

UNIP – UNIVERSIDADE PAULISTA

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

PIM

PROJETO INTEGRADO MULTIDICIPLINAR

TEMA:

**Desenvolvimento de um Sistema Geral de Gerenciamento de Frotas**

Semestre: 4ª semestre 2019

Alunos:

Daniel Henrique M. Linhares – D75GBB-1

Luis Gustavo B. Bravin – N27966-4

Matheus Leme dos Santos – N29221-0

Matheus Manabu P. Ikenaga – D6565F-1

Mauro Vieira Pesente – N307FE-5

**São Paulo**

**2019**

Alunos:

Daniel Henrique M. Linhares – D75GBB-1

Luis Gustavo B. Bravin – N27966-4

Matheus Leme dos Santos – N29221-0

Matheus Manabu P. Ikenaga – D6565F-1

Mauro Vieira Pesente – N307FE-5

**Desenvolvimento de um Sistema Geral de Gerenciamento de Frotas**

Trabalho Semestral para obtenção do título de graduação em Análise e Desenvolvimento de Sistemas apresentado à Universidade Paulista – UNIP

Orientador: Glauco Paparoto

**São Paulo**

**2019**

# RESUMO

Uma empresa de controle de frotas de veículos solicitou para uma equipe de desenvolvedores de Software para realizar, no momento, uma análise de requisitos para um sistema que gerencie a frota de carros de uma empresa no setor de telecomunicações. O sistema deve ter portabilidade para desktop, web e mobile, além de abranger todo o território nacional. A empresa de controle de frotas solicita também que ela possa usar esse sistema não só para o cliente de telecomunicações, mas para todos os outros clientes que ela pode vir a ter. O foco do projeto é o gerenciamento de veículos. A empresa deve auxiliar aos seus clientes a migração para veículos com fontes renováveis e limpas de energia. O sistema deverá contemplar algumas partes acessíveis para que eventuais usuários portadores de deficiência possam utilizá-lo, além de alguns dos itens já gerenciados pela empresa, como Veículos, Seguros, Motoristas (CNH, validade da Habilitação, etc.), Viagens (rotas, data de início e retorno etc.), Entrada/Saída (de garagens credenciadas ou estacionamento), Finanças, Estoque de peças, Pneus, Manutenções Corretivas e Preventivas, Abastecimentos, Quilometragem, Multas, Sinistros, Avisos automáticos (pelo celular ao motorista etc.), Relatórios (multas, consumo, revisões etc.), Aluguel de veículos ao cliente (frota interna pequena).

Palavras-Chave: Análise de Requisitos. Gerenciamento de Frotas. Sistema. Desenvolvimento.

# ABSTRACT

A vehicle fleet control company has requested a team of Software developers to perform, at the time, a requirements analysis for a system that manages a company's fleet of cars in the telecommunications sector. The system must have portability for desktop, web and mobile, besides covering the entire national territory. The fleet control company also requests that it can use this system not only for the telecommunications client, but for all other customers it may have. The focus of the project is vehicle management. The company should assist its customers in migrating to vehicles with renewable and clean energy sources. The system should contemplate some accessible parts so that any disabled users can use it, in addition to some of the items already managed by the company, such as vehicles, insurance, drivers (CNH, validity of authorization to Drive etc.), travel (routes, departure/entry date, etc.), entry/exit (from accredited garages or parking), finance, parts inventory, tires, corrective and preventive maintenance, supplies, mileage, fines, claims, automatic notices (by mobile to Driver etc.), reports (traffic tickets, consumption, reviews etc.), vehicle rental to customer (small internal fleet).

Keywords: Requirements analysis. Fleet management. System. Development.

**Resumo**

[1. Introdução 6](#_Toc25876474)

[2. Metodologia e Ferramentas Utilizada 7](#_Toc25876475)

[3. Requisitos 9](#_Toc25876476)

[3.1. Levantamento de Requisitos 10](#_Toc25876477)

[4. UML 12](#_Toc25876478)

[4.1 Processos do Sistema 14](#_Toc25876479)

[5. Banco de Dados 28](#_Toc25876480)

[6. Gerenciamento de Projetos. 38](#_Toc25876481)

[6.1 Backup e Restauração do Sistema 38](#_Toc25876482)

[7. Testes do Sistema 40](#_Toc25876483)

[8. Manuais 43](#_Toc25876484)

[8.1 Manual de Instalação 43](#_Toc25876485)

[8.2 Manual do Sistema 44](#_Toc25876486)

[9. Conclusão 53](#_Toc25876487)

[10. Referências 54](#_Toc25876488)

# Introdução

Uma empresa de gerenciamento de frotas recebeu um novo cliente, de telecomunicações que possui sua própria frota para gerenciar e eventualmente alugar carros da empresa. Por ser um cliente grande, a empresa de gerenciamento decidiu atualizar o método ao qual trabalha de uma planilha para um programa completo, decidindo contratar uma equipe de desenvolvimento de Softwares para desenvolver um sistema que gerencie não só a frota do cliente, mas que seja possível de fazer a gerencia de todo e qualquer tipo de frota de veículos.

O sistema necessita realizar tudo que a empresa já realizava em suas planilhas, como por exemplo controle de veículos disponíveis, controle dos motoristas e sinistros, entre outros. Além de o sistema também necessita portabilidade, ou seja, funcionar em várias plataformas como computadores e celulares, acessibilidade para portadores de deficiências além de avisos automático para os motoristas. Com a intenção de alcançar essa acessibilidade o programa será desenvolvido em uma versão para Desktop que pode ser acessada sem acesso à internet e também em Web que poderá ser acessada de qualquer equipamento que possua acesso à internet, podendo ser um celular, um computador de qualquer tipo, entre outros.

Para poder concretizar todo o programa, o grupo fez todo o levantamento de requisitos necessários para o sistema, preparou diagramas UML para ajudar o entendimento do sistema e mostrou um exemplo das telas do sistema, tudo isso feito no semestre passado, restando apenas o gerenciamento do projeto e a programação do sistema, para poder entregar esse software de vez para a empresa.

Esse trabalho contempla não só os dados levantados no semestre anterior que foram atualizados com o decorrer do projeto, como também o sistema finalizado e a parte de gerência de projetos.

# Metodologia e Ferramentas Utilizada

Dada a proposta do trabalho que é de tanto levantar todos os requisitos, diagramas de UML e prototipação de telas, que foi feita no semestre anterior, quanto a programação e concretização desse projeto no semestre atual, analisamos todas as metodologias de desenvolvimento e escolhemos o Método Cascata para Desenvolvimento desse projeto, dado o tempo disponível de coleta e facilidade de utilização para programadores que já possuem pouca experiência.

A metodologia Cascata se fundamenta em um desenvolvimento do projeto sem nenhuma mudança no decorrer do projeto, sempre tratando a etapa em que está por completo antes de ir para a próxima, e como foco do projeto já foi definido no início e não haverá mudanças no meio do caminho identificamos essa opção como a mais válida.

O método cascata se baseia nas seguintes etapas:

1. Concepção – A ideia do Projeto.
2. Levantamento de Requisitos – Levantar as necessidades do sistema.
3. Análise de Requisitos – Entender o Procedimento.
4. Homologação de Requisitos – Concluir as informações obtidas.
5. Modelagem – Criar os diagramas UML para melhor entendimento.
6. Projeto – Construção do Software.
7. Testes – Caixa branca/ Caixa Preta.
8. Manutenção – Suporte ao sistema, correções e melhorias.

Para esse Projeto seguiremos todas as etapas mostradas acima.

Essa Metodologia foi utilizada para criar o sistema para duas plataformas, uma para Desktop onde será feito o download do programa e outra em Web para poder acessar de qualquer equipamento contanto que tenha acessa a internet.

Ambas as plataformas serão desenvolvidas em Java, atualmente pertencente a Oracle, é uma linguagem de programação interpretada mundialmente conhecida devido a sua ampla gama de equipamentos no qual a linguagem funciona, qualquer equipamento que possua um *Java Virtual Machine* (JMC) pode rodar os códigos Java, desde computadores de mesa e notebooks até sistemas de GPS, celulares e muitos outros. Para desenvolver em Java foi utilizado a IDE NetBeans, fornecido pela própria empresa dona dessa linguagem, é uma ferramenta que auxilia no desenvolvimento e roda os códigos não só de Java como JavaScript, PHP, HTML, entre outros.

O Banco de Dados do sistema foi desenvolvido utilizando o MySQL, como esse banco também é da própria Oracle, a integração entre o banco e o código Java se torna maior e mais confiável para o sistema, atualmente MySQL é o sistema de banco de dados mais utilizado no mundo todo.

Para poder manter a parte de Web funcionando é necessário um programa de servidor que sustente o código Java e permita a utilização do mesmo, para essa função utilizamos o Apache Tomcat, um servidor Web especializado em Java, é um software livre e foi desenvolvido pela Apache Software Foundation, é o servidor mais confiável para o código de Java devido a ser criado solenemente para essa função.

Outras ferramentas que foram usadas pelo grupo são o Trello, Slack e o Git/GitHub, Trello é uma ferramenta para gerenciamento de projetos que foi utilizada para dividir cada função dos membros do grupo e acompanhar o andar de cada uma delas em conjunto para o projeto inteiro, o Slack é um aplicativo de mensagens profissional em nuvem, com o intuito de centralizar todo o assunto envolta do projeto e poder acompanhar o andamento junto com a equipe, essa ferramenta possui interação com o GitHub, uma plataforma de hospedagem de código e controle de versão, onde todos os arquivos do projeto eram guardados e fornecido para todo o grupo, sempre que algo era adicionado ao GitHub, o Slack informava automaticamente, atualizando a todos os membros.

# Requisitos

O grupo pesquisou por softwares já existentes na área e foram analisados os sistemas: Master Frotas, Produsoft e Unifrota.

Antes de iniciar qualquer projeto, é necessário adquirir uma base conhecimento sobre o produto ao qual está trabalhando, levantar o que é necessário para realizar o sistema ideal para o usuário sem fugir do objetivo principal. Existem várias formas de levantar os requisitos, a escolhida para esse projeto foi a utilização de questionários enviados para empresas já estabelecidas na área de gerenciamento de frotas, o questionário foi desenvolvido buscando entender não só a própria função do gerenciamento como também entender a parte técnica do sistema, esse método é muito útil para obter as informações detalhadas porém não garante uma grande quantidade de respostas, e para complementar os dados adquiridos foi realizado analise das documentações disponíveis publicamente nos sites de softwares já existentes de gerenciamento de frotas, as informações resultantes adquiridas foram o suficiente para começar a desenvolver os requisitos para o sistema.

Foi enviado um e-mail com um questionário elaborado pelo grupo, para entender como é feita a gestão e outras informações com relação ao próprio sistema

O questionário foi:

1. Como é feita a Gerência da Frota de Veículos?
2. Como é feita a Gerência dos Seguros?
3. Como é feito o Cadastro dos Motoristas?
4. A cobrança é feita por Quilometragem ou somente pela Diária?
5. Vocês possuem garagem ou estacionamentos credenciados? Se sim, como é gerenciada a entrada e saída de veículos?
6. Quais as principais receitas e despesas da empresa? (Não é necessário especificar)
7. Vocês possuem um estoque de peças e uma equipe especializada em manutenção ou fazem a manutenção através de uma equipe terceirizada?
8. Como é feito o Controle de Multas e Sinistros?
9. Como é feita a escolha dos veículos da Frota?
10. Vocês fazem algum laudo pericial ao receber o veículo de volta?
11. Vocês utilizam algum ripo de sistema em nuvem?
12. Qual a(s) linguagem(s) de programação utilizada(s) no desenvolvimento do sistema?
13. Qual o banco de Dados utilizado por vocês?
14. Com qual frequência é realizado o backup das aplicações e dos dados?
15. Vocês utilizam alguma metodologia de desenvolvimento? Se sim, qual?
16. O Sistema é multiplataforma?
17. O que você implantaria em seu sistema?
18. Como foram feitos os casos de teste do sistema?
19. Quais são as partes que interagem com o sistema?
20. Vocês possuem certificação ISO/IEC ? Se sim, quais?
21. Quais as principais funcionalidades do sistema?

Infelizmente, nenhuma respondeu ao questionário ou liberou acesso ao sistema para que fosse analisado, então o levantamento foi completamente feito utilizando as documentações disponibilizadas pelas próprias empresas nos sites delas.

Os softwares de gestão presentes hoje no mercado, contém muitas informações nas telas e muito pouco intuitivo, com isso levantamos as informações para a construção do SIGEFROTAS que tem a proposta de ser um sistema em nuvem, responsivo, com um design inovador e completamente intuitivo.

## Levantamento de Requisitos

Um levantamento precisa registrar não só os requisitos funcionais, ou seja, o que realmente é necessário e desejado pelo cliente para o programa fique realmente útil para eles, como também os requisitos não funcionais de usabilidade, portabilidade, confiabilidade do sistema etc. Os Requisitos ficaram da seguinte forma:

|  |  |
| --- | --- |
| **Funcionais** | **Não Funcionais** |
| Gerencia de Veículos | Usabilidade |
| Controle de Seguros | Confiabilidade |
| Controle de Motoristas e Viagens | Portabilidade |
| Gestão de Finanças | Manutenibilidade |
| Controle de Peças e Estoque | Segurança |
| Manutenção dos Veículos | Disponibilidade |
| Controle de Multas e Sinistros | Performance |
| Gerencia de Aluguel de Veículos | Tolerância a Falha |

Tabela 1 – Requisitos Funcionais e não funcionais levantados.

Conforme visto na Tabela 1 foram levantados 16 requisitos, sendo 8 Funcionais e 8 Não Funcionais para o SIGEFROTAS

# UML

Unified Modeling Language mais conhecido como UML é uma linguagem única de modelagem para especificação, visualização, construção e documentação de um projeto de software, a modelagem é uma das principais operações que levam a implementação de um software, a UML gera diagramas que servem para seguir durante uma criação, e esses diagramas ajudam a visualizar o software de forma gráfica e entender como ele funciona, como se fosse a planta de um prédio a ser construído, tanto na visão do cliente como na visão da equipe que vai desenvolver a aplicação.

Existem vários diagramas que podem ser usados na linguagem UML, porém os utilizados nesse trabalho foram os diagramas de classe, caso de uso, sequência e atividades.

Descrevendo o que é cada um e para que serve, o diagrama de classe descreve a estrutura de um sistema, apresentando classes, atributos e métodos, esse diagrama é bastante útil pois define todas as classes que o sistema precisa ter, e cada classe do diagrama representa uma tabela do banco de dados.

O diagrama de caso de uso descreve as principais funcionalidades do sistema e como os usuários interagem com o mesmo sistema, nele temos uma visão geral do que o sistema faz do ponto de vista do usuário.

Diagrama de sequência determina a sequência de eventos que ocorrem em um processo, descreve o fluxo da operação e tempo em que a operação vai ocorrer.

As funcionalidades do diagrama de atividades são detalhar o passo a passo de uma determinada atividade e mostrar as atividades que compõe um processo do sistema, mostrando onde começa e onde encerra as atividades.

O diagrama de Caso de Uso a seguir detalha a interação dos usuários com todo o sistema, não só o consumidor final, como também a interação do gestor, do Administrador, e de qualquer pessoa que necessite utilizar o sistema, sendo o primeiro e mais importante na hora de levantar os requisitos para todo o projeto.

Uma imagem contendo texto, mapa

Descrição gerada automaticamente

Figura 1 – Diagrama de Caso e Uso para o Sigefrotas.

## **4.1 Processos do Sistema**

Depois de levantar todos os requisitos e de preparar o diagrama de Caso e Uso (Figura 1), é possível preparar outros tipos de diagramas que vão definir os processos do sistema e como eles interagem entre si, são esses diagramas o de Classe, Sequência, Atividade e Processos, o de classe defines as classes do sistemas e as interação entre elas, o de sequência define a ordem dos eventos e a ordem que o usuário vai realizar dentro e fora do sistema o de atividade define a ordem de execução da classes e por fim o de processos que retrata os principais elementos de fluxos de processos, dentre esses modelos o único que funciona para todo o sistema é o de classes conforme visto na Figura 2, tanto o de sequência como o de atividade e o de processos possuem versão diferente para cada funcionalidade do sistema.

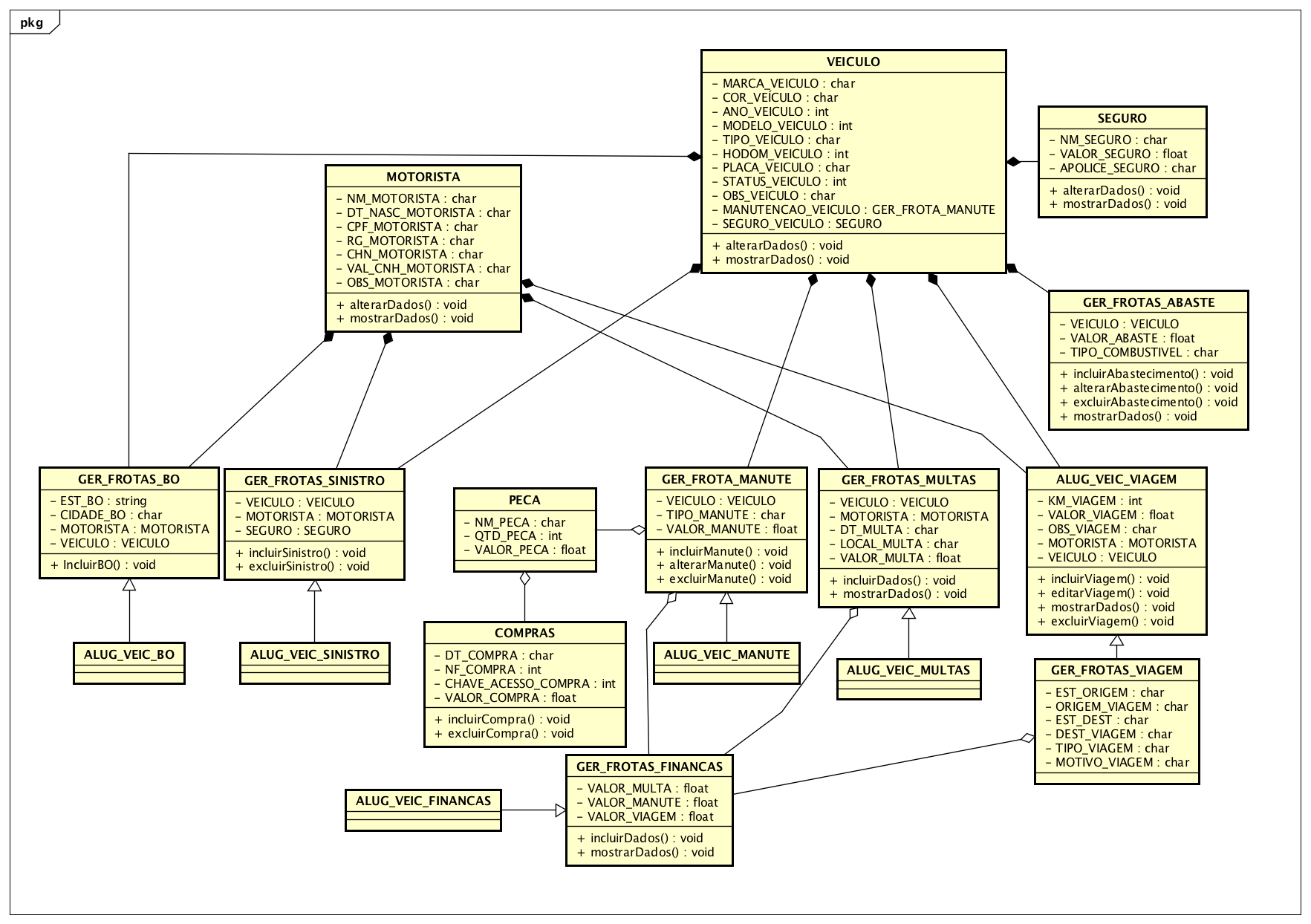


Figura 2 – Diagrama de classes do Sigefrotas.

