



MARCOS SILVEIRA BELLO

**CONSOLE: UM SISTEMA PARA BUSCA E CLASSIFICAÇÃO DE
JOGOS SÉRIOS**

LAVRAS – MG

2021

MARCOS SILVEIRA BELLO

CONSOLE: UM SISTEMA PARA BUSCA E CLASSIFICAÇÃO DE JOGOS SÉRIOS

Relatório técnico apresentado à Universidade Federal de Lavras, como parte das exigências do curso de Sistemas de Informação, para obtenção do título de Bacharel

Prof. DSc. Maurício Ronny de Almeida Souza

Orientador

LAVRAS – MG

2021

**Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos
da Biblioteca Universitária da UFLA**

Bello, Marcos Silveira

Console: Um sistema para busca e classificação de jogos sérios /
Marcos Silveira Bello. 2^a ed. rev., atual. e ampl. – Lavras : UFLA,
2021.

35 p. : il.

TCC–Universidade Federal de Lavras, 2021.

Orientador: Prof. DSc. Maurício Ronny de Almeida Souza.
Bibliografia.

1. TCC. 2. Monografia. 3. Dissertação. 4. Tese. 5. Trabalho
Científico – Normas. I. Universidade Federal de Lavras. II. Título.

CDD-808.066

MARCOS SILVEIRA BELLO

CONSOLE: UM SISTEMA PARA BUSCA E CLASSIFICAÇÃO DE JOGOS SÉRIOS

Relatório técnico apresentado à Universidade Federal de Lavras, como parte das exigências do curso de Sistemas de Informação, para obtenção do título de Bacharel

APROVADA em 07 de Maio de 2021.

Prof. DSc. Maurício Ronny de Almeida Souza	UFLA
Profa. DSc. Renata Teles Moreira	UFLA
Prof. DSc. Carlos dos Santos Portela	UFPA

Prof. DSc. Maurício Ronny de Almeida Souza
Orientador

LAVRAS – MG
2021

AGRADECIMENTOS

Agradeço à minha família por todo amor e apoio durante toda a minha vida. Agradeço ao meu amigo Raphael Henrique pela ajuda durante o desenvolvimento do projeto. Agradeço também aos amigos que fiz durante a graduação, especialmente ao Filipe Rezende Lima.

Agradeço aos professores Renata e Maurício por todo o apoio durante este trabalho. Agradeço também ao professor Carlos dos Santos Portela pela participação na banca e suas recomendações para este trabalho.

RESUMO

O uso de jogos para o ensino tem ganhado bastante destaque, sobretudo com a popularização de abordagens de ensino como o Ensino Baseado em Jogos (*Game-based Learning*, ou GBL). No ensino de computação não é diferente: diversos jogos sérios têm sido propostos para apoiar o ensino de diversos tópicos, desde a programação básica até a engenharia de software. No entanto, um dificultador para o uso do Ensino Baseado em Jogos é a dificuldade de encontrar, de forma centralizada, estes jogos e recomendações para o seu uso. Este relatório técnico tem como objetivo descrever a concepção, levantamento de requisitos e desenvolvimento de um sistema que teve como principal função centralizar a busca e classificação de jogos sérios. O sistema denominado Console, foi desenvolvido com *backend* em linguagem JAVA utilizando *framework spring boot* e o *frontend* utilizando o *framework vue.js*, o padrão de projeto foi o *MVC*.

Palavras-chave: Jogos sérios. Engenharia de software. Ensino baseado em jogos.

ABSTRACT

The use of games for teaching has gained a lot of prominence, especially with the popularization of teaching approaches such as Game-based Learning (GBL). In computing education it is no different: several serious games have been proposed to support the teaching of different topics, from basic programming to software engineering. However, a hindrance to the use of Game-Based Education is a difficulty in finding these games and recommendations for their use. This technical report aims to define definition, requirements identification and development of a system whose main function was to centralize the search and classification of serious games. The system called Console, was developed with *backend* in JAVA using *framework spring boot* and *frontend* using *framework vue.js*, the design pattern was *MVC*.

Keywords: Serious games. Software Engineering. Game Based Learn.

LISTA DE FIGURAS

Figura 3.1 – Exemplo de um arquivo tipo <i>pug</i>	20
Figura 3.2 – Exemplo de um arquivo tipo <i>pug</i> depois da transformado em <i>html</i>	20
Figura 3.3 – Diagrama da estrutura de um projeto MVC	21
Figura 4.1 – Tela de Cadastro de Usuário	24
Figura 4.2 – Tela de autenticação	25
Figura 4.3 – Tela de listagem	26
Figura 4.4 – Tela de Cadastro de jogos	27
Figura 4.5 – Tela de Cadastro de Relato de Uso	28
Figura 4.6 – Tela de Cadastro de Relato de Uso	28
Figura 4.7 – Tela de Visualização do Jogo	29
Figura 4.8 – Modal de Visualização do Relato de Uso	30
Figura 4.9 – Menu Lateral	31
Figura 4.10 – Menu Vertical	31

