Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

ARTHUR PIROLO DE OLIVEIRA

GAMIFICAÇÃO DE TEORIA DOS GRAFOS

Assis/SP
2023

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

ARTHUR PIROLO DE OLIVEIRA

GAMIFICAÇÃO DE TEORIA DOS GRAFOS

Trabalho de conclusão de curso apresentado ao curso de Bacharelado em Ciências da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a):Arthur Pirolo de Oliveira
Orientador(a):Luiz Ricardo Begosso

Assis/SP
2023

Oliveira, Arthur Pirolo de

O482g Gamificação de teoria dos grafos / Arthur Pirolo de Oliveira. -- Assis, 2023.

48p. : il.

Trabalho de Conclusão de Curso (Ciência da Computação) -- Fundação Educacional do Município de Assis (FEMA), Instituto Municipal de Ensino Superior de Assis (IMESA), 2023.
Orientador: Prof. Dr. Luiz Carlos Begosso.

1. Matemática da computação. 2. Inteligência artificial aplicações educacionais. 3. Tecnologia educacional. I Begosso, Luiz Carlos. II Título.

CDD 006.68

GAMIFICAÇÃO DE TEORIA DOS GRAFOS

ARTHUR PIROLO DE OLIVEIRA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
DR. LUIZ RICARDO BEGOSSO

Examinador: _____
CÉLIO DESIRÓ

Dedico este trabalho á minha família, e a todos os meus amigos.

Assis/SP
2023

DEDICATÓRIA

AGRADECIMENTOS

Ao professor, Luiz Ricardo Begosso, pela orientação e pelo constante estímulo transmitido durante o trabalho.

RESUMO

A presente monografia tem como objetivo o desenvolvimento de uma gamificação com fundamento da matéria de teoria dos grafos, com a finalidade de auxiliar um entendimento essencial da área, para nichos como tópicos em rede e teoria da computação.

Palavras-chave: gamificação; Teoria dos Grafos; auxiliar; compreensão ; Divertimento no aprendizado;

ABSTRACT

Abstract: This monograph aims to develop a gamification based on graph theory, with the purpose of helping an essential understanding of the area, for niches such as networking topics and computation theory.

Keywords: gamification; Theory of graphs; assistant; understanding ; Fun in learning;

LISTA DE ILUSTRAÇÕES

Figura 1 :GRAFO COMPLETO, CONEXO E PLANAR....15
Figura 2 : MAPA MENTAL....19
Figura 3 : MODELO BANCO DE DADOS....20
Figura 4 : DIAGRAMA DE UC....21
Figura 5 : Menu....27
Figura 6 : Cadastro aluno....27
Figura 7 : Cadastro professor....27
Figura 8 : Cadastro de aula....28
Figura 9 : Envio mensagem....28
Figura 10 : Mostrar pontuação....29
Figura 11 : Fórum....29
Figura 12 : Escolha aula....30
Figura 13 : Resolver Aula....31
Figura 14 : Mensagem caso erro....31
Figura 15 : Mensagem Caso Acerto....32
Figura 16 : Visão geral Back-End....33
Figura 17 : FRAMEWORK JPA....34
Figura 18 : CLASSE JAVA....35
Figura 19 : Java Controller....36
Figura 20 : Java Repository....36
Figura 21 : application properties....36
Figura 22 : POM parte 1....37
Figura 23 : POM parte 2....38
Figura 24 : Visão geral Front-end....39
Figura 25 : Componente angular....40
Figura 26 : Model Angular....40
Figura 27 : app-routing....40
Figura 28 : Module....41
Figura 29 : PROXY_CONFIG....42
Figura 30 : Package.json....42
Figura 31 : Form professor....43
Figura 32 : Professor service....44
Figura 33 : Professor chamada http....44
Figura 34 : Atribuição de pontos para o aluno....45
Figura 35 : Atribuição de pontos....46

SUMÁRIO

1.INTRODUÇÃO.....10
1.2. OBJETIVO.....10
1.3. JUSTICATIVA.....11
1.4. MOTIVAÇÃO.....11
1.5. PERSPETIVAS DE CONTRIBUIÇÃO.....11
1.6. METOLOGIA.....11
2. ESTUDOS BIBLIOGRÁFICOS.....13
2.1. GAMIFICAÇÃO.....13
2.2. TEORIA DOS GRAFOS.....15
3. TECNOLOGIA UTILIZADAS.....17
4. MODELAGEM DO SOFTWARE.....19
4.1. LEVANTAMENTO DE REQUISITOS.....19
4.2. MAPA MENTAL.....20
4.3. MODELO BANCO DE DADOS.....21
4.4. DIAGRAMA DE CASO DE USO.....22
4.5. ESPECIFICAÇÃO DE UC.....22
4.5.1. Cadastro jogador.....23
4.5.2. Cadastro professor.....24
4.5.3. Cadastro lição.....25
4.5.4. Lição para resolução.....25
4.5.5. Listar pontuação.....26
4.5.6. Perguntas.....26
4.5.7. Respostas.....27
4.5.8. Fórum.....27
4.5.9. Enviar Mensagem.....27
4.6. PROTOTIPAÇÃO DE TELA.....28
5. PROJETO DO SOFTWARE.....33
5.1 BACK-END.....33
5.2 FRONT-END.....39
6. CONCLUSÃO.....47
REFERÊNCIA.....48

1.INTRODUÇÃO

Segundo Portal IDEA(2022), o conceito de gamificação vem sendo muito utilizado atualmente, pois ajuda no processo de aprendizagem de uma forma mais descontraída, e dando um senso de recompensa para que assim o aluno mantenha o interesse em continuar estudando, e devido ao grande interesse dos jovens em *games*, torna-se importante o desenvolvimento deste projeto.

Na ideia de BALDISSERA (2021), a gamificação é uma estratégia de aprendizagem, que se baseia na implementação de aspectos de vídeo game para ensinar e ajudar na compreensão. Para uma gamificação é necessário que haja progressão, ou seja, passar de nível ou etapas, pontuação e rankings, desafios que aumentem a dificuldade gradativamente.

Nos dias de hoje pode-se ver a falta de atenção dos alunos na sala de aula, o que se dá por diversos motivos como questões emocionais, problemas na família, alimentação inadequada, entre outros fatores. Isso é um problema, pois assim os alunos não conseguem abstrair totalmente o conhecimento dado em sala de aula (MATOS,2022).

Há diversos trabalhos usando a gamificação para ambientes acadêmicos como o jogo da segurança contra incêndios feito por SKILAB (2022),que ensina o que se deve fazer quando há um incêndio em ambientes de trabalho, porém com base nas pesquisas feitas para este trabalho não foi encontrado nenhum trabalho que seja baseado na disciplina de Teoria dos Grafos.O objetivo deste jogo é tornar mais divertido, e prender a atenção dos alunos enquanto estudam os conteúdos dessa matéria.

Para o desenvolvimento desse jogo será utilizado um banco de dados, para o *front-end* será utilizado o *framework* Angular, e para fazer o *back-end* será utilizado o Spring-Boot Java.

1.2 OBJETIVO

Essa monografia tem como objetivo desenvolver uma ferramenta gamificada sobre Teoria dos Grafos, para ajudar os alunos a compreender os conceitos fundamentais dessa disciplina, uma vez que este assunto é muito importante e relacionados com outras disciplinas como Tópicos em Redes e Teoria da Computação, tornando também a maneira de se apreender mais divertida e descontraída.

1.3 JUSTIFICATIVA

A presente monografia se justifica pelo desenvolvimento de uma ferramenta que possa auxiliar os alunos no estudo sobre Teoria dos Grafos, engajando os estudantes e tornando o aprendizado mais motivador. A disciplina de Teoria dos Grafos do curso de Bacharelado em Ciência da Computação foi escolhida, pois é o fundamento para os conteúdos de Tópicos em Redes e Teoria da Computação, o que a torna bem importante de se estudar.

1.4 MOTIVAÇÃO

A presente monografia tem por motivação desenvolver uma ferramenta gamificada que auxilie os usuários com a compreensão dos conceitos fundamentais da disciplina de Teoria dos Grafos, apoiando o aprendizado de outras disciplinas que são relacionadas com ela, como Tópicos em Redes e Teoria da Computação.

1.5 PERSPETIVAS DE CONTRIBUIÇÃO

A área de gamificação tem crescido bastante nos últimos anos, pois os jovens tem bastante interesse em jogos. Espera-se que esse trabalho possa contribuir com o processo de ensino de Teoria dos Grafos, aumentando o engajamento dos alunos e a motivação para o aprendizado.

1.6 METOLOGIA

A metodologia empregada para o desenvolvimento desta monografia consistirá de levantamento bibliográfico seguido da implementação de uma ferramenta em linguagem computacional. O trabalho está organizado da seguinte forma: Capítulo 1 é a Introdução onde estão descritos os objetivos e as justificativas do trabalho; o Capítulo 2 apresentará o levantamento bibliográfico sobre gamificação e teoria dos grafos; o Capítulo 3 descreve as tecnologias que serão utilizadas para o desenvolvimento deste projeto; o Capítulo 4 apresentará a modelagem do software a ser implementado; o Capítulo 5 apresentará as conclusões e propostas de desenvolvimento de trabalho futuro.

2. ESTUDOS BIBLIOGRÁFICOS

Neste capítulo, exploraremos de forma minuciosa as obras, artigos científicos, pesquisas acadêmicas e literatura relacionada que fundamentam e enriquecem a compreensão do tema central.

2.1 GAMIFICAÇÃO

Segundo SILVIO JUNIOR (2022),a gamificação envolve a aplicação de funções que fazem os usuários gostarem de jogos, como por exemplo, os pontos, autonomias, placares e os desafios e aplicá-las num ambiente que motive o jogador a utilizar determinada ferramenta.

Segundo SKINNER (2003), o conceito de recompensa é muito importante, para dar um sentido para continuar jogando o jogo e se o jogador for recompensado mesmo que de maneiras pequenas, ele vai se motivar a continuar aprendendo o conteúdo.

MARTINS (2022) diz que os jogos trazem consigo algumas vantagens, sendo elas: Acelera o ciclo de avaliação de desempenho, fazendo com que os funcionários se adaptem de uma forma mais rápida; deixa bem claro os objetivos e as regras dos jogos; Mostrar uma história a ser seguida; Divisão de grandes tarefas em tarefas menores para que as atividades sejam feitas de uma maneira satisfatória tanto para o usuário, quanto para o administrador. Ela diz também que a gamificação engaja, socializa, motiva, ensina e fideliza os jogadores.

