



**Projeto Interdisciplinar  
4º Período**

**UaiFuel**

**Éder Antonio Rangel Marques  
Elias Mantovan Reboucas  
Estevan Navarro Lago  
Pedro Henrique Coimbra Rodrigues Sposito Alves**

Junho de 2021

# Sumário

1	INTRODUÇÃO .....	1
1.1	OBJETIVO .....	1
1.2	JUSTIFICATIVA .....	1
2	ANÁLISE DO SISTEMA PROPOSTO .....	3
2.1	ESCOPO DO SISTEMA .....	3
2.2	NÃO ESCOPO DO SISTEMA .....	3
2.3	REQUISITOS DO SISTEMA .....	3
2.4	GLOSSÁRIO .....	4
2.5	MAPEAMENTO DE PROCESSOS .....	4
2.6	DIAGRAMA DE CASO DE USO .....	5
2.7	CASO DE USO EXPANDIDO .....	7
2.8	DIAGRAMA DE CLASSE .....	10
2.9	TESTES DO SOFTWARE/APLICATIVO/SOLUÇÃO .....	11
	CT001 – REALIZAR ACESSO AO UAIFUEL .....	11
	CT002 – CADASTRO DE MOTORISTA .....	11
	CT003 – CADASTRO DE POSTO .....	12
	CT004 – CADASTRO DE VEÍCULO .....	12
	CT005 – CADASTRO DE COMBUSTÍVEL .....	13
	CT006 – E D I Ç Ã O DE CADASTRO DE MOTORISTA .....	13
	CT007 – E D I Ç Ã O NO CADASTRO DE POSTO .....	14
	CT008 – E D I Ç Ã O NO CADASTRO DE VEÍCULO .....	14
	CT009 – E D I Ç Ã O NO CADASTRO DE COMBUSTÍVEL .....	14
	CT010 – SIMULAÇÃO DA REALIZAÇÃO DE UM ABASTECIMENTO .....	15
	CT011 – EXIBIÇÃO DE RELATÓRIO DE ABASTECIMENTOS .....	16
3	PROJETO DE INTERFACE GRÁFICA (MODELO NÃO FUNCIONAL) .....	18
3.1	MODELO NÃO FUNCIONAL .....	18
3.2	INTERNACIONALIZAÇÃO DO SOFTWARE .....	23
4	PROJETO DE BANCO DE DADOS .....	24
4.1	DIAGRAMA DE ENTIDADE E RELACIONAMENTO (DER) .....	24
4.2	DIAGRAMA DE BANCO DE DADOS .....	25
4.3	QUADRO RESUMO DOS OBJETOS DE BANCO DE DADOS .....	25
4.4	MAPEAMENTO PARA O MODELO LÓGICO (MODELO RELACIONAL) .....	25
4.5	DICIONÁRIO DE DADOS .....	26
4.6	SCRIPT DE BANCO DE DADOS .....	28
4.7	INSERÇÃO NO BANCO DE DADOS .....	32
4.8	VIEWS .....	37
7.1	PROCEDURES .....	39
8	CONSIDERAÇÕES FINAIS .....	47
8.1	DIFICULDADES ENCONTRADAS .....	47
8.2	SUGESTÕES DE MELHORIA .....	47
9	REFERÊNCIAS .....	47

# 1 Introdução

**Título do Projeto:** UaiFuel

**Nome do Grupo:** UaiFuel Inter2

Nr.	Aluno(s) envolvido(s)	E-mail(s)
01	Éder Marques	earmarques@gmail.com
02	Elias Montovani	eliasmt6@gmail.com
03	Estevan Navarro	estevan_lago@hotmail.com
04	Pedro H. Coimbra	pedro.alves9@fatec.sp.gov.br

## 1.1 Objetivo

- Desenvolver uma aplicação web para registro de abastecimentos para postos de combustíveis.
- Os registros devem conter informações do veículo abastecido, do motorista que efetuou o abastecimento, combustíveis, bem como do posto de combustível em que a operação de abastecimento foi realizada.

## 1.2 Justificativa

O presente projeto justifica-se pela ausência ou necessidade de controle nos abastecimentos dos veículos de empresas do ramo de transporte.

As fraudes e os desperdícios são, certamente, dois dos principais obstáculos para os empreendedores quando o assunto é abastecimento de frotas. Infelizmente, os desvios são mais frequentes do que se possa imaginar. Além disso, algumas situações levam a práticas ilícitas – abastecimentos em postos irregulares, ou ainda adulteração nas bombas de combustíveis, tanto nos valores quanto na quantia de litros.

De acordo com o Ipem (Instituto de Pesos e Medida), as fraudes em média reduzem a quantidade de combustível injetado entre 10% e 12%, fazendo com que se perca 2 litros a cada 20 indicados na bomba. Como exemplo, um posto que vende 300 mil litros por mês, pode faturar a mais R\$ 100 mil com a fraude. Outro ponto é o abastecimento de forma irregular, onde o motorista/funcionário costuma abastecer em horários não programados pela empresa, durante os finais de semana e horário pós expedientes, sujeito a

toda sorte de trambiques. Há também o roubo de combustível onde existem casos na qual os motoristas/funcionários colaboram com a situação.

Este projeto é um protótipo embrionário de um sistema de gestão de abastecimento que ligará as duas partes interessadas, o transportador e o posto de combustível. O sistema de gestão oferecerá relatórios para ambos os lados, mas principalmente ao transportador, a fim de que possa conhecer o desempenho de sua frota em termos de consumo por veículo, por motorista e em função de rotas, modelos dos veículos e tipos de operações.

Neste primeiro momento, o projeto trata o lado do posto, abordando a política de descontos às empresas credenciadas.

## 2 Análise do Sistema Proposto

### 2.1 Escopo do Sistema

O projeto se sustenta em três entidades fundamentais, a saber: Motorista, Veículo e Posto de combustível. Todos devem estar devidamente credenciados. As informações pertinentes ao motorista são o nome, senha, status, cpf, e-mails e quanto de crédito dispõe para abastecer. Para o posto, além de nome, senha, status e e-mails, há também o endereço do posto e telefones.

O motorista da empresa conveniada chega a um posto de combustível parceiro, com preços negociados, para realizar o abastecimento. O caixa do posto consulta no sistema se o binômio motorista e veículo estão na base de dados. Se a verificação for positiva o caixa autoriza o frentista a abastecer o veículo. Terminado o abastecimento o frentista entrega ao caixa as informações como litragens e os combustíveis. O caixa por sua vez alimenta as informações no sistema web e registra o abastecimento. As informações pertinentes ao abastecimento de combustíveis são o motorista que realizou o abastecimento, o posto de combustível, o veículo abastecido, a data e hora em que foi realizado, o status em relação ao pagamento, os combustíveis abastecidos com seus respectivos valores e quantidade de litros, o valor total do abastecimento, e depois de efetuado o pagamento, o cupom fiscal.

Um ponto relevante à regra de negócio é o fato de um abastecimento pode ser composto de mais de um tipo de combustível. A título de exemplos, pode-se colocar cinco litros de gasolina aditivada e completar o tanque com gasolina comum, ou um pouco de gasolina num carro a álcool para aumentar a octanagem, ou ainda acrescentar a um abastecimento de diesel um galão de arla.

### 2.2 Não Escopo do Sistema

Por se tratar de um protótipo embrionário de um sistema de gestão de abastecimento maior, o presente projeto não fornecerá os relatórios de desempenho mencionados na justificativa, como:

- Relatórios de médias de consumo de motorista e veículos, em função de rotas, modelos e tipos de operações./chrome

### 2.3 Requisitos do Sistema

Nr.	Requisito
01	O sistema deve permitir o cadastro (inclusão, alteração e remoção) de combustíveis.
02	O sistema deve permitir o cadastro (inclusão, alteração e remoção) de veículos.
03	O sistema deve permitir o cadastro (inclusão, alteração e remoção) de motoristas.
04	O sistema deve permitir o cadastro (inclusão, alteração e remoção) de postos.

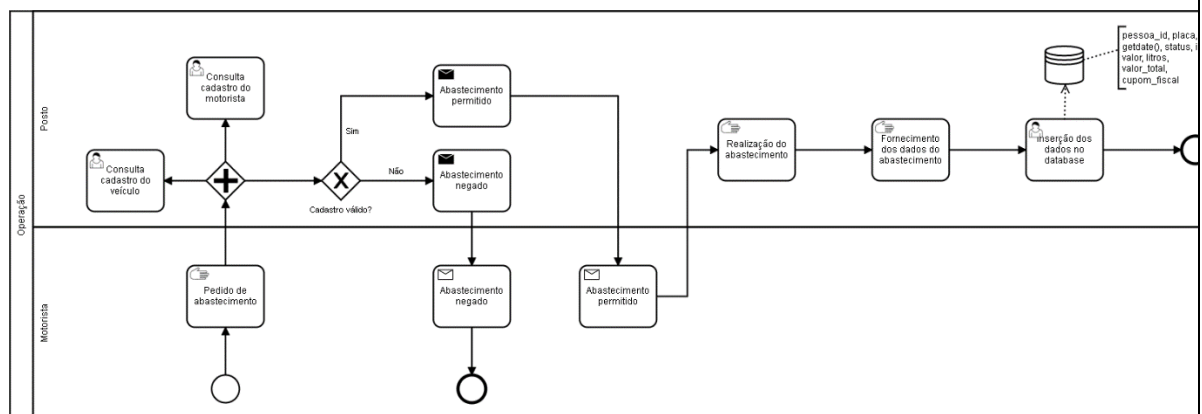
05	O sistema deve permitir o registro de abastecimentos de combustíveis.
06	O sistema deve ser capaz de realizar pesquisas de abastecimentos filtrados por período, status, combustíveis, por motorista, posto e pela placa do veículo.

## 2.4 Glossário

Termo	Comentário
Posto	Estabelecimento comercial cuja atividade principal é a venda de combustíveis
Caixa	é o funcionário do posto de combustível encarregado de operar o caixa.
Motorista	funcionário da transportadora condutor do veículo.
Abastecimento	Materialização do propósito de existir do projeto. Pode ser de vários combustíveis.
Combustível	São os itens vendidos pelo posto: álcool, gasolina, gasolina aditivada, diesel S500, diesel S10, arla, aditivo, etc.
Arla	Aditivo para motores a diesel para redução de poluentes