SUMÁRIO

1	INTRODUÇÃO	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Jogos sérios e Ensino Baseado em Jogos	17
2.2	Uso de jogos sérios no Ensino de Engenharia de Software	18
3	TECNOLOGIAS UTILIZADAS	19
3.1	<i>Backend</i>	19
3.2	<i>Frontend</i>	19
3.3	Padrão de projeto MVC	20
3.4	Git	21
4	Console	23
4.1	Concepção	23
4.2	Desenvolvimento o Sistema	23
4.3	Sistema desenvolvido	24
4.3.1	Registro do usuário	24
4.3.2	Autenticação do usuário	24
4.3.3	Listagem de jogos com filtro de busca	25
4.3.4	Cadastro do jogo	26
4.3.5	Cadastro do relato de uso do jogo	27
4.3.6	Visualização do jogo	29
4.3.7	Menu lateral	29
4.3.8	Menu Vertical	30
4.4	Limitações	30
5	CONCLUSÃO	33
	REFERÊNCIAS	35

1 INTRODUÇÃO

Ensino de engenharia de software é mais do que somente programação, inclui atenção a detalhes como qualidade, cronogramas e aspectos econômicos (Ardís et al., 2015). Além disso, profissionais de softwares tem a necessidades de lidar com problemas não técnicos como gerenciamento e trabalho em equipe (Ardís et al., 2015). Por causa disso é necessário ir além do formato padrão das aulas expositivas por não favorecer o aprendizado (Souza et al., 2017). Uma abordagem usada para superar esses desafios é o uso de jogos educacionais sérios para engenharia de software (Souza et al., 2017).

O *Game Based Learning* ou Ensino Baseado em Jogos consiste na inclusão de jogos no processo de ensino e aprendizagem. Apesar de serem ferramentas poderosas com muitos benefícios, os exemplos aplicados podem ser questionados e não foram rigorosamente definidos (von Wangenheim; Shull, 2009). Além disso jogos não são recomendados para todos os cenários, sendo melhores aplicados para reforço dos conceitos do que para apresentá-los. Na qual outros métodos se mostraram mais eficientes, como leitura (von Wangenheim; Shull, 2009).

Um exemplo de jogo é *Problems and Programmers* (BAKER; NAVARRO; HOEK, 2003) que um jogo de cartas educacional sobre engenharia de software que simula processos do desenvolvimento de software. O jogo apresenta um baralho de cartas físicas que representam processos no desenvolvimento de software, suas partidas são rápidas e através das partidas, cada uma delas acaba ilustrando aspectos do desenvolvimento de um sistema, por ser um jogo competitivo também, os estudantes acabam aprendendo uns com os outros.

Um dos problemas para aplicação desse método é a falta de informação sobre as aplicações desses jogos. Além da dificuldade de encontrá-los, porque estão espalhados por toda literatura científica (Rodrigues; Souza; Figueiredo, 2018). Outra dificuldade é encontrar recomendações sobre o uso de jogos para contextos específicos juntamente com avaliações sobre o uso destes jogos. Existem instrumentos padronizados para avaliação do uso de jogos sérios no contexto educacional, como o MEEGA+, que é baseado em um conjunto de atributos, obtidos através de um questionário usando fatores como a experiência do jogador e a usabilidade (PETRI; WANGENHEIM; BORGATTO, 2018). No entanto, não há um ambiente centralizador para busca de jogos sérios e relatos sobre o uso dos mesmos.

O objetivo deste trabalho é a concepção de uma ferramenta web para centralizar a busca e classificação de jogos sérios. O Console é uma ferramenta Web, projetada para cadastrar e buscar jogos sérios e

avaliações sobre seu uso, no contexto do ensino de Engenharia de Software. Entende-se que a existência de ferramenta de tal sorte pode contribuir para a adoção em maior escala dessas tecnologias educacionais.

Além deste capítulo, este documento está organizado nos seguintes capítulos. O Capítulo 2 apresenta a fundamentação teórica necessária para o entendimento dos conceitos usados neste trabalho. O Capítulo 3 descreve as tecnologias utilizadas no desenvolvimento do software proposto. O Capítulo 4 descreve a ferramenta Console. O Capítulo 5 traz as considerações finais deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica necessária para o entendimento dos conceitos utilizados neste trabalho. Mais especificamente, este capítulo apresenta conceitos relacionados a jogos sérios, ensino baseado em jogos e o uso dessas tecnologias no contexto do ensino de engenharia de software.

2.1 Jogos sérios e Ensino Baseado em Jogos

De acordo com von Wangenheim e Shull (2009) jogo pode ser definido como qualquer competição entre adversários(jogadores) operando sobre regras para um objetivo(vitória ou recompensa). Ainda de acordo com os autores, os jogos sérios, ou educacionais, são descritos como jogos pensados para ensinar certos assuntos, expandir contextos, reforçar desenvolvimento ou ajudar a desenvolver uma habilidade a medida que é jogado.

Super Mario Logic (PANEGALLI GILIANE BERNARDI, 2019) é um jogo sério para ensino de lógica de programação, o jogo é definido por comandos como andar para frente, pular ou abaixar. Os comandos são executados por meio das estruturas de dados de programação, o jogador deve escolher as mais adequadas para cada situação ou objetivo (PANEGALLI GILIANE BERNARDI, 2019).

Castelo dos enigmas (SCAICO DIEGO LOPES MARQUES, 2011) é um jogo sério também para ensino de lógica de programação, o jogo é definido por um castelo, onde jogador explora as salas e cada sala apresenta um desafio que representa uma estrutura de dados (SCAICO DIEGO LOPES MARQUES, 2011).

