

**CENTRO PAULA SOUZA  
FACULDADE DE TECNOLOGIA DE FRANCA  
“Dr. THOMAZ NOVELINO”**

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**LUIS FERNANDO SANTOS BEZERRA  
ANDRÉ PEDRO GOMES**

**SISTEMA WEB PARA CONTROLE DE BOMBA DE COMBUSTÍVEL**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Carlos Alberto Lucas

**FRANCA/SP**

**2023**

# SISTEMA WEB PARA CONTROLE DE BOMBA DE COMBUSTÍVEL

LUIS FERNANDO SANTOS BEZERRA<sup>1</sup>  
ANDRÉ PEDRO GOMES<sup>2</sup>

## Resumo

Este projeto tem como objetivo desenvolver um sistema que vise automatizar o fluxo e controle da bomba de combustível de diesel de uma empresa "X" na cidade de Brodowski, gerenciando a entrada, o estoque mínimo e o manuseio da bomba pelo frentista, além de proporcionar a geração de relatórios para análise. Para atingir este objetivo, desenvolvemos um projeto base, junto com a documentação que explica como funciona o software que gerencia este abastecimento da frota, o controle de estoque e o monitoramento do consumo de combustível. O propósito do projeto foi conseguir modernizar o setor responsável pelo abastecimento, tornando o fluxo operacional mais eficiente, além de reduzir as despesas.

**Palavras-chave:** Abastecimento; combustível; despesas; eficiência; monitoramento; software.

## Abstract

This project aims to develop a system that aims to automate the flow and control of the diesel fuel pump of a company "X" in the city of Brodowski, managing the entrance, the minimum stock and the handling of the pump by the attendant, in addition to providing generating reports for analysis. To achieve this goal, we developed a base project, along with documentation that explains how the software that manages this fleet supply, stock control and fuel consumption monitoring works. The purpose of the project was to modernize the sector responsible for supply, making the operational flow more efficient, in addition to reducing expenses.

**Keywords:** *Refueling, fuel, expenses, efficiency, monitoring, software.*

---

<sup>1</sup> Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico:

<sup>2</sup> Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico:

## 1 - Introdução

A tecnologia está em constante evolução para nos beneficiar em vários aspectos, mas em alguns momentos ainda ficamos presos no passado, principalmente nas indústrias onde faltam treinamentos para funcionários, fazendo com que regredem para uma forma de trabalho arcaica.

A empresa que escolhemos embora seja muito rígida nos seus processos, ainda utiliza papel e caneta para coletar informações, além de outras formas nada funcionais para um dia a dia de trabalho.

Buscamos entender todos os processos através de uma entrevista com o (Coordenador do setor de transportes), levantamos os requisitos necessários para o mapeamento do projeto buscando a melhor solução e divisão de tarefas entre a equipe para conseguirmos atender os prazos de entregas impostos. Assim focamos em entregar a documentação e prototipação necessária para desenvolvimento de um software que controle os abastecimentos, automatização, gerenciamento de motoristas e geração de relatórios para análise.

Por ser uma empresa que mantemos contato direto através de um dos integrantes da dupla, optamos por escolher esse tema devido ao reconhecimento das fragilidades do sistema interno de gerenciamento de combustível. Por ter conhecimento dos fluxos internos, o integrante do grupo que também é colaborador da empresa verificou-se a grande necessidade de uma melhoria nos processos internos nos impulsionando a desenvolver um software para gestão.

Seguindo o modelo de negócio da empresa 'X' Distribuidora, a missão é distribuir produtos de qualidade e atender o maior número de regiões possíveis, de modo a melhorar a qualidade de gerenciamento da empresa, onde impacta diretamente na área de distribuição de produtos.

Desde modo, optamos por desenvolver um sistema que vise automatizar o controle da bomba de combustível de diesel, gerenciando entrada, estoque mínimo e manuseio da bomba pelo frentista.

O objetivo do projeto é desenvolver um software que vise sanar a necessidade de gestão da bomba de combustível da empresa, para isso, optamos por apresentar nossas soluções por meio das seguintes funcionalidades:

Seguindo a iniciativa do projeto, realizamos a divisão em cinco etapas: definição do escopo do projeto, gestão participativa, elaboração da documentação, desenvolvimento e finalização.

Em seu escopo inicial, onde definimos sua primeira etapa sendo iniciada com a realização do walkthrough, a estipulação das atividades da dupla, a entrevista com o cliente e com isso a análise do resultado com base nas respostas do entrevistado.

A segunda etapa, onde foi feita a gestão participativa, foram realizados a matriz SWOT, a definição do SW2H, a criação do modelo EAP, finalizando a etapa com a criação do modelo TAP.

A terceira etapa, onde foi realizada a elaboração da documentação, foi composta pelas criações do BPMN, do diagrama de caso de uso, da documentação de requisitos e documentação do caso de uso, e por fim com a criação dos diagramas de atividade, máquina de estado e sequência.

Na quarta etapa foi realizado a parte de desenvolvimento onde nela foi feita a prototipação e a matriz de rastreabilidade do projeto.

Para finalizar, sendo está a quinta e última etapa do projeto, foi desenvolvido um CRUD do projeto para apresentação.

## **2 - Viabilidade do projeto**

### **2.1 - Matriz SWOT**

Quando vamos tomar uma decisão mais difícil, entramos em uma balança mental e pesamos os prós e contras, para chegar a uma decisão. E a matriz SWOT, é basicamente essa balança em formato de matriz, e ela vai servir de ponto de partida para a empresa enxergar em um contexto geral, que vai ser o ponto de partida para elaborar a melhor estratégia de negócios para a empresa.

Foi identificado que a empresa tem como fraquezas:

- Falta de controle do estoque de combustível;
- Retrabalho passando do papel para uma planilha os dados dos abastecimentos do mês, gerando custo com horas extras;
- Falta de identificação de quem realmente está abastecendo;
- Não saber qual a quantidade mínima de combustível a se ter na bomba do posto.

## 2.2 - 5W2H

O 5W2H vai basicamente pegar toda a análise feita em um planejamento, por exemplo a Matriz SWOT e vai transformar em ações práticas, a onde vai indicar as atividades, os prazos e as responsabilidades de todos os envolvidos em um projeto. Ele pode ser aplicado em empresas ou na vida pessoal, pois é possível organizar tarefas, orçamento mensal de uma casa ou o orçamento para uma viagem. A estrutura do 5W2H, consiste em responder as seguintes perguntas:

**What:** Ação que deve ser executada, ou problema a ser solucionado;

**Why:** Justificativa dos motivos e objetivos daquilo estar sendo executado ou solucionado;

**Who:** Definição de quem será (serão) o(s) responsável(eis) pela execução do que foi planejado;

**Where:** Informação sobre onde cada um dos procedimentos será executado;

**When:** Cronograma sobre quando ocorrerão os procedimentos;

**How:** Explicação sobre como serão executados os procedimentos para atingir os objetivos pré-estabelecidos;

**How Much:** Limitação de quanto custará cada procedimento e o custo total do que será feito?

**Quadro 01: Matriz SWOT**

WHAT	WHY	WHERE	WHEN	WHO	HOW	HOW MUCH
Controlar o estoque mínimo e entrada de combustível	Evitar o risco de não haver combustível na bomba ou a necessidade realizar um segundo pedido	Soft	Junho	Gerente ou responsável	Disparando um alerta ou e-mail para o gerente ou responsável	Sem custo
Evitar a perda de dados	Para reduzir os riscos e esforços na importação dos dados para uma planilha excel	Soft	Junho	Frentista	Cadastrando os motoristas e gerenciando o valor que é abastecido	Sem custo
Confirmar quem realmente abasteceu	Manter um registro dos dados relacionados ao abastecimento	Soft	Junho	Frentista	Utilizando a placa do veículo para verificar a autenticidade do motorista e realizar o controle do abastecimento	Sem custo

Visualizar o panorama geral do abastecimento	Visualização das operações no setor e facilitar a tomada de decisões	Soft	Junho	Gerente ou responsável	Possibilitando a geração de relatórios mensais e gráficos para análise dos dados	Sem custo
--	--	------	-------	------------------------	--	-----------

Fonte: os autores

## 2.3 - Viabilidade

Com o projeto será possível controlar todos os processos que envolvem o abastecimento, assim evitando a perda de dados, além de do controle de usuários, podendo barrar ou não que pode realizar o abastecimento e utilizar o serviço.

Além de poder obter métricas de quanto combustível está sendo consumido diariamente, semanalmente e mensalmente, o que ajuda a analisar se está pensando ou não ter a bomba na empresa. Além de evitar que pessoas abasteçam e acidentalmente “esqueçam” de registrar a saída de combustível.

Também vai ajudar a minimizar gastos com funcionários fazendo horas extras, para transcrever dados, para o Excel, assim é possível obter, agilidade, segurança e economia com esse sistema.

## 3 - Levantamento de Requisitos

### 3.1 Elicitação e especificação dos Requisitos

Quando vamos realizar o processo de elicitação de requisitos, temos 4 modelos bastante usado.

1° **Entrevista:** Conhecida por ser a mais comum das técnicas de elicitação de requisitos. As entrevistas geralmente envolvem discussões individuais de um representante da equipe com o stakeholder do novo sistema, formais ou informais. A onde são colocadas questões formuladas pela equipe de engenharia de requisitos, sobre o processo de trabalho atual do cliente e sobre o sistema que será desenvolvido. E deve ser preparada uma pauta a ser seguida durante a entrevista, para não abandonar nenhuma informação. (RETRAINING, 2023).

