



FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA
LICENCIATURA EM ENGENHARIA INFORMÁTICA

[Engenharia de Software I]

IIIº ano, Iº semestre

**Sistema de Gestão de Frota de
Camiões**

Discentes:

- Dimande, Júlio
- Gulele, Dércio
- Júnior, Abel Jorge Guirrute
- Jeque, Osvaldo
- Machaquele, Andrade
- Mazembe, Dércio
- Rodrigues, Núria

Corpo Docente:

- dr. Vali Issufo
- Monitor: Richaldo Elias

Maputo Junho de 2018



FACULDADE DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

LICENCIATURA EM ENGENHARIA INFORMÁTICA

[Engenharia de Software I]

IIIº ano, Iº semestre

Grupo II

**Sistema de Gestão de Frota de
Camiões**

Índice

1. Introdução	4
2. Glossário	5
3. Requisitos.....	6
3.1. Requisitos de utilizadores.....	6
3.2. Requisitos Funcionais (Casos de usos)	7
3.3. Requisitos Não Funcionais.....	7
4. Modelos do sistema.....	10
4.1. Diagramas de Classes.....	10
4.2. Diagrama de Sequencia.....	19
4.3. Diagrama de Estados.....	22
5. Interface de Utilizador (Protótipos)	24
6. Testes	25

Índice de Diagramas

Figure 1: Diagrama de casos de uso	7
Figure 2: Página de login	8
Figure 3: Página inicial	9
Figure 4: Registo de viagem	9
Figure 5: Diagrama de classes	18
Figure 6: 6DC: Viagem.....	10
Figure 7: DC: Carga 1	11
Figure 8: DC: Gestor 1.....	12
Figure 9: Manutenção	14
Figure 10: Cliente	14
Figure 11: Camião.....	15
Figure 12: Motorista	16
Figure 13: Funcionário.....	17
Figure 14: Supervisor.....	17
Figure 15: Diagrama de sequência criar viagem.....	19
Figure 16: Monitorar Viagem	19
Figure 17: Visualizar Historico.....	20
Figure 18: Verificar Disponibilidade do Camião.....	20
Figure 19: Confirmar Chegada	21
Figure 20: Confirmar Viagem.....	21
Figure 21: Diagrama de estados do camião	22
Figure 22: Alocação do motorista.....	22
Figure 23: Viagem	23

Índice de tabelas

Table 1: caso de teste - confirmar entrega	25
Table 2: caso de teste - registar viagem	26
Table 3: Verificar histórico	26
Table 4: Verificar solicitação.....	26

Prefácio

O documento seguinte destina-se aos clientes e ou interessados em usar o sistema, ou seja usuários que em primeira instância poderão ser o Gestor, “cliente”, motorista e o supervisor, possui versões duas versões em Java, a segunda surgiu com o interesse de melhorar a primeira e incluir interface gráfica, nesta fase encontra-se o sistema disponível numa versão Web pois assim acredita-se que desta maneira o cliente ira expandir o seu negocio para novos horizontes.

1. Introdução

O propósito do sistema desenvolvido é dinamizar e otimizar (armazenando dados que a posterior poderão ser usados para diferentes tomadas de decisão) por meios informáticos o modo de viver, forma de trabalho, a relação custo-benefício entre o cliente e o prestador de serviços e fundamentalmente a questão tempo que é um recurso imprescindível nos negócios. O sistema terá algumas plataformas integradas como um Web Site.

2. Glossário

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir:

DE- Diagrama de Estados;

DC- Diagrama de Classes;

HTML- *HyperText Markup Language*, ou Linguagem de Marcação de Hipertexto.

LP- Linguagem de Programação;

SO- Sistema Operacional;

SGBD- Sistema de Gestão de Base de Dados;

SGFC- Sistema de Gestão de Frota de Camiões;

PWSGC – Programação Web e Sistema de Gestão de Conteúdos;

PHP- *Hypertext Preprocessor*;

RF – Requisitos Funcionais;

RNF- Requisitos não funcionais;

SQL- *Structured Query Language*, ou Linguagem de Consulta Estruturada;

3. Requisitos

Nos **requisitos funcionais** preocupamo-nos em descrever as funcionalidades e os serviços do sistema, ou seja, as funções que o sistema deve fornecer para o cliente e como o sistema se comportará em determinadas situações:

- [RF001] O Sistema deve registar clientes e Motoristas (entrada).
- [RF002] O Sistema deve emitir um histórico de viagens (saída).
- [RF003] O Sistema deve passar uma viagem da situação "solicitada" para "confirmada " quando o Gestor tiver feito a confirmação (mudança de estado)
- [RF004] O Gestor, cliente e motorista o podem consultar seus dados no sistema.
- [RF005] O sistema deve permitir ao gestor e o cliente monitorar a viagem, isto e, verificar quanto tempo falta ate ao destino de chegada.
- [RF006] O sistema deve permitir o gestor fazer toda e qualquer alteração de dados existentes na base de dados;
- [RF007] O sistema garantir sempre que o supervisor de um “*check*” para confirmar a partida de um determinado cliente;

3.1. Requisitos de utilizadores

Nesta etapa pretende-se descrever os serviços fornecidos aos utilizadores.

