LINHA DE PROCESSOS PARA JOGOS SÉRIOS DIGITAIS

Marcos Vinícius Bezerra Benigno

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadores: Cláudia Maria Lima Werner
Eldânae Nogueira Teixeira

Rio de Janeiro
Maio de 2026

LINHA DE PROCESSOS PARA JOGOS SÉRIOS DIGITAIS

Marcos Vinícius Bezerra Benigno

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientadores: Cláudia Maria Lima Werner
Eldânae Nogueira Teixeira

Aprovada por: Prof. Cláudia Maria Lima Werner
Prof. Eldânae Nogueira Teixeira
Prof. Geraldo Bonorino Xexéo
Prof. Edson Alves de Oliveira Junior

RIO DE JANEIRO, RJ – BRASIL
MAIO DE 2026

Benigno, Marcos Vinícius Bezerra

LINHA DE PROCESSOS PARA JOGOS SÉRIOS DIGITAIS/Marcos Vinícius Bezerra Benigno. – Rio de Janeiro: UFRJ/COPPE, 2026.

XV, 156 p.: il.; 29, 7cm.

Orientadores: Cláudia Maria Lima Werner
Eldânae Nogueira Teixeira

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2026.

Referências Bibliográficas: p. 134 – 143.

1. Jogos Sérios. 2. Linha de Processos de Software. 3. Reutilização de Software. I. Werner, Cláudia Maria Lima *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Agradecimentos

A Deus, por ter me conduzido até aqui e por ter guiado meus passos ao longo de toda esta jornada de estudos; da graduação a este momento.

À minha mãe, Ana, ao meu pai, Gilson, e à minha irmã, Carol, pelo apoio ao longo dessa caminhada e por não deixarem de reafirmar, em cada etapa, o valor da formação.

Às minhas orientadoras, por terem conduzido com cuidado meus estudos e minha pesquisa, por terem me incentivado nos momentos de dificuldade e, sobretudo, por não terem desistido, e nem permitido que eu desistisse.

À CAPES e à Fundação COPPETEC, pelo apoio financeiro que tornou possível a minha permanência no mestrado.

Ao PESC/COPPE, pela oportunidade.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

LINHA DE PROCESSOS PARA JOGOS SÉRIOS DIGITAIS

Marcos Vinícius Bezerra Benigno

Maio/2026

Orientadores: Cláudia Maria Lima Werner
Eldânae Nogueira Teixeira

Programa: Engenharia de Sistemas e Computação

Os Jogos Sérios Digitais (JS) constituem um domínio de crescente relevância socioeconômica e educacional, cujo desenvolvimento, no entanto, permanece marcado pela ausência de uma infraestrutura metodológica sistemática e reutilizável. A literatura evidencia que os processos empregados na construção desses sistemas baseiam-se, em sua maioria, em adaptações *ad hoc* e na experiência tácita dos especialistas envolvidos, resultando em desenvolvimentos dispendiosos, pouco rastreáveis e propensos à perda de capital intelectual entre projetos. Diante dessa lacuna, este trabalho propõe a construção e a avaliação de uma Linha de Processos de Software para Jogos Sérios (LPrS-JS), fundamentada na metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016) e no método *ArchToDSSA* (VASCONCELOS e WERNER, 2011), como mecanismo para gerenciar a variabilidade inerente ao domínio e elevar a sistematicidade do desenvolvimento. A pesquisa oferece quatro contribuições principais. A primeira é um método sistemático de análise comparativa de processos heterogêneos, que operacionaliza as etapas de delimitação de escopo, captura de conhecimento, extração de elementos, definição de glossário, construção de vocabulário canônico, análise de mandatoriedade e identificação de variabilidades, culminando na modelagem de uma linha de processos de software em notação *OdysseyProcess-FEX* (TEIXEIRA, 2011). A segunda contribuição é a própria LPrS-JS, construída a partir de nove processos de desenvolvimento de JS identificados na literatura acadêmica, representando o domínio por meio de seis macrofases canônicas organizadas em torno de um vocabulário que integra papéis, artefatos, subatividades e variantes. A terceira contribuição consiste na verificação por instanciação, por meio da derivação de um processo a partir da linha com base em publicação acadêmica inédita ao corpus fundador, demonstrando a capacidade de cobertura e

de derivação da LPrS-JS. A quarta contribuição, atrelada a etapa de avaliação da linha, é um Catálogo de Processos em formato de sistema *web*, que disponibiliza os artefatos da linha em interface navegável, sem exigir familiaridade prévia com o ferramental *Odyssey*. A avaliação, conduzida com seis especialistas por meio de *survey*, indicou resultados positivos nas dimensões de clareza, completude e utilidade prática, com unanimidade no item que verifica a redução de esforço na estruturação de novos processos. A viabilidade de configuração e derivação foi reconhecida em termos de custo-benefício, embora condicionada à disponibilização de materiais de apoio e capacitação prévia. As limitações identificadas apontam para a necessidade de expansão do corpus de processos fundadores, de modo a incorporar contextos industriais e equipes com perfis artísticos, bem como para o desenvolvimento de mecanismos assistidos de derivação.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

SOFTWARE PROCESS LINE FOR DIGITAL SERIOUS GAMES

Marcos Vinícius Bezerra Benigno

May/2026

Advisors: Cláudia Maria Lima Werner
Eldânae Nogueira Teixeira

Department: Systems Engineering and Computer Science

Digital Serious Games (DSG) constitute a domain of growing socioeconomic and educational relevance, whose development, however, remains marked by the absence of a systematic and reusable methodological infrastructure. The literature evidences that the processes employed in the construction of such systems are based, for the most part, on *ad hoc* adaptations and on the tacit experience of the specialists involved, resulting in costly, poorly traceable developments prone to the loss of intellectual capital across projects. Faced with this gap, this work proposes the construction and evaluation of a Software Process Line for Serious Games (SPrL-SG), grounded in the *OdysseyProcessReuse* methodology (TEIXEIRA, 2016) and the *ArchToDSSA* method (VASCONCELOS e WERNER, 2011), as a mechanism to manage the variability inherent to the domain and to raise the systematicity of development. The research offers four main contributions. The first is a systematic method for the comparative analysis of heterogeneous processes, which operationalizes the steps of scope delimitation, knowledge capture, element extraction, glossary definition, canonical vocabulary construction, mandatoriness analysis, and variability identification, culminating in modeling a software process line using the *OdysseyProcess-FEX* notation (TEIXEIRA, 2011). The second contribution is the SPrL-SG itself, constructed from nine DSG development processes identified in the academic literature, representing the domain through six canonical macrophases organized around a vocabulary that integrates roles, artifacts, subactivities, and variants. The third contribution consists of verification by instantiation, through the derivation of a process from the line based on an academic publication not included in the founding corpus, demonstrating the coverage and derivation capacity of the SPrL-SG. The fourth contribution, tied to the line's evaluation stage, is a

Process Catalog in the form of a web system, which makes the line's artifacts available through a navigable interface, requiring no prior familiarity with the *Odyssey* toolset. The evaluation, conducted with six specialists by means of a *survey*, indicated positive results in the dimensions of clarity, completeness, and practical utility, with unanimity on the item verifying the reduction of effort in structuring new processes. The viability of configuration and derivation was recognized in terms of cost-benefit, although conditioned on the availability of supporting materials and prior training. The identified limitations point to the need for expansion of the founding process corpus, so as to incorporate industrial contexts and teams with artistic profiles, as well as to the development of assisted derivation mechanisms.

Sumário

Lista de Figuras **xiii**

Lista de Tabelas **xv**

1 Introdução **1**

1.1 Contexto e motivação 1
1.2 Problema e lacuna 3
1.3 Objetivo geral 5
1.4 Objetivos específicos 5
1.5 Questões de pesquisa (QPs) 6
1.6 Escopo e delimitações 6
1.7 Contribuições 7
1.8 Organização do texto 7

2 Fundamentação Teórica **9**

2.1 Jogos 9
2.1.1 Jogos Digitais 10
2.1.2 Jogos Sérios e Jogos Sérios Digitais 11
2.2 Processos de software 13
2.2.1 Definições e fundamentos 13
2.2.2 Perspectivas históricas e evolução dos modelos 14
2.3 Processos para o desenvolvimento de jogos digitais 15
2.3.1 Especificidades dos processos de desenvolvimento de jogos sérios digitais 16
2.4 Reutilização de artefatos de software 16
2.4.1 Linhas de Produto de Software 17
2.4.2 Linhas de Processo de Software e Variabilidade 18
2.5 Metodologia *OdysseyProcessReuse* 20
2.5.1 Engenharia de Domínio de Processos de Software (EDPS) . . 21
2.6 *ArchToDSSA* e a identificação de opcionalidades e de variabilidades . 23
2.7 Avaliação 24

2.7.1 Avaliação na *Design Science Research* 24
2.7.2 *Survey* e Consulta a Especialistas 24
2.7.3 Mensuração de Atitudes por Meio da Escala *Likert* 25
2.7.4 Estatística Descritiva 25

3 Metodologia **27**
3.1 Classificação da Pesquisa 27
3.2 Visão Geral das Etapas da Pesquisa 28
3.3 Abordagem para Construção do Método 31
3.4 Abordagem para Construção da LPrS-JS 32
3.5 Abordagem para Verificação da LPrS-JS 33
3.6 Abordagem para Avaliação 34
3.7 Ameaças à Validade 35
3.8 Escrita de relato 36

4 Método para a construção da LPrS-JS **37**
4.1 Atividade: Delimitação do escopo 40
4.2 Atividade: Captura de Conhecimento do Domínio 41
4.3 Atividade: Extração de Elementos de Processo 41
4.4 Atividade: Definição de Glossário 42
4.5 Atividade: Definição de Vocabulário Canônico 42
4.5.1 Identificação de Macrofases 43
4.5.2 Identificação de Subatividades Canônicas 44
4.5.3 Construção do Vocabulário Canônico 45
4.6 Atividade: Definição de Mandatoriedade 46
4.7 Atividade: Definição de Variabilidades 47
4.7.1 Agrupamento 47
4.7.2 Definição de variantes 49
4.8 Atividade: Modelagem 50
4.9 Fluxo de inserção de novos processos 51
4.10 Limitações, validade e validação 51

5 Construção da LPrS-JS **52**
5.1 Atividade: Delimitação do escopo 52
5.2 Atividade: Captura de Conhecimento do Domínio 53
5.3 Atividade: Extração de Elementos de Processo 59
5.4 Atividade: Definição de Glossário 60
5.5 Atividade: Definição de Vocabulário Canônico 60
5.5.1 Definição de Macrofases 60
5.5.2 Subtividades canônicas 63

5.5.3 Definição de Vocabulário Canônico 65
5.6 Atividade: Definição de Mandatoriedade 69
5.7 Atividade: Definição de Variabilidades 69
5.7.1 Agrupamento 70
5.7.2 Definição de Variantes 74
5.8 Atividade: Modelagem 78

6 Instanciação **79**
6.1 Procedimento geral de instanciação da LPrS 79
6.2 Instanciação 80
6.2.1 Descrição do processo 80
6.2.2 Fases e Condução do Processo no Artigo 81
6.2.3 Derivação da LPrS-JS 82
6.2.4 Discussão: Pontos Fortes e Pontos de Atenção da LPrS-JS Frente ao Processo 96

7 Avaliação da LPrS-JS **98**
7.1 Método de Avaliação 98
7.2 Perfil dos Participantes 99
7.3 Materiais Disponibilizados - Catálogo 99
7.3.1 Visão Modelagens *Odyssey* 100
7.3.2 Visão Macrofases 101
7.3.3 Visão Subatividades e Variabilidades de MF1 e MF2 101
7.3.4 Visão Catálogo 102
7.3.5 Visão Domínio 103
7.3.6 Visão Glossário 103
7.3.7 Visão Busca Global 104
7.4 Instrumento de Avaliação e Avaliação Piloto 104
7.5 Procedimento de Coleta de Dados e Avaliação Piloto 105
7.5.1 Piloto 106
7.6 Resultados 107
7.6.1 Perfil dos Participantes 107
7.6.2 Achados Quantitativos 111
7.6.3 Triangulação: Perfil, Dados Quantitativos e Respostas Discursivas 118
7.6.4 Síntese da Análise 122

8 Conclusão e Trabalhos Futuros **124**
8.1 Síntese das Contribuições 124
8.2 Respostas às Questões de Pesquisa 126

8.3 Limitações e Ameaças à Validade 127
8.4 Trabalhos Futuros 129
8.4.1 Expansão e Manutenção da Linha 129
8.4.2 Evolução da Modelagem 130
8.4.3 Ferramental e Apoio à Adoção 131
8.4.4 Avaliação de Impacto 132
8.5 Considerações finais 133

Referências Bibliográficas **134**

A Instrumento de Avaliação da LPrS-JS **144**

B Processos do Domínio de Jogos Sérios **149**
B.1 GAMED (ASLAN e BALCI, 2015) 149
B.2 Reha (BARANYI et al., 2019) 150
B.3 eLuna (BREIEN e WASSON, 2022) 150
B.4 Meconesis (CANO et al., 2016) 151
B.5 Urano (PRIETO et al., 2017) 152
B.6 PROGame (ALCOVER et al., 2018) 152
B.7 Exergame (PIROVANO et al., 2016) 153
B.8 QUARTIC (COWLEY, 2014) 154
B.9 iPlus (CARRIÓN-TORO et al., 2020) 155

Lista de Figuras

2.1 Principais elementos da notação *OdysseyProcess-FEX* (TEIXEIRA, 2011) 23

3.1 Visão geral das etapas da pesquisa, com entradas, saídas e correspondência com os capítulos da dissertação. 30

4.1 Visão geral do fluxo de atividades do método para a análise de processos, resultando na construção de uma Linha de Processos de Software para Jogos Sérios. 40

5.1 Modelo de características com macrofases na notação *OdysseyProcess-FEX* (TEIXEIRA, 2011) 62
5.2 Dimensão de papéis do modelo de características, apresentando suas opcionalidades e mandatoriedades. 65
5.3 Dimensão de artefatos do modelo de características, apresentando suas opcionalidades, mandatoriedades e relações de entrada e saída. . 67
5.4 Dimensão da macrofase 1 do modelo de características, apresentando suas possíveis variantes, cardinalidades, opcionalidades e mandatoriedades. 74
5.5 Dimensão da macrofase 2 do modelo de características, apresentando suas possíveis variantes, cardinalidades, opcionalidades e mandatoriedades. 75
5.6 Diagrama do Modelo de Características comportamental de alto nível considerando apenas as macrofases definidas. 78
5.7 Diagrama do Modelo de Características comportamental de alto mais baixo, considerando apenas as subatividades das macrofases 1 e 2. . . 78

7.1 Seção inicial da visão Modelagens *Odyssey*. 100
7.2 Interação em elemento de modelo de características na visão *Odyssey*. 101
7.3 Visão Macrofases, com o cursor do usuário sob o rótulo do processo iPlus, exibindo a descrição do item nativo. 101
7.4 Aba "Por Subatividade" da visão de subatividades. 102

7.5 Aba “Por processo” da visão subatividades. 102
7.6 Visão Catálogo do catálogo de processos. 103
7.7 Visão Domínio do catálogo de processos. 103
7.8 Visão do Glossário. 104
7.9 Visão de Busca Geral. 104
7.10 Nível de formação acadêmica dos participantes. 108
7.11 Área de formação principal. 109
7.12 Anos de experiência em desenvolvimento de software. 110
7.13 Anos de experiência em jogos digitais. 110
7.14 Participação em projetos de jogos sérios (multi-seleção). 111
7.15 Grau de familiaridade com LPS ou LPrS. 111
7.16 Média geral por dimensão avaliativa. 112
7.17 QP1 - Distribuição das respostas *Likert* por item (porcentagem de respondentes). 113
7.18 QP1 - Médias por item com desvio-padrão. Itens ordenados por média crescente. 113
7.19 QP2 - Distribuição das respostas *Likert* por item (% de respondentes). 114
7.20 QP2 - Médias por item com desvio-padrão. 115
7.21 QP3 - Distribuição das respostas *Likert* por item (% de respondentes). 116
7.22 QP3 - Médias por item com desvio-padrão. 116
7.23 Distribuição das respostas na avaliação geral de qualidade. 118

Lista de Tabelas

4.1 Exemplo de tabela de extração de um processo. 42
4.2 Fragmento de tabela de glossário. 42
4.3 Exemplo de definição de macrofase. 44
4.4 Exemplo de derivação de subatividades canônicas. 45
4.5 Fragmento de tabela de papéis do vocabulário canônico. 46
4.6 Fragmento de tabela de análise de mandatoriedade. 47
4.7 Tabela com agrupamento de termos nativos (variantes brutas) que participaram da criação de cada macrofase pro processo originário. . . 48
4.8 Fragmento de tabela de definição de variabilidades. 50

5.1 Trabalhos primários potenciais identificados. 55
5.2 Avaliação de qualidade dos trabalhos primários selecionados. 56
5.3 Relação de apelidos e referências dos trabalhos selecionados. 59
5.4 Mapeamento detalhado dos elementos brutos para as subatividades canônicas. 70

6.1 Elementos canônicos analisados na instância 82

7.1 Perfil dos participantes da avaliação. 108

A.1 Questionário de avaliação da LPrS-JS com especialistas. 144

Capítulo 1

Introdução

1.1 Contexto e motivação

Nas últimas décadas, a indústria de jogos digitais tem-se mostrado como um ramo dinâmico da economia global, com taxas de crescimento que ultrapassam as das indústrias cinematográficas e da música (CHUECA *et al.*, 2024). Somente em 2021, esse ramo tecnológico movimentou receitas superiores a 180 bilhões de dólares, impulsionado pela popularização de dispositivos computacionais e pela maior procura por experiências interativas (CHUECA *et al.*, 2024). A aplicação desses sistemas, no entanto, expandiu-se para além do entretenimento, alcançando relevância em áreas estratégicas como defesa, saúde e educação (ALEEM *et al.*, 2016; LAAMARTI *et al.*, 2014). Nesse movimento, destacam-se os Jogos Sérios (JS), aplicações cujo propósito transcende a diversão e o entretenimento, voltando-se para a instrução, o treinamento ou a passagem de informação aos seus usuários (LAAMARTI *et al.*, 2014; WILKINSON, 2016).