Uma vez definido o diagrama de classes podemos definir os diagrama de sequência, atividades e processos, começando pelos Abastecimento, que definem como será feito o abastecimento do veículo e como o usuário irá registrar isso no sistema, a Figura 3 revela a ordem das ações do usuário, primeiramente abastecendo o veículo, depois indo no sistema na parte de Cadastro de Contas e por fim o sistema irá salvar essa despesa do cliente na base de dados.

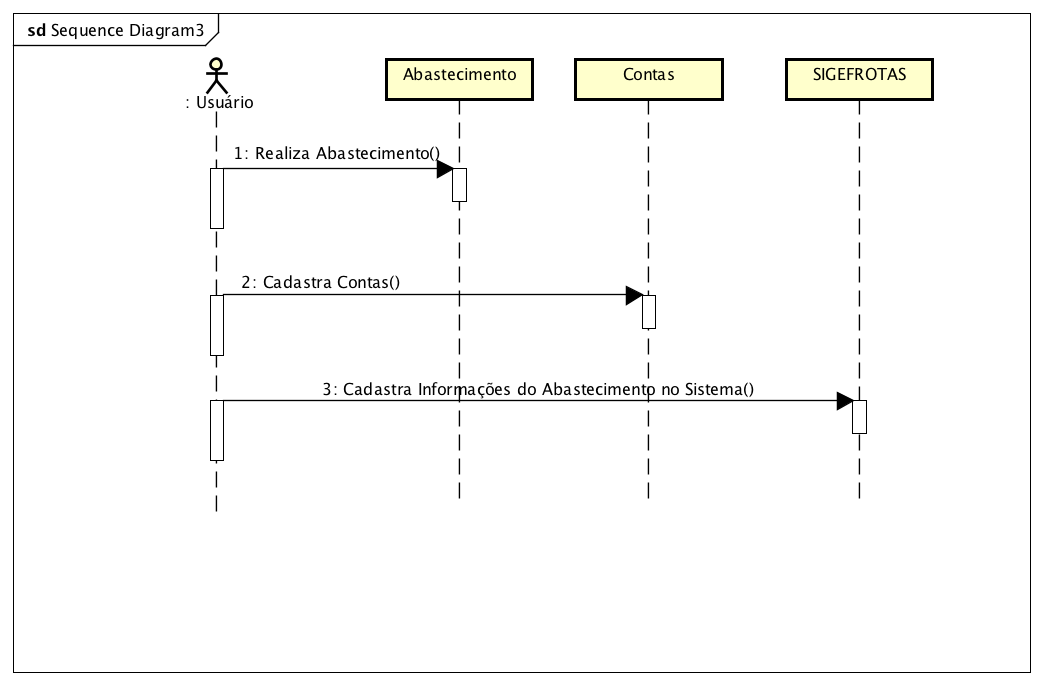


Figura 3 – Diagrama de sequência de Abastecimento

Conforme a figura 4, o processo de Abastecimento inicia com o motorista que utiliza o veículo, consulta a necessidade de abastecimento, caso não precise o processo finaliza, caso precisa abastecer, ao mesmo tempo que o posto irá realizar o abastecimento, o motorista irá realizar o pagamento, o TEF irá validar o pagamento, caso reprovado o processo finaliza, caso aprovado o veículo está abastecido e o motorista informará o valor, quantidade e tipo de combustível abastecido, com isso, o processo é finalizado por completo.

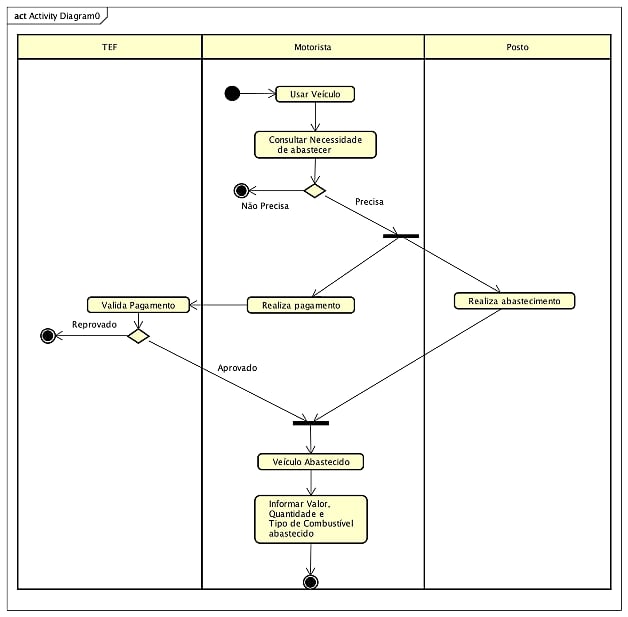


Figura 4 – Diagrama de Atividades de abastecimento.

Conforme a Figura 5, o processo é iniciado com a utilização de um veículo e para que seja realizado é necessário que possua as informações sobre o veículo e as informações sobre o combustível, sua saída sempre será o veículo abastecido e sua busca é a possibilidade de cadastrar o abastecimento no sistema através do celular onde será realizado um cálculo sobre a economia do veículo.

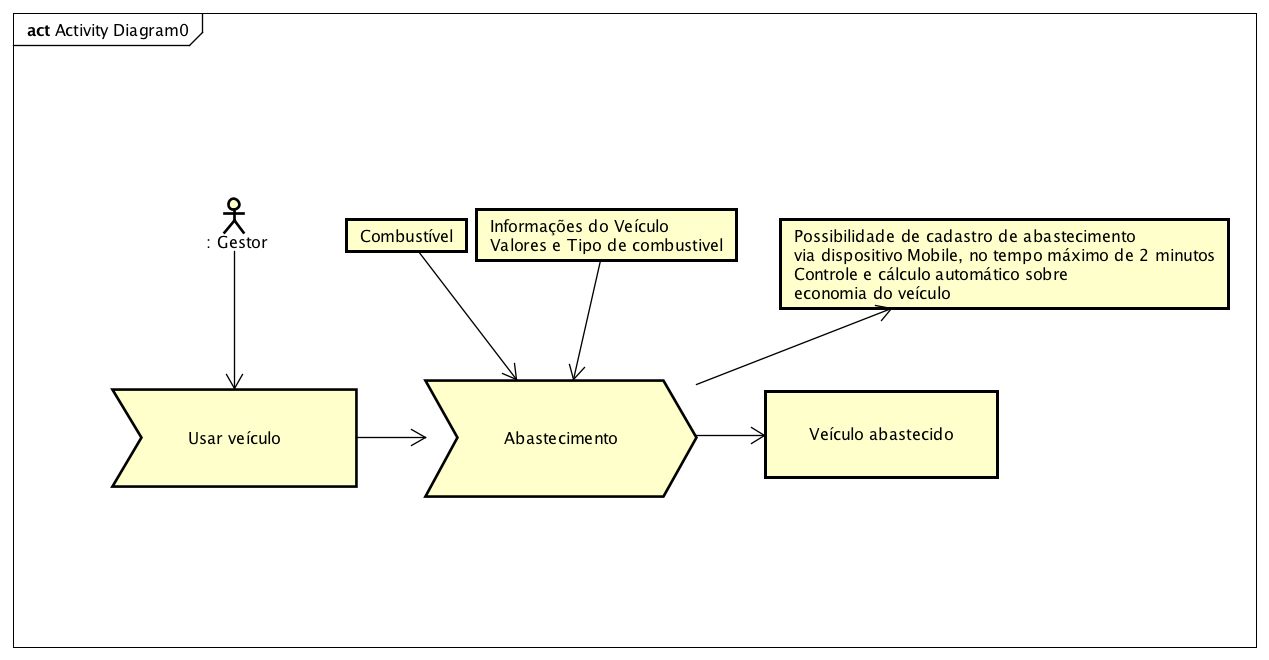


Figura 5 – Diagrama de Processos de Abastecimento.

Feitos os diagramas de abastecimentos o próximo que foi concluído foi o de controle de Peças/Pneus (Figuras 6 e 7) , o processo se inicia com a solicitação de compra de uma peça ou a necessidade de manutenção de um veículo, para que o processo ocorra são necessárias informações referentes às peças, pneus, fluido e as informações sobre o veículo, estoque e valores. Durante o processo sempre é buscada a atualização e abastecimento do estoque, buscando assim a excelência no controle do estoque e tendo como saída a notificação de uma nova requisição de compra.

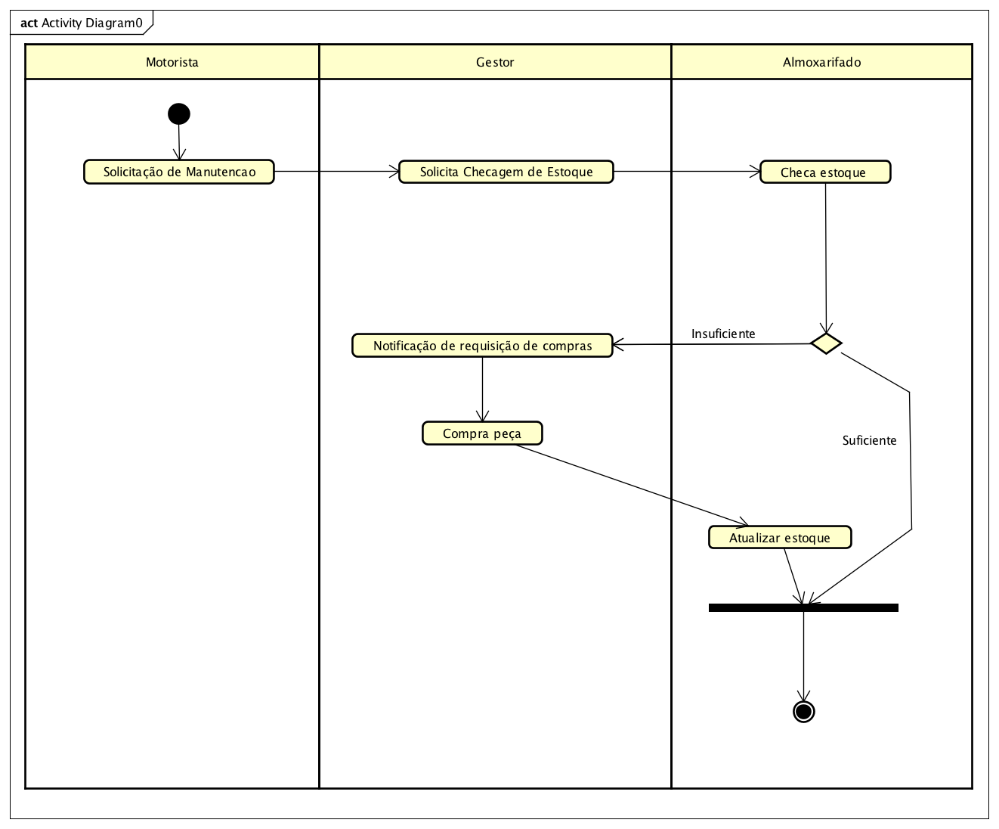


Figura 6 – Diagrama de Atividades sobre Controle de Peças/Pneus

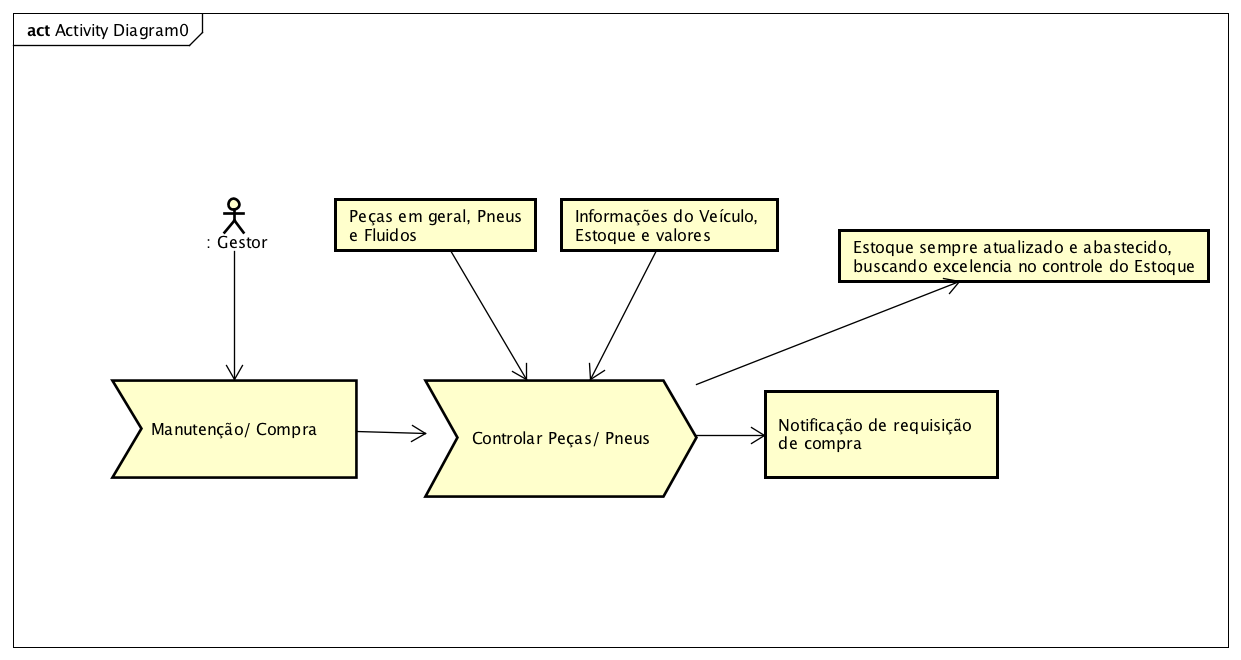


Figura 7 – Diagrama de processos sobre Controle de Peças e Pneus.

Os próximos foram os diagramas de Seguros e Boletins de Ocorrência (B.O.), o processo se inicia com a inserção de um sinistro ou um acidente e para que seja possível a realização do B.O. são necessárias as informações do veículo, do seguro e todas as outras informações que possam ser utilizadas, feito o B.O. as informações do cadastro do veículo são atualizadas e é iniciado o processo de notificação da seguradora. A busca no processo é sempre facilitar e aumentar a velocidade e eficácia da consulta e realização do BO, as Figuras 8, 9 e 10 descrevem os processos onde será realizado o registro e levantamento do B.O.

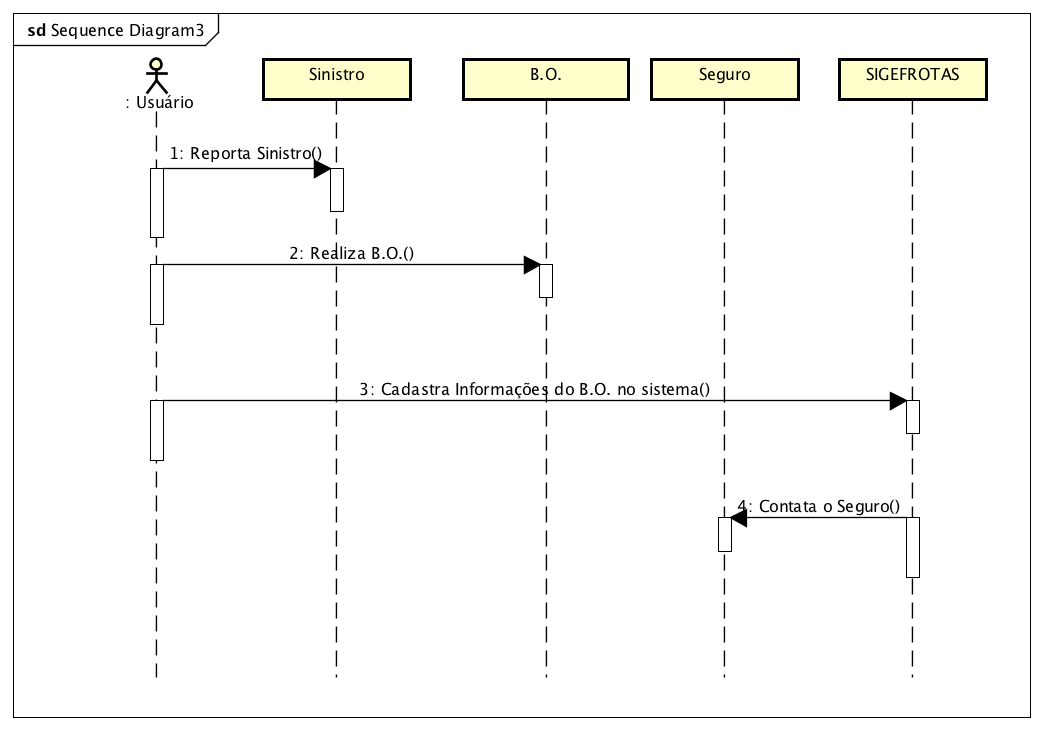


Figura 8 – Diagrama de Sequência de Seguros e Boletins de Ocorrência.

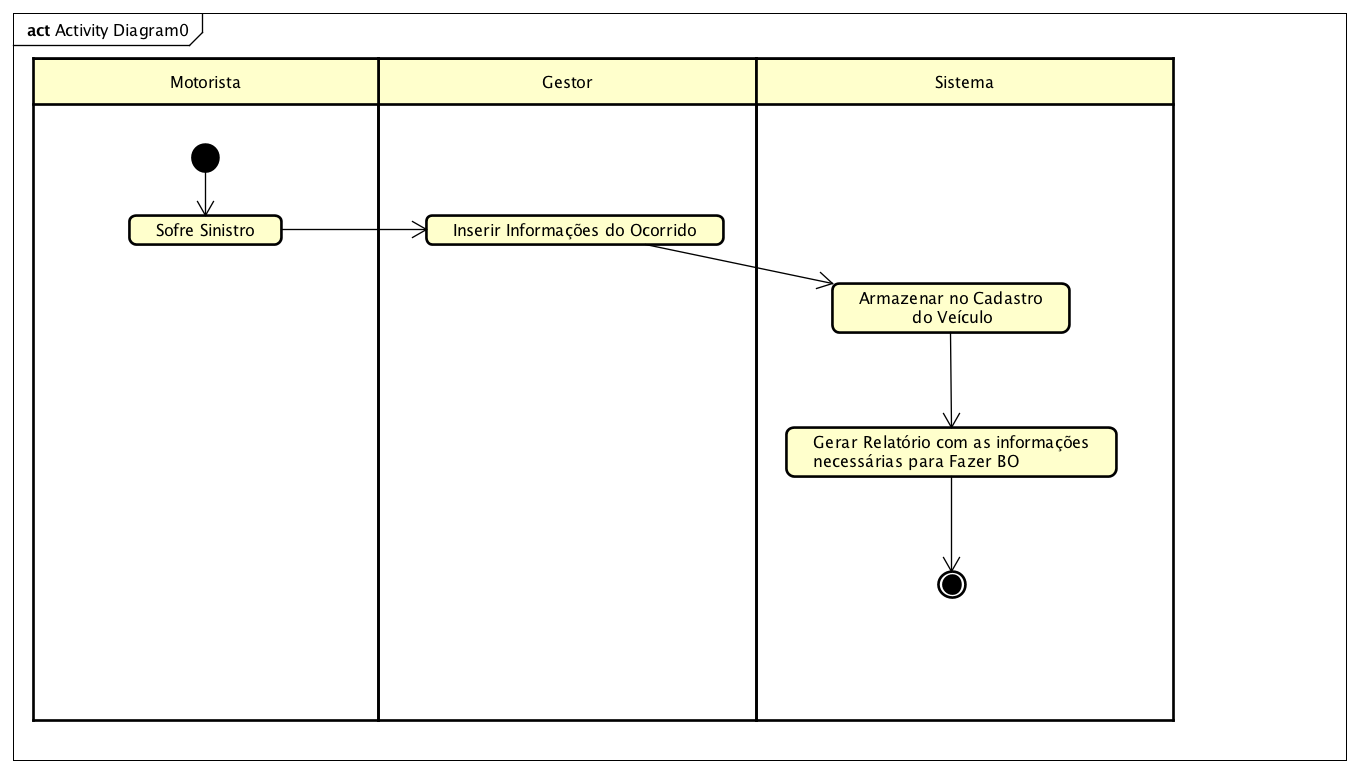


Figura 9 – Diagrama de Atividades sobre Gerar Relatórios de B.O.

