<S.i Vira>

<Lazy fries>
Documento de Arquitetura de Software

Versão <1.0>

[Observação: O template a seguir é fornecido para uso com o Rational Unified Process (RUP). O texto em azul exibido entre colchetes e em itálico (style=InfoBlue) foi incluído para orientar o autor e deve ser excluído antes da publicação do documento. Um parágrafo digitado após esse estilo será automaticamente definido como normal (style=Body Text).]

[Para personalizar campos automáticos no Microsoft Word (que exibem um fundo cinza quando selecionados), escolha File>Properties e substitua os campos Title, Subject e Company pelas informações apropriadas para este documento. Depois de fechar a caixa de diálogo, para atualizar os campos automáticos no documento inteiro, selecione Edit>Select All (ou Ctrl-A) e pressione F9 ou simplesmente clique no campo e pressione F9. Isso deve ser feito separadamente para Cabeçalhos e Rodapés. Alt-F9 alterna entre a exibição de nomes de campos e do conteúdo dos campos. Consulte a ajuda do Word para obter mais informações sobre como trabalhar com campos.]

Histórico da Revisão

Data	Versão	Descrição	Autor
<22/09/2022>	<1.0>	<Criação do minimundo>	<Gabriel>

Índice Analítico

Documento de Arquitetura de Software

1. Introdução

Nesse projeto produziremos um software de logística de transporte para uma fábrica de batata frita congelada. Nosso objetivo é monitorar e otimizar os processos de entrada de matéria prima e saída do produto final (inbound e outbound, respectivamente). Para isso faremos um sistema web em que a fábrica poderá solicitar os serviços das transportadoras parceiras, agendando a carga ou descarga.

Dito isso, nosso sistema terá 3 stakeholders que são membros ativos de todo o processo da logística de transporte: o administrador da fábrica, as transportadoras e o motorista (algumas vezes citado como caminhão). Basicamente o agendamento consiste em permitir que, com um acesso a um módulo de software, as transportadoras visualizem a frota de caminhões disponíveis naquela data e então irão atribuir o serviço a um deles. O motorista recebe na sua agenda do app a data, horário e local, e ao chegar na fábrica esse caminhoneiro fará um check-in, habilitando a geolocalização para que a fábrica monitore o pátio (já que é uma área muito grande). A ideia é que a fábrica possa ter controle dos processos para realizá-los de maneira mais eficiente, uma vez que os caminhões precisam passar por pesagem, checagem de risco, carga/descarga e faturamento em diferentes lugares. Com esse controle é possível ter uma ideia do tempo médio dos caminhões no pátio, reagir a imprevistos de forma ágil, evitar engarrafamento e agilizar o transporte.

Em caso de entrega de insumos (inbound), o caminhão sabe qual vaga (recebe por aplicativo através de notificação) ele vai descarregar e se pesar e entra na fábrica e assim que ele chega ocorre a pesagem com o insumo. Depois ele descarrega o produto. Pesagem sem insumo. Sai da fábrica.

Em caso de transporte do produto final (outbound), o caminhão sabe qual vaga (recebe por aplicativo através de notificação) ele vai fazer a checagem de risco, caso não passe na checagem o agendamento é cancelado e o caminhão volta para a transportadora, caso ele passe na checagem acontece o carregamento, depois disso ele vai para a viagem final.

Contexto:
Área do pátio: 9 km2
Número de docas: 10

-outbound-
Tempo estipulado entrada -> checagem de risco: 5min
Tempo estipulado checagem de risco: 30 min
Tempo estipulado checagem de risco -> carga/descarga: 2min
Tempo estipulado carga/descarga: 20 min
Tempo estipulado carga/descarga -> faturamento: 2min
Tempo estipulado faturamento: 1min
Tempo estipulado faturamento->saída: 4min
Total: 1hr e 04 min

-inbound-
Tempo estipulado entrada -> pesagem: 5min
Tempo estipulado pesagem: 5min
Tempo estipulado pesagem -> carga/descarga: 2min

Tempo estipulado carga/descarga: 20 min
Tempo estipulado carga/descarga -> pesagem: 2min
Tempo estipulado pesagem: 5min
Tempo estipulado pesagem->saída: 2min
Total: 41 min

Nossa ideia é que esse software seja adaptável a diferentes tipos de fábricas, como um produto flexível e escalonável.

1.1 Finalidade

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

2. Metas e Restrições da Arquitetura

Este sistema será dirigido por funcionalidades simples e fáceis de entendimento e implementação, visando estabelecer uma conexão entre transportadora e fábrica. Nossa principal meta é garantir que o desempenho seja sempre priorizado, em questões como resposta em tempo real ao administrador da fábrica e atualização em tempo real dos estados do motorista. Além disso, nós consideramos importante que a qualidade e desempenho do uso de dispositivos móveis não seja defasada pela instabilidade de conexão que vier a ocorrer. Portanto, uma vez agendado, o motorista terá suas operações salvas num estado seguro para evitar perda de informação, por acesso precário à internet, enquanto percorre pelo pátio.

3. Suposições e Dependências

O projeto foi feito dentro do segundo semestre de 2022 e é composto por uma equipe de 7 estudantes universitários de Sistemas de Informação. O grupo é munido de informações sobre a logística de uma transportadora de caminhões e conhecimento incipiente em desenvolvimento web, porém a dificuldade sobre as especificidades de montar um software na parte teórica implicaram bastante na confecção desse projeto. As tecnologias usadas são novas para alguns dos integrantes, que estão aprendendo enquanto produzem e a incompatibilidade de tempo também pesa muito, mas foi ótimo para a melhorar a experiência da equipe.

4. Requisitos Arquiteturalmente Significantes

Requisitos funcionais:

Requisitos Gerais:

- **RF01:** O sistema deve ser capaz de buscar uma vaga disponível e reservá-la.
- **RF02:** O sistema deve ter uma tela de login, que automaticamente loga na função certa baseado

na combinação CPF/CNPJ e senha.