Segundo Sommerville, São declarações, em uma linguagem natural com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar:

- O sistema deve gerar relatórios que mostram o custo da viagem;
- Demonstração da quantidade total de carga para realização de uma viagem;
- Conhecimentos básicos a uma solicitação de frota de camiões;
- O usuário deve ser capaz de fazer uma leitura no documento de requisitos;
- O sistema deve permitir o preenchimento dos dados do utilizador;

3.2. Requisitos Funcionais (Casos de usos)

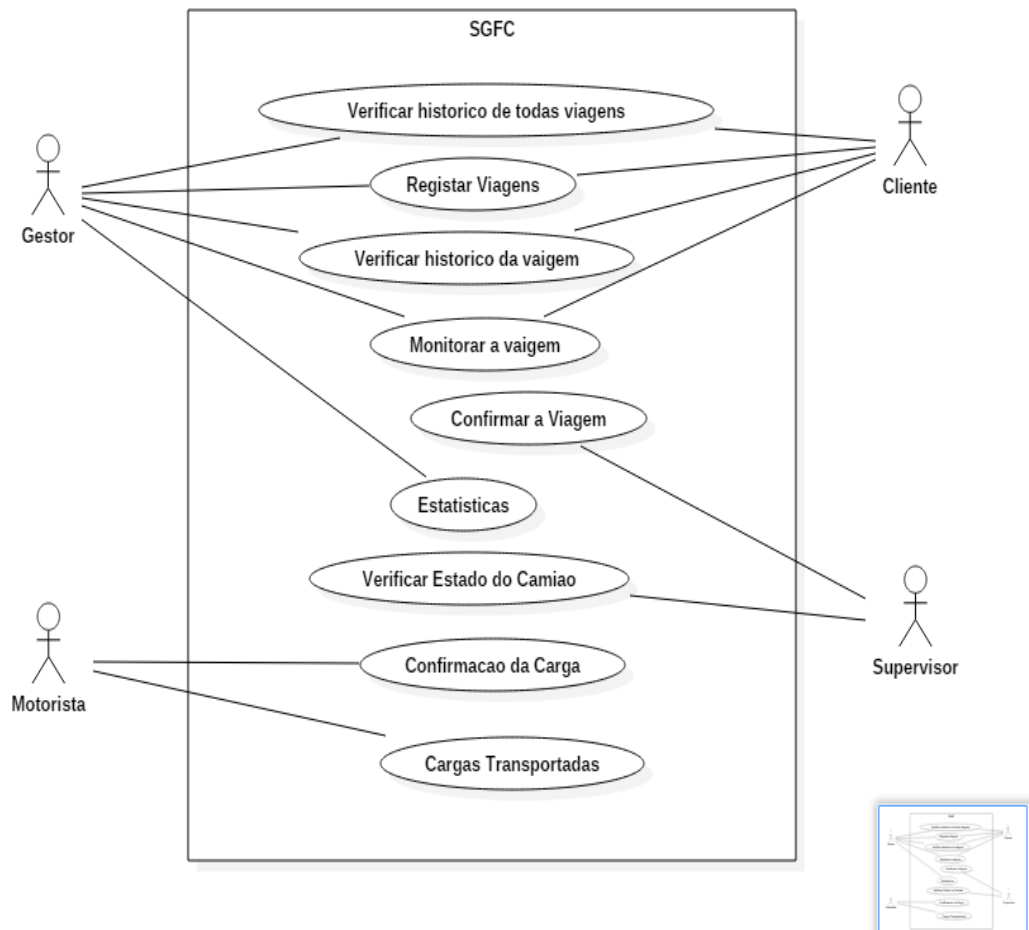


Figure 1: Diagrama de casos de uso

3.3. Requisitos Não Funcionais

Os **requisitos não funcionais** definem propriedades e restrições do sistema como tempo, espaço, linguagens de programação, versões do compilador, SGBD, Sistema Operacional, método de desenvolvimento, etc. Uma dica importante é que os requisitos não funcionais são geralmente mensuráveis e assim devemos preferencialmente associar uma medida ou referência para cada requisito não funcional.

- [RNF001] O sistema deve imprimir o relatório em até 5 segundos para os caso do uso em que pretende-se visualizar histórico.
- [RNF002] Todos os relatórios devem seguir o padrão de relatórios especificado pelo Gestor;
- [RNF003] O sistema deve ser implementado nas LP's HTML, PHP, JavaScript e SQL.
- [RNF003] O sistema deve mostrar constantemente a localização do motorista;

[RNF001] Apresentar Tutorial

Neste tutorial pretende-se indicar as etapas para o processo *registar uma viagem*.