Segundo a equipe TOTVS (2020), aplicar o conceito de gamificação na educação, faz com que os alunos fiquem mais interessados no estudo, pois os coloca como protagonistas de seu aprendizado, além de proporcionar mais motivação e engajamento para as atividades. Além disso, o uso de gamificação apresenta resultados mais eficientes do que os resultados em aulas tradicionais. Os benefícios de aplicar a gamificação são: feedback instantâneo; auxilia a desenvolver habilidades socioemocionais importantes, sendo elas a curiosidade, a criatividade, o raciocínio lógico e analítico, bem como sua capacidade de tomar decisões; gera engajamento na sala de aula, permitindo que as instituições educacionais criem ambientes de trabalho mais lúdicos e participativos; desperta o interesse dos alunos, tornando um exercício de um livro em um desafio lógico e lúdico; promove aulas mais dinâmicas, extrapolando o método tradicional de aprendizagem, que é passivo e sem dinamismo, deixando o aluno como espectador de palestras dos professores e as aulas focam mais na memorização. Com a gamificação, os

resultados são mais visíveis, já que possibilita o acompanhamento do progresso através de uma barra de progresso ou avanço de nível; traz resultados em todos os níveis de ensino, não importando se é um nível básico ou mais avançado. Exemplos de Gamificação:

- Duolingo, que utiliza da gamificação para ensinar aos seus assinantes outras línguas estrangeiras que os mesmos desejam, utilizando um sistema de recompensa para os usuários, quando fazem suas tarefas em dia, além de utilizar metas diárias para fazer com que os alunos voltem diariamente.
- Medieval Swansea, que utiliza a gamificação para ensinar história a seus usuários, utilizando um sistema de progressão, incluindo várias fases diferentes, e fases narrativas.
- Kahoot, que é um aplicativo de perguntas e respostas, podendo ser utilizado de forma mais aberta, ou seja, ele não tem uma especialidade, o professor cria a pergunta , e depois o aluno responde e dependendo do tempo em que ele demora e se ele responde corretamente, ele poderá ganhar uma certa pontuação; no final é apresentado o pódio com as maiores pontuações obtidas pelos participantes.
- Enem Game, que é uma gamificação focada no Exame Nacional do Ensino Médio, que permite a resolução de questões direcionadas, além de conter vídeo aulas, e também livros para a leitura.
- Perguntados, é uma gamificação que faz com que duas pessoas disputem em conhecimentos variados, recebendo pontos pelo tempo demorado para responder e se respondeu corretamente e no final da partida o vencedor receberá as parabenizações, e o ranking de pontos do final da partida fica disponível para compartilhar em suas redes sociais.

Segundo SILVIO JUNIOR(2022), usar um sistema de recompensas é um dos truques mais velhos do mundo do marketing, então se houver um sistema de recompensa, isso irá aumentar tanto a visibilidade do software como também a quantidade de pessoas que a gamificação ajuda.

Porém as recompensas também devem ser imprevisíveis, assim fazendo com que o jogador nunca saiba o que vai ganhar, o fazendo querer usar mais a gamificação para descobrir qual é a sua próxima recompensa.

2.2. TEORIA DOS GRAFOS

Ostroski (2009) diz que A Teoria dos Grafos é uma divisão da Matemática Discreta, a qual estuda a estrutura da matemática que não necessita obrigatoriamente da noção de continuidade, que estuda as estruturas denominadas grafos. O pioneiro desta teoria foi o matemático suíço Leonhard Euler, que elaborou e solucionou o problema de pontes de Königsberg, que consiste na seguinte situação, segundo Christina (2013): no século 18, havia sete pontes, na cidade prussiana de Königsberg, quatro delas ligavam a margens opostas a uma pequena ilha, outras duas ligavam as margens opostas a uma outra ilha, próxima à primeira, e a última ponte ligava as duas ilhas; o problema consiste em: sair do local onde está, atravessar todas as pontes uma só vez e regressar ao ponto de partida. Euler fez uma representação simplificada da cidade, sendo cada ilha e margem associadas aos vértices e cada ponte às arestas. Chegando à conclusão de que o grafo não admite um círculo euleriano, chega-se à solução que é impossível percorrer todas as pontes sem repetir nenhuma e retornar ao ponto de partida.

Além de Euler, outros nomes como Gustav Robert Kirchhoff, Arthur Cayley e Willian Rowan Hamilton foram muito importantes para a fundação da teoria dos grafos. Um grafo é representado pela expressão G = G(V,E), onde V é um conjunto discreto e ordenado dos vértices, e E é um conjunto de linhas que se chamam arestas, sendo que cada aresta está conectada em pelo menos um vértice.

Com a ideia de pontos interligados por linhas, a representação por grafos pode facilitar o estudo e a resolução de problemas. Enfim, mapas podem ser representados, assim facilitando para definição de rotas de transporte, redes de comunicação, distribuição de produtos e também em estruturas químicas de moléculas, que podem ser representadas por grafos. O estudo dos grafos vem crescendo nas últimas décadas devido a avanços tecnológicos computacionais, que permitem a resolução de problemas utilizando algoritmos, com maior eficiência, rapidez e confiabilidade.

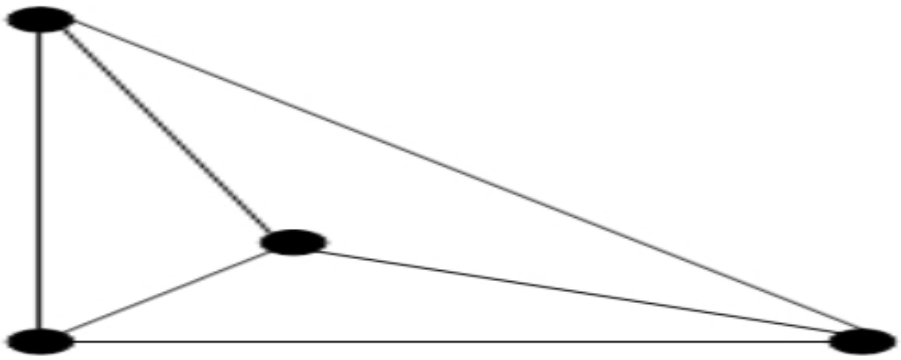

Figura 1:GRAFO COMPLETO, CONEXO E PLANAR
Fonte:https://pt.wikipedia.org/wiki/Teoria_dos_grafos

De acordo com Rabuske (1992), um grafo se define por G = G (V, E), que é uma estrutura construída pela ligação entre V e E, sendo V um conjunto discreto finito e não vazio, e E uma relação binária sobre V. Os componentes de V são representados com pontos, o par ordenado (v,w) ∈ E, onde v,w ∈ V, e as dá por uma linha ligando v a w.

Os componentes do conjunto E são arestas, linhas ou arcos do grafo, e são representados pelas letras minúsculas, podendo ser elas: a,b,c,d.

Os componentes do conjunto V são vértices, pontos ou nós dos gráficos, e são representados por letras minúsculas, podendo ser elas: e,f,g,h ou por números 1,2,3.

Os Grafos são classificados por suas características presentes neles, se tiver uma estrutura de laço ou aresta paralela é denominado multigrafo, se não tem laço ou aresta paralela ele é chamado de grafo simples. Quando o grafo pode ser dividido em dois, de forma que V1 com interseção com V2 for igual a vazio e cada vértice vi pertence ao conjunto V1 está unido a pelo menos um vértice vj pertence ao conjunto V2 , ele é bipartido.

A forma algébrica de representação de um grafo é uma matriz de tamanho n, e assim ele pode ser reconhecido em um algoritmo uma vez que existe a representação de uma matriz.

3. TECNOLOGIAS UTILIZADAS

Para o desenvolvimento deste projeto, para o front-end foi utilizado o typescript usando o Framework Angular/TS. Segundo Marylene Guedes (2021), é um framework javascript de código aberto, que é utilizado principalmente para fazer SPA (Aplicações de Página Única), que funciona basicamente de maneira simples.

A aplicação web é inteiramente construída em uma única página, assim a navegação e interação não necessita recarregar a página. Ele deixa a maneira de se produzir uma aplicação mais simples e otimizada, podendo também fazer aplicações de resolução *Desktop* e também aplicações mobile, fazendo-as dinâmicas, modernas e escaláveis. O Angular não utiliza a virtualização do DOM, ele usa mecânicas próprias de detecção de alteração de interface chamada *Two-Way Data Binding*, mantendo o *model* e a *view* sempre se comunicando, ou seja, sempre que se atualiza um dos dois, o outro é atualizado juntamente. O Angular ainda é baseado em javascript, mas através do TypeScript.

Para o back-end foi utilizado o Java por meio do framework Spring-boot. Segundo Melo (2021), Java é uma linguagem de programação que foi criada pela SunMicroSystems na década de 1990, que depois foi comprada pela Oracle. Diante da sua capacidade de usar o mesmo código em diferentes dispositivos, ele se tornou bem popular sendo usado em sites, computadores, celulares, calculadoras e até mesmo em videogames. O Java é baseado em classes e orientado a objetos, com segurança, portabilidade e alta performance. Porém, também conta com diversas bibliotecas de rotina e APIs para trabalhar com redes, e também um gerenciamento automático de memória. Para a interpretação do código, ele é transformado em um bytecode, que é interpretado e executado pela máquina virtual Java.

Segundo Lima (2021), o Spring Boot é um framework, que torna mais simples a criação de aplicações auto suficientes e robustas. Isso se dá graças aos esforços da plataforma Spring e também a bibliotecas de terceiros, que faz o tempo de configuração da aplicação ser o menor possível. As funcionalidades do Spring Boot são servidores web, gerenciamento de dependências, configuração automática de bibliotecas, funcionalidades para ambientes de produção, além de dispensar a necessidade de configuração XML. O Spring Boot também tem um mecanismo de empacotamento de aplicativos, permitindo a criação de arquivos JAR executáveis com todas as dependências que são necessárias, sendo amplamente usado para desenvolver aplicativos web, RESTful, micro serviços,

análise de dados, além de ser compatível com várias outras tecnologias Java, sendo algumas delas Spring Data, Spring Security.

