An Architectural Approach to Support the Development of Digital Educational Games in Computing

Nikolas Oliver Sales Genesio[1], Williamson Silva[2], Alessandreia Marta de Oliveira[1], Pedro Henrique Dias Valle[3]

[1]Federal University of Juiz de Fora (UFJF)
36036-900 – Juiz de Fora – MG – Brazil

[2]Federal University of Pampa (UNIPAMPA)
97546-550 – Alegrete – RS – Brazil

[3]University of São Paulo (USP)
05508-090 – São Paulo – SP – Brazil

nikolas.genesio@estudante.ufjf.br, williamsonsilva@unipampa.edu.br, alessandreia.oliveira@ufjf.br, pedrohenriquevalle@usp.br

Abstract. *The increasing use of Digital Educational Games (DEGs) to support computing education has been observed over the years to improve student engagement and motivation. However, many of these DEGs face challenges in capturing students' interest, resulting in demotivation and disinterest. In this context, this study presents the Reference Architecture (RA), denominated RADEG, to support the development of DEGs aimed at teaching Computing. The evaluation results indicated that the specialists accepted the RA well, considering different requirements, such as learnability, playability, and maintainability. However, security and entertainment requirements need to be addressed in the evolution of RA.*

1. Introduction

The digital games industry has gained significant prominence within the creative and cultural sector, both for its financial performance and for its promising growth prospects in the coming years [Amélio 2018]. In particular, it ranks second in the global entertainment business hierarchy, surpassed only by television and ahead of the film and music industries [Fortim et al. 2022].

From a pedagogical perspective, digital games are effective tools for supporting the teaching-learning process [Lopes et al. 2020], as they are engaging, promote active learning, provide immediate feedback, and allow for the personalization of educational experiences [Souza et al. 2023]. In this context, Digital Educational Games (DEGs) have been employed as didactic resources by Computer Science educators and researchers, standing out as one of the most popular playful strategies across various educational levels. In addition to complementing the learning process, DEGs contribute significantly to increasing students' motivation, engagement, and attention, thereby positively impacting academic performance [Silva and Franco 2022].

Despite the potential of DEGs, the literature has identified important limitations regarding their effectiveness in Computer Science Education [Aires et al. 2020]. Many DEGs are perceived by students as unappealing due to an excessive emphasis on pedagogical content at the expense of playful elements, which can lead to a lack of motivation, low engagement, and, consequently, hinder learning. Moreover, the lack of approaches that offer systematic support for DEG development contributes to their inefficacy [Petri et al. 2019]. Given the absence of studies that offer architectural contributions to DEG construction, this work introduces RADEG — a Reference Architecture (RA) designed to support the development of DEGs for Computer Science education —and presents the results of an expert evaluation conducted to assess its effectiveness and compliance with pedagogical and technical requirements. According to Bass et al. (2012), an RA is a special type of Software Architecture (SA) that captures the essence of the architectures of a collection of systems within a given domain. Through this RA, the aim is to systematize a process that enables structured support for the development of DEGs targeting Computer Science education.

In this context, some studies indicate that DEG architectures often fail to adequately integrate pedagogical and playful elements, resulting in games that are ineffective for both educational and entertainment purposes [Goldin et al. 2022, Genesio et al. 2024]. Therefore, the development of a Reference Architecture (RA) for DEGs designed to support Computer Science education is crucial for standardizing best practices, promoting component reuse, facilitating development, and ensuring both pedagogical and technical quality, ultimately optimizing the teaching-learning process and student engagement [Carvalho 2017]. Furthermore, by applying principles of Software Engineering (SE) to DEG development, both technical and pedagogical quality can be ensured [Rahimi et al. 2021], thereby enhancing the efficiency and effectiveness of DEGs in Computer Science education [Rahimi et al. 2021].

2. Related Work

Several RAs for specific domains can be found in the literature [Goldin et al. 2022]. However, no RAs specifically tailored for DEGs aimed at Computer Science education have been identified.

