**Universidade Anhembi Morumbi**

Gustavo Brandão RA: 125111358231

Lucas Dias RA: 125111371139

Lucas Rodrigues RA: 125111373334

Vinícius Martin RA: 125111367756

**A3 – Modelagem de Software**

São Paulo

2024

**Sumário**

1. **Introdução................................................................................................1**
   1. Metodologia.........................................................................................1
2. **Desenvolvimento....................................................................................2**
3. **Conclusão..............................................................................................10**
4. **Referências bibliográficas....................................................................11**

**1. Introdução:**

Atualmente, o Brasil enfrenta um grande desafio para aumentar o índice de leitura entre seus cidadãos e para que medidas efetivas sejam tomadas, elas devem ser introduzidas desde a infância. De acordo com o site “Globo”, ocupamos o 39º lugar em um ranking com 43 países em um teste que mede as habilidades de leitura entre crianças.

Por esse motivo, é necessário a introdução de ideias inovadoras que possam auxiliar nesse problema, e uma das principais ideias que obteve êxito nos últimos anos, como é visto em uma matéria da CNN do ano passado ("Gamificação na educação: entenda o que é, importância e como pode ser usada"); é a gamificação. Se trata do ato de transformar o aprendizado em uma experiência divertida e mais próxima do entretenimento, fazendo com que o ensino pareça ou tenha princípios de um jogo.

Levando em consideração o exposto anterior, o presente projeto tem como objetivo principal apresentar um sistema de biblioteca gamificado, em que seus usuários recebem pontuações por leituras completas, metas diárias, semanais e mensais concluídas e desafios feitos. Essas pontuações podem ser usadas para resgatar prêmios ou alugar livros. Além disso, os usuários disputam em um ranking com outras pessoas, fazendo com que seja mais interessante e recompensador conquistar mais pontos.

Isto posto, é esperado que essas propostas facilitem e gerem engajamento para todas as faixas etárias, fazendo com que o hábito da leitura seja mais difundido na sociedade. Desse modo, com mais usuários e engajamento, haveriam mais feedbacks do sistema, que seriam revisados e levados em consideração para melhorias futuras, auxiliando ainda mais na proposta inicial do projeto.

* 1. **Metodologia:**

O desenvolvimento do sistema ocorreu através de princípios ágeis, com diversas etapas divididas por nível de complexidade entre os integrantes do grupo. Esse processo aconteceu na plataforma Monday, utilizada para facilitar a criação de projetos pela metodologia ágil. Seguindo os princípios do Scrum, os prazos foram divididos em sprints, que são tempos de desenvolvimento; de duas semanas, totalizando 3 sprints.

Na primeira sprint foram levantados os requisitos funcionais, não funcionais e os casos de uso, de forma que foi útil para a documentação e para o início da modelagem do software. Na segunda sprint, houve a criação do banco de dados, onde todos os dados sobre os usuários, logins, reservas e livros são armazenados. Por fim, na última sprint foi implementada a parte principal do código do sistema de biblioteca, em que, de fato, o projeto começou a tomar forma na prática. Isto posto, é possível dizer que a metodologia ágil ajudou de forma eficaz no desenvolvimento do projeto e nas metas que existiam para finalizar as etapas nas datas definidas.

**2. Desenvolvimento:**

Ao iniciar o projeto, houve o levantamento de requisitos, em que foi necessário entender quais seriam as partes mais importantes do sistema a serem desenvolvidas em volta da ideia principal de gamificação. Como o sistema não seria um software de biblioteca comum, os requisitos se diferiam de outros sistemas em certos aspectos.

Assim, após conversas com diversas pessoas e com os próprios integrantes do grupo, as funcionalidades principais do sistema foram definidas e os requisitos também. Abaixo, é possível verificar os requisitos funcionais do projeto:

- Realizar login: estudante e funcionário;

- Buscar livro pelo título, autor ou código de identificação;

- Listar todos os livros (ordem de adição ou alfabética);

- Listar todos os estudantes (ordem de adição ou alfabética);

- Reservar e devolver livros;

- Adicionar e remover livro do banco de dados;

- Registrar e remover usuário do sistema;

- Exibir a tabela de classificação dos usuários de acordo com suas pontuações;

- Listar todas as reservas.

Os requisitos não funcionais são vistos a seguir:

- Desempenho;

- Conectividade;

- Escalabilidade;

- Usabilidade;

- Disponibilidade.

