

# **Carga Explosiva**

## **Documento de Arquitetura de Software**

**Versão 1.0**

Documento apresentado na disciplina  
Projeto de Software para o curso de  
Graduação em Ciência da Computação  
da Universidade Federal Fluminense,  
ministrado pelo docente Vânia de  
Oliveira Neves

**Barbara Keren N. C. Guarino  
Caio Guimarães Cardoso  
Gabriel Ramalho Braga  
Gyselle Regina Gonçalves de Mello**

# Sumario

<b>Histórico de Revisão.....</b>	<b>3</b>
<b>Introdução.....</b>	<b>4</b>
Finalidade.....	4
Definições, Acrônimos e Abreviações.....	4
Escopo.....	4
<b>Requisitos Arquiteturalmente Significantes.....</b>	<b>6</b>
<b>Metas e Restrições Arquiteturais.....</b>	<b>7</b>
<b>Padrões Arquiteturais.....</b>	<b>9</b>
<b>Mecanismos Arquiteturais.....</b>	<b>11</b>
<b>Decisões e Justificativas.....</b>	<b>13</b>
<b>Representação Arquitetural.....</b>	<b>15</b>
<b>Diagramas da Arquiteturas.....</b>	<b>15</b>
Casos de Uso.....	15
Diagrama de Sequência do Sistema.....	17
Contratos das Operações.....	23
Diagrama de Classe de Análise.....	26
Diagrama de Comportamento.....	26
Diagrama de Classes Detalhado.....	29
<b>Dimensionamento e Performance.....</b>	<b>29</b>
Volume.....	29
Performance.....	29
<b>Qualidade.....</b>	<b>30</b>
<b>Documentação.....</b>	<b>31</b>

## Histórico de Revisão

Data	Versão	Descrição	Autor
11/04/2024	01.00.00	Criação do Documento	Barbara
10/05/2024	01.01.00	Conclusão Parcial do Documento	Barbara, Gyselle, Gabriel e Caio

# Introdução

Este documento apresenta uma visão abrangente da arquitetura do sistema de gestão de frota desenvolvido pela [equipe]. O sistema visa proporcionar uma solução completa para empresas de transporte e logística, permitindo o gerenciamento eficiente de veículos, motoristas e manutenção. Esta introdução fornecerá uma visão geral do conteúdo do documento, destacando os principais aspectos da arquitetura e sua importância para o sucesso do sistema.

# Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema [nome do sistema], usando diversas visões arquitetônicas para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquitetônicas significativas que foram tomadas em relação ao sistema.

# Definições, Acrônimos e Abreviações

Definições, Acrônimos e Abreviações	Descrição
Frota	Conjunto de veículos pertencentes a uma empresa ou organização
Usuário	Pessoa autorizada a interagir com o sistema de gestão de frota.
API	Interface de Programação de Aplicativos.
CRUD	Create, Read, Update, Delete (Criar, Ler, Atualizar, Excluir)
SQL	Structured Query Language (Linguagem de Consulta Estruturada).
UI	User Interface (Interface do Usuário).
GPS	Global Positioning System (Sistema de Posicionamento Global).
SMTP	Simple Mail Transfer Protocol (Protocolo de Transferência de Correio Simples).

# Escopo

Este Documento de Arquitetura de Software se aplica ao sistema Carga Explosiva, que será desenvolvido pela Carga Explosiva visando empresas de transportes e logísticas.

O sistema trata-se de um software para gestão de frota com suporte para desktops e smartphones, com o objetivo de facilitar o gerenciamento eficiente dos veículos da frota, assim como o gerenciamento de motoristas e da oficina própria que a empresa possui para pequenas manutenções corretivas e preventivas.

O sistema incluirá funcionalidades abrangentes como o cadastro e gestão detalhada de veículos, permitindo acompanhar informações como modelo, placa, ano de fabricação, número do chassi, capacidade de carga, histórico de manutenções, histórico de multas, histórico de sinistros, quilometragem, histórico de abastecimento, entre outros dados vinculados ao veículo.

Outra funcionalidade crucial será o gerenciamento dos motoristas. O sistema permitirá o cadastro completo dos motoristas, incluindo dados pessoais, documentos, habilitação, treinamento, histórico de condução e experiência profissional. Cada motorista poderá ser atribuído a veículos específicos, facilitando o controle das escalas de trabalho e dos períodos de condução.

No que diz respeito a manutenção, o sistema oferece recursos de agendamentos e registros de manutenção, permitindo que os usuários possam agendar serviços preventivos e corretivos, planejar manutenções periódicas com base em quilometragem ou tempo de uso, registrar todas as atividades de manutenção realizadas em cada veículo, além de acompanhar custos, materiais e peças utilizadas em cada manutenção.

Além disso, o sistema controlará o estoque e serviços prestados pela oficina que a empresa possui, permitindo o registro de entradas, saídas, transferências e ajustes dos serviços prestados, assim como, de estoque de peças e materiais. Alertas automáticos serão gerados em caso de estoque baixo, facilitando a reposição de itens essenciais.

Por fim, o sistema fornecerá uma variedade de relatórios e análises sobre o desempenho da frota e das operações de manutenção. Esses relatórios serão personalizáveis e permitirão uma avaliação detalhada do custo total de propriedade, da eficiência operacional, do consumo de combustível e da produtividade dos motoristas.

Em síntese, o sistema de gestão de frota será uma ferramenta essencial para otimizar as operações das empresas de transporte e logística, garantindo um controle detalhado, levando a tomadas de decisão embasadas em dados e redução significativa de custos operacionais.

## Requisitos Arquiteturalmente Significantes

Nesta seção, serão apresentados os requisitos arquitetônicos significativos para o sistema de gestão de frota. Esses requisitos são essenciais para a definição da arquitetura do software e são baseados nas necessidades funcionais, qualidades desejadas e restrições do sistema. Ao identificar e descrever esses requisitos, busca-se garantir que a arquitetura dos sistemas atenda plenamente às expectativas de integração, desempenho, segurança e usabilidade, contribuindo assim para o sucesso e eficiência do sistema como um todo.

Requisitos	Descrição
Integração com Dispositivos Móveis	Necessidade de suportar desktops e smartphones, garantindo integração e sincronização eficientes entre diferentes plataformas.
Segurança	Proteção dos dados contra acessos não autorizados, garantindo confidencialidade e integridade.
Desempenho	Garantia de um desempenho eficiente, especialmente para operações em tempo real.
Flexibilidade e Manutenibilidade	Capacidade de suportar mudanças e atualizações sem impactar negativamente a estabilidade ou funcionalidade do sistema.
Disponibilidade	Garantia de disponibilidade do sistema durante a maior parte do tempo.
Persistência	Necessidade de armazenar e recuperar dados de forma eficiente e confiável.
Autenticação e Autorização	Garantir que apenas usuários autorizados tenham acesso ao sistema e a determinadas funcionalidades.
Armazenamento de Arquivos e Imagens	Capacidade de armazenar e recuperar arquivos e imagens relacionados à frota e oficina.
Envio de E-mail	Capacidade de enviar e-mails para comunicação e notificações.
Localização em Tempo Real	Necessidade de rastrear a localização em tempo real dos veículos da frota.
Interface de Usuário Dinâmica e Amigável	Exigência de um interface intuitiva e responsiva para os usuários interagirem facilmente com o sistema.
Reuso de Componentes	Capacidade de reutilizar componentes de interface do usuário e lógica de negócios para minimizar a duplicação de código e

Requisitos	Descrição
	facilitar a manutenção, seguindo padrões de design e boas práticas de desenvolvimento.
Geração de Relatórios	Necessidade de gerar relatórios personalizáveis sobre o desempenho da frota e das operações de manutenção.

## Metas e Restrições Arquiteturais

A arquitetura do sistema de gestão de frota será guiada por uma abordagem centrada na integração, confiabilidade e usabilidade. O sistema será projetado para atender às necessidades complexas e multifacetadas de gerenciamento de frota, integrando-se de forma eficiente com sistemas legados e oferecendo alto desempenho em condições operacionais e diversas. A robustez durante um longo período de manutenção será uma prioridade, garantindo que o sistema permaneça estável e fácil de manter ao longo do tempo. Além disso, a segurança dos dados e das comunicações será uma preocupação central, assegurando a integridade e confidencialidade das informações em todas as integrações do sistema. A filosofia da arquitetura busca criar uma solução flexível, modular e altamente adaptável, capaz de evoluir com as demandas do mercado e as necessidades em constante mudança dos usuários.

As metas arquitetônicas definem os objetivos que orientarão o design e o desenvolvimento do sistema de gestão de frota, visando garantir sua eficácia, confiabilidade e usabilidade. A arquitetura do sistema será projetada com foco na:

Metas	Descrição
Confiabilidade	A arquitetura deve ser projetada para garantir a confiabilidade do sistema, minimizando o tempo de inatividade não planejado e garantindo a disponibilidade contínua dos serviços.
Usabilidade	A arquitetura deve priorizar a usabilidade, proporcionando uma interface de usuário intuitiva e amigável para garantir uma experiência de uso positiva para os usuários.
Segurança	A arquitetura deve garantir a segurança dos dados e das comunicações, protegendo o sistema contra ameaças externas e internas, e garantindo a integridade e confidencialidade dos dados.
Manutenibilidade	A arquitetura deve ser projetada para facilitar a manutenção do sistema, permitindo atualizações e modificações sem impactar

## Carga Explosiva

Metas	Descrição
	negativamente a estabilidade ou a funcionalidade do sistema.
Portabilidade	A arquitetura deve garantir a portabilidade do sistema, permitindo que ele seja executado em diferentes plataformas e ambientes de execução, como desktops e dispositivos móveis.
Modularidade	A arquitetura deve ser modular, permitindo a divisão do sistema em componentes independentes, facilitando a manutenção e evolução do sistema e promovendo o reuso de componentes.
Tolerância a Falhas	A arquitetura deve ser projetada para lidar com falhas de componentes individuais, garantindo a disponibilidade contínua do sistema e minimizando o impacto de falhas no funcionamento do sistema.
Eficiência	A arquitetura deve garantir a eficiência do sistema, otimizando o uso de recursos e garantindo um desempenho adequado mesmo em condições de carga elevada ou recursos limitados.
Interação	A arquitetura deve facilitar a integração com sistemas externos, permitindo a troca de dados e a comunicação sem problemas entre diferentes sistemas e serviços.

