

Aplicando Aprendizagem Baseada em Jogos no Ensino de Engenharia de Software por meio do Minecraft Education

Applying Game-Based Learning to Software Engineering Education through Minecraft Education

Claudia Werlich¹, Gabriel Caixeta Silva¹, Kauan Henrique Werlich²

¹Faculdade SENAC Joinville
Joinville – SC – Brasil

²Universidade do Estado de Santa Catarina (UDESC)
Joinville – SC – Brasil

{claudia.werlich,gabriel.silva}@prof.sc.senac.br,
kauan.werlich@edu.udesc.br

Abstract. Introduction: Software Engineering education presents challenges related to the practical understanding of life cycle models and student engagement. Game-Based Learning (GBL) has been investigated as an approach capable of promoting active and meaningful learning in computing courses. **Objective:** This study proposes to analyze the application of Game-Based Learning through Minecraft Education, comparing the Waterfall and Incremental life cycle models in a practical activity involving the development of virtual thematic villages. **Methodology or Steps:** The research was conducted with 30 undergraduate students from a Systems Analysis and Development program, organized into teams that applied different development models during the collaborative construction of virtual environments. Data collection occurred through direct observation of the development process, communication between teams, and adaptation to requirement changes. **Results:** The results indicate that the Incremental model favored greater flexibility, communication, and engagement, while the Waterfall model presented limitations when dealing with requirement changes. It is concluded that immersive virtual environments based on games contribute to active and contextualized learning of software development methodologies.

Keywords Game-Based Learning, Software Engineering Education, Waterfall Model, Incremental Model, Minecraft Education, Active Learning.

Resumo. Introdução: O ensino de Engenharia de Software apresenta desafios relacionados à compreensão prática dos modelos de ciclo de vida e ao engajamento dos estudantes. A Aprendizagem Baseada em Jogos (Game-Based Learning – GBL) tem sido investigada como uma abordagem capaz de promover aprendizagem ativa e significativa em cursos da área de computação. **Objetivo:** Este estudo propõe analisar a aplicação de Game-Based Learning por meio do Minecraft Education, comparando os modelos de ciclo de vida Cascata e Incremental em uma atividade prática de desenvolvimento

*de vilas temáticas virtuais. **Metodologia ou Etapas:** A pesquisa foi conduzida com 30 estudantes do curso superior de Análise e Desenvolvimento de Sistemas, organizados em equipes que aplicaram diferentes modelos de desenvolvimento durante a construção colaborativa de ambientes virtuais. A coleta de dados ocorreu por meio de observação direta do processo de desenvolvimento, comunicação entre equipes e adaptação a mudanças de requisitos. **Resultados:** Os resultados indicam que a equipe que utilizou o modelo Incremental apresentou maior capacidade de adaptação às mudanças de requisitos e interações mais frequentes entre os participantes, enquanto o modelo Cascata apresentou dificuldades diante de alterações introduzidas durante o desenvolvimento. Conclui-se que ambientes virtuais imersivos baseados em jogos contribuem para a aprendizagem ativa e contextualizada de metodologias de desenvolvimento de software.*

Palavras-Chave Aprendizagem Baseada em Jogos, Engenharia de Software, Modelo Cascata, Modelo Incremental, Minecraft Education, Metodologias Ativas.

1. Introdução

O ensino de Engenharia de Software apresenta desafios relacionados à compreensão prática dos modelos de ciclo de vida e à sua aplicação em situações reais de desenvolvimento. Embora conceitos como levantamento de requisitos, planejamento, projeto, implementação e testes sejam amplamente abordados em disciplinas teóricas, muitos estudantes encontram dificuldades para compreender como diferentes modelos de processo influenciam a organização do trabalho em equipe, a comunicação entre os participantes e a adaptação a mudanças de requisitos ao longo do projeto [Pressman e Maxim 2020, Sommerville 2019]. Entre os modelos clássicos destacam-se o Cascata, caracterizado por um fluxo sequencial de desenvolvimento, e o Incremental, baseado na entrega progressiva de funcionalidades e na incorporação contínua de mudanças.

