

# GamES: Promovendo a Educação em Engenharia de Software através de Jogos Sérios

Filipe Furtado<sup>1</sup>, Ana Clara Miranda<sup>1</sup>, Pedro Henrique Valle<sup>1</sup>,  
Marcelo Renhe<sup>1</sup>, Alessandra de Oliveira<sup>1</sup>

<sup>1</sup>Depto de Ciência da Computação - Universidade Federal de Juiz de Fora (UFJF)

{brinati, marcelo.caniato}@ice.ufjf.br, ana.laredo@estudante.ufjf.br  
{alessandrea.oliveira, pedrohenrique.valle}@ufjf.br

**Resumo.** Jogos sérios têm sido empregados para facilitar o ensino-aprendizagem da Engenharia de Software (ES) e integrar teoria e prática profissional. Essa abordagem visa promover maior envolvimento dos estudantes na disciplina, ao mesmo tempo que fortalece conceitos teóricos fundamentais. Este estudo descreve o desenvolvimento de um jogo sério em formato de romance visual, concebido para reforçar os conceitos ensinados na disciplina de ES por meio de quizzes interativos e simulações de papéis (role play). São apresentados tanto o processo de concepção do jogo quanto a estrutura do software desenvolvido. Por fim, são discutidas brevemente a situação atual do jogo e as experiências recentes de sua utilização.

**Palavras-Chave** Engenharia de Software, ensino-aprendizagem, jogo sério.

## 1. Introdução

O processo tradicional de ensino de Engenharia de Software (ES) compreende diversos conteúdos teóricos, muitas vezes, falhando em fazer a ponte entre teoria e prática [Mendes et al. 2019], podendo resultar em perda de interesse e abandono da disciplina [Hoed 2016]. Professores têm buscado diferentes métodos para lidar com este problema, sendo um deles o uso de jogos sérios de forma complementar [Santos et al. 2020].

Um jogo sério tem por objetivo principal estar além do entretenimento [Vargas et al. 2014]. O uso de jogos sérios como apoio ao processo de ensino-aprendizagem de ES permite tratar os tópicos da disciplina de forma lúdica, inclusive com simulações de ambientes práticos que ajudam a reforçar o que é visto no modelo tradicional de ensino [Susi et al. 2007, Mendes et al. 2019]. Tal abordagem já apresenta alguns resultados positivos quanto ao aumento de interesse e motivação dos alunos [Kosa et al. 2016]. Diferentes exemplos de jogos podem ser citados nesse contexto. Baker et al. (2005) apresenta um experimento para avaliar os impactos do *card game Problems and Programmers*, que simula um ambiente onde os jogadores assumem papéis específicos em uma equipe de desenvolvimento e participam de várias fases que representam as etapas do ciclo de vida do desenvolvimento de software. Já o trabalho de von Wangenheim et al. (2009) aborda o jogo sério *X-MED*, que visa exercitar a aplicação de métricas de software voltada para a gerência de projetos. No contexto do jogo anterior, *Back to Penelope* [Marín et al. 2019] busca ensinar o processo de estimativa de esforço usando o método COSMIC. Outro exemplo da literatura é o *SCRUMI*, um jogo eletrônico de tabuleiro desenvolvido para ensinar os principais conceitos de Scrum [Souza et al. 2017]. No contexto de teste de software, pode-se mencionar o jogo *IslandTest* [Queiroz et al. 2019], que desafia o jogador a aplicar conceitos de teste em um cenário inspirado pelo seriado *Lost*. Há ainda o trabalho de Vizcaíno et al. (2019), que propõe o jogo sério *GSD-Aware*, criado para ensinar sobre os desafios do desenvolvimento de software global, tópico pouco abordado em cursos de computação. Por fim, os

jogos *AD-RPG - Adaptive Role Playing Game* [Lemos et al. 2020] e *REQUIREMENTS ENGINEERING (Requengin)* [García et al. 2020] abordam a parte de requisitos, sendo que o primeiro possui um enfoque no ensino de elicitación de requisitos, enquanto o último introduz os processos e tarefas da Engenharia de Requisitos conforme o padrão ISO/IEC/IEEE 29148:2011.

