

ThinkTest: uma plataforma para testes de jogos educacionais

Tiago Farias¹, José Carlos Duarte¹, Osvaldo Viana¹, Fernanda Pires¹, Marcela Pessoa¹

¹Universidade do Estado do Amazonas (UEA) - ThinkTEd Lab
Av. Darcy Vargas, 1.200 - Parque Dez de Novembro, Manaus - AM

{tfb.snf21, ojunior, fpires, mspessoa}@uea.edu.br

carlosduarte@icomp.ufam.edu.br

Abstract. *With the significant increase in the development of educational games, it has become essential to validate their effectiveness to ensure that their objectives are achieved. However, developers often face challenges due to a lack of tools to assist in this validation process. This work presents the beginning of the development of ThinkTest, a web platform to facilitate the application of educational game tests, offering a complete environment, allowing the organization of test groups and the construction of personalized forms for data collection, making the organization and the test planning faster and more effective. Based on reports of challenges faced by educational game developers, initial system documentation was defined.*

Resumo. *Com o aumento significativo no desenvolvimento de jogos educacionais, tornou-se essencial validar sua eficácia para garantir que seus objetivos sejam alcançados. No entanto, os desenvolvedores muitas vezes enfrentam desafios devido à falta de ferramentas para auxiliar neste processo de validação. Este trabalho apresenta o início do desenvolvimento do ThinkTest, uma plataforma web para facilitar a aplicação de testes de jogos educacionais, oferecendo um ambiente completo, permitindo a organização de grupos de teste e a construção de formulários personalizados para coleta de dados, tornando a organização e o planejamento de testes mais rápidos e eficazes. Com base em relatos de desafios enfrentados por desenvolvedores de jogos educacionais, foi definida a documentação inicial do sistema.*

1. Introdução

O avanço da tecnologia na educação tem promovido a implementação de novos métodos para melhorar a aprendizagem. Nesse contexto, jogos educacionais têm se destacado como uma nova e eficaz estratégia de ensino [Genesio et al. 2024]. A falta de motivação dos estudantes em conteúdos acadêmicos e escolares pelo método tradicional [Barbosa et al. 2017] torna os jogos educacionais digitais uma solução atrativa, uma vez que combinam elementos lúdicos com objetivos pedagógicos, promovendo um ambiente de aprendizado mais dinâmico e interativo.

Apesar do interesse pela área de estudo, pois parte dos jogos para aprendizagem são desenvolvidos academicamente, desde a graduação até projetos de mestrado, em laboratórios de pesquisa, avaliar estes artefatos ainda é um desafio. Um dos métodos mais usados para validar esses jogos é a aplicação de formulários de teste na etapas de desenvolvimento do sistema, podendo avaliar pontos distintos. Um exemplo de um teste desse

tipo é o MEEGA+ [Petri et al. 2019], um teste em formato de formulário que avalia diversas dimensões de um jogo educacional, como usabilidade, atenção e aprendizagem, auxiliando os desenvolvedores a entender e verificar onde e o que deve ser melhorado nas próximas versões [Petri et al. 2019].

Para investigar esses desafios, este estudo utilizou a metodologia de Grupo Focal. As discussões envolveram desenvolvedores de jogos educacionais, com o objetivo de identificar dificuldades na aplicação e análise de testes. A partir dessas discussões, identificamos problemas enfrentados tanto por desenvolvedores iniciantes quanto por mais experientes. Os resultados indicam que desenvolvedores iniciantes têm dificuldades em encontrar testes validados, enquanto os mais experientes enfrentam desafios na organização e comparação dos dados coletados.

Contudo, o processo de testar um jogo pode ser complexo e trabalhoso, especialmente quando envolve a criação e aplicação de formulários extensos, o que aumenta a dificuldade na análise posterior dos dados. Para enfrentar esses desafios, este trabalho apresenta o ThinkTest, uma plataforma web projetada para facilitar a organização e a coleta de dados em testes de jogos educacionais. A plataforma oferece uma variedade de testes validados pela literatura, que podem ser estudados e aplicados de forma ágil pelos desenvolvedores, além de permitir a criação de formulários personalizados. Além disso, o ThinkTest simplifica a organização dos dados por meio da orquestração em grupos, proporcionando uma visualização mais clara e eficiente das informações.