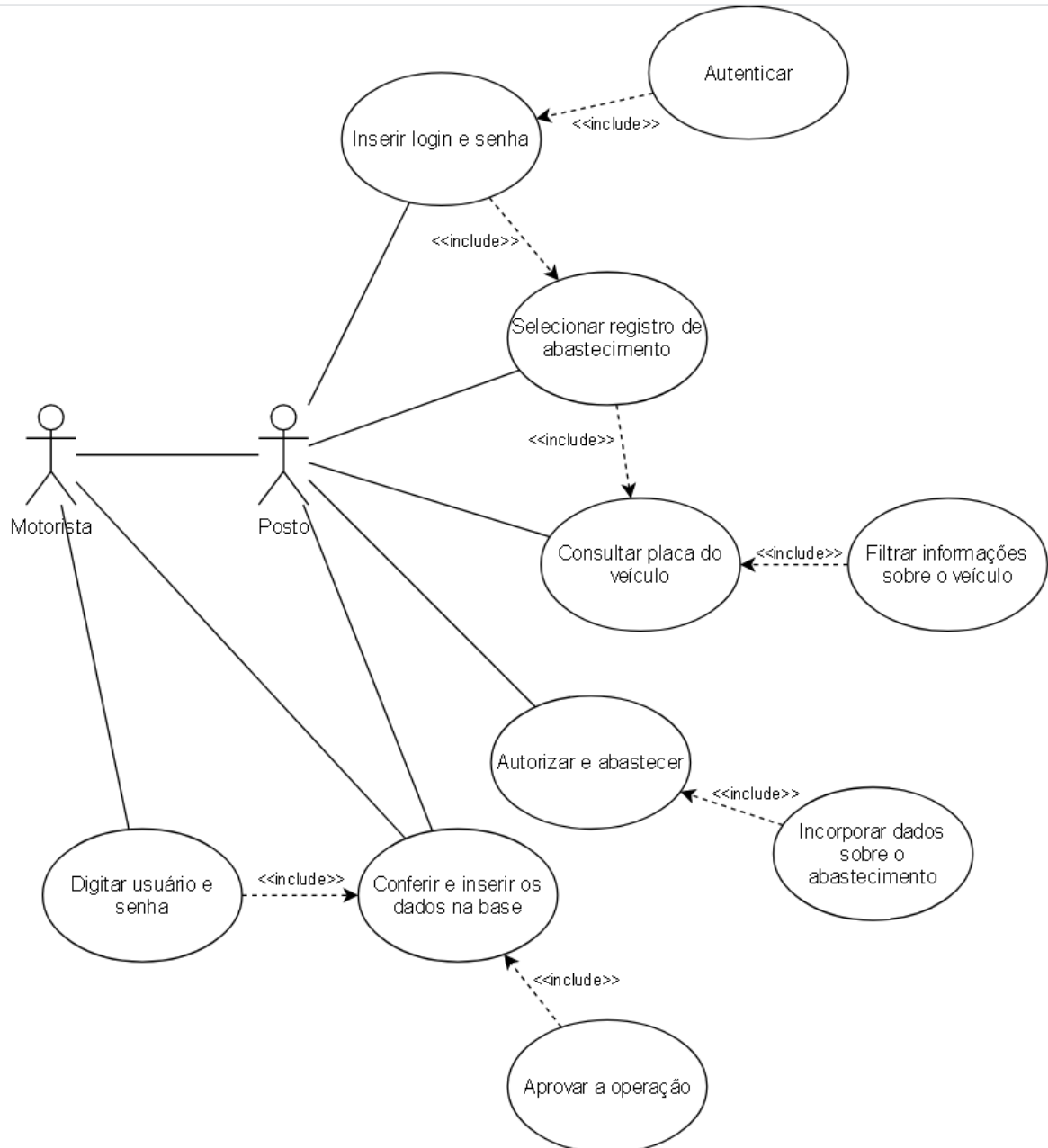
## 2.5 Mapeamento de Processos

### Diagrama BPMN – Abastecimento

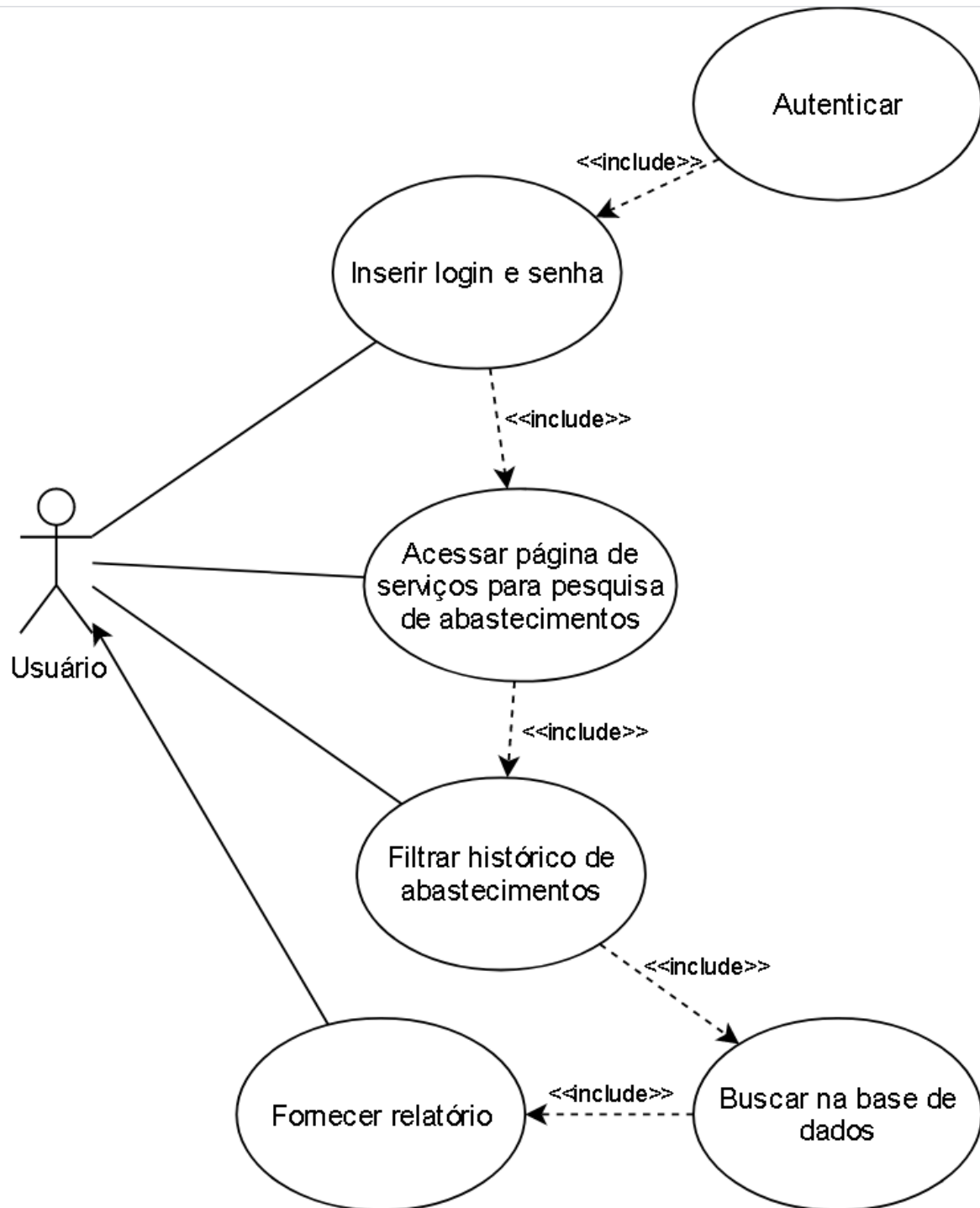


## 2.6 Diagrama de Caso de Uso

### Caso de Uso – Abastecer



Caso de Uso – Pesquisar abastecimentos





## 2.7 Caso de Uso Expandido

**Caso de Uso:** Abastecer

**Ator(es):** Motorista (iniciador), Caixa do posto, Frentista

**Propósito:** Realizar de forma controlada o abastecimento do veículo num certo posto por um determinado motorista até a quantia de crédito especificada.

**Tipo:** Primário e essencial

**Descrição:** Motorista chega no posto e pede para abastecer. Caixa verifica se o motorista e a placa do veículo estão cadastrados e autoriza o Frentista a abastecer. Caixa preenche formulário com dados do abastecimento. Motorista digita código e senha. Motorista é autenticado, registra o abastecimento e Motorista segue viagem.

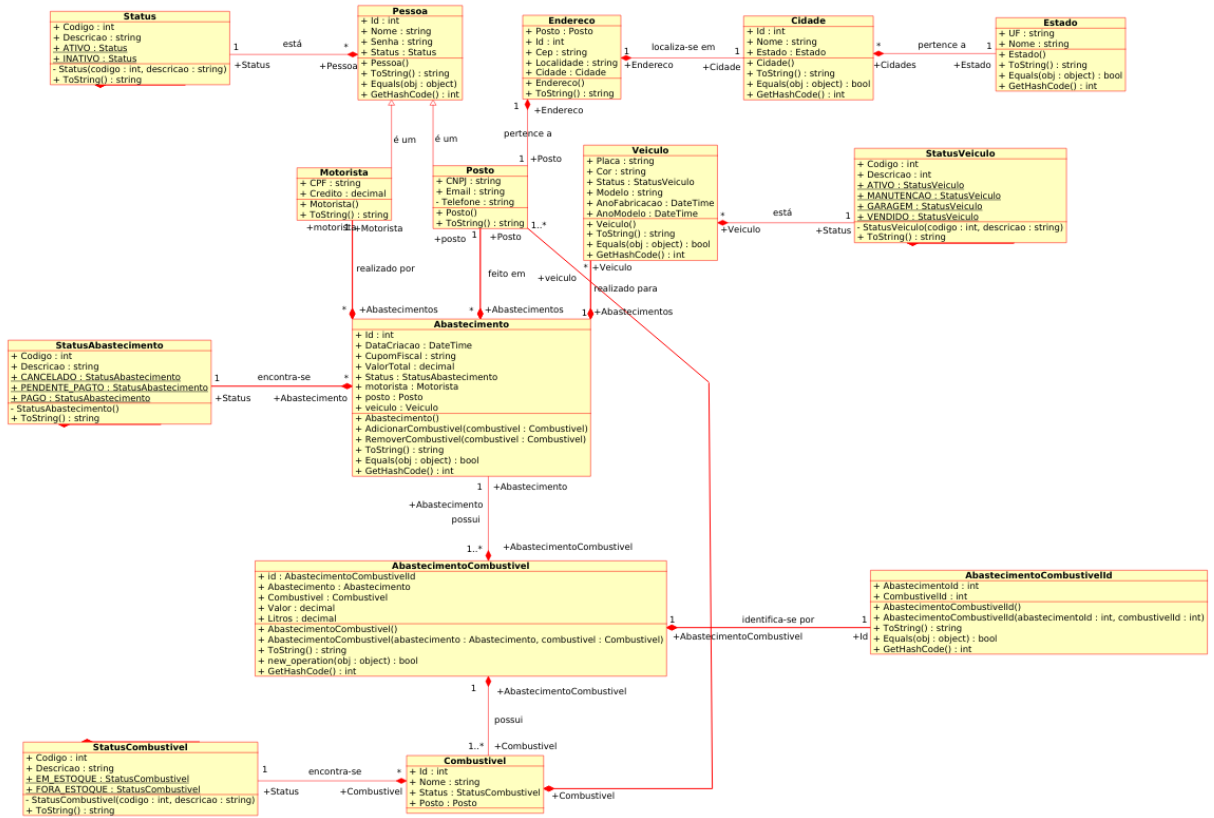
### Sequência Típica de eventos.

Ação do Ator	Resposta do Sistema
1. O Motorista deve informar a placa do Veículo e quais combustíveis deseja abastecer.	
2. Caixa deve abrir a tela inicial do sistema e fornece seu login e senha.	3. O sistema faz a autenticação do Caixa e abre a tela inicial.
4. Caixa seleciona no menu o serviço de registrar abastecimento.	5. Abre a tela de registro de abastecimento.
6 Caixa pesquisa pela placa do veículo.	7. Busca na base todas as placas de veículo e carrega as informações básicas do veículo como modelo e cor.
8. Caixa autoriza o Frentista abastecer o veículo.	
9. Frentista abastece e retorna ao Caixa informando os tipos de combustíveis e quantos litros foram abastecidos.	
10. Caixa digita todos os dados do abastecimento.	
11. Caixa pede ao Motorista que confira as informações e que faça autenticação, digitando o código (id) e senha do motorista.	
12. Motorista digita seu código e sua senha.	
13. Caixa submete formulário.	14. Confere os dados do motorista e registra o abastecimento.
	15. Retorna à confirmação da operação.

16. Caixa recebe aprovação do sistema e informa o Motorista do sucesso da transação.											
17. Motorista segue viagem.											
<p><b>Caso de Uso:</b> Pesquisar abastecimentos</p> <p><b>Ator(es):</b> Usuário do sistema (iniciador).</p> <p><b>Propósito:</b> Realizados os abastecimentos faz-se necessário visualizá-los. Esta visualização deve se dar através de uma pesquisa nos abastecimentos. Para tanto, o sistema deve disponibilizar diversos filtros ou critérios para os abastecimentos. Devemos poder filtrar os abastecimentos em função de motorista, veículo, posto de combustível, por combustível abastecido, pelo status do abastecimento, e por período, com data de início e data de fim.</p> <p><b>Tipo:</b> primário essencial</p> <p><b>Descrição:</b> Usuário do Posto acessa o sistema e define os critério para realizar a pesquisa de abastecimentos.</p>											
<p><b>Sequência Típica de eventos.</b></p> <table border="1"> <thead> <tr> <th>Ação do Ator</th><th>Resposta do Sistema</th></tr> </thead> <tbody> <tr> <td>1. Usuário acessa a tela inicial do sistema e fornece login e senha.</td><td>2. Faz a autenticação de login e senha.</td></tr> <tr> <td></td><td>3. Concede acesso ao sistema e a página de serviços com a funcionalidade de pesquisa de abastecimentos.</td></tr> <tr> <td>4. Usuário acessa a página de pesquisa de abastecimentos, configura os campos para fazer a filtragem desejada e define um período de abastecimentos – um intervalo de tempo com data inicial e data final. O conjunto de critérios de filtragem são: motorista, placa do veículo, posto de combustível, combustível, status do abastecimento e período (data inicial e final).</td><td>5. Busca todos os abastecimentos do período.</td></tr> <tr> <td></td><td>6. O sistema busca na base de dados todos os abastecimentos que atendem aos critérios configurados pelo usuário.</td></tr> </tbody> </table>		Ação do Ator	Resposta do Sistema	1. Usuário acessa a tela inicial do sistema e fornece login e senha.	2. Faz a autenticação de login e senha.		3. Concede acesso ao sistema e a página de serviços com a funcionalidade de pesquisa de abastecimentos.	4. Usuário acessa a página de pesquisa de abastecimentos, configura os campos para fazer a filtragem desejada e define um período de abastecimentos – um intervalo de tempo com data inicial e data final. O conjunto de critérios de filtragem são: motorista, placa do veículo, posto de combustível, combustível, status do abastecimento e período (data inicial e final).	5. Busca todos os abastecimentos do período.		6. O sistema busca na base de dados todos os abastecimentos que atendem aos critérios configurados pelo usuário.
Ação do Ator	Resposta do Sistema										
1. Usuário acessa a tela inicial do sistema e fornece login e senha.	2. Faz a autenticação de login e senha.										
	3. Concede acesso ao sistema e a página de serviços com a funcionalidade de pesquisa de abastecimentos.										
4. Usuário acessa a página de pesquisa de abastecimentos, configura os campos para fazer a filtragem desejada e define um período de abastecimentos – um intervalo de tempo com data inicial e data final. O conjunto de critérios de filtragem são: motorista, placa do veículo, posto de combustível, combustível, status do abastecimento e período (data inicial e final).	5. Busca todos os abastecimentos do período.										
	6. O sistema busca na base de dados todos os abastecimentos que atendem aos critérios configurados pelo usuário.										