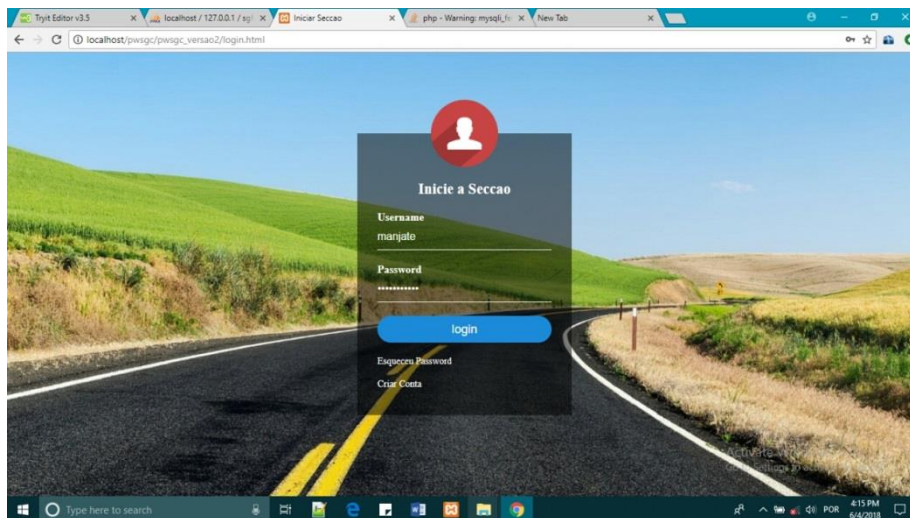


Figure 2: Página de login

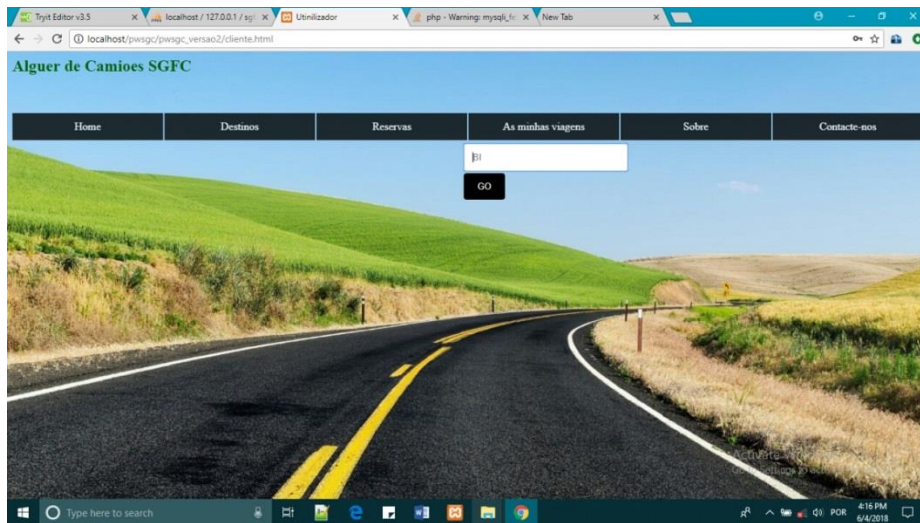


Figure 3: Página inicial

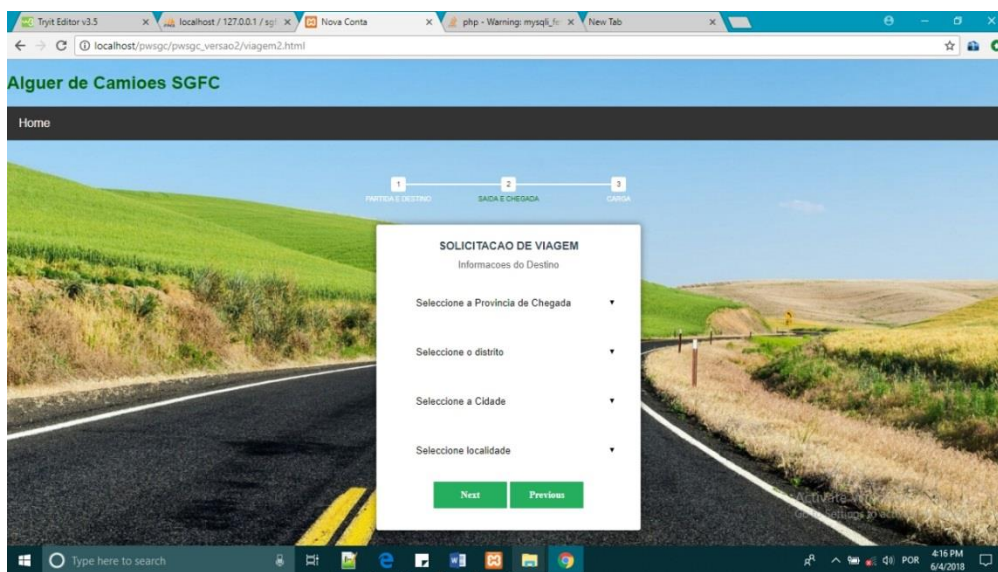


Figure 4: Registo de viagem

Prioridade: ☒ **Essencial** ☐ **Importante** ☐ **Desejável**

4. Modelos do sistema

Na UML, os diagramas de implementação modelam a arquitectura física de um sistema. Os diagramas de implementação mostram os relacionamentos entre os componentes de software.

Pretende-se nesta etapa mostrar como as classes relacionam-se no sistema, a utilidade de cada método, bem como os seus atributos.

4.1. Diagramas de Classes

Para um melhor entendimento do relacionamento entre os diversos atributos e classes neste ponto ir-se-á descrever o que cada elemento desempenha no sistema:

[DC001] Viagem

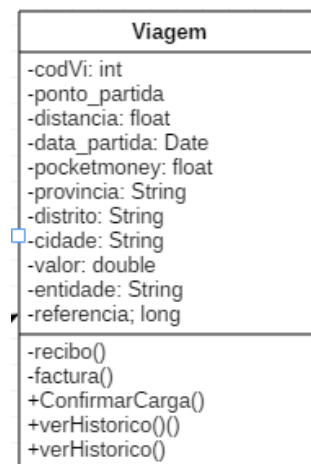


Figure 5: 6DC: Viagem

A classe Viagem, possui os atributos:

`codVi` – que representa o código cada viagem feito.

`ponto_partida` – e local de partida, isto é, o lugar aonde é carregado o camião;

`distancia` – e a distancia a ser percorrida do ponto de partida ate ao destino final;

data_partida- data de partida do camião já carregado;

pocketmoney – valor de ajuda de custo que o motorista ira levar durante a viagem.

provincia- pronvincia de destino da carga;

distrito - distrito de destino da carga;

cidade- cidade de destino da carga;

valor – valor que o cliente ira pagar pelo transporte;

entidade- entidade/instituição na qual se efectua os pagamentos;

referencia- conta para a qual se efectuara o pagamento;

recibo() – documento passado ao cliente depois que e confirmado o pagamento

factura() – documento emitido antes do pagamento;

confirmarCarga() – confirmar a carga a ser transportada;

verHistorico() – visualizar as viagens efectuadas.

