

picê 

DMF

Dimensionamento de Frota



Nome de dos componentes da equipe

Anita Ayres

Bruno Bozelli

Daniel Reis Laranjeira

Daniel Xavier

Edilson Ferreira Gomes

Enzo Viola Bernardes

Felipe Lacerda

Igor Daniel

João Victor

João Gabriel

Jozilene Joana Santos

Maria Júlia Cardoso

Mateus Hiroshi

Matheus Ramos da Silva

Matheus Silkinaite de Brito

Pedro Araújo Dienes

Thomas Ferreira Lopes

Victor Nunes Vinicius Soares de Lima

Yuri Yan Ymoto Simões



Projeto de desenvolvimento de software apresentado ao curso de Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Guarulhos sob orientação dos professores (Jane Eberson, Marcos José Corrêa Bueno)

Criação do Documento

Elaborado por: Marcos Kayan de Santana Santos	Data: 17/05/2022
Orlando Alves Meirelis	

Controle de Versão

Versão	Data	Páginas	Mudanças	Comentários
1.0	08/05/2023	24	Estilização do Layout	Capa, cabeçalho e rodapé adicionados

Composição da Equipe de desenvolvimento



Número de ordem	Função	Descrição das responsabilidades	Nome
2	Analistas	Análise e especificação de requisitos	Anita, Yuri
3	Desenvolvedores	Desenho e Implementação	Felipe Lacerda, Fernando Castelhana, Matheus Ramos, Thomas Ferreira, Pedro Araújo
4	Testadores	Testes e gestão da qualidade	Matheus Silkinaitė, João Victor, Pedro Araujo Dienes
5	Projetista de BD	Documentação e projeto de Banco de dados	Enzo, Luís Henrique, Vinicius Soares
6	Documentador	Desenvolvimento e integração de toda documentação constante nesse arquivo	Daniel Reis Laranjeira



Sumário

Sumário

1. Documento de Visão	6
1.1. Resultado do Projeto.....	6
1.1.1. Finalidade do Projeto.....	6
1.1.2. Diagnóstico da Situação Atual	6
1.1.3. Benefícios Esperados	6
1.2. Caracterização dos Atores	7
1.2.1. Áreas de Negócio	7
1.2.2. Atores Externos	7
1.3. Modelo Funcional.....	7
1.3.1. Diagrama de caso de uso	7
1.3.2. Descrição do caso de uso.....	8
1.4.1. Requisitos Funcionais.....	14
1.4.2. Regras de Negócio	16
1.4.3. Requisitos não funcionais	17
1.5. Modelo Estrutural	20
1.5.1. Diagrama de Classes de Entidade.....	20
1.6. Modelagem e projeto de Banco de Dados	21
2. Solução Tecnológica	24
2.1. Modelo da Arquitetura.....	24
O modelo de arquitetura adotado para o projeto é DAS N+1	24
2.2.2. Descrição da Arquitetura	24
2.2.3. Desenho da Arquitetura	25
3. Glossário de Negócio	44
4. Projeto de Interface de Software.....	45
5. Planejamento da Qualidade de Software.....	46



INTRODUÇÃO

Nós, alunos do curso de Análise e Desenvolvimento de Sistemas – ADS, Fatec-Guarulhos, Turma 2º S/2023, no segundo semestre do curso, iniciamos o desenvolvimento de um projeto de software, sob a orientação da professora Jane Maria dos Santos Eberson. O projeto Dimensionamento de Frota – DMF foi solicitado pelo Professor Mestre Marcos José Corrêa Bueno e tem por objetivo a automatização de um business game, atualmente aplicado pelo professor, de forma lúdica e manual, com caminhões de papéis e de brinquedos, na disciplina de Gestão de Transportes e Roteirização, ministrada no 6º semestre dos cursos de Logística. Os alunos precisam definir as características dos caminhões (comprimento, largura, altura), capacidade, além dos custos, fixos e variáveis, inerentes à tomada de decisão. O jogo aplicado aos alunos tem por finalidade estimular a competição no mercado de transporte visando a delimitação de uma frota a ser contratada por um cliente. Com o business game, o professor fará interação de uma forma mais dinâmica com os alunos. Já os alunos, entre si, serão submetidos à uma atividade comum da área de logística.



Projeto Conceitual

Versão de Orientação a Objetos - UML

Dimensionamento de Frota

1. Documento de Visão

A finalidade deste documento é demonstrar uma descrição das aplicabilidades, atributos e limitações do Sistema de Dimensionamento de Frota – **DMF**. Através deste cliente e outros stakeholders que analisarão a conformidade do projeto com as demandas necessárias para o dimensionamento e aprendizado operacional, desta forma, os desenvolvedores obterão um entendimento de tudo o que deve ser desenvolvido e as perspectivas de acordo com o planejado.

1.1. Resultado do Projeto

Espera-se com o projeto gerar dinamismo, motivação e engajamento dos participantes nas atividades educacionais colaborando para o desafio do professor em manter os alunos interessados durante toda a dinâmica da aula, e desta forma, proporcionando resultados concretos de aprendizado.

1.1.1. Finalidade do Projeto

Gamificação da aula de dimensionamento de frota, empregando o processo de aprendizagem com mecanismos de jogo.

1.1.2. Diagnóstico da Situação Atual

Seq	Descrição do problema
1	<i>A aula prática ocorre com a utilização de recursos manuais</i>
2	<i>Os recursos envolvem planilha, papel, e prospecto de papel</i>
3	A aplicação manual não proporciona a vivência da prática do dimensionamento de frota na vida real.

1.1.3. Benefícios Esperados

Seq	Descrição do benefício
1	<i>A aula se tornará mais prática com a realidade</i>
2	<i>Despertará o interesse e o engajamento de todos</i>



3	<i>O objetivo do ensino atingirá o maior número de alunos</i>
4	<i>Os alunos não irão se esquecer dessa experiência</i>

1.2. Caracterização dos Atores

1.2.1. Áreas de Negócio

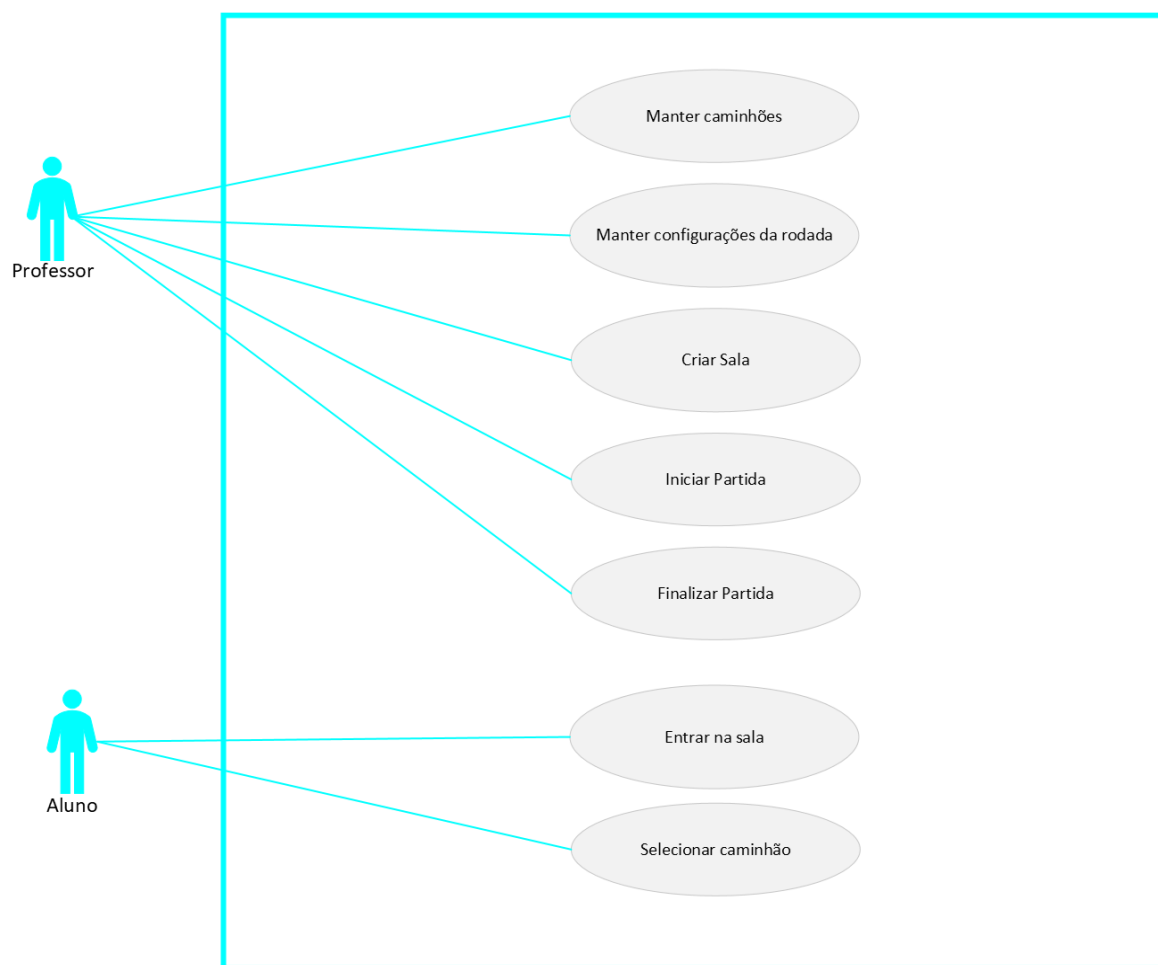
Parte Envolvida:	Professor
Finalidade:	Ensinar os alunos
Atribuição:	Mediador do jogo

1.2.2. Atores Externos

Ator:	Alunos
Finalidade:	Aprender a disciplina, desenvolver a análise crítica, em negociações e tomadas de decisões.
Atribuição:	Jogar o “Business Game” de logística, aprender sobre formas de analisar opções de negociação em situações adversas que ocorrem no cenário real.

1.3. Modelo Funcional

1.3.1. Diagrama de caso de uso



1.3.2. Descrição do caso de uso

Caso de uso Nr:	UC001
Nome Caso de uso:	Manter caminhão - Incluir
Atores:	Professor
Descrição:	Incluir caminhão
Início:	Estar na tela de cadastro
Pré condição:	Caminhão não ter cadastro
Pós condição:	Caminhão cadastrado
Fluxo usuário: normal	1- O professor insere os dados do caminhão 2- O professor confere os dados do caminhão 3- O professor salva o cadastro do caminhão
Fluxo sistema: normal	1- O sistema recebe os dados do caminhão 2- O sistema recebe a confirmação dos dados 3- O sistema armazena os dados do caminhão
Exceções:	1- O sistema informa: caminhão já está cadastrado 2- O sistema informa: Dados incompletos, preencha todos os campos
Alternativo:	Não se aplica

Caso de uso Nr:	UC002
Nome Caso de uso:	Manter caminhão - Excluir



Atores:	Professor
Descrição:	Excluir um caminhão
Início:	Estar na tela de cadastro
Pré condição:	Caminhão deve estar cadastrado
Pós condição:	Caminhão excluído com sucesso
Fluxo usuário: normal	1- O professor localiza o caminhão no cadastro 2- O professor confere os dados do caminhão 3- O professor exclui o caminhão
Fluxo sistema: normal	1- O sistema localiza o caminhão 2- O sistema recebe o comando excluir 3- O sistema exclui o caminhão
Exceções:	1- O sistema informa: Caminhão não cadastrado
Alternativo:	O sistema informa: Informe um caminhão cadastrado

Caso de uso Nr:	UC003
Nome Caso de uso:	Manter caminhão - Atualizar
Atores:	Professor
Descrição:	Atualizar cadastro
Início:	Estar na tela de cadastro de caminhão
Pré condição:	Caminhão deve possuir cadastro
Pós condição:	Cadastro do caminhão atualizado com sucesso
Fluxo usuário: normal	1- O professor localiza o cadastro do caminhão 2- O professor confirma o caminhão 3- O professor altera os dados necessários 4- O professor confirma a atualização 5- O professor salva a alteração
Fluxo sistema: normal	1- O sistema localiza o caminhão 2- O sistema recebe as alterações do cadastro 3- O sistema atualiza os dados do caminhão 4- O sistema armazena as alterações
Exceções:	1- O sistema informa: Caminhão não cadastrado
Alternativo:	O sistema informa: Informe um caminhão cadastrado

Caso de uso Nr:	UC004
Nome Caso de uso:	Manter caminhão - Pesquisar
Atores:	Professor
Descrição:	Pesquisar cadastro
Início:	Estar na tela de cadastro
Pré condição:	Caminhão estar cadastrado
Pós condição:	Sistema localizar e apresentar os dados do caminhão
Fluxo usuário: normal	1- O professor informa os dados do caminhão 2- O professor seleciona a pesquisa do caminhão 3- O professor visualiza os dados do caminhão
Fluxo sistema: normal	1- O sistema recebe os dados da pesquisa 2- O sistema recebe o comando pesquisar 3- O sistema localiza o caminhão 4- O sistema apresenta o cadastro do caminhão
Exceções:	1- O sistema informa caminhão não encontrado



Alternativo:	1- O sistema: Informe os dados válidos para pesquisa
Caso de uso Nr:	UC005
Nome Caso de uso:	Manter configurações da rodada - Incluir
Atores:	Professor
Descrição:	Incluir os dados da carga e dados da rota
Início:	Estar na tela configurações da rodada
Pré condição:	Não ter o cadastro dos dados da carga, dados da rota, tempo da partida, tempo da trava, pacote de caminhões.
Pós condição:	Dados da carga, dados da rota, tempo da partida, tempo da trava, pacote de caminhões cadastrados.
Fluxo usuário: normal	<ol style="list-style-type: none">1- O professor insere os dados da carga, rota, viagem, tempo da trava, tempo de cada rodada e escolhe o pacote de caminhões.2- O professor confere os dados da carga, rota, viagem, tempo da trava, tempo de cada rodada e escolhe o pacote de caminhões.3- O professor salva o cadastro da carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote de caminhões.
Fluxo sistema: normal	<ol style="list-style-type: none">1- O sistema recebe os dados da carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote de caminhões.2- O sistema recebe a confirmação dos dados da carga, rota, viagem, tempo de trava, tempo de cada rodada e pacote de caminhões.3- O sistema armazena os dados da carga, rota, viagem, tempo de trava, tempo de cada rodada e pacote de caminhões.
Exceções:	O sistema informa: Dados incompletos, preencha todos os campos
Alternativo:	Não se aplica

Caso de uso Nr:	UC006
Nome Caso de uso:	Manter configurações da rodada - Excluir
Atores:	Professor
Descrição:	Excluir carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote de caminhões.
Início:	Estar na tela configurações da rodada.
Pré condição:	Carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote de caminhões devem estar cadastrados.
Pós condição:	Carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote de caminhões excluídos com sucesso.
Fluxo usuário: normal	<ol style="list-style-type: none">1- O professor localiza: carga, rota, viagem, tempo da trava, tempo de cada rodada e pacotes de caminhões.2- O professor confere os dados cadastrados para a configuração da rodada.3- O professor seleciona a opção excluir os dados da configuração da rodada.
Fluxo sistema: normal	<ol style="list-style-type: none">1- O sistema localiza: carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote dos caminhões.2- O sistema recebe o comando excluir.3- O sistema exclui os dados da configuração da rodada.