Fioravanti et al. (2010) introduced the EDUCAR architecture, an RA for educational environments developed based on the ProSA-RA process[Nakagawa et al. 2014]. This three-tier client-server architecture enables the definition of functionalities specific to educational environments while separating them from support and organizational activities. However, this architecture has not yet been evaluated.

Van der Vegt et al. (2016) proposed and evaluated the RAGE architecture, designed to facilitate the reuse of game technology components (objects with specific functionalities) across several programming languages, engines, and platforms. RAGE was successfully validated through the implementation and testing of software assets in four major programming languages, promoting the development of reusable assets across different engines and enhancing the quality and diversity of serious games.

Carvalho (2017) presented two main contributions: the Activity Theory-Based Model for Serious Games (ATMSG) and the Service-Oriented Reference Architecture for Serious Games (SORASG), based on the Service-Oriented Architecture (SOA) standard. The evaluation of this architecture demonstrated that the combination of theory and practice (ATMSG and SORASG) provided an effective solution for reducing the high development costs of Digital Serious Games (DSGs), promoting the creation of high-quality DSGs.

Finally, based on the RAs detailed above, the need for an RA specifically designed for the development of DEGs for Computer Science education becomes evident. While the existing RAs address aspects such as component reuse, modularity, and cost optimization within their respective domains, none of them specifically target the unique challenges of DEGs in the context of Computer Science.

3. Establishment of RADEG

For the development of the reference architecture for DEGs, named RADEG (Reference Architecture for Digital Educational Games)[Genesio et al. 2025], the ProSA-RA approach [Nakagawa et al. 2014] was adopted, which defines a set of four steps that guide the development of reference architectures. Each of the ProSA-RA steps is described below.

3.1. Step RA-1: Investigation of Information Sources

Information sources considered included RA for DEG in several domains, such as the RAGE and SORASG architectures presented in Section 2. In addition, a Systematic Mapping Study (SMS) was conducted, following a protocol based on the guidelines proposed by Kitchenham et al. (2007), to identify the main DEGs, providing a broader understanding of the relationship between non-functional requirements (NFRs) and the quality of these games, both in the development process and in the post-implementation phase.

The SMS analyzed studies published between 1994 and 2024. To increase the quantity and quality of the studies found, the Snowballing technique was also employed in two ways: a search of the references cited in the articles (Backward Snowballing, BS) and a search of articles that cited the selected studies (Forward Snowballing, FS) [Felizardo et al. 2016].

In total, 194 primary studies were analyzed. A more detailed reading of these studies led to the identification of characteristics, functionalities, and requirements that informed the construction of RADEG. The complete results of this stage can be found in Genesio et al. (2024).

Step RA-2: Definition of Architectural Requirements

In the second stage of ProSA-RA, the information and domain concepts obtained from the investigation of sources were used to elicit requirements for the design of RADEG. As a result, 57 System Requirements (SRs) common to DEGs for supporting Computer Science education were identified[1]. It is important to note that some SRs were explicitly extracted from the analyzed studies, while others—common to DEGs—were identified implicitly, requiring direct interaction with the game to recognize them.

Subsequently, a detailed analysis of these requirements was conducted to identify the Architectural Requirements. This process included expert discussions and a qualitative analysis to assess the relevance of each requirement, resulting in the identification of

[1] Available at: `https://bit.ly/4310mxh`

10 key Architectural Requirements that support the development of DEGs for Computer Science education, as presented in Table 1.