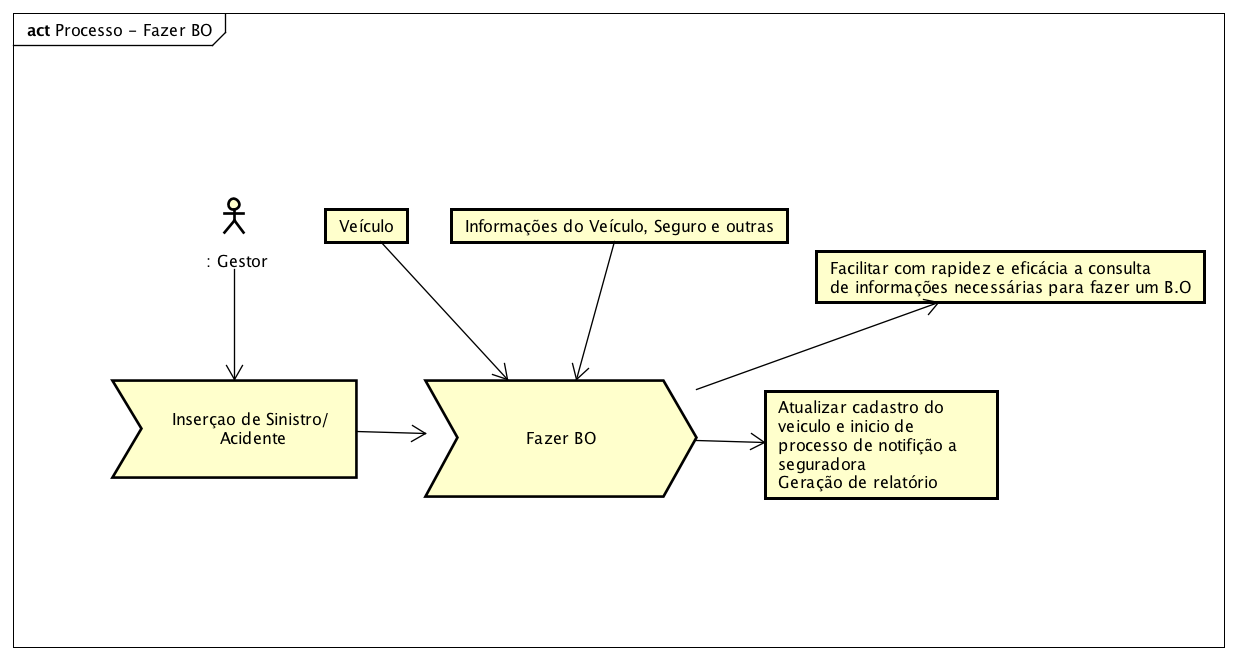


Figura 10 – Diagrama de Processos sobre Gerar Relatórios de B.O.

Um dos processos mais importantes é o de Controle de Viagens, é nele onde a maior proposito do sistema acontece, O processo se inicia com o cliente solicitando viagem, devendo ser informados os dados da viagem, como motorista, veículo, local de partida e de destino, distância e o tipo de viagem a ser realizada, como chamado técnico, manutenções internas etc. Assim que a solicitação de viagem é realizada, é emitido um recibo da viagem com os dados para controle e o sistema e banco de dados são alimentados, gerando dados e relatórios. A meta com a gerência das viagens é aumentar a eficiência, acurácia e rastreabilidade das informações de viagem, aumentando o potencial de crescimento da empresa gerenciada com relatórios detalhados dos serviços realizados, as figuras 11, 12 e 13 demonstram os diagramas de como esse processo tão importante funcionará.

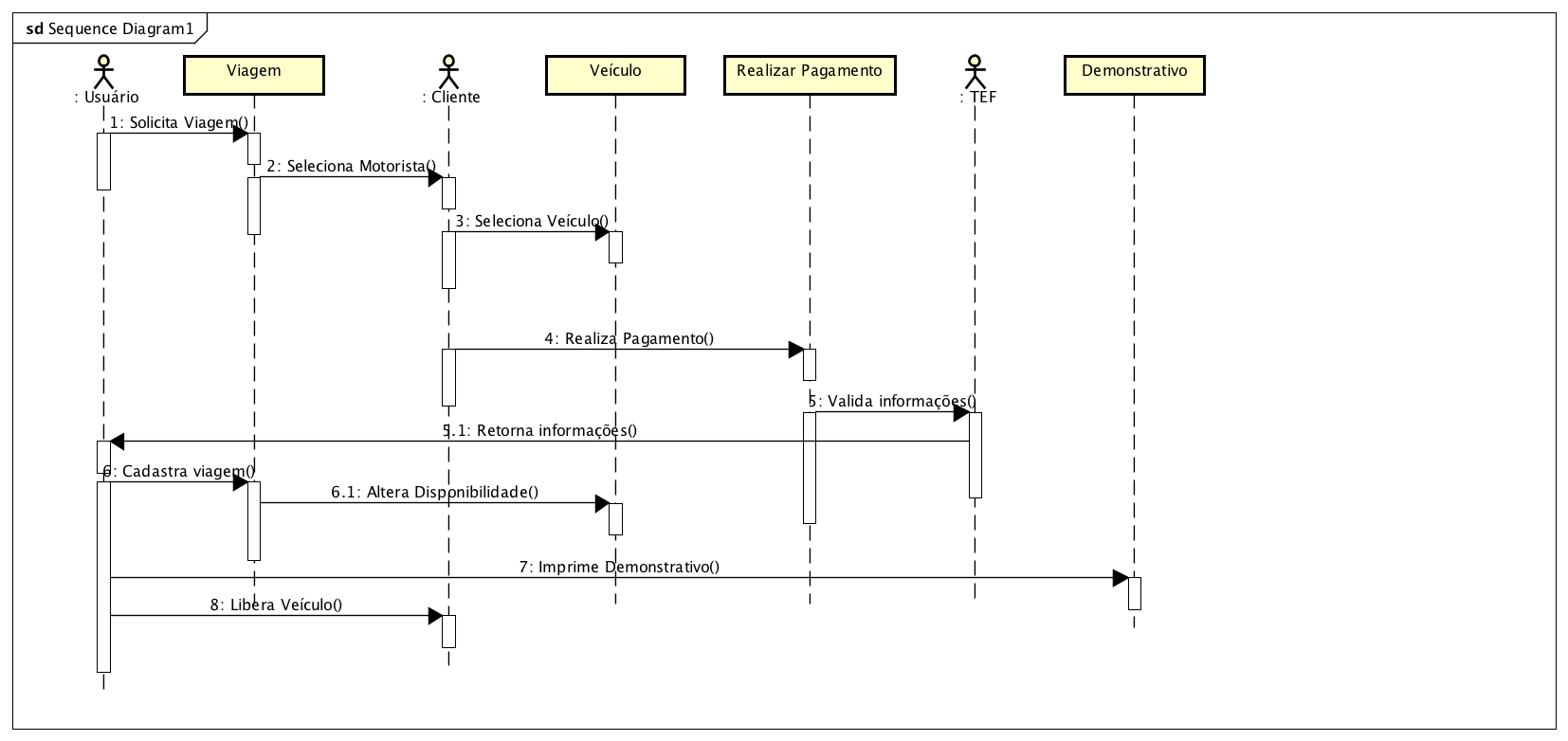


Figura 11 – Diagrama de Sequência sobre Gerência de Viagens

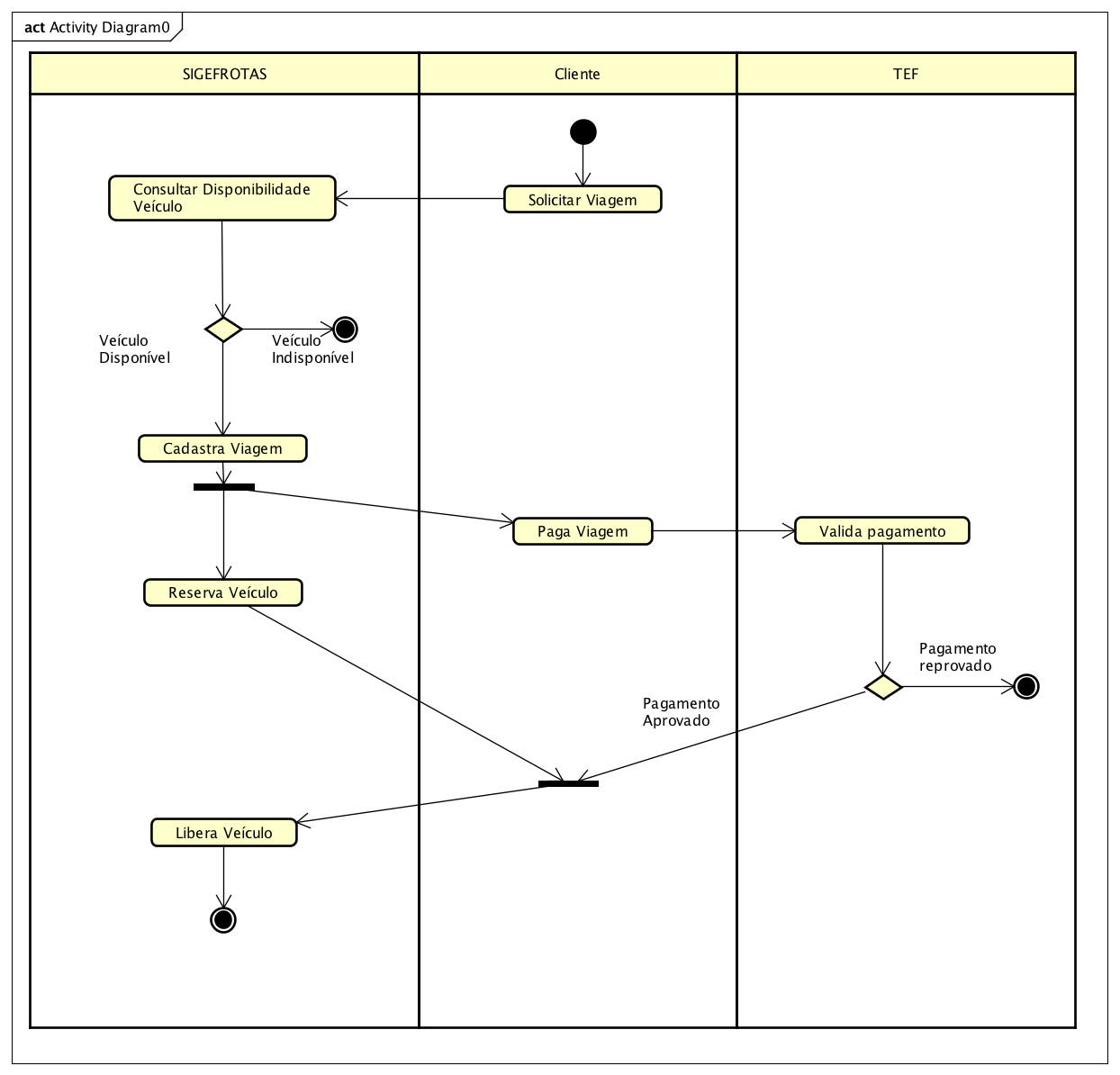
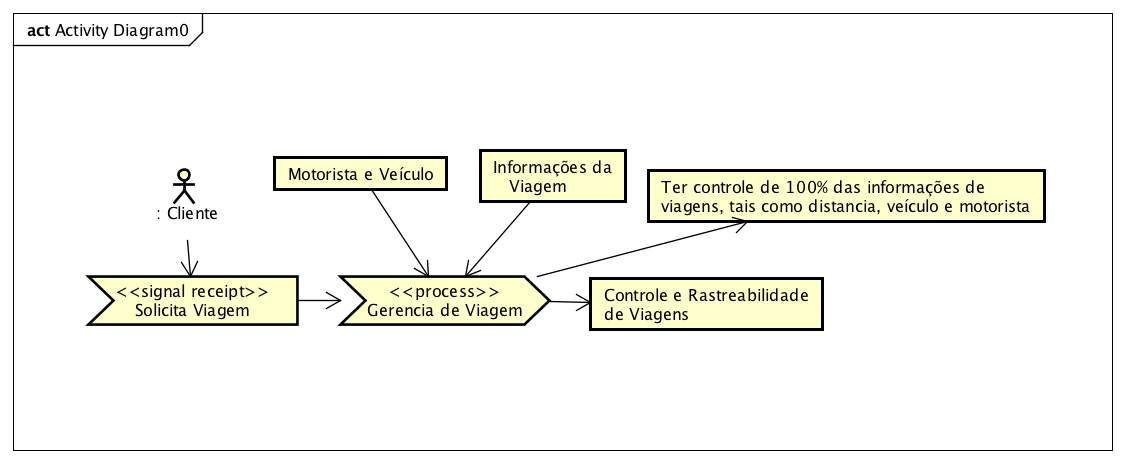


Figura 12 – Diagrama de Atividades sobre Gerência de Viagens

Figura 13 – Diagrama de Processos sobre Gerenciamento de Viagens

Os diagramas de Manutenção corretiva (Figura 16) e Manutenção preventiva (Figura 17) iniciam da mesma forma, com o ator “motorista” utilizando o veículo. A manutenção preventiva é essencial para que a corretiva não seja necessária, apenas em casos excepcionais e de emergência. Para ambos, é necessário inserir informações referentes ao veículo, como quilometragem, a oficina a ser utilizada, o tempo previsto de manutenção, as peças necessárias para serem substituídas e os custos gerados. A meta prevista é reduzir a utilização da manutenção corretiva em milésimos, garantindo a segurança do motorista, conservação das frotas e redução nos custos da empresa, aumentando a previsão de lucros, como o começo de ambas as manutenções é o mesmo as Figuras 14 e 15 servem para os dois.

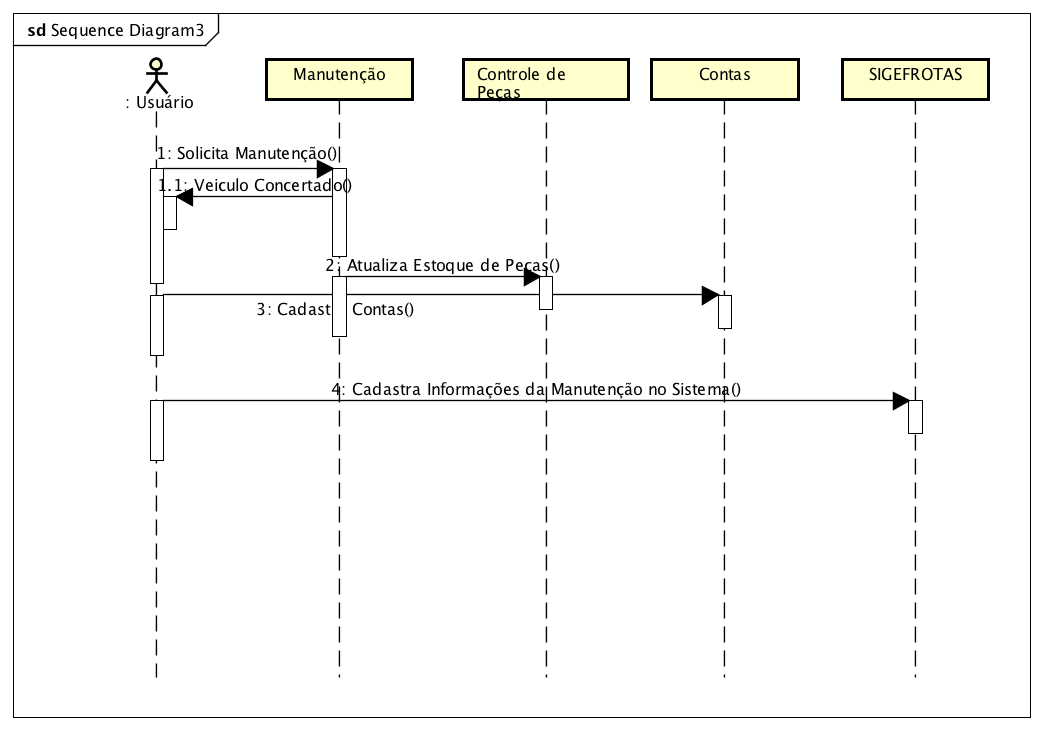


Figura 14 – Diagrama de Sequência de Manutenção

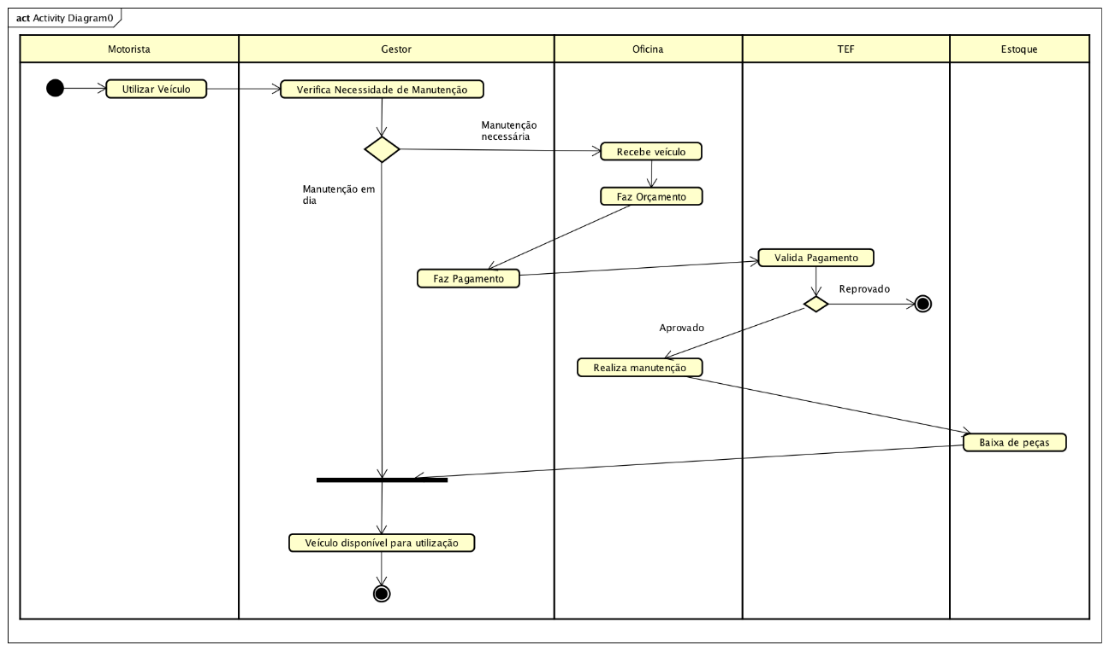


Figura 15 – Diagrama de Atividades de Manutenção.