	7. O sistema preenche uma tabela, localizada abaixo dos campos de pesquisa, com as informações dos abastecimentos retornados.

## 2.8 Diagrama de Classe



## 2.9 Testes do Software/Aplicativo/Solução

<b>CASO Nº</b>	<b>CT001 – Realizar acesso ao UaiFuel.</b>
<b>OBJETIVO DO TESTE</b>	Verificar se o usuário consegue efetuar o login.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Acessar a página de login do UaiFuel através do navegador;</li> <li>2. Inserção de usuário;</li> <li>3. Inserção de senha;</li> <li>4. Pressione o botão para realizar o processo de login.</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	Ao clicar no botão para realizar o login o usuário deve conseguir realizar acesso à primeira tela do UaiFuel, onde será possível navegar pelos cadastros e relatórios.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Se for apresentada um usuário incorreto, deverá apresentar a seguinte mensagem de que o usuário não está cadastrado, ou de login e senha inválidos.</i>

<b>CASO Nº</b>	<b>CT002 – Cadastro de motorista.</b>
<b>OBJETIVO DO TESTE</b>	Verificar se o usuário do sistema consegue cadastrar um motorista.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Clicar no ícone de cadastros de motorista no menu principal;</li> <li>2. Clicar no botão para cadastrar um novo motorista;</li> <li>3. Inserir o nome do motorista;</li> <li>4. Inserir o CPF do motorista;</li> <li>5. Inserir a senha que será utilizada pelo motorista;</li> <li>6. Inserir um ou mais endereços de e-mail do motorista;</li> <li>7. Inserir crédito disponível para o motorista;</li> <li>8. Selecionar o status do cadastro;</li> <li>9. Clicar no botão para finalizar o cadastro.</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	O usuário deve conseguir cadastrar um motorista e receberá uma mensagem confirmando o cadastro.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Se algum campo for inválido este ficará com a cor vermelha e com uma pequena mensagem dizendo "&lt;Nome do campo&gt; inválido &lt;motivo do erro&gt;".</i>

<b>CASO Nº</b>	<b>CT003 – Cadastro de posto.</b>
<b>OBJETIVO DO TESTE</b>	Verificar se o usuário do sistema consegue cadastrar um posto.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Clicar no ícone de cadastros de postos no menu principal;</li> <li>2. Clicar no botão para cadastrar um novo posto;</li> <li>3. Inserir o nome do posto;</li> <li>4. Inserir o CNPJ do posto;</li> <li>5. Inserir a senha que será utilizada pelo posto;</li> <li>6. Inserir um ou mais endereços de e-mail do posto;</li> <li>7. Inserir telefone do posto;</li> <li>8. Inserir um ou mais endereços do posto;</li> <li>9. Selecionar o status do cadastro;</li> <li>10. Clicar no botão para finalizar o cadastro.</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	O usuário deve conseguir cadastrar um posto e receberá uma mensagem confirmando o cadastro.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Se algum campo for inválido este ficará com a cor vermelha e com uma pequena mensagem dizendo "&lt;Nome do campo&gt; inválido &lt;motivo do erro&gt;".</i>

<b>CASO Nº</b>	<b>CT004 – Cadastro de veículo.</b>
<b>OBJETIVO DO TESTE</b>	Verificar se o usuário do sistema consegue cadastrar um veículo.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Clicar no ícone de cadastros de veículo no menu principal;</li> <li>2. Clicar no botão para cadastrar um novo veículo;</li> <li>3. Inserir a placa do veículo;</li> <li>4. Inserir o modelo do veículo;</li> <li>5. Inserir a cor do veículo;</li> <li>6. Inserir o ano de fabricação do veículo;</li> <li>7. Inserir o ano do modelo do veículo;</li> <li>8. Selecionar o status do cadastro;</li> <li>9. Clicar no botão para finalizar o cadastro.</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	O usuário deve conseguir cadastrar um veículo e receberá uma mensagem confirmando o cadastro.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Se algum campo for inválido este ficará com a cor vermelha e com uma pequena mensagem dizendo "&lt;Nome do campo&gt; inválido &lt;motivo do erro&gt;".</i>

<b>CASO Nº</b>	<b>CT005 – Cadastro de combustível.</b>
<b>OBJETIVO DO TESTE</b>	Verificar se o usuário do sistema consegue cadastrar um combustível.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Clicar no ícone de cadastros de combustíveis no menu principal;</li> <li>2. Clicar no botão para cadastrar um novo combustível;</li> <li>3. Inserir o nome do combustível;</li> <li>4. Selecionar o status do cadastro;</li> <li>5. Clicar no botão para finalizar o cadastro.</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	O usuário deve conseguir cadastrar um combustível e receberá uma mensagem confirmando o cadastro.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Se algum campo for inválido este ficará com a cor vermelha e com uma pequena mensagem dizendo "&lt;Nome do campo&gt; inválido &lt;motivo do erro&gt;".</i>

<b>CASO Nº</b>	<b>CT006 – Edição de cadastro de motorista.</b>
<b>OBJETIVO DO TESTE</b>	Verificar se o usuário do sistema consegue editar o cadastro de um motorista.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Clicar no ícone de cadastros de motorista no menu principal;</li> <li>2. Inserir o nome do motorista desejado na caixa de pesquisas para realização de filtragem;</li> <li>3. Clicar no botão para editar o cadastro do motorista desejado;</li> <li>4. Fazer alterações no nome do motorista;</li> <li>5. Fazer alterações no CPF do motorista;</li> <li>6. Fazer alterações na senha do motorista;</li> <li>7. Fazer alterações no(s) endereço(s) de e-mail do motorista;</li> <li>8. Fazer alterações no crédito disponível para o motorista;</li> <li>9. Fazer alterações no status do cadastro;</li> <li>10. Clicar no botão para salvar as alterações realizadas.</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	O usuário deve conseguir editar o cadastro do motorista e receberá uma mensagem confirmando a edição.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Se algum campo for inválido este ficará com a cor vermelha e com uma pequena mensagem dizendo "&lt;Nome do campo&gt; inválido &lt;motivo do erro&gt;".</i>

<b>CASO Nº</b>	<b>CT007 – Edição no cadastro de posto.</b>
<b>OBJETIVO DO TESTE</b>	Verificar se o usuário do sistema consegue editar o cadastro de um posto.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Clicar no ícone de cadastros de postos no menu principal;</li> <li>2. Inserir o nome de um posto desejado na caixa de pesquisas para realização de filtragem;</li> <li>3. Clicar no botão para editar o cadastro do motorista desejado;</li> <li>4. Fazer alterações no nome do posto;</li> <li>5. Fazer alterações no CNPJ do posto;</li> <li>6. Fazer alterações na senha utilizada pelo posto;</li> <li>7. Fazer alterações no(s) endereço(s) de e-mail do posto;</li> <li>8. Fazer alterações no(s) telefone(s) do posto;</li> <li>9. Fazer alteração no(s) endereços(s) do posto;</li> <li>10. Fazer alterações no status do cadastro;</li> <li>11. Clicar no botão para salvar as alterações realizadas.</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	O usuário deve conseguir editar o cadastro do posto e receberá uma mensagem confirmando a edição.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Se algum campo for inválido este ficará com a cor vermelha e com uma pequena mensagem dizendo "&lt;Nome do campo&gt; inválido &lt;motivo do erro&gt;".</i>

<b>CASO Nº</b>	<b>CT008 – Edição no cadastro de veículo.</b>
<b>OBJETIVO DO TESTE</b>	Verificar se o usuário do sistema consegue editar o cadastro de um veículo.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Clicar no ícone de cadastros de veículo no menu principal;</li> <li>2. Inserir o nome de um veículo desejado na caixa de pesquisas para realização de filtragem;</li> <li>3. Clicar no botão para editar o cadastro do veículo desejado;</li> <li>4. Fazer alterações na placa do veículo;</li> <li>5. Fazer alterações no modelo do veículo;</li> <li>6. Fazer alterações na cor do veículo;</li> <li>7. Fazer alterações no status do cadastro</li> <li>8. Clicar no botão para salvar as alterações realizadas.</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	O usuário deve conseguir editar o cadastro do veículo e receberá uma mensagem confirmando a edição.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Se algum campo for inválido este ficará com a cor vermelha e com uma pequena mensagem dizendo "&lt;Nome do campo&gt; inválido &lt;motivo do erro&gt;".</i>

<b>CASO Nº</b>	<b>CT009 – Edição no cadastro de combustível.</b>
----------------	---



<b>OBJETIVO DO TESTE</b>	Verificar se o usuário do sistema consegue editar o cadastro de um combustível.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Clicar no ícone de cadastros de combustíveis no menu principal;</li> <li>2. Inserir o nome de um combustível desejado na caixa de pesquisas para realização de filtragem;</li> <li>3. Clicar no botão para editar o cadastro do combustível desejado;</li> <li>4. Fazer alterações no nome do combustível;</li> <li>5. Fazer alterações no status do cadastro;</li> <li>6. Clicar no botão para salvar as alterações realizadas.</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	O usuário deve conseguir editar o cadastro do combustível e receberá uma mensagem confirmando a edição.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Se algum campo for inválido este ficará com a cor vermelha e com uma pequena mensagem dizendo "&lt;Nome do campo&gt; inválido &lt;motivo do erro&gt;".</i>
<b>CASO Nº</b>	CT010 – Simulação da realização de um abastecimento.

<b>OBJETIVO DO TESTE</b>	Verificar se o usuário do sistema consegue registrar um abastecimento.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Clicar no ícone de abastecimentos no menu principal;</li> <li>2. Clicar no botão para realizar o cadastro do abastecimento;</li> <li>3. Seleção da placa do veículo abastecido para inserção das informações;</li> <li>4. Seleção do nome/CPF do motorista para inserção das informações;</li> <li>5. Inserção da senha do motorista para comprovação de autenticidade.</li> <li>6. Seleção do nome/CNPJ do posto de combustíveis para inserção das informações;</li> <li>7. Seleção dos ids dos combustíveis abastecidos;</li> <li>8. Inserir a litragem dos combustíveis abastecidos;</li> <li>9. Inserir o valor por litro dos combustíveis abastecidos;</li> <li>10. Inserir senha do usuário para validar o preenchimento;</li> <li>11. Clicar no botão para gravar e gerar o abastecimento;</li> <li>12. Verifique se as informações estão corretas;</li> <li>13. Realização do pagamento para geração de cupom fiscal;</li> <li>14. Alteração do status para "Pago";</li> <li>15. Clicar no botão para finalizar o abastecimento.</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	O usuário deve conseguir gerar o abastecimento e todos os dados devem ser exibidos na tela, em seguida o pagamento deve ser efetuado, aparecendo uma mensagem confirmando a realização do abastecimento.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Se algum campo for inválido este ficará com a cor vermelha e com uma pequena mensagem dizendo "&lt;Nome do campo&gt; inválido &lt;motivo do erro&gt;".</i>
<b>CASO Nº</b>	CT011 – Exibição de relatório de abastecimentos.