Para o banco de dados foi utilizado o banco de dados H2, que é um banco de dados relacional de código aberto escrito em Java, projetado para ser rápido, eficiente, com uma utilização mínima de memória, fazendo com que ele seja escolhido popularmente para aplicações com o banco de dados incorporado. Ele suporta SQL além de ser um banco de dados em memória ou de persistência em disco, permitindo diversas conexões simultâneas, possibilitando que os desenvolvedores façam conexão no banco usando APIs padrões do JDBC. Ele é bem usado em aplicações Java, e também é popular no framework Spring.

Lombok é uma biblioteca para o Java, que simplifica o desenvolvimento, eliminando a necessidade de escrever getters, setters, construtores e métodos. Ele permite que os desenvolvedores adicionem anotações simples em classes Java.

4. MODELAGEM SOFTWARE

Neste capítulo, exploraremos em detalhes a tradução da concepção inicial do projeto em um modelo concreto que guiará a implementação prática do software. Através de diagramas, esquemas e descrições precisas, desdobraremos a estrutura e o funcionamento do sistema de forma a permitir uma compreensão abrangente de sua arquitetura e design.

4.1. LEVANTAMENTO DE REQUISITOS

Para o levantamento dos requisitos foi realizada uma entrevista com um professor, além de um estudo sobre recompensa para que as pessoas sintam-se com um objetivo para o uso da ferramenta.

Para a implementação deste projeto, os seguintes requisitos foram identificados:

- Ter um sistema de marcação de pontos.
- Um sistema para dar impressão de progresso.
- Ter uma recompensa por cada fase.
- Separação do game em fases que ficam mais desafiadoras.
- Sistema de placar para ver a evolução da competitividade.
- Seria interessante contar uma história com as fases.
- Relatório para ver quais alunos fizeram a lição.
- Sistema para aumento de complexidade de perguntas.

