

Mestres do Processo Digital: Gamificando o Ensino de Engenharia de Software

Title: Masters of the Digital Process: Gamifying Software Engineering Education

Victor Travassos Sarinho¹

¹Laboratório de Entretenimento Digital Aplicado (LEnDA)
Universidade Estadual de Feira de Santana (UEFS)
Feira de Santana, BA - Brasil

vsarinho@uefs.br

Abstract. Introduction: Teaching Software Engineering continuously seeks strategies to reduce the gap between theoretical knowledge and professional practice in authentic development environments. **Objective:** This paper details the creation of *Mestres do Processo Digital*, a virtual adaptation of a board game designed to teach project management and software processes in a playful and interactive way. **Methodology:** The system simulates the phases of the Rational Unified Process (RUP), requiring the player to manage resources, allocate developers across knowledge areas, and resolve adverse events presented through a dynamic deck of cards. The implementation adopts a declarative, data-oriented approach based on the EnSCEVA model, in which the game logic is specified in a JSON file and interpreted by a developed engine. **Results:** The validation conducted with Computer Engineering students indicated a positive reception, highlighting the relevance of the content to the course and the game's ability to maintain user attention and engagement. **Keywords** Software Engineering, Educational Games, Project Management, RUP, EnSCEVA, Godot Engine.

Resumo. Introdução: O ensino de Engenharia de Software busca constantemente estratégias para reduzir a distância entre o conhecimento teórico e a prática profissional em ambientes de desenvolvimento autênticos. **Objetivo:** Este artigo detalha a criação do jogo *Mestres do Processo Digital*, uma adaptação para o ambiente virtual de um jogo de tabuleiro projetado para ensinar gestão de projetos e processos de software de forma lúdica e interativa. **Metodologia:** O sistema simula as fases do Rational Unified Process (RUP), exigindo que o jogador gerencie recursos, aloque desenvolvedores em áreas de conhecimento e resolva eventos adversos apresentados por meio de um baralho dinâmico de cartas. A implementação utiliza uma abordagem declarativa orientada a dados baseada no modelo EnSCEVA, onde a lógica do jogo é especificada em um arquivo JSON e interpretada por um motor desenvolvido. **Resultados:** A validação realizada com estudantes de Engenharia de Computação indicou uma recepção positiva, com destaque para a relevância do conteúdo para a disciplina e a capacidade do jogo em manter a atenção focada e o engajamento dos usuários. **Palavras-Chave** Engenharia de Software, Jogos Educacionais, Gerenciamento de Projetos, RUP, EnSCEVA, Godot Engine

1. Introdução

Um dos principais desafios na formação de engenheiros de software é reduzir a lacuna entre o conhecimento teórico e sua aplicação prática [Oguz e Oguz 2019]. Embora muitos estudantes dominem os princípios da Engenharia de Software (ES), frequentemente possuem pouca experiência com ambientes reais de desenvolvimento, ferramentas colaborativas e projetos autênticos. Essa limitação pode comprometer sua preparação para lidar com a complexidade dos projetos modernos, reforçando a necessidade de abordagens educacionais que reproduzam contextos próximos à prática profissional [Oguz e Oguz 2019].

Os jogos têm sido reconhecidos como estratégias eficazes para apoiar objetivos educacionais, favorecendo a aprendizagem, a motivação, o interesse e a persistência dos estudantes [Su e Zou 2024, Alotaibi 2024, Soriano-Sánchez et al. 2026]. Além disso, promovem aprendizagem ativa e podem facilitar a compreensão de conceitos complexos em períodos reduzidos de ensino [Von Wangenheim e Shull 2009], servindo tanto para prática e reforço quanto para a representação de conceitos dinâmicos ou abstratos [Hays e Singer 2012].

Nesse contexto, a incorporação de jogos no ensino de ES tem se mostrado uma alternativa promissora aos métodos tradicionais. Jogos permitem criar ambientes imersivos que simulam desafios reais do desenvolvimento de software, possibilitando experimentação, colaboração e tomada de decisão em um ambiente seguro [Kosa et al. 2016]. Além disso, contribuem para o desenvolvimento de competências essenciais, como resolução de problemas, criatividade e trabalho em equipe [Kosa et al. 2016].

Seguindo essa perspectiva, este trabalho apresenta uma adaptação digital do *Mestres do Processo* [Sarinho 2019b, Sarinho 2019a], originalmente concebido como um jogo de tabuleiro para o ensino de gestão de projetos e processos de desenvolvimento de software. Reimaginada como uma experiência para um único jogador, a versão digital busca ampliar a acessibilidade e a interatividade do jogo, oferecendo um ambiente flexível para experimentação individual, autoavaliação e aplicação em contextos educacionais.

