySQL c = new ConexaoMySQL();
144         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

4.3.2 View

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.chart.*?>
<?import javafx.scene.text.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<?import javafx.collections.*?>

<AnchorPane id="AnchorPane" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
     fx:controller="opencarshop.fornecedor.controller.FornecedorController">
   <children>
      <TabPane prefHeight="328.0" prefWidth="1024.0" tabClosingPolicy="UNAVAILABLE">
        <tabs>
          <Tab text="Identificação">
            <content>
              <AnchorPane minHeight="0.0" minWidth="0.0" prefWidth="1024.0">
                     <children>
                        <TextField fx:id="tf_cnpjCadastro" layoutX="14.0" layoutY="87.0" prefHeight="25.0" prefWidth="
     170.0" promptText="CNPJ" />
                        <TextField fx:id="tf_descricaoCadastro" layoutX="15.0" layoutY="142.0" prefHeight="73.0"
     prefWidth="170.0" promptText="Descrição" />
                        <TextField fx:id="tf_razaoCadastro" layoutX="14.0" layoutY="31.0" prefHeight="25.0" prefWidth="
     350.0" promptText="Razão Social" />
                        <Label layoutX="14.0" layoutY="6.0" text="Razão Social" />
                        <Label layoutX="15.0" layoutY="70.0" text="CNPJ" />
                        <Label layoutX="15.0" layoutY="122.0" text="Descrição" />
                     </children>
                  </AnchorPane>
            </content>
          </Tab>
          <Tab text="Contato">
            <content>
              <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="316.0" prefWidth="1024.0">
                     <children>
                        <TextField fx:id="tf_emailCadastro" layoutX="14.0" layoutY="14.0" prefHeight="25.0" prefWidth="
     350.0" promptText="Email" />
                        <TextField fx:id="tf_telefone1Cadastro" layoutX="14.0" layoutY="50.0" prefWidth="170.0"
     promptText="Telefone 1" />
                        <TextField fx:id="tf_telefone2Cadastro" layoutX="194.0" layoutY="50.0" prefWidth="170.0"
     promptText="Telefone 2" />
                        <TextField fx:id="tf_ruaCadastro" layoutX="14.0" layoutY="116.0" prefWidth="350.0" promptText="
     Rua" />
                        <TextField fx:id="tf_cidadeCadastro" layoutX="14.0" layoutY="187.0" prefWidth="350.0"
     promptText="Cidade" />
                        <TextField fx:id="tf_estadoCadastro" layoutX="14.0" layoutY="222.0" prefWidth="170.0"
     promptText="UF" />
                        <TextField fx:id="tf_bairroCadastro" layoutX="194.0" layoutY="222.0" prefWidth="170.0"
     promptText="Bairro" />
                        <TextField fx:id="tf_cepCadastro" layoutX="14.0" layoutY="256.0" prefWidth="170.0" promptText="
     CEP" />
                        <TextField fx:id="tf_numeroCadastro" layoutX="194.0" layoutY="256.0" prefWidth="170.0"
     promptText="Número" />
                        <TextField fx:id="tf_complementoCadastro" layoutX="14.0" layoutY="151.0" prefWidth="350.0"
     promptText="Complemento" />
                        <ComboBox fx:id="cb_tipoCadastro" layoutX="14.0" layoutY="83.0" prefWidth="350.0" promptText="
     Tipo de Endereço">
                            <items>
                            <FXCollections fx:factory="observableArrayList">
                                <String fx:value="Residencial" />
                                <String fx:value="Comercial" />
                            </FXCollections>
                          </items>
                          </ComboBox>
                     </children>
                  </AnchorPane>
            </content>
          </Tab>
        </tabs>
      </TabPane>
      <Button id="btn_cadastrar" layoutX="151.0" layoutY="367.0" mnemonicParsing="false" onAction="#cadastrar" text="
     Cadastrar" />
      <Label fx:id="resultadoCadastro" alignment="CENTER" contentDisplay="CENTER" layoutX="16.0" layoutY="328.0"
     prefHeight="17.0" prefWidth="346.0" textAlignment="CENTER" />
   </children>
</AnchorPane>
```

Código 14: Cadastrar.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?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␣Alterações" />
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

4.3.3 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
42
43     // TELA DE AUTENTICACAO
44     @FXML
45     private Label labelErro;
46     @FXML
47     private TextField tf_cpf;
```

```
48      @FXML
49      private PasswordField pf_senha;
50
51
52      // TELA DE CADASTRO
53      @FXML
54      private TextField tf_cpfCadastro;
55      @FXML
56      private PasswordField pf_senhaCadastro;
57      @FXML
58      private TextField tf_nomeCadastro;
59      @FXML
60      private DatePicker dp_dataNascimentoCadastro;
61
62      @FXML
63      private ComboBox<String> cb_tipoCadastro;
64      @FXML
65      private TextField tf_emailCadastro;
66      @FXML
67      private TextField tf_telefone1Cadastro;
68      @FXML
69      private TextField tf_telefone2Cadastro;
70      @FXML
71      private TextField tf_ruaCadastro;
72      @FXML
73      private TextField tf_cidadeCadastro;
74      @FXML
75      private TextField tf_estadoCadastro;
76      @FXML
77      private TextField tf_bairroCadastro;
78      @FXML
79      private TextField tf_cepCadastro;
80      @FXML
81      private TextField tf_numeroCadastro;
82      @FXML
83      private TextField tf_complementoCadastro;
84
85      @FXML
86      private TextField tf_salarioCadastro;
87      @FXML
88      private ComboBox<String> cb_cargoCadastro;
89      @FXML
90      private DatePicker dp_dataInicioCadastro;
91      @FXML
92      private DatePicker dp_dataTerminoCadastro;
93
94      @FXML
95      private Label resultadoCadastro;
96
97      // TABELA FUNCIONARIO
98      @FXML
99      private TableColumn<Funcionario, String> col_nome;
100     @FXML
101     private TableColumn<Funcionario, String> col_cpf;
102     @FXML
103     private TableColumn<Funcionario, String> col_telefone1;
104     @FXML
105     private TableColumn<Funcionario, String> col_telefone2;
106     @FXML
107     private TableColumn<Funcionario, String> col_email;
108
109     @FXML
110     private TableView<Funcionario> tbl_funcionario;
111
112         @FXML
113     private CheckBox cb_ativo;
114
115     @FXML
116     private Label confirmaAlteracao;
117
118     static Stage stageAnterior;
119
120     public static void setPrevStage(Stage stage){
121          FuncionarioController.stageAnterior = stage;
122     }
123
124
125     @FXML
126     private void autenticar(ActionEvent event)
127     {
128         Funcionario funcionario;
129         FuncionarioDAO func = new FuncionarioDAO();
130         funcionario = func.getFuncionario(tf_cpf.getText());
131
132         if(funcionario.getCpf() != null)
133         {
134             if(funcionario.getSenha().equals(pf_senha.getText()))
135             {
136                 Parent root = null;
137                 try
138                 {
139                     root = FXMLLoader.load(getClass().getResource("/opencarshop/TelaPrincipal.fxml"));
140                     Scene scene = new Scene(root);
141                     Stage nStage = new Stage();
```

```
142                     nStage.setScene(scene);
143                     //nStage.setMaximized(true);
144                     nStage.setMaxHeight(768);
145                     nStage.setMaxWidth(1024);
146                     nStage.setTitle("OpenCarShop");
147                     nStage.setResizable(false);
148                     nStage.show();
149                     //Stage stage = (Stage) cadastroLink.getScene().getWindow();
150
151
152                     stageAnterior.close();
153                     //stage.close();
154                 }
155                 catch (IOException e)
156                 {
157                     e.printStackTrace();
158                 }
159             }
160             else
161             {
162                 labelErro.setText("Login ou senha errado!!!");
163             }
164         }
165         else
166         {
167             labelErro.setText("Login ou senha errado!!!");
168         }
169     }
170
171     @FXML
172     private void cadastrar(ActionEvent event) throws ParseException {
173         //cb_cargoCadastro.setItems(cargos);
174         Funcionario  func    = new Funcionario();
175         Endereco     end     = new Endereco();
176         Contrato     contr   = new Contrato();
177         FuncionarioDAO f     = new FuncionarioDAO();
178         Utilidades u = new Utilidades();
179
180         // OBJETO FUNCIONARIO
181         func.setCpf(tf_cpfCadastro.getText());
182         func.setNome(tf_nomeCadastro.getText());
183         func.setSenha(pf_senhaCadastro.getText());
184         func.setDataNascimento(dp_dataNascimentoCadastro.getValue());
185         func.setDataNascimento(dp_dataNascimentoCadastro.getValue());
186         func.setEmail(tf_emailCadastro.getText());
187         func.setTelefone1(tf_telefone1Cadastro.getText());
188         func.setTelefone2(tf_telefone2Cadastro.getText());
189
190         // OBJETO ENDERECO
191         end.setCEP(tf_cepCadastro.getText());
192         end.setEstado(tf_estadoCadastro.getText());
193         end.setCidade(tf_cidadeCadastro.getText());
194         end.setBairro(tf_bairroCadastro.getText());
195         end.setRua(tf_ruaCadastro.getText());
196         end.setNumero(Integer.parseInt(tf_numeroCadastro.getText()));
197         end.setComplemento(tf_complementoCadastro.getText());
198         end.setTipo(cb_tipoCadastro.getValue().charAt(0));
199
200         // OBJETO CONTRATO
201         contr.setCargo( cb_cargoCadastro.getValue().charAt(0));
202         contr.setSalario(DecimalFormat.getInstance().parse(tf_salarioCadastro.getText()).doubleValue());
203         contr.setDataInicio(dp_dataInicioCadastro.getValue());
204         contr.setDataTermino(dp_dataTerminoCadastro.getValue());
205
206
207         if(f.cadastraFuncionario(func, end, contr))
208         {
209             resultadoCadastro.setText("Cadastrado com sucesso!!");
210         }
211         else
212         {
213             resultadoCadastro.setText("Erro ao cadastrar!! Tente novamente.");
214         }
215     }
216
217 @FXML
218     private void alterarCadastro(ActionEvent event) throws Exception {
219         Funcionario  func    = new Funcionario();
220         func.setCpf(tf_cpfCadastro.getText());
221         func.setNome(tf_nomeCadastro.getText());
222         func.setSenha(pf_senhaCadastro.getText());
223         func.setDataNascimento(dp_dataNascimentoCadastro.getValue());
224         func.setEmail(tf_emailCadastro.getText());
225         func.setTelefone1(tf_telefone1Cadastro.getText());
226         func.setTelefone2(tf_telefone2Cadastro.getText());
227         func.setAtivo(cb_ativo.isSelected());
228
229         FuncionarioDAO f     = new FuncionarioDAO();
230         if(f.alteraFuncionario(func))
231         {
232             confirmaAlteracao.setText("Alteração realizada com sucesso!!");
233         }else
234         {
235             confirmaAlteracao.setText("Erro ao realizar a alteração!!");
```

```
236         }
237     }
238    public void selecionarItemTablelaFuncionario(Funcionario funcionario){
239         if (funcionario.getCpf() != null) {
240               tf_cpfCadastro.setText(funcionario.getCpf());
241               tf_nomeCadastro.setText(funcionario.getNome());
242               pf_senhaCadastro.setText(funcionario.getSenha());
243               tf_emailCadastro.setText(funcionario.getEmail());
244               tf_telefone1Cadastro.setText(funcionario.getTelefone1());
245               tf_telefone2Cadastro.setText(funcionario.getTelefone2());
246               dp_dataNascimentoCadastro.setValue(funcionario.getDataNascimento());
247               cb_ativo.setSelected(funcionario.getAtivo());
248 
249         }
250     }
251 
252 
253     private void carregaTabelaFuncionario() throws Exception
254     {
255         col_nome.setCellValueFactory(new PropertyValueFactory<>("nome"));
256         col_cpf.setCellValueFactory(new PropertyValueFactory<>("cpf"));
257         col_telefone1.setCellValueFactory(new PropertyValueFactory<>("telefone1"));
258         col_telefone2.setCellValueFactory(new PropertyValueFactory<>("telefone2"));
259         col_email.setCellValueFactory(new PropertyValueFactory<>("email"));
260 
261         FuncionarioDAO f = new FuncionarioDAO();
262         List<Funcionario> listaFuncionario = f.getAllFuncionario();
263         ObservableList<Funcionario> observableListFuncionatio;
264 
265         observableListFuncionatio = FXCollections.observableArrayList(listaFuncionario);
266         tbl_funcionario.setItems(observableListFuncionatio);
267     }
268 
269     @Override
270     public void initialize(URL url, ResourceBundle rb) {
271 try {
272             carregaTabelaFuncionario();
273             tbl_funcionario.getSelectionModel().selectedItemProperty().addListener(
274                 (observable, oldValue, newValue) -> selecionarItemTablelaFuncionario(newValue));
275         } catch (Exception ex) {
276             //Logger.getLogger(FuncionarioController.class.getName()).log(Level.SEVERE, null, ex);
277         }
278 
279     }
280 
281 }
```

