**Desenvolvimento de Games**

**Simulador de Falha de Porta - MetroSP**

**Nome dos Alunos:**

Eduardo Jorge Aranha Manduca – 25.00838-4

Gabriel Estavam Gomes – 25.00353-4

Lucas Matias Silveira – 25.01458-0

Caio Onha Ferreira – 24.00937-7

Sumário

[1. Descrição/Resumo do Projeto 3](#_Toc190505773)

[2. Extração de Requisitos 3](#_Toc190505774)

[2.2. Análise da Coleta de Requisitos 3](#_Toc190505775)

[3.1. Requisitos Funcionais 3](#_Toc190505776)

[3.2. Requisitos Não-Funcionais 3](#_Toc190505777)

[3.3. Especificação dos Casos de Uso 3](#_Toc190505778)

[4.1. Diagrama de Casos de Uso 4](#_Toc190505779)

[4.2. Diagrama de Classes 4](#_Toc190505780)

[4.3. Diagrama de Sequência 4](#_Toc190505781)

[4.4. Modelo de Banco de Dados 4](#_Toc190505782)

[4.5. Diagrama de Atividades (opcional) 4](#_Toc190505783)

[4.6. Diagrama de Estados (opcional) 4](#_Toc190505784)

[5. Implementação 4](#_Toc190505785)

[6. Testes 4](#_Toc190505786)

[7. Resultados e Considerações 4](#_Toc190505787)

[8. Registro da Apresentação ao Parceiro 4](#_Toc190505788)

[9. Referências 5](#_Toc190505789)

[10. Apêndice I 5](#_Toc190505790)

## Descrição/Resumo do Projeto

Servir como uma forma de simulador para que o maquinista tenha em mente as tarefas que precisa fazer no seu trabalho.

É um software de um simulador. Ele haverá limitações, e, como qualquer simulação, não pode expressar todas as nuances, os riscos e os deveres da realidade. Ele servirá apenas como um pequeno guia instrutivo e intuitivo para os maquinistas.

Correções e Feedback para o usuário - Quando ele erra, a aplicação deve mostrar o problema e uma possível solução

## Extração de Requisitos

A entrevista permite que possamos nos aprofundar dependendo da resposta dada pelo cliente, oferecendo mais alternativas e escolhas do que um questionário, por exemplo.

Quais responsabilidades de um maquinista são as mais importantes para serem inseridas no jogo (condução do trem e aceleração, paradas, segurança) ?

Qual o tipo de linguagem deve ser usado (formal, informal, popular)?

O game deve ser online ou offline?

Quanto tempo o game deve durar aproximadamente Qual estilo a interface gráfica do game deve ter?

## 2.2. Análise da Coleta de Requisitos

O cliente deseja um jogo realista de simulação de falhas em metrôs, focado em gerenciamento de crises como panes, acidentes e superlotação, direcionado a entusiastas de simulação e estudantes. Priorizou desempenho estável (sem travamentos, tempo de carregamento rápido e suporte a hardware variado), compatibilidade com Windows, macOS e Linux, e acessibilidade (legendas, ajustes de contraste). Salvar progresso automaticamente e proteger dados locais são essenciais.

## 3.1. Requisitos Funcionais

Funcionalidades essenciais do sistema, descritas de forma clara.Exemplo:

|  |  |  |
| --- | --- | --- |
| ID | Descrição do Requisito | Prioridade |
| RF01 | O sistema deve permitir o cadastro de usuários. | Alta |
| RF02 | O usuário deve poder visualizar relatórios. | Média |
| RF03 | O sistema deve enviar notificações por e-mail. | Baixa |
| RF04 | O sistema deve permitir o cadastro de usuários (admin e comum) com login via e-mail corporativo (@metrosp.com.br) e autenticação em rede interna Microsoft. | Alta |
| RF05 | O jogo deve ser um point-and-click com interação em 180° da cabine do trem. | Alta |
| RF06 | O trem deve operar em modo automático, mas permitir mudança para semiautomático quando falhas ocorrerem. | Média |
| RF07 | O console do trem deve exibir 3 telas com informações operacionais. | Alta |
| RF08 | O jogador deve poder visualizar o interior dos carros para vistoria durante falhas. | Alta |
| RF09 | Cada falha deve ter pontuação máxima/mínima, classificando gravidade (ex.: falhas mínimas não exigem ações críticas). | Média |
| RF10 | O sistema deve rastrear erros frequentes dos jogadores (ex.: esquecer chave na cabine) e gerar relatórios. | Média |
| RF11 | O admin deve ter acesso a relatórios de desempenho dos usuários | Baixa |
| RF12 | O jogo deve incluir um modo de visita técnica para inspeção realista do ambiente (opcional). | Baixa |