Requisitos relacionados ao administrador da fábrica:

- **RF03:** O sistema deve permitir que o administrador da fábrica liste, cadastre, exclua e altere transportadoras.
- **RF04:** O sistema deve permitir que o administrador da fábrica crie e apague vagas.
- **RF05:** O sistema deve permitir que o administrador da fábrica solicite o agendamento de carga/descarga a uma transportadora.
- **RF06:** O sistema deve permitir que o administrador da fábrica defina as características para realizar o transporte, por exemplo capacidade do caminhão.
- **RF07:** O sistema deve notificar ao administrador da fábrica as informações do motorista selecionado no agendamento, como por exemplo característica s do seu caminhão.
- **RF08:** O sistema deve permitir que a fábrica visualize a localização dos caminhões dentro do pátio.
- **RF09:** O sistema deve fornecer ao administrador da fábrica uma lista de todos agendamentos em andamento.
- **RF10:** O sistema deve permitir que o administrador da fábrica veja o estado dos caminhões.
- **RF11:** O sistema deve permitir que o administrador da fábrica receba notificações sobre a mudança de estado dos caminhões.
- **RF12:** O sistema deve permitir que o administrador da fábrica veja o histórico de agendamentos.

Requisitos relacionados a Transportadora:

- **RF13:** O sistema deve permitir que a transportadora liste, cadastre, exclua e altere motoristas
- **RF14:** O sistema deve permitir que a transportadora selecione um caminhão para o serviço solicitado.
- **RF15:** O sistema deve fornecer à transportadora uma lista de agendamentos em andamento que ela participa.
- **RF16:** O sistema deve permitir que a transportadora veja o histórico de agendamentos que ela participa.
- **RF17:** O sistema deve permitir que o administrador da fábrica receba notificações sobre um novo agendamento

Requisitos relacionados ao Motorista:

- **RF18:** O sistema deve calcular a rota antes do término do agendamento e disponibilizá-la na interface do motorista.
- **RF19:** O sistema deve notificar o motorista na sua agenda a data, o horário e local.
- **RF20:** O sistema deve dar a opção de check-in e check-out para o motorista.
- **RF21:** O sistema deve ser capaz de deixar o motorista trocar o estado do caminhão ao final de cada processo (sendo eles “pendente”, “em pesagem”, “carga”, “descarga”, “checagem”, “faturamento”, “reagendado”, "concluído", “cancelado”).
- **RF22:** O sistema deve receber a minutagem sempre que o caminhão trocar de estado.

Requisitos relacionados ao Administrador:

- **RF23:** O sistema deve permitir que o administrador liste, cadastre, exclua, altere e crie o usuário administrador da fábrica.

Requisitos não funcionais

Confiabilidade

- **CF-01:** A cada 100 agendamentos, não podem ocorrer mais do que 1 erro na execução do software.

Desempenho

- **DS-01:** A página de visualização de rota deve dar suporte para 1000 usuários motoristas por hora provendo 5 segundos ou menos de resposta no aplicativo, incluindo carregar em tempo real o pátio da fábrica e qual vaga parar.

Integridade

- **IN-01:** O sistema não deve permitir que agentes externos acessem os dados registrados.

Manutenabilidade

- **MA-01:** A média de tempo para restaurar o sistema após uma falha não deve ser maior que 10 minutos.
- **MA-02:** Em caso de acidentes no pátio da fábrica, o sistema deve agendar novamente caminhões para novas vagas em menos de 5 minutos.

Portabilidade:

- **PR-01:** A interface de usuário do sistema deve estar disponível para dispositivos Android e iOS.

Usabilidade

- **US-01:** O sistema deve ser fácil o suficiente para um usuário agendar uma tarefa em 5 minutos.
- **US-02:** A taxa de erro dos usuários ao fazer o agendamento deve ser menor que 5%.

Compatibilidade

- **CP-01:** O sistema deve ser compatível com os sistemas de gerenciamento de banco de dados relacionais MySQL, PostgreSQL, Oracle e SQL Server.
- **CP-02:** O sistema deve poder utilizar sistemas de gerenciamento de banco de dados não-relacionais de forma não-conflitante com os relacionais.

Segurança

- **SE-01:** O sistema deve manter, num período de 24 horas, cópias dos seus dados armazenados em outro dispositivo.
- **SE-02:** Todas as comunicações externas entre o servidor de dados do sistema e clientes devem ser criptografadas.
- **SE-03:** todas as operações realizadas pelos usuários no sistema devem poder ser auditáveis por até um mês.

5. Decisões, Restrições e justificativas

A primeira decisão tomada foi a adoção do modelo MVC, já que ele é bastante simples e prático e faz muito sentido no contexto de desenvolvimento que lida tanto com front-end unido a conexão com banco de dados. Esse modelo norteou várias decisões arquiteturais como os padrões de projeto Controller e Creator.

O tipo de linguagem utilizada, por ser muito atual, fica disposta a muitas atualizações. Essa seria a principal restrição, pois a equipe de desenvolvimento deve ter em mente futuras e possíveis mudanças e upgrades de linguagem ou banco de dados.

6. Mecanismos Arquiteturais

Mecanismos de Análise

Persistência

A persistência dos dados é garantida pelo banco de dados MySQL.

Front-End

O front-end ficará na camada View e será responsável por interagir com o usuário do site e mostrar informações do banco de dados trazidas pela camada Controller. O front-end será feito em Vue e Typescript.

Deploy

O plano para deployment do projeto é hospedar em um host gratuito como o github pages ou heroku.

Padrão de projeto GRASP

Indireção

O padrão de indireção é usado para reduzir o acoplamento entre duas classes e garantir maior reuso por meio de uma classe intermediária que faz a ponte entre eles. Esse padrão já é usado no MVC, pois as classes controle fazem justamente essa ponte entre os modelos e as views.

Controlador

O padrão controlador é usado para separar a camada de interação com usuário da camada de sistema, e é normalmente utilizada para implementar os casos de uso. Como o nosso sistema já utiliza o padrão MVC o controlador é usado em todos os casos de uso, e um mesmo controlador pode resolver vários casos de uso, como a classe produto_controle que faz o CRUD e todas as operações que envolvem produtos.

