**Desenvolvimento de Games**

**Show do Milhão Acadêmico**

**Guilherme Zanelli de Novaes RA: 25.00225-4**

**Rodrigo Ribeiro Persolli RA: 25.00074-6**

**Nicolas Paulino Volpatto RA: 25.01259-2**

**Guilherme Gois Cruz Coelho RA: 25.00702-2**

**Patrick Araujo dos Santos RA: 25.01311-1**

**Tiago Machado Azevedo Vianna RA: 25.00482-1**

Sumário

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

[2. Extração de Requisitos 4](#_Toc190505774)

[2.2. Análise da Coleta de Requisitos 4](#_Toc190505775)

[3.1. Requisitos Funcionais 6](#_Toc190505776)

[3.2. Requisitos Não-Funcionais 7](#_Toc190505777)

[3.3. Especificação dos Casos de Uso 7](#_Toc190505778)

[4.1. Diagrama de Casos de Uso 19](#_Toc190505779)

[4.2. Diagrama de Classes 20](#_Toc190505780)

[4.3. Diagrama de Sequência 21](#_Toc190505781)

[4.4. Modelo de Banco de Dados 21](#_Toc190505782)

[4.5. Diagrama de Atividades (opcional) 23](#_Toc190505783)

[4.6. Diagrama de Estados (opcional) 23](#_Toc190505784)

[5. Implementação 23](#_Toc190505785)

[6. Testes 23](#_Toc190505786)

[7. Resultados e Considerações 23](#_Toc190505787)

[8. Registro da Apresentação ao Parceiro 23](#_Toc190505788)

[9. Referências 23](#_Toc190505789)

[10. Apêndice I 23](#_Toc190505790)

## Descrição/Resumo do Projeto

O objetivo deste projeto é o desenvolvimento de um sistema de jogo educacional interativo, inspirado no formato do programa "Show do Milhão", voltado para a avaliação do conhecimento dos alunos do Colégio Poliedro. Este sistema permitirá que os professores testem o domínio de conteúdos por parte dos alunos, oferecendo uma ferramenta lúdica e envolvente. O jogo será baseado em perguntas de múltipla escolha, com respostas de acerto e erro exibidas de maneira rápida, proporcionando feedback imediato. Além disso, o jogo irá apresentar um nível de dificuldade progressivo, tornando-se mais desafiador à medida que o usuário avança nas fases, com o intuito de avaliar de forma precisa e gradual o aprendizado dos alunos.

O público-alvo deste sistema são os professores e alunos do Colégio Poliedro. Os professores utilizarão a plataforma como ferramenta de avaliação do conhecimento adquirido pelos alunos em diferentes disciplinas, podendo personalizar o conteúdo das questões conforme suas necessidades. Os alunos, por sua vez, serão os usuários finais do sistema, participando ativamente do jogo e testando seus conhecimentos de forma divertida e desafiadora.

O escopo do projeto abrange o desenvolvimento de um jogo educativo com as seguintes características principais:

* Interface gráfica interativa
* Sistema de perguntas e respostas
* Feedback imediato
* Progressão de dificuldade
* Armazenamento de resultados

Para o desenvolvimento do sistema, as seguintes tecnologias serão empregadas:

* Linguagem de Programação: A plataforma será desenvolvida utilizando a linguagem de programação Java.
* Ambiente de Desenvolvimento: O código será desenvolvido no Visual Studio Code, em sua última versão
* Biblioteca Gráfica: A interface gráfica do jogo será construída com a utilização da biblioteca Java Swing.
* Armazenamento de Dados: Banco de dados simples para salvar as informações do usuário e dos resultados das avaliações, podendo ser alterado por um administrador (professor ou coordenador), que poderá adicionar ou retirar perguntas do banco de dados.