Todas as funcionalidades foram desenvolvidas em Java. Os livros, usuários, funcionários, reservas e login de funcionários e usuários são armazenados em um banco de dados na plataforma online Railway, que permite acessá-las de forma prática e segura. Foram implementados métodos para execução dos casos de uso, que serão explicados com mais detalhes nas especificações dos casos de uso.

O sistema do Railway não possui como escrever scripts de criação ou para comandos gerais. Apesar disso, os scripts para criação das tabelas são vistos a seguir:

**Criação tabela usuarios:**

CREATE TABLE usuarios(

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR,

sobrenome VARCHAR,

registroUsuario VARCHAR,

pontos INT,

insignia VARCHAR);

**Criação tabela livros:**

CREATE TABLE livros(

id INT AUTO\_INCREMENT PRIMARY KEY,

titulo VARCHAR,

autor VARCHAR,

codigoLivro VARCHAR,

reservado BOOLEAN);

**Criação tabela loginFuncionario:**

CREATE TABLE loginFuncionario(

id INT AUTO\_INCREMENT PRIMARY KEY,

login VARCHAR,

senha VARCHAR);

**Criação tabela loginUsuario:**

CREATE TABLE loginUsuario(

id INT AUTO\_INCREMENT PRIMARY KEY,

login VARCHAR,

senha VARCHAR);

**Criação tabela reservas:**

CREATE TABLE reservas (

id INT PRIMARY KEY AUTO\_INCREMENT,

id\_usuario INT,

id\_livro INT,

data\_reserva DATE,

data\_devolucao DATE,

devolvido BOOLEAN DEFAULT FALSE,

diffDias INT,

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id),

FOREIGN KEY (id\_livro) REFERENCES livros(id));

Os scripts utilizados no sistema de biblioteca para manipular o banco de dados são vistos a seguir:

**Reserva de Livro:**

"SELECT \* FROM livros";

"UPDATE livros SET reservado = ? WHERE codigoLivro = ?";

"UPDATE usuarios SET pontos = pontos + 5 WHERE registroUsuario = ?";

"SELECT \* FROM usuarios";

"INSERT INTO reservas (id\_usuario, id\_livro, data\_reserva, devolvido) VALUES (?, ?, ADDDATE(CURDATE(), INTERVAL 1 DAY), ?)";

**Devolução de livro:**

"SELECT \* FROM livros";

"UPDATE livros SET reservado = ? WHERE codigoLivro = ?";

"UPDATE reservas SET data\_devolucao = ADDDATE(CURDATE(), INTERVAL 1 DAY), devolvido = TRUE, diffDias = DATEDIFF(data\_devolucao, data\_reserva) WHERE id\_usuario = ? AND id\_livro = ? AND devolvido = FALSE";

"SELECT \* FROM usuarios";

"SELECT diffDias from reservas WHERE id\_usuario = ? and id\_livro = ?";

"UPDATE usuarios SET pontos = pontos + ? WHERE id = ?";

**Procura de livro por código, autor ou título:**

"SELECT \* FROM livros";

**Cadastrar livro:**

"INSERT INTO livros (titulo, autor, codigoLivro, reservado) VALUES (?, ?, ?, ?)";

**Remover livro:**

"DELETE FROM livros WHERE titulo = ? AND autor = ? AND codigoLivro = ?";

**Cadastrar usuário:**

"INSERT INTO usuarios (nome, sobrenome, registroUsuario, pontos) VALUES (?, ?, ?, ?)";

**Remover usuário:**

"DELETE FROM usuarios WHERE nome = ? AND sobrenome = ? AND registroUsuario = ?";

"DELETE FROM reservas WHERE id\_usuario = ?";

"SELECT id FROM usuarios WHERE nome = ? AND sobrenome = ? AND registroUsuario = ?";

**Listagem alfabética livros:**

"SELECT \* FROM livros ORDER BY titulo ASC";

**Listagem alfabética usuários:**

"SELECT \* FROM usuarios ORDER BY nome ASC";

**Listagem por ordem de adição livros:**

"SELECT \* FROM livros";

**Listagem por ordem de adição usuários:**

"SELECT \* FROM usuarios";

**Listagem leaderboard:**

"SELECT \* FROM usuarios ORDER BY pontos DESC";

"UPDATE usuarios SET insignia = ? WHERE id = ?";

**Listagem reservas:**

"SELECT \* FROM reservas";