2. Referencial Teórico e Trabalhos Relacionados

Avaliar jogos educacionais é essencial para determinar a sua eficácia. Na revisão sistemática da literatura conduzida por [Petri and von Wangenheim 2017], analisou-se 117 estudos sobre a avaliação de jogos para educação. Esses estudos foram examinados quanto à definição das avaliações, considerando fatores de análise, desenhos de pesquisa, modelos/métodos de avaliação, tipos de instrumentos de coleta de dados, tamanhos de amostra, replicações e métodos de análise de dados. Uma das principais lições aprendidas com essa revisão sistemática é a falta de padronização nas metodologias de avaliação de jogos educacionais. Muitos dos estudos analisados utilizavam métodos *ad hoc*, com amostras pequenas e variáveis, o que limitava a comparação dos resultados e a replicabilidade dos testes.

A avaliação de uma estratégia instrucional deve determinar seu nível de sucesso, verificando se o público-alvo alcançou os objetivos estabelecidos, abrangendo tanto a aprendizagem do aluno, quanto a qualidade dos elementos, materiais e recursos que compõem a estratégia instrucional [Branch 2009]. No entanto, estudos como [Calderón and Ruiz 2015] revelam uma falta de rigor e padronização nas avaliações de jogos educacionais, que frequentemente são realizadas de maneira *ad hoc*, com populações variando de 1 a 40 alunos e usando métodos simples, como questionários. Nesse contexto, a plataforma ThinkTest surge como uma solução eficaz, permitindo a criação de formulários e a condução de diversos tipos de testes para jogos educacionais. O ThinkTest pretende facilitar a coleta de dados e oferecer métodos avançados de análise, de forma a permitir uma avaliação mais precisa e abrangente, suprimindo a lacuna identificada na literatura.

Com objetivo parecido com o ThinkTest, o Assistant MEEGA+ é um sistema web

com *layout* responsivo que abrange todas as etapas de aplicação do MEEGA+: planejamento e preparação do questionário, execução da avaliação com os usuários e síntese dos dados coletados [Ripasy et al. 2018]. No entanto, uma grande limitação do Assistant MEEGA+ é que ele suporta apenas um único teste, o MEEGA+, restringindo seus possíveis usuários àqueles que buscam testar seus jogos especificamente com esse teste.

Por outro lado, uma diferença importante entre o ThinkTest e as avaliações discutidas na revisão sistemática está na escalabilidade. Enquanto muitos dos estudos avaliados na revisão utilizavam amostras pequenas, o ThinkTest foi concebido para ser escalável, permitindo que os desenvolvedores conduzam testes com um número maior de usuários e repliquem esses testes em diferentes contextos, possibilitando uma análise mais ampla e conclusiva sobre a eficácia dos jogos educacionais.

3. Metodologia

Esta seção descreve a metodologia adotada neste estudo, que forneceu a base para o desenvolvimento da pesquisa e ajudou a compreender melhor as dificuldades enfrentadas pelos desenvolvedores ao testar jogos educacionais. Para isso, foram utilizados métodos qualitativos que envolveram a coleta de dados através de discussões em grupo, seguidas de uma análise criteriosa das respostas obtidas.

3.1. Grupo focal

Para entender melhor como os desenvolvedores se sentem em relação ao processo de avaliação de jogos educacionais, foi realizado um Grupo Focal. Essa metodologia se baseia em uma reunião para debater sobre um assunto específico. No caso desta pesquisa, o objetivo foi entender as principais dificuldades que os desenvolvedores enfrentam ao testar se seus jogos são realmente eficientes no processo de aprendizagem, por meio de discussões em grupo, para captar experiências e percepções.

O Grupo Focal contou com a participação de cinco desenvolvedores de jogos em diferentes níveis de experiência, o que ajudou a identificar problemas enfrentados, tanto por desenvolvedores iniciantes, que possuem barreiras em testar jogos, quanto por desenvolvedores mais experientes, que têm opiniões sólidas e vivências a serem compartilhadas. As pessoas participantes foram recrutadas por conveniência, sendo desenvolvedores conhecidos dos autores. Foram convidados por rede social, e todos os participantes já tinham alguma experiência no desenvolvimento de jogos educacionais.