Padrões de Projeto GOF

Fachada

Será utilizado para, dentre outras funções, realizar o acesso ao subsistema de persistência de dados. Classes de fachada para persistência permitirá a abstração das classes clientes a implementação em si do mecanismo de persistência. Por meio das classes de fachada, as classes clientes terão acesso somente a um pequeno conjunto abstrato de métodos para recuperação, salvamento, atualização ou remoção de dados. Isso permitiria ainda que diferentes implementações de persistência sejam utilizadas

no sistema sem afetar o comportamento das classes clientes, uma vez que essas operações estarão disponíveis através de uma interface.

Mediador

Permite que o sistema esteja em conformidade com o requisito SE-03. Aplicando o padrão Mediador, conseguimos restringir comunicações entre objetos do sistemas para serem realizadas através de um objeto mediador. Esse objeto poderia registrar em um log as operações realizadas pelos usuários, permitindo que elas sejam auditáveis. Da mesma forma, concentrando as comunicações entre as entidades do sistema através de mediadores possibilitaria que eles criptografassem todo o fluxo de mensagens.

7. Camadas da Arquitetura

Arquitetura em camadas:

Esse padrão arquitetural se trata de hierarquia e organização. Com esse padrão uma camada de hierarquia mais alta pode utilizar serviços de uma com hierarquia mais baixa.
Nesse projeto o padrão arquitetural em camadas será usado para instanciar classes e objetos nos arquivos.

Serão usadas 3 camadas no caso do nosso sistema, camada de interface dos usuários, camada da lógica de negócio (solicitação de serviço, acompanhamento do veículo e cadastros no sistema) e a camada do banco de dados.

Single Page Application:

No caso do portal web, nós utilizaremos React, um framework JavaScript para compor o front-end. Uma das características principais desse modelo é o roteamento de conteúdos da página utilizando o Router e Hooks, mecanismos do próprio framework. A arquitetura SPA possibilita uma rápida navegação do usuário, uma vez que suas todos os arquivos de front são carregados de uma vez, o que, por sua vez, melhora o desempenho citado no requisito não funcional de desempenho.

Model - View - Controller (MVC):

A escolha do MVC como um dos padrões de projeto parte da necessidade da interface do usuário ser portável entre diferentes sistemas operacionais. Com o padrão MVC, almejamos favorecer a distribuição de clientes magros, que por sua vez serão compostas unicamente por uma camada de interface gráfica, favorecendo maior manutenibilidade no lado do cliente. A separação da camada de visão possibilita ainda que ela seja distribuída entre os diferentes dispositivos e sistemas operacionais e se comunique com uma única implementação das camadas de controle e modelo desde que seja utilizado um protocolo de comunicação comum, como o HTTP, e favorece a mutabilidade do sistema, uma vez que permite que diferentes camadas tenham suas implementações alteradas ao longo do tempo sem impactos nas demais; de maneira resumida, com o emprego desse padrão almejamos menor acoplamento entre as camadas do sistema.

HTTPS:

O protocolo HTTPS garante uma conexão segura do cliente com o servidor, assegurando o RNF

de integridade.