**Login para funcionário:**

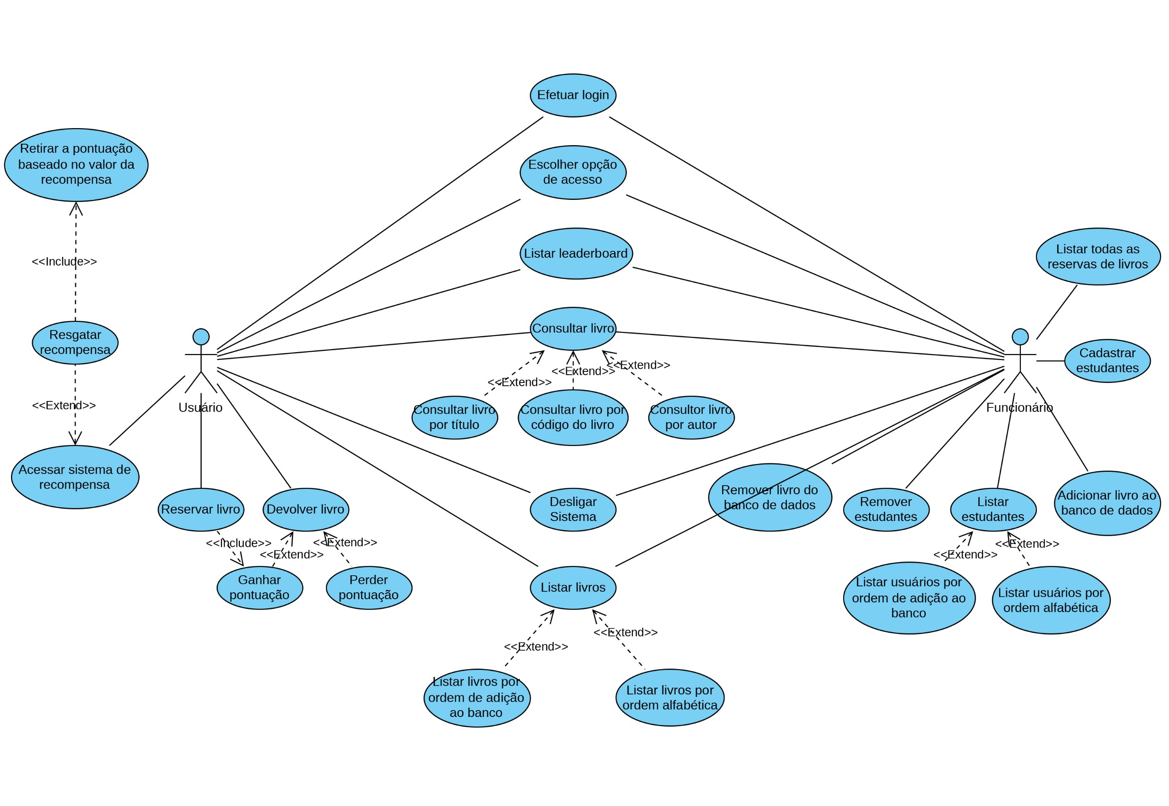
"SELECT \* FROM loginFuncionario";

**Login para usuário:**

"SELECT \* FROM loginUsuario";