Ao integrarem imersão de sentidos, desafio e interatividade, os JS constituem ferramentas eficazes para a aprendizagem, facilitando o desenvolvimento de habilidades (OLIVEIRA *et al.*, 2020). Embora compartilhem traços com aplicações convencionais, os jogos digitais apresentam atributos diferenciados decorrentes de sua natureza criativa e do foco na promoção de experiências subjetivas (CHUECA *et al.*, 2024). Nos JS, essa complexidade é intensificada, e é preciso a mescla de atividades lúdicas com objetivos pedagógicos ou terapêuticos (BENIGNO *et al.*, 2024; LAAMARTI *et al.*, 2014; OLIVEIRA *et al.*, 2020).

Dada a natureza multidisciplinar das equipes envolvidas, a construção de JS demanda processos de desenvolvimento bem estruturados. A criação de um jogo requer colaboração entre profissionais de diferentes frentes, unindo programadores, artistas, *designers* de mídia e especialistas no domínio de aplicação, como pedagogos ou médicos (BENIGNO *et al.*, 2024; O'HAGAN *et al.*, 2014). Essa diversidade de

papéis exige comunicação constante e métodos que integrem recursos artísticos e tecnológicos sob cronogramas pré-definidos (ALEEM *et al.*, 2016). Apesar dessa necessidade, a literatura revela uma carência de processos padronizados e amplamente aceitos, resultando em desenvolvimentos *ad-hoc*, dispendiosos, com pouco potencial de repetições e propensos a erros (BENIGNO *et al.*, 2024; OLIVEIRA *et al.*, 2020; TEIXEIRA *et al.*, 2019).

Diante disso, a área da Engenharia de Software para o Desenvolvimento de Jogos tem buscado adaptar métodos já conhecidos da Engenharia de Software convencional para lidar com a volatilidade de requisitos e a dinamicidade típicas desses sistemas (BENIGNO *et al.*, 2024; CHUECA *et al.*, 2024; OLIVEIRA *et al.*, 2020). A reutilização de software, nesse contexto, surge como uma prática possível para promover a qualidade e reduzir os esforços em novos projetos, aproveitando conhecimentos tecnológicos e gerenciais consolidados (FRAKES e KANG, 2005; KIM e STOHR, 1998) - tendo apoio da afirmação de que processos de software devem ser tratados como o próprio software, sendo passíveis de descrição padronizada, execução e, sobretudo, reutilização sistemática (OSTERWEIL, 1997).

Uma técnica de reutilização que trabalha o reúso apoiando-se nas similaridades e nas variabilidades dos processos é a Linha de Processos de Software (LPrS). Baseada nos preceitos da Linha de Produtos de Software (LPS), a LPrS consiste em um conjunto de processos que compartilham características comuns e variáveis em um domínio específico, desenvolvidos a partir de ativos reutilizáveis (ROMBACH, 2006; TEIXEIRA *et al.*, 2019). A adoção de uma linha de processos permite que organizações definam seus fluxos de trabalho específicos para atender a diferentes necessidades de forma antecipada (ROMBACH, 2006). Essa estratégia mostra-se útil para o cenário dos JS, no qual cada aplicação demanda adaptações restritas a seus objetivos educacionais, clínicos ou terapêuticos (LAAMARTI *et al.*, 2014), permitindo que se mantenha uma base comum de engenharia gerenciada sistematicamente por meio de componentes e variabilidades de processo (BARRETO *et al.*, 2011).

A utilização da técnica de LPrS pode impactar diretamente a produtividade ao representar variabilidades de um domínio de forma explícita. Utilizando blocos modulares, torna-se possível derivar processos customizados para projetos específicos com menor custo e tempo (ROMBACH, 2006; TEIXEIRA *et al.*, 2019). Esse encapsulamento de informações em componentes reutilizáveis melhora a capacidade de manutenção e a compreensão do domínio, permitindo que o processo definido ajuste-se ao seu contexto particular de execução (BARRETO *et al.*, 2011).

Assim, a proposição de uma linha de processos para jogos sérios digitais apresenta-se como uma alternativa para o aproveitamento das diferentes características do domínio desse tipo de sistema, fundamentando-se na necessidade de ge-

renciar a variabilidade inerente a esse ramo de forma sistemática. A modelagem de características permite capturar os elementos comuns e variantes (TEIXEIRA *et al.*, 2019), de forma a oferecer um mecanismo formal para definir processos que garantam eficácia pedagógica e eficiência técnica. Diante da diversidade de contextos em projetos de jogos sérios, as abordagens *bottom-up*, que analisa experiências de processos anteriores, e *top-down*, que parte da observação de necessidades de um domínio, apresentam-se como caminhos possíveis para elevar a maturidade do setor em busca da consolidação de uma infraestrutura de reutilização (ROMBACH, 2006; WASHIZAKI, 2006).

Em termos mais gerais, o objeto de estudo deste trabalho é a linha de processos de software enquanto mecanismo de representação de um domínio, sendo uma estrutura que organiza, de forma abstrata e reutilizável, os elementos comuns e variáveis dos processos de um domínio, a partir da qual processos específicos podem ser derivados. Sob essa ótica, o resultado de maior alcance a que a pesquisa se dirige não é um processo particular, mas um modelo conceitual de referência do domínio de jogos sérios com macrofases, papéis, artefatos e atividades, suas obrigatoriedades e variabilidades, e capaz de servir de base comum para a definição de processos em diferentes contextos. A Linha de Processos de Software para Jogos Sérios construída neste trabalho é uma realização concreta e parcial dessa visão, representando um passo inicial, fundamentado em um conjunto de processos da literatura, em direção a um modelo de referência que se pretende progressivamente mais completo à medida que novos processos e contextos sejam incorporados.

1.2 Problema e lacuna

Os jogos sérios conquistaram relevância socioeconômica e educacional nas últimas décadas, mas seu desenvolvimento ainda possui obstáculos decorrentes da ausência de um corpo metodológico consolidado (BENIGNO *et al.*, 2024; OLIVEIRA *et al.*, 2020; XEXÉO *et al.*, 2021). BENIGNO *et al.* (2024) observam que, embora exista uma multiplicidade de opções para a construção de jogos, a literatura não oferece direcionamento sobre quais métodos são mais apropriados para diferentes contextos. Tampouco há convergência nas definições de processos para um método único. Essa fragmentação cria uma lacuna entre a Engenharia de Software tradicional e as necessidades da indústria de jogos, que exige ciclos de vida capazes de suportar a alta volatilidade de requisitos e a integração constante de ativos criativos (ALEEM *et al.*, 2016; CHUECA *et al.*, 2024).

A heterogeneidade das abordagens atuais dificulta a padronização sistemática de processos de software da área de JS. Revisões da literatura mostraram que os processos de desenvolvimento de jogos sérios baseiam-se, em sua maioria, na expe-

riência individual de especialistas ou em métodos adaptados de forma *ad-hoc* para situações específicas (OLIVEIRA *et al.*, 2020). CHUECA *et al.* (2024) apontam que a diversidade inerente ao setor, que abrange desde desenvolvedores independentes até grandes corporações, amplia o desafio de generalização das soluções. As necessidades de um jogo de ação diferem das de um jogo educativo focado em lógica, por exemplo (CHUECA *et al.*, 2024). A carência de padrões faz com que o conhecimento gerado em um projeto não seja transferido de forma estruturada para um próximo, podendo resultar em perda de capital intelectual e de experiências, conforme afirmado por FRAKES e KANG (2005) no contexto da reutilização.

A falta de rastreabilidade entre objetivos pedagógicos e mecânicas lúdicas representa outra lacuna técnica de impacto na qualidade do software. OLIVEIRA *et al.* (2020) observam que projetos de jogos educativos iniciam com um levantamento de requisitos, mas nem sempre consideram os objetivos de aprendizado ao longo de todo o processo de *design* ou os associam de maneira formal às funcionalidades do sistema. A fragilidade dos artefatos documentais potencializa essa desconexão. O *Game Design Document* (GDD), embora amplamente reconhecido como a peça importante de especificação em sistemas desse tipo, não possui um modelo universalmente aceito, o que pode causar falhas na representação da complexidade das interações pretendidas (BENIGNO *et al.*, 2024; SALAZAR *et al.*, 2012; XEXÉO *et al.*, 2021). SALAZAR *et al.* (2012) advertem, ainda, que documentos de projeto mal estruturados podem levar ao retrabalho e a perdas financeiras significativas durante as fases de produção e pós-produção.

Essa falta de padronização reflete-se na dificuldade de garantir que o produto final cumpra sua proposta híbrida de entretenimento e aprendizado. A ausência de processos sistemáticos conduz a uma inconsistência na avaliação dos resultados, tornando incerto se o jogo é eficaz tanto em termos de entretenimento quanto de instrução (LAAMARTI *et al.*, 2014). Sem um fluxo de processos definido, as equipes multidisciplinares podem enfrentar barreiras de comunicação, visto que pedagogos, artistas e programadores, muitas vezes, carecem de meios padronizados que integrem suas visões distintas e garanta a integridade do sistema (ALEEM *et al.*, 2016; BENIGNO *et al.*, 2024; O'HAGAN *et al.*, 2014). Muitas aplicações acabam por não atender às propostas originais, resultando em ferramentas que falham em engajar o usuário ou em transmitir o conhecimento para o qual foram projetadas (LAAMARTI *et al.*, 2014).

Em uma outra frente, a literatura também destaca a ineficiência na reutilização de ativos de processo como uma lacuna que impede o amadurecimento da área de reutilização de software. TEIXEIRA *et al.* (2019) afirmam que o conceito de uma infraestrutura de reutilização integrada para processos ainda é pouco explorado, e as ferramentas existentes não apoiam de forma eficiente todas as etapas de uma

linha de processos, desde a análise de similaridades até a derivação de instâncias específicas. Sem mecanismos padronizados para gerenciar a variabilidade inerente a diferentes gêneros de jogos sérios, o setor permanece em um estado de baixa previsibilidade, caracterizado por desenvolvimentos dispendiosos e propensos a falhas (ALEEM *et al.*, 2016; BENIGNO *et al.*, 2024; OLIVEIRA *et al.*, 2020).

1.3 Objetivo geral

Frente ao exposto nas subseções anteriores, este trabalho objetiva a sistematização e o amadurecimento do desenvolvimento de jogos sérios por meio da Engenharia de Linha de Processos de Software, propondo a construção e a avaliação de uma Linha de Processos de Software para Jogos Sérios (LPrS-JS) representada por modelagens de características, tendo como base, sobretudo, a metodologia *Odyssey-ProcessReuse* (TEIXEIRA, 2016). O modelo a ser desenvolvido oferece uma estrutura para representar as variabilidades e as opcionalidades características do domínio dos jogos sérios, constituindo um passo em direção a um modelo conceitual de referência do domínio, modelando uma representação abrangente dos seus processos, a partir do qual se torna possível a derivação de processos mais sistemáticos e eficientes. Ao aplicar técnicas de Engenharia de Linha de Processos de Software, pretende-se elevar a sistematicidade no desenvolvimento desses jogos, reduzindo a dependência de conhecimentos tácitos e evitando que erros técnicos e pedagógicos, já identificados em projetos anteriores, se repitam.

1.4 Objetivos específicos

O alcance do objetivo geral desdobra-se em cinco metas que estruturam o desenvolvimento desta pesquisa, desde o levantamento de conhecimento sobre o domínio até a verificação dos artefatos propostos:

(a) Mapear processos existentes na literatura para a construção de jogos sérios, atentando para as especificidades pedagógicas, terapêuticas e tecnológicas que caracterizam esse domínio;

(b) Propor um método de análise entre processos de desenvolvimento de jogos sérios digitais que permita identificar convergências e divergências nas abordagens encontradas na literatura;

(c) Modelar uma linha de processo de software, contemplando as principais características de parte do domínio identificado;

(d) Verificar a capacidade de derivação da LPrS-JS, examinando se a linha permite configurar e derivar processos coerentes e adequados às especificidades do domínio de jogos sérios;

(e) Avaliar, mediante consulta a especialistas do domínio, a clareza e a cobertura, a utilidade prática percebida e a viabilidade de configuração e derivação dos artefatos que compõem a LPrS-JS.

1.5 Questões de pesquisa (QPs)

As lacunas metodológicas identificadas na Engenharia de Software voltada ao desenvolvimento de jogos motivam questões relativas à abordagem de linha de processos para esse domínio, isto é, ao potencial de representar o domínio de jogos sérios por meio de uma linha de processos e aos benefícios dela decorrentes; e não a um artefato isolado. Essas questões são investigadas empiricamente por meio da Linha de Processos de Software para Jogos sérios construída neste trabalho, tomada como instância representativa dessa abordagem:

- QP1: A representação do domínio de jogos sérios por meio de uma linha de processos descreve o desenvolvimento desse tipo de sistema com clareza e completude?
- QP2: A adoção de uma linha de processos para o domínio de jogos sérios oferece utilidade prática e é percebida como ganho de produtividade pelos profissionais envolvidos no desenvolvimento?
- QP3: A derivação de processos específicos a partir de uma linha de processos para jogos sérios mostra-se viável considerando o esforço exigido para sua configuração e derivação?

1.6 Escopo e delimitações

Esta pesquisa delimita seu escopo ao desenvolvimento de JS em formato digital, aplicações cuja finalidade transcende o entretenimento puro, buscando instruir ou treinar seus usuários (BENIGNO *et al.*, 2024; LAAMARTI *et al.*, 2014). Estão contemplados apenas sistemas que empregam dispositivos eletrônicos como plataforma de mediação do aprendizado. Jogos analógicos e esportes físicos, portanto, não integram este estudo. A integração entre mecânicas lúdicas e objetivos pedagógicos, aspecto enfatizado por LAAMARTI *et al.* (2014), constitui o eixo da pesquisa aqui desenvolvida.

Métodos destinados exclusivamente a jogos de entretenimento, sem fins pedagógicos ou terapêuticos validados, situam-se além dos limites deste trabalho. Processos de gamificação de sistemas existentes também foram excluídos, uma vez que a gamificação opera através da inserção de elementos lúdicos isolados em contextos não

lúdicos, distanciando-se da construção de um produto de software de jogo completo e integrado (OLIVEIRA *et al.*, 2020; O'HAGAN *et al.*, 2014).

1.7 Contribuições

Este trabalho oferece avanços no amadurecimento do desenvolvimento de jogos sérios por meio da Engenharia de Linha de Processos de Software. Propõe-se um método sistemático, inspirado no método *ArchToDSSA* (VASCONCELOS e WERNER, 2011) e na metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016), para análise de processos de software, capaz de identificar similaridades e variabilidades em abordagens documentadas na literatura. Tal método viabiliza a consolidação de conhecimentos advindos de projetos anteriores e de proposições de processos, formando uma modelagem de características reutilizáveis.

A LPrS-JS representa outra contribuição. Por meio da modelagem de características, a linha gerencia as variabilidades do domínio e permite a derivação de processos customizados com menor esforço.

Uma terceira contribuição é o desenvolvimento de uma prova de conceito, que consiste na derivação de um processo a partir da LPrS-JS construída tomando como referência modelo existente na literatura acadêmica, de modo a observar sua capacidade de cobertura dos elementos. Essa atividade demonstra a viabilidade prática da linha.

Os artefatos de processo gerados, como modelos e notações gráficas, catálogo *web*, constituem ativos reutilizáveis para a comunidade de Engenharia de Software voltada ao desenvolvimento de jogos sérios.