## 3.2. Requisitos Não-Funcionais

Restrições e características do sistema:

* Desempenho.
* Segurança.
* Compatibilidade.

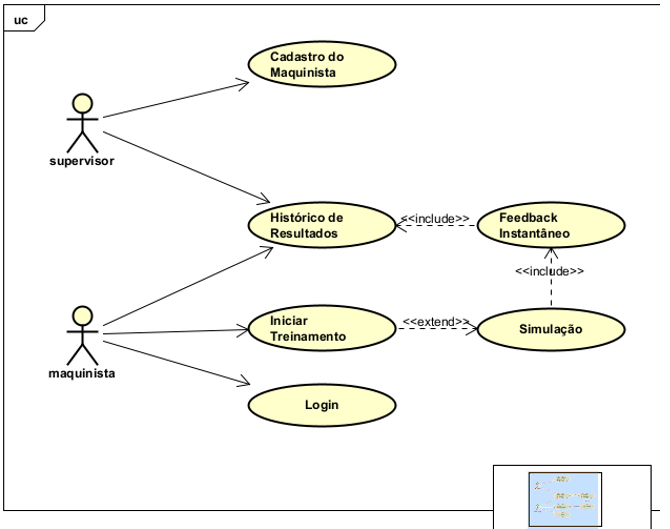
Exemplo:

|  |  |  |
| --- | --- | --- |
| ID | Descrição do Requisito | Prioridade |
| RNF01 | O sistema deve ser acessível em dispositivos móveis. | Alta |
| RNF02 | O sistema deve suportar até 1000 usuários simultâneos. | Média |
| RNF03 | O tempo de resposta do sistema não deve ultrapassar 2 segundos. | Alta |
| RNF04 | Os dados do usuário devem ser armazenados de forma criptografada. | Alta |
| RNF05 | O sistema deve estar disponível 99,9% do tempo. | Alta |

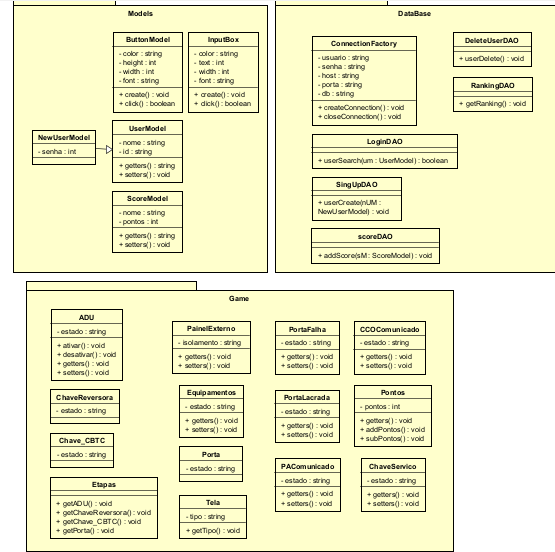
## 3.3. Especificação dos Casos de Uso

|  |  |  |
| --- | --- | --- |
| N° | Ator | Descrição |
| 1 | Supervisor | Sua função é cadastrar o maquinista, ver os históricos de resultados e receber feedbacks do resultado da simulação |
| 2 | Maquinista | Sua função é fazer login, iniciar treinamento e ver os resultados da simulação. |

