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

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

1.6.3 Características do Usuário

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

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

1.6.4 Restrições Gerais

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

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

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

1.6.5 Suposições e Dependências

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

1.7 Requisitos específicos

1.7.1 Requisitos Funcionais

...

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

RF1 Autenticar Funcionário (Pr.: 1):

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

RF2 Cadastrar Cliente (Pr.: 1):

O sistema deve permitir ao funcionário cadastrar clientes.

RF3 Inativar Cliente (Pr.: 1):

O sistema deve permitir ao funcionário inativar cadastro de clientes.

RF4 Atualizar Cliente (Pr.: 1):

O sistema deve permitir ao funcionário atualizar cadastro de clientes.

RF5 Listar Cliente (Pr.: 1):

O sistema deve listar os clientes para o funcionário.

RF6 Cadastrar Funcionário (Pr.: 1):

Descrição: O sistema deve permitir ao gerente cadastrar funcionários.

RF7 Atualizar Funcionário (Pr.: 1):

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

RF8 Inativar Funcionário (Pr.: 1):

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

RF9 Listar Funcionário. (Pr.: 1):

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

RF10 Cadastrar Serviço (Pr.: 1):

O sistema deve permitir ao funcionário cadastrar serviços.

RF11 Atualizar Serviço (Pr.: 1):

O sistema deve permitir ao funcionário atualizar serviços.

RF12 Listar Serviço (Pr.: 1):

O sistema deve listar os Serviços para o funcionário.

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

O sistema deve permitir ao funcionário finalizar ordens de serviços.

RF14 Cadastrar Fornecedor (Pr.: 1):

O sistema deve permitir ao gerente cadastrar fornecedores.

RF15 Atualizar Fornecedor (Pr.: 1):

O sistema deve permitir ao gerente atualizar fornecedores.

RF16 Inativar Fornecedor (Pr.: 1):

O sistema deve permitir ao gerente inativar fornecedores.

RF17 Listar Fornecedor (Pr.: 1):

O sistema deve listar os Fornecedores para o gerente.

RF18 Cadastrar Veículo (Pr.: 1):

Descrição: O sistema deve permitir ao funcionário cadastrar veículos.

RF19 Listar Veículos (Pr.: 1):

O sistema deve permitir ao funcionário listar os veículos.

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

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

RF21 Cadastrar Peça (Pr.: 1):

O sistema deve permitir ao funcionário cadastrar peças.

RF22 Atualizar Peça (Pr.: 1):

O sistema deve permitir ao funcionário atualizar dados da peças.

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

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

RF24 Inativar Peça (Pr.: 1):

O sistema deve permitir ao funcionário a inativação de peças.

RF25 Listar Peças (Pr.: 1):

O sistema deve permitir ao funcionário listar as peças..

RF26 Orçar serviços (Pr.: 1):

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

RF27 Vender Peça (Pr.: 1):

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

RF28 Vender Veículo (Pr.: 1):

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

RF29 Autorizar Serviço (Pr.: 1):

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

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

O sistema deve armazenar os pagamentos das vendas de veículos.

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

O sistema deve armazenar os pagamentos das vendas de peças.

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

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

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

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

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

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

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

O sistema deve efetuar a exclusão de fornecedores.

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

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

RF37 Verificar disponibilidade (Pr.: 1):

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

RF38 Atualizar Estoque (Pr.: 1):

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

RF39 Relatório de Faturamento (Pr.: 1):

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

RF40 Relatório de Cliente (Pr.: 1):

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

1.7.2 Requisitos Não Funcionais

RNF1 Integridade (Pr.: 1):

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

RNF2 Tempo de Resposta (Pr.: 1):

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

RNF3 Usuários Simultâneos (Pr.: 1):

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

RNF4 Interface gráfica (Pr.: 1):

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

RNF5 Portabilidade (Pr.: 1):

O sistema deverá ser independente de plataforma de sistema operacional.

2 Plano de Projeto

Descrever Plano de Projeto

2.1 Motivação

Motivação para o projeto:

- Motivação 1
- Motivação 2
- Motivação 3

3 Casos de Uso

Nome: Autenticar Funcionário.
Descrição: Autenticação dos funcionários para uso do sistema.
Identificador: CDU01.
Ator Primário: Funcionário.
...
Fluxo principal

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

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

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

Nome: Cadastrar Cliente.
Descrição: Funcionário cadastra dados do cliente no sistema.
Identificador: CDU02.
Ator Primário: Funcionário
Precondição: Funcionário deve estar autenticado no sistema.
...
Fluxo principal

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

Fluxo Alternativo(Cliente já cadastrado)

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

Nome: Orçar Serviços.
Descrição: Gerar orçamento de um serviço.
Identificador: CDU03
Ator Primário: Funcionário
Precondição: Funcionário deve estar autenticado no sistema.
...
Fluxo principal

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

Nome: Vender Veículo
Descrição: Realizar venda de veículo
Identificador: CDU04
Ator Primário: Funcionário
Precondição: Funcionário deve estar autenticado no sistema.
...
Fluxo principal

Funcionário	Sistema
1 - Selecionar opção de veículos.	
	2 - Exibir tela menu de veículos.
3 - Selecionar opção de “Vender Veículo”	
	4 - Exibir veículos para venda.
5 - Selecionar veículo para venda.	
6 - Selecionar opção “Continuar.	
	7 - Exibir tela para seleção de cliente
8 - 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 Arquitetura

5 Códigos

5.1 Pacote Cliente

5.1.1 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 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;
    }
}
```

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;
```

```java
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();
        ConexaoMySQL c = new ConexaoMySQL();
        Connection conn = null;
        conn = c.conectar();
        try {
            PreparedStatement stmt = conn.prepareStatement(query);
            ResultSet resultado = stmt.executeQuery();
            while (resultado.next()) {
                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);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        conn.close();
        return retorno;
    }

    public Cliente buscar(Cliente cliente) {
        String sql = "SELECT * FROM Cliente WHERE cpf=?";
        Cliente retorno = new Cliente();
        Utilidades u = new Utilidades();
```

```
        try {
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, cliente.getCpf());
            ResultSet resultado = stmt.executeQuery();
            if (resultado.next()) {
                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 = cliente;
            }
        } catch (SQLException ex) {
            Logger.getLogger(ClienteDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
        return retorno;
    }

    public Boolean alteraCliente(Cliente cli) throws SQLException {
        String query = "UPDATE Cliente SET nome=?, dataNascimento=?, email=?, telefone1=?, telefone2=?, ativo=? WHERE cpf=?";

        try {
            conn = c.conectar();
            PreparedStatement stmt = conn.prepareStatement(query);

            stmt.setString(1, cli.getNome());
            stmt.setDate(2, Date.valueOf(cli.getDataNascimento()));
            stmt.setString(3, cli.getEmail());
            stmt.setString(4, cli.getTelefone1());
            stmt.setString(5, cli.getTelefone2());
            stmt.setBoolean(6, cli.getAtivo());
            stmt.setString(7, cli.getCpf());
            stmt.execute();
            conn.close();
            return true;
        } catch (Exception ex) {

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

    public Connection getConnection() {
        return conn;
    }

    public void setConnection(Connection connection) {
        this.conn = connection;
    }
}
```