## Extração de Requisitos

Entrevistas com os stakeholders do sistema (professores, coordenadores e alunos), a fim de obter compreensão aprofundada a respeito da idealização de projeto. Além de explicitar as reais expectativas e necessidades dos usuários, em relação ao sistema de pontuação, perguntas e feedbacks.

Extração através do brainstorming, ao juntar os usuários aos programadores para uma sessão de ideias irrestritas, onde depois serão filtradas mantendo somente as que possam ser realizadas.

## 2.2. Análise da Coleta de Requisitos

O projeto tem como objetivo desenvolver um jogo educacional interativo com o objetivo de reforçar conteúdos já abordados, especialmente nas disciplinas de Química, Física e Biologia, a partir do oitavo ano do Ensino Fundamental II. O público-alvo é composto por estudantes do Ensino Fundamental II e Ensino Médio da rede Poliedro. O jogo será realizado em casa, durante o período de lazer, e não terá caráter de avaliação, o que evita qualquer tipo de competição direta entre os estudantes.

A interface do jogo deverá seguir um layout minimalista, focado no conteúdo e sem distrações visuais. É recomendável utilizar a identidade visual da rede Poliedro, observando a sua paleta de cores e estilo. O jogo terá um apresentador virtual inspirado no apresentador Silvio Santos (apenas voz, sem uso de imagens) Os sons e os efeitos podem ser exibidos, desde que haja a opção de silenciar completamente o jogo.

O cadastro será realizado exclusivamente pelos professores, contendo as seguintes informações: nome completo do aluno, série, idade e e-mail institucional (Poliedro) os professores também serão responsáveis pelo registro e aplicação das questões, que devem estar relacionadas às disciplinas e séries específicas, com progressão de dificuldade.

Ao iniciar o jogo, o aluno poderá selecionar a disciplina que deseja cursar. As questões serão apresentadas em formato de múltipla escolha, com cinco opções (A, B, C, D e E), que podem ser sorteadas ou selecionadas de acordo com a série. Cada partida terá 12 questões, com checkpoints nas questões 3, 6 e 10, o que permitirá ao jogador "salvar" o seu progresso. Se o aluno errar uma questão, poderá optar por reiniciar do início ou retornar com o valor do último checkpoint.

O aluno terá acesso a três tipos de ajuda ao longo do jogo: eliminar duas alternativas, pular duas questões ou, eventualmente, comprar ajudas adicionais com uma moeda fictícia. A última pergunta, conhecida como "pergunta do milhão", não estará disponível para auxílio. Ao confirmar uma resposta, o sistema exibirá uma mensagem que perguntará: "Você tem plena certeza disso?" como um reforço à escolha.

Ao final da jornada, os estudantes receberão uma recompensa em dinheiro fictício, de acordo com o seu desempenho. Um sistema de ranking interno trimestral poderá ser utilizado pelos professores para monitorar aqueles que acumularam mais valor, sem a exibição pública para os alunos, de modo a evitar o estímulo à competição.

