**RENT A CAR Locadora de Veiculos**

**Especificação de Requisitos de Software**

**Para Controle de Agendamento de Clientes para Frota**

**Versão 1.2**

**Integrantes:**

Michael Advincola Roriz

Sergio Marangoni

**Índice Analítico**

[1. Introdução](http://www.wthreex.com/rup/portugues/webtmpl/templates/req/rup_srsuc.htm#1.)

[1.1 Finalidade](http://www.wthreex.com/rup/portugues/webtmpl/templates/req/rup_srsuc.htm#1.1)

[1.2 Escopo](http://www.wthreex.com/rup/portugues/webtmpl/templates/req/rup_srsuc.htm#1.2)

[1.3 Definições, Acrônimos e Abreviações](http://www.wthreex.com/rup/portugues/webtmpl/templates/req/rup_srsuc.htm#Abbreviations)

[1.4 Referências](http://www.wthreex.com/rup/portugues/webtmpl/templates/req/rup_srsuc.htm#1.4)

[2](http://www.wthreex.com/rup/portugues/webtmpl/templates/req/rup_srsuc.htm#/h) - Requisitos e Regras de Negócios

2.1 - Requisitos Funcionais

2.2 - Requisitos Não Funcionais

2.3 - Regras de Negócios

[3](http://www.wthreex.com/rup/portugues/webtmpl/templates/req/rup_srsuc.htm#/h) - Relatórios de Caso de Uso

3.1 - Diagrama de Caso de Uso

3.2 - Especificação de Caso de Uso

4 - Matrizes de Relacionamento

4.1 - Matriz Requisitos Funcionais X Caso de Uso

4.2 - Matriz Requisitos Funcionais X Regra de Negócios

5. Modelagem Banco de Dados

5.1 Modelo Conceitual

5.2 Diagrama Entidade Relacionamento

5.3 DDL - Data Definition Language

**Especificação de Requisitos de Software**

# **1.** **Introdução**

## 1.1 Finalidade

Coletar, analisar e definir necessidades e recursos de nível superior do Rent A Car Locadora de Veículos com base em conhecimentos adquiridos no curso Análise e Desenvolvimento de Sistemas.

## 1.2 Escopo

O objetivo do Sistema é criar um agendamento eletrônico de locação de veículos, que permita a empresa ter um controle eficiente de veículos disponíveis para locação e veículos já alocados.

## 1.3 Definições, Acrônimos e Abreviações

|  |  |  |  |
| --- | --- | --- | --- |
| **RF** | Requisito Funcional | **RNF** | Requisito não funcional |
| **RN** | Regras de negócio | **UC** | Caso de Uso |
| **CAL** | Controle de Agendamento de Locação | **BD** | Banco de Dados |
| **JAVA** | Linguagem de programação | **UML** | É uma linguagem para visualização, especificação, construção e documentação |
| **SQL** | Linguagem de pesquisa declarativa padrão para banco de dados relacionais |  |  |

## 1.4 Referências

**2.**                  **Requisitos e Regra de Negócios**

**2.1**               **Requisitos Funcionais**

|  |  |  |
| --- | --- | --- |
| **Requisito** | **Importância** | **Complexidade** |
| *[RF1] - O Sistema deve permitir o controle de agendamento de locações, bem como a data da retirada e devolução prevista de cada veículo.* | *Alta* | *Alta* |
| *[RF2] - O Sistema deve prover um cadastro de clientes* | *Alta* | *Média* |
| *[RF3] - O sistema deve permitir o cancelamento de uma reserva, pela pessoa responsável pela entrada de dados no sistemas, de acordo com política da empresa.* | *Alta* | *Média* |
| *[RF4] - O sistema deve permitir consultas ao acervo. As consultas poderão ser feitas informando uma (ou uma combinação) de informações do veículo, dentre elas: marca (ou parte dele), modelo, ou categoria (ex: conversível).* | *Média* | *Média* |