Nesse contexto, metodologias ativas têm sido adotadas como alternativa para aproximar teoria e prática, colocando o estudante no centro do processo de aprendizagem por meio da resolução de problemas, da colaboração e do desenvolvimento de projetos [Irbadra et al. 2025]. Entre essas abordagens, a Aprendizagem Baseada em Jogos (*Game-Based Learning* – GBL) utiliza jogos digitais como ambiente de aprendizagem, favorecendo motivação, experimentação e aprendizagem significativa [Silva et al. 2024, Werbach e Hunter 2012]. No ensino de Engenharia de Software, essa abordagem permite simular situações próximas às encontradas em projetos reais, proporcionando aos estudantes oportunidades para experimentar processos de desenvolvimento, analisar problemas e avaliar diferentes estratégias de solução.

O Minecraft Education foi selecionado como ambiente de aprendizagem por oferecer um espaço virtual colaborativo que possibilita representar visualmente estruturas, artefatos e processos. Embora não reproduza um ambiente profissional de desenvolvimento de *software*, sua flexibilidade permite simular cenários nos quais os estudantes precisam elicitar requisitos, planejar atividades, organizar equipes, implementar soluções e responder a mudanças ao longo do desenvolvimento. Essas características tornam o ambiente adequado para explorar, de forma prática, diferenças

entre modelos de processo.

Apesar do crescente número de estudos sobre GBL no ensino de Computação, observa-se que a maior parte das pesquisas concentra-se em aspectos como motivação, ensino de programação ou desenvolvimento de habilidades gerais. São menos frequentes os trabalhos que investigam o uso de ambientes virtuais imersivos para apoiar o ensino de Engenharia de Software e, particularmente, aqueles que utilizam o Minecraft Education para comparar diferentes modelos de ciclo de vida em uma mesma atividade prática.

Diante desse cenário, este trabalho investiga a aplicação da Aprendizagem Baseada em Jogos por meio do Minecraft Education para comparar os modelos Cascata e Incremental em uma atividade colaborativa de Engenharia de Software. Para isso, foi conduzido um estudo de caso com estudantes de graduação, no qual equipes desenvolveram vilas temáticas virtuais utilizando diferentes modelos de processo. A proposta permitiu observar aspectos relacionados ao engajamento, à colaboração, à adaptação a mudanças de requisitos e ao desenvolvimento de competências técnicas e interpessoais, contribuindo para a discussão sobre o potencial pedagógico de ambientes virtuais imersivos no ensino de Engenharia de Software.

2. Trabalhos Relacionados

A GBL tem sido amplamente investigada como estratégia para promover aprendizagem ativa em diferentes níveis de ensino, especialmente em áreas STEAM, nas quais a experimentação, a resolução de problemas e a colaboração desempenham papel central na construção do conhecimento [Gui et al. 2023]. Evidências de meta-análises indicam que o uso de jogos digitais educacionais pode contribuir para melhorias no desempenho acadêmico, no engajamento e na motivação dos estudantes quando integrado a atividades alinhadas aos objetivos de aprendizagem [Gui et al. 2023].

No contexto da Computação, estudos apontam que ambientes baseados em jogos favorecem a compreensão de conceitos abstratos, o desenvolvimento do pensamento crítico e a aprendizagem colaborativa [Lampropoulos e Kinshuk 2024, Videnovik et al. 2023]. Essas características tornam o GBL particularmente adequado para disciplinas que envolvem processos complexos e tomada de decisão, como Engenharia de Software.

Entre os ambientes utilizados para fins educacionais, o Minecraft Education destaca-se por oferecer um espaço virtual colaborativo que favorece a construção de soluções, a comunicação entre estudantes e a resolução de problemas. Estudos indicam que sua utilização contribui para o desenvolvimento de competências como criatividade, pensamento crítico, colaboração e aprendizagem contextualizada [Alawajee e Delafield-Butt 2021, Liu 2024, Marnewick e Chetty 2021].

Além disso, [Suwannik 2025] demonstra que o ambiente pode ser utilizado para o ensino de diferentes paradigmas de programação, incluindo conceitos fundamentais, programação orientada a objetos, programação orientada a eventos e programação paralela, evidenciando a versatilidade do Minecraft Education como plataforma para o ensino de Computação. Também é destacado o potencial do Redstone para representar conceitos computacionais de forma visual e interativa, ampliando as possibilidades de aprendizagem prática [Liu 2024].