Para garantir o sucesso do sistema de gestão de frota, é essencial identificar e considerar cuidadosamente as restrições arquitetônicas que podem impactar o design e a implementação da solução. Essas restrições representam limitações e requisitos não negociáveis que devem ser levados em consideração ao definir a arquitetura do sistema.

Restrições	Descrição
Conformidade com Padrões	A arquitetura deve aderir a padrões de desenvolvimento de software reconhecidos e amplamente aceitos, garantindo a interoperabilidade e a manutenção do sistema.
Tecnologia de Plataforma	O sistema deve ser desenvolvido utilizando tecnologias que suportem tanto desktop quanto dispositivos móveis, garantindo uma experiência consistente em todas as plataformas.
Segurança	A arquitetura deve priorizar a segurança, incorporando práticas e mecanismos robustos para proteger os dados contra acesso não autorizado e ataques cibernéticos.
Desempenho	O sistema deve ser otimizado para garantir um desempenho eficiente, minimizando os tempos de resposta e maximizando a capacidade de processamento em todas as condições de uso.



Restrições	Descrição
Confiabilidade	A arquitetura deve ser projetada para garantir a confiabilidade do sistema, minimizando o tempo de inatividade não planejado e as interrupções nas operações de frota.

Considerando as metas e restrições definidas, a arquitetura do sistema de gestão de frota será projetada para atender não apenas aos requisitos funcionais, mas também às expectativas de confiabilidade, usabilidade e segurança. Ao identificar e abordar esses elementos desde o início do processo de design, buscamos garantir que o sistema seja robusto, capaz de manter a disponibilidade contínua das operações da frota, e capaz de evoluir com as necessidades do negócio e as demandas do mercado.

## Padrões Arquiteturais

A arquitetura de software desempenha um papel fundamental na concepção e implementação de sistemas robustos e eficientes. Para o desenvolvimento do sistema de gestão de frota, optamos por adotar uma abordagem que combina elementos da arquitetura **Cliente-Servidor** com a estrutura em **Camadas**. Essa mistura de padrões arquiteturais visa aproveitar as vantagens de cada abordagem, proporcionando uma solução que atende às necessidades complexas e multifacetadas do sistema.

A mistura das arquiteturas Cliente-Servidor e em Camadas oferece uma abordagem flexível e escalável para o desenvolvimento do software em desenvolvimento. Nesta abordagem, o sistema é dividido em duas partes principais: o cliente e o servidor. No lado cliente, são desenvolvidos aplicativos desktop e móveis que atuam como interfaces de usuário para interação com o sistema. No lado servidor, a lógica de processamento e armazenamento de dados é organizada em camadas distintas: apresentação, lógica de negócios e persistência.

### Camadas no Lado do Cliente:

**Camada de Segurança (Autenticação do Cliente):** Esta camada é responsável por autenticar os usuários do sistema, garantindo que apenas usuários autorizados possam acessar as funcionalidades do sistema. Aqui, os usuários fornecem suas credenciais de login para se autenticar no sistema antes de poderem interagir com a interface do usuário.

**Camada de Apresentação (Interface do Usuário):** Nesta camada, são desenvolvidas as

interfaces do usuário que os clientes interagem para acessar e utilizar o sistema. Aqui, os aplicativos desktop e móveis são criados para proporcionar uma experiência intuitiva e responsiva aos usuários, permitindo que eles visualizem e interajam com as informações de maneira eficiente.

**Camada de Aplicação (Lógica do Cliente):** Esta camada é responsável por processar a entrada do usuário e preparar as solicitações para serem enviadas ao servidor. Aqui, a lógica de negócio relacionada à interação do cliente com o sistema é implementada, garantindo que as operações sejam realizadas de acordo com as regras estabelecidas e que os dados sejam formatados corretamente antes de serem enviados.

**Camada de Comunicação (Comunicação de Rede):** Nesta camada, são estabelecidas e gerenciadas as comunicações entre o cliente e o servidor. Aqui, são implementados os métodos necessários para enviar e receber dados do servidor, garantindo uma comunicação eficiente e confiável entre as partes.

### **Camadas no Lado do Servidor:**

**Camada de Segurança (Autenticação e Autorização):** Esta camada é responsável por garantir a autenticação e autorização dos usuários, bem como a proteção dos dados contra acessos não autorizados. Ela inclui a implementação de mecanismos de autenticação, como login com nome de usuário e senha, e autorização baseada em papéis para controlar o acesso às funcionalidades do sistema. Além disso, ela gerencia sessões e tokens de autenticação para garantir a segurança contínua do sistema.

**Camada de Apresentação (Recebimento de Requisições):** Responsável por receber as solicitações dos clientes e encaminhá-las para o processamento adequado. Aqui, os endpoints de API são implementados para lidar com as requisições dos clientes, garantindo que cada solicitação seja roteada para o destino correto dentro do servidor.

**Camada de Aplicação (Lógica de Negócios):** Esta camada é responsável por processar as solicitações recebidas dos clientes e realizar operações no sistema. Aqui, a lógica de negócios é implementada para executar as operações necessárias, como cadastro de veículos, atribuição de motoristas, entre outros, de acordo com as regras de negócio estabelecidas.

**Camada de Persistência (Acesso a Dados):** Responsável por interagir com o banco de dados para armazenar e recuperar informações sobre veículos, motoristas, manutenção, entre

outros. Aqui, são implementados os métodos para realizar operações de CRUD no banco de dados, garantindo que os dados sejam armazenados e recuperados de forma eficiente e segura.

A mistura das arquiteturas Cliente-Servidor e em Camadas oferece uma solução robusta e flexível para o sistema de gestão de frota. Ao combinar elementos de centralização, segurança e controle do servidor com a flexibilidade, reutilização de componentes e manutenibilidade das camadas separadas, conseguimos desenvolver um sistema que atende às necessidades complexas e em constante evolução do nosso cenário operacional. Esta abordagem arquitetural permite confiabilidade e eficiência, garantindo uma experiência positiva para os usuários finais e uma base sólida para futuras expansões e atualizações do sistema.

## Mecanismos Arquiteturais

Os mecanismos arquiteturais são elementos essenciais para garantir o funcionamento adequado e eficiente de um sistema de software. Eles representam as estratégias e técnicas utilizadas para lidar com aspectos cruciais da arquitetura, como persistência de dados, comunicação entre componentes e tratamento de erros. Nesta seção, serão listados e descritos diversos mecanismos arquitetônicos utilizados no sistema, destacando suas finalidades, atributos e funções.

Mecanismo Arquitetural	Finalidade	Atributos	Funções
Persistência de Dados	Garantir a persistência e recuperação eficiente de dados no sistema, permitindo que informações importantes sejam armazenadas de forma confiável.	Confiabilidade, Eficiência, Consistência	Realizar operações de CRUD no banco de dados, gerenciar transações, otimizar consultas e garantir a integridade dos dados.
Comunicação entre Componentes	Facilitar a comunicação e troca de dados entre os diversos componentes do sistema, permitindo a integração e interação entre eles.	Confiabilidade, Eficiência, Segurança.	Implementar protocolos de comunicação, gerenciar mensagens, garantir a entrega e recebimento de dados de forma assíncrona ou síncrona.
Tratamento de Erros	Lidar com situações de exceção e erros de forma	Robustez, Resiliência,	Capturar exceções, registrar erros, notificar

## Carga Explosiva

Mecanismo Arquitetural	Finalidade	Atributos	Funções
	adequada, minimizando impactos negativos no funcionamento do sistema e proporcionando uma experiência de usuário consistente.	Monitoramento.	usuários ou administradores, implementar estratégias de fallback e recuperação.
Autenticação e Autorização	Garantir que apenas usuários autorizados tenham acesso ao sistema e a determinadas funcionalidades, protegendo os dados contra acessos não autorizados.	Segurança, Privacidade, Confiabilidade.	Verificar credenciais de usuário, gerenciar sessões e token de autenticação, controlar permissões de acesso a recursos e funcionalidades.
Geração de Relatórios	Permitir a geração e personalização de relatórios sobre o desempenho da frota e das operações de manutenção, fornecendo insights valiosos para tomada de decisões.	Flexibilidade, Eficiência, Customização.	Coletar dados relevantes, processar informações, formatar e apresentar relatórios de maneira clara e compreensível.
Filtragem e Validação de Dados de Entrada	Prevenir ataques de segurança, como inserção de códigos maliciosos, e garantir a integridade dos dados processados pelo sistema, filtrando e validando informações de entrada.	Segurança, Integridade, Confiabilidade.	Validar formatos de dados, sanitizar entradas para evitar SQL Injection, XSS e outros ataques, rejeitar dados suspeitos ou malformados.
Notificação e Alerta	Notificar os usuários sobre eventos importantes e alertar sobre situações críticas, garantindo uma resposta rápida a problemas e necessidades urgentes de acordo com a necessidade de cada usuário.	Responsividade, Confiabilidade, Eficiência.	Enviar notificações por e-mail, mensagens push, configurar alertas automáticos para situações como estoque baixo, vencimento de documentos, manutenções pendentes, entre outros.
Backup e Recuperação	Garantir a integridade e disponibilidade dos dados do sistema, possibilitando a recuperação de informações	Segurança, Confiabilidade, Resiliência.	Realizar backups regulares dos dados do sistema, armazenar cópias de segurança em

Mecanismo Arquitetural	Finalidade	Atributos	Funções
	em caso de falhas ou desastres.		loais seguros, implementar procedimentos de recuperação de desastres.
Controle de Acesso	Gerenciar os direitos de acesso dos usuários ao sistema, garantindo que cada usuário tenha permissões adequadas de acordo com sua função e responsabilidades.	Segurança, Confidencialidade, Conformidade.	Definir políticas de acesso, controlar e restringir o acesso a recursos e funcionalidades com base nas permissões atribuídas a cada usuário, monitorar atividades de acesso para garantir conformidade com as políticas de segurança.