**2.2**               **Requisitos Não Funcionais**

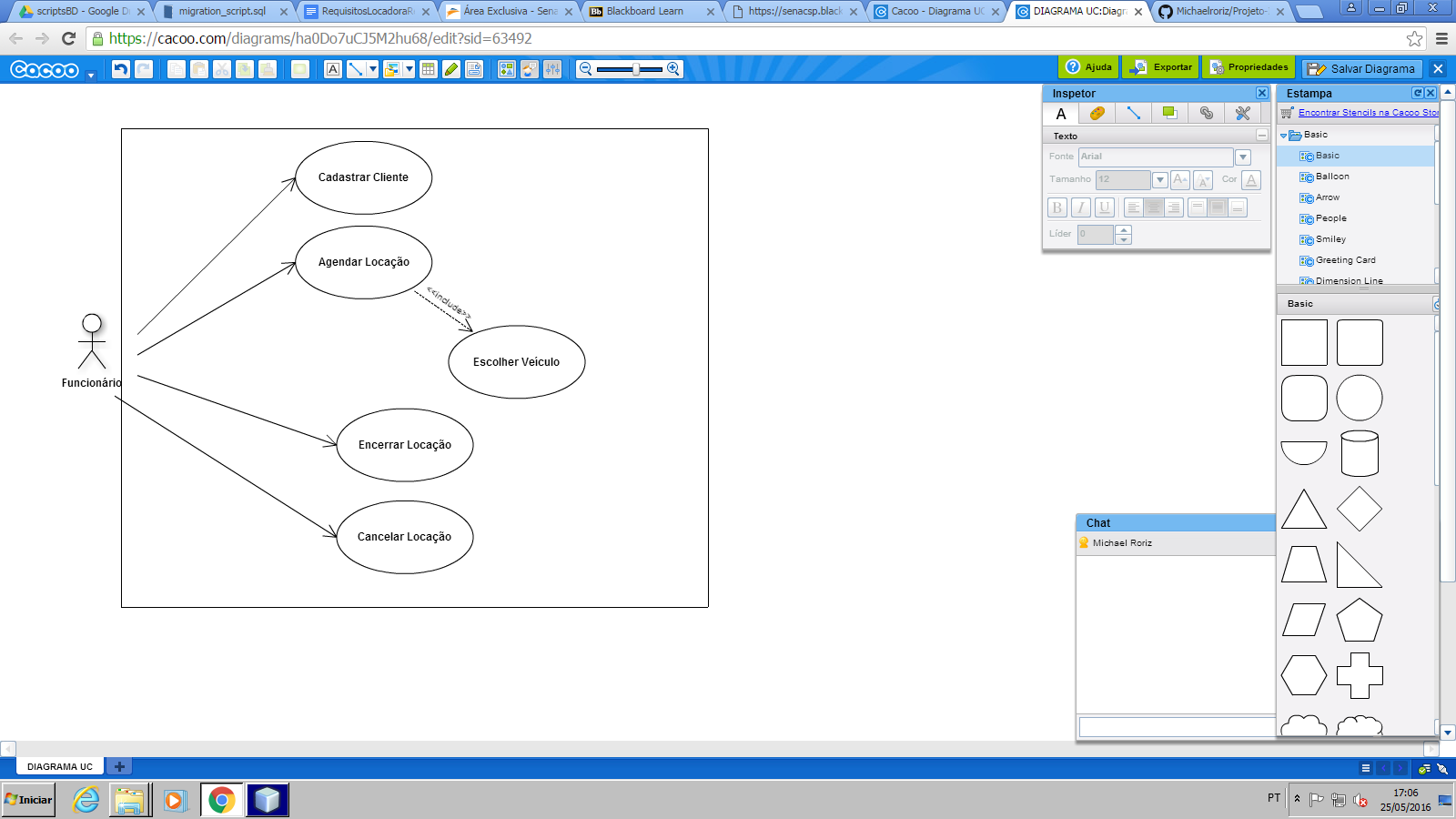
|  |  |  |  |
| --- | --- | --- | --- |
| **Requisito** | **Tipo** | **Importância** | **Complexidade** |
| *[RNF1] - O aplicativo deve ser desenvolvido em linguagem JAVA* | *Suportabilidade* | *Alta* | *Alta* |
| *[RNF2] - Esta aplicação deverá possuir uma interface gráfica que será desenvolvida em JAVA SWING\** | *Interface* | *Alta* | *Alta* |
| *[RNF3] - Esta aplicação deverá utilizar o SQLITE como banco de dados* | *Suportabilidade* | *Alta* | *Alta* |
| *[RNF4] - O sistema deve ser fácil de usar, devendo-se evitar a digitação desnecessária de informações, de modo a dar agilidade ao processo.* | *Facilidade de Operação* | *Média* | *Média* |
| *[RNF5] - O sistema deve ser fácil de manter, de modo a acomodar novas funcionalidades ou até mesmo adaptação para outras locadoras.* | *Manutenibilidade* | *Alta* | *Alta* |
| *[RNF6] - O sistema deve permitir o cliente cancelar a locação ou será cancelado automaticamente caso o cliente não confirme locação no prazo determinado.* | *Cancelamento* | *Alta* | *Alta* |