Código 16: FuncionarioController.java

4.4 Pacote Peça

4.4.1 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     /**
18      * @return the id
19      */
20     public int getId() {
21         return id;
22     }
23 
24     /**
25      * @param id the id to set
26      */
27     public void setId(int id) {
28         this.id = id;
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.peca = 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.peca.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      public ItemPeca buscar(ItemPeca itemPeca) {
63          String sql = "SELECT␣*␣FROM␣itensDeVenda␣WHERE␣id=?";
64          ItemPeca retorno = new ItemPeca();
65
66          try {
67              PreparedStatement stmt = connection.prepareStatement(sql);
68              stmt.setInt(1, itemPeca.getId());
69              ResultSet resultado = stmt.executeQuery();
70              if (resultado.next()) {
71                  Peca peca = new Peca();
72                  VendaPeca venda = new VendaPeca();
73                  itemPeca.setId(resultado.getInt("id"));
74                  itemPeca.setQuantidadeVendida(resultado.getInt("quantidadeVendida"));
75                  itemPeca.setValor(resultado.getDouble("valor"));
76
77                  peca.setId(resultado.getInt("id"));
78                  venda.setId(resultado.getInt("id"));
79
80                  //Obtendo os dados completos do Cliente associado à Venda
81                  PecaDAO pecaDAO = new PecaDAO();
82                  pecaDAO.setConnection(connection);
83                  peca = pecaDAO.buscar(peca);
84
85                  VendaPecaDAO vendaDAO = new VendaPecaDAO();
86                  vendaDAO.setConnection(connection);
87                  venda = vendaDAO.buscar(venda);
88
89                  itemPeca.setPeca(peca);
90                  itemPeca.setVendaPeca(venda);
91
92                  retorno = itemPeca;
93              }
94          } catch (SQLException ex) {
95              Logger.getLogger(ItemPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
96          }
97          return retorno;
98
99      }
100
101     public List<ItemPeca> listar() {
102         String sql = "SELECT␣*␣FROM␣ItemPeca";
103         List<ItemPeca> retorno = new ArrayList<>();
104         try {
105             PreparedStatement stmt = connection.prepareStatement(sql);
106             ResultSet resultado = stmt.executeQuery();
107             while (resultado.next()) {
108                 Peca peca = new Peca();
109                 ItemPeca item = new ItemPeca();
110                 VendaPeca venda = new VendaPeca();
111
112                 item.setId(resultado.getInt("id"));
113                 item.setValor(resultado.getDouble("valor"));
114                 item.setQuantidadeVendida(resultado.getInt("quantidadeVendida"));
115
116                 peca.setId(resultado.getInt("id"));
117                 venda.setId(resultado.getInt("id"));
118
119                 PecaDAO pecaDAO = new PecaDAO();
120                 pecaDAO.setConnection(connection);
121                 peca = pecaDAO.buscar(peca);
122
123                 VendaPecaDAO vendaDAO = new VendaPecaDAO();
124                 vendaDAO.setConnection(connection);
125                 venda = vendaDAO.buscar(venda);
126
127                 item.setPeca(peca);
128                 item.setVendaPeca(venda);
129
```

```
                retorno.add(item);
            }
        } catch (SQLException ex) {
            Logger.getLogger(ItemPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
        return retorno;
    }

    public List<ItemPeca> listarPorVenda(VendaPeca venda) {
        String sql = "SELECT * FROM ItemPeca WHERE id=?";
        List<ItemPeca> retorno = new ArrayList<>();
        try {
            PreparedStatement stmt = connection.prepareStatement(sql);
            stmt.setInt(1, venda.getId());
            ResultSet resultado = stmt.executeQuery();
            while (resultado.next()) {
                ItemPeca item = new ItemPeca();
                Peca peca = new Peca();
                VendaPeca v = new VendaPeca();
                item.setId(resultado.getInt("id"));
                item.setQuantidadeVendida(resultado.getInt("quantidadeVendida"));
                item.setValor(resultado.getDouble("valor"));

                peca.setId(resultado.getInt("id"));
                v.setId(resultado.getInt("id"));

                //Obtendo os dados completos do Produto associado ao Item de Venda
                PecaDAO pecaDAO = new PecaDAO();
                pecaDAO.setConnection(connection);
                peca = pecaDAO.buscar(peca);

                item.setPeca(peca);
                item.setVendaPeca(v);

                retorno.add(item);
            }
        } catch (SQLException ex) {
            Logger.getLogger(ItemPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
        return retorno;
    }

}
```

Código 18: ItemPecaDAO.java

```
package opencarshop.peca.model;

import java.io.Serializable;

public class Peca implements Serializable {

    private int id;
    private String nome;
    private double valor;
    private int quantidade;
    private boolean ativa;

    public Peca() {

    }

    public Peca(int id, String nome, double preco, int quantidade, boolean ativa) {
        this.id = id;
        this.nome = nome;
        this.valor = preco;
        this.quantidade = quantidade;
        this.ativa = ativa;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public double getValor() {
        return valor;
    }

    public void setValor(double valor) {
        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             }
125         } catch (SQLException ex) {
126             Logger.getLogger(PecaDAO.class.getName()).log(Level.SEVERE, null, ex);
127         }
128         return retorno;
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
151 }
```

Código 20: PecaDAO.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package opencarshop.peca.model;

import java.sql.Date;
import java.util.List;
import opencarshop.cliente.model.Cliente;

public class VendaPeca {

    private int id;
    private List<ItemPeca> itemsVendidos;
    private Date dataVenda;
    private double valor;
    private boolean pago;
    private Cliente cliente;

    public VendaPeca() {

    }

    /**
     * @return the id
     */
    public int getId() {
        return id;
    }

    /**
     * @param id the id to set
     */
    public void setId(int id) {
        this.id = id;
    }

    /**
     * @return the itemsVendidos
     */
    public List<ItemPeca> getItemsVendidos() {
        return itemsVendidos;
    }

    /**
     * @param itemsVendidos the itemsVendidos to set
     */
    public void setItemsVendidos(List<ItemPeca> itemsVendidos) {
        this.itemsVendidos = itemsVendidos;
    }

    /**
     * @return the dataVenda
     */
    public Date getDataVenda() {
        return dataVenda;
    }

    /**
     * @param dataVenda the dataVenda to set
     */
    public void setDataVenda(Date dataVenda) {
        this.dataVenda = dataVenda;
    }

    /**
     * @return the valor
     */
    public double getValor() {
        return valor;
    }

    /**
     * @param valor the valor to set
     */
    public void setValor(double valor) {
        this.valor = valor;
    }

    /**
     * @return the pago
     */
    public boolean isPago() {
        return pago;
    }

    /**
```

```
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.peca.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 à 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 à 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             return retorno;
```

```
        } catch (SQLException ex) {
            Logger.getLogger(VendaPecaDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
        return retorno;
    }

}
```

Código 22: VendaPecaDAO.java

4.4.2 View

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<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">
    <children>
        <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">
            <items>
                <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="160.0" prefWidth="100.0">
                    <children>
                        <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">
                            <columns>
                                <TableColumn fx:id="tableColumnPecaNome" prefWidth="75.0" text="Nome" />
                                <TableColumn fx:id="tableColumnPecaQuantidade" prefWidth="75.0" text="Quantidade" />
                            </columns>
                            <columnResizePolicy>
                                <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
                            </columnResizePolicy>
                        </TableView>
                    </children>
                </AnchorPane>
                <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="160.0" prefWidth="100.0">
                    <children>
                        <Label layoutX="112.0" layoutY="14.0" text="Detalhes␣Da␣Peca">
                            <font>
                                <Font name="System␣Bold" size="14.0" />
                            </font>
                        </Label>
                        <GridPane layoutX="54.0" layoutY="90.0" AnchorPane.leftAnchor="75.0">
                            <columnConstraints>
                                <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
                                <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
                            </columnConstraints>
                            <rowConstraints>
                                <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                                <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                                <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                                <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                            </rowConstraints>
                            <children>
                                <Label text="Código" />
                                <Label text="Nome" GridPane.rowIndex="1" />
                                <Label text="Preco" GridPane.rowIndex="2" />
                                <Label text="Quantidade" GridPane.rowIndex="3" />
                                <Label fx:id="labelPecaCodigo" GridPane.columnIndex="1" />
                                <Label fx:id="labelPecaNome" GridPane.columnIndex="1" GridPane.rowIndex="1" />
                                <Label fx:id="labelPecaPreco" GridPane.columnIndex="1" GridPane.rowIndex="2" />
                                <Label fx:id="labelPecaQuantidade" GridPane.columnIndex="1" GridPane.rowIndex="3" />
                            </children>
                        </GridPane>
                        <Group AnchorPane.bottomAnchor="20.0" AnchorPane.rightAnchor="20.0">
                            <children>
                                <Button fx:id="buttonInserir" layoutX="97.0" layoutY="293.0" mnemonicParsing="false"
     onAction="#handleButtonInserir" prefHeight="25.0" prefWidth="64.0" text="Inserir" />
                                <Button fx:id="buttonAlterar" layoutX="169.0" layoutY="293.0" mnemonicParsing="false"
     onAction="#handleButtonAlterar" prefHeight="25.0" prefWidth="64.0" text="Alterar" />
                                <Button fx:id="buttonInativar" layoutX="240.0" layoutY="293.0" mnemonicParsing="false"
     onAction="#handleButtonInativar" text="Inativar" />
                            </children>
                        </Group>
                    </children>
                </AnchorPane>
            </items>
        </SplitPane>
    </children>
</AnchorPane>
```

Código 23: CadastroPeca.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<AnchorPane id="AnchorPane" prefHeight="183.0" prefWidth="340.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="opencarshop.peca.controller.CadastroPecaDialog">
    <children>
        <GridPane layoutX="20.0" layoutY="8.0" prefHeight="120.0" prefWidth="298.0">
            <columnConstraints>
                <ColumnConstraints hgrow="SOMETIMES" maxWidth="168.0" minWidth="10.0" prefWidth="128.0" />
                <ColumnConstraints hgrow="SOMETIMES" maxWidth="248.0" minWidth="10.0" prefWidth="218.0" />
            </columnConstraints>
            <rowConstraints>
                <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
            </rowConstraints>
            <children>
                <Label fx:id="labelPecaNome" text="Nome" />
                <Label fx:id="labelPecaPreco" text="Preco" GridPane.rowIndex="1" />
                <Label fx:id="labelPecaQuantidade" text="Quantidade" GridPane.rowIndex="2" />
                <TextField fx:id="textFieldPecaNome" GridPane.columnIndex="1" />
                <TextField fx:id="textFieldPecaPreco" GridPane.columnIndex="1" GridPane.rowIndex="1" />
                <TextField fx:id="textFieldPecaQuantidade" GridPane.columnIndex="1" GridPane.rowIndex="2" />
            </children>
        </GridPane>
        <Group layoutX="15.0" AnchorPane.bottomAnchor="20.0" AnchorPane.rightAnchor="20.0">
            <children>
                <Button fx:id="buttonConfirmar" layoutX="129.0" layoutY="137.0" mnemonicParsing="false" onAction="#handleButtonConfirmar" text="Confirmar" />
                <Button fx:id="buttonCancelar" layoutX="208.0" layoutY="137.0" mnemonicParsing="false" onAction="#handleButtonCancelar" text="Cancelar" />
            </children>
        </Group>
    </children>
</AnchorPane>
```

Código 24: CadastroPecaDialog.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.chart.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>