[DC002] Carga

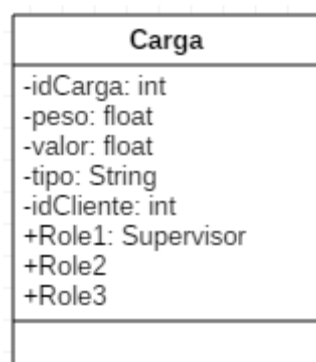


Figure 6: DC: Carga 1

A classe carga tem o seguinte atributos/metodos:

idCarga - código único identificador da carga;

peso- peso da carga a ser transportada;

valor- este atributo refere-se ao valor o qual a carga esta avaliada;

tipo- e a carga/ tipo de carga a ser transportada., Ex. Cimento.

idCliente- o código único identificador que relaciona a carga e o cliente

[DC003] Gestor



Figure 7: DC: Gestor 1

A classe Gestor possui os seguintes:

nome - e o nome do gestor do sistema e ou da empresa;

Os métodos abaixo são responsáveis por permitir o gestor a:

`criarViagem()` – criar uma viagem;

`cancelarViagem()` – cancelar uma viagem, que já havia sido criada;

`confSaida()` – confirmar a saída/viagem dos caminhões;

`adicionarCli()` – adiciona um novo cliente;

`editDados()` – alterar dados referentes ao cliente;

`addMotorista()` – adiciona um motorista novo;

`deleteMotorista()` – remove um motorista no sistema;

`editarViagem()` – altera os dados da viagem;

`addDadosManu()` – estado de um determinado caminhão em manutenção;

`deleteManu()` – remove dados de manutenção de um caminhão;

`editarManu()` – altera os dados de manutenção;

`addCamiao()` – adiciona um novo caminhão no sistema;

`deletarCamaio()` – remove um caminhão do sistema;

`editardadoscamao()` – altera atributos de um caminhão;

`adicionarCarg()` – adiciona carga de uma viagem;

`removeCarg()` – remove uma determinada carga do sistema;

`alteraCarg()` – altera os atributos de uma carga;

`verHistorico()` – retorna todas as operações (viagens) feitas no sistema;

[DC004] Manutenção

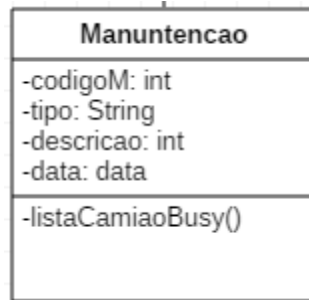


Figure 8: Manutenção

A classe manutenção possui os seguintes atributos/métodos:

`codigoM` – código que identifica o um camião que foi a manutenção;

`tipo` – indica o tipo de manutenção que o camião efectuou. Ex. Reparar radiador;

`descricao` – Descreve as circunstancias que levaram o camião a manutenção;

`data` – o dia em que o camião foi a manutenção;

`listaCamiaoBusy()` – visualiza a lista de camiões em manutenção;

[DC005] Cliente

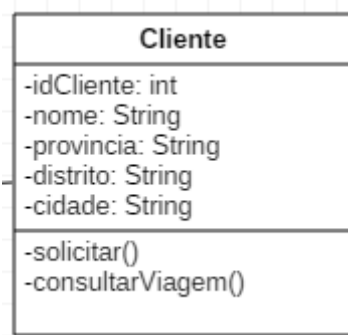


Figure 9: Cliente

A classe Cliente possui os seguintes atributos/métodos:

`idCliente` – código único identificador de cada cliente;

nome- nome do cliente;

provincia- província de onde o cliente solicita a viagem;

distrito- distrito de onde o cliente solicita a viagem;

cidade- cidade onde o cliente solicita a viagem;

solicitar() - permite ao solicitar uma viagem;

consultarViagem() - consulta o estado da viagem solicitado;

[DC006] Camiao

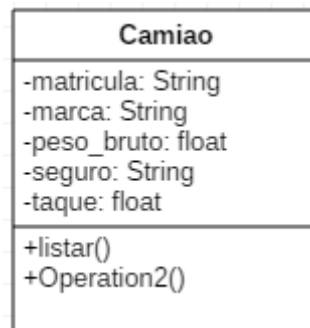


Figure 10: Camião

A classe Camiao possui os seguintes atributos/métodos:

matricula- matricula do camião;

marca- marca do camião;

peso_bruto- carga máxima que o camião suporta;

seguro- verifica se o camião tem ou não seguro;

taque- indica o numero de litros que o tanque suporta;

`listar()` – lista todos os camiões;

[DC006] Motorista

Motorista
-id: int -dataNasc: Date -salario: float -nome: String
-confChegada() -Descarregado()

Figure 11: Motorista

A classe Motorista possui os seguintes atributos/métodos:

`id`- código identificador do motorista;

`dataNasc`- data de nascimento do motorista.

`salario`- salario do motorista;

`nome` - nome completo do motorista

`confChegada()` – confirmação de chegada do motorista;

`Descarregado()` – confirmação da entrega da carga;

[DC007] Funcionario

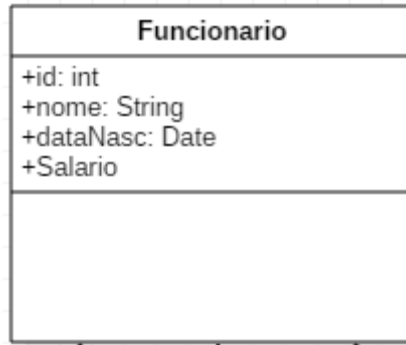


Figure 12: Funcionário

A classe Funcionario e também “pai” das classes Gestor , Motorista e Supervisor possui os seguintes atributos/métodos:

Id - código identificador do funcionário;

salario- salario do funcionário;

nome - nome completo do funcionário;

dataNasc- data de nascimento do funcionário.