Table 1. System Requirements and Architectural Requirements

ID	System Requirement	Architectural Requirement
AR-1	The concepts presented in the game need to be learned	The RA must provide mechanisms to ensure that the concepts presented in the game are learned
AR-2	The game needs to be easy to play	The RA must have tutorials to ensure that the game is easy to play
AR-3	The game elements need to be playful	The RA must allow the use of resources so that the game elements are playful
AR-4	The game must offer players a comfortable and enjoyable experience	The RA must ensure good performance to provide a comfortable and enjoyable experience for players
AR-5	The game must ensure the protection of players' data and provide a secure environment	The RA must implement robust security measures and privacy policies to ensure that player data is protected and that the gaming environment is secure
AR-6	The game needs to be easy to maintain	The RA must be well documented to facilitate maintenance and future updates of the game
AR-7	The game needs to be visually organized to improve the player's experience	The RA must enable clear and effective visual organization, including well-designed screen features
AR-8	The game's concepts, objectives, rules and mechanics need to be easily understood by players	The RA must provide clear and understandable information for presenting the rules and functionalities of the game
AR-9	The game must be fun and engaging for players	The RA must offer interactive elements and stimulating game mechanics
AR-10	The game must allow interactions between the player and the game system, responding to the player's actions quickly and meaningfully	The RA must respond quickly and meaningfully to the player's actions to ensure interactive gameplay

3.2. Step RA-3: Reference Architecture Design

Based on the elicited Architectural Requirements, the design of the RA was carried out in Step RA-3. To properly describe RADEG, the following key objectives were identified: to contribute to the development process, to promote the reuse of software components, and to improve the quality, playability, and maintainability of DEGs. Considering the Architectural Requirements, the main stakeholders were identified: students, teachers, educational institutions, and DEG development professionals. Students use DEGs as learning tools, while teachers apply them in classroom settings. Educational institutions (such as schools and universities) adopt DEGs to enhance their teaching and learning processes, and developers—either companies or individuals—are responsible for creating these DEGs.

Based on the interests of these stakeholders, architectural views were defined following the "4+1" view model, which organizes the description of a SA using five concurrent views, each addressing specific concerns [Kruchten 1995]. Accordingly, the architectural views adopted include: Logical View, Process View, Physical View, Development View, and Use Case View. In addition, the Module View was also considered, as it identifies the main implementation modules of the system and the relationships among them.

Due to space constraints, these views are not detailed here but are available online[2].

After investigating various architectural styles and patterns, RADEG was designed based on a client–server architecture. Figure 1 illustrates the overall structure of RADEG, defined according to the selected architectural style and pattern. On the **Client** side, various platforms on which a DEG may run are represented, such as Desktop, Mobile, or Web. On the Application Server side, domain-specific educational modules are presented, as described below.

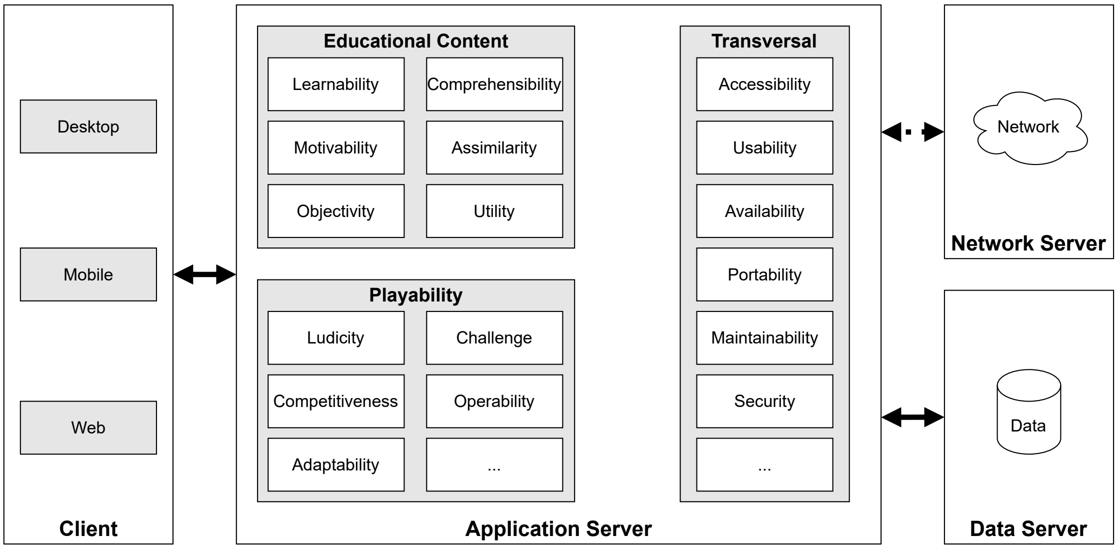

Figure 1. General Structure of RADEG

For the Educational Content, requirements were selected to facilitate the absorption of didactic material integrated within the DEGs, thereby creating an optimal balance between entertainment and learning. By incorporating these requirements into the development of DEGs, it is expected to enhance student learning in the classroom by increasing motivation, engagement, and attention, which positively impacts academic performance.

