

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Uma Arquitetura de Referência para Jogos Educativos Digitais de Apoio ao Ensino de Computação

Nikolas Oliver Sales Genesio

JUIZ DE FORA
MARÇO, 2025

Uma Arquitetura de Referência para Jogos Educacionais Digitais de Apoio ao Ensino de Computação

NIKOLAS OLIVER SALES GENESIO

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Alessandra Marta de Oliveira

Coorientador: Pedro Henrique Dias Valle

JUIZ DE FORA

MARÇO, 2025

UMA ARQUITETURA DE REFERÊNCIA PARA JOGOS EDUCACIONAIS DIGITAIS DE APOIO AO ENSINO DE COMPUTAÇÃO

Nikolas Oliver Sales Genesio

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS
EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTE-
GRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Alessandreia Marta de Oliveira
Doutora em Computação - IC/UFF

Pedro Henrique Dias Valle
Doutor em Ciências da Computação e Matemática Computacional - ICMC/USP

André Luiz de Oliveira
Doutor em Ciências da Computação e Matemática Computacional - ICMC/USP

Gleiph Ghiotto Lima de Menezes
Doutor em Computação - IC/UFF

JUIZ DE FORA
10 DE MARÇO, 2025

A DEUS.

*Aos meus pais, que, sob muito sol, fizeram-me
chegar até aqui, na sombra.*

À minha família, pelo apoio e sustento.

Aos meus amigos e professores.

Resumo

A utilização crescente de Jogos Educacionais Digitais (JEDs) para apoiar o ensino de Computação tem sido observada ao longo dos anos, visando melhorar o engajamento e a motivação dos alunos. No entanto, muitos desses JEDs enfrentam desafios em captar o interesse dos alunos, resultando em desmotivação e desinteresse. Nesse contexto, este trabalho propõe uma Arquitetura de Referência (AR) para apoiar o desenvolvimento de JEDs voltados à Educação em Computação. A AR busca promover a reutilização de componentes e facilitar a evolução e a manutenção dos JEDs, além de equilibrar os aspectos pedagógicos e de entretenimento, proporcionando uma experiência educacional mais envolvente e eficaz para os estudantes. Para o estabelecimento dessa AR, foi utilizado o ProSA-RA (*Process based on Software Architecture – Reference Architecture*), um processo que sistematiza o projeto, representação e avaliação de ARs. Os resultados alcançados por meio da avaliação utilizando o *checklist* FERA (*Framework for Evaluation of Reference Architectures*) e o método ATAM (*Architecture Tradeoff Analysis Method*) indicaram que a AR está bem alinhada com os interesses dos *stakeholders* e considera diversos atributos de qualidade essenciais para o desenvolvimento de JEDs. No entanto, foram constatadas necessidades de refinamento devido aos riscos não identificados previamente, de modo que ajustes sejam realizados para aumentar a eficiência da AR no desenvolvimento dos JEDs.

Palavras-chave: Arquitetura de Referência, Jogos Educacionais Digitais, Educação em Computação.

Abstract

The increasing use of Digital Educational Games (DEGs) to support Computing teaching has been observed over the years, aiming 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 proposes a Reference Architecture (RA) to support the development of DEGs aimed at Computing Education. The RA seeks to promote the reuse of components and facilitate the evolution and maintenance of DEGs, as well as balancing the pedagogical and entertainment aspects, providing a more engaging and effective educational experience for students. To establishing this RA, was used the ProSA-RA (Process based on Software Architecture - Reference Architecture), a process that systematizes the design, representation and evaluation of RAs. The results of the evaluation using the FERA (Framework for Evaluation of Reference Architectures) checklist and the ATAM (Architecture Tradeoff Analysis Method) indicated that the RA is well aligned with the interests of the stakeholders and considers various quality attributes that are essential for the development of DEGs. However, refinement needs were identified due to previously unidentified risks, so that adjustments can be made to increase the RA's efficiency in the development of DEGs.

Keywords: Reference Architecture, Digital Educational Games, Computing Education.

Agradecimentos

Primeiramente, agradeço a DEUS, que permitiu a realização deste trabalho, sustentando-me a cada passo até a presente etapa. A Ele, que é antes de todas as coisas e em quem todas as coisas subsistem, dedico toda honra. Tudo que foi feito aqui é por Ele, através Dele e para Ele, e somente Ele é digno de ser exaltado e glorificado. Sem Sua graça, nada disso teria sido alcançado, e por isso, minha gratidão é eterna.

Em especial, gostaria de agradecer meus pais, Rogério e Eliana, por todo o apoio, carinho e orações que me ofereceram ao longo desta trajetória. Sou imensamente grato também pelo trabalho no Frango Assado do Rogerão, que foi essencial para que eu pudesse seguir estudando. Não há palavras que possam descrever a imensa gratidão que sinto por ser abençoado com pais tão maravilhosos.

Agradeço também à minha avó Edmea, por suas orações e apoio contínuos durante minha trajetória acadêmica. Este trabalho é dedicado, com muito carinho, à memória do meu avô Armando, que sempre acreditou em mim e em meu sonho de ingressar na universidade. Estendo meus agradecimentos a todos os meus familiares, que sempre torceram por mim e celebraram cada conquista.

Em especial, aos meus melhores amigos, Júlio César e Yasmin, minha eterna gratidão. Vocês tornaram essa caminhada mais leve e foram meus companheiros, presentes nos momentos bons e nos desafios, sempre prontos a oferecer palavras de conforto e motivação. Agradeço também aos amigos que fiz na universidade, em especial Ágata e Kleiton, que sempre estiveram ao meu lado em cada etapa do curso.

Aos meus professores, sou grato por todo conhecimento transmitido e pelos conselhos que contribuíram para minha formação. Em especial, à minha orientadora, professora Alessandra Marta de Oliveira, obrigado pelo apoio, ensinamentos e carinho ao longo deste trabalho. Que Deus a recompense por toda sua ajuda. Ao meu coorientador, professor Pedro Henrique Dias Valle, expresse minha gratidão por ser um exemplo de dedicação e amizade. Que Deus lhe recompense abundantemente por tudo.

*“Consagre ao SENHOR tudo o que você
faz, e os seus planos serão bem-sucedidos”.*

Provérbios 16:3

Conteúdo

Lista de Figuras	8
Lista de Tabelas	9
Lista de Abreviações	10
1 Introdução	11
1.1 Apresentação do Tema	11
1.2 Contextualização	12
1.3 Descrição do Problema	14
1.4 Motivação	15
1.5 Questão de Pesquisa	16
1.6 Objetivos	16
1.7 Metodologia	17
1.8 Organização	17
2 Fundamentação Teórica e Trabalhos Relacionados	18
2.1 Jogos Educacionais Digitais	19
2.1.1 Desenvolvimento de Jogos Educacionais Digitais	20
2.1.2 Ensino de Computação	21
2.1.3 Desafios e Perspectivas	22
2.2 Arquitetura de Software	23
2.2.1 Terminologia e Conceitos Básicos	24
2.2.2 Representações de Arquiteturas de Software	27
2.3 Arquitetura de Referência	30
2.3.1 Classificações de Arquiteturas de Referência	32
2.3.2 Processo para Estabelecer uma Arquitetura de Referência	34
2.4 Trabalhos Relacionados	37
2.4.1 Análise Comparativa	40
2.5 Considerações Finais	42
3 Estabelecimento da RADEG	43
3.1 RA-1: Investigação das Fontes de Informação	43
3.1.1 Grupo 1: JEDs para a Educação em Computação	44
3.1.2 Grupo 2: ARs para Jogos e Sistemas Educacionais	45
3.2 RA-2: Estabelecimento dos Requisitos Arquiteturais	46
3.2.1 Identificação dos Requisitos do Sistema	46
3.2.2 Identificação dos Requisitos Arquiteturais	46
3.3 RA-3: Projeto da Arquitetura de Referência	47
3.3.1 Objetivos, <i>Stakeholders</i> e Interesses	48
3.3.2 Visão Geral (<i>Overview</i>)	49
3.3.3 Visão Lógica (<i>Logical View</i>)	50
3.3.4 Visão de Processo (<i>Process View</i>)	52
3.3.5 Visão Física (<i>Physical View</i>)	54
3.3.6 Visão de Desenvolvimento (<i>Development View</i>)	55

3.3.7	Visão de Caso de Uso (<i>Use Case View</i>)	57
3.3.8	Visão de Módulo (<i>Module View</i>)	58
3.4	RA-4: Avaliação da Arquitetura de Referência	59
3.5	Considerações Finais	60
4	Avaliação da RADEG	62
4.1	Avaliação Preliminar	62
4.2	<i>Architecture Tradeoff Analysis Method</i> (ATAM)	67
4.3	Fase 0: Parceria e preparação	68
4.4	Fase 1: Avaliação (inicial)	69
4.4.1	Etapa 1: Preparação do ATAM	69
4.4.2	Etapa 2: Apresentação do <i>Business Driver</i>	69
4.4.3	Etapa 3: Apresentação da Arquitetura	69
4.4.4	Etapa 4: Identificação das Abordagens Arquitetônicas	70
4.4.5	Etapa 5: Geração da Árvore de Atributos de Qualidade	70
4.4.6	Etapa 6: Análise das Abordagens Arquiteturais	76
4.5	Fase 2: Avaliação (continuação)	79
4.5.1	Etapa 7: <i>Brainstorming</i> e Priorização de Cenários	80
4.5.2	Etapa 8: Análise das Abordagens Arquiteturais com os Cenários . .	81
4.5.3	Etapa 9: Apresentação dos resultados	85
4.6	Fase 3: Acompanhamento	86
4.7	Ameaças à Validade	86
4.8	Considerações Finais	87
5	Conclusão	88
5.1	Contribuição Científica	88
5.2	Publicações	89
5.3	Trabalhos Futuros	89
	Bibliografia	91
6	Apêndices	101
6.1	Apêndice A Requisitos do Sistema	101

Lista de Figuras

2.1	Relacionamento entre Modelo de Referência, Padrão Arquitetural, Arquitetura de Referência e Concreta (Adaptado de Bass, Clements e Kazman, 2003).	27
2.2	Modelo de Visualização 4+1 (Adaptado de Kruchten, 1995).	29
2.3	Interação de <i>Stakeholders</i> e Contextos entre Arquiteturas Concretas e de Referência (Adaptado de Angelov, Trienekens e Grefen, 2008).	31
2.4	Estrutura do ProSA-RA (Adaptado de Nakagawa et al., 2014).	36
3.1	Processo de busca em quatro etapas para o MSL.	44
3.2	Visão Geral da RADEG.	50
3.3	Diagrama de Classes da RADEG.	51
3.4	Diagrama de Atividades da RADEG.	53
3.5	Diagrama de Implantação da RADEG.	55
3.6	Diagrama de Pacotes da RADEG.	56
3.7	Diagrama de Caso de Uso da RADEG.	57
3.8	Diagrama de Componentes da RADEG.	60
4.1	Etapas do ATAM (Adaptado de Kazman et al., 1998).	68
4.2	Árvore de Atributos de Qualidade.	71
4.3	Partes de um Cenário de Atributo de Qualidade (Adaptado de Bass, Clements e Kazman, 2012).	72
4.4	Um cenário geral para Aprendizizibilidade.	73
4.5	Um cenário geral para Jogabilidade.	73
4.6	Um cenário geral para Ludicidade.	74
4.7	Um cenário geral para Usabilidade.	74
4.8	Um cenário geral para Segurança.	74
4.9	Um cenário geral para Manutenibilidade.	75
4.10	Um cenário geral para Acessibilidade.	75
4.11	Um cenário geral para Compreensibilidade.	75
4.12	Um cenário geral para Diversão.	76
4.13	Um cenário geral para Interatividade.	76
4.14	Avaliação dos cenários.	80

Lista de Tabelas

2.1	Comparação dos trabalhos relacionados.	41
4.1	<i>Checklist</i> FERA para avaliação de AR (adaptado de Santos et al. 2013) . .	63
4.2	Respostas das Perguntas da <i>checklist</i> FERA	64
4.3	Observações dos Atributos na AR	77
4.4	Análise do Cenário de Aprendizibilidade	82
4.5	Análise do Cenário de Manutenibilidade	82
4.6	Análise do Cenário de Ludicidade	83
4.7	Análise do Cenário de Usabilidade	83
4.8	Análise do Cenário de Acessibilidade	83
4.9	Análise do Cenário de Jogabilidade	84
4.10	Análise do Cenário de Segurança	84
4.11	Análise do Cenário de Compreensibilidade	84
4.12	Análise do Cenário de Diversão	85
4.13	Análise do Cenário de Interatividade	85

Lista de Abreviações

AR	Arquitetura de Referência
AS	Arquitetura de Software
ATAM	<i>Architecture Tradeoff Analysis Method</i>
DCC	Departamento de Ciência da Computação
FERA	<i>Framework for Evaluation of Reference Architectures</i>
JEDs	Jogos Educacionais Digitais
MSL	Mapeamento Sistemático da Literatura
MVC	<i>Model View Controller</i>
ProSA-RA	<i>Process based on Software Architecture – Reference Architecture</i>
QP	Questão de Pesquisa
RAModel	<i>Reference Architecture Model</i>
SBC	Sociedade Brasileira de Computação
SysML	<i>Systems Modeling Language</i>
UFJF	Universidade Federal de Juiz de Fora
UML	<i>Unified Modeling Language</i>

1 Introdução

Este capítulo apresenta, de forma detalhada, uma introdução ao tema desse trabalho, fornecendo uma contextualização sobre a relevância dos Jogos Educacionais Digitais (JEDs) no cenário atual e a necessidade de uma Arquitetura de Referência (AR) para auxiliar no desenvolvimento e evolução desses JEDs. Além disso, também é apresentado o problema central da pesquisa, a motivação para o desenvolvimento de uma AR, a questão de pesquisa, os objetivos delineados e a metodologia adotada para atingir tais objetivos.

Nesse cenário, este Capítulo está organizado da seguinte forma: A Seção 1.1 introduz o tema dessa pesquisa. A Seção 1.2 apresenta os aspectos contextuais. Na Seção 1.3 são detalhados os problemas que sustentam o desenvolvimento dessa pesquisa. Na Seção 1.4 é apresentada a motivação para a realização dessa pesquisa. A Seção 1.5 apresenta a Questão de Pesquisa. Na Seção 1.6 são detalhados os objetivos principais e específicos da pesquisa. A Seção 1.7 detalha a metodologia para alcançar os objetivos propostos. Por fim, a Seção 1.8 apresenta a organização geral de todo o trabalho.

1.1 Apresentação do Tema

Recentemente, a Sociedade Brasileira de Computação (SBC) tem se dedicado ativamente à elaboração de referenciais curriculares (SBC, 2017; ZORZO et al., 2017) para a integração da Computação na Educação básica e superior (BITTENCOURT; SANTANA; ARAUJO, 2021). Nesse sentido, os pesquisadores brasileiros da área de Computação têm produzido diversas iniciativas para a inserção bem-sucedida da Computação nos diferentes níveis da Educação (SANTANA; ARAUJO; BITTENCOURT, 2020). Dentre essas iniciativas, destaca-se o uso de JEDs como ferramentas de apoio ao ensino (CLEMENTINO et al., 2022). No entanto, existem diversos desafios a serem superados para a efetiva implementação da Computação na Educação básica e superior (BRUM; CRUZ, 2017).

No contexto da Engenharia de Software (ES), a subárea de Arquitetura de Software (AS), tem ganhado destaque nos últimos anos (WOODS; ERDER; PUREUR, 2021).

Seu foco é compreender os principais elementos de software, a relação entre esses elementos e suas propriedades (OLIVEIRA et al., 2022). Além disso, uma AS bem projetada pode permitir que um sistema atenda aos principais requisitos de um projeto (VENTERS et al., 2018), incluindo desempenho, confiabilidade, portabilidade, manutenibilidade e interoperabilidade (VENTERS et al., 2018).

Nesse cenário, a AS desempenha um papel crucial nas propriedades do sistema (SHAW; GARLAN, 2005), e as decisões tomadas em nível arquitetural podem impactar diretamente na realização dos objetivos de negócio (ANGELOV; TRIENEKENS; GREFFEN, 2008), assim como no cumprimento dos requisitos funcionais e de qualidade (ANGELOV; TRIENEKENS; GREFFEN, 2008). Desenvolver um sistema de forma planejada e testá-lo de forma abrangente são ações importantes, mas nenhuma dessas atividades pode compensar uma arquitetura inadequada (GARCIA et al., 2021).

Dentre os tipos de AS existentes, um tipo especial de arquitetura, a AR, tem emergido como uma estrutura que oferece diretrizes essenciais para a especificação de arquiteturas concretas pertencentes a um determinado domínio de aplicação (BASS; CLEMENTS; KAZMAN, 2012). Além de estabelecer uma compreensão bem fundamentada dos domínios específicos, as ARs também promovem a reutilização de conhecimentos especializados em *design*, simplificando o desenvolvimento, a padronização e a evolução de sistemas de software (NAKAGAWA; OQUENDO; BECKER, 2012).

Considerando sua relevância, ARs podem ser encontradas em diversos domínios (ARSANJANI et al., 2007; DUARTE, 2012; RODRIGUEZ, 2018) e têm-se mostrado bastante relevantes para o desenvolvimento de sistemas nesses contextos (NAKAGAWA; OQUENDO; BECKER, 2012). Para facilitar e sistematizar a criação de ARs, processos, modelos e diretrizes também têm sido propostos (BAYER, 2004; MULLER, 2008a; NAKAGAWA et al., 2009).

1.2 Contextualização

Nos últimos anos, observou-se um avanço na criação de ferramentas tecnológicas para apoiar o processo de ensino-aprendizado em Computação (SILVA et al., 2018). Contudo, apesar da Computação e do perfil dos alunos terem-se modificado significativamente (LO-

PES et al., 2021), grande parte dos cursos de graduação nessa área ainda adota métodos de ensino tradicionais, como aulas expositivas, que frequentemente não acompanham essas transformações (FROSI; JAQUES, 2020).

Para muitos estudantes, o método tradicional de ensino é considerado ineficaz e desmotivador (FEICHAS; SEABRA; SOUZA, 2021), comprometendo a aprendizagem dos conceitos de programação e contribuindo para a evasão em disciplinas iniciais de cursos de Computação (ALVIM; BITTENCOURT; DURAN, 2024). Em particular, aprender a programar é amplamente reconhecido como a tarefa mais desafiadora e, ao mesmo tempo, a mais crucial a ser adquirida pelos alunos nos cursos de Computação (OLIVEIRA; PEREIRA, 2019).

Devido a esse fato e embora as disciplinas introdutórias sejam fundamentais, observa-se um alto índice de reprovação (MACHADO et al., 2018), o que, consequentemente, pode levar à desistência das disciplinas e/ou até mesmo a inúmeras evasões desses cursos (CLASSE; CASTRO, 2020b), gerando preocupação entre os docentes que as ministram (TENÓRIO; RODRIGUES, 2018).

Apesar dos professores procurarem novas abordagens instrucionais de apoio ao ensino de Computação, é difícil manter os alunos engajados e evitar a evasão, tornando-se um dos principais desafios a serem superados (CARVALHO et al., 2019). Nesse contexto, um número crescente de estratégias tem se destacado por sua capacidade de captar a atenção dos alunos, transformando o processo de ensino-aprendizagem (CASTRO; SIQUEIRA, 2019).

Nessa perspectiva, os Jogos Educacionais Digitais (JEDs) surgem como uma alternativa (GAROZI; JUNIOR; COSTA, 2021), pois proporcionam um processo de aprendizagem interativo e comunicativo, permitindo ao aluno resolver problemas de forma lúdica (SILVA; FERNANDES; SANTOS, 2018). Os JEDs, além de incluir elementos de entretenimento (PAIVA; TORI, 2017), são projetados especificamente para ensinar sobre um tema específico, ampliar conceitos, reforçar o desenvolvimento, auxiliar no aprendizado de uma habilidade ou mudança de atitude enquanto os jogadores se divertem (PETRI; WANGENHEIM; BORGATTO, 2019).

1.3 Descrição do Problema

Considerando esses objetivos, os JEDs têm desempenhado um papel significativo no aprimoramento do processo de ensino-aprendizagem em diversas disciplinas na área da Computação (PETRI; WANGENHEIM; BORGATTO, 2019), incluindo Engenharia de Software, Programação, Redes de Computadores, Algoritmos e Segurança (ASSUMPÇÃO et al., 2022). Além de contornar a complexidade dos conteúdos de Computação (HONDA et al., 2022), os JEDs também promovem engajamento e motivação (MACENA et al., 2022), tornando a aprendizagem mais significativa (SILVA; FERNANDES; FERNANDES, 2022).

No entanto, muitos JEDs não têm se mostrado eficazes (AIRES; BARBOSA; MADEIRA, 2020). Isso ocorre porque, além de serem considerados chatos pelos alunos devido ao foco excessivo no conteúdo pedagógico e ao tratamento do entretenimento como algo secundário (FALCÃO et al., 2017), também falham em seu objetivo principal de ensinar (ARAÚJO; ARANHA; MADEIRA, 2018). Com isso, observa-se que os alunos não conseguem se motivar e se engajar com esses jogos da mesma forma que fazem com jogos voltados ao entretenimento (NASCIMENTO; ISHITANI, 2021).

Além disso, observa-se que muitos JEDs não são utilizados de forma sustentável, em decorrência de fatores como o desalinhamento com os objetivos pedagógicos, as dificuldades na avaliação de sua eficácia e os desafios relacionados ao seu desenvolvimento (PETRI; WANGENHEIM; BORGATTO, 2019). Ademais, verifica-se que muitos professores desconhecem a existência de diversos JEDs e, mesmo os que os conhecem, enfrentam dificuldades ao integrá-los em suas práticas pedagógicas. A falta de formação específica e de recursos adequados contribui para essa lacuna, limitando o potencial desses JEDs como ferramentas de ensino (SILVA; FRANCO, 2022).

Um dos fatores que contribuem para essa problemática é que os JEDs são frequentemente desenvolvidos de forma *ad-hoc* (BATTISTELLA; WANGENHEIM, 2016), sem seguir uma metodologia de desenvolvimento do jogo (PETRI; WANGENHEIM; BORGATTO, 2019). Além disso, eles são comumente avaliados de forma superficial quanto ao seu impacto no processo de ensino-aprendizagem (BATTISTELLA; WANGENHEIM, 2016). Por fim, percebe-se que a maioria dos estudos é conduzida com amostras pequenas, o que compromete a confiabilidade e a validade dos resultados obtidos. Consequentemente,

a qualidade e a eficácia desses jogos tornam-se questionáveis (PETRI; WANGENHEIM; BORGATTO, 2019).

Diante disso, é imprescindível utilizar metodologias e/ou ferramentas que possam formalizar e apoiar o desenvolvimento de JEDs (AIRES; BARBOSA; MADEIRA, 2020), para torná-los mais envolventes para os estudantes (SILVA et al., 2021). Isso envolve o equilíbrio dos aspectos pedagógicos necessários para o processo de ensino-aprendizagem com os elementos de entretenimento que atraem jogadores de todas as idades (AIRES; BARBOSA; MADEIRA, 2020). Considerando a importância de desenvolver JEDs de alta qualidade (PETRI; WANGENHEIM; BORGATTO, 2019), é importante concentrar esforços no estabelecimento das arquiteturas de software desses JEDs (MIZUTANI; DAROS; KON, 2021).

1.4 Motivação

Apesar de já existirem algumas iniciativas que utilizam metodologias para a reutilização de diversos elementos característicos dos jogos (AIRES; BARBOSA; MADEIRA, 2020), como os *frameworks*, que são artefatos que promovem a reutilização de códigos ou conceitos de forma simplificada (VICTAL; MENEZES, 2015), ainda não foram identificados trabalhos que estabeleçam a criação de uma AR para o domínio dos JEDs, especialmente voltados para a Educação em Computação. O surgimento cada vez mais frequente de JEDs voltados para a Educação em Computação destaca a necessidade de ARs para esse domínio.

Além disso, observa-se que, em geral, as equipes de desenvolvimento de JEDs frequentemente não se preocupam com os processos, métodos e ferramentas empregados para auxiliar o desenvolvimento desses jogos (MADEIRA; SERRANO, 2020). Isso é potencializado pelo fato de que a maioria das pesquisas é realizada com amostras pequenas, o que gera dúvidas sobre a confiabilidade e validade dos resultados, assim como a qualidade e eficácia dos JEDs (PETRI; WANGENHEIM; BORGATTO, 2019).

Em resumo, a falta de sistematização no desenvolvimento de JEDs e também a qualidade desses produtos de software podem ser identificadas como um problema de pesquisa a ser abordado neste trabalho. Portanto, uma AR para apoiar o desenvolvimento

de JEDs pode auxiliar na melhor organização, modularização e, consequentemente, reutilização de módulos, aumentando a produtividade e a capacidade de evolução dos sistemas desse domínio, além de garantir que características de qualidade sejam satisfeitas.

1.5 Questão de Pesquisa

Diante dos desafios mencionados anteriormente, elaborou-se a seguinte questão de pesquisa (QP):

- Como apoiar o desenvolvimento de Jogos Educacionais Digitais, em nível arquitetural de software, para apoiar a Educação em Computação?

1.6 Objetivos

Motivado pela dificuldade na sistematização do desenvolvimento dos JEDs de apoio à Educação em Computação e pela ausência de qualidade dos jogos existentes, o objetivo principal deste trabalho é estabelecer uma AR para apoiar e sistematizar o desenvolvimento de JEDs para apoio à Educação em Computação. A AR proposta abrange o conhecimento do domínio e os conceitos dos JEDs, com o intuito de contribuir para a área dos JEDs por meio da disponibilização de uma arquitetura que facilite o desenvolvimento, reúso, manutenibilidade, produtividade, modularidade e evolução de JEDs voltados para a Educação em Computação.

Para alcançar o objetivo principal da pesquisa, foram estabelecidos os seguintes objetivos específicos:

- Identificar Fontes de Informação: Explorar as fontes de informação que descrevem o conhecimento necessário para compreender o domínio dos JEDs. Adicionalmente, identificar os requisitos, tecnologias e características relacionadas aos JEDs para a Educação em Computação por meio de investigações em trabalhos científicos, utilizando revisões e mapeamentos sistemáticos da literatura;
- Estabelecer a Análise Arquitetural: Definir os requisitos arquiteturais e construir a AR, levando em conta os problemas e soluções identificados;

- Estabelecer a Síntese Arquitetural: Modelar a arquitetura em visões arquiteturais, demonstrando suas propriedades, componentes e conexões em diferentes perspectivas de desenvolvimento;
- Realizar a Avaliação da AR proposta: Realizar um estudo de caso para avaliar a AR e obter evidências que possam responder à questão de pesquisa e identificar possíveis melhorias a serem implementadas.

1.7 Metodologia

Para alcançar os objetivos mencionados, pretende-se desenvolver uma AR considerando diversas etapas. Inicialmente, são selecionadas as principais fontes de informação que permitem compreender as características e particularidades do domínio de sistemas voltados para JEDs na Educação em Computação por meio de um Mapeamento Sistemático da Literatura (MSL). Com base nessas informações, procede-se à análise arquitetural, onde são estabelecidos os atributos de qualidade e os requisitos arquiteturais da AR.

Em seguida, na etapa de síntese arquitetural, define-se a AR para os JEDs no contexto da Educação em Computação, incluindo seus objetivos, elementos e descrições arquiteturais. Por fim, a AR proposta é avaliada por meio de inspeções utilizando o *checklist* FERA e o método ATAM, incluindo a análise dos resultados obtidos a partir dessas avaliações.

1.8 Organização

Este trabalho está organizado como a seguir. Além dessa introdução, no Capítulo 2 são apresentados conceitos e terminologias relacionados a JEDs, AS e AR, além dos trabalhos relacionados. A AR proposta é apresentada no Capítulo 3. Em seguida, a avaliação realizada é apresentada no Capítulo 4. Por fim, no Capítulo 5 são discutidos os principais resultados deste trabalho, as contribuições e perspectivas de trabalhos futuros.

2 Fundamentação Teórica e Trabalhos

Relacionados

Este capítulo realiza uma análise da crescente relevância dos jogos digitais nos contextos acadêmico e industrial, com destaque nos Jogos Educacionais Digitais (JEDs), e discute a importância de métodos e processos de software, como as Arquiteturas de Software (AS) e Arquiteturas de Referência (AR), para o desenvolvimento de sistemas de alta qualidade. Além disso, apresenta uma análise de estudos que, embora não estejam diretamente focados no desenvolvimento de JEDs para a Educação em Computação, trazem conceitos, estruturas e soluções que podem ser adaptados ou servir de base para o desenvolvimento de uma AR específica para esse contexto. Diversas ARs já foram propostas para domínios variados, como para teste de software (OLIVEIRA, 2010), sistemas robóticos baseados em serviço (FEITOSA, 2013), indústria automotiva (STARON; STARON; DURISIC, 2017), televisão digital (DUARTE, 2012), navegadores da Internet (GROSSKURTH; GODFREY, 2005), Computação em Nuvem (LIU et al., 2011), entre outros. No entanto, até o momento, não foi identificada uma AR específica para apoiar o desenvolvimento de JEDs para a Educação em Computação.

Portanto, esse Capítulo está organizado da seguinte forma: Na Seção 2.1 são abordados os aspectos relacionados aos JEDs, incluindo a aprendizagem baseada em JEDs, o desenvolvimento, a integração com o currículo escolar, além dos desafios e perspectivas desses jogos. Na Seção 2.2 são apresentados os conceitos, terminologias e representações pertinentes à AS. Na Seção 2.3 são discutidos os conceitos, terminologias e classificações, além de processos para estabelecer ARs. Na Seção 2.4 são detalhados os estudos e uma análise comparativa sobre esses estudos. Por fim, na Seção 2.5 são discutidas as considerações finais deste capítulo.

2.1 Jogos Educacionais Digitais

Desde seu surgimento em 1962, os jogos digitais passaram a fazer parte da cultura, influenciando aspectos sociais, políticos, econômicos e tecnológicos (MELO et al., 2020). Atualmente, os jogos são parte integrante da vida de muitas pessoas, especialmente dos mais jovens, sendo acessíveis em diversas plataformas, como consoles, computadores e dispositivos móveis, e abrangendo uma variedade de gêneros (SOUZA; PRATES, 2022). Esse cenário despertou o interesse de pesquisadores de diversas áreas, como *design*, Ciência da Computação e Literatura, sendo a educação um dos principais campos em que os jogos são aplicados atualmente (MELO et al., 2020).

Os jogos digitais têm revolucionado o ambiente educacional, destacando-se pela sua atratividade e pela capacidade de apoiar os alunos no desenvolvimento cognitivo, tanto no aspecto lógico quanto no crítico (MAYER et al., 2022). Por isso, os jogos digitais estão sendo incorporados às salas de aula como ferramentas lúdicas no processo de ensino-aprendizagem (SILVA et al., 2022).

Os Jogos Educacionais Digitais (JEDs) são softwares que integram conteúdos e atividades com finalidades pedagógicas, promovendo tanto o entretenimento quanto o aprimoramento do processo de ensino-aprendizagem (LAUTERT; PIRES; LAGO, 2023). Além disso, esses JEDs desempenham um papel significativo no apoio ao processo educacional, favorecendo a retenção de informações, estimulando a autonomia, a criatividade, o cumprimento de instruções e incentivando a tomada de decisões (SAMPAIO; PEREIRA, 2022). Com isso, os JEDs se mostram ferramentas instrucionais eficazes devido às suas características, pois conseguem divertir e engajar ao mesmo tempo que facilitam o aprendizado (DUTRA et al., 2021).