Código 2: ClienteDAO.java

5.1.2 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.cliente.controller.ClienteController">
    <children>
        <TabPane prefHeight="494.0" prefWidth="1024.0" tabClosingPolicy="UNAVAILABLE">
            <tabs>
                <Tab text="Identificacao">
                    <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="N mero" />
                            <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 Endere o">
                                <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="Endere o:" />
                            <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="N mero:" />
                            <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" />
                <TableColumn id="col_email" fx:id="col_email" prefWidth="193.0" text="Email" />
            </columns>
        </TableView>
        <TextField fx:id="tf_nomeCadastro" layoutX="846.0" layoutY="95.0" prefWidth="150.0" promptText="Nome" />
        <TextField fx:id="tf_emailCadastro" layoutX="846.0" layoutY="151.0" promptText="Email" />
        <TextField fx:id="tf_telefone1Cadastro" layoutX="675.0" layoutY="208.0" promptText="Telefone 1" />
        <TextField fx:id="tf_telefone2Cadastro" layoutX="846.0" layoutY="208.0" promptText="Telefone 2" />
        <DatePicker fx:id="dp_dataNascimentoCadastro" layoutX="675.0" layoutY="151.0" prefWidth="150.0" promptText="Data de nascimento" />
        <Label layoutX="844.0" layoutY="78.0" text="Nome:" />
        <Label layoutX="846.0" layoutY="134.0" text="Email:" />
```

```
        <Label layoutX="678.0" layoutY="191.0" text="Telefone 1:" />
        <Label layoutX="847.0" layoutY="191.0" text="Telefone 2:" />
        <Label layoutX="678.0" layoutY="134.0" text="Data de Nascimento:" />
        <CheckBox fx:id="cb_ativo" layoutX="674.0" layoutY="263.0" mnemonicParsing="false" text="Ativo" />
        <Button fx:id="btn_alterarCadastro" layoutX="782.0" layoutY="359.0" mnemonicParsing="false" onAction="#
    alterarCadastro" text="Salvar Altera  es" />
        <TextField fx:id="tf_cpfCadastro" editable="false" layoutX="675.0" layoutY="95.0" />
        <Label layoutX="675.0" layoutY="78.0" text="CPF:" />
        <Label fx:id="confirmaAlteracao" alignment="CENTER" contentDisplay="CENTER" layoutX="736.0" layoutY="325.0"
     prefHeight="17.0" prefWidth="199.0" textAlignment="CENTER" />
    </children>
</AnchorPane>
```

Código 4: Buscar.fxml

5.1.3 Controller

```
package opencarshop.cliente.controller;

import java.net.URL;
import java.text.DecimalFormat;
import java.text.ParseException;
import java.util.List;
import java.util.ResourceBundle;
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.ComboBox;
import javafx.scene.control.DatePicker;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import opencarshop.Endereco;
import opencarshop.cliente.model.Cliente;
import opencarshop.cliente.model.ClienteDAO;
import opencarshop.util.Utilidades;

public class ClienteController implements Initializable {

    // TELA DE CADASTRO
    @FXML
    private TextField tf_cpfCadastro;
    @FXML
    private TextField tf_nomeCadastro;
    @FXML
    private DatePicker dp_dataNascimentoCadastro;
    @FXML
    private TextField tf_emailCadastro;
    @FXML
    private TextField tf_telefone1Cadastro;
    @FXML
    private TextField tf_telefone2Cadastro;

    @FXML
    private ComboBox<String> cb_tipoCadastro;
    @FXML
    private TextField tf_ruaCadastro;
    @FXML
    private TextField tf_cidadeCadastro;
    @FXML
    private TextField tf_estadoCadastro;
    @FXML
    private TextField tf_bairroCadastro;
    @FXML
    private TextField tf_cepCadastro;
    @FXML
    private TextField tf_numeroCadastro;
    @FXML
    private TextField tf_complementoCadastro;

    @FXML
    private Label resultadoCadastro;

    // TABELA CLIENTE
    @FXML
    private TableColumn<Cliente, String> col_nome;
    @FXML
    private TableColumn<Cliente, String> col_cpf;
    @FXML
    private TableColumn<Cliente, String> col_telefone1;
    @FXML
    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("Altera  o realizada com sucesso!!");
131         } else {
132             confirmaAlteracao.setText("Erro ao realizar a altera  o!!");
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) {
```

```
        try {
            carregaTabelaCliente();
            tbl_cliente.getSelectionModel().selectedItemProperty().addListener(
                    (observable, oldValue, newValue) -> selecionarItemTablelaCliente(newValue));
        } catch (Exception ex) {
            //Logger.getLogger(ClienteController.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

}
```

Código 5: ClienteController.java

5.2 Pacote Fornecedor

5.2.1 Model

```
package opencarshop.fornecedor.model;

public class Fornecedor {

    private String cnpj;
    private String razaoSocial;
    private String email;
    private String telefone1;
    private String telefone2;
    private String descricao;

    public String getCnpj() {
        return cnpj;
    }

    public void setCnpj(String cnpj) {
        this.cnpj = cnpj;
    }

    public String getRazaoSocial() {
        return razaoSocial;
    }

    public void setRazaoSocial(String razaoSocial) {
        this.razaoSocial = razaoSocial;
    }

    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 String getDescricao() {
        return descricao;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
}
```

Código 6: Fornecedor.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.
 */
```

```java
package opencarshop.fornecedor.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
import opencarshop.Endereco;
import opencarshop.util.ConexaoMySQL;
import opencarshop.util.Utilidades;

/**
 *
 * @author JomarR
 */
public class FornecedorDAO {

    public Fornecedor getFornecedor(String cnpj) {
        ConexaoMySQL c = new ConexaoMySQL();
        Connection conn = null;
        PreparedStatement stmt = null;
        Utilidades u = new Utilidades();

        String query = "SELECT * FROM Fornecedor WHERE cnpj=?";

        Fornecedor fornecedor = new Fornecedor();
        try {
            conn = c.conectar();
            stmt = conn.prepareStatement(query);
            stmt.setString(1, cnpj);

            ResultSet resultado = stmt.executeQuery();

            if (resultado.next()) {
                fornecedor.setCnpj(resultado.getString("cnpj"));
                fornecedor.setDescricao(resultado.getString("razaoSocial"));
                fornecedor.setEmail(resultado.getString("email"));
                fornecedor.setTelefone1(resultado.getString("telefone1"));
                fornecedor.setTelefone2(resultado.getString("telefone2"));
                fornecedor.setDescricao(resultado.getString("descricao"));
            }

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

    public boolean cadastraFornecedor(Fornecedor fornecedor, Endereco end) {
        ConexaoMySQL c = new ConexaoMySQL();
        Connection conn = null;

        PreparedStatement stmtEnd = null;

        PreparedStatement stmtFornecedor = null;

        String queryEnd = "INSERT INTO Endereco (cep, estado, cidade, bairro, rua, numero, complemento, tipo) VALUES (?,?,?,?,?,?,?,?)";
        String queryFornecedor = "INSERT INTO Fornecedor (cnpj, razaoSocial, email, telefone1, telefone2, descricao, endereco) VALUES (?,?,?,?,?,(select LAST_INSERT_ID()))";

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

            stmtEnd = conn.prepareStatement(queryEnd);
            stmtFornecedor = conn.prepareStatement(queryFornecedor);

            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()));

            stmtFornecedor.setString(1, fornecedor.getCnpj());
            stmtFornecedor.setString(2, fornecedor.getRazaoSocial());
            stmtFornecedor.setString(3, fornecedor.getEmail());
            stmtFornecedor.setString(4, fornecedor.getTelefone1());
            stmtFornecedor.setString(5, fornecedor.getTelefone2());
            stmtFornecedor.setString(6, fornecedor.getDescricao());

            stmtEnd.execute();
            stmtFornecedor.execute();

            conn.commit();

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

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

Código 7: FornecedorDAO.java

5.2.2 View

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