8. Visões da Arquitetura

Diagrama de Classes

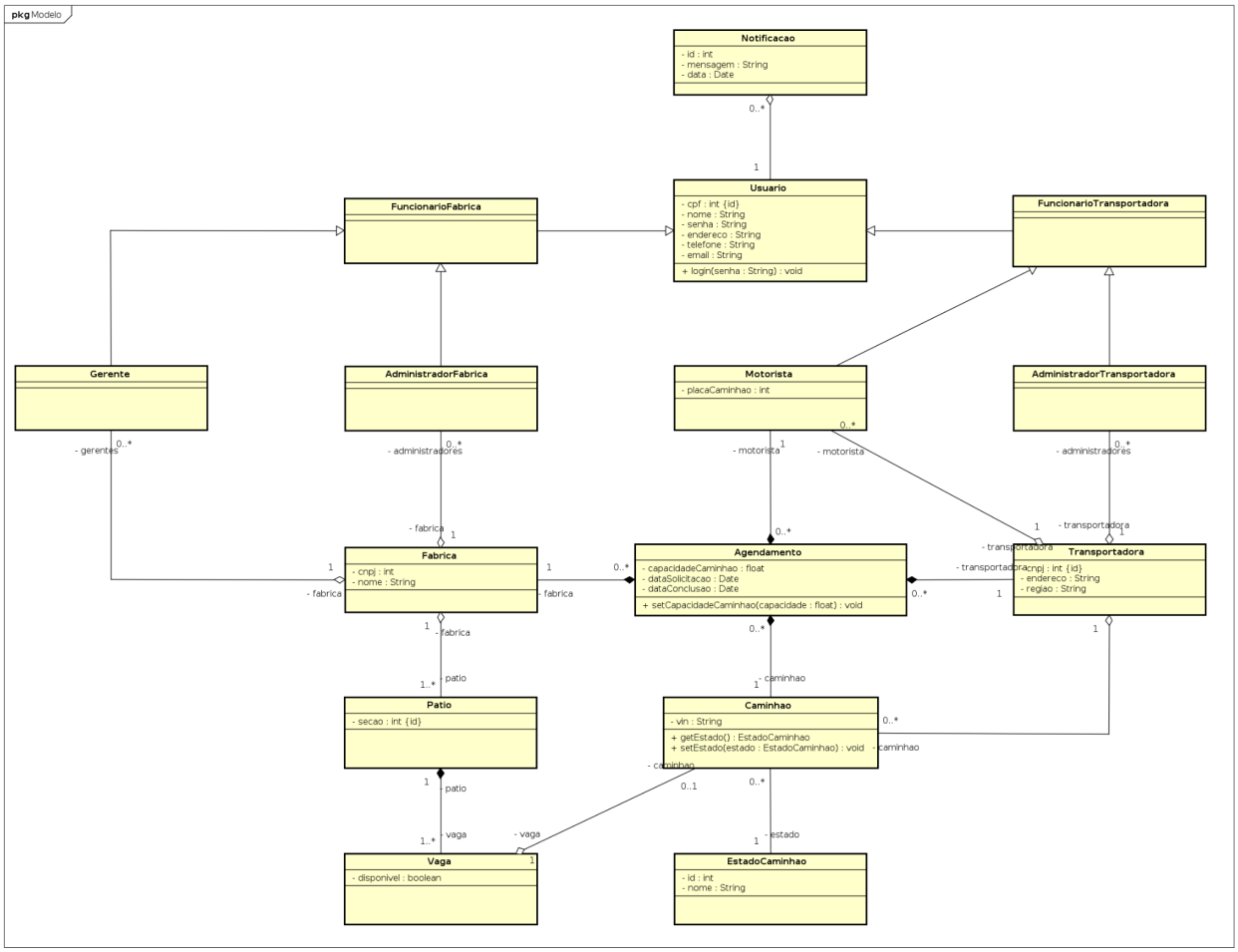


Diagrama de Implantação

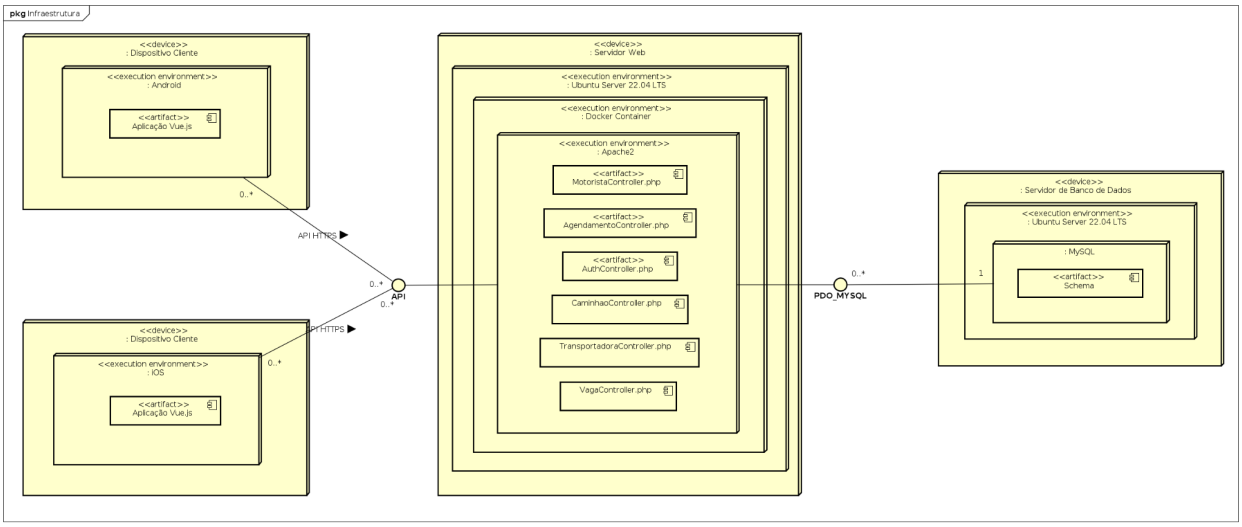
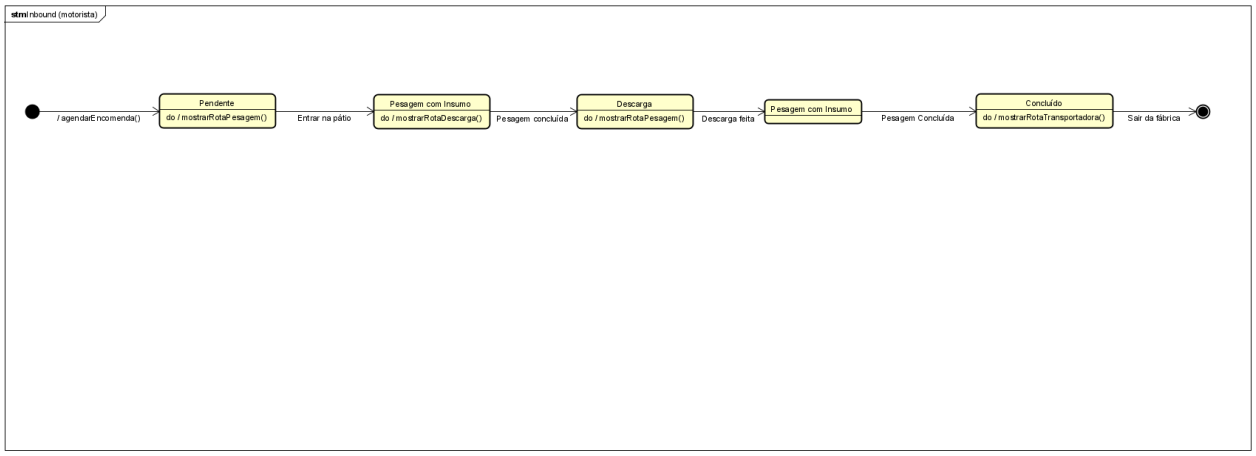
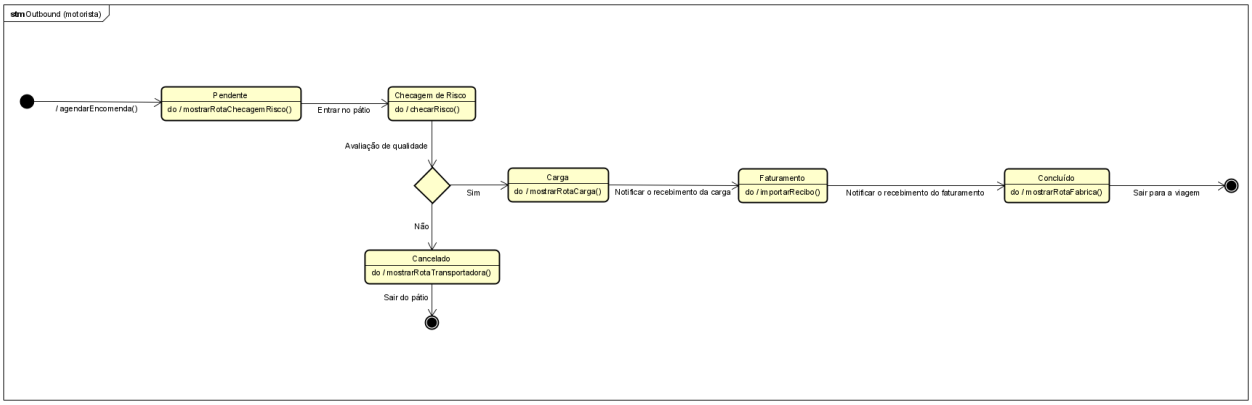