[DC008] Supervisor

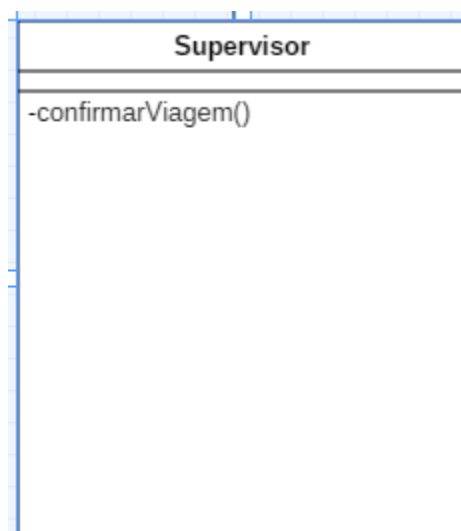


Figure 13: Supervisor

A classe Supervisor possui o seguinte método:

`confirmarViagem()` - indica se o caminhão esta em condições partir ou não.

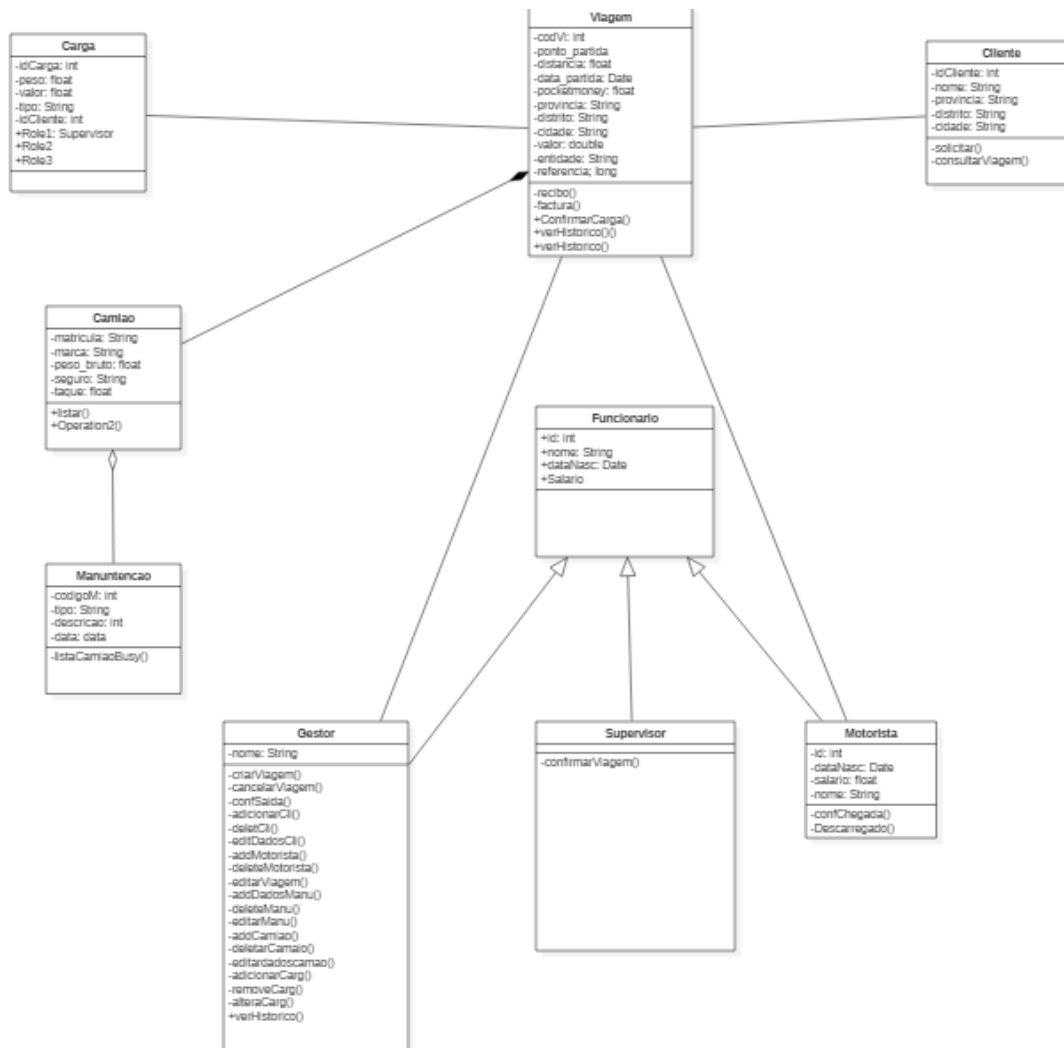


Figure 14: Diagrama de classes

4.2. Diagrama de Sequencia

Código: [DS001] Registrar Viagem

Descrição: Depois que o cliente solicita uma viagem esta só e confirmada pelo gestor depois que o cliente efectua o pagamento pelo transporte.

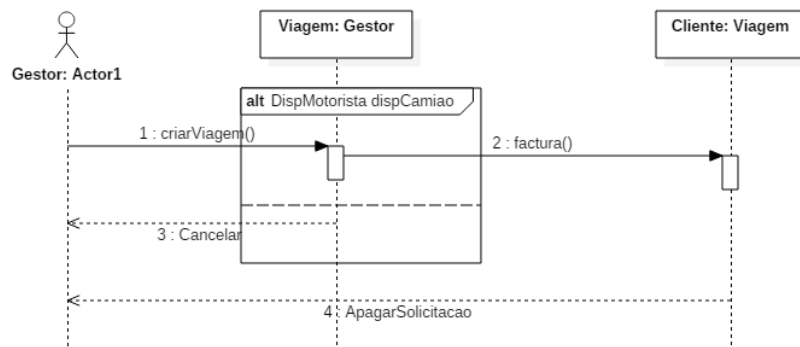


Figure 15: Diagrama de sequência criar viagem

Código: [DS002] Monitorar Viagem

Descrição: O cliente/gestor tem acesso ao sistema por meio de uma autenticação, uma das formas de monitoramento para os dois atores, e a verificação do tempo restante na janela principal do sistema (outra e a comunicação via telefónica motorista-gestor-cliente) .