```
<?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" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://
     javafx.com/fxml/1" fx:controller="opencarshop.fornecedor.controller.FornecedorController">
    <children>
        <Button layoutX="504.0" layoutY="159.0" mnemonicParsing="false" text="Cadastrar" />
        <Button layoutX="506.0" layoutY="210.0" mnemonicParsing="false" text="Atualizar" />
        <Button layoutX="510.0" layoutY="261.0" mnemonicParsing="false" text="Inativar" />
        <TableView prefHeight="343.0" prefWidth="448.0">
            <columns>
                <TableColumn prefWidth="75.0" text="CNPJ" />
                <TableColumn prefWidth="75.0" text="Razao Social" />
                <TableColumn prefWidth="75.0" text="Email" />
                <TableColumn prefWidth="75.0" text="Telefone" />
                <TableColumn prefWidth="75.0" text="Telefone 2" />
                <TableColumn prefWidth="75.0" text="Descri  o" />
            </columns>
        </TableView>
    </children>
</AnchorPane>
```

Código 9: Buscar.fxml

5.2.3 Controller

```
package opencarshop.fornecedor.controller;

import java.net.URL;
import java.text.ParseException;
import java.util.List;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Hyperlink;
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.Endereco;
import opencarshop.fornecedor.model.Fornecedor;
import opencarshop.fornecedor.model.FornecedorDAO;
import opencarshop.util.Utilidades;
```

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

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

Código 10: FornecedorController.java

5.3 Pacote Funcionário

5.3.1 Model

```java
package opencarshop.funcionario.model;

import java.time.LocalDate;
import java.util.Date;

public class Contrato {

    private Character cargo;
    private Double salario;
    private LocalDate dataInicio;
    private LocalDate dataTermino;

    public Character getCargo() {
        return cargo;
    }

    public void setCargo(Character cargo) {
        this.cargo = cargo;
    }

    public Double getSalario() {
        return salario;
    }

    public void setSalario(Double salario) {
        this.salario = salario;
    }

    public LocalDate getDataInicio() {
        return dataInicio;
    }

    public void setDataInicio(LocalDate dataInicio) {
        this.dataInicio = dataInicio;
    }

    public LocalDate getDataTermino() {
        return dataTermino;
    }

    public void setDataTermino(LocalDate dataTermino) {
        this.dataTermino = dataTermino;
    }

}
```

Código 11: Contrato.java

```java
package opencarshop.funcionario.model;

import java.time.LocalDate;

public class Funcionario {

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

    public Boolean getAtivo() {
        return ativo;
    }

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

    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 String getSenha() {
        return senha;
    }

    public void setSenha(String senha) {
```

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

Código 12: Funcionario.java

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

```
55
56          PreparedStatement stmtEnd = null;
57          PreparedStatement stmtCon = null;
58          PreparedStatement stmtFun = null;
59
60          String queryEnd = "INSERT␣INTO␣Endereco␣(cep,␣estado,␣cidade,␣bairro,␣rua,␣numero,␣complemento,tipo)␣VALUES␣
        (?,?,?,?,?,?,?,?)";
61          String queryFun = "INSERT␣INTO␣Funcionario␣(cpf,␣nome,␣senha,␣dataNascimento,␣email,␣telefone1,␣telefone2,␣
        endereco,␣ativo)␣VALUES␣(?,?,?,?,?,?,(select␣LAST_INSERT_ID()),?)";
62          String queryCon = "INSERT␣INTO␣Contrato␣(cargo,␣salario,␣dataInicio,␣dataTermino,funcionario)␣VALUES␣
        (?,?,?,?,?)";
63
64          try {
65              conn = c.conectar();
66              conn.setAutoCommit(false);
67
68              stmtEnd = conn.prepareStatement(queryEnd);
69              stmtFun = conn.prepareStatement(queryFun);
70              stmtCon = conn.prepareStatement(queryCon);
71
72              stmtEnd.setString(1, end.getCEP());
73              stmtEnd.setString(2, end.getEstado());
74              stmtEnd.setString(3, end.getCidade());
75              stmtEnd.setString(4, end.getBairro());
76              stmtEnd.setString(5, end.getRua());
77              stmtEnd.setInt(6, end.getNumero());
78              stmtEnd.setString(7, end.getComplemento());
79              stmtEnd.setString(8, Character.toString(end.getTipo()));
80
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;
```

```
        try {
            conn = c.conectar();
            PreparedStatement stmt = conn.prepareStatement(query);

            stmt.setString(1, func.getNome());
            stmt.setString(2, func.getSenha());
            stmt.setDate(3, Date.valueOf(func.getDataNascimento()));
            stmt.setString(4, func.getEmail());
            stmt.setString(5, func.getTelefone1());
            stmt.setString(6, func.getTelefone2());
            stmt.setBoolean(7, func.getAtivo());
            stmt.setString(8, func.getCpf());
            stmt.execute();
            conn.close();
            return true;
        } catch (Exception ex) {

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

Código 13: FuncionarioDAO.java

5.3.2 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="Identifica  o">
                    <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="Descri  o" />
                                <TextField fx:id="tf_razaoCadastro" layoutX="14.0" layoutY="31.0" prefHeight="25.0"
     prefWidth="350.0" promptText="Raz o␣Social" />
                                <Label layoutX="14.0" layoutY="6.0" text="Raz o␣Social" />
                                <Label layoutX="15.0" layoutY="70.0" text="CNPJ" />
                                <Label layoutX="15.0" layoutY="122.0" text="Descric o" />
                            </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="N mero" />
                                <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␣Endere o">
                                    <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.*?>