Com relação ao prisma ético deste trabalho, o qual se baseia na análise de ações-respostas organizadas e obtidas a partir de um brincar participativo e espontâneo de atividades funcionais para fins de ensino, entende-se que o mesmo se enquadra na Resolução CNS nº 510, de 2016, segundo a qual: “Não serão registrados nem avaliados pelo sistema CEP/CONEP: [...] VII – pesquisa que objetiva o aprofundamento teórico de situações que emergem espontânea e contingencialmente na prática profissional, desde que não revelem dados que possam identificar o sujeito”; e “atividade realizada com o intuito exclusivamente de educação, ensino ou treinamento sem finalidade de pesquisa científica, de alunos de graduação, de curso técnico, ou de profissionais em especialização” [Brasil 2016].

2. Trabalhos Relacionados

Uma busca no *Google Scholar*¹ utilizando os termos *Software Engineering*, *Digital Games* e *Project Management* revela diferentes jogos sérios voltados ao ensino de gestão

¹<https://scholar.google.com/>

e processos em ES. Entre eles, destacam-se o *SimSE* [Navarro e van der Hoek 2004], que simula o gerenciamento de projetos de software por meio de decisões sobre equipe, orçamento e tarefas; o *SimuES-W* [Suescún Monsalve et al. 2018], que promove aprendizagem colaborativa baseada em casos reais e discussão de soluções; o *The Incredible Manager* [Dantas et al. 2004], que utiliza modelos de Dinâmica de Sistemas para desenvolver habilidades de tomada de decisão em projetos; e o *ProDec* [Calderón e Ruiz 2013], que combina treinamento e avaliação em um ambiente virtual de gestão de projetos. Em conjunto, esses jogos evidenciam o potencial das abordagens baseadas em simulação para apoiar a aprendizagem experiencial em ES. Nesse contexto, a versão digital de *Masters of the Process* busca oferecer uma alternativa acessível e interativa para o ensino de gestão de projetos e processos de software em sala de aula.

Com relação ao jogo usado como modelo para este trabalho, *Masters of the Process* [Sarinho 2019b, Sarinho 2019a] se apresenta como um jogo de tabuleiro desenvolvido para o ensino de gestão de projetos e processos de software. Neste jogo, os participantes devem administrar recursos, alocar profissionais e lidar com eventos adversos durante as diferentes fases de um projeto, simulando desafios típicos da Engenharia de Software de forma colaborativa e lúdica.

Para a versão digital proposta neste trabalho, preserva-se os principais objetivos pedagógicos e mecânicas centrais do jogo original, incluindo a gestão de orçamento, a alocação de desenvolvedores, o uso de cartas de evento e a progressão pelas fases do processo de desenvolvimento. Entretanto, a adaptação introduz elementos não presentes na versão física, como execução automatizada das regras, feedback imediato, sistema de ranqueamento, condições explícitas de vitória e derrota, acompanhamento automático do estado do projeto em uma partida e suporte ao aprendizado individual.

Assim, como uma das principais contribuições deste trabalho, tem-se não apenas na digitalização do jogo, mas também a investigação de como os recursos computacionais desenvolvidos podem potencializar a experiência educacional originalmente proposta pelo jogo de tabuleiro, mantendo seus objetivos pedagógicos e ampliando suas possibilidades de aplicação em contextos de ensino.

3. Metodologia

O desenvolvimento do *Mestres do Processo Digital* foi orientado por objetivos educacionais relacionados à compreensão de processos de software, gerenciamento de recursos e tomada de decisão em projetos de desenvolvimento. Para isso, foram preservados os principais elementos pedagógicos do jogo de tabuleiro original, cuja proposta é promover aprendizagem experiencial por meio da simulação de cenários típicos da Engenharia de Software.

Embora metodologias ágeis sejam amplamente adotadas pela indústria atualmente, o RUP foi mantido como estrutura de referência por apresentar fases, papéis e artefatos claramente definidos, facilitando a representação de conceitos fundamentais de planejamento, acompanhamento e controle de projetos. Dessa forma, o foco do jogo não está no treinamento de um processo específico, mas na compreensão de princípios gerais de gestão e desenvolvimento de software que permanecem relevantes independentemente da metodologia utilizada.

A partir dessa concepção pedagógica, a implementação do jogo foi realizada por

meio de uma abordagem declarativa orientada a dados baseada no modelo EnSCEvA [Sarinho 2024], no qual toda a lógica do jogo é especificada em um arquivo JSON² posteriormente interpretado por um motor de execução. Para isso, foi desenvolvido um script em JavaScript responsável por gerar a especificação do jogo a partir de três elementos centrais: entidades, assertivas e cenas.