Entre os ambientes utilizados para fins educacionais, o Minecraft Education destaca-se por oferecer um espaço virtual colaborativo que favorece a construção de soluções, a comunicação entre estudantes e a resolução de problemas. Os trabalhos relacionados relatam contribuições para o desenvolvimento de competências como criatividade, pensamento crítico, colaboração e aprendizagem contextualizada. Esses resultados sugerem que ambientes virtuais imersivos podem aproximar conceitos teóricos de situações práticas de aprendizagem.

Embora os benefícios do GBL estejam bem documentados na literatura, a maior parte das pesquisas concentra-se em aspectos relacionados ao ensino de programação, motivação ou desenvolvimento de habilidades gerais em Computação. Estudos que exploram especificamente o ensino de Engenharia de Software em ambientes virtuais imersivos ainda são menos frequentes e, até onde foi possível identificar, não foram encontrados trabalhos que utilizassem o Minecraft Education para comparar, em uma mesma intervenção pedagógica, diferentes modelos de ciclo de vida de *software*.

Diante dessa lacuna, este trabalho investiga a utilização do Minecraft Education como ambiente de Aprendizagem Baseada em Jogos para apoiar o ensino de Engenharia de Software, comparando, em uma atividade colaborativa, os modelos Cascata e Incremental. Dessa forma, busca-se ampliar as evidências sobre o potencial de ambientes virtuais imersivos para favorecer a aprendizagem de processos de desenvolvimento de *software*.

3. Metodologia

Este estudo caracteriza-se como uma pesquisa aplicada, de abordagem qualitativa e caráter exploratório, conduzida por meio de um estudo de caso educacional. Essa estratégia foi adotada por possibilitar a investigação de fenômenos contemporâneos em seu contexto real de ocorrência, permitindo analisar como a Aprendizagem Baseada em Jogos influencia o processo de ensino e aprendizagem em Engenharia de Software.

Participaram da pesquisa 30 estudantes da quarta fase do curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas, matriculados na disciplina de Engenharia de Software, no quarto semestre do curso. Antes do início da intervenção, foi aplicado um questionário para identificar a familiaridade dos participantes com o Minecraft Education, indicando que 28% já possuíam experiência prévia com a plataforma. Todos os estudantes foram informados sobre os objetivos da pesquisa e participaram voluntariamente da atividade. Os dados foram analisados de forma agregada, preservando o anonimato dos participantes.

3.1. Estrutura da Intervenção Baseada em Jogos

A intervenção pedagógica foi estruturada com base nos princípios da GBL, utilizando o ambiente virtual Minecraft Education como suporte ao processo de ensino-aprendizagem. Essa abordagem possibilita a criação de contextos simulados nos quais os estudantes podem experimentar, na forma prática, conceitos fundamentais da Engenharia de Software, contribuindo para a aprendizagem ativa e contextualizada [Videnovik et al. 2023].

Os estudantes foram organizados em duas equipes, escolhidas por seus respectivos líderes. Os líderes também foram responsáveis pela divisão de tarefas entre os integrantes.

Cada equipe responsabilizou-se pelo desenvolvimento de uma vila temática virtual, concebida como uma analogia a um sistema de *software*. Nessa analogia, cada vila representava um sistema de software em desenvolvimento. A atividade teve como objetivo possibilitar a aplicação prática e comparativa de modelos de ciclo de vida de *software*, amplamente utilizados na Engenharia de Software. Foram adotados: (i) o modelo Cascata, caracterizado por um processo sequencial, estruturado em fases bem definidas, como levantamento de requisitos, projeto, implementação e validação, e (ii) o modelo Incremental, baseado na entrega progressiva de funcionalidades e na adaptação contínua a mudanças [Pressman e Maxim 2020].

A proposta buscou simular um ambiente de desenvolvimento de *software* no qual os estudantes precisaram elicitar e gerenciar requisitos, planejar atividades, implementar soluções e validar resultados. Durante a execução, foram introduzidas mudanças de requisitos, refletindo a natureza dinâmica dos projetos de *software* e exigindo dos estudantes a capacidade de adaptação e replanejamento, aspectos amplamente discutidos na literatura de Engenharia de Software [Sommerville 2019]. Durante a atividade observou-se elevada participação dos estudantes nas tarefas propostas, caracterizada pela interação frequente entre os membros das equipes, pela negociação de soluções e pela tomada colaborativa de decisões relacionadas à construção das vilas.