## 4.1. Diagrama de Casos de Uso

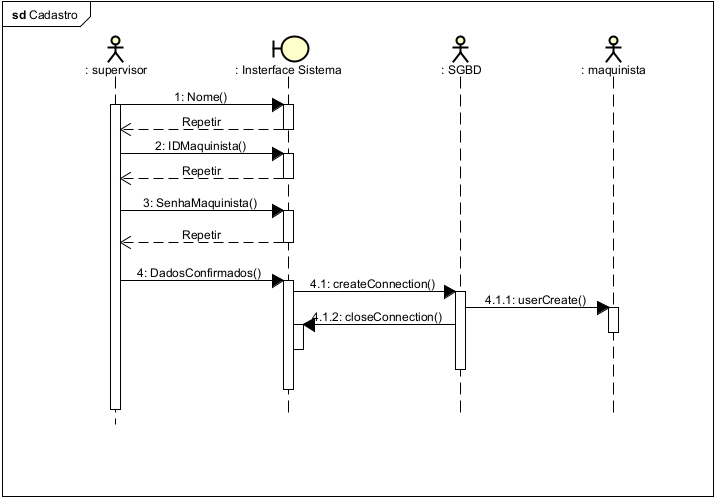


## 4.2. Diagrama de Classes

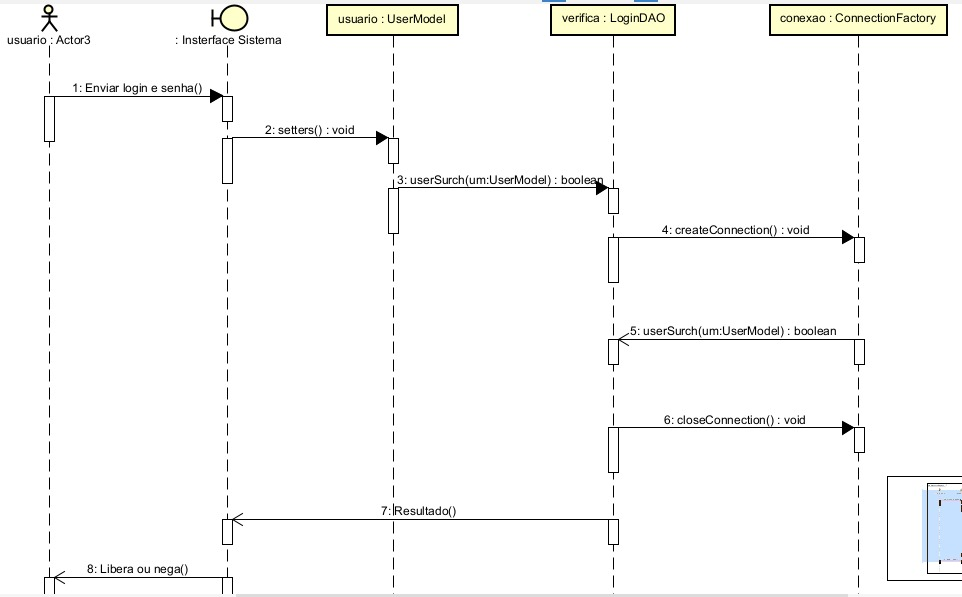


## 4.3. Diagrama de Sequência

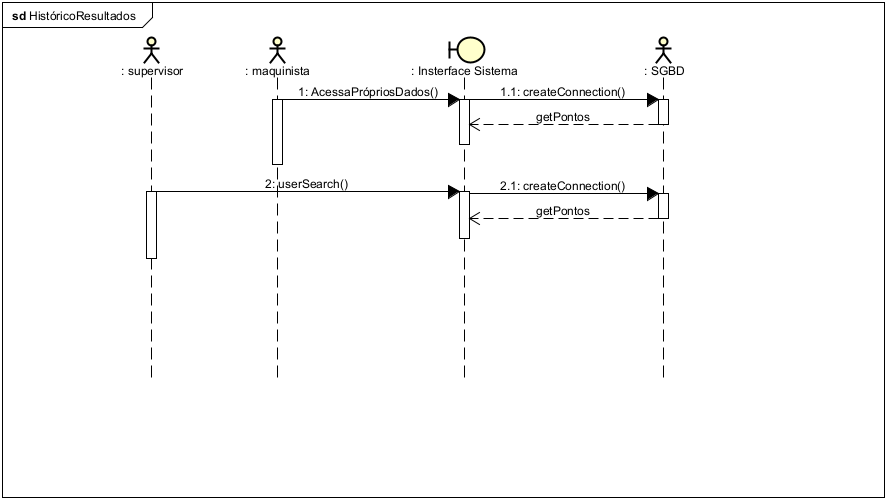
Cadastro maquinista



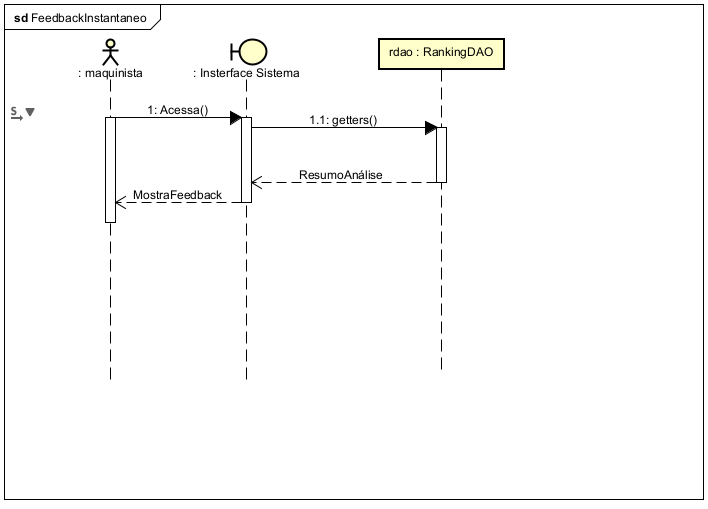
Login maquinista



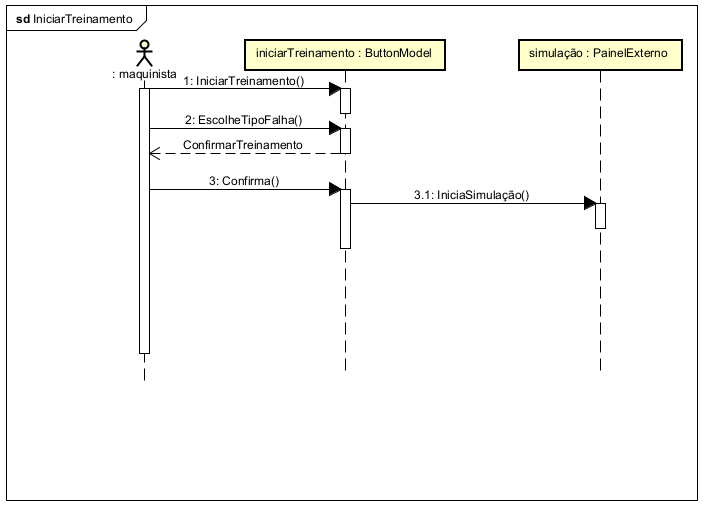
Histórico de resultados