<b>OBJETIVO DO TESTE</b>	Verificar se o usuário do sistema consegue realizar uma filtragem dos abastecimentos para exibição de relatório.
<b>PASSOS</b>	<ol style="list-style-type: none"> <li>1. Clicar no ícone de abastecimentos no menu principal;</li> <li>2. Clicar no botão para criar um relatório e selecionar as informações pertinentes;</li> </ol> <p><b>Filtragem</b></p> <ol style="list-style-type: none"> <li>3. Seleção da placa do veículo para filtrar dentre todos os veículos abastecidos;</li> <li>4. Seleção do nome/CPF do motorista para filtrar dentre todos os motoristas que abasteceram;</li> <li>5. Seleção do nome/CNPJ do posto de combustíveis para filtrar os abastecimentos dentre todos os postos cadastrados no sistema;</li> <li>6. Seleção de um id/nome de um combustível para filtrar por combustível abastecido;</li> <li>7. Seleção da litragem dos combustíveis abastecidos para filtrar pelo montante de combustível abastecido;</li> <li>8. Seleção do valor total do abastecimento para filtrar pelo valor total do abastecimento;</li> <li>9. Seleção de um período para selecionar os abastecimentos realizados dentro de um determinado intervalo de tempo;</li> <li>10. Inserir senha do usuário para validar o preenchimento;</li> <li>11. Clicar no botão para gerar o relatório dos abastecimentos com as informações requeridas;</li> </ol>
<b>CRITÉRIOS DE ÊXITO</b>	O usuário deve conseguir gerar o relatório de todos os dados requeridos, estes devem ser exibidos na tela, em seguida deve-se gerar o relatório com sucesso, aparecendo uma mensagem confirmando a geração do relatório.
<b>CRITÉRIOS DE TRATAMENTO DE ERROS</b>	<i>Caso alguma operação sofra algum tipo de erro, aparecerá uma mensagem "Não foi possível filtrar as informações pois &lt;motivo do incidente&gt;"</i>

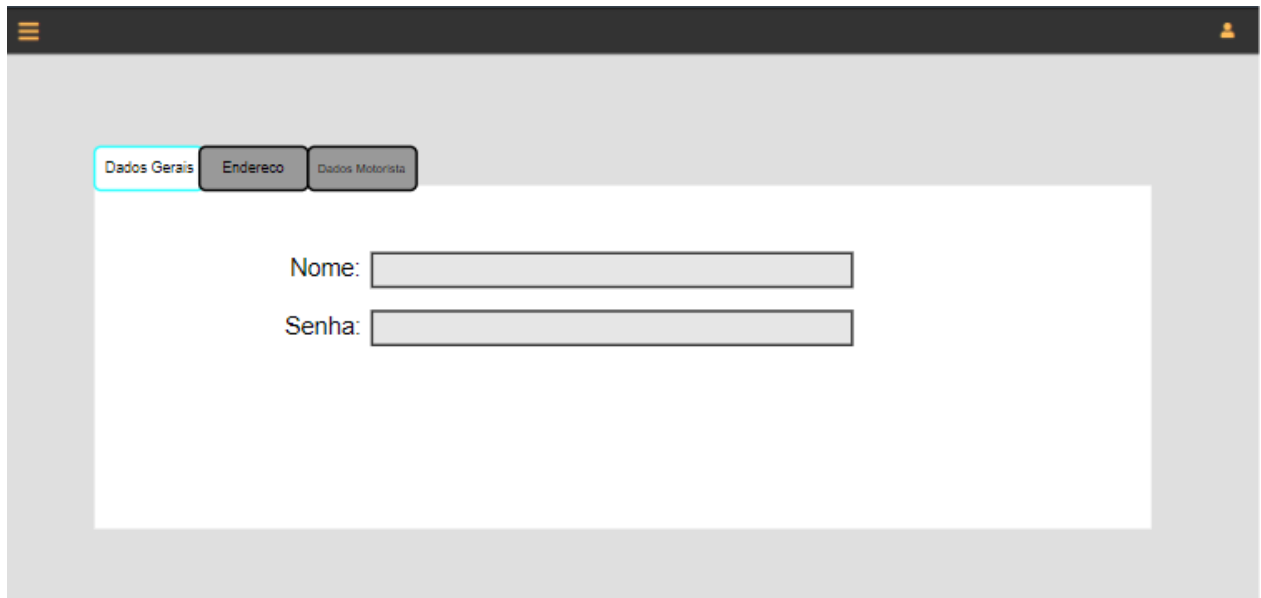
### 3 Projeto de Interface Gráfica (Modelo não Funcional)

#### 3.1 Modelo Não Funcional

##### Interface 01: Cadastro de Motorista

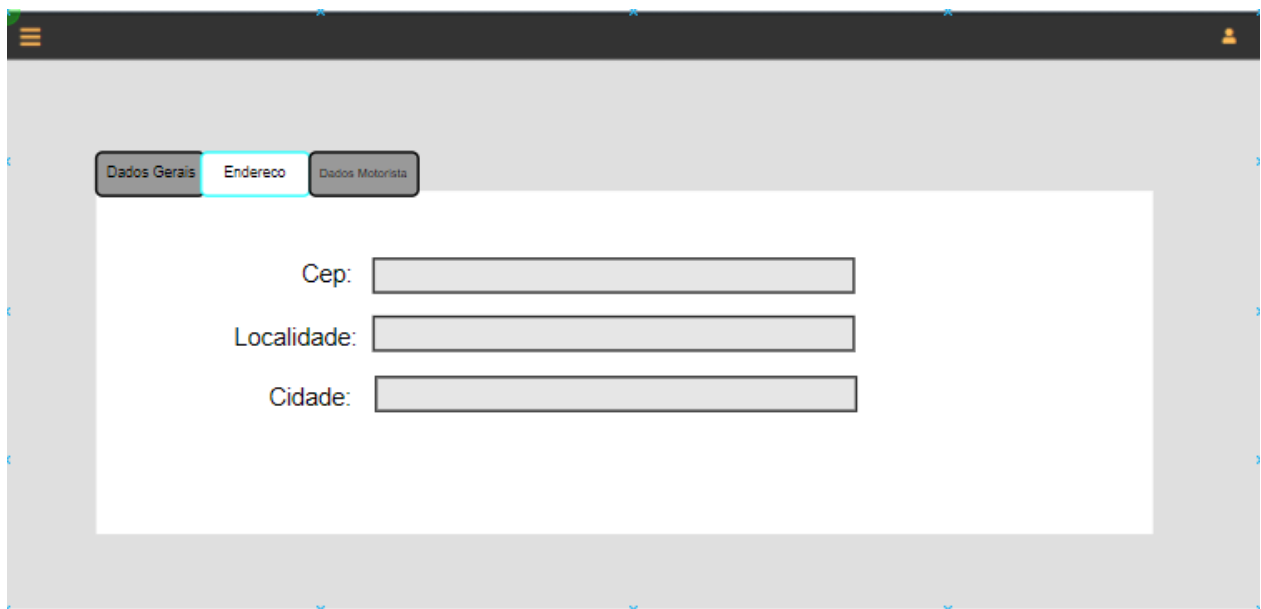
**Descrição:** As seguintes imagens são responsáveis pelo cadastro de motorista no sistema.

**Nome:** FrmCadMotorista



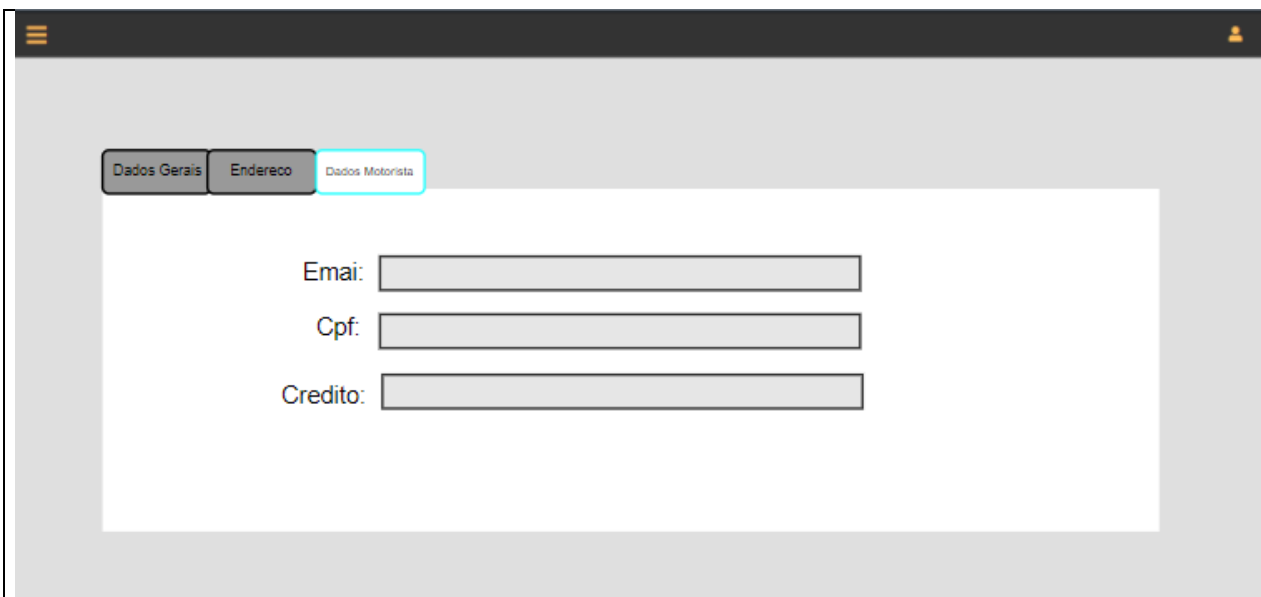
The screenshot shows a web application interface for driver registration. At the top, there is a dark header bar with a hamburger menu icon on the left and a user profile icon on the right. Below the header, there are three tabs: 'Dados Gerais' (highlighted with a blue border), 'Endereco', and 'Dados Motorista'. The main content area is a light gray rectangle containing two text input fields. The first field is labeled 'Nome:' and the second is labeled 'Senha:'.

Cadastro de motorista – Aba de Dados Gerais.



The screenshot shows the same web application interface, but with the 'Endereco' tab highlighted with a blue border. The main content area is a light gray rectangle containing three text input fields. The first field is labeled 'Cep:', the second is labeled 'Localidade:', and the third is labeled 'Cidade:'.

Cadastro de motorista – Aba de Endereço.



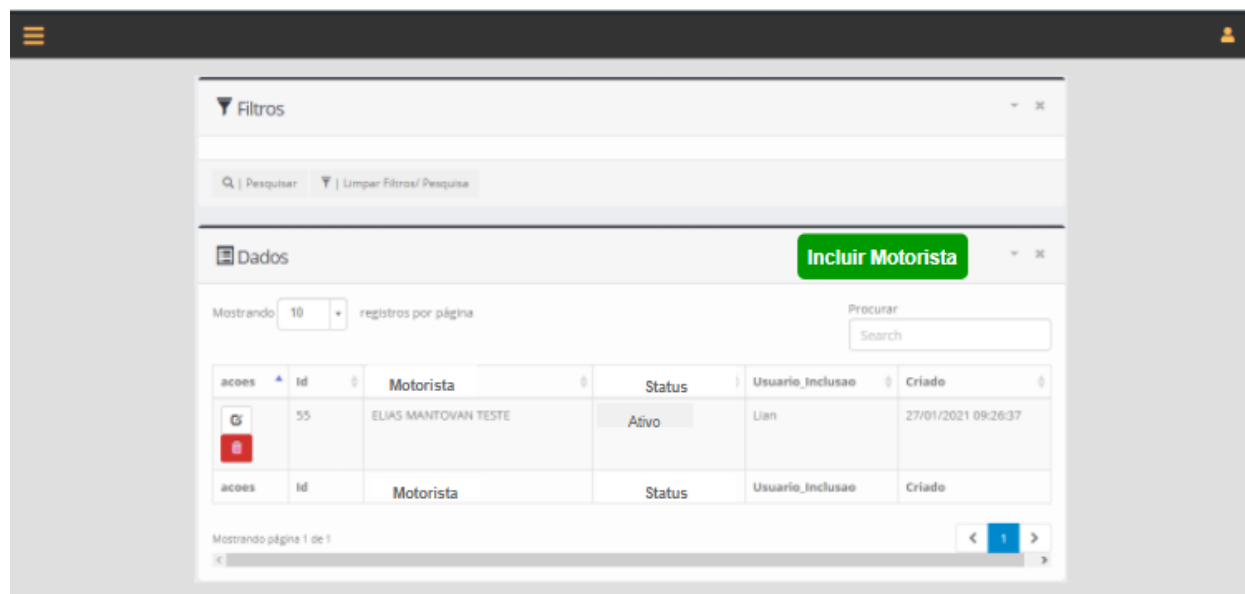
The screenshot shows a web interface with a dark header containing a menu icon and a user profile icon. Below the header, there are three tabs: 'Dados Gerais', 'Endereco', and 'Dados Motorista' (which is highlighted). The 'Dados Motorista' tab contains a white form with three input fields labeled 'Email:', 'Cpf:', and 'Credito:'.

Cadastro de motorista – Aba de Dados do Motorista.