<AnchorPane id="AnchorPane" prefHeight="400.0" prefWidth="600.0" xmlns:fx="http://javafx.com/fxml/1" xmlns="http://javafx.com/javafx/8" fx:controller="opencarshop.peca.controller.GraficosVendasPorMesController">
    <children>
        <BarChart fx:id="barChart" layoutX="37.0" layoutY="18.0" prefHeight="364.0" prefWidth="500.0" title="Vendas␣por␣Mês">
            <xAxis>
                <CategoryAxis fx:id="categoryAxis" side="BOTTOM" />
            </xAxis>
            <yAxis>
                <NumberAxis fx:id="numberAxis" side="LEFT" />
            </yAxis>
        </BarChart>
    </children>
</AnchorPane>
```

Código 25: GraficosVendasPorMes.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<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.peca.controller.VendaPecasController">
   <children>
```

```
12        <SplitPane dividerPositions="0.46153846153846156" layoutX="73.0" layoutY="21.0" 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="398.0" prefWidth="269.0">
15                 <children>
16                    <TableView fx:id="tableViewVendas" layoutX="28.0" layoutY="29.0" prefHeight="398.0" prefWidth="275.0"
       AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0
      ">
17                      <columns>
18                        <TableColumn fx:id="tableColumnVendaCodigo" prefWidth="61.0" text="Código" />
19                        <TableColumn fx:id="tableColumnVendaData" prefWidth="63.0" text="Data" />
20                          <TableColumn fx:id="tableColumnVendaValorTotal" prefWidth="66.0" text="Valor Total" />
21                          <TableColumn fx:id="tableColumnVendaNomeCliente" prefWidth="98.0" text="Nome do Cliente" />
22                      </columns>
23                    </TableView>
24                 </children>
25              </AnchorPane>
26            <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="160.0" prefWidth="100.0">
27                 <children>
28                    <Label layoutX="100.0" layoutY="14.0" text="Detalhes da Venda">
29                       <font>
30                          <Font name="System Bold" size="14.0" />
31                       </font>
32                    </Label>
33                    <GridPane layoutX="25.0" layoutY="62.0" prefHeight="146.0" prefWidth="251.0">
34                      <columnConstraints>
35                        <ColumnConstraints hgrow="SOMETIMES" maxWidth="113.0" minWidth="10.0" prefWidth="84.0" />
36                        <ColumnConstraints hgrow="SOMETIMES" maxWidth="202.0" minWidth="10.0" prefWidth="180.0" />
37                      </columnConstraints>
38                      <rowConstraints>
39                        <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
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="Código" />
47                          <Label text="Data" GridPane.rowIndex="1" />
48                          <Label text="Valor" GridPane.rowIndex="2" />
49                          <Label text="Pago" GridPane.rowIndex="3" />
50                          <Label text="Cliente" GridPane.rowIndex="4" />
51                          <Label fx:id="labelVendaCodigo" GridPane.columnIndex="1" />
52                          <Label fx:id="labelVendaData" GridPane.columnIndex="1" GridPane.rowIndex="1" />
53                          <Label fx:id="labelVendaValor" GridPane.columnIndex="1" GridPane.rowIndex="2" />
54                          <Label fx:id="labelVendaPago" GridPane.columnIndex="1" GridPane.rowIndex="3" />
55                          <Label fx:id="labelVendaCliente" GridPane.columnIndex="1" GridPane.rowIndex="4" />
56                       </children>
57                    </GridPane>
58                    <Group AnchorPane.bottomAnchor="20.0" AnchorPane.rightAnchor="20.0" />
59                    <Button fx:id="buttonInserir" layoutX="100.0" layoutY="342.0" mnemonicParsing="false" onAction="#
      handleButtonInserir" prefHeight="25.0" prefWidth="65.0" text="Inserir" />
60                    <TableView layoutX="22.0" layoutY="219.0" prefHeight="102.0" prefWidth="257.0">
61                      <columns>
62                        <TableColumn fx:id="tableColumnNomePeca" prefWidth="75.0" text="Peca" />
63                        <TableColumn fx:id="tableColumnQuantidadePeca" prefWidth="75.0" text="Quantidade" />
64                          <TableColumn fx:id="tableColumnValorPeca" prefWidth="75.0" text="Preco" />
65                      </columns>
66                    </TableView>
67                 </children>
68              </AnchorPane>
69          </items>
70        </SplitPane>
71     </children>
72 </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://
       javafx.com/fxml/1" fx:controller="opencarshop.peca.controller.VendaPecasDialogController">
10     <children>
11        <GridPane layoutX="81.0" layoutY="33.0" prefHeight="251.0" prefWidth="324.0">
12           <columnConstraints>
13              <ColumnConstraints hgrow="SOMETIMES" maxWidth="95.0" minWidth="7.0" prefWidth="60.0" />
14              <ColumnConstraints hgrow="SOMETIMES" maxWidth="267.0" minWidth="10.0" prefWidth="264.0" />
15           </columnConstraints>
16           <rowConstraints>
17              <RowConstraints maxHeight="37.0" minHeight="10.0" prefHeight="25.0" vgrow="SOMETIMES" />
18              <RowConstraints maxHeight="64.0" minHeight="10.0" prefHeight="26.0" vgrow="SOMETIMES" />
19              <RowConstraints maxHeight="77.0" minHeight="10.0" prefHeight="23.0" vgrow="SOMETIMES" />
20              <RowConstraints maxHeight="91.0" minHeight="10.0" prefHeight="27.0" vgrow="SOMETIMES" />
21              <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" />
```

```
            </rowConstraints>
            <children>
                <Label text="Cliente" />
                <Label text="Data" GridPane.rowIndex="1" />
                <Label text="Pago" GridPane.rowIndex="2" />
                <Label text="Peca" GridPane.rowIndex="3" />
                <Label text="Itens" GridPane.rowIndex="4" />
                <Label text="Valor" GridPane.rowIndex="5" />
                <ComboBox fx:id="comboBoxVendaCliente" prefHeight="25.0" prefWidth="173.0" promptText="Selecione␣o␣cliente␣..." GridPane.columnIndex="1" />
                <DatePicker fx:id="datePickerVendaData" GridPane.columnIndex="1" GridPane.rowIndex="1" />
                <CheckBox fx:id="checkBoxVendaPago" mnemonicParsing="false" GridPane.columnIndex="1" GridPane.rowIndex="2" />
                <TableView fx:id="tableViewItensDeVenda" prefHeight="200.0" prefWidth="200.0" GridPane.columnIndex="1" GridPane.rowIndex="4">
                    <columns>
                        <TableColumn fx:id="tableColumnItemDeVendaPeca" prefWidth="127.0" text="Peca" />
                        <TableColumn fx:id="tableColumnItemDeVendaQuantidade" prefWidth="75.0" text="Quantidade" />
                        <TableColumn fx:id="tableColumnItemDeVendaValor" prefWidth="60.0" text="Valor" />
                    </columns>
                </TableView>
                <TextField fx:id="textFieldVendaValor" editable="false" GridPane.columnIndex="1" GridPane.rowIndex="5" />
                <HBox prefHeight="100.0" prefWidth="200.0" GridPane.columnIndex="1" GridPane.rowIndex="3">
                    <children>
                        <ComboBox fx:id="comboBoxVendaPeca" prefHeight="25.0" prefWidth="179.0" promptText="Selecione␣a␣peca␣..." />
                        <TextField fx:id="textFieldVendaQuantidade" prefHeight="25.0" prefWidth="40.0" />
                        <Button fx:id="buttonAdicionar" mnemonicParsing="false" onAction="#handleButtonAdicionar" prefHeight="25.0" prefWidth="68.0" text="+" />
                    </children>
                </HBox>
            </children>
        </GridPane>
        <Button fx:id="buttonConfirmar" layoutX="252.0" layoutY="297.0" mnemonicParsing="false" onAction="#handleButtonConfirmar" text="Confirmar" />
        <Button fx:id="buttonCancelar" layoutX="343.0" layoutY="297.0" mnemonicParsing="false" onAction="#handleButtonCancelar" text="Cancelar" />
    </children>
</AnchorPane>
```

Código 27: VendaPecasDialog.fxml

4.4.3 Controller

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package opencarshop.peca.controller;

import java.io.IOException;
import java.net.URL;
import java.sql.Connection;
import java.util.List;
import java.util.ResourceBundle;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import opencarshop.peca.model.PecaDAO;
import opencarshop.util.ConexaoMySQL;
import opencarshop.peca.model.Peca;

/**
 *
 * @author Dimitri
 */
public class CadastroPeca implements Initializable {

    @FXML
    private TableView<Peca> tableViewPecas;
    @FXML
    private TableColumn<Peca, String> tableColumnPecaNome;
    @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 manipulação 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 alterações na seleção 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 buttonConfirmarClicked = showCadastroPecaDialog(peca);
107         System.out.println(peca.getNome());
108         if (buttonConfirmarClicked) {
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 buttonConfirmarClicked = showCadastroPecaDialog(peca);
119             if (buttonConfirmarClicked) {
120                 pecaDAO.atualizar(peca);
121                 carregarTableViewPeca();
122             }
123         } else {
124             Alert alert = new Alert(Alert.AlertType.ERROR);
125             alert.setContentText("Por␣favor,␣escolha␣uma␣peca␣na␣Tabela!");
126             alert.show();
127         }
128     }
129
130     @FXML
131     public void handleButtonInativar() throws IOException {
132         Peca peca = tableViewPecas.getSelectionModel().getSelectedItem();
133         if (peca != null) {
134             pecaDAO.inativar(peca);
135             carregarTableViewPeca();
136         } else {
137             Alert alert = new Alert(Alert.AlertType.ERROR);
```

```
138             alert.setContentText("Por␣favor,␣escolha␣uma␣peca␣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 Estágio de Diálogo (Stage Dialog)
149         Stage dialogStage = new Stage();
150         dialogStage.setTitle("Cadastro␣de␣Pecas");
151         Scene scene = new Scene(page);
152         dialogStage.setScene(scene);
153 
154         // Setando o peca 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.isButtonConfirmarClicked();
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 buttonConfirmarClicked = 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 isButtonConfirmarClicked() {
51         return buttonConfirmarClicked;
52     }
53 
54     public void setButtonConfirmarClicked(boolean buttonConfirmarClicked) {
55         this.buttonConfirmarClicked = buttonConfirmarClicked;
56     }
57 
58     public Peca getPeca() {
59         return peca;
60     }
```

```
61
62      public void setPeca(Peca peca) {
63          this.peca = peca;
64          this.textFieldPecaNome.setText(peca.getNome());
65          String preco = String.valueOf(peca.getValor());
66          this.textFieldPecaPreco.setText(preco);
67          String quantidade = String.valueOf(peca.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.peca.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.peca.model.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 manipulação 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 an 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 mês 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";
```

```
            case 3:
                return "Mar";
            case 4:
                return "Abr";
            case 5:
                return "Mai";
            case 6:
                return "Jun";
            case 7:
                return "Jul";
            case 8:
                return "Ago";
            case 9:
                return "Set";
            case 10:
                return "Out";
            case 11:
                return "Nov";
            case 12:
                return "Dez";
            default:
                return "";
        }
    }

}
```

Código 30: GraficosVendasPorMesController.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package opencarshop.peca.controller;

import java.io.IOException;
import java.net.URL;
import java.sql.Connection;
import java.sql.SQLException;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;
import java.util.ResourceBundle;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import opencarshop.peca.model.PecaDAO;
import opencarshop.peca.model.ItemPecaDAO;
import opencarshop.peca.model.ItemPeca;
import opencarshop.peca.model.Peca;
import opencarshop.peca.model.VendaPeca;
import opencarshop.peca.model.VendaPecaDAO;
import opencarshop.util.ConexaoMySQL;

public class VendaPecasController implements Initializable {

    @FXML
    private TableView<VendaPeca> tableViewVendas;
    @FXML
    private TableColumn<VendaPeca, Integer> tableColumnVendaCodigo;
    @FXML
    private TableColumn<VendaPeca, LocalDate> tableColumnVendaData;
    @FXML
    private TableColumn<VendaPeca, String> tableColumnVendaValorTotal;
    @FXML
    private TableColumn<VendaPeca, String> tableColumnVendaNomeCliente;

    @FXML
    private TableView<ItemPeca> tableViewPecasVendidas;
    @FXML
    private TableColumn<ItemPeca, Peca> tableColumnNomePeca;
    @FXML
    private TableColumn<ItemPeca, Double> tableColumnValorPeca;
    @FXML
    private TableColumn<ItemPeca, Integer> tableColumnQuantidadePeca;

    @FXML
    private Button buttonInserir;
    @FXML
```

```
    private Label labelVendaCodigo;
    @FXML
    private Label labelVendaData;
    @FXML
    private Label labelVendaValor;
    @FXML
    private Label labelVendaPago;
    @FXML
    private Label labelVendaCliente;

    private List<VendaPeca> listVendas;
    private ObservableList<VendaPeca> observableListVendas;

    private final ConexaoMySQL database = new ConexaoMySQL();
    private Connection connection;
    private final VendaPecaDAO vendaDAO = new VendaPecaDAO();
    private final ItemPecaDAO itemDeVendaDAO = new ItemPecaDAO();
    private final PecaDAO pecaDAO = new PecaDAO();

    @Override
    public void initialize(URL location, ResourceBundle resources) {
        try {
            connection = database.conectar();
            vendaDAO.setConnection(connection);
        } catch (Exception ex) {
            Logger.getLogger(VendaPecasController.class.getName()).log(Level.SEVERE, null, ex);
        }
        carregarTableViewVendas();
        selecionarItemTableViewVendas(null);
        // Listen acionado diante de quaisquer alterações na seleção de itens do TableView
        tableViewVendas.getSelectionModel().selectedItemProperty().addListener(
                (observable, oldValue, newValue) -> selecionarItemTableViewVendas(newValue));
    }

    public void mostrarItemTable(ItemPeca peca) {
        tableColumnNomePeca.setCellValueFactory(new PropertyValueFactory<>("peca"));
        tableColumnValorPeca.setCellValueFactory(new PropertyValueFactory<>("valor"));
        tableColumnQuantidadePeca.setCellValueFactory(new PropertyValueFactory<>("quantidadeVenda"));
    }

    public void selecionarItemTableViewVendas(VendaPeca venda) {
        if (venda != null) {
            labelVendaCodigo.setText(String.valueOf(venda.getId()));
            labelVendaData.setText(String.valueOf(venda.getDataVenda()));
            labelVendaValor.setText(String.format("%.2f", venda.getValor()));
            String pago;
            if (venda.isPago() == true) {
                pago = "Sim";
            } else {
                pago = "Nao";
            }
            labelVendaPago.setText(pago);
            labelVendaCliente.setText(venda.getCliente().getNome());

            for (ItemPeca i : venda.getItemsVendidos()) {
                //tableViewPecasVendidas.getSelectionModel().selectedItemProperty().addListener(
                //((observable, oldValue, newValue) -> mostrarItemTable(i));
                mostrarItemTable(i);
            }

        } else {
            labelVendaCodigo.setText("");
            labelVendaData.setText("");
            labelVendaValor.setText("");
            labelVendaPago.setText("");
            labelVendaCliente.setText("");
        }
    }

    public void carregarTableViewVendas() {
        tableColumnVendaCodigo.setCellValueFactory(new PropertyValueFactory<>("id"));
        tableColumnVendaData.setCellValueFactory(new PropertyValueFactory<>("dataVenda"));
        tableColumnVendaValorTotal.setCellValueFactory(new PropertyValueFactory<>("valor"));
        tableColumnVendaNomeCliente.setCellValueFactory(new PropertyValueFactory<>("cliente"));

        listVendas = vendaDAO.listar();

        observableListVendas = FXCollections.observableArrayList(listVendas);
        tableViewVendas.setItems(observableListVendas);
    }

    @FXML
    public void handleButtonInserir() throws IOException, SQLException {
        VendaPeca venda = new VendaPeca();
        List<ItemPeca> listItensDeVenda = new ArrayList<>();
        venda.setItemsVendidos(listItensDeVenda);
        boolean buttonConfirmarClicked = showVendaPecasDialog(venda);

        if (buttonConfirmarClicked) {
            try {
                vendaDAO.inserir(venda);

                connection.setAutoCommit(false);
                vendaDAO.setConnection(connection);
```

```
                itemDeVendaDAO.setConnection(connection);
                pecaDAO.setConnection(connection);

                for (ItemPeca itemPeca : venda.getItemsVendidos()) {
                    Peca peca = itemPeca.getPeca();
                    itemPeca.setVendaPeca(vendaDAO.buscarUltimaVenda());
                    itemDeVendaDAO.inserir(itemPeca);
                    peca.setQuantidade(peca.getQuantidade() - itemPeca.getQuantidadeVendida());
                    pecaDAO.atualizar(peca);
                }
                connection.commit();
                carregarTableViewVendas();
            } catch (SQLException ex) {
                try {
                    connection.rollback();
                } catch (SQLException ex1) {
                    Logger.getLogger(VendaPecasController.class.getName()).log(Level.SEVERE, null, ex1);
                }
                Logger.getLogger(VendaPecasController.class.getName()).log(Level.SEVERE, null, ex);
            }
        }

    }

    public boolean showVendaPecasDialog(VendaPeca venda) throws IOException {

        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(VendaPecasDialogController.class.getResource("/opencarshop/peca/view/VendaPecasDialog.fxml")
     );
        AnchorPane page = (AnchorPane) loader.load();

        // Criando um Estágio de Diálogo (Stage Dialog)
        Stage dialogStage = new Stage();
        dialogStage.setTitle("Registro de Vendas");
        Scene scene = new Scene(page);
        dialogStage.setScene(scene);
        // Setando a Venda no Controller.
        VendaPecasDialogController controller = loader.getController();
        controller.setDialogStage(dialogStage);
        controller.setVenda(venda);
        // Mostra o Dialog e espera até que o usuário o feche
        dialogStage.showAndWait();
        return controller.isButtonConfirmarClicked();
    }

}
```