Ensino Baseado em Jogos, (do inglês *Game Based Learn*) é a aplicação destes jogos no contexto educacionais. É uma ferramenta poderosa de ensino, com muitos benefícios, como aumento do interesse, motivação e aumento da eficácia de aprendizagem (von Wangenheim; Shull, 2009). Porém, esta ferramenta não é recomendada para todos os cenários, como apresentar novos conceitos, onde outros métodos podem apresentar melhores resultados (von Wangenheim; Shull, 2009). Outro problema é que muitos estudos envolvendo aplicação desses jogos não apresentam informações satisfatórias sobre a efetividade (von Wangenheim; Shull, 2009), junto com a falta de recomendações e avaliações sobre aplicações dos jogos em diferentes contextos. Além disso, outra dificuldade apresentada está em buscar esses jogos, já que estão espalhados por toda literatura científica(Rodrigues; Souza; Figueiredo, 2018).

2.2 Uso de jogos sérios no Ensino de Engenharia de Software

Engenharia de software é uma área muito abrangente envolvendo aspectos que não são somente a programação, como qualidade, cronograma e parte econômica (Ardis et al., 2015). O importante desafio para ensino de engenharia de software, vem de sua natureza, que tem raízes na computação e surgiu como disciplina de engenharia e abrange tanto a teoria quanto a prática (Ardis et al., 2015). Estas características tem impacto direto na quantidade material que professores precisam cobrir durante a disciplina (Ardis et al., 2015).

Por conta do desafio prático da disciplina, vem sendo adotando outros métodos de ensino, como jogos sérios, para que estudantes tenha mais engajamento e entenda mais sobre a importância dos processos de engenharia de software (von Wangenheim; Shull, 2009). Dois exemplos de jogos sérios no contexto do ensino de engenharia de software:

SimSE é um jogo digital de simulação, cujo objetivo é permitir aos estudantes praticar um dos processo de engenharia de software simulado (ou subprocesso) em uma interface totalmente gráfica, interativa e divertida. Ele fornece feedback instantâneo e gráfico que permite aos estudantes aprender a complexa causa e efeito dos relacionamentos subjacentes aos processos de engenharia de software (NAVARRO; HOEK, 2009).

Pex4Fun é um jogo online jogado direto do navegador, que permite o usuário escrever, compilar e executar código para aprender conceitos de programação, praticar habilidades de codificação e analisar o comportamento do código de forma interativa. Apresenta desafios de quebra cabeças e duelo de códigos além de material de estudo adicional (TILLMANN; HALLEUX; XIE, 2011).

3 TECNOLOGIAS UTILIZADAS

O sistema foi desenvolvido como um sistema *web*, ou seja, projetado para ser utilizado através de um navegador na internet. O sistema pode ser dividido entre o *frontend*, por onde o usuário interage com sistema, visualizando e cadastrando informações, e *backend* onde as informações são processadas e salvas no banco de dados e disponibilizadas para o *frontend* via *HTTP*.

3.1 Backend

O *backend* é responsável pelo processamento das informações, recebendo e enviando, e dividido entre o banco de dados que o armazenamento e organização de informações sobre um domínio específico, ou seja, é o agrupamento de dados que abordam no mesmo assunto, e que precisam ser armazenados para segurança ou para consultas futuras. Para utilizar um banco de dados é necessário o uso de um sistema gerencial de banco de dados (SGBD) que é um *software* que disponibiliza recursos para manipular as informações presentes no banco de dados e interagir com o usuário. O SGBD utilizado para o desenvolvimento foi o PostgreSQL que é desenvolvido como projeto de código aberto e disponibiliza recursos para manipulação do banco de dados.

Já a outra parte do *backend* é a *API(Application Programming Interface)* que foi desenvolvido com o uso do *framework spring boot* para a plataforma Java que facilita o desenvolvimento disponibilizando várias ferramentas como acesso ao banco de dados através do mapeamento objeto-relacional, segurança, como o uso do *JWT(JSON Web Token)* e o envio de informações via *HTTP*.

3.2 Frontend

Frontend é a parte visual da aplicação, na qual se interage, visualizando ou cadastrando dados. O desenvolvimento foi feito utilizando o *vue.js* que é um *framework javascript* de código-aberto criado pelo Evan You, e foi lançado em fevereiro de 2014 ¹. Ele permite a criação de componentes que podem ser reutilizados, desse jeito, simplificando o desenvolvimento e facilitando a criação de páginas web. Dentro do uso desse *framework* foi utilizado a *template engine pug* ² que facilita o desenvolvimento das páginas *HTML* não sendo necessário o uso das *tags*, deixando o código mais limpo e compreensível.

¹ <https://vuejs.org/>

² <https://pugjs.org/api/getting-started.html>

Figura 3.1 – Exemplo de um arquivo tipo *pug*

```

1 doctype html
2 html(lang='pt')
3 head
4   title Pug
5 body
6   h1 hello world with pug
7   div.container
8     p pug is very simple!
9

```

Figura 3.2 – Exemplo de um arquivo tipo *pug* depois da transformado em *html*

```

1 <!DOCTYPE html>
2 <html lang="pt">
3 <head>
4   <title>Pug</title>
5 </head>
6 <body>
7   <h1>hello world with pug</h1>
8   <div class="container">
9     <p>pug is very simple!</p>
10  </div>
11 </body>
12 </html>
13

```

Além disso, foi utilizado o *Element*³ uma biblioteca auxiliar no desenvolvimento da interface, padronizando o estilo *css*, e disponibilizando componentes já prontos, deste modo não é necessário criar elementos visuais do zero, gerando uma interface mais consciente e agradável. Porém uma das limitações é que o sistema não é compatível para ser aberto em navegadores *mobile*.