3.2. Procedimentos da Atividade

A intervenção consistiu no desenvolvimento colaborativo de duas vilas temáticas em ambiente virtual, estruturadas para representar sistemas de *software* sob diferentes modelos de ciclo de vida, conforme detalhado na Tabela 1. As vilas foram estruturadas com base em dois contextos distintos: uma cidade futurista, com foco em tecnologia e inovação, e uma cidade sustentável, voltada à sustentabilidade e à indústria verde, alinhada aos Objetivos de Desenvolvimento Sustentável (ODS) [United Nations 2015], especialmente às metas relacionadas à inovação, infraestrutura e cidades sustentáveis.

Tabela 1. Estrutura das vilas temáticas

Vila	Foco	Elementos obrigatórios	Elementos opcionais
Futurista (Smart City)	Tecnologia e automação	• Casas inteligentes • Transporte automatizado • Central de energia • Torre de comunicação • Arranha-céus	• Hospital inteligente • Banco digital • Drones • Centro de controle urbano • Parque tecnológico
Sustentável (Industrial 4.0)	Sustentabilidade e indústria verde	• Hortas automatizadas • Reciclagem • Lago artificial • Casas ecológicas • Biblioteca digital	• Eco-escola • Usina de energia limpa • Transporte verde • Parque ecológico • <i>Hub</i> de inovação

As atividades práticas ocorreram ao longo de quatro semanas, totalizando 12 horas-aula, e envolveram o desenvolvimento das vilas no ambiente *Minecraft Education*. Durante esse período, uma equipe desenvolveu a cidade futurista seguindo o modelo Cascata, enquanto a outra desenvolveu a cidade sustentável utilizando o modelo Incremental. Ambos os grupos receberam inicialmente os mesmos requisitos

estruturais para suas respectivas propostas, sendo posteriormente submetidos a alterações de requisitos que simulavam mudanças solicitadas por um cliente ao longo do desenvolvimento. Essa dinâmica permitiu aos estudantes vivenciar, na prática, as diferenças entre uma abordagem sequencial e outra incremental quanto ao planejamento, à organização das atividades e à adaptação às mudanças.

Como exemplo das produções desenvolvidas pelos estudantes, a Figura 1 apresenta as duas vilas construídas durante a atividade. A Figura 1(a) corresponde à cidade futurista, desenvolvida segundo o modelo Cascata, enquanto a Figura 1(b) apresenta a cidade sustentável, desenvolvida com base no modelo Incremental.

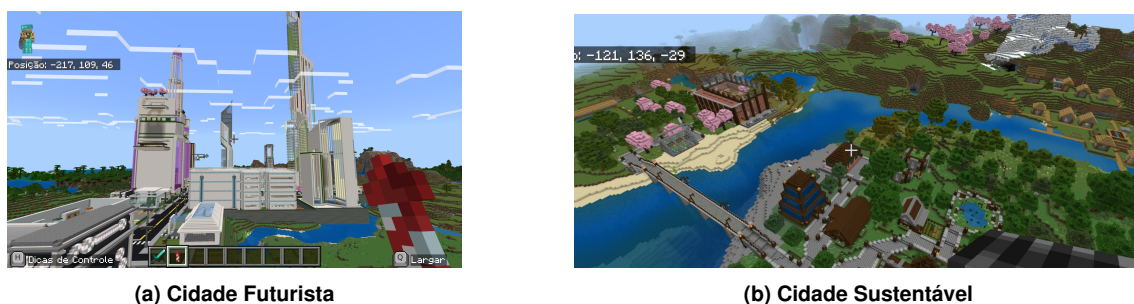


Figura 1. Vilas desenvolvidas pelos estudantes: (a) cidade futurista — modelo Cascata; (b) cidade sustentável — modelo Incremental.

A diversidade de abordagens evidencia a capacidade dos estudantes de relacionar conceitos técnicos com diferentes contextos sociais e tecnológicos, ressaltando o potencial do GBL em ambientes imersivos, como o Minecraft Education, para promover aprendizagem ativa, aplicada e significativa. Nesse processo, ambas as equipes enfrentaram desafios que envolveram construção colaborativa, organização do projeto e tomada de decisão, contribuindo para o desenvolvimento integrado de competências técnicas e socioemocionais.