2° **Prototipação:** Pode ser considerada como a confecção de uma versão inicial do produto, preparada para coletar o feedback das partes interessadas e levantar

alterações que devem ser incorporadas nas próximas edições. Poder ser feito um protótipo em papel, a onde se tem representações de interfaces gráficas com diferentes níveis de fidelidade, desde um wireframe desenhado à mão em pequenos pedaços de papel até uma embalagem de sabonete ou protótipo de experiência: Que é a simulação da experiência do serviço que prevê alguns de seus desempenhos através do uso de interações específicas envolvidas. (RETRAINING, 2023).

**3° Brainstorming:** É uma discussão informal onde cada participante pode expressar suas ideias livremente para que um novo sistema seja desenvolvido. Basicamente é uma reunião em que conjunto de ideias de vários profissionais dele ou vários setores, são discutidas, sem grandes críticas. Assim se dividindo em duas fases a de coleta de ideias e fase de refinamento, a onde as melhores ideias para o problema em questão, vão sendo refinadas, até chegar no objetivo a ser cumprido. (RETRAINING, 2023).

**4° Card sorting:** Antigamente algumas empresas não se preocupavam em oferecer uma boa experiência de usuário, a seus clientes, o que levou muitas a ruína, mas com o passar dos anos essa visão foi mudando, e entre as ferramentas de elicitación de requisitos surgiu o Card sorting uma ferramenta de design, que consiste em estruturar a arquitetura de informação, por meio de cartões ou post-its com conteúdo, os usuários fazem agrupamentos, organizando os itens de acordo com seu mapa mental. O usuário organiza os temas em categorias que façam sentido para ele, podendo também nomear cada grupo. (RETRAINING, 2023).

O que vai ajudar os desenvolvedores de projetos, essa técnica facilita a criação de sistemas e plataformas, tornando-as mais intuitivas e focadas no usuário, reduzindo tempo e custos para empresas.

Somente a partir da elicitación de requisitos, é possível entender as expectativas e necessidades do usuário final em relação ao projeto, além de facilitar as definições do projeto entre diversas partes interessadas, como gerentes, clientes, analistas de negócios, desenvolvedores, entre outros. Também ajuda a evitar problemas surpresas durante o desenvolvimento do projeto, sendo ele causador de atraso na en-

trega ou financeiro, além de estabelecer uma boa comunicação entre as partes envolvidas, facilitando o desenvolvimento e adaptações do projeto, quando necessário.

Para desenvolvimento desse projeto, escolhemos o método de entrevista para realizar a elicitación de requisitos com nosso cliente (gerente de produção da empresa), pois nesse método conseguimos abordar questões já elaboradas, como uma visão pré-formulada do projeto, a entrevista foi realizada via Whatsapp, pois a empresa se encontra em outra cidade, e devido à alta demanda de tarefas do cliente, seria o meio de comunicação mais viável, Enviamos todas as perguntas e cliente foi respondendo conforme conseguia, nas questões enviadas, tivemos como objetivo extrair as maiores dores e objetivos que a empresa tinha com o posto de gasolina interno. E a partir das informações obtidas, realizamos a adaptação em alguns pontos que já havíamos idealizados para o projeto, e visualizamos novos pontos a serem abordados, durante o desenvolvimento do projeto.

### **3.2 BPMN (Business Process Model and Notation)**

O BPMN é uma metodologia que ajuda a organizar os processos de negócios, de modo que nada se perca ao levantar os processos, principalmente quando se esbarra em vários setores ou em diferentes níveis hierárquicos.

Essa metodologia de gerenciamento de processos é representada através de um fluxograma, que modela as etapas de ponta a ponta, de um processo de negócios planejado, para fins de padronização, treinamento de funcionários, gestão da qualidade e conformidade de auditoria.

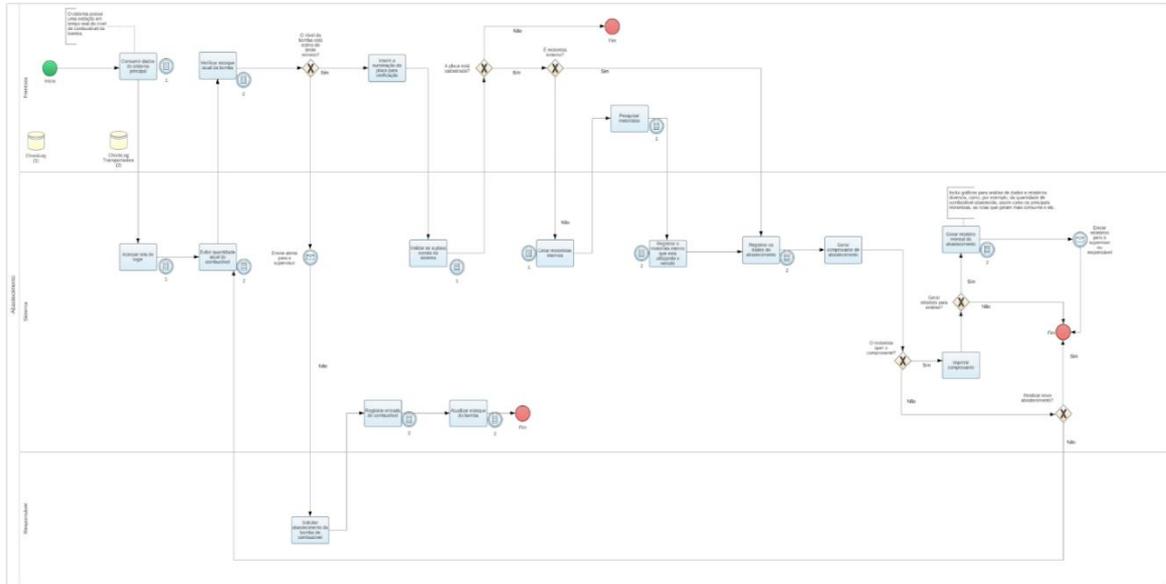
De acordo com o glossário da Gartner, que se refere ao tema por *Business Process Modeling*, a definição é:

*“A modelagem de processos de negócios vincula a estratégia de negócios ao desenvolvimento de sistemas de TI para garantir o valor do negócio.*

*Ele combina visualizações de processos/fluxo de trabalho, funcionais, organizacionais e de dados/recursos com métricas subjacentes, como custos, tempos de ciclo e responsabilidades, para fornecer uma base para analisar cadeias de valor, custos baseados em atividades, gargalos, caminhos críticos e ineficiências.* (LUID-CHART, 2023)

Ou seja, é uma forma de ajudar no mapeamento de processos, utilizando símbolos padronizados em uma representação gráfica.

**Figura 01: BPMN**



**Fonte: os autores**

### 3.3 Requisitos Funcionais

Os requisitos funcionais são a declaração de como o sistema deve se portar e podem ser divididos em duas partes: função e comportamento. A função é o que o sistema faz (por exemplo “registrar as saídas de combustível”). O comportamento é como o sistema vai fazer isso (por exemplo” registrar quantos litros estão saindo em uma operação de abastecimento, e a partir desses dados, subtrair da quantidade em litros do estoque atual, e registrar essa quantidade no histórico da operação”). Quando for criar requisitos deve-se ser específico sobre o que o sistema deve fazer, alcançável, para atender o prazo de terminado e relevante para o objetivo do negócio.

**Quadro 2 – Requisitos Funcionais do sistema**

<p><b>RF001-</b> Consumir dados do sistema principal</p>	<p>Categoria:  <input checked="" type="checkbox"/> Oculto  <input type="checkbox"/> Evidente</p>	<p>Prioridade:  <input checked="" type="checkbox"/> Altíssima  <input type="checkbox"/> Alta  <input type="checkbox"/> Média  <input type="checkbox"/> Baixa</p>
--	--	--

<b>Descrição:</b> Através da API fornecida pelo sistema principal de cadastro e gerenciamento da empresa, os dados de cadastro dos frentistas e motoristas serão consumidos a fim de gerar as credenciais de acesso ao sistema e listagem de veículos.		
<b>RF002</b> -Exibir quantidade atual de combustível	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> Na dashboard do sistema, o usuário terá acesso à informação em tempo real da quantidade de combustível presente na bomba.		
<b>RF003</b> -Verificar estoque	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> Antes de cada abastecimento, o sistema realizará uma verificação para validar a quantidade atual de combustível disponível na bomba, essa ação será necessária para servir de gatilho para o disparo de e-mail ao responsável..		
<b>RF004</b> - Enviar alerta para usuário responsável	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> Utilizando-se da integração sistema-bomba, assim que houver a necessidade de realizar um novo abastecimento devido a pouca quantidade de combustível no tanque, o sistema enviará um alerta via e-mail para o responsável, de forma que o mesmo realize a solicitação de abastecimento da bomba de combustível.		
<b>RF005</b> -Solicitar abastecimento da bomba de combustível	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> Após o responsável receber o alerta que a bomba chegou ao seu estoque mínimo, o sistema deve permitir que o mesmo solicite o abastecimento da bomba de combustível.		
<b>RF006</b> -Registrar entrada e saída de combustível	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> Assim que a solicitação for realizada pelo responsável e um novo lote de abastecimento for recebido, o sistema que estará integrado com a bomba deverá automaticamente registrar a nova entrada que foi realizada. Também serão registradas as solicitações de abastecimento realizadas pelo supervisor responsável.		
<b>RF007</b> -Atualizar estoque da bomba	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> Após a inserção da nova entrada de combustível, o estoque referente a quantidade atual presente na bomba será atualizado		