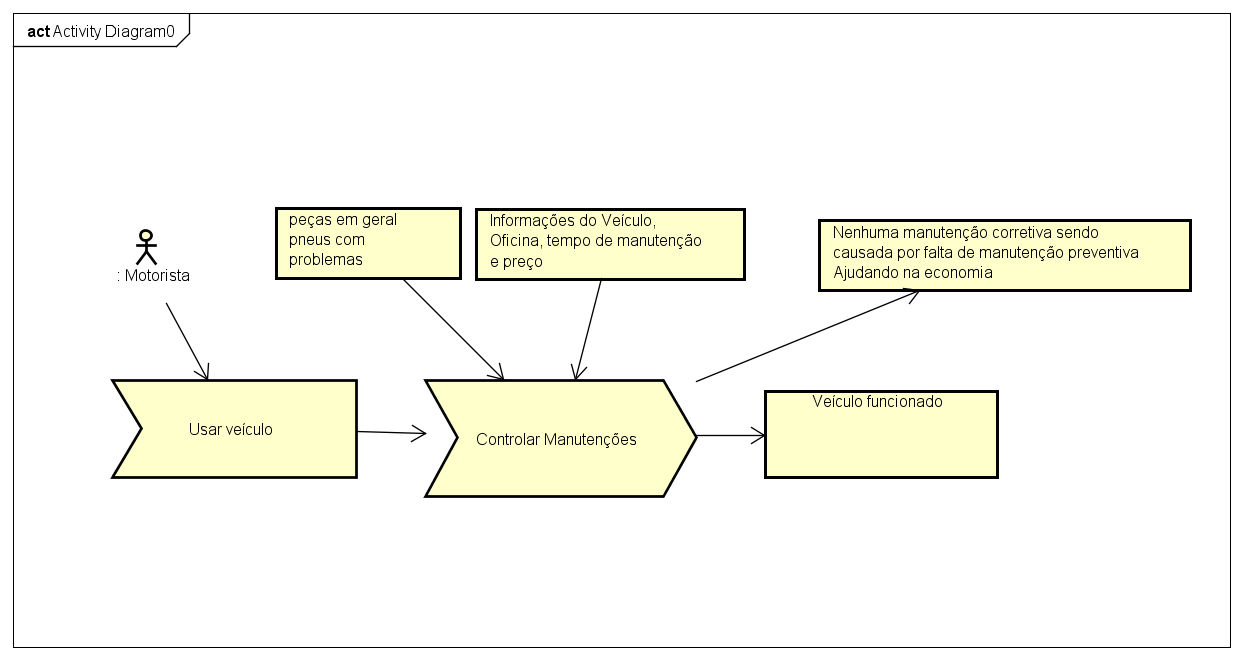


Figura 16 – Diagrama de Processos de Manutenção Corretiva.

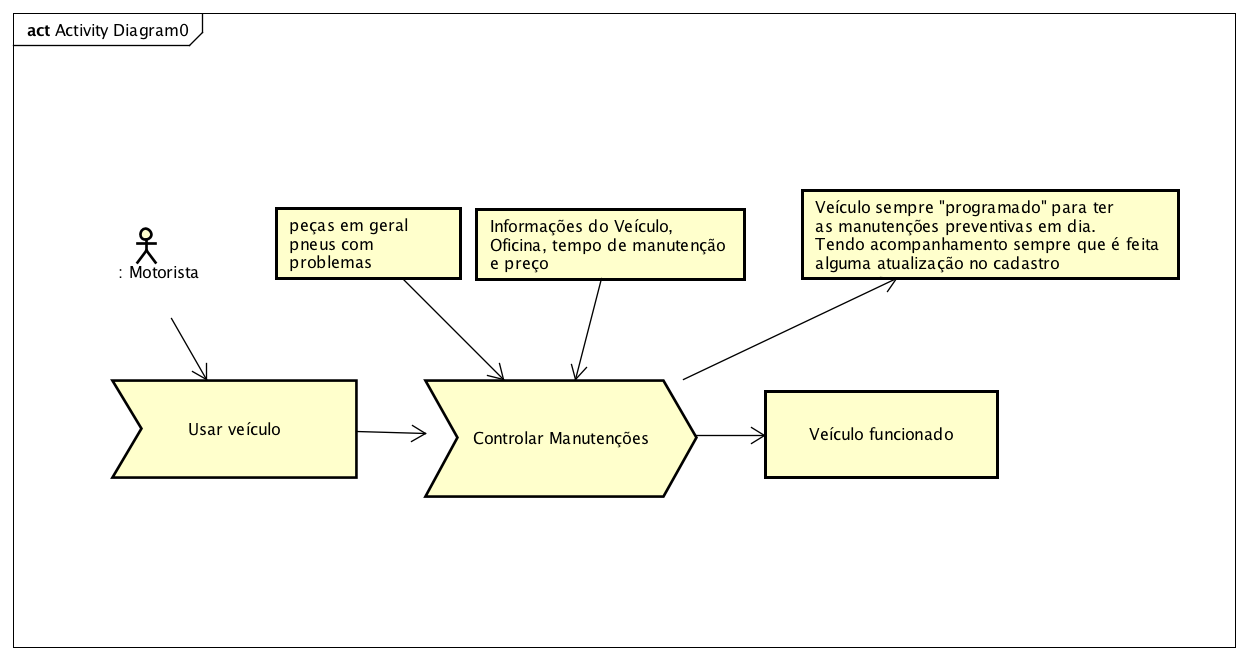


Figura 17 – Diagrama de Processos de Manutenção Preventiva.

Os diagramas de multas e sinistros (Figuras 18 e 19) apresenta a solução para facilitação dos serviços de controle de veículos, aumentando a rastreabilidade sobre as ações dos motoristas com relação a multas e auxílio em gerenciar os sinistros gerados por acidentes ou possível furto/roubo. O sistema é iniciado ao realizar uma viagem, onde caso haja alguma ocorrência, o motorista deverá relatar em sistema qual o tipo de situação. A saída final é o início do procedimento de realização do Boletim de Ocorrência e a atualização do cadastro das multas e sinistros, gerando modificações financeiras e de relatórios gerenciais, como os mesmos não tem uma ação direta de usuários não foi feito um diagrama de Sequência.

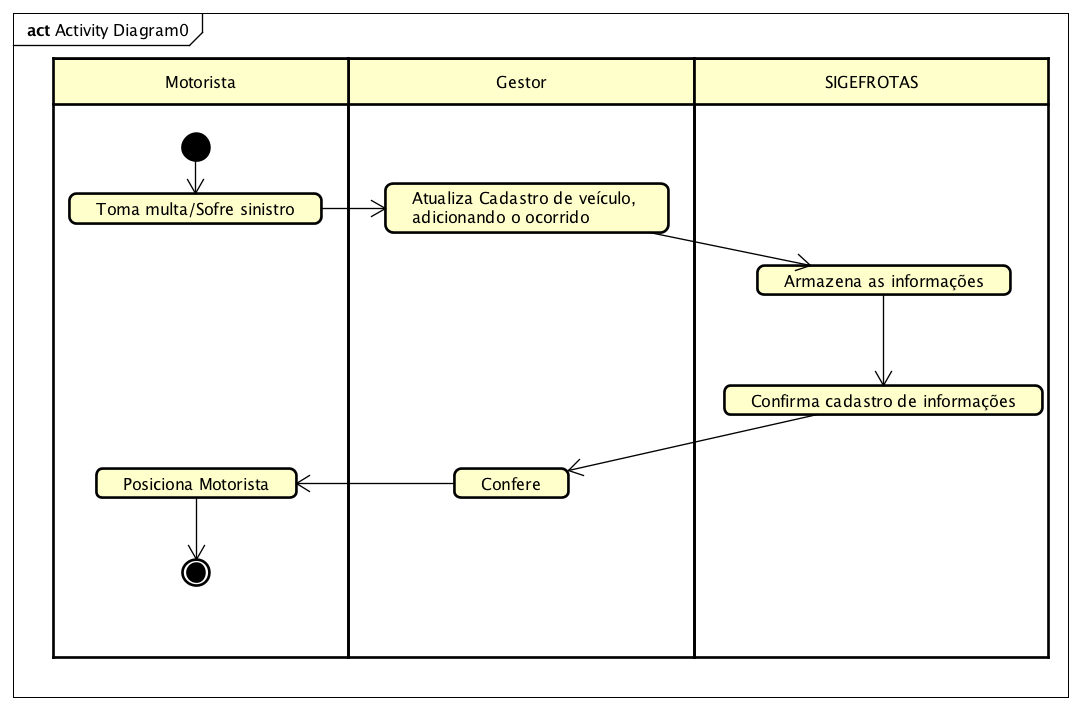


Figura 18 – Diagrama de Atividades sobre Controle de Multas e Sinistros.

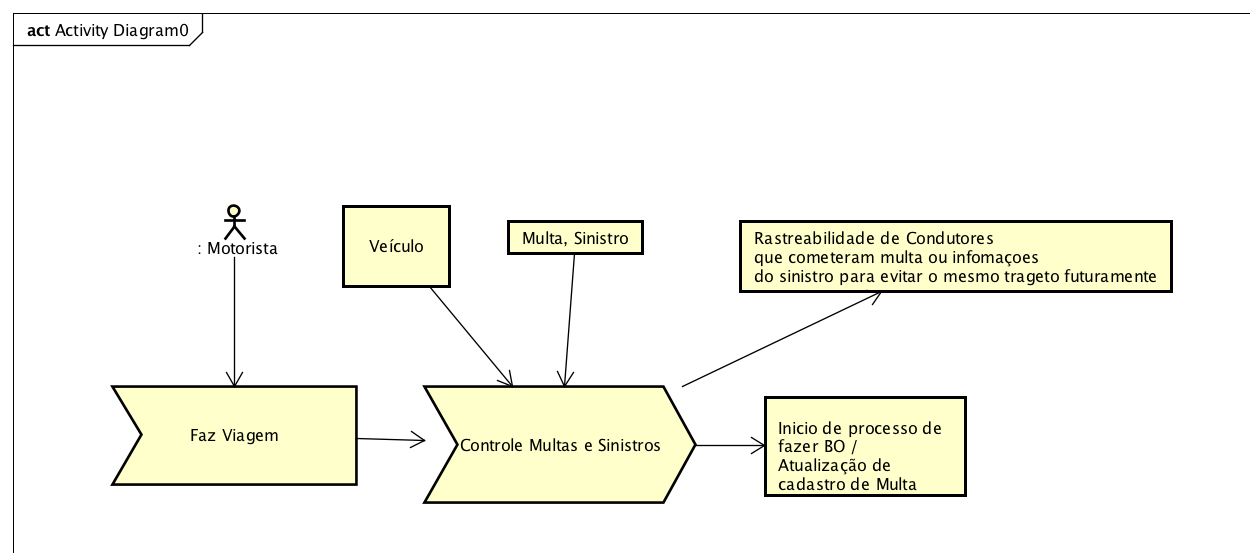


Figura 19 – Diagrama de Processos sobre Multas e Sinistros

Os diagramas abaixo (Figuras 20 e 21) representam a notificação de sinistros e acidentes, ativados pela saída do diagrama anterior. Esta atividade se baseia em realizar a notificação à seguradora, adicionando o boletim de ocorrência gerado, as informações do veículo e qual a seguradora cadastrada na apólice. A saída dessa atividade consiste em automatizar a notificação para a seguradora, atualização dos cadastros de veículos, motorista e sinistro, e gerar relatórios administrativos. Nossa meta é automatizar 100% o sistema de gerenciamento de sinistros, retirando a necessidade de burocracias com relação a solicitações às seguradoras.

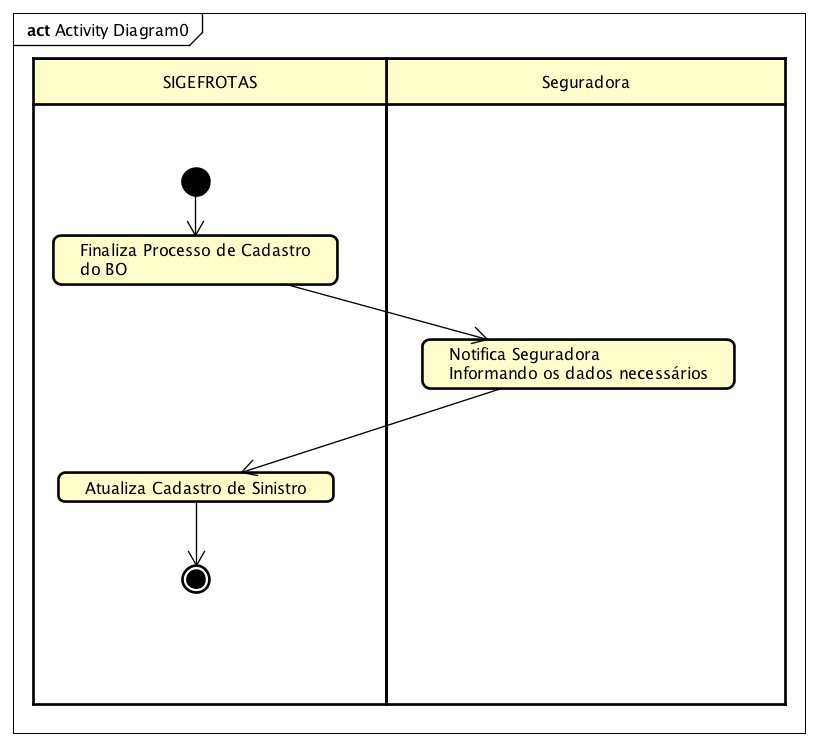


Figura 20 – Diagrama de Atividades sobre Notificação de Seguros

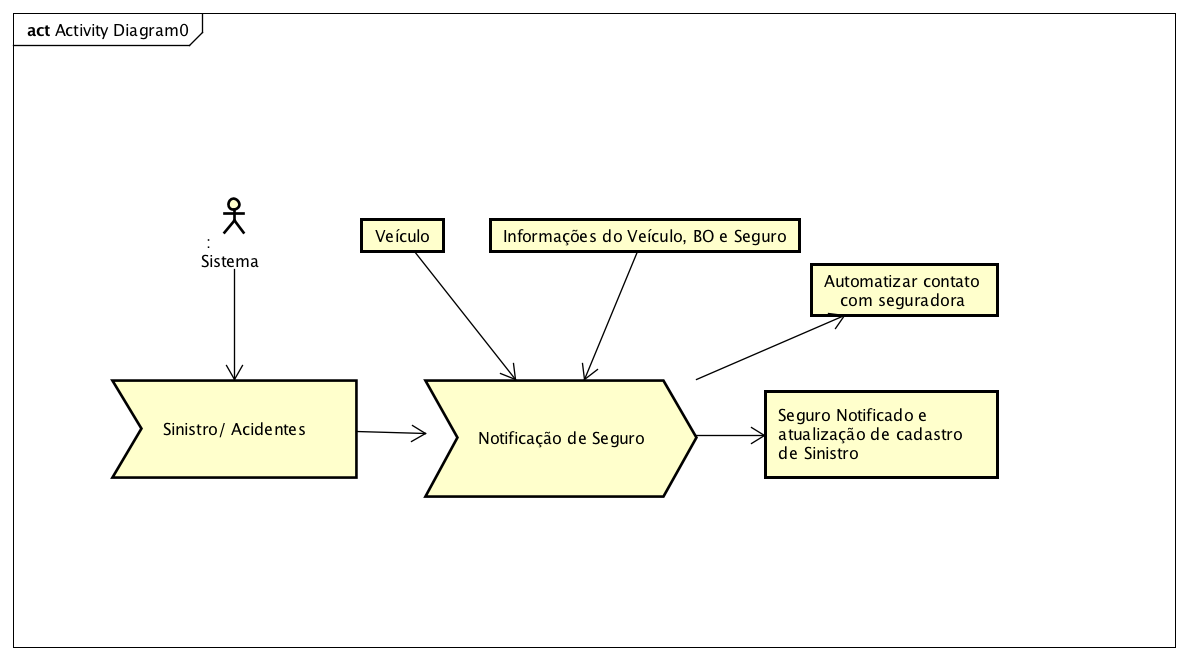


Figura 21 – Diagrama de Processos sobre Notificação de Seguros.

Estes diagramas (Figuras 22 e 23), representam a contratação do seguro dos veículos, sendo ativado o trigger assim que um veículo for adquirido, sendo requisitado a contratação do seguro. As informações necessárias para a contratação são todos os dados do veículo, como renovação, placa, laudos e documentos. A saída do diagrama é o veículo com o seguro ativo e número de apólice cadastrado no sistema, pronto para ser conduzido por um motorista cadastrado. A meta é sempre manter o gestor notificado quanto a vigência do contrato, reduzindo os riscos e aumentando as chances de renovação antecipada com descontos, gerando aumento do caixa da empresa.