Os mecanismos arquiteturais desempenham um papel crucial na definição e implementação de uma arquitetura de software robusta e eficiente. Eles fornecem as bases necessárias para lidar com desafios complexos e garantir o funcionamento adequado do sistema. Ao entender e aplicar corretamente esses mecanismos, é possível criar um sistema confiável, seguro, capaz de atender às necessidades do negócio e dos usuários finais.

## Decisões e Justificativas

Nesta seção, apresentamos as decisões arquiteturas fundamentais tomadas durante o processo de concepção do sistema, bem como as justificativas que embasam essas escolhas. Ao longo do desenvolvimento do projeto, foram consideradas diversas alternativas e abordagens arquitetônicas, levando em conta os requisitos, metas e restrições.

Decisão	Justificativa
Escolhas da Arquitetura Cliente-Servidor em Camadas	A escolha da arquitetura Cliente-Servidor em Camadas, permite uma clara separação de responsabilidades entre o cliente e o servidor, facilitando o desenvolvimento, manutenção, segurança e escalabilidade do sistema. Além disso, facilita a implementação de segurança em diferentes níveis, ademais, proporciona maior flexibilidade para adaptação a diferentes plataformas e ambientes

## Carga Explosiva

Decisão	Justificativa
	de execução, como desktops e dispositivos móveis.
Implementação de Camadas de Segurança em Cliente e Servidor	Essa decisão é crucial para atender aos requisitos de segurança, autenticação e autorização. A implementação de medidas de segurança em ambas as camadas ajuda a proteger os dados contra acessos não autorizados e garantir a integridade e confidencialidade das informações, contribuindo para a confiabilidade do sistema.
Desempenho e Disponibilidade	Em um ambiente operacional de gestão de frota, onde operações em tempo real podem ser críticas, o desempenho e a disponibilidade são fundamentais. A arquitetura Cliente-Servidor em Camadas facilita a otimização do desempenho e garante a disponibilidade contínua do sistema, minimizando o tempo de inatividade não planejado.
Usabilidade e Manutenibilidade	Uma interface de usuário amigável é essencial para garantir uma experiência positiva para os usuários do sistema. A separação em camadas facilita o desenvolvimento de uma interface de usuário intuitiva e responsiva, contribuindo para a usabilidade e manutenibilidade do sistema ao longo do tempo.
Adoção de Práticas de Reuso de Componentes	Essa decisão apoia a meta de manutenção e a necessidade de flexibilidade. Ao reutilizar componentes, é possível minimizar a duplicidade de código, facilitando a manutenção e evolução do sistema ao longo do tempo, sem comprometer a estabilidade ou funcionalidade do sistema.

## Representação Arquitetural

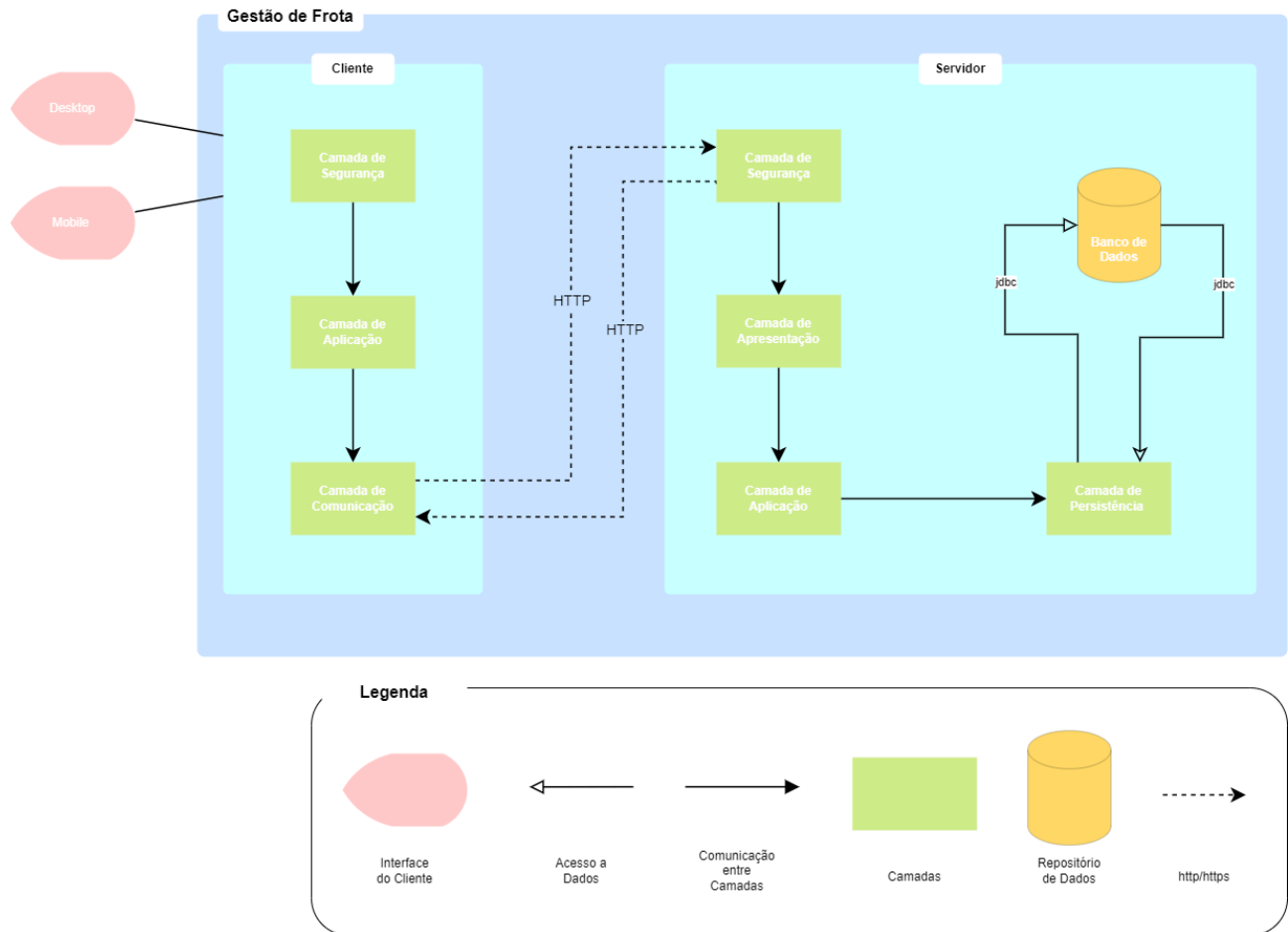


Imagem 1 - Visão Geral da Arquitetura do Software

## Diagramas da Arquiteturas

### Casos de Uso

Nesta seção, serão apresentados os casos de uso do sistema, fornecendo uma visão das interações entre os usuários e o sistema.

# Carga Explosiva

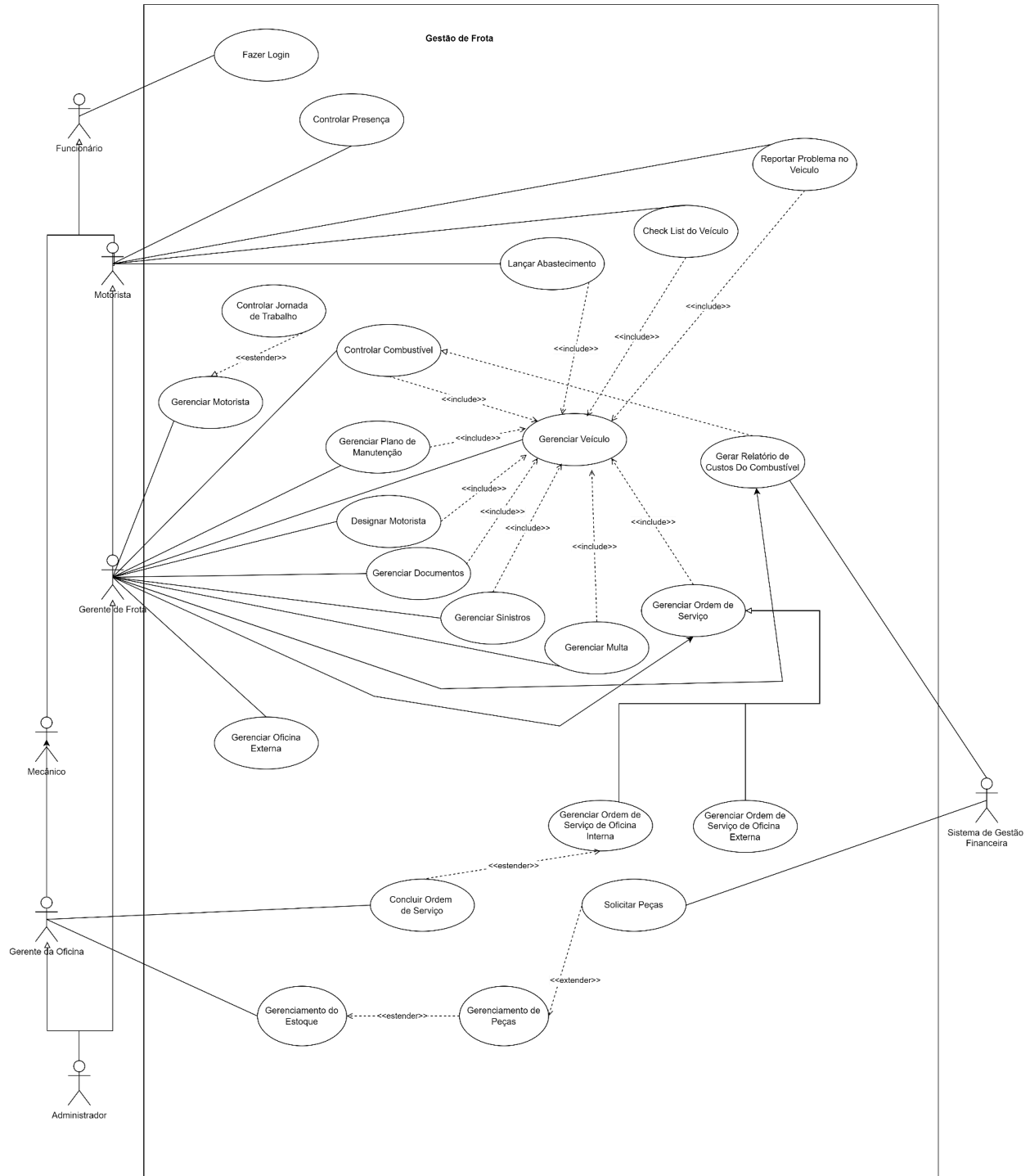
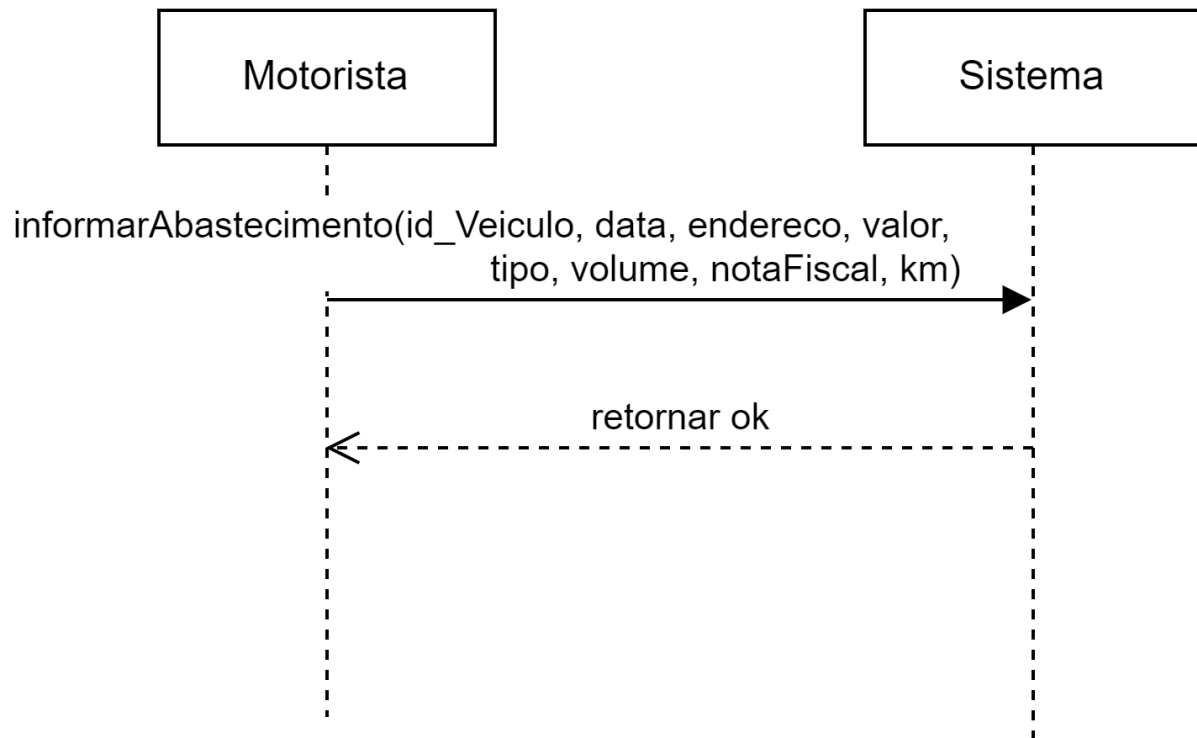