1.8 Organização do texto

Esta dissertação está estruturada em oito capítulos que documentam o percurso metodológico da pesquisa. A introdução, apresentada neste Capítulo, estabelece o contexto, a motivação e as questões de pesquisa que orientam o estudo, assim como a organização do texto. A fundamentação teórica é desenvolvida no Capítulo 2, onde são abordados os conceitos relevantes para esta dissertação.

A metodologia utilizada para este trabalho é apresentada no Capítulo 3, seguido do método de construção utilizado para a construção da linha de processos no Capítulo 4. A construção da LPrS-JS é apresentado no Capítulo 5. O Capítulo 6 apresenta a derivação de um processo a partir da linha modelada no capítulo anterior.

A avaliação da linha de processos proposta encontra-se no Capítulo 7. A discussão dos resultados obtidos e o fechamento do trabalho com as conclusões, as

limitações identificadas e as perspectivas para investigações futuras na área são apresentados no Capítulo 8.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica deste trabalho, com a definição de conceitos relevantes para o desenvolvimento da pesquisa, de modo a embasar os desenvolvimentos dos demais capítulos. Dentre os temas abordados, temos: jogos, processos de software, processos de desenvolvimento de jogos digitais, reutilização de artefatos de software, a sistemática *OdysseyProcessReuse* (TEIXEIRA, 2016), o método *ArchToDSSA* (VASCONCELOS e WERNER, 2011) e métodos de avaliação de pesquisa.

2.1 Jogos

O conceito de jogo, embora intuitivamente entendido, possui múltiplas definições que variam conforme a disciplina e o enfoque de pesquisa. STENROS (2016) observa que a definição constitui uma questão inicial nos estudos ludológicos, com as últimas décadas produzindo diversas propostas conceituais, gerando uma pluralidade que reflete a complexidade do objeto, mais do que uma convergência (STENROS, 2016). Essa heterogeneidade de conceitos pode ser atribuída a natureza multidisciplinar do campo, onde diferentes tradições teóricas buscam capturar um fenômeno dinâmico e contextual, servindo funções explicativas ou analíticas conforme o problema que está em investigação (ARJORANTA, 2019).

Sob uma perspectiva histórica, o jogo tem sido reconhecido como componente integrante das sociedades e fator determinante no desenvolvimento civilizacional. Dessa maneira, HUIZINGA (1955) caracteriza o jogo como função fundamental que precede a cultura, definindo-o como atividade voluntária executada dentro de limites fixos de tempo e espaço, segundo regras livremente aceitas mas obrigatórias, acompanhada por sentimentos de tensão e alegria (HUIZINGA, 1955). Para esse autor, a civilização não apenas emerge do jogo, mas se desenrola como jogo, o que denomina *sub specie ludi* (HUIZINGA, 1955). WILKINSON (2016) complementa essa perspectiva ao destacar que, desde o século XIX, essa prática do jogo tem sido assu-

mida como impulsionadora do desenvolvimento, reforçando comportamentos sociais e funcionando como meio de experimentação de realidades (WILKINSON, 2016).

A categorização tipológica dos jogos tem sido abordada por meio de estruturas comportamentais e funcionais. Isso pode ser visto em CAILLOIS (2001), que propõe quatro categorias fundamentais baseadas no impulso predominante: *agon* (competição), *alea* (sorte), *mimicry* (simulação) e *ilinx* (vertigem) (CAILLOIS, 2001; SALEN TEKINBAŞ e ZIMMERMAN, 2003). Essa classificação, ainda, situa os jogos em *algo* entre os conceitos de *paidia*, que significa exuberância espontânea e improvisação livre, e *ludus*, que é caracterizado pela subordinação a convenções arbitrárias e regras imperativas (CAILLOIS, 2001). SALEN e ZIMMERMAN (2003), por sua vez, sintetizam essa discussão definindo o jogo como um sistema no qual participantes se envolvem em conflito artificial, regrado, resultando em desfecho quantificável (SALEN TEKINBAŞ e ZIMMERMAN, 2003).

As perspectivas apresentadas, desde a visão antropológica sobre o jogo como fundamento cultural, passando pela classificação comportamental até a sua definição, convergem em aspectos fundamentais: a voluntariedade, a delimitação por regras e a geração de experiências significativas. Esta compreensão do jogo é essencial para examinar sua transferência ao meio digital, que preserva esses elementos fundamentais enquanto introduz possibilidades de interação.

2.1.1 Jogos Digitais

Estabelecidas as bases conceituais do jogo como fenômeno cultural e social, torna-se necessário examinar como essas características manifestam-se quando mediadas pela tecnologia digital. Os jogos digitais representam a transferência da experiência lúdica tradicional para o meio computacional, preservando características fundamentais do jogo enquanto incorporam capacidades tecnológicas específicas. XEXÉO *et al.* (2021) definem os jogos digitais como atividades de entretenimento que utilizam dispositivos eletrônicos, como computadores e celulares (BENIGNO *et al.*, 2024; XEXÉO *et al.*, 2021). Complementarmente, na perspectiva da Ciência da Computação, tais jogos constituem aplicações de software complexas que viabilizam a interação entre jogador e máquina através de interface visual (TAVINOR, 2008).

Dessa maneira, a distinção entre o jogo analógico e o digital reside, em primeiro lugar, na automação de sistemas complexos e na manipulação de informações através do suporte eletrônico. Enquanto em jogos tradicionais os próprios jogadores mantêm e aplicam as regras, nos jogos digitais, o computador atua como executor e árbitro, possibilitando níveis de complexidade estrutural que não são possíveis em meios físicos (SALEN TEKINBAŞ e ZIMMERMAN, 2003; SCHELL, 2014).

Em uma ótica mercadológica, o consumo de *videogames* tem demonstrado cres-

cimento, estabelecendo-se como a maior indústria de entretenimento em termos de receita. CHUECA *et al.* (2024) apontam que a indústria de *videogames* ultrapassou as receitas dos setores cinematográfico e musical, refletindo demanda crescente por experiências interativas (CHUECA *et al.*, 2024). Foi observado que os *videogames*, ainda, integraram-se ao cotidiano humano, o que foi impulsionado pela expansão do acesso a dispositivos móveis e ferramentas de desenvolvimento (BENIGNO *et al.*, 2024; CHUECA *et al.*, 2024; WHITSON, 2019). No contexto da composição dos jogos digitais, SCHELL (2014) identifica a "tétrada elemental"como elementos essenciais: mecânicas (procedimentos e regras), história (sequência de eventos), estética (aspectos sensoriais) e tecnologia (meio que possibilita o jogo) (SCHELL, 2014); o que decompõe o jogo digital para uma melhor compreensão.

2.1.2 Jogos Sérios e Jogos Sérios Digitais

Observando os jogos sob uma perspectiva utilitária, existe o conceito de jogos sérios, que constituem uma categoria projetada com finalidade que transcende o entretenimento. ABT (1970) define o termo como jogos com propósito educacional explícito planejado, não destinados unicamente à diversão, embora devam manter o engajamento do usuário (ABT, 1970; MOUAHEB *et al.*, 2012). Essa definição é estendida ao canal computacional, definindo o jogo sério digital como a combinação de aspectos "sérios", ensino, aprendizagem ou comunicação, com facilitadores lúdicos provenientes dos *videogames* (ALVAREZ e DJAOUTI, 2011). Dessa maneira, a diferenciação entre jogos sérios e jogos digitais sérios reside no canal: o primeiro termo abrange jogos de tabuleiro ou RPGs de mesa com fins pedagógicos, por exemplo, enquanto o segundo, refere-se especificamente à implementação dessas finalidades em ambiente digital (ALVAREZ e DJAOUTI, 2011; LAAMARTI *et al.*, 2014).

Portanto, esse tipo de aplicação foca na transformação do destinatário, seja na melhoria de habilidades, compreensão de fenômenos ou mudança comportamental. Através de teorias pedagógicas construtivistas, os jogos sérios garantem a motivação e proporcionam a aprendizagem em ambientes desafiadores e realistas (DE GLORIA *et al.*, 2014; MOUAHEB *et al.*, 2012). Diversos domínios de aplicação são possíveis nesse tipo de jogo, como a educação formal, saúde, treinamento militar, reabilitação física, planejamento urbano e publicidade (LAAMARTI *et al.*, 2014). Ademais, os jogos digitais sérios funcionam como ferramentas para ensino-aprendizagem também por permitirem simulação de situações complexas ou perigosas de forma segura e com custo reduzido comparado ao treinamento real (DE GLORIA *et al.*, 2014; OLIVEIRA *et al.*, 2020).

Essa classificação desse tipo de jogo como categoria separada dos jogos comuns decorre da inclusão de dimensão pedagógica ausente nos jogos comerciais convenci-

onais. BREUER e BENTE (2010) contrastam que, enquanto o objetivo de um jogo comum é o entretenimento, o jogo sério envolve pedagogia, atividades que instruem e transmitem conhecimento ou habilidade como meta primária (BREUER e BENTE, 2010; LAAMARTI *et al.*, 2014). Distingue-se também dos jogos sérios a gamificação, que caracteriza-se como a adição de elementos isolados de jogos, mecânicas de pontos e *rankings* a atividades de aprendizagem não-lúdicas para aumentar o engajamento, enquanto o jogo sério constitui experiência de jogo completamente dedicada para fins educacionais (LANDERS e CALLAN, 2011). Por fim, DE GLORIA *et al.* (2014) observam que o *design* de jogos sérios exige equilíbrio entre demandas de um jogo convencional adicionado da camada de requisitos para medir e demonstrar resultados de aprendizagem específicos (DE GLORIA *et al.*, 2014), tornando-o um artefato de maior complexidade.

Para além de sua definição conceitual, os jogos sérios digitais consolidaram-se como campo de estudo estabelecido e como segmento da indústria, com crescimento no volume de publicações acadêmicas e no mercado ao longo das últimas décadas LAAMARTI *et al.* (2014); WILKINSON (2016). Esse campo caracteriza-se por sua natureza interdisciplinar, que reúne contribuições das ciências sociais, da psicologia, da educação e da computação, fator que enriquece a área mas também origina tensões expressas na multiplicidade de definições, na diversidade de métodos de avaliação e na variedade de conceituações WILKINSON (2016). Sob a ótica da computação, o desenvolvimento desse tipo de aplicação insere-se na Engenharia de Software para o Desenvolvimento de Jogos, subárea que adapta métodos consolidados da Engenharia de Software convencional às particularidades do domínio, sem que se observe, no entanto, convergência quanto a uma metodologia única adequada aos diferentes contextos de projeto BENIGNO *et al.* (2024); OLIVEIRA *et al.* (2020).

Diante da ausência de um método consolidado, a caracterização da Engenharia de Software para jogos sérios digitais apoia-se em instrumentos diversos. No plano da classificação, propõem-se taxonomias que organizam os jogos sérios segundo atividade, modalidade, estilo de interação, ambiente e área de aplicação LAAMARTI *et al.* (2014). No plano do ciclo de vida, tendo como base o desenvolvimento de jogos convencionais, observa-se a organização do desenvolvimento em macrofases de pré-produção, produção e pós-produção ALEEM *et al.* (2016), apoiada por atividades recorrentes como a definição inicial de objetivos, a prototipagem e o teste de jogabilidade em ciclos iterativos ALEEM *et al.* (2016); BENIGNO *et al.* (2024); OLIVEIRA *et al.* (2020). No plano do projeto, reúnem-se diretrizes e fatores de sucesso extraídos da literatura, com destaque para a engenharia de software centrada no usuário, a participação de equipes multidisciplinares e de representantes do público-alvo, e a incorporação de modelos instrucionais ao design do jogo BENIGNO *et al.* (2024); LAAMARTI *et al.* (2014); OLIVEIRA *et al.* (2020). Em

todos esses planos, persiste o desafio do balanceamento do componente lúdico e do propósito sério, de modo que o entretenimento atue como meio para o alcance dos objetivos pretendidos LAAMARTI *et al.* (2014).

2.2 Processos de software

A Engenharia de Software consolidou-se a partir da necessidade de conferir rigor técnico e previsibilidade à produção de sistemas computacionais. Essa disciplina caracteriza-se pela aplicação de abordagens sistemáticas, disciplinadas e quantificáveis ao ciclo de vida dos produtos tecnológicos (ALSAQQA *et al.*, 2020). Nessa perspectiva, o processo de software representa o conjunto de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos empregados na concepção, desenvolvimento, implantação e manutenção de um produto de software (ALSAQQA *et al.*, 2020; FUGGETTA, 2000). Tais estruturas fornecem um roteiro capaz de minimizar esforços e custos desnecessários, transformando o desenvolvimento em um esforço coletivo e gerenciável, distante da assistematicidade (ALSAQQA *et al.*, 2020), sendo a qualidade do resultado final vinculada à organização das atividades e aos métodos empregados pela equipe durante a construção do sistema (ALSAQQA *et al.*, 2020; FUGGETTA, 2000).

2.2.1 Definições e fundamentos

A premissa da disciplina de processos estabelece uma correlação entre a qualidade do processo de desenvolvimento e a qualidade do produto final (FUGGETTA, 2000). Ao estruturar o ciclo de vida, as organizações buscam produzir sistemas mais confiáveis e de manutenção menos dispendiosa, especialmente em cenários marcados por funcionalidades de complexidade crescente (ALSAQQA *et al.*, 2020; MATHARU *et al.*, 2015). A formalização dos processos visa substituir práticas informais de codificar e reparar, frequentemente associadas à imprevisibilidade, por métodos que garantam conformidade com requisitos e controle sobre cronogramas e orçamentos (ADENOWO e ADENOWO, 2020; AND PRABOWO *et al.*, 2022).

OSTERWEIL (1997) introduziu um conceito base ao propor que processos de software são, em essência, software (FUGGETTA, 2000; OSTERWEIL, 1997). Segundo essa tese, as descrições de processos podem ser formalizadas e programadas mediante linguagens de modelagem que incorporam conceitos típicos de algoritmos convencionais (OSTERWEIL, 1997). Essa perspectiva permite que os processos sejam analisados, testados, automatizados e evoluídos continuamente, tratando a gestão do desenvolvimento como uma disciplina de programação em um domínio não tradicional, onde os objetos manipulados são especificações, códigos e casos

de teste (OSTERWEIL, 1997). A materialização do processo em modelos formais possibilita a detecção precoce de inconsistências e o maior aproveitamento do conhecimento gerencial, permitindo que práticas eficazes sejam preservadas e reutilizadas pela organização (OSTERWEIL, 1997).

2.2.2 Perspectivas históricas e evolução dos modelos

Os modelos de processo evoluíram de estruturas sequenciais para abordagens dinâmicas e adaptativas, refletindo a busca da indústria por maior flexibilidade diante da volatilidade de requisitos (MATHARU *et al.*, 2015; ROYCE, 2021). O modelo cascata, cujas bases apontam para o trabalho de ROYCE (1970), estabeleceu uma abordagem linear e sucessiva composta por especificação de requisitos, projeto de sistema e software, implementação e testes, operação e manutenção (SOMMERVILLE, 1996). Embora facilite o controle gerencial de recursos, sua rigidez revela-se problemática em ambientes dinâmicos. A ausência de ciclos de *feedback* entre as etapas dificulta mudanças após o congelamento das especificações iniciais, resultando na descoberta tardia de erros críticos e no desalinhamento com as necessidades reais dos usuários (ROYCE, 2021; SOMMERVILLE, 1996).

As limitações das abordagens puramente sequenciais motivaram o surgimento de modelos voltados para rapidez e interatividade, como o Desenvolvimento Rápido de Aplicações. Proposto por MARTIN (1991), o RAD prioriza a entrega acelerada de sistemas funcionais mediante equipes pequenas, construção de protótipos e participação dos usuários (BEYNON-DAVIES *et al.*, 1999; MARTIN, 1991). O método introduziu o conceito de *timeboxing*, segundo o qual o escopo do projeto é ajustado para caber em prazos fixos, servindo de intermediário entre a rigidez do modelo cascata e a ausência de processos formais (MARTIN, 1991). O RAD admite que o sistema seja construído em pequenos incrementos, permitindo que os usuários validem versões iniciais e refinem os requisitos futuros de forma iterativa (MARTIN, 1991; SOMMERVILLE, 1996).