## Interface 02: Index Motoristas

**Descrição:** Responsável pela exibição dos motoristas cadastrados no sistema.

**Nome:** FrmVisualiMotoristas



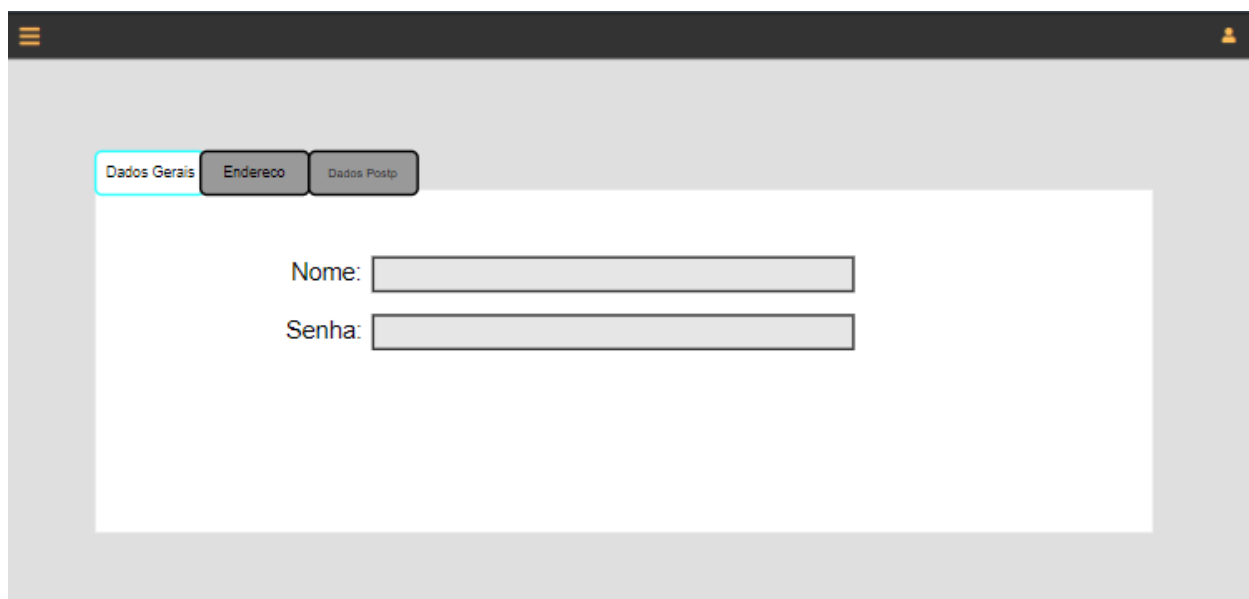
The screenshot displays a web interface for managing drivers. It features a 'Filtros' (Filters) section at the top with a search bar and a 'Limpar Filtros/Pesquisa' button. Below this is a 'Dados' section with a green 'Incluir Motorista' button. The main area contains a table with columns for 'acoes', 'Id', 'Motorista', 'Status', 'Usuario\_Inclusao', and 'Criado'. The table shows one driver: ELIAS MANTOVAN TESTE, with status 'Ativo', created by 'Lian' on 27/01/2021 09:26:37. The interface also includes pagination controls at the bottom.

acoes	Id	Motorista	Status	Usuario_Inclusao	Criado
 	55	ELIAS MANTOVAN TESTE	Ativo	Lian	27/01/2021 09:26:37

### Interface 03: Cadastro de Postos

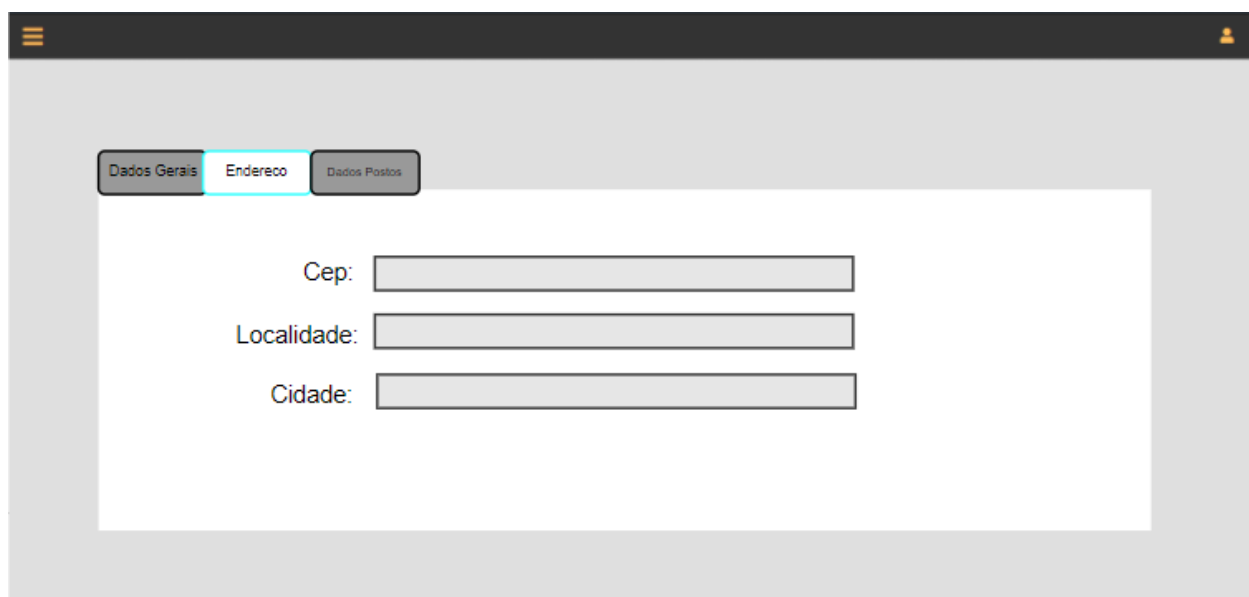
**Descrição:** As seguintes imagens são responsáveis pelo cadastro dos postos de combustíveis no sistema.

**Nome:** FrmCadPostos



The screenshot shows a web application interface with a dark header bar containing a menu icon on the left and a user profile icon on the right. Below the header, there are three tabs: 'Dados Gerais' (highlighted with a blue border), 'Endereco', and 'Dados Postos'. The main content area is white and contains two text input fields. The first field is labeled 'Nome:' and the second is labeled 'Senha:'.

Cadastro dos postos – Aba de Dados Gerais.



The screenshot shows the same web application interface, but with the 'Endereco' tab highlighted with a blue border. The main content area is white and contains three text input fields. The first field is labeled 'Cep:', the second is labeled 'Localidade:', and the third is labeled 'Cidade:'.

Cadastro dos postos – Aba de Endereço.

The screenshot shows a web application interface with a dark header containing a menu icon and a user profile icon. Below the header, there are three tabs: 'Dados Gerais', 'Endereco', and 'Dados Posto', with the last one being active. The main content area is a white box containing three input fields labeled 'Cnpj:', 'Email:', and 'Telefone:'.



Cadastro dos postos – Aba de Dados do Posto.

#### Interface 04: Index Postos

**Descrição:** Responsável pela exibição dos postos de combustíveis cadastrados no sistema.

**Nome:** FrmVisualiPostos

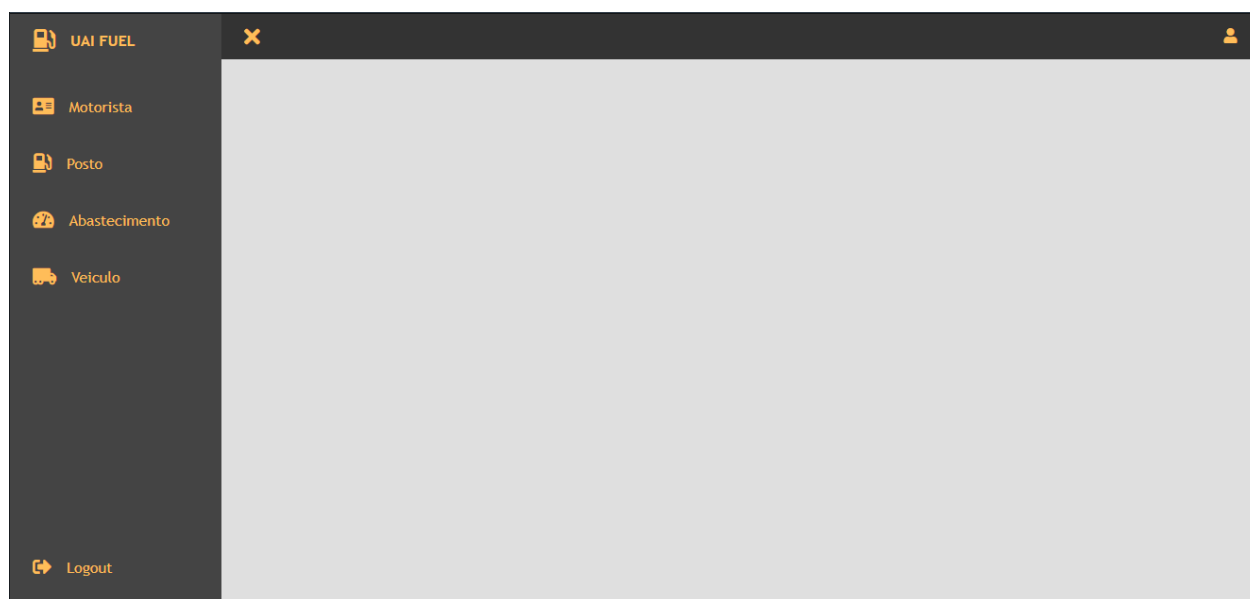
The screenshot displays the 'Index Postos' interface. At the top, there's a 'Filtros' section with a search bar and a 'Pesquisar' button. Below this is a 'Dados' section with a green 'Incluir Posto' button. The main part of the interface is a table with columns: 'acoes', 'Id', 'Posto', 'Status', 'Usuario\_Inclusao', and 'Criado'. The first row shows a record with Id 35, Posto 'Posto Postinho', Status 'Ativo', Usuario\_Inclusao 'Lian', and Criado '27/01/2021 09:26:37'. Below the table, there's a pagination bar showing 'Mostrando página 1 de 1'.

acoes	Id	Posto	Status	Usuario_Inclusao	Criado
 	35	Posto Postinho	Ativo	Lian	27/01/2021 09:26:37
acoes	Id	Posto	Status	Usuario_Inclusao	Criado

**Interface 05:** Tela inicial do sistema

**Descrição:** Quando o sistema for acessado, essa será a primeira página que o usuário irá visualizar, usando-a para navegar pelo sistema.

**Nome:** FrmHome





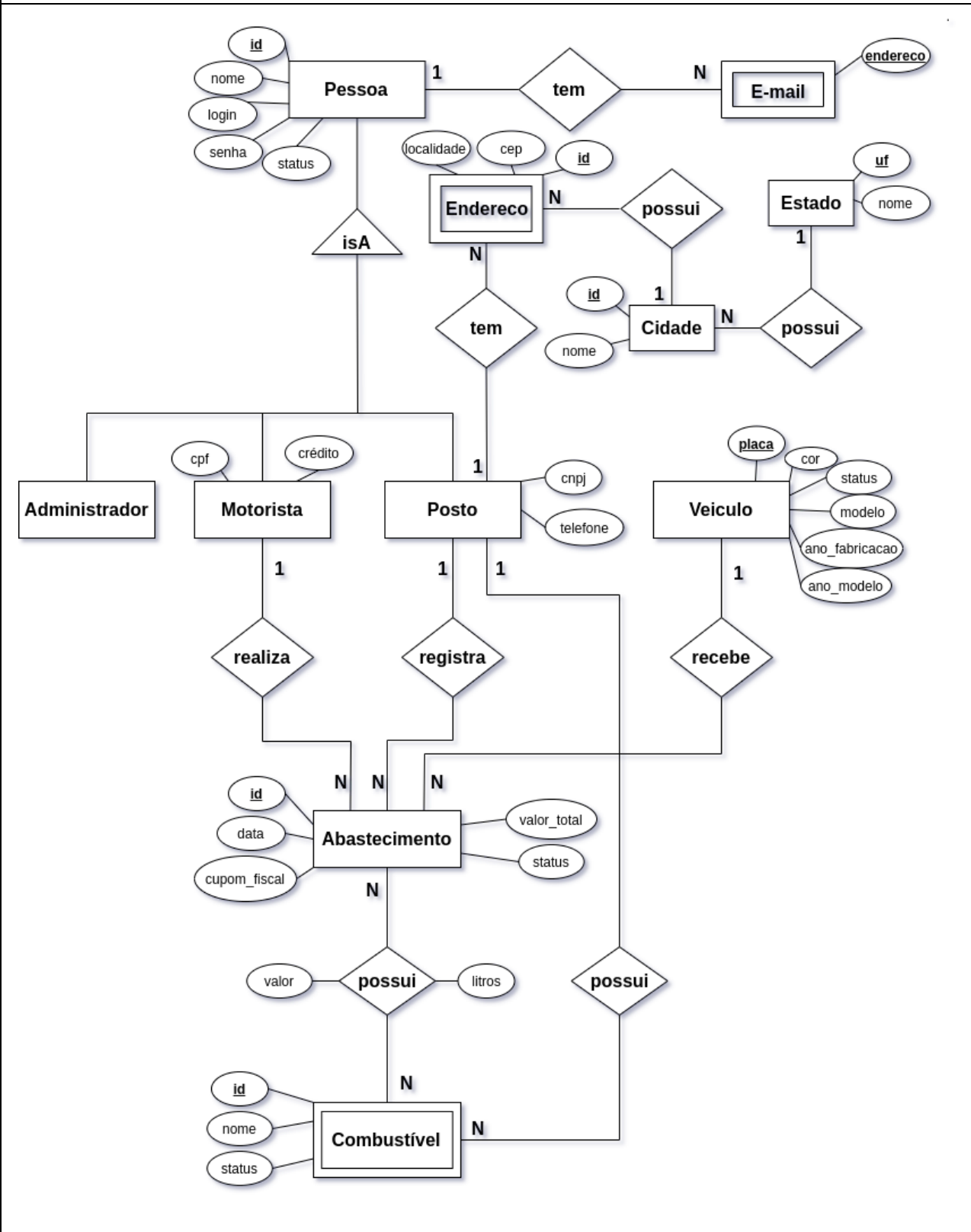
### 3.2 Internacionalização do software

<coloque aqui a correlação de todos os termos utilizados na interface gráfica nos idiomas Português e Inglês, conforme abaixo:>