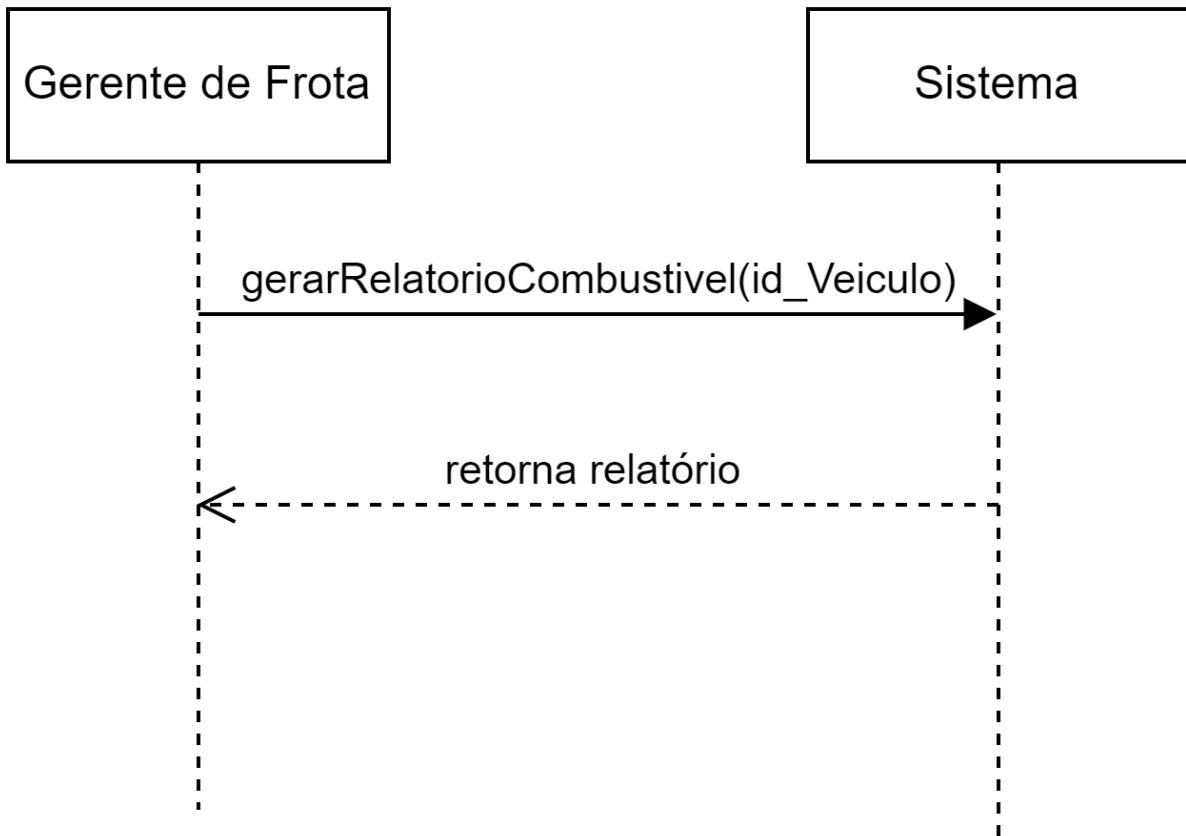
Imagem II - Diagrama de Casos de Uso



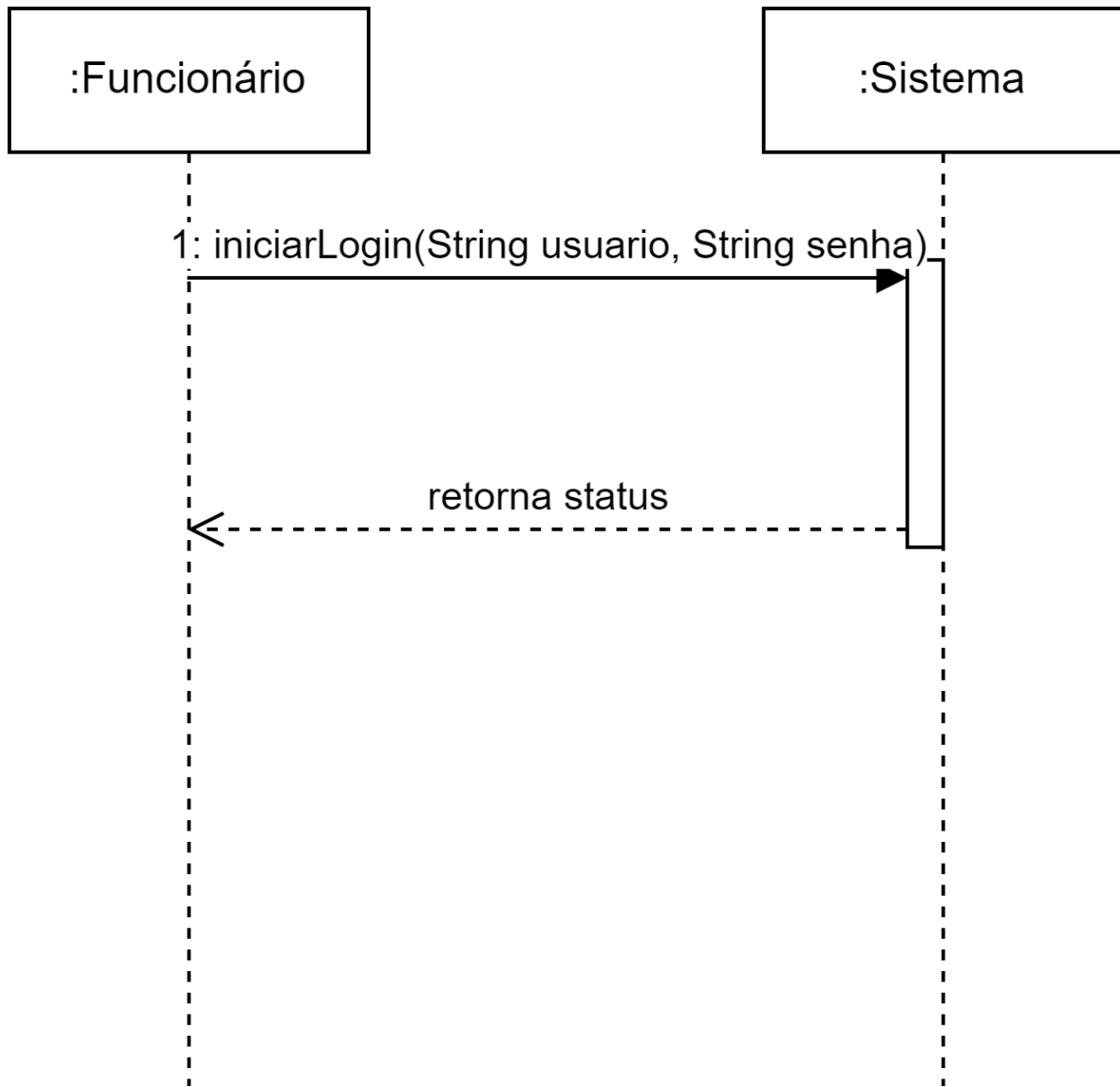
## Diagrama de Sequência do Sistema



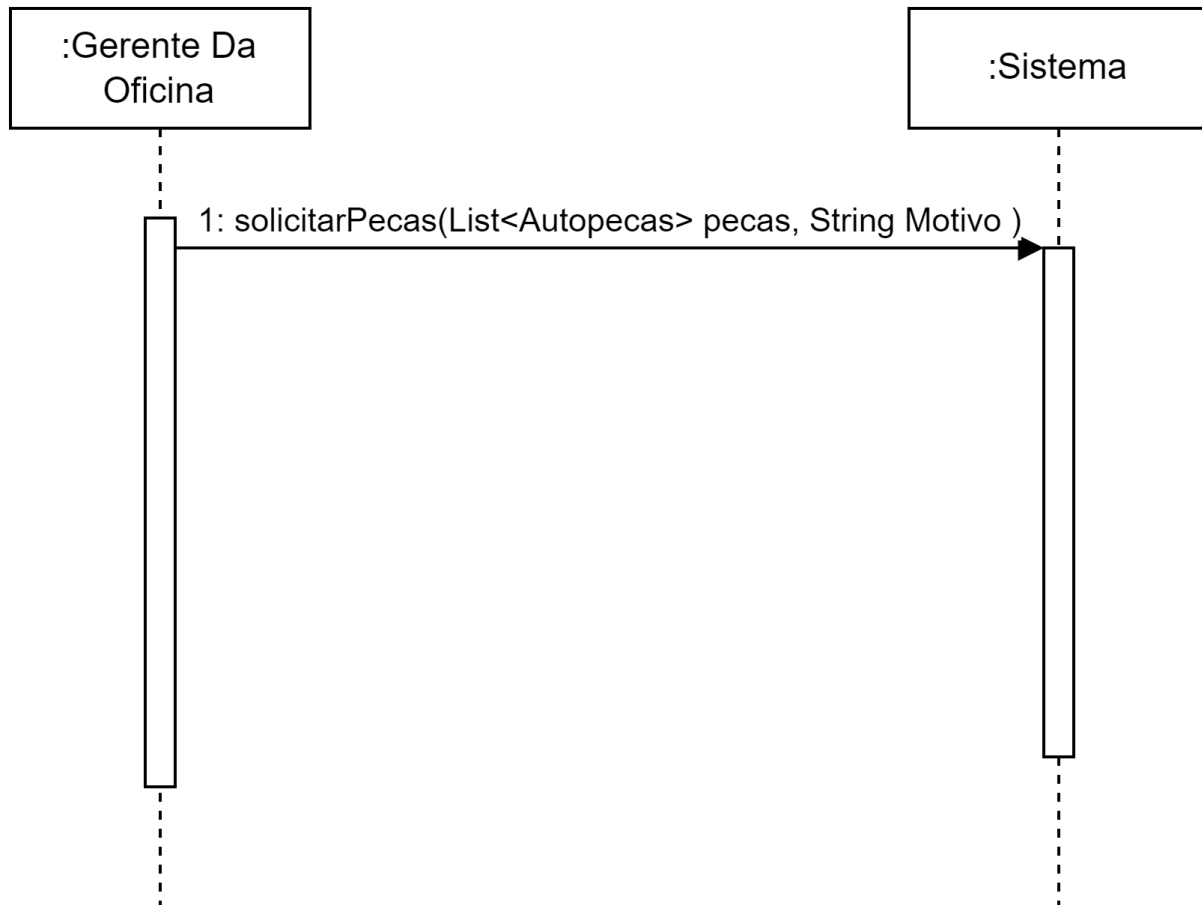
**Imagem III** - Diagrama de Sequência do Sistema - Lançar