O Grupo Focal foi realizado de forma remota pela plataforma de reunião Google Meet e teve duração de aproximadamente uma hora. A condução da discussão seguiu um roteiro estruturado para garantir que todos os participantes tivessem a oportunidade de compartilhar suas experiências e opiniões. O Grupo Focal foi norteado por seis perguntas, descritas a seguir:

- Como e quando você começou a desenvolver jogos?
- Com eram feitos os testes em jogos educacionais?
- Como descobriu os testes de jogos educacionais?
- Quais dificuldades você enfrenta ao aplicar os testes?
- Quais dificuldades você enfrenta ao organizar os dados do testes?
- Quais dificuldades você enfrenta ao analisar os resultados dos testes?

A análise dos dados foi realizada por meio de uma análise temática, onde as respostas foram categorizadas em temas chave relacionados aos desafios no processo de avaliação de jogos. Após uma análise dos dados do grupo focal foi possível notar que, para desenvolvedores mais iniciantes, foi evidente que umas das principais dificuldades para avaliar jogos educacionais seria encontrar testes validados para serem utilizados. Devido à falta de experiência, esses desenvolvedores muitas vezes desconhecem testes e possuem dificuldade em procurá-los e estudá-los para entender seus objetivos e o que eles avaliavam. Esses desafios se tornam barreiras significativas no processo de teste, dificultando não apenas a aplicação dos testes, mas também a análise posterior dos resultados, já que os desenvolvedores frequentemente não compreendem completamente os propósitos e as métricas desses testes. Consequentemente, isso pode levar a avaliações imprecisas e a dificuldades em melhorar os jogos educacionais com base no *feedback* obtido.

Enquanto isso, desenvolvedores mais experientes enfrentam dificuldades principalmente no processo de aplicação e organização dos dados para análise de resultados. Eles relatam que construir o teste para aplicação é uma atividade maçante, pois é necessário obter as perguntas do teste e, com o auxílio de alguma ferramenta como um software de construção de formulários, reescrevê-las uma por uma para que os testadores possam respondê-las. Esse processo manual está altamente suscetível a erros, e um pequeno deslize pode invalidar ou comprometer toda uma sessão de teste. Além disso, há uma grande dificuldade em cruzar dados entre sessões de teste. Desenvolvedores muitas vezes desejam comparar diferentes sessões para avaliar o progresso das melhorias entre as versões do jogo, a fim de validar se as alterações feitas foram realmente eficazes. A falta de ferramentas automatizadas para facilitar essa comparação torna o processo ainda mais desafiador e propenso a erros.

Nesse contexto, o Grupo Focal se revelou uma ferramenta importante para avaliar as dificuldades que os desenvolvedores enfrentam e como o ThinkTest poderia ajudar a resolvê-las. As barreiras identificadas durante a discussão se tornaram a base para definir os requisitos e funcionalidades do sistema, com o objetivo de eliminá-las e proporcionar um processo de avaliação mais eficaz e rápido. A plataforma, portanto, foi projetada para atender às necessidades reais dos desenvolvedores.

3.2. Requisitos

A etapa de elicitação de requisitos é essencial durante o desenvolvimento, pois são definidos as funcionalidades que o sistema necessitará. E eles ajudam a garantir que todas as funcionalidades necessárias sejam implementadas corretamente e que o sistema atenda às expectativas dos usuários finais.

A Tabela 1 apresenta a lista de requisitos levantados, juntamente com a descrição de cada um dos requisitos. É possível observar que os requisitos RF01 e RF02 são responsáveis pelas funcionalidades de organização de grupos de testes, sendo o primeiro responsável por definir a página home que o usuário visualizará todos os grupos que faz parte, e o segundo por manipular as informações dos grupos como nome, descrição, código do grupo, dentre outros. Nesse contexto, como descrito no RF02, usuários serão capazes de ingressar novos grupos a partir do código do grupo a qual desejam participar. Além disso, usuários desenvolvedores poderão cadastrar jogos no sistema a qual desejam testar posteriormente, descrito no RF04. Por fim, os requisitos RF05, RF06 e RF07

são responsáveis por descrever como os usuários irão interagir com os testes de jogos no sistema, podendo respectivamente: manipular dados de testes, responder um teste e visualizar os resultados das respostas.