Diante disso, este trabalho propõe o jogo sério GamES, que utiliza *quizzes* e elementos lúdicos para aumentar o engajamento e a eficácia no ensino-aprendizagem de ES. A seguir, o GamES e sua implementação são descritos, seguidos das considerações finais.

## 2. Game High Concept

O primeiro passo na concepção do jogo foi definir os elementos que seriam utilizados. Para superar a dificuldade em ligar o contexto profissional a conceitos vistos em aula, a narrativa e o *role-play* foram introduzidos como pilares do jogo, estruturando-o como um romance visual (*visual novel*). Esses elementos foram encaixados em um enredo inspirado na crise do software, um dos primeiros tópicos da disciplina de ES. No jogo, o jogador vivencia uma hipotética segunda crise do ponto de vista de um engenheiro de software, resolvendo problemas que surgem para ele e seus colegas. Para que o jogador consiga enfrentar a crise, os cenários da narrativa levam-no a recordar o histórico de evolução da ES. Esse e outros tópicos da ementa da disciplina são explorados em 6 semanas, estruturadas na seguinte ordem: Análise de Requisitos, Ferramentas CASE, Gerência de Projetos, Processo de Software, Qualidade de Software e Gerência de Configuração de Software. Além disso, há uma Semana 0 como tutorial do jogo.

Para introduzir os conceitos de ES, optou-se pelo *quiz* como a principal ferramenta de aquisição de conhecimento. As perguntas são diretamente relacionadas ao conteúdo da disciplina, conectando o que é discutido em sala com o contexto do jogo. Foi adicionado um retorno aos jogadores na forma de um e-mail, informando as perguntas respondidas corretamente ao fim de uma semana.

A Figura 1 apresenta os principais cenários do jogo. O cenário que introduz o enredo pode ser visto na Figura 1a. O cenário do *quiz* é apresentado na Figura 1b, enquanto a etapa de *feedback* é ilustrada na Figura 1c. Por fim, a Figura 1d apresenta a sala onde os empregados da empresa se encontram no meio do dia. Além de reforçar os conceitos apresentados em outros cenários, a sala é um ambiente para os jogadores conhecerem melhor os personagens do jogo, ajudando a diversificar a história.

Há ainda um elemento de satisfação da empresa, que é influenciado pelo número de questões acertadas pelo jogador. Caso muitos erros sejam cometidos, o jogador pode chegar ao “final ruim” do jogo. Com o objetivo de não apenas penalizar o desempenho do jogador, também foi criado um sistema de recompensas para reforçar os acertos. Ao final de cada semana, um relatório é apresentado ao jogador, com a atribuição de moedas em função do resultado do *quiz*. Essas moedas podem ser usadas na compra de itens para customização da sala do *quiz*. A loja de itens pode ser acessada diretamente a partir da tela do relatório, como pode ser observado nas Figuras 2a e 2b.

## 3. Visão Geral do Software

Foi utilizado o motor de jogos Unity<sup>1</sup>, além da Inky<sup>2</sup> para o desenvolvimento do jogo. Para estruturar e implementar as narrativas e diálogos, criando ramificações significativas

---

<sup>1</sup><https://unity.com/pt>

<sup>2</sup><https://www.inklestudios.com/ink/>



Figura 1. Principais cenários do jogo



Figura 2. Cenários que representam as últimas cenas da semana no jogo.

na história baseadas nas escolhas dos jogadores. Por fim, Aseprite<sup>3</sup> foi a escolhida para a criação de arte e animação em *pixel art*. A estrutura geral do GameS (Figura 3), é relativamente simples, utilizando principalmente a ferramenta de criação de cenas da Unity. Existem cinco gerenciadores principais para o funcionamento do jogo. O **gerenciador de dados** armazena todas as informações do jogador, desde a quantidade de perguntas que ele acertou até em qual semana ele se encontra, usadas para a geração do relatório semanal. O **gerenciador de enredo** controla o carregamento das etapas do romance visual quando requisitado, enquanto o **gerenciador de quiz** carrega as perguntas a serem apresentadas e provê *feedback* ao jogador. O **gerenciador de loja** controla quais objetos foram comprados e quais foram ativados para visualização no cenário do *quiz*. Por fim, o **gerenciador de personagem** controla as ações do jogador nos cenários possíveis.