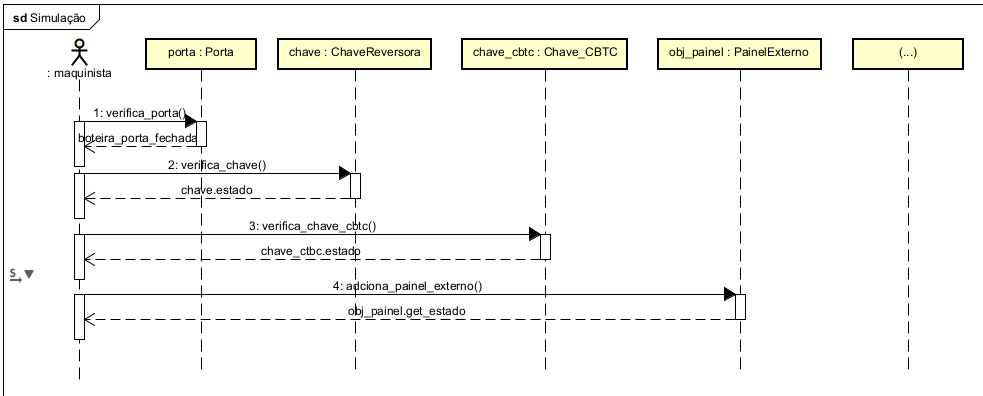
Feedback instantâneo



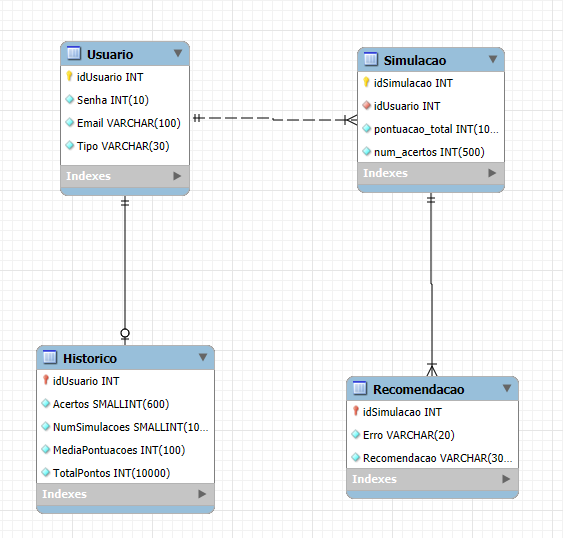
Iniciar treinamento



Simulação

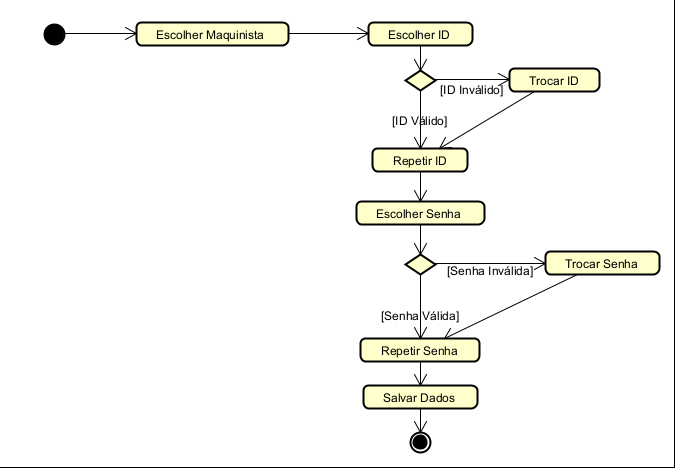


## 4.4. Modelo de Banco de Dados

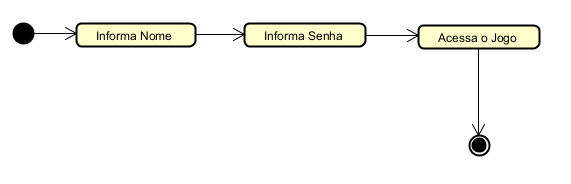


## 4.5. Diagrama de Atividades (opcional)

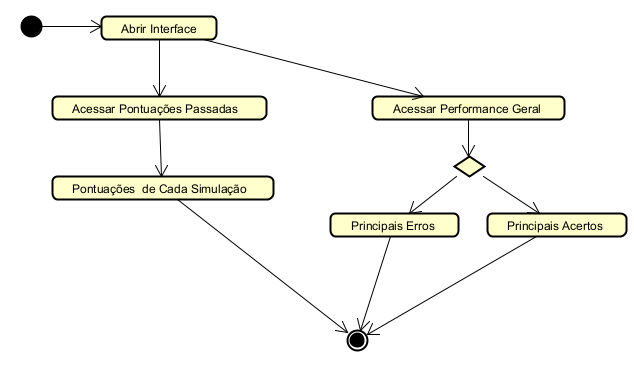
Cadastro Maquinista