<b>RF008-</b> Validar veículos e motoristas	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> Nos lançamentos dos abastecimentos de combustível, será necessário validar a existência do veículo cuja placa foi inserida no sistema, consumindo os dados do provedor principal, assim como o motorista que detém o veículo.		
<b>RF009-</b> Exibir listagem de motoristas internos	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> Em casos de abastecimento para motoristas afiliados da empresa, após análise e verificação da placa, será necessária a criação de uma tela que liste todos os motoristas internos da empresa, a fim de selecionar o que se encontra utilizando o veículo com determinada placa (ação necessária devido a rotação de veículos internos).		
<b>RF010-</b> Pesquisar veículos e motoristas	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema possuirá um campo de busca para facilitar a filtragem dos dados buscados.		
<b>RF011-</b> Registrar motorista interno.	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> Após a seleção do motorista interno do veículo no qual está sendo abastecido, o mesmo será registrado no banco de dados simultaneamente com os dados do abastecimento.		
<b>RF012-</b> Registrar abastecimentos efetuados	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deverá calcular e registrar todos os abastecimentos efetuados, gerenciando o estoque atual para não estar inferior ao limite estipulado.		
<b>RF013-</b> Gerar comprovante de abastecimento	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> Após feito o cálculo e registro do abastecimento realizado, o sistema deverá gerar um XML com as informações completas do abastecimento, como meio de comprovar a ação.		
<b>RF014-</b> Imprimir o comprovante de abastecimento.	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa

<b>Descrição:</b> Após o documento de comprovante de abastecimento ser gerado, o sistema deve automaticamente fazer a impressão do mesmo, para que o motorista tenha um meio de comprovar a ação realizada.		
<b>RF015-</b> Gerar relatórios	<b>Categoria:</b> <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	<b>Prioridade:</b> <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve gerar relatórios mensais, através dos dados adquiridos de entradas e saídas de abastecimento da bomba de combustível.		

**Fonte: os autores**

### 3.4 Requisitos Não Funcionais

Os requisitos funcionais descrevem como o sistema deve funcionar, já os não funcionais explicam as limitações e restrições do sistema a ser projetado, basicamente eles vão descrever como o requisito funcional realizará tal tarefa.

Esses requisitos não tem nenhum impacto na funcionalidade do aplicativo. Além disso, existe uma prática comum de subclassificar os requisitos não funcionais em várias categorias, como:

- Interface de Usuário
- Confiabilidade
- Segurança
- atuação
- Manutenção
- Standards

Subclassificar os requisitos não funcionais é uma boa prática. Ajuda na hora de criar uma lista de verificação dos requisitos que devem ser atendidos no sistema a ser projetado.

**Quadro 3 – Requisitos Não Funcionais do sistema**

<b>RNF001-</b> Segurança de acesso	O acesso ao sistema se dará apenas através de uma conexão via VPN da empresa	Tipo Segurança	<input type="checkbox"/> Desejável <input checked="" type="checkbox"/> Obrigatório	<input checked="" type="checkbox"/> Permanente <input type="checkbox"/> Transitório
<b>RNF002-</b> Tela de login	O sistema contará com uma tela de login para inserção das credenciais previamente autorizadas de cada usuário	Tipo Segurança	<input type="checkbox"/> Desejável <input checked="" type="checkbox"/> Obrigatório	<input checked="" type="checkbox"/> Permanente <input type="checkbox"/> Transitório
<b>RNF003-</b> Uso limitado à desktop	O sistema será criado com a finalidade de ser utilizado em ambientes desktop	Tipo Compatibilidade	<input type="checkbox"/> Desejável <input checked="" type="checkbox"/> Obrigatório	<input checked="" type="checkbox"/> Permanente <input type="checkbox"/> Transitório

<b>RNF004-</b> Design Responsivo	Embora seja desenvolvido visando desktops, a responsividade é necessária em casos onde o acesso mobile seja requerido	Tipo Compatibilidade	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF005-</b> Acessibilidade	Padronizar a paleta de cores utilizada no sistema para facilitar a visualização de pessoas daltônicas	Tipo Performace	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF006-</b> Utilização de tags intuitivas	O sistema contará com o uso de tags HTML assistivas que auxiliam na utilização de leitores de telas	Tipo Performace	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF007-</b> Sistema Operacional Windows	O sistema deverá ser desenvolvido para ser utilizado em um ambiente Windows	Tipo compatibilidade	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF008-</b> Disponibilidade	A confiabilidade do sistema deverá ser extremamente alta, considerando a utilização para fins de abastecimento	Tipo Compatibilidade/Performace	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF009-</b> Linguagem de programação	A linguagem utilizada para desenvolvimento será React para a visualização do usuário e Node para o lado do servidor	Tipo Compatibilidade	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF010-</b> Banco de dados	O SGBD utilizado será MySQL visando a estabilidade e integridade dos dados.	Tipo Segurança	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF011-</b> Tempo de resposta	Ao se tratar de um sistema que terá uma frequência de uso elevada, possuir o menor tempo de resposta possível é essencial para o funcionamento.	Tipo Performance	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF013-</b> Especificações da impressão	A impressão do comprovante, caso o motorista deseje, será realizada em uma folha A4 e irá possuir a quantidade abastecida, o tipo de combustível e os dados do veículo e motorista	Tipo Compatibilidade	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório

Fonte: os autores

### 3.5 Regras de Negócio

Uma regra de negócio vai definir a necessidade de uma empresa, validando ou restringindo algo, por exemplo em uma plataforma para e-commercercs: “. Os clientes só vão ter acesso ao serviço de e-mail da plataforma, caso estejam no plano platinum”

ou seja, uma restrição a um serviço, definida pelos planos de pagamentos ofertados pela empresa.

Assim pode-se dizer que regras de negócio são limites impostos às operações, de forma que elas atendam todas as políticas e objetivos das empresas, e elas devem ser completas, com início, meio e fim, serem possíveis de mensurar, atualizadas e estar em consonância com a legislação.

**Quadro 4 – Regras de Negócio do sistema.**

<b>RN001 – Funcionário verificado</b>
<b>Descrição:</b> O frentista deve estar cadastrado no sistema pelo seu supervisor, para poder operar a bomba.
<b>RN002 - Somente pessoas cadastradas abasteceram</b>
<b>Descrição:</b> O motorista que quiser abastecer deve estar cadastrado no banco de dados da empresa, tendo autorização do gerente do setor. (TRIGGER)
<b>RN003 - Tratar os dados somente no sistema</b>
<b>Descrição:</b> Os dados do abastecimento devem ser movimentados no sistema.
<b>RN004 – Estoque mínimo</b>
A bomba deve ter o estoque mínimo de 3.000 litros para começar os abastecimentos. (TRIGGER)
<b>RN005 – Rotação de motoristas</b>
Existe rotação de motoristas para veículos internos
<b>RN006 – Comprovante de abastecimento</b>
No fim de cada abastecimento o frentista deve imprimir o comprovante de abastecimento a fim do motorista ter controle dos gastos.
<b>RN007 – Experiência comprovada</b>
O usuário deverá ter conhecimento básico de informática e experiência no manuseio da bomba de combustível
<b>RN008 - Experiência comprovada</b>
O funcionário responsável pelo abastecimento deverá possuir um treinamento comprovado na brigada de incêndio em casos em que seja necessário evitar possíveis acidentes.
<b>RN009 – Introdução do sistema</b>
Será disponibilizado um overview básico do funcionamento do sistema para que o funcionário possa se situar no uso do ambiente.

**Fonte: os autores**

### 3.6 Casos de Uso

Esse diagrama de UML é composto basicamente por quatro partes:

**Cenário:** Que são os eventos, de quando o usuário interage com o sistema.

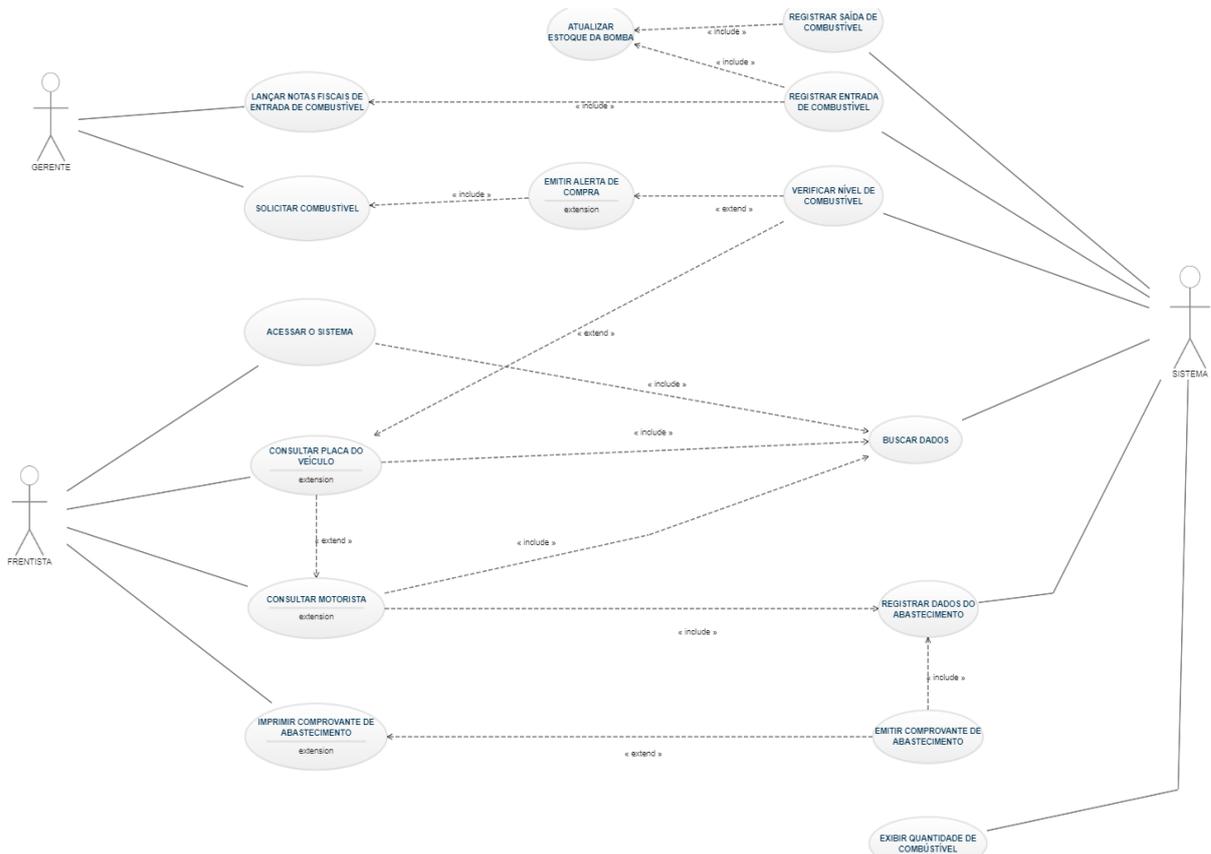
**Ator:** Um tipo de usuário do sistema.