Código 31: VendaPecasController.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package opencarshop.peca.controller;

import java.net.URL;
import java.sql.Date;
import java.util.List;
import java.util.ResourceBundle;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.CheckBox;
import javafx.scene.control.ComboBox;
import javafx.scene.control.DatePicker;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.stage.Stage;
import opencarshop.cliente.model.Cliente;
import opencarshop.peca.model.ItemPeca;
import opencarshop.peca.model.Peca;
import opencarshop.peca.model.PecaDAO;
import opencarshop.cliente.model.ClienteDAO;
import opencarshop.peca.model.VendaPeca;

public class VendaPecasDialogController implements Initializable {

    @FXML
    private ComboBox comboBoxVendaCliente;
    @FXML
    private DatePicker datePickerVendaData;
    @FXML
    private CheckBox checkBoxVendaPago;
    @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<>("peca"));
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              for (Cliente c : listClientes) {
92                  System.out.println(c.getNome());
93              }
94
95          } catch (Exception ex) {
96              Logger.getLogger(VendaPecasDialogController.class.getName()).log(Level.SEVERE, null, ex);
97          }
98          observableListClientes = FXCollections.observableArrayList(listClientes);
99          comboBoxVendaCliente.setItems(observableListClientes);
100         System.out.println(comboBoxVendaCliente.getItems());
101
102     }
103
104     public void carregarComboBoxPecas() {
105         listPecas = pecaDAO.listar();
106         observableListPecas = FXCollections.observableArrayList(listPecas);
107         comboBoxVendaPeca.setItems(observableListPecas);
108     }
109
110     public Stage getDialogStage() {
111         return dialogStage;
112     }
113
114     public void setDialogStage(Stage dialogStage) {
115         this.dialogStage = dialogStage;
116     }
117
118     public boolean isButtonConfirmarClicked() {
119         return buttonConfirmarClicked;
120     }
121
122     public VendaPeca getVenda() {
123         return this.venda;
124     }
125
126     public void setVenda(VendaPeca venda) {
127         this.venda = venda;
128     }
129
130     @FXML
131     public void handleButtonAdicionar() {
132         Peca peca;
133         ItemPeca itemPeca = new ItemPeca();
134
135         if (comboBoxVendaPeca.getSelectionModel().getSelectedItem() != null) {
136             peca = (Peca) comboBoxVendaPeca.getSelectionModel().getSelectedItem();
137             int quantidade = Integer.parseInt(textFieldVendaQuantidade.getText());
```

```
138             if (peca.getQuantidade() >= quantidade) {
139                 itemPeca.setPeca(peca);
140                 itemPeca.setValor(peca.getValor() * quantidade);
141                 itemPeca.setQuantidadeVendida(quantidade);
142 
143                 venda.getItemsVendidos().add(itemPeca);
144                 venda.setValor(venda.getValor() + itemPeca.getValor());
145 
146                 observableListItensDeVenda = FXCollections.observableArrayList(itemPeca);
147                 tableViewItensDeVenda.getItems().add(itemPeca);
148 
149                 textFieldVendaValor.setText(String.format("%.2f", venda.getValor()));
150             } else {
151                 Alert alert = new Alert(Alert.AlertType.ERROR);
152                 alert.setHeaderText("Problemas na escolha do produto!");
153                 alert.setContentText("Não existe a quantidade de produtos disponíveis no estoque!");
154                 alert.show();
155             }
156         }
157     }
158 
159     public void handleButtonConfirmar() {
160 
161         Cliente c = (Cliente) comboBoxVendaCliente.getSelectionModel().getSelectedItem();
162         venda.setCliente(c);
163         venda.setPago(checkBoxVendaPago.isSelected());
164 
165         Date date = java.sql.Date.valueOf(datePickerVendaData.getValue());
166         venda.setDataVenda(date);
167 
168         buttonConfirmarClicked = true;
169         dialogStage.close();
170     }
171 
172     @FXML
173     public void handleButtonCancelar() {
174         getDialogStage().close();
175     }
176 
177 }
```

Código 32: VendaPecasDialogController.java

4.5 Pacote Serviço

4.5.1 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 = "Não";
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 (descrição, valorPadrão, 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("descrição"));
54                  servico.setValorPadrao(resultado.getDouble("valorPadrão"));
55                  servico.setValorFixo(resultado.getBoolean("valorFixo"));
56                  servico.setValorF(resultado.getBoolean("valorFixo"));
57                  servico.setValorP(resultado.getDouble("valorPadrão"));
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␣descrição=?,␣valorPadrão=?,␣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

4.5.2 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://
        javafx.com/fxml/1" fx:controller="opencarshop.servico.controller.ServicoController">
13      <children>
14          <AnchorPane maxHeight="383.0" maxWidth="1000.0" minHeight="200.0" minWidth="350.0" prefHeight="383.0" prefWidth
        ="457.0">
15              <children>
16                  <TextField fx:id="tfdescricao" alignment="TOP_LEFT" depthTest="ENABLE" layoutX="14.0" layoutY="31.0"
        nodeOrientation="LEFT_TO_RIGHT" promptText="Descrição␣do␣serviço">
17                      <tooltip>
18                          <Tooltip text="User␣Name␣will␣need␣to␣Login" />
19                      </tooltip>
20                      <font>
21                          <Font size="11.0" />
22                      </font>
23                      <cursor>
24                          <Cursor fx:constant="DEFAULT" />
25                      </cursor>
26                  </TextField>
27                  <TextField fx:id="tfvalor" layoutX="165.0" layoutY="31.0" promptText="ex:␣50,00">
28                      <font>
29                          <Font size="11.0" />
30                      </font>
31                  </TextField>
32                  <Button fx:id="btnSignUp" layoutX="316.0" layoutY="74.0" mnemonicParsing="false" onAction="#
        cadastraServico" text="Cadastrar" />
33                  <Label layoutX="14.0" layoutY="14.0" text="Desccrição:" />
34                  <Label layoutX="165.0" layoutY="14.0" text="Valor:" />
35                  <CheckBox fx:id="chkValue" layoutX="312.0" layoutY="35.0" mnemonicParsing="false" text="Valor␣Fixo" />
36                  <Label fx:id="labelErroServ" layoutX="106.0" layoutY="176.0" textFill="#ee0303">
37                      <font>
38                          <Font size="11.0" />
```

```
                    </font>
                </Label>
            </children>
        </AnchorPane>
    </children>


</AnchorPane>
```

Código 35: Cadastrar.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<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">
    <children>
        <TableView fx:id="tbl_servico" prefHeight="650.0" prefWidth="446.0">
            <columns>
                <TableColumn fx:id="col_descricao" minWidth="0.0" prefWidth="250.0" text="Descrição" />
                <TableColumn fx:id="col_valor" minWidth="0.0" prefWidth="110.0" text="Valor" />
                <TableColumn fx:id="col_tpvalor" minWidth="0.0" prefWidth="85.0" text="Valor␣fixo" />
            </columns>
        </TableView>
        <TextField fx:id="tfdescricao" layoutX="632.0" layoutY="255.0" prefWidth="200.0" promptText="Descrição" />
        <TextField fx:id="tfvalor" layoutX="632.0" layoutY="301.0" prefWidth="200.0" promptText="Valor" />
        <CheckBox fx:id="chkValue" layoutX="632.0" layoutY="342.0" mnemonicParsing="false" text="Valor␣Fixo" />
        <Button layoutX="688.0" layoutY="399.0" mnemonicParsing="false" onAction="#alterarServico" text="Alterar␣
     Serviço" />
        <TextField fx:id="tfid" editable="false" layoutX="632.0" layoutY="209.0" prefWidth="200.0" promptText="ID␣
     serviço" />
        <Label fx:id="confirmaAtualizacao" alignment="CENTER" layoutX="634.0" layoutY="365.0" prefHeight="17.0"
     prefWidth="200.0" />
        <Label layoutX="634.0" layoutY="186.0" text="ID␣Serviço:" />
        <Label layoutX="634.0" layoutY="234.0" text="Descrição␣do␣serviço:" />
        <Label layoutX="634.0" layoutY="280.0" text="Valor␣do␣serviço:" />
    </children>
</AnchorPane>
```

Código 36: Buscar.fxml

4.5.3 Controller

```
package opencarshop.servico.controller;

import java.net.URL;
import java.sql.SQLException;
import java.util.List;
import java.util.ResourceBundle;
import java.util.regex.Pattern;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.CheckBox;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import opencarshop.servico.model.Servico;
import opencarshop.servico.model.ServicoDAO;

public class ServicoController implements Initializable {

    /**
     * Initializes the controller class.
     */
    //Variavéis
    @FXML
    private TextField tfdescricao;

    @FXML
    private TextField tfvalor;

    @FXML
    private CheckBox chkValue;

    @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 serviços
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          //istancia objeto serviço para configuraçao de atributos
60          Servico servico;
61          //instacia objeto para inserção de objeto cadasrtado 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() > (int) 45) || ("".equals(descri))) {
69              labelErroServ.setText("Descrição␣deve␣ter␣até␣45␣caracteres");
70
71          }
72          if (Pattern.matches("[a-zA-Z]+", valor) == true) {
73              labelErroServ.setText("por␣favor␣insira␣apenas␣numeros␣e␣virgula/ponto");
74          }
75
76          //passando para double
77          valor = valor.replace(",", ".");
78          Double valorDouble = Double.parseDouble(valor);
79
80          //Criando objeto serviço
81          servico = new Servico(descri, valorDouble, chkValue.isSelected());
82
83          if (!servDao.insertServico(servico)) {
84
85              labelErroServ.setText("Serviço␣cadastrado...retornando␣");
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("Alteração␣realizada␣com␣sucesso!!");
118         } else {
119             confirmaAtualizacao.setText("Erro␣ao␣realizar␣a␣alteração!!");
120         }
121
122     }
123
124     public void selecionarItemTablelaServico(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) -> selecionarItemTablelaServico(newValue));
140         } catch (Exception ex) {
141
142         }
143
144     }
145
146 }
```

Código 37: ServicoController.java

4.6 Pacote Veiculo

4.6.1 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
20     public String getModelo() {
21         return modelo;
22     }
23
24     public void setModelo(String modelo) {
25         this.modelo = modelo;
26     }
27
28     public String getVersao() {
29         return versao;
30     }
31
32     public void setVersao(String versao) {
33         this.versao = versao;
34     }
35
36     public int getAno() {
37         return ano;
38     }
39
40     public void setAno(int ano) {
41         this.ano = ano;
42     }
43
44     public int getQuantidade() {
45         return quantidade;
46     }
47
48     public void setQuantidade(int quantidade) {
49         this.quantidade = quantidade;
50     }
51
52     public double getValor() {
53         return valor;
54     }
55
56     public void setValor(double valor) {
57         this.valor = valor;
58     }
59
60     public boolean isOpcionalVidrosEletricos() {
61         return opcionalVidrosEletricos;
62     }
63
64     public void setOpcionalVidrosEletricos(boolean opcionalVidrosEletricos) {
```

```
65          this.opcionalVidrosEletricos = opcionalVidrosEletricos;
66      }
67
68      public boolean isOpcionalTravasEletricas() {
69          return opcionalTravasEletricas;
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,␣
     opcionalAltoFalantes)"
95                  + "VALUES(?,?,?,?,?,?,?,?,?,?)";
96          try {
97              conn = c.conectar();
98              conn.setAutoCommit(false);
99
100             stmt = conn.prepareStatement(query);
101
102             stmt.setString(1, veiculo.getModelo());
103             stmt.setInt(2, veiculo.getAno());
104             stmt.setString(3, veiculo.getVersao());
105             stmt.setInt(4, veiculo.getQuantidade());
106             stmt.setDouble(5, veiculo.getValor());
107             stmt.setBoolean(6, veiculo.isOpcionalVidrosEletricos());
108             stmt.setBoolean(7, veiculo.isOpcionalTravasEletricas());
109             stmt.setBoolean(8, veiculo.isOpcionalAr());
110             stmt.setBoolean(9, veiculo.isOpcionalFarolNeblina());
111             stmt.setBoolean(10, veiculo.isOpcionalAltoFalantes());
112
113             stmt.execute();
114             conn.commit();
115
116             conn.close();
117             return true;
118         } catch (Exception e) {
119             e.printStackTrace();
120             return false;
121         }
122
123     }
124 }
```