## 3.1. Requisitos Funcionais

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

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Descrição do Requisito | Prioridade | Requisitos Relacionados |
| RF01 | O sistema deverá utilizar um banco de dados | Alta | RNF01 |
| RF02 | O sistema deverá permitir a personalização de cores da interface (fundo, perguntas e letras). | Baixa | RNF05, RNF06, RNF07 |
| RF03 | O sistema deverá permitir que o usuário escolha o tema ou disciplina das perguntas (ex: Matemática, História). | Média | RF10 |
| RF04 | O sistema deverá apresentar ao usuário a resposta correta em menos de 1 segundo | Média | RNF04 |
| RF05 | O sistema deverá registrar as respostas inseridas pelo usuário | Média | RNF02, RNF03, RF15, RF12 |
| RF06 | O sistema deverá permitir autenticação do usuário | Alta | RNF08 |
| RF07 | O sistema deverá possuir checkpoints | Média | RNF08 |
| RF08 | O sistema deverá oferecer quatro tipos de ajuda ao jogador (ex: eliminar uma resposta, pular pergunta). | Média | RNF09 |
| RF09 | O sistema deverá exibir um ranking de pontuações dos alunos. | Média | RNF12 |
| RF10 | O sistema deverá permitir os professores removerem e/ou adicionarem perguntas. | Alta | RF03, RNF02, RNF03 |
| RF11 | O sistema deverá mostrar ao final do jogo quantas perguntas o usuário respondeu corretamente | Média | RF02, RF03, RF10, RNF02, RNF03 |
| RF12 | O sistema deverá possuir um tempo limite para responder cada pergunta | Média | RF05 |
| RF13 | O Sistema deverá permitir que o administrador filtre as estatísticas por turma e matéria. | Alta | RF03 |
| RF14 | O sistema deve sortear perguntas aleatórias de um banco amplo sem repetições na mesma partida. | Média | RF17 |
| RF15 | O sistema deverá possuir trilha sonora/áudio para acertos e erros. | Baixa | RNF14 |
| RF16 | O sistema deverá permitir definir níveis (fácil, médio, difícil) manualmente para as perguntas. | Média | RF03, RF10 |
| RF17 | O sistema deverá aplicar algoritmos de randomização para a ordem das alternativas. | Média | RF14 |

## 3.2. Requisitos Não-Funcionais

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Descrição do Requisito | Categoria | Prioridade | Requisitos Relacionados |
| RNF01 | O sistema deverá utilizar o MySQL | Interoperalidade | Alta | RF01 |
| RNF02 | O sistema deverá dificultar as perguntas após uma sequência de cinco acertos | Implementação | Baixa | RF05 |
| RNF03 | O sistema não deverá repetir perguntas em uma mesma sessão. | Implementação | Média | RF05 |
| RNF04 | O sistema deverá conectar-se ao banco de dados em menos de 10 segundos | Desempenho | Baixa | RF01, RNF01 |
| RNF05 | O sistema deverá possuir uma cor de fundo azul escuro (#0716A0) | Padronização | Baixa | RF02 |
| RNF06 | O sistema deverá possuir uma cor para o fundo das perguntas amarela (#E3EE0A) | Padronização | Baixa | RF02 |
| RNF07 | O sistema deverá possuir uma cor das letras branca (#FFFDFD) | Padronização | Baixa | RF02 |
| RNF08 | O Sistema não deverá permitir checkpoints nas últimas três perguntas | Implementação | Alta | RF07 |
| RNF09 | O Sistema não deverá permitir utilizar ajudas nas últimas três perguntas | Implementação | Alta | RF08 |
| RNF10 | O jogo deverá ser responsivo e funcionar corretamente em diferentes tamanhos de tela. | Usabilidade | Média | RF02 |
| RNF11 | O sistema deverá funcionar sem falhas críticas por pelo menos 1 hora de jogo contínuo. | Confiabilidade | Alta | RNF13, RNF15, RNF16 |
| RNF12 | O sistema deverá permitir o ranking ser visível somente para o professor. | Implementação | Baixa | RF09 |
| RNF13 | |  | | --- | | O sistema deve ser compilável com Java versão 17 ou superior. | | Portabilidade | Alta | RF10, RNF15, RNF16 |
| RNF14 | O sistema deverá permitir ao usuário ativar/desativar efeitos sonoros e trilha sonora. | Implementação | Baixa | RF15 |
| RNF15 | O sistema deverá ser capaz de lidar com picos de acesso de até 100 alunos simultâneos. | Desempenho | Baixa | RF10, RNF16 |
| RNF16 | O sistema deve ser construído seguindo boas práticas de Clean Code. | Arquitetura | Alta | RF10, RNF15 |

## 3.3. Especificação dos Casos de Uso

Especificação de Atores

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| *Aluno* | *Usuário responsável por utilizar os checkpoints, responder as perguntas, visualizar o histórico de respostas e utilizar ajuda.* |
| *Professor* | *Usuário responsável por visualizar o ranking, gerenciar os alunos, as perguntas e os checkpoints. Além disso, esse usuário possui todas as funções de um “Aluno”* |
| *Coordenador* | *Usuário responsável por gerenciar os professores, além de possuir todas as funções de um usuário “Professor”* |

**Especificação de Casos de Uso – Cadastrar jogador**

**Breve Descrição:**

Os usuários devem se cadastrar no sistema para pode jogar o jogo. Os dados a serem informados são: nome de exibição (o nome que aparecerá no ranking), série escolar e e-mail institucional.