**Use Case:** Tarefa ou funcionalidade realizada pelo ator.

**Comunicação:** Ligação do ator, com o caso de uso

Todas essas partes se unem para formar um cenário dos processos executados pelos usuários, representando cada etapa da tarefa e funcionalidades. Assim é possível ter uma visão ampla de como os processos devem funcionar, por isso em todo projeto deve se iniciar pelo desenho do sistema, e não na codificação.

**Figura 02: Diagrama de Caso de Uso**



Fonte: os autores

**Quadro 5 – Use Case: Acessar o sistema principal**

Caso de Uso - Acessar o sistema principal	
<b>ID</b>	UC 001
<b>Descrição</b>	Esse caso tem por objetivo efetuar o login do frentista no sistema
<b>Ator Primário</b>	Frentista

<b>Pré-condição</b>	Consumir os dados do sistema principal.
<b>Cenário Principal</b>	1 - Quando o sistema consome os dados da API da empresa. 2- Frentista preenche usuário e senha 3 - Sistema valida os dados de login e inicia a sessão.
<b>Pós-condição</b>	1 - Carregar dados do funcionário 2 - Campos obrigatórios preenchidos 3 - Login autorizado
<b>Cenário Alternativo</b>	Frentista não está cadastrado no banco de dados da empresa
<b>Inclusão</b>	Buscar dados
<b>Extensão</b>	Nenhuma

**Fonte: os autores**

#### **Quadro 6 – Use Case: Consultar placa**

<b>Caso de Uso - Consultar placa</b>	
<b>ID</b>	UC 002
<b>Descrição</b>	Esse caso tem por objetivo verificar se a placa do veículo está cadastrada, na base de dados da empresa.
<b>Ator Primário</b>	Frentista
<b>Pré-condição</b>	A placa do veículo já estar cadastrada
<b>Cenário Principal</b>	1 - Frentista preenche a numeração da placa 2- Sistema efetua a busca no banco de dados 3 - Frentista seleciona o veículo
<b>Pós-condição</b>	1 - Sistema carrega os dados dos veículos de transporte 2 - Verificação da placa 3 - Sistema traz os dados do veículo
<b>Cenário Alternativo</b>	1 - Placa não está cadastrada na base de dados 2 - Frentista pode sair do sistema a qualquer momento
<b>Inclusão</b>	Buscar dados
<b>Extensão</b>	Consultar motorista

**Fonte: os autores**

#### **Quadro 7 – Use Case: - Consultar motorista**

<b>Caso de Uso - Consultar motorista</b>	
<b>ID</b>	UC 003
<b>Descrição</b>	Esse caso tem por objetivo verificar se o motorista é interno ou externo.
<b>Ator Primário</b>	Frentista
<b>Pré-condição</b>	O motorista está ligado a uma placa de veículo ou a empresa.
<b>Cenário Principal</b>	1- Sistema efetua a busca no banco de dados 2 - Frentista seleciona o motorista
<b>Pós-condição</b>	1 - O sistema carrega os dados prontos para serem usados. 2 - O sistema traz os dados do motorista selecionado
<b>Cenário Alternativo</b>	Não possui
<b>Inclusão</b>	Buscar dados e Registrar os dados do abastecimento
<b>Extensão</b>	Não possui

Fonte: os autores

#### Quadro 8 – Use Case: Registrar dados do abastecimento

<b>Caso de Uso - Registrar dados do abastecimento</b>	
<b>ID</b>	UC 004
<b>Descrição</b>	Esse caso tem por objetivo salvar os dados da placa, motorista e a quantidade combustível abastecido no veículo.
<b>Ator Primário</b>	Sistema
<b>Pré-condição</b>	De todos os campos obrigatórios estarem preenchidos
<b>Cenário Principal</b>	1 - Frentista abastece o veículo
<b>Pós-condição</b>	1 - O sistema ao registrar a quantidade de combustível abastecida, atualiza o estoque da bomba de combustível.
<b>Cenário Alternativo</b>	1 - API da empresa estar fora do ar
<b>Inclusão</b>	Emitir comprovante de abastecimento
<b>Extensão</b>	Não possui

Fonte: os autores

#### Quadro 9 – Use Case: Imprimir comprovante de abastecimento

<b>Caso de Uso - Imprimir comprovante de abastecimento</b>	
<b>ID</b>	UC 005
<b>Descrição</b>	Esse caso tem por objetivo imprimir um comprovante com os dados do motorista, e a quantidade de combustível abastecida.
<b>Ator Primário</b>	FRENTISTA
<b>Pré-condição</b>	UC 004
<b>Cenário Principal</b>	1 - Após o registro dos dados de abastecimento, o sistema emite o comprovante. 2 - O frentista confirma, se o motorista quer ou não o comprovante
<b>Pós-condição</b>	1 - O sistema disponibiliza o comprovante para impressão. 2 - O sistema manda o arquivo para ser impresso.
<b>Cenário Alternativo</b>	1 - Se o motorista não quiser o comprovante, vai apenas ficar salvo no sistema
<b>Inclusão</b>	Não possui
<b>Extensão</b>	Não possui

**Fonte: os autores**

**Quadro 10 – Use Case: Emitir Relatórios**

<b>Caso de Uso - Emitir Relatórios</b>	
<b>ID</b>	UC 006
<b>Descrição</b>	Inclui gráficos para análise de dados e relatórios diversos, como, quantidade de combustível abastecido, assim como os principais motoristas, as rotas que geram mais consumo e etc.
<b>Ator Primário</b>	Sistema
<b>Pré-condição</b>	Conexão com o banco de dados, para consulta dos registros de saída e entrada de combustível.
<b>Cenário Principal</b>	1 - Emitir registros de abastecimentos por período, motorista ou placa. 2 - Emitir relatórios de custos por período, motorista ou placa.
<b>Pós-condição</b>	1 - Disponibiliza na tela do sistema todos os registros, dando possibilidade para extraí-los em .csv. 2 - Disponibiliza gráficos informativos dos custos, tendo possibilidade de extraí-los em .csv .
<b>Cenário Alternativo</b>	1 - Se não houver abastecimento, não terá dados para serem analisados. Assim não gerando o relatório.

<b>Inclusão</b>	Exibir a quantidade de combustível.
<b>Extensão</b>	Enviar relatório para o gerente.

Fonte: os autores

#### Quadro 11 – Use Case: Buscar Dados

<b>Caso de Uso - Buscar Dados</b>	
<b>ID</b>	UC 007
<b>Descrição</b>	Acessar os dados do BD da empresa, através de uma API e trazer os dados necessários para execução do sistema de gestão da bomba de combustível.
<b>Ator Primário</b>	Sistema
<b>Pré-condição</b>	Ter a API para fazer a ponte entre o sistema da empresa, o sistema de gestão da bomba de combustível.
<b>Cenário Principal</b>	1 - Conectar a API ao sistema da empresa e sistema de gestão da bomba.
<b>Pós-condição</b>	1 - Trazer os dados dos veículos, placas e outros dados necessários para execução do sistema.
<b>Cenário Alternativo</b>	Nenhum
<b>Inclusão</b>	Acessar o Sistema, Consultar Placa dos Veículos e Consultar Motorista.
<b>Extensão</b>	Não possui

Fonte: os autores

#### Quadro 12 – Use Case: Verificar Nível de Combustível

<b>Caso de Uso - Verificar Nível de Combustível</b>	
<b>ID</b>	UC 008
<b>Descrição</b>	Este caso tem por objetivo verificar o nível da bomba de combustível e alertar o gerente para solicitar combustível, caso o estoque esteja abaixo do nível mínimo.
<b>Ator Primário</b>	Sistema
<b>Pré-condição</b>	O sistema está conectado à bomba de combustível.
<b>Cenário Principal</b>	1 - Sistema deve estar conectado a bomba de combustível 2 - Nível mínimo de estoque é definido para verificação, em tempo real.