Exceções:	O sistema informa: dados de pesquisa não localizados.
Alternativo:	O sistema informa: Informe os dados cadastrados em sistema.

Caso de uso Nr:	UC007
Nome Caso de uso:	Manter configurações da rodada - Atualizar
Atores:	Professor
Descrição:	Atualizar cadastro da carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote de caminhões.
Início:	Estar na tela configurações da rodada.
Pré condição:	Carga, rota, viagem, tempo da trava, tempo de cada rodada e quantidade de rodadas deve possuir cadastro.
Pós condição:	Cadastro da carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote de caminhões atualizada com sucesso.
Fluxo normal usuário:	<ol style="list-style-type: none">1- O professor localiza o cadastro da carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote de caminhões.2- O professor confirma os dados das configurações da rodada.3- O professor altera os dados necessários.4- O professor confirma a atualização.5- O professor salva a alteração.
Fluxo normal sistema:	<ol style="list-style-type: none">1- O sistema localiza a carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote de caminhões.2- O sistema recebe as alterações.3- O sistema atualiza os dados.4- O sistema armazena as alterações.
Exceções:	O sistema informa: Dados não localizados.
Alternativo:	O sistema informa: Informe as configurações para a rodada cadastrada.

Caso de uso Nr:	UC008
Nome Caso de uso:	Manter configurações da rodada - Pesquisar
Atores:	Professor
Descrição:	Pesquisar cadastro da carga, rota, viagem, tempo da trava, tempo de cada rodada e pacotes de caminhões.
Início:	Estar na tela configurações da rodada.
Pré condição:	Carga, rota, viagem, tempo da trava, tempo de cada rodada e pacotes de caminhões devem estar cadastrados.
Pós condição:	Sistema localizar e apresentar os dados da carga, rota, viagem, tempo da trava, tempo de cada rodada e pacotes de caminhões.
Fluxo normal usuário:	<ol style="list-style-type: none">1- O professor informa os dados da carga, rota, viagem, tempo da trava, tempo de cada rodada e pacotes de caminhões.2- O professor seleciona a pesquisa das configurações da rodada.3- O professor visualiza os dados das configurações da rodada.
Fluxo normal sistema:	<ol style="list-style-type: none">1- O sistema recebe os dados da pesquisa.2- O sistema recebe o comando pesquisar.3- O sistema localiza os dados das configurações da rodada.4- O sistema apresenta o cadastro da carga, rota, viagem, tempo da trava, tempo de cada rodada e pacote de caminhões.
Exceções:	2- O sistema informa dados das configurações da rodada não localizado
Alternativo:	2- O sistema: Informe os dados válidos para pesquisa.



Caso de uso Nr:	UC009
Nome Caso de uso:	Criar sala
Atores:	Professor
Descrição:	Criar uma sala para hospedar os jogadores durante a partida.
Início:	Estar na etapa de criação de sala.
Pré condição:	Não estar com nenhuma sala ativa.
Pós condição:	Criar sala que hospede os jogadores.
Fluxo normal usuário:	1- O professor seleciona a opção criar sala. 2- O professor insere o nome / código da sala. 3- O professor seleciona a opção criar e voltar
Fluxo normal sistema:	1- O sistema recebe o nome / código da sala. 2- O sistema habilita uma nova sala para hospedar os jogadores. 3- O sistema gera a nova sala com o nome / código que foi inserido.
Exceções:	1- O sistema informa: para criar uma nova sala, deve-se encerrar a outra existente. 2- O sistema informa: para criar uma nova sala é preciso inserir um nome válido.
Alternativo:	O sistema informa: inserir o nome da sala.

Caso de uso Nr:	UC013
Nome Caso de uso:	Iniciar partida
Atores:	Professor
Descrição:	Iniciar uma partida do jogo.
Início:	Quando finalizado todas as configurações de rodada.
Pré condição:	Ter alguma sala ativa e jogadores hospedados.
Pós condição:	Dar início a uma partida.
Fluxo normal usuário:	1- O professor seleciona a opção de iniciar a partida. 2- O professor confirma a ação.
Fluxo normal sistema:	1- O sistema recebe o comando de iniciar a partida. 2- O sistema executa o comando e a partida é iniciada.
Exceções:	O sistema informa: Para iniciar a partida você precisa preencher todos os campos informados.
Alternativo:	O sistema informa: Tem certeza de que deseja iniciar a partida?

Caso de uso Nr:	UC014
Nome Caso de uso:	Finalizar partida
Atores:	Professor
Descrição:	Finalizar uma partida do jogo.
Início:	Após o início de uma partida.
Pré condição:	Ter uma partida válida em andamento.
Pós condição:	Finaliza a partida mostrando o ranking.
Fluxo normal usuário:	1- O professor seleciona a opção de finalizar a partida. 2- O professor confirma a ação.



Fluxo normal sistema:	4- O sistema recebe o comando de finalizar a partida. 5- O sistema executa o comando e a partida é encerrada. 6- O sistema mostra os dados da partida e o ranking.
Exceções:	Não se aplica.
Alternativo:	1- O sistema informa: Tem certeza de que deseja finalizar a partida?

Caso de uso Nr:	UC015
Nome Caso de uso:	Entrar na sala.
Atores:	Alunos
Descrição:	Entrar na sala criada pelo professor.
Início:	Estar na tela de login.
Pré condição:	Não estar com o usuário criado na sala criada pelo professor.
Pós condição:	Estar com o usuário criado pelo professor.
Fluxo normal usuário:	4- O aluno visualiza a sala criada pelo professor. 5- O aluno insere o nome na sala. 6- O aluno entra na sala.
Fluxo normal sistema:	7- O sistema recebe o nome do aluno na sala. 8- O sistema libera a entrada do aluno na sala.
Exceções:	3- O sistema informa: Para entrar na sala é necessário informar o nome completo.
Alternativo:	Não se aplica

Caso de uso Nr:	UC017
Nome Caso de uso:	Selecionar caminhão.
Atores:	Alunos
Descrição:	Seleciona o caminhão, fazer análise, decidir escolher ou liberar e selecionar outro.
Início:	Estar na tela de selecionar caminhão.
Pré condição:	Não estar com nenhum caminhão selecionado.
Pós condição:	Estar com um caminhão selecionado.
Fluxo normal usuário:	1- O aluno seleciona o caminhão. 2- O aluno faz a análise do caminhão. 3- O aluno decide escolher o caminhão. 4- O aluno terá que definir o caminhão dentro do tempo estipulado.
Fluxo normal sistema:	1- O sistema bloqueia caminhão selecionado. 2- O sistema apresenta as informações do caminhão para análise. 3- O sistema trava a troca do caminhão pelo tempo determinado. 4- O sistema informa final da partida.
Exceções:	1- O sistema informa: caminhão indisponível para análise. 2- O sistema informa: troca de caminhão bloqueada.
Alternativo:	Não se aplica.



1.4. Requisitos do Projeto

1.4.1. Requisitos Funcionais

Número		Prioridade	Revisado	
RF001	O sistema deverá ter a tela para o login de acesso do professor, com a opção não tenho cadastro, ao selecionar, o mesmo, deverá ser redirecionado para a tela de cadastro.	Alta	<Sim>	<Não>
RF002	O sistema deverá ter uma tela para cadastro de novo professor com nome, e-mail e senha.	Alta		
RF003	O sistema deverá ter a tela de cadastro dos caminhões com as informações do nome comprimento, largura, altura e capacidade, custo fixo mensal e custo variável.	Alta		



RF004	O sistema deverá ter a opção de consultar os caminhões cadastrados.	Baixa		
RF005	O sistema deverá ter a tela para o cadastro das configurações para a rodada com os campos: tempo da partida, tempo da trava, dados da carga: comprimento, peso, largura, altura e quantidade e dados da rota: distância, dias e valor da pernoite.	Alta.		
RF006	Na tela de configurações da rodada deverá ter a opção selecionar caminhões, o professor deverá ser redirecionado para a tela de seleção de caminhões.	Alta		
RF007	O sistema deverá ter uma tela para seleção de caminhões.	Alta		
RF008	A tela de seleção de caminhões apresentará a lista de caminhões para o professor selecionar.	Alta		
RF009	O sistema deverá ter a tela de criar pacote de caminhões	Alta		
RF010	Na tela criar pacote de caminhões, o sistema deverá apresentar a lista de caminhões cadastrados para o professor selecionar e criar novo pacote.	Alta		
RF011	O sistema deve ter um campo voltado a criação da sala, para inserir o nome / código e criar a sala que hospedará os jogadores.	Alta		
RF012	O sistema deve ter uma opção de iniciar a partida.	Alta		
RF013	O sistema deve ter uma opção para executar a ação de finalizar a partida a critério do professor, a qualquer momento.	Alta		
RF014	O sistema deverá ter uma tela para o login do aluno com o campo para o nome completo e o nome da sala, criada pelo professor.	Alta		
RF015	O sistema deverá ter uma tela para o aluno analisar o caminhão.	Alta		



RF016	Na tela do aluno deverá ter o tempo da partida e o tempo para a escolha do caminhão.	Alta		
RF017	Na tela do aluno deverá ter uma lista das imagens dos caminhões para a seleção de um caminhão por vez para a análise.	Alta		
RF018	Ao Selecionar o caminhão, o mesmo, ficará indisponível para os demais alunos até a sua liberação.	Alta		
RF019	Na tela do aluno após a seleção do caminhão o sistema apresentará, as especificações do caminhão, os cálculos dos custos em relação aos seus parâmetros com os dados da carga, rota e viagem para a análise.	Alta		
RF020	Se o aluno não escolher o caminhão dentro do tempo para a escolha o sistema deverá liberar o caminhão automaticamente, ao término do tempo.	Alta		
RF021	O aluno poderá analisar quantos caminhões forem possíveis dentro do tempo da partida, respeitando o tempo para a escolha do caminhão.	Alta		
RF022	A tela do aluno deverá ter um campo com o histórico do último caminhão analisado dentro de uma mesma partida.	Baixa		
RF023	Ao término do tempo de cada rodada o sistema apresentará o vencedor da rodada.	Baixa		
RF024	Ao término de todas as rodadas o sistema apresentará o ranking das rodadas.	Alta		
RF025	O jogo deverá ter um design e uma estética lúdica.	Média		

1.4.2. Regras de Negócio



Número	Descrição
RN001	O aluno poderá selecionar somente um caminhão por vez para análise.
RN002	O caminhão selecionado ficará disponível para os demais alunos somente após a sua liberação.
RN003	O aluno poderá trocar de caminhão quantas vezes forem possíveis somente dentro do tempo estipulado para a rodada.
RN004	O sistema possui uma trava com um tempo mínimo para a análise, após o término da trava, o caminhão será liberado automaticamente, caso o aluno não selecione a opção escolher caminhão.
RN005	O professor irá estabelecer o tempo da trava para análise do caminhão.
RN006	O professor estabelecerá o tempo de duração para cada rodada.
RN007	O professor estabelecerá a quantidade de rodadas
RN008	O professor irá liberar a entrada dos usuários.
RN009	O professor deve iniciar a partida quando estiver pronto e todas as informações fornecidas.
RN010	O sistema fará o reconhecimento para mostrar o ganhador de cada rodada e ao final, o vencedor geral com o <i>ranking</i> .
RN011	O sistema encerrará a partida quando todas as rodadas forem terminadas.
RN012	O aluno entra em uma sala somente com um nome válido.

1.4.3. Requisitos não funcionais

Número	Descrição	Prioridade	Revisado	
RNF001	O software deve ser acessado pelo administrador do jogo (professor) através de login e senha.	Alta		
RNF002	O software deve ser acessado pelos jogadores (alunos) apenas através de login (nome da sala), não sendo necessário senha.	Alta		
RNF003	As informações do caminhão (comprimento, largura, altura) e da carga deverão ser protegidas contra acesso não autorizado. Ou seja, só poderão ser criadas, atualizadas e selecionadas para o jogo pelo administrador do jogo (professor).	Alta		



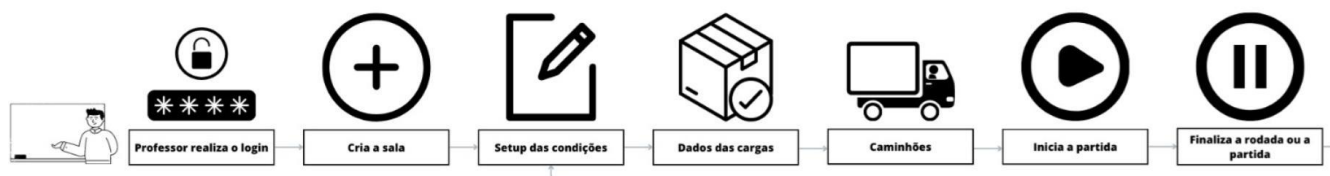
RNF004	O software deverá ser capaz de cadastrar novas informações de caminhões e cargas conforme necessidade do administrador do jogo.	Alta		
RNF005	O administrador do jogo (professor) deve criar, atualizar ou deletar informações de caminhões e cargas com agilidade e facilidade.	Alta		
RNF006	Os jogadores (alunos) devem ter facilidade em visualizar as informações do jogo (características da carga e dos caminhões, duração da rodada, etc.) e ter clareza para executar as tarefas solicitadas do jogo.	Alta		
RNF007	Os alunos devem ter facilidade em escolher e selecionar os caminhões durante a rodada do jogo.	Alta		
RNF008	O software deve estar disponível em período integral nos computadores e servidores onde estiver instalado.	Alta		
RNF009	O software deve ser compatível com os computadores e servidores das instituições do Centro Paula Souza, com alterações mínimas.	Alta		
RNF010	O software armazenará os dados de login dos jogadores (alunos).	Alta		
RNF011	O software deve ser capaz de suportar uma sala de aula com 40 alunos.	Alta		
RNF012	O software deve escalar, ou seja, suportar manipulações (cadastros, exclusões etc.) de dados de forma ilimitada	Alta		
RNF013	O software deve cumprir todas as leis e regulamentos aplicáveis.	Alta		
RNF014	O software deve ser confiável e atender aos requisitos do usuário.	Alta		

Requisitos de Implementação

Apresenta os requisitos que estabelecem o que os usuários esperam da solução em termos de implementação.