O uso de JEDs no ensino de Computação tem se mostrado uma estratégia instrucional eficaz, promovendo uma aprendizagem ativa e o desenvolvimento de novos conhecimentos, habilidades e atitudes (CLEMENTINO et al., 2022). Nesse contexto, observa-se um crescimento significativo nas pesquisas voltadas para a integração do ensino com a diversão por meio do desenvolvimento de JEDs (GENESIO et al., 2024b). No entanto, direcionar o entusiasmo que muitos estudantes têm pelos jogos para atividades educacionais é um desafio significativo (FEICHAS; SEABRA; SOUZA, 2021).

Portanto, é fundamental promover iniciativas que apoiem o desenvolvimento de JEDs de alta qualidade, bem como criar novos métodos para orientar as equipes de desenvolvimento, garantindo que as soluções propostas proporcionem melhores resultados para o público-alvo (PETRI; WANGENHEIM; BORGATTO, 2019).

2.1.1 Desenvolvimento de Jogos Educacionais Digitais

O desenvolvimento de jogos é complexo, demorado e exige uma combinação de habilidades técnicas e criativas (DAIREL; CATTELAN; ARAÚJO, 2023). Nesse contexto, esse processo exige uma equipe multidisciplinar, integrando conhecimentos de programação, artes, *design* e música para criar um produto de qualidade alinhado aos padrões de mercado (ARAÚJO; ARANHA; MADEIRA, 2018). Com isso, enquanto alguns desenvolvedores têm formação em Computação, outros vêm de áreas criativas e podem não possuir o mesmo nível de conhecimento técnico para programar um jogo (FILHO; LOPES; SHITSUKA, 2019).

Vale a pena mencionar que o processo de desenvolvimento de um jogo pode variar conforme o orçamento, o tamanho da equipe, a finalidade do jogo, a plataforma, o público-alvo, o gênero, entre outros fatores (AIRES; BARBOSA; MADEIRA, 2020). No entanto, esse processo geralmente passa por várias etapas antes do lançamento ao público, incluindo concepção, prototipagem, desenvolvimento, testes e implantação (KRAMARZEWSKI; NUCCI, 2018). O sucesso de cada uma dessas etapas depende da execução bem-sucedida da etapa anterior, impactando diretamente o resultado do jogo (AIRES; BARBOSA; MADEIRA, 2020). Além disso, é crucial que os dados gerados por esses jogos sejam analisados sob a perspectiva do percurso de aprendizagem e forneçam *feedback* tanto para professores quanto para alunos (ROCHA; BITTENCOURT; ISOTANI, 2015).

Para apoiar o desenvolvimento de JEDs, diversas abordagens têm sido empregadas para auxiliar nesse processo, como o uso de *frameworks* (OLIVEIRA et al., 2018). Esses *frameworks* economizam tempo e recursos financeiros ao oferecer uma abstração elevada que representa o mundo real por meio de um esquema conceitual, além de permitir a reutilização de códigos testados por outros desenvolvedores (OLIVEIRA et al., 2018). Outra abordagem é a aplicação de Inteligência Artificial (IA) no desenvolvimento

de JEDs, por meio de Aprendizado de Máquina (AM), que permite que o agente aprenda automaticamente a tomar boas decisões com base em sua experiência (TEIXEIRA et al., 2020). Finalmente, as arquiteturas possibilitam a criação dinâmica de JEDs com escalabilidade, oferecendo ferramentas e recursos para a criação, personalização e modificação flexível e interativa dos jogos (DAIREL; CATTELAN; ARAÚJO, 2023).

2.1.2 Ensino de Computação

O processo de ensino-aprendizagem de tópicos associados à Computação não é uma tarefa trivial, pois envolve conceitos complexos e uma carga significativa de conhecimentos que não são facilmente assimilados, especialmente para estudantes iniciantes (ARIMOTO; OLIVEIRA, 2019). Nesse cenário, os alunos enfrentam diversas dificuldades durante o aprendizado de conceitos de Computação, que vão desde a falta de compreensão do raciocínio lógico e dificuldade em desenvolver um pensamento abstrato até a falta de motivação, que muitas vezes é causada pelas próprias dificuldades (MARTINS; RODRIGUES; KNIHS, 2019).

Para muitos estudantes de Computação, o método tradicional de ensino, fundamentado em aulas expositivas, é ineficaz e desmotivador (ARIMOTO; OLIVEIRA, 2019), comprometendo a aprendizagem dos conceitos de programação e contribuindo para a evasão nas disciplinas iniciais dos cursos de Computação (KRZYZANOWSKI et al., 2019). Embora os professores, de modo geral, estejam em busca de novas abordagens instrucionais, manter o engajamento dos alunos e evitar a evasão continua sendo um dos principais desafios a serem enfrentados (SILVA; FERNANDES; SANTOS, 2018).

Diante do aumento das reprovações e da evasão em cursos de Computação, estratégias didáticas têm ganhado destaque ao buscar prender a atenção dos alunos, oferecendo-lhes maior protagonismo nas aulas e transformando a forma de ensino e aprendizagem (CLASSE; CASTRO, 2020a). Entre essas estratégias, os JEDs têm se mostrado uma ferramenta instrucional eficaz para apoiar o processo de ensino-aprendizagem no contexto da Educação em Computação (GENESIO et al., 2024b).

Desse modo, os JEDs são frequentemente utilizados para oferecer aos estudantes mais oportunidades práticas de aprendizagem em um ambiente seguro e controlado na área

da Computação (PETRI; WANGENHEIM; BORGATTO, 2019). Além disso, esses JEDs na Computação promovem o aumento do interesse dos alunos pela área, além de estimular sua autonomia, criatividade e pensamento crítico (GARCÊZ; OLIVEIRA, 2022). Por fim, a Base Nacional Comum Curricular (BNCC) da Computação reforça a importância da integração das tecnologias digitais no processo educacional, reconhecendo os JEDs como uma ferramenta pedagógica eficaz (BRASIL, 2022).

2.1.3 Desafios e Perspectivas

Embora se discutam os benefícios dos JEDs para a aprendizagem, há poucas evidências de que eles produzam resultados educacionais válidos e duradouros (PETRI; WANGENHEIM; BORGATTO, 2019). Além disso, a inadequação dos métodos utilizados pelas equipes de desenvolvimento resulta em soluções frequentemente ineficazes, não reproduzíveis e com evidências pouco claras de resultados positivos (MADEIRA; SERRANO, 2020). Com isso, a eficácia dos JEDs é frequentemente questionada devido à percepção dos alunos de que são entediantes, o que leva à desmotivação e ao desinteresse (ALCANTARA; SANTOS, 2023).

Um dos fatores que contribuem para esse problema é o desenvolvimento *ad-hoc* dos JEDs, sem seguir uma metodologia de desenvolvimento formalizada, o que resulta em avaliações superficiais de seu impacto no processo de ensino-aprendizagem dos estudantes (BATTISTELLA; WANGENHEIM, 2016). Além disso, a falta de rigor científico, comum na comunidade de Informática na Educação, e o uso frequente de amostras pequenas nos estudos comprometem a confiabilidade e a validade dos resultados, levantando dúvidas sobre a qualidade e a efetividade desses JEDs (PETRI; WANGENHEIM; BORGATTO, 2019).

Para que os resultados sejam considerados válidos, é necessário revisar os processos de desenvolvimento para manter o equilíbrio entre aprendizagem e entretenimento no JED, analisando quais estruturas têm potencial para promover o desenvolvimento cognitivo (MELO et al., 2020). Portanto, é essencial que os JEDs sejam bem projetados para evitar desvio de seu propósito central e para proporcionar um aprendizado efetivo e significativo para os alunos (GENESIO et al., 2024a).

Dessa forma, é essencial que os JEDs sejam bem planejados e estruturados desde a concepção inicial para garantir uma alta qualidade (AIRES; BARBOSA; MADEIRA, 2020). Para alcançar isso, é necessário garantir um alinhamento adequado entre mecânica, estética, narrativa e tecnologia (XEXÉO; TAUCEI, 2021). Além disso, é fundamental que os JEDs integrem entretenimento e elementos educacionais para garantir que sejam eficazes nos resultados do processo de ensino-aprendizagem (PETRI; WANGENHEIM; BORGATTO, 2019).

2.2 Arquitetura de Software

Seguindo as tendências e demandas do mercado e da academia, os sistemas de software estão se expandindo e se tornando cada vez mais complexos (BASSANI, 2022). Em resposta a essa demanda, é necessário adotar abordagens mais sistemáticas e eficientes para a estruturação e desenvolvimento desses sistemas (PETRI; WANGENHEIM; BORGATTO, 2019).

Nesse contexto, estruturas e processos, nomenclaturas, gerenciamento de configuração e arquiteturas desempenham papéis fundamentais no desenvolvimento de sistemas (LAND, 2002). Em particular, as Arquiteturas de Software (AS) são essenciais para a organização, estruturação e qualidade dos sistemas, considerando suas metodologias de desenvolvimento, o uso de modelos, arquiteturas de referência, padrões de projeto e estilos e táticas arquiteturais (BASS; CLEMENTS; KAZMAN, 2012).

A primeira menção ao termo Arquitetura de Software ocorreu em 1969 (KRUCHTEN; OBBINK; STAFFORD, 2006). Até o final da década de 1980, o termo arquitetura era predominantemente associado à arquitetura de sistemas, referindo-se à estrutura física de um computador (KRUCHTEN; OBBINK; STAFFORD, 2006). No entanto, o conceito de Arquitetura de Software como uma disciplina distinta começou a emergir no início da década de 1990, evidenciado pelo aumento das contribuições tanto da indústria quanto da academia (BOEHM, 2006).

De acordo com Wasserman (1996), a AS é vista como o principal fator na determinação dos aspectos de qualidade e manutenibilidade do software. Além disso, o

Software Engineering Institute (SEI)¹ enfatiza a forte relação entre os requisitos de qualidade dos sistemas de software e suas arquiteturas, considerando-as como os elementos essenciais para a qualidade dos sistemas.

Uma AS pode ser considerada como um conjunto de decisões importantes sobre a organização de um sistema de software, a seleção dos elementos estruturais e respectivas interfaces que compõem o sistema, bem como o seu comportamento, tal como especificado nas colaborações entre esses elementos (BOOCH; RUMBAUGH; JACOBSON, 1999). Em particular, Booch et al. (1999) complementa a composição desses elementos estruturais e comportamentais em subsistemas progressivamente maiores e o estilo de arquitetura que orienta essa organização - esses elementos e respectivas interfaces, as suas colaborações e a sua composição.

Por fim, segundo Bass, Clements e Kazman (2012), a AS de um programa ou sistema de computação é a estrutura, ou estruturas do sistema, que inclui elementos de software, as propriedades visíveis externamente desses elementos e as relações entre eles. Além disso, os autores acrescentam que as propriedades visíveis externamente se referem às suposições que outros componentes podem fazer de um componente, como seus serviços fornecidos, características de desempenho, tratamento de falhas, uso de recursos compartilhados e assim por diante.

2.2.1 Terminologia e Conceitos Básicos

Na área de AS, diversos conceitos e terminologias têm sido definidos para estabelecer um vocabulário comum entre pesquisadores. Os termos mais frequentemente utilizados nesse vocabulário são: Arquitetura Concreta, Instância Arquitetural, Tática Arquitetural, Estilo Arquitetural, Padrão Arquitetural, Modelo de Referência e AR.

A **Arquitetura Concreta**, um sinônimo de AS, consiste em um conjunto coerente de padrões e especificações que orientam o desenvolvimento de cada componente de um sistema de software (SEI)¹. Segundo Angelov, Trienekens e Grefen (2008), o termo Arquitetura Concreta refere-se à descrição arquitetural de um sistema de software real. Essa descrição define um conjunto de funcionalidades e aborda determinadas qualidades

¹Disponível em: <https://www.sei.cmu.edu/>

relacionadas ao sistema, aos negócios e à arquitetura, conforme as exigências dos *stakeholders*.

As qualidades do sistema, como disponibilidade e modificabilidade, são requisitos dos *stakeholders* para o sistema a ser desenvolvido. As qualidades de negócios, como custo e tempo de lançamento no mercado, são objetivos empresariais que influenciam a arquitetura do sistema. Já as qualidades de arquitetura, como integridade conceitual e construtibilidade, dizem respeito às características da própria arquitetura (ANGELOV; TRIENEKENS; GREFEN, 2008).

Para um sistema de software específico, pode-se descrever sua arquitetura, a qual é chamada de **Instância Arquitetural** (NAKAGAWA, 2006). Assim, o termo Instância Arquitetural é frequentemente utilizado como sinônimo de Arquitetura Concreta, especialmente em contextos de aplicação ou problemas específicos, frequentemente derivados da utilização de uma AR (NAKAGAWA, 2006). Por sua vez, a **Tática Arquitetural** refere-se a uma decisão de *design* que impacta a obtenção de atributos de qualidade específicos em um sistema de software (BASS; CLEMENTS; KAZMAN, 2012). Nesse contexto, as táticas oferecem aos *designers* métodos comprovados para melhorar ou gerenciar esses atributos, orientando a tomada de decisões de *design* que estejam alinhadas ao comportamento desejado do sistema (BASS; CLEMENTS; KAZMAN, 2012).

O **Estilo Arquitetural** é uma especialização de tipos de elementos e relações, juntamente com um conjunto de restrições sobre como podem ser utilizados (BASS; CLEMENTS; KAZMAN, 2012). Os estilos arquiteturais permitem aplicar conhecimento especializado de *design* a uma determinada classe de sistemas e apoiar essa classe de *design* de sistemas com ferramentas, análises e implementações específicas do estilo. Por sua vez, Kruchten (2004) define que o estilo arquitetural pode ser definido pela seleção de uma estrutura arquitetônica, por *middleware*, por um conjunto recomendado de padrões ou por uma técnica ou ferramenta de descrição de arquitetura. Alguns exemplos de estilos arquiteturais são *Pipe-And-Filter*, Cliente-Servidor e Arquitetura Orientada a Serviços (SOA) (KRUCHTEN, 2004).

Um **Padrão Arquitetural** descreve um problema de *design* recorrente específico que surge em contextos de *design* específicos e apresenta *designs* de arquitetura bem com-

provados para sua solução (FRANK et al., 1996). Além disso, Sommerville (2011) define o Padrão Arquitetural como uma descrição abstrata e estilizada de boas práticas que foram experimentadas e testadas em diversos sistemas e ambientes. Ele deve descrever uma organização de sistema que obteve sucesso em implementações anteriores, incluindo informações sobre quando o uso desse padrão é apropriado, além de seus pontos fortes e fracos (SOMMERVILLE, 2011). Um exemplo comum é o padrão MVC (*Model View Controller*), que separa a funcionalidade do sistema em três componentes: modelo, visualização e controlador, sendo o controlador responsável por mediar as interações entre o modelo e a visualização.

Padrões e Estilos Arquiteturais são usados como conceitos similares. Porém, a diferença entre padrão e estilo arquitetural é que o primeiro sugere um *design* de arquitetura baseado no problema e no contexto, enquanto o último se concentra no *design* arquitetural, com orientação mais leve sobre quando um estilo específico pode ou não ser útil (GARLAN et al., 2010).

Já o **Modelo de Referência**, conforme definido por Bass, Clements e Kazman (2012), refere-se à divisão de funcionalidades apresentadas de maneira conjunta e ao fluxo de dados pelo qual suas partes se relacionam. Esses modelos são independentes de padrões específicos, tecnologias, implementações ou outros detalhes concretos (MACKENZIE et al., 2006), podendo variar desde modelos conceituais simples até ontologias bem estabelecidas (OLIVEIRA et al., 2010). Além disso, Bass, Clements e Kazman (2012) destacam que os Modelos de Referência podem auxiliar na criação de ARs, uma vez que os conceitos e funcionalidades definidos nesses modelos podem ser mapeados para entidades de AS.

Por fim, as **Arquiteturas de Referência** (ARs) são definidas como mapeamentos entre Modelos de Referência e elementos de software que implementam as funcionalidades descritas nesses modelos. Nesse contexto, uma AR consiste em componentes de software e nos relacionamentos entre eles, que implementam funcionalidades relativas às partes definidas no Modelo de Referência. A Figura 2.1 ilustra o relacionamento entre Modelos de Referência e Estilos Arquiteturais, que podem ser utilizados para estabelecer ARs. A partir do estabelecimento da AR, são projetadas Arquiteturas Concretas voltadas para a construção de sistemas.

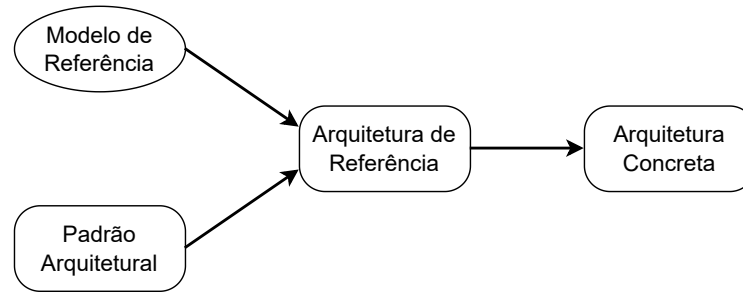


Figura 2.1: Relacionamento entre Modelo de Referência, Padrão Arquitetural, Arquitetura de Referência e Concreta (Adaptado de Bass, Clements e Kazman, 2003).

2.2.2 Representações de Arquiteturas de Software

Devido à importância da AS no desenvolvimento de sistemas, existem esforços direcionados à sua representação adequada (FARSHIDI; JANSEN; WERF, 2020). No entanto, geralmente, as AS são representadas de forma informal, utilizando modelos que contêm componentes e conectores (*box-and-lines*) (NAKAGAWA; ANTONINO, 2022). Os componentes representam as principais funcionalidades do sistema e os dados armazenados nele, enquanto os conectores indicam a comunicação entre os componentes, definindo as interações entre eles (LAND, 2002). Embora esse tipo de representação arquitetural seja útil para fornecer uma visão geral do sistema, é crucial destacar que, se usada isoladamente, pode gerar problemas relacionados a interpretações enganosas, ambíguas e inconsistentes (GARLAN et al., 2010).

Devido à necessidade de oferecer representações formais para descrever AS com uma sintaxe e estrutura conceitual que permitissem sua caracterização (NAKAGAWA; ANTONINO, 2022), diversas linguagens de descrição arquitetural (do inglês, *Architecture Description Languages* - ADL) têm sido propostas e aplicadas na especificação de projetos arquiteturais (MEDVIDOVIC; ROSENBLUM; TAYLOR, 1999; GARLAN; ALLEN; OCKERBLOOM, 1994; MORICONI; QIAN; RIEMENSCHNEIDER, 1995).

Essas linguagens fornecem abstrações adequadas para a modelagem de grandes sistemas, assegurando detalhes suficientes para definir as propriedades desejadas (MEDVIDOVIC et al., 2002). Além disso, cada ADL adota uma abordagem específica para a especificação e evolução de uma arquitetura, abordando questões de avaliação relacionadas à modelagem e explorando aspectos detalhados do sistema (MEDVIDOVIC et

al., 2002). No entanto, devido à natureza específica das ADLs, a integração de modelos desenvolvidos com essas linguagens a outros artefatos de desenvolvimento pode ser um desafio (MEDVIDOVIC et al., 2002).

Além do uso de linguagens de descrição arquitetural, a adoção de diferentes visões para representar AS foram amplamente investigadas (KRUCHTEN, 1995; MEDVIDOVIC et al., 2002; COMMITTEE et al., 2000). De acordo com Buschmann et al. (1996), cada visão arquitetural reflete um aspecto específico da arquitetura, que corresponde a determinadas propriedades do sistema de software. Nesse contexto, por meio dessas visões, é possível observar diferentes características do sistema, dependendo do ponto de vista adotado na análise da arquitetura (LAND, 2002). No entanto, ainda não há um consenso entre os pesquisadores da área de AS sobre quais visões são essenciais e quais métodos e técnicas são mais adequados para expressá-las (COMMITTEE et al., 2000). Apesar disso, as visões estrutural e comportamental da AS têm sido as mais consideradas (COMMITTEE et al., 2000).

Kruchten (1995) propôs um conjunto mais amplo de visões para a representação de arquiteturas de software, conhecido como o modelo de visualização 4+1 (*4+1 View Model*), que organiza a descrição de uma AS em cinco visualizações simultâneas, cada uma focada em um conjunto específico de preocupações. Os arquitetos capturam suas decisões de *design* em quatro dessas visualizações e utilizam a quinta para ilustrá-las e validá-las.

Nesse modelo, a **Visão Lógica** aborda principalmente os requisitos funcionais, representando os serviços que o sistema deve fornecer aos usuários finais. A **Visão de Processo** trata dos requisitos não funcionais, como desempenho, disponibilidade e tolerância a falhas, além de definir como os *threads* de controle executam as operações das classes da visão lógica. A **Visão de Desenvolvimento** foca na organização dos módulos de software no ambiente de desenvolvimento, onde o software é dividido em componentes, como bibliotecas ou subsistemas, organizados em camadas com interfaces bem definidas. Já a **Visão Física** considera requisitos como disponibilidade, confiabilidade e escalabilidade, abordando as diferentes configurações físicas para desenvolvimento, teste e implantação, garantindo que o mapeamento do software para os nós seja flexível

e cause mínimo impacto no código-fonte (KRUCHTEN, 1995). A Figura 2.2 ilustra esse modelo de visualização 4+1, usado para mostrar o sistema sob várias perspectivas, como usuários finais, programadores, integradores de sistemas e engenheiros de sistemas.

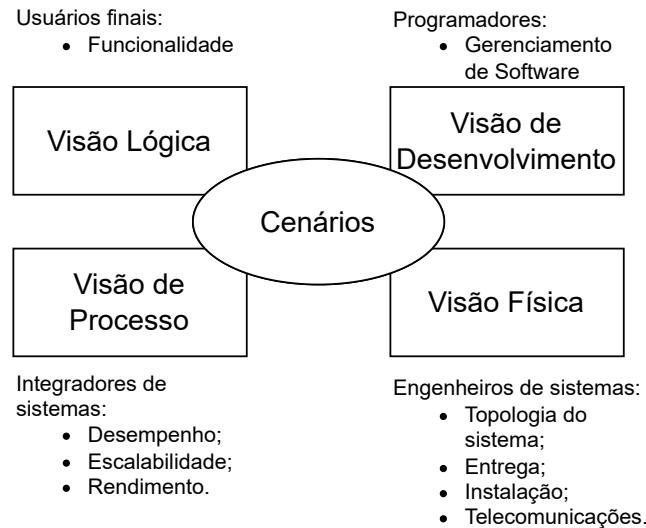


Figura 2.2: Modelo de Visualização 4+1 (Adaptado de Kruchten, 1995).

A UML (*Unified Modeling Language*)² tem se destacado nos últimos anos como uma linguagem relevante para a representação de AS, sendo amplamente utilizada tanto no meio acadêmico quanto na indústria (JÚNIOR; FARIAS; SILVA, 2022). A UML oferece um grande conjunto de construções predefinidas e é suportada por diversas ferramentas de modelagem (MEDVIDOVIC et al., 2002). Além disso, é considerada uma linguagem semi-formal, com uma representação mais próxima da implementação, o que facilita sua compreensão pelos desenvolvedores do sistema (JÚNIOR; FARIAS; SILVA, 2022). De acordo com Garlan et al. (2010), o uso da UML para documentar AS traz vantagens, como a familiaridade dos desenvolvedores, a facilidade de mapeamento para a implementação e o suporte de ferramentas especializadas.

Além disso, diversos perfis especializados da UML têm surgido, como a SysML (*Systems Modeling Language*)³, voltada para modelar requisitos, comportamentos, estruturas e parâmetros de sistemas, permitindo a integração com outros modelos de engenharia (BAJAJ; FRIEDENTHAL; SEIDEWITZ, 2022). A SysML é uma linguagem semi-formal usada na especificação, análise, projeto e verificação de sistemas complexos, abrangendo

²Disponível em: <https://www.uml.org/>

³Disponível em: <https://sysml.org/>

hardware, software, pessoas e procedimentos (BOUWMAN; LUTTIK; WAL, 2021). Ela oferece vários tipos de diagramas, como os de Atividades, Sequência, Casos de Uso e Requisitos, para apoiar todo o ciclo de vida do sistema, conforme o padrão ISO/IEC/IEEE 42010 (JÚNIOR; MISRA; SOARES, 2019).

2.3 Arquitetura de Referência

Arquiteturas de referência (ARs) têm ganhado destaque na área de AS, fornecendo diretrizes mais precisas para a especificação de arquiteturas concretas em um determinado domínio de aplicação (ANGELOV; TRIENEKENS; GREFEN, 2008). Como resultado, as ARs exercem uma influência direta na qualidade e no *design* de um conjunto de arquiteturas concretas, bem como nos sistemas derivados dessas arquiteturas (ANGELOV; GREFEN; GREEFHORST, 2009).

Uma das primeiras definições de AR foi proposta por Kruchten (2000), que descreve uma AR como um padrão arquitetural predefinido, ou conjunto de padrões, possivelmente parcial ou completamente instanciados, projetados e comprovados para uso em contextos comerciais e técnicos específicos, juntamente com artefatos de suporte para permitir seu uso. Além disso, o autor acrescenta que esses artefatos são frequentemente coletados de projetos anteriores.

Por outro lado, alguns pesquisadores não incluem padrões arquiteturais em suas definições, mas focam em elementos de software de alto nível. Nesse contexto, Garland e Anthony (2003) definem uma AR como o conjunto de elementos de alto nível envolvidos em aplicações de um domínio específico, juntamente com suas interações. Da mesma forma, Bass, Clements e Kazman (2012) afirmam que uma AR faz o mapeamento de funcionalidades para elementos de software (que implementam cooperativamente essas funcionalidades) e os fluxos de dados entre eles.

Adicionalmente, Muller (2008) também acrescenta que ARs podem ser utilizadas tanto como guia para o desenvolvimento de arquiteturas concretas quanto como ferramenta de padronização, promovendo a interoperabilidade entre sistemas ou seus componentes. Assim, uma AR pode ser aplicada em diferentes contextos, resultando no desenvolvimento de instâncias arquiteturais variadas, conforme os *stakeholders* e os objetivos

específicos de cada organização, como mostrado na Figura 2.3. O principal objetivo das ARs é facilitar o entendimento de um determinado domínio, fornecendo um vocabulário comum, além de esclarecer as partes que o compõem e as relações entre elas (MULLER, 2008a).

Além disso, Angelov, Grefen e Greefhorst (2009) definem uma AR como uma arquitetura genérica para uma classe de sistemas de informação, que serve como base para o *design* de arquiteturas específicas dessa classe. De forma semelhante, Muller (2008) afirma que uma AR é criada capturando os fundamentos das arquiteturas existentes e levando em consideração as necessidades e oportunidades futuras, que vão desde tecnologias específicas, até padrões, modelos de negócios e segmentos de mercado.

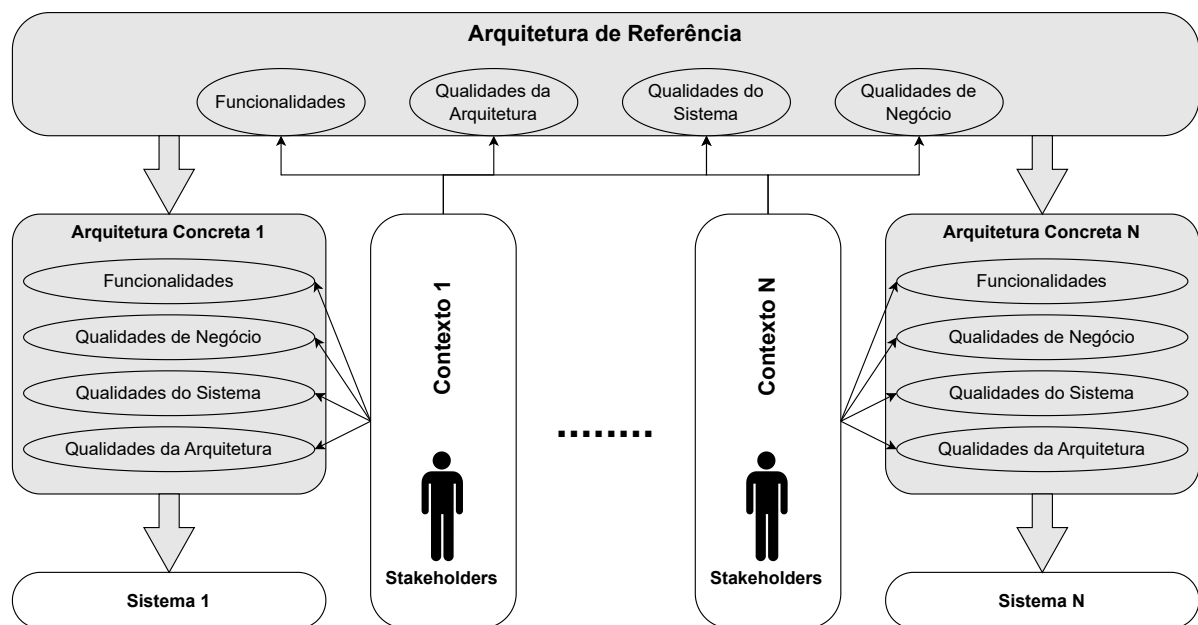


Figura 2.3: Interação de *Stakeholders* e Contextos entre Arquiteturas Concretas e de Referência (Adaptado de Angelov, Trienekens e Grefen, 2008).

Embora as definições acima variem, elas compartilham a mesma essência: a reutilização de conhecimento sobre o desenvolvimento de software em um determinado domínio, especialmente no que se refere ao *design* arquitetural (NAKAGAWA et al., 2014). Nesse contexto, Nakagawa et al. (2014) afirmam que uma AR se refere a uma arquitetura que abrange o conhecimento sobre como projetar arquiteturas concretas de sistemas de um determinado domínio de aplicação. Ela deve englobar regras de negócios, estilos arquiteturais (às vezes chamados de padrões arquiteturais, que também podem abordar atributos

de qualidade), melhores práticas de desenvolvimento de software (como decisões arquitetônicas, restrições do domínio, legislação e normas), além dos elementos de software que suportam o desenvolvimento de sistemas para esse domínio. Tudo isso deve ser sustentado por uma terminologia de domínio unificada, clara e amplamente compreendida.

2.3.1 Classificações de Arquiteturas de Referência

ARs têm sido desenvolvidas por pesquisadores e pela indústria em diversos contextos e com variados objetivos. De acordo com Angelov, Grefen e Greefhorst (2009), uma AR pode ser classificada com base em: (i) o contexto para o qual foi criada; (ii) seu propósito de uso; e (iii) os recursos utilizados em sua construção. A seguir, essa classificação é apresentada:

- **Contexto de Aplicação:** ARs podem ser desenvolvidas para atender às necessidades de uma única organização, que busca criar uma família de produtos semelhantes, ou para múltiplas organizações que compartilham características em comum, como localização geográfica ou domínio de mercado;
- **Propósito de Uso:** Dois principais usos podem ser identificados: padronização e facilitação. A padronização de arquiteturas concretas visa garantir a interoperabilidade entre sistemas, estabelecendo um consenso arquitetural. Já a facilitação busca fornecer diretrizes, na forma de boas práticas e padrões, para o desenvolvimento de arquiteturas concretas;
- **Insumos Utilizados:** Uma AR pode ser projetada antes da existência de qualquer sistema, baseada no conhecimento do domínio, sendo chamada de AR preliminar. Alternativamente, pode ser desenvolvida a partir da experiência acumulada com sistemas já implementados, sendo então classificada como uma AR clássica.