**Fluxo Básico**

Este caso de uso se inicia quando o ator aluno escolhe a opção de “Cadastrar”.

1. *Abrir a tela de cadastro*
2. *Preencher os dados pessoais*
3. *Salvar os dados*
4. *Registrar dados no banco de dados*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:** Caso o usuário não preencha os dados ou os preencha de forma incorreta, o sistema deve informar o erro e solicitar os dados ao usuário novamente.

**[FA2] Fluxo Alternativo 2:** Caso o usuário tente se cadastrar, mas já exista um cadastro com seus dados, o sistema deve informar e impedir que o mesmo cadastro seja efetuado mais de uma vez

**Pré-condições**

O sistema deve estar vinculado ao banco de dados MySQL

**Pós-condições**

Os dados do usuário deverão ser registrados no banco de dados MySQL

**Regras de Negócio:**

[RN1] Os usuários devem se registrar antes de jogar, os professores podem realizar o cadastro dos alunos, os coordenadores podem realizar o cadastro dos professores e dos alunos, não é possível realizar o cadastro da mesma pessoa mais de uma vez.

**Especificação de Casos de Uso – Manter professores <<crud>>**

**Breve Descrição:**

*Este caso de uso se refere à permissão dos usuários cadastrados no sistema como “Coordenador” de gerenciar o ator “Professor”. Os coordenadores podem cadastrar novos professores ao sistema, remover professores do sistema, alterar informações já existentes dos professores no sistema e ver quantos e quais professores estão cadastrados no sistema.*

**Fluxo Básico**

Este caso de uso tem como objetivo consultar, incluir, alterar e excluir professores no sistema pelo ator coordenador.

1. Se cadastrar no sistema como “Coordenador”

2*. Abrir o menu principal*

*3. Clicar em “Gerenciar professores”*

*4. Gerenciar os professores conforme desejado*

*5. Clicar em “Salvar”*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:** Caso o coordenador decida excluir um professor, seus dados devem ser retirados do banco de dados.

**[FA2] Fluxo Alternativo 2:** Caso o coordenador decida alterar as informações de um professor, as alterações devem ser registradas no banco de dados.

**[FA3] Fluxo Alternativo 3:** Caso o coordenador decida adicionar um professor ao sistema, ele poderá cadastrá-lo no sistema, inserindo seu nome e e-mail institucional. Essa alteração deverá ser registrada no banco de dados.

**Pré-condições**

O sistema deve estar vinculado ao banco de dados MySQL e o usuário deve estar cadastrado no sistema com a função de “Coordenador”.

**Pós-condições**

As alterações feitas devem ser salvas no banco de dados MySQL.

**Regras de Negócio:**

[RN1] O cadastro de um novo professor ao sistema deve ser feito com o e-mail institucional do poliedro, e a senha do cadastro deve ser uma escolha pessoal do professor.

**Especificação de Casos de Uso – Manter alunos <<crud>>**

**Breve Descrição:**

*Os atores do sistema “Professor” ou “Coordenador” podem gerenciar os alunos, alterando suas informações, excluindo-os do sistema, adicionando novos alunos ao sistema ou apenas verificar quantos e quais alunos estão cadastrados no sistema.*

**Fluxo Básico**

Este caso de uso tem como objetivo consultar, incluir, alterar e excluir alunos no sistema pelos atores coordenador e professor.