Abastecimento



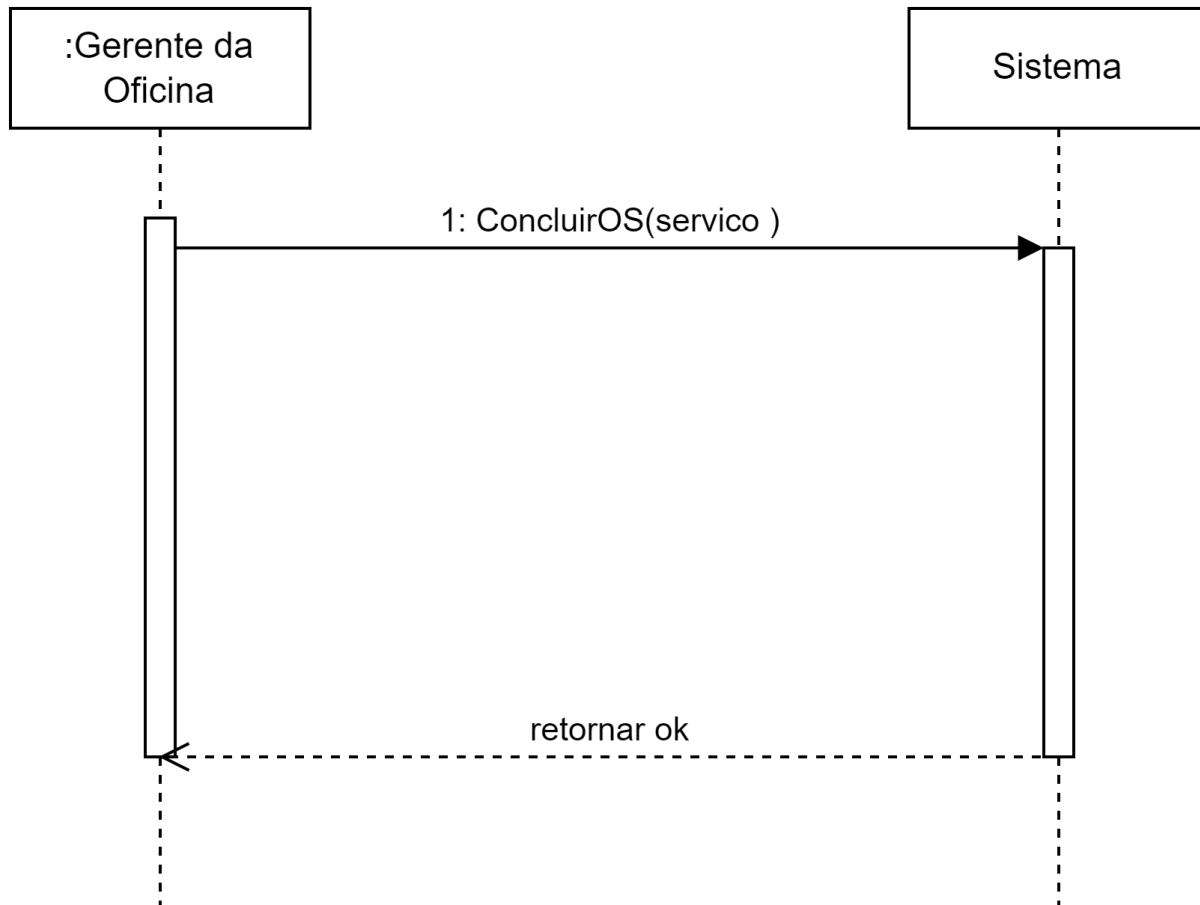
**Imagem IV** - Diagrama de Sequência do Sistema - Gerar Relatório de Custo de Combustível



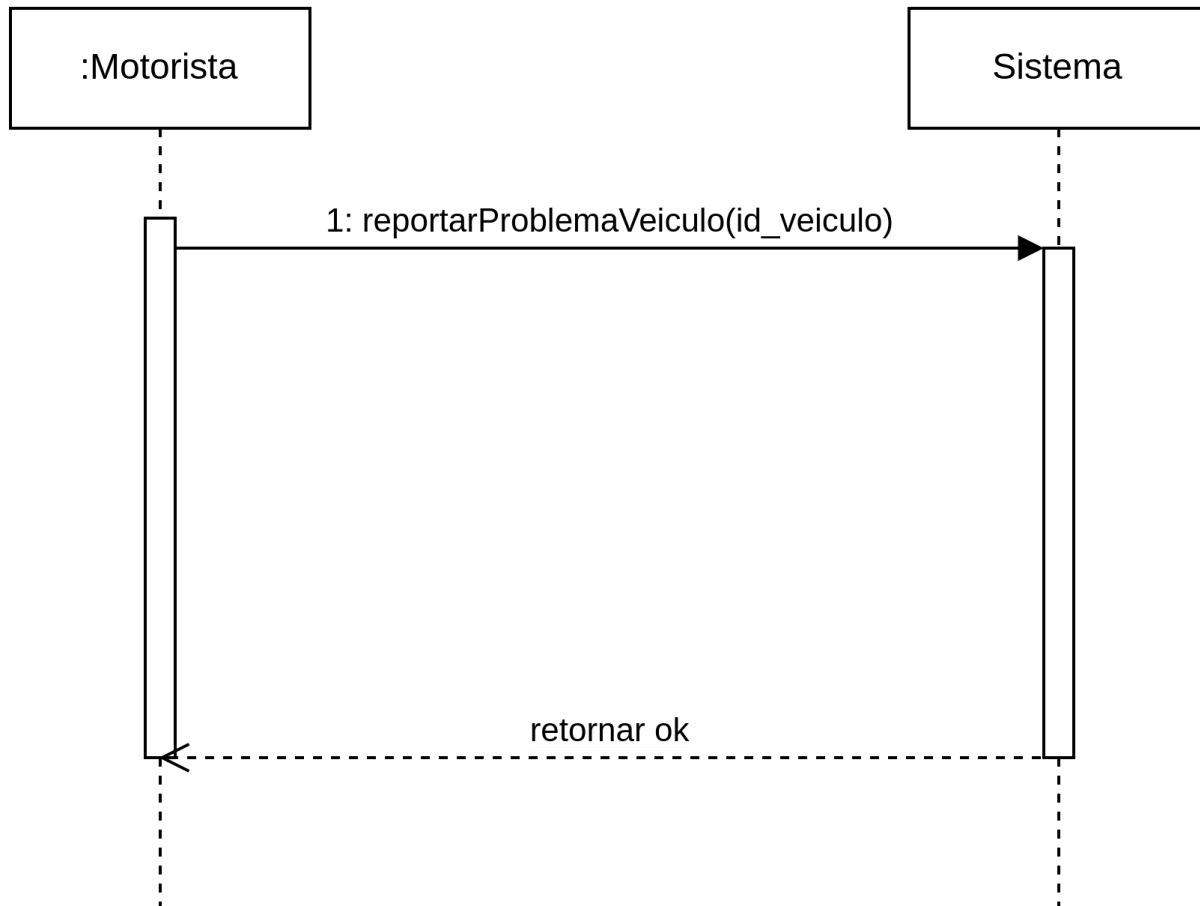
**Imagem V** - Diagrama de Sequência do Sistema - Fazer Login