Com base na classificação do contexto, propósito e insumos da AR, Angelov, Grefen e Greefhorst (2009) identificam seis tipos distintos de AR, descritos a seguir:

- **Tipo 1:** Classificação para um contexto de múltiplas organizações, para padronizar uma AR, baseada em insumos clássicos. Esse tipo é utilizado para alcançar

consenso arquitetural entre as organizações e, por isso, é projetado com descrições de componentes e suas interfaces em altos níveis de abstração. O detalhamento dos componentes não é necessário, pois cada organização pode definir os detalhes concretos que forem mais relevantes para seu contexto;

- **Tipo 2:** Destinado a um contexto de uma única organização, para padronizar a AR, utilizando insumos clássicos. Têm por objetivo apoiar a criação de uma família de sistemas com características semelhantes. Essas arquiteturas possuem características mais concretas, podendo especificar detalhes como tecnologias, aplicações existentes e padrões;
- **Tipo 3:** Projetado para múltiplas organizações, para facilitar ARs futuras, utilizando insumos baseados em experiências anteriores (clássico). Esse tipo de arquitetura é utilizado na construção de produtos de software, oferecendo uma descrição mais concreta e fornecendo orientações para o desenvolvimento;
- **Tipo 4:** Destinado a uma única organização, para facilitar a criação de ARs futuras, utilizando insumos baseados em experiências anteriores (clássico). Semelhante ao Tipo 2, porém projetado para facilitar o desenvolvimento e descrito de forma mais informal;
- **Tipo 5:** Classificado para um contexto de múltiplas organizações, as ARs desse tipo são desenvolvidas a partir do conhecimento de domínio, antes da existência de sistemas concretos, sendo por isso chamadas de preliminares. Essas arquiteturas definem os componentes necessários para sua implementação, podendo também sugerir algoritmos e protocolos que facilitem a interação entre esses componentes;
- **Tipo 6:** Destinado a uma única organização, esse tipo de AR é geralmente desenvolvido por organizações líderes de mercado, devido ao alto esforço envolvido em sua criação.

No desenvolvimento dessas arquiteturas, diversos tipos de *stakeholders* podem estar envolvidos. A criação de uma AR pode ser conduzida por empresas de software,

organizações de usuários, centros de pesquisa, *designers* de software, gerentes de projeto, organizações de padronização ou uma combinação desses grupos (ANGELOV; GREFEN; GREEFHORST, 2009). Como cada conjunto de ARs possui suas especificidades, diferentes tipos de instituições frequentemente concentram esforços em sua construção (ANGELOV; GREFEN; GREEFHORST, 2009).

Além disso, a previsão dos resultados obtidos a partir de uma AR depende do tipo a que ela pertence (ANGELOV; GREFEN; GREEFHORST, 2009). Arquiteturas desenvolvidas de forma clássica, baseadas na observação de sistemas já implementados, permitem uma estimativa de resultados mais precisa, enquanto as arquiteturas preliminares apresentam maior dificuldade na previsão de seus resultados (ANGELOV; TRIENEKENS; GREFEN, 2008).

2.3.2 Processo para Estabelecer uma Arquitetura de Referência

A sistematização do processo para estabelecer ARs é uma atividade de grande relevância (NAKAGAWA et al., 2009). Nesse contexto, é possível encontrar diversos estudos relacionados ao desenvolvimento de ARs (BAYER et al., 2004; MULLER, 2008b; GALSTER; AVGERIOU, 2011; CLOUTIER et al., 2010; ANGELOV; GREFEN; GREEFHORST, 2012; NAKAGAWA et al., 2014). Esses estudos oferecem processos, diretrizes, princípios e recomendações para aprimorar a engenharia de ARs (CLOUTIER et al., 2010).

Nakagawa et al. (2014) propuseram o ProSA-RA (*Process based on Software Architecture – Reference Architecture*), que é um processo que sistematiza o projeto, a representação e a avaliação de uma AR. Esse processo oferece um conjunto de quatro etapas que orientam o desenvolvimento de uma AR, desde a investigação das fontes de informação do domínio – utilizadas para a definição dos requisitos arquiteturais – até as fases de projeto e avaliação da arquitetura. É válido salientar que vários estudos já aplicaram o ProSA-RA no desenvolvimento de suas ARs (ALLIAN et al., 2023; CAMARGO et al., 2024; PORTOCARRERO et al., 2017; OSSHIRO et al., 2018). Neste trabalho final de curso, o ProSA-RA foi escolhido como o método para o desenvolvimento da AR proposta, sendo, portanto, detalhado a seguir.

Para a aplicação do ProSA-RA, conforme ilustrado na Figura 2.4, os seguintes

passos devem ser seguidos:

- **Passo RA-1: Investigação das Fontes de Informação:** são selecionadas as principais fontes de informação do domínio, que devem fornecer dados sobre os processos, atividades e tarefas a serem automatizados. Essas fontes podem incluir pessoas envolvidas no domínio (como usuários, pesquisadores e especialistas), sistemas já desenvolvidos (quando disponíveis), publicações, documentos relacionados e ontologias. O resultado dessa etapa é um conjunto de fontes de informação, utilizadas na próxima fase para definir os requisitos da AR;
- **Passo RA-2: Estabelecimento dos Requisitos Arquiteturais:** com base nas fontes de pesquisa selecionadas, as informações do domínio são elicitadas, resultando em um conjunto de requisitos para a AR. Em resumo, identificam-se primeiramente os requisitos dos sistemas do domínio e, em seguida, definem-se os requisitos específicos da AR, ou seja, os Requisitos Arquiteturais. Por fim, esses conceitos são classificados como transversais ou não-transversais, para identificar aspectos arquiteturais. Em geral, conceitos que estão associados a muitos requisitos ou a requisitos não-funcionais possuem natureza transversal. Em síntese, as saídas dessa etapa são os requisitos da AR e um conjunto de conceitos relacionados;
- **Passo RA-3: Projeto da Arquitetura de Referência:** A partir dos Requisitos Arquiteturais e dos conceitos identificados na etapa anterior, é elaborada uma descrição arquitetural da AR, considerando também padrões e estilos arquiteturais. Em seguida, a descrição arquitetural é construída utilizando visões arquiteturais. Especificamente, as visões de módulos, tempo de execução, implantação e conceitual têm sido sugeridas para descrever essas arquiteturas, embora outras visões possam ser relevantes, dependendo dos diferentes *stakeholders*, do uso ou da natureza da AR. Ao final, essa etapa resulta em uma descrição arquitetônica composta por um conjunto de visões e artefatos adicionais da AR;
- **Passo RA-4: Avaliação da Arquitetura de Referência:** Por fim, inspeções utilizando *checklists* são empregadas para avaliar a qualidade da AR. *Checklists* consistem em questionários que orientam os revisores na identificação de defeitos

em documentos. Nesse caso, são utilizados para inspecionar a documentação da AR estabelecida. Essa técnica permite a avaliação de características de qualidade, como manutenibilidade, desempenho, segurança, usabilidade e reúso. Além disso, defeitos como omissões, ambiguidades, inconsistências e informações incorretas são identificados.

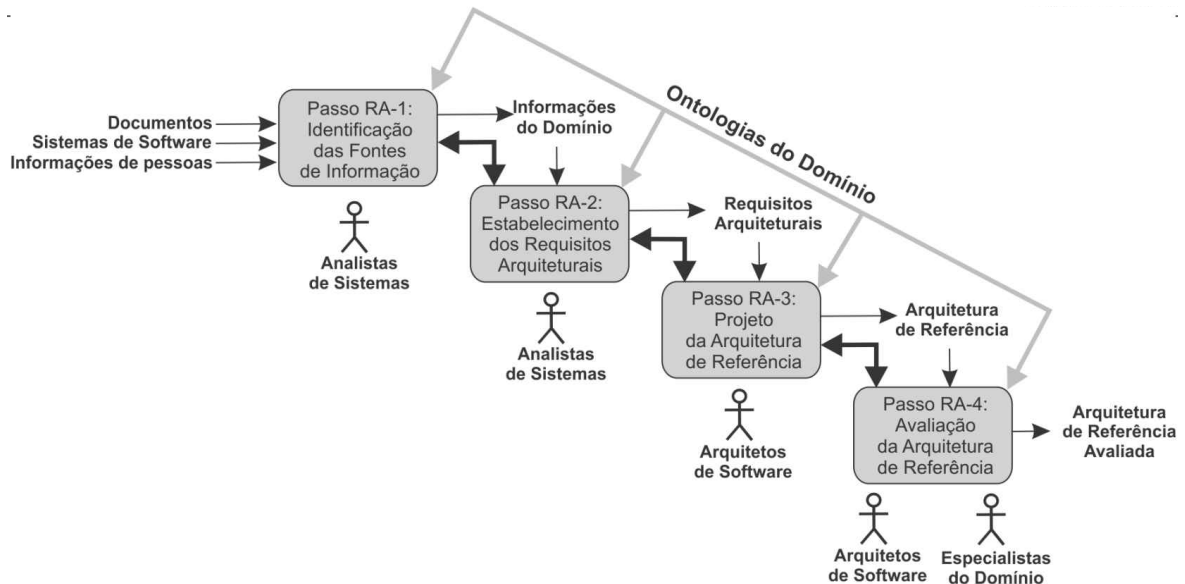


Figura 2.4: Estrutura do ProSA-RA (Adaptado de Nakagawa et al., 2014).

Os quatro passos definidos no ProSA-RA são apoiados pelo RAModel (*Reference Architecture Model*), um modelo de referência para ARs que fornece informações sobre todos os possíveis elementos e seus relacionamentos que poderiam estar presentes em ARs, independentemente do domínio de aplicação ou do propósito dessas arquiteturas (NAKAGAWA; OQUENDO; BECKER, 2012). Em resumo, o RAModel é constituído por quatro grupos de elementos:

- **Domínio:** Contém elementos relacionados a informações específicas e autocontidas do contexto de atuação humana no mundo real, como legislações, padrões e processos de certificação do domínio, que influenciam sistemas e ARs associadas;
- **Aplicação:** Contém elementos que fornecem um bom entendimento da AR, suas capacidades e limitações. Também contém elementos relacionados às regras de negócios (ou funcionalidades) que podem estar presentes em sistemas de software construídos a partir da AR;

- **Infraestrutura:** Refere-se a elementos que podem ser utilizados para construir sistemas de software baseados na arquitetura de referência. Esses elementos são responsáveis por permitir que os sistemas automatizem, por exemplo, processos, atividades e tarefas de um determinado domínio;
- **Elementos Transversais:** Contém elementos que geralmente estão distribuídos e/ou interligados com os elementos dos outros três grupos (domínio, aplicação e infraestrutura). A comunicação (classificada como interna e externa) nos sistemas de software baseados na AR, assim como a terminologia e as decisões de domínio, aparecem de forma disseminada e interconectada na descrição dos outros grupos, sendo, portanto, considerados elementos transversais.

2.4 Trabalhos Relacionados

Fioravanti et al. (2010) discutiram a relevância das ARs na construção de sistemas educacionais e apresentam a arquitetura EDUCAR, uma AR para ambientes educacionais, desenvolvida com base no processo ProSA-RA. A EDUCAR foi projetada para apoiar o desenvolvimento de ambientes educacionais, atendendo a necessidades específicas desse domínio, como avaliação de aprendizagem, comunicação, documentação e personalização. Os principais módulos da EDUCAR incluem Autoria, Ensino, Colaboração e Comunicação, Administração, Documentação e Personalização, cada um responsável por funcionalidades específicas, como criação e disponibilização de conteúdo didático, comunicação entre usuários e gerenciamento de cursos e usuários. A partir da EDUCAR, é possível identificar as funcionalidades essenciais de ambientes educacionais, ou seja, o conjunto central de ferramentas que os compõem. Por fim, os autores não realizaram uma avaliação da AR e indicaram como trabalhos futuros uma inspeção baseada em *checklist* para avaliar a efetividade da AR.

Van der Vegt et al. (2016) realizaram um estudo que discutiu como a complexidade do *design* dos jogos sérios (JS) pode limitar sua eficácia pedagógica. Nesse contexto, eles desenvolveram uma pesquisa para descrever o desenvolvimento da RAGE, uma AR que facilita a reutilização de componentes tecnológicos de JS em diversas lin-

guagens de programação, motores de jogos e plataformas. A RAGE propõe um modelo de desenvolvimento baseado em componentes, permitindo a criação de ativos de software reutilizáveis. Esses ativos abrangem funcionalidades como análise de dados dos jogadores, reconhecimento de emoções e personalização do jogo, podendo ser facilmente integrados em diferentes plataformas sem a necessidade de os desenvolvedores reescreverem essas funcionalidades.

O processo de construção da RAGE foi realizado por meio de uma abordagem de desenvolvimento baseada em componentes. O objetivo principal era garantir a interoperabilidade e portabilidade dos componentes em diferentes motores de jogo, plataformas e linguagens de programação. Já a sua avaliação foi realizada através da validação técnica de seus componentes em quatro linguagens principais de programação: C#, C++, Java e TypeScript/JavaScript. O processo envolveu a implementação de um componente básico de software, que foi testado quanto à sua capacidade de registro, comunicação entre componentes, interoperabilidade com motores de jogo e a utilização de padrões. Esses testes também verificaram a capacidade dos componentes de armazenar e recuperar dados, chamar serviços da web e gerenciar configurações e eventos. Além disso, o motor Unity foi utilizado como plataforma de integração para o componente C#, demonstrando a viabilidade da arquitetura. Segundo os autores, a longo prazo, a expectativa é que a RAGE promova a coesão na indústria de JS, aumentando seu potencial em áreas como educação e treinamento.

Carvalho (2017) conduziu um estudo para apoiar a criação e desenvolvimento de Jogos Sérios (JS) como ferramentas para educação e treinamento. A pesquisa buscou reduzir os custos de desenvolvimento desses jogos, assegurando que cumpram tanto metas educacionais quanto de entretenimento. No estudo, foram apresentadas duas principais contribuições nesse trabalho: o Modelo baseado na teoria da atividade para JS (*Activity Theory-based Model for Serious Games* - ATMSG) e uma AR orientada a serviços para JS (*Service-Oriented Reference Architecture for Serious Games* - SORASG). O ATMSG é um modelo conceitual que liga os objetivos educacionais e de entretenimento a mecânicas de jogo específicas, facilitando a compreensão de como os elementos do jogo impactam os resultados de aprendizagem. Já a SORASG baseia-se no padrão de Arquitetura Orientada

a Serviços (SOA), visando aumentar a reutilização e a eficiência no desenvolvimento de JS. Essa arquitetura também busca padronizar o desenvolvimento de jogos em diferentes gêneros e temas, diminuindo o custo e o tempo de produção.

A eficácia da SORASG foi demonstrada por meio de dois métodos principais. Primeiramente, foi aplicada a Metodologia de Análise de Trade-offs Arquiteturais (*Architecture Trade-off Analysis Method* - ATAM), que avalia a arquitetura em termos de atributos de qualidade, como modificabilidade, desempenho e escalabilidade. Em segundo lugar, a AR foi validada por meio de uma implementação prática, onde um jogo existente foi modificado para utilizar os serviços fornecidos pela SORASG. Esse estudo de caso mostrou como componentes reutilizáveis podem ser integrados de forma eficiente, reduzindo o tempo de desenvolvimento e os custos, enquanto ainda alcança os objetivos educacionais e de entretenimento esperados. Com essas abordagens, os resultados indicam que a combinação do modelo ATMSG e da arquitetura SORASG oferece uma solução eficaz para reduzir os custos de desenvolvimento de JS, sem comprometer a qualidade.

Por fim, Mizutani e Kon (2020) realizaram um estudo para apresentar a *Unlimited Rulebook* (URB), uma AR voltada para mecânicas econômicas em jogos digitais, visando reduzir os custos de desenvolvimento dessas mecânicas, que se caracterizam por serem imprevisíveis, instáveis e complexas. A proposta da URB se apoia no reaproveitamento de conhecimento arquitetural, utilizando técnicas como Despacho de Predicados (*Predicate Dispatching*), o padrão Sistema-Componente-Entidade (*Entity-Component-System* - ECS) e o estilo arquitetural Modelo de Objeto Adaptável (*Adaptive Object-Model* - AOM).

Assim, a URB organiza o subsistema econômico dos jogos em um modelo de referência baseado em três conceitos principais: Integração de subsistemas (define as interfaces e limita o acesso aos dados e comportamentos do subsistema econômico), Modelo de Mecânicas (separa os tipos de recursos e as operações disponíveis no sistema econômico) e Desenvolvimento Iterativo (facilita a modificação das mecânicas econômicas, permitindo a adição de novas funcionalidades sem impactar o código existente).

A URB foi desenvolvida utilizando o processo ProSA-RA. Inicialmente, informações sobre o domínio de mecânicas econômicas foram coletadas por meio de uma revisão sistemática de literatura, consultas a especialistas e análise de jogos e motores de código

aberto. Com o apoio de especialistas, padrões e requisitos recorrentes foram identificados e posteriormente sintetizados em requisitos arquiteturais gerais, agrupados em conceitos de domínio. A arquitetura foi iterativamente refinada com base nessas análises e melhorada a cada ciclo de desenvolvimento.

Além disso, a avaliação da URB foi realizada por meio de quase-experimentos com estudantes de cursos de programação de jogos. Nesses experimentos, os alunos foram divididos em grupos, cada um responsável por desenvolver dois tipos de jogos: um jogo de RPG baseado em turnos e um jogo de defesa de torres em tempo real. A primeira versão dos jogos foi desenvolvida sem a URB, enquanto a segunda utilizou a arquitetura proposta. Os resultados preliminares mostraram que a URB reduziu ligeiramente a quantidade de modificações necessárias para implementar e estender mecânicas econômicas, tornando o processo mais eficiente, especialmente no caso de mecânicas que sofrem alterações durante a execução do jogo. Além disso, provas de conceito mostraram que a URB simplifica a implementação de mecânicas complexas e dinâmicas, demonstrando sua flexibilidade ao lidar com sobreposições de regras e modificações em tempo de execução.

2.4.1 Análise Comparativa

A comparação entre as arquiteturas EDUCAR, RAGE, SORASG e URB evidencia diferentes abordagens no desenvolvimento de AR, cada uma com foco em domínios e justificativas específicas, conforme ilustrado na Tabela 2.1.

A AR EDUCAR, voltada para ambientes educacionais, busca preencher a lacuna de uma AR consolidada nesse campo, enquanto a RAGE e a SORASG focam no desenvolvimento de jogos sérios, com ênfase em reutilização de componentes e modularidade. A URB, por outro lado, se concentra em jogos digitais com uma abordagem voltada à redução de custos e refatoração de mecânicas econômicas. A justificativa para o desenvolvimento de cada AR reflete as demandas específicas de seus domínios, tornando cada uma única em seu contexto de aplicação.

No que se refere ao processo de desenvolvimento, tanto a EDUCAR quanto a URB utilizaram o ProSA-RA, enquanto a RAGE adotou um modelo baseado em componentes, e a SORASG implementou uma abordagem de arquitetura orientada a serviços.

Tabela 2.1: Comparação dos trabalhos relacionados.

Funcionalidade	Arquitetura de Referência			
	EDUCAR	RAGE	SORASG	URB
Domínio	Ambientes educacionais	Jogos sérios	Jogos sérios	Jogos digitais
Justificativa	Disponibilizada	Disponibilizada	Disponibilizada	Disponibilizada
Desenvolvimento	ProSA-RA	Desenvolvimento baseado em componentes	<i>Service-Oriented Architecture</i>	ProSA-RA
Fontes de Informações	Detalhado	Não foi explicitado	Detalhado	Detalhado
Atributos de Qualidade	Não foi explicitado	Detalhado	Detalhado	Não foi explicitado
Requisitos Arquiteturais	Detalhado	Não foi explicitado	Não foi explicitado	Detalhado
<i>Stakeholders</i>	Detalhado	Detalhado	Detalhado	Não foram mencionados
Visões Arquiteturais	Detalhado	Não foi explicitado	Detalhado	Detalhado
Avaliação	Não foi realizado	Realizado	Realizado	Realizado
Nível de Maturidade	Implementação	Uso em situação real	Uso em situação real	Uso em situação real

Esse aspecto diferencia as ARs em termos de metodologias e etapas seguidas para a sua construção. As fontes de informação variam em detalhamento, com EDUCAR e SORASG apresentando dados mais detalhados, enquanto a RAGE e URB oferecem informações de forma mais superficial. A avaliação de atributos de qualidade também varia significativamente: a RAGE enfatiza portabilidade e manutenibilidade, enquanto a SORASG aborda modularidade e interoperabilidade. A EDUCAR e a URB, no entanto, não detalham explicitamente esses atributos.

Em termos de maturidade, as ARs RAGE, SORASG e URB apresentam uso em situações reais, enquanto a EDUCAR está em estágio de implementação para ser avaliada. As visões arquiteturais são mais desenvolvidas nas ARs SORASG e URB, que incluem diagramas detalhados, enquanto a RAGE não apresenta essas visões explicitamente. O processo de avaliação também é mais estruturado em RAGE e SORASG, que passaram por validações técnicas, enquanto a EDUCAR e URB carecem de processos de avaliação

formalizados. A análise comparativa demonstra que, embora compartilhem objetivos semelhantes de reutilização e otimização, as ARs diferem amplamente em suas abordagens e níveis de detalhamento.

2.5 Considerações Finais

Neste capítulo foram apresentados tópicos relacionados a JEDs, AS e AR, os quais são essenciais para o desenvolvimento deste trabalho. Além disso, foram discutidos os principais estudos que servem de base para o desenvolvimento da AR proposta.

Com o crescimento da área de jogos digitais, especialmente os JEDs, os quais são voltados para a Educação em Computação, é evidente a necessidade de uma abordagem mais estruturada para o desenvolvimento dos JEDs, garantindo que eles alcancem seus objetivos. Nesse cenário, o uso de métodos sistemáticos, como AS e, mais especificamente, ARs, exerce um papel crucial na sistematização dos processos de desenvolvimento. Portanto, a utilização de AS e ARs no desenvolvimento de JEDs para a Educação em Computação representa uma oportunidade de sistematização, tanto nos aspectos técnicos quanto educacionais.

Além disso, a partir da análise comparativa das ARs EDUCAR, RAGE, SORASG e URB, fica evidente a importância de uma AR voltada para o desenvolvimento de JEDs para a Educação em Computação. Embora as ARs existentes abordem aspectos como reutilização de componentes, modularidade e otimização de custos em seus respectivos domínios, nenhuma delas é direcionada especificamente para os desafios dos JEDs na Computação.

Por fim, a ausência de uma AR específica para a Educação em Computação, aliada ao fato de que grande parte dos estudantes apresenta um perfil exigente quanto à qualidade, jogabilidade e interatividade dos JEDs, evidencia a necessidade do desenvolvimento de uma solução específica para esse domínio. Com isso, o desenvolvimento de uma AR é essencial para suprir essa lacuna, possibilitando a criação de JEDs de alta qualidade e favorecendo o reúso de componentes de software. Dessa forma, a adoção dessa AR contribuirá para o engajamento dos estudantes, a otimização do processo de ensino-aprendizagem em Computação e a eficiência no desenvolvimento de JEDs.

3 Estabelecimento da RADEG

Este capítulo apresenta a AR para JEDs voltados ao ensino de Computação, denominada RADEG (do inglês, *Reference Architecture for Digital Educational Games*). Essa arquitetura tem como principal objetivo apoiar a fase de projeto arquitetural no desenvolvimento de JEDs para a Educação em Computação. A RADEG foi elaborada utilizando o processo ProSA-RA (NAKAGAWA et al., 2014), que define um conjunto de quatro passos que guiam o desenvolvimento de ARs. Para representar a RADEG, foram adotadas seis visões arquiteturais: Visão Lógica (*Logical View*), Visão de Processo (*Process View*), Visão Física (*Physical View*), Visão de Desenvolvimento (*Development View*), Visão de Caso de Uso (*Use Case View*) e Visão de Módulo (*Module View*). Além disso, são destacados os principais objetivos, *stakeholders* e interesses considerados durante o desenvolvimento da AR.

O Capítulo está organizado conforme as etapas definidas pelo ProSA-RA. Na Seção 3.1 são descritas as fontes de informação utilizadas na identificação dos Requisitos Arquiteturais da RADEG. A Seção 3.2 apresenta o conjunto de Requisitos Arquiteturais identificados a partir da análise dessas fontes. Na Seção 3.3 são detalhados o projeto da RADEG, suas seis visões arquiteturais e sua representação geral. A Seção 3.4 discute os resultados obtidos na avaliação da arquitetura proposta. Por fim, a Seção 3.5 apresenta as considerações finais do capítulo.

3.1 RA-1: Investigação das Fontes de Informação

No primeiro passo do ProSA-RA, denominado Investigação das Fontes de Informação, foram identificadas diversas fontes para a obtenção de informações sobre o domínio de JEDs voltados ao ensino de Computação. Nesse contexto, as fontes foram agrupadas em dois grupos principais: (i) JEDs para a Educação em Computação; e (ii) ARs para Jogos e Sistemas Educacionais. Conforme a singularidade da pesquisa, foram levantadas ARs de outros domínios. A seguir, são apresentadas as fontes de informação utilizadas,

destacando como cada uma contribuiu para o estabelecimento da AR proposta.

3.1.1 Grupo 1: JEDs para a Educação em Computação

Visando identificar os principais JEDs voltados para a Educação em Computação, foi conduzido um MSL. O estudo analisou estudos entre os anos de 1994 e 2024, utilizando como fontes a Biblioteca Digital da Sociedade Brasileira de Computação (SBC-OpenLib)⁴ e a base de dados Scopus⁵. Além disso, foi aplicada a técnica de *Snowballing* de duas formas para aumentar a quantidade e a qualidade dos estudos identificados: a busca na lista de referências dos estudos, chamada de *Backward Snowballing* (BS), e a busca nas citações dos estudos, *Forward Snowballing* (FS) (FELIZARDO et al., 2016).

Como resultado, foram selecionados 194 estudos, cuja análise permitiu a identificação de características, funcionalidades e requisitos relevantes para o desenvolvimento da AR proposta neste trabalho. A Figura 3.1 apresenta uma visão geral das etapas seguidas no MSL.

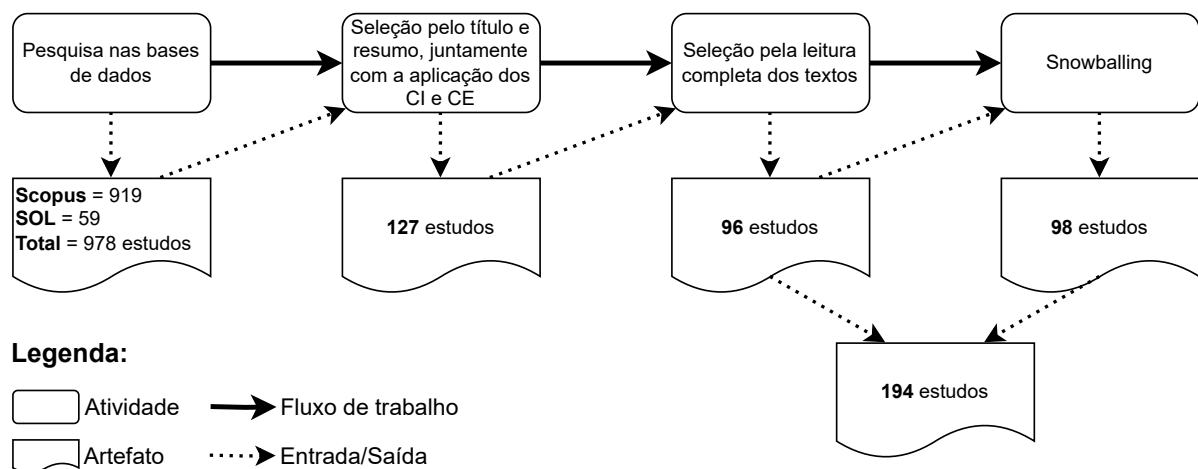


Figura 3.1: Processo de busca em quatro etapas para o MSL.

No MSL foram investigadas questões relacionadas aos nomes, áreas de aplicação, processos de desenvolvimento, *frameworks*, atributos de qualidade e métodos de avaliação dos JEDs. Os resultados demonstraram que as áreas de Lógica de Programação e Engenharia de Software concentram a maioria dos jogos analisados, seguidas por Estrutura de Dados e Redes de Computadores. Dentre os principais requisitos identificados no MSL,

⁴Disponível em: <https://sol.sbc.org.br/index.php/indice>

⁵Disponível em: <https://www.scopus.com>

destacam-se aprendizibilidade, jogabilidade, ludicidade, usabilidade, compreensibilidade, motivabilidade e interatividade. Esses elementos são considerados fundamentais para proporcionar uma experiência de ensino eficiente e agradável, contribuindo para a efetividade dos JEDs no processo educacional.

Além disso, verificou-se que ferramentas como Unity e Construct são amplamente utilizadas no desenvolvimento de JEDs, destacando-se por sua facilidade de uso e pelos recursos oferecidos. Contudo, foi identificada uma ausência significativa no uso de processos de desenvolvimento desses JEDs, contribuindo para a criação de jogos ineficazes para a aprendizagem de Educação em Computação.

Em relação à avaliação dos JEDs, observou-se que a maioria dos trabalhos se baseou em questionários, com uma aplicação de modelos sistemáticos. O resultado completo desse MSL pode ser consultado em Genesio et al. (2024).

3.1.2 Grupo 2: ARs para Jogos e Sistemas Educacionais

Para compilar os principais aspectos no desenvolvimento de JEDs para a Educação em Computação, foi realizada uma busca por ARs para esse domínio. No entanto, foi identificado que ainda não existem ARs específicas para apoiar JEDs voltados para a Educação em Computação. Assim, na Subseção 2.4, são apresentados alguns estudos que, embora não sejam diretamente focados em JEDs para a Educação em Computação, abordam conceitos, estruturas e soluções que podem servir de base para o desenvolvimento da RADEG.

Em suma, foram identificadas quatro ARs: **EDUCAR** (FIORAVANTI; NAKAGAWA; BARBOSA, 2010), **RAGE** (VEGT et al., 2016), **SORASG** (CARVALHO, 2017) e **URB** (MIZUTANI; KON, 2020). Essas ARs visam melhorar a eficiência, a reutilização de componentes e a flexibilidade no desenvolvimento de jogos, além de atenderem a requisitos específicos de cada domínio. Além disso, elas buscam reduzir custos e tempo de desenvolvimento, enquanto garantem a eficácia pedagógica e a flexibilidade na implementação de mecânicas de jogo. Dessa forma, essas abordagens oferecem alternativas mais eficientes e adaptáveis, facilitando a criação de ambientes e jogos que atendem a necessidades educacionais e de entretenimento de forma eficaz.

3.2 RA-2: Estabelecimento dos Requisitos Arquiteturais

Durante o passo RA-2 do ProSA-RA, denominado Estabelecimento de Requisitos Arquiteturais, as informações e os conceitos do domínio, obtidos a partir da investigação de fontes de informação, foram empregados para a eliciação dos requisitos da AR proposta. Essa etapa foi estruturada em duas tarefas principais: (i) Identificação dos Requisitos do Sistema; e (ii) Identificação dos Requisitos Arquiteturais. A seguir, cada uma dessas tarefas é apresentada.