A Figura 4 ilustra o fluxo de interações do usuário com o jogo para uma semana da narrativa. O jogador começa em um cenário de romance visual, avançando no diálogo com a tecla espaço. Em seguida, ele entra na sala do *quiz*, onde se move livremente e

<sup>3</sup><https://www.aseprite.org>

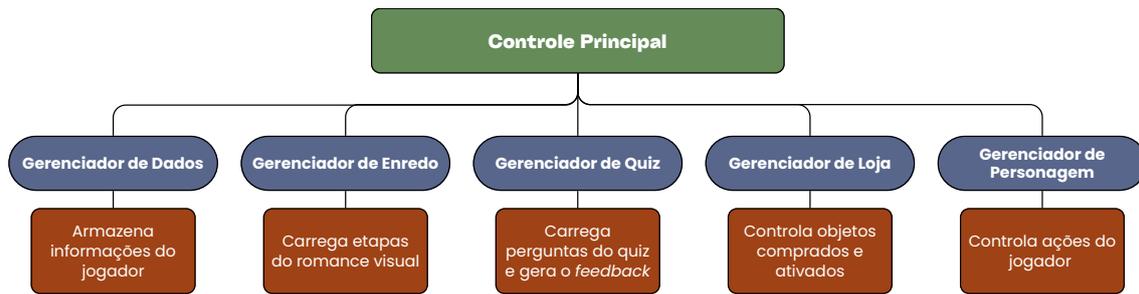


Figura 3. Gerenciadores desenvolvidos para controle dos cenários.

repete o ciclo de perguntas cinco vezes. Depois, o jogador acessa a sala de descanso, podendo interagir com o ambiente e conclui essa cena com uma etapa de novela visual. O jogador então visualiza o cenário de *feedback* com as questões certas e erradas, seguido pelo cenário de relatório, onde pode acessar a loja de itens ou continuar. Na loja, ele pode comprar itens e adicioná-los à sala do *quiz*. Finalmente, o jogador entra em um último cenário de novela visual; dependendo do desempenho no *quiz*, ele retorna à primeira cena ou vai para uma cena final que encerra o jogo. O jogo também encerra após todas as semanas serem concluídas.

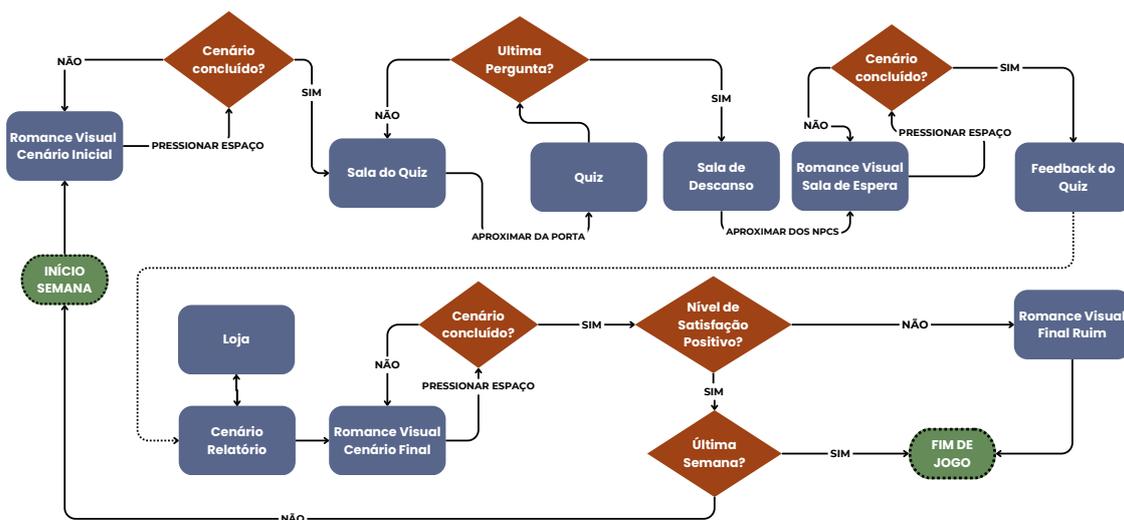


Figura 4. Fluxo de interações do usuário no decorrer de uma semana do jogo.