4.2. MAPA MENTAL

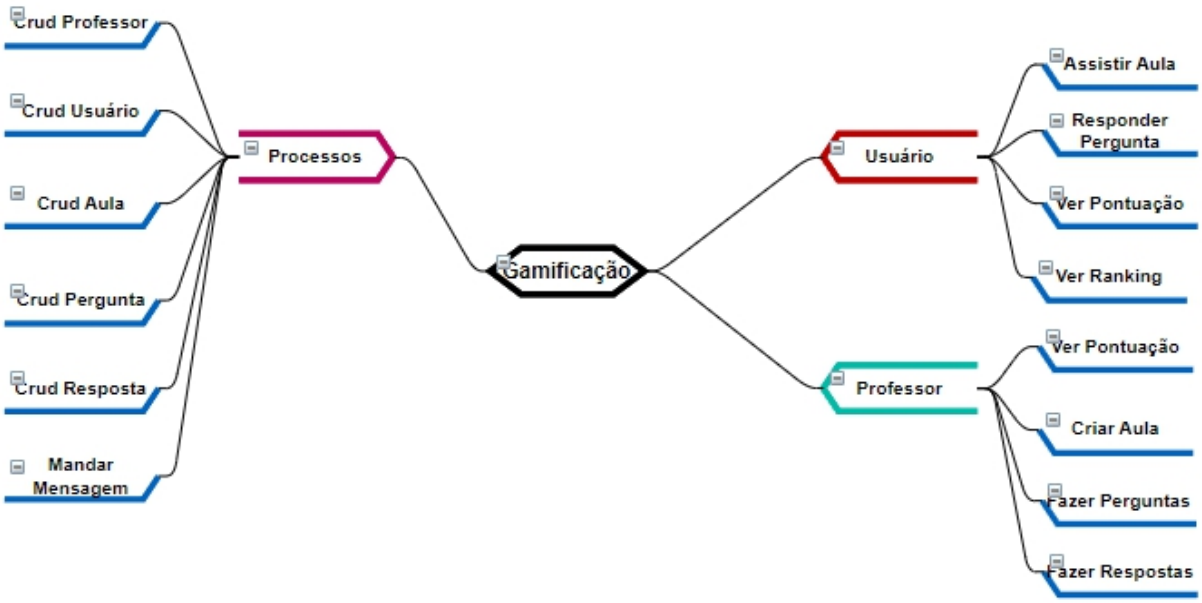

Figura 2: MAPA MENTAL

4.3. MODELO DE BANCO DE DADOS

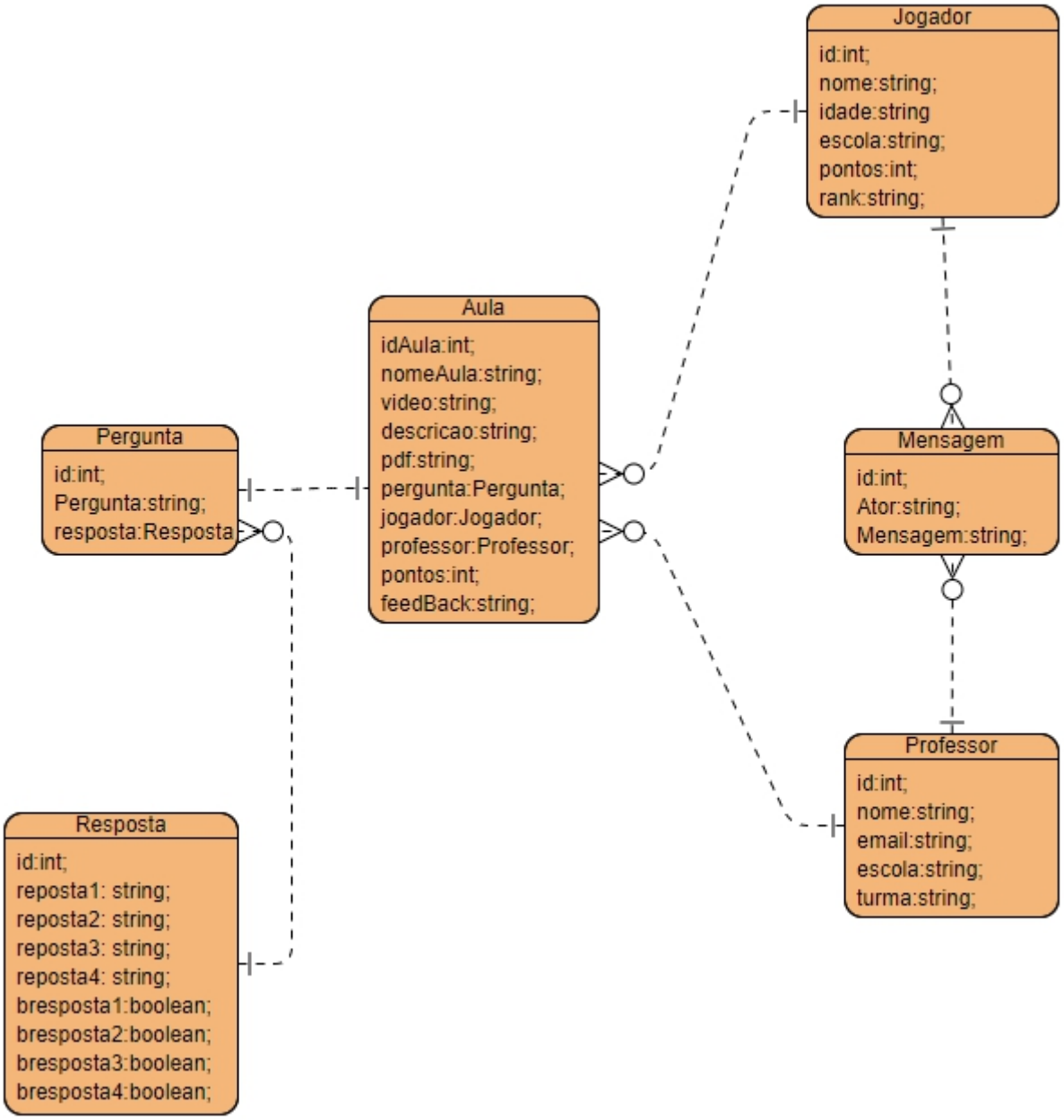

Figura 3: MODELO BANCO DE DADOS

4.4. DIAGRAMA DE CASO DE USO

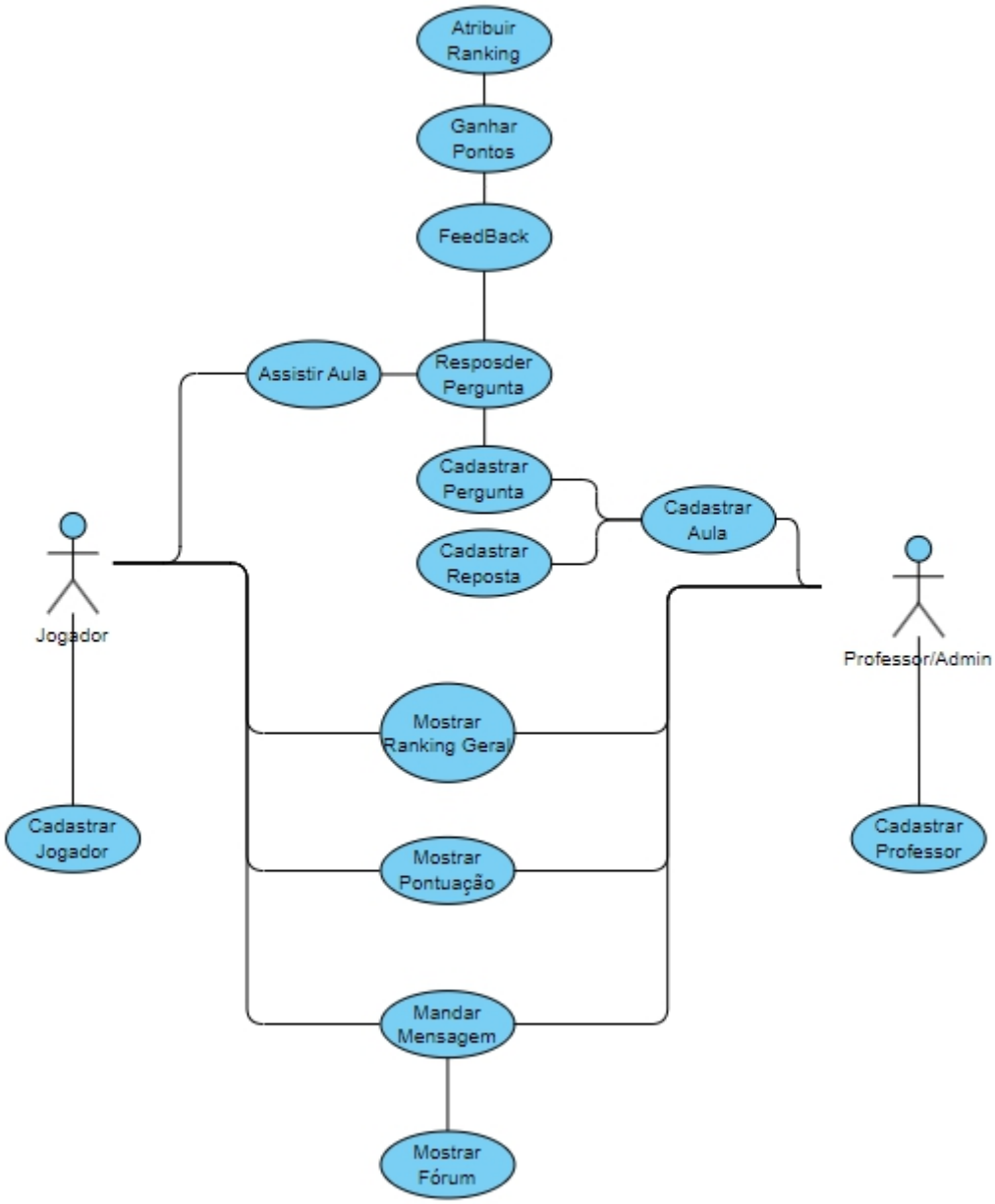

Figura 4: DIAGRAMA DE UC

4.5. ESPECIFICAÇÃO DE CASO DE USO

4.5.1 Cadastro jogador

1-Atores	Usuários, Administrador.
2-Funções	Criar, Deletar, Atualizar, Listar
3-Descrição	Criar, usuário cria uma conta para conseguir utilizar a gamificação por causa da pontuação da sua conta quando pressionado [a1], caso o usuário já esteja cadastrado [a2]. Deletar, o usuário pode também excluir sua conta quando desejar pelo botão [a3], pode cancelar a exclusão da conta [a4]. Atualizar o usuário pode também atualizar seus dados cadastrais clicando em “meu perfil” e depois clicando em atualizar dados. Listar, na pontuação, terá uma visualização do rank que o usuário está diante do outros [a1] , quando é o administrador ele pode ver todas as pessoas que estão inscritas no site.
4-Fluxo	A1- botão verde, quando pressionado será mandado o formulário para o back-end e sendo armazenado na tabela de usuários. A3-botão vermelho, quando pressionado será mandado para o back-end, sendo mandado a ordem de exclusão para armazenamento.
5-Fluxo Alternativo	A2-Será cancelado o envio para o banco de dados e voltará a mensagem “Usuário já cadastrado”. A4- Será cancelado o envio para o banco de dados.

4.5.2 Cadastro professor

1-Atores	Professor, Administrador.
2-Funções	Criar ,Deletar , Atualizar ,Listar
3-Descrição	Criar, professor cria para conseguir utilizar a gamificação, o professor deve ter o cadastro [**a1**] na mesma, para poder postar as atividades.Caso o professor já esteja inscrito [**a2**]**.** Deletar, o professor pode também excluir sua conta quando desejar pelo botão [**a2**], pode cancelar a exclusão da conta [**a3**]. Atualizar, o professor pode também atualizar seus dados cadastrais clicando em “meus perfil” depois clicando em atualizar dados. Listar, os professores inscritos.
4-Fluxo	A1- botão verde, quando pressionado será mandado o formulário para o back-end e sendo armazenado na tabela de professores. A3-botão vermelho, quando pressionado será mandado para o back-end, sendo mandado a ordem de exclusão para a tabela de professores.
5-Fluxo Alternativo	A2-Será cancelado o envio para o banco de dados e voltará a mensagem “Professor já cadastrado”. A4- Será cancelado o envio para o banco de dados.

4.5.3 Cadastro lição

1-Atores	Professor, Administrador.
2-Funções	Criar, Deletar ,Atualizar, Listar
3-Descrição	Criar, professor poderá criar [**a1**] lições para os alunos, que serão assistidas ou lidas e depois, uma pergunta será feita.Poderá cancelar com [**a3**]**.** Deletar, o professor poderá deletar uma lição [**a2**]. Atualizar, o professor poderá atualizar os dados da tarefa enviadas pelo mesmo. Listar, os Alunos inscritos, conseguiram ver as lições disponíveis.
4-Fluxo	A1- botão verde, quando pressionado será mandado o formulário para o back-end e sendo armazenado na tabela de lições. A2-botão vermelho, quando pressionado será mandado para o back-end, sendo mandado a ordem de exclusão para a tabela de professores.
5-Fluxo Alternativo	A3-Botão vermelho usado para cancelar qualquer uma das ações podem ser a criação, a deleção, e a atualização

4.5.4 Lição para resolução

1-Atores	Administrador, Aluno
2-Funções	Atribuir pontuação aos alunos, listar tarefas para os alunos.
3-Descrição	Atribuir pontuação, caso o aluno acerte a pergunta feita no final da aula, será atribuido uma pontuação pela dificuldade da pergunta [a1], caso o aluno erre a resposta fazer

	[a2]. As lições disponíveis ficaram visíveis aos alunos.
4-Fluxo	A1 - sendo elas iniciante explorador [0-99], aprendiz dos vértices[100-499], explorador de caminhos[500-999], Mestre das Arestas[1000 - a frente].
5-Fluxo Alternativo	A2- demonstrar a alternativa correta e retornar o feedback, do por que está errado.

4.5.5 Listar pontuação

1-Atores	Aluno
2-Funções	Mostra as pontuações de acordo com os dados da aplicação, e de acordo com os pontos o ranking do aluno é atribuído.
3-Descrição	Ao clicar em “pontuações” [a1] os alunos são listados de acordo com a pontuação e o ranking.
4-Fluxo	A1 - lista os alunos com suas pontuações e rankings.
5-Fluxo Alternativo	A2 - Retornar a pagina inicial.

4.5.6 Perguntas

1-Atores	Professor
2-Funções	Criar, Deletar, Listar.
3-Descrição	Fica no cadastro de aula, junto as respostas. Criar, para conseguir criar uma pergunta o professor deve ter o cadastro na gamificação, para poder postar as perguntas, clicar em cadastrar [a1] Deletar, o professor pode também excluir sua pergunta quando desejar pelo botão [**a2**]. Listar, mostra as perguntas inscritos.
4-Fluxo	A1- Cadastra a pergunta.
5-Fluxo Alternativo	A2 - Retornar a pagina inicial.

4.5.7 Respostas

1-Atores	Professor
2-Funções	Criar, Deletar, Listar.
3-Descrição	Fica no cadastro de aula, junto as perguntas. Criar, para conseguir criar uma resposta o professor deve ter o cadastro[a1] na gamificação, para poder postar as respostas.Ele pode voltar através do menu [a2] Listar, mostra as respostas inscritas.
4-Fluxo	A1 - cadastra a resposta
5-Fluxo Alternativo	A2 - Retornar a pagina inicial.

4.5.8 Fórum

1-Atores	Aluno, Professor
2-Funções	Listar
3-Descrição	Ao clicar no menu no “fórum”[a1], se houver a necessidade de voltar clicar em no menu onde deseja [a2].
4-Fluxo	a1- Lista as mensagens enviadas para o fórum, podendo ser tanto na relação aluno-professor como professor-aluno.
5-Fluxo Alternativo	a2- sai do fórum e vai a algum para onde o usuário deseja.

4.5.9 Mandar mensagem

1-Atores	Aluno, Professor
2-Funções	Listar
3-Descrição	Ao clicar, no link da navbar, “mandar mensagem”[a1], depois preencher as informações e clicar no botão “enviar mensagem”[a2]. Ao clicar, no link da navbar, “fórum”[a3]
4-Fluxo	a1- Abre a tela para preenchimento de informações. a2- Envia a mensagem para o fórum. A3-listas as mensagens do fórum
5-Fluxo Alternativo	

4.6. PROTOTIPAÇÃO DE TELA

Menu

Essa tela é o menu da ferramenta responsável pelo roteamento da mesma.

Figura 5: Menu

Cadastrar Alunos

Essa tela é a responsável pelo cadastro do aluno.

Os campos do aluno devem ser preenchidos com as informações e depois ao clicar no botão “salvar”, estará cadastrado, no momento do cadastro ele recebe os atributos ” 0” (zero) pontos sendo atribuído o rank “Iniciante explorador”.

Figura 6: Cadastro aluno

Cadastro Professor

Essa é a tela responsável pelo cadastro do professor, os campos do professor devem ser preenchidos com as informações e depois ao clicar no botão salvar, ele vai ser cadastrado.

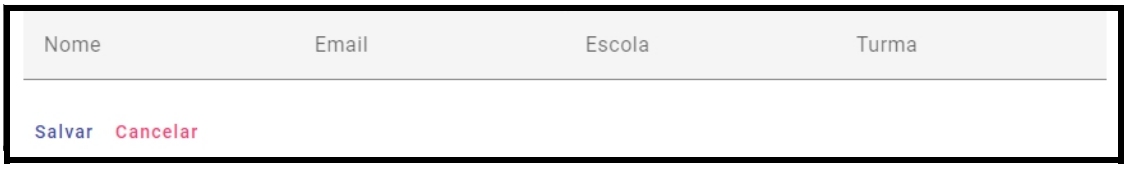

Figura 7: Cadastro professor

Cadastro Aula

Essa é a tela responsável pelo cadastro de aula, os campos devem ser preenchidos com as informações e depois ao clicar no botão salvar ,ela será cadastrada.

Os campos de respostas (1,2,3 e 4) aceitam apenas “sim” ou “não” e fazem correspondência direta na linha superior da aplicação.