Diagrama de Estados do Motorista

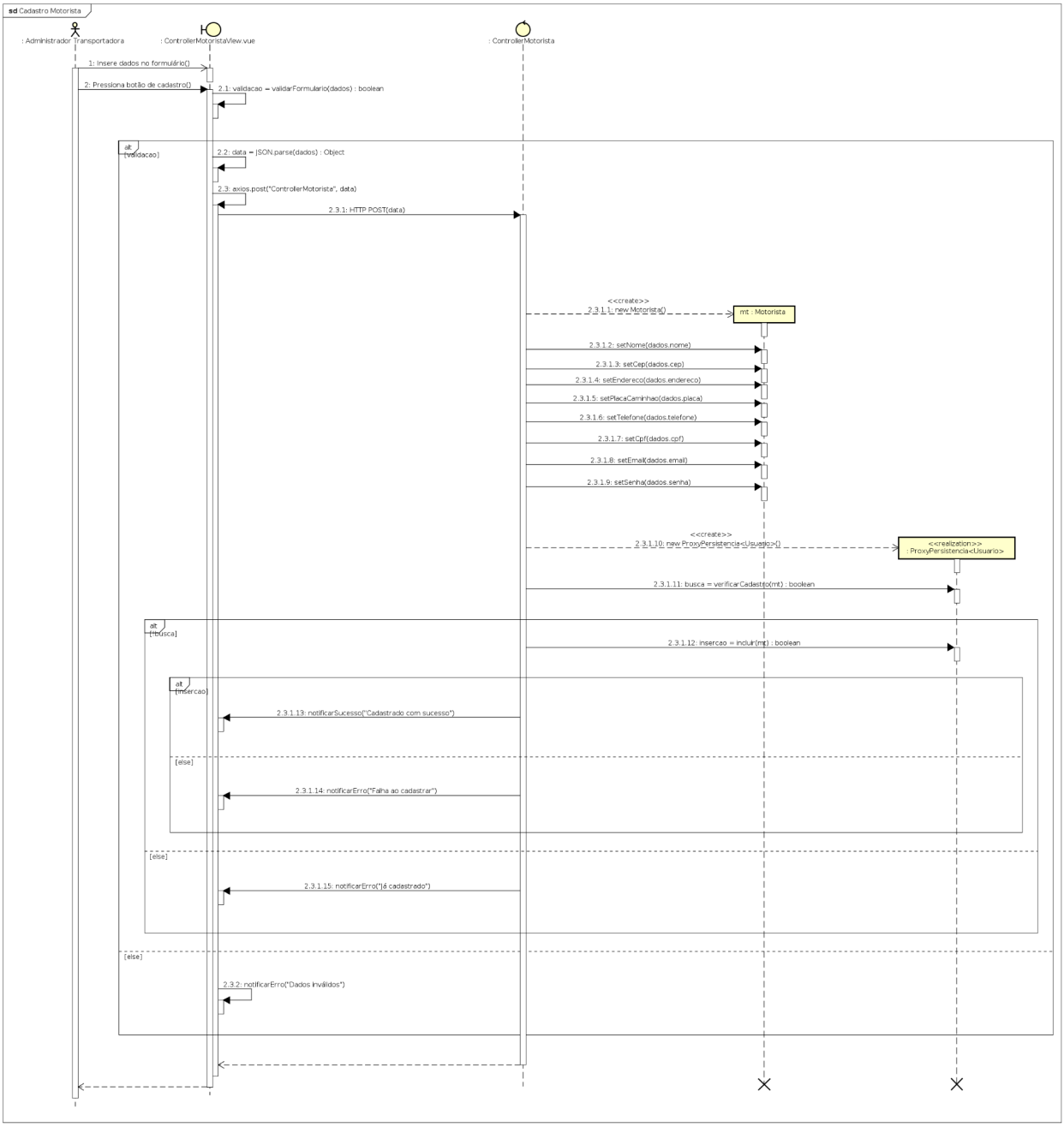
Inbound



Outbound



Diagramas de Sequência



Casos de Uso

Stakeholders

ATOR	PAPEL
Administrador	Responsável por gerenciar a aplicação, criando e modificando administrador da fábrica.
Administrador da Fábrica	Tem controle sobre agendamento e monitora o pátio. Pode cadastrar transportadoras e outro administrador.
Transportadora	Adicionar ou remover motoristas disponíveis. Pode cadastrar motoristas.
Motorista	Visualizar rota e mudar de caminhão de estado.

Descrição Detalhada dos Casos de Uso:

UC01: Estrutura Detalhada

Nome do caso de uso	UC01 - Inbound (chegada de matéria prima)
Ator(es)	Motorista
Descrição	Quando a transportadora atribui o serviço ao motorista ele é notificado no app com os dados: local de retirada, local da fábrica, dia e horário. Além disso, o status do caminhão sai de “pendente” para “aceito”. Chegando na fábrica o motorista faz check-in, mudando o status do caminhão de “aceito” para “em pesagem” e habilitando a geolocalização. O caminhão se dirige para a pesagem ainda cheio e quando conclui essa fase o motorista indica no app para que seu status mude para “em descarga” e ele seja direcionado para a vaga reservada. Quando o motorista indicar o fim dessa fase seu status volta para “segunda pesagem”, pois agora com o caminhão vazio a fábrica vai fazer a conta da quantidade de matéria que chegou. Por fim, o motorista indica o fim do processo e seu status muda para “concluído”.