Requisitos	
Identificação	Descrição
RF01 – Visualizar página <i>home</i>	Ao acessar a página home, o desenvolvedor terá uma visualização dos grupos disponíveis. Nessa visualização, será apresentado um ícone para criar um grupo ou participar de um existente.
RF02 – Manter grupo	O sistema deve permitir o gerenciamento completo dos grupos por meio das operações CRUD (<i>Create, Read, Update, Delete</i>).
RF03 – Participar de um grupo	O Usuário poderá participar de um grupo ao fornecer o código de identificação do grupo desejado
RF04 – Manter jogo	O sistema deve permitir o gerenciamento completo dos jogos por meio das operações CRUD (<i>Create, Read, Update, Delete</i>).
RF05 – Manter teste	O sistema deve permitir o gerenciamento completo dos testes por meio das operações CRUD (<i>Create, Read, Update, Delete</i>).
RF06 – Responder teste	O sistema deve permitir que usuários respondam testes.
RF07 – Visualizar respostas	O sistema deve permitir que os donos dos grupos possam visualizar os resultados das respostas dos testes.

Tabela 1. Requisitos Funcionais.

4. Resultados

Após uma análise dos desafios que os desenvolvedores enfrentam ao avaliar jogos educacionais e dos requisitos do sistema, foi concebida uma série de *mockups*, desenvolvidos no Figma, que compõem a interface do sistema ThinkTest.

A Figura 1 apresenta uma das telas principais do sistema, a tela de Grupo na visão de um usuário desenvolvedor. Nesta tela, será possível visualizar as informações de um grupo como nome, descrição e código para participação. Nela usuários do sistema podem interagir entre si por meio de testes adicionais pelo dono do grupo para que os testadores possam avaliar seus jogos.

A tela de criação de teste apresentada na Figura 2 possibilita o usuário adicionar ou criar testes em grupos cujo é dono. O usuário desenvolvedor poderá definir um nome, um jogo que será testado, uma descrição, data e hora limite e por fim selecionar um teste validado pela literatura que já esteja disponível no sistema ou criar um novo teste.

Antes de iniciar o restante das documentações, foi planejada a arquitetura do sistema. Com auxílio do nível 2 do modelo C4, que fornece uma ampla visualização dos módulos do sistema e como se comunicam entre si, ilustrado na Figura 3, o sistema foi



Figura 1. Tela de Grupo.

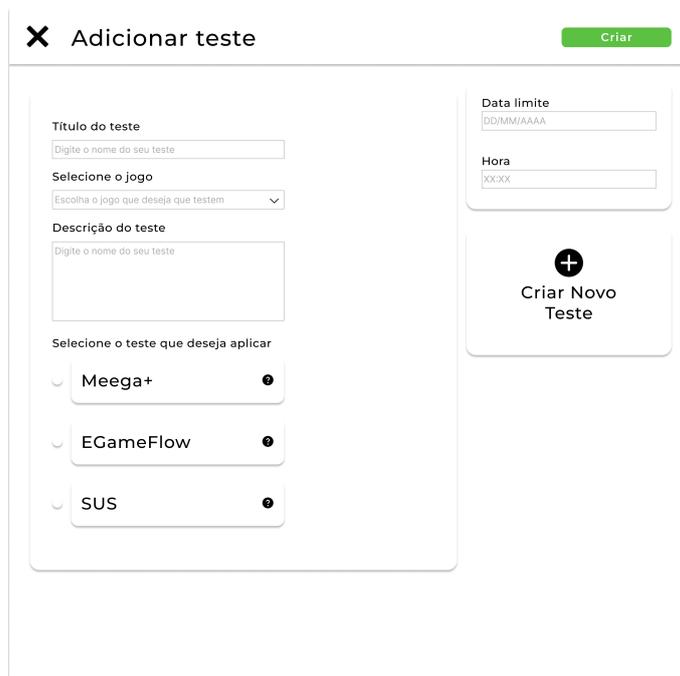


Figura 2. Tela de Criação de Teste.

divido em três *containers* (módulos). Primeiramente, o FrontEnd Web será o *container* que o usuário terá contato direto para exercer as funções do sistema, sendo responsável por renderizar as telas onde apresentará os dados vindo da API, que por sua vez é o *container* responsável por fornecer e processar os dados do banco de dados em PostgreSQL.