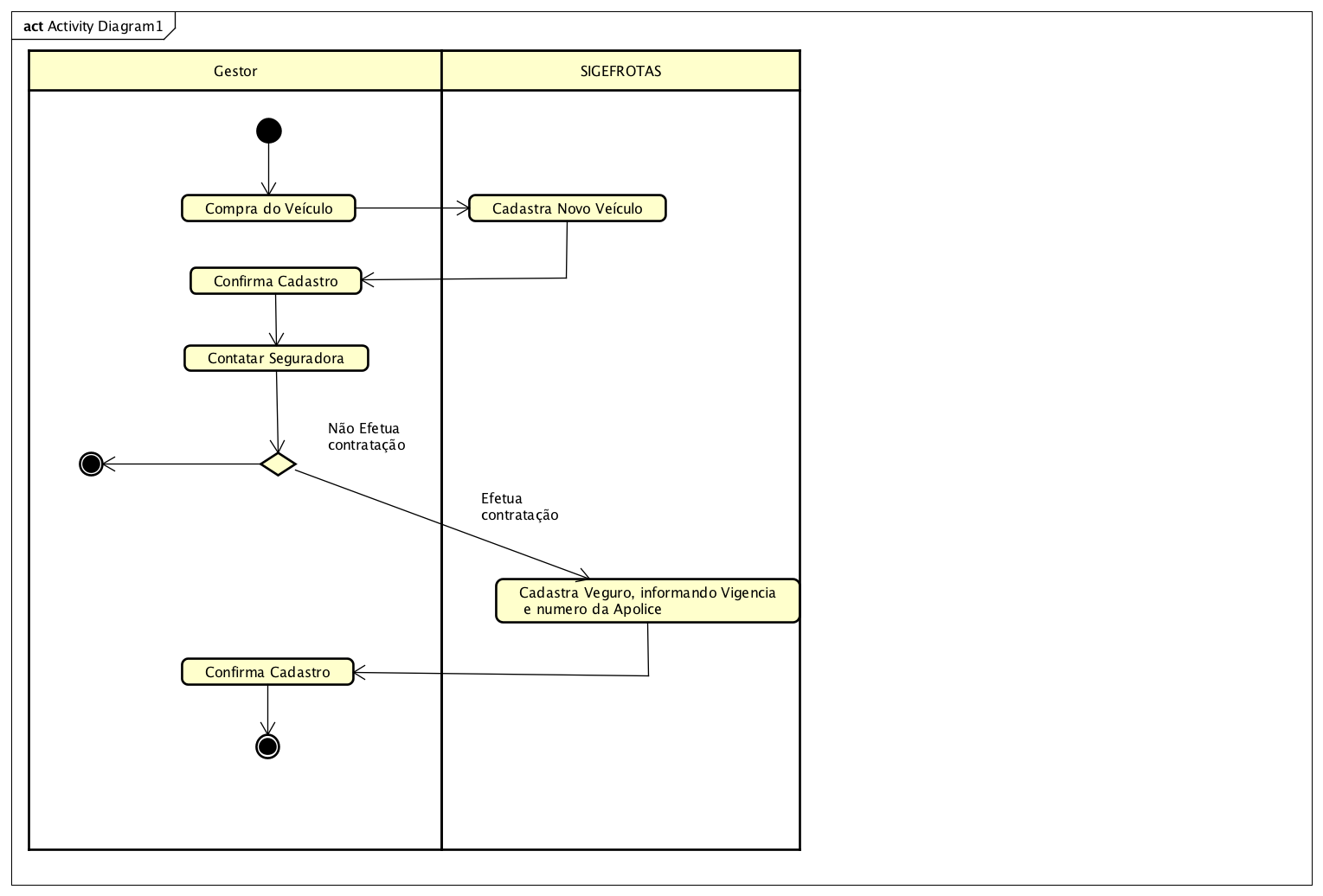


Figura 22 – Diagrama de Atividades sobre Contratação de Seguros

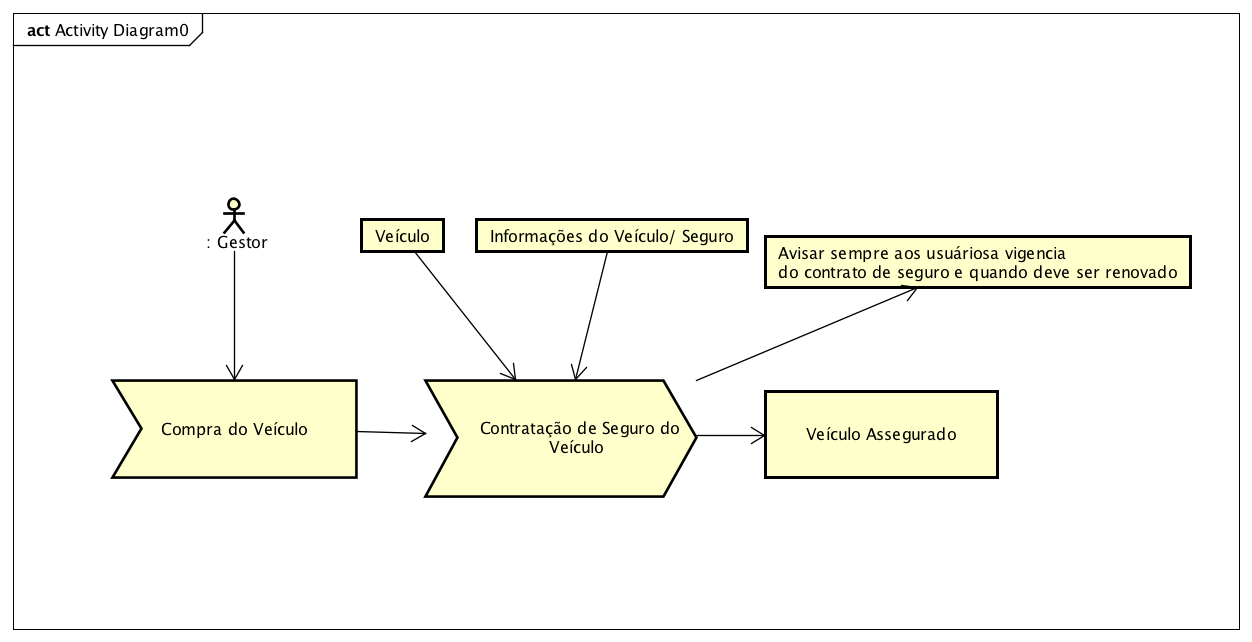


Figura 23 – Diagramas de Processos sobre Contratação de Seguros

Com tudo do sistema já esquematizado, é possível definir os estados de cada função principal, para poder definir se um veículo, ou um motorista está disponível para um serviço ou caso o cliente queira checar a situação de uma viagem, é sempre bom definir um estado que demonstrem o andamento desses objetos, isso é definido por um diagrama de estado, que situa o usuário sobre a situação de processo, por exemplo a figura 24 define o os possíveis estados de um carro:

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 24 – Diagrama de Estado do Veículo

Depois do estado do veículo, um outro ponto muito importa é saber o estado do motorista que irá realizar a viagem, como demonstra a figura 25:

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 25 – Diagrama de Estado do Motorista

Com os estados do veículo e do motorista definido é possível iniciar uma viagem, que será o trajeto realizado pelo motorista no veículo para ir realizar uma determinada tarefa, o diagrama de estado da viagem é bem simples como demonstra a figura 26:

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 26 – Diagrama de Estado da Viagem

# Banco de Dados

Para a criação do Banco de Dados, utilizamos a seguinte Query:

USE MASTER;

CREATE DATABASE GER\_FROTAS;

GO

USE GER\_FROTAS;

GO

CREATE TABLE MOTORISTA

( CD\_MOTORISTA INT NOT NULL,

NM\_MOTORISTA CHAR (80),

DT\_NASC\_MOTORISTA DATETIME,

CPF\_MOTORISTA CHAR(11),

RG\_MOTORISTA CHAR(10),

CNH\_MOTORISTA CHAR (12) NOT NULL,

VAL\_CNH\_MOTORISTA DATETIME NOT NULL,

OBS\_MOTORISTA varchar(120),

PRIMARY KEY (CD\_MOTORISTA))

GO

CREATE TABLE VEICULO

( CD\_VEICULO INT NOT NULL,

MARCA\_VEICULO varchar (15),

COR\_VEICULO varchar (20),

ANO\_VEICULO int,

MODELO\_VEICULO int,

TIPO\_VEICULO varchar (20),

HODOM\_VEICULO NUMERIC(8),

PLACA\_VEICULO varchar (8),

STATUS\_VEICULO int,

MANUTENCAO\_VEICULO int,

OBS\_VEICULO varchar (120),

SEGURO\_VEICULO int,

PRIMARY KEY (CD\_VEICULO))

GO

CREATE TABLE SEGURO

( CD\_SEGURO INT NOT NULL,

CD\_VEICULO\_SEGURO INT NOT NULL,

VALOR\_SEGURO NUMERIC (7,2),

PRIMARY KEY (CD\_SEGURO),

CONSTRAINT VEICULO\_SEGURO FOREIGN KEY (CD\_VEICULO\_SEGURO) REFERENCES VEICULO (CD\_VEICULO))

GO

CREATE TABLE FORNECEDOR

( CD\_FORNEC int NOT NULL,

END\_FORNEC varchar (120),

OBS\_FORNEC varchar (120),

PRIMARY KEY (CD\_FORNEC))

GO

CREATE TABLE PECAS

( CD\_PECA int NOT NULL,

NOME\_PECA varchar (40),

QTD\_PECA int,

VALOR\_PECA NUMERIC (7,2),

CD\_FORNEC\_PECA INT NOT NULL,

PRIMARY KEY (CD\_PECA),

CONSTRAINT FORNEC\_PECA FOREIGN KEY (CD\_FORNEC\_PECA) REFERENCES FORNECEDOR (CD\_FORNEC))

GO

CREATE TABLE GER\_FROTA\_VIAGEM

( CD\_VIAGEM int NOT NULL,

ORIGEM\_VIAGEM varchar (150),

EST\_ORIGEM\_VIAGEM varchar (50),

DEST\_VIAGEM varchar (150),

EST\_DEST\_VIAGEM varchar (50),

KM\_VIAGEM NUMERIC (5),

TIPO\_VIAGEM varchar (50),

MOTIVO\_VIAGEM varchar (20),

VALOR\_VIAGEM NUMERIC (7,2),

OBS\_VIAGEM varchar (120),

CD\_MOTORISTA\_VIAGEM INT NOT NULL,

CD\_VEICULO\_VIAGEM INT NOT NULL,

PRIMARY KEY (CD\_VIAGEM),

CONSTRAINT MOTORISTA\_VIAGEM FOREIGN KEY (CD\_MOTORISTA\_VIAGEM) REFERENCES MOTORISTA (CD\_MOTORISTA),

CONSTRAINT VEICULO\_VIAGEM FOREIGN KEY (CD\_VEICULO\_VIAGEM) REFERENCES VEICULO (CD\_VEICULO))

GO

CREATE TABLE ALUG\_VEIC\_VIAGEM

( CD\_VIAGEM int NOT NULL,

KM\_VIAGEM NUMERIC (5),

MOTIVO\_VIAGEM varchar (20),

VALOR\_VIAGEM NUMERIC (7,2),

OBS\_VIAGEM varchar (120),

CD\_MOTORISTA\_VIAGEM INT NOT NULL,

CD\_VEICULO\_VIAGEM INT NOT NULL,

PRIMARY KEY (CD\_VIAGEM),

CONSTRAINT MOTORISTA\_VIAGEM\_ALUG FOREIGN KEY (CD\_MOTORISTA\_VIAGEM) REFERENCES MOTORISTA (CD\_MOTORISTA),

CONSTRAINT VEICULO\_VIAGEM\_ALUG FOREIGN KEY (CD\_VEICULO\_VIAGEM) REFERENCES VEICULO (CD\_VEICULO))

GO

CREATE TABLE GER\_FROTA\_ABASTE

( CD\_ABASTE INT NOT NULL,

CD\_VEICULO\_ABASTE INT NOT NULL,

VALOR\_ABASTE NUMERIC (7,2),

TIPO\_COMBUSTIVEL varchar (40),

PRIMARY KEY (CD\_ABASTE),

CONSTRAINT VEICULO\_ABASTE FOREIGN KEY (CD\_VEICULO\_ABASTE) REFERENCES VEICULO (CD\_VEICULO))

GO

CREATE TABLE GER\_FROTA\_BO

( CD\_BO INT NOT NULL,

CD\_VEICULO\_BO INT NOT NULL,

CD\_MOTORISTA\_BO INT NOT NULL,

EST\_BO varchar (50),

CIDADE\_BO varchar (50),

PRIMARY KEY (CD\_BO),

CONSTRAINT VEICULO\_BO FOREIGN KEY (CD\_VEICULO\_BO) REFERENCES VEICULO (CD\_VEICULO),

CONSTRAINT MOTORISTA\_BO FOREIGN KEY (CD\_MOTORISTA\_BO) REFERENCES MOTORISTA (CD\_MOTORISTA))

GO

CREATE TABLE ALUG\_VEIC\_BO

( CD\_BO INT NOT NULL,

CD\_VEICULO\_BO INT NOT NULL,

CD\_MOTORISTA\_BO INT NOT NULL,

EST\_BO varchar (50),

CIDADE\_BO varchar (50),

PRIMARY KEY (CD\_BO),

CONSTRAINT VEICULO\_BO\_ALUG FOREIGN KEY (CD\_VEICULO\_BO) REFERENCES VEICULO (CD\_VEICULO),

CONSTRAINT MOTORISTA\_BO\_ALUG FOREIGN KEY (CD\_MOTORISTA\_BO) REFERENCES MOTORISTA (CD\_MOTORISTA))

GO

CREATE TABLE GER\_FROTA\_MANUTE

( CD\_MANUTE INT NOT NULL,

CD\_VEICULO\_MANUTE INT NOT NULL,

TP\_MANUTE varchar (30),

VALOR\_MANUTE NUMERIC (7,2),

PRIMARY KEY (CD\_MANUTE),

CONSTRAINT VEICULO\_MANUTE FOREIGN KEY (CD\_VEICULO\_MANUTE) REFERENCES VEICULO (CD\_VEICULO))

GO

CREATE TABLE ALUG\_VEIC\_MANUTE

( CD\_MANUTE INT NOT NULL,

CD\_VEICULO\_MANUTE INT NOT NULL,

TP\_MANUTE varchar (30),

VALOR\_MANUTE NUMERIC (7,2),

PRIMARY KEY (CD\_MANUTE),

CONSTRAINT VEICULO\_MANUTE\_ALUG FOREIGN KEY (CD\_VEICULO\_MANUTE) REFERENCES VEICULO (CD\_VEICULO))

GO

CREATE TABLE GER\_FROTA\_MULTAS

( CD\_MULTA INT NOT NULL,

CD\_VEICULO\_MULTA INT NOT NULL,

CD\_MOTORISTA\_MULTA INT NOT NULL,

DT\_MULTA datetime,

LOCAL\_MULTA varchar (120),

VALOR\_MULTA NUMERIC (7,2),

PRIMARY KEY (CD\_MULTA),

CONSTRAINT VEICULO\_MULTA FOREIGN KEY (CD\_VEICULO\_MULTA) REFERENCES VEICULO (CD\_VEICULO),

CONSTRAINT MOTORISTA\_MULTA FOREIGN KEY (CD\_MOTORISTA\_MULTA) REFERENCES MOTORISTA (CD\_MOTORISTA))

GO

CREATE TABLE ALUG\_VEIC\_MULTAS

( CD\_MULTA INT NOT NULL,

CD\_VEICULO\_MULTA INT NOT NULL,

CD\_MOTORISTA\_MULTA INT NOT NULL,

DT\_MULTA datetime,

LOCAL\_MULTA varchar (120),

VALOR\_MULTA NUMERIC (7,2),

PRIMARY KEY (CD\_MULTA),

CONSTRAINT VEICULO\_MULTA\_ALUG FOREIGN KEY (CD\_VEICULO\_MULTA) REFERENCES VEICULO (CD\_VEICULO),

CONSTRAINT MOTORISTA\_MULTA\_ALUG FOREIGN KEY (CD\_MOTORISTA\_MULTA) REFERENCES MOTORISTA (CD\_MOTORISTA))

GO

CREATE TABLE GER\_FROTA\_FINANCAS

( CD\_FINANCA INT NOT NULL,

VALOR\_MULTA NUMERIC (7,2) NOT NULL,

VALOR\_MANUTE NUMERIC (7,2) NOT NULL,

VALOR\_VIAGEM NUMERIC (7,2) NOT NULL,

CD\_SEGURO\_FINANCA INT NOT NULL,

CD\_PECA\_FINANCA INT NOT NULL,

CD\_VIAGEM\_FINANCA INT NOT NULL,

CD\_ABASTE\_FINANCA INT NOT NULL,

CD\_MANUTE\_FINANCA INT NOT NULL,

CD\_MULTA\_FINANCA INT NOT NULL,

PRIMARY KEY (CD\_FINANCA),

CONSTRAINT VLR\_SEG FOREIGN KEY (CD\_SEGURO\_FINANCA) REFERENCES SEGURO (CD\_SEGURO),

CONSTRAINT VLR\_PECA FOREIGN KEY (CD\_PECA\_FINANCA) REFERENCES PECAS (CD\_PECA),

CONSTRAINT VLR\_VIAGEM FOREIGN KEY (CD\_VIAGEM\_FINANCA) REFERENCES GER\_FROTA\_VIAGEM (CD\_VIAGEM),

CONSTRAINT VLR\_MANUT FOREIGN KEY (CD\_MANUTE\_FINANCA) REFERENCES GER\_FROTA\_MANUTE(CD\_MANUTE),

CONSTRAINT VLR\_MULTA FOREIGN KEY (CD\_MULTA\_FINANCA) REFERENCES GER\_FROTA\_MULTAS (CD\_MULTA),

CONSTRAINT VLR\_ABASTE FOREIGN KEY (CD\_ABASTE\_FINANCA) REFERENCES GER\_FROTA\_ABASTE(CD\_ABASTE))

GO

CREATE TABLE ALUG\_VEIC\_FINANCAS

( CD\_FINANCA INT NOT NULL,

VALOR\_MULTA NUMERIC (7,2) NOT NULL,

VALOR\_MANUTE NUMERIC (7,2) NOT NULL,

VALOR\_VIAGEM NUMERIC (7,2) NOT NULL,

CD\_SEGURO\_FINANCA INT NOT NULL,

CD\_PECA\_FINANCA INT NOT NULL,

CD\_VIAGEM\_FINANCA INT NOT NULL,

CD\_MANUTE\_FINANCA INT NOT NULL,

CD\_MULTA\_FINANCA INT NOT NULL,

PRIMARY KEY (CD\_FINANCA),

CONSTRAINT VLR\_SEG\_ALUG FOREIGN KEY (CD\_SEGURO\_FINANCA) REFERENCES SEGURO (CD\_SEGURO),

CONSTRAINT VLR\_PECA\_ALUG FOREIGN KEY (CD\_PECA\_FINANCA) REFERENCES PECAS (CD\_PECA),

CONSTRAINT VLR\_VIAGEM\_ALUG FOREIGN KEY (CD\_VIAGEM\_FINANCA) REFERENCES ALUG\_VEIC\_VIAGEM(CD\_VIAGEM),

CONSTRAINT VLR\_MULTA\_ALUG FOREIGN KEY (CD\_MULTA\_FINANCA) REFERENCES ALUG\_VEIC\_MULTAS (CD\_MULTA),

CONSTRAINT VLR\_MANUT\_ALUG FOREIGN KEY (CD\_MANUTE\_FINANCA) REFERENCES GER\_FROTA\_MANUTE(CD\_MANUTE))

GO

CREATE TABLE GER\_FROTA\_SINISTRO

( CD\_SINISTRO INT NOT NULL,

CD\_VEICULO\_SINISTRO INT NOT NULL,

CD\_MOTORISTA\_SINISTRO INT NOT NULL,

CD\_SEGURO\_SINISTRO INT NOT NULL,

PRIMARY KEY (CD\_SINISTRO),

CONSTRAINT VEICULO\_SINISTRO FOREIGN KEY (CD\_VEICULO\_SINISTRO) REFERENCES VEICULO (CD\_VEICULO),

CONSTRAINT MOTORISTA\_SINISTRO FOREIGN KEY (CD\_MOTORISTA\_SINISTRO) REFERENCES MOTORISTA (CD\_MOTORISTA),

CONSTRAINT SEGURO\_SINISTRO FOREIGN KEY (CD\_SEGURO\_SINISTRO) REFERENCES SEGURO (CD\_SEGURO))

GO

CREATE TABLE ALUG\_VEIC\_SINISTRO

( CD\_SINISTRO INT NOT NULL,

CD\_VEICULO\_SINISTRO INT NOT NULL,

CD\_MOTORISTA\_SINISTRO INT NOT NULL,

CD\_SEGURO\_SINISTRO INT NOT NULL,

PRIMARY KEY (CD\_SINISTRO),

CONSTRAINT VEICULO\_SINISTRO\_ALUG FOREIGN KEY (CD\_VEICULO\_SINISTRO) REFERENCES VEICULO (CD\_VEICULO),

CONSTRAINT MOTORISTA\_SINISTRO\_ALUG FOREIGN KEY (CD\_MOTORISTA\_SINISTRO) REFERENCES MOTORISTA (CD\_MOTORISTA),

CONSTRAINT SEGURO\_SINISTRO\_ALUG FOREIGN KEY (CD\_SEGURO\_SINISTRO) REFERENCES SEGURO (CD\_SEGURO))

GO

CREATE TABLE COMPRAS

( CD\_COMPRA INT NOT NULL,

DT\_COMPRA DATETIME,

NF\_COMPRA NUMERIC (30),

CHAVE\_ACESSO\_COMPRA VARCHAR (120),

VALOR\_COMPRA NUMERIC (7,2),

CD\_PECA\_COMPRA INT NOT NULL,

PRIMARY KEY (CD\_COMPRA),

CONSTRAINT COMPRA FOREIGN KEY (CD\_PECA\_COMPRA) REFERENCES PECAS (CD\_PECA))

GO

Com essa Query, foi criado o seguinte Diagrama Relacional (Figura 27):