Referências cruzadas	RF18, RF19, RF20, RF21.
Pré-condições	O motorista deverá estar logado no sistema (UC07) e ter sido atribuído a um serviço de inbound (UC10).
Pós-condições	Caso o processo seja bem sucedido, ele deve constar no histórico do administrador como “concluído”.

Cenário típico

Ator	Sistema
	1.O sistema notifica os dados do serviço ao motorista
2.O motorista se dirige ao local e faz check-in	
	3.O sistema troca o status do caminhão para “em pesagem”.
4.O motorista vai ao local, realiza a pesagem e indica a conclusão ao sistema.	
	5.O Sistema troca o status do caminhão para “em descarga” e indica a rota para a vaga ao motorista.
6.O motorista vai até a vaga reservada, realiza a descarga e indica a conclusão ao sistema.	
	7.O sistema troca o status do caminhão para “segunda pesagem”.
8.O motorista vai ao local, realiza a pesagem e indica a conclusão ao sistema.	
	9.O sistema troca o status do caminhão para “concluído”.

UC02: Estrutura Detalhada

Nome do caso de uso	**UC02 - Outbound (Saída do produto)**

Ator(es)	Motorista
Descrição	Quando a transportadora atribui o serviço ao motorista ele é notificado no app com os dados: local da fábrica, destino, dia e horário. Além disso, o status do caminhão sai de “pendente” para “aceito”. Chegando na fábrica o motorista faz check-in, mudando o status do caminhão de “aceito” para “em checagem” e habilitando a geolocalização. O caminhão se dirige para a checagem e quando conclui essa fase o motorista indica no app para que seu status mude para “em carga” e ele seja direcionado para a vaga reservada. Quando o motorista indicar o fim dessa fase seu status volta para “segunda pesagem”, pois agora com o caminhão vazio a fábrica vai fazer a conta da quantidade de matéria que chegou. Por fim, o motorista indica o fim do processo e seu status muda para “concluído”.
Referências cruzadas	RF18, RF19, RF20, RF21.
Pré-condições	O motorista deverá estar logado no sistema (UC07) e ter sido atribuído a um serviço de inbound (UC10).
Pós-condições	Caso o processo seja bem sucedido, ele deve constar no histórico do administrador como “concluído”.

Cenário típico

Ator	Sistema
	1.O sistema notifica os dados do serviço ao motorista
2.O motorista se dirige ao local e faz check-in	

	3.O sistema troca o status do caminhão para “em pesagem”.
4.O motorista vai ao local, realiza a pesagem e indica a conclusão ao sistema..	
	5.O Sistema troca o status do caminhão para “em descarga” e indica a rota para a vaga ao motorista.
6.O motorista vai até a vaga reservada, realiza a descarga e indica a conclusão ao sistema.	
	7.O sistema troca o status do caminhão para “segunda pesagem”.

Cenário(s) alternativo(s)

4.a. Alguma das informações não foi fornecida ou alguma das fornecidas está fora do formato esperado:

1. O sistema deverá indicar quais informações estão fora de formatação e qual o formato correto.
2. Retornar ao passo 2 do fluxo principal.

UC03: Estrutura Detalhada

Nome do caso de uso	**UC03 - Criar Vagas**
Ator(es)	Administrador da fábrica
Descrição	Um administrador da fábrica acessa o cadastro de vagas e então cadastra uma nova vaga informando os seguintes dados: ID, Tipo (pesagem, checagem, carga, descarga), Horário da última vez utilizada, ID do último caminhão estacionado, ID do próximo caminhão, Horário de chegada do próximo e Horário de partida do atual.

Referências cruzadas	RF01, RF04,
Pré-condições	O administrador da fábrica deverá estar logado no sistema e a sua conta deve possuir o cargo “administrador da fábrica”.
Pós-condições	Caso o cadastro seja bem sucedido, deverá ser criada e armazenada uma nova conta de funcionário com as informações fornecidas.

Cenário típico

Ator	Sistema
1. O administrador da fábrica acessa o cadastro de funcionários.	
	2. O sistema exibe um formulário com as informações necessárias para o cadastro (ID, Tipo (pesagem, checagem, carga, descarga), Horário da última vez utilizada, ID do último caminhão estacionado, ID do próximo caminhão, Horário de chegada do próximo e Horário de partida do atual.)
2. O administrador da fábrica preenche os campos do cadastro com os dados do funcionário	
3. O administrador da fábrica confirma o cadastro.	
	4. O sistema informa que o cadastro foi bem sucedido.

Cenário(s) alternativo(s)

4.a. Alguma das informações não foi fornecida ou alguma das fornecidas está fora do formato esperado:

1. O sistema deverá indicar quais informações estão fora de formatação e qual o formato correto.

2. Retornar ao passo 2 do fluxo principal.

UC04: Estrutura Detalhada

Nome do caso de uso	UC04 - Cadastro de um administrador da fábrica.
Ator(es)	administrador da fábrica.
Descrição	Um administrador da fábrica acessa o cadastro de administradores da fábrica e então cadastra um novo administrador da fábrica informando os seguintes dados: Nome, CEP, Endereço, CPF, Data de Nascimento, Telefone para Contato, E-mail, Senha.
Referências cruzadas	RF02, RF03, UC09
Pré-condições	O administrador da fábrica deverá estar logado no sistema.
Pós-condições	Caso o cadastro seja bem sucedido, deverá ser criada e armazenada uma nova conta de administrador da fábrica com as informações fornecidas.

Cenário típico

Ator	Sistema
1. O administrador da fábrica acessa o cadastro dos administradores da fábrica.	
	2. O sistema exibe um formulário com as informações necessárias para o cadastro (Nome, CEP, Endereço, CPF, Data de Nascimento, Telefone para Contato, E-mail, Senha.)
2. O administrador da fábrica preenche os campos do cadastro com os dados do novo administrador da fábrica	
3. O administrador da fábrica confirma o cadastro.	
	4. O sistema informa que o cadastro foi bem sucedido.