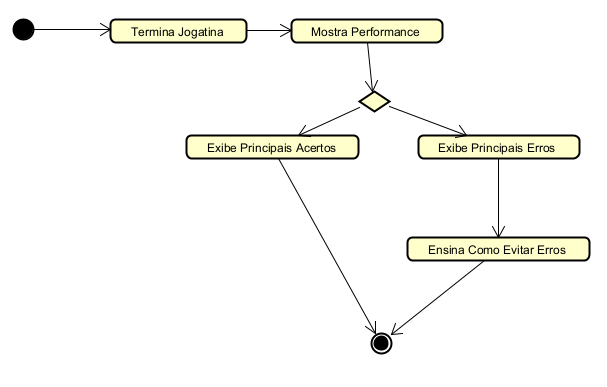
Login Maquinista



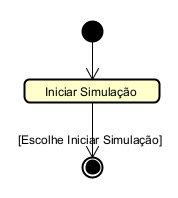
Histórico de Resultados



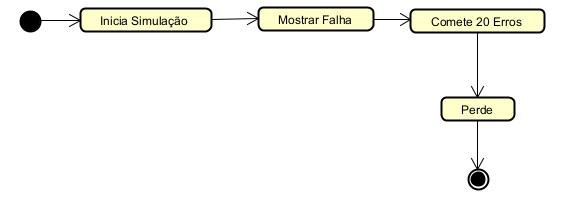
Feedback Instantâneo



Iniciar Treinamento



Simulação



## 4.6. Diagrama de Estados (opcional)

Representação de estados de objetos durante seu ciclo de vida.

## 5. Implementação

* Link para o repositório de código-fonte.