#### 4. Considerações Finais

Este trabalho apresentou o GameES, um jogo sério voltado para o processo de ensino-aprendizagem de ES que adota elementos de jogo como o *quiz*, a narrativa interativa e o *role-play*. Estes elementos são integrados de forma coesa para proporcionar uma experiência imersiva e educativa. A utilização de um romance visual permitiu não apenas a contextualização dos conceitos teóricos de ES, mas também a aplicação prática desses conceitos em um cenário simulado de crise do software. GameES foi avaliado em duas turmas da disciplina de ES. Durante essas iterações, foi possível observar a eficácia do jogo como ferramenta de ensino-aprendizagem, embora alguns desafios tenham sido identificados. Como trabalho futuro, um estudo experimental para avaliar o impacto de usar o jogo ao longo do semestre letivo está sendo planejado. Isso pode incluir análises longitudinais para monitorar a retenção de conhecimento e o engajamento dos alunos após o uso contínuo do jogo ao longo de semestres acadêmicos.

## Referências

- Baker, A., Navarro, E. O., e Van Der Hoek, A. (2005). An experimental card game for teaching software engineering processes. *Journal of Systems and Software*, 75:3–16.
- García, I., Pacheco, C., León, A., e Calvo-Manzano, J. A. (2020). A serious game for teaching the fundamentals of iso/iec/ieee 29148 systems and software engineering–lifecycle processes–requirements engineering at undergraduate level. *Computer Standards & Interfaces*, 67:103377.
- Hoed, R. M. (2016). Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de computação.
- 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.
- Lemos, E., Conceição, M., Nunes, F., Medina, R., e Bernardi, G. (2020). Ad-rpg: Elicitando requisitos na idade média utilizando um jogo educacional. In *Anais do XXXI Simpósio Bras. de Informática na Educação*, pages 1763–1772, POA, RS, Brasil. SBC.
- Marín, B., Vera, M., e Giachetti, G. (2019). An adventure serious game for teaching effort estimation in software engineering. In *IWSM-Mensura*, pages 71–86.
- Mendes, J., Costa, Y., Frazão, K., Santos, R., Santos, D., e Rivero, L. (2019). Identificação das expectativas e dificuldades de alunos de graduação no ensino de engenharia de software. In *Anais do XXVII Workshop Educação em Computação*, pages 334–347. SBC.
- Queiroz, R., Pinto, F., e Silva, P. (2019). Islandtest: jogo educativo para apoiar o processo ensino-aprendizagem de testes de software. In *Anais do XXVII Workshop sobre Educação em Computação*, pages 533–542, Porto Alegre, RS, Brasil. SBC.
- Santos, S. H. N., Costa, Y. d. J. S., dos Santos, D. V., Barradas Filho, A. O., Junior, J. B. B., e Cabrejos, L. J. E. R. (2020). Identificando jogos sérios para o ensino de engenharia de software no brasil através de um mapeamento sistemático. *Research, Society and Development*, 9(7):e329973702–e329973702.
- Souza, A., Seabra, R., Ribeiro, J., e Rodrigues, L. (2017). Scrumi: Um jogo virtual sério de tabuleiro para o ensino do framework scrum. In *Anais do XIII Simpósio Brasileiro de Sistemas de Informação*, pages 520–526, Porto Alegre, RS, Brasil. SBC.
- Susi, T., Johannesson, M., e Backlund, P. (2007). Serious games: An overview.
- Vargas, J. A., García-Mundo, L., Genero, M., e Piattini, M. (2014). A systematic mapping study on serious game quality. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, pages 1–10.
- Vizcaíno, A., García, F., Guzmán, I. G. R. D., e Moraga, M. Á. (2019). Evaluating gsd-aware: A serious game for discovering global software development challenges. *ACM Transactions on Computing Education (TOCE)*, 19(2):1–23.
- von Wangenheim, C., Thiry, M., Kochanski, D., Steil, L., Silva, D., e Lino, J. (2009). Desenvolvimento de um jogo para ensino de medição de software. In *Anais do VIII Simpósio Brasileiro de Qualidade de Software*, pages 46–60, POA, RS, Brasil. SBC.