Cenário(s) alternativo(s)

4. Alguma das informações não foi fornecida ou alguma das fornecidas está fora do formato esperado:

- O sistema deverá indicar quais informações estão fora de formatação e qual o formato correto.
- retornar ao passo 2 do fluxo principal.

UC05: Estrutura Detalhada

Nome do caso de uso	UC05 - Cadastro de uma transportadora.
Ator(es)	administrador da transportadora, transportadora
Descrição	Um administrador da fábrica acessa o cadastro de transportadoras e então cadastra uma nova transportadora informando os seguintes dados: Nome, CEP, Endereço, CNPJ, Telefone para Contato, E-mail, Senha.
Referências cruzadas	RF02, RF03, UC08
Pré-condições	O administrador da fábrica deverá estar logado no sistema.
Pós-condições	Caso o cadastro seja bem sucedido, deverá ser criada e armazenada uma nova conta de transportadora com as informações fornecidas.

Cenário típico

Ator	Sistema
1. O administrador da fábrica acessa o cadastro das transportadoras.	
	2. O sistema exibe um formulário com as

	informações necessárias para o cadastro (Nome, CEP, Endereço, CNPJ, Telefone para Contato, E-mail, Senha.)
2. O administrador da fábrica preenche os campos do cadastro com os dados da nova transportadora	
3. O administrador da fábrica confirma o cadastro.	
	4. O sistema informa que o cadastro foi bem sucedido.

Cenário(s) alternativo(s)

4.a. Alguma das informações não foi fornecida ou alguma das fornecidas está fora do formato esperado:

- O sistema deverá indicar quais informações estão fora de formatação e qual o formato correto.
- Retornar ao passo 2 do fluxo principal.

UC06: Estrutura Detalhada

Nome do caso de uso	UC06 - Cadastro de um motorista.
Ator(es)	transportadora
Descrição	A transportadora acessa o cadastro de motoristas e então cadastra um novo motorista informando os seguintes dados: Nome, CEP, Endereço, Placa do Caminhão, Telefone para Contato, CPF, E-mail, Senha.
Referências cruzadas	RF02, RF13, UC08
Pré-condições	A transportadora deverá estar logada no sistema.

Pós-condições	Caso o cadastro seja bem sucedido, deverá ser criada e armazenada uma nova conta de motorista com as informações fornecidas.

Cenário típico

Ator	Sistema
1. A transportadora acessa o cadastro dos motoristas.	
	2. O sistema exibe um formulário com as informações necessárias para o cadastro (Nome, CEP, Endereço, Placa do Caminhão, Telefone para Contato, CPF, E-mail, Senha)
2. A transportadora preenche os campos do cadastro com os dados do novo motorista	
3. A transportadora confirma o cadastro.	
	4. O sistema informa que o cadastro foi bem sucedido.

Cenário(s) alternativo(s)

4.a. Alguma das informações não foi fornecida ou alguma das fornecidas está fora do formato esperado:

1. O sistema deverá indicar quais informações estão fora de formatação e qual o formato correto.
2. Retornar ao passo 2 do fluxo principal.

UC07: Estrutura Detalhada

Nome do caso de uso	**UC07 -** Login do motorista
Ator(es)	Motorista
Descrição	O motorista acessa sua conta pelo login.

Referências cruzadas	RF02
Pós-condições	O motorista pode acessar o aplicativo e usar seus serviços.

Cenário típico

Ator	Sistema
1. O motorista entra no aplicativo e acessa a opção de login.	
2. O motorista informa seus dados nos campos adequados (CPF e senha).	
	3. O sistema confirma se os dados informados corroboram com os da conta.
	4. O sistema permite a entrada do cliente.
5. O motorista acessa a página “Home” do aplicativo.	

Cenário(s) alternativo(s)

2.a O motorista esqueceu a senha:

1. O motorista aperta o botão de “Esqueceu a senha?”.
2. O sistema envia um e-mail com um link de redefinição de senha.
3. O motorista informa uma senha nova e confirma a senha nova em outro campo.
4. O sistema registra a alteração de informações.
5. Retorna ao passo 2 do fluxo principal.

3.a Dados não foram localizados:

1. Exibe uma mensagem indicando que os dados informados estão incorretos
2. Volta ao passo 2 do fluxo principal

UC08: Estrutura Detalhada

UC09: Estrutura Detalhada

Nome do caso de uso	**UC09 -** Login do administrador

Ator(es)	Administrador
Descrição	O administrador acessa sua conta pelo login.
Referências cruzadas	RF01
Pós-condições	O administrador pode acessar o sistema e usar seus serviços.

Cenário típico

Ator	Sistema
1. O administrador entra no portal e acessa a opção de login.	
2. O administrador informa seus dados nos campos adequados (CPF e senha).	
	3. O sistema confirma se os dados informados estão de acordo com os da conta.
	4. O sistema permite a entrada do administrador.
5. O administrador acessa a página “Home” do portal web.	

Cenário(s) alternativo(s)

2.a O administrador esqueceu a senha:

1. O administrador aperta o botão de “Esqueceu a senha?”.
2. O sistema envia um e-mail com um link de redefinição de senha.
3. O administrador informa uma senha nova e confirma a senha nova em outro campo.
4. O sistema registra a alteração de informações.
5. Retorna ao passo 2 do fluxo principal.

3.a Dados não foram localizados:

Exibe uma mensagem indicando que os dados informados estão incorretos

Volta ao passo 2 do fluxo principal.

UC010: Estrutura Detalhada

Nome do caso de uso	UC10 - Visualizar o pátio
Ator(es)	Administrador da fábrica
Descrição	O administrador visualiza a localização geográfica de todos os caminhões em relação às vagas e afins, para evitar possíveis conflitos físicos dentro do pátio.
Referências cruzadas	RF08, RF10
Gatilho	Apertou o botão de “Visualizar pátio” no portal web.
Pré-condições	O administrador deverá estar logado no sistema e a sua conta deve possuir o cargo “Administrador”.
Pós-condições	O administrador deve ficar ciente da posição em tempo real dos caminhões.