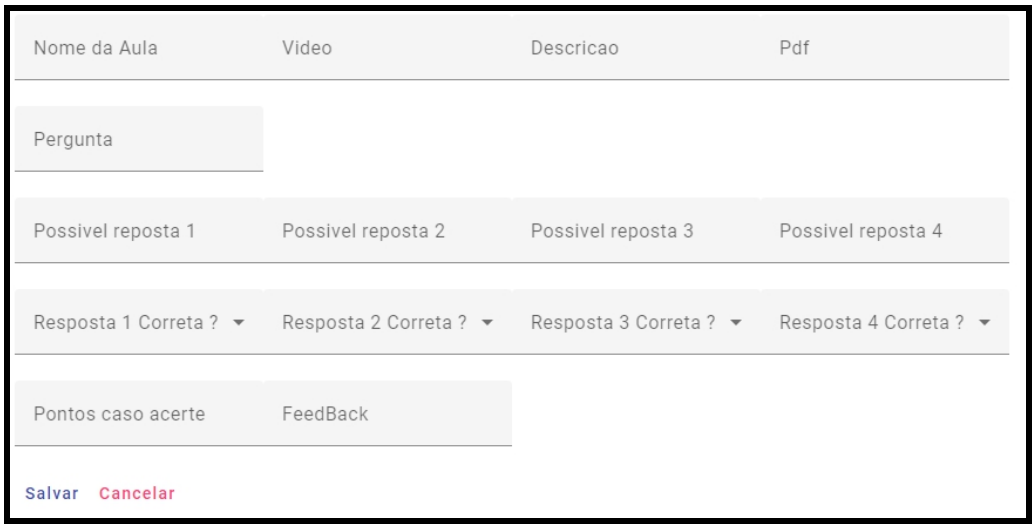

Figura 8: Cadastro de aula

Envio de mensagem

Essa tela é a responsável pelo envio de mensagens ao fórum, nela é colocado o ator responsável pelas mensagens que irão aparecer para os jogadores/professores.

Figura 9: Envio mensagem

Mostrar pontuação

Essa tela é responsável por mostrar a pontuação e ranking de cada jogador.

Name	pontos	Rank	
arthur	1000	Mestre das Arestas	recalcular
guilherme	500	Explorador de Caminhos	recalcular
lucas	500	Explorador de Caminhos	recalcular
gustavo	200	Aprendiz dos Vértices	recalcular
lione	25	Iniciante Explorador	recalcular

Figura 10: Mostrar pontuação

Fórum

Essa tela mostra as mensagens enviadas pelos jogadores/professores.

Figura 11: Fórum

Escolha Aula

Essa tela demonstra as aulas criadas pelos professores, ao clicar em “Entrar na aula”, a tela muda para resolver aula, aqui é demonstrado o nome , a descrição, e o total de pontos ganhos na aula caso acerte em sua totalidade.

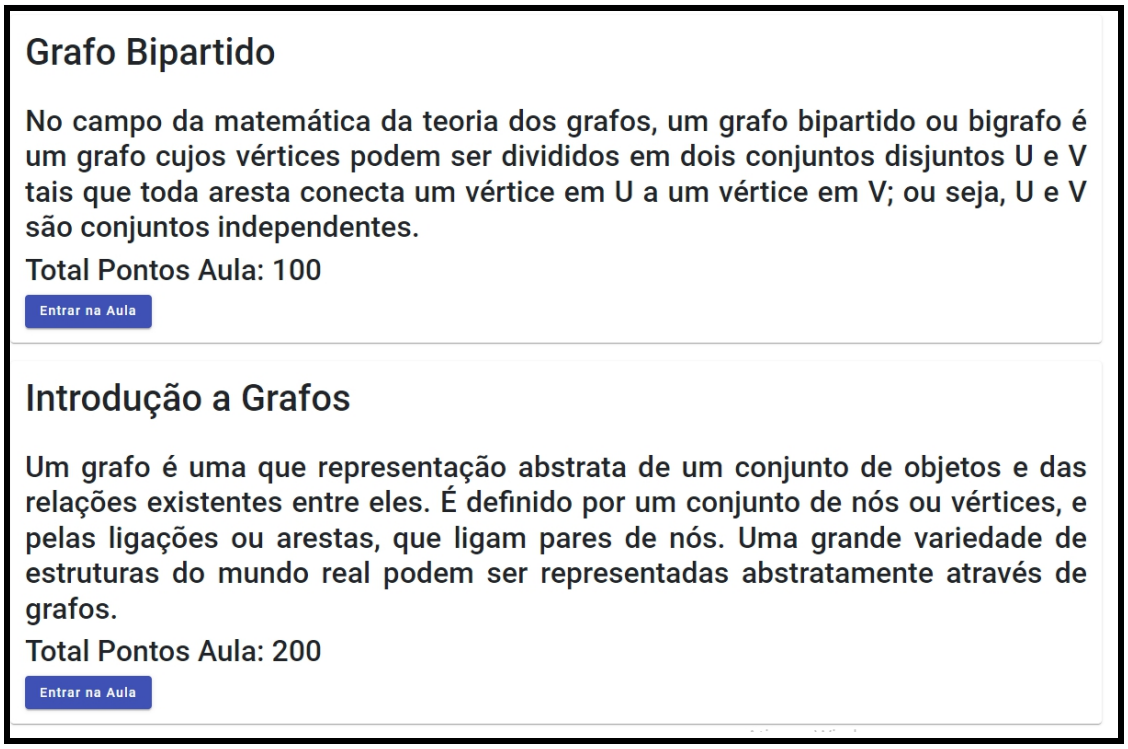

Figura 12: Escolha aula

Resolver Aula

Essa é a tela de resolução de tarefa, onde consta todos os atributos da aula, o aluno seleciona a alternativa que acredita estar correta, e pressiona o botão confirmar resposta, o feedback é retornado nos alerts abaixo. Caso erre aparece a figura 14, caso acerte figura 15.

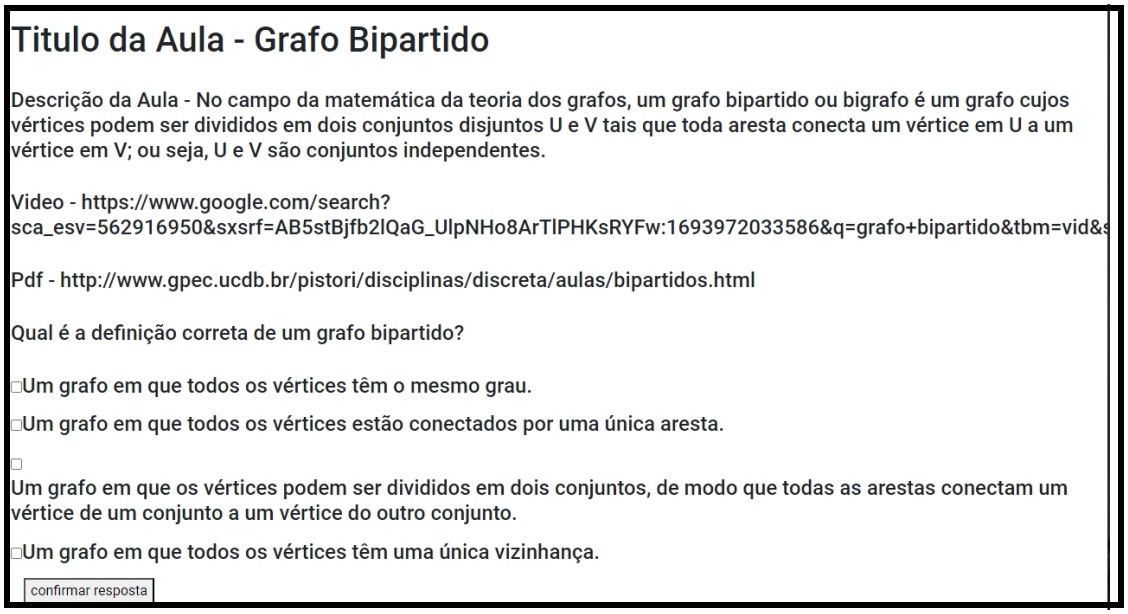

Figura 13: Resolver Aula

Mensagem Caso Erro

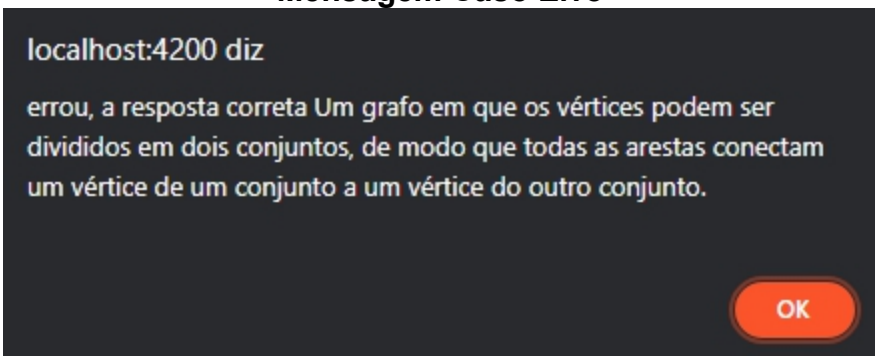

Figura 14: Mensagem caso erro

Mensagem Caso Acerto

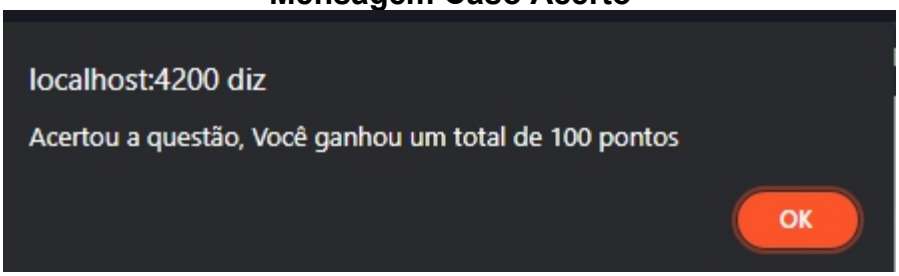

Figura 15: Mensagem Caso Acerto

5. PROJETO DE SOFTWARE

Como foi especificado no capitulo 3, a gamificação foi produzida a partir do Angular na parte do front-end, Spring-Boot Java e como banco de dados o banco H2.Essas tecnologias foram selecionadas devido à sua eficácia e adequação aos requisitos específicos do projeto, contribuindo para a criação de um ambiente de gamificação envolvente e funcional. Elas desempenharam papéis complementares na implementação bem-sucedida do software, proporcionando uma experiência aprimorada tanto para os desenvolvedores quanto para os usuários finais.