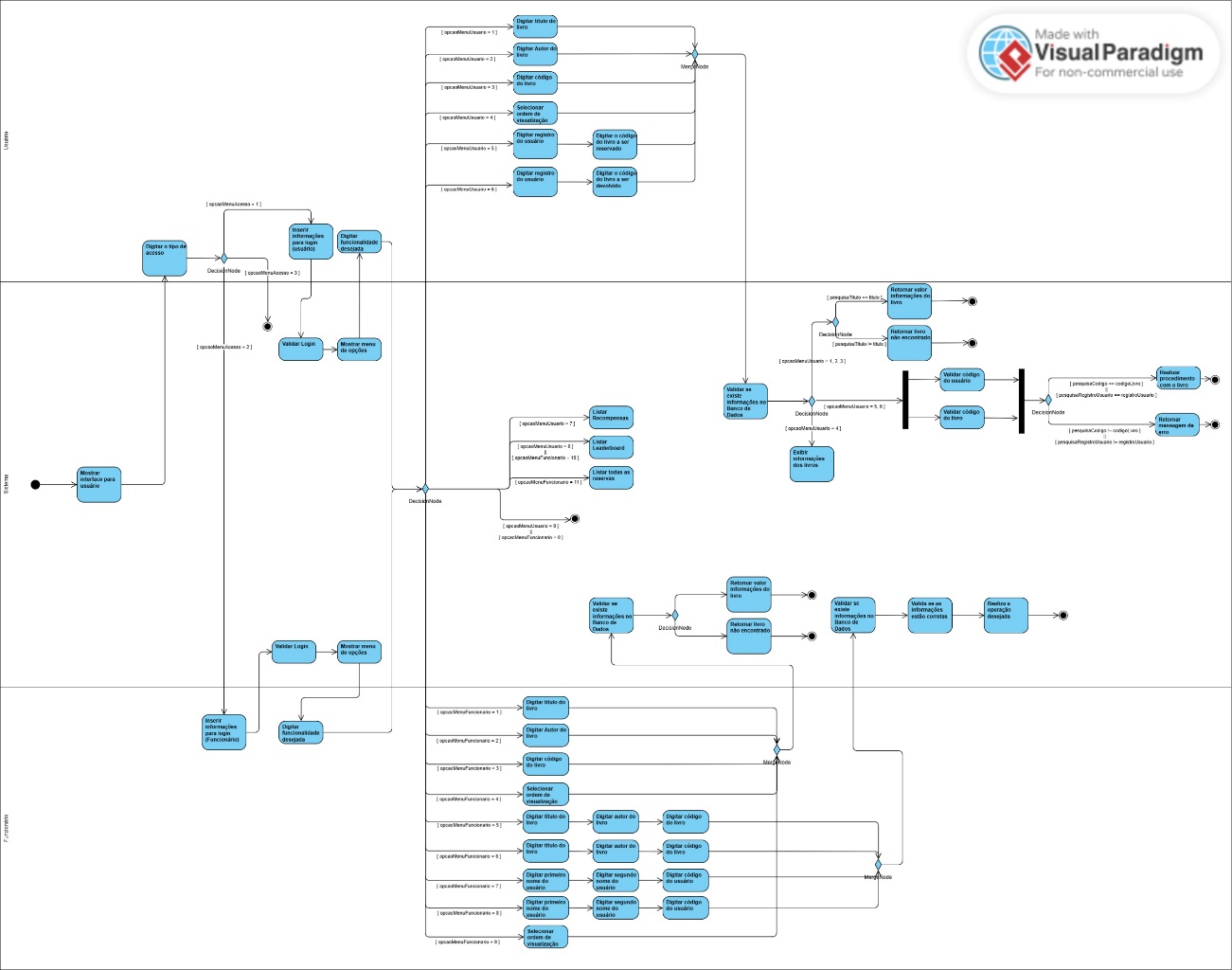
Ademais, foram desenvolvidos: o diagrama de casos de uso, o diagrama de atividades e o diagrama de classes. Eles são vistos a seguir:

**Figura 1 – Diagrama de casos de uso**



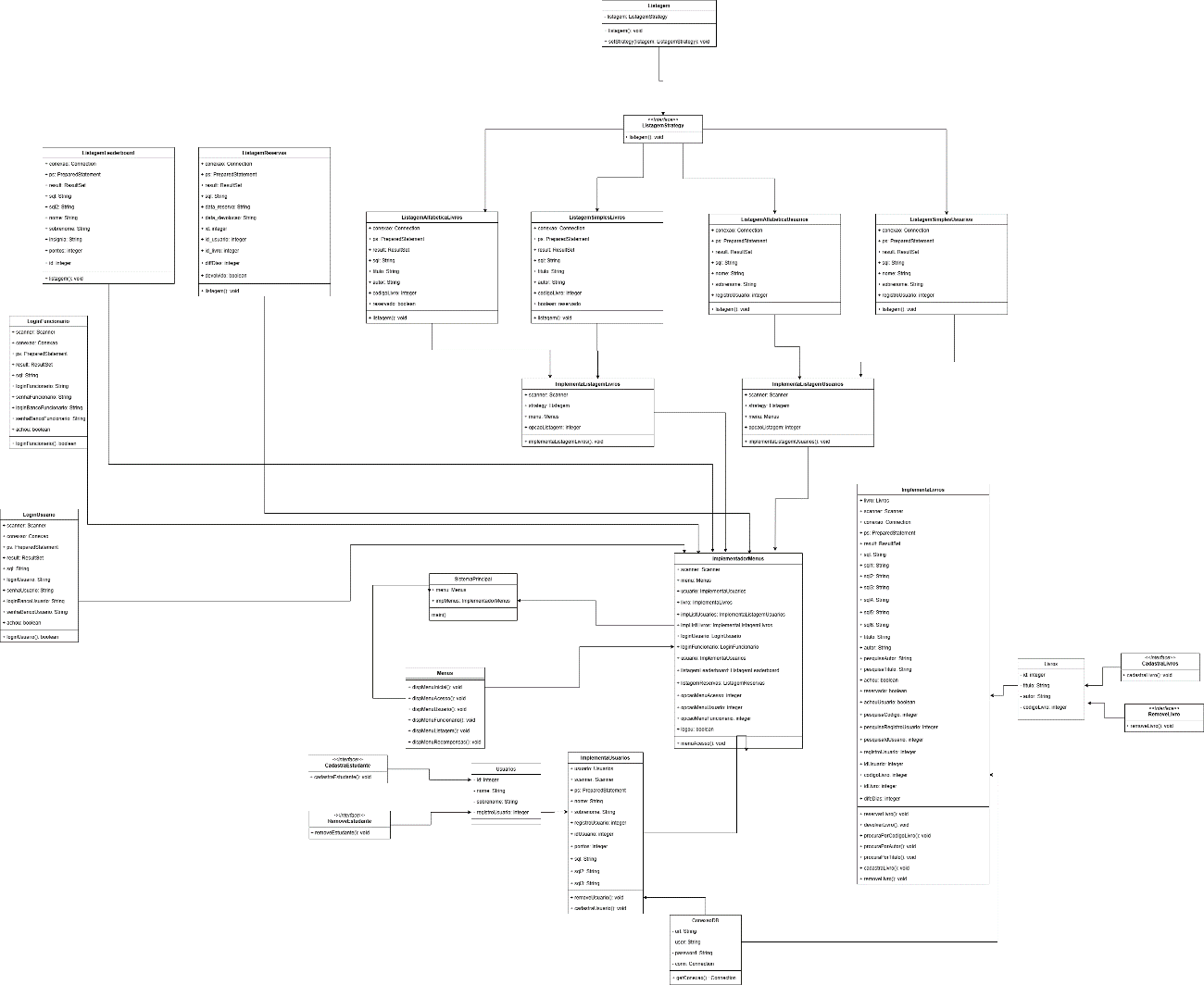
Fonte: Elaborada pelos autores.

**Figura 2 – Diagrama de atividades**



Fonte: Elaborada pelos autores.

**Figura 3 – Diagrama de classes**



Fonte: Elaborada pelos autores.

Por fim, é possível visualizar a seguir a especificação de cada um dos casos de uso do sistema:

Identificação: UC001

Nome: Escolher opção de acesso

Descrição: Escolher acessar o sistema como usuário comum ou funcionário.

Ator principal: Usuário, Funcionário

Pré-condições: N/A

Fluxo principal:

1. O ator escolhe com qual tipo de usuário deseja acessar o sistema.

2. O sistema exibe a tela de login para o tipo de usuário escolhido.

Pós-condições: O ator visualiza a tela de login para o tipo de usuário que escolheu

Identificação: UC002

Nome: Efetuar login

Descrição: Permite que o ator se autentique no sistema utilizando suas credenciais.

Ator principal: Usuário, Funcionário

Pré-condições: O ator deve ter escolhido que tipo de usuário ele deseja acessar

Fluxo principal:

1. O ator acessa a interface de login.

2. O ator insere suas credenciais.

3. O sistema verifica as credenciais e autentica o ator.

4. O sistema redireciona o ator para a interface principal que corresponde ao tipo de usuário escolhido pelo ator.

Pós-condições: O ator está autenticado no sistema.

Identificação: UC003

Nome: Consultar livro

Descrição: O ator procura por livros no sistema.

Ator principal: Usuário, Funcionário

Pré-condições: O ator deve estar autenticado no sistema.

Fluxo principal:

1. O ator acessa a área de consulta de livros.

2. O ator insere os critérios de busca.

3. O sistema apresenta os resultados da busca.

4. O ator visualiza, com sucesso, as informações dos livros encontrados.

Extensões:

UC004 – Consultar livro por título

UC005 – Consultar livro por código do livro

UC006 – Consultar livro por autor

Pós-condições: O sistema exibe os livros que estão de acordo com os critérios inseridos.

Identificação: UC004

Nome: Consultar livro por título

Descrição: O ator procura por livros inserindo o título.

Ator principal: Usuário, Funcionário

Pré-condições: O ator deve estar autenticado no sistema.

Fluxo principal:

1. O ator insere o título do livro.

2. O sistema retorna os livros que correspondem ao título informado.

Pós-condições: O sistema exibe os livros relacionados ao título.

Identificação: UC005

Nome: Consultar livro por código do livro

Descrição: O ator procura por livros inserindo o código específico do livro.