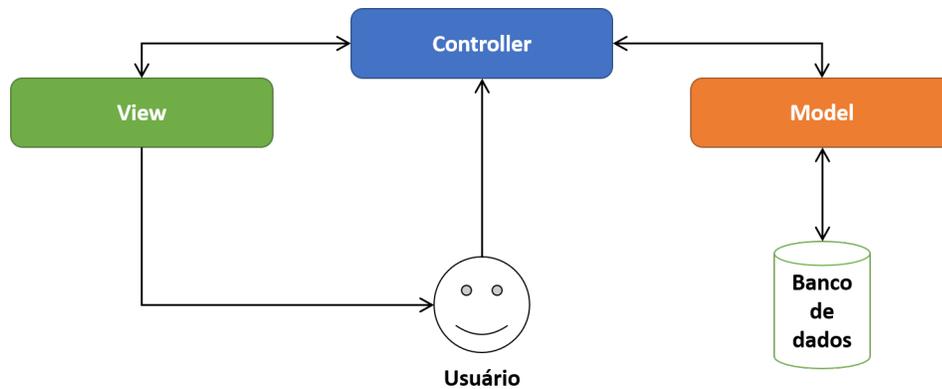
3.3 Padrão de projeto MVC

O padrão *Model-view-control* (MVC) é um padrão de projeto de software formulado na década de 1970, criado na Xerox Parc, por Trygve Reenskaug, focado no reuso de código e a separação de conceitos em três camadas interconectadas, onde a apresentação de dados e a interação do usuário (*front-end*) são separadas dos meios que interagem com banco de dados (ALVE, 2011). O projeto aplicado ao padrão foi dividido aplicado da seguinte forma, *backend* responsável pela camada *model*, que são as *services* do projeto, onde estão aplicadas as regras de negocio para leitura e escrita de dados, pela camada *controller*, que são

³ <https://element.eleme.io/en-US>

as *controllers* do projeto, onde estão as requisições do usuário, disponibilizando os dados via *HTTP*. Já a camada *view* é responsabilidade do *frontend* onde há a representação visual para o usuário.

Figura 3.3 – Diagrama da estrutura de um projeto MVC



3.4 Git

O Git é um sistema de versionamento de código fonte aberto que permite que múltiplos colaboradores trabalhem em um mesmo projeto ao mesmo tempo (MICROSOFT, 2018). O Git tem um fluxo de trabalho de contribuição e uma terminologia exclusivos para dar suporte ao modelo distribuído. Por exemplo, não há um bloqueio de arquivo normalmente associado a operações de check-out/check-in. Na verdade, o Git se preocupa com alterações em um nível ainda maior, comparando arquivos byte por byte (MICROSOFT, 2018).

O Git usa uma estrutura em camadas para armazenar e gerenciar o conteúdo de um projeto. O repositório é a maior unidade de armazenamento, sendo que, um repositório pode ter uma ou mais *branches*. O *branch* é uma unidade de armazenamento que contém os arquivos e pastas que compõe o projeto (MICROSOFT, 2018).

Os *branches* separam fluxos de trabalho (normalmente conhecidos como versões), assim as 18 contribuições sempre são feitas para um *branch* específico e estão no escopo do mesmo. Todos os repositórios contêm um *branch* padrão (normalmente chamado “master”) e um ou mais *branches* que serão mesclados novamente no *branch* master. O *branch* master funciona como a versão atual e “única fonte de verdade” para o projeto, sendo que todos os outros *branches* no repositório são criados dele (MICROSOFT, 2018). Os colaboradores interagem com o Git para atualizar e manipular repositórios nos níveis locais (ferramentas

como o console Git Bash) ou através do GitHub (serviço de hospedagem na Web para repositórios git) (MICROSOFT, 2018).

4 CONSOLE

Este capítulo apresenta informações relevantes da concepção, requisitos e desenvolvimento do sistema Console.

4.1 Concepção

O sistema foi concebido para atender a demanda de centralizar jogos sérios, para isso foi necessário entender as informações principais dos jogos e porque essas informações são relevantes e como elas seriam registradas e exibidas. Além disso, como seria feito o relacionamento dos jogos com as outras partes do sistema.

O sistema também foi concebido para registrar relatos de uso de um jogo para que outras pessoas possam entender o resultado da aplicação de um jogo em determinado contexto, e descobrir qual jogo pode se adaptar melhor ao contexto necessário baseado nestes relatos.

Por isso foi preciso compreender quais dados de um relato são relevantes, como isso se relacionaria com o jogo e outras partes do sistema. Além disso foi necessário escolher uma ferramenta para avaliação do relato de uso para que houvesse uma padronização dos relatos, com isso foi utilizado a (*MEEGA+*).

A escolha para sistema Web foi realizada para que o sistema fosse mais amplamente utilizado, sem a necessidade o uso de um sistema operacional próprio ou a utilização de outra tipo de ferramenta, sendo necessário somente o uso de um navegador e acesso a internet.

O nome da ferramenta console foi escolhida para remeter a um console de jogos, ou seja, uma plataforma na qual se tem uma grande biblioteca de jogos disponíveis.

4.2 Desenvolvimento o Sistema

Para o desenvolvimento desse sistema foi utilizado o padrão de projeto *MVC*, que separa aplicação em camadas (bancos de dados, *frontend* e *backend*). Foi criado um banco de dados utilizando o Sistema de Gerenciamento de Banco de Dados (SGBD) PostgreSQL, para salvar as informações do sistema. O PostgreSQL foi escolhido por ser projeto de código aberto e pela familiaridade do autor com o SGBD.