**2.3**               **Regras de Negócios**

|  |  |
| --- | --- |
| **Regra de Negócio** | **Definição** |
| *[RN001] - O sistema deverá gerar relatório consolidado* | *O relatório deverá exibir a posição atual do controle de agendamento, mostrando quais as reservas existem em sistema.* |
| *[RN002] - O sistema deverá gerar relatório de veículos* *alocados e locados.* | *O Relatório deverá exibir os veículos que foram alocados e os que estão disponíveis no sistema.* |
| *[RN003] - O sistema deverá certificar se a reserva foi confirmada.* | *Uma reserva expira quando passadas mais do que 24h de sua comunicação para o cliente.* |
| *[RN004] - O sistema não deverá aceitar o cadastro de clientes com idade inferior à dezoito anos.* | *Apenas maiores de idade podem ter o cadastro no site.* |
| *[RN005] - O sistema deverá limitar o local de retirada e entrega apenas para locadora de veículos.* | *O cliente obrigatoriamente terá que retirar o veículo na locadora após agendamento realizado e confirmado e devolver o veículo na locadora ao final da locação.* |

**3.**                  **Relatórios de Caso de Uso**

**3.1**               **Diagrama de Caso de Uso**



**3.2 Caso de Uso: Cadastrar Cliente [UC0001]**

**Descrição**

*Realiza a inserção de um novo registro de cliente no banco de dados do sistema, armazenando suas informações pessoais para o mesmo efetuar o agendamento da locação desejada. Uma mensagem será informada de confirmação do cadastro ter sido efetuado com sucesso ou uma mensagem de erro caso os dados estejam incorretos ou já exista um cadastro do mesmo.*

**Pré-condições**

*Não deve existir um cliente já cadastrado com o mesmo CPF.*

*Idade do cliente não pode ser inferior à 18.*

**Fluxo de Eventos**

**Fluxo Básico**

1. O funcionário escolhe a opção “Fazer Cadastro” na tela inicial do sistema.
2. O funcionário preenche os campos “Nome”, “Sexo”, “Data de nascimento”, “CPF”, “Endereço”, “Numero” , “Complemento”,”Bairro”, “Cidade”, “UF” estabelecidos pelo cliente.
3. O funcionário confirma clicando em “Cadastrar”, ou “Cancelar” para sair da tela de cadastro.
4. Após clicar em “Cadastrar”, o sistema irá armazenar os dados estabelecidos pelo cliente.

**Fluxos Alternativos**

**(A01) – CPF já existente no Banco de Dados ou dado(s) inválido(s)**

Caso o CPF já exista no banco de dados o sistema recusará a inclusão. Caso o CPF seja invalido o mesmo conterá letras ou caracteres especiais. Neste caso igualmente será recusado o cadastro do cliente.

***Caso de Uso:* Agendar Locação[UC0002]**

**Descrição**

*Definição da data da locação e a data combinada para o retorno do veículo.*

**Pré-condições**

O usuário deverá estar cadastrado no sistema.

**Fluxo de Eventos**

1. O funcionário escolhe a opção “Agendar Locação” na tela inicial do sistema.
2. Aciona o caso de uso UC003 - Escolher Veículo.
3. O funcionário preenche os campos “Data de retirada” e “Horário de retirada”, “Data de devolução” e “Horário de devolução” estabelecido pelo cliente.
4. O funcionário confirma clicando em “Agendar”.
5. O sistema armazena os dados estabelecidos pelo cliente.

**Fluxos Alternativos**

**(A01) – Não há horários disponíveis**

Caso o cliente estabeleça um horário e data (Passo 3) que coincida com outro horário no sistema, o sistema sugere um horário mais próximo disponível para o cliente.

**Caso de Uso: Escolher Veículo [UC0003]**

**Descrição**

*Cliente será informado sobre a lista de carros disponíveis para locação pelo funcionário.*

**Pré-condições**

*Não deverá exibir os veículos que já estão locados.*

**Fluxo de Eventos**

1. Será exibido ao funcionário a lista de veículos que estão disponíveis para a locação e o cliente decide sobre a escolha do veículo que se encaixa nas suas necessidades.
2. O funcionário seleciona o veículo escolhido pelo cliente.
3. O funcionário clica na opção “Avançar” para para confirmar.