As entidades, definidas por meio da função *add_entity*, representam os objetos do jogo e são descritas a partir de propriedades tipadas, como sprites, botões, textos, zonas, temporizadores e recursos de áudio. As assertivas, criadas com a função *add_assert*, correspondem às regras de transição de estado, sendo compostas por condições de avaliação (*eval*) e ações de execução (*exec*), ambas baseadas nos valores das entidades previamente declaradas. Por sua vez, as cenas, definidas por *add_scene*, representam as diferentes telas do jogo, organizando componentes visuais atrelados às entidades e eventos associados às assertivas. Por fim, a geração do arquivo JSON final (*mestres.json*) é realizada pela função *make_json_file*, responsável por serializar a estrutura declarativa do jogo em formato JSON, tornando-a apta para interpretação pelo motor de execução.

O motor de execução foi implementado utilizando a Godot Engine³, por meio de um interpretador customizado capaz de processar arquivos JSON que seguem o modelo EnSCEvA [Sarinho 2024]. Esse interpretador utiliza a sintaxe da linguagem GDScript do Godot para avaliação de expressões e execução de comandos. A arquitetura do motor é baseada em um sistema de eventos reativo, no qual interações do jogador, como cliques, arrastos, colisões e temporizações, disparam assertivas que promovem alterações no estado global do jogo.

Com relação a modelagem do jogo, tem-se nas entidades a representação dos elementos do domínio do jogo, incluindo cartas, componentes de interface, variáveis de estado e recursos do sistema. Como exemplo, o conjunto de cartas do jogo é modelado como uma coleção de objetos contendo atributos como título, descrição, tipo e ações associadas, os quais impactam diretamente o estado do jogo, como alterações no orçamento, geração de bugs e consumo de recursos. Já o estado global do jogo é mantido em uma entidade central denominada *game*, responsável por armazenar variáveis de controle, como orçamento (*currentBudget*), contadores de bugs (*bugTokens*), desenvolvedores (*developerTokens*) e fichas de trabalho (*productionTokens*).

As regras do jogo são formalizadas por meio das assertivas, que combinam condições (*eval*) e ações (*exec*) executadas em resposta a eventos específicos. Essa abordagem permite a definição de comportamentos reativos, incluindo transições de estado, cálculo de pontuação e verificação de condições de vitória ou derrota. Dessa forma, o sistema consegue avaliar dinamicamente critérios como quantidade de bugs e orçamento disponível para determinar o desempenho do jogador.

Por fim, as cenas organizam os elementos visuais e interativos em diferentes estados do jogo, tais como menu inicial, deck de cartas avulso, tela de jogo, tutorial, tela de vitória e tela de derrota (Figura 1). Cada cena define os eventos associados, como cliques, temporizadores e colisões, e os respectivos componentes visuais que irão renderizar os dados das entidades associadas, incluindo sprites, botões, textos e recursos

²<https://www.json.org/json-en.html>

³<https://godotengine.org/>

de áudio, estruturando assim o fluxo de interação do usuário.