Durante a execução da atividade, foram introduzidas mudanças de requisitos de forma controlada, com o objetivo de analisar a capacidade de adaptação das equipes e as limitações inerentes a cada modelo de desenvolvimento. Essas mudanças simularam situações reais de projetos de *software*, exigindo replanejamento, comunicação e tomada de decisão sob restrições. Ambas as equipes receberam requisitos obrigatórios e opcionais, garantindo comparabilidade entre os projetos e liberdade criativa controlada.

3.3. Coleta de Dados

A observação direta foi adotada por permitir a análise das interações, comportamentos e estratégias dos estudantes em ambiente real, considerando o fenômeno em seu contexto natural [González 2020]. Trata-se de uma técnica relevante na pesquisa qualitativa, pois possibilita examinar, de forma detalhada, as dinâmicas e práticas dos participantes em cenários práticos [Silva et al. 2023].

A coleta de dados ocorreu por meio de observação direta em todas as etapas da atividade, do planejamento à apresentação final dos projetos, permitindo o acompanhamento contínuo do comportamento dos estudantes em tempo real. A observação direta foi conduzida pela docente responsável pela disciplina, que

acompanhou todas as etapas da atividade. Os registros produzidos durante o desenvolvimento, incluindo o diário de campo, os artefatos elaborados pelas equipes e os relatórios finais, foram utilizados como fontes complementares de evidência para a análise qualitativa. Foram utilizados como instrumentos de coleta: (i) Observação sistemática das interações entre os estudantes; (ii) Registros em diário de campo; (iii) Análise dos artefatos produzidos pelas equipes; (iv) Relatórios finais dos grupos e (v) Apresentação dos ambientes desenvolvidos. A observação direta justifica-se pela sua capacidade de fornecer evidências empíricas detalhadas sobre as interações, comportamentos e estratégias dos estudantes em contexto real de aprendizagem.

3.4. Questões de Pesquisa

Com base nos objetivos do estudo, foram definidas cinco Questões de Pesquisa (QP), apresentadas na Tabela 2. Essas questões orientaram a coleta, a organização e a análise dos dados obtidos durante a intervenção.

Tabela 2. Questões de Pesquisa e respectivos objetivos

QP	Questão	Aspecto investigado
QP1	Qual é o impacto do uso de um ambiente virtual imersivo no engajamento dos estudantes no ensino de Engenharia de Software?	Engajamento discente
QP2	Quais diferenças podem ser observadas na aplicação dos modelos Cascata e Incremental em um contexto de Aprendizagem Baseada em Jogos?	Comparação entre modelos
QP3	Como o ambiente virtual imersivo contribui para a compreensão prática de conceitos de Engenharia de Software?	Aprendizagem prática
QP4	Como os estudantes respondem a mudanças de requisitos em um ambiente de Aprendizagem Baseada em Jogos?	Adaptação às mudanças
QP5	Quais competências técnicas e interpessoais são desenvolvidas a partir da atividade proposta?	Desenvolvimento de competências

3.5. Técnica de Análise dos Dados

Os dados foram analisados por meio de análise temática, técnica apropriada para identificar padrões de comportamento e interpretar as interações observadas durante a atividade. Inicialmente, os registros do diário de campo, as observações, os artefatos produzidos e os relatórios das equipes foram examinados de forma integrada. Em seguida, os trechos relacionados às questões de pesquisa foram codificados e organizados nas seguintes categorias de análise: (i) planejamento e organização do projeto; (ii) comunicação e colaboração; (iii) adaptação às mudanças de requisitos; (iv) impacto do modelo de desenvolvimento adotado; e (v) engajamento dos estudantes.

Por fim, a triangulação de dados foi realizada a partir da combinação de diferentes fontes de evidência, incluindo observações, relatórios, artefatos produzidos e apresentados, com o objetivo de aumentar a confiabilidade dos resultados.

4. Resultados e Discussão

A análise foi conduzida com base em observação direta, análise dos artefatos produzidos e relatos das equipes coletados durante a intervenção pedagógica, organizados de acordo com as questões de pesquisa definidas. A Tabela 3 apresenta uma síntese das respostas às questões de pesquisa, consolidando os principais achados do estudo.