<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">
  <children>
    <TableView fx:id="tbl_funcionario" 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" />
        <TableColumn id="col_email" fx:id="col_email" prefWidth="193.0" text="Email" />
      </columns>
    </TableView>
    <TextField fx:id="tf_nomeCadastro" layoutX="846.0" layoutY="95.0" prefWidth="150.0" promptText="Nome" />
    <TextField fx:id="tf_emailCadastro" layoutX="846.0" layoutY="151.0" promptText="Email" />
    <TextField fx:id="tf_telefone1Cadastro" layoutX="675.0" layoutY="208.0" promptText="Telefone 1" />
    <TextField fx:id="tf_telefone2Cadastro" layoutX="846.0" layoutY="208.0" promptText="Telefone 2" />
    <DatePicker fx:id="dp_dataNascimentoCadastro" layoutX="675.0" layoutY="262.0" promptText="Data de nascimento" />
    <Label layoutX="844.0" layoutY="78.0" text="Nome:" />
    <Label layoutX="675.0" layoutY="134.0" text="Senha:" />
    <PasswordField fx:id="pf_senhaCadastro" layoutX="675.0" layoutY="151.0" promptText="Senha" />
    <Label layoutX="846.0" layoutY="134.0" text="Email:" />
    <Label layoutX="678.0" layoutY="191.0" text="Telefone 1:" />
    <Label layoutX="847.0" layoutY="191.0" text="Telefone 2:" />
    <Label layoutX="678.0" layoutY="245.0" text="Data de Nascimento:" />
    <CheckBox fx:id="cb_ativo" layoutX="903.0" layoutY="266.0" mnemonicParsing="false" text="Ativo" />
    <Button fx:id="btn_alterarCadastro" layoutX="782.0" layoutY="359.0" mnemonicParsing="false" onAction="#alterarCadastro" text="Salvar Altera  es" />
    <TextField fx:id="tf_cpfCadastro" editable="false" layoutX="675.0" layoutY="95.0" />
    <Label layoutX="675.0" layoutY="78.0" text="CPF:" />
    <Label fx:id="confirmaAlteracao" alignment="CENTER" contentDisplay="CENTER" layoutX="736.0" layoutY="325.0" prefHeight="17.0" prefWidth="199.0" textAlignment="CENTER" />
  </children>
</AnchorPane>
```

Código 15: Buscar.fxml

5.3.3 Controller

```
package opencarshop.funcionario.controller;

import java.io.IOException;
import java.net.URL;
import java.text.DecimalFormat;
import java.text.ParseException;
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.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
```

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

```
113
114     @FXML
115     private CheckBox cb_ativo;
116
117     @FXML
118     private Label confirmaAlteracao;
119
120     @FXML
121     private void autenticar(ActionEvent event) {
122         Funcionario funcionario;
123         FuncionarioDAO func = new FuncionarioDAO();
124         funcionario = func.getFuncionario(tf_cpf.getText());
125
126         if (funcionario.getCpf() != null) {
127             if (funcionario.getSenha().equals(pf_senha.getText())) {
128                 Parent root = null;
129                 try {
130                     root = FXMLLoader.load(getClass().getResource("/opencarshop/TelaPrincipal.fxml"));
131                     Scene scene = new Scene(root);
132                     Stage nStage = new Stage();
133                     nStage.setScene(scene);
134                     //nStage.setMaximized(true);
135                     nStage.setMaxHeight(768);
136                     nStage.setMaxWidth(1024);
137                     nStage.setTitle("OpenCarShop");
138                     nStage.setResizable(false);
139                     nStage.show();
140                     Stage stage = (Stage) cadastroLink.getScene().getWindow();
141                     stage.close();
142                 } catch (IOException e) {
143                     e.printStackTrace();
144                 }
145             } else {
146                 labelErro.setText("Login ou senha errado!!!");
147             }
148         } else {
149             labelErro.setText("Login ou senha errado!!!");
150         }
151     }
152
153     @FXML
154     private void cadastrar(ActionEvent event) throws ParseException {
155         //cb_cargoCadastro.setItems(cargos);
156         Funcionario func = new Funcionario();
157         Endereco end = new Endereco();
158         Contrato contr = new Contrato();
159         FuncionarioDAO f = new FuncionarioDAO();
160         Utilidades u = new Utilidades();
161
162         // OBJETO FUNCIONARIO
163         func.setCpf(tf_cpfCadastro.getText());
164         func.setNome(tf_nomeCadastro.getText());
165         func.setSenha(pf_senhaCadastro.getText());
166         func.setDataNascimento(dp_dataNascimentoCadastro.getValue());
167         func.setEmail(tf_emailCadastro.getText());
168         func.setTelefone1(tf_telefone1Cadastro.getText());
169         func.setTelefone2(tf_telefone2Cadastro.getText());
170         func.setAtivo(true);
171
172         // OBJETO ENDERECO
173         end.setCEP(tf_cepCadastro.getText());
174         end.setEstado(tf_estadoCadastro.getText());
175         end.setCidade(tf_cidadeCadastro.getText());
176         end.setBairro(tf_bairroCadastro.getText());
177         end.setRua(tf_ruaCadastro.getText());
178         end.setNumero(Integer.parseInt(tf_numeroCadastro.getText()));
179         end.setComplemento(tf_complementoCadastro.getText());
180         end.setTipo(cb_tipoCadastro.getValue().charAt(0));
181
182         // OBJETO CONTRATO
183         contr.setCargo(cb_cargoCadastro.getValue().charAt(0));
184         contr.setSalario(DecimalFormat.getInstance().parse(tf_salarioCadastro.getText()).doubleValue());
185         contr.setDataInicio(dp_dataInicioCadastro.getValue());
186         contr.setDataTermino(dp_dataTerminoCadastro.getValue());
187
188         if (f.cadastraFuncionario(func, end, contr)) {
189             resultadoCadastro.setText("Cadastrado com sucesso!!");
190         } else {
191             resultadoCadastro.setText("Erro ao cadastrar!! Tente novamente.");
192         }
193     }
194
195     @FXML
196     private void alterarCadastro(ActionEvent event) throws Exception {
197         Funcionario func = new Funcionario();
198         func.setCpf(tf_cpfCadastro.getText());
199         func.setNome(tf_nomeCadastro.getText());
200         func.setSenha(pf_senhaCadastro.getText());
201         func.setDataNascimento(dp_dataNascimentoCadastro.getValue());
202         func.setEmail(tf_emailCadastro.getText());
203         func.setTelefone1(tf_telefone1Cadastro.getText());
204         func.setTelefone2(tf_telefone2Cadastro.getText());
205         func.setAtivo(cb_ativo.isSelected());
206
```

```
        FuncionarioDAO f = new FuncionarioDAO();
        if (f.alteraFuncionario(func)) {
            confirmaAlteracao.setText("Altera  o realizada com sucesso!!");
        } else {
            confirmaAlteracao.setText("Erro ao realizar a altera  o!!");
        }
    }

    private void carregaTabelaFuncionario() throws Exception {
        col_nome.setCellValueFactory(new PropertyValueFactory<>("nome"));
        col_cpf.setCellValueFactory(new PropertyValueFactory<>("cpf"));
        col_telefone1.setCellValueFactory(new PropertyValueFactory<>("telefone1"));
        col_telefone2.setCellValueFactory(new PropertyValueFactory<>("telefone2"));
        col_email.setCellValueFactory(new PropertyValueFactory<>("email"));