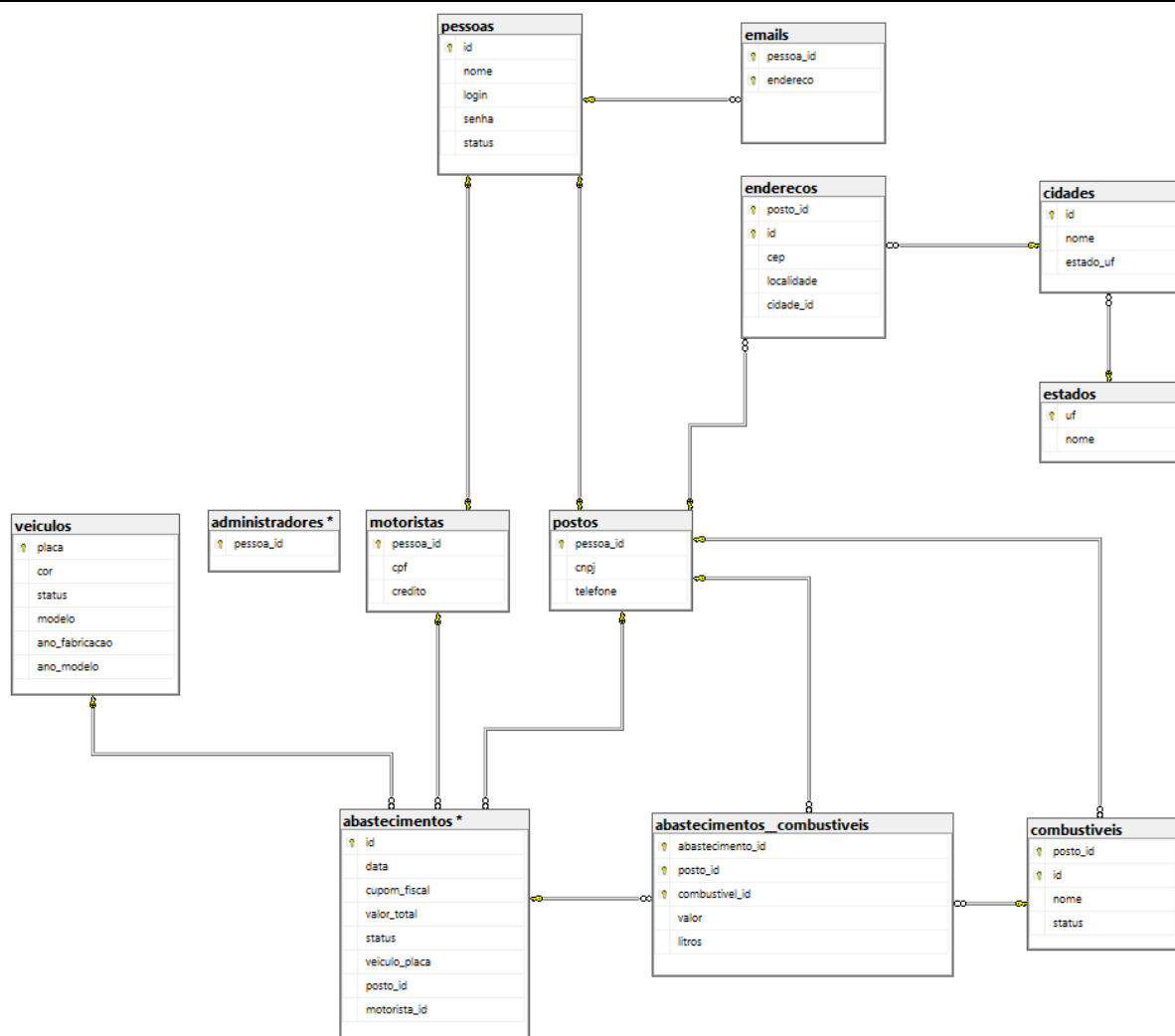
Termos utilizados para a Internacionalização	
Português	Inglês
Usuário	User
Senha	Password
Motoristas Cadastrados	Registered Drivers
Exibir	Show
Registros	Records
Filtro	Filters
Id	Id
Nome	Name
Status	Status
CPF	CPF
Crédito	Credit
Ação	Action
Anterior	Previous
Próximo	Next
Novo Motorista	New Driver
Dados Gerais	General Data
E-mails	E-mails
Login	Login
Guardar	Save
Endereço	Address
Postos Cadastrados	Registered Gas Station
CNPJ	CNPJ
Telefone	Phone
Novo Posto	New Gas Station
Salvar	Save
Localidade	Location
CEP	CEP
Estado	State
Cidade	City
Selecione	Select
Veículos Cadastrados	Registered Vehicles
Placa	Plate
Cor	Color
Modelo	Model
Ano Fabricação	Fabrication Year
Ano Modelo	Model Year
Novo Veículo	New Vehicles
Combustíveis Cadastrados	Registered Fuel
Abastecimentos Registrados	Registered Car Fuel
Novo Abastecimento	New Car Fuel
Valor	Value
Adicionar	Add
Efetuar Abastecimento	Make Car Fuel

## 4 Projeto de Banco de Dados

### 4.1 Diagrama de Entidade e Relacionamento (DER)



## 4.2 Diagrama de Banco de Dados



### 4.3 Quadro Resumo dos Objetos de Banco de Dados

Objeto de Banco de Dados	Quantidade
Tabelas	<12>
Views	<10>
Procedures	<preencher>

#### 4.4 Mapeamento para o Modelo Lógico (Modelo Relacional)

```
Veiculos (placa, cor, status, modelo, ano_fabricacao, ano_modelo)
Estados(uf, nome)
Cidades(id, nome, #estado_uf)
Pessoas (id, nome, login, senha, status)
```

Postos(#pessoa\_id, cnpj, telefone, endereco)  
 Motoristas (#pessoa\_id, cpf, credito)  
 Administradores (#pessoa\_id)  
 Emails (#pessoa\_id, endereco)  
 Enderecos (#posto\_id, id, cep, localidade, #cidade\_id)  
 Combustiveis (#posto\_id, id, descricao, status)  
 Abastecimentos (id, data, cupom\_fiscal, valor\_total, status, #veiculo\_placa, #posto\_id, #motorista\_id)  
 Abastecimentos\_\_Combustiveis (#abastecimento\_id, #posto\_id, #combustivel\_id, valor, litros)

#### 4.5 Dicionário de Dados

Tabela	Veiculos									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
placa	CHAR	7	-0-	-0-	Não	Sim	Não	Sim	-0-	-0-
cor	CHAR	20	-0-	-0-	Não	Não	Não	Não	-0-	-0-
status	INT	4	10	0	Não	Não	Não	Não	-0-	-0-
modelo	VARCHAR	20	-0-	-0-	Não	Não	Não	Não	-0-	-0-
ano_fabricacao	INT	4	10	0	Não	Não	Não	Não	-0-	-0-
ano_modelo	INT	4	10	0	Não	Não	Não	Não	-0-	-0-

Tabela	Estados									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
uf	CHAR	2	-0-	-0-	Não	Sim	Não	Sim	-0-	-0-
nome	VARCHAR	75	-0-	-0-	Não	Não	Não	Não	-0-	-0-

Tabela	Cidades									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
id	INT	4	10	0	Não	Sim	Não	Sim	-0-	-0-
nome	VARCHAR	120	-0-	-0-	Não	Não	Não	Não	-0-	-0-
estado_uf	CHAR	2	-0-	-0-	Não	Não	Sim	Não	estados	-0-

Tabela	Pessoas									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
id	INT	4	10	0	Não	Sim	Não	Sim	-o-	-o-
nome	VARCHAR	100	-o-	-o-	Não	Não	Não	Não	-o-	-o-
login	VARCHAR	100	-o-	-o-	Não	Não	Não	Sim	-o-	-o-
senha	VARCHAR	100	-o-	-o-	Não	Não	Não	Não	-o-	-o-
status	INT	4	10	0	Não	Não	Não	Não	-o-	-o-

Tabela	Postos									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
peessoa_id	INT	4	10	0	Não	Sim	Sim	Sim	peessoas	-o-
cnpj	VARCHAR	20	-o-	-o-	Não	Não	Não	Sim	-o-	-o-
telefone	VARCHAR	30	-o-	-o-	Não	Não	Não	Não	-o-	-o-

Tabela	Motoristas									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
peessoa_id	INT	4	10	0	Não	Sim	Sim	Sim	peessoas	-o-
cpf	VARCHAR	15	-o-	-o-	Não	Não	Não	Sim	-o-	-o-
credito	MONEY	8	19	4	Não	Não	Não	Não	-o-	-o-

Tabela	Administradores									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
peessoa_id	INT	4	10	0	Não	Sim	Sim	Sim	peessoas	-o-

Tabela	Emails									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
peessoa_id	INT	4	10	0	Não	Sim	Sim	Não	peessoas	-o-
endereco	VARCHAR	50	-o-	-o-	Não	Sim	Não	Não	-o-	-o-

Tabela	Enderecos									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
posto_id	INT	4	10	0	Não	Sim	Sim	Não	postos	-o-
id	INT	4	10	0	Não	Sim	Não	Não	-o-	-o-
cep	CHAR	9	-o-	-o-	Não	Não	Não	Não	-o-	-o-
localidade	VARCHAR	200	-o-	-o-	Não	Não	Não	Não	-o-	-o-
cidade_id	INT	4	10	0	Não	Não	Sim	Não	cidades	-o-

Tabela	Abastecimentos									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
id	INT	4	10	0	Não	Sim	Não	Sim	-o-	-o-
data	DATETIME	8	-o-	-o-	Não	Não	Não	Não	-o-	-o-
cupom_fiscal	VARCHAR	50	-o-	-o-	Sim	Não	Não	Sim	-o-	-o-
valor_total	MONEY	8	19	4	Sim	Não	Não	Não	-o-	>= 0
status	INT	4	10	0	Não	Não	Não	Não	-o-	(1, 2, 3)
veiculo_placa	CHAR	7	-o-	-o-	Não	Não	Sim	Não	veiculos	-o-
posto_id	INT	4	10	0	Não	Não	Sim	Não	postos	-o-
motorista_id	INT	4	10	0	Não	Não	Sim	Não	motoristas	-o-

Tabela	Abastecimentos__Combustiveis									
Colunas	Tipo	Tamanho	Precisão	Escala	Null	PK	FK	UK	Referência	Check
abastecimento_id	INT	4	10	0	Não	Sim	Sim	Não	abastecimentos	-o-
posto_id	INT	4	10	0	Não	Sim	Sim	Não	combustiveis	-o-
combustivel_id	INT	4	10	0	Não	Sim	Sim	Não	combustiveis	-o-
valor	MONEY	8	19	4	Não	Não	Não	Não	-o-	-o-
litros	DECIMAL	9	18	0	Não	Não	Não	Não	-o-	-o-

#### 4.6 Script de Banco de Dados

```

-----
create database uaifuel
go
use uaifuel
go

```

```
-- veiculos -----
create table veiculos
(
    placa          char(7)          not null,
    cor            char(20)         not null,
    status         int              not null,
    modelo         varchar(20)      not null,
    ano_fabricacao int              not null,
    ano_modelo     int              not null

    constraint pk_veiculo primary key (placa),
    constraint ak_veiculo unique(placa)
)
go

-- estados -----
create table estados
(
    uf          char(2) not null,
    nome        varchar(75) not null,

    constraint pk_estado primary key (uf),
    constraint ak_estado unique (uf)
)
go

-- cidades -----
create table cidades
(
    id          int not null identity,
    nome        varchar(120) not null,
    estado_uf   char(2) not null,

    constraint pk_cidade primary key(id),
    constraint fk_cidade__estado foreign key(estado_uf) references estados
)
go

-- pessoas -----
create table pessoas
(
    id          int not null identity,
    nome        varchar(100) not null,
    login       varchar(100) not null,
    senha       varchar(100) not null,
    status      int not null,

    constraint pk_pessoa primary key (id),
    constraint ak_pessoa_login unique(login)
)
go

-- postos -----
```

```

create table postos
(
    pessoa_id    int                not null,
    cnpj          varchar(20)        not null,
    telefone      varchar(30)        not null

    constraint pk_posto primary key(pessoa_id),
    constraint fk_posto__pessoa foreign key(pessoa_id) references pessoas,
    constraint ak_posto_cnpj unique(cnpj)
)
go

-- motoristas -----

create table motoristas
(
    pessoa_id    int                not null,
    cpf          varchar(15)        not null,
    credito      money              not null default 0,

    constraint pk_motorista primary key(pessoa_id),
    constraint fk_motorista__pessoa foreign key(pessoa_id) references pessoas,
    constraint ak_motorista_cpf unique(cpf)
)
go

-- administrador -----

create table administradores
(
    pessoa_id    int                not null,

    constraint pk_administrador primary key(pessoa_id),
    constraint fk_administrador__pessoa foreign key(pessoa_id) references pessoas
)
go

-- emails -----

create table emails
(
    pessoa_id    int                not null,
    endereco     varchar(50)        not null,

    constraint pk_email primary key (pessoa_id, endereco),
    constraint fk_email__pessoa foreign key (pessoa_id) references pessoas
)
go

-- enderecos -----

create table enderecos
(
    posto_id     int                not null,
    id           int                not null identity,
    cep          char(9)            null,
    localidade   varchar(200)       not null,
    cidade_id    int                not null