5.1 BACK-END

Visão Geral do Back-end ,figura 16:

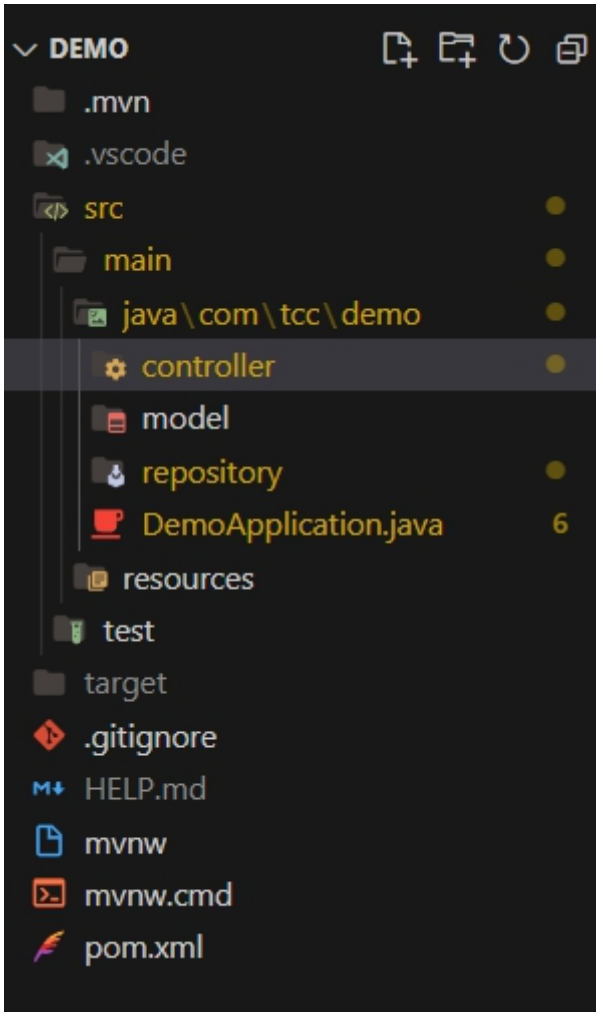

Figura 16: Visão geral Back-End

Na parte do *back-end* foi usado o *framework* JPA (Java Persistence API), que serve para facilitar o acesso ao banco de dados.

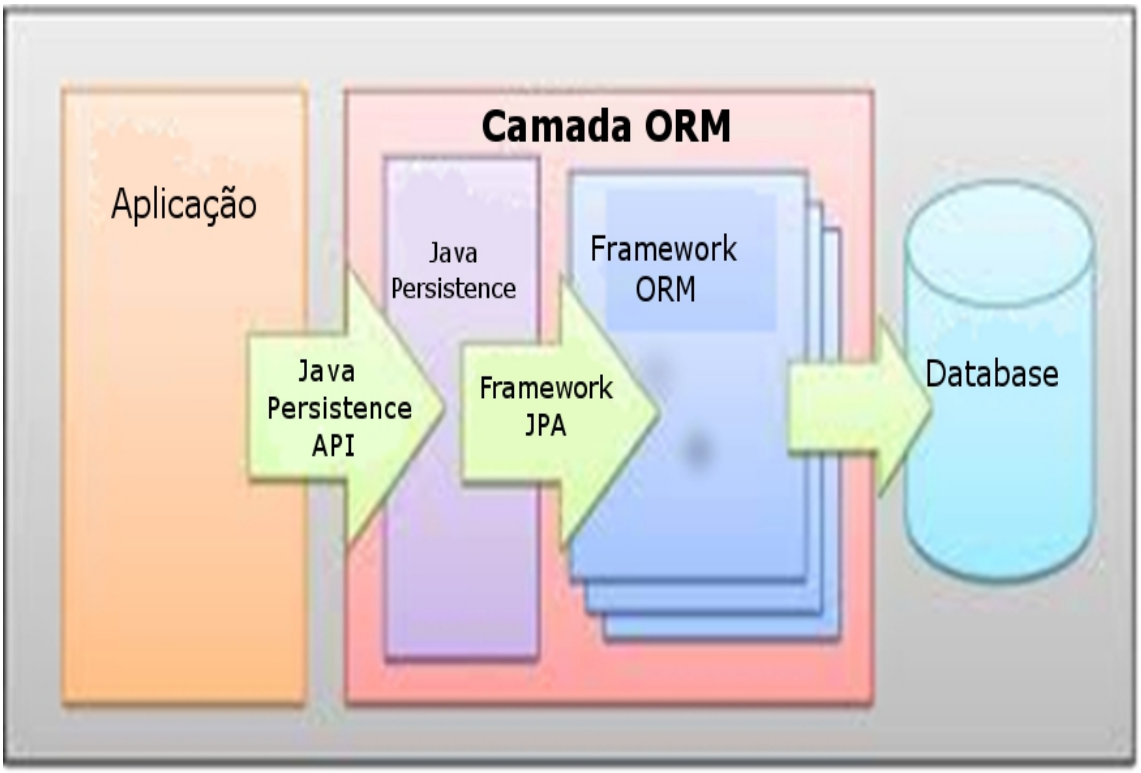

Figura 17: FRAMEWORK JPA
Fonte: Endel A

A anotação @Entity, como visto na figura 18, serve para demonstrar que o modelo de objeto é uma entidade, para sua criação no banco de dados, e para deixar mais simples a criação do banco de dados é usado a anotação @column(name="columns name"), que cria a coluna com o nome identificado pelo name.

A anotação ID significa que a coluna por ela ocupada tem a classificação como chave primária a anotação @GeneratedValue(strategy = generationType.IDENTITY), a para a geração do número do ID, com o strategy como parâmetro, que serve para falar como será a estratégia de geração nesse caso é pra seguir por ordem numérica dependendo do número de ID já existente.

Lombok é uma biblioteca que auxilia no desenvolvimento do projeto pois com ela é possível criar os getters e setters de maneira muito mais veloz usando anotações, já o @Data é uma anotação que cria getters e setters das variáveis do objeto. (getters pegar o valor e o setters por um valor)

```java
@Data
@Entity
public class Jogador {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @Column(name = "nome")
    private String nome;
    @Column(name = "idade")
    private String idade;
    @Column(name = "escola")
    private String escola;
    @Column(name = "pontos")
    private Number pontos;
    @Column(name = "rank")
    private String rank;
}
```

Figura 18: CLASSE JAVA

O projeto é estruturado da seguinte forma: há os controllers, models e repository, no qual cada um deles tem sua devida função.

Como é possível visualizar na figura 19, os **controllers** usados para a criação dos métodos como o Listar, Criar, Deletar, Atualizar. Nele há as seguintes anotações @RestController que serve para transformar o controlador em um especialista em solicitações RESTful, @RequestMapping serve para demonstrar em qual URL que a API vai estar, @AllArgsConstructor cria um construtor com todos os atributos de parâmetro, @CrossOrigin que serve para resolver o erro de CORS politic, @GetMapping demonstra que o método é de get.

```java
@RestController
@RequestMapping("/api/jogador")
@AllArgsConstructor
public class JogadorController {

    private JogadorRepository jogadorRepository;


    @CrossOrigin
    @GetMapping
    public List<Jogador> listJogador(){
        return jogadorRepository.findAll();
    }
```

Figura 19: Java Controller

O repository foi criado para abstração do banco de dados,além de criar um padrão de acesso a dados no banco de dados.

```java
@Repository
public interface AulaRepository extends JpaRepository<Aula, Long> {

}
```

Figura 20: Java Repository

A anotação @Repository, conforme demonstrado na figura 20 vem do JPA, serve para gerenciar a camada de persistência de dados, quando usado ela demarca que é um componente que lida com acesso de dados.

Conexão com o banco de dados por meio do application.properties:

```
application.properties ×
src > main > resources > application.properties
1  spring.datasource.url = jdbc:h2:mem:testdb
2  spring.datasource.driverClassName=org.h2.Driver
3  spring.datasource.username=sa
4  spring.datasource.password=password
5  spring.jpa.database-plataform=org.hibernate.dialect.H2Dialect
6  spring.jpa.show-sql=true
7  #http://localhost:8080/h2-console/
8
```

Figura 21: application properties

Como ilustrado na Figura 21, observa-se uma configuração detalhada da fonte de dados (datasource). Na primeira linha, está definida a URL da datasource, indicando o local onde o banco de dados está hospedado. Na segunda linha, é especificado o driver que será utilizado para se conectar ao banco de dados. A terceira linha identifica o nome de usuário que será utilizado para acessar o banco de dados, enquanto a quarta linha define a senha correspondente a esse usuário. Na quinta linha, é estabelecido o dialeto que será aplicado; neste caso, o H2Dialect. Por fim, a sexta linha contém uma instrução importante que determina que as consultas SQL sejam registradas e exibidas no console para fins de depuração e acompanhamento do comportamento do sistema. Essas configurações desempenham um papel fundamental na configuração e na conexão bem-sucedida com o banco de dados, garantindo o correto funcionamento do software.

Arquivo pom, em projetos maven é o responsável por definir plugins, frameworks,banco de dados, e várias outras coisas como lombok demonstrado nas figuras 22 e 23.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4      <modelVersion>4.0.0</modelVersion>
5      <parent>
6          <groupId>org.springframework.boot</groupId>
7          <artifactId>spring-boot-starter-parent</artifactId>
8          <version>3.1.1</version>
9          <relativePath/> <!-- lookup parent from repository -->
10     </parent>
11     <groupId>com.tcc</groupId>
12     <artifactId>demo</artifactId>
13     <version>0.0.1-SNAPSHOT</version>
14     <name>demo</name>
15     <description>Demo project for Spring Boot</description>
16     <properties>
17         <java.version>17</java.version>
18     </properties>
19     <dependencies>
20         <dependency>
21             <groupId>org.springframework.boot</groupId>
22             <artifactId>spring-boot-starter-data-jpa</artifactId>
23         </dependency>
24         <dependency>
25             <groupId>org.springframework.boot</groupId>
26             <artifactId>spring-boot-starter-web</artifactId>
27         </dependency>
28
29         <dependency>
30             <groupId>org.springframework.boot</groupId>
31             <artifactId>spring-boot-devtools</artifactId>
32             <scope>runtime</scope>
33             <optional>true</optional>
34         </dependency>
35         <dependency>
36             <groupId>com.h2database</groupId>
37             <artifactId>h2</artifactId>
38             <scope>runtime</scope>
39         </dependency>
```

Figura 22: POM parte 1