Regarding Gameplay, the selected requirements encompass the set of characteristics that define the player's experience within the DEG environment, ensuring both enjoyment and effective learning throughout the game session. The Cross-Cutting functionalities include aspects common to all DEGs, such as accessibility and usability, which must be integrated holistically. The architecture was also designed to accommodate the continuous evolution of these functionalities during DEG execution.

Finally, the Data Server manages the storage and retrieval of DEG data, while the Network Server handles network connectivity in cases where the DEG includes online features.

3.3. Step RA-4: Evaluation of the Reference Architecture

The evaluation of RADEG was conducted using 23 questions from the FERA checklist (Framework for Evaluation of Reference Architectures) [Santos et al. 2013]. FERA consists of questions that guide reviewers in identifying issues within documents describing

[2]Available at: `https://bit.ly/3SrKnCl`

a reference architecture. The checklist contains a total of 93 multiple-choice questions, with answers ranging from “fully satisfactory” to “fully unsatisfactory,” as well as fields for general comments. FERA is divided into four stages, each containing questions that reflect the perspectives of different stakeholders, such as software architects, domain experts, managers, and developers.

The RADEG evaluation involved eight experts, all professionals in Software Engineering (SE) with solid knowledge in the domains of DEGs and SAs. Participants were invited by convenience sampling, selecting individuals with experience in SE and DEGs. It is important to note that the invitations were sent via email, and all participants received detailed information about the research, emphasizing that participation was entirely voluntary.

For the evaluation process, participants were initially provided with information regarding the application domain of the reference architecture and the Informed Consent Form for their participation in the study. Next, a profile characterization questionnaire was administered to collect non-identifiable demographic and professional data. Subsequently, a document[3] containing the RADEG views and additional information, such as objectives, involved stakeholders, and risk considerations, was made available. Finally, the FERA questions were provided via Google Forms, with a two-week deadline for responses.

To ensure participants were able to adequately address the questions, a contact channel was made available within the form to clarify any doubts during the evaluation. Table 2 presents the 23 selected questions from the original 93 in FERA, adapted for the educational domain, with response options “yes,” “no,” and “partially”.

4. Results Analysis and Discussion

Following the evaluation conducted using FERA, all evaluator responses were categorized, recorded, and compiled, as illustrated in Table 3. To ensure a qualitative analysis of the responses and adequately address the observations made, Grounded Theory was adopted as the analysis method. This method was chosen for its ability to transform and abstract data into theory by continuously exploring and analyzing the information, ensuring that the developed theories remain grounded in reality [Strauss and Corbin 1994].

The analysis process followed the three main stages described by Strauss and Corbin (1994) . In open coding, the data are segmented and categorized, generating initial concepts and emerging categories. In axial coding, these categories are refined and linked to subcategories, creating a more detailed understanding of the relationships among them. Finally, in selective coding, the core categories are integrated, resulting in a theoretical model that synthesizes the main findings and offers a coherent explanation of the observed patterns [Strauss and Corbin 1994].