```



```

        constraint pk_endereco          primary key (posto_id, id),
        constraint fk_endereco__posto   foreign key (posto_id      ) references postos,
        constraint fk_endereco__cidade foreign key (cidade_id     ) references cidades
    )
go

-- combustiveis -----

create table combustiveis
(
    posto_id      int          not null,
    id            int          not null      identity,
    nome          varchar(50)   not null,
    status        int          not null      default 1,

    constraint pk_combustiveis          primary key (posto_id, id),
    constraint fk_combustiveis__posto   foreign key (posto_id      ) references
postos
)
go

-- abastecimentos -----

create table abastecimentos
(
    id            int          not null      identity,
    data          datetime     not null,
    cupom_fiscal  varchar(50)   null,
    valor_total   money        null         default 0,
    status        int          not null,
    veiculo_placa char(7)       not null,
    posto_id      int          not null,
    motorista_id  int          not null,

    constraint pk_abastecimento          primary key (id),
    constraint fk_abastecimento__veiculo foreign key (veiculo_placa)
        references veiculos,
    constraint fk_abastecimento__posto   foreign key (posto_id)
        references postos,
    constraint fk_abastecimento__motorista foreign key (motorista_id)
        references motoristas,
    constraint ak_abastecimento_cupom_fiscal unique (cupom_fiscal),
    constraint ck_abastecimento_valor_total check (valor_total >= 0),
    constraint ck_abastecimento_status     check (status in (1,2,3))
)
go

```

```
-- abastecimentos__combustiveis -----

create table abastecimentos__combustiveis
(
    abastecimento_id    int          not null,
    posto_id            int          not null,
    combustivel_id       int          not null,
    valor               money        not null,
    litros              decimal      not null

    constraint pk_abastecimento_combustivel primary key(abastecimento_id, posto_id,
combustivel_id),
    constraint fk_ac__abastecimento          foreign key(abastecimento_id)
        references abastecimentos,
    constraint fk_ac__posto                  foreign key(posto_id)
        references postos,
    constraint fk_ac__combustivel            foreign key(posto_id, combustivel_id)
        references combustiveis
)
go
```

#### 4.7 Inserção no banco de dados

```
-----

use uaifuel
go

-- estados -----

insert into estados (uf, nome)
values ('AC', 'Acre'),
       ('AL', 'Alagoas'),
       ('AM', 'Amazonas'),
       ('AP', 'Amapá'),
       ('BA', 'Bahia'),
       ('CE', 'Ceará'),
       ('DF', 'Distrito Federal'),
       ('ES', 'Espírito Santo'),
       ('GO', 'Goiás'),
       ('MA', 'Maranhão'),
       ('MG', 'Minas Gerais'),
       ('MS', 'Mato Grosso do Sul'),
       ('MT', 'Mato Grosso'),
       ('PA', 'Pará'),
       ('PB', 'Paraíba'),
       ('PE', 'Pernambuco'),
       ('PI', 'Piauí'),
       ('PR', 'Paraná'),
       ('RJ', 'Rio de Janeiro'),
       ('RN', 'Rio Grande do Norte'),
       ('RO', 'Rondônia'),
       ('RR', 'Roraima'),
       ('RS', 'Rio Grande do Sul'),
       ('SC', 'Santa Catarina'),
       ('SE', 'Sergipe'),
```

```

        ('SP', 'São Paulo'),
        ('TO', 'Tocantins');

go

-- cidades -----
insert into cidades(nome, estado_uf)
values ('São José do Rio Preto', 'SP'),
       ('São Paulo', 'SP'),
       ('Tanabi', 'SP'),
       ('Campinas', 'SP')

go

-- veiculo -----
insert into veiculos (placa, cor, status, modelo, ano_fabricacao, ano_modelo)
values ('DTU3934', 'CINZA COSMOS', 1, 'GOL', 2007, 2007),
       ('FUS1955', 'BRANCO', 1, 'FUSCA', 1955, 1955),
       ('FUS1974', 'AMARELO', 1, 'FUSCA', 1974, 1974),
       ('FIA1147', 'AZUL', 1, 'FIAT', 1981, 1982),
       ('CCC1111', 'VERDE', 1, 'SCANIA 4587 BITREM', 2018, 2018),
       ('CCC2222', 'BRANCO PÉROLA', 1, 'VOLVO 7854', 2020, 2021),
       ('CCC3333', 'AZUL ESCURO', 2, 'VOLVO 7854', 2020, 2021),
       ('BRA4C44', 'ROXO', 1, 'IVECO 458 GRANELEIRO', 2021, 2021)

go

-- pessoas -----
insert into pessoas (nome, login, senha, status)
values ('AUTOPOSTO PINATTO', 'pinatto', '123', 1),
       ('ELIAS MANTOVANI', 'elias', '123', 1),
       ('POSTO CINQUENTÃO', 'posto50', '123', 1),
       ('PEDRO COIMBRA', 'pedro', '123', 1),
       ('ESTEVAN NAVARRO LAGO', 'estevan', '123', 1),
       ('HARRY HPOTTER', 'potter', '123', 2),
       ('POSTO 474', 'posto474', '123', 1),
       ('POSTO VILLA SETTE', 'sette', '123', 2),
       ('Administrador do Sistema', 'admin', 'admin', 1)

go

-- postos -----
insert into postos (pessoa_id, cnpj, telefone)
values (1, '79.263.278/0001-01', '(17) 981492059'),
       (3, '42.931.155/0001-91', '(17) 981492059'),
       (7, '19.499.378/0001-00', '(17) 810362496'),
       (8, '92.773.681/0001-49', '(17) 925478123')

go

-- motoristas -----
insert into motoristas (pessoa_id, cpf, credito)
values (2, '924.813.930-23', 1200),
       (4, '471.874.450-40', 5000),
       (5, '027.910.030-29', 0.50),
       (6, '374.675.570-00', -2800.30)

go

-- emails -----
insert into emails values(1, 'pinattoposto@gmail.com')

```

```

insert into emails values(1, 'autopostopinatto@hotmail.com')
insert into emails values(2, 'eliasmantovan@fatec.sp.gov.br')
insert into emails values(3, 'cinquentao@yahoo.com.br')
insert into emails values(4, 'pedro@fatec.sp.gov.br')
insert into emails values(4, 'coimbra@gmail.com')
insert into emails values(4, 'sposito@bol.com.br')
insert into emails values(6, 'harry@fantasy.com.br')
insert into emails values(8, 'villasette@hotmail.com')
go

-- enderecos -----

insert into enderecos (posto_id, cep, localidade, cidade_id)
      values (1, '15055-200', 'Paulo Vidalli, 285', 5264),
             (3, '15050-453', 'XV de Novembro, 1515', 5270),
             (7, '15054-030', 'Rodovia Euclides da Cunha, km 477', 5302),
             (8, '15046-778', 'Santos Dumont, 3281', 4814)
go

-- combustiveis -----

insert into combustiveis (posto_id, nome, status)
      values (1, 'ÁLCOOL', 1),
             (1, 'GASOLINA', 1),
             (1, 'GASOLINA ADITIVADA', 2),
             (1, 'DIESEL S10', 1),
             (1, 'DIESEL S500', 1),
             (3, 'ÁLCOOL', 1),
             (3, 'GASOLINA', 1),
             (3, 'GASOLINA ADITIVADA', 2),
             (3, 'DIESEL S10', 1),
             (3, 'DIESEL S500', 1),
             (3, 'ARLA', 1),
             (3, 'ADITIVO', 1),
             (3, 'QUEROSENE', 2),
             (7, 'ÁLCOOL', 1),
             (7, 'GASOLINA', 1),
             (7, 'GASOLINA ADITIVADA', 1),
             (7, 'DIESEL S10', 1),
             (7, 'DIESEL S500', 1),
             (8, 'ÁLCOOL', 1),
             (8, 'GASOLINA', 1),
             (8, 'GASOLINA ADITIVADA', 1),
             (8, 'DIESEL S10', 1),
             (8, 'DIESEL S500', 1)
go

-- abastecimentos -----

insert into abastecimentos (data, cupom_fiscal, valor_total, status, veiculo_placa,
                           motorista_id, posto_id )
      values (getdate(), '26597419017464', 0.0, 2, 'DTU3934', 5, 3),
             (getdate()-1, '03647950546329', 0.0, 1, 'FIA1147', 2, 1 ),
             (getdate()-2, '97501348564102', 0.0, 2, 'CCC1111', 6, 7 ),
             (getdate()-3, '69871230459670', 0.0, 1, 'BRA4C44', 4, 8 )
Go

```

```
-- N--      =====
-- abastecimentos__combustiveis -----

insert into abastecimentos__combustiveis (abastecimento_id, posto_id, combustivel_id,
                                          valor, litros)
      values (1, 3, 6, 6.20, 52),
             (1, 3, 8, 3.86, 100),
             (1, 3, 12, 2.55, 20)
go

insert into abastecimentos__combustiveis (abastecimento_id, posto_id, combustivel_id,
                                          valor, litros)
      values (2, 1, 1, 6.80, 30  ),
             (2, 1, 3, 5.50, 15.84),
             (2, 1, 5, 30.00, 1  )
go

insert into abastecimentos__combustiveis (abastecimento_id, posto_id, combustivel_id,
                                          valor, litros)
      values (3, 7, 17, 4.10, 750),
             (3, 7, 16, 2.55, 250)
go

insert into abastecimentos__combustiveis (abastecimento_id, posto_id, combustivel_id,
                                          valor, litros)
      values (4, 8, 19, 4.10, 500),
             (4, 8, 21, 2.55, 100)
go

-- UPDATE abastecimentos -----

update abastecimentos
set valor_total =
(
    select sum(valor * litros)
    from abastecimentos__combustiveis
    where abastecimento_id = 1
)
where id = 1
go

update abastecimentos
set valor_total =
(
    select sum(valor * litros)
    from abastecimentos__combustiveis
    where abastecimento_id = 2
)
where id = 2
go

update abastecimentos
set valor_total =
(
    select sum(valor * litros)
    from abastecimentos__combustiveis
    where abastecimento_id = 3
)
where id = 3
go
```

```
update abastecimentos
set valor_total =
(
    select sum(valor * litros)
    from abastecimentos__combustiveis
    where abastecimento_id = 4
)
where id = 4
go
```

## 4.8 Views

1. Visualização da tabela motoristas.

```
create view v_motoristas
as
select p.id as MotoristaId, p.nome as MotoristaNome
from pessoas as p inner join motoristas as m
on p.id = m.pessoa_id
go
```

```
-- Teste de execução -
Select * from v_motoristas
```

2. Visualização da tabela postos.

```
create view v_postos
as
select p.id as PostoId, p.nome as PostoNome
from pessoas as p
inner join postos as pt
on p.id = pt.pessoa_id
go
```

```
-- Teste de execução -
Select * from v_postos
```

3. Visualização da tabela abastecimentos.

```
create view v_abastecimentos
as
select a.id as Id, a.data as Data, a.cupom_fiscal as CupomFiscal, a.valor_total as
ValorTotal, a.status as Status, a.veiculo_placa as VeiculoPlaca, po.PostoId,
po.PostoNome, m.MotoristaId, m.MotoristaNome
from abastecimentos a
inner join v_postos po on a.posto_id = po.PostoId
inner join v_motoristas m on a.motorista_id = m.MotoristaId
inner join veiculos v on a.veiculo_placa = v.placa
go
```

```
-- Teste de execução -
Select * from v_abastecimentos
```

4. Visualização da tabela de administradores

```
create or alter view v_administrador
as
select
a.pessoa_id AdministradorId,
p.nome Nome
from
administradores a,
pessoas p
where
p.id = a.pessoa_id
```

```
-- Teste de execução -
Select * from v_abastecimentos
```

## 5 Visualização da tabela de estados

```
create or alter view v_estado
as
select
    uf sigla,
nome
from
    estados
```

```
-- Teste de execução -
Select * from v_estado
```

## 6 Visualização da tabela de cidades

```
create or alter view v_cidade
as
select
    e.nome Estado,
c.nome
from
    cidades c,
estados e
where
    e.uf = c.estado_uf
```

```
-- Teste de execução -
Select * from v_cidade
```

## 7 Visualização da tabela de combustíveis

```
create or alter view v_combustivel as
select
c.id,
p.PostoNome Posto,
c.nome Combustivel
from
v_postos p,
combustiveis c
where
p.PostoId=c.posto_id
```

```
-- Teste de execução -
Select * from v_combustivel
```