**Caso de Uso: Cancelar da Locação [UC0004]**

**Descrição**

Caso o cliente não confirme o agendamento dentro de um período de 24 horas, seu agendamento será cancelado.

**Pré-condições**

O cliente deve estar cadastrado e o agendamento deverá estar com o status “Aguardando Confirmação”

**Fluxo de Eventos**

1. O funcionário entra em contato com o cliente para confirmar a locação.
2. O funcionário após ser notificado de que o cliente não tem interesse em prosseguir a locação, escolhe a opção “Cancelar Locação”
3. O funcionário confirma a operação de cancelamento.

**Caso de Uso: Encerrar da Locação [UC0005]**

**Descrição**

*Após o término do prazo de Locação o cliente realiza a devolução do veículo.*

**Pré-condições**

É necessário que a locação tenha sido efetivada e que o veículo tenha sido retirado pelo cliente.

**Fluxo de Eventos**

1. O funcionário retorna para efetivar a devolução do veículo.
2. Após a verificação do veículo pelo funcionário o mesmo realiza a baixa em sistema do veículo.
3. O funcionário confirma o recebimento do veículo para a locadora.

# 

# **4.** **Matrizes de Relacionamento**

**4.1 - Matriz Requisitos Funcionais X Caso de Uso**

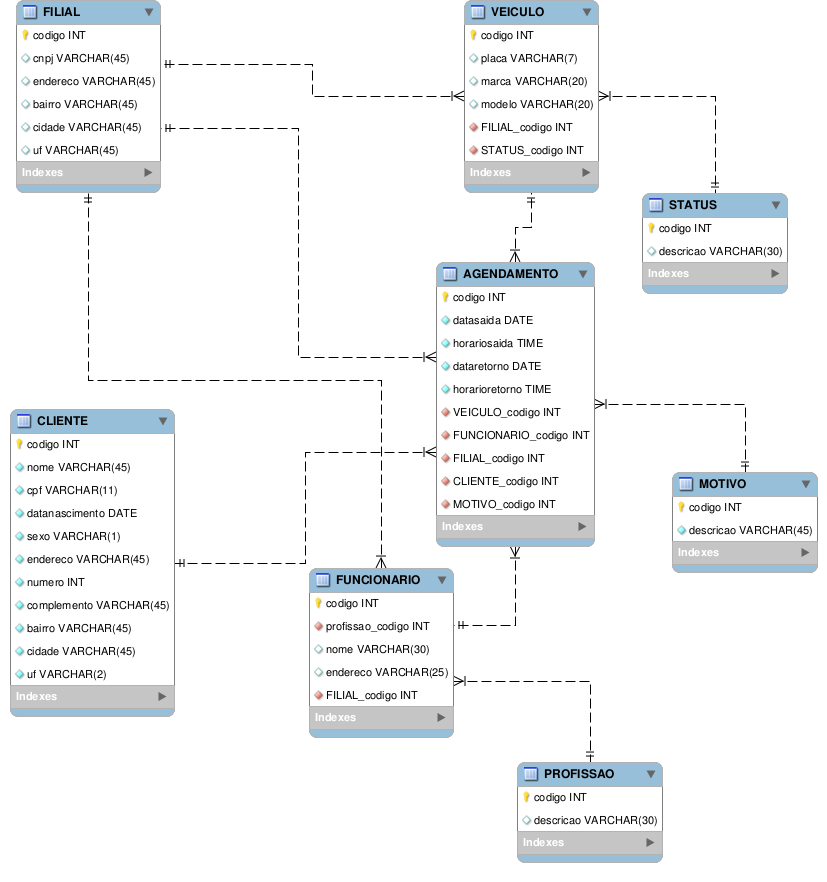
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisitos/Caso de Uso** | **UC001** | **UC002** | **UC003** | **UC004** | **UC005** |
| **[RF1]** |  | **x** |  |  | **x** |
| **[RF2]** | **x** |  |  |  |  |
| **[RF3]** |  |  |  | **x** |  |
| **[RF4]** |  |  | **x** |  |  |