        FuncionarioDAO f = new FuncionarioDAO();
        List<Funcionario> listaFuncionario = f.getAllFuncionario();
        ObservableList<Funcionario> observableListFuncionatio;

        observableListFuncionatio = FXCollections.observableArrayList(listaFuncionario);
        tbl_funcionario.setItems(observableListFuncionatio);
    }

    public void selecionarItemTablelaFuncionario(Funcionario funcionario) {
        if (funcionario.getCpf() != null) {
            tf_cpfCadastro.setText(funcionario.getCpf());
            tf_nomeCadastro.setText(funcionario.getNome());
            pf_senhaCadastro.setText(funcionario.getSenha());
            tf_emailCadastro.setText(funcionario.getEmail());
            tf_telefone1Cadastro.setText(funcionario.getTelefone1());
            tf_telefone2Cadastro.setText(funcionario.getTelefone2());
            dp_dataNascimentoCadastro.setValue(funcionario.getDataNascimento());
            cb_ativo.setSelected(funcionario.getAtivo());

        }
    }

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        try {
            carregaTabelaFuncionario();
            tbl_funcionario.getSelectionModel().selectedItemProperty().addListener(
                    (observable, oldValue, newValue) -> selecionarItemTablelaFuncionario(newValue));
        } catch (Exception ex) {
            //Logger.getLogger(FuncionarioController.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

}
```

Código 16: FuncionarioController.java

5.4 Pacote Serviço

5.4.1 Model

```
package opencarshop.servico.model;

public class Servico {

    private String descricao;
    private double valorPadrao;
    private boolean valorFixo;
    private int id;

    private String valorF;
    private String valorP;

    public Servico() {
    }

    public Servico(String desc, double valor, boolean bfixo) {

        this.descricao = desc;
        this.valorPadrao = valor;
        this.valorFixo = bfixo;

    }

    public String getDescricao() {
        return descricao;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
```

```
31
32      public double getValorPadrao() {
33          return valorPadrao;
34      }
35
36      public void setValorPadrao(double valorPadrao) {
37          this.valorPadrao = valorPadrao;
38      }
39
40      public boolean getValorFixo() {
41          return valorFixo;
42      }
43
44      public void setValorFixo(boolean valorFixo) {
45          this.valorFixo = valorFixo;
46      }
47
48      public int getId() {
49          return id;
50      }
51
52      public void setId(int id) {
53          this.id = id;
54      }
55
56      public String getValorP() {
57          return valorP;
58      }
59
60      public void setValorP(double v) {
61          String v2 = Double.toString(v).replace(".", ",");
62          this.valorP = "R$␣" + v2;
63      }
64
65      public String getValorF() {
66          return valorF;
67      }
68
69      public void setValorF(boolean t) {
70          if (t) {
71              this.valorF = "Sim";
72          } else {
73              this.valorF = "N  o";
74          }
75      }
76  }
```

Código 17: Servico.java

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

```
44          ConexaoMySQL c = new ConexaoMySQL();
45          Connection conn = null;
46          conn = c.conectar();
47
48          try {
49              PreparedStatement stmt = conn.prepareStatement(query);
50              ResultSet resultado = stmt.executeQuery();
51              while (resultado.next()) {
52                  Servico servico = new Servico();
53                  servico.setDescricao(resultado.getString("descri  o"));
54                  servico.setValorPadrao(resultado.getDouble("valorPadr o"));
55                  servico.setValorFixo(resultado.getBoolean("valorFixo"));
56                  servico.setValorF(resultado.getBoolean("valorFixo"));
57                  servico.setValorP(resultado.getDouble("valorPadr o"));
58                  servico.setId(resultado.getInt("id"));
59                  retorno.add(servico);
60              }
61          } catch (Exception e) {
62              e.printStackTrace();
63
64          }
65          conn.close();
66          return retorno;
67      }
68
69      public Boolean alteraServico(Servico srv) throws SQLException {
70          String query = "UPDATE Servico SET descri  o=?, valorPadr o=?, valorFixo=? WHERE id=?";
71
72          ConexaoMySQL c = new ConexaoMySQL();
73          Connection conn = null;
74          try {
75              conn = c.conectar();
76              PreparedStatement stmt = conn.prepareStatement(query);
77
78              stmt.setString(1, srv.getDescricao());
79              stmt.setDouble(2, srv.getValorPadrao());
80              stmt.setBoolean(3, srv.getValorFixo());
81              stmt.setInt(4, srv.getId());
82
83              stmt.execute();
84              conn.close();
85              return true;
86          } catch (Exception ex) {
87
88              Logger.getLogger(ServicoDAO.class.getName()).log(Level.SEVERE, null, ex);
89              return false;
90          }
91      }
92
93  }
```

Código 18: FuncionarioDAO.java

5.4.2 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="Descri  o do servi o">
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>
```

```
                </TextField>
                <Button fx:id="btnSignUp" layoutX="316.0" layoutY="74.0" mnemonicParsing="false" onAction="#
    cadastraServico" text="Cadastrar" />
                <Label layoutX="14.0" layoutY="14.0" text="Desccri  o:" />
                <Label layoutX="165.0" layoutY="14.0" text="Valor:" />
                <CheckBox fx:id="chkValue" layoutX="312.0" layoutY="35.0" mnemonicParsing="false" text="Valor␣Fixo" />
                <Label fx:id="labelErroServ" layoutX="106.0" layoutY="176.0" textFill="#ee0303">
                    <font>
                        <Font size="11.0" />
                    </font>
                </Label>
            </children>
        </AnchorPane>
    </children>


</AnchorPane>
```