O movimento ágil, formalizado pelo Manifesto Ágil em 2001, consolidou o paradigma adaptativo (ABRAHAMSSON *et al.*, 2002; BECK *et al.*, 2001). Essa mudança de orientação valoriza indivíduos e interações mais que processos e ferramentas, e software em funcionamento mais que documentação abrangente (BECK *et al.*, 2001; FUGGETTA e DI NITTO, 2014). Métodos como *Scrum* (SCHWABER, 2004) e *Extreme Programming* (BECK, 1999) fundamentam-se em ciclos curtos de desenvolvimento, as iterações ou *sprints*, que promovem a mitigação de riscos e a capacidade de resposta imediata a mudanças (ALSAQQA *et al.*, 2020; DINGSØYR *et al.*, 2012; MATHARU *et al.*, 2015). Enquanto os modelos tradicionais baseiam-se em uma lógica preditiva, orientada a planos estáveis, as abordagens ágeis reconhe-

cem a incerteza intrínseca à construção de software, utilizando o *feedback* contínuo do cliente como o principal direcionador da qualidade e do sucesso do produto (ALSAQQA *et al.*, 2020; FUGGETTA e DI NITTO, 2014; MATHARU *et al.*, 2015).

2.3 Processos para o desenvolvimento de jogos digitais

Os jogos digitais, embora sejam produtos de software, apresentam características que exigem métodos de construção distintos daqueles estabelecidos pela Engenharia de Software convencional (BENIGNO *et al.*, 2024). A natureza multidisciplinar desses sistemas integra áudio, arte, inteligência artificial e fatores humanos, demandando abordagens que priorizam experiências interativas e entretenimento em detrimento da busca estrita por produtividade (ALEEM *et al.*, 2016; MURPHY-HILL *et al.*, 2014). A divergência fundamental reside nos requisitos subjetivos e próprios do domínio, como diversão e sentimento, que dificilmente podem ser descritos de maneira técnica ou mensurável como ocorre em aplicações de negócio (O'HAGAN *et al.*, 2014).

A gestão de requisitos em projetos de jogos caracteriza-se por volatilidade acentuada, com o *design* ocorrendo frequentemente de forma simultânea à implementação. Esse cenário requer processos flexíveis e adaptativos, favorecendo metodologias ágeis e híbridas que suportem mudanças constantes e ciclos de *feedback* contínuos (MURPHY-HILL *et al.*, 2014; O'HAGAN *et al.*, 2014). A estrutura das equipes também diferencia o processo, marcada por heterogeneidade de papéis que exige dos profissionais capacidade de comunicação com especialistas de outras áreas (MURPHY-HILL *et al.*, 2014; O'HAGAN *et al.*, 2014).

Entre os artefatos de desenvolvimento, destaca-se o *Game Design Document* (GDD), que serve como guia central para a visão criativa (BENIGNO *et al.*, 2024). Observa-se, ainda, hesitação no uso de testes automatizados devido à sua obsolescência acelerada perante mudanças criativas, elevando a importância do teste de jogabilidade humano e do uso de protótipos funcionais como ferramentas de validação (MURPHY-HILL *et al.*, 2014). Parâmetros subjetivos como o divertimento do jogo tornam-se requisitos críticos, tendo de ser observados por meio de instrumentação e coleta de métricas de experiência do usuário ao longo de todo o ciclo de vida (BENIGNO *et al.*, 2024).

2.3.1 Especificidades dos processos de desenvolvimento de jogos sérios digitais

No subdomínio dos jogos sérios, o processo incorpora camadas adicionais de complexidade em razão de seu propósito pedagógico ou utilitário explícito, que deve coexistir com o componente de entretenimento (BENIGNO *et al.*, 2024; LAAMARTI *et al.*, 2014). A construção dessas aplicações exige integração sistemática de teorias de aprendizagem e *design* instrucional ao fluxo de engenharia de software (OLIVEIRA *et al.*, 2020). O sucesso depende do equilíbrio entre diversão e objetivo sério, de modo que a experiência lúdica sirva como meio pelo qual o aprendizado é atingido (LAAMARTI *et al.*, 2014).

Ciclos iterativos e prototipagem tornam-se de importância para explorar ideias e ajustar o equilíbrio entre desafio lúdico e transmissão de conhecimento (OLIVEIRA *et al.*, 2020). Essa realidade demanda métodos de avaliação híbridos, que permitam verificar tanto a jogabilidade quanto o cumprimento de metas instrucionais. A diferenciação de *stakeholders* é mais acentuada nestes processos, exigindo a inclusão de especialistas no domínio do conhecimento e de representantes do público-alvo para ciclos constantes de validação. O desenvolvimento de jogos sérios acrescenta, portanto, a obrigação de vincular mecânicas de interação diretamente aos resultados comportamentais ou educacionais pretendidos, utilizando métodos para monitorar a eficácia pedagógica (BENIGNO *et al.*, 2024).

Os processos para jogos digitais e jogos sérios consideram métodos comuns da Engenharia de Software, mas requerem adaptações conforme as diferenciações dos domínios e dos projetos específicos (BENIGNO *et al.*, 2024). Existe, em razão disso, o risco de perda de informação entre processos, uma vez que a ausência de um padrão único para o desenvolvimento dificulta a consolidação de práticas estáveis (BENIGNO *et al.*, 2024; MURPHY-HILL *et al.*, 2014).

2.4 Reutilização de artefatos de software

A reutilização de software caracteriza o processo pelo qual recursos desenvolvidos previamente, provenientes de qualquer fase do ciclo de vida, são empregados na construção de novas aplicações (KIM e STOHR, 1998). FRAKES e KANG (2005) descrevem essa prática como o uso sistemático de ativos de software ou conhecimento relacionado para a criação de novos sistemas, sendo a reusabilidade uma propriedade que expressa a probabilidade de um ativo ser efetivamente utilizado novamente (FRAKES e KANG, 2005). A área consolidou-se historicamente a partir da Conferência da OTAN de 1968, com a proposição da transição para uma indústria de componentes produzidos em massa como resposta à crise do software (KRUEGER,

1992; STANDISH, 1984). Desde então, o setor evoluiu de abordagens centradas na recuperação de código para métodos sistemáticos e estruturados, como as linhas de produto e os ecossistemas de software (COSTA *et al.*, 2018).

O potencial de reutilização em uma organização relaciona-se à maturidade de seus processos e à capacidade de identificar ativos com comportamento estável em contextos similares (BARRETO *et al.*, 2011). BABAR *et al.* (2010) demonstram que a adoção sistemática do reuso visa à redução de custos de desenvolvimento, à diminuição do tempo de lançamento no mercado e ao incremento da qualidade final do produto (BABAR *et al.*, 2010), isso em razão de os artefatos passarem por sucessivos processos de validação e melhoria a cada projeto. A literatura destaca ainda que a prática promove maior previsibilidade e produtividade na definição de processos e produtos (COSTA *et al.*, 2018).

É importante pontuar que a reutilização não se restringe ao código-fonte, conforme observado por FRAKES e TERRY (1996); diversos artefatos de software são passíveis de reuso, incluindo arquiteturas, modelos de projeto, especificações de requisitos, planos de teste, estimativas, documentação e interfaces humanas (FRAKES e TERRY, 1996). KIM e STOHR (1998) prosseguem com essa ótica, ao indicar que o reuso de conhecimento abstrato e artefatos de *design* oferece um nível mais avançado comparado à simples reutilização de módulos de código, reduzindo o esforço intelectual necessário para transpor o conceito inicial de um sistema para sua implementação final (KIM e STOHR, 1998; KRUEGER, 1992).

2.4.1 Linhas de Produto de Software

A Engenharia de Linha de Produto de Software (LPS) representa uma das abordagens mais consolidadas para a reutilização sistemática. BABAR *et al.* (2010) caracterizam uma LPS como um conjunto de sistemas intensivos em software que compartilham características comuns para atender às necessidades de um segmento de mercado ou missão específica (BABAR *et al.*, 2010). A estratégia fundamenta-se na construção de produtos individuais a partir de uma base de ativos centrais, projetados para serem compartilhados por todo o portfólio da organização (BACHMANN e CLEMENTS, 2005).

O gerenciamento eficaz das similaridades e variabilidades entre os sistemas determina o sucesso da aplicação de linhas de produto. As similaridades representam as características estáveis e comuns a todos os membros da linha, enquanto as variabilidades definem os pontos em que os produtos podem diferir para atender requisitos particulares de cada projeto (COSTA *et al.*, 2018). O tratamento adequado das variabilidades permite que a LPS ofereça a flexibilidade necessária sem comprometer a integridade dos artefatos centrais (BOSCH *et al.*, 2002).

Essa abordagem divide-se em dois processos distintos e interdependentes. A Engenharia de Domínio, frequentemente denominada desenvolvimento *para* reutilização, concentra-se na definição e construção dos ativos reutilizáveis da linha. Paralelamente, a Engenharia de Aplicação, ou desenvolvimento *com* reutilização, lida com a seleção e adaptação desses ativos para a criação de produtos específicos, resolvendo os pontos de variação conforme o contexto do projeto (COSTA *et al.*, 2018; ROMBACH, 2006).

A escalabilidade da produção e a melhoria da qualidade técnica através da padronização arquitetural representam vantagens do uso de LPS (CLEMENTS e NORTHROP, 2002). ROMBACH (2006) observa que organizações conseguem reduzir o risco e os custos de desenvolvimento, aumentando simultaneamente a previsibilidade sobre o comportamento dos sistemas derivados (ROMBACH, 2006).

2.4.2 Linhas de Processo de Software e Variabilidade

A abordagem de Linha de Processos de Software (LPrS) adapta os fundamentos das LPS, aplicando-os ao cenário de modelagem e administração de processos de desenvolvimento (BARRETO *et al.*, 2011; TEIXEIRA, 2011, 2016). As LPrS podem ser definidas como um conjunto de processos pertencentes a um domínio particular, ou destinados a um propósito específico, que compartilham características comuns e são construídos a partir de ativos de processos reutilizáveis (TEIXEIRA *et al.*, 2019; WASHIZAKI, 2006). A técnica é relevante com a constatação de que a diversidade de objetivos organizacionais e as particularidades de cada projeto do âmbito de software tornam a definição de processos uma tarefa complexa e dispendiosa (DE CARVALHO *et al.*, 2014; TEIXEIRA, 2011). Fluxos de trabalho, que funcionam corretamente em um projeto específico, podem ser ineficazes em outras situações, devido a requisitos divergentes, às diferentes limitações organizacionais e à diversidade de maturidade técnica dos colaboradores envolvidos (SIMMONDS e BASTARRICA, 2011; TEIXEIRA, 2011), o que evidencia a inadequação de um único modelo de desenvolvimento genérico para todos os contextos.

A variabilidade em uma LPrS constitui o elemento principal para explicitar tanto os pontos de similaridades quanto os de divergências entre os processos de uma família, demandando tratamento específico (TEIXEIRA, 2011, 2016). A abordagem propõe uma iniciativa na qual a organização prepara sistematicamente seus ativos para necessidades antecipadas, contrastando com o ajuste manual tradicionalmente conduzido de forma *ad hoc* (DE CARVALHO *et al.*, 2014; TEIXEIRA *et al.*, 2019). A adoção dessa estratégia promove incremento na produtividade da atividade de definição de processos, melhoria na qualidade e adequação dos fluxos gerados, além de mitigar riscos associados a definições inadequadas (COSTA *et al.*, 2018; TEIXEIRA,

2011). Tais benefícios derivam da capacidade de capturar o conhecimento de especialistas em módulos, permitindo gerenciar a complexidade inerente aos múltiplos projetos possíveis por meio de uma arquitetura de processos (BARRETO *et al.*, 2011; ROMBACH, 2006; TEIXEIRA, 2011).

Seguindo a lógica da Engenharia de Linha de Produtos, a LPrS organiza suas atividades em dois ciclos de desenvolvimento distintos e interdependentes (ROMBACH, 2006; TEIXEIRA, 2011). A Engenharia de Domínio de Processos de Software foca no desenvolvimento *para* reutilização, sendo responsável pela identificação, aquisição e representação do conhecimento do domínio, incluindo a especificação dos ativos centrais e de suas variabilidades (COSTA *et al.*, 2018; TEIXEIRA, 2011). A Engenharia de Aplicação de Processos de Software opera no desenvolvimento *com* reutilização, ciclo no qual os ativos da linha são selecionados, adaptados e combinados para derivação de um processo específico que atenda às demandas de um projeto particular (COSTA *et al.*, 2018; TEIXEIRA, 2011). Embora compartilhe a base conceitual da LPS, a LPrS lida com fluxos de trabalho, papéis e artefatos de processo, exigindo notações que suportem aspectos comportamentais e de execução (SIMMONDS e BASTARRICA, 2011; TEIXEIRA, 2011).

A literatura atual apresenta uma pluralidade de abordagens e metodologias para o emprego das técnicas de LPrS (COSTA *et al.*, 2018; TEIXEIRA *et al.*, 2019), variando desde métodos baseados em componentes e arquiteturas de referência até propostas focadas em anotações, operações de adaptação e modelagem de características (KUHRMANN *et al.*, 2014; TEIXEIRA, 2016; TEIXEIRA *et al.*, 2019). A evolução desta área reflete o amadurecimento da disciplina de reutilização, embora a integração de ferramentas de apoio e a dependência do conhecimento de especialistas para a resolução de variabilidades permaneçam em investigação (COSTA *et al.*, 2018; TEIXEIRA *et al.*, 2019).

Quanto à estratégia de construção, a literatura distingue três abordagens para a criação de linhas KRUEGER (2002); SILVA *et al.* (2011): a proativa, em que os ativos são desenvolvidos a priori, a partir do planejamento antecipado dos produtos previstos; a reativa, em que uma versão já existente da linha evolui de forma incremental à medida que novos requisitos surgem; e a extrativa, em que produtos, ou, no caso de uma linha de processos, processos, já existentes são analisados para deles extrair similaridades e variabilidades, derivando-se uma versão inicial da linha. A abordagem predominantemente *bottom-up* adotada nesta pesquisa alinha-se à estratégia extrativa, pois a linha é construída a partir da análise de processos previamente documentados na literatura; o componente *top-down* complementar, em que necessidades do domínio orientam a definição de pontos de variação ainda não cobertos pelas fontes, confere ao processo um caráter parcialmente proativo.

2.5 Metodologia *OdysseyProcessReuse*

A metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016) apresenta-se como uma abordagem direcionada ao reaproveitamento sistemático, fundindo os princípios das LPrSs com o paradigma de Desenvolvimento Baseado em Componentes de Processos (DBCP). Esta seção utiliza a publicação de TEIXEIRA (2016) como fonte de conteúdo para as definições desta metodologia.

A metodologia *OdysseyProcessReuse* foi concebida para preencher lacunas apontadas em estudos anteriores, especialmente a carência de uma base que permita estruturar e documentar dados reaproveitáveis, dividindo as instâncias de variação de um domínio em módulos independentes. O principal objetivo da abordagem é simplificar a localização de opções adequadas para estruturar processos à medida de cada projeto, ampliando as possibilidades de reaproveitamento.

A estrutura da *OdysseyProcessReuse* sustenta-se em cinco elementos que guiam o ciclo de vida do desenvolvimento voltado à reutilização: (1) o método para a Engenharia de Linhas de Processos de Software (ELPrS), que sistematiza as atividades de construção e uso da linha; (2) níveis de abstração complementares, que permitem o tratamento da variabilidade e da opcionalidade e representação da linha por meio de visões de características e de componentes; (3) recursos de correlação entre os artefatos nos variados níveis, utilizando regras de transformação para assegurar a consistência dessas representações; (4) diretrizes para a união de componentes, o que eleva o grau de modularidade da LPrS; e (5) táticas de verificação (tanto sintática quanto semântica) da linha criada, apoiadas em listas de inspeção dos artefatos, com o intuito de evitar que falhas se estendam aos processos gerados.

A execução da metodologia organiza-se em duas fases principais. A Engenharia de Domínio de Processos de Software (EDPS) volta-se ao desenvolvimento *para* reutilização, subdividindo-se em Análise do Domínio de Processos de Software (ADPS), que identifica similaridades e variabilidades em alto nível de abstração, e Projeto do Domínio de Processos de Software (PDPS), responsável por estruturar a linha em blocos modulares e encapsulados, denominados componentes. A Engenharia de Aplicação de Processos de Software (EAPS) foca no desenvolvimento *com* reutilização. Esta sistematização permite que o conhecimento do domínio seja extraído de variadas origens, incluindo padrões de mercado e fluxos de trabalho já estabelecidos, utilizando uma estratégia conhecida como *bottom-up* sempre que a base de dados for composta pelo histórico real da própria empresa.

A origem da *OdysseyProcessReuse* vincula-se às pesquisas do Grupo de Reutilização de Software da COPPE/UFRJ, tendo a sua implementação concretizada sob a forma de um módulo adicional ao ambiente *Odyssey*. Esta ferramenta fornece a base tecnológica exigida para desenvolver a linha através de modelos de caracterís-

ticas e componentes de processos e controlar a rastreabilidade entre os artefatos dos diferentes níveis de abstração. O recurso a variados níveis de abstração, aliado a integração de múltiplos modelos, possibilita uma abordagem em que o tratamento das dificuldades características das áreas de processo seja em etapas delimitadas, abrangendo desde as definições de caráter mais teórico até aos elementos práticos de implementação. A sistemática propõe um ambiente unificado que suporta o alinhamento entre os objetivos estratégicos da organização e as funções operacionais dos projetos de software.