Código 39: VeiculoDAO.java

4.6.2 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
```

```
<AnchorPane id="AnchorPane" prefHeight="312.9609375" prefWidth="1024.0" xmlns:fx="http://javafx.com/fxml/1" xmlns="
     http://javafx.com/javafx/2.2" fx:controller="opencarshop.veiculo.controller.VeiculoController">
    <children>
        <AnchorPane layoutX="0.0" layoutY="0.0" minHeight="0.0" minWidth="0.0" prefHeight="263.0" prefWidth="1024.0">
            <children>
                <TextField fx:id="tf_ano" layoutX="14.0" layoutY="65.0" prefHeight="25.0" prefWidth="170.0" promptText=
     "Ano" />
                <TextField fx:id="tf_versao" layoutX="195.0" layoutY="65.0" prefHeight="25.0" prefWidth="170.0"
     promptText="Versão" />
                <TextField fx:id="tf_modelo" layoutX="14.0" layoutY="14.0" prefHeight="25.0" prefWidth="
     350.0000999999975" promptText="Modelo" />
                <TextField fx:id="tf_qntd" layoutX="14.0" layoutY="114.0" prefHeight="25.0" prefWidth="170.0"
     promptText="Quantidade" />
                <TextField fx:id="tf_valor" layoutX="195.0" layoutY="114.0" prefHeight="25.0" prefWidth="170.0"
     promptText="Valor" />
                <CheckBox fx:id="cb_VidrosEletricos" layoutX="28.0" layoutY="161.0" mnemonicParsing="false" text="
     Vidros␣Elétricos" />
                <CheckBox fx:id="cb_AltoFalante" layoutX="277.0" layoutY="161.0" mnemonicParsing="false" text="Alto␣
     Falantes" />
                <CheckBox fx:id="cb_Ar" layoutX="154.0" layoutY="161.0" mnemonicParsing="false" text="Ar␣Condicionado"
     />
                <CheckBox fx:id="cb_FarolNeblina" layoutX="154.0" layoutY="189.0" mnemonicParsing="false" text="Farol␣
     Neblina" />
                <CheckBox fx:id="cb_TravasEletricas" layoutX="28.0" layoutY="189.0" mnemonicParsing="false" text="
     Travas␣Eletricas" />
            </children>
        </AnchorPane>
        <Button fx:id="btn_cadastrar" layoutX="149.0" layoutY="232.0" mnemonicParsing="false" onAction="#
     cadastrarVeiculo" text="Cadastrar" />
        <Label fx:id="lb_result" labelFor="$btn_cadastrar" layoutX="69.0" layoutY="263.0" prefHeight="36.0" prefWidth="
     220.0" text="" textFill="#ff3333">
            <font>
                <Font size="14.0" />
            </font>
        </Label>
    </children>
</AnchorPane>
```

Código 40: Cadastrar.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>


<AnchorPane id="AnchorPane" prefHeight="768.0" prefWidth="1201.0" xmlns:fx="http://javafx.com/fxml/1" xmlns="http://
     javafx.com/javafx/2.2" fx:controller="opencarshop.veiculo.controller.VeiculoController">
    <children>
        <TableView layoutX="-11.0" layoutY="0.0" prefHeight="650.0" prefWidth="602.0">
            <columns>
                <TableColumn id="col_modelo" prefWidth="75.0" text="Modelo" />
                <TableColumn id="col_ano" prefWidth="75.0" text="Ano" />
                <TableColumn id="col_versao" prefWidth="75.0" text="Versão" />
                <TableColumn id="col_qntd" prefWidth="75.0" text="Quantidade" />
                <TableColumn id="col_valor" prefWidth="75.0" text="Valor" />
                <TableColumn id="col_vidrosEletricos" maxWidth="5000.0" minWidth="10.0" prefWidth="100.0" text="Vidros␣
     Elétricos" />
                <TableColumn id="col_travasEletricas" maxWidth="5000.0" minWidth="10.0" prefWidth="100.0" text="Travas␣
     Elétricas" />
                <TableColumn id="col_ar" maxWidth="5000.0" minWidth="10.0" prefWidth="112.0" text="Ar␣Condicionado" />
                <TableColumn id="col_farolNeblina" maxWidth="5000.0" minWidth="10.0" prefWidth="91.0" text="Farol␣
     Neblina" />
                <TableColumn id="col_altoFalantes" maxWidth="5000.0" minWidth="10.0" prefWidth="84.0" text="Alto␣
     Falantes" />
            </columns>
        </TableView>
        <TextField id="tf_modelo" layoutX="624.0" layoutY="29.0" prefHeight="25.0" prefWidth="286.0" promptText="Modelo
     " />
        <Button id="btn_buscar" layoutX="767.0" layoutY="212.0" mnemonicParsing="false" text="Buscar" />
    </children>
</AnchorPane>
```

Código 41: Buscar.fxml

4.6.3 Controller

```
package opencarshop.veiculo.controller;

import java.net.URL;
import java.util.List;
import java.util.Properties;
import java.util.ResourceBundle;
```

```java
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.CheckBox;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import opencarshop.veiculo.model.Veiculo;
import opencarshop.veiculo.model.VeiculoDAO;

public class VeiculoController implements Initializable {

    //Variavéis
    @FXML
    private Label lb_result;

    @FXML
    private TextField tf_modelo;

    @FXML
    private TextField tf_ano;

    @FXML
    private TextField tf_versao;

    @FXML
    private TextField tf_qntd;

    @FXML
    private TextField tf_valor;

    @FXML
    private CheckBox cb_VidrosEletricos;

    @FXML
    private CheckBox cb_TravasEletricas;

    @FXML
    private CheckBox cb_Ar;

    @FXML
    private CheckBox cb_FarolNeblina;

    @FXML
    private CheckBox cb_AltoFalante;

    //TABELA VEICULO
    @FXML
    private TableColumn<Veiculo, String> col_modelo;

    @FXML
    private TableColumn<Veiculo, String> col_ano;

    @FXML
    private TableColumn<Veiculo, String> col_versao;

    @FXML
    private TableColumn<Veiculo, String> col_qntd;

    @FXML
    private TableColumn<Veiculo, String> col_valor;

    @FXML
    private TableColumn<Veiculo, String> col_vidrosEletricos;

    @FXML
    private TableColumn<Veiculo, String> col_travasEletricas;

    @FXML
    private TableColumn<Veiculo, String> col_ar;

    @FXML
    private TableColumn<Veiculo, String> col_farolNeblina;

    @FXML
    private TableColumn<Veiculo, String> col_altoFalantes;

    // tabela
    @FXML
    private TableView<Veiculo> tbl_veiculo;

    @FXML
    private void cadastrarVeiculo(ActionEvent event) {

        //instanciando objeto
        Veiculo veiculo = new Veiculo();
        //instancia objeto para inserção de objeto cadastrado no banco
        VeiculoDAO veiculoDao = new VeiculoDAO();

        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

4.7 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 = "root";
10     //Senha do mysql
11     private final String PASSWORD = "keomas123456";
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://localhost/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
```

```
     * @param senha
     * @return uma conexão com o banco de dados
     * @throws Exception
     */
    public Connection conectar() throws Exception {
        Class.forName("com.mysql.jdbc.Driver"); //Faz com que a classe seja carregada pela JVM
        //Cria a conexão com o banco de dados
        Connection connection = DriverManager.getConnection(DATABASE_URL, USERNAME, PASSWORD);
        return connection;
    }
}
```

Código 43: ConexaoMySQL.java

```
package opencarshop.util;

import java.time.Instant;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.ZoneId;
import java.util.Date;
import javafx.scene.control.DatePicker;

public class Utilidades {

    /**
     * Converte LocalDate para Date
     *
     * @param datePicker
     * @return date
     */
    public Date toDate(LocalDate datePicker) {
        if (datePicker == null) {
            return null;
        }
        LocalDate ld = datePicker;
        Instant instant = ld.atStartOfDay().atZone(ZoneId.systemDefault()).toInstant();
        Date date = Date.from(instant);

        return date;
    }

    /**
     * Converte Date para LocalDate
     *
     * @param d
     * @return LocalDate
     */
    public LocalDate toLocalDate(Date d) {
        Instant instant = Instant.ofEpochMilli(d.getTime());
        LocalDate localDate = LocalDateTime.ofInstant(instant, ZoneId.systemDefault()).toLocalDate();
        return localDate;
    }

}
```

Código 44: Utilidades.java

4.8 Outros

```
package opencarshop;

public class Endereco {

    private String CEP;
    private String estado;
    private String cidade;
    private String bairro;
    private String rua;
    private int numero;
    private String complemento;
    private Character tipo;

    public String getCEP() {
        return CEP;
    }

    public Character getTipo() {
        return tipo;
    }

    public void setTipo(Character tipo) {
        this.tipo = tipo;
    }

```

```
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   import opencarshop.funcionario.controller.FuncionarioController;
10
11  public class OpenCarShop extends Application {
12
13      @Override
14      public void start(Stage stage) throws Exception {
15          Parent root = FXMLLoader.load(getClass().getResource("funcionario/view/Autenticar.fxml"));
16
17          Scene scene = new Scene(root);
18
19          FuncionarioController.setPrevStage(stage);
20
21          stage.setScene(scene);
22          stage.getIcons().add(new Image("recursos/icones/account-circle-white.png"));
23          stage.setTitle("Autenticação");
24          stage.show();
25      }
26
27      /**
28       * @param args the command line arguments
29       */
30      public static void main(String[] args) {
31          launch(args);
32      }
33
34  }
```

Código 46: OpenCarShop.java

```
package opencarshop;

import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.control.Label;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.StackPane;
import javafx.scene.text.Text;

public class TelaPrincipalController implements Initializable {

    @FXML
    private StackPane acContent;

    @FXML
    private Text tx_info;

    @FXML
    private void cadastrarFuncionario(ActionEvent event) {
        System.out.println("Cadastrar Funcionario");
        tx_info.setText("Cadastrar Funcionario");
        FXMLLoader fxmlLoader = new FXMLLoader();
        try {
            fxmlLoader.load(getClass().getResource("/opencarshop/funcionario/view/Cadastrar.fxml").openStream());
        } catch (IOException e) {

        }
        AnchorPane root = fxmlLoader.getRoot();
        acContent.getChildren().clear();
        acContent.getChildren().add(root);
    }

    @FXML
    private void listarFuncionario(ActionEvent event) {
        System.out.println("Buscar Funcionario");
        tx_info.setText("Listar Funcionario");
        FXMLLoader fxmlLoader = new FXMLLoader();
        try {
            fxmlLoader.load(getClass().getResource("/opencarshop/funcionario/view/Buscar.fxml").openStream());
        } catch (IOException e) {

        }
        AnchorPane root = fxmlLoader.getRoot();
        acContent.getChildren().clear();
        acContent.getChildren().add(root);
    }

    @FXML
    private void cadastrarFornecedor(ActionEvent event) {
        System.out.println("Cadastrar Fornecedor");
        tx_info.setText("Cadastrar Fornecedor");
        FXMLLoader fxmlLoader = new FXMLLoader();
        try {
            fxmlLoader.load(getClass().getResource("/opencarshop/fornecedor/view/Cadastrar.fxml").openStream());
        } catch (IOException e) {

        }
        AnchorPane root = fxmlLoader.getRoot();
        acContent.getChildren().clear();
        acContent.getChildren().add(root);
    }

    @FXML
    private void listarFornecedor(ActionEvent event) {
        System.out.println("Buscar Fornecedor");
        tx_info.setText("Listar Fornecedor");
        FXMLLoader fxmlLoader = new FXMLLoader();
        try {
            fxmlLoader.load(getClass().getResource("/opencarshop/fornecedor/view/Buscar.fxml").openStream());
        } catch (IOException e) {

        }
        AnchorPane root = fxmlLoader.getRoot();
        acContent.getChildren().clear();
        acContent.getChildren().add(root);
    }

    @FXML
    private void cadastrarCliente(ActionEvent event) {
        System.out.println("Cadastrar Cliente");
        tx_info.setText("Cadastrar Cliente");
        FXMLLoader fxmlLoader = new FXMLLoader();
        try {
```

```
89              fxmlLoader.load(getClass().getResource("/opencarshop/cliente/view/Cadastrar.fxml").openStream());
90          } catch (IOException e) {
91
92          }
93          AnchorPane root = fxmlLoader.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 Serviço");
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         }
```

```
        AnchorPane root = fxmlLoader.getRoot();
        acContent.getChildren().clear();
        acContent.getChildren().add(root);
    }

    @FXML
    private void listarPeca(ActionEvent event) {
        System.out.println("Buscar Peca");
        tx_info.setText("Listar Peça");
        FXMLLoader fxmlLoader = new FXMLLoader();
        try {
            fxmlLoader.load(getClass().getResource("/opencarshop/peca/view/Buscar.fxml").openStream());
        } catch (IOException e) {

        }
        AnchorPane root = fxmlLoader.getRoot();
        acContent.getChildren().clear();
        acContent.getChildren().add(root);
    }

    @FXML
    private void venderPeca(ActionEvent event) {
        System.out.println("Vender Peca");
        tx_info.setText("Vender Peça");
        FXMLLoader fxmlLoader = new FXMLLoader();
        try {
            fxmlLoader.load(getClass().getResource("/opencarshop/peca/view/VendaPecas.fxml").openStream());
        } catch (IOException e) {

        }
        AnchorPane root = fxmlLoader.getRoot();
        acContent.getChildren().clear();
        acContent.getChildren().add(root);

    }

    @FXML
    private void relatorioVendaPecaMes(ActionEvent event) {
        System.out.println("Relatorio Peca/Mes");
        tx_info.setText("Relatorio Peca/Mes");

        FXMLLoader fxmlLoader = new FXMLLoader();
        try {
            fxmlLoader.load(getClass().getResource("/opencarshop/peca/view/GraficosVendasPorMes.fxml").openStream());
        } catch (IOException e) {

        }
        AnchorPane root = fxmlLoader.getRoot();
        acContent.getChildren().clear();
        acContent.getChildren().add(root);
    }

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

}
```