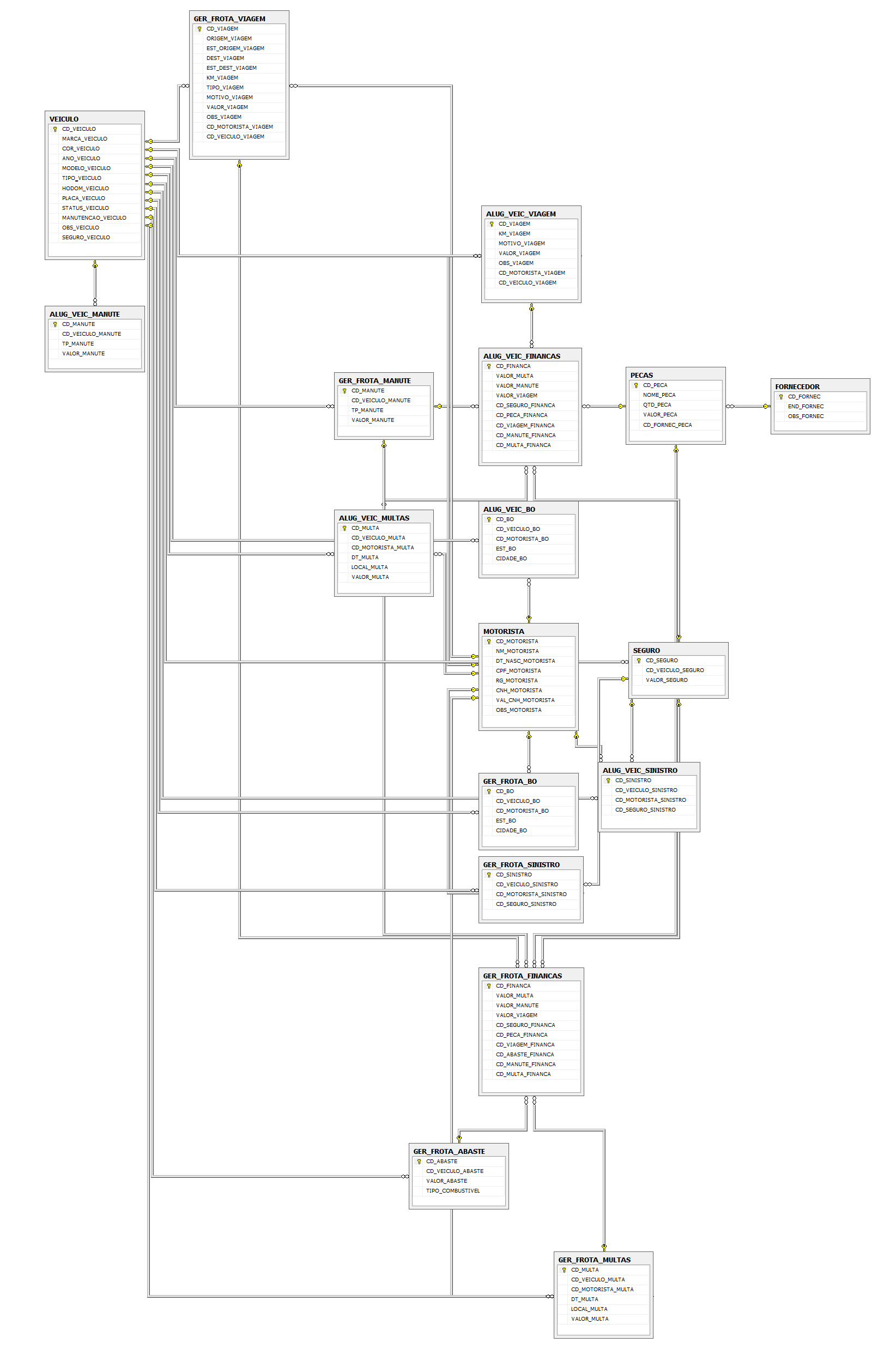


Figura 27 – Diagrama relacional do Banco de Dados

# Gerenciamento de Projetos.

A EAP permite uma estruturação de um projeto de forma simplificada e contém desde o começo até o final do trabalho necessário para conclusão do projeto, com o objetivo de identificar as partes que compõe o projeto. Possui algumas atividades principais como, facilitar identificação das fases do projeto, facilitar identificação dos responsáveis, identificar as atividades do projeto, facilitar identificação de riscos, entre outros.

Pode-se construir uma EAP de várias maneiras conforme o tipo do projeto, as formas mais comuns são, por Fases, por Entregas e por Equipes. Por fases temos uma "visão cronológica" dos acontecimentos no projeto, fácil entendimento para leigos e facilita o posterior gerenciamento das atividades. Por entregas mostra partes necessárias para compor a entrega do projeto, visualiza claramente as partes que compõe o projeto, facilita a discussão de soluções técnicas e caminhos alternativos e facilita a identificação de riscos técnicos. Por equipes podemos visualizar os pacotes de trabalho a partir da divisão de equipes do projeto, são boas para ocasiões em que o projeto tem equipes com responsabilidades muito diferentes. Sendo assim, o escolhido para gerenciar o projeto foi o de fases, pois por podermos ver uma "linha de vida" do projeto, torna mais viável e condizente, como mostra a Figura 28.

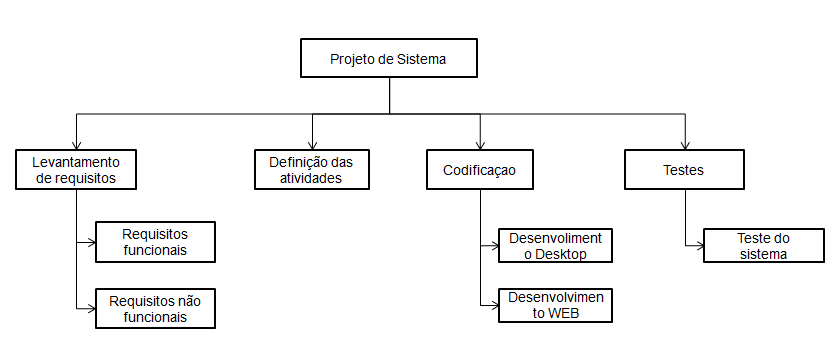


Figura 28 - EAP gerenciamento do projeto

## **6.1 Backup e Restauração do Sistema**

Backup ou cópia de segurança é um procedimento indispensável para qualquer rede/sistema, é a única forma de recuperar informações em caso de perca de informações, essa perda de dados pode causar prejuízos inimagináveis, tanto financeiro quando no relacionamento com os clientes. Então, para o procedimento de backup existem 3 tipos dos quais poderíamos utilizar, sendo o completo, incremental e o diferencial.

O completo usa-se quando necessita de uma cópia completa de todos os arquivos, por conta disso ele é o mais demorado dentre eles e o que ocupa mais espaço.

Já o diferencial faz cópia apenas das modificações divergentes ao último backup completo, fazendo assim, sua velocidade de backup ser superior e utiliza menos espaço em disco do que o backup completo.

O incremental faz copias das alterações relativas ao último backup, é o mais rápido entre os três e o que requer menor espaço de armazenamento.

Seguindo as características acima, será feito o backup completo no primeiro dia para armazenar todas as informações que já foram obtidas, após isso, será feito todo dia um backup incremental as 00:00 e uma vez na semana será feito um backup diferencial, serão feitas duas copias desse backup, uma armazenada na nuvem e a outra em um servidor físico.

# Testes do Sistema

Todos os sistemas depois ou durante o desenvolvimento precisa ser testado e aprovado para poder seguir para a próxima etapa, os testes são importantes para identificar Falhas no sistema, erros de programação, entrada de dados incorretos, todas as falhas que podem conter e que impeçam ou dificultem a utilização do e navegação do usuário.

Os testes passam por uma planilha contendo passo a passo de cada etapa, cada uma válida uma função do sistema, depois de tudo ser aprovado essa tabela é enviada para a equipe de desenvolvimento, caso sejam encontrados erros são feitos os reparos e reenviado para o pessoal de testes, caso o sistema passe em todas as etapas ele segue para a próxima etapa.

Para o Sigefrotas foi desenvolvida a seguinte tabela de testes para o sistema (figura 29):

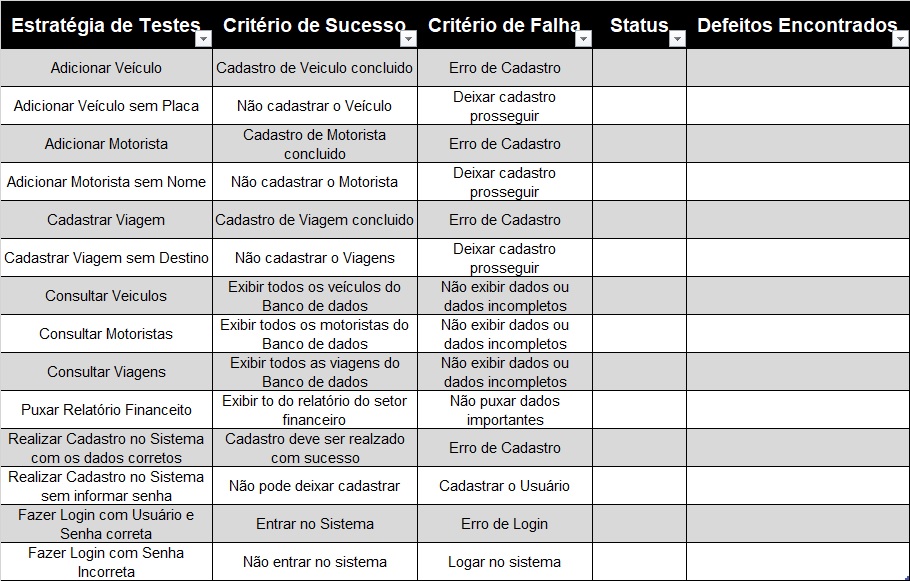


Figura 29 – Tabela de Testes do Sigefrotas.

Com base nessa tabela foram feitos os testes do Sigefrotas, e foi obtido o seguinte resultado (figura 30):

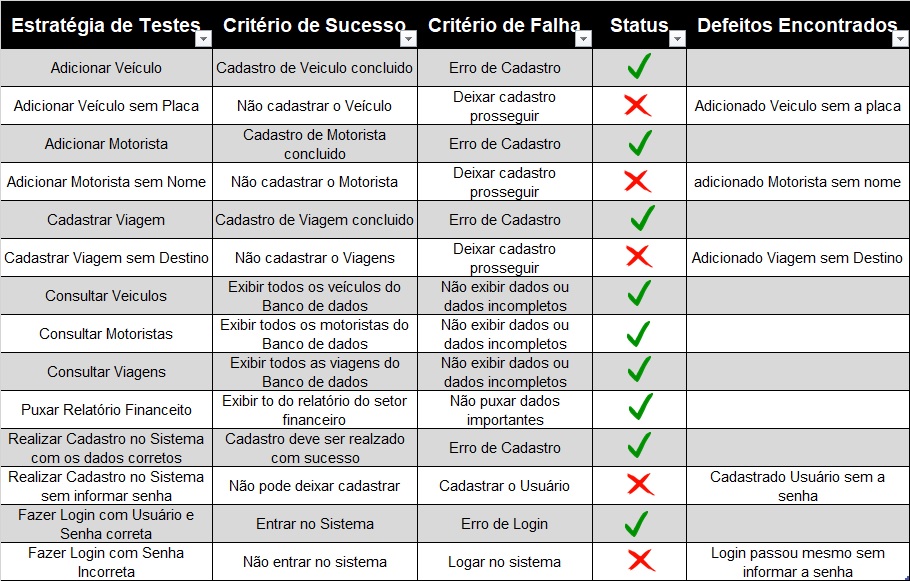


Figura 30 – Tabela de Testes do Sigefrotas com erros de sistema

Ao analisar o resultado foi possível identificar que os dados estavam sendo guardados com sucesso, porém ai deixar um campo vazio ou com algum caractere inválido o sistema não identificava o erro e seguia salvando os dados, esses dados foram repassados para a equipe de Desenvolvimento que começou a identificar o motivo desse erro acontecer, sendo que faltavam linhas de código que restringissem esses campos como obrigatórios.

Depois de resolver o problema a equipe de desenvolvimento enviou o sistema para realizar os testes mais uma vez, essa rotina tem que ser repetida sempre que for identificado um erro, ou até que passem em todos os testes e que nenhum problema seja encontrado, feitos os testes saíram o seguinte resultado (figura 31):

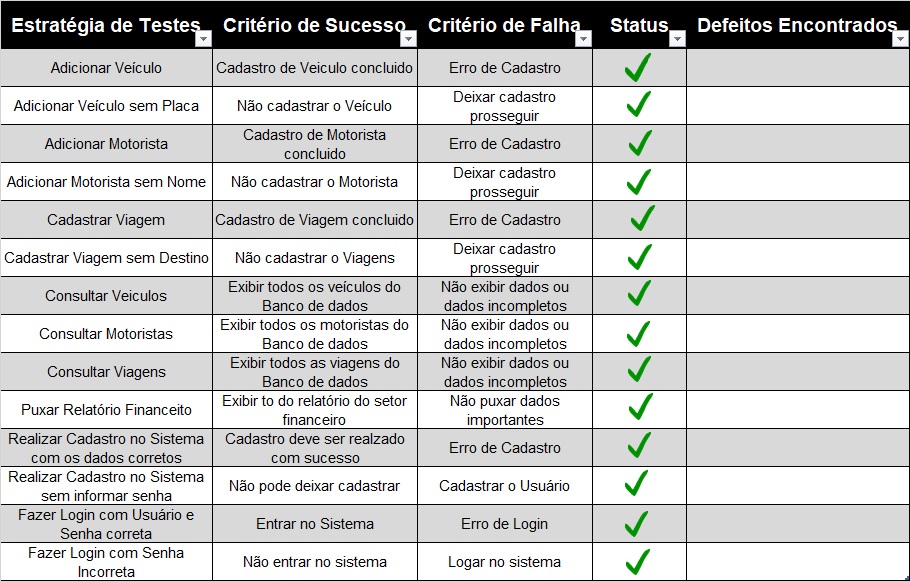


Figura 31 – tabela de Testes do Sigefrotas com erros corrigidos

Com a aprovação na parte dos testes, o sistema está um passo mais próximo de ser concluído e ser entregue para o cliente.

# Manuais

## **8.1 Manual de Instalação**

Os Manuais são muito importantes para os usuários de um sistema, seja qual for, ensina-los desde como instalar o seu produto até como o utilizar é uma boa forma de não só garantir que o usuário utilize de forma correta e tirar maior proveito do sistema, como também garante que haja uma menor quantidade de erros do programas, por isso foram desenvolvidos dois manuais para auxiliar o usuário a ter acesso ao seu programa, o primeiro é o de instalação, ensinando como instalar o produto em seu equipamento caso não queira utilizar em seu navegador, e o segundo manual é o de utilização do equipamento.

**Manual de Instalação do Sistema SIGEFROTAS**

Para realizar a instalar o SIGEFROTAS em seus equipamentos siga os passos a seguir:

1. Após a compra de seu Sistema, você receberá no e-mail informado durante seu cadastro uma mensagem com os links para Download do Sigefrotas.

1. No e-mail clique agora em “Download Setup”, após isso será iniciado o download de um arquivo chamado “Sigefrotas\_Setup.exe”

1. Execute o arquivo e abrirá o Instalador do programa.

1. Logo na primeira tela ele irá solicitar que selecione se deseja instalar em 64 Bits, ou em 32 Bits, marque a opção que seja compatível com a arquitetura do seu sistema operacional.
2. Depois selecione o local onde será instalado os arquivos do programa, de padrão vem a opção “C:\Program Files\Sigefrotas” para 64 Bits e “C:\Program Files (x86)\Sigefrotas” para 32 Bits, mas é possível alterar para algum outro de sua preferência, ao terminar clique em “Avançar”.

1. Agora na tela de Termos e Uso, recomendamos que leia os Termos antes de aceitá-los, clique em “Concordo” e depois em “Avançar”
2. Após isso é só clicar em “Instalar” e aguardar o fim da instalação do seu Sistema.

1. Depois de terminado clique “Concluir” e o processo de instalação de seu sistema terá sido concluído com sucesso.

## **8.2 Manual do Sistema**

**Passo 1:** Acessar o sistema de frotas pelo computador.

**Passo 2:** Caso já seja cadastrado basta digitar nos campos designados Login e Senha e clicar em *Entrar*. Em caso de esquecimento clique na opção *Esqueci minha Senha.* Caso não seja cadastrado clique em *Criar Cadastro*, como visto na figura 32.

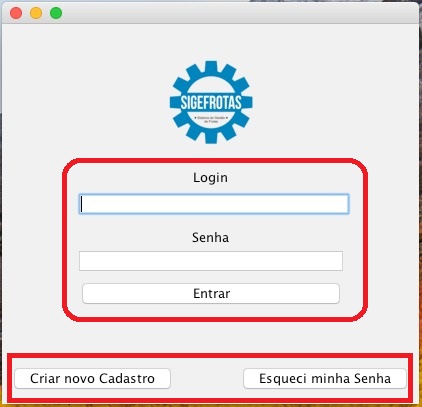


Figura 32 – Tela de Login do Sigefrotas

**Passo 3:** Caso não seja cadastrado faça o cadastro, digite Login, Senha, E-mail e Data de Nascimento, clique em *Cadastrar Novo Usuário.* Caso já possua cadastro clique em *Já Possui Cadastro? Clique aqui,* como visto na figura 33.

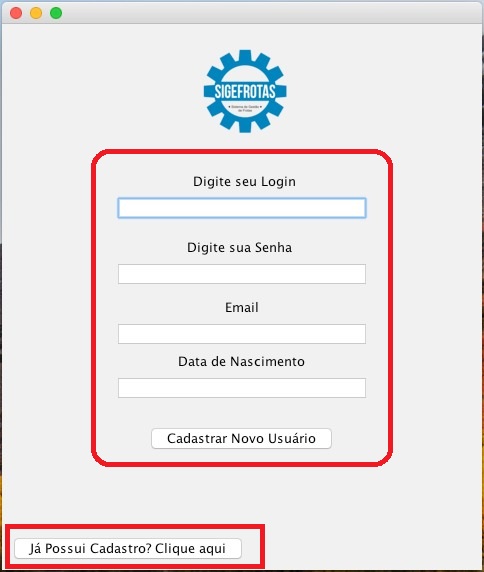


Figura 33 – Tela de Cadastro do Sigefrotas

**Passo 4:** Escolher opção desejada dentre as telas de *Cadastro e Consultas, Gerenciamento de Frotas, Locação de Veículos, Financeiro e Relatórios, Compras e Abastecimentos* e *Manutenção,* como visto na figura 34.

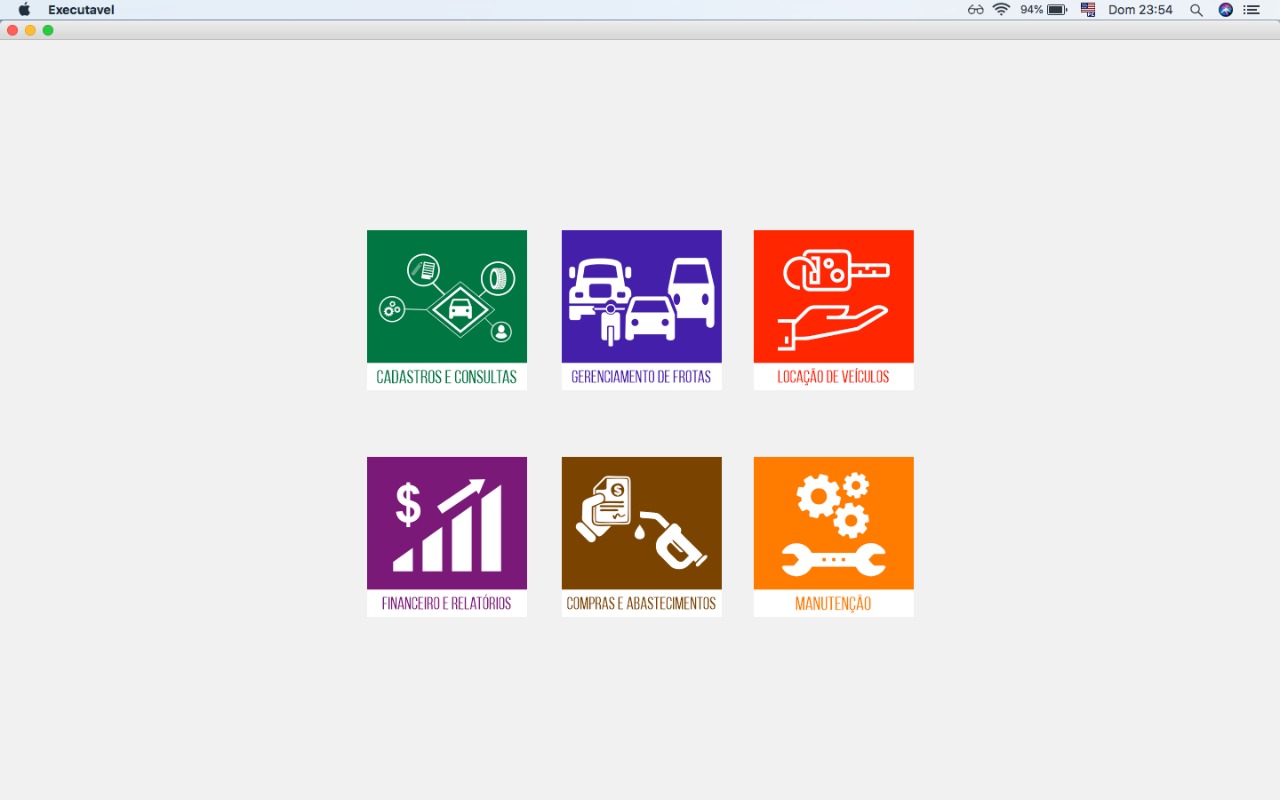


Figura 34 – Tela Inicial do Sigefrotas

Será exibido um menu com opções de cadastro e consulta, para realizar, basta clicar em *Incluir* ou *Consultar* respectivamente para cada ação desejada, como visto na figura 35.