2.5.1 Engenharia de Domínio de Processos de Software (EDPS)

A Engenharia de Domínio de Processos de Software (EDPS), no contexto da metodologia *OdysseyProcessReuse*, constitui o processo de desenvolvimento voltado para a reutilização, tendo como objetivo a especificação de artefatos que organizem o conhecimento de um domínio de forma a promover o reuso sistemático. Esta fase fundamenta-se na premissa de que a complexidade da definição de processos pode ser atenuada pela identificação de aspectos similares e variáveis entre os processos, permitindo que a informação seja estruturada em diferentes níveis de abstração complementares. A EDPS organiza-se operacionalmente através de um fluxo sequencial de etapas e atividades.

A Análise do Domínio de Processos de Software (ADPS) refere-se à identificação e representação do conhecimento em um plano mais alto nível e conceitual de abstração, compreendendo as seguintes atividades:

1. Delimitação do escopo do domínio;
2. Captura do conhecimento do domínio, que pode adotar uma abordagem *bottom-up*, ao extrair informações de processos legados e já executados anteriormente; e *top-down*, partindo de necessidades definidas pela organização ou modelos de referência;
3. Análise de similaridades, variabilidades e opcionalidades;
4. Modelagem de Características do Domínio, utilizando a notação *OdysseyProcess-FEX* (TEIXEIRA, 2011), para representar elementos como atividades, tarefas, papéis, produtos de trabalho e ferramentas;
5. Modelagem Contextual do Domínio, empregando a técnica *UbiFEX*, com o fim de mapear os aspectos do contexto que impactam a escolha dos elementos;
6. Identificação de subescopos do domínio (atividade opcional);

7. Inspeção do modelo de características, por meio da técnica PVMCheck (*Process Variability Modeling Checking*) (TEIXEIRA, 2016), para garantir a consistência sintática e semântica da linha desenvolvida.

O Projeto do Domínio de Processos de Software (PDPS) estrutura a linha por meio de uma visão modular e encapsulada, incluindo as seguintes atividades:

1. Construção do Modelo de Componentes de Processo a partir do mapeamento do Modelo de Características;

2. Complementação de informações dos componentes, adicionando aos componentes os seus requisitos de entrada e saída, métricas de performance e diretrizes de aplicação;

3. Agrupamento dos módulos com o objetivo de minimizar as dependências e maximizar a unidade lógica da arquitetura;

4. Inspeção do modelo de componentes de processos.

A modelagem utiliza a linguagem *OdysseyProcess-FEX* (TEIXEIRA, 2011), que provê suporte à representação explícita de variabilidades e opcionalidades. O papel das variabilidades é fundamental para gerenciar as alternativas de um domínio, permitindo categorizar os componentes estruturais em três frentes: aqueles que são estáticos em qualquer cenário (invariantes), os espaços que permitem adaptação (pontos de variação) e as opções disponíveis para preencher esses mesmos espaços (variantes). Ao possibilitar a demarcação de itens como sendo de uso obrigatório (mandatório) ou opcional, a linguagem confere a maleabilidade necessária para instanciar processos voltados a contextos particulares e diversos, observando a consistência do domínio.

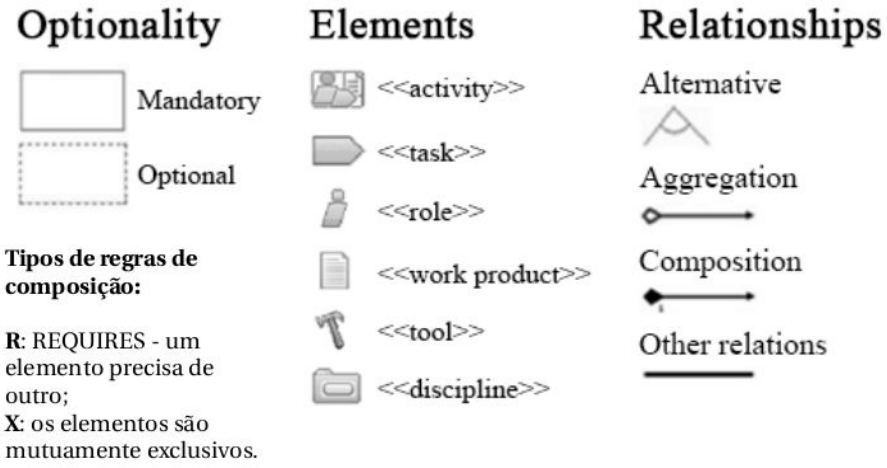
Os artefatos gerados durante a EDPS servem como blocos de construção para a Engenharia de Aplicação. A representação baseada em um Modelo de Características registra a lógica abstrata e as normas de integração, como exclusões mútuas e dependências. Por outro lado, a modelagem por componentes oferece um panorama no qual os itens operam de forma isolada, interagindo por meio de interfaces de comunicação. É essa compartimentação que viabiliza a ligação direta de insumos considerados de baixo nível, como roteiros de automação ou modelos de documentos, a cada bloco estrutural independentemente encapsulado.

A infraestrutura para a execução destas atividades é provida pelo ambiente *Odyssey*, ferramenta desenvolvida pelo Grupo de Reutilização de Software da COPPE/UFRJ. O ambiente foi adaptado para suportar as perspectivas de visualização da *OdysseyProcess-FEX* (TEIXEIRA, 2011), permitindo o gerenciamento de rastros entre os modelos de análise e projeto. O uso do ferramental *Odyssey*[1] é importante

[1]http://reuse.cos.ufrj.br/infoprojeto/3

para garantir a validação automática das diretrizes estruturais subjacentes ao modelo conceitual e para facilitar a identificação de alternativas viáveis de processos, reduzindo o esforço do engenheiro de processos durante a organização do domínio. Os principais elementos de representação da notação *OdysseyProcess-FEX* (TEIXEIRA, 2011) estão expressos na Figura 2.1.

Figura 2.1: Principais elementos da notação *OdysseyProcess-FEX* (TEIXEIRA, 2011)



Fonte: Adaptado de COSTA *et al.* (2018)

2.6 *ArchToDSSA* e a identificação de opcionalidades e de variabilidades

No campo da engenharia de domínio, a identificação de similaridades e variabilidades entre artefatos de um mesmo contexto constitui uma etapa importante para a construção de infraestruturas de reutilização. Assim, o método *ArchToDSSA* (VASCONCELOS e WERNER, 2011) consiste em uma abordagem voltada para a criação de arquiteturas de referência a partir da análise comparativa de arquiteturas de sistemas legados pertencentes a um mesmo domínio (VASCONCELOS e WERNER, 2011).

A execução do método organiza-se em três etapas. A primeira, de detecção de opcionalidades, consiste na comparação dos elementos das arquiteturas fornecidas com base em critérios de equivalência, que abrangem a igualdade de nomes, a identificação de sinônimos por meio de um dicionário de domínio e a quantidade de fragmentos textuais coincidentes entre os nomes dos elementos. Os elementos com correspondência são classificados como comuns, enquanto os demais recebem a classificação de opcionais. A segunda etapa, de detecção de variabilidades, localiza os

pontos do modelo em que configurações são possíveis. A terceira etapa, de geração do DSSA, consolida os resultados anteriores em uma arquitetura de referência sobre a qual o engenheiro de domínio realiza as escolhas finais quanto aos elementos opcionais e variáveis (VASCONCELOS e WERNER, 2011).

O raciocínio do método *ArchToDSSA*, de identificar convergências e divergências entre artefatos de um mesmo domínio para consolidar uma base, é relevante para iniciativas no campo da engenharia de domínio, que apresentam características similares.

2.7 Avaliação

A avaliação de artefatos produzidos no contexto da pesquisa em Engenharia de Software constitui etapa importante para a demonstração de sua utilidade e validade prática. A avaliação de artefatos de pesquisa deve apoiar-se em fundamentos metodológicos de forma a embasar a estratégia de coleta de dados e os métodos de análise.

2.7.1 Avaliação na *Design Science Research*

A pesquisa em sistemas de informação atua na interseção entre pessoas, organizações e tecnologia, exigindo a articulação de paradigmas complementares para o seu desenvolvimento (HEVNER *et al.*, 2004). Nesse campo, o paradigma da ciência do *design*, *Design Science Research* (DSR), orienta investigações cujo objetivo central é a criação de inovações e artefatos tecnológicos projetados para solucionar problemas, com o propósito de ampliar as fronteiras das áreas de conhecimento através de contribuições verificáveis à base de conhecimento da área (HEVNER *et al.*, 2004). Os artefatos construídos no âmbito da DSR abrangem modelos, métodos e instâncias de sistemas (HEVNER *et al.*, 2004).

A construção dessas soluções pressupõe a aplicação de teorias já pré-estabelecidas, e a eficácia do produto final deve ser demonstrada por meio de métodos de avaliação (HEVNER *et al.*, 2004). O ciclo iterativo entre construção e avaliação proporciona um entendimento do domínio do problema, assegurando que a intervenção proposta seja relevante para o ambiente em que se insere e que possua utilidade prática (HEVNER *et al.*, 2004).

2.7.2 *Survey* e Consulta a Especialistas

O planejamento de um estudo requer a seleção de métodos de coleta e análise de dados de acordo com os propósitos da investigação. Dentre as abordagens disponíveis, a pesquisa do tipo *survey* destaca-se pela capacidade de produzir descrições

quantitativas e capturar atitudes ou opiniões de uma população-alvo através da aplicação de um instrumento padronizado (FREITAS *et al.*, 2000). Em situações nas quais as questões investigadas comportam algum grau de incerteza ou carecem de técnicas analíticas, o julgamento coletivo de indivíduos experientes assume o papel de recurso metodológico, tornando-se alternativa ao conhecimento empírico direto (J. *et al.*, 1976).

Nesse contexto de complexidade, a consulta estruturada a especialistas torna-se um mecanismo útil para garantir a validade dos instrumentos e a consistência das inferências produzidas. A técnica Delphi, por exemplo, representa uma estratégia para estruturar o processo comunicativo entre especialistas, de modo a permitir a extração de opiniões sobre problemas sem a necessidade de interações presenciais que poderiam introduzir viéses de influência (J. *et al.*, 1976).

2.7.3 Mensuração de Atitudes por Meio da Escala *Likert*

A captura sistemática de percepções e disposições humanas exige a adoção de métricas padronizadas. O método desenvolvido por LIKERT (1932) configura-se como uma técnica amplamente utilizada para a medida de atitudes em instrumentos de pesquisa, fundamentando-se na premissa de que as respostas a proposições verbais constituem indicadores de tendências comportamentais. O procedimento envolve a formulação de afirmações objetivas acompanhadas de um conjunto de alternativas de resposta distribuídas em uma escala que vai da aprovação total à desaprovação total, passando por um ponto central de indecisão ou neutralidade (LIKERT, 1932).

Para fins de quantificação, atribuem-se valores numéricos consecutivos a cada uma das categorias de resposta. O valor individual do respondente é então obtido pelo cálculo da média ou da soma simples dos valores assinalados ao longo do instrumento (LIKERT, 1932). Essa abordagem dispensa o uso de calibração de pesos específicos por item, o que, além de simplificar a manutenção do instrumento, tende a produzir coeficientes de maior confiabilidade e facilitar o tratamento estatístico das atitudes coletadas (LIKERT, 1932).

2.7.4 Estatística Descritiva

O tratamento quantitativo dos dados coletados demanda a aplicação de medidas de estatística descritiva para a síntese e organização das observações brutas. No que diz respeito às medidas de tendência central, a média aritmética é calculada pela razão entre a soma de todas as observações e o número total de elementos da amostra, sendo empregada para representar o comportamento geral do conjunto de dados (MORETTIN e BUSSAB, 2010). A mediana, por sua vez, indica o valor que ocupa a posição central da série de dados ordenada, apresentando como vantagem

metodológica sua baixa sensibilidade a observações extremas ou atípicas (MORETTIN e BUSSAB, 2010). A moda corresponde ao valor de maior frequência absoluta no conjunto analisado (MORETTIN e BUSSAB, 2010).

A caracterização de uma distribuição exclusivamente por suas medidas de centralidade omite informações sobre a homogeneidade da amostra. Para suprir essa limitação, recorre-se às medidas de dispersão, entre as quais se destacam a variância e o desvio padrão (MORETTIN e BUSSAB, 2010). O desvio padrão, definido como a raiz quadrada positiva da variância amostral, mensura a magnitude do afastamento das observações em relação à média, compartilhando a mesma unidade de medida da variável original (MORETTIN e BUSSAB, 2010). A articulação dessas medidas de centralidade e de dispersão consolida a base analítica para diferentes tipos de análises de dados, servindo como técnicas centrais para a observação em populações.

Capítulo 3

Metodologia

Este Capítulo apresenta os métodos utilizados para a realização desta pesquisa, bem como definições a respeito do tipo de pesquisa executada.

3.1 Classificação da Pesquisa

Esta pesquisa classifica-se, quanto à sua natureza, como aplicada, uma vez que tem como produto a construção de um artefato: um método de análise de processos e uma Linha de Processos de Software para Jogos Sérios (LPrS-JS), com utilidade prática para equipes envolvidas no desenvolvimento desse tipo de aplicação. Quanto à abordagem, a pesquisa apresenta-se como predominantemente qualitativa, visto que os procedimentos centrais envolvem análise interpretativa de documentações de processos, síntese de vocabulário e modelagem de variabilidades; contudo, elementos quantitativos estão presentes de forma complementar, especialmente no critério de mandatoriedade ao construir o modelo da LPrS, que utiliza frequência de ocorrência dos elementos entre os processos analisados como critério objetivo de decisão.

Do ponto de vista de seus objetivos, a pesquisa possui caráter exploratório e descritivo (GIL, 2002). Exploratório porque o domínio de linhas de processos para jogos sérios ainda carece de consolidação metodológica; descritivo porque a pesquisa caracteriza o domínio, identificando e organizando seus elementos, variabilidades e estrutura antes de modelá-los na linha de processos.

Quanto a sua estratégia, este trabalho adota uma metodologia própria, estruturada em cinco etapas e inspirada no *Design Science Research* (DSR) HEVNER *et al.* (2004), paradigma que orienta investigações cujo objetivo é a concepção, construção e avaliação de artefatos que respondam a problemas práticos identificados em um domínio HEVNER *et al.* (2004). Dessa abordagem, a pesquisa incorpora seus elementos de construção: um problema extraído do ambiente, a ausência de uma infraestrutura sistemática para o desenvolvimento de JS; um artefato projetado para respondê-lo, a LPrS-JS e o método que viabiliza sua construção; enquadrados nas

categorias de método e instância previstas pela DSR HEVNER *et al.* (2004); e uma etapa de avaliação destinada a evidenciar sua utilidade, conduzida por verificação por instanciação e por avaliação de percepção com especialistas.

É importante observar, contudo, uma delimitação metodológica. A DSR, em sua forma completa, pressupõe a iteração entre construção e avaliação, em que os resultados da avaliação realimentam sucessivos refinamentos do artefato HEVNER *et al.* (2004). Nesta pesquisa, em razão das restrições de tempo e de escopo próprias de uma pesquisa de mestrado, o ciclo foi percorrido uma única vez, em uma execução sequencial de construção, verificação e avaliação, sem reentrada no projeto do artefato a partir das percepções coletadas. Por essa razão, opta-se por caracterizar a abordagem como uma metodologia própria inspirada na DSR, e não como uma aplicação integral do paradigma. O refinamento do artefato a partir dos resultados da avaliação é reconhecido como desdobramento natural deste trabalho e tratado na seção de trabalhos futuros.

3.2 Visão Geral das Etapas da Pesquisa

O percurso metodológico desta pesquisa organiza-se em cinco etapas sequenciais e interdependentes, ilustradas na Figura 3.1. Cada etapa corresponde a um produto ou resultado que alimenta a etapa seguinte, compondo um ciclo de construção e avaliação do artefato proposto.

A primeira etapa consiste na fundamentação teórica, cujo objetivo é estabelecer o referencial teórico que sustenta as decisões metodológicas da pesquisa. Foram abordados os conceitos de JS digitais, processos de software para jogos, reutilização de software, linhas de produto e de processos de software, a metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016), o método *ArchToDSSA* (VASCONCELOS e WERNER, 2011) e métodos de avaliação de pesquisa. Essa etapa fornece tanto o embasamento conceitual quanto a identificação das lacunas que motivam a pesquisa, sendo seus resultados apresentados no Capítulo 2.

A segunda etapa corresponde à apresentação do método de análise e construção da LPrS-JS. Com base nas lacunas identificadas na literatura e nos preceitos da metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016) e no procedimento *ArchToDSSA* (VASCONCELOS e WERNER, 2011), foi desenvolvido um método para analisar processos de software de domínios heterogêneos, sintetizar seus elementos comuns e variantes, e produzir os insumos necessários para a modelagem de uma linha de processos. Esse método é detalhado no Capítulo 4.