Código 47: TelaPrincipalController.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.effect.*?>
<?import javafx.scene.text.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<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.TelaPrincipalController">
    <children>
        <StackPane prefHeight="740.0" prefWidth="1024.0">
            <children>
                <BorderPane prefHeight="200.0" prefWidth="200.0">
                    <top>
                        <MenuBar prefWidth="25.0" BorderPane.alignment="CENTER">
                            <menus>
                                <Menu mnemonicParsing="false" text="Cliente">
                                    <items>
                                        <MenuItem mnemonicParsing="false" onAction="#cadastrarCliente" text="Cadastrar"
     />
                                        <MenuItem mnemonicParsing="false" onAction="#listarCliente" text="Listar" />
                                    </items>
                                </Menu>
                                <Menu mnemonicParsing="false" text="Serviço">
                                    <items>
                                        <MenuItem mnemonicParsing="false" onAction="#cadastrarServico" text="Cadastrar"
     />
                                        <MenuItem mnemonicParsing="false" onAction="#listarServico" text="Listar" />
```

```
                                    </items>
                                </Menu>
                                <Menu mnemonicParsing="false" text="Peça">
                                    <items>
                                        <MenuItem mnemonicParsing="false" onAction="#venderPeca" text="Vender" />
                                        <SeparatorMenuItem mnemonicParsing="false" />
                                        <MenuItem mnemonicParsing="false" onAction="#cadastrarPeca" text="Gerenciar" />
                                        <SeparatorMenuItem mnemonicParsing="false" />
                                        <MenuItem mnemonicParsing="false" onAction="#relatorioVendaPecaMes" text="
    Relatório␣Venda/Mes" />
                                    </items>
                                </Menu>
                                <Menu mnemonicParsing="false" text="Veiculo">
                                    <items>
                                        <MenuItem mnemonicParsing="false" text="Vender" />
                                        <SeparatorMenuItem mnemonicParsing="false" />
                                        <MenuItem mnemonicParsing="false" onAction="#cadastrarVeiculo" text="Cadastrar"
     />
                                        <MenuItem mnemonicParsing="false" onAction="#listarVeiculo" text="Listar" />
                                    </items>
                                </Menu>
                                <Menu mnemonicParsing="false" text="Funcionário">
                                    <items>
                                        <MenuItem mnemonicParsing="false" onAction="#cadastrarFuncionario" text="
    Cadastrar" />
                                        <MenuItem mnemonicParsing="false" onAction="#listarFuncionario" text="Listar" /
    >
                                    </items>
                                </Menu>
                                <Menu mnemonicParsing="false" text="Fornecedor">
                                    <items>
                                        <MenuItem mnemonicParsing="false" onAction="#cadastrarFornecedor" text="
    Cadastrar" />
                                        <MenuItem mnemonicParsing="false" onAction="#listarFornecedor" text="Listar" />
                                    </items>
                                </Menu>
                            </menus>
                        </MenuBar>
                    </top>
                    <center>
                        <StackPane fx:id="acContent" prefHeight="646.0" prefWidth="1024.0" BorderPane.alignment="CENTER
    " />
                    </center>
                    <bottom>
                        <ToolBar prefHeight="14.0" prefWidth="1024.0" BorderPane.alignment="CENTER">
                            <items>
                                <Text id="tx_info" fx:id="tx_info" strokeType="OUTSIDE" strokeWidth="0.0" text="
    OpenCarShop">
                                    <font>
                                        <Font name="System␣Bold" size="18.0" />
                                    </font>
                                </Text>
                            </items>
                        </ToolBar>
                    </bottom>
                </BorderPane>
            </children>
        </StackPane>
    </children>
</AnchorPane>
```

Código 48: TelaPrincipal.fxml

4.9 SQL

```
-- MySQL Script generated by MySQL Workbench
-- 10/31/16 22:22:22
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-- -----------------------------------------------------
-- Schema opencarshop
-- -----------------------------------------------------
DROP SCHEMA IF EXISTS `opencarshop` ;

-- -----------------------------------------------------
-- Schema opencarshop
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `opencarshop` DEFAULT CHARACTER SET utf8 ;
USE `opencarshop` ;

-- -----------------------------------------------------
-- Table `opencarshop`.`Endereco`
-- -----------------------------------------------------
```

```
24  DROP TABLE IF EXISTS `opencarshop`.`Endereco` ;
25
26  CREATE TABLE IF NOT EXISTS `opencarshop`.`Endereco` (
27    `id` INT NOT NULL AUTO_INCREMENT,
28    `cep` VARCHAR(9) NULL DEFAULT '_____-___',
29    `estado` CHAR(2) NOT NULL,
30    `cidade` VARCHAR(45) NOT NULL,
31    `bairro` VARCHAR(45) NOT NULL,
32    `rua` VARCHAR(45) NULL,
33    `numero` INT NULL,
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    `cargo` 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` DATE 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,
```

```
  `razaoSocial` VARCHAR(50) NOT NULL,
  `descricao` VARCHAR(45) NULL,
  `email` VARCHAR(50) NULL,
  `telefone1` VARCHAR(45) GENERATED ALWAYS AS (),
  `telefone2` VARCHAR(45) NULL,
  `endereco` INT NOT NULL,
  `ativo` TINYINT(1) NULL DEFAULT 1,
  PRIMARY KEY (`cnpj`),
  INDEX `fk_Fornecedor_Endereco_idx` (`endereco` ASC),
  CONSTRAINT `fk_Fornecedor_Endereco`
    FOREIGN KEY (`endereco`)
    REFERENCES `opencarshop`.`Endereco` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`Veiculo`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`Veiculo` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`Veiculo` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `modelo` VARCHAR(45) NOT NULL,
  `ano` YEAR NOT NULL,
  `versao` VARCHAR(45) NOT NULL,
  `opcionalVidrosEletricos` TINYINT(1) NULL DEFAULT 0,
  `opcionalTravasEletricas` TINYINT(1) NULL DEFAULT 0,
  `opcionalAr` TINYINT(1) NULL DEFAULT 0,
  `opcionalFarolNeblina` TINYINT(1) NULL DEFAULT 0,
  `opcionalAltoFalantes` TINYINT(1) NULL DEFAULT 0,
  `quantidade` INT NULL,
  `valor` DECIMAL(10,2) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`Peca`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`Peca` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`Peca` (
  `id` INT NOT NULL,
  `descricao` VARCHAR(45) NULL,
  `valor` VARCHAR(45) NULL,
  `tipo` CHAR(1) NULL,
  `quantidade` INT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`Servico`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`Servico` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`Servico` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `descricao` VARCHAR(45) NOT NULL,
  `valorPadrao` DECIMAL(10,2) NOT NULL,
  `valorFixo` TINYINT(1) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`OrcamentoServico`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`OrcamentoServico` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`OrcamentoServico` (
  `id` INT NOT NULL,
  `placa` VARCHAR(45) NOT NULL,
  `data` DATE NULL,
  `funcionario` VARCHAR(15) NOT NULL,
  PRIMARY KEY (`id`, `funcionario`),
  INDEX `fk_OrcamentoServico_Funcionario1_idx` (`funcionario` ASC),
  CONSTRAINT `fk_OrcamentoServico_Funcionario1`
    FOREIGN KEY (`funcionario`)
    REFERENCES `opencarshop`.`Funcionario` (`cpf`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`ServicoSelecionado`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`ServicoSelecionado` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`ServicoSelecionado` (
  `orcamento` INT NOT NULL,
```

```
  `servico` INT NOT NULL,
  PRIMARY KEY (`orcamento`, `servico`),
  INDEX `fk_OrcamentoServico_has_Servico_Servico1_idx` (`servico` ASC),
  INDEX `fk_OrcamentoServico_has_Servico_OrcamentoServico1_idx` (`orcamento` ASC),
  CONSTRAINT `fk_OrcamentoServico_has_Servico_OrcamentoServico1`
    FOREIGN KEY (`orcamento`)
    REFERENCES `opencarshop`.`OrcamentoServico` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_OrcamentoServico_has_Servico_Servico1`
    FOREIGN KEY (`servico`)
    REFERENCES `opencarshop`.`Servico` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`VeiculoFornecido`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`VeiculoFornecido` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`VeiculoFornecido` (
  `Fornecedor_cnpj` VARCHAR(15) NOT NULL,
  `Veiculo_id` INT NOT NULL,
  PRIMARY KEY (`Fornecedor_cnpj`, `Veiculo_id`),
  INDEX `fk_Fornecedor_has_Veiculo_Veiculo1_idx` (`Veiculo_id` ASC),
  INDEX `fk_Fornecedor_has_Veiculo_Fornecedor1_idx` (`Fornecedor_cnpj` ASC),
  CONSTRAINT `fk_Fornecedor_has_Veiculo_Fornecedor1`
    FOREIGN KEY (`Fornecedor_cnpj`)
    REFERENCES `opencarshop`.`Fornecedor` (`cnpj`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Fornecedor_has_Veiculo_Veiculo1`
    FOREIGN KEY (`Veiculo_id`)
    REFERENCES `opencarshop`.`Veiculo` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`PecaForncida`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`PecaForncida` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`PecaForncida` (
  `fornecedor` VARCHAR(15) NOT NULL,
  `peca` INT NOT NULL,
  PRIMARY KEY (`fornecedor`, `peca`),
  INDEX `fk_Fornecedor_has_Peca_Peca1_idx` (`peca` ASC),
  INDEX `fk_Fornecedor_has_Peca_Fornecedor1_idx` (`fornecedor` ASC),
  CONSTRAINT `fk_Fornecedor_has_Peca_Fornecedor1`
    FOREIGN KEY (`fornecedor`)
    REFERENCES `opencarshop`.`Fornecedor` (`cnpj`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Fornecedor_has_Peca_Peca1`
    FOREIGN KEY (`peca`)
    REFERENCES `opencarshop`.`Peca` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`PecaNecessarias`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`PecaNecessarias` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`PecaNecessarias` (
  `peca` INT NOT NULL,
  `orcamentoServico` INT NOT NULL,
  PRIMARY KEY (`peca`, `orcamentoServico`),
  INDEX `fk_Peca_has_OrcamentoServico_OrcamentoServico1_idx` (`orcamentoServico` ASC),
  INDEX `fk_Peca_has_OrcamentoServico_Peca1_idx` (`peca` ASC),
  CONSTRAINT `fk_Peca_has_OrcamentoServico_Peca1`
    FOREIGN KEY (`peca`)
    REFERENCES `opencarshop`.`Peca` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Peca_has_OrcamentoServico_OrcamentoServico1`
    FOREIGN KEY (`orcamentoServico`)
    REFERENCES `opencarshop`.`OrcamentoServico` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`OrdemServico`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`OrdemServico` ;

```

```
CREATE TABLE IF NOT EXISTS `opencarshop`.`OrdemServico` (
  `id` INT NOT NULL,
  `data` VARCHAR(45) NULL,
  `valorFinal` DECIMAL(10,2) NULL,
  `desconto` DECIMAL(6,2) NULL,
  `orcamento` INT NOT NULL,
  `cliente` VARCHAR(15) NOT NULL,
  `funcionario` VARCHAR(15) NOT NULL,
  PRIMARY KEY (`id`, `orcamento`, `cliente`, `funcionario`),
  INDEX `fk_OrdemServico_OrcamentoServico1_idx` (`orcamento` ASC),
  INDEX `fk_OrdemServico_Cliente1_idx` (`cliente` ASC),
  INDEX `fk_OrdemServico_Funcionario1_idx` (`funcionario` ASC),
  CONSTRAINT `fk_OrdemServico_OrcamentoServico1`
    FOREIGN KEY (`orcamento`)
    REFERENCES `opencarshop`.`OrcamentoServico` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_OrdemServico_Cliente1`
    FOREIGN KEY (`cliente`)
    REFERENCES `opencarshop`.`Cliente` (`cpf`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_OrdemServico_Funcionario1`
    FOREIGN KEY (`funcionario`)
    REFERENCES `opencarshop`.`Funcionario` (`cpf`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`Venda`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`Venda` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`Venda` (
  `codigo` INT NOT NULL AUTO_INCREMENT,
  `funcionario` VARCHAR(15) NOT NULL,
  `cliente` VARCHAR(15) NOT NULL,
  PRIMARY KEY (`codigo`),
  INDEX `fk_Funcionario_has_Cliente_Cliente1_idx` (`cliente` ASC),
  INDEX `fk_Funcionario_has_Cliente_Funcionario1_idx` (`funcionario` ASC),
  CONSTRAINT `fk_Funcionario_has_Cliente_Funcionario1`
    FOREIGN KEY (`funcionario`)
    REFERENCES `opencarshop`.`Funcionario` (`cpf`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Funcionario_has_Cliente_Cliente1`
    FOREIGN KEY (`cliente`)
    REFERENCES `opencarshop`.`Cliente` (`cpf`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`ItemVeiculo`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`ItemVeiculo` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`ItemVeiculo` (
  `chasi` VARCHAR(45) NOT NULL,
  `valorFinal` DECIMAL(10,2) NULL,
  `desconto` DECIMAL(6,2) NULL,
  `veiculo` INT NOT NULL,
  `venda` INT NOT NULL,
  PRIMARY KEY (`chasi`, `veiculo`, `venda`),
  INDEX `fk_Veiculo_has_Funcionario_Veiculo1_idx` (`veiculo` ASC),
  INDEX `fk_ItemVeiculo_Venda1_idx` (`venda` ASC),
  CONSTRAINT `fk_Veiculo_has_Funcionario_Veiculo1`
    FOREIGN KEY (`veiculo`)
    REFERENCES `opencarshop`.`Veiculo` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_ItemVeiculo_Venda1`
    FOREIGN KEY (`venda`)
    REFERENCES `opencarshop`.`Venda` (`codigo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `opencarshop`.`ItemPeca`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `opencarshop`.`ItemPeca` ;