<https://github.com/EduardoManduca/MetroSP>

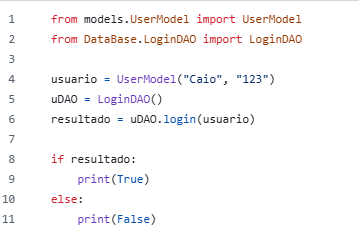
* Tecnologias utilizadas (linguagens, frameworks, padrões de arquitetura)

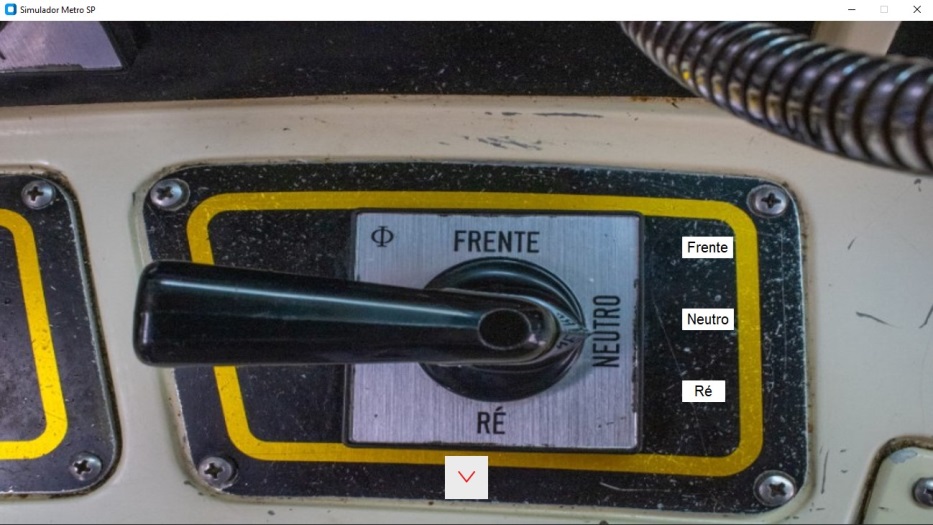
GitHub, Python, Pygame, CustomTkinter, Figma, MySQL Workbench, Astah, Diagramas UML, Visual Studio Code

## 6. Testes

* Testes unitários, de integração e funcionais.