<b>Pós-condição</b>	1 - Informação de "OK" para utilização do sistema. 2 - Nível aparente para o frentista, para começar a abastecer os veículos.
<b>Cenário Alternativo</b>	1 - O sistema não está conectado à bomba, assim não possuindo dados para verificação. 2 - Nível mínimo da bomba não definido, assim impedindo a verificação, podendo ocasionar um furo de estoque.
<b>Inclusão</b>	Não possui
<b>Extensão</b>	Emitir alerta de compra

Fonte: os autores

**Quadro 13 – Use Case: Lançar notas fiscais de entrada de combustível**

<b>Caso de Uso - Lançar notas fiscais de entrada de combustível</b>	
<b>ID</b>	UC 009
<b>Descrição</b>	Este caso de uso tem como objetivo lançar as notas fiscais de combustível para alimentar a bomba de combustível.
<b>Ator Primário</b>	Gerente
<b>Pré-condição</b>	O sistema está conectado à bomba de combustível.
<b>Cenário Principal</b>	1 - Gerente da entrada na nota fiscal gerando estoque suficiente para o sistema liberar o abastecimento.
<b>Pós-condição</b>	1 - Estoque adiciona a quantidade de litros de combustível no sistema, atualizando o saldo da bomba.
<b>Cenário Alternativo</b>	1 - Nota fiscal está divergente da quantidade ou preço comprado.
<b>Inclusão</b>	Registrar entrada de combustível.
<b>Extensão</b>	Não possui

Fonte: os autores

**Quadro 14 – Use Case: Solicitar Combustível**

<b>Caso de Uso - Solicitar Combustível</b>	
<b>ID</b>	UC 010
<b>Descrição</b>	Este caso de uso tem como objetivo solicitar combustível, fazendo a compra do mesmo.
<b>Ator Primário</b>	Gerente
<b>Pré-condição</b>	A bomba de combustível está no nível próximo ao nível mínimo.

<b>Cenário Principal</b>	1 - O Gerente verifica o nível da bomba e se preciso, solicita a compra de combustível.
<b>Pós-condição</b>	1 - Pedido de compra é adicionado no sistema, aguardando a entrada da nota fiscal.
<b>Cenário Alternativo</b>	1 - A bomba tem combustível suficiente e não precisa de solicitar combustível.
<b>Inclusão</b>	Emitir alerta de compra
<b>Extensão</b>	Não possui

**Fonte: os autores**

#### **Quadro 15 – Use Case: Registrar Saída de combustível**

<b>Caso de Uso - Registrar Saída de combustível</b>	
<b>ID</b>	UC 011
<b>Descrição</b>	Este caso de uso tem como objetivo registrar a saída de combustível com os abastecimentos.
<b>Ator Primário</b>	Sistema
<b>Pré-condição</b>	Sistema disponível e com nível suficiente para abastecimento.
<b>Cenário Principal</b>	1 - Assim que o abastecimento for feito é registrado a saída de combustível da bomba.
<b>Pós-condição</b>	1 - Sistema verifica a bomba para continuar o abastecimento.
<b>Cenário Alternativo</b>	1 - Bomba não tem combustível suficiente para o abastecimento e registro da saída da bomba.
<b>Inclusão</b>	Atualizar estoque da bomba.
<b>Extensão</b>	Não possui

**Fonte: os autores**

#### **Quadro 16 – Use Case: Registrar Entrada de combustível**

<b>Caso de Uso - Registrar Entrada de combustível</b>	
<b>ID</b>	UC 012
<b>Descrição</b>	Este caso de uso tem como objetivo registrar a entrada de combustível com as notas e os pedidos de compra do gerente.
<b>Ator Primário</b>	Sistema
<b>Pré-condição</b>	Nota fiscal registrada no sistema.

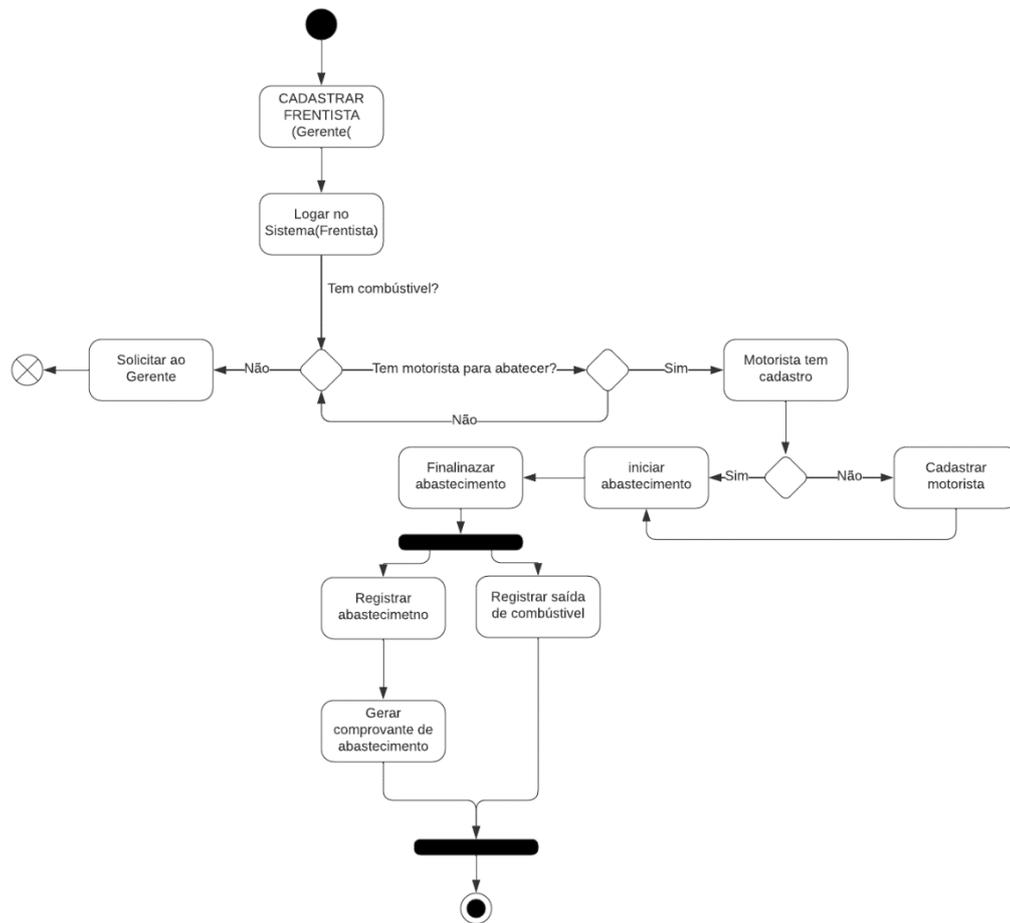
<b>Cenário Principal</b>	1 - Assim que a nota for dada entrada, o sistema adiciona o saldo no estoque para uso da empresa.
<b>Pós-condição</b>	1 - Sistema libera a bomba para abastecimento.
<b>Cenário Alternativo</b>	1 - A quantidade de combustível para dar entrada é maior que o limite da bomba.
<b>Inclusão</b>	Atualizar estoque da bomba. Lançar notas fiscais de entrada de combustível.
<b>Extensão</b>	Não possui.

Fonte: os autores

### 3.7 Diagrama de Atividades

Esse diagrama de UML é semelhante a fluxogramas porque mostra o fluxo entre as ações em uma atividade, mas ele também mostra fluxos paralelos e simultâneos e fluxos alternativos. O diagrama de atividades ilustra graficamente como será o funcionamento do software (em nível micro ou macro), como será a execução de alguma de suas partes, como será a atuação do sistema na realidade de negócio na ele está inserido. Através dele vai ser possível mostrar os requisitos funcionais e aspectos específicos, além de documentá-los.

Figura 03: Diagrama de Atividades



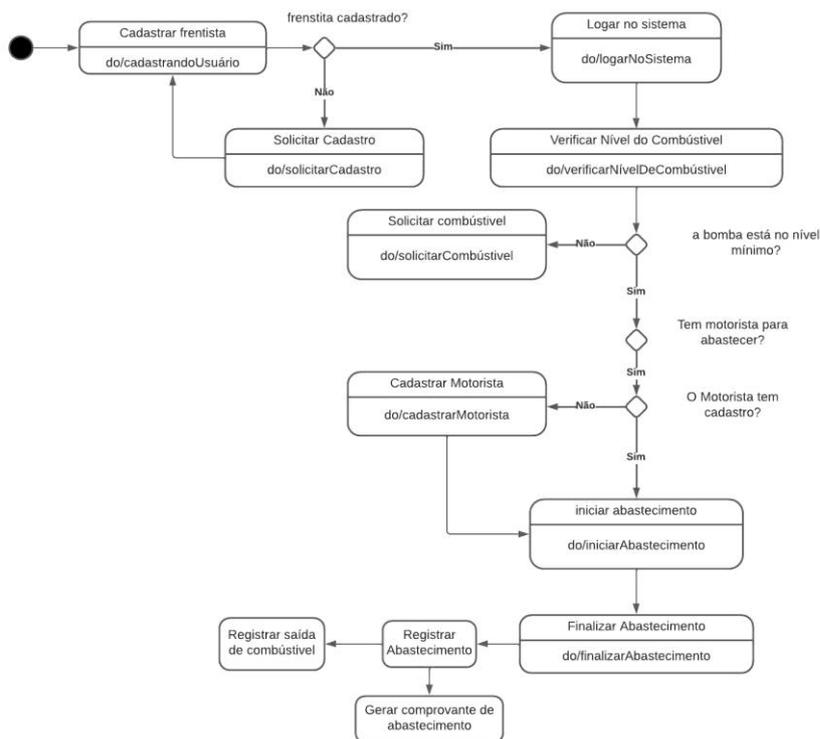
Fonte: os autores

### 3.8 Diagrama de Estados

Na modelagem UML, uma máquina de estado é uma especificação do comportamento dinâmico de objetos de classe individuais, casos de uso e sistemas inteiros. Os elementos que o compõem são:

- Estado(simples): Quando se tem uma condição ou evento na vida do objeto que atende a uma condição.
- Estado inicial: Inicia a modelagem dos estados de um elemento.
- Estado final: indicando o final dos estados modelados.
- Estado composto: que possui os sub-estados.