**4.2 - Matriz Requisitos Funcionais X Regra de Negócios**

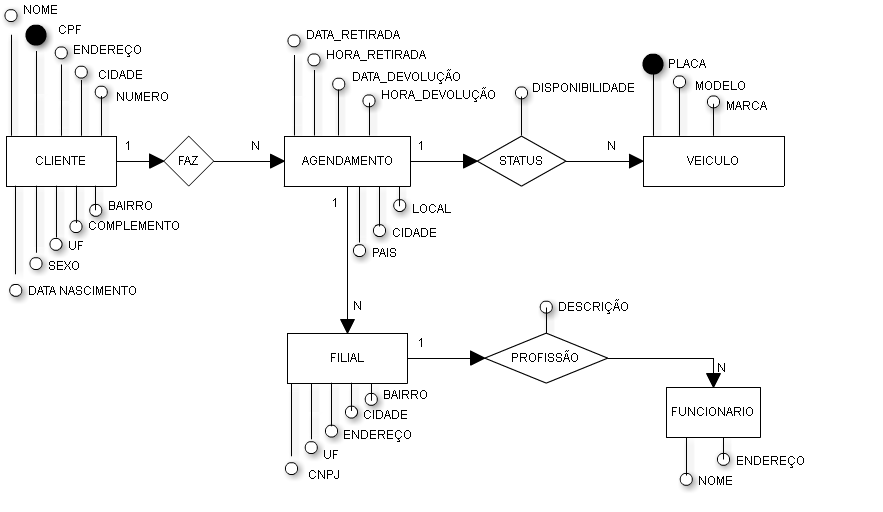
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisitos/Regra de Negócios** | **RN001** | **RN002** | **RN003** | **RN004** | **RN005** |
| **[RF1]** | **x** | **x** |  |  | x |
| **[RF2]** |  |  |  | **x** |  |
| **[RF3]** |  |  | **x** |  |  |
| **[RF4]** | **x** | **x** |  |  |  |

# **5.** **Matrizes de Relacionamento**

## 5.1 Modelo Lógico



## 5.2 Diagrama Entidade Relacionamento



## 

## 

## 

## 

## 

## 

## 

## 5.3 DDL - Data Definition Language

-- MySQL Script generated by MySQL Workbench

-- Sex 10 Jun 2016 12:13:20 BRT

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

-- -----------------------------------------------------

-- Schema DBagendamento

-- -----------------------------------------------------

DROP TABLE IF EXISTS AGENDAMENTO;

DROP TABLE IF EXISTS VEICULO;

DROP TABLE IF EXISTS FUNCIONARIO;

DROP TABLE IF EXISTS MOTIVO;

DROP TABLE IF EXISTS PROFISSAO;

DROP TABLE IF EXISTS FILIAL;

DROP TABLE IF EXISTS STATUS;

DROP TABLE IF EXISTS CLIENTE;

-- -----------------------------------------------------

-- Schema DBagendamento

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `DBagendamento` DEFAULT CHARACTER SET utf8 ;

USE `DBagendamento` ;

-- -----------------------------------------------------

-- Table `DBagendamento`.`CLIENTE`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `DBagendamento`.`CLIENTE` (

`codigo` INT AUTO\_INCREMENT,

`nome` VARCHAR(45) NOT NULL,

`cpf` VARCHAR(11) NOT NULL,

`datanascimento` DATE NOT NULL,

`sexo` VARCHAR(1) NOT NULL,

`endereco` VARCHAR(45) NOT NULL,

`numero` INT NOT NULL,

`complemento` VARCHAR(45) NOT NULL,

`bairro` VARCHAR(45) NOT NULL,

`cidade` VARCHAR(45) NOT NULL,

`uf` VARCHAR(2) NOT NULL,

PRIMARY KEY (`codigo`),

UNIQUE INDEX `cpf\_UNIQUE` (`cpf` ASC))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `DBagendamento`.`STATUS`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `DBagendamento`.`STATUS` (

`codigo` INT AUTO\_INCREMENT,

`descricao` VARCHAR(30) NULL,

PRIMARY KEY (`codigo`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `DBagendamento`.`FILIAL`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `DBagendamento`.`FILIAL` (

`codigo` INT AUTO\_INCREMENT,

`cnpj` VARCHAR(45) NULL,

`endereco` VARCHAR(45) NULL,

`bairro` VARCHAR(45) NULL,

`cidade` VARCHAR(45) NULL,

`uf` VARCHAR(45) NULL,

PRIMARY KEY (`codigo`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `DBagendamento`.`PROFISSAO`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `DBagendamento`.`PROFISSAO` (

`codigo` INT AUTO\_INCREMENT,

`descricao` VARCHAR(30) NULL,

PRIMARY KEY (`codigo`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `DBagendamento`.`MOTIVO`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `DBagendamento`.`MOTIVO` (