Cenário típico

Ator	Sistema
1. Este caso de uso começa quando um administrador clica no botão de “Visualizar pátio” no portal web.	
	2. O sistema informa que é necessário escolher um mapa de monitoramento e exibe as opções dentre as seções do pátio.
3. O administrador escolhe uma seção do pátio.	
	4. O sistema registra a opção e exibe aquela parte do pátio.
5. O administrador é notificado das posições dos caminhões.	

Cenário(s) alternativo(s)

- **5.a O administrador troca de seção de mapa**
    1. Retorna para o passo 3 do fluxo principal.

UC011: Estrutura Detalhada

Nome do caso de uso	**UC11 -** agendamento
Ator(es)	Administrador da fábrica, transportadora
Descrição	O administrador da fábrica escolhe uma transportadora para efetuar um transporte, definindo características específicas do caminhão. Dito isso, é enviado um veículo apropriado para a fábrica, escolhido pela transportadora. Quando todas as partes estiverem cientes, uma vaga é reservada e o tempo do processo é calculado. Logo após, o motorista e o administrador recebem notificações do agendamento.
Referências cruzadas	RF01, RF05, RF06, RF07, RF09,RF12, RF14, RF15, RF16, RF17, RF18, CF01, IN01, US01, US02, PR01
Gatilho	
Pré-condições	O administrador deverá estar logado no sistema e a sua conta deve possuir o cargo “Administrador”.
Pós-condições	O caminhão segue com o processo de transporte da carga (inbound ou outbound).

Cenário típico

Ator	Sistema
1. O administrador clica em “Fazer um agendamento”.	
	2. O sistema oferece uma lista de

	transportadoras, em um menu “dropdown”, e um campo de entrada “input” para especificação de característica de caminhões.
3. O administrador seleciona a transportadora e descreve as características desejadas.	
4. O administrador confirma sua pesquisa.	
	5. O sistema notifica a transportadora sobre o agendamento a ser realizado e as características desejadas pelo administrador.
6. A transportadora aceita o serviço.	
	7. O sistema reserva as vagas disponíveis para compor a rota.
	8. O sistema notifica o administrador (dados do motorista e caminhão), a transportadora (conclusão do agendamento) e o motorista (receber a rota).

Cenário(s) alternativo(s)

- **2.a Não há transportadoras cadastradas**
    1. O sistema exibe mensagem de erro e cancela o agendamento.
- **6.a Não há caminhões cadastrados ou disponíveis.**
    1. O sistema exibe mensagem de erro e cancela o agendamento.

- **Operacional:** Descreva os nós físicos do sistema e os processos, threads e componentes que rodam em cada um desses nós. Esta visão não é necessária se o sistema roda num único processo e num único thread.

9. Qualidade

Portanto, concluímos que o projeto arquitetado da forma exposta neste documento fornece maior manutenibilidade principalmente em relação ao modo de persistência de dados e mudanças de banco, além de facilitar o crescimento e expansão no futuro usando como base os padrões arquiteturais feitos ou facilmente trocando uma parte modular do código por outra que supra as novas necessidades. Outro aspecto importante que ressalta a qualidade do código estruturado é a facilidade de integrar novos programadores no time, pois o código fica mais legível e em caso de dúvida se pode recorrer a este próprio documento.

Enfim, a qualidade de um software é algo difícil de ser julgada, principalmente quando em dois

programas distintos todos os requisitos e funcionalidades são supridos, mas tendo em vista todos os benefícios citados neste documento, é possível enxergar com mais clareza a maior qualidade do software final depois de estruturado e implementado com todas essas diretrizes.

10. SOLID

Princípio da responsabilidade única

A classe do caminhão está infringindo o princípio da responsabilidade única, poderia ser criada uma classe separada para manter a mudança de estado dos caminhões e o tempo em cada estado (que pode ser uma informação útil para a fábrica).

A classe Fábrica está infringindo o princípio da responsabilidade única, suas tarefas poderiam ser divididas em outras classes como que derivam de uma classe mais simples como Funcionário que pode ser a classe mãe de AdiministradorFabrica e Gerente e esses podem dividir as funções que foram atribuídas a classe Fábrica.

Princípio de aberto/fechado

O princípio de aberto/fechado foi infringido ao mesmo tempo em que o princípio de responsabilidade única, ao darmos muitas funções para o administrador da fábrica. Nossa solução foi criar o agente administrador como um cargo acima, capaz de criar e modificar os funcionários da fábrica.

Princípio da inversão de dependência

As classes que atualmente implementam a persistência ferem o princípio da inversão de dependência, pois cada uma das classes no diagrama anterior implementam separadamente seus métodos relacionados a persistência; aplicando o princípio, essas classes passarão a utilizar interfaces de classes de fachada, que farão a abstração dos métodos de persistência. Isso ainda possibilitará que a implementação da persistência seja transparente às classes clientes, ao passo que possibilita empregar ainda diferentes meios e tecnologias, como SGBDs SQL, NoSQL, documentos JSON ou XML, dentre outros.

Princípio da substituição de Liskov

O princípio da substituição de Liskov não está sendo atendido, uma vez que existem objetos do tipo administrador e motorista, ambos com a função de realizar login, porém não existe nenhuma interface ou classe mãe que prevê essa funcionalidade em comum para que seja permitido que independente do tipo objeto passado em determinado método não seja necessário alterar as propriedades do programa para seu funcionamento desde que a classe mãe ou interface declare o comportamento em comum.

Princípio da segregação de interface

O princípio da segregação de interface está sendo infringido pela classe Motorista, onde ela podia ser uma extensão de um classe mãe compartilhada por exemplo com o administrador, já que eles tem métodos em comum, como login, e ter suas implementações especificidades de métodos internos neles mesmos.