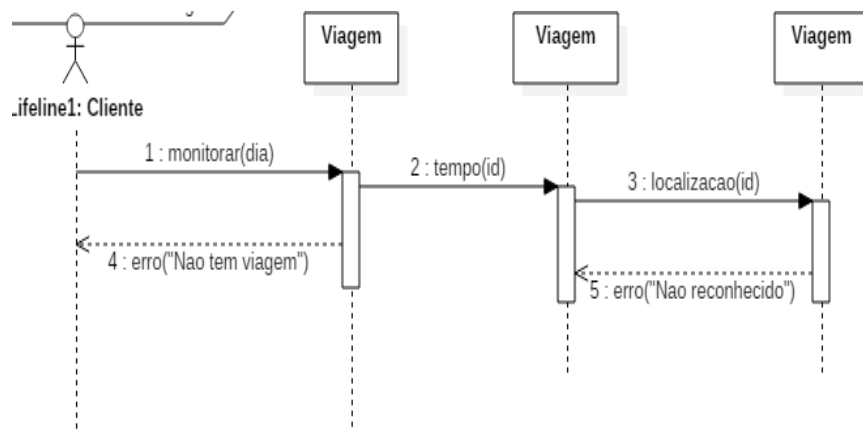


Figure 16: Monitorar Viagem

Código: [DS002] Visualizar Histórico

Descrição: O cliente/gestor tem acesso ao sistema por meio de uma autenticação, na janela principal selecciona a opção histórico, uma vez seleccionada a opção ira retornar toda a informação referente as viagens já realizadas (em particular para o cliente e geral para o gestor).

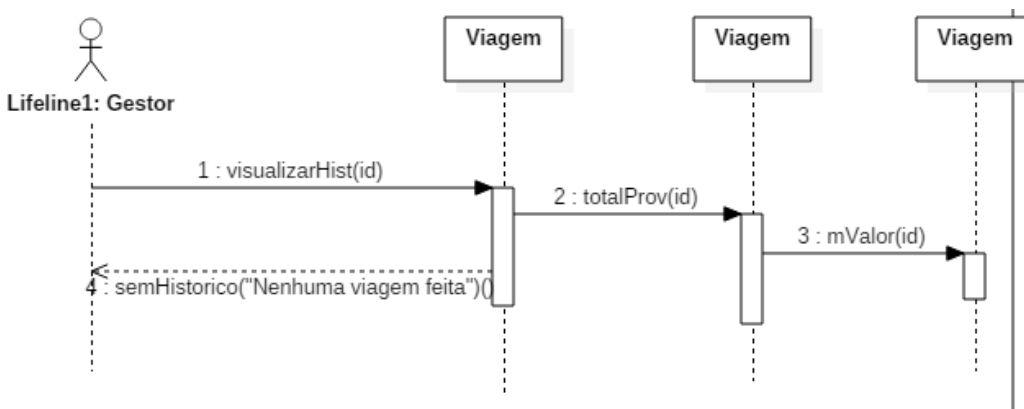


Figure 17: Visualizar Historico

Código: [DS002] Verificar Disponibilidade do Camião

Descrição:

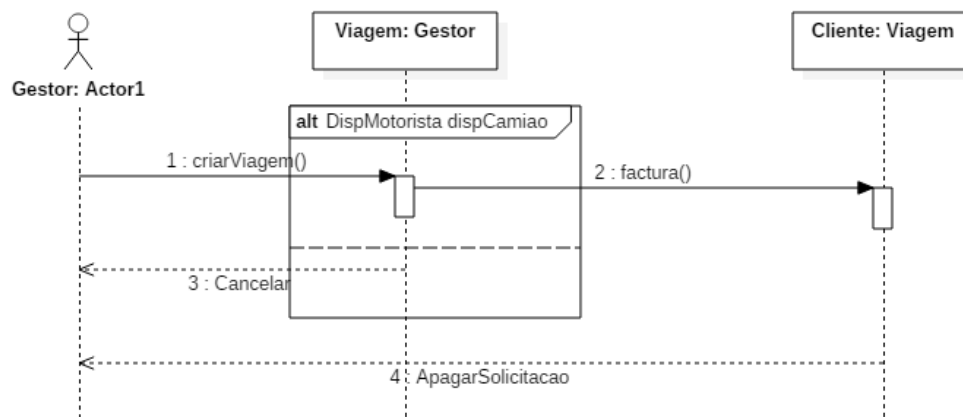


Figure 18: Verificar Disponibilidade do Camião

Código: [DS003] Confirmar Chegada

Descrição: Depois que o motorista chegar ao local de entrega da carga, dá um “*check*” pra confirmar efectivamente a sua chega.

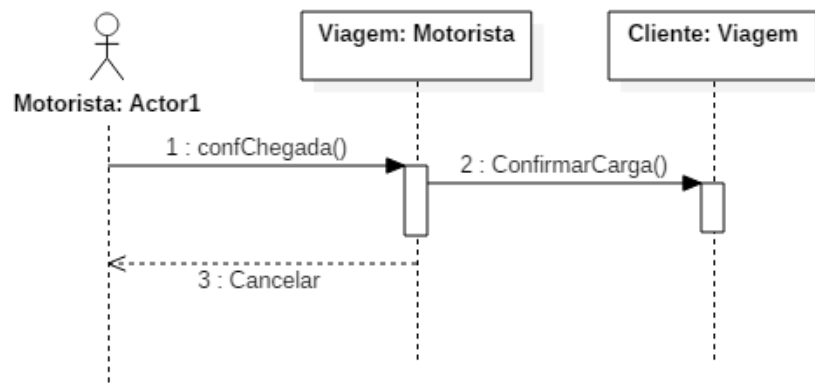


Figure 19: Confirmar Chegada

Código: [DS004] Confirmar Viagem

Descrição: O supervisor chega ao sector do gerente, para que possa ter acesso a senha, com o objectivo de fazer uma verificação geral da solicitação e registo da viagem feita pelo cliente.