Tabela 3. Síntese das respostas às questões de pesquisa

QP	Objetivo	Síntese dos Resultados
QP1	Analisar o impacto do ambiente virtual imersivo no engajamento dos estudantes	Os estudantes demonstraram participação ativa, colaboração e envolvimento contínuo nas atividades.
QP2	Comparar a aplicação dos modelos Cascata e Incremental	A equipe que utilizou o modelo Incremental apresentou maior facilidade para incorporar a mudança de requisitos.
QP3	Analisar a contribuição do ambiente imersivo na aprendizagem.	O ambiente virtual favoreceu a compreensão prática de conceitos de Engenharia de Software.
QP4	Analisar a adaptação dos estudantes às mudanças de requisitos.	Os estudantes adaptaram-se mais facilmente às mudanças de requisitos no modelo Incremental do que no modelo Cascata.
QP5	Identificar as competências desenvolvidas pelos estudantes	Desenvolvidas competências técnicas (planejamento e elicitação de requisitos) e interpessoais (comunicação, colaboração e tomada de decisão).

A análise qualitativa, baseada na observação direta e triangulação com os artefatos produzidos, evidenciou que o ambiente imersivo mostrou impacto significativo no engajamento e na participação ativa dos estudantes (QP1 e QP3). Esse engajamento não se restringiu à execução das tarefas, mas manifestou-se na intensificação das interações colaborativas, na negociação de soluções e na tomada de decisões relacionadas à construção das vilas. Tais evidências corroboram a literatura de Engenharia de Software ao indicar que a aprendizagem se torna mais efetiva quando os estudantes são inseridos em contextos que simulam práticas reais de desenvolvimento, nos quais comunicação, colaboração e compreensão do domínio são fundamentais [Sommerville 2019].

Além disso, a possibilidade de visualizar concretamente os artefatos produzidos no ambiente virtual reforça a ideia de que abordagens experiencialmente orientadas contribui para a internalização de conceitos abstratos, conforme discutido por [Pressman e Maxim 2020], especialmente no que se refere à compreensão das relações entre requisitos, projeto e implementação. Sob essa ótica, o ambiente baseado em jogo atuou como mediador do processo de aprendizagem, promovendo não apenas engajamento, mas também maior profundidade na construção do conhecimento.

No que se refere à comparação entre os modelos de desenvolvimento e à resposta às mudanças de requisitos (QP2 e QP4), os resultados corroboram a literatura clássica de Engenharia de Software. A introdução de um novo requisito após aproximadamente 60% da execução evidenciou as limitações do modelo Cascata em contextos sujeitos à variabilidade. Como destacam [Sommerville 2019], esse modelo pressupõe estabilidade dos requisitos e forte dependência entre fases, dificultando a incorporação de mudanças

tardias. Na prática, a equipe que utilizou o modelo Cascata precisou interromper o desenvolvimento, descartar parte do trabalho realizado e não conseguiu integrar o novo requisito sem comprometer o prazo.

Em contrapartida, a equipe que adotou o modelo Incremental demonstrou maior capacidade de adaptação, incorporando a nova funcionalidade de forma mais eficiente, em consonância com a abordagem adaptativa descrita por [Pressman e Maxim 2020]. Quanto às competências interpessoais (QP5), a formação espontânea de lideranças e a divisão do trabalho em subequipes evidenciaram o desenvolvimento de habilidades de coordenação, comunicação e gestão, competências reconhecidas na literatura como essenciais para equipes de desenvolvimento de software.

De forma geral, os resultados sugerem que o uso de GBL em ambientes imersivos constitui uma estratégia eficaz para o ensino de Engenharia de Software. A comparação entre os modelos de ciclo de vida evidenciou diferenças práticas que dificilmente seriam percebidas apenas por meio de abordagens expositivas. A triangulação dos dados coletados reforça a consistência dos achados, indicando que a abordagem proposta contribui tanto para a compreensão conceitual quanto para o desenvolvimento de competências aplicadas.

5. Conclusão

Este estudo investigou o uso da GBL como estratégia pedagógica no ensino de Engenharia de Software, por meio da utilização de um ambiente virtual imersivo baseado no Minecraft Education. Os resultados observados neste estudo indicam que a atividade baseada em GBL favoreceu elevada participação dos estudantes durante a intervenção, sugerindo potencial para apoiar experiências de aprendizagem contextualizadas. A utilização do ambiente virtual favoreceu a experimentação, a resolução de problemas e o trabalho colaborativo, elementos centrais para o desenvolvimento de competências na área de Engenharia de Software.