Ator principal: Usuário, Funcionário

Pré-condições: O ator deve estar autenticado no sistema.

Fluxo principal:

1. O ator insere o código do livro.

2. O sistema retorna os livros que correspondem ao código informado.

Pós-condições: O sistema exibe os livros relacionados ao código.

Identificação: UC006

Nome: Consultar livro por autor

Descrição: O ator procura por livros inserindo o nome do autor.

Ator principal: Usuário, Funcionário

Pré-condições: O ator deve estar autenticado no sistema.

Fluxo principal:

1. O ator insere o nome do autor.2. O sistema retorna os livros que correspondem ao nome do autor.

Pós-condições: O sistema exibe os livros relacionados ao nome do autor.

Identificação: UC007

Nome: Listar livros

Descrição: Mostra a lista completa de livros disponíveis no sistema.

Ator principal: Usuário, Funcionário

Pré-condições: O ator deve estar autenticado no sistema.

Fluxo principal:

1. O ator seleciona a opção de listar livros.

2. O sistema exibe a lista de livros disponíveis.

Extensões:

UC008 – Listar livros por ordem de adição ao banco

UC009 – Listar livros por ordem alfabética

Pós-condições: O sistema exibe a lista de livros.

Identificação: UC008

Nome: Listar livros por ordem de adição ao banco

Descrição: Mostra a lista completa de livros disponíveis ordenados pela data em que foram adicionados ao sistema.

Ator principal: Usuário, Funcionário

Pré-condições: O ator deve estar autenticado no sistema.

Fluxo principal:

1. O ator seleciona a opção de listar livros por ordem de adição.

2. O sistema exibe a lista de livros ordenada pela data de adição.

Pós-condições: O ator visualiza os livros ordenados pela data de adição.

Identificação: UC009

Nome: Listar livros por ordem alfabética

Descrição: Mostra a lista completa de livros disponíveis no sistema em ordem alfabética.

Ator principal: Usuário, Funcionário

Pré-condições: O ator deve estar autenticado no sistema.

Fluxo principal:

1. O ator seleciona a opção de listar livros por ordem alfabética.

2. O sistema exibe a lista de livros ordenada em ordem alfabética.

Pós-condições: O ator visualiza os livros organizados por ordem alfabética.

Identificação: UC010

Nome: Listar leaderboard

Descrição: Mostra o ranking de usuários baseado em pontuações no sistema.

Ator principal: Usuário, Funcionário

Pré-condições: Deve existir usuários com pontuações.

Fluxo principal:

1. O ator seleciona a opção de listar leaderboard.

2. O sistema exibe o ranking de usuários.

Pós-condições: O ator visualiza o ranking de usuários.

Identificação: UC011

Nome: Reservar livro

Descrição: O usuário realiza a reserva de um livro disponível no sistema.

Ator principal: Usuário

Pré-condições: O usuário deve estar autenticado no sistema e o livro deve estar disponível para reserva.

Fluxo principal:

1. O usuário seleciona a opção de reservar livro.

2. O usuário seleciona o livro desejado.

3. O sistema registra a reserva do livro.

Inclusão:

UC012 – Ganhar pontuação

Pós-condições: O livro é reservado pelo usuário.

Identificação: UC012

Nome: Ganhar pontuação

Descrição: O sistema atribui pontos aos usuários.

Ator principal: N/A (inclusão)

Pré-condições: O usuário deve estar autenticado no sistema e realizar ações que gerem pontuações.

Fluxo principal:

1. O usuário reserva um livro ou devolve na data certa.

2. O sistema verifica se as condições foram seguidas.

3. O sistema atribui a pontuação ao usuário.

4. O sistema atualiza a pontuação total do usuário.

Pós-condições: A pontuação do usuário é atualizada no sistema.

Identificação: UC013

Nome: Perder pontuação

Descrição: O sistema retira pontos dos usuários.

Ator principal: N/A (inclusão)

Pré-condições: O usuário deve estar autenticado no sistema e realizar ações que causem perda de pontuações.

Fluxo principal:

1. O usuário devolve um livro na data errada.

2. O sistema atualiza a pontuação total do usuário.

Pós-condições: A pontuação do usuário é atualizada no sistema.

Identificação: UC014

Nome: Devolver livro

Descrição: O usuário devolve um livro que foi pego anteriormente e o sistema atualiza a pontuação do usuário.

Ator principal: Usuário

Pré-condições: O usuário deve estar autenticado no sistema e ter um livro emprestado.