## 7.1 Procedures

1. Procedure para pesquisa de abastecimentos para o relatório.

```
CREATE OR ALTER PROCEDURE procAbastecimento
@v_acao int = null,
@v_data datetime = null,
@v_cupomFiscal varchar(50) = null,
@v_valorTotal money = null,
@v_status int = null,
@v_veiculoPlaca varchar(7) = null,
@v_motoristaId int = null,
@v_postoId int = null,

@v_combustivelId int = null,
@v_valor money = null,
@v_litros decimal = null,
@v_abastecimentoId int = null
AS
BEGIN
    IF @v_acao = 1
        BEGIN
            insert into abastecimentos
            (data, cupom_fiscal, valor_total, status, veiculo_placa,
            motorista_id, posto_id)
            values
            (CONVERT(DATETIME, @v_data), @v_cupomFiscal, @v_valorTotal,
            @v_status, @v_veiculoPlaca, @v_motoristaId, @v_postoId);

            SET @v_abastecimentoId = (select @@IDENTITY);

            insert into abastecimentos__combustiveis
            (abastecimento_id, posto_id, combustivel_id, valor, litros)
            values
            (@v_abastecimentoId, @v_postoId, @v_combustivelId, @v_valor,
            @v_litros);

            SELECT @v_abastecimentoId AS AbastecimentoId;

        END;

    IF @v_acao = 2
        BEGIN
            insert into abastecimentos__combustiveis
            (abastecimento_id, posto_id, combustivel_id, valor, litros)
            values
            (@v_abastecimentoId, @v_postoId, @v_combustivelId, @v_valor,
            @v_litros);

        END;

    IF @v_acao = 3
        BEGIN
            delete from abastecimentos__combustiveis
            where abastecimento_id = @v_abastecimentoId
            and posto_id = @v_postoId;

            delete from abastecimentos where id = @v_abastecimentoId

        END;

END
GO
```

-- Teste de execução --

```
----- INSERT -----
exec procAbastecimento    @v_acao = 1,
                           @v_data = '20200618',
                           @v_cupomFiscal = '32323232',
                           @v_valorTotal = 340,
                           @v_status = 2,
                           @v_veiculoPlaca = 'FUS1974',
                           @v_motoristaId = 2,
                           @v_postoId = 7,
                           @v_combustivelId = 15,
                           @v_valor = 5.2,
                           @v_litros = 40
```

```
----- UPDATE -----
exec procAbastecimento    @v_acao = 2,
                           @v_abastecimentoId = 5,
                           @v_postoId = 7,
                           @v_combustivelId = 17,
                           @v_valor = 5.47,
                           @v_litros = 30
```

```
----- DELETE -----
exec procAbastecimento    @v_acao = 3,
                           @v_abastecimentoId = 5,
                           @v_postoId = 7
```

2. Procedure para inserção, atualização e exclusão na tabela veículos.

CREATE OR ALTER PROCEDURE procVeiculo

```
@v_acao int = null,
@placa varchar(7) = null,
@cor varchar(20) = null,
@status int = null,
@modelo varchar(20) = null,
@ano_fabricacao int = null,
@ano_modelo int = null
```

AS  
BEGIN

```
-----INSERT-----
    IF @v_acao = 1
        BEGIN
            insert into veiculos
            (placa, cor, status, modelo, ano_fabricacao, ano_modelo)
        values
            (@placa, @cor, @status, @modelo, @ano_fabricacao, @ano_modelo);

        END;

-----UPDATE-----
    IF @v_acao = 2
        BEGIN
            update veiculos
            set
```

```

cor = @cor,
status = @status,
modelo = @modelo,
ano_fabricacao = @ano_fabricacao,
ano_modelo = @ano_modelo

where

placa = @placa

END;

-----DELETE-----
IF @v_acao = 3
BEGIN
delete from veiculos where placa = @placa
END;

END

-----INSERT-----
exec procVeiculo
    @v_acao = 1,
    @placa = 'EFP9696',
    @cor = 'Titanium',
    @status = 1,
    @modelo = 'Twingu',
    @ano_fabricacao = 1998,
    @ano_modelo = 1990

-----UPDATE-----
exec procVeiculo
    @v_acao = 2,
    @placa = 'EFP9696',
    @cor = 'Vinho',
    @status = 1,
    @modelo = 'Twingu',
    @ano_fabricacao = 1998,
    @ano_modelo = 1990

-----DELTE-----
exec procVeiculo
    @v_acao = 3,
    @placa = 'EFP9696'

```

3. Procedure para inserção, atualização e exclusão na tabela combustiveis.

```
CREATE OR ALTER PROCEDURE procCombustivel
```

```

@v_acao int = null,
@combustivelId int = null,
@postoId int = null,
@nome varchar(50) = null,
@status int = null

```

```

AS
BEGIN

```

```

-----INSERT-----
IF @v_acao = 1

```

```

BEGIN

    insert into combustiveis
    (posto_id, nome, status)
values
    (@postoId, @nome, @status);

END;

-----UPDATE-----
IF @v_acao = 2
BEGIN
    update combustiveis
    set
        nome = @nome,
        status = @status

    where
        id = @combustivelId

    END;

-----DELETE-----
IF @v_acao = 3
BEGIN
    delete from combustiveis where id = @combustivelId

    END;

END

-----INSERT-----
exec procCombustivel
    @v_acao = 1,
    @postoId = 7,
    @nome = 'GAZOLINA ADITIVADA',
    @status = 1

select * from combustiveis

-----UPDATE-----
exec procCombustivel
    @v_acao = 2,
    @combustivelId = 24,
    @postoId = 7,
    @nome = 'GASOLINA ADITIVADA',
    @status = 1

-----DELTE-----
exec procCombustivel
    @v_acao = 3,
    @combustivelId = 24

```

4. Procedure para inserção, atualização e exclusão na tabela motoristas.

CREATE OR ALTER PROCEDURE procMotorista

```

@v_acao int = null,
@motoristaId int = null,
@login varchar(100) = null,
@nome varchar(100) = null,
@senha varchar(100) = null,

```

```

@cpf varchar(15) = null,
@credito money = null,
@status int = null,
@email varchar(50) = null

AS
BEGIN

-----INSERT-----
    IF @v_acao = 1
        BEGIN

            insert into pessoas
                (nome, login, senha, status)
            values
                (@nome, @login, @senha, @status);

            SET @motoristaId = (SELECT @@IDENTITY);

            insert into motoristas
                (pessoa_id, cpf, credito)
            values
                (@motoristaId, @cpf, @credito);

            insert into emails
                (pessoa_id, endereco)
            values
                (@motoristaId, @email);

            END;

-----UPDATE-----
    IF @v_acao = 2
        BEGIN

            update pessoas
            set
                nome = @nome,
                login = @login,
                senha = @senha,
                status = @status
            where
                id = @motoristaId;

            update motoristas
            set
                cpf = @cpf,
                credito = @credito
            where
                pessoa_id = @motoristaId;

            update emails
            set
                endereco = @email
            where
                pessoa_id = @motoristaId;

            END;

-----DELETE-----
    IF @v_acao = 3
        BEGIN
            delete from emails where pessoa_id = @motoristaId;
        
```

```

        delete from motoristas where pessoa_id = @motoristaId;

        delete from pessoas where id = @motoristaId;

    END;

END

-----INSERT-----
exec procMotorista
    @v_acao = 1,
    @login = 'ashuashua',
    @senha = '123456',
    @nome = 'REINALDO JANEQUINE',
    @cpf = '41638200866',
    @credito = 15000,
    @status = 1,
    @email = 'contato_reinaldo@gmail.com'

-----UPDATE-----
exec procMotorista
    @v_acao = 2,
    @motoristaId = 10,
    @login = 'rei_naldo',
    @senha = '123456',
    @nome = 'REINALDO JANEQUINE',
    @cpf = '416.382.008-66',
    @credito = 15000,
    @status = 1,
    @email = 'reinaldo@gmail.com'

-----DELETE-----
exec procMotorista
    @v_acao = 3,
    @motoristaId = 10

```

5. Procedure para inserção, atualização e exclusão nas tabelas pessoa, postos, email e endereço.

```

CREATE OR ALTER PROCEDURE procPosto

    @v_acao int = null,
    @postoId int = null,
    @login varchar(100) = null,
    @nome varchar(100) = null,
    @senha varchar(100) = null,
    @cnpj varchar(20) = null,
    @telefone varchar(30) = null,
    @status int = null,
    @cep varchar(9) = null,
    @localidade varchar(20) = null,
    @cidadeId int = null,
    @idEndereco int = null,
    @email varchar(50) = null

AS
BEGIN

```

```

-----INSERT-----
IF @v_acao = 1
    BEGIN

        insert into pessoas
            (nome, login, senha, status)
        values
            (@nome, @login, @senha, @status);

        SET @postoId = (SELECT @@IDENTITY);

        insert into postos
            (pessoa_id, cnpj, telefone)
        values
            (@postoId, @cnpj, @telefone);

        insert into enderecos
            (posto_id, cep, localidade, cidade_id)
        values
            (@postoId, @cep, @localidade, @cidadeId);

        insert into emails
            (pessoa_id, endereco)
        values
            (@postoId, @email);

    END;

-----UPDATE-----
IF @v_acao = 2
    BEGIN

        update pessoas
        set
            nome = @nome,
            login = @login,
            senha = @senha,
            status = @status
        where
            id = @postoId;

        update postos
        set
            cnpj = @cnpj,
            telefone = @telefone
        where
            pessoa_id = @postoId;

        update enderecos
        set
            cep = @cep,
            localidade = @localidade,
            cidade_id = @cidadeId
        where
            posto_id = @postoId
            and id = @idEndereco;

        update emails
        set
            endereco = @email
        where
            pessoa_id = @postoId;
    
```

```

END;

-----DELETE-----
IF @v_acao = 3
BEGIN
    delete from emails where pessoa_id = @postoId;

    delete from enderecos where @postoId = @postoId;

    delete from postos where pessoa_id = @postoId;

    delete from pessoas where id = @postoId;
END;

END

-----INSERT-----
exec procPosto
    @v_acao = 1,
    @login = 'ashuashua',
    @senha = '123456',
    @nome = 'POSTO JANEQUINE',
    @cnpj = '41638200866',
    @telefone = '70707070',
    @status = 1,
    @cep = '15081470',
    @localidade = 'RODOVIA BR-153',
    @cidadeId = 1553,
    @email = 'contato_reinaldo@gmail.com'

SELECT * FROM enderecos

-----UPDATE-----
exec procPosto
    @v_acao = 2,
    @login = 'ashuashua',
    @senha = '123456',
    @nome = 'POSTO JANEQUINE',
    @cnpj = '41638200866',
    @telefone = '70707070',
    @status = 1,
    @cep = '15081470',
    @localidade = 'RODOVIA BR-153',
    @cidadeId = 1553,
    @email = 'contato_reinaldo@gmail.com',
    @postoId = 11,
    @idEndereco = 5

-----DELETE-----
exec procPosto
    @v_acao = 3,
    @postoId = 11

```



## 8 Considerações Finais

### 8.1 Dificuldades Encontradas

A maior dificuldade que encontramos foi com relação à conciliação de todos os trabalhos em todas as matérias, devido ao curto período disponível para o desenvolvimento do projeto. Dificuldades com codificação ou algum outro recurso necessário no desenvolvimento são aspectos solucionáveis, pois dependem apenas da pesquisa e orientação dos professores para serem superados. Houve também dificuldades devido a troca de professores durante o semestre, assim como a pandemia e o Covid-19, que foram empecilhos durante nosso desenvolvimento.

### 8.2 Sugestões de Melhoria

Deixar o aluno inteirado de todas as atividades que serão requisitadas no projeto interdisciplinar logo no início do semestre, para que não fiquemos confusos com relação ao que devemos fazer e quando, assim como uma clara apresentação do que deve ser entregue ao final do período. Também seria ideal todos os professores trabalharem tendo em mente que o interdisciplinar está ocorrendo (professores que participam e que não participam do interdisciplinar), possibilitando uma melhor organização na entrega de atividades. Desta forma, o tempo seria mais bem gerido e todas as tarefas poderiam ser feitas da melhor forma possível.

## 9 Referências

- SOMMERVILLE, I. Engenharia de software. 8.ed. São Paulo: Addison Wesley, 2007.
- LARMAN, C. Utilizando UML e padrões: uma introdução à análise e projeto orientados a objetos e ao desenvolvimento iterativo. Bookman, 2007.
- BOOCH, G.; RUMBAUGH, J; JACOBSON, I. UML: Guia do usuário. Elsevier, 2006.
- ARAÚJO, E. C. ASP.NET Core MVC: Aplicações modernas em conjunto com o Entity Framework. Editora Casa do Código, 2018.
- SHARP, J. Microsoft Visual C# 2013 – Passo a Passo, Editora Bookman, 2014.
- SANDERS, W.; MUANG, L.; PENBERTHY, W.; Developing ASP.Net MVC 4 Web Applications (MCSD Exame Ref. 70-486), 1.ed. Microsoft Press, 2013.
- ASP.NET Core. Disponível em: <https://docs.microsoft.com/pt-br/aspnet/core>. Acesso em: 15/05/2021
- STELLMAN, A.; GREENE, J. Use a Cabeça: C#, 2.ed. Editora Alta Books.
- ELMASRI, R.; NAVATHE, S. B. Sistemas de Banco de Dados: Fundamentos e Aplicações. Pearson, 2005.
- HARRINGTON, J. L. Projeto de bancos de dados relacionais: teoria e prática. Campus, 2002.
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de Banco de Dados. Campus, 2006.
- BORGES, S. Modelagem de Dados: 10 exemplos práticos. Pará de Minas: VirtualBooks, 2014.
- DATE, C. J. Introdução a sistemas de banco de dados. 8 ed. Rio de Janeiro: Elsevier, 2003.