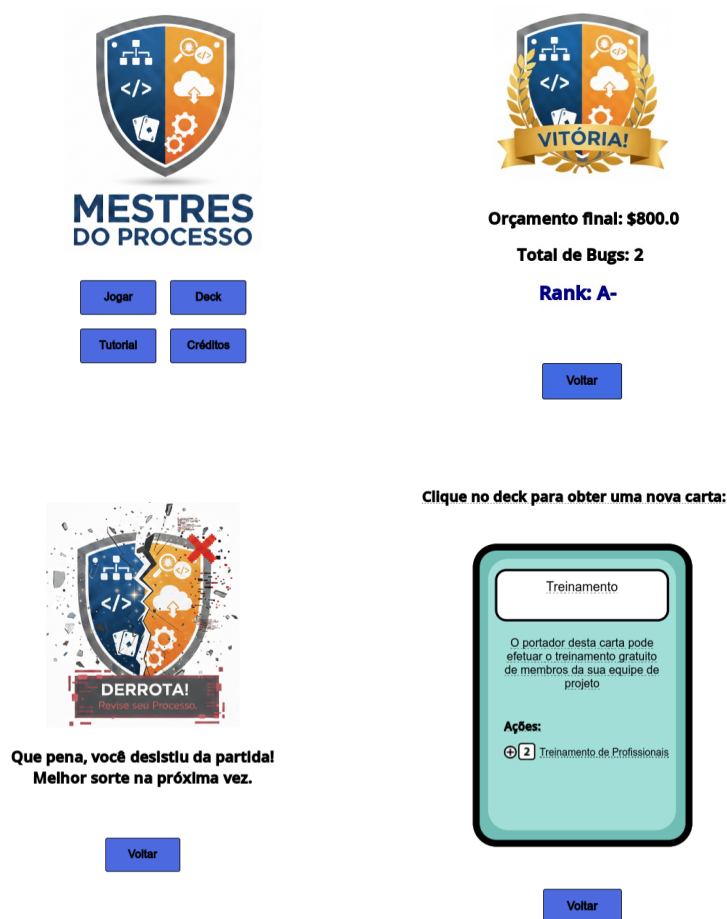


Figura 1. Telas de menu inicial, vitória, derrota e deck de cartas do jogo.

4. Resultados Obtidos

4.1. Jogo Produzido

*Mestres do Processo Digital*⁴ busca simular de forma lúdica a gestão de recursos na condução de projetos de software. Para isso, a interface principal do jogo (Figura 2) apresenta um painel de trabalho central com sete áreas de conhecimento dispostas circularmente, as quais irão receber desenvolvedores conforme a respectiva área alocada.

O jogo reproduz as quatro fases do RUP, Concepção, Elaboração, Construção e Transição, com cada uma exigindo a entrega de um número específico de fichas de trabalho (tokens de produção) por área de conhecimento (Análise de Negócio, Gerência, Análise & Projeto, Implementação, Testes, Configuração & Manutenção, Implantação & Suporte), conforme indicado no painel de fase do lado esquerdo da Figura 2. A progressão entre as fases só ocorre quando todas as fichas de trabalho exigidas são entregues (*drag'n'drop* no painel de fases) e não existem bugs pendentes nas áreas correspondentes.

⁴Disponível em: <https://vsarinho.github.io/Mestres-do-Processo/>

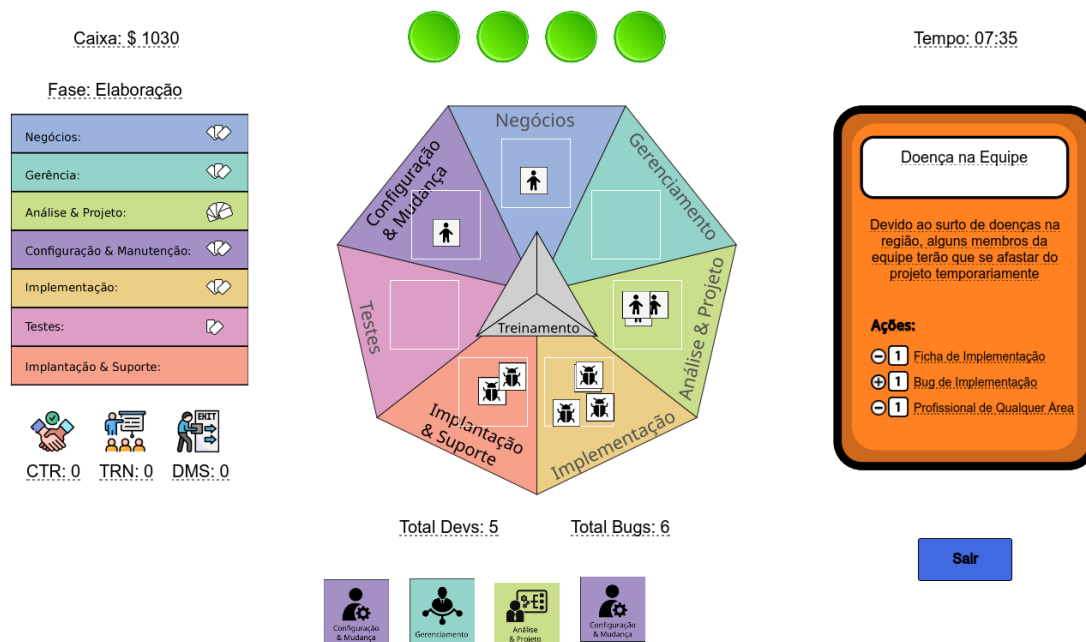


Figura 2. Tela principal do jogo com painel de trabalho, fichas de trabalho, bugs e desenvolvedores alocados.