3.2.1 Identificação dos Requisitos do Sistema

Nesta etapa, foram levantados os requisitos necessários para automatizar os processos, atividades e tarefas dos JEDs desenvolvidos com base na AR proposta. Assim, foram identificados 57 Requisitos de Sistema (RS) comuns aos JEDs, para apoiar a Educação em Computação. Vale ressaltar que alguns desses requisitos foram extraídos explicitamente dos estudos analisados, enquanto outros foram identificados de forma implícita, por meio da experiência prática com os jogos. A lista completa dos 57 RS identificados encontra-se no Apêndice A.

3.2.2 Identificação dos Requisitos Arquiteturais

Após a definição dos RS, cada requisito foi analisado em termos de relevância e viabilidade de implementação, resultando na identificação de um conjunto de Requisitos Arquiteturais para a AR proposta. Esse processo foi conduzido por meio de discussões entre especialistas e uma análise qualitativa, que avaliou a importância relativa de cada requisito. Nesse contexto, essa abordagem viabilizou uma tomada de decisão mais estruturada, garantindo alinhamento com os objetivos do sistema e as necessidades dos *stakeholders*. Como resultado, foram identificados 10 Requisitos Arquiteturais principais, os quais oferecem suporte ao desenvolvimento de JEDs voltados para a Educação em Computação, conforme detalhado a seguir.

- **RA-1:** A AR deve fornecer mecanismos para garantir que os conceitos apresentados

no jogo sejam aprendidos;

- **RA-2:** A AR deve ter tutoriais para garantir que o jogo seja fácil de jogar;
- **RA-3:** A AR deve permitir a utilização de recursos para que os elementos do jogo sejam lúdicos;
- **RA-4:** A AR deve garantir um bom desempenho para proporcionar uma experiência confortável e agradável aos jogadores;
- **RA-5:** A AR deve implementar medidas de segurança, robustez e políticas de privacidade para garantir que os dados dos jogadores estejam protegidos e que o ambiente de jogo seja seguro;
- **RA-6:** A AR deve ser bem documentada para facilitar a manutenção e futuras atualizações do jogo;
- **RA-7:** A AR deve viabilizar uma organização visual clara e eficaz, incluindo recursos de tela bem projetados;
- **RA-8:** A AR deve fornecer informações claras e compreensíveis para a apresentação das regras e funcionalidades do jogo;
- **RA-9:** A AR deve oferecer elementos interativos e mecânicas de jogo estimulantes;
- **RA-10:** A AR deve responder de forma rápida e significativa às ações do jogador para garantir uma jogabilidade interativa.

3.3 RA-3: Projeto da Arquitetura de Referência

No terceiro passo do ProSA-RA, denominado Projeto da Arquitetura de Referência, os Requisitos Arquiteturais elicitados são utilizados como base para o projeto da AR. Para documentar a RADEG de forma mais completa, são apresentados os objetivos, *stakeholders* e interesses (Seção 3.3.1), além da Visão Geral (*Overview*) (Seção 3.3.2). Além disso, seis visões arquiteturais também são apresentadas: Visão Lógica (*Logical View*) (Seção 3.3.3), Visão de Processo (*Process View*) (Seção 3.3.4), Visão Física (*Physical*

View) (Seção 3.3.5), Visão de Desenvolvimento (*Development View*) (Seção 3.3.6), Visão de Caso de Uso (*Use Case View*) (Seção 3.3.7) e Visão de Módulo (*Module View*) (Seção 3.3.8).

3.3.1 Objetivos, *Stakeholders* e Interesses

Os principais objetivos estabelecidos para a RADEG são contribuir para:

- O desenvolvimento de JEDs;
- Melhoria da qualidade, jogabilidade e interatividade dos JEDs;
- Melhor modularização (processos sistemáticos) dos JEDs;
- Contribuir para o reúso de elementos de software no contexto dos JEDs;
- A garantia de que os JEDs estejam sempre disponíveis;
- Melhor manutenibilidade de JEDs.

Com base nos Requisitos Arquiteturais e objetivos, citados anteriormente, foram definidos os *stakeholders*. São eles:

- **Estudantes:** principais usuários dos JEDs, que se beneficiam diretamente do aprendizado proporcionado pelos jogos;
- **Professores:** implementam os JEDs em sala de aula;
- **Instituições de Ensino:** escolas e universidades que permitem o uso dos JEDs para melhorar o processo de ensino-aprendizagem dos alunos;
- **Profissionais de Desenvolvimento de JEDs:** empresas ou pessoas que desenvolvem os JEDs.

A seguir, são apresentados os principais interesses identificados com base nos *stakeholders* analisados, os quais foram determinados por meio de discussões entre os membros do presente trabalho:

- **Aprendizibilidade:** espera-se que a RADEG facilite a criação de jogos que promovam a aprendizagem eficaz de conceitos de Computação;
- **Jogabilidade:** espera-se que a RADEG suporte uma variedade de mecânicas de jogo, permitindo flexibilidade na criação de experiências interativas e envolventes, essenciais para estimular o interesse e a curiosidade dos estudantes.
- **Ludicidade:** espera-se que a RADEG permita a criação de jogos que sejam divertidos e motivadores, promovendo o engajamento dos estudantes por meio de elementos lúdicos;
- **Usabilidade:** espera-se que a RADEG facilite o desenvolvimento de jogos com interfaces e controles intuitivos, permitindo que os jogadores e professores interajam com facilidade;
- **Segurança:** espera-se que a RADEG priorize a segurança dos dados dos usuários, especialmente informações de estudantes;
- **Manutenibilidade:** espera-se que a RADEG seja projetada para facilitar a manutenção dos jogos desenvolvidos, permitindo correções de *bugs*, atualizações e melhorias contínuas com um esforço reduzido.
- **Acessibilidade:** espera-se que a RADEG suporte a criação de jogos acessíveis para usuários com diferentes habilidades e limitações;
- **Reúso:** espera-se que a RADEG possibilite o reúso de informações e componentes em diferentes níveis, incluindo documentações, módulos e padrões;
- **Modularidade:** espera-se que a RADEG seja modular, permitindo a adição, remoção ou modificação de componentes de forma simples.

3.3.2 Visão Geral (*Overview*)

Com base na investigação de estilos e padrões arquiteturais, decidiu-se construir a RADEG com uma arquitetura Cliente-Servidor, pois esta oferece suporte a uma jogabilidade envolvente, atualizações em tempo real, escalabilidade e segurança, aprimorando assim

a experiência de aprendizagem dos jogadores. A Figura 3.2 apresenta a estrutura da RADEG, definida conforme o estilo e padrão arquitetural definidos.

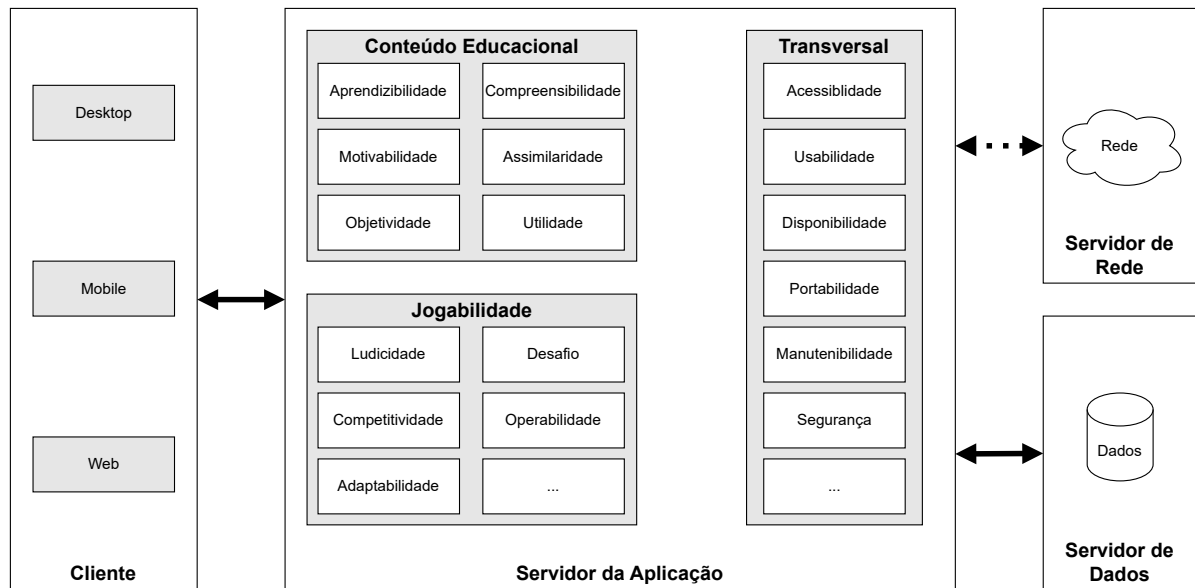


Figura 3.2: Visão Geral da RADEG.

No lado do **Cliente**, são apresentadas as diferentes plataformas nas quais o JED pode ser executado, como *Desktop*, *Mobile* ou *Web*. No **Servidor de Aplicação**, estão representados os módulos específicos do domínio educacional, detalhados a seguir. O **Conteúdo Educacional** refere-se a um conjunto independente de materiais educacionais projetados para ensino-aprendizagem, que integram diversos requisitos para otimizar a experiência de aprendizagem. Já a **Jogabilidade**, engloba os elementos interativos que envolvem os jogadores durante o aprendizado, determinando como o JED pode efetivamente transmitir o Conteúdo Educacional enquanto mantém o envolvimento e a diversão do jogador. As funcionalidades **Transversais** traduzem as funcionalidades e propriedades comuns ao sistema geral. Por fim, o **Servidor de Dados** gerencia o armazenamento e a recuperação de dados do JED, enquanto o **Servidor de Rede** cuida da conectividade à rede, caso o JED tenha funcionalidades on-line.

3.3.3 Visão Lógica (*Logical View*)

Como mencionado anteriormente, a Visão Lógica foca nos Requisitos Funcionais, ou seja, nos serviços que o sistema deve fornecer aos seus usuários. Para isso, o sistema é de-

composto em um conjunto de abstrações-chave, provenientes principalmente do domínio do problema, representadas na forma de objetos ou classes de objetos. Dessa maneira, a decomposição não se limita à análise funcional, mas também é utilizada para identificar mecanismos e elementos de *design* comuns nas diversas partes do sistema (KRUCHTEN, 1995). Para representar essa visão, foi elaborado um Diagrama de Classes, conforme apresentado na Figura 3.3, que exibe um conjunto de classes e seus relacionamentos lógicos.

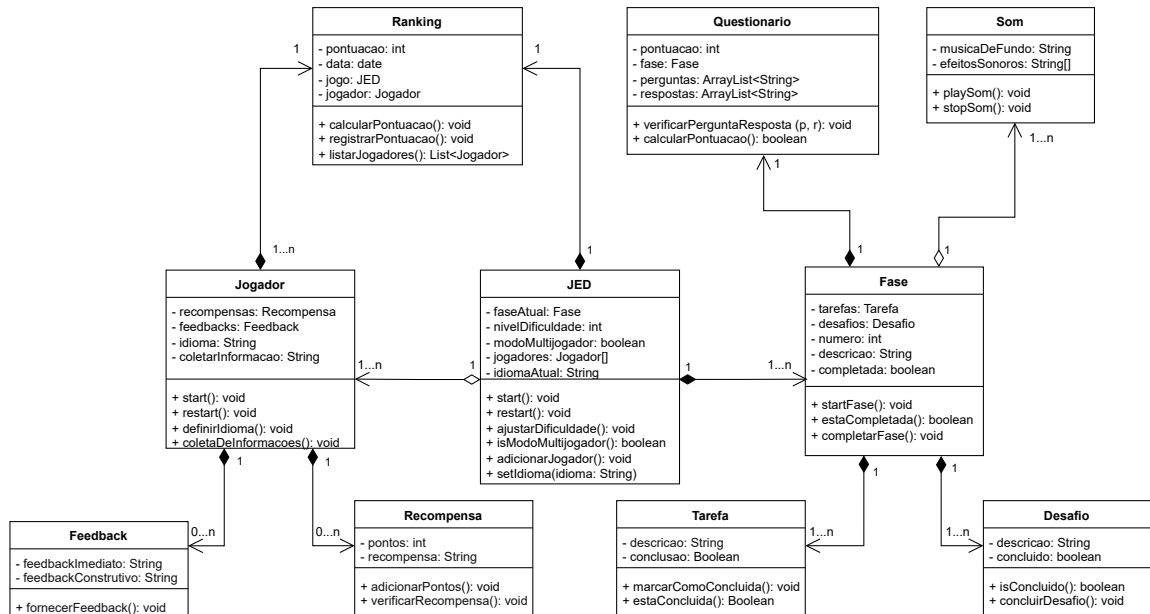


Figura 3.3: Diagrama de Classes da RADEG.

Para a elaboração do diagrama, foram identificadas, em particular, 13 funcionalidades comuns nos JEDs voltados para o apoio à Educação em Computação. Essas funcionalidades incluem:

- **Pontuação:** pode motivar os jogadores, criando uma sensação de realização e competição (*ranking*);
- **Feedback:** o JED deve fornecer *feedback* imediato e construtivo, permitindo que o jogador compreenda seus erros e identifique oportunidades de aprimoramento;
- **Tarefas:** tarefas claras, como tutoriais e dicas, devem orientar o processo de aprendizagem e auxiliar os jogadores a saber o que fazer em cada etapa;
- **Recompensas:** o JED deve ter um sistema de recompensas para motivar os jogadores a atingirem seus objetivos;

- **Personalização:** todas as informações importantes devem ser exibidas na tela em formato de texto, e o JED deve permitir tradução para diferentes idiomas;
- **Ferramentas de Aprendizagem Interativas:** elementos interativos, como questionários lúdicos, devem ser incluídos em cada fase;
- **Restart:** permite que os jogadores recomecem o JED pode ser útil para aprender com os erros e tentar novas estratégias;
- **Som:** incluir música de fundo e efeitos sonoros pode tornar o jogo mais envolvente, além de ser utilizado para fornecer *feedback* ou instruções ao usuário;
- **Aprendizagem Adaptativa:** o JED deve permitir que os usuários ajustem o nível de dificuldade de acordo com suas habilidades;
- **Desafios:** o JED deve incluir desafios para testar as habilidades dos jogadores;
- **Fases:** o JED deve ter fases ou níveis diferentes para proporcionar uma experiência de aprendizagem simulando um passo a passo;
- **Coleta de informações:** o JED deve permitir que os usuários forneçam *feedback* sobre a experiência de jogo;
- **Modo multijogador:** o JED deve ter um modo multijogador para promover o aprendizado colaborativo e o trabalho em equipe.

3.3.4 Visão de Processo (*Process View*)

A Visão de Processo considera aspectos relacionados aos Requisitos Não-Funcionais, como desempenho e disponibilidade do sistema, explorando os processos do sistema, como eles se comunicam e concentrando-se no comportamento em tempo de execução (KRUCHTEN, 1995). Além disso, esses processos representam o nível no qual essa visão pode ser controlada taticamente (iniciada, recuperada, reconfigurada, desligada e assim por diante). Com isso, os processos podem ser replicados para distribuir a carga de processamento ou melhorar a disponibilidade do sistema (KRUCHTEN, 1995).

Para representar essa visão, um Diagrama de Atividades foi construído, conforme mostrado na Figura 3.4. Nesse contexto, esse Diagrama ilustra o fluxo de execução de um JED e as interações entre o jogador e o sistema.

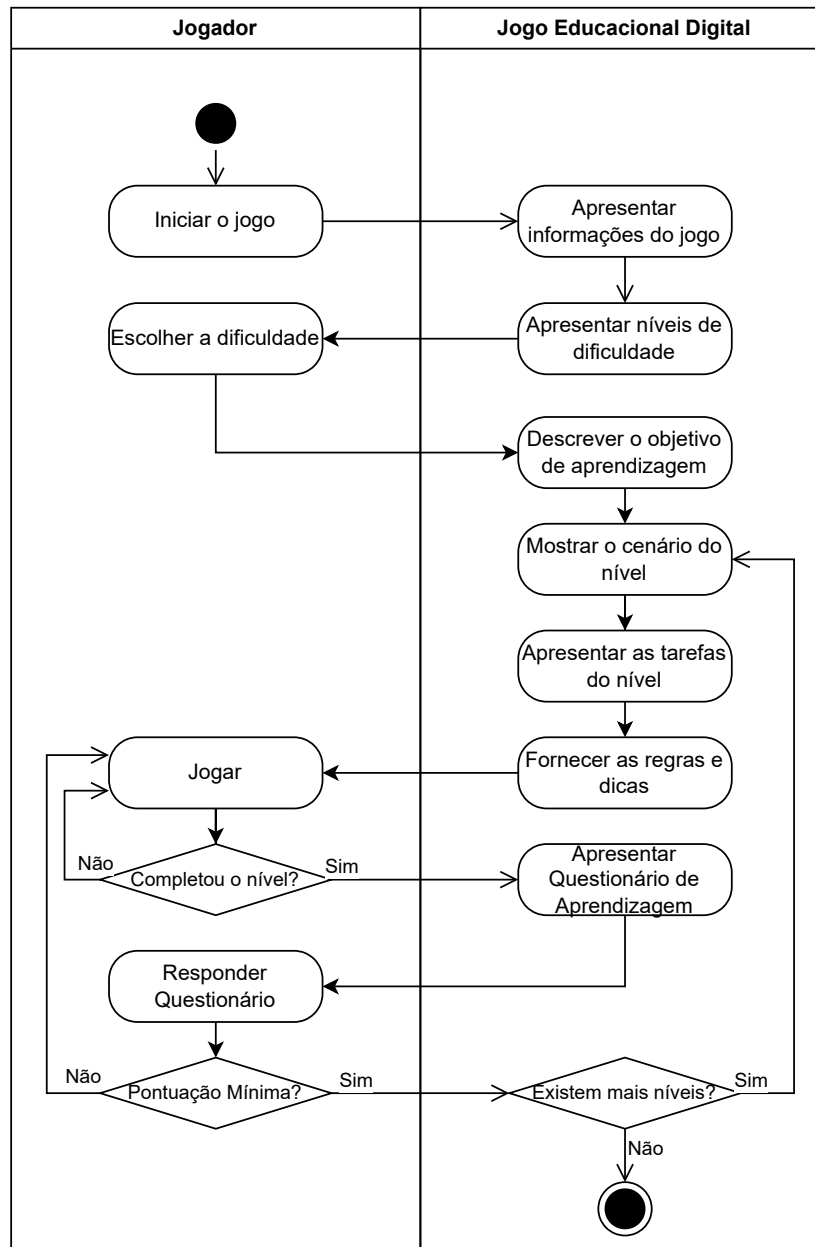


Figura 3.4: Diagrama de Atividades da RADEG.

O processo começa com o jogador clicando em Iniciar, seguido pela apresentação das informações básicas (nome do JED, créditos e configurações de som e idioma) e opções de dificuldade. Após escolher o nível de dificuldade, o JED explica os objetivos educacionais e mostra o cenário correspondente, onde o jogador deve completar tarefas baseadas em instruções claras. Durante a execução do nível, o jogador realiza tarefas

específicas dentro do ambiente do JED, baseadas em regras e dicas fornecidas pelo sistema. Ao concluir o nível, um questionário de avaliação é apresentado para avaliar o aprendizado adquirido. Caso o jogador alcance a pontuação mínima necessária, ele pode avançar para o próximo nível; caso contrário, o sistema permite a repetição do atual. Ao concluir, o jogador pode escolher um novo nível de dificuldade ou sair do JED.

3.3.5 Visão Física (*Physical View*)

A Visão Física descreve o mapeamento do software no hardware e reflete seu aspecto distribuído, levando em consideração alguns RNFs do sistema, como disponibilidade do sistema, confiabilidade (tolerância a falhas), desempenho (rendimento) e escalabilidade (KRUCHTEN, 1995). Para representar essa visão, um Diagrama de Implantação foi construído, como ilustrado na Figura 3.5. Nesse diagrama, é possível visualizar a arquitetura de um JED distribuído entre dispositivos clientes e servidores.

Os dispositivos clientes representam diferentes plataformas nas quais o JED pode ser executado, como *desktops*, *smartphones* ou navegadores Web. Cada dispositivo cliente utiliza o componente Jogo Educacional Digital para acessar os serviços fornecidos pelo servidor. O servidor, por sua vez, é o núcleo central responsável pelo processamento e execução das funcionalidades do JED. Ele é composto por três principais componentes: Servidor do JED (responsável pelo processamento central do jogo educacional digital), Servidor da Rede (gerencia a comunicação e conexão entre os dispositivos clientes e o servidor) e Banco de Dados (armazena os dados do jogo, incluindo progresso dos jogadores, pontuações e outros dados persistentes).

Dessa forma, os dispositivos clientes executam o JED e enviam solicitações ao servidor central. O servidor do JED processa essas solicitações, acessa ou atualiza os dados necessários no banco de dados e retorna respostas adequadas aos dispositivos clientes. Essa estrutura modular permite que o JED seja executado em diversas plataformas, oferecendo uma experiência consistente.

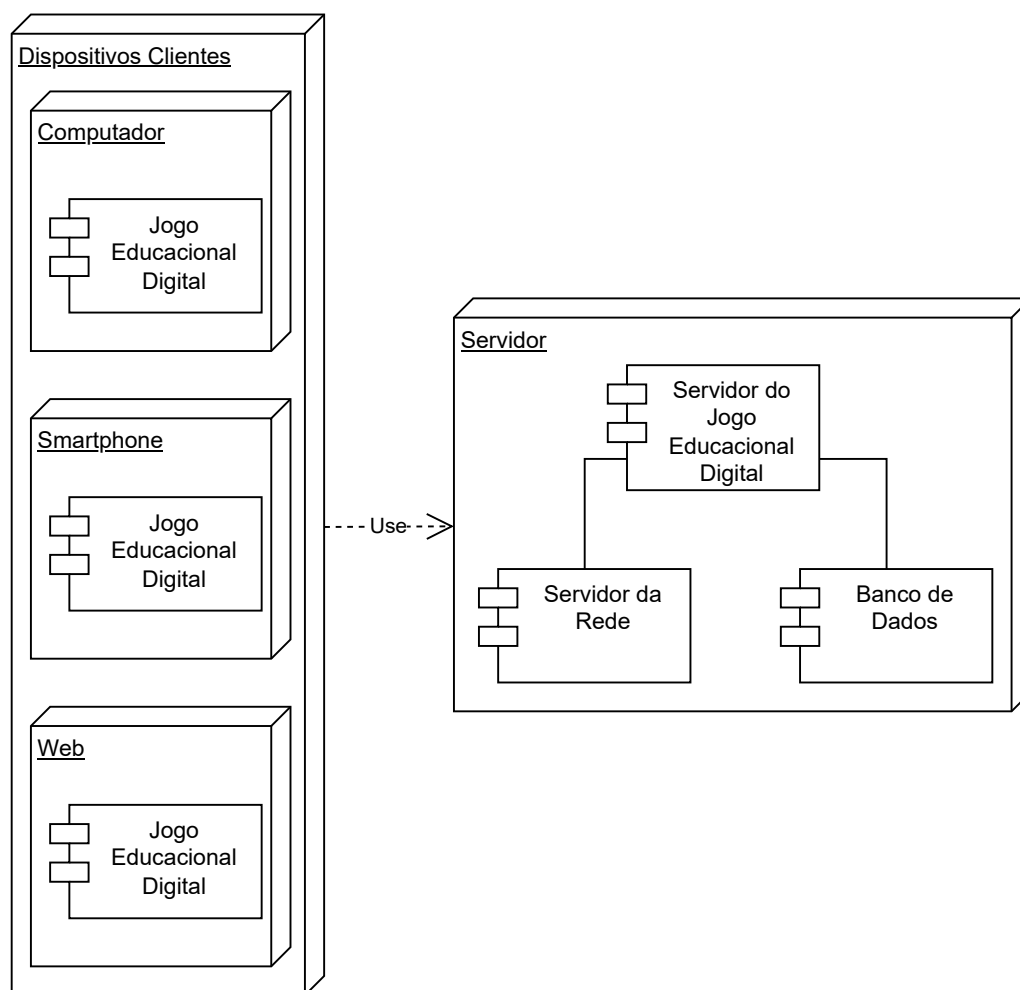


Figura 3.5: Diagrama de Implantação da RADEG.

3.3.6 Visão de Desenvolvimento (*Development View*)

A Visão de Desenvolvimento foca na organização dos módulos de software reais no ambiente de desenvolvimento de software. Nesse contexto, o software é modularizado em componentes menores, como subsistemas, permitindo que esses elementos sejam desenvolvidos de forma independente por equipes ou desenvolvedores distintos. Assim, esses subsistemas são organizados de maneira hierárquica, com as camadas superiores interagindo com as inferiores por meio de interfaces bem definidas e restritas, promovendo encapsulamento, reutilização e facilidade de manutenção (KRUCHTEN, 1995). Para representar essa visão, foi construído um Diagrama de Pacotes com 5 sub pacotes principais, como representado na Figura 3.6.

O pacote JED é o principal, que contém a lógica central do jogo. Ele é responsável por coordenar as atividades e o fluxo de dados entre os diferentes pacotes do sistema,

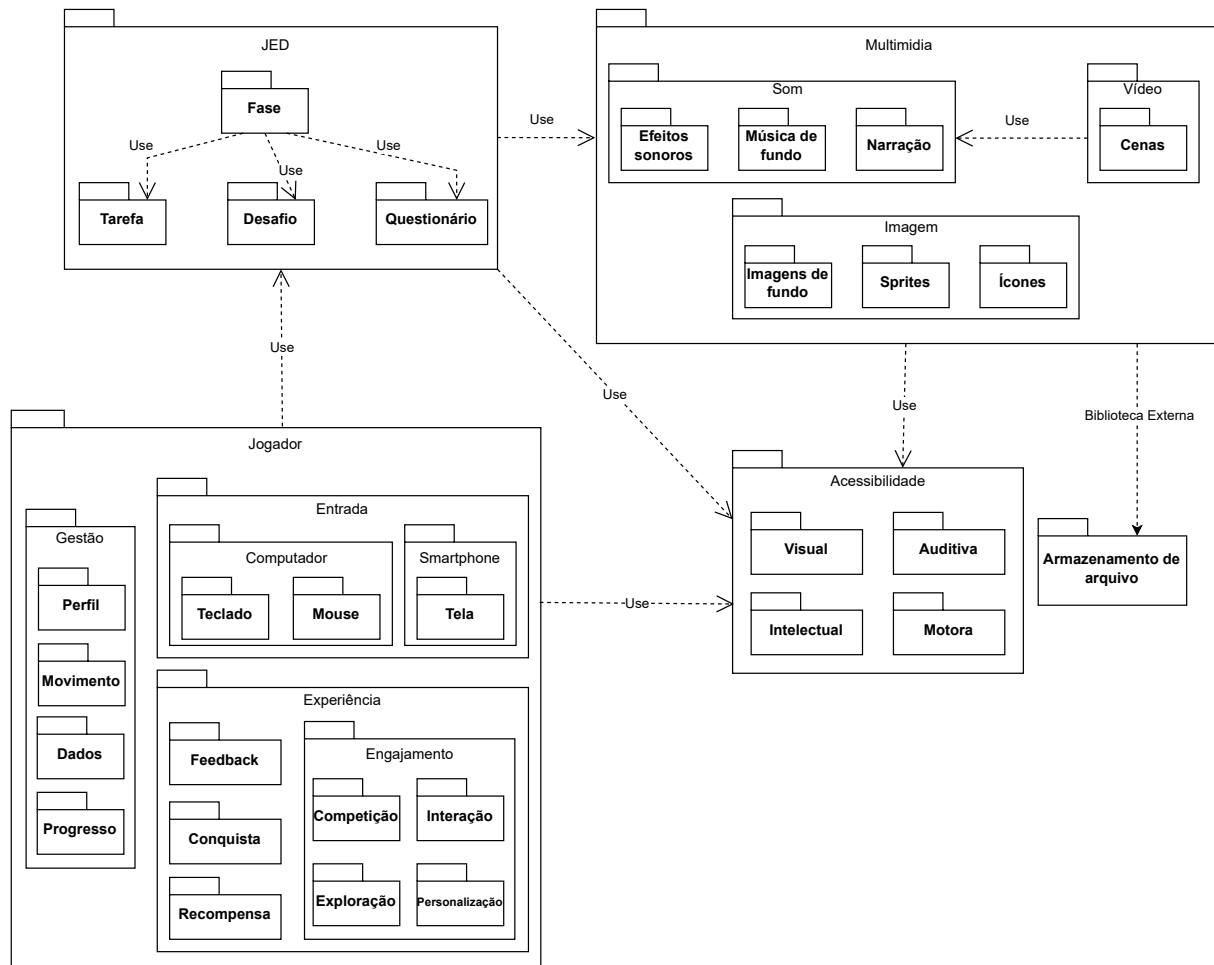


Figura 3.6: Diagrama de Pacotes da RADEG.

como Jogador, Multimídia e Acessibilidade, garantindo a execução eficiente do JED. Por sua vez, o pacote Jogador gerencia todas as funcionalidades relacionadas à experiência do usuário no JED. Dentro deste pacote, encontram-se sub pacotes que gerenciam as funcionalidades do JED e as configurações do sistema, como dispositivos de entrada, além de também gerenciar as experiências do jogador.

Já o pacote Multimídia é essencial para a imersão do jogador, gerenciando os elementos visuais e sonoros, além de organizar os recursos que compõem a interface gráfica. Por meio de sub pacotes como Som e Imagem, o JED pode reproduzir efeitos sonoros, música de fundo e narrações que criam um ambiente mais envolvente. Da mesma forma, a exibição de imagens de fundo, *sprites* e ícones contribui para a construção de um ambiente gráfico atrativo, impactando diretamente na experiência do usuário. A presença de vídeos, organizados dentro do sub pacote Vídeo, complementa o jogo com a reprodução de cenas que podem ser utilizadas para ilustrar histórias ou ensinar conceitos.

Para garantir que as informações de multimídia sejam mostradas ao longo do JED, o pacote Armazenamento de arquivo é utilizado. Ele gerencia o progresso no JED, preferências e configurações para mostrar os arquivos para o jogador. Por fim, o pacote Acessibilidade garante que o jogo seja acessível a todos os tipos de jogadores, incluindo aqueles com deficiências visuais, auditivas, motoras e intelectuais. Esse pacote oferece funcionalidades que adaptam o jogo para atender às necessidades específicas de cada jogador, proporcionando a todos uma experiência positiva dentro do JED.

3.3.7 Visão de Caso de Uso (*Use Case View*)

A Visão de Caso de Uso utiliza um subconjunto selecionado de cenários importantes, ou seja, instâncias de casos de uso, para demonstrar a integração perfeita dos elementos nas quatro visualizações anteriores (KRUCHTEN, 1995). Para ilustrar isso, foi empregado o Diagrama de Caso de Uso, conforme exemplificado na Figura 3.7.