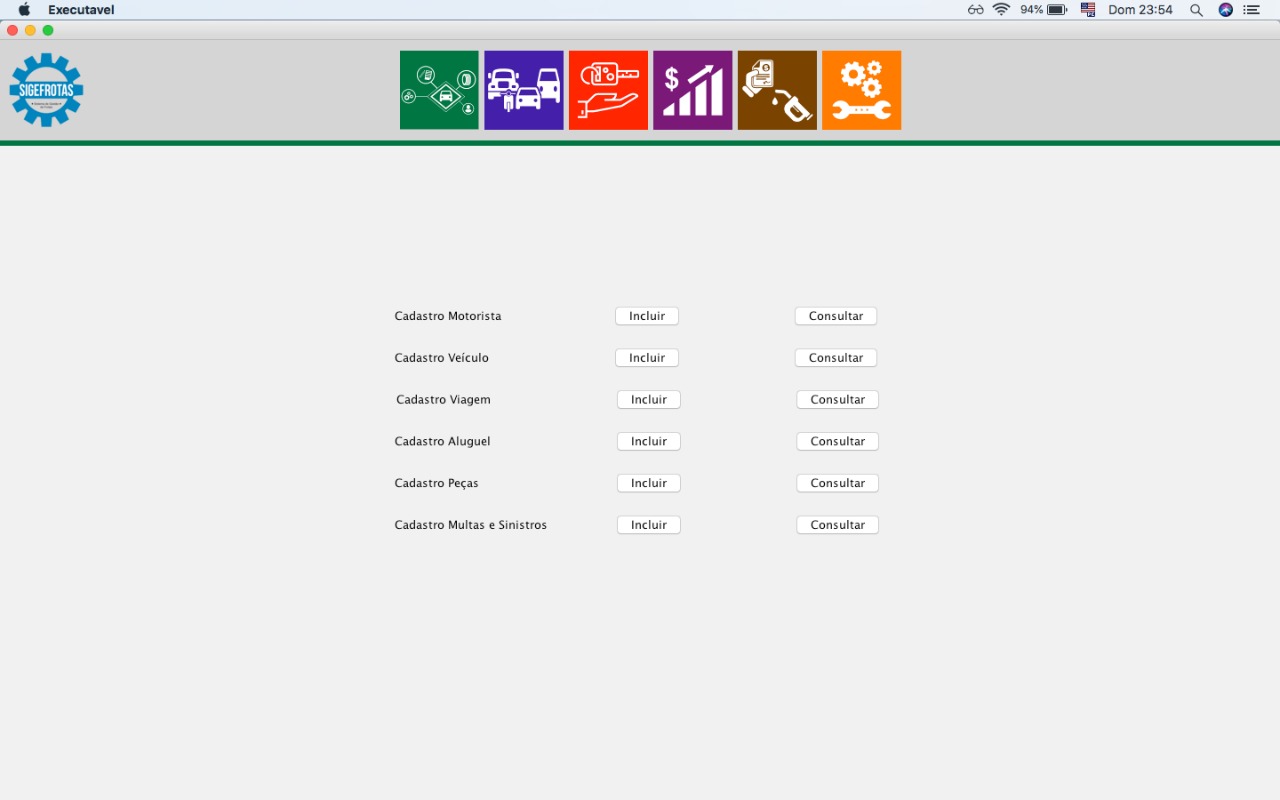
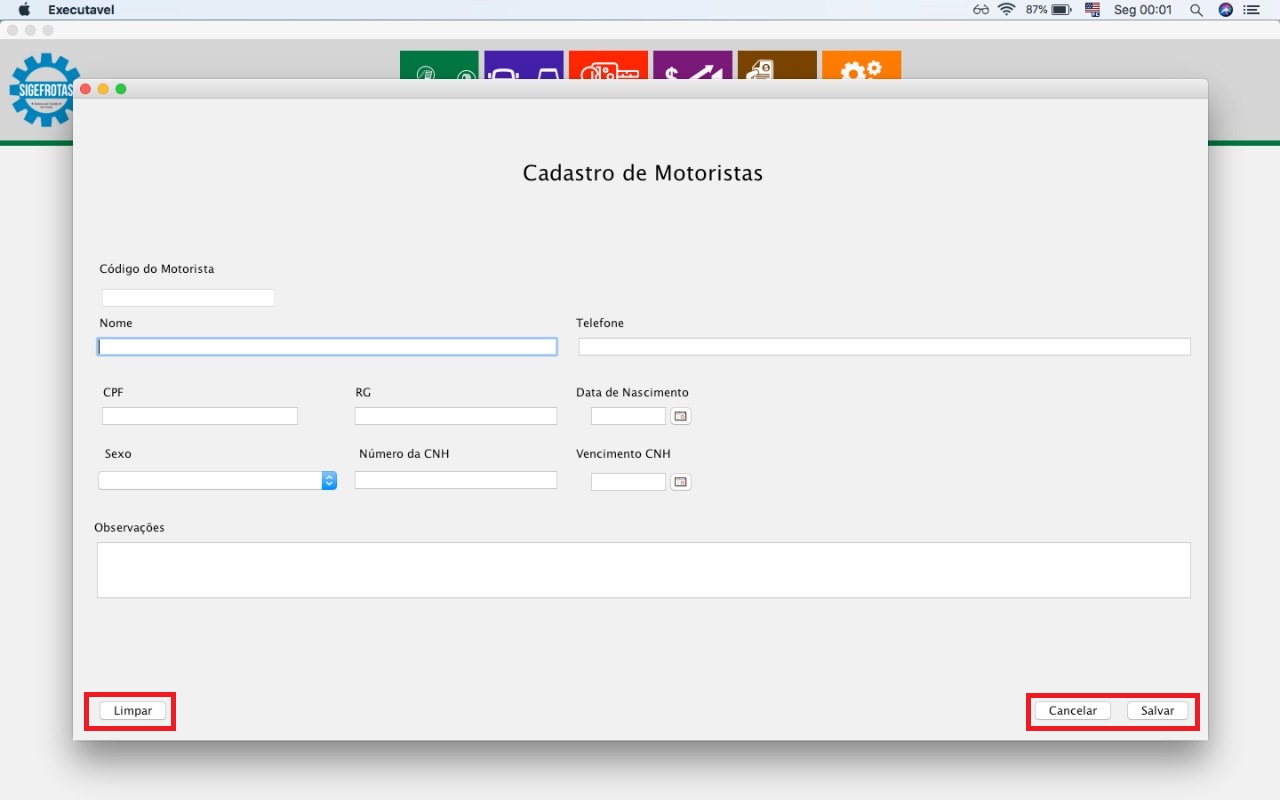


Figura 35 – Tela de Gerência de Frotas

Caso seja selecionado a opção de cadastro de motoristas, algumas informações serão necessárias para realização do cadastro como *Nome, Telefone, CPF, RG, Data de Nascimento, Sexo, Número da CNH* e *Vencimento CNH*, ao terminar de preencher clique em *Salvar*, ou se preferir pode apagar todos os dados clicando em *Limpar,* ou se desistir do cadastro clique em *Cancelar*, como visto na figura 36.

Figura 36 – Tela de cadastro de Motorista

Caso tenha selecionado a opção para consultar motorista será exibido uma lista com todos os motoristas já cadastrados, se quiser excluir, selecione um cadastro e clique em *Excluir Registro*, clique em *Editar Registro* para editar alguma informação ou clique em *Voltar* para retornar ao menu anterior, como visto na figura 37.

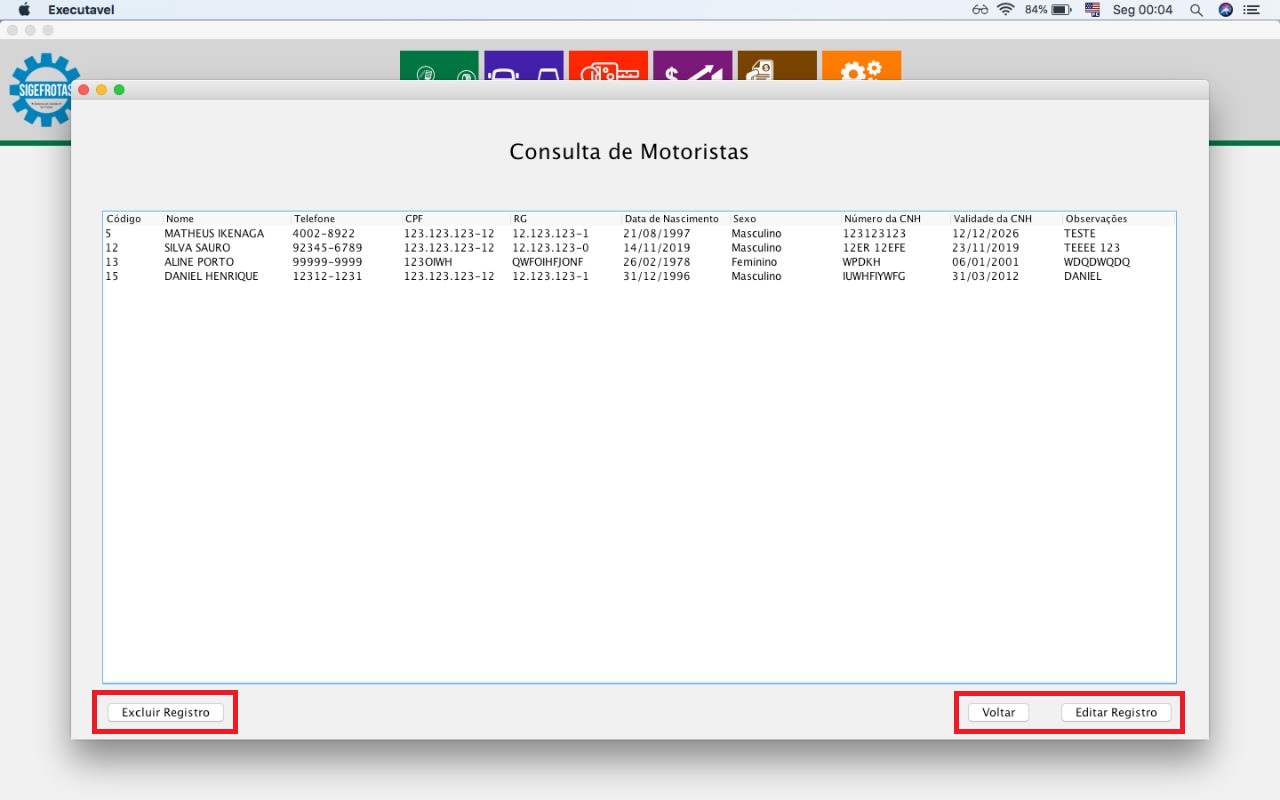
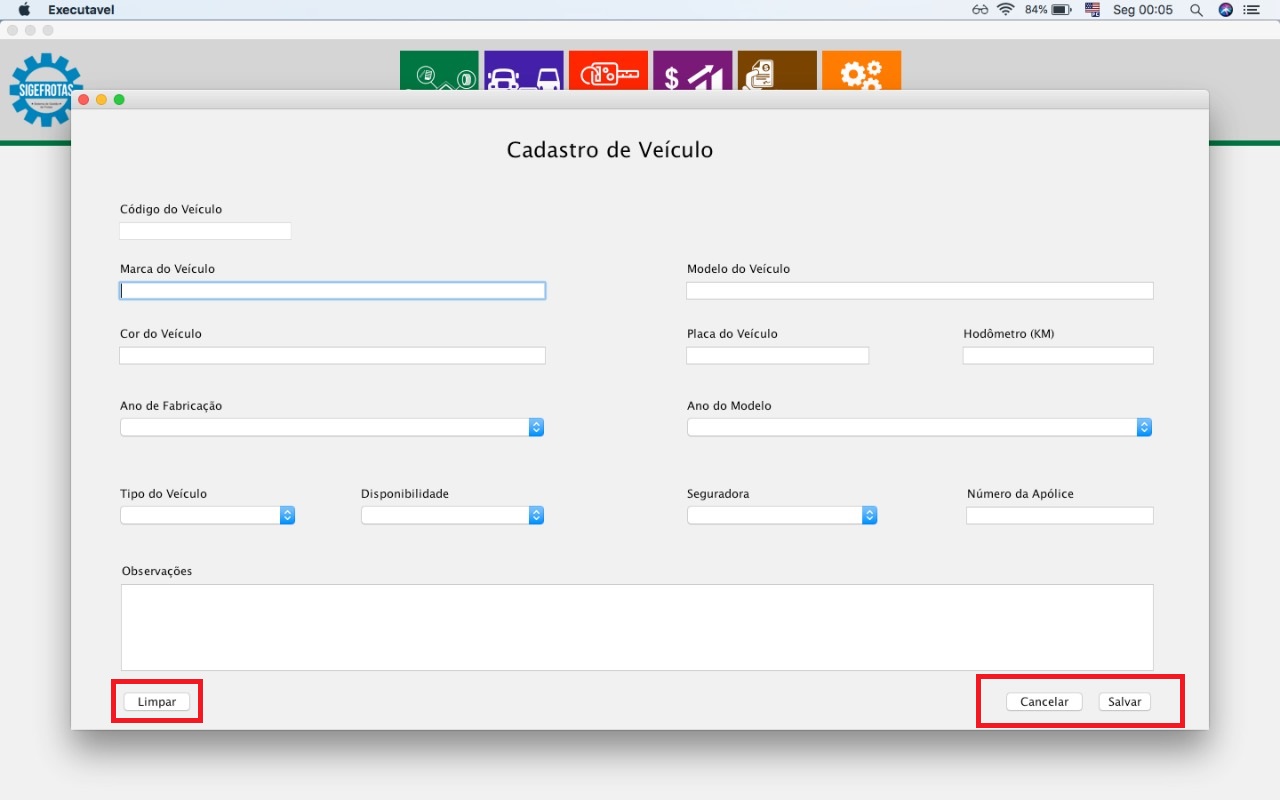


Figura 37 – Tela de consulta de Motoristas.

Caso seja selecionado a opção de cadastro de veículo, será necessário preencher os campos *Marca do Veículo, Modelo do Veículo, Cor do Veículo, Placa do Veículo, Hodômetro (KM), Ano de Fabricação* e *Ano do Modelo*, ao terminar de preencher clique em *Salvar*, ou se preferir pode apagar todos os dados clicando em *Limpar,* ou se desistir do cadastro clique em *Cancelar*, como visto na figura 38.

Figura 38 – Tela de Cadastro de Veículos

Caso seja selecionado a opção para consultar veículo será apresentado todos os veículos cadastrados, se quiser excluir, selecione um cadastro e clique em *Excluir Registro*, clique em *Editar Registro* para editar alguma informação ou clique em *Voltar* para retornar ao menu anterior, como visto na figura 39.

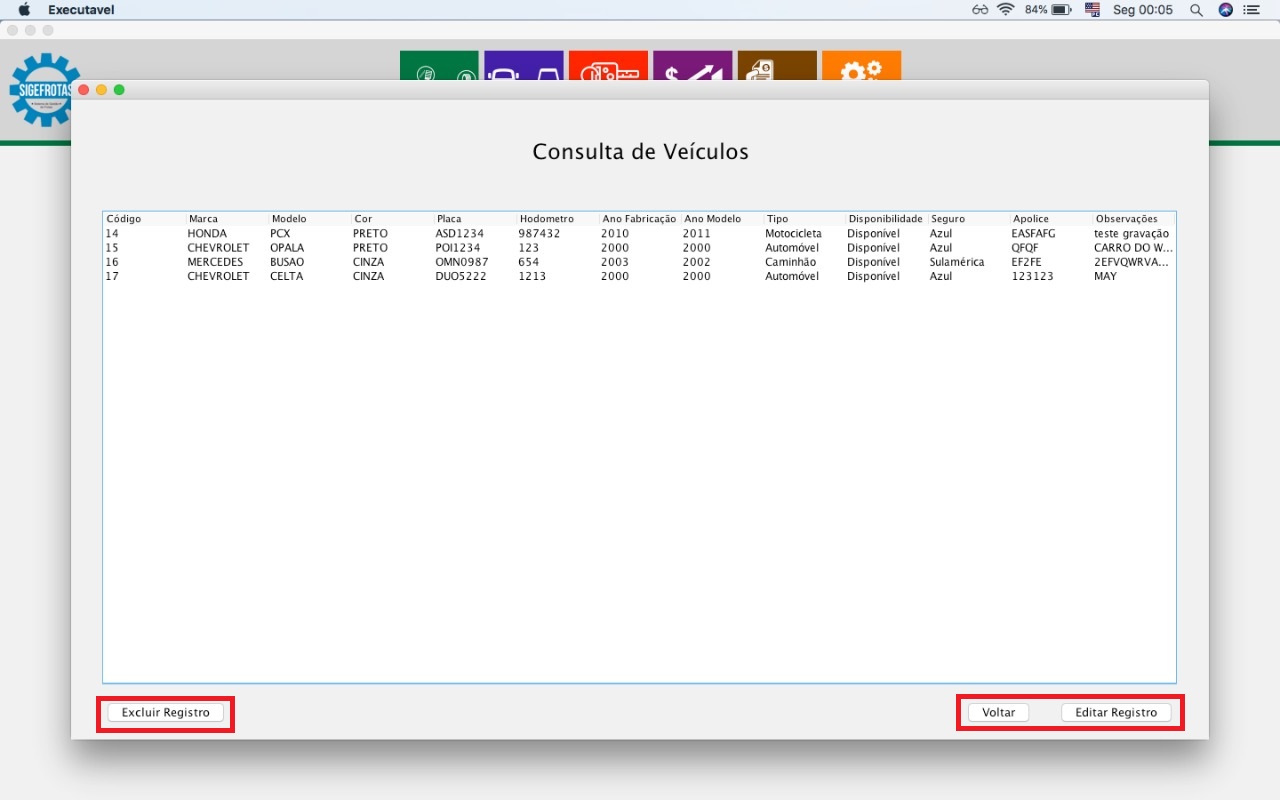


Figura 39 – Tela de Consulta de Veículos

Caso tenha escolhido a opção para cadastrar viagem será necessário preencher a cidade e estado de origem e destino da viagem respectivamente e também os campos restantes, ao terminar de preencher clique em *Salvar*, ou se preferir pode apagar todos os dados clicando em *Limpar,* ou se desistir do cadastro clique em *Cancelar*, como visto na Figura 40.