A linguagem escolhida para desenvolvimento *backend* foi o Java, utilizando o *framework Spring Boot* já que o mesmo apresenta inúmeras ferramentas que auxiliam no desenvolvimento, como a de segurança, acesso ao banco de dados e o envio de informações via HTTP. Para o *frontend* foi utilizado o *framework Vue* com *Element*, um template, que auxilia no desenvolvimento, não sendo necessário a criação

de elementos visuais do zero além de se ter uma UI mais agradável. O repositório do código do sistema se encontra no GitHub, onde a *branch* develop se encontra com a versão em desenvolvimento, já a *branch* master se encontra com a versão do código já testada e homologada.

4.3 Sistema desenvolvido

As subseções a seguir descrevem os principais requisitos funcionais desenvolvidos e presentes na versão final do projeto.

4.3.1 Registro do usuário

A tela de registro do usuário é responsável pelo cadastro de usuário, mostrado na Figura 4.1, a tela apresenta os seguintes campos, nome, a data de nascimento, a filiação (instituição de ensino), o *e-mail* (que deve ser único no sistema), e senha (que deve ter no mínimo 8 dígitos). Caso o e-mail indicado não seja único, o sistema mostrará uma mensagem.

Figura 4.1 – Tela de Cadastro de Usuário

A imagem mostra a interface de usuário para o cadastro de um novo usuário. O formulário é intitulado "Cadastro" e contém os seguintes campos obrigatórios:

- Nome:** Campo de texto com ícone de pessoa.
- Data de Nascimento:** Campo de data com máscara "dd/mm/aaaa" e ícone de calendário.
- Universidade:** Campo de texto com ícone de pessoa.
- Email:** Campo de texto com ícone de envelope.
- Senha:** Campo de texto com ícone de cadeado.

Na base do formulário, há dois botões: "Cadastrar" (em azul) e "Cancelar" (em cinza).

4.3.2 Autenticação do usuário

A tela de autenticação de usuário é responsável pela autenticação do usuário, como mostrado na Figura 4.2, permitindo acesso a outras funções do sistema, como cadastro de jogos(ver Seção 4.3.4) ou relatos de uso (ver Seção 4.3.5).

A tela tem dois campos, como mostrado na Figura 4.2, um para *e-mail* e outro para senha, sendo os dois obrigatórios. Caso o *e-mail* ou senha não estejam cadastrados, o sistema exibirá uma mensagem de falha na autenticação por causa do usuário e/ou senha inválidos.

Figura 4.2 – Tela de autenticação

A imagem mostra a interface de autenticação do sistema, intitulada "/Console_". Ela contém dois campos de entrada: "Email" com um ícone de envelope e "Senha" com um ícone de cadeado. Abaixo dos campos, há um botão azul "Entrar" e um botão cinza "Cadastrar".

4.3.3 Listagem de jogos com filtro de busca

A tela de listagem é a inicial do sistema, como mostrado na Figura 4.3, que exibe uma tabela com os jogos cadastrados no banco de dados, exibindo quatro informações básicas, nome do jogo, licença, tipo e a nota. A licença do artigo que refere-se a permissão de acesso e distribuição. O tipo é se o jogo é digital ou físico. A nota é a média das notas dos relatos de uso.

A tela também apresenta um filtro de busca com os seguintes campos: nome do jogo, tipo de jogo, estilo de jogo e áreas de ensino. Nome do jogo é um campo de tipo texto, quando buscado qualquer nome semelhante será retornado. Tipo de jogo é um campo do tipo seleção com três opções, todos, físico e digital, quando selecionado será retornado ou todos os jogos ou jogos físicos ou digitais. Estilo de jogo é um campo de seleção, listando todos os estilos cadastrados no sistema, podendo ter múltiplas seleções, quando buscado retorna qualquer jogo que tenha qualquer um dos estilos selecionados. Áreas de ensino é um campo de seleção, listando todas as áreas cadastradas no sistema, podendo ter múltiplas seleções, quando buscado retorna qualquer jogo que tenha qualquer uma das áreas selecionados.

Além disso, a tabela de listagem também apresenta um botão de ações, que permite ao usuário visualizar o jogo (ver Seção 4.3.6), e caso esteja autenticado ao sistema, permite o cadastro de um relato de uso (ver Seção 4.3.5).

Figura 4.3 – Tela de listagem

Filtros

Nome do jogo:

Tipo de jogo: Todos Físico Digital

Estilo de jogo:

Áreas de ensino:

Listagem

Nome	licença	Tipo	Nota	Ações
Jogo 1	Licença	Digital	0/5 ★	Ações
Jogo 10	Licença	Físico	0/5 ★	Ações
Jogo 2	Licença	Físico	0/5 ★	Ações
Jogo 3	Licença	Físico	0/5 ★	Ações
Jogo 4	Licença	Físico	0/5 ★	Ações
Jogo 5	Licença	Digital	0/5 ★	Ações
Jogo 6	Licença	Físico	0/5 ★	Ações
Jogo 7	Licença	Físico	0/5 ★	Ações
Jogo 8	Licença	Digital	0/5 ★	Ações
Jogo 9	Licença	Físico	0/5 ★	Ações