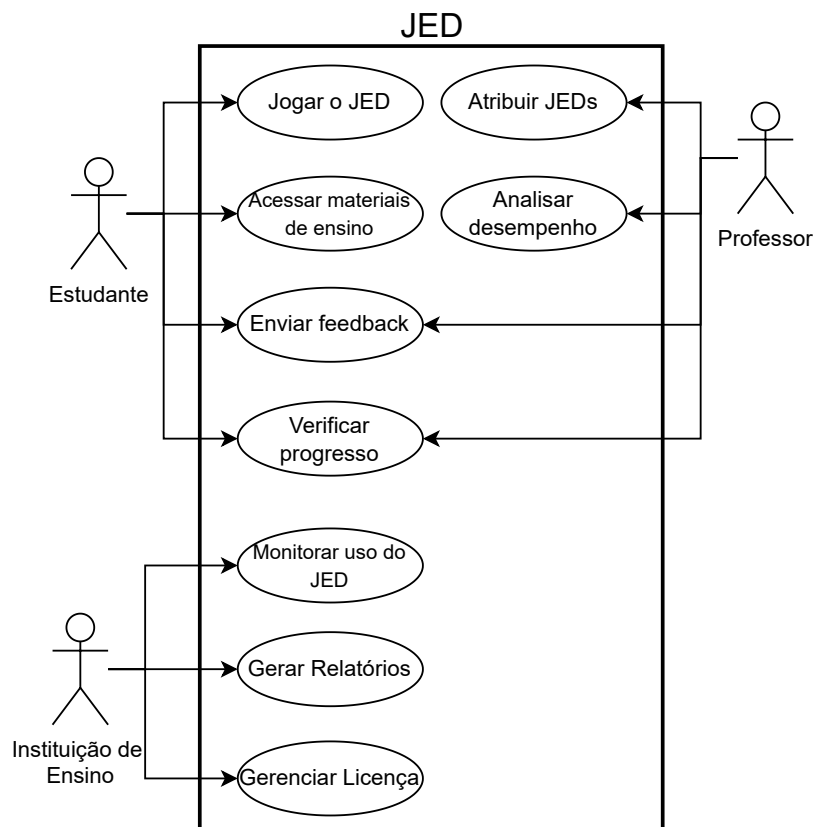


Figura 3.7: Diagrama de Caso de Uso da RADEG.

Esse Diagrama de Caso de Uso ilustra as interações entre os atores principais - Estudante, Professor e Instituição de Ensino - e o sistema JED.

O Estudante é o usuário central do sistema, sendo sua principal função jogar o JED. A interação do estudante com o sistema envolve realizar uma série de atividades, como acessar materiais de ensino, enviar *feedbacks* e verificar seu progresso no JED. O Professor, por outro lado, desempenha um papel de apoio e supervisão dentro do sistema. Uma de suas principais funções é atribuir JEDs aos estudantes, ou seja, selecionar e distribuir JEDs adequados às necessidades de sua turma. Além disso, o professor também tem a capacidade de verificar o progresso de cada estudante, acompanhando o desenvolvimento de suas habilidades ao longo do jogo. Com base nessa análise, o professor pode fornecer *feedback* personalizado, ajudando os alunos a melhorar seu desempenho.

Por fim, a Instituição de Ensino também desempenha um papel importante no gerenciamento e manutenção do sistema. Ela é responsável por gerenciar as licenças de uso do JED, garantindo que o acesso esteja disponível para professores e estudantes. Além disso, a instituição monitora o uso do JED, acompanhando como o jogo está sendo utilizado e garantindo que ele esteja alinhado aos seus objetivos educacionais. Além disso, a instituição também pode gerar relatórios, que permitem que ela tome decisões sobre a continuidade ou ajustes na implementação do jogo. Em resumo, o Estudante, como jogador, participa das atividades de aprendizagem, o Professor, como controlador, orienta e monitora esse aprendizado, e a Instituição de Ensino, como administradora, garante que o sistema funcione de forma eficaz.

3.3.8 Visão de Módulo (*Module View*)

A Visão de Módulo identifica as principais unidades de implementação, ou módulos, de um sistema, juntamente com as relações entre essas unidades (BASS; CLEMENTS; KAZMAN, 2012). Isso ajuda a determinar como alterações em uma parte do sistema podem impactar outras partes, influenciando a capacidade do sistema de suportar modificabilidade, portabilidade e reutilização (BASS; CLEMENTS; KAZMAN, 2012).

Para representar essa visão, foi criado um Diagrama de Componentes que ilustra a organização e as relações entre os componentes dentro do JED (Figura 3.8). Esses componentes trabalham em conjunto para criar a experiência geral do JED. A seguir, estão os 9 módulos identificados por meio das fontes de informações:

- **Acompanhamento:** acompanha o progresso do jogador no JED, fornecendo *feedbacks* para melhorar a experiência;
- **Mecânica do Jogo:** trata da mecânica central do JED, abrangendo controles e regras;
- **Educacional:** responsável por apresentar conteúdo educacional ao jogador, como questionários e tarefas;
- **Dados:** responsável por armazenar e recuperar as informações geradas ao longo do uso do JED, como perfis de jogadores e pontuações;
- **Interface do Usuário:** responsável pela camada de apresentação do JED, incluindo *design* das telas, menus, botões, placares e outros elementos visuais que impactam na usabilidade;
- **Acessibilidade:** responsável por recursos que tornam o JED acessível a todos os tipos de jogadores, inclusive pessoas com deficiências (PcD);
- **Rede:** responsável pela rede, como conexão a um servidor, caso o JED tenha funcionalidades on-line;
- **Enredo, Sons e Animações:** gerencia a narrativa do JED, incluindo enredo, diálogos e desenvolvimento do personagem, além de controlar os efeitos sonoros e visuais do JED, melhorando a experiência geral do jogador;
- **Segurança:** responsável por proteger os dados dos jogadores e garantir que o JED seja utilizado de maneira segura, implementando medidas de proteção contra acessos não autorizados e garantindo que o sistema esteja protegido contra ataques externos.

3.4 RA-4: Avaliação da Arquitetura de Referência

No quarto passo do ProSA-RA, denominado Avaliação da Arquitetura de Referência, são conduzidas análises destinadas a evidenciar a qualidade da AR proposta (NAKAGAWA et al., 2014). Para atingir esse objetivo, utilizam-se inspeções por meio de *checklists*

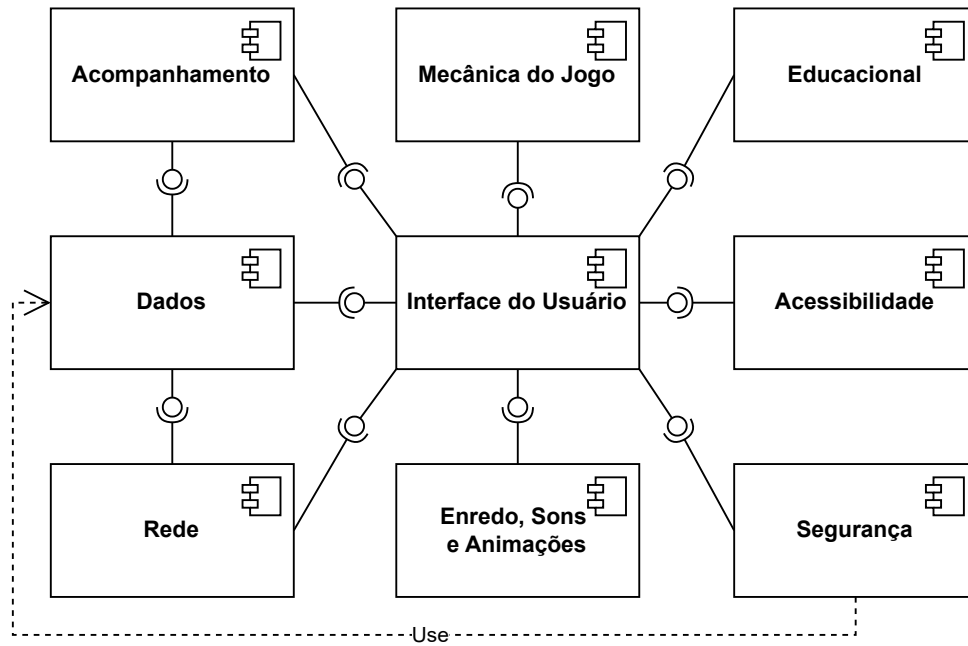


Figura 3.8: Diagrama de Componentes da RADEG.

e métodos de avaliação arquitetural. No contexto deste trabalho, adotou-se o *checklist* FERA (*Framework for Evaluation of Reference Architectures*) (SANTOS et al., 2013), e o método ATAM (*Architecture Tradeoff Analysis Method*) (KAZMAN et al., 1998), como abordagens principais para a avaliação. No Capítulo 4, são apresentados os detalhes desse processo de avaliação.

3.5 Considerações Finais

Dada a importância dos JEDs para a Educação em Computação, este capítulo apresentou a RADEG, uma AR projetada para apoiar o desenvolvimento desses jogos. Por meio da RADEG, buscou-se fornecer uma sistematização que facilite o desenvolvimento de JEDs, promovendo a integração do entretenimento e elementos educacionais para garantir a eficácia desses JEDs no apoio à Educação em Computação.

Além disso, o desenvolvimento de uma AR representa um desafio significativo, especialmente em áreas recentes, onde ainda não existem arquiteturas concretas e maduras. Nesse sentido, é essencial que o processo de elaboração de uma AR contemple etapas bem definidas, desde a fase inicial com o levantamento de trabalhos relacionados, a análise de normas técnicas da área e o estudo de sistemas já desenvolvidos, até a realização de uma

avaliação arquitetural. Nesse contexto, o ProSA-RA desempenhou um papel fundamental para o estabelecimento da RADEG, fornecendo etapas bem definidas.

Por fim, considerando a necessidade de validar a RADEG, o próximo capítulo apresenta uma análise baseada no *checklist* FERA e no método ATAM.

4 Avaliação da RADEG

Este capítulo apresenta o *checklist* FERA e o método ATAM, ambos utilizados para avaliar arquiteturas de sistemas. O FERA é composto por 93 questões de múltipla escolha, cujas respostas estão na escala *likert*, as quais variam entre “totalmente satisfatório” e “totalmente insatisfatório”, além de campos para comentários que orientam os avaliadores na identificação de problemas nos documentos de representação da AR (SANTOS et al., 2013). Já o ATAM visa determinar se a arquitetura atende às necessidades esperadas, identificando *trade-offs* entre atributos, refinando requisitos e promovendo a comunicação entre *stakeholders* (KAZMAN et al., 1998).

A estrutura do Capítulo está organizada da seguinte forma: A Seção 4.1 detalha a avaliação preliminar realizada com o FERA. A Seção 4.2 apresenta uma breve introdução do ATAM, juntamente com suas etapas. O restante das seções estão organizadas conforme as etapas definidas pelo ATAM. Na Seção 4.3 são descritos os principais detalhes da avaliação. A Seção 4.4 apresenta a definição de objetivos de negócio, apresentação da arquitetura, identificação de abordagens arquiteturais, criação e priorização de atributos de qualidade, e análise de riscos e *trade-offs*. Na Seção 4.5 são detalhados a priorização e análise dos cenários pelos *stakeholders*, além da apresentação dos resultados. A Seção 4.6 discute a elaboração do relatório final. Na Seção 4.7, são apresentadas as ameaças à validade. Por fim, a Seção 4.8 apresenta as considerações finais do capítulo.

4.1 Avaliação Preliminar

A avaliação inicial da RADEG foi conduzida por meio do *checklist* FERA. A partir do uso do FERA foi possível validar se a AR: (i) está corretamente representada, fornecendo informações gerais como riscos potenciais, restrições e escopo; (ii) contém um conjunto adequado de pontos de vista, perspectivas e modelos arquiteturais; (iii) tem em sua documentação, informações importantes, como decisões arquitetônicas, melhores práticas e orientações, políticas e regras, normas internacionais e interfaces entre módulos; (iv)

considera atributos de qualidade importantes para seu domínio; (v) pode ser facilmente instanciada; e (vi) pode ser alterada, se necessário, para melhorar a sua documentação.

Nesse contexto, a avaliação da RADEG foi conduzida utilizando um conjunto de 23 perguntas do FERA, selecionadas por sua relevância para o contexto deste trabalho, conforme ilustrado na Tabela 4.1. As opções de resposta foram adaptadas para sim (a documentação está totalmente em conformidade com o critério avaliado pela pergunta), não (a documentação não está em conformidade com o critério avaliado pela pergunta) e parcialmente (a documentação está parcialmente em conformidade com o critério avaliado pela pergunta), permitindo uma análise mais detalhada.

Tabela 4.1: *Checklist* FERA para avaliação de AR (adaptado de Santos et al. 2013)

ID	Tópico	Pergunta
P1	Construção e Conteúdo	O atributo Aprendizibilidade foi considerado?
P2	Construção e Conteúdo	O atributo Jogabilidade foi considerado?
P3	Construção e Conteúdo	O atributo Ludicidade foi considerado?
P4	Construção e Conteúdo	O atributo Usabilidade foi considerado?
P5	Construção e Conteúdo	O atributo Segurança foi considerado?
P6	Construção e Conteúdo	O atributo Manutenibilidade foi considerado?
P7	Construção e Conteúdo	O atributo Acessibilidade foi considerado?
P8	Construção e Conteúdo	O atributo Compreensibilidade foi considerado?
P9	Construção e Conteúdo	O atributo Diversão foi considerado?
P10	Construção e Conteúdo	O atributo Interatividade foi considerado?
P11	Discussão	A AR explana claramente quem são os <i>stakeholders</i> e seus interesses?
P12	Discussão	A AR explana claramente os <i>stakeholders</i> em potencial de uma arquitetura instanciada?
P13	Discussão	Os <i>viewpoints</i> gerados abordam os interesses de todos os <i>stakeholders</i> ?
P14	Discussão	Os <i>viewpoints</i> gerados abordam os interesses de <i>stakeholders</i> em potencial de uma arquitetura instanciada?
P15	Discussão	A AR está consistente com as práticas do domínio e os padrões mandatórios?
P16	Discussão	Os <i>viewpoints</i> incluem interesses que não são específicos de domínio, de interesses de <i>stakeholders</i> ?
P17	Discussão	Para cada <i>viewpoint</i> , os modelos estão claramente definidos? Os modelos provêm informação para determinar se os interesses foram alcançados?
P18	Discussão	Os objetivos do domínio estão claramente articulados e priorizados?
P19	Discussão	Há alguma forma de rastreabilidade entre os objetivos do domínio e os requisitos levantados?
P20	Discussão	Há alguma forma de rastreabilidade entre os objetivos do domínio e soluções técnicas?
P21	Discussão	Que critérios são usados para determinar se a AR suporta os objetivos do domínio?
P22	Discussão	É possível visualizar as instâncias da AR?
P23	Discussão	Os riscos de introduzir esta AR estão claramente documentados?

Na avaliação da RADEG utilizando o FERA, participaram oito especialistas, todos ES com conhecimentos sólidos no domínio de JEDs e também AS. Diante disso,

após a condução da avaliação utilizando o FERA, todas as respostas dos avaliadores foram categorizadas, registradas e compiladas, conforme ilustrado na Tabela 4.2.

Tabela 4.2: Respostas das Perguntas da *checklist* FERA

Resultados	Construção e Conteúdo	Discussão da AR		Total Itens
		Stakeholders	Especialistas do Domínio	
Ambos Sim	P1, P2, P3, P4, P6, P7, P8, P10	P11 a P17	P18, P21 e P22	18
Ambos Parcial				
Sim e Parcial	P9		P19 e P23	3
Ambos Não				
Não e outra resp.	P5 (Não, Parcialmente)		P20 (Sim, Não)	2

Para garantir a análise qualitativa das respostas e responder adequadamente às observações feitas, foi utilizado a *Grounded Theory* como método de análise (STRAUSS; CORBIN, 1994). Esse método é utilizado para transformar e abstrair dados em uma teoria, explorando e analisando continuamente as informações, garantindo que as teorias desenvolvidas permaneçam fundamentadas na realidade (STRAUSS; CORBIN, 1994).

Nesse contexto, o processo de análise seguiu as três etapas principais descritas por Strauss e Corbin (1994). Na codificação aberta, os dados são segmentados e categorizados, gerando conceitos iniciais e categorias emergentes. Na codificação axial, essas categorias são refinadas e conectadas a subcategorias, criando um entendimento mais detalhado das relações entre elas. Finalmente, na codificação seletiva, as categorias centrais são integradas, resultando em um modelo teórico que sintetiza as principais descobertas e oferece uma explicação coerente para os padrões observados (STRAUSS; CORBIN, 1994).

Na primeira fase da análise, conhecida como codificação aberta, todas as respostas dos especialistas foram examinadas em detalhes para identificar categorias e padrões iniciais. As respostas foram organizadas em dois tópicos principais: Construção e Conteúdo e Discussão da AR facilitando a análise das semelhanças e diferenças entre elas. No tópico de Construção e Conteúdo, surgiram categorias como a clareza da documentação, a consideração de requisitos de diversão e segurança, e a completude das visões apresentadas.

A análise revelou que a maioria dos especialistas considerou a documentação da RADEG clara e detalhada. No entanto, foram identificadas algumas lacunas, especialmente em relação à explicitação dos requisitos de segurança e diversão, que foram

abordados de forma implícita e sem o detalhamento necessário, gerando questionamentos entre os especialistas. Observou-se que, embora a RADEG tenha tratado de elementos lúdicos, o requisito de diversão não estava claramente representado nas visões da AR. Da mesma forma, a segurança dos dados foi mencionada, mas faltavam informações precisas sobre as medidas para garantir a proteção dos dados e prevenir acessos não autorizados. Apesar disso, os demais requisitos foram plenamente compreendidos e aceitos pelos participantes da avaliação. No tópico de Discussão da AR, as categorias emergentes incluíram a clareza na identificação dos *stakeholders* e seus interesses, porém, os especialistas relataram dificuldades em visualizar a rastreabilidade entre os objetivos do domínio e as soluções técnicas propostas pela arquitetura.

Na segunda fase, a codificação axial, as categorias emergentes da fase anterior foram relacionadas entre si, proporcionando uma compreensão mais aprofundada dos dados. Essa análise revelou que, para que a RADEG seja considerada completa, é fundamental que os aspectos lúdicos, como a diversão, sejam tratados com maior clareza e destaque. Além disso, embora a segurança dos dados seja reconhecida como importante, sua ausência como um requisito explícito na RADEG pode gerar incertezas quanto à proteção dos dados dos usuários. Essa lacuna indica que a segurança precisa ser abordada de forma mais visível e explícita, tanto na documentação quanto nas visões arquiteturais. A dificuldade em visualizar a rastreabilidade entre os objetivos dos *stakeholders* e as soluções técnicas também levantou questões sobre a adequação da AR para atender aos objetivos específicos do domínio. Isso sugere que um mapeamento direto entre os objetivos do domínio e suas soluções técnicas seria valioso para reforçar o alinhamento da AR com os interesses dos *stakeholders*.

Na última fase, a codificação seletiva, as categorias foram refinadas em torno de duas teorias principais. A primeira teoria emergente é que, para que a RADEG atenda de forma eficaz às expectativas dos *stakeholders* e às exigências do domínio de JEDs, é essencial que os requisitos de diversão e segurança sejam explicitamente tratados na documentação. A ausência de uma abordagem clara para esses requisitos afeta tanto a aceitação da AR pelos especialistas quanto sua capacidade de ser instanciada de forma eficaz. A segunda teoria sugere que a inclusão de uma visão que mapeie diretamente

os objetivos do domínio para as soluções técnicas, poderia melhorar a rastreabilidade e a compreensão da RADEG. Isso contribuiria para garantir que as soluções propostas estejam claramente alinhadas com os objetivos educacionais e lúdicos do domínio.

Apesar da RADEG ter sido amplamente aceita em termos de clareza e organização, a análise mais profunda das respostas revelou que questões cruciais, como a explicitação dos requisitos de segurança e diversão, bem como a rastreabilidade dos objetivos, precisam ser abordadas de forma mais clara. O uso do método *Grounded Theory* permitiu uma avaliação detalhada, destacando a importância de explicitar requisitos essenciais na documentação e garantir que as soluções técnicas propostas estejam diretamente alinhadas com os objetivos dos *stakeholders* e do domínio.

Além disso, a RADEG apresentou uma base sólida para o desenvolvimento de JEDs, sendo amplamente aceita pelos especialistas. No entanto, a avaliação sugere ajustes necessários para melhorar a documentação. Esses ajustes são fundamentais para garantir que a RADEG possa ser efetivamente instanciada e sustentada ao longo do tempo, promovendo o desenvolvimento de JEDs de alta qualidade.

A análise quantitativa revelou que, em 18 das 23 perguntas, todos os oito especialistas responderam “Sim”, indicando uma aceitação ampla da AR em relação aos atributos avaliados. Essas perguntas incluem P1, P2, P3, P4, P6, P7, P8, P10, P11, P12, P13, P17, P15, P16, P17, P18, P19 e P22, sugerindo que a AR foi bem-sucedida ao atender 78% dos critérios analisados. A concordância completa nessas perguntas reflete a confiança dos especialistas na capacidade da AR de proporcionar uma experiência de uso eficiente e agradável.

Algumas perguntas, entretanto, apresentaram respostas mais diversificadas, revelando áreas onde a AR pode necessitar de ajustes. P5 destacou-se por ter a avaliação mais crítica, com 6 especialistas respondendo “Não” e apenas 2 respondendo “Parcialmente”. Isso aponta para uma preocupação generalizada sobre a capacidade da AR de lidar adequadamente com aspectos de segurança, o que deve ser considerado uma área de melhoria prioritária. Outra área que apresentou discordância foi o requisito de diversão (P9), no qual 4 especialistas responderam “Sim” e 4 responderam “Parcialmente”, sugerindo que o elemento lúdico da AR não está sendo completamente percebido como parte essencial.

Essa divergência sugere que, embora metade dos especialistas tenha reconhecido a diversão como um atributo positivo e bem implementado, a outra metade não percebeu essa característica de forma tão evidente, apontando para a necessidade de a AR reforçar melhor o requisito de diversão.

Além disso, P19 obteve 7 respostas “Sim” e 1 resposta “Parcialmente”, o que sugere que, apesar do alto nível de aceitação, um especialista identificou a necessidade de melhorias na rastreabilidade entre os objetivos do domínio e os requisitos levantados. Nesse contexto, embora a maioria tenha considerado que a AR atende bem a esse critério, é necessário realizar ajustes para garantir que a rastreabilidade esteja completamente clara e bem documentada. Da mesma forma, P23, com 6 respostas “Sim” e 2 “Parcialmente”, indica que, embora a documentação da AR tenha sido bem avaliada pela maioria, ainda há espaço para aprimoramento. O fato de dois especialistas considerarem a documentação apenas parcialmente satisfatória ressalta a importância de revisar e melhorar a clareza da dos riscos na documentação.

Por fim, P20 recebeu 7 respostas “Sim” e 1 resposta “Não”, indicando que embora a grande maioria dos especialistas tenha concordado que a AR apresenta uma rastreabilidade adequada entre os objetivos do domínio e as soluções técnicas, existe a necessidade de uma revisão e potencial melhoria na forma como essa rastreabilidade é documentada e demonstrada. Desse modo, é necessário ter uma visão mais transparente de como cada solução técnica mapeia diretamente para os objetivos estratégicos do domínio.

4.2 *Architecture Tradeoff Analysis Method (ATAM)*

O método de avaliação ATAM requer a participação colaborativa de três grupos principais: a equipe de avaliação, os tomadores de decisão e os *stakeholders*. A equipe de avaliação é composta por especialistas externos ao projeto cuja arquitetura está sendo analisada. Normalmente, essa equipe é formada por três a cinco profissionais selecionados com base em seu conhecimento em AS, garantindo uma avaliação qualificada. Os tomadores de decisão, por sua vez, são aqueles que possuem autoridade para representar o projeto de desenvolvimento, sendo responsáveis por definir diretrizes e, se necessário, implementar mudanças na arquitetura. Por fim, os *stakeholders* englobam todas as pessoas que têm

interesse no sucesso do sistema e, para a avaliação, recomenda-se a participação de um grupo de aproximadamente 12 a 15 pessoas (BASS; CLEMENTS; KAZMAN, 2012).

O ATAM divide-se em quatro áreas principais: (1) coleta de cenários e requisitos, (2) análise de visões arquitetônicas e realização de cenários, (3) construção e avaliação de modelos, e (4) identificação de *trade-offs*. Essas quatro áreas são divididas em nove etapas, como ilustra a Figura 4.1. Inicialmente, cenários e requisitos são definidos, e uma arquitetura inicial é proposta. Em seguida, cada atributo de qualidade é analisado separadamente quanto ao impacto no *design*. Na etapa crítica, pontos de compensação entre atributos são identificados, permitindo ajustes na arquitetura, nos modelos ou nos requisitos. Esse ciclo iterativo promove melhorias contínuas, garantindo que a arquitetura atenda às necessidades do sistema de forma eficaz (KAZMAN et al., 1998).

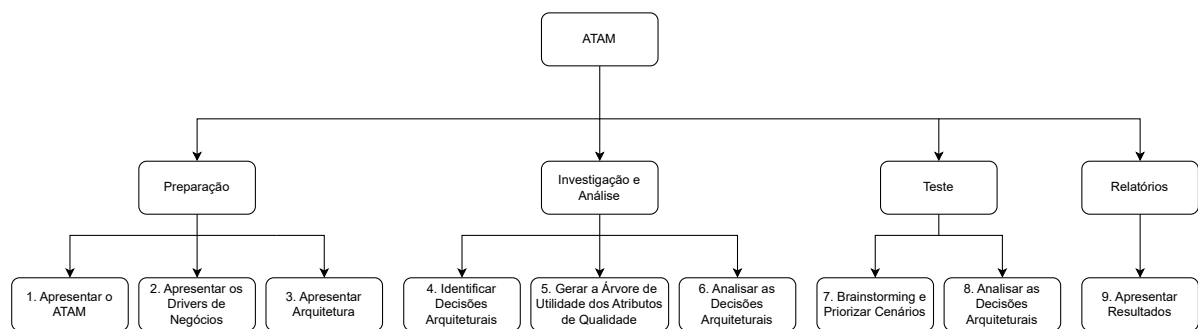


Figura 4.1: Etapas do ATAM (Adaptado de Kazman et al., 1998).

4.3 Fase 0: Parceria e preparação

Nessa fase, os tomadores de decisão selecionam uma equipe de avaliação composta por profissionais com experiência adequada para a análise da RADEG. Posteriormente, a equipe de avaliação e os tomadores de decisão realizam reuniões para definir os detalhes e a logística do processo de avaliação (BASS; CLEMENTS; KAZMAN, 2012).

Inicialmente, foram identificados alguns participantes, e convites foram enviados via e-mail. Como resultado, 3 professores, cada um com mais de 6 anos de experiência na área de modelagem de software, aceitaram participar da avaliação. Em seguida, foram agendados os horários das avaliações com cada um dos participantes, e o propósito da avaliação foi explicado.

4.4 Fase 1: Avaliação (inicial)

Nessa fase, a equipe de avaliação se reuniu com os tomadores de decisão para iniciar o processo de coleta e análise de informações. Esse processo foi dividido em seis etapas, descritas nas subseções a seguir.

4.4.1 Etapa 1: Preparação do ATAM

Inicialmente, foi realizada uma apresentação⁶ para introduzir o ATAM à equipe de avaliação e aos *stakeholders*. Durante essa etapa, foram apresentadas as fases do ATAM, bem como o processo metodológico que seria seguido ao longo da avaliação.

4.4.2 Etapa 2: Apresentação do *Business Driver*

Nessa etapa, foram reunidas as informações apresentadas na Seção 2, proporcionando uma visão geral do contexto em que a RADEG foi desenvolvida. Dessa forma, foram apresentados⁶ os principais conceitos relacionados a RADEG, para contextualizar tanto a equipe de avaliação quanto os *stakeholders*.

4.4.3 Etapa 3: Apresentação da Arquitetura

Na presente etapa, um dos membros da equipe de tomadores de decisão realizou uma apresentação⁶ da RADEG, com ênfase em como ela atende às diretrizes de negócios estabelecidas. A apresentação abordou os 10 Requisitos Arquiteturais da RADEG (conforme descrito na Subseção 3.2.2), além dos principais objetivos definidos para a RADEG, os *stakeholders* e os principais interesses identificados. Também foram apresentadas as seis visões arquiteturais e a visão geral da modelagem da RADEG (conforme descrito na Seção 3.3).

Durante a avaliação, a equipe de avaliação solicitou esclarecimentos e fez anotações sobre as táticas ou padrões arquitetônicos empregados, registrando observações que contribuíram para a análise da RADEG.

⁶Disponível em: <https://encurtador.com.br/Trr9H>

4.4.4 Etapa 4: Identificação das Abordagens Arquitetônicas

Após a apresentação da etapa anterior, nesta breve etapa, a equipe de avaliação identificou as abordagens arquiteturais consideradas para a RADEG. Essas abordagens foram identificadas com base nas informações apresentadas, sendo registradas e analisadas pela equipe para fornecer uma base para as etapas subsequentes da avaliação.

4.4.5 Etapa 5: Geração da Árvore de Atributos de Qualidade

Nesta etapa, os atributos de qualidade são priorizados em detalhes por meio de uma Árvore de Atributos de Qualidade. As Árvores de Atributos de Qualidade têm como função identificar e priorizar os objetivos dos atributos de qualidade mais importantes para o sistema (BASS; CLEMENTS; KAZMAN, 2012).

Os Requisitos Arquiteturais (ou atributos de qualidade) da RADEG foram previamente priorizados na Subseção 3.2.2, e, a partir dessa priorização, foi construída a Árvore de Atributo de Utilidade, conforme ilustrado na Figura 4.2. Essa estrutura ajudou a organizar e representar visualmente as prioridades dos atributos de qualidade, facilitando a análise e a tomada de decisões sobre a RADEG.

A Árvore de Atributos de Qualidade, apresentada na Figura 4.2, possui **qualidade** como nó raiz, pois esse é o objetivo central da RADEG: garantir a qualidade de um JED. Os nós diretamente abaixo da raiz representam os atributos de qualidade desejados no sistema, ou seja, os atributos priorizados para atender aos objetivos da RADEG.

Abaixo desses atributos de qualidade, encontram-se os itens específicos desejados para cada atributo. Em seguida, descreve-se como cada item deve ser implementado no JED. Vale ressaltar que, entre o item e a descrição da sua implementação, é indicada a importância e complexidade do item, classificadas em uma escala de alta (A), média (M) ou baixa (B). Para determinar essas classificações, foram realizadas análises e discussões, garantindo que a importância e a complexidade de cada cenário fossem avaliadas de forma precisa e alinhadas aos objetivos da RADEG.