```
40          <dependency>
41              <groupId>org.projectlombok</groupId>
42              <artifactId>lombok</artifactId>
43              <optional>true</optional>
44          </dependency>
45          <dependency>
46              <groupId>org.springframework.boot</groupId>
47              <artifactId>spring-boot-starter-test</artifactId>
48              <scope>test</scope>
49          </dependency>
50          <dependency>
51          <groupId>org.springframework.boot</groupId>
52          <artifactId>spring-boot-starter-data-jpa</artifactId>
53          </dependency>
54      </dependencies>
55
56      <build>
57          <plugins>
58              <plugin>
59                  <groupId>org.springframework.boot</groupId>
60                  <artifactId>spring-boot-maven-plugin</artifactId>
61                  <configuration>
62                      <excludes>
63                          <exclude>
64                              <groupId>org.projectlombok</groupId>
65                              <artifactId>lombok</artifactId>
66                          </exclude>
67                      </excludes>
68                  </configuration>
69              </plugin>
70          </plugins>
71      </build>
72
73  </project>
74
```

Figura 23: POM parte 2

5.2 Front-End

Conforme foi especificado no capítulo 3, o *front-end* foi feito com o *framework* Angular, com o seguinte formato figura 24:

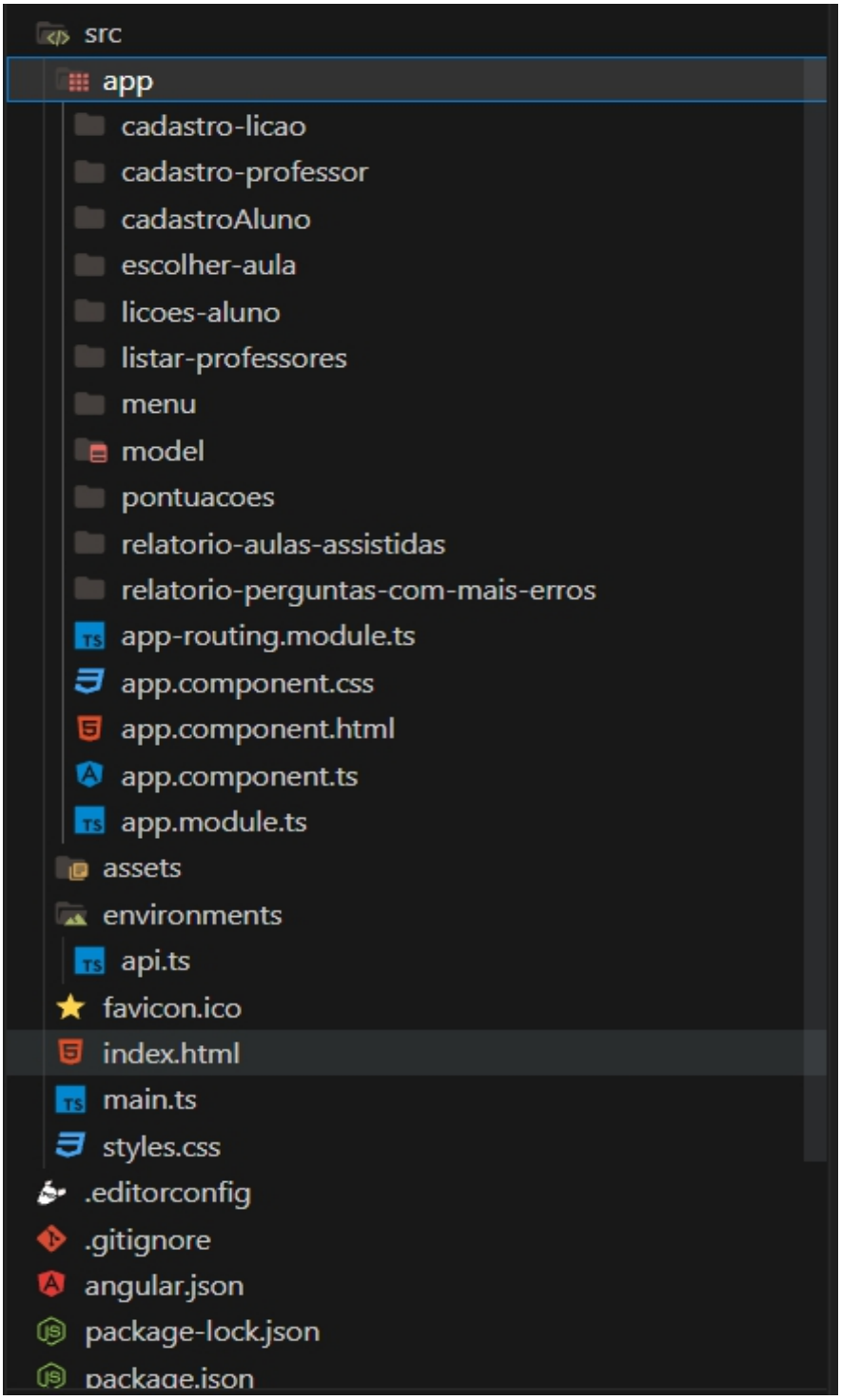

Figura 24: Visão geral Front-end

Na pasta App, como demonstrado na figura 25, há os componentes, que são formados por uma página html que é responsável pelas tags do html, uma css é a página de estilização, e uma em Typescript que é responsável por criar, listar, deletar e atualizar.

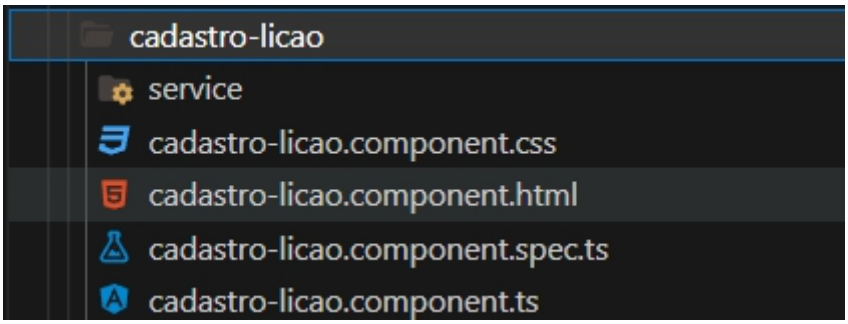

Figura 25: Componente angular

A Pasta Model é responsável pelos objetos como por exemplo a figura 26 o modelo de professor:

```
export interface Professor{
    idProfessor:number;
    nome:string;
    email:string;
    escola:string;
    turma:string;
}
```

Figura 26: Model Angular

No app-routing.module.ts figura 27,é feito o roteamento das páginas, porém é necessário importar o componente:

```
const routes: Routes = [
  { path: '', redirectTo: '/menu', pathMatch: 'full' },
  { path: 'menu', component: MenuComponent },
  { path: 'cadastroAluno', component: CadastroAlunoComponent },
  { path: 'cadastroProfessor', component: CadastroProfessorComponent },
  { path: 'cadastroLicao', component: CadastroLicaoComponent },
  { path: 'Pontuacoes', component: PontuacoesComponent },
  { path: 'LicaoAluno', component: LicoesAlunoComponent },
  { path: 'relatorioAula', component: RelatorioAulasAssistidasComponent },
  { path: 'relatorioPergunta', component: RelatorioPerguntasComMaisErrosComponent },
  { path: 'escolhaAula', component:EscolherAulaComponent},
  { path: 'listarProfessor', component:ListarProfessoresComponent}
];
```

Figura 27: app-routing

Porém além da configuração da rota nesse module também é necessário chamar o componente no app.component.html com a tag <router-outlet> que atua como um “buraco” a ser preenchido pela página que é chamada na url como por exemplo: www.localhost:4200/cadastroProfessor

No app.module.ts, serve para fazer importações de módulos que o projeto vai usar como por exemplo: HttpClientModule, que é responsável pelos chamados http como o get, post, delete, update, demonstrado na figura 28.

```
@NgModule({
  declarations: [
    AppComponent,
    MenuComponent,
    PontuacoesComponent,
    LicoesAlunoComponent,
    CadastroLicaoComponent,
    CadastroAlunoComponent,
    CadastroProfessorComponent,
    RelatorioPerguntasComMaisErrosComponent,
    RelatorioAulasAssistidasComponent,
    EscolherAulaComponent,
    ListarProfessoresComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule,
    BrowserAnimationsModule,
    MatTableModule,
    ReactiveFormsModule,
    MatFormFieldModule,
    MatInputModule,
    MatButtonModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Figura 28: Module

Proxy.conf.js, mostrado na figura 29, ele foi criado para resolver um problema de cors policy, assim criando uma conexão entre o front-end e back-end, assim eliminando o erro do cors, nesse arquivo é necessário mostrar a localidade onde o back-end está compilado, assim como a URL da API.

```
const PROXY_CONFIG = [
    {
        context:['/api'],
        target: ['http://localhost:8080'],
        secure: false,
        logLevel:'debug'
    }
];

module.exports = PROXY_CONFIG;
```

Figura 29: PROXY_CONFIG

Package.json, figura 30, tem os objetivos de gerenciamento de dependências, Scripts NPM, informações do projeto, Versões das dependências, publicação e distribuição. Como visto na imagem abaixo o ng serve tem como arquivo complementar o proxy.conf.js, que serve para iniciar sempre que dado o ng serve.

```
{
  "name": "gamificacao",
  "version": "0.0.0",
  ▷ Debug
  "scripts": {
    "ng": "ng",
    "start": "ng serve --proxy-config proxy.conf.js",
    "build": "ng build",
    "watch": "ng build --watch --configuration development",
    "test": "ng test"
  },
```

Figura 30: Package.json

Exemplo do component.html figura 31, ele foi feito usando as tags do html, além delas há também as propriedades do angular material.

```
<form [formGroup]="form">
    <mat-form-field>
      <mat-label>Nome</mat-label>
      <input matInput placeholder="Nome" formControlName="nome">
    </mat-form-field>

    <mat-form-field>
      <mat-label>Email</mat-label>
      <input matInput placeholder="Email" formControlName="email">
    </mat-form-field>

    <mat-form-field>
      <mat-label>Escola</mat-label>
      <input matInput placeholder="Escola" formControlName="escola">
    </mat-form-field>

    <mat-form-field>
      <mat-label>Turma</mat-label>
      <input matInput placeholder="Turma" formControlName="turma">
    </mat-form-field>
    <br>
    <button mat-button color="primary" (click)="onSubmit()">Salvar</button>
    <button mat-button color="accent" (click)="onCancel()">Cancelar</button>
</form>
```