< 1 2 >

4.3.4 Cadastro do jogo

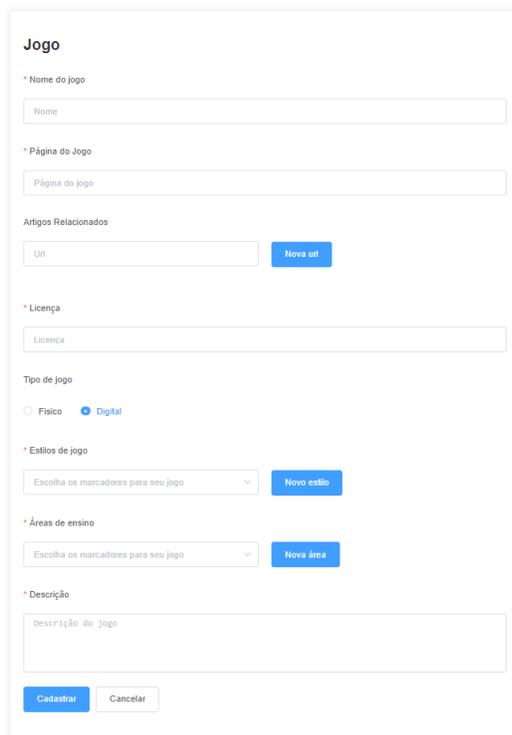
A tela de cadastro do jogo é responsável pelo registro de novos jogos no sistema, como mostrado na Figura 4.4. A tela apresenta os seguintes campos a serem preenchidos: nome do jogo, página do jogo, artigos relacionados, licença, o tipo de jogo, estilos de jogo, as áreas de ensino e descrição.

O nome do jogo é um campo simples de texto representando o nome do jogo. Página do jogo é um campo de tipo texto que representa um link para página do jogo. Artigos relacionados representa um campo do tipo texto para link de artigos relacionados ao jogo permitindo múltiplos links. Licença é um campo do tipo texto simples que representa a licença do jogo. o tipo de jogo é um campo do tipo seleção com duas opções, físico ou digital, que representa se o jogo é físico ou um eletrônico. Estilos de jogos é um campo de seleção, listando todos estilos de jogos cadastrados, representando o estilo de jogo, permitindo múltipla seleção, caso não tenha estilo de jogo cadastrado mais adequado é possível cadastrar um novo estilo de jogo. Áreas de ensino é um campo de seleção, listando todas as áreas de ensino cadastradas, representando as áreas de ensino do jogo, como por exemplo, se é da engenharia de software, desenvolvimento ou gestão,

permitindo múltipla seleção, caso não tenha área de ensino cadastrada mais adequada é possível cadastrar uma nova área de ensino.

Quando o botão cadastrar é acionado, há validação dos campos obrigatórios, caso seja verdadeiro o jogo é cadastrado. Já o botão cancelar retorna o usuário para a tela de listagem.

Figura 4.4 – Tela de Cadastro de jogos



O formulário, intitulado "Jogo", contém os seguintes campos e elementos:

- * Nome do jogo:** Campo de texto com o placeholder "Nome".
- * Página do Jogo:** Campo de texto com o placeholder "Página do jogo".
- Artigos Relacionados:** Campo de texto com o placeholder "Url" e um botão "Nova url" em azul.
- * Licença:** Campo de texto com o placeholder "Licença".
- Tipo de jogo:** Grupo de botões de opção com "Físico" e "Digital" (selecionado).
- * Estilos de jogo:** Campo de seleção com o placeholder "Escolha os marcadores para seu jogo" e um botão "Novo estilo" em azul.
- * Áreas de ensino:** Campo de seleção com o placeholder "Escolha os marcadores para seu jogo" e um botão "Nova área" em azul.
- * Descrição:** Campo de texto com o placeholder "Descrição do jogo".

Na base do formulário, há dois botões: "Cadastrar" em azul e "Cancelar" em cinza.

4.3.5 Cadastro do relato de uso do jogo

Esta tela é responsável pelo cadastro de relatos de uso referentes a jogos cadastrados no sistema, conforme as Figuras 4.5 e 4.6. A primeira parte do cadastro de relato de uso se refere aos cadastro de informações básicas (Figura 4.5): local aplicado, disciplina, o período da aplicação, o motivo e o numero de alunos. Local aplicado é um campo do tipo texto simples representando o local onde o jogo foi aplicado. Disciplina é um campo de tipo texto simples representando a disciplina que jogo foi aplicado. O período da aplicação é dois campos do tipo data representando o período em que o jogo foi aplicado. Motivo é um campo do tipo de texto representando o motivo para aplicação do jogo. Número de alunos é um campo do tipo número, representando o número de alunos onde o jogo foi aplicado.

A segunda parte do cadastro de relato de uso corresponde a uma avaliação de uso. Para garantir um padrão na avaliação dos jogos sérios, optamos por incluir uma parte do questionário do MEEGA+ (PETRI; WANGENHEIM; BORGATTO, 2018), referente a avaliação por parte do professor. O questionário é constituído de 13 perguntas com respostas em escala Likert de 1 (discordo totalmente) à 5 (concordo totalmente), conforme apresentado na Figura 4.6.

Figura 4.5 – Tela de Cadastro de Relato de Uso

The screenshot shows a form titled "Relato de Uso" with the following fields:

- * Local aplicado:** A text input field containing "Local aplicado".
- * Disciplina:** A text input field containing "Disciplina".
- * Data inicio da aplicação:** A date picker field showing "dd/mm/aaaa".
- * Data fim da aplicação:** A date picker field showing "dd/mm/aaaa".
- * Motivo:** A text area containing "MOTIVO".
- * Alunos:** A text input field containing "0".