O jogo começa com o jogador clicando no deck de cartas, o qual é formado por um baralho dinâmico composto por 62 cartas de evento, divididas em três categorias:

- **Cartas Laranja (Negativas):** Representam eventos adversos do cotidiano de projetos reais, introduzindo bugs, removendo recursos ou impondo penalidades financeiras. Exemplos incluem “Erros de Estimativa de Produção”, “Mudança de Requisitos”, “Ataque de Hackers” e “Assédio Moral no Trabalho”. Cada carta especifica quantos bugs de quais áreas são gerados, ou quantas fichas/desenvolvedores/dinheiro são perdidos.
- **Cartas Azuis (Positivas):** Eventos favoráveis que incrementam o orçamento, geram fichas extras ou concedem ações especiais gratuitas. Exemplos incluem “Vendas Acima do Esperado”, “Inovação Tecnológica Concluída”, e cartas de “Treinamento”, “Contratação” e “Demissão”, onde estas últimas permitem ao jogador reconfigurar sua equipe sem custo monetário.
- **Cartas Verdes (Neutras):** Representam estabilidade no projeto, sem impacto direto nas variáveis do jogo. Sua função é criar respiro estratégico entre eventos significativos, além de aumentar a variabilidade do baralho.

Bugs são representados visualmente por tokens com ícone de inseto, posicionados aleatoriamente dentro das zonas de área correspondentes. São entidades colidíveis que bloqueiam a entrega de fichas de trabalho, onde uma área de conhecimento com bugs ativos impede que fichas ali depositadas sejam contabilizadas para a passar de fase. O jogador resolve bugs arrastando fichas de trabalho compatíveis (mesma área) sobre eles. Ao colidirem, ambos os tokens são removidos, representando assim o esforço corretivo que consome a capacidade produtiva da equipe na conclusão da fase do projeto.

A simulação da gestão de pessoas constitui um dos diferenciais pedagógicos do jogo. O jogador pode: (i) *contratar* novos desenvolvedores ao custo de \$50 cada, clicando

no ícone de contrato; (ii) *demitir* profissionais ao custo de \$70 cada, arrastando-os para a zona de demissão; e (iii) *treinar* e realocar desenvolvedores entre áreas ao custo de \$20 por treinamento, arrastando-os para uma das sete áreas de conhecimento desejadas do painel de trabalho central. Cada desenvolvedor alocado em uma área específica gera automaticamente uma ficha de trabalho correspondente a cada duas rodadas, simulando ciclos de produtividade iterativa de uma empresa.

A partida termina caso uma de três condições de derrota seja atingida: orçamento zerado (`currentBudget` ≤ 0), acúmulo crítico de 30 ou mais bugs, ou esgotamento do tempo de uma partida (20 min). Caso o jogador obtenha a conclusão bem-sucedida de todas as quatro fases do RUP, ele vence a partida recebendo uma classificação de *rank* que varia de E (mínimo) a S (máximo). O *rank* é calculado com base no orçamento remanescente e no número total de bugs acumulados durante toda a partida (Tabela 1).

Tabela 1. Sistema de classificação por rank ao final da partida, baseado no orçamento remanescente e no número total de bugs obtidos.

Rank	Bugs Acumulados	Orçamento Mínimo
S	≤ 2	> 1000
A+	≤ 5	> 800
A–	≤ 8	> 500
B+	≤ 11	> 400
B–	≤ 14	> 300
C+	≤ 18	> 200
C–	≤ 22	> 100
D	≤ 26	> 50
E	≤ 30	> 0

Para finalizar, as principais características do jogo original foram preservadas, tais como: etapas do processo a serem cumpridas; fichas de produção; contratação, treinamento e demissão de funcionários; cartas de evento; e o controle do orçamento com incremento mensal representado por quatro turnos. Entre as modificações implementadas, o jogo passou a operar em modo *single player*, incorporando diferentes condições de derrota que incentivam o jogador a concluir as etapas do processo de desenvolvimento de forma lúdica e estratégica.

No que se refere às mecânicas adotadas, observa-se um sistema equilibrado entre aspectos econômicos e de qualidade do projeto de software. Destaca-se, nesse contexto, a dinâmica de economia circular, na qual a cada quatro rodadas processa receitas e despesas recorrentes correspondentes ao recebimento mensal do projeto e ao pagamento dos salários dos desenvolvedores, exigindo do jogador um gerenciamento estratégico do tamanho da equipe em relação ao orçamento disponível.

Adicionalmente, o sistema de ranqueamento final (E–S) tem como base uma matriz que correlaciona o orçamento remanescente ao acúmulo de bugs, penalizando estratégias que priorizam progresso acelerado em detrimento da qualidade do produto. Por fim, a variabilidade das partidas é assegurada por um mecanismo de aleatoriedade controlada na distribuição das categorias de cartas, garantindo que cada sessão seja distinta, ao mesmo tempo em que se preserva o balanceamento pedagógico da simulação.