Código 19: 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="Descri  o" />
                <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="Descri  o" />
        <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␣
    Servi o" />
        <TextField fx:id="tfid" editable="false" layoutX="632.0" layoutY="209.0" prefWidth="200.0" promptText="ID␣
    servi o" />
        <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␣Servi o:" />
        <Label layoutX="634.0" layoutY="234.0" text="Descri  o␣do␣servi o:" />
        <Label layoutX="634.0" layoutY="280.0" text="Valor␣do␣servi o:" />
    </children>
</AnchorPane>
```

Código 20: Buscar.fxml

5.4.3 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.
     */
    //Variav is
    @FXML
```

```
29      private TextField tfdescricao;
30
31      @FXML
32      private TextField tfvalor;
33
34      @FXML
35      private CheckBox chkValue;
36
37      @FXML
38      private TextField tfid;
39
40      @FXML
41      private Label labelErroServ;
42
43      @FXML
44      private Label confirmaAtualizacao;
45
46  //Colunas da tabela listar servi os
47      @FXML
48      private TableColumn<Servico, String> col_descricao;
49      @FXML
50      private TableColumn<Servico, Double> col_valor;
51      @FXML
52      private TableColumn<Servico, String> col_tpvalor;
53      @FXML
54      private TableView<Servico> tbl_servico;
55
56      //metodos
57      @FXML
58      public void cadastraServico(ActionEvent event) {
59          //istancia objeto servi o para configura ao de atributos
60          Servico servico;
61          //instacia objeto para inser  o 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("Descri  o␣deve␣ter␣at ␣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 servi o
81          servico = new Servico(descri, valorDouble, chkValue.isSelected());
82
83          if (!servDao.insertServico(servico)) {
84
85              labelErroServ.setText("Servi o␣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("Altera  o␣realizada␣com␣sucesso!!");
118         } else {
119             confirmaAtualizacao.setText("Erro␣ao␣realizar␣a␣altera  o!!");
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 21: ServicoController.java

5.5 Pacote Veiculo

5.5.1 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
```

```
    public void setValor(double valor) {
        this.valor = valor;
    }

    public boolean isOpcionalVidrosEletricos() {
        return opcionalVidrosEletricos;
    }

    public void setOpcionalVidrosEletricos(boolean opcionalVidrosEletricos) {
        this.opcionalVidrosEletricos = opcionalVidrosEletricos;
    }

    public boolean isOpcionalTravasEletricas() {
        return opcionalTravasEletricas;
    }

    public void setOpcionalTravasEletricas(boolean opcionalTravasEletricas) {
        this.opcionalTravasEletricas = opcionalTravasEletricas;
    }

    public boolean isOpcionalAr() {
        return opcionalAr;
    }

    public void setOpcionalAr(boolean opcionalAr) {
        this.opcionalAr = opcionalAr;
    }

    public boolean isOpcionalFarolNeblina() {
        return opcionalFarolNeblina;
    }

    public void setOpcionalFarolNeblina(boolean opcionalFarolNeblina) {
        this.opcionalFarolNeblina = opcionalFarolNeblina;
    }

    public boolean isOpcionalAltoFalantes() {
        return opcionalAltoFalantes;
    }

    public void setOpcionalAltoFalantes(boolean opcionalAltoFalantes) {
        this.opcionalAltoFalantes = opcionalAltoFalantes;
    }

}
```

Código 22: Veiculo.java

```
package opencarshop.veiculo.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
import opencarshop.util.ConexaoMySQL;

public class VeiculoDAO {

    public List<Veiculo> getAllVeiculo() throws Exception {
        String query = "SELECT * FROM Veiculo";
        List<Veiculo> retorno = new ArrayList<>();
        ConexaoMySQL c = new ConexaoMySQL();
        Connection conn = null;
        conn = c.conectar();
        try {
            PreparedStatement stmt = conn.prepareStatement(query);
            ResultSet resultado = stmt.executeQuery();
            while (resultado.next()) {

                Veiculo veiculo = new Veiculo();
                veiculo.setModelo(resultado.getString("modelo"));
                veiculo.setAno(resultado.getInt("ano"));
                veiculo.setVersao(resultado.getString("versao"));
                veiculo.setQuantidade(resultado.getInt("qntd"));
                veiculo.setValor(resultado.getDouble("valor"));
                veiculo.setOpcionalAltoFalantes(resultado.getBoolean("altoFalantes"));
                veiculo.setOpcionalAr(resultado.getBoolean("ar"));
                veiculo.setOpcionalFarolNeblina(resultado.getBoolean("farolNeblina"));
                veiculo.setOpcionalTravasEletricas(resultado.getBoolean("travasEletricas"));
                veiculo.setOpcionalVidrosEletricos(resultado.getBoolean("vidrosEletricos"));

                retorno.add(veiculo);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        conn.close();
        return retorno;
    }

    /*
```

```
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 23: VeiculoDAO.java

5.5.2 View

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
```

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


<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="Vers o" />
                <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␣El tricos" />
                <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 24: 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="Vers o" />
                <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␣
     El tricos" />
                <TableColumn id="col_travasEletricas" maxWidth="5000.0" minWidth="10.0" prefWidth="100.0" text="Travas␣
     El tricas" />
                <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 25: Buscar.fxml

5.5.3 Controller

```
1  package opencarshop.veiculo.controller;
2
3  import java.net.URL;
4  import java.util.List;
5  import java.util.Properties;
6  import java.util.ResourceBundle;
7  import javafx.collections.FXCollections;
8  import javafx.collections.ObservableList;
9  import javafx.event.ActionEvent;
10 import javafx.fxml.FXML;
11 import javafx.fxml.Initializable;
12 import javafx.scene.control.CheckBox;
13 import javafx.scene.control.Label;
14 import javafx.scene.control.TableColumn;
15 import javafx.scene.control.TableView;
16 import javafx.scene.control.TextField;
17 import javafx.scene.control.cell.PropertyValueFactory;
18 import opencarshop.veiculo.model.Veiculo;
19 import opencarshop.veiculo.model.VeiculoDAO;
20
21 public class VeiculoController implements Initializable {
22
23     //Variav is
24     @FXML
25     private Label lb_result;
26
27     @FXML
28     private TextField tf_modelo;
29
30     @FXML
31     private TextField tf_ano;
32
33     @FXML
34     private TextField tf_versao;
35
36     @FXML
37     private TextField tf_qntd;
38
39     @FXML
40     private TextField tf_valor;
41
42     @FXML
43     private CheckBox cb_VidrosEletricos;
44
45     @FXML
46     private CheckBox cb_TravasEletricas;
47
48     @FXML
49     private CheckBox cb_Ar;
50
51     @FXML
52     private CheckBox cb_FarolNeblina;
53
54     @FXML
55     private CheckBox cb_AltoFalante;
56
57     //TABELA VEICULO
58     @FXML
59     private TableColumn<Veiculo, String> col_modelo;
60
61     @FXML
62     private TableColumn<Veiculo, String> col_ano;
63
64     @FXML
65     private TableColumn<Veiculo, String> col_versao;
66
67     @FXML
68     private TableColumn<Veiculo, String> col_qntd;
69
70     @FXML
71     private TableColumn<Veiculo, String> col_valor;
72
73     @FXML
74     private TableColumn<Veiculo, String> col_vidrosEletricos;
75
76     @FXML
77     private TableColumn<Veiculo, String> col_travasEletricas;
78
79     @FXML
80     private TableColumn<Veiculo, String> col_ar;
81
82     @FXML
83     private TableColumn<Veiculo, String> col_farolNeblina;
84
85     @FXML
86     private TableColumn<Veiculo, String> col_altoFalantes;
87
88     // tabela
89     @FXML
90     private TableView<Veiculo> tbl_veiculo;
91
```

```
92      @FXML
93      private void cadastrarVeiculo(ActionEvent event) {
94
95          //instanciando objeto
96          Veiculo veiculo = new Veiculo();
97          //instancia objeto para inser  o de objeto cadastrado no banco
98          VeiculoDAO veiculoDao = new VeiculoDAO();
99
100         veiculo.setModelo(tf_modelo.getText());
101         veiculo.setAno(Integer.parseInt((tf_ano.getText())));
102         veiculo.setVersao(tf_versao.getText());
103         veiculo.setQuantidade(Integer.parseInt(tf_qntd.getText()));
104         veiculo.setValor(Double.parseDouble(tf_valor.getText()));
105         veiculo.setOpcionalAltoFalantes(Boolean.parseBoolean(cb_AltoFalante.getText()));
106         veiculo.setOpcionalAr(Boolean.parseBoolean(cb_Ar.getText()));
107         veiculo.setOpcionalFarolNeblina(Boolean.parseBoolean(cb_FarolNeblina.getText()));
108         veiculo.setOpcionalTravasEletricas(Boolean.parseBoolean(cb_TravasEletricas.getText()));
109         veiculo.setOpcionalVidrosEletricos(Boolean.parseBoolean(cb_VidrosEletricos.getText()));
110
111         if (veiculoDao.insertVeiculo(veiculo)) {
112             lb_result.setText("Ve culo 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 26: VeiculoController.java

5.6 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 usu rio do mysql
9       private final String USERNAME = "desenvolvimento";
10      //Senha do mysql
```

```
11      private final String PASSWORD = "ds123";
12      //Dados de caminho, porta e nome da base de dados que ir  ser feita a conex o
13      private final String DATABASE_URL = "jdbc:mysql://tharlysson.com:3306/opencarshop";
14
15      /**
16       * Cria uma conex o com o banco de dados MySQL utilizando o nome de usu rio
17       * e senha fornecidos
18       *
19       * @param username
20       * @param senha
21       * @return uma conex o com o banco de dados
22       * @throws Exception
23       */
24      public Connection conectar() throws Exception {
25          Class.forName("com.mysql.jdbc.Driver"); //Faz com que a classe seja carregada pela JVM
26          //Cria a conex o com o banco de dados
27          Connection connection = DriverManager.getConnection(DATABASE_URL, USERNAME, PASSWORD);
28          return connection;
29      }
30  }
```