Figura 4.6 – Tela de Cadastro de Relato de Uso

The screenshot shows a section titled "Perguntas" with 13 evaluation questions, each followed by a five-point Likert scale (represented by five stars):

- Houve algo interessante no início do jogo que capturou a atenção dos alunos.
- Os alunos estavam tão envolvidos no jogo que perderam a noção do tempo.
- Observei que os alunos esqueceram sobre o ambiente ao seu redor enquanto jogavam o jogo.
- Os alunos se divertiram com o jogo.
- Aconteceu alguma situação durante o jogo (elementos do jogo, competição, etc.) que fez os alunos sorrirem.
- Observei que este jogo é adequadamente desafiador aos alunos.
- O jogo ofereceu novos desafios aos alunos (ofereceu novos obstáculos, situações ou variações) com um ritmo adequado.
- O jogo não se tornou monótono nas suas tarefas (repetitivo ou com tarefas chatas).
- Os alunos interagiram durante o jogo.
- O jogo promove momentos de cooperação e/ou competição entre os jogadores.
- Observei que os alunos se sentiram bem interagindo com outras pessoas durante o jogo.
- Observei que ao completar as tarefas do jogo os alunos tiveram um sentimento de realização.
- Observei que é devido ao esforço pessoal dos alunos que eles conseguem avançar no jogo.

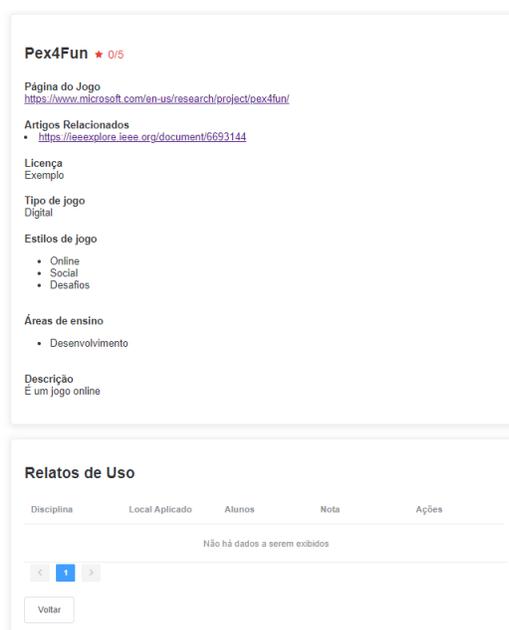
At the bottom of the form, there are two buttons: "Cadastrar" (highlighted in blue) and "Cancelar".

4.3.6 Visualização do jogo

Esta tela é responsável pela visualização da entidade jogo, como mostrado na Figura 4.7, a tela mostra todas as informações da entidade jogo, como o nome, a média da nota dos casos de uso, a página do jogo, artigos relacionados, licença, tipo de jogo, estilos de jogo, áreas de ensino e a descrição.

A tela também apresenta uma tabela exibindo todos os casos de uso, mostrando a disciplina, o local aplicado, a quantidade de alunos e a média das notas junto com botão de ações que ao ser acionado exibe uma modal, como mostrado na Figura 4.8, exibindo todas as informações do caso de uso, como as informações básicas e o questionário.

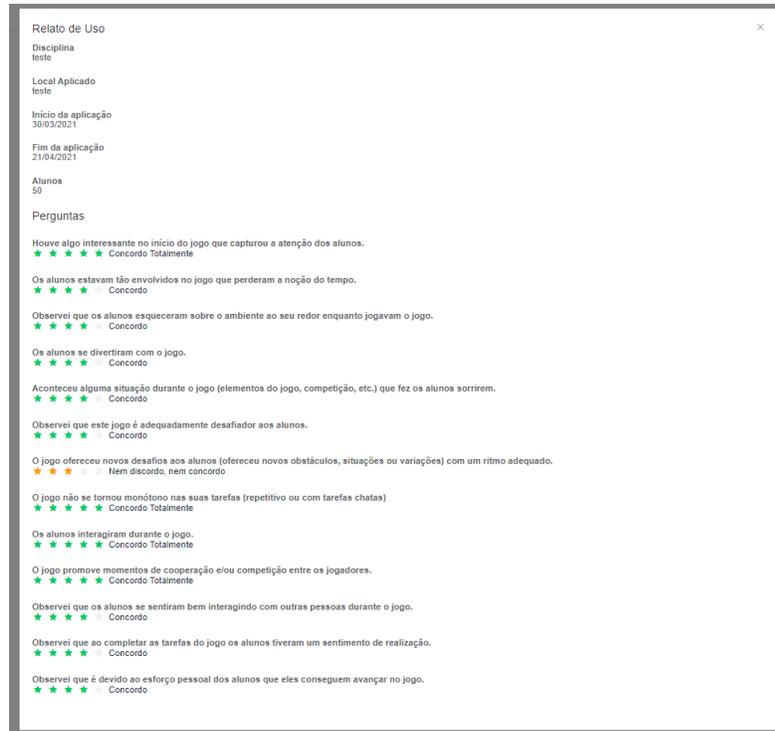
Figura 4.7 – Tela de Visualização do Jogo



4.3.7 Menu lateral

O sistema apresenta um menu lateral, como mostrado na Figura 4.9, com as opções para a listagem dos jogos (ver Seção 4.3.3) e para cadastro do jogo (ver Seção 4.3.4). Caso o usuário não esteja autenticado (ver Seção 4.3.2) a opção de cadastro de jogo ficará bloqueada exibindo uma mensagem pedindo a autenticação.