In the first phase of the analysis, known as open coding, all expert responses were examined in detail to identify categories and patterns. The responses were organized into two main topics: “Construction and Content” and “Discussion of the Reference Architecture (RA),” facilitating the analysis of similarities and differences among them. Within the

[3] Available at: `https://bit.ly/3FgclxC`

Table 2. Checklist FERA (adapted from [Santos et al. 2013])

ID	Topic	Question
Q1	Construction and Content	Was the Learnability attribute considered?
Q2		Was the Playability attribute considered?
Q3		Was the Ludicity attribute considered?
Q4		Was the Usability attribute considered?
Q5		Was the Security attribute considered?
Q6		Was the Maintainability attribute considered?
Q7		Was the Accessibility attribute considered?
Q8		Was the Comprehensibility attribute considered?
Q9		Was the Fun attribute considered?
Q10		Was the Interactivity attribute considered?
Q11	Discussion	Does the RA clearly explain who the stakeholders are and their interests?
Q12		Does the RA clearly explain the potential stakeholders of an instantiated RA?
Q13		Do the viewpoints generated address the interests of all stakeholders?
Q14		Do the viewpoints generated address the interests of the stakeholders of an instantiated RA?
Q15		Is the RA consistent with domain practices and mandatory standards?
Q16		Do the viewpoints include interests that are not domain-specific, stakeholder interests?
Q17		For each viewpoint, are the models clearly defined? Do the models provide information to determine whether the interests have been achieved?
Q18		Are the domain's objectives clearly articulated and prioritized?
Q19		Is there any traceability between domain objectives and requirements raised?
Q20		Is there any traceability between domain objectives and technical solutions?
Q21		What criteria are used to determine whether the RA supports the domain's objectives?
Q22		Is it possible to view the RA instances?
Q23		Are the risks of introducing this RA clearly documented?

Table 3. Evaluators' answers

Results	Construction and Content	RA Discussion		Total Items
		Stakeholders	Experts	
Both Yes	Q1, Q2, Q3, Q4, Q6, Q7, Q8 and Q10	Q11 to Q17	Q18, Q21 and Q22	18
Both Partial				
Yes and Partial	Q9		Q19 and Q23	3
Both No				
No and other answer	Q5 (No, Partial)		Q20 (Yes, No)	2

Construction and Content topic, categories such as documentation clarity, consideration of fun and security requirements, and completeness of the presented views emerged.

The analysis revealed that most experts considered the RADEG documentation clear and detailed. However, some gaps were identified, especially regarding the explicitness of security and fun requirements, which were addressed implicitly and without the necessary detail, generating questions among the experts. It was observed that although RADEG addressed playful elements, the fun requirement was not represented in the RA views. Likewise, data security was mentioned, but precise information about measures to ensure data protection and prevent unauthorized access was lacking. Despite this, the other requirements were fully understood and accepted by the evaluators.

In the second phase, axial coding, the categories emerging from the previous phase were related to one another, providing a deeper understanding of the data. This analysis revealed that for RADEG to be considered complete, it is essential that playful aspects, such as fun, be treated with greater clarity and prominence. Moreover, although data security is recognized as important, its absence as an explicit requirement in RADEG may raise uncertainties regarding user data protection. This gap indicates that security needs to be addressed more visibly and explicitly, both in the documentation and in the architectural views. The difficulty in visualizing traceability between stakeholders' objectives and technical solutions also raised questions about the adequacy of the RA to meet the domain's specific objectives.

In the final phase, selective coding, the categories were refined around two main theories. The first emerging theory is that, for RADEG to effectively meet the expectations of stakeholders and the demands of the JED domain, fun and security requirements must be explicitly addressed in the documentation. The absence of a clear approach to these requirements affects both the acceptance of the RA by experts and its capacity to be effectively instantiated. The second theory suggests that including a view that directly maps domain objectives to technical solutions could improve traceability and understanding of RADEG. This would help ensure that the proposed solutions are clearly aligned with the educational and playful goals of the domain.

Although RADEG was widely accepted in terms of clarity and organization, deeper analysis of the responses revealed that crucial issues, such as the explicitness of security and fun requirements, as well as objective traceability, need to be addressed more clearly. The use of the Grounded Theory method allowed for a detailed evaluation, highlighting the importance of explicitly stating essential requirements in the documentation.