1. *Realizar o login no sistema como “Professor” ou “Coordenador”*
2. *Ir ao menu principal*
3. *Escolher a opção « Gerenciar alunos »*
4. *Gerenciar os alunos como desejado*
5. *Clicar em « Salvar »*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:**Caso um professor ou coordenador decida excluir um aluno do sistema, as informações dele devem ser removidas do banco de dados.

**[FA2] Fluxo Alternativo 2:**Caso um professor ou coordenador decida alterar uma informação de um aluno, as alterações devem ser registradas no banco de dados.

**[FA3] Fluxo Alternativo 3:** Caso um professor ou coordenador decida adicionar um aluno ao sistema, ele poderá cadastrá-lo no sistema, inserindo seu nome e e-mail institucional. Essa alteração deverá ser registrada no banco de dados.

**Pré-condições**

O sistema deve estar vinculado ao banco de dados MySQL e um coordenador ou professor já devem estar cadastrados no banco de dados.

**Pós-condições**

As alterações feitas devem ser salvas no banco de dados MySQL

**Regras de Negócio:**

[RN1] Apenas usuários registrados no sistema com a função de “Professor” ou “Coordenador” podem gerenciar os alunos.

**Especificação de Casos de Uso – Manter perguntas <<crud>>**

**Breve Descrição:**

*Os atores “Professor” ou “Coordenador” podem criar, ver, alterar e apagar perguntas do banco de questões do sistema.*

**Fluxo Básico**

Este caso de uso tem como objetivo consultar, incluir, alterar e excluir perguntas no sistema pelos atores coordenador e professor.

*1. Fazer o login no sistema como « Professor » ou « Coordenador »*

*2. Ir ao menu principal*

*3. Selecionar a opção « Gerenciar perguntas »*

*4. Gerenciar as perguntas do sistema*

*5. Clicar em « Salvar »*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:**Caso o usuário deseje criar uma pergunta, ele deve informar o enunciado, as alternativas incorretas, a alternativa correta e um feedback da alternativa correta.

**[FA2] Fluxo Alternativo 2:** Caso o usuário deseje apagar uma pergunta, ele deverá selecionar a opção “Apagar pergunta”, dentro da tela de gerenciamento das perguntas, e então clicar para excluir a questão desejada.

**[FA3] Fluxo Alternativo 3:** Caso o usuário deseje alterar uma pergunta existente, basta selecionar a opção “Alterar pergunta” dentro da tela de gerenciamento das perguntas, selecionar a pergunta que ele deseja alterar, e então alterá-la conforme desejado.

**Pré-condições**

O usuário deve estar cadastrado no sistema como “Professor” ou “Coordenador”.

**Pós-condições**

O banco de dados deverá salvar as alterações feitas pelo usuário.

**Regras de Negócio:**

[RN1] Apenas usuários registrados no sistema com a função de “Professor” ou “Coordenador” podem gerenciar as perguntas.

**Especificação de Casos de Uso – Manter checkpoints <<crud>>**

**Breve Descrição:**

Os atores “Professor” ou “Coordenador” podem adicionar checkpoints ao sistema, verificar quantos checkpoints existem e onde eles estão localizados, alterar a localização dos checkpoints já existentes e remover checkpoints presentes no sistema.

**Fluxo Básico**

Este caso de uso tem como objetivo consultar, incluir, alterar e excluir checkpoints no sistema pelos atores coordenador e professor.

*1. Fazer login no sistema como um « Professor » ou « Coordenador »*

*2. Ir ao menu principal*

*3. Clicar em « Gerenciar checkpoints »*

*4. Gerenciar os checkpoints*

*5. Clicar em « Salvar »*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:**Caso o usuário deseje criar um checkpoint, ele deverá informar o local que esse checkpoint estará localizado.