Requisitos Tecnológicos

Evidencia as tecnologias a serem utilizadas para a implementação da solução.



Nº	Nome do Requisito	Implementação do Requisito
<Nº>	Plataforma	desktop



<Nº>	Tecnologia	Visual Studio Windows forms .NET C# Access
<Nº>	Implantação	Disponibilização Instalação Cadastro Verificação de cadastro

Requisitos Inversos

Apresenta os requisitos que não serão contemplados durante o atendimento da solicitação, ou seja, aqueles requisitos que são tão desejáveis, próximos e naturais de serem atendidos, que podem induzir à ideia de que serão atendidos, quando na verdade não serão. A intenção é deixar bem claro o que não vai definitivamente ser atendido com a solução.

1.5. Modelo Estrutural

1.5.1. Diagrama de Classes de Entidade

1.5.2. Relação de Classes de Objetos persistentes



1.6. Modelagem e projeto de Banco de Dados

<Breve explanação das decisões sobre projeto de banco de dados>

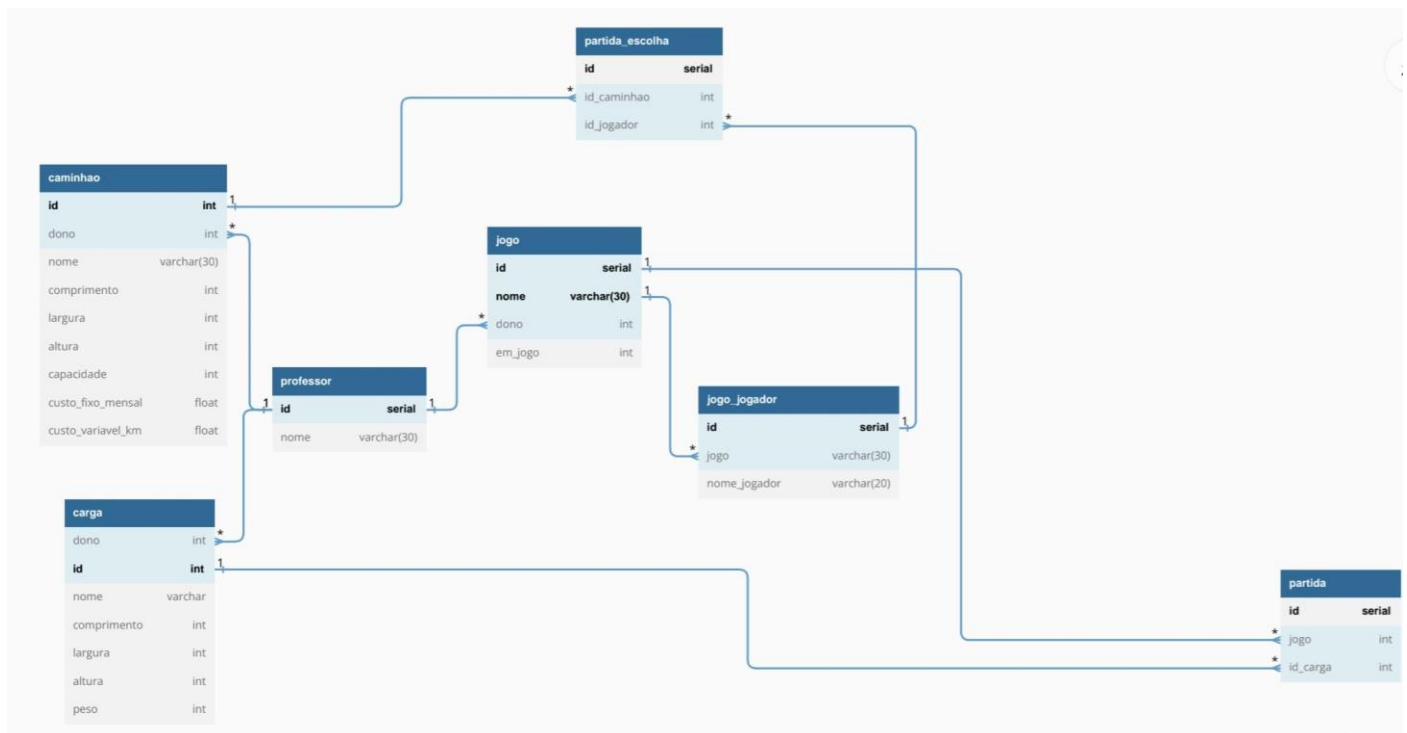
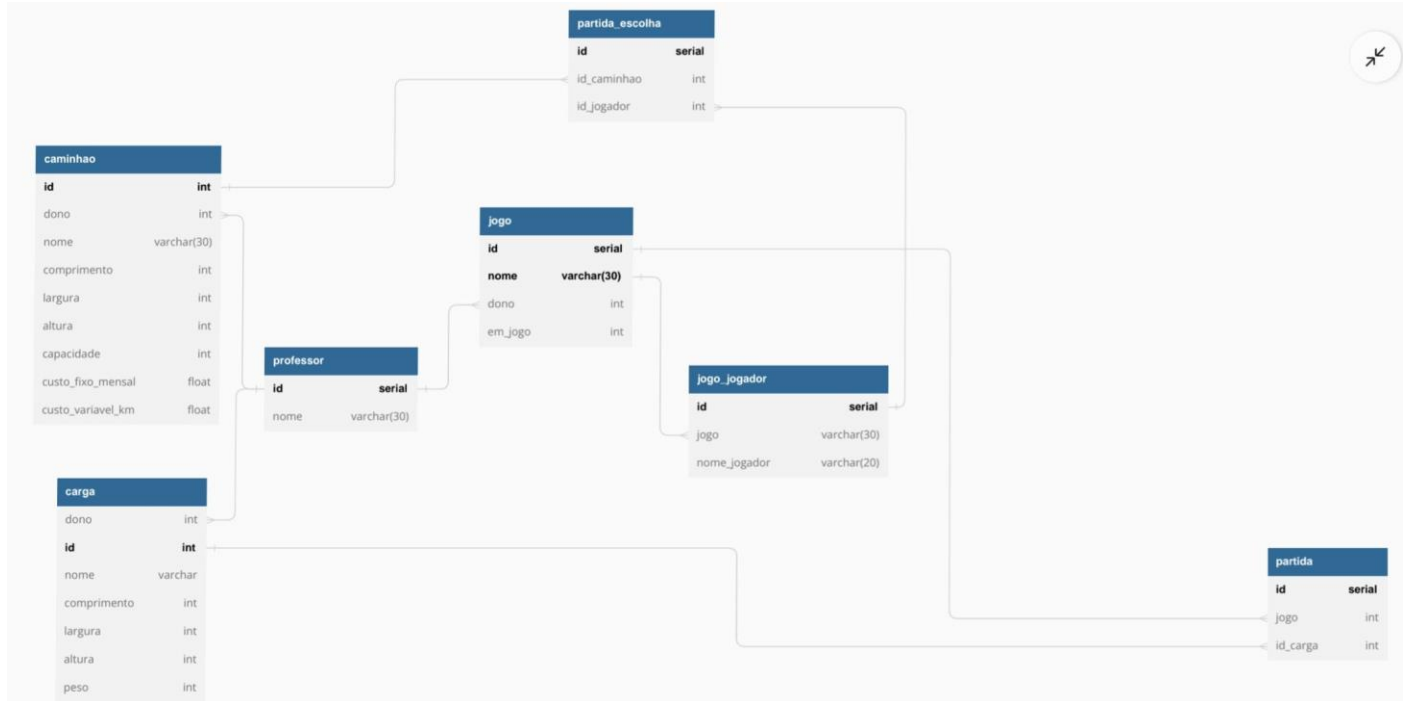
1.6.1 Estrutura de Dados

Relacionado a Banco de Dados foi decidido a utilização da plataforma “Acess”, como ferramenta de desenvolvimento.

Em reuniões com os desenvolvedores, foi pensado em relação ao projeto quais itens seriam armazenados no Banco de Dados, como:

- Dados do caminhão
- Dados da carga
- Dados do professor
- Dados do aluno
- Dados do jogo (sala, escolha do caminhão)

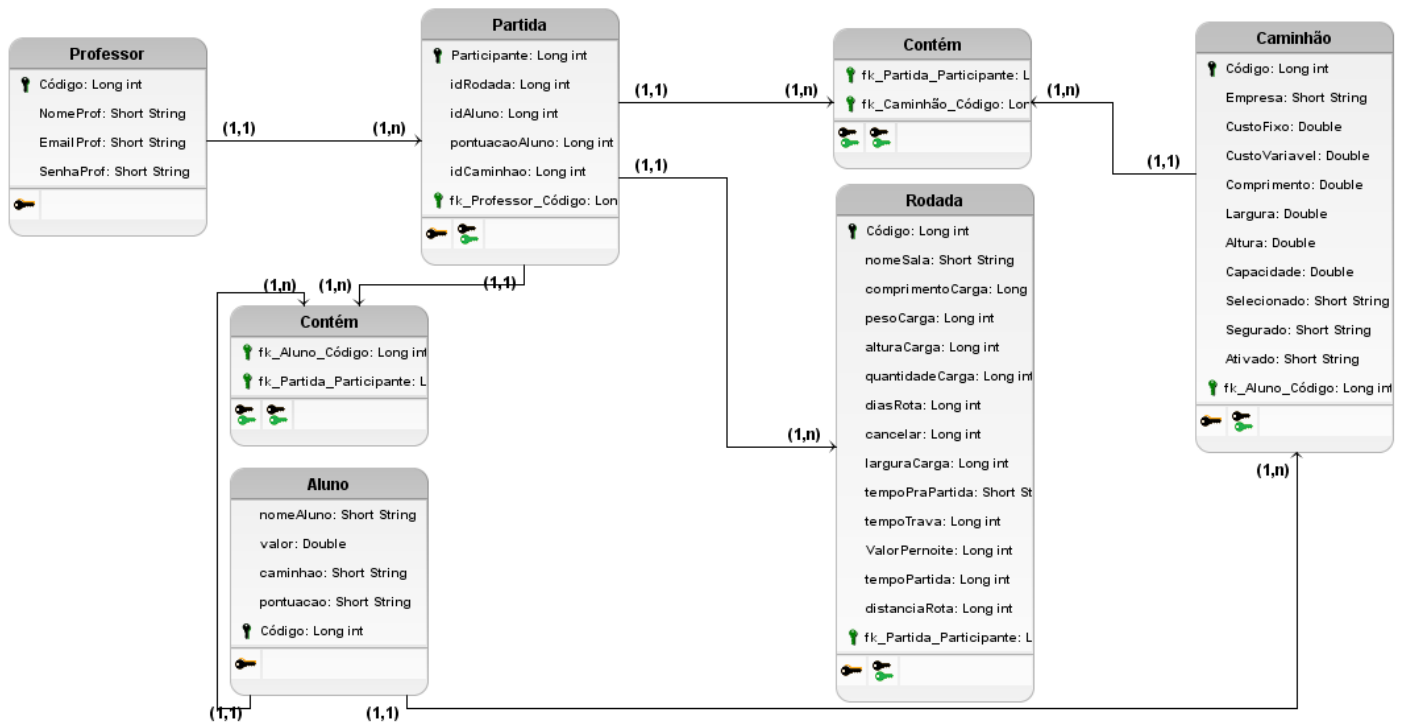
Para a interpretação do Banco de Dados, utilizando-se múltiplos acessos foi pensado utilizar a rede compartilhada entre os computadores da sala.



22



1.6.3. Modelo Lógico



1.6.4. Modelo Físico



2. Solução Tecnológica

A necessidade do cliente em gameficar os exercícios de dimensionamento de frota levaram ao desenvolvimento do software com a seguinte arquitetura demonstrada nos tópicos abaixo:

2.1. Modelo da Arquitetura

O modelo de arquitetura adotado para o projeto é DAS N+1

2.2.2. Descrição da Arquitetura

Esse DAS resume a arquitetura a partir de múltiplas visões abaixo demonstradas:

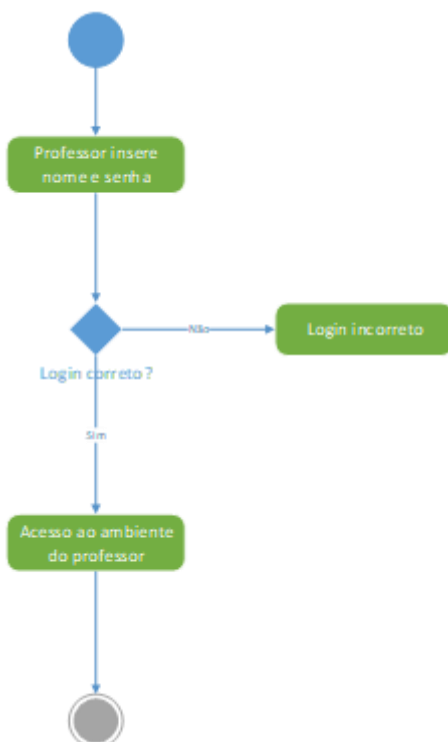
- Visão Implementação – Para a visão da implementação, a equipe de desenvolvedores elaborou diagramas de atividade para cada atividade dentro do sistema. As atividades foram separadas para facilitar o entendimento, pois juntas formariam um diagrama que, por conta de seu tamanho, poderia não evidenciar cada uma de suas partes.
- Visão da Implantação – Para a visão da implantação, a equipe de banco de dados elaborou diagramas de sequência para cada parte dentro do sistema. Os diagramas foram separados para facilitar o entendimento.
- Visão de Processos – As visões arquiteturais permitem diversos panoramas, visando abordar um dos pontos importantes do projeto, os analistas de sistemas desenvolveram o diagrama de classe que compõe a visão de processos para ilustrar os processos e as linhas de execução suas responsabilidades, cooperação e alocação de elementos lógicos como camadas, subsistemas, classes, entre outros.
- Visão Conceitual – Para a visão conceitual foi implementado um diagrama de caso de uso no qual engloba o funcionamento do sistema como um todo. Denota-se os papéis dos atores e suas possíveis ações para um entendimento coeso do sistema.
- Visão Lógica - A visão lógica oferece uma visão visual das opções de design arquitetônico disponíveis, ajudando a tomar decisões informadas sobre a arquitetura de software. Ele mostra as diferentes escolhas, critérios de decisão e resultados potenciais, simplificando o processo de seleção de arquitetura.