4.2. Validação do Jogo

Para realizar a validação do jogo, uma coleta de dados foi realizada, seguindo os critérios do questionário MEEGA+ [Giani Petri 2019], logo após a realização de partidas do jogo com alunos de uma turma de ES de um curso de Engenharia da Computação no período de Março a Abril de 2026, por meio de link para um *google form*. Dos entrevistados no processo, o perfil dos jogadores obtidos reflete um público majoritariamente masculino, com idade entre 18 e 28 anos, e com uma exposição bastante frequente ao uso de jogos digitais (Figura 3) num total de 34 entrevistados.



Figura 3. Informações demográficas obtidas na pesquisa de validação.

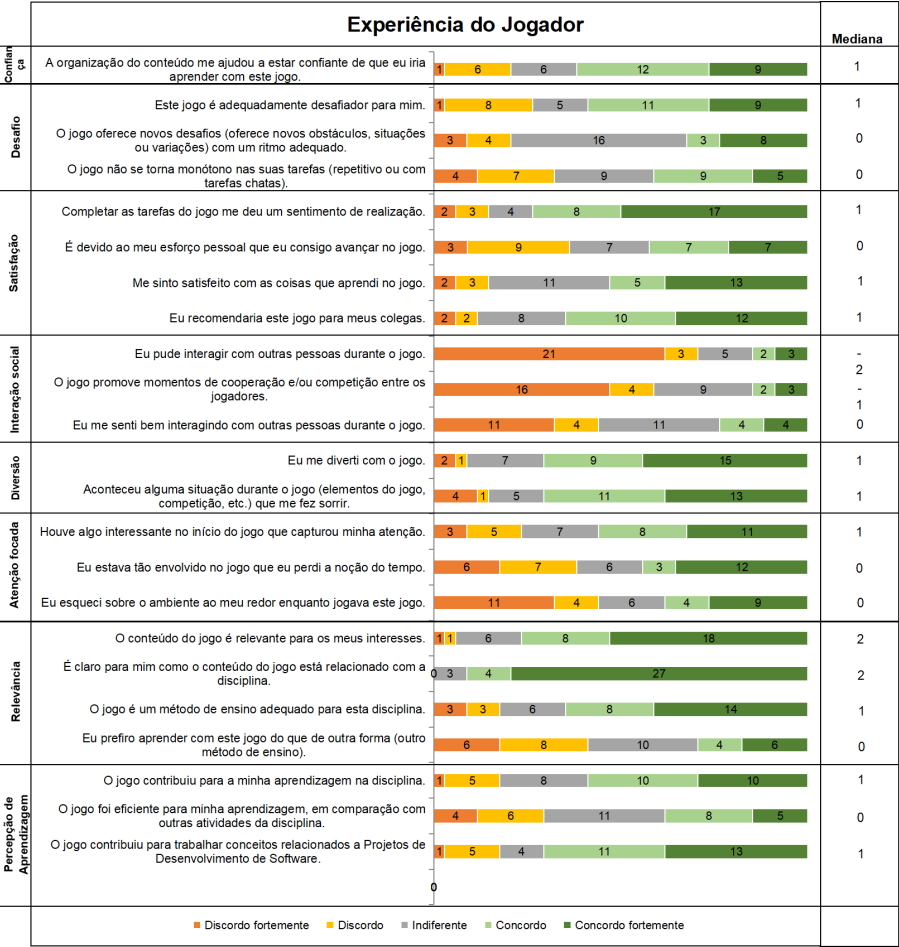


Figura 4. Respostas do questionário referentes a experiência do jogador.

Com relação as respostas adquiridas no questionário aplicado, estas demonstram os diferentes aspectos abordados pelo questionário MEEGA+ (Figura 4 e Figura 5). No

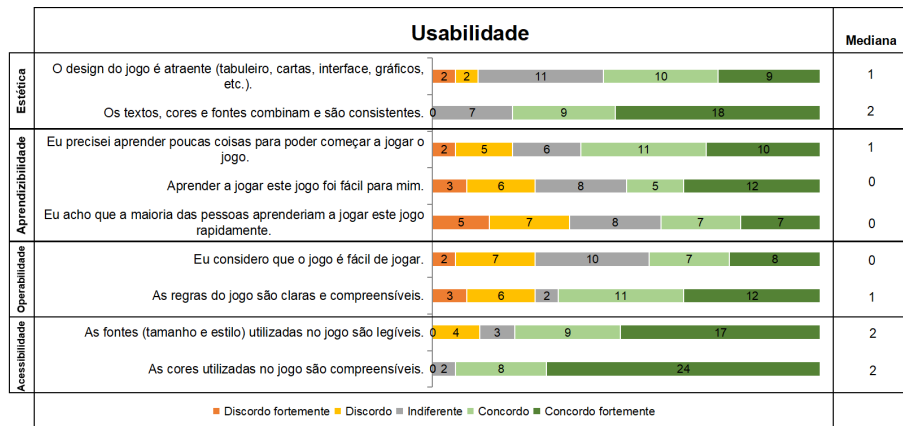


Figura 5. Respostas do questionário referentes a usabilidade do jogo.