Figura 04: Diagrama de Estados



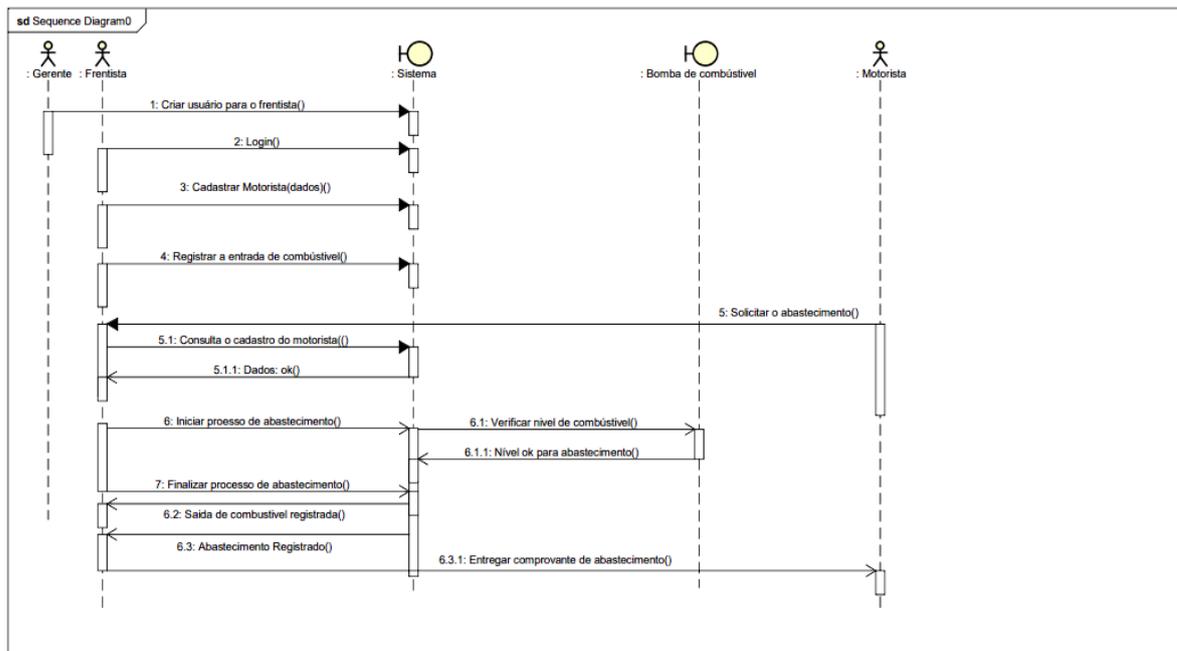
Fonte: os autores

### 3.9 Diagrama de Sequência

O diagrama de Sequência vai fazer a modelagem das interações entre objetos em um único caso de uso, mostrando diferentes partes do sistema interagindo entre si para realizar uma função, e a ordem em que as interações ocorrem quando um determinado caso de uso é executado.

Esse tipo de diagrama, é extremamente útil para sistemas mais complexos onde é necessário entender o comportamento do sistema em detalhes, pois ele representa através de linhas de vida um grupo de objetos, e as mensagens que eles trocam durante a interação.

Figura 05: Diagrama de Sequência



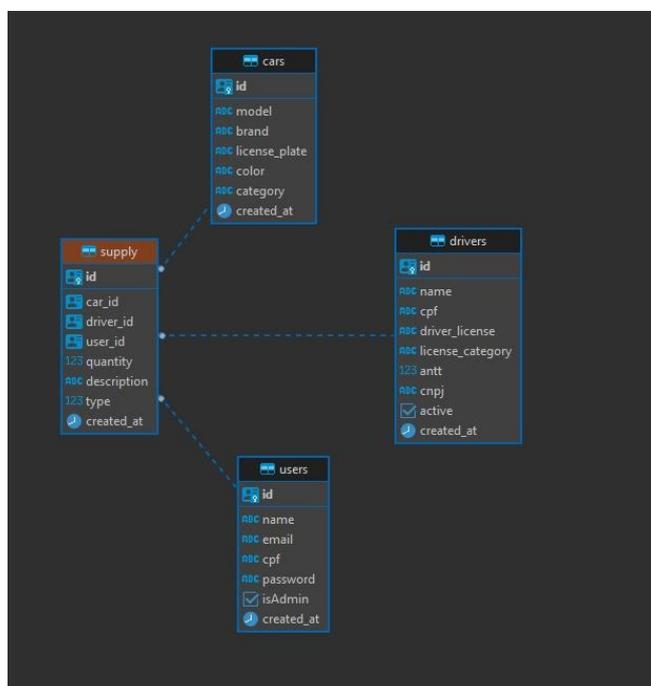
Fonte: os autores

### 3.10 Diagrama Entidade-Relacionamento

Nesse diagrama foi gerado após a estruturação do banco de dados, nele é exibido como as entidades (pessoas, objetos e conceitos) interagem.

Através dele é possível visualizar de forma mais clara a estrutura do sistema, o que facilita a identificar as lacunas entre os relacionamentos e fazer as correções necessárias, para adequar o funcionamento entre as funcionalidades.

Figura 06: Diagrama Entidade-Relacionamento



Fonte: os autores

### 3.11 Matriz de Rastreabilidade

Após o levantamento de requisitos, é necessário um refinamento e de um gerenciamento deles, assim surge a Matriz de Rastreabilidade, que não deixa a origem e o motivo do requisito se perderem, garantido que os objetivos do projeto e do negócio estejam alinhados no momento da entrega. Assim garantindo segurança, controle, precisão e clareza de informações, pois quando os objetivos estão bem definidos, eles vão auxiliar no acompanhamento do projeto.

A **Matriz de Rastreabilidade de Requisitos** é uma ferramenta validada internacionalmente e ela consta do Project Management Body of Knowledge (PMBOK®), do Project Management Institute (PMI), ou seja, uma das padronizações mais aceitas na gestão de projetos.

**Quadro 17: Matriz de Rastreabilidade de RF X RN**

	RN 01	RN 02	RN 03	RN 04	RN 05	RN 06	RN 07	RN 08	RN 09
RF01	X								
RF02			X	X					X
RF03	X		X	X					X
RF05			X	X					
RF06			X	X					
RF07			X	X			X		X
RF08		X			X		X		X
RF09		X			X		X		X
RF 10		X			X		X		X
RF11		X			X				X
RF12		X	X		X	X			
RF13		X	X		X	X			
RF14		X	X		X	X	X		
RF15			X						

Fonte: os autores

Aqui tanto os requisitos funcionais, quanto os requisitos não funcionais estão sendo relacionados com os casos de uso.

Na coleta de requisitos, o cliente relatou que necessitava de algumas transações no sistema para utilização em seu processo de negócios. Cada uma dessas transações representa um requisito funcional, ou seja, uma parte do sistema que deve ser construída. Para que um requisito funcional seja construído e especificado, ele dá origem a um Caso de Uso, na modelagem orientada a objetos. Portanto, um RF (Requisito Funcional) dará origem a um UC (Caso de Uso).

## **4 Ferramentas e Métodos ou Desenvolvimento**

### **4.1 Ferramentas**

As ferramentas escolhidas para o projeto foram selecionadas com base em sua eficiência, escalabilidade e suporte à comunidade. Além disso, essas ferramentas têm documentação abrangente, tutoriais e recursos disponíveis na comunidade de desenvolvedores, o que torna mais fácil para os desenvolvedores aprenderem e implementarem as soluções. A escolha dessas ferramentas também foi influenciada pela preferência pessoal da equipe de desenvolvimento e experiência prévia no uso delas. As licenças das ferramentas são de código aberto, o que significa que são gratuitas e podem ser usadas para fins comerciais e pessoais. Os sites oficiais para cada uma das ferramentas são:

SWOT = <https://miro.com/>

5W2H = <https://docs.google.com>

BPMN = <https://app.genmymodel.com/>

DOCUMENTAÇÃO = <https://docs.google.com>

TELAS = <https://chakra-ui.com/>

Miro é uma ferramenta de lousa online, que proporciona ferramentas para criar diagramas.

O docs do google nos ajudou com a criação da tabela 5W2H. É uma ferramenta bem completa que possui funcionalidades semelhantes ao Word da Microsoft, porém é grátis.

O GenMyModel é uma ferramenta que nos auxilia na criação de diagramas geralmente usados na elicitação de requisitos, como o BPMN.

Assim como mencionado acima, o docs nos auxiliou na criação da documentação do TG como um todo, ferramenta bem completa para configurações e tratamento de textos.

Chakra UI é uma ferramenta que reproduz templates em ReactJs para a criação de front-end de aplicações.

## **4.2 Métodos ou Desenvolvimento**

Durante o projeto, seguimos as etapas da metodologia Scrum, começando com o planejamento do projeto e a definição do escopo. Criamos um Backlog com as principais funcionalidades do sistema, e definimos as sprints e o Sprint Backlog com as tarefas específicas para cada sprint.

Ao longo do projeto, realizamos reuniões diárias de acompanhamento do projeto, a fim de garantir que as tarefas estivessem sendo realizadas conforme o planejado e que não houvesse impedimentos.

Desenvolvemos o projeto utilizando tecnologias famosas no mercado. No front usamos ReactJs utilizando templates do Chakra UI. No back-end usamos Node com framework Express. Subimos a aplicação com uma imagem no Docker e de banco usamos o Postgress.