A terceira etapa consiste na aplicação do método e construção da LPrS-JS, configurando a prova de conceito da pesquisa. O método proposto foi executado sobre um conjunto de nove processos de desenvolvimento de JS identificados na literatura

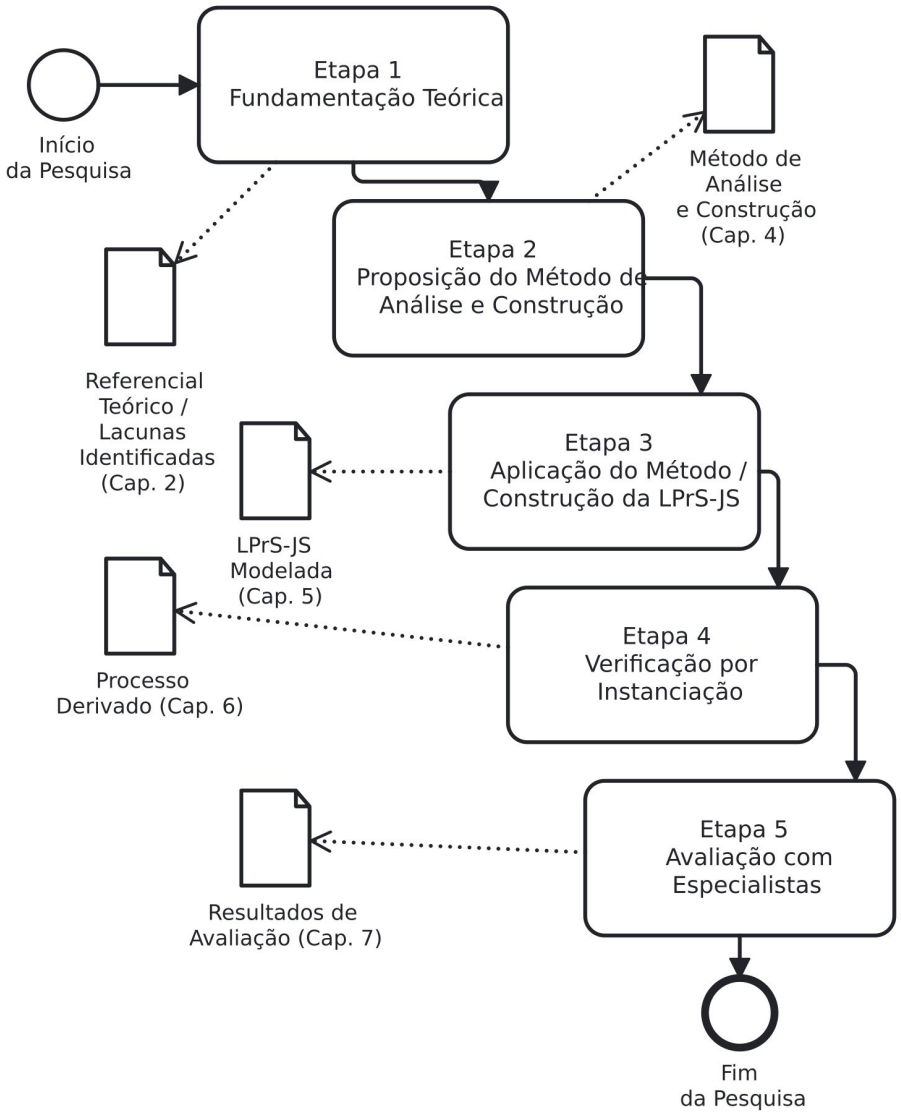
acadêmica, resultando na LPrS-JS modelada no ferramental *Odyssey*. Essa etapa é descrita no Capítulo 5.

A quarta etapa trata da verificação da LPrS-JS por instanciação. Um processo foi derivado a partir da linha construída, tomando como referência um modelo presente na literatura técnica. Essa atividade tem por objetivo verificar a capacidade de derivação da linha e sua adequação à variabilidade do domínio. Os resultados são apresentados no Capítulo 6.

A quinta e última etapa corresponde à avaliação da LPrS-JS por um painel de especialistas, com o objetivo de verificar a qualidade, a clareza e a aplicabilidade prática dos artefatos gerados. Essa etapa é descrita e conduzida no Capítulo 7.

A Figura 3.1 sintetiza esse percurso, indicando as entradas e saídas de cada etapa e sua correspondência com os capítulos da dissertação.

Figura 3.1: Visão geral das etapas da pesquisa, com entradas, saídas e correspondência com os capítulos da dissertação.



Fonte: Elaborado pelo autor (2026).

Embora organizadas como metodologia própria, as cinco etapas descritas possuem correspondência com os elementos da DSR HEVNER *et al.* (2004).

- A primeira etapa, fundamentação teórica (Capítulo 2), cumpre dupla função no paradigma, estabelecendo a relevância da pesquisa, ao caracterizar o problema a partir das lacunas do domínio de JS, e observa seu rigor, ancorando as decisões metodológicas na base de conhecimento existente com base nas metodologias *OdysseyProcessReuse* TEIXEIRA (2016) e *ArchToDSSA* VASCONCELOS e WERNER (2011).

- A segunda etapa, de proposição do método de análise (Capítulo 4), corresponde a uma primeira construção, na qual projeta-se o artefato do tipo método,

destinado a operacionalizar a análise comparativa de processos heterogêneos.

- A terceira etapa, aplicação do método e construção da LPrS-JS (Capítulo 5), dá continuidade à construção, agora produzindo o artefato do tipo Linha de Processo: o modelo de características da linha. Essa etapa configura, simultaneamente, a prova de conceito do método proposto na etapa anterior.

- A quarta etapa, verificação por instanciação (Capítulo 6), corresponde à demonstração do artefato, evidenciando a capacidade da linha de derivar processos coerentes para um contexto concreto do domínio, com avaliação de sua adequação interna.

- A quinta etapa, avaliação por especialistas (Capítulo 7), corresponde à avaliação do artefato no sentido da DSR, destinada a obter evidências sobre sua utilidade, clareza e viabilidade na perspectiva de potenciais usuários, dimensão que a verificação por si só não abrange.

- Por fim, a comunicação dos resultados e sua devolução à base de conhecimento, etapa também prevista pela DSR HEVNER *et al.* (2004), materializa-se no fim deste texto, especialmente na síntese das contribuições e nas respostas às questões de pesquisa apresentadas no Capítulo 8.

3.3 Abordagem para Construção do Método

A proposição do método descrito no Capítulo 4 partiu da identificação de uma lacuna operacional específica: a metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016), embora ofereça um ciclo bem definido para a Engenharia de Domínio de Processos de Software, não prescreve um procedimento detalhado para a etapa de análise comparativa entre processos heterogêneos, aquela em que processos com granularidades, nomenclaturas e estruturas distintas precisam ser colocados lado a lado para que similaridades e variabilidades sejam identificadas. Essa lacuna é relevante no domínio de JS, onde, conforme apontado no Capítulo 2, os processos documentados na literatura foram desenvolvidos de forma independente, por equipes com formações distintas e para contextos de aplicação variados, resultando em uma heterogeneidade terminológica e estrutural.

Diante disso, o método proposto foi construído como um instrumento que operacionaliza a etapa de análise de domínio da *OdysseyProcessReuse* (TEIXEIRA, 2016) para esse contexto específico. Sua elaboração seguiu um processo de elaboração progressiva: partindo das diretrizes gerais da metodologia base (delimitação de escopo, captura de conhecimento, análise de similaridades e variabilidades, e modelagem)

foram definidos procedimentos para cada uma dessas etapas, com heurísticas, artefatos intermediários e critérios de decisão explícitos, com base também no método ArchToDSSA (VASCONCELOS e WERNER, 2011). O resultado é um método que pode ser executado por um engenheiro de processos sem depender exclusivamente do julgamento tácito de especialistas para cada decisão de agrupamento ou classificação.

A construção do método também incorporou decisões específicas ao contexto de fontes acadêmicas. Na abordagem *bottom-up* prevista pela *OdysseyProcessReuse* (TEIXEIRA, 2016), o insumo natural são processos legados de uma organização, enquanto a abordagem *top-down* parte, em um nível mais alto, de necessidades organizacionais e modelos de referência. Como este trabalho não dispõe de processos organizacionais reais, os processos documentados na literatura acadêmica, já executados ou não, foram adotados como substitutos equivalentes, representando, ao mesmo tempo, registros do domínio com diferentes níveis de detalhamento e foco, permitindo análises para a extração de elementos comuns e variantes - adotando, portanto, uma abordagem *híbrida (top-down e bottom-up)*. Essa adaptação é justificada pela natureza do domínio de JS, cuja produção de conhecimento metodológico tem ocorrido predominantemente no ambiente acadêmico, conforme evidenciado por BENIGNO et al. (2024).

3.4 Abordagem para Construção da LPrS-JS

A construção da LPrS-JS constitui a etapa central desta pesquisa, na qual o método proposto no Capítulo 4 é efetivamente aplicado a um conjunto de processos de desenvolvimento de JS documentados na literatura acadêmica. Essa etapa corresponde à Engenharia de Domínio de Processos de Software prevista pela *OdysseyProcessReuse* (TEIXEIRA, 2016), sendo seu produto principal o modelo de características da linha, construído e armazenado no ferramental *Odyssey*.

A escolha da abordagem *híbrida (top-down e bottom-up)* para a condução dessa engenharia de domínio decorre da natureza das fontes disponíveis. Conforme discutido na seção anterior, os processos da literatura acadêmica podem assumir o papel de processos legados, fornecendo um corpo de conhecimento acumulado sobre execuções passadas do domínio, bem como podem servir como referências de domínio caso não executados.

A seleção dos processos que compõem o corpo fundador da linha partiu da revisão terciária de BENIGNO et al. (2024) e seguiu uma descida sistemática do nível das revisões secundárias para o dos estudos primários propositores de processos. Dos 22 estudos secundários analisados naquela revisão, 11 tratavam de jogos sérios; destes, 6 foram retidos por apresentarem, em suas seções de resultados e discussão, referências diretas a artigos que propõem processos de desenvolvimento de JS. A

partir dessas 6 revisões, foram identificados 70 trabalhos primários potenciais, dos quais 67 puderam ser recuperados. Esses 67 trabalhos foram submetidos a um procedimento de avaliação de qualidade, resultando em 28 considerados aptos. Por fim, 9 processos foram selecionados dentre os aptos, segundo critérios de diversidade de área de aplicação (educação formal, saúde e reabilitação física), variedade de nível de detalhamento documental e representatividade do domínio. O detalhamento de cada estágio, com as revisões e os trabalhos envolvidos, é apresentado na Seção 5.2.

A execução do método sobre esse conjunto de processos percorre as etapas de extração de elementos, definição de glossário, construção do vocabulário canônico, análise de mandatoriedade, identificação de variabilidades e, por fim, modelagem no ferramental *Odyssey* com uso da linguagem *OdysseyProcess-FEX* (TEIXEIRA, 2011). O detalhamento de cada uma dessas etapas, bem como os artefatos intermediários e os resultados obtidos, encontra-se no Capítulo 5.

3.5 Abordagem para Verificação da LPrS-JS

A verificação da LPrS-JS foi conduzida por meio da instanciação de um processo a partir da linha construída, tomando como referência um modelo de desenvolvimento de JS presente na literatura acadêmica. Essa estratégia de verificação fundamenta-se na premissa de que uma linha de processos deve ser capaz de derivar processos coerentes para diferentes contextos do domínio, com esforço reduzido em relação à definição de um processo do zero (ROMBACH, 2006; TEIXEIRA, 2016).

A instanciação parte da identificação do contexto do processo de referência, seu domínio de aplicação, público-alvo e objetivos principais; e prossegue com a seleção das características da linha que melhor correspondem a esse contexto, percorrendo as macrofases, papéis, artefatos e atividades modelados. O resultado da instanciação é um processo derivado, com suas escolhas de variantes justificadas e suas eventuais adaptações registradas. Esse registro permite verificar tanto a cobertura da linha quanto a clareza das regras de derivação.

Os detalhes da instanciação, incluindo o contexto, as escolhas realizadas, as verificações de conformidade e as adaptações necessárias, são apresentados no Capítulo 6.

Do ponto de vista das questões de pesquisa, esta etapa contribui para a resposta à QP1, ao verificar se a linha cobre o processo de construção de JS com clareza e completude; e à QP3, ao registrar o esforço e as dificuldades encontradas durante a configuração e derivação de novos modelos de processo.

3.6 Abordagem para Avaliação

A avaliação da LPrS-JS tem por objetivo verificar a qualidade e a aplicabilidade prática dos artefatos gerados, complementando a verificação por instanciação realizada na etapa anterior. Enquanto a verificação demonstra que a linha é capaz de derivar processos coerentes, a avaliação busca evidências sobre se os artefatos construídos são compreensíveis, úteis e viáveis na perspectiva de profissionais com experiência no domínio. Essa distinção é relevante porque um artefato tecnicamente correto pode, ainda assim, apresentar barreiras de adoção decorrentes de clareza insuficiente, vocabulário inadequado ou esforço de uso elevado.

A avaliação é estruturada em torno das três questões de pesquisa definidas no Capítulo 1. A QP1, sobre cobertura e rastreabilidade da linha, é observada por meio de questões que investigam se os especialistas reconhecem, nos artefatos produzidos, os elementos centrais do domínio de JS e se conseguem identificar a correspondência entre os componentes da linha e as práticas que conhecem. A QP2, sobre utilidade prática e ganho de produtividade percebido, é investigada a partir da percepção dos participantes sobre o valor da linha como apoio à tomada de decisão em projetos reais. A QP3, sobre viabilidade considerando o esforço de configuração e derivação, é avaliada com base nas experiências relatadas durante a interação com os artefatos.

Os participantes da avaliação serão especialistas recrutados a partir de contatos acadêmicos dos orientadores e do grupo de pesquisa, selecionados por possuírem experiência comprovada em ao menos uma das seguintes áreas: desenvolvimento de JS, engenharia de software, processos de software ou reutilização de software. O critério de inclusão exige experiência prática ou acadêmica relevante no domínio, de modo a garantir que as percepções coletadas sejam fundamentadas em conhecimento do campo.

O instrumento consistirá em um questionário estruturado, organizado em seções que contém questões a serem respondidas de acordo com a escala *Likert* e questões que recebem texto livre. As questões que utilizam a escala *Likert* (LIKERT, 1932) fazem uso de cinco pontos para mensurar percepções de clareza, completude, utilidade e viabilidade dos artefatos. As respostas serão analisadas por meio de estatísticas descritivas, de modo a buscar padrões e tendências de percepção entre os participantes. As questões de texto livre incluem questionamentos abertos, que permitirão aos participantes registrar observações, críticas e sugestões não capturadas pelos itens fechados. O questionário foi elaborado com base nas três questões de pesquisa e revisado pelos orientadores antes de sua aplicação, de modo a verificar a adequação e a clareza dos itens. O instrumento completo é apresentado no Capítulo 7.

Antes do preenchimento do questionário, os participantes terão acesso aos arte-

fatos da linha produzidos na etapa anterior em um sistema *web* tipo catálogo, a ser apresentado no Capítulo 7, de modo a fundamentar suas respostas em uma interação concreta com o material avaliado. O tempo estimado para essa interação e o preenchimento do questionário, bem como as instruções fornecidas aos participantes, são detalhados no Capítulo correspondente.

A análise dos dados coletados combina duas abordagens. Para os itens quantitativos, serão calculadas frequências e medidas de tendência central, com o objetivo de identificar padrões de percepção e pontos de consenso ou divergência entre os participantes. Não se pretende inferência estatística, o que é coerente com o caráter exploratório da avaliação e com o tamanho previsto do painel. As respostas abertas serão analisadas de acordo com o perfil do respondente, de modo a observar as nunaces específicas de acordo com a experiência. Os procedimentos detalhados de avaliação, incluindo o instrumento completo e os critérios de análise, são descritos no Capítulo 7.

3.7 Ameaças à Validade

Pesquisas de caráter construtivo e interpretativo estão sujeitas a ameaças que podem comprometer a confiabilidade e a generalização de seus resultados. Esta seção identifica as principais ameaças reconhecidas nesta pesquisa e descreve as medidas adotadas para mitigá-las, organizadas segundo quatro dimensões de validade (WHITSON, 2019) em pesquisas empíricas e construtivas: interna, de construto, externa e de conclusão.

1. **Validade interna.** A principal ameaça à validade interna reside na dimensão interpretativa da análise de processos. A extração de elementos, a definição do vocabulário canônico, a identificação de macrofases, subatividades e variabilidades envolvem julgamentos sobre equivalência semântica entre termos e estruturas originalmente concebidos em contextos distintos, por autores com formações e objetivos diferentes. Esse processo não pode ser inteiramente dissociado da experiência e da perspectiva dos pesquisadores envolvidos.