CREATE TABLE IF NOT EXISTS `opencarshop`.`ItemPeca` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `valorFinal` DECIMAL(10,2) NULL,
  `desconto` DECIMAL(6,2) NULL,
  `peca` INT NOT NULL,
  `venda` INT NOT NULL,
  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

5 Diagramas

5.1 Diagrama de Casos de Uso

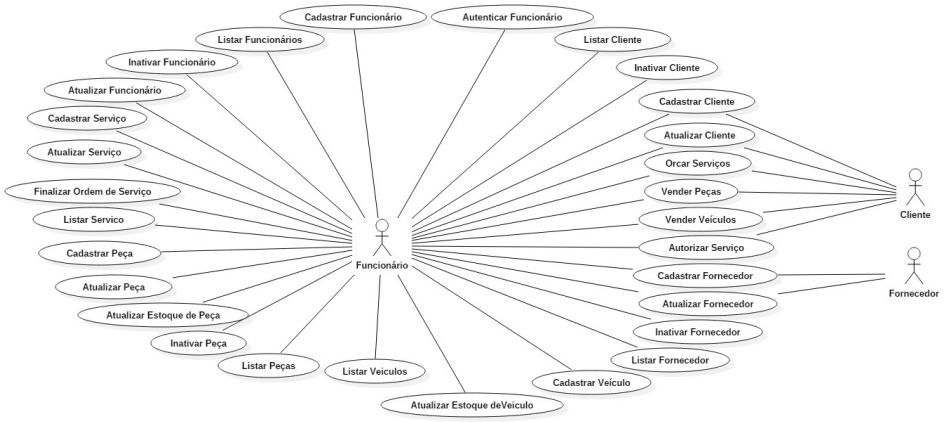

Figura 1: Diagrama de Casos de Uso

5.2 Diagrama de Classes - Analise

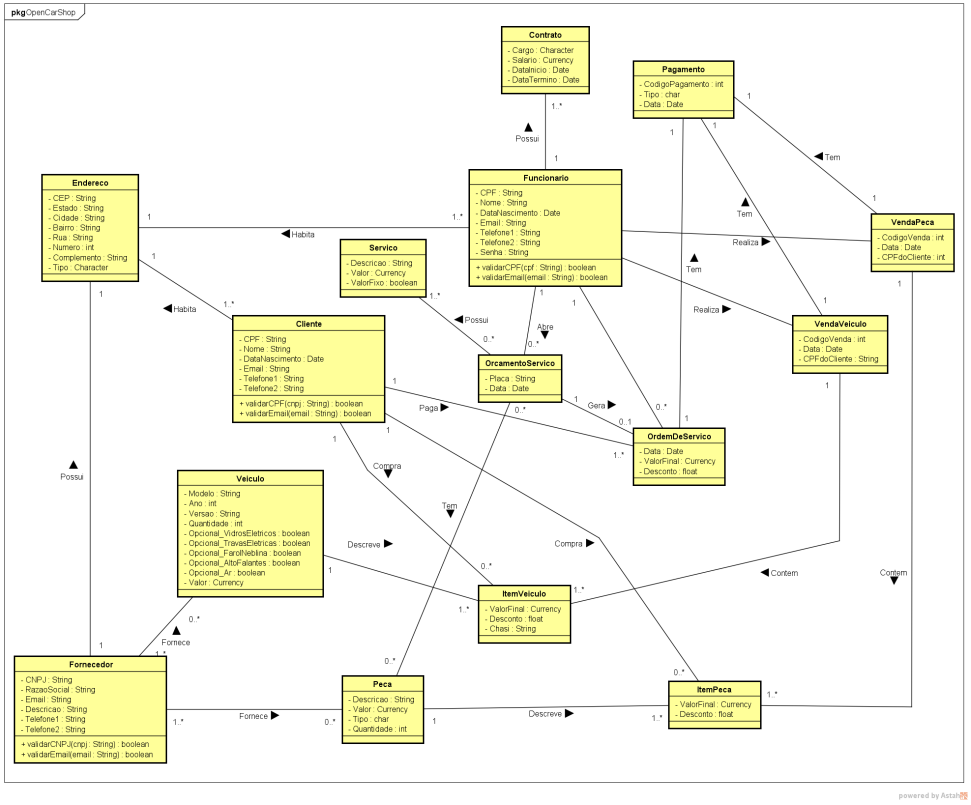

Figura 2: Diagrama de Classes - Analise

5.3 Diagrama de Classes - Projeto

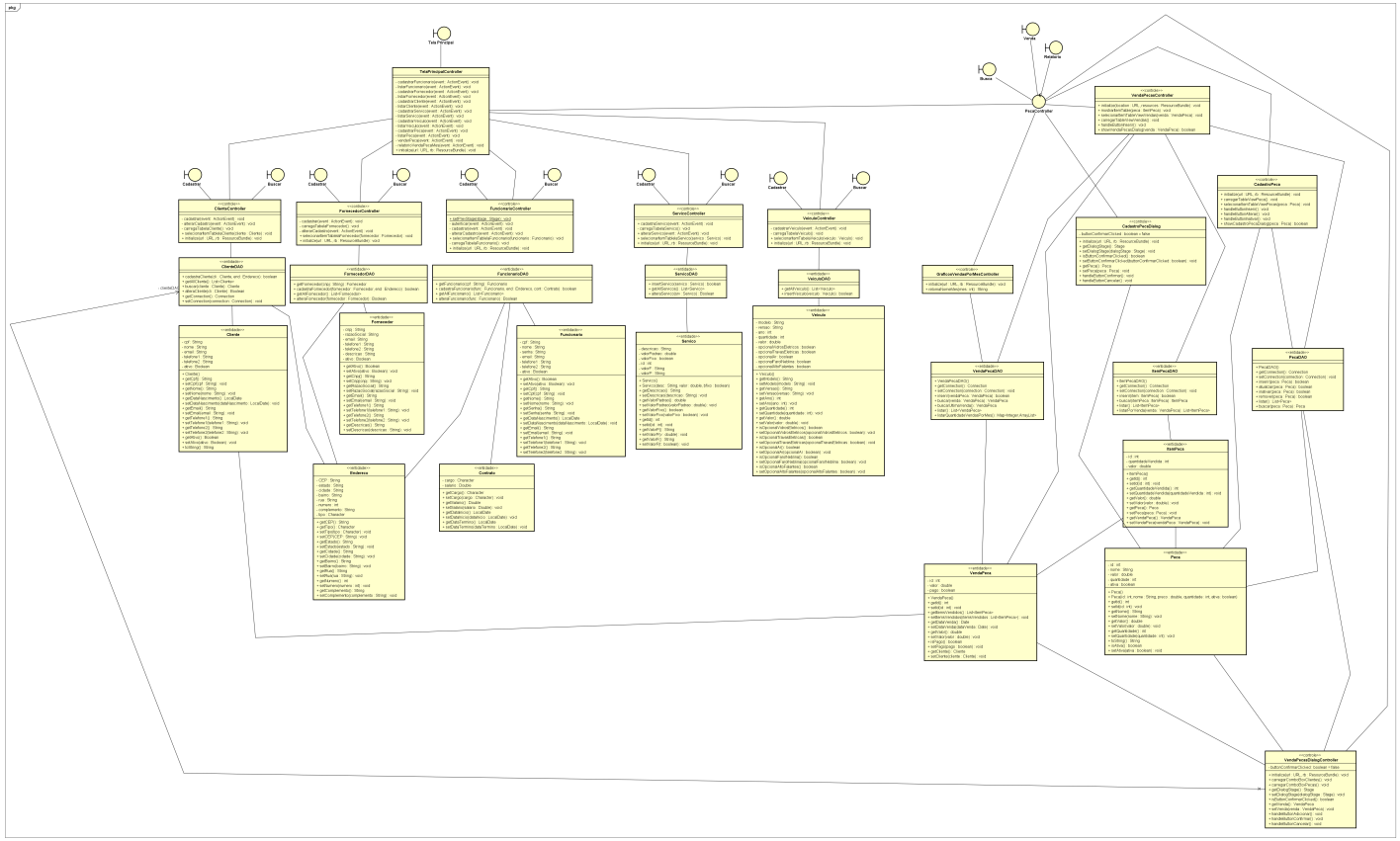

Figura 3: Diagrama de Classes - Projeto

5.4 Diagramas de Atividade

5.4.1 CDU01

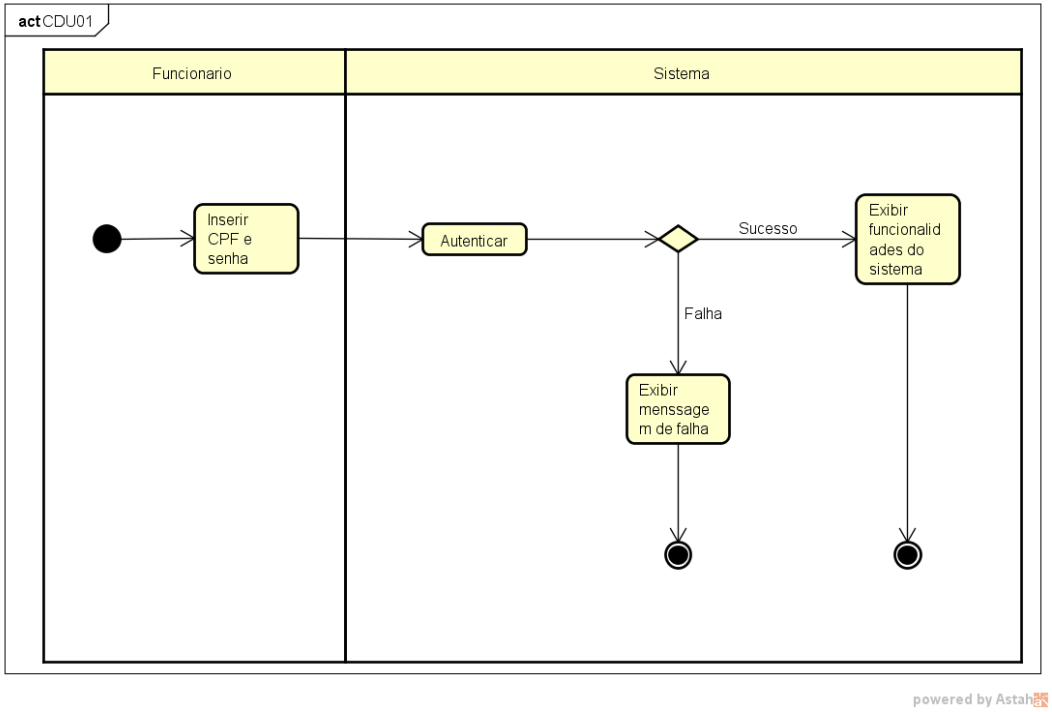

Figura 4: CDU01

5.4.2 CDU02

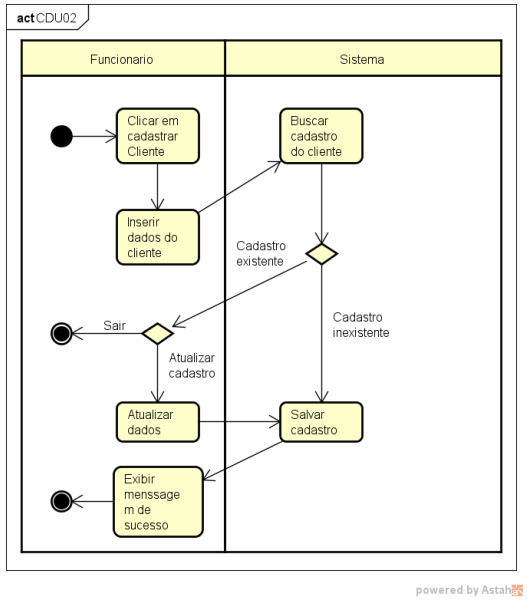