Outra etapa crucial no desenvolvimento do ThinkTest foi o planejamento do

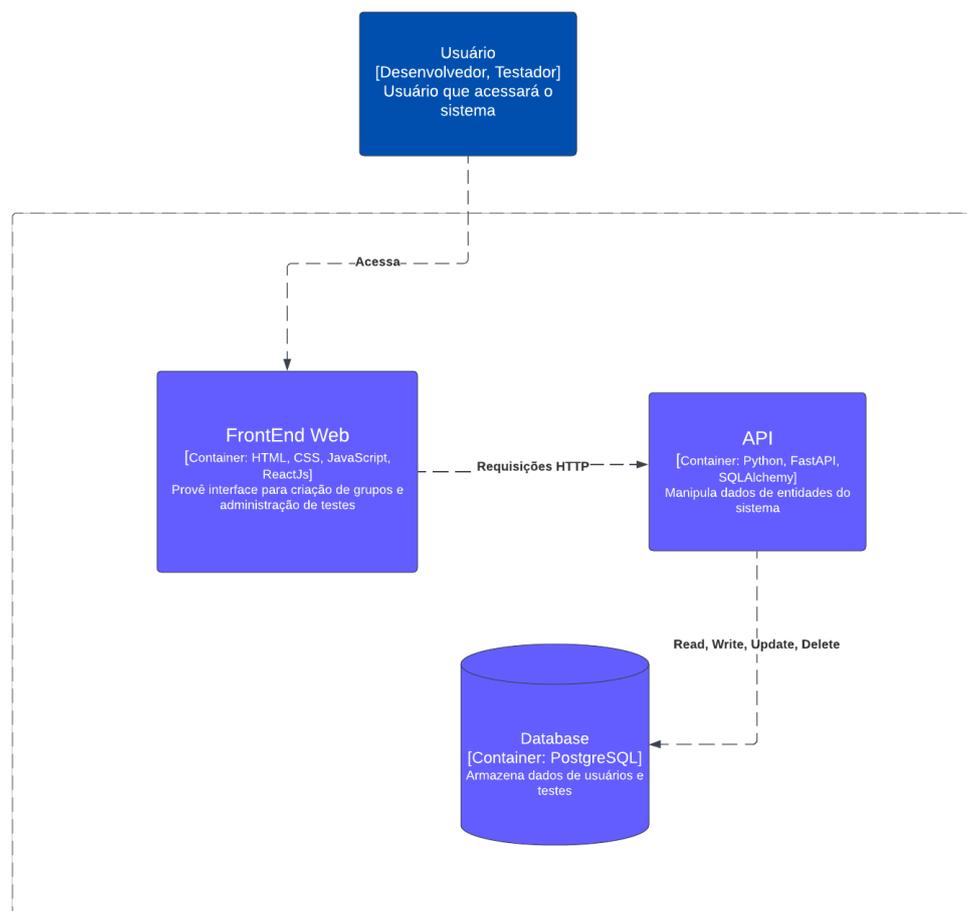


Figura 3. Modelo C4: nível 2.

banco de dados, uma parte essencial que seria responsável por armazenar informações de usuários e testes, viabilizando todas as funcionalidades do sistema. Para uma melhor visualização das tabelas que comporiam o sistema, desenvolveu-se base de dados para listar as entidades atuantes no sistema e seus relacionamentos.

Para criar um sistema capaz de gerar testes em formato de formulário de forma genérica, adaptável a diversas situações, foi necessário abstrair os conceitos de testes. Assim, definiu-se em Testes, Seções, Perguntas e Alternativas. Essas entidades têm como propósito, respectivamente, armazenar informações de um teste, representar as seções de perguntas, armazenar os dados das perguntas e armazenar os conteúdos das alternativas.

Após um mapeamento de tais entidade foi realizado um série de testes em SQL. Nesta etapa, era essencial validar se a modelagem seria capaz de cumprir as funcionalidades do sistema. Para tanto, populou-se um banco de dados PostgreSQL com as tabelas oriundas do mapeamento e executou-se uma série de *queries* simulando funcionalidades do sistema. Os testes incluíram simulações de criação de grupos, criação de testes e resposta a testes. Como resultado, a modelagem demonstrou-se capaz de satisfazer os requisitos do sistema, confirmando sua viabilidade e eficácia para suportar as operações planejadas.