Furthermore, RADEG provided a solid foundation for the development of JEDs and was broadly accepted by the experts. However, the evaluation suggests necessary adjustments to improve the documentation. These adjustments are fundamental to ensure that RADEG can be effectively instantiated and sustained over time, promoting the development of high-quality JEDs. Quantitative analysis revealed that in 18 out of 23 questions, all eight experts answered "Yes," indicating broad acceptance of the RA concerning the evaluated attributes. These questions include Q1, Q2, Q3, Q4, Q6, Q7, Q8, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q18, Q19, and Q22, suggesting that the RA successfully met 78% of the analyzed criteria.

Some questions, however, presented more diverse answers, revealing areas where the RA may require adjustments. Question Q5 stood out as the most critical, with six experts responding "No" and only two responding "Partially". This points to a widespread concern regarding the RA's ability to adequately address security aspects, which should be considered a priority area for improvement. Another area of disagreement was the fun requirement (Q9), in which four experts responded "Yes" and four responded "Partially", suggesting that the playful element of the RA is not fully perceived as an essential component. This divergence indicates that, although half of the experts recognized fun as a positive and well-implemented attribute, the other half did not perceive this characteristic as clearly, pointing to the need for the RA to better reinforce the fun requirement.

Additionally, Q19 received seven "Yes" responses and one "Partially", suggesting that despite the high level of acceptance, one expert identified the need for improvements in traceability between domain objectives and elicited requirements. While the majority considered the RA to meet this criterion, adjustments are necessary to ensure that traceability is fully clear and well documented. Q23, with six “Yes” responses and two “Partially”, indicates that although the RA documentation was well rated by most, there is still room for improvement. The fact that two experts considered the documentation only partially satisfactory highlights the importance of reviewing and improving the clarity of risks in the documentation.

Finally, Q20 received seven “Yes” responses and one “No”, indicating that although the vast majority of experts agreed that the RA presents adequate traceability between domain objectives and technical solutions, there is a need for review and potential improvement in how this traceability is documented and demonstrated. Thus, a more transparent view is necessary regarding how each technical solution directly maps to the strategic domain objectives.

5. Threats to Validity

This study collected empirically grounded evidence regarding RADEG and utilized expert feedback for its evaluation, thereby incorporating certain threats to validity. These threats were identified and discussed based on the guidelines proposed by Wohlin *et al.* (2012).

Regarding **internal validity**, the human factor may have influenced the construction of the RA, impacting its overall structure. To mitigate this threat, ProSA-RA was employed, a process that systematizes the development, design, and evaluation of reference architectures. Furthermore, the human factor may also have affected the prioritization and transformation of system requirements (SRs) into architectural requirements. However, this process was conducted by different experts to eliminate biases during the prioritization and transformation of requirements.

Concerning **external validity**, the number of participants in the evaluation of RADEG through FERA may limit the validity of the results. Nevertheless, to minimize this threat, RADEG was evaluated by eight specialists with knowledge in SE, RA, and DEGs. Their expertise and experience provided reliable assessments, as well as substantial feedback on RADEG.

Finally, to reduce the threat to **construct validity** in the evaluation of RADEG, the FERA checklist was used. This checklist allows for assessing the construction and descriptions of reference architectures, including questions that reflect the perspectives of different stakeholders.

6. Conclusion and Future Work

This work presented the development and evaluation of RADEG, a RA designed to support the development of DEGs for Computer Science education. Based on the conducted evaluations, it was concluded that the RA addresses several important quality attributes, such as learnability, playability, usability, and maintainability. However, areas requiring refinement were identified, particularly regarding the requirements for fun and security. The lack of a clear approach to these requirements affects both the acceptance of the RA by experts and its capacity to be effectively instantiated.

In short, the RA provides a solid foundation for the development of DEGs, but adjustments in documentation and a more detailed specification of certain quality attributes are necessary to ensure an effective and sustainable instantiation over time. Completing these refinements will contribute to increasing experts' confidence and the RA's efficiency in developing high-quality DEGs.

As future work, it is proposed to update the documentation to specify the fun and security requirements, as well as the risks associated with the introduction of RADEG. Furthermore, traceability should be enhanced to guarantee that all essential aspects are adequately communicated and understood by the involved experts. Through this RA, it is also expected to promote increased reuse of software components in DEGs, facilitating development while encompassing all essential aspects for the success of these games.