Somado a isso, a comparação entre os modelos de desenvolvimento indicou que abordagens iterativas, como o modelo Incremental, apresentam melhor adaptação em contextos dinâmicos, especialmente quando inseridas em ambientes baseados em jogos, nos quais mudanças e interações ocorrem de forma contínua. A principal contribuição deste trabalho consiste na proposição de uma abordagem baseada em GBL em ambiente virtual imersivo, que possibilita a simulação de cenários reais de desenvolvimento de *software* e promove a integração significativa entre teoria e prática.

Entre as ameaças à validade deste estudo, destacam-se o número restrito de participantes e a condução da atividade em um único contexto educacional, o que pode limitar a generalização dos resultados. Adicionalmente, a predominância de análise qualitativa pode introduzir vieses de interpretação, embora a utilização de múltiplas fontes de dados tenha buscado mitigar esse efeito.

Como trabalhos futuros, propõe-se investigar, em ambientes de Aprendizagem Baseada em Jogos, diferentes estratégias de organização do desenvolvimento de software, incluindo abordagens tradicionais, ágeis e híbridas. O objetivo não consiste apenas em analisar a adoção de *frameworks* específicos, mas compreender como os estudantes desenvolvem competências para avaliar criticamente os *trade-offs* associados a diferentes

processos, reconhecendo que a adequação de uma abordagem depende das características do projeto, da equipe e do contexto organizacional. Além disso, sugere-se complementar a análise qualitativa com instrumentos quantitativos capazes de mensurar os impactos do GBL sobre a aprendizagem e o engajamento dos participantes.

Referências

- Alawajee, O. e Delafield-Butt, J. (2021). Minecraft in education benefits learning and social engagement. *International Journal of Game-Based Learning*, 11(4):19–56.
- González, F. E. (2020). Reflexões sobre alguns conceitos da pesquisa qualitativa. *Revista Pesquisa Qualitativa*, 8(17):155–183.
- Gui, Y., Cai, Z., Yang, Y., Kong, L., Fan, X., e Tai, R. H. (2023). Effectiveness of digital educational game and game design in stem learning: a meta-analytic review. *International Journal of STEM Education*, 10(36).
- Irabedra, J., Solari, M., e Mousqués, G. (2025). Active learning pedagogical strategies in software engineering university courses. *CLEI Electronic Journal*, 28.
- Lampropoulos, G. e Kinshuk (2024). Virtual reality and gamification in education: a systematic review. *Educational Technology Research and Development*, 72:1691–1785.
- Liu, Q. (2024). Minecraft redstone: The gateway to engineering. *Lecture Notes in Education Psychology and Public Media*, 59:95–100.
- Marnewick, C. e Chetty, J. (2021). Mining and crafting a game to teach research methodology. *International Journal of Educational Technology in Higher Education*, 18(62).
- Pressman, R. S. e Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education, New York, 9th edition.
- Silva, P. B., Souza, P. V. S., e Freire, F. d. S. (2023). Observação como técnica de pesquisa qualitativa: panorama em periódicos brasileiros. *Revista Ciências Sociais em Perspectiva*, 22(42):43–64.
- Silva, S. V., Freire, E. F. d. S., e Santos, R. P. d. (2024). Gamificação em engenharia de software: evidências do processo de ensino-aprendizagem. *ETD - Educação Temática Digital*, 26(00):e024003.
- Sommerville, I. (2019). *Software Engineering*. Pearson, Boston, 10th edition.
- Suwannik, W. (2025). Minecraft: An engaging platform to learn programming. In *Proceedings of the 2025 IEEE/ACIS 23rd International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 543–546. IEEE.
- United Nations (2015). Transforming our world: the 2030 agenda for sustainable development. <https://sdgs.un.org/2030agenda>.
- Videnovik, M., Vold, T., Kjøning, L., Madevska Bogdanova, A., e Trajkovik, V. (2023). Game-based learning in computer science education: a scoping literature review. *International Journal of STEM Education*, 10(54).
- Werbach, K. e Hunter, D. (2012). *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton Digital Press, Philadelphia.