Figura 4.8 – Modal de Visualização do Relato de Uso



4.3.8 Menu Vertical

O sistema apresenta um menu vertical, como mostrado na Figura 4.10, com a opção usuário, exibe duas opções, login que redireciona para tela de autenticação (ver Seção 4.3.2) e a opção cadastro que redireciona para tela de cadastro de usuário 4.3.1. Caso o usuário esteja autenticado, passara a exibir o email do mesmo com a opção para se desligar do sistema.

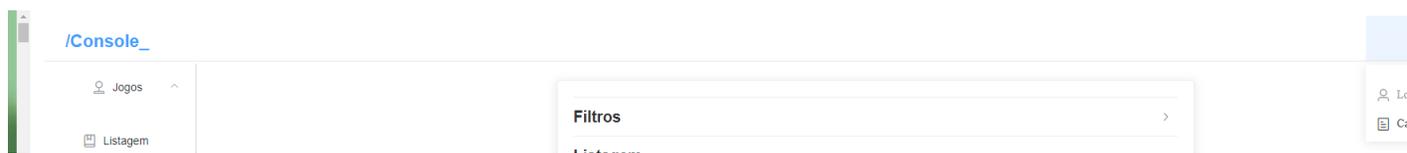
4.4 Limitações

A ferramenta apresenta limitações quanto ao seu uso em plataformas móveis, acessibilidade e não há uma identidade visual própria. Outras limitações estão relacionadas a falta de integração contra outras plataformas para autenticação ou recuperação de dados relacionados aos jogos.

Figura 4.9 – Menu Lateral



Figura 4.10 – Menu Vertical



5 CONCLUSÃO

O objetivo deste documento foi apresentar a ferramenta para cadastro de jogos sérios e seus relatos de uso e com isso, espera-se conseguir disseminar mais o uso dessas ferramentas de ensino para engenharia de software. Além disso, espera-se que o sistema ganhe a implementação de uma ferramenta para geração de relatórios e *dashboards* para melhor visualização dos jogos cadastrados no sistema.

Como trabalhos futuros pretende-se avaliar o sistema junto a educadores e pesquisadores em ensino de computação e realizar ajustes necessários com base no feedback de especialistas. Também é planejado um estudo de usabilidade para melhorar este aspecto não-funcional do sistema. Por fim, pretende-se publicar e acompanhar o uso do sistema.

Com o desenvolvimento deste sistema propiciou ao aluno experienciar vários conhecimentos práticos que me foram ensinados ao decorrer da graduação, além de experimentar alguns conceitos que só me foram mostrados na teoria. Conhecimentos estes adquiridos ao longo do curso de Sistemas de Informação. Em especial, as disciplinas Engenharia de Software, aplicado ao levantamento de requisitos e concepção da ferramenta, Redes de Computadores, aplicado ao desenvolvimento da *API*, Sistema distribuídos, aplicado a parte do desenvolvimento do servidor, da Programação Web, e Banco de Dados apresentaram conteúdos compatíveis aos necessários para a elaboração deste projeto.

REFERÊNCIAS

- ALVE, J. L. W. J. B. Padrão de arquitetura mvc: Model-viewcontroller. In: . [S.l.: s.n.], 2011.
- Ardis, M. et al. Se 2014: Curriculum guidelines for undergraduate degree programs in software engineering. **Computer**, v. 48, n. 11, p. 106–109, 2015.
- BAKER, A.; NAVARRO, E.; HOEK, A. van der. Problems and programmers: An educational software engineering card game. In: . [S.l.: s.n.], 2003. p. 614– 619. ISBN 0-7695-1877-X.
- MICROSOFT. **Conceitos básicos do Git e do GitHub para o Docs**. 2018. <<https://docs.microsoft.com/pt-br/contribute/git-github-fundamentals>>. Acesso em 17 fev. 2021.
- NAVARRO, E.; HOEK, A. van der. Multi-site evaluation of simse. In: . [S.l.: s.n.], 2009. v. 41, p. 326–330.
- PANEGALLI GILIANE BERNARDI, A. Z. C. F. S. Super mario logic: um jogo sério para auxiliar no processo de ensino e aprendizagem de lógica de programação. In: . [S.l.: s.n.], 2019.
- PETRI, G.; WANGENHEIM, C. Gresse von; BORGATTO, A. Meega+: A method for the evaluation of educational games for computing education. In: . [S.l.: s.n.], 2018.
- Rodrigues, P.; Souza, M.; Figueiredo, E. Games and gamification in software engineering education: A survey with educators. In: **2018 IEEE Frontiers in Education Conference (FIE)**. [S.l.: s.n.], 2018. p. 1–9.
- SCAICO DIEGO LOPES MARQUES, M. A. d. A. J. C. d. S. S. V. M. N. P. D. Combinando diversão e educação: Castelo dos enigmas, um jogo sério para o ensino de algoritmos. In: . [S.l.: s.n.], 2011.
- Souza, M. R. A. et al. Games for learning: bridging game-related education methods to software engineering knowledge areas. In: **2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)**. [S.l.: s.n.], 2017. p. 170–179.
- TILLMANN, N.; HALLEUX, J.; XIE, T. Pex4fun: Teaching and learning computer science via social gaming. In: . [S.l.: s.n.], 2011. p. 546–548.
- von Wangenheim, C. G.; Shull, F. To game or not to game? **IEEE Software**, v. 26, n. 2, p. 92–94, 2009.