2.2.3. Desenho da Arquitetura

- Visão Implementação:

Efetuar login de professor

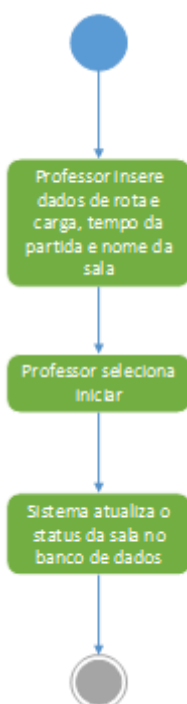


Cadastrar professor

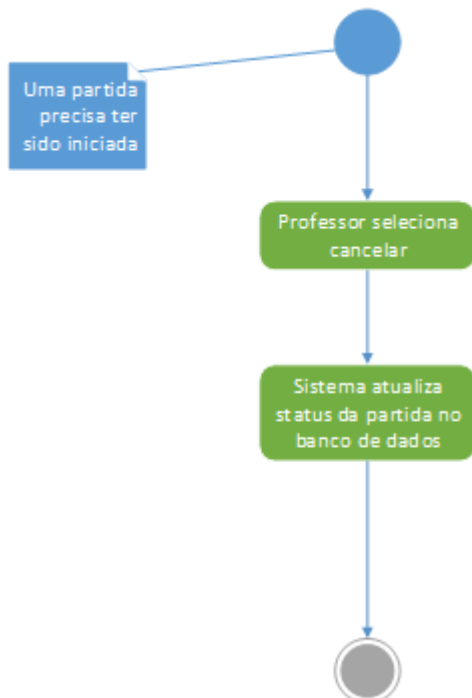




Iniciar partida

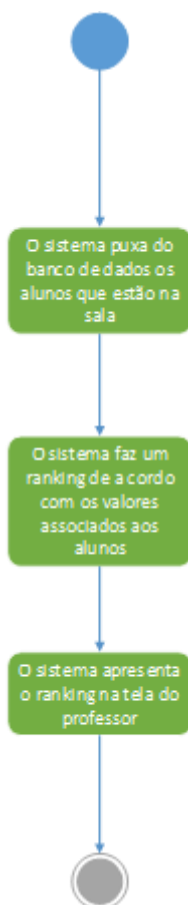


Cancelar partida

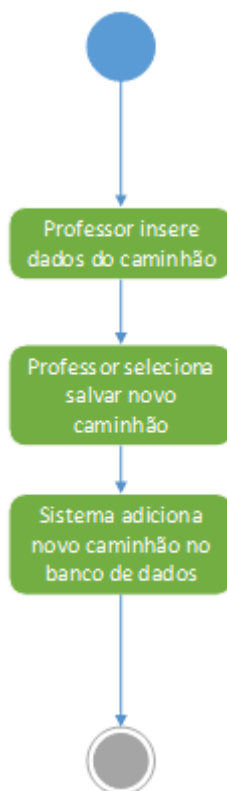




Listar alunos

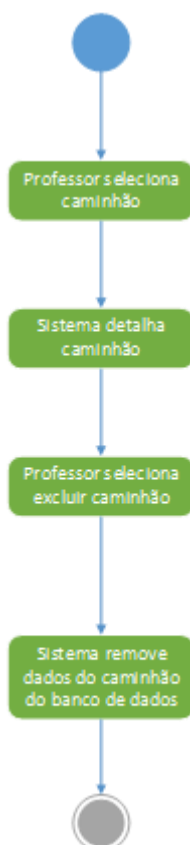


Adicionar caminhão

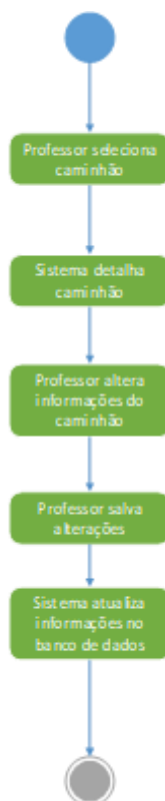




Excluir caminhão

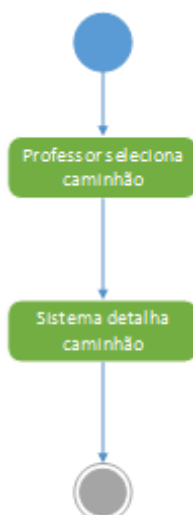


Alterar caminhão

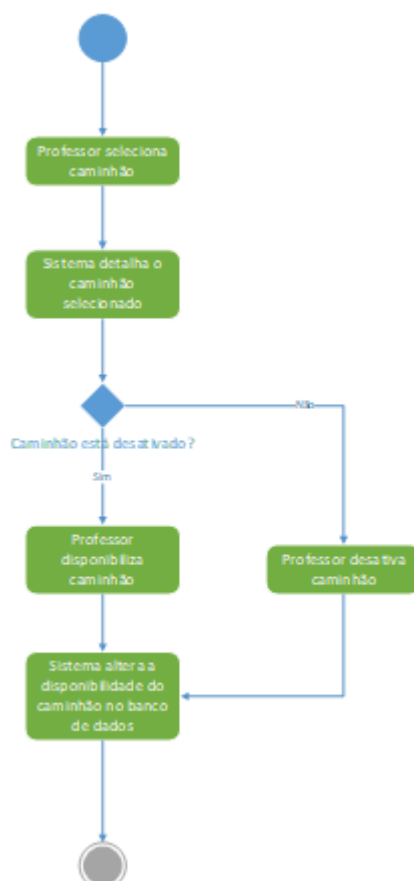




Visualizar caminhão

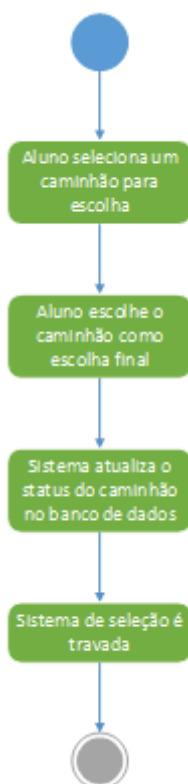


Alterar disponibilidade do caminhão

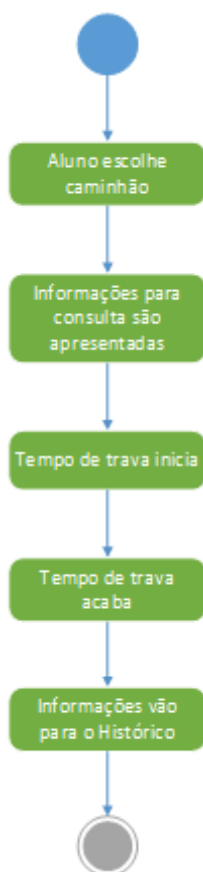




Escolher caminhão

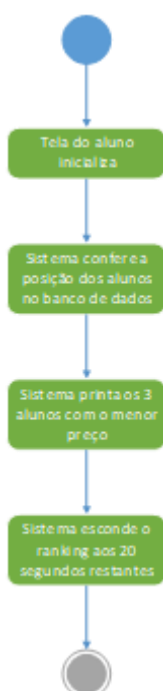


Fazer histórico de caminhão

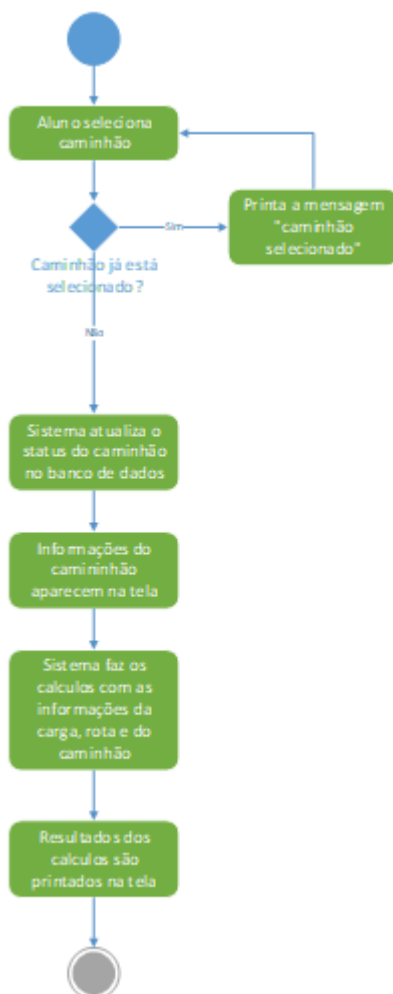




Fazer ranking

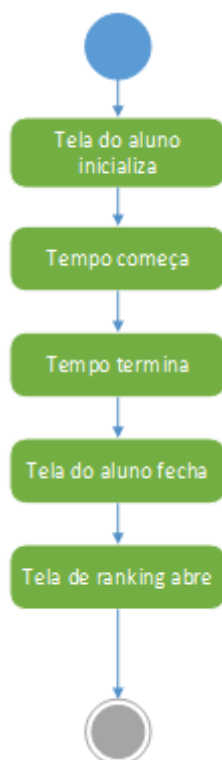


Selecionar caminhão

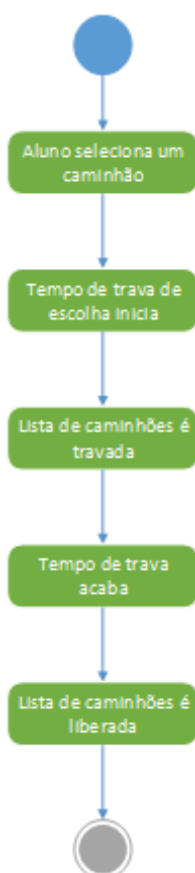




Administrar temporizador da partida



Travar seleção de caminhão





Efetuar login como aluno

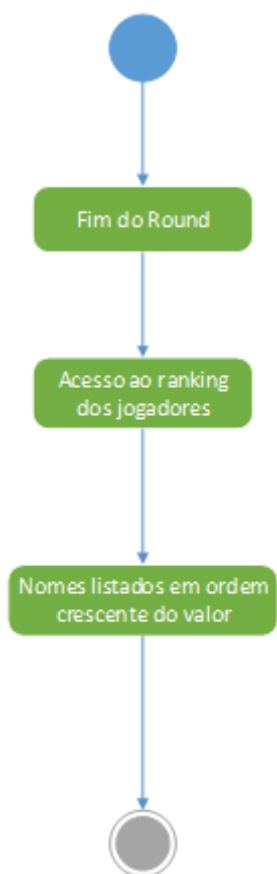


Acessar lobby





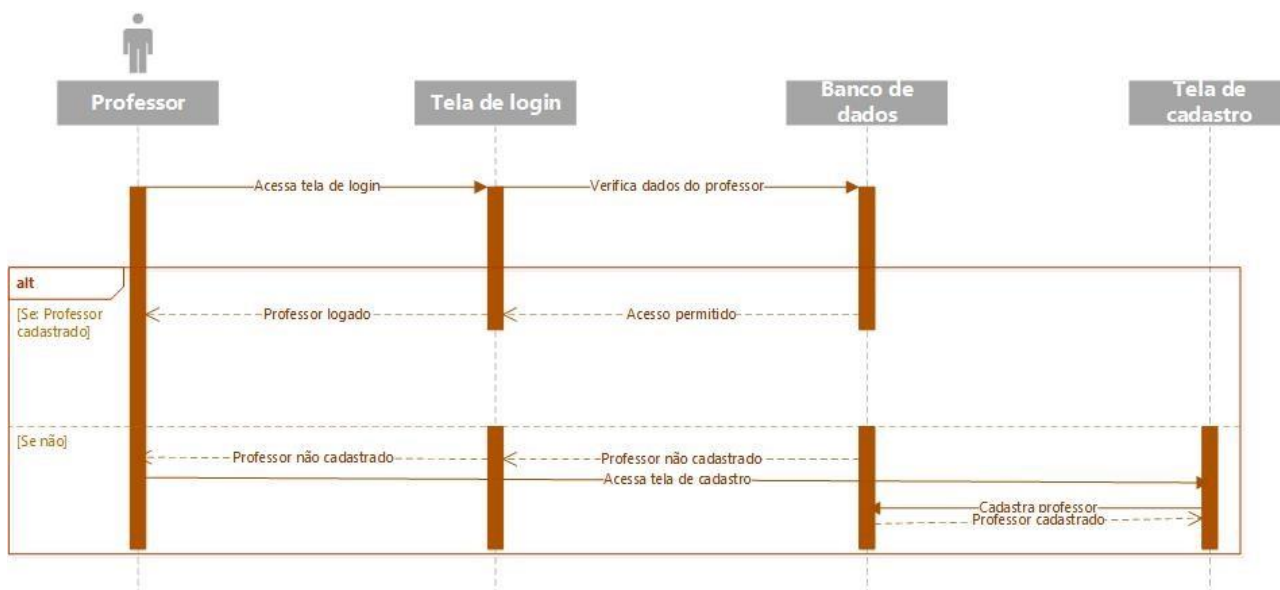
Fazer ranking do fim da partida



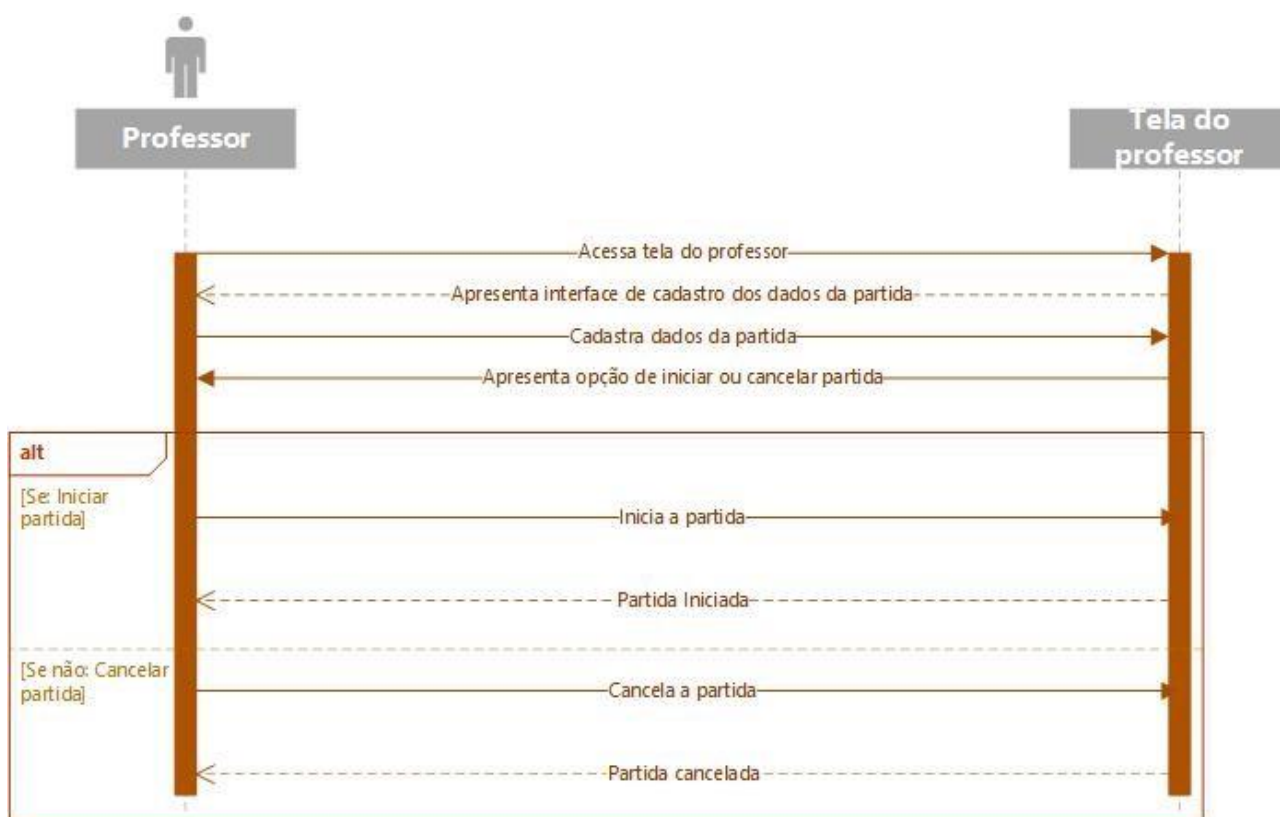


- Visão Implantação:

Efetuar login e cadastro de professor

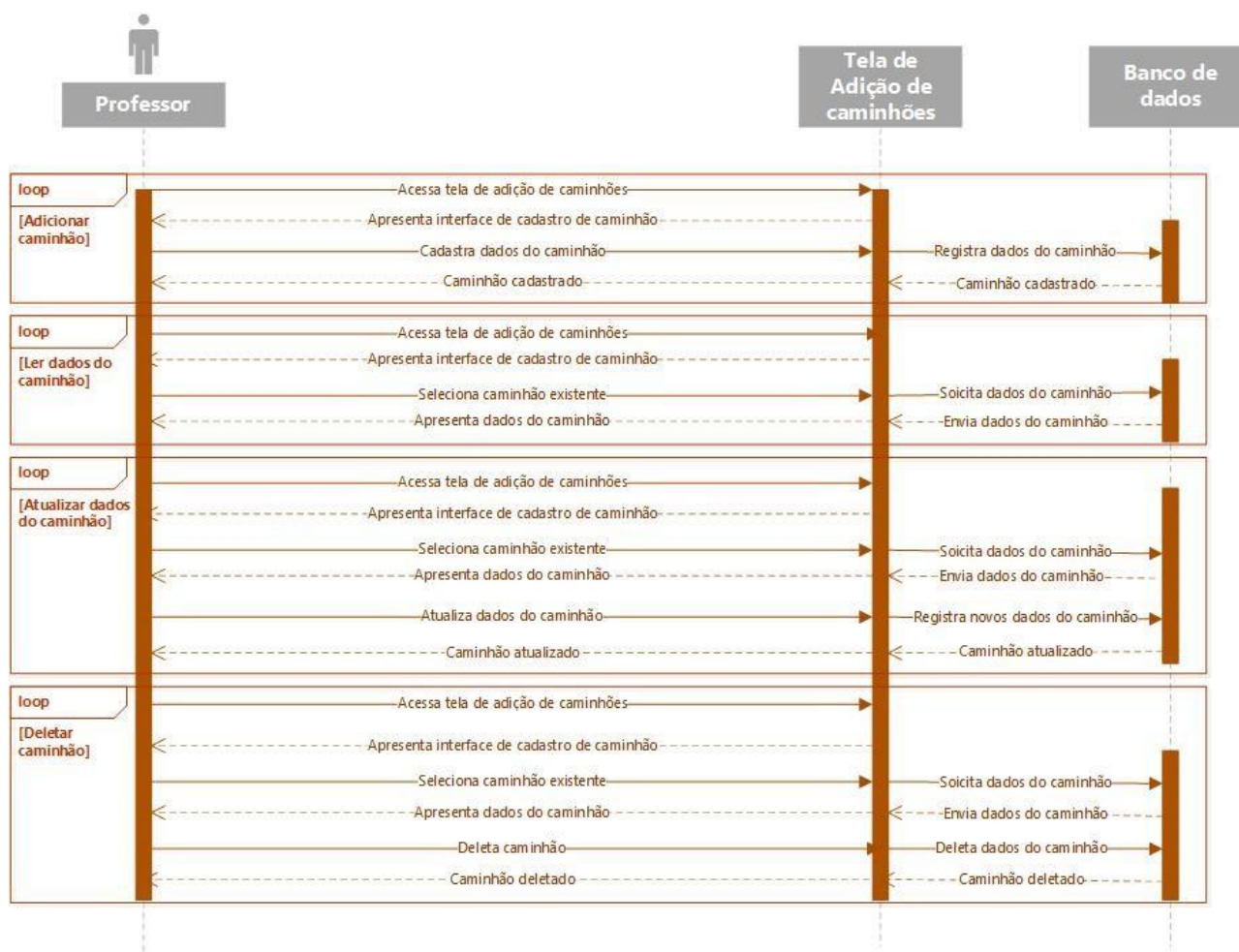


Tela da partida – Visão do professor

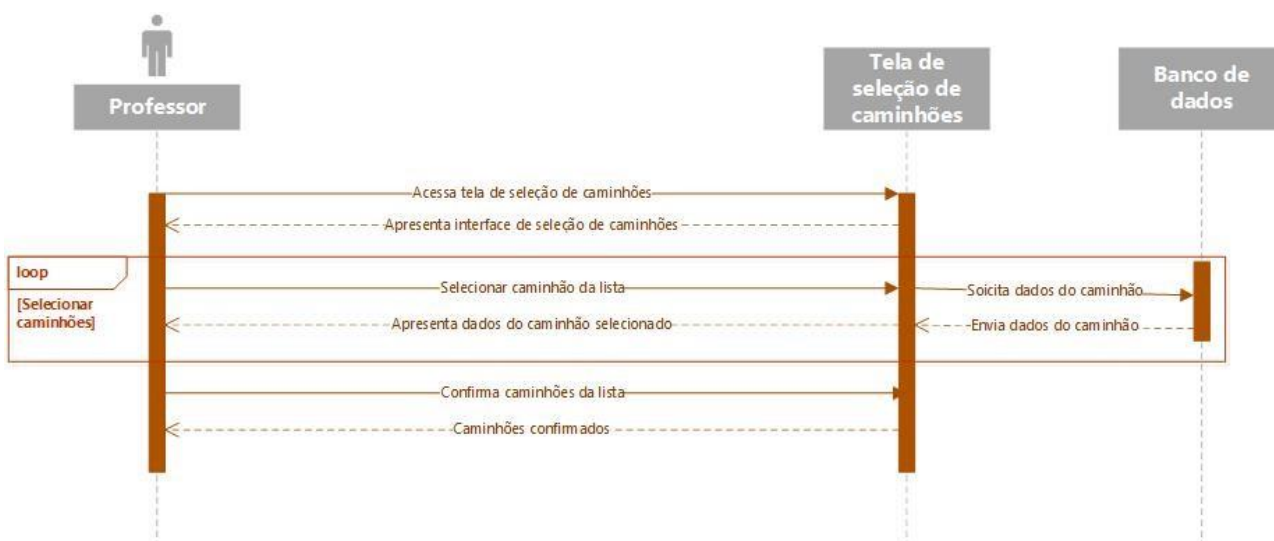




Adicionar, excluir, alterar e visualizar caminhão

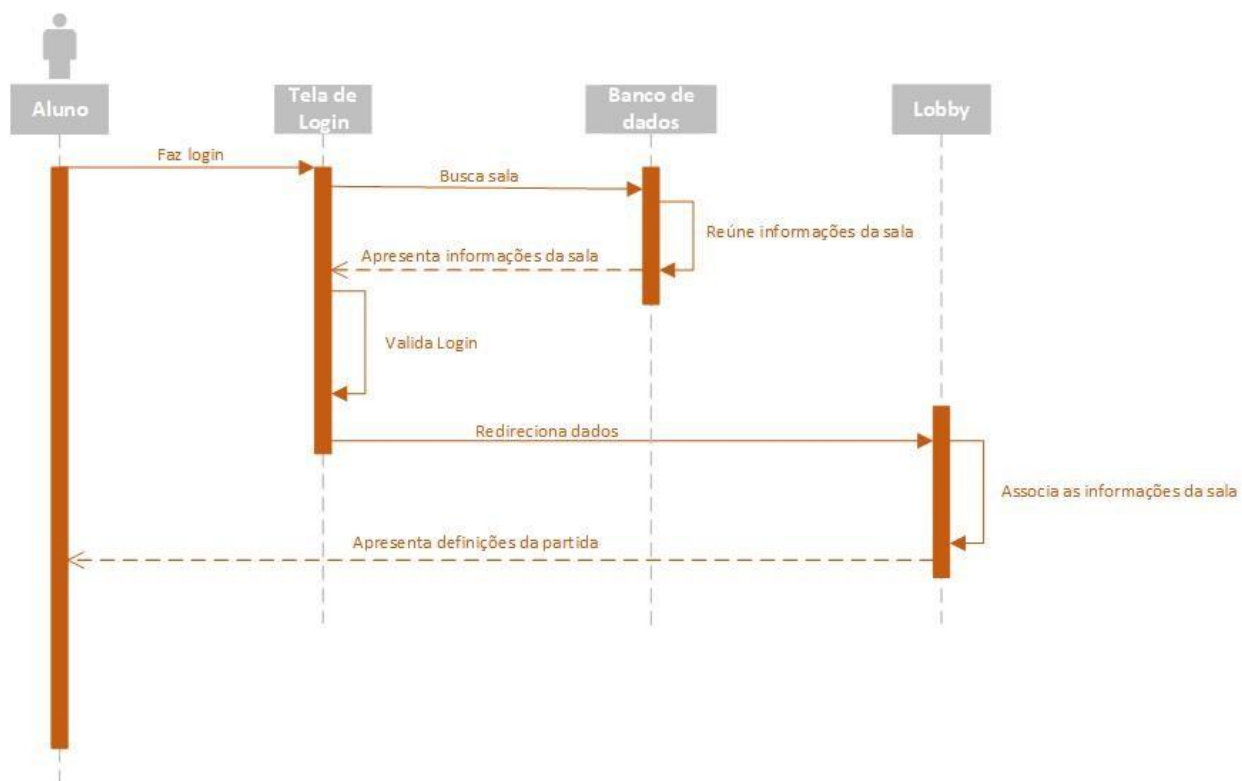


Selecionar caminhões da partida

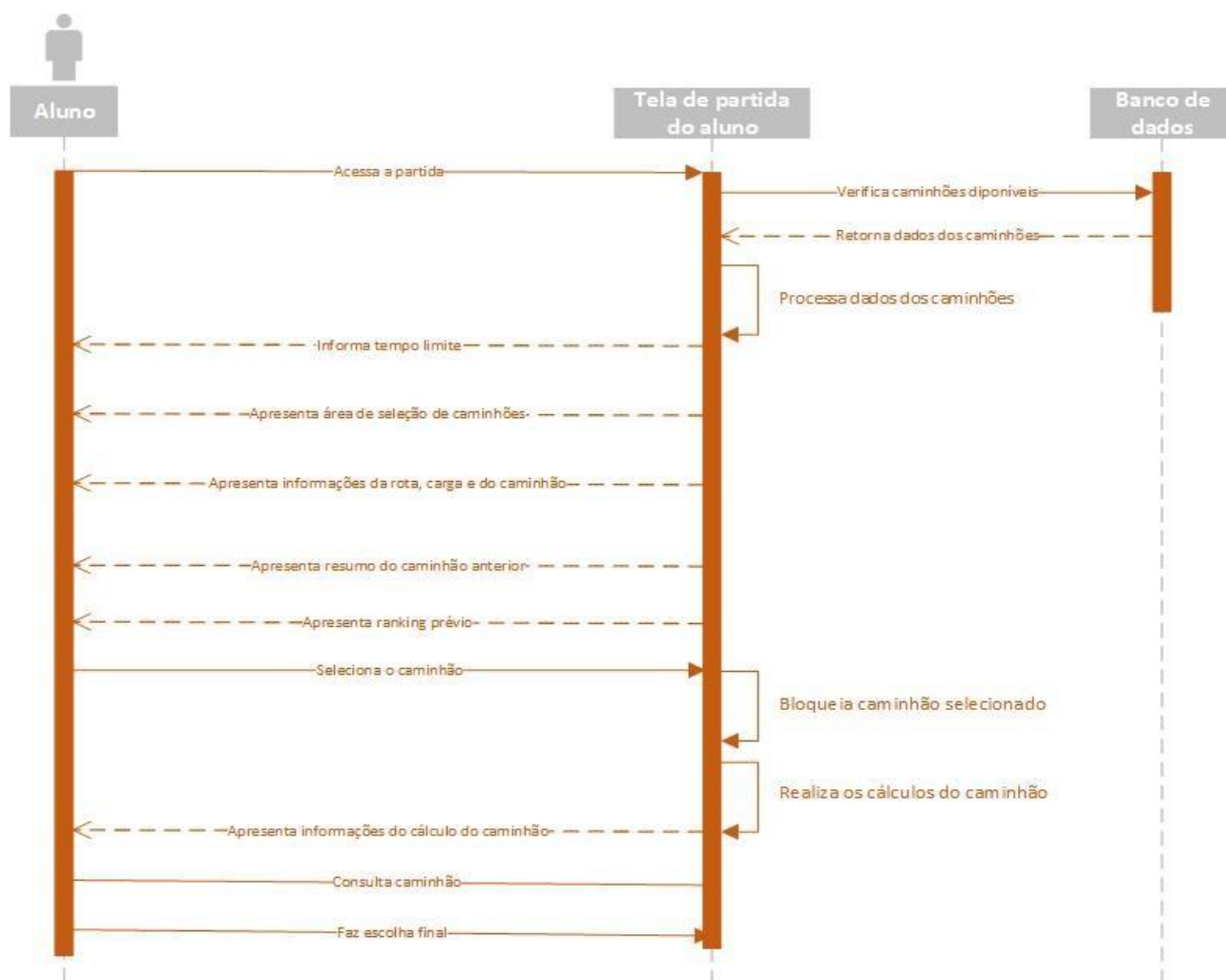




Efetuar login e lobby aluno

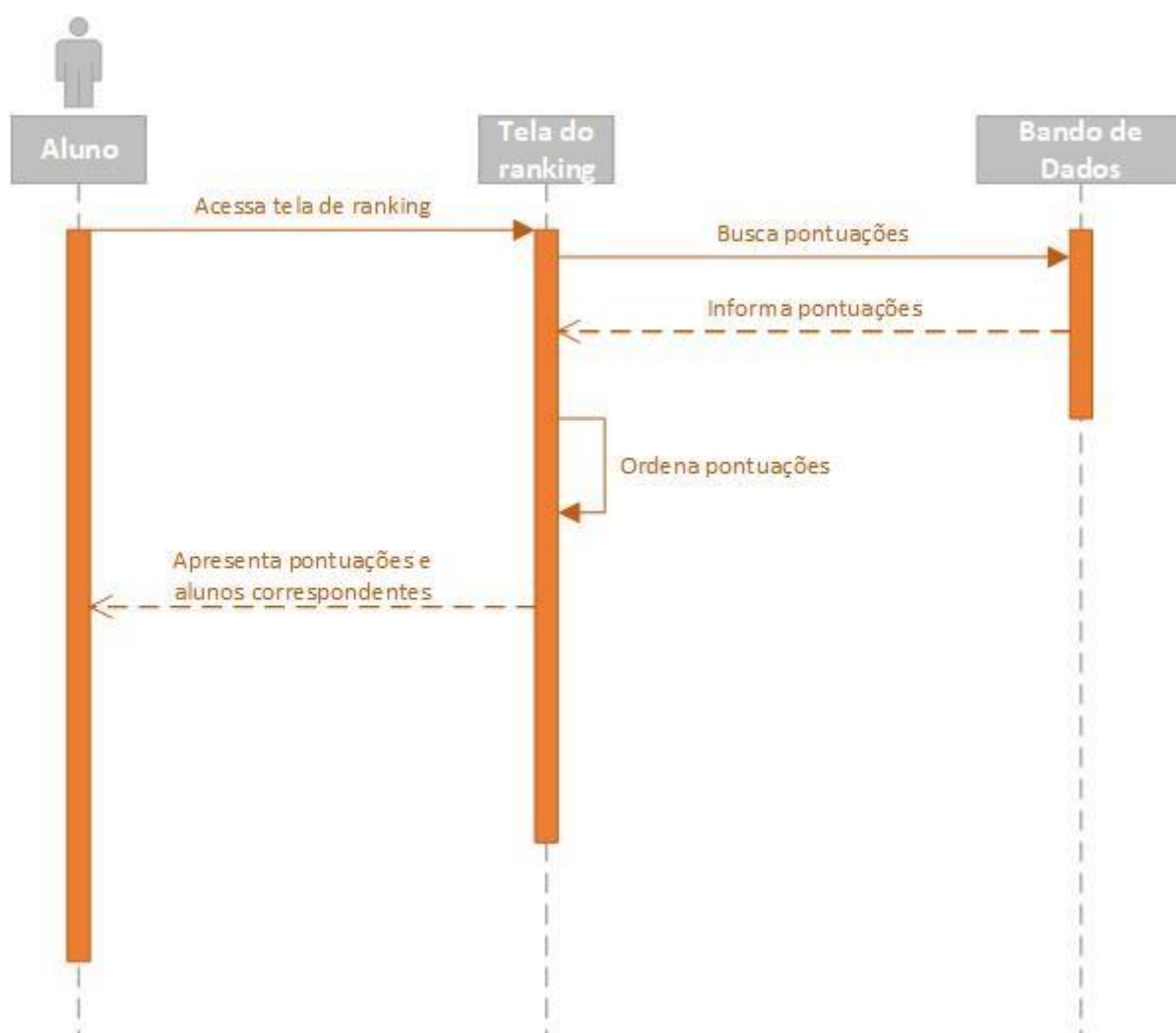


Tela de partida – Visão do aluno





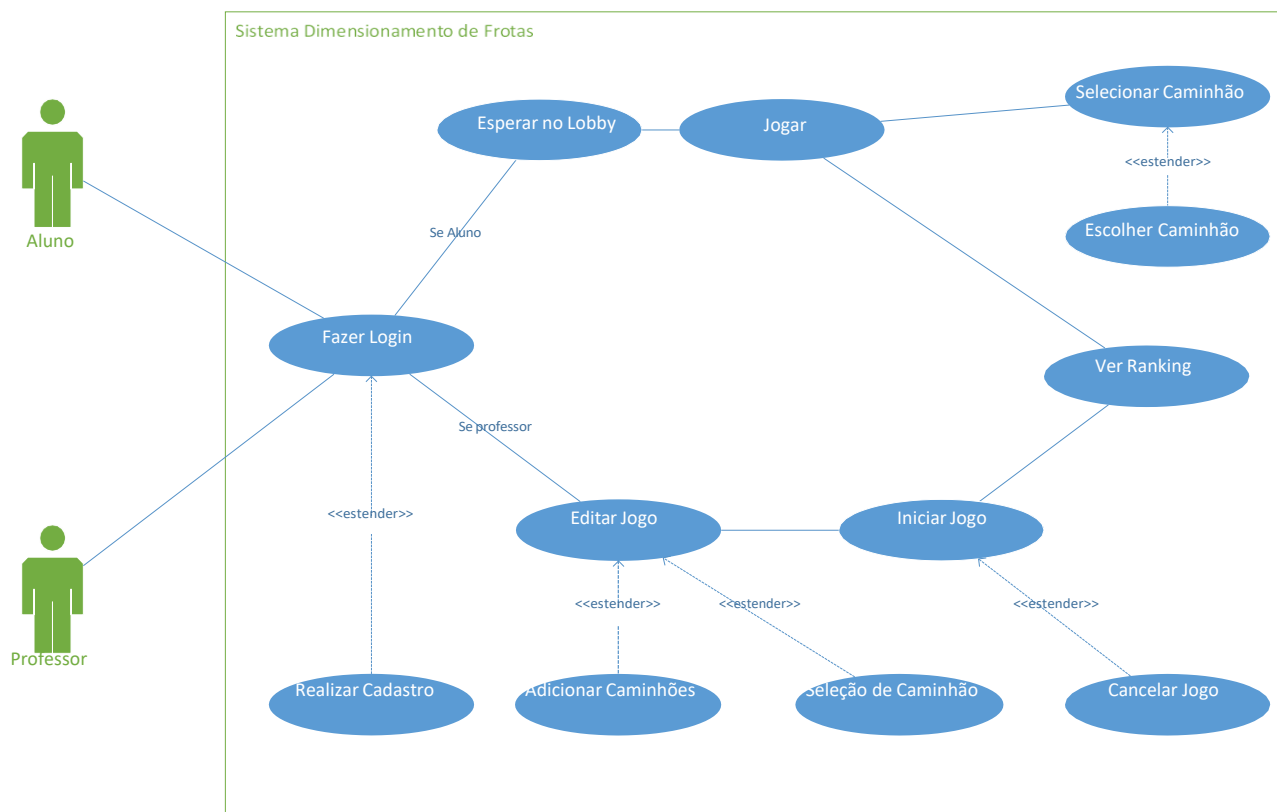
Ranking





- Visão Conceitual:

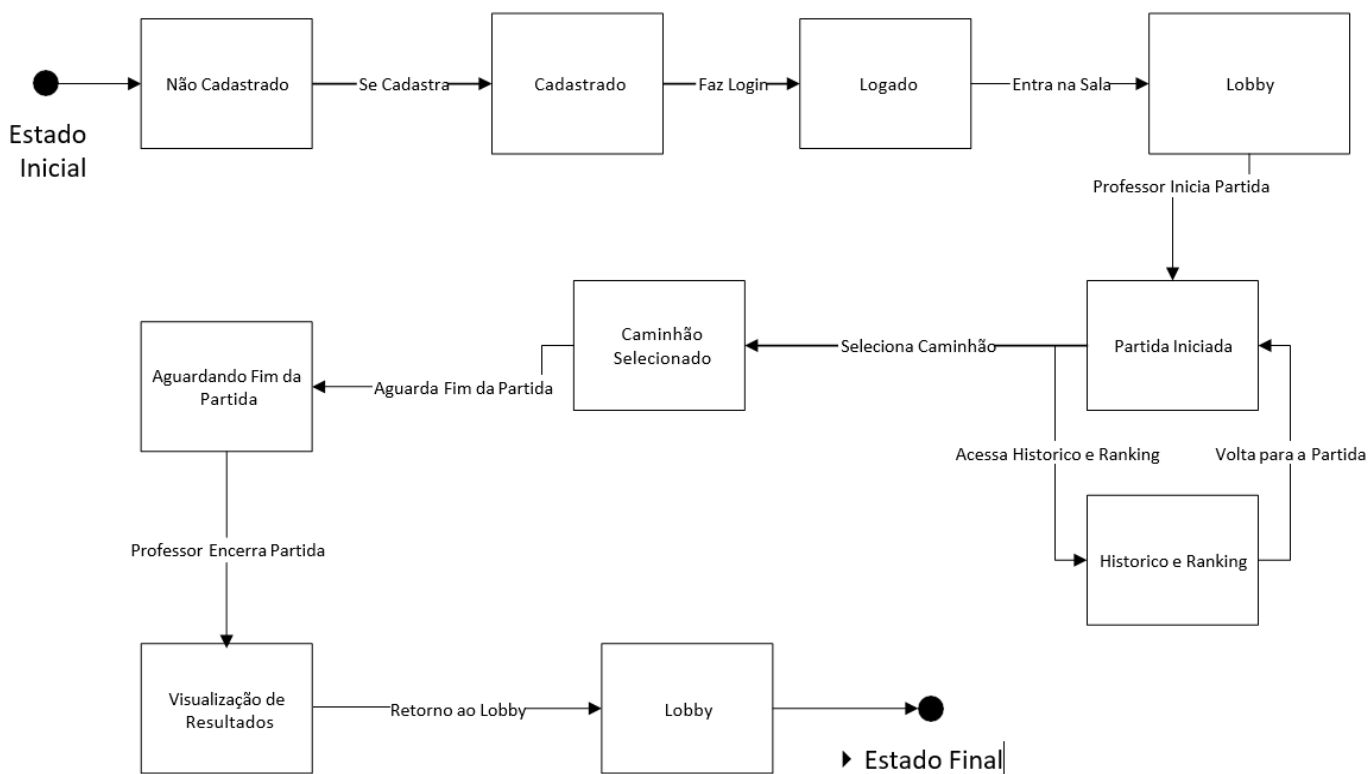
Diagrama de Caso de Uso





- Visão Lógica:

Diagrama de Seleção



[illegible]



3. Glossário de Negócio

Apresenta os termos de negócio considerados significativos para o entendimento do negócio.

Termo	Capacidade da carga em dimensão
Descrição	É o confronto da dimensão da carga em comprimento (m), largura (m) e altura (m) com essas mesmas dimensões do caminhão (capacidade).

Termo	Capacidade da carga em peso
Descrição	É o quociente da divisão da capacidade em (Kg) do caminhão e o peso da carga (kg).

Termo	Custos fixos
Descrição	São Custos que independem de volume produzido. Em transportes, são os custos que ocorrem independente da quilometragem rodada.

Termo	Custos fixos mensais
Descrição	são relativos às despesas que ela tem todos os meses, independentemente da quantidade de produção.

Termo	Custos fixos variáveis
Descrição	são aqueles que não oscilam muito e que não estão atrelados ao volume de vendas ou de produção.

Termo	Custo pernoite
Descrição	Custo de passar a noite num local. Por exemplo em hotel.

Termo	Custos variáveis