**[FA2] Fluxo Alternativo 2:**Caso o usuário deseje alterar um checkpoint existente no sistema, ele deverá informar qual checkpoint ele deseja alterar e onde esse checkpoint será colocado.

**[FA3] Fluxo Alternativo 3:** Caso o usuário deseje remover um checkpoint existente no sistema, ele deverá informar o local em que esse checkpoint está localizado.

**Pré-condições**

O sistema deve estar conectado ao banco de dados MySQL, o qual terá armazenado a quantidade de checkpoints e a localização de cada um.

**Pós-condições**

As alterações feitas serão registradas no banco de dados

**Regras de Negócio:**

[RN1] Apenas usuários registrados no sistema com a função de “Professor” ou “Coordenador” podem gerenciar os checkpoints.

**Especificação de Casos de Uso – Autenticar usuário**

**Breve Descrição:**

Os usuários podem se autenticar no sistema. Para isso, eles devem estar previamente cadastrados no sistema, e na autenticação o usuário deve informar o e-mail e a senha.

**Fluxo Básico**

Este caso de uso se inicia quando o ator aluno escolhe a opção de “login”.

1. *Abrir a tela de cadastro*
2. *Informar o e-mail e senha*
3. *Clicar em « Login »*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:** Caso os dados do usuário não coincidam com os dados presentes no banco de dados, o sistema deve informar a incoerência e solicitar os dados ao usuário novamente.

[FA2] Fluxo Alternativo 2: Caso o usuário não tenha um cadastro prévio no sistema, ele deve se cadastrar primeiro, informando o nome, e-mail e senha.

**Pré-condições**

O sistema deve estar vinculado ao banco de dados MySQL, o qual possuirá informações do cadastro de todos os usuários, permitindo a verificação de autenticação.

**Pós-condições**

O usuário, após se autenticar, terá acesso às funcionalidades do sistema.

**Regras de Negócio:**

[RN1] Os usuários devem se cadastrar no sistema antes de usá-lo.

**Especificação de Casos de Uso – Utilizar checkpoint**

**Breve Descrição:**

Os usuários, conforme forem acertando as questões, eventualmente chegarão aos checkpoints. Ao acertar a pergunta do checkpoint, ele será desbloqueado, servindo como uma espécie de garantia, que impede a perda total dos pontos em caso de erro de questões futuras.

**Fluxo Básico**

Este caso de uso se inicia quando o ator aluno erra uma questão.

*1. Acertar a questão do checkpoint*

*2. Errar uma questão futura*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:** Caso o usuário erre a questão do checkpoint, ele não terá acesso ao mesmo.

**[FA2] Fluxo Alternativo 2:** Caso o usuário acerte outra pergunta de checkpoint, o checkpoint anterior será descartado e o novo será atribuído em seu lugar.

**[FA3] Fluxo Alternativo 3:**Caso o usuário erre uma questão após desbloquear algum checkpoint, a pontuação do jogador será atribuída com base em seu último checkpoint.

**Pré-condições**

O sistema deve estar vinculado ao banco de dados MySQL, que contém o banco de perguntas e as respostas corretas e incorretas para cada pergunta. O banco de dados também deve conter informações a respeito da quantidade de checkpoints no jogo, da localização de todos os checkpoints e da atribuição de pontos por pergunta correta.

**Pós-condições**

O sistema terá armazenado a pontuação atual do usuário, assim como a pontuação indicada pelo último checkpoint que o usuário possui.

**Regras de Negócio:**

[RN1] Qualquer usuário pode utilizar os checkpoints, caso os tenha desbloqueado. Seu uso é feito automaticamente ao final do jogo, somente em caso de derrota.

**Especificação de Casos de Uso – Visualizar histórico de perguntas**

**Breve Descrição:**

O usuário poderá visualizar o seu histórico de perguntas. O histórico mostra todas as questões que o usuário já respondeu anteriormente, junto a um feedback para cada pergunta, mostrando qual é a alternativa correta de cada uma e mostrando também uma justificativa para a alternativa correta. No histórico também é possível ver qual alternativa o usuário colocou em cada questão que ele respondeu.