A partir da priorização dos atributos de qualidade, foram analisados os cenários de caso de uso, observando a ocorrência dos atributos de qualidade e sua importância no contexto do comportamento e dos resultados esperados no JED. Com isso, foram criados

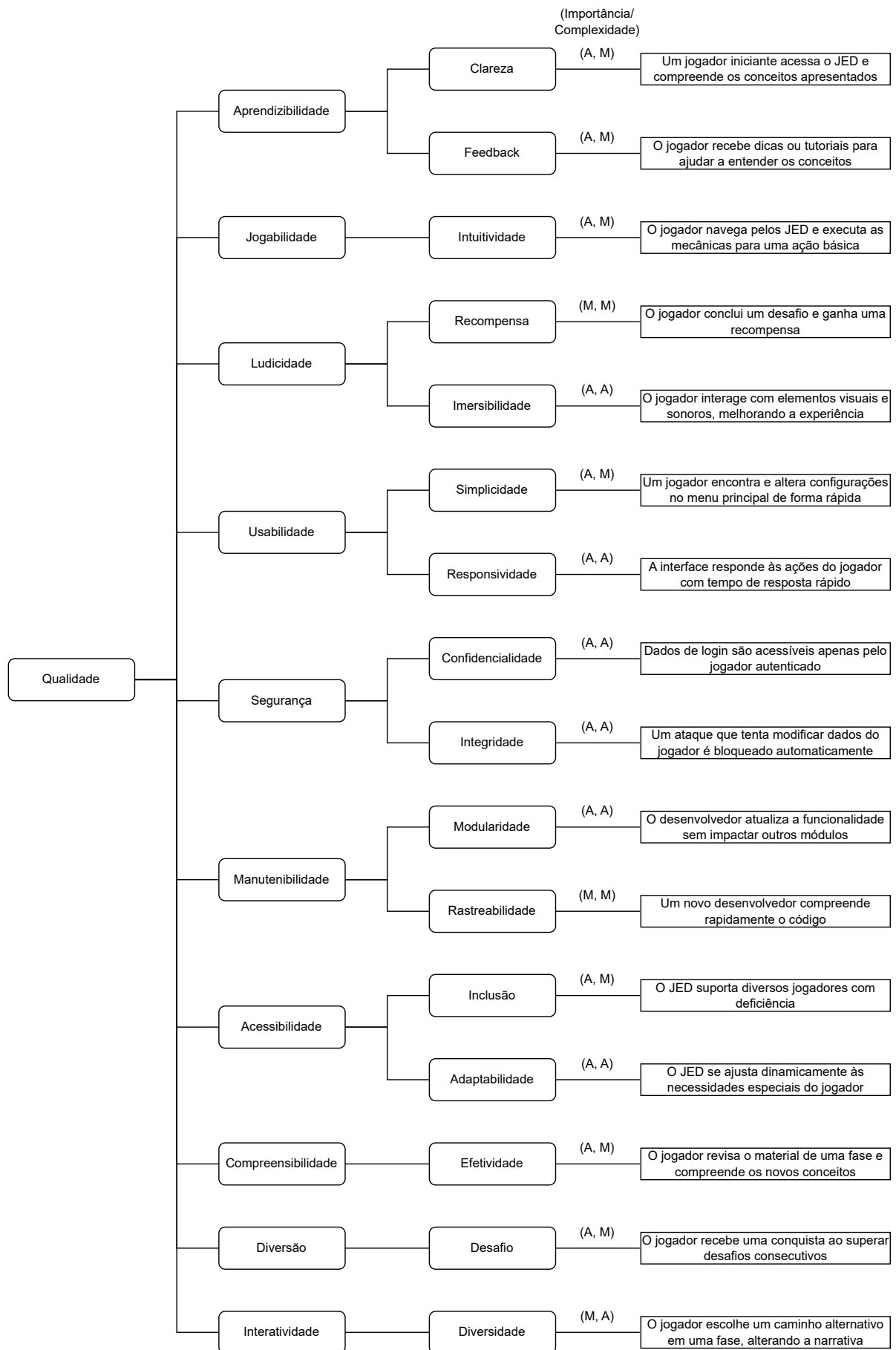


Figura 4.2: Árvore de Atributos de Qualidade.

dez cenários diferentes, um para cada atributo, apresentados a seguir.

Para especificar um requisito de atributo de qualidade, é possível formalizá-lo em seis partes, como ilustrado na Figura 4.3. A primeira parte, **Fonte de Estímulo**, refere-se a algum atuador que gerou o estímulo. A segunda parte, **Estímulo**, é uma condição que exige uma resposta quando chega ao sistema. Em seguida, o **Ambiente** especifica o contexto ou modo em que o sistema está sendo executado. O **Artefato**, por sua vez, é o elemento estimulado, podendo ser uma coleção de sistemas, o sistema completo ou uma parte dele. A **Resposta** é a atividade realizada como resultado da chegada do estímulo. Por fim, a última etapa, a **Medida de Resposta**, estabelece que, quando a resposta ocorre, ela deve ser mensurável de alguma forma, permitindo que o requisito seja testado e validado adequadamente (BASS; CLEMENTS; KAZMAN, 2012).

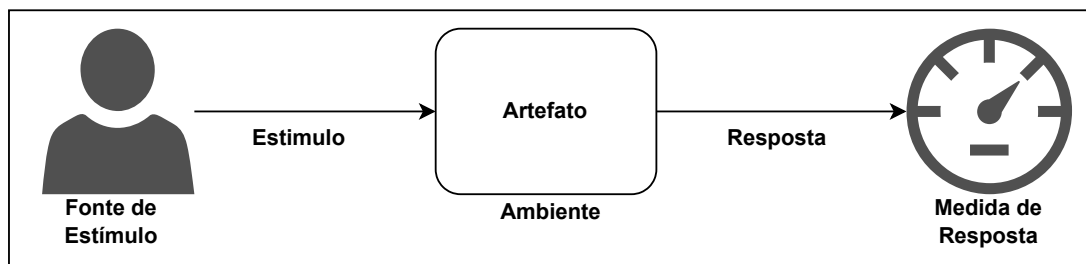


Figura 4.3: Partes de um Cenário de Atributo de Qualidade (Adaptado de Bass, Clements e Kazman, 2012).

O primeiro cenário, **Aprendizibilidade** (Figura 4.4), avalia como os mecanismos da RADEG garantem a aprendizagem de conceitos educacionais.

O segundo cenário, **Jogabilidade**, ilustrado na Figura 4.5, visa avaliar a eficácia da RADEG em garantir que o JED seja fácil de usar e acessível a todos os usuários.

Por sua vez, o terceiro cenário, **Ludicidade**, conforme ilustrado na Figura 4.6, destaca a importância dos elementos lúdicos para aumentar o engajamento e a retenção do aprendizado no contexto dos JEDs.

O quarto cenário, **Usabilidade**, apresentado na Figura 4.7, assegura que a RADEG proporcione uma experiência agradável desde a interação inicial com o JED, garantindo sua facilidade de uso.

Já o quinto cenário, **Segurança** (Figura 4.8), assegura que o ambiente do JED seja seguro para o jogador, protegendo seus dados pessoais e evitando acessos não auto-

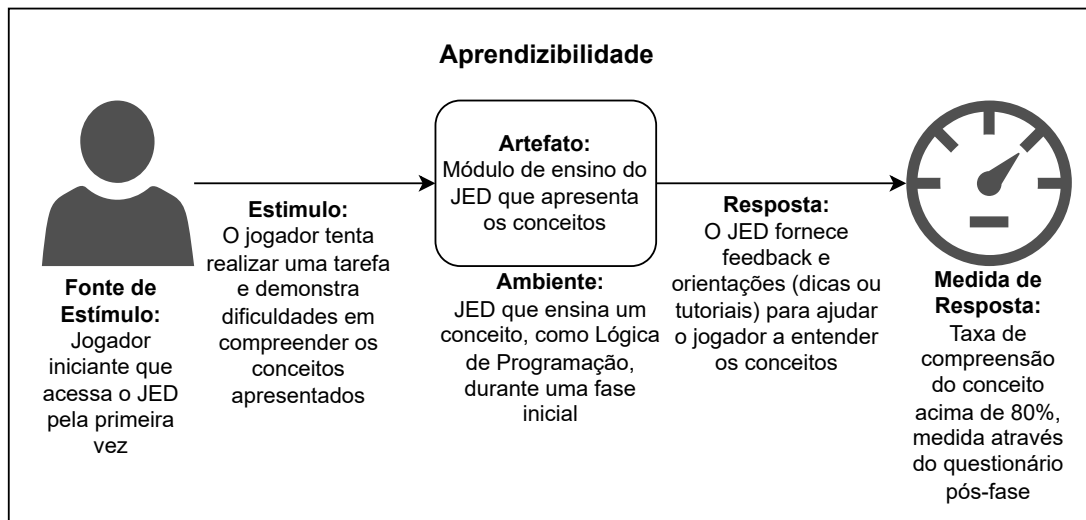


Figura 4.4: Um cenário geral para Aprendizizibilidade.

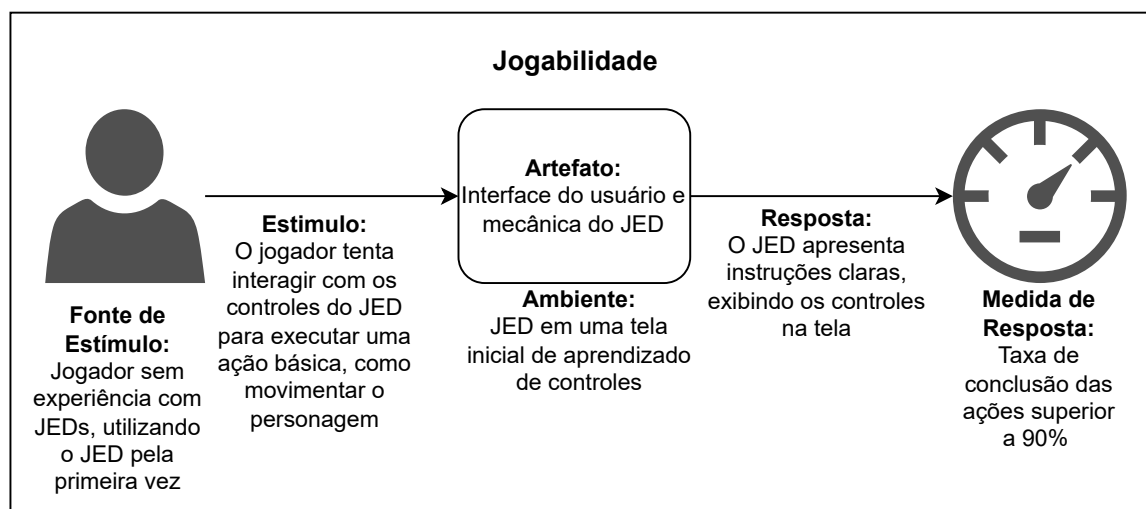


Figura 4.5: Um cenário geral para Jogabilidade.

rizados.

Por sua vez, o sexto cenário, **Manutenibilidade**, conforme ilustrado na Figura 4.9, reforça que a RADEG facilita a manutenção do JED, permitindo a inclusão de novas funcionalidades de forma eficiente.

O sétimo cenário, **Acessibilidade**, apresentado na Figura 4.10, avalia como a RADEG apoia a acessibilidade, especialmente no contexto de jogadores com deficiência motora, garantindo que o jogo seja inclusivo.

Já o oitavo cenário, **Compreensibilidade** (Figura 4.11), se concentra em como o jogador pode compreender e integrar conceitos durante o JED, facilitando o aprendizado.

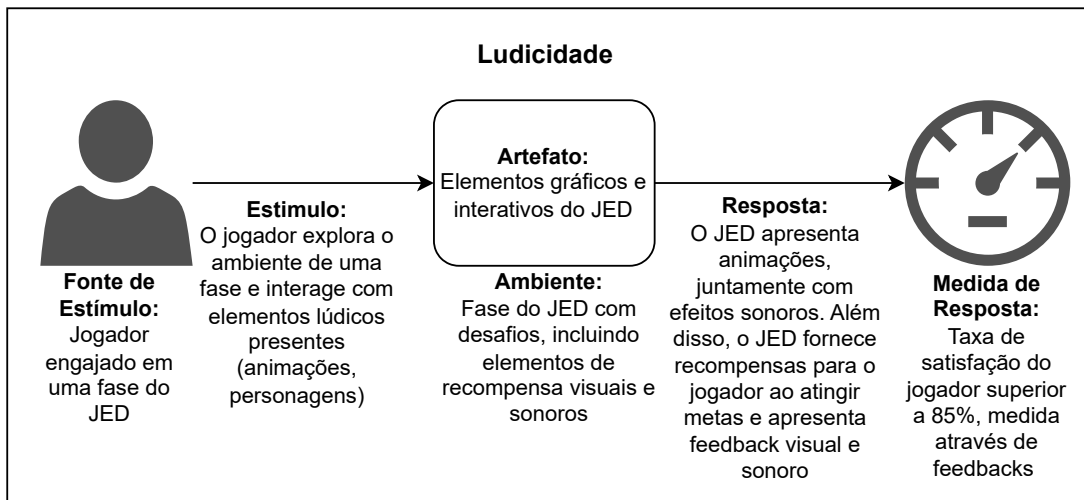


Figura 4.6: Um cenário geral para Ludicidade.

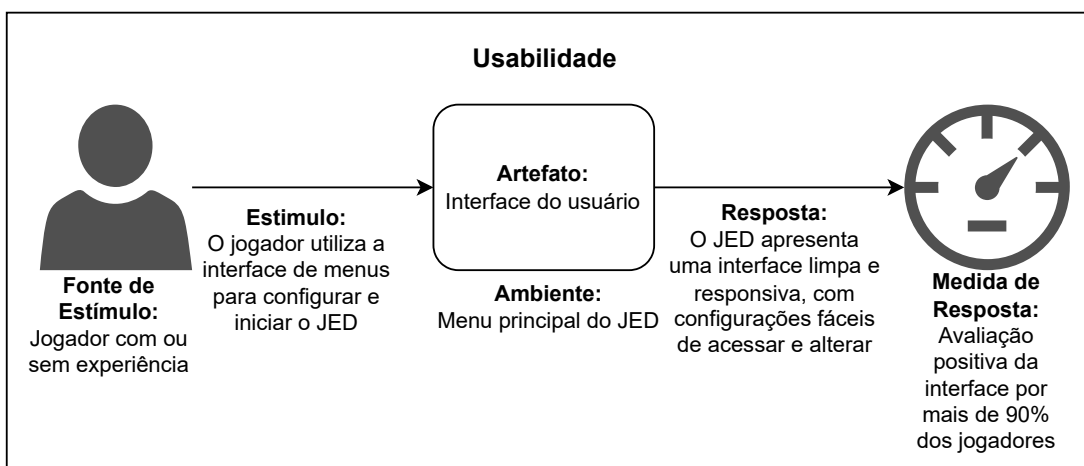


Figura 4.7: Um cenário geral para Usabilidade.

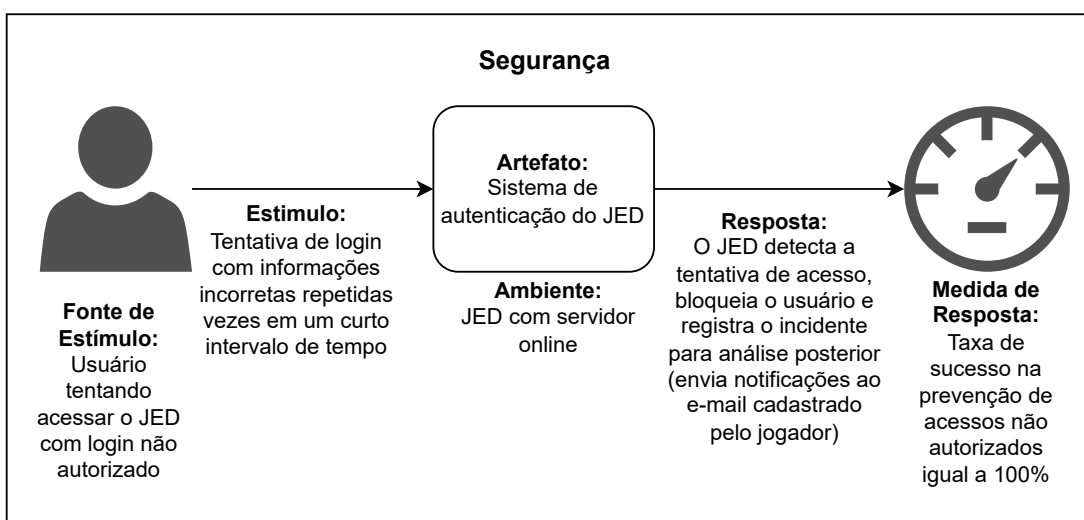


Figura 4.8: Um cenário geral para Segurança.

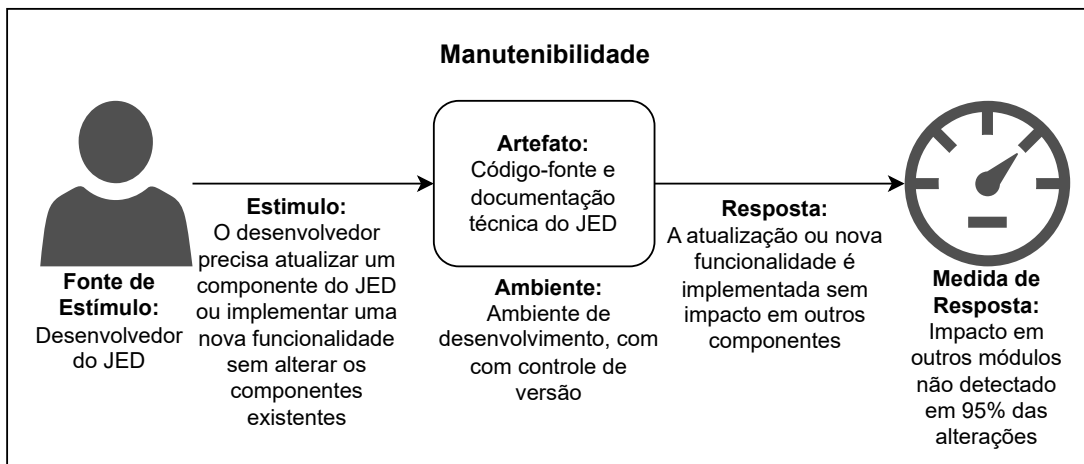


Figura 4.9: Um cenário geral para Manutenibilidade.

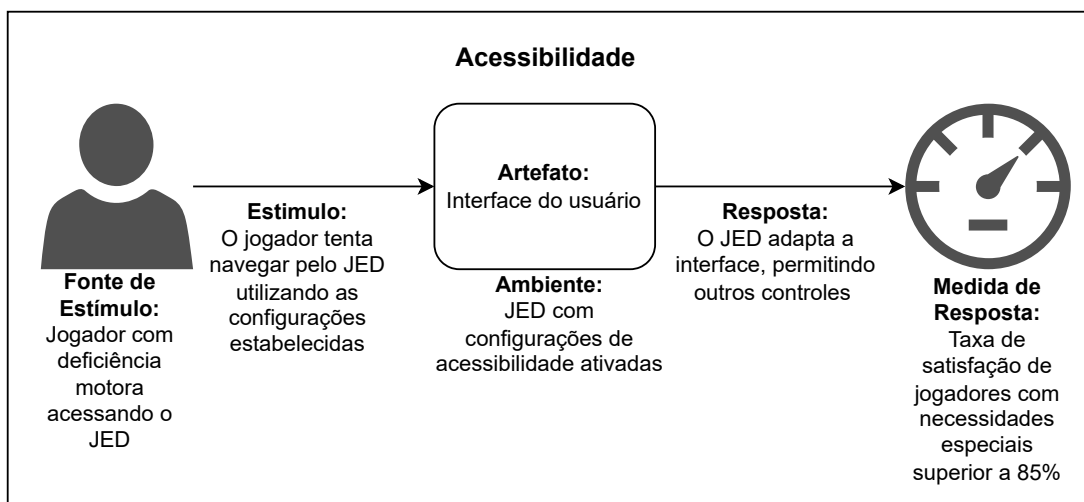


Figura 4.10: Um cenário geral para Acessibilidade.

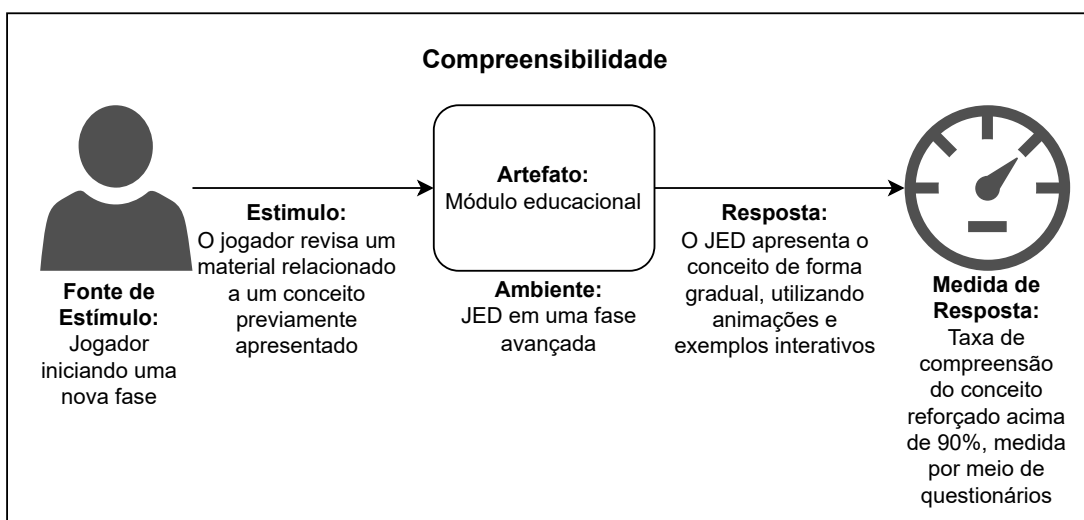


Figura 4.11: Um cenário geral para Compreensibilidade.

O nono cenário, **Diversão**, ilustrado na Figura 4.12, assegura que o JED seja envolvente, incorporando elementos gráficos e interativos para manter o aluno motivado e focado no aprendizado.

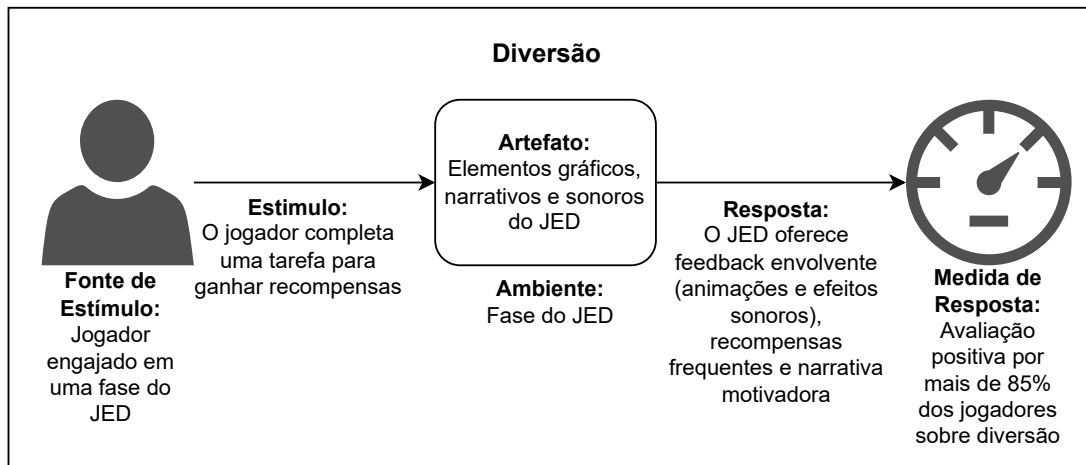


Figura 4.12: Um cenário geral para Diversão.

Por fim, o décimo cenário, **Interatividade**, conforme ilustrado na Figura 4.13, garante que o JED permita ao jogador interagir com os conteúdos e tomar decisões, promovendo uma experiência mais dinâmica e participativa.

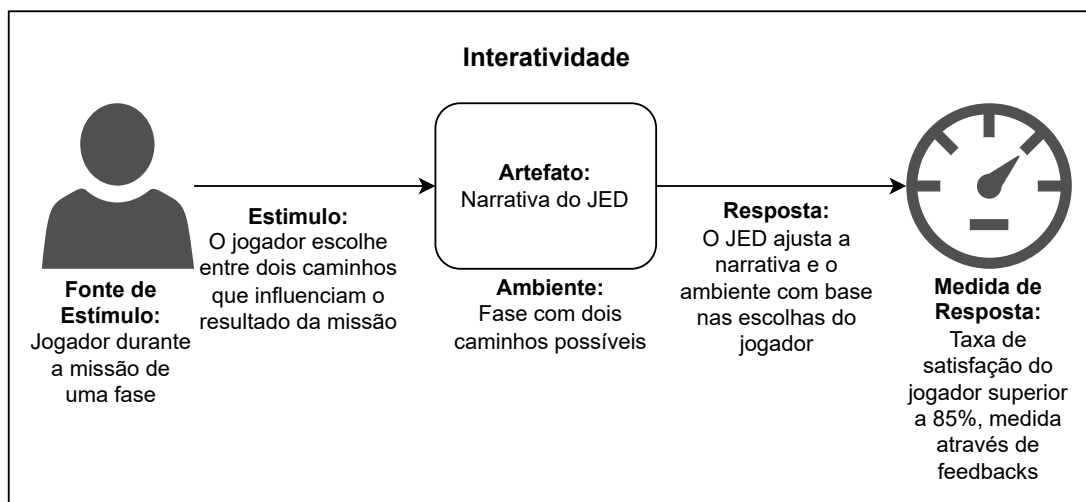


Figura 4.13: Um cenário geral para Interatividade.

4.4.6 Etapa 6: Análise das Abordagens Arquiteturais

Nessa etapa, a equipe de avaliação analisou individualmente os cenários mais bem classificados, conforme identificados na árvore de utilidades, para verificar se os atributos foram

considerados na RADEG, assim como os riscos, não riscos, pontos de sensibilidade e *trade-offs* associados. Ao final desse processo, a equipe obteve uma visão clara dos elementos mais relevantes da RADEG, bem como uma lista estruturada dos riscos e compensações envolvidos.

A avaliação ocorreu imediatamente após as apresentações descritas nas etapas anteriores. Durante esse processo, os três professores avaliadores (identificados como PA1, PA2 e PA3) preencheram um formulário específico⁷, no qual registraram suas observações quanto à consideração dos atributos na RADEG (Tabela 4.3), assim como os riscos, não riscos, pontos de sensibilidade e *trade-offs* associados a cada atributo de qualidade da RADEG.

Tabela 4.3: Observações dos Atributos na AR

ID	Descrição	Resposta
P1	O atributo Aprendizibilidade foi considerado?	Sim e Parcial
P2	O atributo Jogabilidade foi considerado?	Ambos Sim
P3	O atributo Ludicidade foi considerado?	Sim e Parcial
P4	O atributo Usabilidade foi considerado?	Sim e Parcial
P5	O atributo Segurança foi considerado?	Sim e Parcial
P6	O atributo Manutenibilidade foi considerado?	Ambos Sim
P7	O atributo Acessibilidade foi considerado?	Sim e Parcial
P8	O atributo Compreensibilidade foi considerado?	Sim e Parcial
P9	O atributo Diversão foi considerado?	Sim e Parcial
P10	O atributo Interatividade foi considerado?	Sim e Parcial

Primeiramente, na análise do atributo **Aprendizibilidade** (P1), a avaliação resultou em respostas mistas, indicando que esse atributo foi parcialmente atendido. Nesse contexto, PA2 observou que, apesar da RADEG contemplar esse cenário de forma parcial, não há uma descrição explícita dos elementos arquiteturais que tornam o JED fácil de aprender nas visões geral e lógica da RADEG. Além disso, apontou, como ponto sensível, a falta de detalhamento dos componentes que contribuem para a aprendizagem do usuário. Como aprimoramento, sugeriu a adoção do conceito de Facilidade de Aprendizagem (*Learnability*), conforme definido no padrão ISO 25010, em substituição ao termo aprendizagem, visando maior alinhamento com padrões de qualidade de software. Em relação ao atributo **Jogabilidade** (P2), todos os avaliadores concordaram que ele foi devidamente considerado na RADEG, sem observações adicionais.

No caso de P3, referente ao atributo **Ludicidade**, que gerou opiniões divergen-

⁷Disponível em: <https://encurtador.com.br/jMvwQ>

tes, os avaliadores chegaram à conclusão de que a avaliação foi parcialmente satisfatória. PA2 apontou a necessidade de incluir uma definição mais precisa do conceito de lúdico e seus sub-atributos, como identidade, imersão, interatividade, níveis de complexidade, análise de desempenho e ser instrutivo. Além disso, sugeriu que esses elementos fossem explicitamente incorporados à RADEG como sub-atributos do conceito de ludicidade, proporcionando maior clareza e alinhamento com boas práticas de *design* de JEDs. Quanto ao atributo **Usabilidade** (P4), que teve respostas diferentes, os avaliadores PA1 e PA2 também apresentaram observações relevantes. PA1 destacou que a usabilidade foi abordada apenas no contexto do menu, sem uma visão abrangente do JED como um todo, sugerindo que a representação da usabilidade contemplasse a experiência completa do usuário no JED, e não apenas elementos isolados da interface. Por sua vez, PA2 ressaltou que a experiência do usuário está diretamente relacionada à satisfação decorrente do uso do JED. Dessa forma, apontou a necessidade de adicionar o atributo experiência do usuário e seus sub-atributos às visões geral e lógica da RADEG, garantindo uma especificação mais alinhada com modelos de qualidade centrados no usuário.

A questão referente ao atributo **Segurança** (P5), que obteve avaliações positivas, mas também parcialmente satisfatórias, PA2 sugeriu que os conceitos de segurança e privacidade sejam formalmente incorporados às visões lógica e geral da RADEG, garantindo uma especificação mais completa e alinhada com padrões de qualidade de software. Segundo PA2, no contexto de JEDs, segurança envolve mecanismos para prevenir erros do usuário e auxiliar na recuperação após interações incorretas, sendo um fator essencial de usabilidade. Já a privacidade, um sub-atributo da confidencialidade, refere-se à proteção dos dados dos jogadores contra acessos não autorizados, assegurando conformidade com boas práticas de segurança da informação. Dessa forma, a recomendação de PA2 foi que esses aspectos sejam formalmente incorporados à RADEG, garantindo maior clareza e alinhamento com padrões de qualidade de software. Quanto a P6, semelhante a P2, todos os avaliadores concluíram que a **Manutenibilidade** foi adequadamente considerada na RADEG, sem observações adicionais.

Em relação a P7, PA2 destacou que a **Acessibilidade** está ligada à usabilidade para usuários com limitações cognitivas, perceptivas ou motoras. Além disso, destacou

que a acessibilidade deve seguir quatro princípios: perceptível, operável, compreensível e robusto. Dessa forma, PA2 recomendou incluir esses princípios nas visões geral, lógica e de desenvolvimento da RADEG. Quanto ao atributo **Compreensibilidade** (P8), PA2 apontou que esse atributo está relacionado à comunicabilidade, que, por sua vez, se conecta à jogabilidade. Assim, sugeriu a criação de uma hierarquia entre compreensibilidade, comunicabilidade e jogabilidade na visão geral da RADEG, a fim de proporcionar uma abordagem mais estruturada e coerente na organização desses atributos.

Para o atributo **Diversão** (P9), PA2 destacou que diversão é um sub-atributo do conceito de lúdico e pode estar associada à experiência de usuário, que se refere à satisfação gerada pelo uso do sistema interativo. Além disso, em relação ao atributo **Interatividade** (P10), PA2 apontou que interatividade se relaciona à comunicabilidade, pois envolve os princípios de *discoverability* (facilidade de identificar ações na interface) e *feedback*, sugerindo a inclusão desses atributos nas visões geral, lógica ou de desenvolvimento da RADEG.

Por fim, em relação aos comentários sobre os diagramas, os avaliadores PA1 e PA3 fizeram observações relevantes. PA1 apontou questões nos diagramas de classe, implantação e pacotes, sugerindo melhorias nos relacionamentos entre os elementos. Já PA3 destacou que não ficou claro como a RADEG se adequaria diante da inclusão de novos requisitos ou da ausência de algum deles. Com base nessas observações, os diagramas são revisados e o processo de adequação da RADEG ajustado.