Código 27: ConexaoMySQL.java

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

Código 28: Utilidades.java

5.7 Outros

5.8 SQL

```
1   -- MySQL Script generated by MySQL Workbench
2   -- 10/30/16 15:58:11
3   -- Model: New Model    Version: 1.0
4   -- MySQL Workbench Forward Engineering
5
6   SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
7   SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
8   SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
9
10  -- -----------------------------------------------------
11  -- Schema opencarshop
12  -- -----------------------------------------------------
13  DROP SCHEMA IF EXISTS `opencarshop` ;
```

```
14
15  -- -----------------------------------------------------
16  -- Schema opencarshop
17  -- -----------------------------------------------------
18  CREATE SCHEMA IF NOT EXISTS `opencarshop` DEFAULT CHARACTER SET utf8 ;
19  USE `opencarshop` ;
20
21  -- -----------------------------------------------------
22  -- Table `opencarshop`.`Endereco`
23  -- -----------------------------------------------------
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` DATETIME NULL,
96    `email` VARCHAR(50) NULL,
97    `telefone1` VARCHAR(45) GENERATED ALWAYS AS (),
98    `telefone2` VARCHAR(45) NULL,
99    `ativo` TINYINT(1) NULL DEFAULT 1,
100   `endereco` INT NOT NULL,
101   PRIMARY KEY (`cpf`),
102   INDEX `fk_Cliente_Endereco1_idx` (`endereco` ASC),
103   CONSTRAINT `fk_Cliente_Endereco1`
104     FOREIGN KEY (`endereco`)
105     REFERENCES `opencarshop`.`Endereco` (`id`)
106     ON DELETE NO ACTION
107     ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;


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

CREATE TABLE IF NOT EXISTS `opencarshop`.`Fornecedor` (