Figura 31: Form professor

Exemplo do typescript figura 32, que serve como intermediário entre o html e o service, nele é feito a criação dos forms além de fazer a ligação do objeto, e os input a partir do formControlName.

```
@Component({
  selector: 'app-cadastro-professor',
  templateUrl: './cadastro-professor.component.html',
  styleUrls: ['./cadastro-professor.component.css']
})
export class CadastroProfessorComponent {
form:FormGroup;

  constructor(private formBuilder: FormBuilder,private cadastroProfessorService:CadastroProfessorService){
   this.form = this.formBuilder.group({
    nome:[null],
    email:[null],
    escola:[null],
    turma:[null],
   });
  }

  onSubmit(){
    this.cadastroProfessorService.save(this.form.value).subscribe(data => console.log(data));
  }
```

Figura 32: Professor service

Exemplo do Service, figura 33: utilizado como o serviço para a conexão entre o typescript e o back-end, como demonstrado é passado a URL da API, junto a URL do controlador do back-end, neste caso é o cadastro de professor.

```
export class CadastroProfessorService {

  constructor(private httpClient: HttpClient) { }

  save(data: Professor){
   return this.httpClient.post<Professor>(`${api.url}/api/professor`,data);
  }
}
```

Figura 33: Professor chamada http

Funcionalidade dos pontos, quando o aluno responde a pergunta de forma correta os pontos da aula em sua totalidade e também e mostrado um *alert* com a quantidade de pontos que ganhou ao responder, caso erro ele tem um número de quatro tentativas, porém a cada erro e tirado um ponto da quantidade de total de pontos, como é possível ver na figura 34.

```
if(this.resposta1selecionada === true){
  if(br1 === true){
    pontosAula = this.aula.pontos;
    console.log(br1);
    alert("Acertou a questão, Você ganhou um total de " + pontosAula + " pontos");
    this.aula.aluno.pontos = pontosAula;
  }
  if(br1 === false){
    alert("errou, a resposta correta " + this.aula.feedback);
    pontosAula = pontosAula - 1;
}
}

if(this.resposta2selecionada === true){
  if(br2 === true){
    pontosAula = this.aula.pontos;
    console.log(br2);
    alert("Acertou a questão, Você ganhou um total de " + pontosAula + " pontos");
    this.aula.aluno.pontos = pontosAula;
  }
  if(br2 === false){
    alert("errou, a resposta correta " + this.aula.feedback);
    pontosAula = pontosAula - 1;
}
}

if(this.resposta3selecionada === true){
  if(br3 === true){
    pontosAula = this.aula.pontos;
    console.log(br3);
    alert("Acertou a questão, Você ganhou um total de " + pontosAula + " pontos");
    this.aula.aluno.pontos = pontosAula;
  }
  if(br3 === false){
  alert("errou, a resposta correta " + this.aula.feedback);
  pontosAula = pontosAula - 1;
}
}
```

Figura 34: Atribuição de pontos para o aluno

A atribuição de rank é feito a partir da quantidade de pontos do jogador a regra defina a na figura 35.

```
ranks = [
  { nome: 'Iniciante Explorador', pontosMinimos: 0 },
  { nome: 'Aprendiz dos Vértices', pontosMinimos: 100 },
  { nome: 'Explorador de Caminhos', pontosMinimos: 500 },
  { nome: 'Mestre das Arestas', pontosMinimos: 1000 },
];

// Variável para armazenar o rank atual do usuário
rankAtual: string = 'Iniciante'; // Rank inicial padrão

// Lógica para determinar o rank com base nos pontos
determinarRank(jogador:Jogador) {
  for (let i = this.ranks.length - 1; i >= 0; i--) {
    if (jogador.pontos >= this.ranks[i].pontosMinimos) {
      jogador.rank = this.ranks[i].nome;
      break;
    }
  }
}
```

Figura 35: Atribuição de pontos

6.CONCLUSÃO

A gamificação é uma estratégia da aplicação de jogos nas atividades do dia a dia, que são utilizadas tanto na indústria, como na educação, ele necessita de certos elementos, como storytelling, desafios, recompensa, pontos, classificação, feedback. Ela está sendo amplamente usada hoje em dia, como na educação com Enem Game, Duolingo, Kahoot, e também é utilizado em empresas como na Microsoft, Cisco, Google, para melhor aprendizagem das pessoas. Essas ferramentas podem ser tanto simples como um jogo de perguntas e respostas, como mais complexas envolvendo a criação de uma historia e um objetivo final.

Conforme foi especificado no capitulo 2, a teoria dos grafos é uma ramo da matemática onde se estuda objetos de um determinado conjunto, os quais recebem o nome de grafos, G(V,E), onde V são os vértices e E são as arestas.

Contudo, a implementação da gamificação não é uma solução universal. É necessário considerar cuidadosamente o contexto educacional, os objetivos de ensino e as características dos alunos. Ainda assim, os resultados promissores deste estudo sugerem que a gamificação pode ser uma estratégia valiosa para tornar a teoria dos grafos mais acessível e atraente, contribuindo para uma aprendizagem mais profunda e significativa.

REFERÊNCIAS

ALVARO OSTROSKI ,LUCIA MENONCINI, Teoria dos grafos e suas aplicações. UTFPR, 2009. Disponível em <http://revistas.utfpr.edu.br/pb/index.php/SysScy/article/view/709/465>.Acesso em 12 março. 2023.

ANA CINTIA MATOS, Falta de interesse nos estudos. 2022. Disponível em <https://colegiocbv.com.br/escola-de-pais/falta-de- interesse-nos-estudos-por-que-isso-acontece-com-o-seu-filho/>. Acesso em: 18 outubro. 2022.

ANA LAURA GODINHO LIMA, O problema de falta de atenção na escola.

CLEBER CAMPOROMI, Precisamos falar sobre o typescript, 2018. Disponível em <https://www.treinaweb.com.br/blog/precisamos-falar-sobre-o-typescript/>

CLEYSON LIMA, O que é SpringBoot ? treina Web,2021. Disponível em <https://www.treinaweb.com.br/blog/o-que-e-o-spring-boot>

DIEGO MELO, O que é java ?, tecnoblog, 2021. Disponível em <https://tecnoblog.net/responde/o-que-e-java-guia-para-iniciantes/>

EQUIPE SEB, gamificação em sala de aula: saiba como funciona !.2022, Disponível em <https://novosalunos.com.br/gamificacao-sala-de aula/#:~:text=A%20gamifica%C3%A7%C3%A3o%20%C3%A9%20uma%20metodologia,os%20alunos%20interessados%20nas%20atividades.>. Acesso em: 07 novembro. 2022.

EQUIPE TOTVS, Gamificação da educação: vantagens, exemplos e como promover em sala de aula. 2020, Disponível em <https://www.totvs.com/blog/instituicao-de-ensino/gamificacao-na-educacao/>

ISABELA CHRISTINA QUEIROZ SOUZA, O PROBLEMA DAS PONTES DE Königsberg. UFPA, 2013. Disponível em <https://matematica.icen.ufpa.br/images/tccs/TCC_FACMAT_12.pdf>.

JORNAL DA USP, 2018. Disponível em <https://jornal.usp.br/artigos/o-problema-da-falta-de-atencao-na-escola/>. Acesso em: 18 outubro. 2022. BRUNO HOLANDA, Teoria dos Grafos. Portal IDEA. 2017, Disponível em <https://www.obm.org.br/content/uploads/2017/01/Nivel1_grafos_bruno.pdf>. Acesso em: 15 outubro . 2022.

LUCIANE MARIA, VANIA RIBAS ULBRICHT, CLAUDIA REGINA BATISTA, TRACÍSIO VANZIN, Gamificação educação,2014.

MARYLENE GUEDES, O que é Angular e para que serve ?,Treina web, 2021. Disponível em <https://www.treinaweb.com.br/blog/o-que-e-o-angular-e-para-que-serve>

NOEMI DEBORA. 12 benefícios de introduzir a gamificação na aprendizagem. 2019. Disponível em <https://escolasdisruptivas.com.br/steam/gamificacao-na-aprendizagem/>. Acesso em: 07 novembro 2022.

OLIVIA BALDISSERA, o que é gamificação e comom ela aumenta o engajamento,PÓSPUCPRDIGITAL, 2021. Disponível em <https://posdigital.pucpr.br/blog/gamificacao-engajamento>. Acesso em: 15 outubro . 2022.

Portal IDEA, O que é gamificação na educação e como ela pode mudar o segmento. Portal IDEA,2022. Disponível em <https://portalidea.com.br/blog/o-que-e-gamificacao-na-educacao>. Acesso em: 15 outubro . 2022.

RABUSKE, M.A. Introdução á teoria dos grafos. Florianópolis: UFSC, 1992.

ROMULO MARTINS, O que é gamificação. Blog Gupy,2022. Disponível em <https://www.gupy.io/blog/o-que-e-gamificacao>. Acesso em: 20 novembro. 2022.

SILVIO JUNIOR,BURRHUS FREDERIC SKINNER, Gamificação introdução e conceito básicos. 2020.

SKINNER, B. (2003). Ciência e Comportamento Humano. São Paulo – Livraria Martins Fontes Editora, Lda.

WPENSAR, gamificação em sala de aula: o que é e como aplicar de maneira eficaz, Disponível em <https://blog.wpensar.com.br/pedagogico/gamificacao-em-sala-de-aula/>.Acesso em 06 novembro. 2022.

ENDEL A,

https://www.google.com/search?q=JPA+(Java+Persistence+API)&sca_esv=556587111&sxsrf=AB5stBj83v5LkUHu-Yt-RS5miQsoE6DUaQ:1691963076312&tbm=isch&source=iu&ictx=1&vet=1&fir=k-2Qo6lUqRECCM%252CgQlGe7XHBW922M%252C_%253B58bvFM5Gm3Nc5M%252C8sxVh-sK7DuheM%252C_%253B2H649qNZEbbTZM%252C8sxVh-sK7DuheM%252C_%253Bvym3_WF3RGYsxM%252CWgSvh5OTPZNagM%252C_%253Bk_oL-Id1x57Z6M%252CXwJftiJBAT4iyM%252C_%253BVf_SEC5eThvmqM%252CVltKXQ8nBwEVIM%252C_&usg=AI4_-kSgL3JWxIKxLinK1_w4ojgXSqouuA&sa=X&ved=2ahUKEwjGxYvRzdqAAxVdpZUCHSKdAVgQ_h16BAhUEAE#imgrc=2H649qNZEbbTZM.