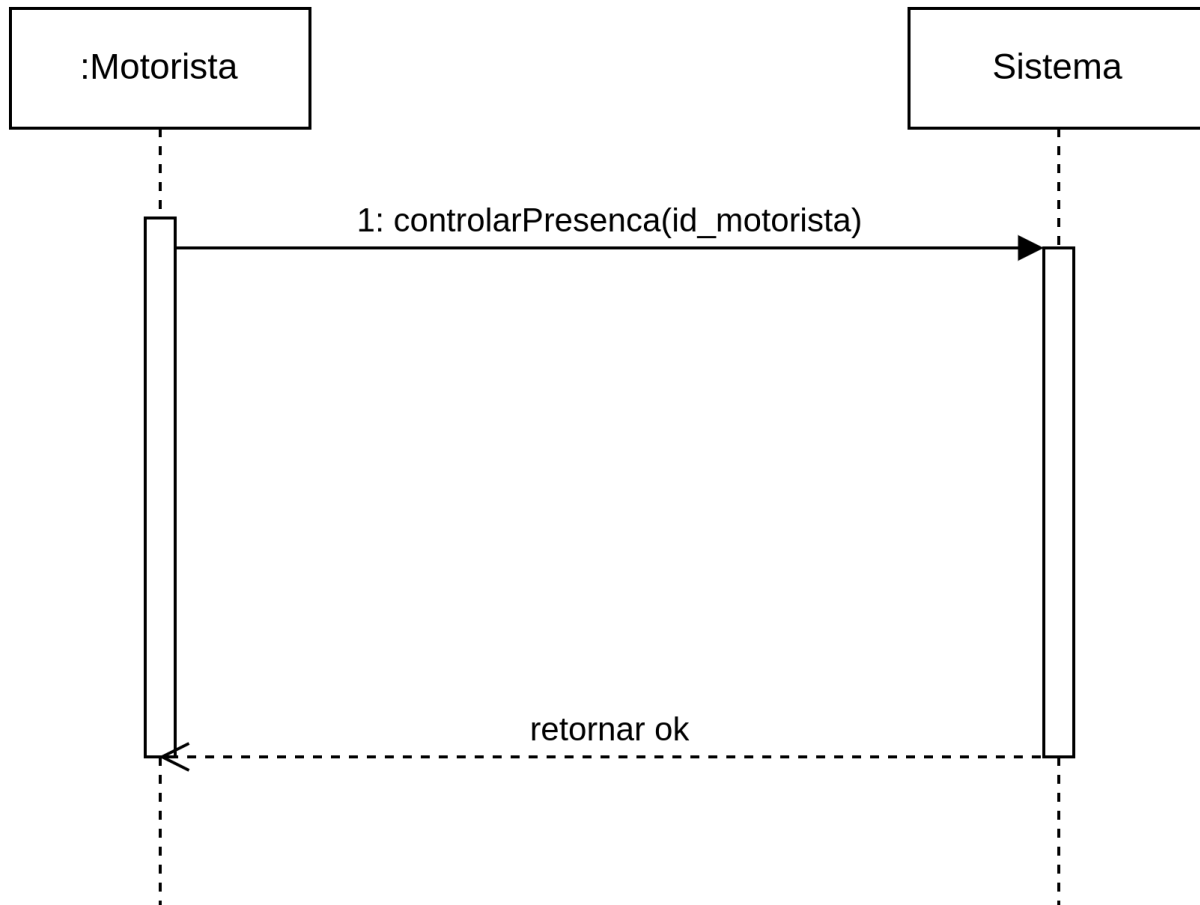
**Imagem VI** - Diagrama de Sequência do Sistema - Solicitar Peças



**Imagem VII** - Diagrama de Sequência do Sistema - Concluir Os



**Imagem VIII** - Diagrama de Sequência do Sistema - Relatar Problema Veículo



*Imagem IX - Diagrama de Sequência do Sistema - Controlar Presença*

## Contratos das Operações

A seção de contrato de operação define os termos e condições para o uso do sistema <Nome>.

Solicitar Peças	
Nome da Operação	solicitarPecas
Parâmetros de Entrada	- List<ItemDoEstoque> pecas - String motivo
Referências Cruzadas	Casos de Uso: “Solicitar Peças”
Pré-Condições	- Estar logado como Gerente - Existe pelo menos uma ItemDoEstoque

## Carga Explosiva

### Solicitar Peças

Pós-Condições	- Uma instância da classe Pedido é criada
---------------	---

### Gerar Relatório de Custo de Combustível

Nome da Operação	gerarRelatorioCombustivel
Parâmetros de Entrada	- id_Veiculo
Referências Cruzadas	Casos de Uso: "Gerar Relatório de Custo de Combustível"
Pré-Condições	<ul style="list-style-type: none"><li>- Estar logado como Gerente</li><li>- Veículo está cadastrado</li><li>- Existir pelo menos um abastecimento cadastrado ao veículo informado</li></ul>
Pós-Condições	- Uma instância da classe Relatório do Veículo é criada

### Lançar Abastecimento

Nome da Operação	informarAbastecimento
Parâmetros de Entrada	<ul style="list-style-type: none"><li>- id_Veiculo</li><li>- Date data</li><li>- List&lt;String&gt; endereco</li><li>- Float valor</li><li>- char tipo</li><li>- Float volume</li><li>- String notaFiscal</li><li>- Float km</li></ul>
Referências Cruzadas	Casos de Uso: "Lançar Abastecimento"
Pré-Condições	<ul style="list-style-type: none"><li>- Esta logado como Motorista</li><li>- Motorista está designado para veículo</li></ul>
Pós-Condições	- Uma instância da classe Abastecimento é criada

### Reportar Problema no Veículo

Nome da Operação	reportarProblemaVeiculo
Parâmetros de Entrada	- id_veiculo



## Carga Explosiva

Reportar Problema no Veículo	
Referências Cruzadas	Casos de Uso: "Reportar Problema no Veículo"
Pré-Condições	<ul style="list-style-type: none"><li>- Estar logado como Motorista</li><li>- Veículo está cadastrado</li></ul>
Pós-Condições	<ul style="list-style-type: none"><li>- O veículo foi identificado</li><li>- Um objeto "ProblemaRelatado" foi criado e associado ao motorista</li></ul>

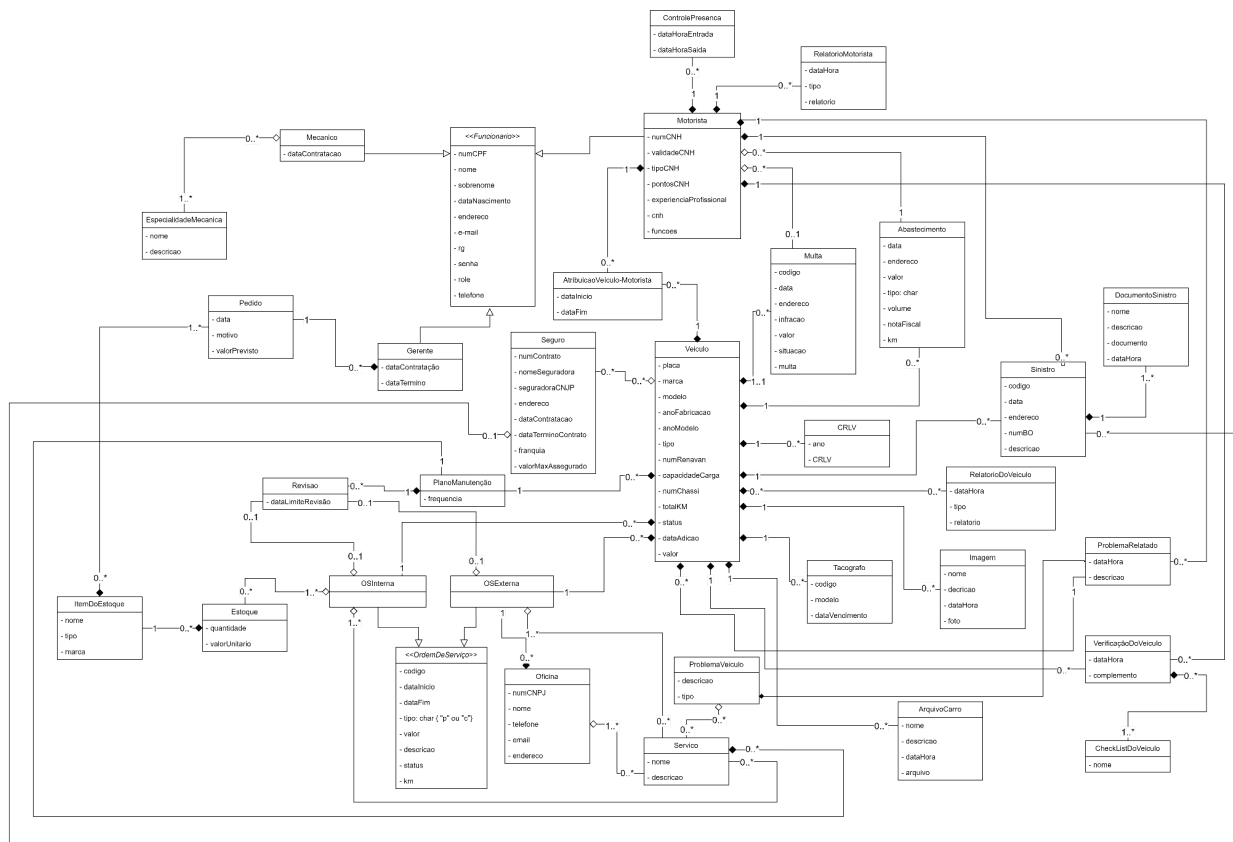
Controlar Presença	
Nome da Operação	controlarPresenca
Parâmetros de Entrada	<ul style="list-style-type: none"><li>- id_motorista</li></ul>
Referências Cruzadas	Casos de Uso: "Controlar Presença"
Pré-Condições	<ul style="list-style-type: none"><li>- Estar logado como Motorista</li></ul>
Pós-Condições	<ul style="list-style-type: none"><li>- Um objeto "ControlePresenca" foi criado e associado ao motorista</li></ul>