Descrição	São todos os custos que acompanham (proporcionalmente) a produção. Em transporte, são custos que impactam no quilômetro rodado. Exemplos: combustível, pneu, câmara (se utilizar), óleo (motor, câmbio, freio, diferencial), filtros (óleo, ar, ar-condicionado, combustível, freio, correias e suspensão).

Termo	Dados da carga (são fornecidos pelo professor)
Descrição	comprimento (m), largura (m), altura (m), peso (kg), quantidade.

Termo	Dados da rota (são fornecidos pelo professor)
Descrição	distância (km), dias.

Termo	Tempo de trava
Descrição	<p>É o tempo que será disponibilizado pelo programa para que os alunos façam a escolha do caminhão selecionado. Caso não seja realizada a escolha do caminhão dentro do tempo estipulado, o caminhão será liberado para escolha dos demais alunos participantes do <i>game</i>.</p> <p>Obs.: Inicialmente, o programa será implementado com o tempo de trava de 10 segundos. Esse tempo poderá ser alterado posteriormente pelo professor.</p>

Termo	
Descrição	



4. Projeto de Interface de Software

<apresenta a criação dos layouts de tela>



5. Planejamento da Qualidade de Software

5.1 Objetivos

O planejamento de testes tem por seu objetivo garantir e controlar a qualidade do software, visando a satisfação do cliente através de um produto compatível, de boa qualidade, dentro dos prazos e das conformidades.

5.2 Pontos fundamentais para a garantia da qualidade do software

Dentro dos pontos fundamentais estão os métodos, que fornecem uma base técnica de como construir o software incluindo um amplo conjunto de tarefas visando a funcionabilidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade do software tudo operando dentro das conformidades. Além das ferramentas e os processos, os quais criam um ambiente de apoio ao desenvolvimento do software e mantém unidas todas as operações.