Após a criação dos mock ups do sistema, das análises do requisitos e modelagem do banco de dados foi iniciado o desenvolvimento da aplicação, começando pela API construída em Python utilizando FastAPI. A tecnologia foi escolhida por conta da sua facilidade de implementação e facilidade de documentação, haja vista que o framework possui uma documentação automática das rotas com Swagger, facilitando o desenvolvimento e os testes dos endpoints. A API possui rotas necessárias para funcionalidades do sistema, sendo responsável por administrar a comunicação com o Banco de Dados, e manipulação de entidades como Grupos, Jogos e Testes, de modo que seja possível realizar os principais requisitos do sistema como manter jogos, grupos e testes.

No início do desenvolvimento do FrontEnd para o ThinkTest, o foco foi na construção de uma base funcional utilizando React para interagir com a API desenvolvida em FastAPI. Nesta fase inicial, foram criados componentes básicos e implementada a integração com os endpoints da API. O objetivo principal era assegurar que a comunicação entre o FrontEnd e a API funcionasse corretamente, permitindo a realização de testes e a validação de algumas funcionalidades essenciais.

5. Considerações finais

Neste trabalho, apresentamos o desenvolvimento inicial do ThinkTest, começando pela investigação com metodologias como grupo focal para entender as dificuldades que desenvolvedores enfrentam ao avaliar seus jogos. Isso validou a importância de criar uma plataforma para auxiliar nesse processo. Após a análise dos desafios, realizamos uma série de documentações para guiar o desenvolvimento do sistema.

Entre as dificuldades encontradas durante o progresso do trabalho, a modelagem do banco de dados foi especialmente complexa. Criar uma série de entidades que pudessem satisfazer e resolver os problemas identificados na etapa de pesquisa exigiu um esforço significativo. Diversos testes SQL foram realizados até que a modelagem atingisse sua versão atual, garantindo que ela atendesse aos requisitos do sistema e proporcionasse uma base sólida para as funcionalidades do ThinkTest.

Para os próximos passos do desenvolvimento da plataforma, a prioridade será continuar o desenvolvimento do FrontEnd e finalizar sua estilização conforme os mockups, visando proporcionar uma melhor experiência de usuário. Além disso, serão realizados testes com usuários, incluindo os participantes do grupo focal apresentado neste artigo, para validar se a plataforma é realmente eficaz e se soluciona os problemas enfrentados por desenvolvedores ao avaliar seus jogos no dia a dia. Esses testes serão fundamentais para ajustar a plataforma conforme o feedback recebido e garantir que o ThinkTest atenda às necessidades dos desenvolvedores de jogos educacionais de forma eficiente e prática.

Referências

- Barbosa, H., Reolon, M., and von Wangenheim, C. G. (2017). Heurísticas para avaliação de jogos educativos digitais—revisão sistemática de literatura.
- Branch, R. M. (2009). *Instructional design: The ADDIE approach*, volume 722. Springer.
- Calderón, A. and Ruiz, M. (2015). A systematic literature review on serious games evaluation: An application to software project management. *Computers & Education*, 87:396–422.

- Genesio, N. O. S., de Oliveira, A. M., Oliveira, E. W., and Valle, P. H. D. (2024). Panorama de estudos sobre jogos educacionais digitais em educação em computação. In *Workshop sobre Educação em Computação (WEI)*, pages 737–749. SBC.
- Petri, G. and von Wangenheim, C. G. (2017). How games for computing education are evaluated? a systematic literature review. *Computers & education*, 107:68–90.
- Petri, G., von Wangenheim, C. G., and Borgatto, A. F. (2019). Meega+: Um modelo para a avaliação de jogos educacionais para o ensino de computação. *Revista Brasileira de Informática na Educação*, 27(03):52–81.
- Ripasy, R., Petri, G., von Wangenheim, C. G., Conte, T., and Marques, A. B. (2018). Assistant meega+: Uma ferramenta de apoio para avaliação de jogos educacionais usando modelo meega+. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 29, page 615.