Fazer Login	
Nome da Operação	fazerLogin
Parâmetros de Entrada	<ul style="list-style-type: none"><li>- usuario</li><li>- senha</li></ul>
Referências Cruzadas	Casos de Uso: "Fazer Login"
Pré-Condições	<ul style="list-style-type: none"><li>- Estar online</li><li>- Estar na tela inicial</li><li>- Estar deslogado</li></ul>
Pós-Condições	<ul style="list-style-type: none"><li>- Usuário logado</li></ul>

Concluir OS	
Nome da Operação	ConcluirOS
Parâmetros de Entrada	<ul style="list-style-type: none"><li>- Os Interna</li></ul>
Referências Cruzadas	Casos de Uso: "Concluir OS"

Concluir OS	
Pré-Condições	<ul style="list-style-type: none"> <li>- Estar logado</li> <li>- OS existe</li> <li>- OS não está concluída</li> </ul>
Pós-Condições	<ul style="list-style-type: none"> <li>- Os concluída</li> </ul>

## Diagrama de Classe de Análise



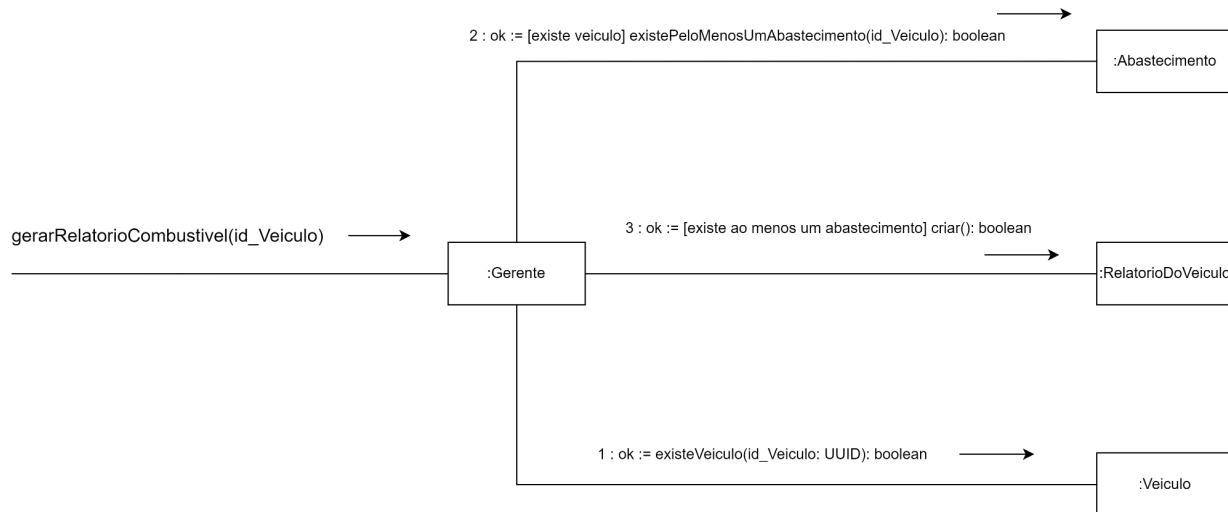
**Imagem X- Diagrama de Classe de Análise**

## Diagrama de Comportamento

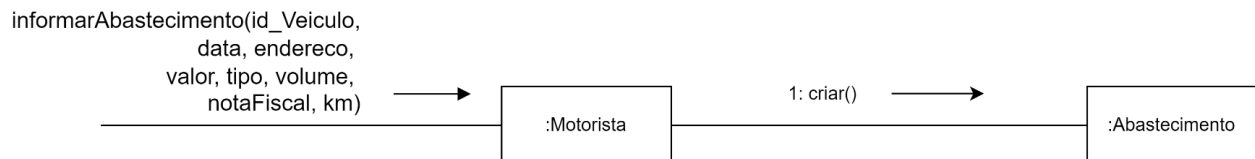
A seção de diagrama de comportamento é crucial para entender como os diversos componentes do sistema interagem entre si para realizar as funcionalidades desejadas. Aqui, os Diagramas de Interação oferecem uma representação visual das trocas de mensagens e interações entre objetos durante a execução de cenários específicos. Esses diagramas ajudam a

## Carga Explosiva

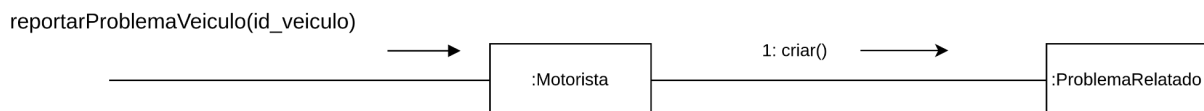
elucidar não apenas a sequência de eventos, mas também os papéis e responsabilidades de cada objeto envolvido, proporcionando uma visão abrangente e detalhada do comportamento dinâmico do sistema em diferentes situações.