**Fluxo Básico**

Este caso de uso se inicia quando o ator aluno escolher a opção histórico de perguntas.

*1. Fazer o login*

*2. Jogar o jogo pelo menos uma vez*

*3. Voltar ao menu principal*

*4. Escolher a opção « Histórico de perguntas »*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:** Caso o usuário deseje sair do histórico, basta pressionar o botão “Sair”.

**[FA2] Fluxo Alternativo 2:** Dentro do histórico de perguntas, o usuário poderá filtrar as questões com base na matéria, ano ou dificuldade.

**Pré-condições**

1. O sistema deve estar vinculado ao banco de dados MySQL, o qual será responsável por armazenar todas as informações das questões respondidas pelo usuário (alternativa que o usuário marcou, alternativa correta da questão, matéria, ano e dificuldade). Todas essas informações devem ser mostradas no histórico de perguntas do usuário.

*2*. O usuário deve ter participado do jogo pelo menos uma vez, para que exista de fato um histórico de perguntas.

**Pós-condições**

O sistema deve apresentar o histórico de perguntas ao usuário.

**Regras de Negócio:**

[RN1] Apenas usuários cadastrados no sistema podem acessar o histórico de perguntas. Nenhum usuário pode acessar o histórico de perguntas de outro usuário, cada histórico é estritamente privado.

**Especificação de Casos de Uso – Visualizar ranking**

**Breve Descrição:**

O ranking poderá ser acessado apenas por usuários cadastrados no sistema como professor ou coordenador. O ranking exibe os alunos com maior pontuação do sistema, em ordem decrescente. A pontuação dos usuários é sempre atualizada ao final do jogo. Os professores e coordenadores podem filtrar o ranking por série e turma dos usuários, porém o ranking, em um primeiro momento, apresenta os usuários com as maiores pontuações do sistema independentemente de sua série e turma escolar.

**Fluxo Básico**

Este caso de uso se inicia quando o ator “Professor” ou “Coordenador” escolher a opção visualizar ranking.

*1. Fazer o login no sistema como « Professor » ou « Coordenador »*

*2. Acessar o menu principal*

*3. Escolher a opção « Visualizar ranking »*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:** O usuário poderá filtrar o ranking por série e/ou turma escolar, possibilitando uma análise mais específica.

**[FA2] Fluxo Alternativo 2:** Caso o usuário não esteja conectado no sistema como “Professor” ou “Coordenador”, a opção “Visualizar ranking” não aparecerá em seu menu principal.

**[FA3] Fluxo Alternativo 3:** Assim como o usuário pode filtrar o ranking, ele também poderá remover esse filtro, possibilitando uma análise geral dos usuários.

**Pré-condições**

Os usuários que desejam visualizar o ranking dos alunos devem estar cadastrados no sistema com a função de “Professor” ou “Coordenador

**Pós-condições**

O sistema irá mostrar o ranking para o usuário.

**Regras de Negócio:**

[RN1] Apenas usuários cadastrados no sistema como “Professor” ou “Coordenador” podem visualizar o ranking. Além disso, os usuários do sistema devem estar cadastrados no mesmo para que eles possam aparecer no ranking.

**Especificação de Casos de Uso – Responder perguntas**

**Breve Descrição:**

Qualquer usuário cadastrado no sistema é capaz de responder as perguntas do jogo. Para isso, basta clicar com o mouse em uma alternativa, e o sistema irá informar se a alternativa selecionada é está correta ou incorreta.

**Fluxo Básico**

Este caso de uso se inicia quando o ator “Aluno” clicar em uma alternativa.

*1. Fazer login no sistema*

*2. Clicar em « Jogar »*

*3. Ler a questão e clicar em uma alternativa*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:** Caso o usuário selecione a alternativa correta, sua pontuação aumentará e ele irá prosseguir com o jogo.