`codigo` INT AUTO\_INCREMENT,

`descricao` VARCHAR(45) NOT NULL,

PRIMARY KEY (`codigo`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `DBagendamento`.`FUNCIONARIO`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `DBagendamento`.`FUNCIONARIO` (

`codigo` INT AUTO\_INCREMENT,

`profissao\_codigo` INT NOT NULL,

`nome` VARCHAR(30) NULL,

`endereco` VARCHAR(25) NULL,

`FILIAL\_codigo` INT NOT NULL,

PRIMARY KEY (`codigo`),

INDEX `fk\_FUNCIONARIO\_profissao\_idx` (`profissao\_codigo` ASC),

INDEX `fk\_FUNCIONARIO\_FILIAL1\_idx` (`FILIAL\_codigo` ASC),

CONSTRAINT `fk\_FUNCIONARIO\_profissao`

FOREIGN KEY (`profissao\_codigo`)

REFERENCES `DBagendamento`.`PROFISSAO` (`codigo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_FUNCIONARIO\_FILIAL1`

FOREIGN KEY (`FILIAL\_codigo`)

REFERENCES `DBagendamento`.`FILIAL` (`codigo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `DBagendamento`.`VEICULO`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `DBagendamento`.`VEICULO` (

`codigo` INT AUTO\_INCREMENT,

`placa` VARCHAR(7) NULL,

`marca` VARCHAR(20) NULL,

`modelo` VARCHAR(20) NULL,

`FILIAL\_codigo` INT NOT NULL,

`STATUS\_codigo` INT NOT NULL,

PRIMARY KEY (`codigo`),

UNIQUE INDEX `placa\_UNIQUE` (`placa` ASC),

INDEX `fk\_VEICULO\_FILIAL1\_idx` (`FILIAL\_codigo` ASC),

INDEX `fk\_VEICULO\_STATUS1\_idx` (`STATUS\_codigo` ASC),

CONSTRAINT `fk\_VEICULO\_FILIAL1`

FOREIGN KEY (`FILIAL\_codigo`)

REFERENCES `DBagendamento`.`FILIAL` (`codigo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_VEICULO\_STATUS1`

FOREIGN KEY (`STATUS\_codigo`)

REFERENCES `DBagendamento`.`STATUS` (`codigo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `DBagendamento`.`AGENDAMENTO`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `DBagendamento`.`AGENDAMENTO` (

`codigo` INT NOT NULL AUTO\_INCREMENT,

`datasaida` DATE NOT NULL,

`horariosaida` TIME NOT NULL,

`dataretorno` DATE NOT NULL,

`horarioretorno` TIME NOT NULL,

`VEICULO\_codigo` INT NOT NULL,

`FUNCIONARIO\_codigo` INT NOT NULL,

`FILIAL\_codigo` INT NOT NULL,

`CLIENTE\_codigo` INT NOT NULL,

`MOTIVO\_codigo` INT NOT NULL,

PRIMARY KEY (`codigo`),

INDEX `fk\_AGENDAMENTO\_VEICULO1\_idx` (`VEICULO\_codigo` ASC),

INDEX `fk\_AGENDAMENTO\_FUNCIONARIO1\_idx` (`FUNCIONARIO\_codigo` ASC),

INDEX `fk\_AGENDAMENTO\_FILIAL1\_idx` (`FILIAL\_codigo` ASC),

INDEX `fk\_AGENDAMENTO\_CLIENTE1\_idx` (`CLIENTE\_codigo` ASC),

INDEX `fk\_AGENDAMENTO\_MOTIVO1\_idx` (`MOTIVO\_codigo` ASC),

CONSTRAINT `fk\_AGENDAMENTO\_VEICULO1`

FOREIGN KEY (`VEICULO\_codigo`)

REFERENCES `DBagendamento`.`VEICULO` (`codigo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_AGENDAMENTO\_FUNCIONARIO1`

FOREIGN KEY (`FUNCIONARIO\_codigo`)

REFERENCES `DBagendamento`.`FUNCIONARIO` (`codigo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_AGENDAMENTO\_FILIAL1`

FOREIGN KEY (`FILIAL\_codigo`)

REFERENCES `DBagendamento`.`FILIAL` (`codigo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_AGENDAMENTO\_CLIENTE1`

FOREIGN KEY (`CLIENTE\_codigo`)

REFERENCES `DBagendamento`.`CLIENTE` (`codigo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_AGENDAMENTO\_MOTIVO1`

FOREIGN KEY (`MOTIVO\_codigo`)

REFERENCES `DBagendamento`.`MOTIVO` (`codigo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;