A metodologia Scrum permitiu que desenvolvêssemos a documentação de forma iterativa e incremental, garantindo que as funcionalidades mais importantes fossem desenvolvidas primeiro. Além disso, a metodologia permitiu que respondêssemos rapidamente às mudanças que surgiram conforme o avanço do projeto. Assim foi possível entregar uma documentação para o Stakeholder sobre o sistema que pode ser implementado dentro da empresa, que vai resolver as dores apontadas por ele.

## **5 Resultados e Discussão (este item é obrigatório)**

Com a evolução do projeto, tivemos vários empasses que nos colocaram em situações de desafio à nossa posição. Tivemos que entrar em acordo quanto ao que seria realmente usável no sistema para não desperdiçarmos recursos no desenvolvimento. Em conversa entre o stakeholder e os integrantes do time entramos no consenso do que era realmente necessário para a empresa.

Outro ponto que nos levou à discussão foram as tecnologias a serem usadas, pois queríamos um software que entregasse desempenho e facilidade em seu uso.

Com a conversa fluindo entramos em acordo de usar o Javascript como linguagem principal do projeto, que nos forneceria todo o apoio necessário no desenvolvimento.

Como resultado geral do projeto, o que mais valeu foi a experiência de participar de um caso real do começo ao fim. Desde a proposta, passando pela elicitação de requisitos e finalmente desenvolvimento em código, tivemos uma ótima primeira (de muitas) convivências com a área, proporcionando momentos de baixa e alta intensidade e ampliando nosso conhecimento na área.

Nesta tela fazemos o cadastro dos usuários, motoristas e veículos.

**Figura 07: Cadastro de motoristas**

The screenshot shows a web interface for registering drivers. On the left, there is a sidebar with 'Cadastros' and 'Busca' tabs. Under 'Cadastros', 'Motoristas' is selected. The main form is titled 'Cadastro de motoristas' and contains the following fields:

- Nome: Text input field.
- CPF: Text input field.
- Registro CNH: Text input field.
- Categoria CNH: Dropdown menu with the text 'Selecione a categoria'.
- ANTT: Text input field.
- CNPJ: Text input field.
- A purple 'Cadastrar' button is located at the bottom left of the form.

Fonte: os autores

Nesta outra tela podemos consultar os motoristas, usuários e veículos cadastrados

**Figura 08: Listagem dos motoristas**

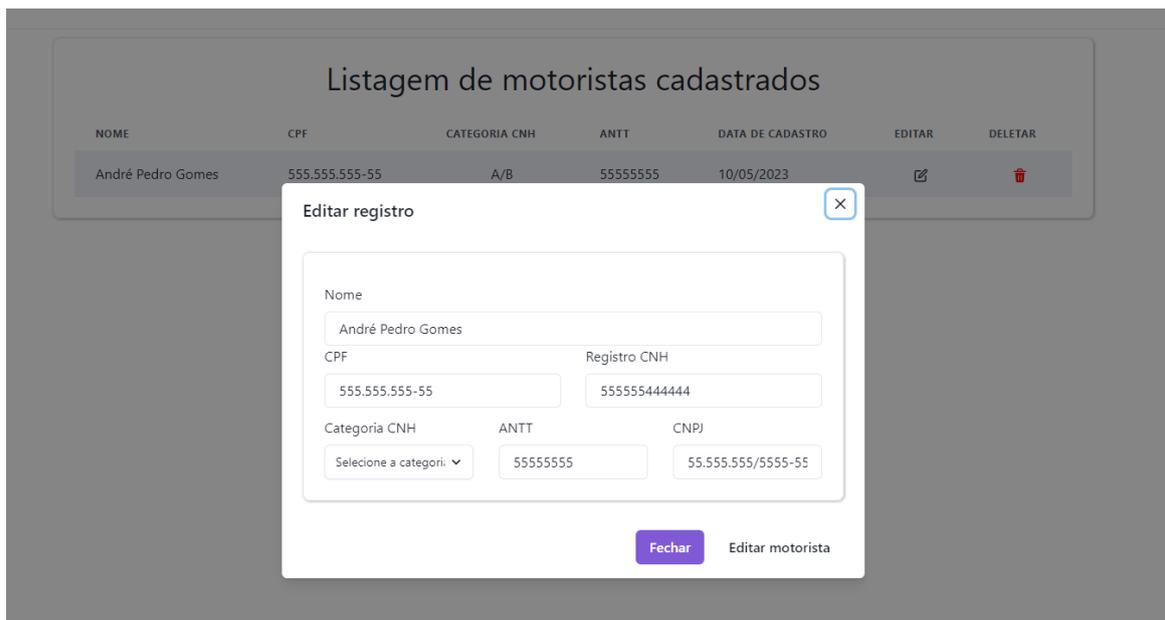
The screenshot shows a web interface for listing registered drivers. On the left, there is a sidebar with 'Busca' and 'Cadastros' tabs. Under 'Cadastros', 'Motoristas' is selected. The main table is titled 'Listagem de motoristas cadastrados' and has the following columns:

CPF	CATEGORIA CNH	ANTT	DATA DE CADASTRO	EDITAR	DELETAR
Andre Pedro Gomes	555.555.555-55	A/B	55555555	10/05/2023	[EDITAR] [DELETAR]

Fonte: os autores

Aqui podemos alterar os dados pesquisados.

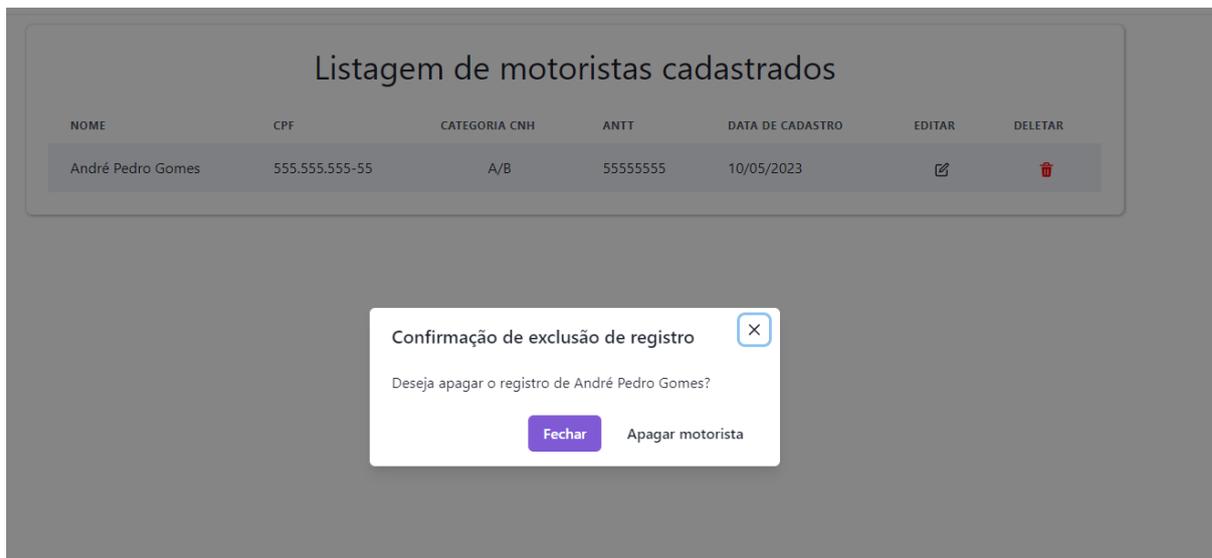
**Figura 09: Editar registros**



Fonte: os autores

Aqui podemos deletar os registros pesquisados.

Figura 10: Confirmação de exclusão



Fonte: os autores

Figura 11: Tela de código

```
@injectable()
class CreateDriverUseCase {
  constructor(
    @inject("DriversRepository")
    private driversRepository: IDriversRepository
  ) {}

  async execute({
    name,
    cpf,
    driver_license,
    license_category,
    antt,
    cnpj,
    active,
  }: IRequest): Promise<Drivers> {
    const driverAlreadyExist = await this.driversRepository.findByCpf(cpf);
    if (driverAlreadyExist) {
      throw new AppError("Motorista já existe!");
    }
    const driver = await this.driversRepository.create({
      name,
      cpf,
      driver_license,
      license_category,
      antt,
      cnpj,
      active,
    });

    return driver;
  }
}
```

Fonte: os autores

Aqui fazemos a verificação se o motorista já existe, se não ele cadastra usando o repository já criado.

Figura 12: Tela de código

```

@Inject()
class DeleteDriverUseCase {
  constructor(
    @Inject("DriversRepository")
    private driversRepository: IDriversRepository
  ) {}

  async execute(id: string): Promise<DeleteResult> {
    const driverExist = await this.driversRepository.findById(id);
    if (!driverExist) {
      throw new AppError("Motorista não existe!");
    }
    const deletedUser = await this.driversRepository.deleteDriver(id);

    return deletedUser;
  }
}

export { DeleteDriverUseCase };

```

Fonte: os autores

Nesta parte do código fazemos o delete do motorista, verificando primeiro se ele existe para prosseguir com a ação.

Figura 13: Tela de código

```

@Inject()
class GetAllDriversUseCase {
  constructor(
    @Inject("DriversRepository")
    private driversRepository: IDriversRepository
  ) {}

  async execute(): Promise<Drivers[]> {
    const drivers = await this.driversRepository.getAllDrivers();

    return drivers;
  }
}

export { GetAllDriversUseCase };

```

Fonte: os autores

Nesta parte executamos a busca dos motoristas, trazendo os dados que estão no banco.