**[FA2] Fluxo Alternativo 2:** Caso o usuário selecione uma alternativa incorreta, o sistema irá informá-lo, calcular sua pontuação, mostrar ao usuário e irá encerrar o jogo. O usuário, no entanto, pode jogar novamente.

**Pré-condições**

As perguntas e suas informações (enunciado, alternativas corretas com explicação e alternativas incorretas) devem estar previamente registradas no banco de dados MySQL,

**Pós-condições**

O banco de dados irá informar ao usuário se ele acertou a questão ou não. Caso ele acerte, o jogo prosseguirá. Caso contrário, o jogo irá se encerrar, mostrando sua pontuação final.

**Regras de Negócio:**

[RN1] Qualquer usuário pode responder as perguntas do jogo, contanto que ele esteja previamente cadastrado no sistema.

**Especificação de Casos de Uso – Utilizar ajuda**

**Breve Descrição:**

O ator “Aluno” cadastrado no sistema pode, durante o jogo, solicitar ajuda. Existem 2 tipos de ajuda: “eliminar alternativas” e “pular questão”. As ajudas auxiliam o usuário no decorrer do jogo, possibilitando-o alcançar níveis maiores em sua jogatina.

**Fluxo Básico**

Este caso de uso se inicia quando o ator “Aluno” clicar em ajuda.

*1. Cadastrar-se no jogo*

*2. Ir ao menu principal*

*3. Clicar em « Jogar »*

*4. Clicar em « Ajuda »*

*5. Selecionar a ajuda que ele deseja*

**Fluxos Alternativos**

**[FA1] Fluxo Alternativo 1:** Caso o usuário escolha a ajuda “eliminar alternativas”, o sistema informará ao usuário 2 ou mais alternativas incorretas da questão em que a ajuda foi solicitada.

**[FA2] Fluxo Alternativo 2:** Caso o usuário escolha a ajuda “pular questão”, ele poderá avançar para a próxima questão sem precisar responder à pergunta.

**[FA3] Fluxo Alternativo 3:** Caso o usuário deseje utilizar a ajuda, mas ele não tenha mais ajudas disponíveis, o Sistema deverá informá-lo e nenhuma ajuda será utilizada.

**Pré-condições**

O sistema deve estar conectado ao banco de dados MySQL, o qual terá armazenado a quantidade de ajudas restantes e o tipo de cada uma delas.

**Pós-condições**

O sistema agirá de acordo com a ajuda solicitada e atualizará a quantidade de ajudas restantes.

**Regras de Negócio:**

[RN1]O usuário pode utilizar ajudas nas rodadas que ele desejar, menos na última rodada (pergunta do milhão). O usuário pode utilizar mais de uma ajuda na mesma pergunta. O usuário tem uma quantidade limitada de ajudas a serem utilizadas.

## 4.1. Diagrama de Casos de Uso

Gráfico, Diagrama, Gráfico de bolhas

O conteúdo gerado por IA pode estar incorreto.

## 4.2. Diagrama de Classes

Diagrama

O conteúdo gerado por IA pode estar incorreto.

## 4.3. Diagrama de Sequência

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.Diagrama, Desenho técnico

O conteúdo gerado por IA pode estar incorreto.

Diagrama, Desenho técnico

O conteúdo gerado por IA pode estar incorreto.

Diagrama, Esquemático

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

## 4.4. Modelo de Banco de Dados

Diagrama

O conteúdo gerado por IA pode estar incorreto.

## 4.5. Diagrama de Atividades (opcional)

Fluxo de processos do sistema, se necessário.

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

## 6. Testes

* Testes unitários, de integração e funcionais.
* Evidências como prints ou links para os testes.

## 7. Resultados e Considerações

* Prints das telas do sistema.
* Comparação entre requisitos e entrega final.
* Dificuldades enfrentadas e sugestões de melhorias.

## 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.