que se refere à experiência do jogador, os itens relacionados à diversão apresentaram médias elevadas, sugerindo que o jogo conseguiu capturar o interesse dos alunos e manter sua atenção ao longo da atividade. Esse resultado é reforçado pelos indicadores de atenção focada, que demonstram baixa ocorrência de distrações, indicando que os jogadores permaneceram imersos durante a interação. A dimensão de desafio também apresentou resultados positivos, evidenciando que o jogo foi percebido como adequadamente balanceado. Esse equilíbrio é fundamental para sustentar o estado de fluxo e favorecer a motivação intrínseca dos jogadores. Além disso, os indicadores de satisfação e confiança sugerem que os participantes se sentiram recompensados ao completar as tarefas propostas, bem como confiantes em sua capacidade de progredir no jogo. Quanto à relevância percebida, os resultados indicam que os jogadores reconheceram a utilidade do jogo para o aprendizado, reforçando seu potencial como ferramenta educacional complementar. Por outro lado, a dimensão de interação social apresentou resultados negativos, configurando-se como um dos principais pontos de fragilidade do jogo. Esses resultados indicam que os participantes perceberam limitações nas oportunidades de colaboração, comunicação ou competição significativa entre os jogadores.

Em relação à usabilidade, os gráficos apontam avaliações predominantemente positivas nos aspectos de estética, indicando que os elementos visuais foram considerados agradáveis e adequados. A dimensão de aprendizagem também apresentou bons resultados, sugerindo que os jogadores conseguiram compreender rapidamente as regras e mecânicas do jogo, mesmo sem suporte extensivo. A operabilidade foi outro ponto de destaque, com avaliações que indicam facilidade de interação e controle, demonstrando que a interface do jogo não impôs barreiras significativas ao usuário. Esse fator é crucial para garantir que o foco do jogador permaneça na experiência e nos objetivos educacionais, e não na superação de dificuldades técnicas.

Com relação aos comentários realizados pelos jogadores que testaram o jogo, diversos participantes destacaram como principais pontos fortes a dinamicidade e o caráter envolvente da experiência, que contribuem para manter o interesse ao longo da partida. Também foi valorizado o conceito central do jogo, especialmente sua proposta de simular o gerenciamento de processos e equipes de desenvolvimento, o que reforça sua adequação ao contexto educacional. Além disso, o formato inspirado em jogos de tabuleiro, aliado ao uso de cartas, foi percebido como um elemento positivo por facilitar

a compreensão das mecânicas e tornar a experiência mais acessível e lúdica.

Com relação aos pontos fracos, os comentários convergem principalmente para as cartas de efeito negativo consideradas desproporcionais por alguns jogadores em determinadas situações. Também foram apontadas dificuldades pontuais na compreensão de regras e mecânicas específicas, indicando a necessidade de maior clareza nas instruções e no suporte oferecido ao jogador durante a partida. Adicionalmente, alguns participantes relataram inconsistências na execução de certos elementos do jogo, sugerindo a presença de limitações técnicas ou de implementação que podem impactar a fluidez da experiência. Embora esses aspectos não tenham sido amplamente recorrentes, sua ocorrência indica oportunidades de refinamento importantes para versões futuras.

Por fim, parte dos jogadores afirmou não identificar pontos fracos relevantes, reforçando a percepção positiva geral do jogo. Em conjunto, os resultados qualitativos indicam que o jogo apresenta uma base consistente e potencial educacional significativo, ao mesmo tempo em que evidencia a necessidade de ajustes no nível de dificuldade e na clareza das mecânicas para aprimorar a experiência final do usuário.

5. Conclusões e Trabalhos Futuros

O desenvolvimento do Mestres do Processo Digital demonstrou a viabilidade de transformar dinâmicas de tabuleiros físicos em experiências digitais para o ensino de Engenharia de Software, promovendo aprendizagem ativa e imersiva. A arquitetura adotada, baseada no padrão EnSCEVA, mostrou-se eficaz ao desacoplar a especificação declarativa (JSON) do mecanismo de execução (Godot), favorecendo a manutenção, a extensibilidade e o reuso dos componentes pedagógicos.