Figura 14: Tela de código

```
@injectable()
class PutDriverUseCase {
  constructor(
    @inject("DriversRepository")
    private driversRepository: IDriversRepository
  ) {}

  async execute(
    id: string,
    {
      name,
      cpf,
      driver_license,
      license_category,
      antt,
      cnpj,
      active,
    }: IPutDriverDTO
  ): Promise<UpdateResult> {
    const driverExist = await this.driversRepository.findById(id);
    if (!driverExist) {
      throw new AppError("Driver do not exist");
    }
    const updatedDriver = await this.driversRepository.updateDriver(id, {
      name,
      cpf,
      driver_license,
      license_category,
      antt,
      cnpj,
      active,
    });

    return updatedDriver;
  }
}
```

Fonte: os autores

Aqui esta sendo feita a alteração dos dados pelo método Put, verificando primeiro se o motorista existe.

### **Considerações finais**

Ao finalizar o projeto de desenvolvimento de um sistema para uma bomba de combustível dentro de uma empresa de transportes, gostaríamos de ressaltar as conquistas alcançadas e os desafios superados. O objetivo principal era criar um sistema eficiente e inovador que reduzisse os custos da empresa, eliminasse falhas de dados e proporcionasse um maior controle sobre as informações. Além disso, tínhamos em mente a possibilidade de expansão do sistema para outras empresas, não apenas do setor de transportadoras.

Considerando a expansão do sistema para outras empresas, acreditamos que os resultados alcançados podem ser replicados em diferentes setores da indústria. O sistema foi projetado de forma modular e escalável, o que facilita sua adaptação a diferentes contextos e necessidades. Com as devidas customizações, é possível implantar o sistema em empresas de logística, transportadoras, postos de combustível e até mesmo em frotas de veículos de outros segmentos. Essa possibilidade de expansão torna o projeto uma solução atrativa para um público mais amplo, abrindo oportunidades de mercado promissoras.

Para o futuro, temos intenções de masterizar este projeto e trabalhar não apenas na empresa informada, mas também ampliar nossa área de prospecção para o agro, onde temos uma grande oportunidade de fortalecer este movimento dando mais importância para o controle e profissionalismo desta área tão importante para a população.

Em conclusão, o projeto de desenvolvimento do sistema para a bomba de combustível atingiu com sucesso seus objetivos iniciais. Foi possível eliminar falhas de dados e proporcionar um controle mais amplo sobre as informações. Além disso, a perspectiva de expansão para outras empresas demonstra o potencial e a versatilidade do sistema. Com uma implementação bem-sucedida, acreditamos que o projeto pode ser um importante diferencial competitivo para as empresas que o adotarem, contribuindo para a melhoria de sua eficiência operacional e para o aumento de sua rentabilidade.

## Referências

### Referências da internet:

ARTIA. EAP (Estrutura Analítica do Projeto). Disponível em: <<https://artia.com/blog/como-fazer-eap-na-gestao-de-projetos/>>. Acesso em: 25 abr. 2023.

EUAX. TUDO sobre MATRIZ SWOT: o que é, passo a passo e dicas para aplicar no planejamento estratégico. Disponível em: <<https://www.euax.com.br/2020/03/matriz-swot/>>. Acesso em: 22 abr. 2023.

RETRAINING. Técnicas de Elicitação de Requisitos: Entrevista. Disponível em: <<https://retraining.inf.ufsc.br/guia/app/classificacoes/tecnicas-de-elicacao-de-requisitos/entidades/tecnicas-de-elicacao-de-requisitos-entrevista>>. Acesso em: 25 abr. 2023.

RETRAINING. Técnicas de Elicitação de Requisitos: Card sorting. Disponível em: <<https://retraining.inf.ufsc.br/guia/app/classificacoes/tecnicas-de-elicacao-de-requisitos/entidades/tecnicas-de-elicacao-de-requisitos-card-sorting>>. Acesso em: 25 abr. 2023.

RETRAINING. Técnicas de Elicitação de Requisitos: Prototipação. Disponível em: <<https://retraining.inf.ufsc.br/guia/app/classificacoes/tecnicas-de-elicacao-de-requisitos/entidades/tecnicas-de-elicacao-de-requisitos-prototipacao>>. Acesso em: 23 abr. 2023.

RETRAINING. Técnicas de Elicitação de Requisitos: Brainstorming. Disponível em: <<https://retraining.inf.ufsc.br/guia/app/classificacoes/tecnicas-de-elicacao-de-requisitos/entidades/tecnicas-de-elicacao-de-requisitos-brainstorming>>. Acesso em: 23 abr. 2023.

LUCIDCHART. O que é BPMN. Disponível em: <[https://www.lucidchart.com/pages/pt/o-que-e-bpmn#discovery\\_\\_top](https://www.lucidchart.com/pages/pt/o-que-e-bpmn#discovery__top)>. Acesso em: 25 abr. 2023.

SEBRAE. 5W2H: o que é, para que serve e por que usar na sua empresa. Disponível em: <<https://www.sebrae-sc.com.br/blog/5w2h-o-que-e-para-que-serve-e-por-que-usar-na-sua-empresa>>. Acesso em: 29 abr. 2023.

ACERTBR. Técnicas de elicitação de requisitos, 10 jan. 2022. Disponível em: <<https://acertbr.com.br/tecnicas-de-elicitaçao-de-requisitos/#:~:text=Essa%C3%A9cnica%20consiste%20em%20reuni%C3%B5es,o%20sistema%20que%20ser%C3%A1%20desenvolvido.>>. Acesso em: 29 abr. 2023.

FIA BUSINESS SCHOOL. Card Sorting: O que é e como utilizar (Guia Completo). Disponível em: <<https://fia.com.br/blog/card-sorting-o-que-e-e-como-utilizar-guia-completo/>>. Acesso em: 25 abr. 2023.

TOTVS. BPMN: entenda o que é a modelagem de processos de negócios, como fazer e sua importância, 10 mar. 2023. Disponível em: <<https://www.totvs.com/blog/gestao-industrial/bpmn/>>. Acesso em: 30 abr. 2023.

VISUREOLUTIONS. O que são Requisitos Funcionais: Exemplos, Definição, Guia Completo. Disponível em: <<https://visuresolutions.com/pt/blog/functional-requirements/>>. Acesso em: 27 abr. 2023.

SYDLE. O que são Regras de Negócio? Qual a importância? Veja exemplos, 06 abr. 2022. Disponível em: <<https://www.sydle.com/br/blog/regras-de-negocio-5f6333be1e43744c69d995e0>>. Acesso em: 27 abr. 2023.

ATEOMOMENTO. Entendendo o Diagrama de Atividades da UML, 26 out. 2016. Disponível em: <<https://www.ateomomento.com.br/uml-diagrama-de-atividades/>>. Acesso em: 28 abr. 2023.

ROBSONCAMARGO. Como a matriz de rastreabilidade ajuda no gerenciamento de projetos, 28 mai. 2019. Disponível em: <<https://robsoncamargo.com.br/blog/matriz-de-rastreabilidade/>>. Acesso em: 28 abr. 2023.

## Anexo 01

### Perguntas e respostas realizadas na elicitação de requisitos.

1. Como funciona o posto em geral?

**R:** Existem dois tipos de motoristas: os externos e os internos, os externos abastecem a um valor reduzido para realizar o frete e assim que for realizado, são pagos descontando o valor abastecido.

Sempre que a bomba chega em um determinado estoque limite, é necessário pedir um novo carregamento de combustível.

O valor a ser descontado do frete, assim como a quantidade abastecida é registrada em papel para que eventualmente seja inserida em uma planilha Excel para controle.

2. O controle do abastecimento é necessário?

**R:** Sim, muito necessário

3. O que podemos fazer para melhorar o controle do processo de abastecimento?

**R:** Automatizar o processo de entrada dos dados de abastecimento e gerenciar isso pelo software para evitar perda de dados sensíveis.

4. Existe alguma perda de material no abastecimento?

**R:** Durante o abastecimento é zero.

5. Você tem um controle do quanto é gasto por abastecimento para os caminhões internos e externos?

**R:** Não existe um controle disso, atualmente, caso seja necessário analisar esses dados, todas as planilhas devem ser consultadas.

6. Vocês têm o controle do custo da operação de abastecimento?

**Resposta:** Não tem.

7. Você acha uma planilha, um banco de dados eficiente?

**Resposta:** Não, pois os dados ficam desorganizados e em lugares diferentes.

8. Você tem controle de quais funcionários realizam o abastecimento?

**Resposta:** Não de uma forma eficiente (apenas por papel).

9. Os funcionários podem abastecer veículos próprios? Se sim, existe um limite?

**Resposta:** Apenas os donos e família podem abastecer veículos próprios.

10. Existe um controle de manutenção preventiva dos veículos?

**Resposta:** Não tem.

11. Há uma conferência se a pessoa que está abastecendo pode realmente abastecer? Existe um controle se o veículo terceirizado pode abastecer?

**Resposta:** Não existe conferência, o controle é realizado na guarita, apenas por contato visual.

12. E se o veículo terceirizado quiser realizar o pagamento durante o abastecimento?

**Resposta:** O motorista não possui essa opção.

13. Se o valor abastecido for maior do que o frete que será recebido pelo terceirizado, como funciona? Existe um controle se ele vai pagar o valor devido?

**Resposta:** O valor fica pendente e é pago no próximo frete, o valor fica em haver e depois ele faz um frete para cobrir o valor.

14. Quais relatórios serão necessários?

**Resposta:** Relatórios mensais para análise de entrada, saída, valores e dados da bomba de combustível.

15. Você tem alguma sugestão de funcionalidade que seria útil?

**Resposta:** Integrar o software com o GPS para ver, em tempo real, a localização do motorista interno.