4.5 Fase 2: Avaliação (continuação)

Nessa fase, os especialistas em JEDs passam a integrar os procedimentos e a análise contínua. Dessa forma, a etapa 1 é repetida para que esses especialistas compreendam o método e os papéis que devem desempenhar. Nesse contexto, os tomadores de decisão recapitulam os resultados das etapas 2 a 6 e compartilham a lista atualizada de riscos, não riscos, pontos de sensibilidade e *trade-offs*. Assim, os especialistas são devidamente informados sobre os resultados obtidos até o momento, permitindo a continuidade do processo de avaliação (BASS; CLEMENTS; KAZMAN, 2012). As três etapas restantes, apresentadas nas próximas subseções, puderam, então, ser conduzidas.

4.5.1 Etapa 7: *Brainstorming* e Priorização de Cenários

Na presente etapa, os tomadores de decisão solicitam que os especialistas realizem um *brainstorming* para identificar cenários operacionalmente significativos, com foco nas funções individuais de cada especialista. Enquanto a geração da árvore de utilidade (Subseção 4.4.5) é utilizada principalmente para entender os atributos de qualidade, o objetivo do *brainstorming* de cenários é obter uma visão mais ampla da comunidade de especialistas, ou seja, compreender o que o sucesso do sistema representa para eles (BASS; CLEMENTS; KAZMAN, 2012).

Para essa avaliação, adotou-se a seguinte abordagem: a cada especialista foi alocado um número de votos equivalente a 30% do total de cenários, permitindo que cada um distribuisse seus votos conforme julgasse mais adequado (BASS; CLEMENTS; KAZMAN, 2012). No caso da RADEG, cada especialista recebeu 3 votos, considerando a existência de 10 cenários (Subseção 4.4.5). Desse modo, a avaliação contou com a participação de 12 especialistas, totalizando 36 votos. A Figura 4.14 apresenta os resultados da votação para cada cenário.

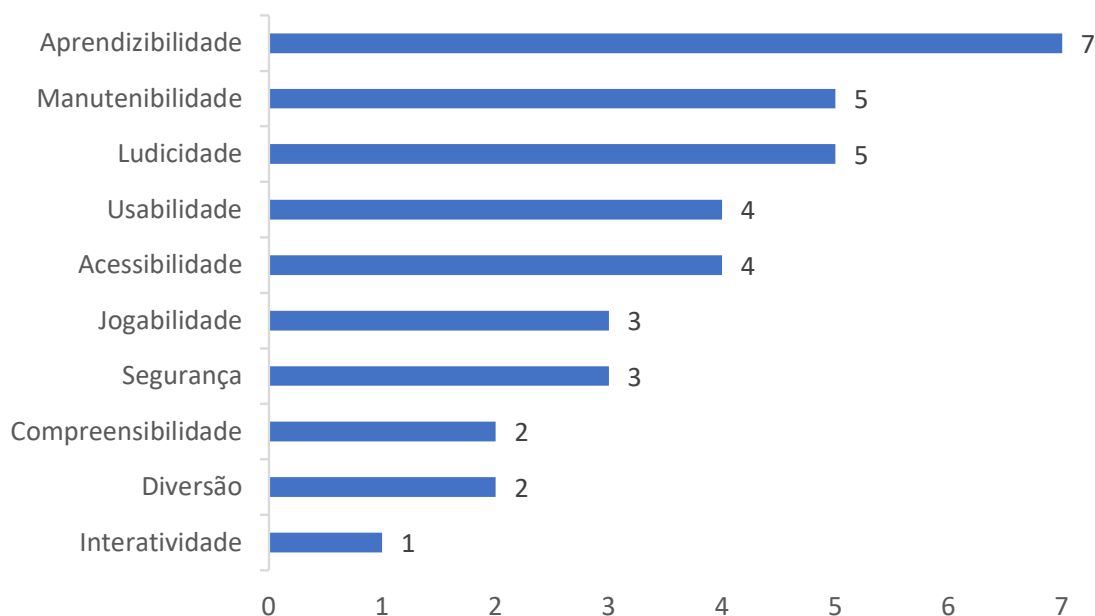


Figura 4.14: Avaliação dos cenários.

A partir da análise dos resultados, observou-se que o cenário de **Aprendizabilidade** manteve-se como o mais votado, evidenciando a preocupação dos especialistas com a eficácia da aprendizagem dos conceitos apresentados no JED. Contudo, algumas

divergências em relação à árvore de utilidade foram identificadas.

Os cenários de **Manutenibilidade** e **Ludicidade** foram os segundos mais votados, com 5 votos cada. No entanto, esse resultado difere da árvore de utilidade, onde o segundo cenário mais relevante é a jogabilidade. Essa diferença indica que, para os especialistas, a capacidade de manutenção do JED e a presença de elementos lúdicos são mais importantes no desenvolvimento para o sucesso dos JEDs.

Além disso, os cenários de **Usabilidade** e **Acessibilidade** receberam 4 votos cada, mantendo a posição da usabilidade conforme definido na árvore de utilidade, mas com uma variação na posição da acessibilidade, que recebeu maior prioridade pelos especialistas. Já os cenários de **Jogabilidade** e **Segurança**, que obtiveram 3 votos cada, apresentaram diferenças em relação à árvore de utilidade, onde a jogabilidade possuía uma prioridade mais elevada (segunda posição).

Ademais, os cenários de **Compreensibilidade**, **Diversão** e **Interatividade** receberam as menores quantidades de votos, mas permaneceram nas mesmas posições da árvore de utilidade. Com isso, foi possível perceber um alinhamento entre as respostas dos especialistas e a ordem na árvore de utilidade.

Por fim, com base na análise comparativa entre os resultados da votação dos especialistas e a árvore de utilidade, observa-se que 6 dos 10 cenários mantiveram a mesma posição, representando uma concordância de 60% entre as duas classificações. Desse modo, apesar de algumas diferenças, a avaliação da RADEG por meio dos especialistas demonstrou um alinhamento positivo com as expectativas dos desenvolvedores.

4.5.2 Etapa 8: Análise das Abordagens Arquiteturais com os Cenários

Nesta etapa, são realizadas as mesmas atividades da Etapa 6 (Subseção 4.4.6), utilizando os cenários com melhor classificação, conforme obtidos na Etapa 7 (Subseção 4.5.1). Dessa forma, as abordagens arquiteturais foram inspecionadas, e cada cenário foi detalhado. A seguir, apresenta-se a descrição da avaliação de cada cenário concreto.

O primeiro cenário, conforme ilustrado na Tabela 4.4, avalia a **Aprendizibilidade**, destacando a necessidade de integrar tutoriais e mecanismos de *feedbacks* para

facilitar a compreensão dos jogadores iniciantes. Porém, o principal desafio é equilibrar a inserção de orientações sem comprometer a diversão.

Tabela 4.4: Análise do Cenário de Aprendizibilidade

Resumo do Cenário	O jogador iniciante acessa o JED pela primeira vez, encontra dificuldades para entender os conceitos e recebe <i>feedback</i> e orientações do sistema
Objetivos de Negócio	Melhorar a compreensão dos conceitos pelos jogadores iniciantes, assegurando eficácia no aprendizado
Atributo de Qualidade	Aprendizibilidade e Compreensibilidade
Análise Arquitetural	A RADEG deve integrar módulos específicos para ensino, como tutoriais interativos, dicas e <i>feedbacks</i> imediatos para medir a compreensão do jogador; necessidade de detalhar, na visão geral e lógica, os elementos que facilitam o aprendizado, alinhando-os às diretrizes do ISO 25010
Riscos	Se os elementos de <i>feedback</i> e tutoriais não forem eficazes, a taxa de compreensão pode ficar abaixo do desejado; desalinhamento com os critérios do ISO 25010 pode comprometer a usabilidade e, consequentemente, o engajamento dos jogadores
Trade-offs	Os tutoriais e dicas podem melhorar a compreensão, mas também podem interromper o fluxo do JED, afetando negativamente a diversão

O segundo cenário, na Tabela 4.5, trata da **Manutenibilidade**, visando facilitar modificações e adições de funcionalidades sem impactar negativamente outros componentes do JED.

Tabela 4.5: Análise do Cenário de Manutenibilidade

Resumo do Cenário	O desenvolvedor modifica ou adiciona funcionalidades no JED sem impactar outros componentes
Objetivos de Negócio	Facilitar a manutenção e evolução do JED, garantindo que novas implementações ocorram com mínimo impacto
Atributo de Qualidade	Extensibilidade e Manutenibilidade
Análise Arquitetural	A especificação atente ao cenário
Riscos	
Trade-offs	

A Tabela 4.6 apresenta o cenário de **Ludicidade**, que reforça a necessidade de aumentar o engajamento dos jogadores por meio de elementos lúdicos. Porém, o desafio consiste em evitar que o excesso desses elementos prejudique o aprendizado.

Já a Tabela 4.7 trata da **Usabilidade**, reforçando a necessidade de uma interface intuitiva e acessível para todos os perfis de jogadores. Com isso, seu risco principal é concentrar esforços apenas no menu principal.

No contexto da **Acessibilidade**, a Tabela 4.8 destaca a importância de mecanismos adaptativos, assegurando que jogadores com deficiência motora possam utilizar o JED de forma eficiente. Ao inserir esses mecanismos, deve-se evitar que a complexidade

Tabela 4.6: Análise do Cenário de Ludicidade

Resumo do Cenário	O jogador interage com elementos lúdicos do JED, como animações e personagens, recebendo recompensas e <i>feedbacks</i> visuais e sonoros ao atingir metas
Objetivos de Negócio	Aumentar o engajamento e a satisfação do jogador por meio de elementos lúdicos
Atributo de Qualidade	Atratividade, Diversão e Ludicidade
Análise Arquitetural	Incluir módulos gráficos e interativos que garantam animações, efeitos sonoros, <i>feedbacks</i> imediatos e recompensas
Riscos	Falta de integração dos sub-atributos pode comprometer satisfação do jogador; excesso de elementos lúdicos pode comprometer o aprendizado
Trade-offs	A implementação de elementos lúdicos aumentam o engajamento do jogador, mas também podem aumentar a complexidade de desenvolvimento

Tabela 4.7: Análise do Cenário de Usabilidade

Resumo do Cenário	O jogador, independentemente da experiência, navega pelo menu principal do JED para configurar e iniciar o jogo
Objetivos de Negócio	Garantir uma interface acessível e fácil de usar para todos os jogadores
Atributo de Qualidade	Navegabilidade, Simplicidade e Usabilidade
Análise Arquitetural	A RADEG deve garantir um <i>design</i> de interface intuitivo, responsivo e acessível que se estenda por todo o JED, não se limitando apenas ao menu; incluir diretrizes que assegurem eficácia, eficiência e satisfação, seguindo princípios de <i>design</i> de usabilidade, como as heurísticas de Nielsen
Riscos	Foco exclusivo no menu pode levar a uma percepção incompleta de usabilidade; falta de integração dos sub-atributos da experiência pode comprometer a satisfação do jogador
Trade-offs	

afete a usabilidade.

Tabela 4.8: Análise do Cenário de Acessibilidade

Resumo do Cenário	O jogador com deficiência motora acessa o JED e navega pela interface utilizando as configurações de acessibilidade disponíveis
Objetivos de Negócio	Garantir que jogadores com deficiência motora possam utilizar o JED de forma eficiente
Atributo de Qualidade	Acessibilidade e Adaptabilidade
Análise Arquitetural	A RADEG deve suportar configurações de acessibilidade; incluir, nas visões geral, lógica e de desenvolvimento, mecanismos que adaptem a interface conforme as necessidades dos jogadores
Riscos	Não implementar adequadamente os padrões de acessibilidade pode reduzir a satisfação dos jogadores com necessidades especiais
Trade-offs	Incluir configurações de acessibilidade pode aumentar a complexidade de desenvolvimento e prejudicar a usabilidade, se não forem implementadas corretamente

O cenário de **Jogabilidade**, representado na Tabela 4.9, ressalta a necessidade de controles intuitivos para novos jogadores do JED.

Em relação à **Segurança**, a Tabela 4.10 avalia mecanismos de autenticação e proteção contra acessos não autorizados.

O cenário de **Compreensibilidade**, apresentado na Tabela 4.11, enfatiza a im-

Tabela 4.9: Análise do Cenário de Jogabilidade

Resumo do Cenário	O jogador sem experiência utiliza o JED pela primeira vez e interage com os controles para realizar uma ação básica, recebendo instruções claras na tela
Objetivos de Negócio	Garantir uma experiência intuitiva e acessível para novos jogadores
Atributo de Qualidade	Jogabilidade, Navegabilidade e Simplicidade
Análise Arquitetural	A especificação atente ao cenário
Riscos	
Trade-offs	

Tabela 4.10: Análise do Cenário de Segurança

Resumo do Cenário	Um jogador tenta acessar o JED com informações inválidas em um curto período, ativando o sistema de autenticação, que bloqueia o acesso e registra o incidente
Objetivos de Negócio	Prevenir acessos não autorizados e proteger a conta dos jogadores
Atributo de Qualidade	Confiabilidade e Segurança
Análise Arquitetural	A RADEG deve garantir mecanismos de autenticação que detecte tentativas de login incorretas em curto período e bloqueie automaticamente o acesso; incluir padrões de segurança e privacidade
Riscos	Falhas no monitoramento e bloqueio podem permitir acessos indevidos e comprometer os dados dos jogadores
Trade-offs	A implementação de mecanismos de segurança pode aumentar a complexidade de desenvolvimento

portância da revisão de conceitos para reforçar o aprendizado dos jogadores. Porém, o seu desafio principal está em manter essa revisão sem afetar a fluidez do JED.

Tabela 4.11: Análise do Cenário de Compreensibilidade

Resumo do Cenário	O jogador, ao iniciar uma nova fase, revisa um conceito previamente apresentado por meio de animações e exemplos
Objetivos de Negócio	Reforçar a compreensão dos conceitos para melhorar a retenção do aprendizado
Atributo de Qualidade	Aprendizabilidade, Comunicabilidade, Compreensibilidade, Jogabilidade e Multimodalidade
Análise Arquitetural	A RADEG deve garantir mecanismos para reforçar a compreensão dos conceitos; criar uma hierarquia que alinhe compreensibilidade, comunicabilidade e jogabilidade
Riscos	A ausência de uma integração entre os requisitos pode comprometer o entendimento dos conceitos, o afetando negativamente o desempenho dos jogadores
Trade-offs	As revisões de conceitos podem tornar o JED mais lento, afetando a diversão e interatividade

Na Tabela 4.12, o cenário de **Diversão** destaca a importância de proporcionar uma experiência envolvente por meio de recompensas e *feedbacks*.

Por fim, a Tabela 4.13 apresenta o cenário de **Interatividade**, que visa aumentar o engajamento dos jogadores ao permitir escolhas que impactam a narrativa e o ambiente do JED. O principal desafio é garantir que essas escolhas sejam claras e relevantes para o jogador.

Tabela 4.12: Análise do Cenário de Diversão

Resumo do Cenário	O jogador completa uma tarefa na fase do JED, ganhando recompensas e recebendo <i>feedbacks</i> por meio de animações e efeitos sonoros
Objetivos de Negócio	Garantir que o JED proporcione uma experiência divertida e motivadora para os jogadores
Atributo de Qualidade	Atratividade, Diversão e Envolvibilidade
Análise Arquitetural	A especificação atente ao cenário
Riscos	
Trade-offs	

Tabela 4.13: Análise do Cenário de Interatividade

Resumo do Cenário	O jogador escolhe entre dois caminhos durante uma fase, e o JED ajusta a narrativa e o ambiente conforme a escolha
Objetivos de Negócio	Aumentar a imersão e a satisfação do jogador ao oferecer escolhas que impactam a narrativa
Atributo de Qualidade	Imersibilidade e Interatividade
Análise Arquitetural	A RADEG deve garantir o ajuste dinâmico à narrativa e ao ambiente com base nas escolhas do jogador; incluir comunicabilidade às visões geral, lógica e de desenvolvimento da RADEG
Riscos	Se as escolhas não forem claramente apresentadas, a interatividade pode se tornar irrelevante, reduzindo a satisfação do jogador
Trade-offs	Oferecer múltiplos caminhos interativos pode prejudicar a usabilidade

4.5.3 Etapa 9: Apresentação dos resultados

Por fim, nesta etapa, todas as informações coletadas são resumidas e apresentadas à equipe de avaliação. A forma de apresentação dos resultados é alterada conforme a análise arquitetural, sendo estruturada de maneira a descrever, de forma concisa, os principais resultados das atividades de inspeção (Etapa 8 - Subseção 4.5.2). Com base na análise das abordagens arquiteturais e dos cenários concretos realizados na etapa anterior, conclui-se que a especificação da RADEG apresenta diversos riscos, os quais são descritos a seguir.

De modo geral, os principais riscos estão relacionados à complexidade de equilibrar alguns requisitos, como usabilidade, acessibilidade, segurança e ludicidade, sem comprometer a experiência do jogador ou os objetivos do JED. Nesse contexto, um desafio importante é a integração de elementos pedagógicos e lúdicos, onde há o risco de que o excesso de animações, *feedbacks* e recompensas possa prejudicar o aprendizado ao desviar o foco dos conceitos essenciais. Por outro lado, a ausência desses elementos pode comprometer o engajamento e a satisfação dos jogadores.

Além disso, a usabilidade e acessibilidade enfrentam riscos associados à implementação inadequada de padrões, o que pode dificultar o acesso ao JED por diferentes

perfis de jogadores. Já na segurança, a ausência de mecanismos eficazes para prevenção de acessos não autorizados pode comprometer a proteção dos dados dos jogadores.

Por fim, há riscos associados à interatividade e à compreensibilidade dos conceitos apresentados. Caso o JED não ofereça instruções claras ou revisões eficazes, a experiência do jogador pode ser prejudicada, impactando diretamente os objetivos do JED. Assim, esses desafios na RADEG devem ser abordados com soluções que garantam o engajamento dos jogadores e a sustentabilidade do JED.

4.6 Fase 3: Acompanhamento

Nessa fase, a equipe de avaliação é responsável por elaborar e entregar um relatório final escrito ao solicitante da avaliação (BASS; CLEMENTS; KAZMAN, 2012). No entanto, essa fase não se fez necessária no presente estudo, uma vez que se trata de um trabalho acadêmico e não houve um solicitante externo para a avaliação.

4.7 Ameaças à Validade

Nesta Subseção são discutidas as ameaças à validade que podem impactar os resultados e conclusões deste trabalho. Tais ameaças foram categorizadas em validade externa, validade interna e validade de construção. Essas ameaças foram definidas com base nas diretrizes propostas em Wohlin et al. (2012).

A **validade externa** refere-se à generalização dos resultados para contextos além deste trabalho. Uma ameaça dessa categoria é a seleção dos estudos dos JEDs a partir de duas bases de dados específicas (SBC-OpenLib e Scopus), o que limita a abrangência dos estudos identificados. Para mitigar isso, aplicou-se a técnica de *snowballing* para identificar novos estudos relevantes. Além disso, a participação de um número restrito de especialistas na avaliação da RADEG pode comprometer a representatividade dos resultados. Porém, a diversidade de conhecimentos dos especialistas garantiu uma análise mais aprofundada e representativa da RADEG.

Quanto à **validade interna**, relacionada à influência do fator humano, uma ameaça principal é a subjetividade na identificação dos RS e RA, que envolveu inter-

pretação dos estudos e experiência prática dos pesquisadores. Essa subjetividade pode ter influenciado a seleção e priorização de requisitos, especialmente os implícitos. Para minimizar essa ameaça, o processo de extração e categorização dos requisitos foi conduzido de forma colaborativa, com discussões e análises qualitativas entre os pesquisadores e especialistas, garantindo consistência e alinhamento com os objetivos do trabalho.

Por fim, a **validade de construção**, que refere-se à adequação das medidas e métodos utilizados, também apresenta ameaças. Uma delas é a adequação do *checklist* FERA e do método ATAM para avaliar RADEG. Embora esses métodos não sejam voltados para JEDs, sua adaptação para o contexto dos JEDs foi feita minuciosamente. Essa adaptação incluiu a seleção de 23 perguntas mais relevantes e a definição estruturada de 10 cenários, além da votação entre especialistas. Com isso, foi possível reduzir os vieses e, conseqüentemente, a subjetividade, garantindo uma análise mais objetiva e alinhada aos objetivos desse trabalho.

4.8 Considerações Finais

Dada a importância da avaliação de AR, este capítulo apresentou a avaliação da RADEG por meio do *checklist* FERA e do método ATAM. Com base nas avaliações realizadas por meio do FERA, concluiu-se que a AR considera diversos atributos de qualidade importantes, como aprendizibilidade, jogabilidade, usabilidade e manutenibilidade. No entanto, foram identificadas necessidades de refinamento, principalmente nos requisitos de diversão e segurança, além de ajustes na documentação e maior detalhamento de alguns atributos de qualidade para garantir uma instância eficaz e sustentável ao longo do tempo.

Por fim, a aplicação do ATAM permitiu a identificação e priorização de cenários por meio de *brainstorming*, a construção da árvore de utilidades, a identificação de riscos, a documentação de não riscos, além da determinação de pontos de sensibilidade e *trade-offs*. Nesse contexto, ao reconhecer os riscos envolvidos, possibilita-se ajustar a RADEG para obter resultados mais positivos em um estudo de caso, se comparado a uma abordagem sem a avaliação pelo ATAM.

5 Conclusão

5.1 Contribuição Científica

Como principal contribuição deste trabalho, destaca-se o estabelecimento da RADEG, uma AR de apoio ao desenvolvimento de JEDs para a Educação em Computação. A RADEG proposta inclui a disponibilização de novos artefatos, como documentos de requisitos, padrões educacionais/pedagógicos e taxonomias, para apoiar o desenvolvimento desses jogos de forma mais estruturada e eficiente. Além disso, espera-se que a RADEG ajude a desenvolver JEDs que não apenas atendam aos aspectos técnicos, mas que também considerem o perfil dos estudantes da área, otimizando o processo de ensino-aprendizagem em Computação.

Além disso, pode-se considerar como contribuição adicional o conjunto de Requisitos do Sistema, Requisitos Arquiteturais, *stakeholders* e interesses identificados ao longo desse processo. Esses elementos podem servir como base para o desenvolvimento de outros JEDs ou para estabelecer ARs em diferentes domínios, considerando que grande parte dos requisitos levantados são genéricos e aplicáveis a diversos contextos.

As avaliações realizadas por meio do FERA e do ATAM mostraram-se benéficas, permitindo a identificação da necessidade de refinamentos e ajustes, contribuindo para o aumento da confiança dos especialistas e para a eficiência da AR no desenvolvimento de JEDs de alta qualidade. Além disso, os riscos também foram identificados e, por meio dos ajustes, é possível garantir um maior alinhamento das decisões arquiteturais com os interesses dos *stakeholders*. Dessa forma, a avaliação permitiu uma nova observação sobre a RADEG, contribuindo para o reconhecimento de ajustes previamente não identificados.

Nesse contexto, ao longo da realização deste estudo, foi possível aprofundar o conhecimento sobre as características dos JEDs voltados para a Educação em Computação. O MSL resultou em 194 estudos analisados, a partir dos quais foram extraídos os requisitos para a implementação da RADEG. Além disso, houve um aprofundamento no processo de modelagem, com a inserção dos Requisitos Arquiteturais em seis visões arquiteturais.

A aplicação do ATAM, conforme mencionado anteriormente, foi fundamental para revelar riscos durante o projeto da RADEG, contribuindo para uma visão mais completa dela.

5.2 Publicações

Os resultados alcançados ao longo deste trabalho incluem a publicação de três artigos, destacando a importância e a relevância dos JEDs:

- GENESIO, NIKOLAS OLIVER SALES; OLIVEIRA, ALESSANDREIA MARTA DE; OLIVEIRA, EDMAR WELINGTON; VALLE, PEDRO HENRIQUE DIAS. Panorama de Estudos sobre Jogos Educacionais Digitais em Educação em Computação. In: Workshop sobre Educação em Computação, 2024, Brasil. Anais do XXXII Workshop sobre Educação em Computação (WEI 2024);
- GENESIO, NIKOLAS OLIVER SALES; OLIVEIRA, ALESSANDREIA MARTA DE; CHAVES, LUCIANO JEREZ; SILVA, WILLIAMSON ALISON FREITAS; VALLE, PEDRO HENRIQUE DIAS. Um Mapeamento Sistemático sobre Jogos Educacionais Digitais para o Ensino-Aprendizagem de Estrutura de Dados. In: Simpósio Brasileiro de Informática na Educação, 2024, Brasil. Anais do XXXV Simpósio Brasileiro de Informática na Educação (SBIE 2024);
- GENESIO, NIKOLAS OLIVER SALES; OLIVEIRA, ALESSANDREIA MARTA DE; VALLE, PEDRO HENRIQUE DIAS. Uma Abordagem Arquitetural para o Desenvolvimento de Jogos Educacionais Digitais na Educação em Computação. In: Laboratório de Ideias - Simpósio Brasileiro de Educação em Computação, 2025, Brasil. No prelo.

5.3 Trabalhos Futuros

Apesar das dificuldades enfrentadas, como a participação de voluntários para a avaliação e entendimento completo do ATAM, foi possível concluir as quatro etapas do ProSA-RA. Visando a dar continuidade ao trabalho desenvolvido, propõe-se a evolução da RADEG através dos resultados obtidos por meio do *checklist* FERA e do método ATAM. Além

disso, após a evolução da RADEG, pode ser conduzida uma nova avaliação com o objetivo de verificar se as modificações implementadas resultaram nas melhorias esperadas.

Como parte da avaliação contínua da RADEG, sugere-se a implementação de um estudo de caso para verificar se os objetivos estabelecidos para a implementação da RADEG foram alcançados por meio do desenvolvimento de um JED baseado nela. Além disso, futuras pesquisas podem envolver a reimplementação de JEDs utilizando a RADEG, possibilitando uma avaliação comparativa entre esses JEDs.

Por fim, espera-se que a RADEG contribua para a evolução dos JEDs, promovendo maior qualidade nesses JEDs e favorecendo aspectos relacionados à análise, entendimento e reúso. Nesse contexto, pode-se realizar uma análise quantitativa do aumento desses aspectos proporcionado pelo uso da RADEG. Com essa abordagem, pode-se obter mais evidências dos benefícios da utilização da RADEG no desenvolvimento de JEDs sustentáveis.

Bibliografia

- AIRES, S. F.; BARBOSA, J. F.; MADEIRA, C. A. Desenvolvendo jogos educacionais digitais inovadores e instigantes com o framework playeduc. *Sociedade Brasileira de Computação*, 2020.
- ALCANTARA, L. Y. de A.; SANTOS, C. A. dos. A gamificação como metodologia inovadora. *GAMIFICAR EM SALA DE AULA: Ensino Fundamental Volume 3*, Editora CRV, p. 13, 2023.
- ALLIAN, A. P.; SILVA, L. F.; OLIVEIRAJR, E.; NAKAGAWA, E. Y. Vmtools-ra: a reference architecture for software variability tools. *Journal of Universal Computer Science*, Pensoft Publishers, v. 29, n. 7, p. 649, 2023.
- ALVIM, Í. V.; BITTENCOURT, R. A.; DURAN, R. S. Evasão nos cursos de graduação em computação no brasil. In: SBC. *Anais do IV Simpósio Brasileiro de Educação em Computação*. [S.l.], 2024. p. 1–11.
- ANGELOV, S.; GREFEN, P.; GREEFHORST, D. A classification of software reference architectures: Analyzing their success and effectiveness. In: IEEE. *2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture*. [S.l.], 2009. p. 141–150.
- ANGELOV, S.; GREFEN, P.; GREEFHORST, D. A framework for analysis and design of software reference architectures. *Information and Software Technology*, Elsevier, v. 54, n. 4, p. 417–431, 2012.
- ANGELOV, S.; TRIENEKENS, J. J.; GREFEN, P. Towards a method for the evaluation of reference architectures: Experiences from a case. In: SPRINGER. *Software Architecture: Second European Conference, ECSA 2008 Paphos, Cyprus, September 29-October 1, 2008 Proceedings 2*. [S.l.], 2008. p. 225–240.
- ARAÚJO, W. O. de; ARANHA, E. H. da S.; MADEIRA, C. A. G. Geração procedural de conteúdo aplicada a jogos digitais educacionais. *Jornada de Atualização em Informática na Educação*, v. 7, n. 1, p. 1–20, 2018.
- ARIMOTO, M.; OLIVEIRA, W. Dificuldades no processo de aprendizagem de programação de computadores: um survey com estudantes de cursos da área de computação. In: SBC. *Anais do XXVII Workshop sobre Educação em Computação*. [S.l.], 2019. p. 244–254.
- ARSANJANI, A.; ZHANG, L.-J.; ELLIS, M.; ALLAM, A.; CHANNABASAVAIAH, K. S3: A service-oriented reference architecture. *IT professional*, IEEE, v. 9, n. 3, p. 10–17, 2007.
- ASSUMPÇÃO, M. de; JUNIOR, M. M. C.; FELINTO, A. S.; AYLON, L. B. R. Manna-x: Projeto, desenvolvimento e avaliação de um jogo multidisciplinar para ensino na ciência da computação. In: SBC. *Anais Estendidos do XXI Simpósio Brasileiro de Jogos e Entretenimento Digital*. [S.l.], 2022. p. 918–928.

- BAJAJ, M.; FRIEDENTHAL, S.; SEIDWITZ, E. Systems modeling language (sysml v2) support for digital engineering. *Insight*, Wiley Online Library, v. 25, n. 1, p. 19–24, 2022.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software Architecture in Practice*. 3rd. ed. [S.l.]: Addison-Wesley Professional, 2012. ISBN 0321815734.
- BASSANI, P. S. Estudos de software e o protagonismo do código. In: SBC. *Anais Estendidos do XVIII Simpósio Brasileiro de Sistemas de Informação*. [S.l.], 2022. p. 245–252.
- BATTISTELLA, P.; WANGENHEIM, C. G. von. Games for teaching computing in higher education—a systematic review. *IEEE Technology and Engineering Education*, sn, v. 9, n. 1, p. 8–30, 2016.
- BAYER, J. View-based software documentation. Fraunhofer IRB Verlag, 2004.
- BAYER, J.; FORSTER, T.; GANESAN, D.; GIRARD, J.-F.; JOHN, I.; KNODEL, J.; KOLB, R.; MUTHIG, D. Definition of reference architectures based on existing systems. *Fraunhofer IESE, March*, 2004.
- BITTENCOURT, R. A.; SANTANA, B. L.; ARAUJO, L. G. J. Computação fundamental: currículo e livros didáticos de computação para o ensino fundamental ii. *Revista Brasileira de Informática na Educação*, v. 29, p. 662–691, 2021.
- BOEHM, B. A view of 20th and 21st century software engineering. In: *Proceedings of the 28th international conference on Software engineering*. [S.l.: s.n.], 2006. p. 12–29.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *The Unified Modeling Language User Guide*. Pearson Education, 1999. (The Addison-Wesley object technology series). ISBN 9788177583724. Disponível em: <https://books.google.com.br/books?id=a5J49FoFKq8C>.
- BOUWMAN, M.; LUTTIK, B.; WAL, D. van D. A formalisation of sysml state machines in mcrl2. In: SPRINGER. *International Conference on Formal Techniques for Distributed Objects, Components, and Systems*. [S.l.], 2021. p. 42–59.
- BRASIL. *Computação na Educação Básica - Complemento à BNCC*. 2022. Retrieved February 05, 2024 from <http://portal.mec.gov.br/docman/fevereiro-2022-pdf/236791-anexo-ao-parecer-cneceb-n-2-2022-bncc-computacao/file>.
- BRUM, M. G.; CRUZ, M. E. K. da. Gamificação para o ensino de computação na educação básica. In: SBC. *Anais do XXV Workshop sobre Educação em Computação*. [S.l.], 2017.
- CAMARGO, M. P. de O.; PEREIRA, G. dos S.; ALMEIDA, D.; BENTO, L. A.; DORANTE, W. F.; AFFONSO, F. J. Ra4self-cps: a reference architecture for self-adaptive cyber-physical systems. *IEEE Latin America Transactions*, IEEE, v. 22, n. 2, p. 113–125, 2024.
- CARVALHO, L.; SANTOS, A.; NAKAMURA, F.; OLIVEIRA, E. Detecção precoce de evasão em cursos de graduação presencial em computação: um estudo preliminar. In: SBC. *Anais do xxvii workshop sobre educação em computação*. [S.l.], 2019. p. 233–243.
- CARVALHO, M. B. Serious games for learning: a model and a reference architecture for efficient game development. 2017.