Os resultados da validação quantitativa e qualitativa indicam que o jogo cumpre seu papel como ferramenta complementar de ensino, sendo percebido pelos estudantes como relevante para a disciplina e capaz de sustentar o engajamento e o estado de fluxo durante a experiência. Ainda assim, a análise evidenciou oportunidades de melhoria, especialmente no refinamento da curva de dificuldade e na ampliação das mecânicas de interação social, que permanecem limitadas na versão atual para um único jogador.

Apesar desses avanços, os resultados devem ser interpretados à luz de algumas limitações do estudo. A amostra, definida por conveniência e composta por um número reduzido de participantes em um contexto específico, pode restringir a generalização dos achados. Além disso, a avaliação baseou-se predominantemente em medidas subjetivas, como experiência do usuário e percepção de aprendizagem, o que pode introduzir vieses, ainda que fundamentadas em modelos validados. A ausência de um pré-teste também limita a mensuração mais precisa dos ganhos de aprendizagem, enquanto o próprio contexto de aplicação em sala pode ter influenciado as respostas dos participantes.

Como trabalhos futuros, pretende-se realizar estudos futuros com amostras mais amplas e delineamentos experimentais controlados, reintroduzir elementos colaborativos do jogo original por meio de modos híbridos, refinar a curva de dificuldade com base nos feedbacks obtidos e implementar um sistema de tutorial adaptativo com dicas contextuais para apoiar a aprendizagem. Adicionalmente, planeja-se expandir o conjunto de cenários e cartas, abrangendo um espectro mais amplo de situações reais de gestão de projetos, bem como conduzir estudos que avaliem a retenção de conhecimento e a transferência das competências desenvolvidas para contextos profissionais.

Referências

- Alotaibi, M. S. (2024). Game-based learning in early childhood education: a systematic review and meta-analysis. *Frontiers in psychology*, 15:1307881.
- Brasil (2016). Conselho nacional de saúde, homologação da resolução cns no 510, de 07 de abril de 2016, nos termos do decreto de delegação de competência de 12 de novembro de 1991. https://bvsms.saude.gov.br/bvs/saudelegis/cns/2016/res0510_07_04_2016.html. (Acessado em 20/01/2025).
- Calderón, A. e Ruiz, M. (2013). Prodec: a serious game for software project management training. In *Proceedings of the 8th International Conference on Software Engineering Advances, Venice, Italy*.
- Dantas, A. R., de Oliveira Barros, M., e Werner, C. M. L. (2004). A simulation-based game for project management experiential learning. In *SEKE*, volume 19.
- Giani Petri, Christiane Gresse von Wangenheim, A. F. B. (2019). Meega+: Um modelo para a avaliação de jogos educacionais para o ensino de computação.
- Hays, R. T. e Singer, M. J. (2012). *Simulation fidelity in training system design: Bridging the gap between reality and training*. Springer Science & Business Media.
- Kosa, M., Yilmaz, M., O'Connor, R., e Clarke, P. (2016). Software engineering education and games: a systematic literature review. *Journal of Universal Computer Science*, 22(12):1558–1574.
- Navarro, E. O. e van der Hoek, A. (2004). Simse: An interactive simulation game for software engineering education. In *CATE*, volume 1, pages 12–17.
- Oguz, D. e Oguz, K. (2019). Perspectives on the gap between the software industry and the software engineering education. *Ieee Access*, 7:117527–117543.
- Sarinho, V. T. (2019a). A board game proposal for teaching software management and software development process. pages 559–562.
- Sarinho, V. T. (2019b). Masters of the process: a board game proposal for teaching software management and software development process. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, pages 532–536.
- Sarinho, V. T. (2024). Epubforge - a simplified game engine based on features for the production of ebook games. In *XXIII Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames) - Anais Estendidos*, Porto Alegre, RS, Brasil. Sociedade Brasileira de Computação.
- Soriano-Sánchez, J. G., Quijano López, R., e Airado Rodríguez, D. (2026). The impact of game-based learning on motivation, self-efficacy, and academic achievement in the natural sciences: A meta-analysis. *Education Sciences*, 16(1):122.
- Su, F. e Zou, D. (2024). A systematic review of game-based assessment in education in the past decade. *Knowledge Management & E-Learning*, 16(3):451–476.
- Suescún Monsalve, E., Toro, M., Mazo, R., Velasquez, D., Vallejo, P., Cardona, J. F., Rincón, R., Werneck, V. M., e Leite, J. C. S. d. P. (2018). Simules-w: A collaborative game to improve software engineering teaching. *Computación y Sistemas*, 22(3):953–983.

Von Wangenheim, C. G. e Shull, F. (2009). To game or not to game? *IEEE software*, 26(2):92–94.