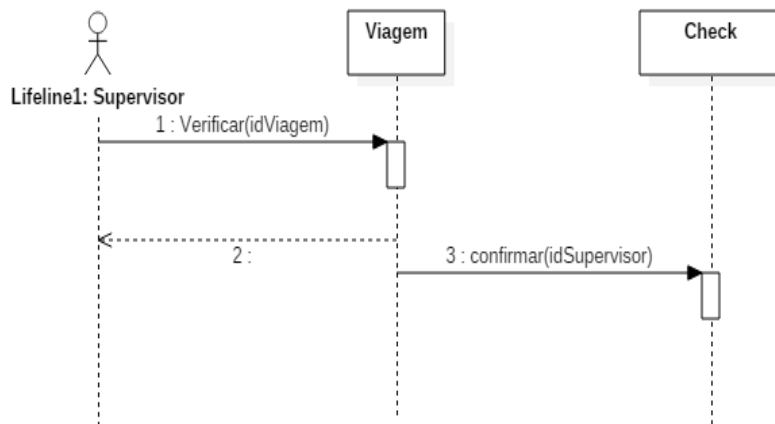


Figure 20: Confirmar Viagem

4.3. Diagrama de Estados

Código: [DE001] Camião

Descrição: Neste DE pretende-se ilustrar os possíveis estados em que um camião na frota pode se encontrar:

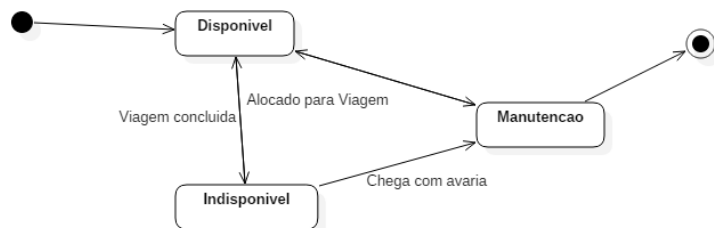


Figure 21: Diagrama de estados do camião

Código: [DE002] Alocar Motorista

Descrição: Em que estados um motorista pode se encontrar na frota? Esta pergunta é respondida com o auxílio do seguinte diagrama:

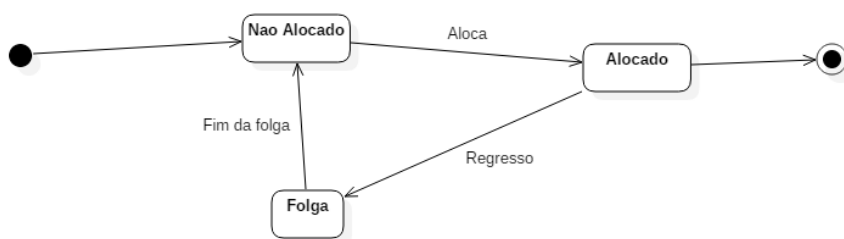


Figure 22: Alocação do motorista

Código: [DE001] Viagem

Descrição: Neste diagrama descrevemos os estados em que uma viagem pode ser encontrar:

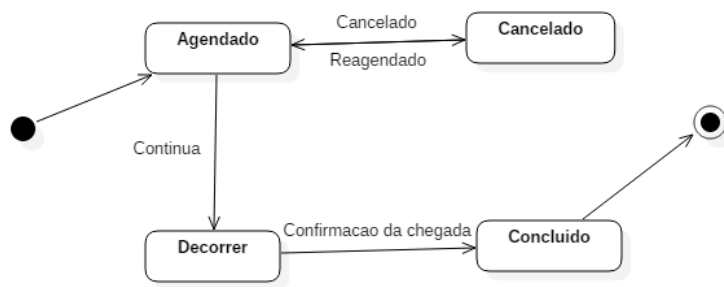
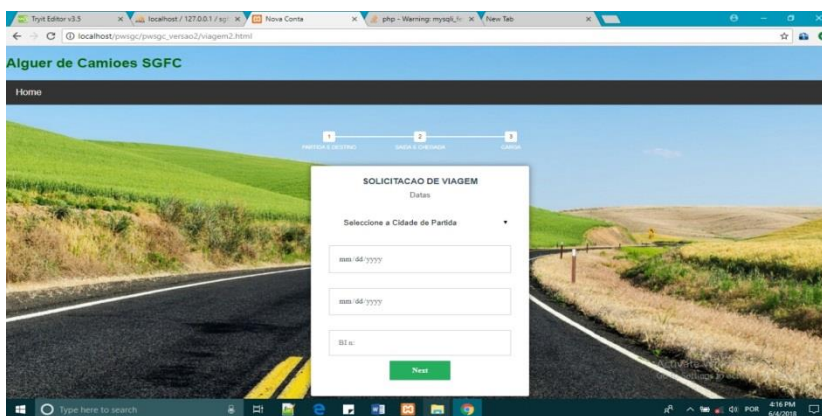
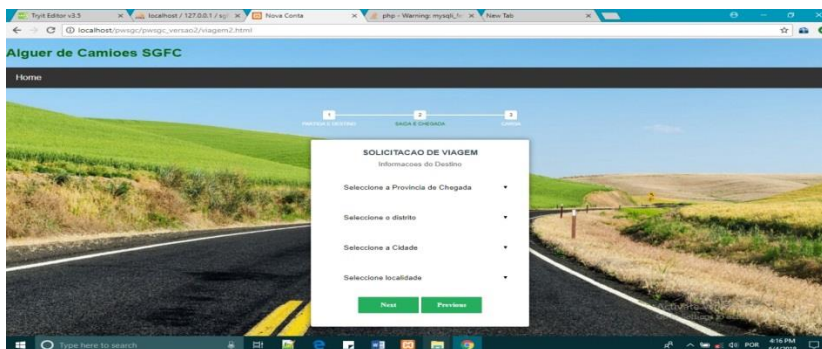
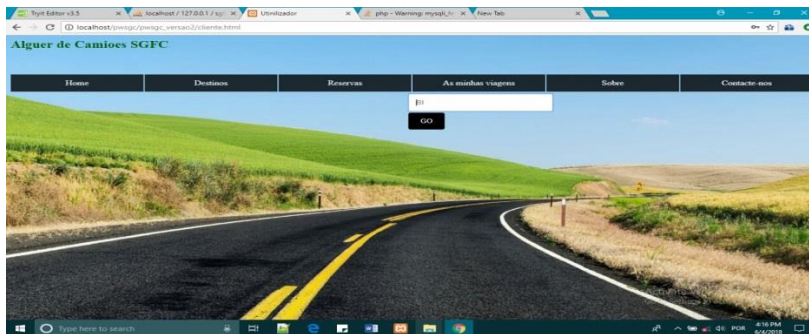
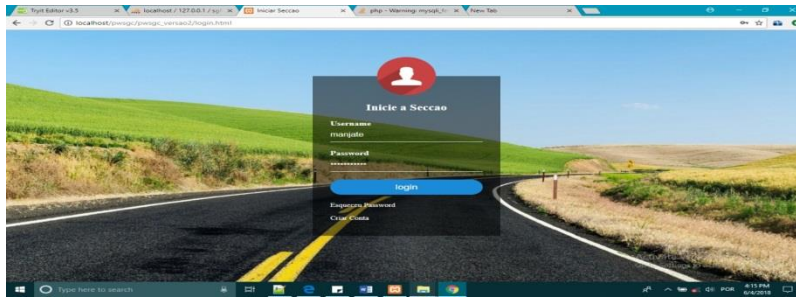