Fluxo principal:

1. O usuário seleciona a opção de devolver livro.

2. O usuário seleciona o livro a ser devolvido.

3. O sistema verifica se o livro foi devolvido na data correta ou não.

4. O sistema retira ou atribui pontuações dependendo se o livro foi entregue na data correta ou não.

5. O sistema registra a devolução do livro.

Extensão:

UC012 – Ganhar pontuação

UC013 – Perder pontuação

Pós-condições: O sistema registra a devolução e a pontuação é atualizada para o usuário.

Identificação: UC015

Nome: Acessar sistema de recompensa

Descrição: O usuário acessa o sistema de recompensas

Ator principal: Usuário

Pré-condições: O usuário deve estar autenticado no sistema.

Fluxo principal:

1. O usuário seleciona a opção de exibir o sistema de recompensas.

2. O sistema exibe as recompensas disponíveis.

Inclusão:

UC016 – Resgatar recompensa

Pós-condições: O usuário visualiza as recompensas disponíveis.

Identificação: UC016

Nome: Resgatar recompensa

Descrição: O usuário pode resgatar recompensar usando sua pontuação.

Ator principal: Usuário

Pré-condições: O usuário deve possuir pontuação suficiente para resgatar a recompensa.

Fluxo principal:

1. O usuário seleciona a recompensa desejada.

2. O sistema verifica se o usuário possui a pontuação necessária.

3. O sistema registra o resgate e desconta a pontuação do usuário.

Extensões:

UC017 – Retirar a pontuação baseado no valor da recompensa

Pós-condições: O resgate da recompensa é registrado e a pontuação do usuário é atualizada.

Identificação: UC017

Nome: Retirar a pontuação baseado no valor da recompensa

Descrição: O sistema desconta a pontuação do usuário com base no valor da recompensa resgatada.

Ator principal: N/A (inclusão)

Pré-condições: N/A

Fluxo principal:

1. O sistema verifica a quantidade de pontos da recompensa.

2. O sistema reduz a quantidade de pontos da recompensa da pontuação total do usuário.

Pós-condições: O sistema atualiza a pontuação do usuário.

Identificação: UC018

Nome: Desligar sistema

Descrição: Desliga o sistema

Ator principal: Usuário, Funcionário

Pré-condições: O ator deve estar autenticado no sistema.

Fluxo principal:

1. O ator seleciona a opção de desligar o sistema.

2. O sistema encerra e fecha a interface.

Pós-condições: O sistema é encerrado.

Identificação: UC019

Nome: Cadastrar estudantes

Descrição: Registra novos estudantes no banco de dados do sistema.

Ator principal: Funcionário

Pré-condições: O funcionário deve estar autenticado no sistema.

Fluxo principal:

1. O funcionário seleciona a opção de cadastrar estudantes.

2. O funcionário insere os dados do estudante.

3. O sistema registra o estudante no banco de dados.

Pós-condições: O estudante é registrado no sistema.

Identificação: UC020

Nome: Remover estudantes

Descrição: Remove estudantes do banco de dados.

Ator principal: Funcionário

Pré-condições: O funcionário deve estar autenticado no sistema.

Fluxo principal:

1. O funcionário seleciona a opção de remover estudantes.

2. O funcionário seleciona o estudante a ser removido.

3. O sistema remove o estudante do banco de dados.

Pós-condições: O estudante é removido do sistema.

Identificação: UC021

Nome: Listar estudantes

Descrição: Mostra a lista de estudantes registrados no sistema.

Ator principal: Funcionário

Pré-condições: O funcionário deve estar autenticado no sistema.

Fluxo principal:

1. O funcionário seleciona a opção de listar estudantes.

2. O sistema exibe a lista de estudantes registrados.

Extensões:

UC022 – Listar usuários por ordem de adição ao banco

UC023 – Listar usuários por ordem alfabética

Pós-condições: O funcionário visualiza a lista de estudantes no sistema.

Identificação: UC022

Nome: Listar usuários por ordem de adição ao banco

Descrição: Mostra os usuários em ordem de data de adição no sistema.

Ator principal: Funcionário

Pré-condições: O funcionário deve estar autenticado no sistema.

Fluxo principal:

1. O funcionário seleciona a opção de listar usuários por ordem de data de adição no sistema.

2. O sistema exibe os usuários na ordem de adição ao banco de dados.

Pós-condições: O funcionário visualiza os usuários organizados pela data de adição ao banco de dados.