References

Aires, S. F., Barbosa, J. F., and Madeira, C. A. (2020). Desenvolvendo jogos educacionais digitais inovadores e instigantes com o framework playeduc. *Sociedade Brasileira de Computação*.

Amélio, C. O. (2018). A indústria e o mercado de jogos digitais no brasil. *XVII SBGames, Foz do Iguaçu, Paraná, Brasil*, pages 1497–1506.

Bass, L., Clements, P., and Kazman, R. (2012). *Software Architecture in Practice*. Addison-Wesley Professional, 3rd edition.

Carvalho, M. B. (2017). Serious games for learning: a model and a reference architecture for efficient game development.

Felizardo, K. R., Mendes, E., Kalinowski, M., Souza, É. F., and Vijaykumar, N. L. (2016). Using forward snowballing to update systematic reviews in software engineering. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 1–6.

Fioravanti, M. L., Nakagawa, E. Y., and Barbosa, E. F. (2010). Educar: uma arquitetura de referência para ambientes educacionais. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 1.

Fortim, I., Nakano, D. N., Sakuda, L. O., and dos Santos, E. F. (2022). Pesquisa indústria brasileira de games 2022. *TIC CULTURA*, page 113.

Genesio, N., Oliveira, A., and Valle, P. (2025). Uma abordagem arquitetural para o desenvolvimento de jogos educacionais digitais na educação em computação. In *Anais Estendidos do V Simpósio Brasileiro de Educação em Computação*, pages 30–32, Porto Alegre, RS, Brasil. SBC.

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.

Goldin, T., Rauch, E., Pacher, C., and Woschank, M. (2022). Reference architecture for an integrated and synergetic use of digital tools in education 4.0. *Procedia Computer Science*, 200:407–417.

Kitchenham, B., Charters, S., et al. (2007). Guidelines for performing systematic literature reviews in software engineering.

Kruchten, P. B. (1995). The 4+ 1 view model of architecture. *IEEE software*, 12(6):42–50.

Lopes, A. F., Santana, T. S., and Braga, A. H. (2020). O ensino de pensamento computacional por meio de jogos desplugados e olimpíadas científicas: um relato de experiência nos anos finais do ensino fundamental. In *Anais do XXVIII Workshop sobre Educação em Computação*, pages 96–100. SBC.

Nakagawa, E. Y., Guessi, M., Maldonado, J. C., Feitosa, D., and Oquendo, F. (2014). Consolidating a process for the design, representation, and evaluation of reference architectures. In *2014 IEEE/IFIP Conference on Software Architecture*, pages 143–152. IEEE.

Petri, G., Gresse von Wangenheim, C., 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(3).

Rahimi, S., Shute, V., Kuba, R., Dai, C.-P., Yang, X., Smith, G., and Fernández, C. A. (2021). The use and effects of incentive systems on learning and performance in educational games. *Computers & Education*, 165:104135.

Santos, J. F. M., Guessi, M., Galster, M., Feitosa, D., and Nakagawa, E. Y. (2013). A checklist for evaluation of reference architectures of embedded systems (s). In *SEKE*, volume 13, pages 1–4.

Silva, L. F. and Franco, M. H. I. (2022). Jogos educacionais digitais no apoio ao processo de alfabetizaçao e letramento: Revisao sistemática da literatura. *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*, pages 453–462.

Souza, G. G., Andrade, V. B., Xexéo, G., Silva, F., and de Oliveira, S. (2023). Superbactérias: um jogo educativo sobre o risco do uso inadequado de antibióticos. In *Anais Estendidos do XXII Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 1317–1322. SBC.

Strauss, A. and Corbin, J. (1994). Grounded theory methodology: An overview.

Van der Vegt, W., Westera, W., Nyamsuren, E., Georgiev, A., and Ortiz, I. M. (2016). Rage architecture for reusable serious gaming technology components. *International Journal of Computer Games Technology*, 2016(1):5680526.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.