### Anexo III - Diagrama de Interação - Gerar Relatório de Custo de Combustível

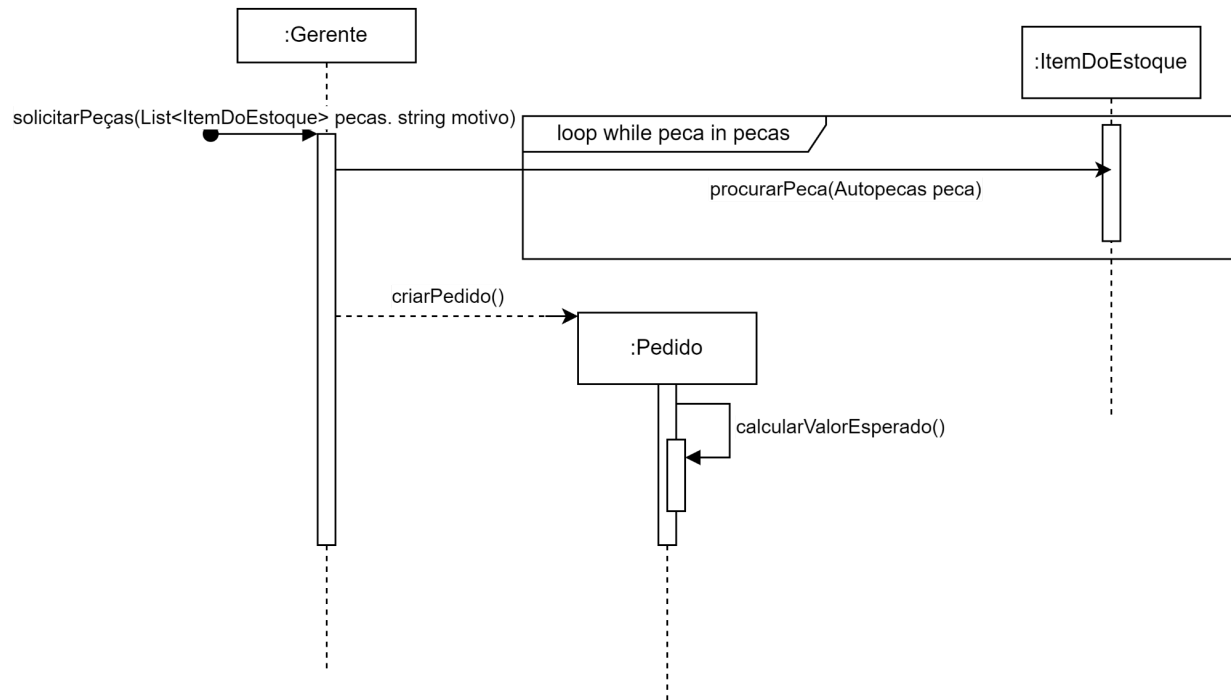


### Imagem XI - Diagrama de Interação - Lançar Abastecimento

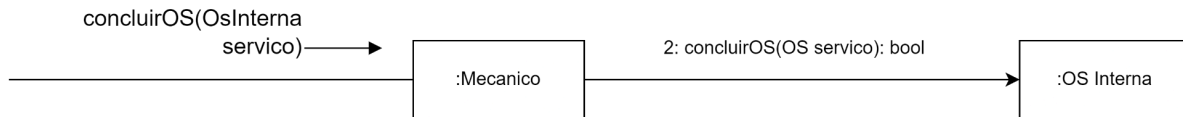


### Imagem XII - Diagrama de Interação - Reportar Problema Veículo

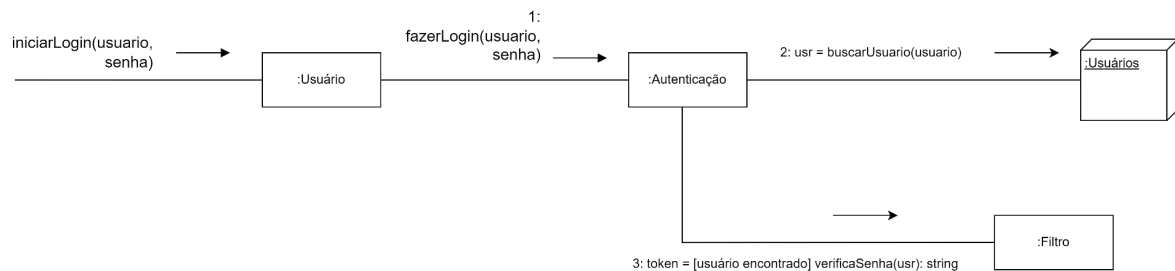
## Carga Explosiva



**Imagem XIII - Diagrama de Sequência - Solicitar Peças**

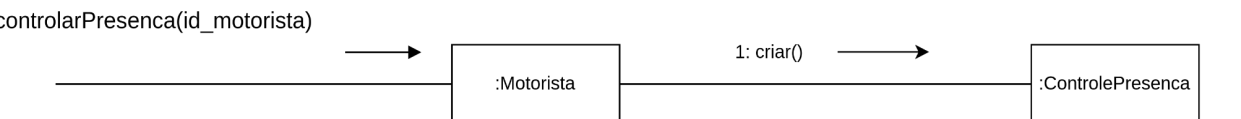


**Imagem XIV- Diagrama de Interação - Concluir OS**



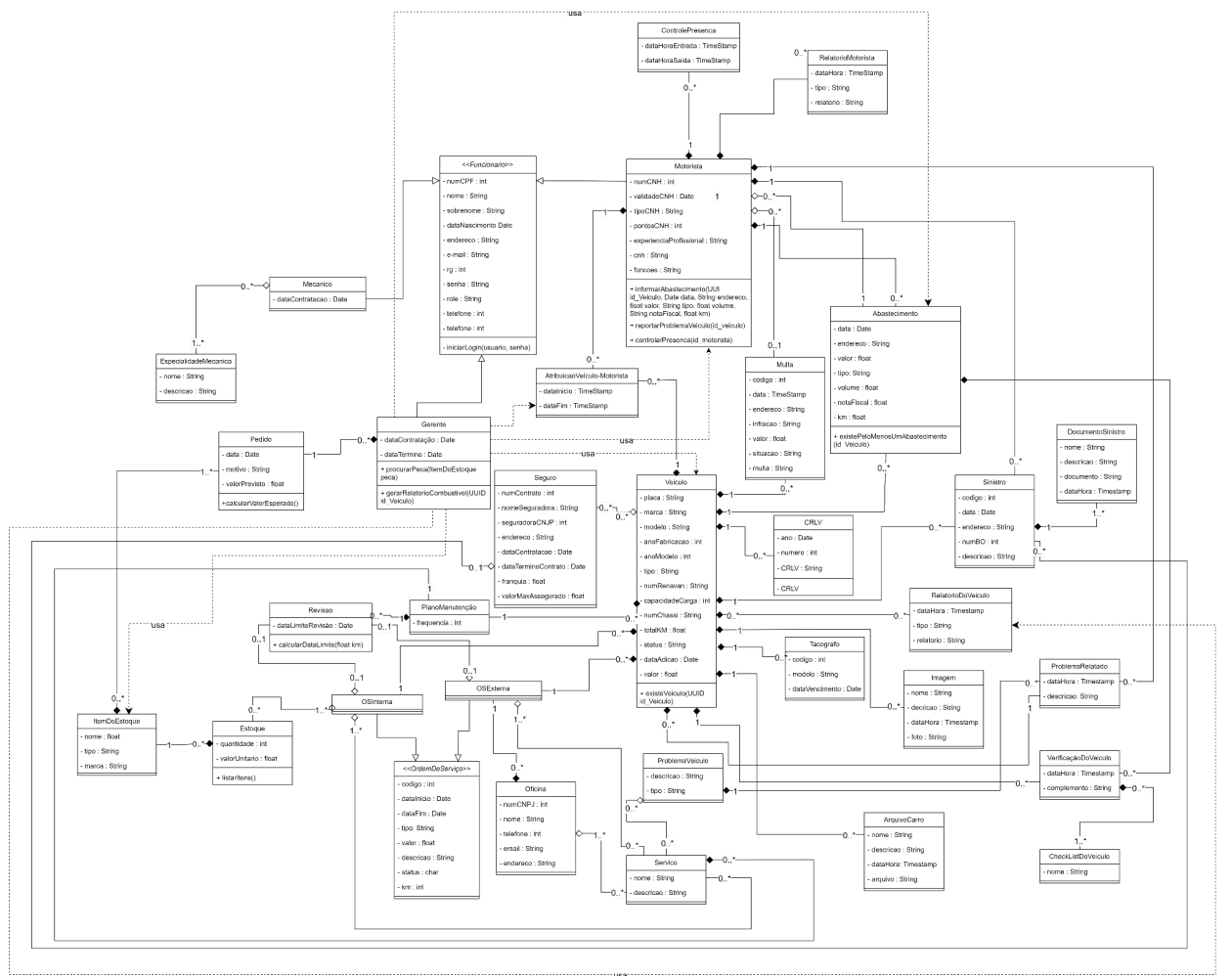
**Imagem XV - Diagrama de Interação - Fazer Login**

### Carga Explosiva



**Imagem XVI - Diagrama de Interação - Controlar Presença**

### Diagrama de Classes Detalhado



**Imagem XVII - Diagrama de Classe Detalhado**

# Dimensionamento e Performance

Esta seção aborda as considerações relacionadas ao dimensionamento e desempenho do sistema de Gestão de Frota, incluindo o volume esperado de usuários, acessos diários e mensais, bem como os tempos de resposta esperados para transações críticas.

## Volume

- Número estimado de usuarios: 100 a 500 usuarios
- Número estimado de acessos diários: 70% dos usuários
- Número estimado de acessos por mês: Aproximadamente 8000.
- Tempo de seção de um usuário: 8 horas

## Performance

Para garantir uma experiência de usuário satisfatória e a eficiência do sistema, são estabelecidos os seguintes objetivos de desempenho:

- **Tempo máximo de resposta para operações críticas:** As transações críticas do sistema, como cadastro de veículos, atribuição de motoristas e registro de manutenções, devem ter um tempo máximo de resposta de até 3 segundos para garantir a responsividade do sistema.
- **Tempo máximo de processamento de relatórios:** A geração de relatórios personalizáveis sobre o desempenho da frota e das operações de manutenção deve ser concluída dentro de um intervalo de tempo aceitável, com um tempo máximo de processamento de 10 segundos para relatórios típicos e 20 segundos para relatórios mais complexos.
- **Otimização de consultas e operações de banco de dados:** Devem ser implementados práticas de otimização de consultas e operações de banco de dados para garantir uma recuperação eficiente de dados e minimizar os tempos de resposta em todas as interações com o banco de dados.

## Qualidade

Nesta seção, mostra como a arquitetura de software contribui para diversos aspectos de qualidade do sistema, incluindo extensibilidade, confiabilidade, portabilidade, segurança e outros.

## Carga Explosiva

Item	Descrição
Extensibilidade	A arquitetura do sistema foi projetada com extensibilidade em mente, permitindo que novas funcionalidades sejam facilmente adicionadas e integradas ao sistema existente. Isso é alcançado por meio da modularidade da arquitetura em camadas que facilita a identificação e isolamento de componentes específicos para modificação ou substituição sem afetar outras partes do sistema. Além disso, a adoção de padrões arquitetônicos, como a separação clara entre a lógica de negócios e a apresentação, facilita a extensão do sistema sem comprometer sua estabilidade ou integridade.
Confiabilidade	A confiabilidade do sistema é garantida por meio de várias medidas arquiteturais. A separação em camadas facilita o isolamento de falhas e a implementação de estratégias de recuperação de falhas em níveis específicos da arquitetura. Além disso, a redundância de componentes críticos e a implementação de mecanismos de monitoramento e registro de erros contribuem para a detecção precoce e a mitigação de falhas, garantindo a disponibilidade contínua do sistema.
Portabilidade	A arquitetura do sistema foi projetada para garantir sua portabilidade em diferentes plataformas e ambientes de execução, incluindo desktops e dispositivos móveis. Isso é alcançado por meio da adoção de tecnologias e padrões de desenvolvimento que são compatíveis com múltiplas plataformas, permitindo que o sistema seja executado de forma consistente e eficiente em diferentes ambientes. Além disso, a separação clara entre a lógica de negócios e a apresentação facilita a adaptação do sistema a diferentes requisitos e restrições de plataforma.
Segurança	A segurança do sistema é uma prioridade e é abordada em várias camadas da arquitetura. Medidas de segurança, como autenticação e autorização de usuários, são implementadas tanto no cliente quanto no servidor para proteger os dados contra acessos não autorizados. Além disso, a criptografia de dados em trânsito e em repouso, juntamente com implementação de práticas de segurança recomendadas, garantem a integridade e confidencialidade das informações do sistema.
Usabilidade	A usabilidade do sistema é aprimorada pela arquitetura centrada no usuário, que prioriza a criação de interfaces intuitivas e responsivas. A separação entre a lógica de apresentação e a lógica de negócios permite o desenvolvimento de interfaces de usuário flexíveis e personalizáveis, adaptadas às necessidades e preferências dos usuários finais. Além disso, a implementação de práticas de design centradas no usuário garante uma experiência de uso positiva, promovendo a eficiência e a satisfação do

## Carga Explosiva

Item	Descrição
	usuário.
Manutenibilidade	A manutenibilidade e sistema é garantida pela modularidade da arquitetura, que facilita a identificação e correção de defeitos, bem como a implantação e a adoção de padrões de codificação consistentes facilitam a compreensão e a manutenção do código-fonte, garantindo a estabilidade e a evolução contínua do sistema ao longo do tempo.

A qualidade do sistema é fundamental para garantir sua eficácia, confiabilidade e usabilidade. Através da definição de requisitos de qualidade claros e mensuráveis, podemos assegurar que o sistema atenda às expectativas dos usuários e às necessidades do negócio. Ao adotar práticas de desenvolvimento de software robustas, realizar testes rigorosos e implementar processos de garantia de qualidade eficazes, podemos mitigar riscos, identificar e corrigir defeitos precocemente, e fornecer um produto final de alta qualidade. Investir na qualidade desde as fases iniciais do projeto é essencial para evitar retrabalho, garantir a satisfação do cliente e alcançar o sucesso a longo prazo do sistema.

## Documentação

Esta seção contém os links para os documentos online apresentados neste presente documento.

Documento	Link
Pasta do Trabalho	<a href="#">Carga Explosiva</a>
Documento de Arquitetura de Software	<a href="#">W Documento de Arquitetura de Software.d...</a>
Diagrama de Casos de Uso	<a href="#">Documento Draw.io</a>
Diagrama de Classe de Análise	<a href="#">Documento Draw.io</a>
Diagramas de Sequência do Sistema	<a href="#">Documento Draw.io</a>
Diagramas de Interação	<a href="#">Documento Draw.io</a>
Diagrama de Classe Detalhado	<a href="#">Documento Draw.io</a>
Diagrama de Visão Geral da Arquitetura	<a href="#">Documento Draw.io</a>