Identificação: UC023

Nome: Listar usuários por ordem alfabética

Descrição: Mostra os usuários em ordem alfabética.

Ator principal: Funcionário

Pré-condições: O funcionário deve estar autenticado no sistema.

Fluxo principal:

1. O funcionário seleciona a opção de listar usuários por ordem alfabética.

2. O sistema exibe os usuários em ordem alfabética.

Pós-condições: O funcionário visualiza os usuários em ordem alfabética.

Identificação: UC024

Nome: Adicionar livro ao banco de dados

Descrição: Adiciona novos livros ao banco de dados.

Ator principal: Funcionário

Pré-condições: O funcionário deve estar autenticado no sistema.

Fluxo principal:

1. O funcionário seleciona a opção de adicionar livros ao banco de dados.

2. O funcionário insere as informações do livro.

3. O sistema registra o livro no banco de dados.

Pós-condições: O livro é adicionado ao sistema e fica disponível.

Identificação: UC025

Nome: Remover livro do banco de dados

Descrição: Remove livros do banco de dados.

Ator principal: Funcionário

Pré-condições: O funcionário deve estar autenticado no sistema e o livro não deve estar reservado ou emprestado.

Fluxo principal:

1. O funcionário seleciona a opção de remover livros do banco de dados.

2. O funcionario seleciona o livro a ser removido.

3. O sistema verifica se o livro não está reservado ou emprestado.

4. O sistema remove o livro do banco de dados.

Pós-condições: O livro é removido do sistema.

Identificação: UC026

Nome: Listar todas as reservas de livros

Descrição: Mostra todos as reservas de livro realizadas no sistema.

Ator principal: Funcionário

Pré-condições: O funcionário deve estar autenticado no sistema.

Fluxo principal:

1. O funcionário seleciona a opção de listar reservas de livros.

2. O sistema exibe todas as reservas.

Pós-condições: O funcionário visualiza a lista de reservas.

**3. Conclusão:**

É possível constatar que a criação do presente projeto, junto da ideia de gamificação, representou uma opção inovadora para futuros sistemas que serão desenvolvidos nesse contexto, estimulando o hábito de leitura e gerando engajamento entre os usuários. Ademais, o uso da metodologia ágil, assim como citado no tópico de metodologia, foi de extrema importância e garantiu que as etapas fossem entregues nos prazos e feitas da maneira correta ao serem dividas por níveis de complexidade e entregues para diferentes integrantes do grupo.

Isto posto, é mister notar que avaliações futuras dos usuários serão de extrema importância para melhorias no projeto e na ideia em si dele, fazendo com que o sistema amadureça e fortaleça o seu objetivo principal, que é de fazer com que os usuários possam ler de maneira divertida e, com isso, aumentar seu tempo de leitura. Essa abordagem não apenas incentiva a leitura, mas também permite que os usuários desenvolvam competências como concentração, disciplina e capacidade de solucionar desafios, ampliando os benefícios além do objetivo principal. Além disso, o processo de refinamento constante permitirá que o projeto não apenas alcance seus objetivos iniciais, mas também se torne uma referência no uso da gamificação para incentivar práticas educativas e culturais.

**4. Referências bibliográficas:**

G1. **Em teste de habilidades de leitura entre crianças, Brasil ocupa 39º lugar em ranking com 43 países.** Disponível em: <https://g1.globo.com/educacao/noticia/2023/05/16/criancas-do-brasil-tem-habilidades-de-leitura-inferiores-as-do-azerbaijao-e-do-uzbequistao-mostra-teste-internacional.ghtml>

CNN Brasil. **Gamificação na educação: entenda o que é, importância e como pode ser usada.** Disponível em: <https://www.cnnbrasil.com.br/lifestyle/gamificacao-na-educacao/>.

‌TOTVS. **Metodologia ágil: o que é e quais são as 10 principais.** Disponível em: <https://www.totvs.com/blog/negocios/metodologia-agil/>

‌ALURA. **Metodologia Ágil - O que é?** Disponível em: <https://www.alura.com.br/artigos/o-que-e-metodologia-agil?srsltid=AfmBOoq7xbyATK43ahCErV6L6fPDL6u6MTVEFHEVDR02nQI-h1CDQj5C>

Lucidchart. **Diagrama de caso de uso UML: O que é, como fazer e exemplos.** Disponível em: < https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>

IBM. Diagramas de Classes. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>

Railway. **A3 – Modelagem de Software.** Disponível em: <https://railway.app/project/3f7cbb4a-baef-43e8-8052-12a7e5f7fcad>