texte\_conexão , para login

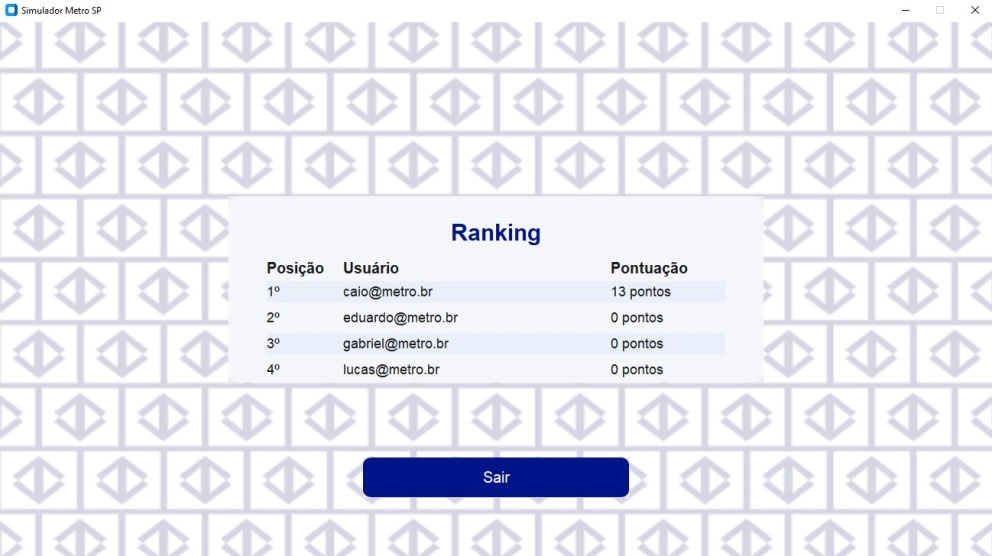
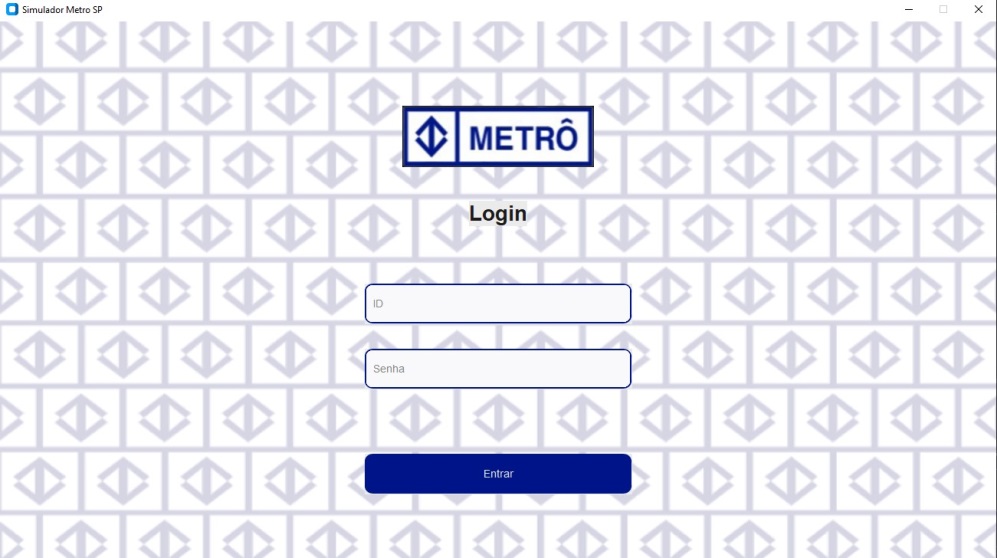


* Evidências como prints ou links para os testes. 



## 7. Resultados e Considerações

* Prints das telas do sistema.



* Comparação entre requisitos e entrega final.

A maioria dos requisitos foram atendidos. A maior mudança que ocorreu foi principalmente em relação aos requisitos do metrô. Mudamos muito do nosso planejamento inicial, que era mais abstrato e idealizado, justamente para atender a esses requisitos. Então, o trabalho ficou distante da visão inicial do grupo, mas com o propósito de priorizar os requisitos do metrô, que são mais importantes.  
Os requisitos com prioridade alta, como: “O jogo deve ser um point-and-click com interação em 180° da cabine do trem.” foram atendidos, enquanto alguns de prioridade mais baixa como notificações por email foram abandonados ao longo do projeto.

* Dificuldades enfrentadas e sugestões de melhorias.

Uma das dificuldades enfrentadas foi que, no primeiro semestre de aula estávamos aprendendo a modelar orientado a objeto ao mesmo tempo que programar orientado a objeto. Ou seja, os primeiros diagramas UML foram feitos antes de aprendermos algumas matérias de POO, e ficaram inconsistentes. Porém, conseguimos consertar muitos erros no decorrer do curso de POO.

## 8. Registro da Apresentação ao Parceiro

* Data da apresentação.
* Feedback recebido.
* Ajustes solicitados.

## 9. Referências

Listar fontes utilizadas no formato ABNT.

## 10. Apêndice I

Materiais adicionais, como roteiros de entrevistas ou questionários.