5.3 Requisitos não-funcionais

5.3.1 Confiabilidade

São essenciais para garantir que o software funcione de maneira estável, esperada e livre de falhas

Número	Descrição	Prioridade	Revisado	
RNFC001	Segurança de acesso: Evitar acesso não autorizado	Alta/Media/Baixa		
RNFC002	Possuir indicação de baixa frequência à falhas.	Alta/Media/Baixa		
RNFC003	Tolerância a falhas, manter nível de desempenho em caso de falha em partes do sistema.	Alta/Media/Baixa		
RNFC004	Recuperabilidade, a capacidade de se restabelecer e restaurar após uma falha e fornecer informações sobre	Alta/Media/Baixa		
RNFC005	Possuir backups regulares e restaurar o software a um estado anterior em caso de falha	Alta/Media/Baixa		
RNFC006	O software deve manter registros detalhados de eventos para que as causas de falhas possam ser rastreadas.	Alta/Media/Baixa		



RNFC007	Manutenções regulares para prevenir falhas, como atualizações de segurança regulares ou manutenção de rotina.	Alta/Media/Baixa		
RNFC008	Manter a validade e integridade dos dados afim de prevenir a corrupção de informações importantes.	Alta/Media/Baixa		
RNFC009	Descreve como o software deve isolar falhas em um componente para evitar que elas afetem o sistema como um todo.	Alta/Media/Baixa		
RNFC010	Testes de Estresse afim de avaliar como ele se comporta sob cargas extremas e identificar possíveis pontos de falha.	Alta/Media/Baixa		
RNFC011	Planos e procedimentos para recuperar o software em caso de desastres naturais ou falhas catastróficas.	Alta/Media/Baixa		
RNFC012	O software deve lidar com a perda de dados de maneira a minimizar o impacto e a perda de informações críticas.	Alta/Media/Baixa		

5.3.2 Usabilidade

São vitais para garantir que o software seja amigável e fácil de usar para seus usuários.

Número	Descrição	Prioridade	Revisado	
RNFU001	Ser inteligível, ou seja, facilidade de entendimento dos conceitos utilizados;	Alta/Media/Baixa		
RNFU002	Ser apreensível, ou seja, facilidade de aprendizado	Alta/Media/Baixa		



RNFU003	Ser operável, ou seja, facilidade de operar e controlar a operação	Alta/Media/Baixa		
RNFU004	O usuário realizar tarefas com um número mínimo de cliques ou etapas, possuir eficiência de navegação	Alta/Media/Baixa		
RNFU005	Fornecer respostas rápidas às ações do usuário.	Alta/Media/Baixa		
RNFU005	A interface do usuário deve ser consistente em toda a aplicação, garantindo que os elementos de design, como botões e menus, sejam apresentados de maneira uniforme.	Alta/Media/Baixa		
RNFU006	O software deve ser acessível a pessoas com deficiências.	Alta/Media/Baixa		
RNFU007	O texto e outros elementos na interface do usuário devem ser fáceis de ler e ver, incluindo tamanhos de fonte adequados e contrastes apropriados.	Alta/Media/Baixa		
RNFU008	O software deve fornecer feedback claro e imediato ao usuário sobre as ações realizadas, como confirmações de envio ou mensagens de erro.	Alta/Media/Baixa		
RNFU009	Deve haver ajuda contextual disponível, bem como documentação do software, como manuais de usuário ou tutoriais.	Alta/Media/Baixa		
RNFU010	O software deve ser projetado de forma que os novos usuários possam aprender a utilizá-lo rapidamente, com um processo de integração suave visando a facilidade de aprendizado.	Alta/Media/Baixa		
RNFU011	A interface do usuário deve ser intuitiva, de modo que as ações e funcionalidades sejam compreendidas sem a necessidade de treinamento extenso	Alta/Media/Baixa		
RNFU012	O software deve ser capaz de lembrar o estado anterior do usuário, como as páginas visitadas ou as configurações, para que os usuários possam retomar onde pararam.	Alta/Media/Baixa		



RNFU013	O software deve incluir mecanismos para coletar feedback dos usuários e usar esses comentários para melhorar a usabilidade.	Alta/Media/Baixa		
RNFU014	Garante que o software seja utilizável em vários tipos de sistemas operacionais	Alta/Media/Baixa		

5.4 Processo de Verificação e Validação

O produto a ser desenvolvido foi verificado através de revisões de códigos, nos quais os desenvolvedores atuam verificando o código uns dos outros a fim de garantir consistência de que boas práticas e padrões estejam estabelecidos. Além de testes unitários, que atuam para verificar pequenas partes individuais do código (unidades) para garantir que funcionem corretamente isoladamente.

Quanto a validação, teste de integração atuam para validar as partes unitárias estão em consonância com outras partes, testes funcionais garantem que o software esteja de acordo com os requisitos funcionais estabelecidos, como acerto nos cálculos de carga, acesso ao login tanto de aluno quanto professor, em geral que o software se comporte como esperado.

5.5 Gerenciamento da Garantia da Qualidade de Software:

O projeto fora guiado através de auditorias, revisões e melhorias contínuas. Reuniões foram agendadas para discutir o andamento do projeto, enumeração dos requisitos (tais como confiabilidade, confidencialidade, conformidade, durabilidade, manutenção, estética, percepção, correção, eficiência, entre outros) e diálogos com os desenvolvedores a respeito do andamento do artefato, que por sua vez forneceram respaldo sobre a continuidade do projeto. Revisões a partir dessas auditorias, serviram para encaminhar o rumo a ser seguido a partir das auditorias, principalmente assegurando que o desenvolvimento esteja dentro dos conformes estabelecidos nos resultados das auditorias e no plano de desenvolvimento de qualidade.

5.6 Equipes de revisões e auditorias (garantir a qualidade do processo)

A equipe de revisões e auditoria é realizada por engenheiros de software para engenheiros de software, com a finalidade de transparecer as inconformidades. As auditorias são as revisões realizadas pela equipe da garantia de qualidade de software(SQA), com o propósito de assegurar que as diretrizes de qualidade estejam sendo seguidas no trabalho de engenharia de software. Por exemplo, uma auditoria do processo de revisão pode ser feita para assegurar que as revisões estejam sendo exercida de forma a encaminhar a maior probabilidade de descoberta de erros.

A qualidade do processo de desenvolvimento de software é garantida pelas equipes de revisões e auditorias que são responsáveis por avaliar e verificar a conformidade com os padrões, processos e práticas estabelecidas.

Geralmente as equipes de revisões e auditorias são compostas por indivíduos com conhecimento e experiência em **desenvolvimento de software, garantia de qualidade e auditoria, podendo incluir: engenheiros de software, Analistas de garantia de qualidade, auditores internos ou externos e gerentes de projetos.**

Algumas atividades desempenhadas pelas equipes de revisões e auditorias são:

Revisões: Revisar artefatos de software, como requisitos, projetos, código e documentação, para identificar erros, inconsistências e áreas de melhorias, com o objetivo de descobrir erros na função lógica, ou implementação, verificar se o software atende aos requisitos, garantir que o software foi



representado de acordo com os padrões pré-definidos, obter software desenvolvido de maneira uniforme. As revisões são realizadas como uma reunião e será bem-sucedida se planejada apropriadamente, controlada e se todos os envolvidos participarem.

Auditoria: O processo de desenvolvimento de software deve ser avaliado com um todo, englobando políticas, procedimentos, ferramentas e práticas, para garantir a conformidade com os padrões e regulamentos.

Relatórios e Recomendações: As descobertas das revisões e auditorias devem ser documentadas e as melhorias devem ser recomendadas.

Monitoramento e Acompanhamento: As ações corretivas e preventivas implementadas com base nas recomendações devem ser monitoradas.

Os Benefícios das equipes de revisões e auditorias são:

Melhoria da Qualidade: Erros e inconsistências são identificados e corrigidos de maneira precoce, impedindo problemas posteriores.

Conformidade com Padrões: Assegura que o processo de desenvolvimento do software atenda aos padrões e regulamentos aplicáveis.

Otimização de Processos: Constatam áreas de melhorias no processo, indicando mudanças para tornar mais eficiente e eficaz.

Prevenção de Riscos: Diminuem riscos identificando e deduzindo potenciais problemas antes que eles se tornem críticos.

Confiança do Cliente: Prover confiança aos clientes e partes interessadas de que o software está sendo desenvolvido de acordo com as melhores práticas.

5.7. Principais Documentações a Serem Empregadas

Plano de garantia da Qualidade de Software: Define o escopo, objetivos, responsabilidades, e atividades do cronograma de garantia de qualidade do software.

Plano de teste: Descreve os tipos de testes a serem executados, estratégias de teste, casos de teste e programa de execução.

Casos de teste: Documentam os testes específicos a serem executados, incorporando os requisitos testados, os dados de entrada e os resultados esperados.

Relatórios de defeitos: Descrevem defeitos ou problemas identificados durante os testes, incluindo sua descrição, gravidade, prioridade e status.

Relatório de testes: Resenha dos resultados dos testes, integrando o número de testes executados, defeitos encontrados e o status geral da qualidade de software.

Procedimento de teste: Prove instruções passo a passo sobre como executar testes específicos, incluindo as ferramentas e técnicas a serem usadas.

Padrões de codificações: Designa as convenções e práticas de codificação a serem seguidas pelos desenvolvedores para garantir a consistência e a qualidade de código.

Processo do gerenciamento de Mudanças: Discorre o processo para gerenciar e controlar as alterações no software e na documentação.

Processos de gerenciamento de riscos: Detectam e avaliam os riscos potenciais para a qualidade do software e definem estratégias para mitigá-los.

Planos de melhorias da qualidade: Fornece um roteiro para melhorar continuamente a qualidade do software e os processos de desenvolvimento.

Benefícios do uso da documentação: Clareza e consistência, garante que todos os envolvidos possuam uma compreensão clara dos requisitos, processos e procedimentos.

Rastreabilidade: A documentação permite rastrear defeitos, alterações, e melhorias ao longo do ciclo de vida do software.

Reutilização: A documentação pode ser usada novamente em projetos futuros, economizando tempo e esforço.

Conformidade: A documentação ajuda a atender aos padrões e regulamentos da indústria.

Melhoria Contínua: A documentação fornece uma base para identificar áreas de melhoria e implementar mudanças para melhorar a qualidade do software.



5.8 Referencias a Ferramentas, Técnicas e Metodologias

Ferramentas: Utilizamos o Visual Studio para realizar todas as versões do software.

Técnicas: Aplicamos tanto a técnica de teste de caixa preta quanto a técnica de teste de caixa branca, para garantir que todas as funcionalidades estavam funcionando conforme o esperado.

Metodologias: Seguimos a metodologia de Desenvolvimento Incremental, uma abordagem de desenvolvimento onde o produto é projetado, implementado e testado incrementalmente até que esteja completo.

5.9 Política de Gerenciamento de Riscos

5.9.1 Introdução e Objetivo

A presente política tem como objetivo estabelecer um processo onde ocorra uma colaboração entre desenvolvedores e testadores, com o objetivo de desenvolver o software solicitado sem maiores problemas e eliminando dificuldades.

5.9.2 Papéis e Responsabilidades

- **Testadores:** Planejar e executar testes para garantir que o software funcione corretamente.
- **Desenvolvedores:** Escrever e manter o código do software, além de corrigir bugs.
- **Analistas:** Coletar e documentar os requisitos do projeto e analisar o sistema e sugerir melhorias.
- **Documentadores:** Criar e manter a documentação do projeto, incluindo manuais de usuário e guias técnicos.
- **Banco de Dados:** Gerenciar e Garantir a integridade do banco de dados

5.9.3 Processo de Gerenciamento de Riscos

- **Identificação de Riscos:** Utilizando testes constantes no software.
- **Avaliação de Riscos:** Classificar riscos quanto à probabilidade e impacto.
- **Mitigação de Riscos:** Desenvolver estratégias para antecipar os riscos.
- **Monitoramento e Revisão:** Monitoramento constante com revisões e melhorias.

5.9.4 Metodologias e Ferramentas

- **Desenvolvimento Incremental**
- **Visual Studio**

5.9.5 Comunicação e Relatórios

Comunicação imediata de riscos significativos à equipe de gerenciamento de riscos.



5.10 Referências a padrões, práticas, convenções e métricas:

O projeto foi desenvolvido em base alguns padrões para que se garantam o atendimento dos requisitos de qualidade estabelecidos, tais como a formatação das variáveis no código, ou seja, uso de um padrão estabelecido para nomenclatura, modelo padronizado das estruturas condicionais, assim como também a organização das classes do projeto. Práticas como reuniões a cada duas semanas, organização dos versionamentos na hora de subir o projeto e o feedback dos desenvolvedores de suas alterações ajudaram a manter e estabelecer um padrão que se manteve aos fins dos projetos. Foi de intensa utilização, comentários no código-fonte, convenção no qual auxiliava o entendimento claro e informal a respeito das alterações, funcionamento ou possíveis erros. Para quantificar o desenvolvimento, métricas como conformidade com padrões definidos serviam para determinar o quanto do código está nos conformes, tempo médio de resposta retornava uma avaliação de desempenho e a satisfação do usuário um feedback claro de como o jogo seria percebido e como poderia ser melhorado.

5.11 Detalhes do Ciclo do Processo de Verificação

Ciclo foi feito baseado na ideia utilizada na metodologia, ou seja, a cada nova versão disponível do software ocorria um teste geral da estrutura para verificar se não ocorreu nenhuma quebra ou inconsistências.

5.12 Principais Atividades de Verificação

Em construção

5.13 Definição de Papeis e Responsabilidades da Equipe de Verificação

Em construção

5.14 Principais Documentos a Serem Gerados

- **Especificação de Requisitos do Cliente:** Documento que descreve detalhadamente as necessidades do cliente em termos de capacidade de transporte, exigências de tamanho dos caminhões (comprimento, largura, altura), e outras especificações técnicas.
- **Especificação Técnica dos Caminhões:** Detalhamento das características físicas dos caminhões disponíveis para a frota, incluindo dimensões (comprimento, largura, altura), capacidade de carga, tipo de motorização, entre outros.
- **Plano de Operação e Logística:** Documento que descreve como será realizada a operação de transporte, incluindo alocação de caminhões, rotas, horários, e estimativas de custos operacionais.
- **Plano de Custos:** Detalhamento dos custos associados à operação dos caminhões, incluindo custos fixos (depreciação, seguro, licenciamento) e variáveis (combustível, manutenção, pedágios).
- **Relatório de Competição:** Documento que apresenta os resultados da competição entre os alunos, incluindo análise das escolhas feitas, desempenho das frotas em termos de eficiência e custos, e lições aprendidas.



5.15 Referência à Técnicas, Métodos e Ferramentas Empregados

- **Análise de Viabilidade Econômica:** Utilização de métodos como Valor Presente Líquido (VPL) e Taxa Interna de Retorno (TIR) para avaliar a viabilidade financeira das decisões de investimento em caminhões.
- **Análise de Custo-Benefício:** Comparação entre os custos operacionais previstos e os benefícios esperados, como tempo de entrega reduzido, para determinar a eficiência das escolhas de frota.
- **Simulação de Monte Carlo:** Para avaliar a incerteza e riscos associados aos custos operacionais, permitindo uma visão mais realista dos resultados esperados.
- **Modelagem e Otimização de Rotas:** Uso de softwares de otimização de rotas para minimizar custos operacionais e tempo de transporte.