Figure 23: Viagem

5. Interface de Utilizador (Protótipos)



6. Testes

[Casos de testes]

Caso de teste	V 11- Confirmação da entrega carga
Pre-condições	1.Login feito pelo motorista. 2.Ter Entrado com o codigo da viagem.
Procedimentos	1. O actor (motorista) seleciona a opção confirmar a carga. 2. O sistema deve responder com um campo para inserir o código da Viagem. 3. Deve entrar co o código 4. O sistema apresenta os dados da Viagem e um botão confirmar Entrega. 5. O motorista clica no botão enviar a confirmação.
Resultado Esperado	Obter a mensagem “ Confirmado Com Sucesso” Enviar mensagem de confirmação da carga ao cliente e ao Gestor.
Dados de entrada	
Crterios Especiais	Não se aplica
Ambiente	
Implementacao	Automatica
Interação	Uma vez

Table 1: caso de teste - confirmar entrega

Caso de teste	V 1- Registrar Viagem
Pre-condições	Ter recebido uma notificação Ter clicado no botão notificações A notificação deve ter assunto: Solicitação de viagem
procedimentos	1.O gestor, recebe a notificação e clica no botão “notificação” que carrega uma lista de notificações. 2. O actor deve abrir as notificações com assunto solicitação. 3.O sistema deve exibir um formulario devidamente preenchido pelo cliente e no final da pagina um botão registrar . 4.Ao clicar no botão devera ser disponibilizado uma pagina com as opções de alocação de do camião, motorista e a carga á viagem, e a data de partida. 5.Por fim deve clicar no botão confirmar viagem que regista a viagem na BD(Base de dados) e uma mensagem é enviada ao cliente, supervisor e motorista.
Resultado Esperado	Registrar Viagem na BD e notificar o cliente, supervisor e motorista sobre a viagem agendada.
Dados de entrada	
Crterios Especiais	Não se aplica
Ambiente	

Implementacao	Manual
Interação	Uma vez

Table 2: caso de teste - registrar viagem

Caso de teste	V2- Verificar o histórico
Pre-condições	Ter clicado no botão Ver Historico Na Pagina Inicial do Gestor.
procedimentos	1.O actor(Gestor) deve clicar no botão ver historico localizado no topo da página do lado direito do botão registrar. 2.O sistema deve responder redirecionando para outra página, que faz a listagem do historico
Resultado Esperado	carregar o historico de todas as viagens.
Dados de entrada	
Critérios Especiais	--
Ambiente	
Implementacao	Manual
Interação	1 interação

Table 3: Verificar histórico

Caso de teste	V3- Verificar a solicitação
Pre-condições	1.Login feito pelo cliente 2. ter preenchido o formulario de solicitação da viaagem e clicado no botão enviar.
procedimentos	1. Depois de clicado a opção solicitar viagem, o sistema devera responder com um formulario , onde o cliente devera preencher o mesmo com os dados da viagem. 2.Devera clicar no botão Enviar Solicitação.
Resultado Esperado	1.Obter a Mensagem “ Solicitação enviada”; 2. Gestor deve receber a solicitação
Dados de entrada	Data de partida e a Carga
Critérios Especiais	--
Ambiente	
Implementacao	Manual
Interação	1 interação

Table 4: Verificar solicitação

7. Conclusão

Em jeito de conclusão pode-se dizer que alcançou-se o primeiro objectivo que é a criação do sistema (SGFC) e segundo, seguindo todos os requisitos do utilizador e regras de negocio tendo como alicerce as técnicas adoptadas na Engenharia de Software mas com certos aspectos ainda por ser melhorados, que são destinados aos estudantes ou formados profissionais na área.