  `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` (
  `veiculo` INT NOT NULL,
  `valorFinal` DECIMAL(10,2) NULL,
  `chasi` VARCHAR(45) NULL,
  `desconto` DECIMAL(6,2) NULL,
  `venda` INT NOT NULL,
  PRIMARY KEY (`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`
```

```
390 -- -----------------------------------------------------
391 DROP TABLE IF EXISTS `opencarshop`.`ItemPeca` ;
392
393 CREATE TABLE IF NOT EXISTS `opencarshop`.`ItemPeca` (
394   `id` INT NOT NULL AUTO_INCREMENT,
395   `valorFinal` DECIMAL(10,2) NULL,
396   `desconto` DECIMAL(6,2) NULL,
397   `peca` INT NOT NULL,
398   `venda` INT NOT NULL,
399   PRIMARY KEY (`id`, `peca`, `venda`),
400   INDEX `fk_ItemPeca_Peca1_idx` (`peca` ASC),
401   INDEX `fk_ItemPeca_Venda1_idx` (`venda` ASC),
402   CONSTRAINT `fk_ItemPeca_Peca1`
403     FOREIGN KEY (`peca`)
404     REFERENCES `opencarshop`.`Peca` (`id`)
405     ON DELETE NO ACTION
406     ON UPDATE NO ACTION,
407   CONSTRAINT `fk_ItemPeca_Venda1`
408     FOREIGN KEY (`venda`)
409     REFERENCES `opencarshop`.`Venda` (`codigo`)
410     ON DELETE NO ACTION
411     ON UPDATE NO ACTION)
412 ENGINE = InnoDB;
413
414
415 SET SQL_MODE=@OLD_SQL_MODE;
416 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
417 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Código 29: OpenCarShop.sql

6 Diagramas

6.1 Diagrama de Casos de Uso

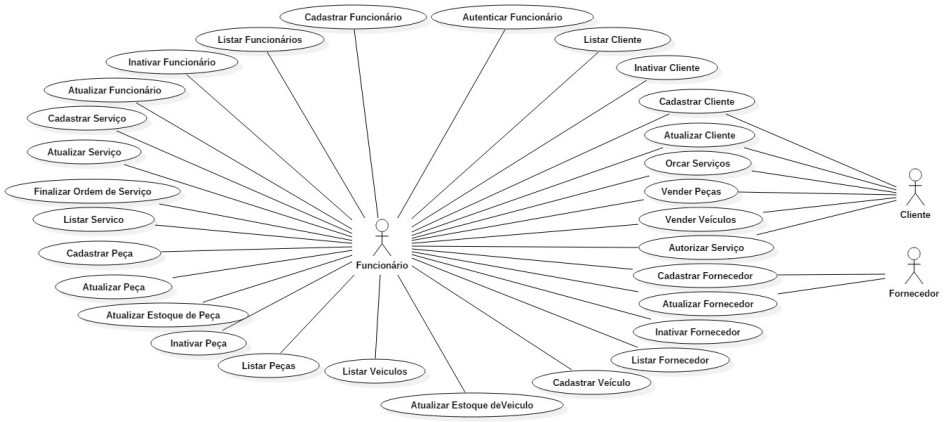

Figura 1: Diagrama de Casos de Uso

6.2 Diagrama de Classes - Analise

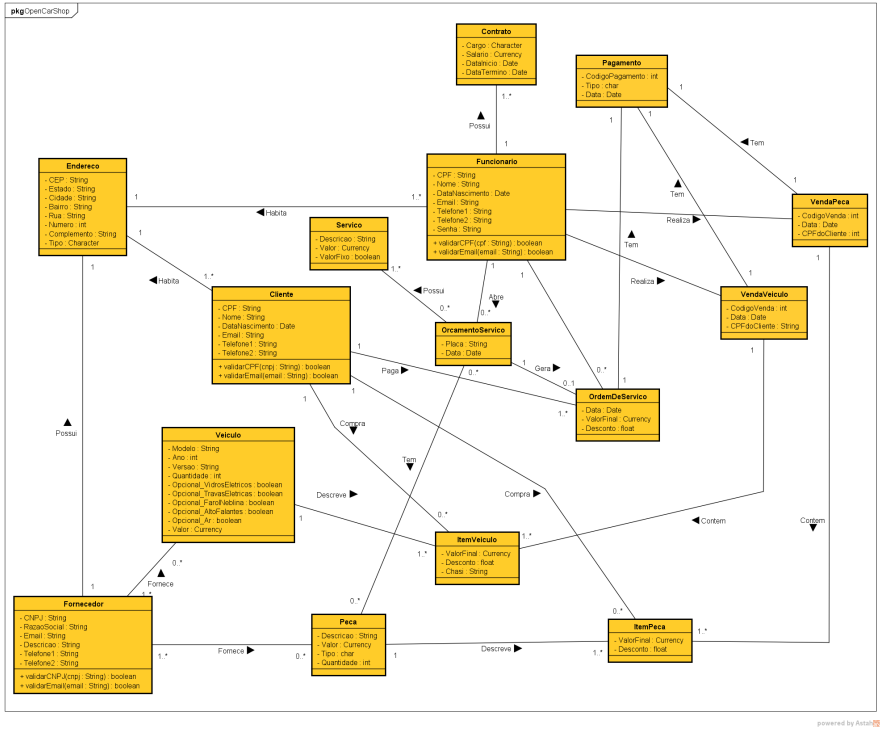

Figura 2: Diagrama de Classes - Analise

6.3 Diagrama de Classes - Projeto

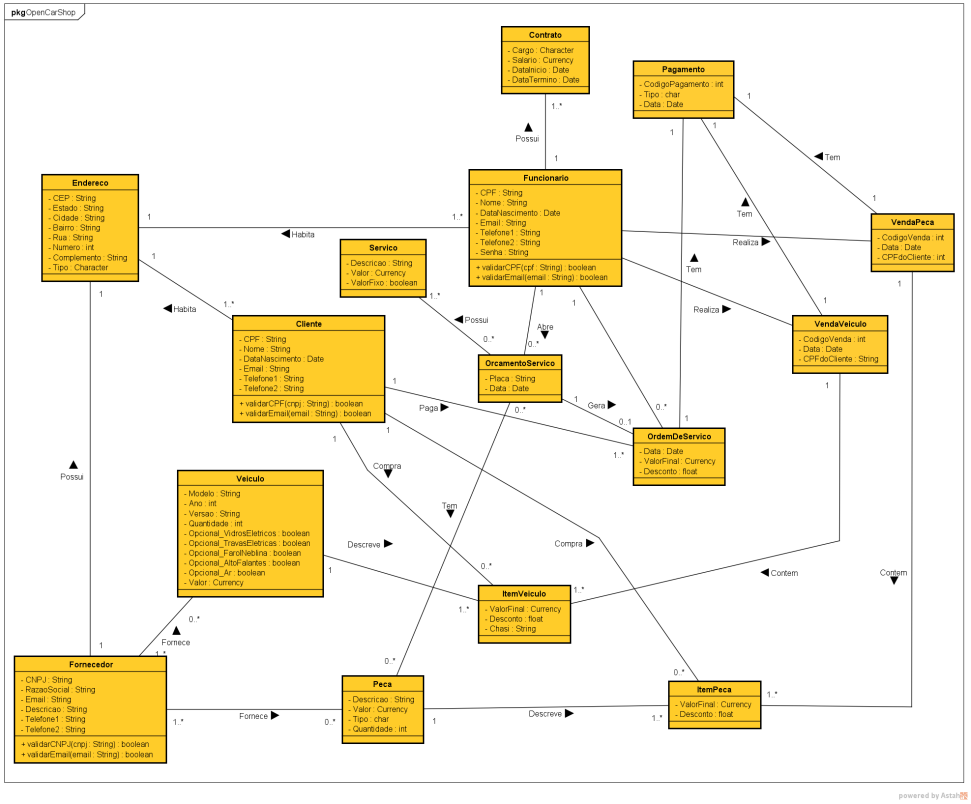

Figura 3: Diagrama de Classes - Projeto

6.4 Diagrama Entidade Relacionamento

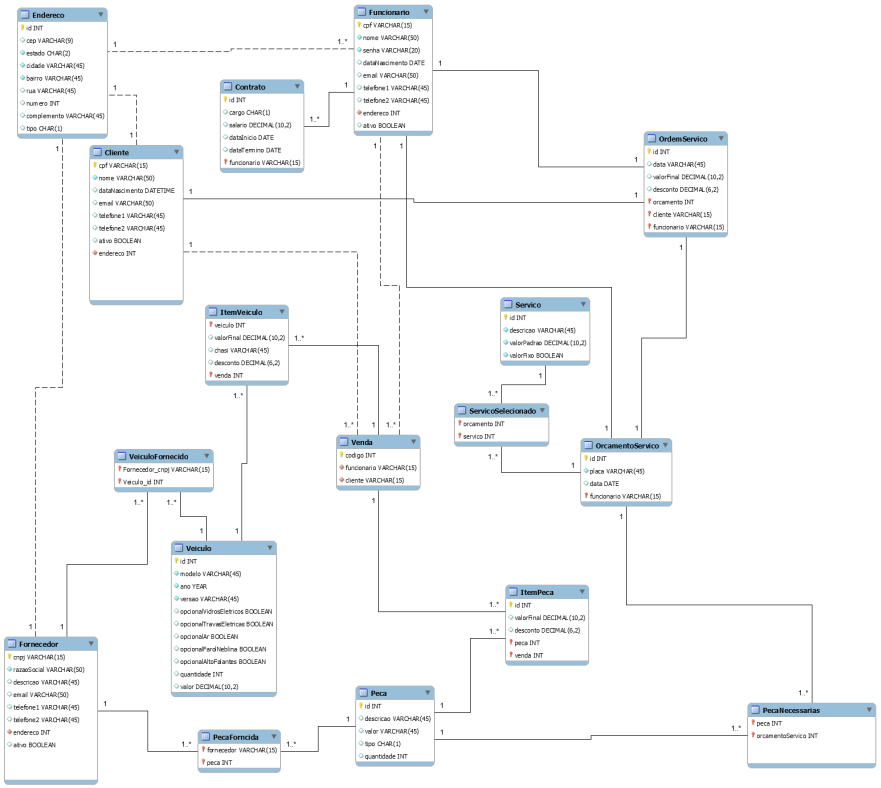

Figura 4: Diagrama de Entidade Relacionamento