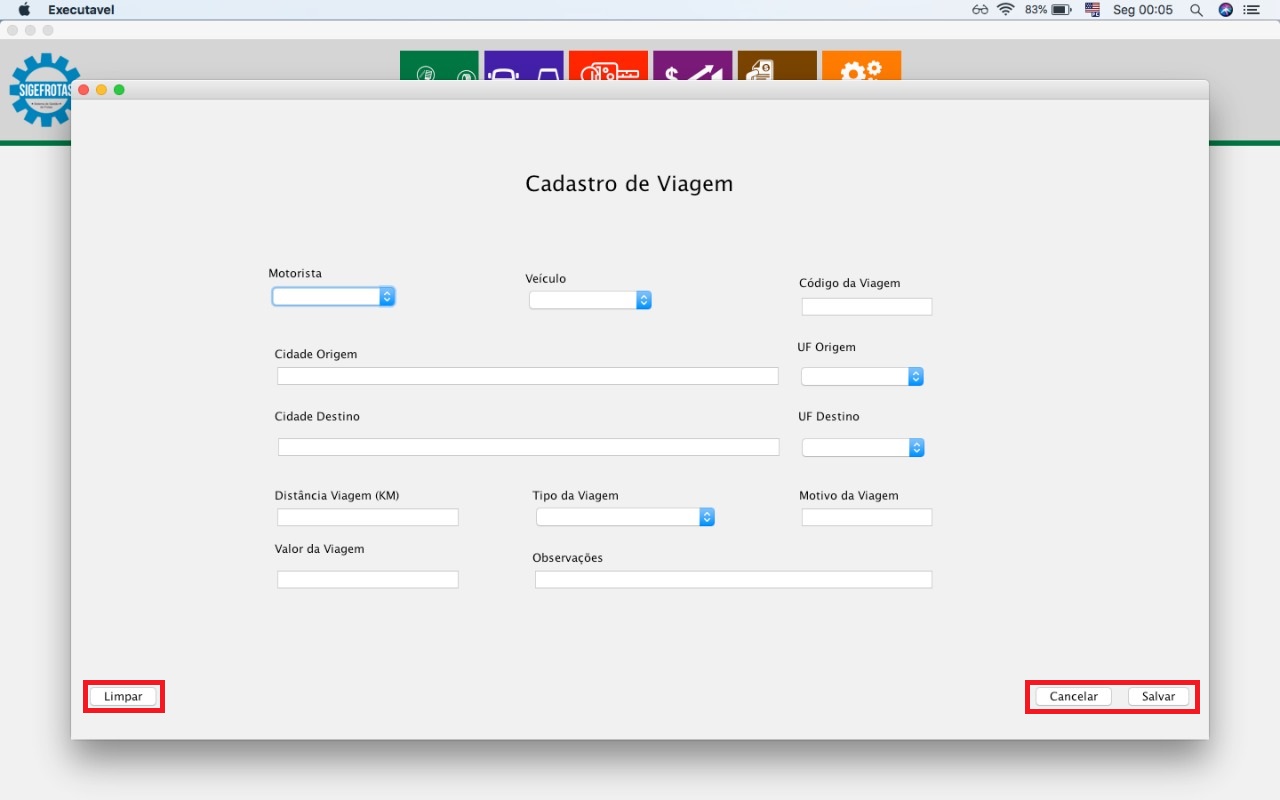


Figura 40 – Tela de Cadastro de Viagem

Caso tenha escolhido a opção para cadastrar aluguel será necessário preencher Motorista, Veiculo, Cidade Origem, UF Origem, Cidade Destino, UF Destino e também algumas outras informações restantes, ao terminar de preencher clique em *Salvar*, ou se preferir pode apagar todos os dados clicando em *Limpar,* ou se desistir do cadastro clique em *Cancelar*, como visto na Figura 41.

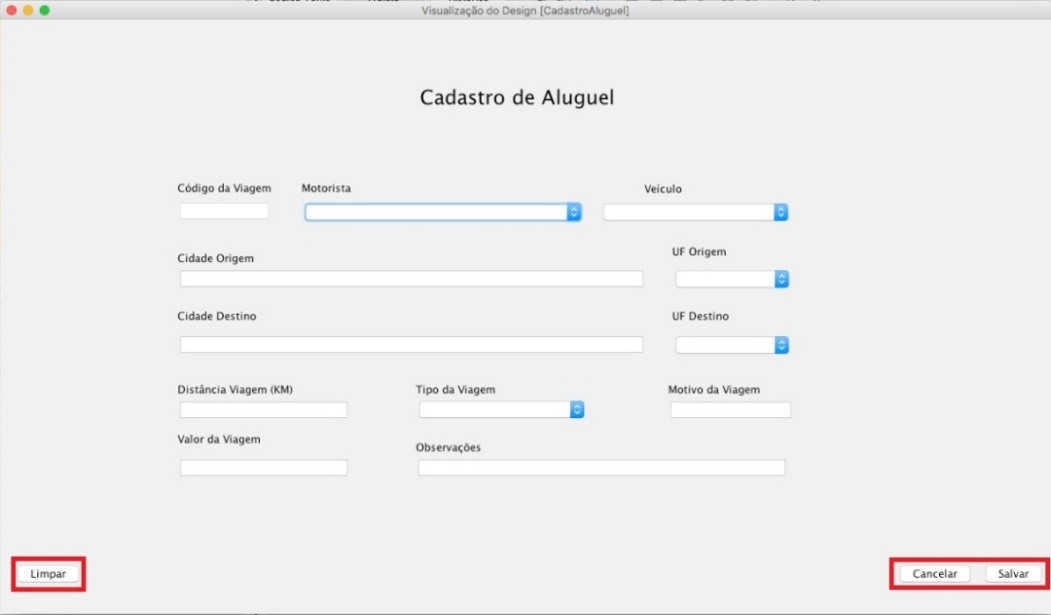


Figura 41 – Tela de Cadastro de Aluguel

Caso seja selecionado a opção para consultar peças será apresentado todas as peças cadastradas, se quiser excluir, selecione um cadastro e clique em *Excluir Registro*, clique em *Editar Registro* para editar alguma informação ou clique em *Voltar* para retornar ao menu anterior, como visto na Figura 42:

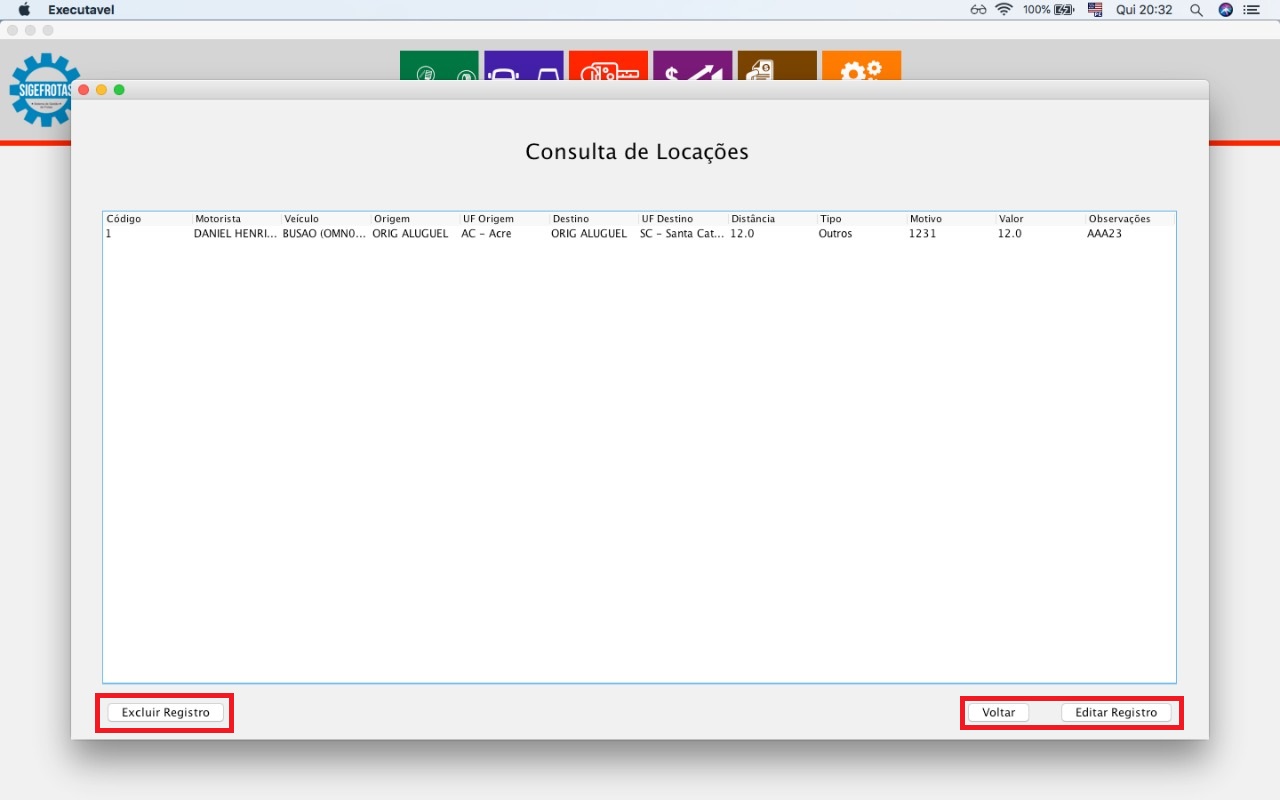


Figura 42 – Tela de Consulta de Locação

Caso seja escolhida opção para cadastrar peças será necessário preencher o *Nome da Peça, Quantidade em Estoque* e *Valor da Peça,* ao terminar de preencher clique em *Salvar*, ou se preferir pode apagar todos os dados clicando em *Limpar,* ou se desistir do cadastro clique em *Cancelar*, como visto na figura 43.

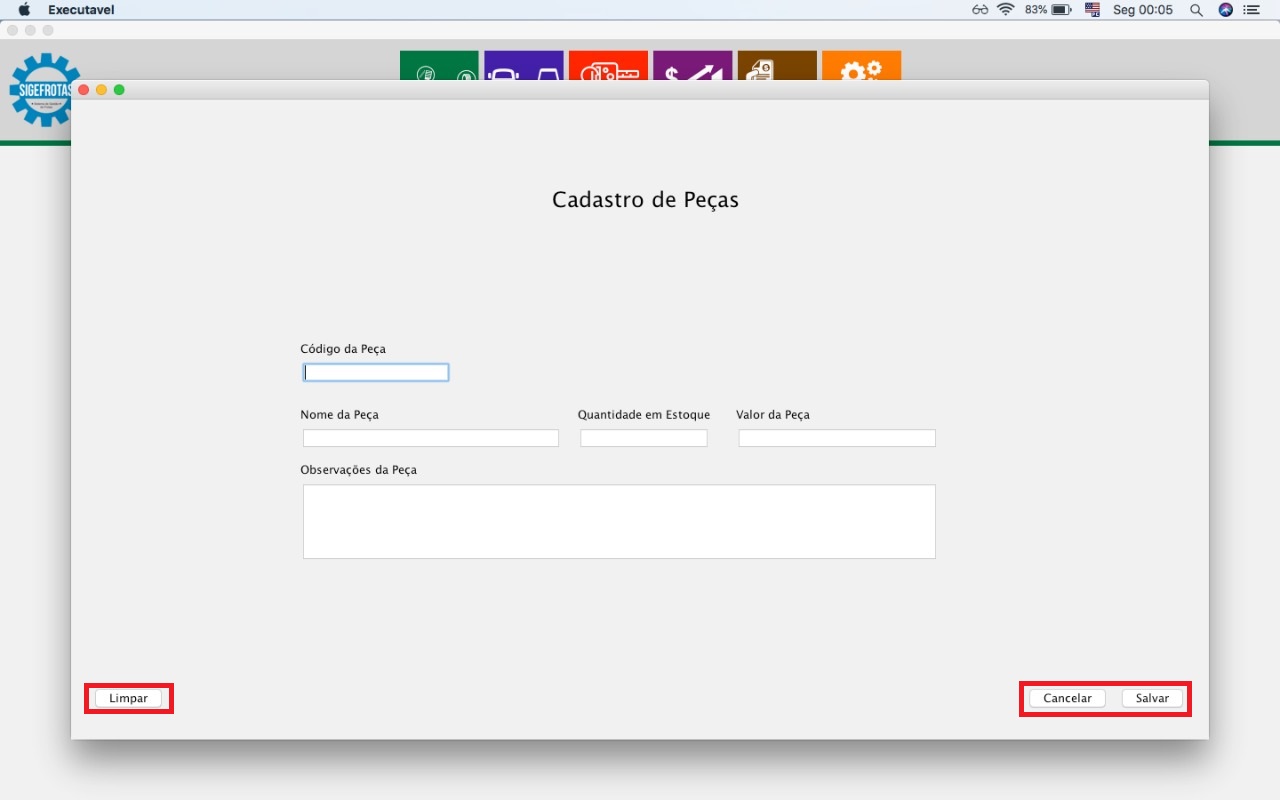


Figura 43 – Tela de Cadastro de peças

Se a opção for a de *Financeiro e Relatórios,* será exibido um menu com suas opções, para realizar, basta clicar em *Incluir* ou *Consultar* respectivamente para cada ação desejada, como visto na Figura 44.

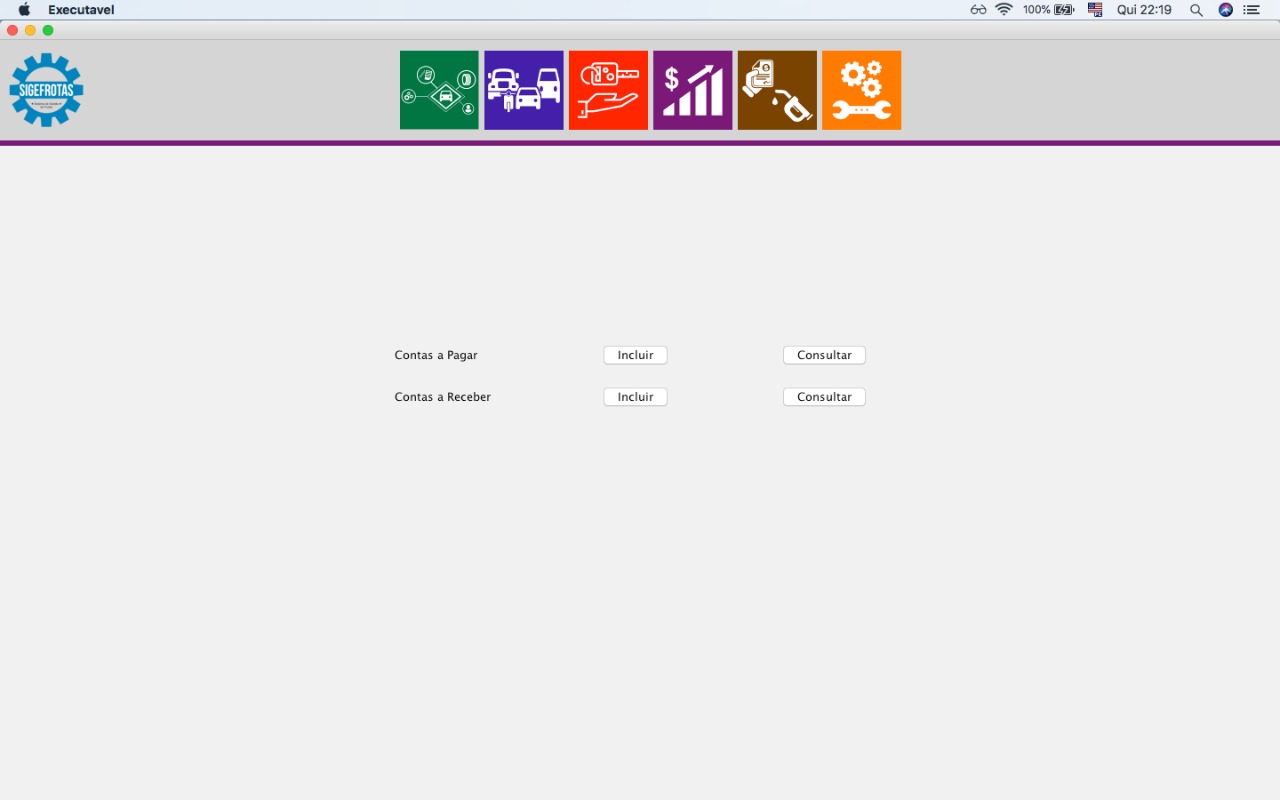


Figura 44 – Tela do Financeiro

Se a escolha for verificar quais contas tem a pagar será necessário preencher o *Valor total da Conta, Categoria, Total Pago e Veiculo,* ao terminar de preencher clique em *Salvar*, ou se preferir pode apagar todos os dados clicando em *Limpar,* ou se desistir do cadastro clique em *Cancelar*, como visto na Figura 45.



Figura 45 – Tela de Conta a Pagar

# Conclusão

O intuito do projeto era o desenvolvimento de um sistema completo para o gerenciamento de frotas de carros, sendo um tema bem atual conforme a grande quantidade de aplicativos de aluguel de carro, controle de frotas e de transporte que tem surgido atualmente. Essa revolução no meio de transporte que trouxe mais praticidade para o trabalhador médio tem sido o foco de grandes projetos e muitas empresas vem surgindo para aproveitar essa grande demanda, tudo isso torna a relevância do sistema muito grande, entendendo tudo que precisa para poder gerenciar uma frota completa de carros, passando por problemas de multa e trânsito, controle de garagem, e tudo mais.

Esse projeto possibilitou um melhor entendimento de como é desenvolvido um sistema, desde as dificuldade e algumas frustrações com o levantamento de requisitos desde a complexidade e nível de abstração para o desenvolvimento do sistema em si, tendo que compreender e trabalhar em cima das divergências do grupo, constantes choques de ideias para o projeto que levavam as discussões, e por fim toda a gratificação de ver o sistema executar as funções conforme o planejado.

Um controle de frota precisa de muitas coisas, como pensar no motorista que irá realizar o caminho, o controle das viagens, como lidar com roubos, acidentes de trajeto, multas e sinistros, peças, mecânico, gasolina entre muitas outras, e passar isso para o sistema foi um verdadeiro desafio, principalmente ter que pensar em tudo isso pensando nos mais diferentes tipos de pessoas que utilizarão o sistema, desde motoristas a gerentes, passando pelos mais diferentes níveis de conhecimento de computação, foi tomado um esforço considerado nessa parte, mas um esforço recompensado já que alcançamos um bom desenvolvimento nessa questão e o produto final ficou bem confortável para o cliente.

Por fim o sistema foi desenvolvido com sucesso e apresentado para o cliente, cumprindo todos os requisitos necessário e integrando com uma portabilidade fornecida pela parte Web como o funcionamento em máquinas menos eficientes fornecidos pela parte de Desktop.

# Referências

# Vazquez, Carlos Eduardo. Engenharia de Requisitos. Software Orientado ao Negócio. Editora Brasport, 2016

# Sites:

# Empresa de Gerenciamento de Frotas Unifrotas.

<http://www.unifrota.com.br/> - Acesso em: 04 de maio de 2019

**Empresa de Gerenciamento de Frotas Produsoft .**

<https://www.produsoft.com.br/informacoes> - Acesso em: 03 de maio de 2019

**Empresa de Gerenciamento de Frotas Masterfrota.**

<https://www.masterfrota.com.br/website/website_home.aspx> - Acesso em: 02 de maio de 2019

**Desenvolvendo um diagrama de Caso de Uso.**

<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408> - Acesso em: 05 de junho de 2019

# O que é um Fluxograma de Processos.

<https://certificacaoiso.com.br/o-que-e-fluxograma-de-processos/> - Acesso em: 02 de junho de 2016