5.16 Referência à Padrões, Políticas e Formatos Adotados

- **Padrões de Segurança e Manutenção:** Cumprimento das normas de segurança veicular e regulamentações de manutenção preventiva e corretiva dos caminhões.
- **Políticas de Sustentabilidade:** Inclusão de critérios ambientais na escolha da frota, como eficiência de combustível e emissões de CO2.
- **Formatos de Relatórios Padronizados:** Uso de formatos específicos para relatórios de custos, especificações técnicas e análises de viabilidade, garantindo clareza e comparabilidade das informações.
- **Políticas de Compliance:** Adoção de práticas que estejam em conformidade com leis e regulamentações locais relacionadas ao transporte rodoviário.

5.17 Relatórios a Serem Produzidos

No plano mestre de validação, vários relatórios são produzidos para documentar o processo de verificação de software, a seguir será documentado as descrições dos relatórios que serão produzidas durante o processo de verificação:

5.17.1 Relatório de Verificação:

- **Descrição:** Detalha os processos de verificação a serem utilizadas ao longo do projeto.
- **Objetivo:** Garantir que todas as etapas do desenvolvimento sejam monitoradas e que os requisitos de qualidade sejam atendidos.

5.17.2 Relatório de Progresso:

- **Descrição:** Apresenta o andamento das atividades de verificação, mostrando as tarefas concluídas, e as em andamento.
- **Objetivo:** Monitorar o cumprimento do cronograma e identificar problemas que possam impactar na entrega do projeto.

5.17.3 Relatório de Testes:

- **Descrição:** Mostra os resultados obtidos nos testes realizados.
- **Objetivo:** Avaliar a conformidade do software com os requisitos especificados e identificar falhas a serem corrigidas.



5.17.4 Relatório de Análise de Riscos:

- **Descrição:** Identifica e avalia os riscos do projeto.
- **Objetivo:** Reduzir a probabilidade de ocorrência de problemas e minimizar seu impacto caso ocorram.

5.17.5 Relatório de Análise de Defeitos:

- **Descrição:** Lista e categoriza os defeitos encontrados durante a verificação, detalhando a gravidade e status.
- **Objetivo:** Acompanhar problemas e garantir que todos os defeitos sejam tratados.

5.17.6 Relatório de Validação:

- **Descrição:** Confirma que o software desenvolvido cumpra suas funções no ambiente real.
- **Objetivo:** Assegurar que o produto final funcione corretamente, mas também satisfaça os usuários.

5.18 Cronograma das Etapas da Verificação

O cronograma das etapas, inclui as fases de verificação que vão do planejamento até a execução e revisão.

Fase	Duração
Planejamento e Verificação	1 dia
Verificação do ambiente de testes	1-2 dias
Testes de Sistema	1 dia
Testes de Aplicação do Software	1-3 dias
Correção de Defeitos	1-5 dias
Revisão e Análise de Resultados	3 dias
Testes Finais	1-2 dias
Verificação Final e Documentação	7 dias

Planejamento e Verificação:

- **Descrição:** Esta fase envolve a definição das estratégias e metodologias de verificação, identificação dos recursos necessários, e elaboração do cronograma detalhado.
- **Duração:** 1 dia
- **Responsáveis:** Gerente de Projeto, Analista de Qualidade
- **Entregáveis:** Plano de Verificação

Verificação do Ambiente de Testes:

- **Descrição:** Configuração do ambiente de testes, incluindo a instalação de ferramentas, preparação de dados de teste e configuração de hardware/software necessário.
- **Duração:** 1-2 dias
- **Responsáveis:** Equipe de Testes
- **Entregáveis:** Ambiente de Testes Configurado



Testes de Sistema:

- **Descrição:** Realização de testes de sistema para verificar o comportamento do software como um todo, assegurando que todos os módulos e componentes interagem corretamente.
- **Duração:** 1 dia
- **Responsáveis:** Equipe de Testes
- **Entregáveis:** Relatórios de Teste de Sistema

Testes de Aplicação do Software

- **Descrição:** Execução de testes de aplicação para validar se o software atende aos requisitos funcionais e não funcionais especificados. Isso inclui testes de funcionalidade, usabilidade, desempenho e segurança.
- **Duração:** 1-3 dias
- **Responsáveis:** Equipe de Testes, Usuários Finais
- **Entregáveis:** Relatórios de Teste de Aplicação

Correção de Defeitos

- **Descrição:** Correção dos defeitos identificados durante as fases de teste. Esta etapa pode variar em duração dependendo da quantidade e gravidade dos defeitos encontrados.
- **Duração:** 1-5 dias
- **Responsáveis:** Desenvolvedores
- **Entregáveis:** Software Corrigido

Revisão e Análise de Resultados

- **Descrição:** Análise dos resultados de todos os testes realizados e identificação de defeitos críticos e não críticos. Avaliação da conformidade do software com os requisitos especificados.
- **Duração:** 3 dias
- **Responsáveis:** Analista de Qualidade, Gerente de Projeto
- **Entregáveis:** Relatório de Análise de Resultados

Testes Finais

- **Descrição:** Realização de testes finais para verificar se todas as correções foram implementadas corretamente e confirmar que o software está pronto para entrega.
- **Duração:** 1-2 dias
- **Responsáveis:** Equipe de Testes
- **Entregáveis:** Relatório de Testes Finais

Verificação Final e Documentação

- **Descrição:** Execução de uma verificação final para garantir que todas as atividades de verificação foram completadas corretamente e preparação da documentação final. Isso inclui a compilação de todos os relatórios e a revisão final do software.
- **Duração:** 7 dias
- **Responsáveis:** Equipe de Testes, Analista de Qualidade
- **Entregáveis:** Relatório Final de Verificação, Documentação do Projeto



5.19 Riscos e Contingências

O gerenciamento de riscos é essencial para a verificação de software, ele mitiga o impacto dos problemas e garante uma retomada segura ao trabalho, logo então, segue de exemplo algumas etapas que podem ser seguidas no projeto:

Riscos de Recursos

- **Descrição:** Falta de recursos
- **Mitigação:**
 - Planejamento antecipado de recursos.
 - Garantir que todos estejam aptos a realizar suas tarefas.
- **Contingência:**
 - Redistribuição de tarefas dentro da equipe.
 - Priorizar atividades importantes.

Riscos de Cronograma

- **Descrição:** Atrasos no cronograma de verificação devido a imprevistos.
- **Mitigação:**
 - Estabelecimento de prazos seguros.
 - Comunicação transparente, garantindo que tudo esteja nos conformes.
- **Contingência:**
 - Extensão do prazo.
 - Redução do escopo.
 - Trabalho adicional para cumprir o prazo.

Riscos de Qualidade de Software

- **Descrição:** Defeitos críticos no software que impactam a qualidade final.
- **Mitigação:**
 - Execução de testes para identificar problemas mais cedo.
 - Revisão de código constantes.
 - Uso de metodologias para correção e teste de forma efetiva.
- **Contingência:**
 - Priorização dos defeitos críticos para correção imediata.
 - Planejamento de releases incrementais para lidar com defeitos menores posteriormente.
 - Mobilização de uma equipe de resposta rápida para resolver problemas críticos.



Riscos Técnicos

- **Descrição:** Problemas técnicos no geral.
- **Mitigação:**

Plataformas para desenvolvimento seguras.

Manutenção regular e atualizações de software.

Equipe com conhecimento adequado em cada área.

- **Contingência:**

Troca de ferramentas ou reconfigurar o ambiente.

A manutenção preventiva.

Treinamento extra para habilitar a equipe, caso a função atribuída seja displicente.

5.20 Detalhamento das Atividades de Validação

Tem por objetivo garantir que o software seja validado de acordo com os requisitos funcionais, de desempenho, usabilidade, segurança, compatibilidade e aceitação pelo cliente antes do lançamento.

Atividades de Validação:

1. Testes Funcionais:

- Verificar funcionalidades principais.
- Testes de integração e regressão.

2. Testes de Desempenho:

- Avaliação de performance sob diferentes cargas.
- Testes de estresse.

3. Testes de Usabilidade:

- Avaliar a interface do usuário.
- Testes de acessibilidade.

4. Testes de Segurança:

- Identificação e correção de vulnerabilidades.
- Conformidade com políticas de segurança.

5. Testes de Compatibilidade:

- Verificação em diferentes ambientes.
- Testes de compatibilidade com dispositivos.

6. Testes de Aceitação pelo Cliente:

- Demonstração e validação com cliente.



5.21 Definição dos Grupos de Validação

Tem por objetivo identificar os grupos envolvidos no processo de validação do software.

Grupos de Validação:

1. Equipe Interna de QA (Quality Assurance):

- Responsável por testes técnicos e relatórios.

2. Usuários Beta:

- Selecionados para testes em ambiente real.

3. Cliente:

- Participação em revisões e validações finais.

5.22 Area do Software Verificadas

Tem por objetivo especificar as áreas do software que serão alvo de verificação durante o processo de validação.

Áreas de Verificação:

1. Funcionalidades Principais:

- Verificação detalhada de todas as funcionalidades conforme requisitos.

2. Interfaces de Usuário:

- Avaliação da usabilidade e experiência do usuário.

3. Integração com Sistemas Externos:

- Testes de integração conforme especificações.

4. Segurança:

- Avaliação de vulnerabilidades e conformidade com políticas.

5. Desempenho:

- Testes para garantir performance adequada.

6. Documentação:

- Revisão da documentação do usuário e técnica.

7. Compatibilidade:

- Verificação em diferentes ambientes.



5.23 Lista de Componentes a Serem Criados

5.23.1 Casos de Teste

- Casos de Teste Funcionais: Verificam se o software realiza as funções esperadas conforme os requisitos especificados.
- Casos de Teste Não Funcionais: Avaliam aspectos como desempenho, segurança, usabilidade e compatibilidade.
- Casos de Teste de Regressão: Garantem que alterações no software não introduzam novos defeitos em partes previamente funcionais.

5.23.2 Planos de Teste

- Plano de Teste de Unidade: Define os testes para verificar a funcionalidade individual de componentes ou módulos do software.
- Plano de Teste de Integração: Verifica a interação entre diferentes módulos do software.
- Plano de Teste de Sistema: Avalia o sistema como um todo, verificando se ele atende aos requisitos funcionais e não funcionais.
- Plano de Teste de Aceitação: Conduzido para assegurar que o software atende às necessidades e expectativas do cliente ou usuário final.

5.23.3 Ambiente de Testes

- Ambiente de Teste de Desenvolvimento: Utilizado pelos desenvolvedores para testes iniciais.
- Ambiente de Teste de QA (Quality Assurance): Ambientes controlados para execução de testes formais.
- Ambiente de Teste de Produção: Espelho do ambiente de produção real, usado para testes finais antes da liberação.

5.24 Lista de Documentos Produzidos

5.24.1 Documentos de Planejamento

- Plano Mestre de Validação: Documento principal que descreve o processo de validação, escopo, objetivos, metodologia e cronograma.
- Plano de Garantia da Qualidade de Software (PGQS):** Descreve os procedimentos e padrões de qualidade a serem seguidos.

5.24.2 Documentos de Execução

- Especificações de Caso de Teste: Descrição detalhada dos casos de teste, incluindo passos, dados de entrada, resultados esperados e critérios de sucesso.
- Relatórios de Defeitos: Documentação de quaisquer defeitos encontrados durante o processo de teste, incluindo detalhes do defeito, impacto e status da resolução.

5.24.3 Documentos de Resultados

- Relatórios de Execução de Teste: Resumo dos testes realizados, incluindo resultados e análises.
- Relatório de Validação: Documento final que resume todo o processo de validação, incluindo a conformidade com os requisitos e a recomendação para a liberação do software.



5.25 Técnicas, Ferramentas e Padrões a Serem Empregados

5.25.1 Técnicas de Teste

- Testes Manuais: Realizados por testadores humanos seguindo scripts de teste.
- Testes Automatizados: Utilização de ferramentas para a execução automática de casos de teste, especialmente úteis para testes de regressão.
- Testes de Caixa Branca: Focam na estrutura interna do software.
- Testes de Caixa Preta: Focam nas entradas e saídas do software sem considerar a estrutura interna.

5.25.2 Ferramentas de Teste

- Ferramentas de Gestão de Testes: Como TestRail e Zephyr, para planejamento, execução e rastreamento de testes.
- Ferramentas de Automação de Testes: Como Selenium, JUnit, e TestNG, para automação de testes.
- Ferramentas de Monitoramento e Relatório: Como Jenkins e SonarQube, para integração contínua e análise de qualidade de código.

5.25.3 Padrões de Qualidade

- ISO/IEC 9126: Padrão internacional para a avaliação da qualidade do software.
- IEEE 829: Padrão para documentação de testes de software.
- CMMI (Capability Maturity Model Integration): Modelo de melhoria de processos para o desenvolvimento de software.



5.26 Cronograma Detalhado

SEMANA	PERÍODO	ATIVIDADES	RESPONSÁVEIS
1	01/04/2024 - 05/04/2024	Reunião inicial – Revisão da documentação de Requisitos; Criação do Plano de teste	Equipe de Teste, Desenvolvedores
2	08/04/2024 – 12/04/2024	Revisão Detalhada dos requisitos funcionais e não funcionais.	Equipe de Teste
3	15/04/2024 - 19/04/2024	Revisão do desempenho e usabilidade	Equipe de Teste e Desenvolvedores
4 e 5	22/04/2024 – 03/05/2024	Execução dos casos de teste funcionais	Equipe de Teste, Desenvolvedores
6	06/05/2024 - 10/05/2024	Execução de teste de estresse	Equipe de Teste, Desenvolvedores
7	13/05/2024 – 17/05/2024	Teste de segurança	Equipe de Teste, Desenvolvedores
8	20/05/2024 – 24/05/2024	Checklist dos requisitos funcionais e não funcionais	Equipe de Teste, Desenvolvedores
9	27/05/2024 – 31/05/2024	Documentação final dos resultados de validação e da lista de aprovação	Equipe de Teste

5.27 Lista de Aprovação

TIPO DE TESTE	REALIZADO		Testador: Equipe de Teste
	SIM	NÃO	OBSERVAÇÃO





CONSIDERAÇÕES FINAIS



REFERÊNCIAS

Pressman, Roger S.

Engenharia de Software: uma abordagem profissional / Roger S. Pressman, Bruce R. Maxim.