   Para mitigar essa ameaça, o método proposto prevê o registro explícito das decisões interpretativas em campos de Notas e Observações nos artefatos intermediários, tornando o raciocínio realizado a cada agrupamento rastreável. Os critérios quantitativos adotados na etapa de mandatoriedade, baseados em frequência de ocorrência, também contribuem para reduzir a dependência de julgamentos puramente subjetivos nas decisões de classificação dos elementos como mandatórios ou opcionais.

2. **Validade de construto.** Há o risco de que os construtos definidos, macro-

fases, subatividades, vocabulário canônico e variantes não capturem adequadamente toda a extensão do domínio de JS, seja por limitações do conjunto de processos analisados, seja por decisões de agrupamento que obscureçam distinções relevantes. O uso de nove processos-fonte com diferentes focos de aplicação e níveis de detalhamento busca ampliar a cobertura do domínio, reduzindo o risco de que a linha reflita apenas uma subárea ou uma perspectiva específica.

3. **Validade externa.** Os resultados desta pesquisa foram obtidos a partir de processos documentados na literatura acadêmica, o que pode limitar sua generalização para contextos industriais, nos quais os processos costumam ser menos formalizados e mais dependentes de conhecimento tácito das equipes. Da mesma forma, a linha construída reflete o estado do domínio tal como documentado até o momento da coleta, podendo não contemplar práticas emergentes ainda não sistematizadas na literatura.

4. **Validade de conclusão.** A validade de conclusão diz respeito à confiabilidade das interpretações extraídas dos dados.

   Coerentemente com a natureza do artefato e com os objetivos da pesquisa, a avaliação foi concebida como um estudo exploratório de percepção, conduzido com um painel reduzido de especialistas, e não como um estudo inferencial. Os itens quantitativos não são, portanto, tratados como estimativas de uma população, mas como indicadores descritivos das percepções desse painel específico. A principal ameaça nessa dimensão é a superinterpretação, ao ler medidas derivadas de poucos respondentes como se possuíssem estabilidade estatística. Para mitigá-la, a síntese quantitativa privilegia a distribuição das respostas, em vez de médias isoladas, não recorre a inferência estatística e ancora as conclusões, sobretudo, na triangulação com as respostas discursivas e o perfil de cada respondente.

3.8 Escrita de relato

A escrita desta dissertação utilizou de ferramentas de inteligência artificial generativa para ajustes no corpo textual, de forma a fornecer uma melhor apresentação de ideias e organização textual.

Capítulo 4

Método para a construção da LPrS-JS

Este capítulo descreve o método para alinhar, analisar e comparar processos de desenvolvimento de jogos sérios, observando suas características similares e variantes, visando a construção de uma LPrS por meio de uma abordagem *híbrida (bottom-up e top-down)*. A abordagem *bottom-up*, segundo TEIXEIRA (2016), consiste na construção da linha de processos a partir de processos já utilizados anteriormente e/ou conhecimentos extraídos de execuções passadas; usando esse conjunto como base para identificar o que é comum e o que varia entre os processos. Já a abordagem *top-down* parte das necessidades do contexto ou de modelos pré-existentes para a criação da linha. A abordagem híbrida escolhida corresponde, na taxonomia de KRUEGER (2002), a uma estratégia extrativa complementada por elementos proativos SILVA *et al.* (2011). O método desenvolvido inclui diretrizes e um conjunto de heurísticas para a identificação da presença/obrigatoriedade de elementos, bem como suas variações e opções, e derivação de um vocabulário canônico agregado a um glossário, que dá sustentação terminológica aos elementos definidos. O procedimento apresentado neste capítulo absorveu procedimentos descritos no método *ArchToDSSA* (VASCONCELOS e WERNER, 2011) e na metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016).

O objetivo do procedimento é fornecer mecanismos práticos de análise de processos originalmente descritos com granularidades, nomenclaturas e estruturas distintas. Apesar dessa heterogeneidade, o foco é tornar esses processos comparáveis e produzir uma estrutura estável que possa ser usada para apoiar a modelagem de uma LPrS para Jogos Sérios.

Para o procedimento de construção da linha, a metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016) foi utilizada como base para a construção do método, sobretudo as etapas referentes à análise de domínio, bem como o método *ArchToDSSA* (VASCONCELOS e WERNER, 2011) foi utilizado como base de derivação mais focada

na análise de opcionalidades e variabilidades. Em ambos os casos, etapas foram absorvidas e adaptadas ao longo do método.

A escolha da *OdysseyProcessReuse* TEIXEIRA (2016) e da notação *OdysseyProcess-FEX* TEIXEIRA (2011) justifica-se por três razões. Primeiro, pelo contexto de desenvolvimento, este trabalho integra-se à linha de pesquisa do Grupo de Reutilização de Software da COPPE/UFRJ, responsável pela concepção e evolução tanto da abordagem quanto do ambiente *Odyssey*, o que assegura continuidade, acesso ao ferramental e suporte metodológico. Segundo, pela adequação ao artefato: a revisão sistemática de abordagens de LPrS conduzida por TEIXEIRA *et al.* (2019) evidencia que poucas notações cobrem, de forma integrada, o conjunto de elementos de processo com suas opcionalidades, os pontos de variação, as relações de dependência e exclusão mútua e os fluxos de controle; capacidades reunidas pela *OdysseyProcess-FEX*, que organiza a variabilidade em dois níveis (características e componentes) e incorpora cardinalidade e tratamento de fluxos. Terceiro, pelo foco em engenharia de domínio, como a contribuição desta pesquisa é a construção da linha, a abordagem alinha-se à lacuna apontada pela mesma revisão quanto à escassez de infraestruturas de reutilização integradas.

Abordagens alternativas foram consideradas, entre elas as baseadas em *SPEM* OBJECT MANAGEMENT GROUP (2008); OLIVEIRA JUNIOR *et al.* (2013); PAZIN *et al.* (2018), como *SMartySPEM* OLIVEIRA JUNIOR *et al.* (2013) e *vSPEM* PAZIN *et al.* (2018) TEIXEIRA *et al.* (2019), mas a representação por modelos de características da *OdysseyProcess-FEX* foi preferida por explicitar a variabilidade do domínio no nível de abstração adequado à proposta e por dispor de mecanismos de mapeamento que preservam consistência e rastreabilidade entre os níveis de características e de componentes TEIXEIRA *et al.* (2019).

Operacionalmente, a análise de domínio é conduzida como um ciclo incremental: inicia-se pela delimitação do escopo (definindo fronteiras e o conjunto inicial de processos que fará parte do domínio), segue-se para a captura do conhecimento (identificando fontes, escolhendo técnicas e registrando/armazenando as informações obtidas), evolui para a análise de similaridades, opcionalidades e variabilidades (mapeando equivalências, distinguindo elementos mandatórios e opcionais e determinando pontos de variação e suas variantes) e, na última etapa, consolida-se a modelagem da análise do domínio, materializando esse conhecimento em um modelo de características. O fluxo é sequencial, porém prevê retornos entre etapas sempre que houver o interesse de incorporação de novos processos.

Desse modo, o método de análise descrito compõe os passos para a construção da LPrS, integrado as etapas da análise de similaridades da metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016) e do método *ArchToDSSA* (VASCONCELOS e WERNER, 2011), bem como a identificação de escopo, captura de conhecimento

provenientes e a notação *OdysseyProcess-FEX* (TEIXEIRA, 2011) da *OdysseysProcessReuse* (TEIXEIRA, 2016).

Sob uma ótica de alto nível, o método é executado com as seguinte etapas, conforme apresentado na Figura 4.1, que serão detalhadas nas seções seguintes:

1. **Delimitação do escopo:** definição dos objetivos da linha a ser construída e da abordagem de análise escolhida;

2. **Captura de conhecimento do domínio de processos de software**: coleta de fontes de informação sobre os processos do domínio de jogos sérios;

3. **Extração de elementos de processo**: leitura e análise de documentos coletados para identificação de elementos relevantes, como macrofases, papéis, artefatos, atividades e tarefas;

4. **Definição de glossário**: para cada processo, é construído um glossário com as definições dos elementos identificados na etapa anterior;

5. **Definição de vocabulário canônico**: associação de elementos que tiverem significado semelhante em um único elemento como um termo canônico;

6. **Definição de mandatoriedade**: organização dos processos e avaliação da presença de elementos canônicos para a definição de mandatoriedade e opcionalidade;

7. **Definição de variabilidades**: identificação de elementos variáveis entre os processos;

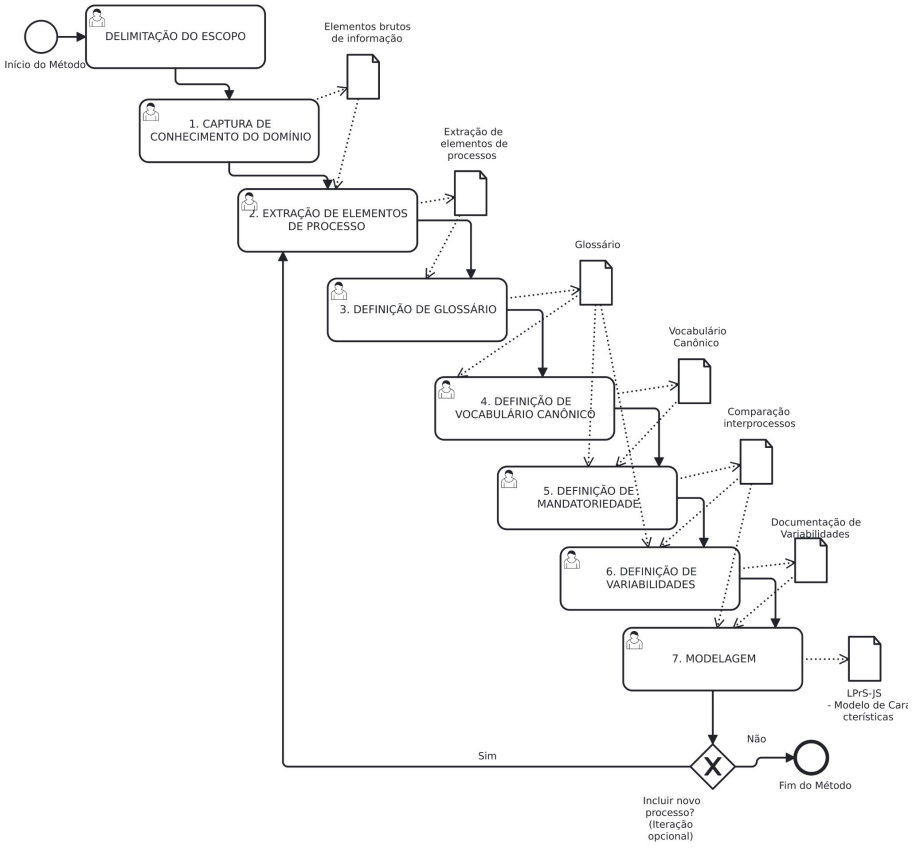
8. **Modelagem**: realização da transferência de conhecimento do domínio para uma representação usando um modelo de características. Neste trabalho, serão utilizados a notação *OdysseyProcess-FEX* (TEIXEIRA, 2011) e o ferramental *Odyssey* para o apoio da modelagem.

9. **Iteração (opcional)**: inclusão de novo processo na linha.

Este método foi pensado para ser flexível, com a possibilidade de inclusão de novos processos, e rastreável, com decisões de agrupamentos e interpretação registradas. Seu resultado é um modelo de características que pode ser reutilizado para a derivação de novos processos específicos de projetos.

A Figura 4.1 apresenta o diagrama da visão geral do fluxo de atividades do método, bem como seus artefatos de entrada e saída, que serão tratados nas seções seguintes deste Capítulo.

Figura 4.1: Visão geral do fluxo de atividades do método para a análise de processos, resultando na construção de uma Linha de Processos de Software para Jogos Sérios.



Fonte: Elaborado pelo autor (2026).

4.1 Atividade: Delimitação do escopo

Conforme definido por TEIXEIRA (2016), a análise de domínio da metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016) inicia-se na etapa de definição de escopo da linha de processos a ser construída. Neste momento, é necessário que sejam definidos os objetivos e a abordagem de análise escolhida.

Para o método aqui descrito, é fixado como objetivo a construção de uma linha de processos para o domínio de Jogos Sérios, com foco no aproveitamento de informações de processos definidos em um corpo de artigos selecionados na literatura acadêmica. É fixada, também, a utilização da abordagem híbrida (*top-down e bottom-up*) - conforme definida anteriormente.

É recomendada a utilização de 5 a 15 processos, inicialmente, para a execução do método, de modo a garantir um número de processos variado para possibilitar uma análise que corresponderá a resultados mais abrangentes, e um limite de análise a um número que seja viável a manipulação e a assimilação de informações.

As decisões tomadas durante a etapa de delimitação do escopo devem ser regis-

tradas e mantidas, ainda que de forma sucinta, descrevendo o norte para construção da linha, de modo a explicitar os objetivos, os meios de informação, a abordagem escolhida e observações extras, permitindo serem consultadas ao longo da execução do método.

4.2 Atividade: Captura de Conhecimento do Domínio

Nesta etapa, ocorre a captura de conhecimentos do domínio. A abordagem *bottom-up* prevê a utilização de dados de processos legados, de execuções passadas e definições anteriores de uma organização para a composição de um corpo de conhecimento passível de análise daquele domínio. A abordagem *top-down* parte, em um nível mais alto, de necessidades dos construtores da linha e de práticas pré-existentes, não necessariamente executadas.

Esta etapa tem como saída os artefatos que contêm as descrições dos processos de software para JS ou sobre suas execuções.

4.3 Atividade: Extração de Elementos de Processo

Nesta etapa, cada processo, capturado na etapa anterior, tem seus elementos identificados e registrados. Deve-se extrair de suas documentações os seguintes elementos de processo para uma descrição detalhada:

- **Unidades de trabalho** (atividades, tarefas e passos) realizadas em cada etapa do processo. É importante manter a associação entre unidades de trabalho, por exemplo, manter o rastro das tarefas que compõem uma atividade;
- **Artefatos** produzidos e consumidos (documentos, protótipos, modelos, etc.); e
- **Papéis e atores** envolvidos no desenvolvimento.

Recomenda-se que o conhecimento extraído seja organizado usando uma tabela como artefato de representação. A tabela deve especificar, por registro de atividade, o nome da atividade (aqui chamado de termo nativo), os papéis envolvidos, os artefatos, bem como suas tarefas associadas. Cada linha pode conter uma campo específico para notas relevantes para a caracterização daqueles elementos de processo.

A saída desta etapa consiste em tabelas de extração dos processos. A Tabela 4.1 traz o fragmento de exemplo de artefato de saída relacionado a um processo.

Tabela 4.1: Exemplo de tabela de extração de um processo.

Atividade	**Tarefa**	**Papel**	**Artefato**	**Notas**
Problem formulation	*Identify education problem*	*Stakeholders, Decision makers*	*Output: Education Problem Specification (problem statement)*	Ponto de partida do ciclo; define tema e contexto.
Problem formulation	*Explain significance*	*Stakeholders*	*Output: Education Problem Specification (rationale)*	Justifica a criticidade do problema.

4.4 Atividade: Definição de Glossário

Na posse dos elementos extraídos na etapa anterior, em uma nova tabela, cada termo nativo de processo relevante deve ter sua definição registrada. A estrutura recomendada para a tabela contém as colunas de tipo/categoria do elemento, nome de elemento (termo nativo) e definição. Esse inventário constitui a matéria-prima para a construção da linha base de comparação e a síntese do vocabulário canônico.

A saída dessa etapa consiste no glossário de cada processo selecionado. A Tabela 4.2 traz o fragmento de exemplo de artefato de saída relacionado a um processo.

Tabela 4.2: Fragmento de tabela de glossário.

Categoria	**Termo**	**Definição**
Activity	*Preparation*	Fase inicial onde educadores trabalham sozinhos para identificar metas curriculares, áreas de conteúdo, demografia do aluno e a situação de aprendizagem.
Activity	*Co-Design*	Fase multidisciplinar onde educadores e desenvolvedores exploram e descobrem os elementos do jogo (agentes, mundo, objetos e eventos) e como eles interagem.

4.5 Atividade: Definição de Vocabulário Canônico

Nesta atividade, ocorre a construção do Vocabulário Canônico (VC) do conjunto de processos. O VC é um artefato em quatro dimensões (macrofases, atividades,

artefatos e papéis) usado para identificar e descrever a equivalência de termos nativos com termos canônicos equivalentes: como um agrupamento de termos sob um mesmo rótulo generalizado.