- CASTRO, R. M. de; SIQUEIRA, S. Técnicas alternativas de ensino (aprendizagem ativa) para disciplinas da computação: Um mapeamento sistemático no contexto brasil. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2019. v. 25, n. 1, p. 1409–1413.
- CLASSE, T. M. de; CASTRO, R. M. de. Ludificando os fundamentos de computação através de aprendizagem ativa. In: SBC. *Anais do XXVIII Workshop sobre Educação em Computação*. [S.l.], 2020. p. 116–120.
- CLASSE, T. M. de; CASTRO, R. M. de. Melhorando o aproveitamento dos alunos em fundamentos de computação e sistemas através de atividades lúdicas. *Revista Novas Tecnologias na Educação*, v. 18, n. 2, p. 296–305, 2020.
- CLEMENTINO, E. G.; SILVA, T. R. da; ARANHA, E. H. da S.; SANTOS, F. G. dos. Jogos não digitais para ensino de computação—um mapeamento sistemático. In: SBC. *Simpósio Brasileiro de Informática na Educação (SBIE)*. [S.l.], 2022. p. 540–550.
- CLOUTIER, R.; MULLER, G.; VERMA, D.; NILCHIANI, R.; HOLE, E.; BONE, M. The concept of reference architectures. *Systems Engineering*, Wiley Online Library, v. 13, n. 1, p. 14–27, 2010.
- COMMITTEE, A. W. G. of the S. E. et al. Recommended practice for architectural description of software intensive systems. *IEEE Standards Department, Piscataway, New Jersey, USA*, 2000.
- DAIREL, J. G. de M.; CATTELAN, R. G.; ARAÚJO, R. D. Uma proposta de arquitetura computacional para autoria de jogos digitais educacionais com suporte a técnicas de analíticas de aprendizagem-studx. In: SBC. *Anais Estendidos do XII Congresso Brasileiro de Informática na Educação*. [S.l.], 2023. p. 241–246.
- DUARTE, L. S. *Estabelecimento de uma arquitetura de referência para aplicações de televisão digital*. Tese (Doutorado) — Universidade de São Paulo, 2012.
- DUTRA, T. C.; TONDORF, D. F.; ZILS, T. A.; FERREIRA, A. E.; GASPARINI, I.; HOUNSELL, M. da S.; MASCHIO, E. Métodos de avaliação de ihc no contexto de jogos sérios educacionais: Um mapeamento sistemático. In: SBC. *Anais do XXXII Simpósio Brasileiro de Informática na Educação*. [S.l.], 2021. p. 564–575.
- FALCÃO, T. P.; PERES, F.; OLIVEIRA, G.; MORAIS, D. Design participativo de jogos educacionais na comunidade de prática do projeto demults. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2017. v. 6, n. 1, p. 1344.
- FARSHIDI, S.; JANSEN, S.; WERF, J. M. van der. Capturing software architecture knowledge for pattern-driven design. *Journal of Systems and Software*, Elsevier, v. 169, p. 110714, 2020.
- FEICHAS, F. A.; SEABRA, R. D.; SOUZA, A. D. de. Gamificação no ensino superior em ciência da computação: Uma revisão sistemática da literatura. *Revista Novas Tecnologias na Educação*, v. 19, n. 1, p. 443–452, 2021.
- FEITOSA, D. *Simus-uma arquitetura de referência para sistemas multirrobóticos de serviço*. Tese (Doutorado) — Universidade de São Paulo, 2013.

- FELIZARDO, K. R.; MENDES, E.; KALINOWSKI, M.; SOUZA, É. F.; VIJAYKUMAR, N. L. 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*. [S.l.: s.n.], 2016. p. 1–6.
- FILHO, J. U. V.; LOPES, R. de S.; SHITSUKA, R. Informática e filosofia: o desenvolvimento de um jogo computacional como processo de aprendizagem. *Research, Society and Development*, Grupo de Pesquisa Metodologias em Ensino e Aprendizagem em Ciências, v. 8, n. 11, p. 148111446, 2019.
- FIORAVANTI, M. L.; NAKAGAWA, E. Y.; BARBOSA, E. F. Educar: uma arquitetura de referência para ambientes educacionais. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2010. v. 1, n. 1.
- FRANK, B.; REGINE, M.; HANS, R.; PETER, S.; MICHAEL, S. Pattern-oriented software architecture: a system of patterns. *Wiley, ISBN 0*, v. 471, n. 95889, p. 7, 1996.
- FROSI, F. O.; JAQUES, P. A. Jogos digitais para o ensino de programação: uma revisão sistemática das pesquisas publicadas no brasil entre 2015 e 2019. *Anais do XIX Simpósio Brasileiro de Jogos e Entretenimento Digital-SBGames, Recife/PE, Brasil*, p. 653–652, 2020.
- GALSTER, M.; AVGERIOU, P. Empirically-grounded reference architectures: a proposal. In: *Proceedings of the joint ACM SIGSOFT conference–QoSA and ACM SIGSOFT symposium–ISARCS on Quality of software architectures–QoSA and architecting critical systems–ISARCS*. [S.l.: s.n.], 2011. p. 153–158.
- GARCÊZ, A. V.; OLIVEIRA, J. M. L. de. Os benefícios da utilização de jogos digitais para o ensino de programação para crianças. *Research, Society and Development*, v. 11, n. 17, p. e239111738122–e239111738122, 2022.
- GARCIA, J.; MIRAKHORLI, M.; XIAO, L.; ZHAO, Y.; MUJHID, I.; PHAM, K.; OKUTAN, A.; MALEK, S.; KAZMAN, R.; CAI, Y. et al. Constructing a shared infrastructure for software architecture analysis and maintenance. In: *IEEE. 2021 IEEE 18th International Conference on Software Architecture (ICSA)*. [S.l.], 2021. p. 150–161.
- GARLAN, D.; ALLEN, R.; OCKERBLOOM, J. Exploiting style in architectural design environments. *ACM SIGSOFT software engineering notes*, ACM New York, NY, USA, v. 19, n. 5, p. 175–188, 1994.
- GARLAN, D.; BACHMANN, F.; IVERS, J.; STAFFORD, J.; BASS, L.; CLEMENTS, P.; MERSON, P. *Documenting Software Architectures: Views and Beyond*. 2nd. ed. [S.l.]: Addison-Wesley Professional, 2010. ISBN 0321552687.
- GARLAND, J. *Large-scale software architecture: a practical guide using UML*. [S.l.]: John Wiley & Sons, 2003.
- GAROZI, P. H. T.; JUNIOR, M. M. C.; COSTA, Y. M. Labirinto gramágico: Um jogo educativo para o ensino de gramáticas regulares. In: *SBC. Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*. [S.l.], 2021. p. 489–498.

GENESIO, N.; OLIVEIRA, A.; CHAVES, L.; SILVA, W.; VALLE, P. Um mapeamento sistemático sobre jogos educacionais digitais para o ensino-aprendizagem de estrutura de dados. In: *Anais do XXXV Simpósio Brasileiro de Informática na Educação*. Porto Alegre, RS, Brasil: SBC, 2024. p. 819–837. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbie/article/view/31287>.

GENESIO, N. O. S.; OLIVEIRA, A. M. de; OLIVEIRA, E. W.; VALLE, P. H. D. Panorama de estudos sobre jogos educacionais digitais em educação em computação. In: SBC. *Workshop sobre Educação em Computação (WEI)*. [S.l.], 2024. p. 737–749.

GROSSKURTH, A.; GODFREY, M. W. A reference architecture for web browsers. In: IEEE. *21st IEEE International Conference on Software Maintenance (ICSM'05)*. [S.l.], 2005. p. 661–664.

HONDA, F.; PIRES, F.; PESSOA, M.; MELO, R. Aplicando learning design na ludificação de percurso em grafos: uma jornada de aprendizagem. In: SBC. *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*. [S.l.], 2022. p. 609–620.

JÚNIOR, A. A.; MISRA, S.; SOARES, M. S. A systematic mapping study on software architectures description based on iso/iec/ieee 42010: 2011. In: SPRINGER. *Computational Science and Its Applications–ICCSA 2019: 19th International Conference, Saint Petersburg, Russia, July 1–4, 2019, Proceedings, Part V 19*. [S.l.], 2019. p. 17–30.

JÚNIOR, E. W.; FARIAS, K.; SILVA, B. da. On the use of uml in the brazilian industry: A survey. *Journal of Software Engineering Research and Development*, v. 10, p. 10–1, 2022.

KAZMAN, R.; KLEIN, M.; BARBACCI, M.; LONGSTAFF, T.; LIPSON, H.; CARRIERE, J. The architecture tradeoff analysis method. In: IEEE. *Proceedings. fourth ieee international conference on engineering of complex computer systems (cat. no. 98ex193)*. [S.l.], 1998. p. 68–78.

KRAMARZEWSKI, A.; NUCCI, E. D. *Practical game design: Learn the art of game design through applicable skills and cutting-edge insights*. [S.l.]: Packt Publishing Ltd, 2018.

KRUCHTEN, P. *The rational unified process: an introduction*. [S.l.]: Addison-Wesley Professional, 2004.

KRUCHTEN, P.; OBBINK, H.; STAFFORD, J. The past, present, and future for software architecture. *IEEE software*, IEEE, v. 23, n. 2, p. 22–30, 2006.

KRUCHTEN, P. B. The 4+ 1 view model of architecture. *IEEE software*, IEEE, v. 12, n. 6, p. 42–50, 1995.

KRZYZANOWSKI, L.; JR, C. B.; JR, R. S.; TOSTES, R. A. Ensino de programação: um estudo preliminar nos cursos de licenciatura em computação no brasil. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2019. v. 8, n. 1, p. 21.

LAND, R. A brief survey of software architecture. *Mälardalen Real-Time Research Center (MRTC) Report*, Citeseer, p. 1–15, 2002.

- LAUTERT, C.; PIRES, C. C. K.; LAGO, M. Jogos digitais como estratégia pedagógica para a compreensão dos conceitos matemáticos: Um estudo de caso no ensino fundamental-anos finais. In: SBC. *Anais do XXIX Workshop de Informática na Escola*. [S.l.], 2023. p. 1262–1272.
- LIU, F.; TONG, J.; MAO, J.; BOHN, R.; MESSINA, J.; BADGER, L.; LEAF, D. et al. Nist cloud computing reference architecture. *NIST special publication*, v. 500, n. 2011, p. 1–28, 2011.
- LOPES, W.; FERNANDES, I.; AUGUSTO, P.; MADEIRA, C. A. G. Estratégia de gamificação aplicada ao ensino remoto emergencial em tempos de covid-19. In: SBC. *Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*. [S.l.], 2021. p. 437–444.
- MACENA, J.; PIRES, F.; PESSOA, M.; MELO, R. Hello food: um jogo para praticar conceitos de algoritmos para iniciantes na computação. In: SBC. *Anais Estendidos do XXI Simpósio Brasileiro de Jogos e Entretenimento Digital*. [S.l.], 2022. p. 1066–1075.
- MACHADO, L. D. P.; BERKENBROCK, C. D. M.; BIANECK, G. A.; SIPLE, I. Z. Uma ferramenta colaborativa para apoiar a aprendizagem de programação de computadores. *Revista Brasileira de Computação Aplicada*, v. 10, n. 1, p. 23–29, 2018.
- MACKENZIE, C. M.; LASKEY, K.; MCCABE, F.; BROWN, P. F.; METZ, R.; HAMILTON, B. A. Reference model for service oriented architecture 1.0. *OASIS standard*, v. 12, n. S18, p. 1–31, 2006.
- MADEIRA, C.; SERRANO, P. H. Games e gamificação na educação: alternativas para aumentar o sucesso das soluções. *Revista Tecnologias na Educação*, v. 33, n. 21, p. 1–21, 2020.
- MARTINS, V. F.; RODRIGUES, B.; KNIHS, E. Uso de jogos para o ensino de programação: Relato de experiência. In: *Proceedings of XX CLEI-Conferencia Latinoamericana de Informática*. [S.l.: s.n.], 2019.
- MAYER, R.; VARELA, P.; ALBONICO, M.; ROHLING, A.; STEFFEN, V. Experiências de um jogo educacional digital para auxiliar no processo de ensino-aprendizagem de transformações químicas para o ensino médio. In: SBC. *Anais do XXVIII Workshop de Informática na Escola*. [S.l.], 2022. p. 59–67.
- MEDVIDOVIC, N.; ROSENBLUM, D. S.; REDMILES, D. F.; ROBBINS, J. E. Modeling software architectures in the unified modeling language. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, ACM New York, NY, USA, v. 11, n. 1, p. 2–57, 2002.
- MEDVIDOVIC, N.; ROSENBLUM, D. S.; TAYLOR, R. N. A language and environment for architecture-based software development and evolution. In: *Proceedings of the 21st international conference on Software engineering*. [S.l.: s.n.], 1999. p. 44–53.
- MELO, R.; PIRES, F.; LIMA, P.; PESSOA, M.; OLIVEIRA, D. B. F. de. Metodologias para a criação de jogos educacionais: um mapeamento sistemático da literatura. *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, SBC, p. 572–581, 2020.
- MIZUTANI, W. K.; DAROS, V. K.; KON, F. Software architecture for digital game mechanics: A systematic literature review. *Entertainment Computing*, Elsevier, v. 38, p. 100421, 2021.

- MIZUTANI, W. K.; KON, F. Unlimited rulebook: A reference architecture for economy mechanics in digital games. In: IEEE. *2020 IEEE International Conference on Software Architecture (ICSA)*. [S.l.], 2020. p. 58–68.
- MORICONI, M.; QIAN, X.; RIEMENSCHNEIDER, R. A. Correct architecture refinement. *IEEE transactions on software engineering*, IEEE, v. 21, n. 4, p. 356–372, 1995.
- MULLER, G. A reference architecture primer. *Eindhoven Univ. of Techn., Eindhoven, White paper*, p. 24, 2008.
- MULLER, G. Right sizing reference architectures; how to provide specific guidance with limited information. In: WILEY ONLINE LIBRARY. *INCOSE International Symposium*. [S.l.], 2008. v. 18, n. 1, p. 2047–2054.
- NAKAGAWA, E. Y. *Uma contribuição ao projeto arquitetural de ambientes de engenharia de software*. Tese (Doutorado) — Universidade de São Paulo, 2006.
- NAKAGAWA, E. Y.; ANTONINO, P. O. An overview of reference architectures. *Reference Architectures for Critical Domains: Industrial Uses and Impacts*, Springer, p. 5–15, 2022.
- NAKAGAWA, E. Y.; GUESSI, M.; MALDONADO, J. C.; FEITOSA, D.; OQUENDO, F. Consolidating a process for the design, representation, and evaluation of reference architectures. In: IEEE. *2014 IEEE/IFIP Conference on Software Architecture*. [S.l.], 2014. p. 143–152.
- NAKAGAWA, E. Y.; MARTINS, R. M.; FELIZARDO, K.; MALDONADO, J. C. Towards a process to design aspect-oriented reference architectures. In: *Proceedings of the XXXV Latin American Informatics Conference (CLEI 2009)*. [S.l.: s.n.], 2009. p. 1–10.
- NAKAGAWA, E. Y.; OQUENDO, F.; BECKER, M. Ramodel: A reference model for reference architectures. In: IEEE. *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*. [S.l.], 2012. p. 297–301.
- NASCIMENTO, R.; ISHITANI, L. Elementos presentes em jogos eletrônicos que motivam meninas a não abandonarem um jogo. *SBC-Proceedings of SBGames— ISSN*, p. 2179–2259, 2021.
- OLIVEIRA, B. R.; GARCÉS, L.; LYRA, K. T.; SANTOS, D. S.; ISOTANI, S.; NAKAGAWA, E. Y. An overview of software architecture education. In: SBC. *Anais do XXV Congresso Ibero-Americano em Engenharia de Software*. [S.l.], 2022. p. 76–90.
- OLIVEIRA, C. M.; PEREIRA, R. Desenvolvimento do pensamento computacional no ensino superior em ciência da computação. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2019. v. 8, n. 1, p. 1502.
- OLIVEIRA, L. B. R. d. *Estabelecimento de uma arquitetura de referência orientada a serviços para ferramentas de teste de software*. Tese (Doutorado) — Universidade de São Paulo, 2010.
- OLIVEIRA, L. B. R. D.; FELIZARDO, K. R.; FEITOSA, D.; NAKAGAWA, E. Y. Reference models and reference architectures based on service-oriented architecture: A systematic review. In: SPRINGER. *Software Architecture: 4th European Conference, ECSA 2010, Copenhagen, Denmark, August 23-26, 2010. Proceedings 4*. [S.l.], 2010. p. 360–367.

- OLIVEIRA, R. N. R. de; CARDOSO, R. P.; BRAGA, J. C. B.; ROCHA, R. V. da. Frameworks para desenvolvimento de jogos educacionais: uma revisão e comparação de pesquisas recentes. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2018. v. 29, n. 1, p. 854.
- OSSHIRO, M.; NAKAGAWA, E. Y.; PAIVA, D. M.; LANDRE, G.; PALMA, E.; CAGNIN, M. I. Cambuci: A service-oriented reference architecture for software asset repositories. In: SPRINGER. *Information Technology-New Generations: 14th International Conference on Information Technology*. [S.l.], 2018. p. 589–595.
- PAIVA, C. A.; TORI, R. Jogos digitais no ensino: processos cognitivos, benefícios e desafios. *XVI Simpósio Brasileiro de Jogos e Entretenimento Digital*, p. 1–4, 2017.
- PETRI, G.; WANGENHEIM, C. G. von; BORGATTO, A. F. Meega+: Um modelo para a avaliação de jogos educacionais para o ensino de computação. *Revista Brasileira de Informática na Educação*, v. 27, n. 03, p. 52–81, 2019.
- PORTOCARRERO, J. M.; DELICATO, F. C.; PIRES, P. F.; COSTA, B.; LI, W.; SI, W.; ZOMAYA, A. Y. Ramses: a new reference architecture for self-adaptive middleware in wireless sensor networks. *Ad Hoc Networks*, Elsevier, v. 55, p. 3–27, 2017.
- ROCHA, R. V. da; BITTENCOURT, I. I.; ISOTANI, S. Análise, projeto, desenvolvimento e avaliação de jogos sérios e afins: uma revisão de desafios e oportunidades. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2015. v. 26, n. 1, p. 692.
- RODRIGUEZ, L. M. G. *A reference architecture for healthcare supportive home systems from a systems-of-systems perspective*. Tese (Doutorado) — Université de Bretagne Sud; Universidade de São Paulo (Brésil), 2018.
- SAMPAIO, L. P.; PEREIRA, C. P. Jogo digital educativo para auxílio a crianças com autismo. In: SBC. *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*. [S.l.], 2022. p. 597–608.
- SANTANA, B. L.; ARAUJO, L. G. J.; BITTENCOURT, R. A. Computação e sociedade: Uma proposta de educação em computação para o oitavo ano do ensino fundamental ii. In: SBC. *Anais do XXVI Workshop de Informática na Escola*. [S.l.], 2020. p. 81–90.
- SANTOS, J. F. M.; GUESSI, M.; GALSTER, M.; FEITOSA, D.; NAKAGAWA, E. Y. A checklist for evaluation of reference architectures of embedded systems (s). In: *SEKE*. [S.l.: s.n.], 2013. v. 13, p. 1–4.
- SBC. *Referenciais de Formação em Computação: Educação Básica*. 2017. Disponível em: <<https://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>>.
- SHAW, M.; GARLAN, D. Formulations and formalisms in software architecture. *Computer Science Today: Recent Trends and Developments*, Springer, p. 307–323, 2005.
- SILVA, D.; PIRES, F.; MELO, R.; PESSOA, M. Glboard: um sistema para auxiliar na captura e análise de dados em jogos educacionais. In: SBC. *Anais Estendidos do XXI Simpósio Brasileiro de Jogos e Entretenimento Digital*. [S.l.], 2022. p. 959–968.
- SILVA, J. V. da; JUNIOR, B. A. da S.; FOSS, L.; CAVALHEIRO, S. A. da C. A criação de jogos para o ensino de computação: uma análise comparativa. In: SBC. *Anais do VI Workshop-Escola de Informática Teórica*. [S.l.], 2021. p. 1–8.

- SILVA, L. da; FRANCO, M. Jogos educacionais digitais no apoio ao processo de alfabetização e letramento: Revisão sistemática da literatura. In: *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*. Porto Alegre, RS, Brasil: SBC, 2022. p. 453–462. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbie/article/view/22430>.
- SILVA, M. R. A.; FERNANDES, K. T.; FERNANDES, G. L. S. A prototipagem em papel no desenvolvimento de jogos digitais em sala de aula. In: SBC. *Anais Estendidos do XXI Simpósio Brasileiro de Jogos e Entretenimento Digital*. [S.l.], 2022. p. 572–581.
- SILVA, R. R.; FERNANDES, J.; SANTOS, R. Panorama da utilização de jogos digitais no ensino de programação no nível superior na última década: Uma revisão sistemática da literatura. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2018. v. 29, n. 1, p. 535.
- SILVA, T. R. da; CORDEIRO, J. R.; SANTOS, R. S. F. dos; SANTOS, F. G. dos; ARANHA, E. H. da S.; SILVA, F. G. Uma análise do cenário nacional do uso de jogos para o ensino e aprendizagem de computação. *XVII SBGames. Foz do Iguaçu*, p. 10, 2018.
- SOMMERVILLE, I. Software engineering 9th edition. *ISBN-10*, v. 137035152, p. 18, 2011.
- SOUZA, J. G. de; PRATES, R. O. Modelo para o desenvolvimento de jogos educacionais digitais por educadores do ensino fundamental brasileiro. In: SBC. *Anais Estendidos do XXI Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais*. [S.l.], 2022. p. 192–200.
- STARON, M.; STARON, M.; DURISIC, D. Autosar standard. *Automotive Software Architectures: An Introduction*, Springer, p. 81–116, 2017.
- STRAUSS, A.; CORBIN, J. Grounded theory methodology: An overview. Sage Publications, Inc, 1994.
- TEIXEIRA, T. da S.; BORGES, H. B.; MATOS, S. N.; LUZ, V. S. G. da; RIBEIRO, T. L. S. Pegagente: Modelagem de agentes por aprendizado de reforço em jogos educacionais. *Revista Novas Tecnologias na Educação*, v. 18, n. 2, p. 225–234, 2020.
- TENÓRIO, E. S.; RODRIGUES, C. Licenciatura em computação: uma análise dos estilos de aprendizagem dos discentes pelo modelo de felder-silverman. In: *V Congresso Nacional de Educação*. [S.l.: s.n.], 2018.
- VEGT, W. Van der; WESTERA, W.; NYAMSUREN, E.; GEORGIEV, A.; ORTIZ, I. M. Rage architecture for reusable serious gaming technology components. *International Journal of Computer Games Technology*, Wiley Online Library, v. 2016, n. 1, p. 5680526, 2016.
- VENTERS, C. C.; CAPILLA, R.; BETZ, S.; PENZENSTADLER, B.; CRICK, T.; CROUCH, S.; NAKAGAWA, E. Y.; BECKER, C.; CARRILLO, C. Software sustainability: Research and practice from a software architecture viewpoint. *Journal of Systems and Software*, Elsevier, v. 138, p. 174–188, 2018.
- VICTAL, E. R. D. N.; MENEZES, C. S. Avaliação para aprendizagem baseada em jogos: Proposta de um framework. *XIV Simpósio Brasileiro de Jogos e Entretenimento Digital*, p. 970–977, 2015.

WASSERMAN, A. I. Toward a discipline of software engineering. *IEEE software*, IEEE, v. 13, n. 6, p. 23–31, 1996.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. *Experimentation in software engineering*. [S.l.]: Springer Science & Business Media, 2012.

WOODS, E.; ERDER, M.; PUREUR, P. *Continuous Architecture in Practice: Software Architecture in the Age of Agility and DevOps*. [S.l.]: Addison-Wesley Professional, 2021.

XEXÉO, G.; TAUCEI, B. Endo-gdc: Projetando jogos educacionais. *Sociedade Brasileira de Computação*, 2021.

ZORZO, A. F.; NUNES, D.; MATOS, E. S.; STEINMACHER, I.; LEITE, J. C.; ARAUJO, R.; CORREIA, R. C.; MARTINS, S. Referenciais de formação para os cursos de graduação em computação 2017. Sociedade Brasileira de Computação, 2017.

6 Apêndices

6.1 Apêndice A Requisitos do Sistema

Esse Apêndice apresenta os 57 Requisitos do Sistema (RS) identificados durante a pesquisa, fundamentais para a automatização dos processos, atividades e tarefas dos JEDs desenvolvidos com base na RADEG.

- **RS-1:** O jogo precisa atender aos padrões e expectativas do jogador;
- **RS-2:** O jogo precisa ter uma organização visual para melhorar a experiência do jogador;
- **RS-3:** O jogo precisa ser capaz de se ajustar dinamicamente às necessidades individuais dos alunos;
- **RS-4:** Os conceitos apresentados no jogo precisam ser aprendidos;
- **RS-5:** O jogo precisa integrar novos conhecimentos e habilidades ao entendimento existente do jogador;
- **RS-6:** O jogo precisa ser visual e interativamente atraente para envolver os jogadores;
- **RS-7:** O jogo precisa incentivar e facilitar o trabalho em equipe;
- **RS-8:** O jogo precisa incluir elementos que promovam uma competição saudável;
- **RS-9:** O jogo deve garantir que os alunos possam compreender e interagir facilmente com o conteúdo do jogo;
- **RS-10:** O jogo deve ajudar os jogadores a manter o foco e o interesse;
- **RS-11:** O jogo deve facilitar a comunicação e a interação contínuas entre jogadores e recursos on-line relevantes;

- **RS-12:** O jogo precisa ter confiança dos jogadores;
- **RS-13:** O jogo deve garantir que os conceitos estejam ligados e apresentados em uma sequência coerente para apoiar uma aprendizagem eficaz;
- **RS-14:** O jogo deve oferecer desafios envolventes e significativos aos jogadores;
- **RS-15:** O jogo deve funcionar perfeitamente, sem atrasos ou travamentos;
- **RS-16:** O jogo deve estar acessível aos usuários sempre que precisarem;
- **RS-17:** O jogo deve ser divertido e envolvente, tornando o aprendizado uma experiência agradável;
- **RS-18:** O jogo deve oferecer uma variedade de conteúdos e atividades para atender a diferentes estilos e preferências de aprendizagem;
- **RS-19:** O jogo deve atingir seus objetivos educacionais, garantindo que os jogadores aprendam os conceitos;
- **RS-20:** O jogo deve utilizar recursos computacionais de forma eficiente;
- **RS-21:** O jogo deve ser adequado e apropriado ao público-alvo, atendendo a todas as normas e regulamentos educacionais;
- **RS-22:** O jogo deve envolver e cativar os jogadores ao longo da experiência de aprendizagem;
- **RS-23:** O jogo deve ser capaz de acomodar um número crescente de usuários;
- **RS-24:** O jogo deve funcionar de forma confiável e consistente sob diversas condições, sem travar ou apresentar bugs significativos;
- **RS-25:** O jogo precisa ter uma boa experiência;
- **RS-26:** O jogo deve permitir experimentação flexível e segura por parte dos jogadores;
- **RS-27:** O jogo deve ser projetado para incorporar facilmente novos conteúdos, recursos ou modificações;

- **RS-28:** O jogo precisa ser fácil de aprender a jogar;
- **RS-29:** O jogo precisa ser flexível para adoção de diferentes metas educacionais;
- **RS-30:** O enredo do jogo precisa garantir a imersão do jogador no ambiente do jogo;
- **RS-31:** O jogo deve ser capaz de atender às necessidades, preferências e ritmos de aprendizagem únicos de cada jogador;
- **RS-32:** O jogo deve envolver os jogadores, permitindo-lhes interagir com o conteúdo, fazer escolhas e realizar feedback;
- **RS-33:** Os problemas apresentados no jogo precisam ser intuitivos;
- **RS-34:** O jogo precisa ser fácil de jogar;
- **RS-35:** Os elementos do jogo precisam ser lúdicos;
- **RS-36:** O jogo deve atender a diferentes necessidades e objetivos educacionais;
- **RS-37:** O jogo precisa ter uma facilidade de manutenção;
- **RS-38:** O jogo precisa estimular a aprendizagem através da motivação;
- **RS-39:** O jogo deve incorporar diversas formas de mídia (texto, imagens, áudio, vídeo) para aprimorar o aprendizado e o envolvimento;
- **RS-40:** A interface do jogo deve ser intuitiva e fornecer caminhos claros para os jogadores se moverem pelo jogo e acessar diferentes recursos;
- **RS-41:** O jogo deve apresentar informações de forma precisa, evitando interpretações subjetivas;
- **RS-42:** O jogo precisa ser claro quanto às regras e funcionalidades;
- **RS-43:** O conteúdo e a mecânica do jogo devem se alinhar com as metas educacionais e os objetivos de aprendizagem;

- **RS-44:** O jogo deve ser projetado para rodar em diversas plataformas (desktop, mobile, Web) sem modificações significativas;
- **RS-45:** O jogo precisa ter potencial para uso em sala de aula no apoio ao ensino;
- **RS-46:** O jogo deve ser viável, eficiente e eficaz na realização dos seus objetivos educacionais;
- **RS-47:** O nível de dificuldade ou complexidade do jogo deve aumentar gradualmente à medida que o jogador avança;
- **RS-48:** O jogo deve atender aos objetivos educacionais e aos resultados de aprendizagem de forma eficaz, garantindo a qualidade;
- **RS-49:** O jogo deve fornecer documentação clara e controle de versão de todos os códigos e ativos;
- **RS-50:** O jogo deve ser estruturado para facilitar a reutilização de código, ativos e conteúdo educacional em vários módulos ou versões;
- **RS-51:** O jogo deve satisfazer as expectativas dos jogadores;
- **RS-52:** O jogo deve manter a integridade dos dados do jogador e evitar acesso não autorizado;
- **RS-53:** O conteúdo do jogo deve ser apropriado para a faixa etária alvo e respeitar as sensibilidades culturais;
- **RS-54:** O jogo deve ser fácil de entender, navegar e operar, especialmente em termos de regras e interface do usuário;
- **RS-55:** O jogo deve oferecer uma experiência confortável e agradável aos jogadores;
- **RS-56:** O jogo deve permitir a aplicabilidade prática dos conhecimentos e habilidades adquiridos;
- **RS-57:** O jogo precisa ter uma viabilidade técnica.