Figura 5: CDU02

5.4.3 CDU03

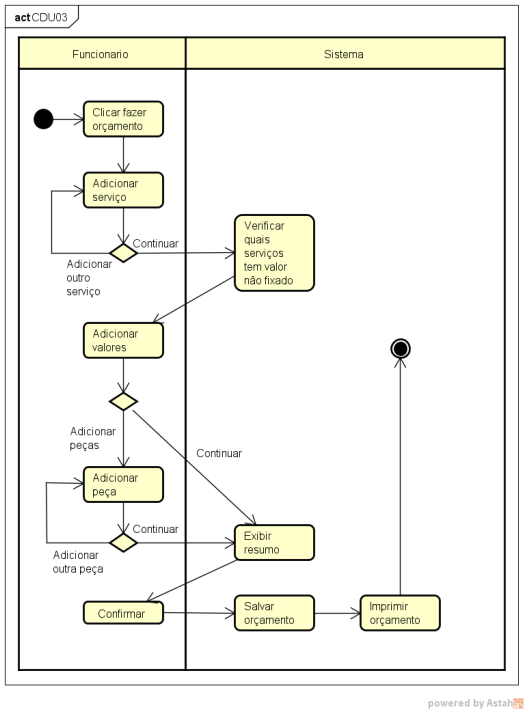

Figura 6: CDU03

5.4.4 CDU04

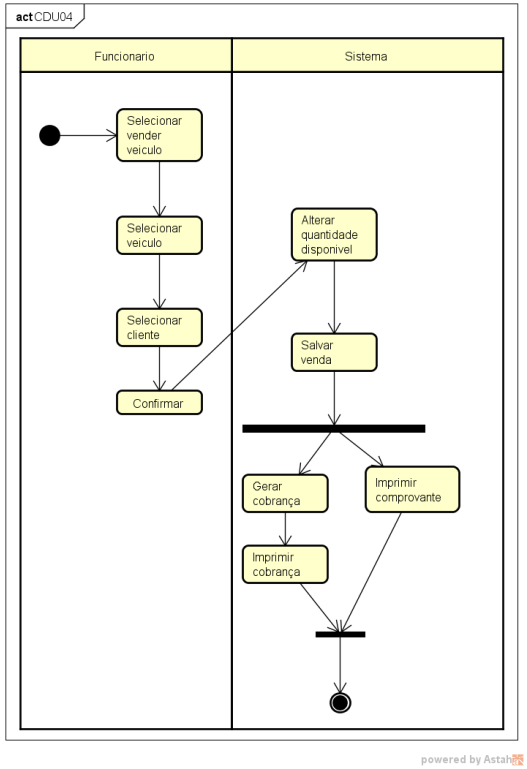

Figura 7: CDU04

5.4.5 CDU05

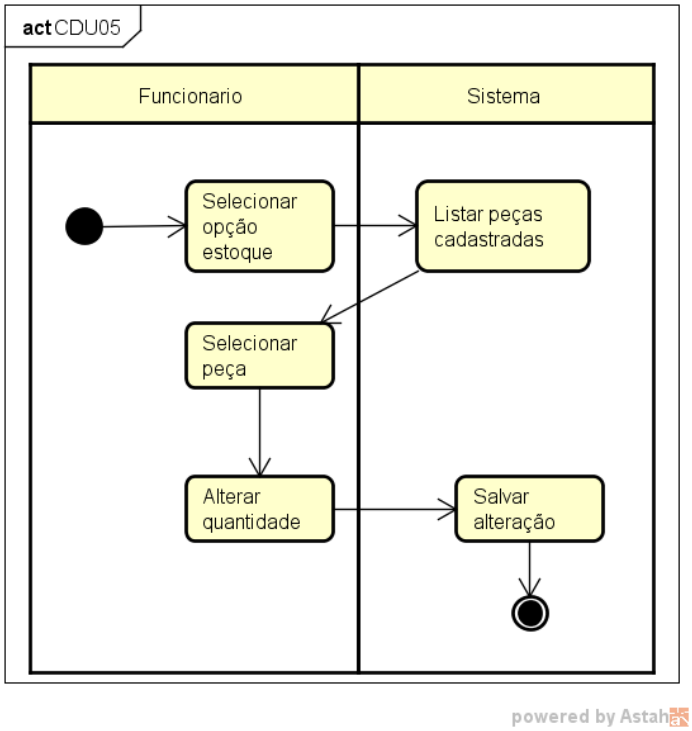

Figura 8: CDU05

5.5 Diagramas de Sequencia

5.5.1 CDU01

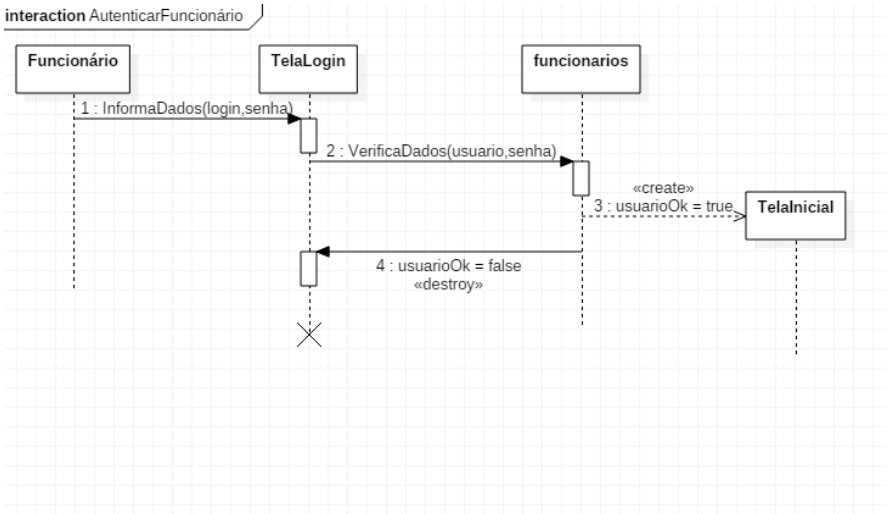

Figura 9: CDU01

5.5.2 CDU02

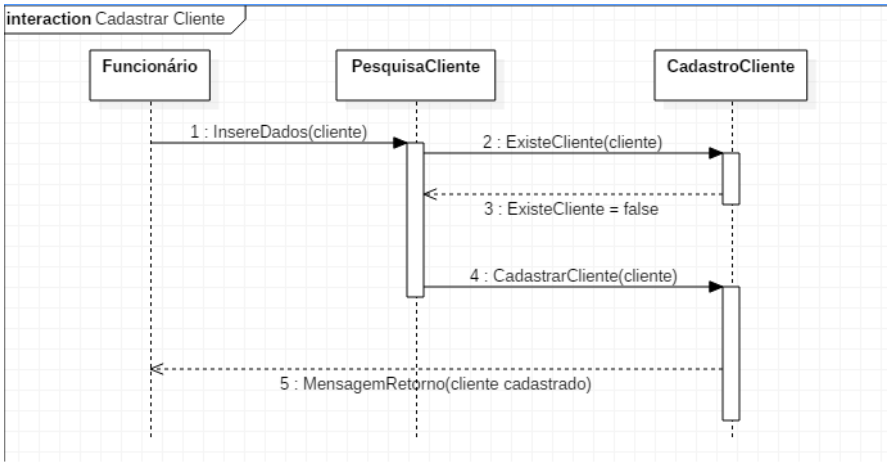

Figura 10: CDU02

5.5.3 CDU03

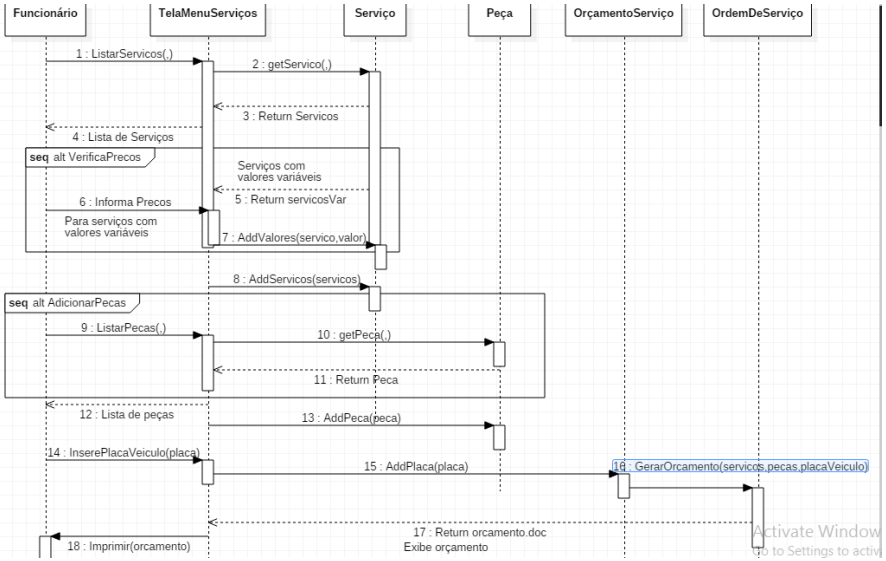

Figura 11: CDU03

5.5.4 CDU04

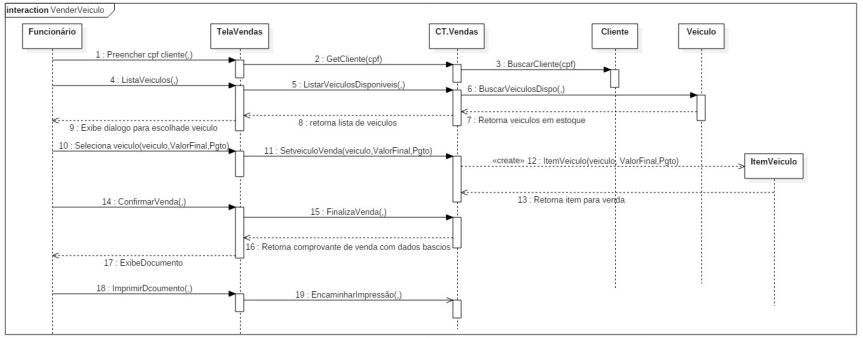

Figura 12: CDU04

∞

5.5.5 CDU05

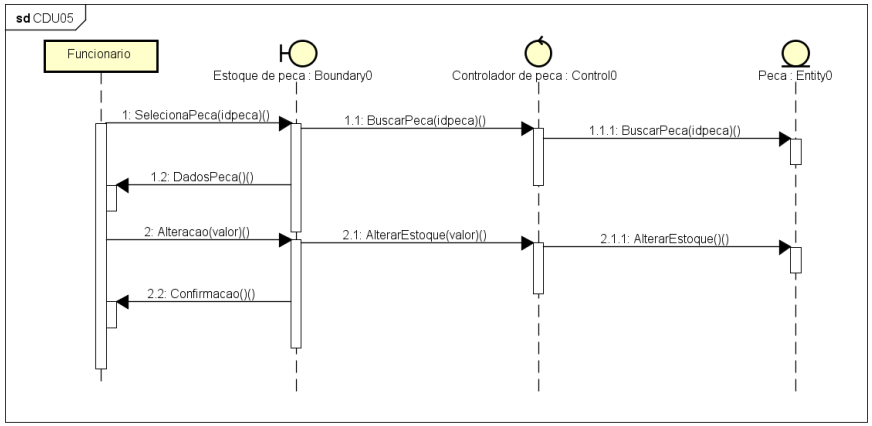

Figura 13: CDU05

5.6 Diagrama Entidade Relacionamento

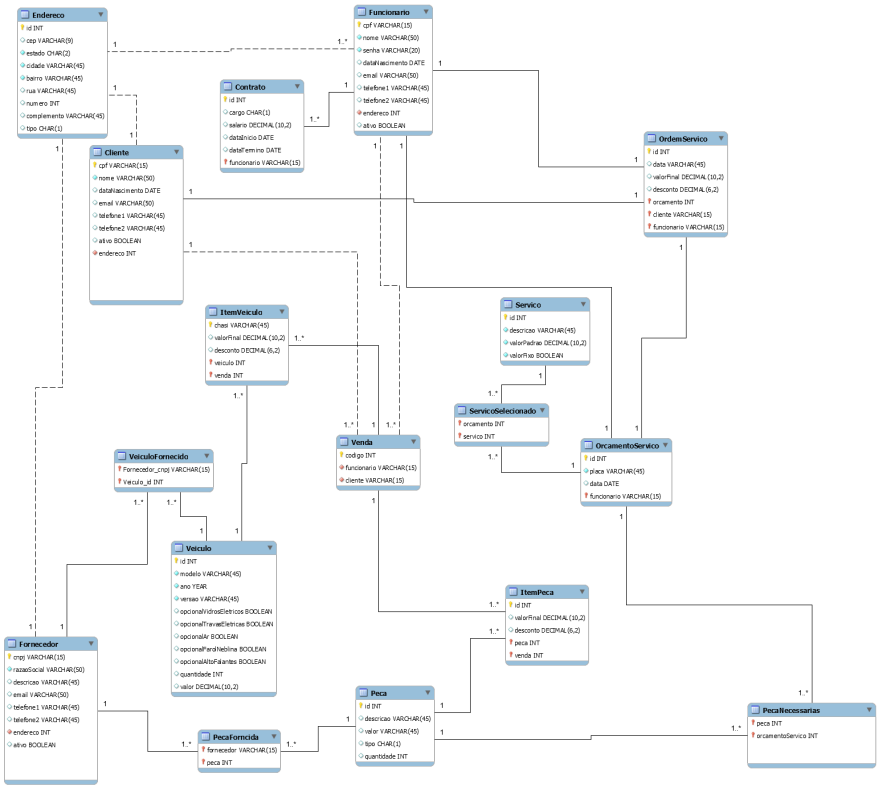

Figura 14: Diagrama de Entidade Relacionamento