Vale ressaltar que o VC não considera o grão mais fino das atividades, chamados de tarefas, mas as considera em conjunto com as atividades para a definição de elementos chamados macrofases. As macrofases consistem em grandes etapas que fazem parte do ciclo de vida de um processo de software, sendo unidades de trabalho agrupadas em etapas maiores. As macrofases podem ser definidas como fases do ciclo de vida de desenvolvimento de jogos como um "esqueleto" de etapas agrupadas. A decisão do uso desse tipo de estrutura foi tomada para se ter uma maior capacidade de agrupamento de atividades e para tornar a linha mais generalizável.

Esta etapa consiste em três subatividades: identificação de macrofases, identificação de subatividades e construção de vocabulário canônico.

4.5.1 Identificação de Macrofases

A identificação das macrofases é feita de forma *bottom-up*: inicialmente, observando o glossário de cada processo, as atividades e as tarefas são identificadas e são observadas as semânticas de suas descrições, bem como são observados os artefatos de entrada e saída e os papéis envolvidos; isso com o auxílio, também, do artefato de extração de elementos de processo construído. Com esse entendimento, tarefas e atividades com função semelhante são agrupadas, essas recebendo um nome para identificação e uma descrição única (termo canônico). É recomendado que o registro seja feito em uma tabela, e também seja armazenado o nome do processo o qual participou na identificação da macrofase para a manutenção de rastreabilidades com as fontes de informações.

O resultado é um conjunto de macrofases que é capaz de acomodar as variações dos processos analisados, suficientemente genérico para inclusão de processos futuros e, ao mesmo tempo, específico o bastante para não perder o significado das transições relevantes.

A Tabela 4.3 apresenta um exemplo de definição de uma macrofase.

Tabela 4.3: Exemplo de definição de macrofase.

Macrofase canônica	**Processo**	**"Macrofase nativa"**	**Descrição da macrofase canônica**
MF1 - Problema & Objetivos	eLuna	*Preparation*	Fase de entendimento do contexto, dos *stakeholders* e do problema a ser tratado, na qual são definidos o escopo e os objetivos de aprendizagem/impacto do jogo sério.
MF1 - Problema & Objetivos	iPlus	*Identification + Pedagogical Objectives*	Fase de entendimento do contexto, dos *stakeholders* e do problema a ser tratado, na qual são definidos o escopo e os objetivos de aprendizagem/impacto do jogo sério.

Essas macrofases compõem um eixo no ciclo de desenvolvimento determinando fases maiores de ações e resultados de valor: cada atividade, papel ou artefato de um processo é posteriormente ancorado em uma dessas macrofases em uma fase futura do processo.

4.5.2 Identificação de Subatividades Canônicas

Uma vez estabelecidas as macrofases, o passo seguinte consiste em detalhar a estrutura interna de cada uma delas por meio da identificação e derivação de suas subatividades canônicas. Semelhante à etapa anterior, este procedimento utiliza a base de dados extraída, mas com um foco mais granular.

Observam-se as tarefas, práticas e atividades nativas dos processos analisados que foram previamente ancoradas em uma determinada macrofase. A partir da análise semântica de suas descrições, propósitos operacionais e artefatos manipulados, agrupam-se aquelas que possuem o mesmo objetivo central de engenharia, *design* ou avaliação.

Para cada grupo formado, deriva-se uma subatividade canônica. A esta atividade são atribuídos um nome e uma descrição consolidada, que sintetizam a intenção de todas as atividades nativas que a compõem. Adicionalmente, durante este mapeamento, registra-se a frequência com que a atividade canônica é encontrada nos processos analisados.

O resultado desta etapa é um catálogo de subatividades canônicas que compõem o fluxo de cada macrofase, mantendo a rastreabilidade com as metodologias de origem.

A Tabela 4.4 apresenta um exemplo de como uma atividade canônica é derivada e estruturada a partir dos processos analisados.

Tabela 4.4: Exemplo de derivação de subatividades canônicas.

Macrofase	**Subatividade canônica**	**Descrição**	**Freq. (0–9)**	**Mandatório**
MF1 - Problema & Objetivos	1.1 Entender o problema	Compreender o problema educacional, terapêutico ou de negócio a ser resolvido, identificando o estado da arte e necessidades.	9	Sim
MF1 - Problema & Objetivos	1.2 Identificar contexto e escopo	Delimitar o domínio de aplicação, compreender o público-alvo, situação dos aprendizes e as características gerais do jogo/terapia.	6	Sim
MF1 - Problema & Objetivos	1.3 Identificar *stakeholders*	Mapear os participantes, tomadores de decisão, especialistas e usuários finais envolvidos no processo.	4	Não

4.5.3 Construção do Vocabulário Canônico

A construção do VC segue um processo parecido com a identificação das macrofases e das subatividades, mas, aqui, sem a realização de agrupamento de elementos de tipos diferentes. Para cada dimensão, entre papéis e artefatos, a observação dos registros do glossário de um processo é realizada, bem como de suas definições. Os elementos com termo nativo de função semelhante são agrupados, recebendo um termo e uma definição canônicos. Aqui, também é recomendado o registro do nome do processo que forneceu o termo nativo.

A Tabela 4.5 apresenta o fragmento de uma dimensão de um exemplo de VC.

Tabela 4.5: Fragmento de tabela de papéis do vocabulário canônico.

Papel canônico	**Processo**	**Papel nativo (equivalente)**	**Responsabilidades (canônicas)**
Facilitador (moderador)	iPlus	*iPlus facilitator*	Conduz dinâmicas/-sessões e organiza o trabalho
Facilitador (moderador)	GAMED	*Moderator; Scribe*	Conduz dinâmicas/-sessões e organiza o trabalho

4.6 Atividade: Definição de Mandatoriedade

Nessa etapa, a mandatoriedade e a opcionalidade de elementos são avaliadas em um contexto de composição da LPrS-JS. A análise consiste em observar a frequência de participação de um dado elemento no conjunto de processos considerados para construção da linha. Desta forma, para cada elemento canônico deve ser associado seu termo nativo no processo de origem, caso exista. Essa análise pode ser realizada através de um artefato de apoio como uma tabela. A análise é realizada por macrofase como guia de organização dos termos canônicos a serem analisados para cada tipo de elemento (macrofase, atividade, artefato e papel), sendo organizadas em uma primeira coluna, seguido da coluna com o termo canônico relacionado que será avaliado.

As colunas subsequentes da tabela representam os processos e, em cada célula, contém o termo nativo do elemento correspondente; sendo mantido em branco (ou com um traço "-") caso o elemento não apareça naquele processo. As últimas colunas contém uma breve descrição daquele item canônico, seguido de uma coluna de frequência, que significa em quantos processos aquele termo canônico está presente. Por fim, encontra-se a coluna de mandatoriedade, onde, observando o valor calculado da frequência, é decidido se o elemento canônico é mandatório ou opcional.

O critério para a definição se um elemento é mandatório consiste na definição de um valor limiar coerente, a depender do aplicador do método. Por exemplo, caso 60% dos processos possuam aquele elemento, ele é obrigatório; caso contrário, é opcional.

A Tabela 4.6 apresenta um fragmento de uma tabela de análise de mandatoriedade de uma dimensão.

Tabela 4.6: Fragmento de tabela de análise de mandatoriedade.

Macrofase	Papel canônico	iPlus	GAMED	QUARTIC	...	Descrição	Freq.	Mand.
MF1 - Problema & Objetivos / Transversal	Facilitador (moderador)	*iPlus facilitator*	*Moderator; Scribe*	-	...	Conduz dinâmicas / sessões e organiza o trabalho	2	Não
MF1 - Problema & Objetivos / Transversal	Decisor / Patrocinador	*Product owner*	*Decision makers; Sponsor; Stakeholders*	*Client*	...	Define escopo e aceita marcos	3	Não

4.7 Atividade: Definição de Variabilidades

Nesta etapa, as variabilidades da linha são identificadas e classificadas. Com base nos achados da mandatoriedade dos elementos, na etapa anterior, são definidos como pontos de variação obrigatórios os elementos mandatórios, e, os demais, opcionais.

Esta etapa é dividida em duas subfases para cada dimensão entre macrofases, papéis, artefatos e subatividades: agrupamento e definição de variantes. Os artefatos Glossário, Vocabulário Canônico e Análise de Mandatoriedade são utilizados para a concretização das subfases desta etapa.

4.7.1 Agrupamento

Nesta subfase, o termo canônico é mapeado para os termos nativos de cada processo. Esses termos nativos representam variantes brutas para cada termo canônico. A Tabela 4.7 apresenta uma prévia de um agrupamento, que contém o tipo de elemento, termo canônico, classificação de mandatoriedade/opcionalidade de acordo com a macrofase, processos que possuem o elemento e as variantes brutas.

Tabela 4.7: Tabela com agrupamento de termos nativos (variantes brutas) que participaram da criação de cada macrofase pro processo originário.

Tipo de Elemento	**Elemento Canônico**	**Classificação**	**Processos que possuem**	**Variantes brutas**
Macrofases	MF1 - Problema & Objetivos	Mandatório	eLuna, iPlus, Reha, GAMED, MECONESIS, URANO, QUARTIC, EXERGAME, PROGame	• *Preparation (eLuna)* • *Identification + Pedagogical Objectives (iPlus)* • *Research / Idea definition (Reha)* • *Problem formulation; Game idea generation; Requirements development (GAMED)* • *Analysis (MECONESIS)* • *Pre-phases 1–3 (Educational challenges; Type of game; Initial story & characters) (URANO)* • *Order 1 - Requirements (QUARTIC)* • *1) Exercise definition (EXERGAME)* • *0) Project initiation (PROGame)*

Continua na próxima página

Tabela 4.7 – continuação da página anterior

Tipo de Elemento	Elemento Canônico	Classificação	Processos que possuem	Variantes brutas
Macrofases	MF2 - *Design* do jogo & Prototipagem (iterativa)	Mandatório	eLuna, iPlus, Reha, GAMED, MECONESIS, URANO, QUARTIC, EXERGAME, PROGame	• *Co-Design + Co-Specification (eLuna)* • *Ludic Game Script (GDD) + GamePlay (iPlus)* • *Iterative prototype development with therapist (Reha)* • *Game design (spiral) (GAMED)* • *Pre-production (MECONESIS)* • *Phases 1–6 (Chapters/Scenes/Educational/Emotional/Adaptation/Collaboration design) (URANO)* • *Order 2–3 - Mapping & Contextualisation (QUARTIC)* • *2) Virtualization (VE) + 3) Game design (EXERGAME)* • *2) Modeling (design per increment) (PROGame)*

4.7.2 Definição de variantes

Nesta subfase, as variantes brutas de cada registro proveniente do artefato produto da subfase de agrupamento são avaliadas com base em suas definições no glossário de cada processo. São realizadas análises semânticas para a identificação de propósitos semelhantes nos conceitos das variantes brutas. Os resultados são subagrupamentos de variantes brutas, formando as variantes finais da linha. Essas variantes são definidas em uma tabela com campos para o nome do agrupamento abstrato, variantes brutas originárias, observações a respeito da análise semântica, nome e tipo de variante. Nesta etapa, devem ser definidas as regras de composição entre variantes, exclusividade mútua e *require* quando se aplicarem, além de cardinalidades conforme a semântica dos elementos.

A Tabela 4.8 apresenta um fragmento de exemplo de saída desta subfase, que também é a saída da etapa de definição de variabilidades.

Tabela 4.8: Fragmento de tabela de definição de variabilidades.

Agrupamento abstrato	**Variantes brutas**	**Observação**	**Variante Obrigatória, indicação de uso**
A. Definição de Objetivos de Domínio (Foco em definir o que ensinar ou tratar)	*Preparation*	Foca exclusivamente no trabalho dos educadores para identificar metas curriculares e demografia, sem mencionar tecnologia ainda.	Foco Pedagógico/Terapêutico - Usar se o time for composto majoritariamente por especialistas de domínio
	Identification + Pedagogical Objectives	Foca nas necessidades educacionais e consenso participativo sobre objetivos pedagógicos.	
	1) Exercise definition	Foca estritamente nos requisitos da terapia (exercício físico) e restrições de saúde.	
B. Engenharia de Requisitos e Análise (Foco formal em levantar requisitos e escopo)	*Project initiation*	Engenharia: determinação de necessidades, contexto e restrições técnicas/de projeto.	Foco em Engenharia - Usar se o projeto for grande e exigir contrato/especificação formal de software
	Order 1 Requirements	Planejamento formal a partir de metas do cliente. Viés de engenharia de sistemas.	
	Analysis	Análise de requerimentos com foco em UX e necessidades. Abordagem técnica de levantamento.	
	Problem formulation; Game idea generation; Requirements development	Embora cite "ideia", o peso da definição está na especificação de requisitos funcionais e não-funcionais.	

4.8 Atividade: Modelagem

Nesta última etapa, com base nos artefatos construídos nas etapas anteriores, o modelo de características da LPrS-JS deve ser construído. No contexto desse trabalho, é utilizado o ferramental *Odyssey* e a linguagem *OdysseyProcess-FEX* (TEIXEIRA, 2011), considerando as definições desenvolvidas, bem como os achados mandatórios, opcionais e suas variantes. Com a utilização da ferramenta, é possível a inserção de diversos detalhes e definições ao modelo, além da exportação para diferentes formatos possibilitar uma melhor distribuição da linha.

4.9 Fluxo de inserção de novos processos

Conforme apresentado no diagrama da Figura 4.1, é possível incluir novos processos à LPrS-JS após essa estar consolidada. Isso pode ser realizado executando parte do método novamente, agora apenas para este processo, a partir da etapa de extração de dados, já na posse dos documentos descritivos do processo de software a incluir. Nas demais etapas, o executante do método deve proceder com atualizações nos artefatos de análise já existentes, de modo a mapear corretamente os elementos do novo processo.

4.10 Limitações, validade e validação

Há uma dimensão interpretativa irredutível quando se alinham processos idealizados com objetivos e públicos distintos e documentados em diferentes níveis, sendo impossível desassociar, também, os achados ao nível de análise interpretativa dos pesquisadores envolvidos e de suas experiências prévias. O método, no entanto, admite essa circunstância e prevê o registro de interpretações em campos de Notas e Observações; e as decisões quantitativas não dependem de julgamentos subjetivos, de modo a mitigar a variação proveniente de subjetividades.

Cabe explicitar a estratégia de validação do método. As suposições que o sustentam, entre elas, a de que processos heterogêneos podem ser tornados comparáveis por meio de um vocabulário canônico, a de que a frequência de ocorrência é um critério adequado para definir mandatoriedade e a de que macrofases obtidas por agrupamento semântico preservam o significado das transições; não foram submetidas a uma validação direta e independente. Sua validação dá-se de forma indireta, pelo uso e aplicação integral do método à construção de uma linha de processos, descrita no Capítulo 5, constituindo a prova de conceito que evidencia que o procedimento é viável e capaz de produzir, a partir de processos reais documentados na literatura, um modelo de características coerente. Evidências indiretas adicionais provêm da instanciação do Capítulo 6, que exercita o resultado da modelagem, e da avaliação por especialistas do Capítulo 7. Reconhece-se que essa validação indireta não substitui um estudo dedicado às suposições do método, o que é registrado como direção de trabalho futuro.

Capítulo 5

Construção da LPrS-JS

Foi conduzida uma prova de conceito aplicando o procedimento detalhado no Capítulo 4 a um conjunto de processos de desenvolvimento de JS documentados na literatura acadêmica, de modo a validar o método proposto criando uma linha de processos para JS. Este capítulo apresenta os resultados dessa aplicação, demonstrando como o método foi operacionalizado e quais foram os artefatos gerados.

Os objetivos desta prova de conceito incluem: (a) a demonstração da viabilidade prática do método proposto; (b) construção de um vocabulário a partir de processos fundadores reais; (c) identificação de macrofases, subatividades, possivelmente, tarefas, papéis e artefatos comuns ao domínio; (d) geração de artefatos de análise; (e) estudo de similaridades e variabilidades em processos do domínio de JS analisado; e (f) derivação de uma LPrS-JS como artefato relevante para representação e consolidação do conhecimento do domínio analisado.

As versões em alta resolução de todos os modelos de características apresentados neste capítulo estão disponíveis no catálogo web da LPrS-JS[1].

Os modelos de características apresentados nesta seção e nas seguintes utilizam a notação *OdysseyProcess-FEX* (TEIXEIRA, 2011), cujos elementos gráficos, mandatoriedade e opcionalidade, pontos de variação, variantes, cardinalidades e as regras de composição require (R) e exclusão mútua (X); foram apresentados na Seção 2.5.1 (Figura 2.1). A leitura das figuras a seguir pressupõe essas convenções.

As seções subsequentes detalham a execução e os detalhes sobre a execução do método.

5.1 Atividade: Delimitação do escopo

Para uma perspectiva mais direcionada ao longo da execução do método, o escopo foi delimitado da seguinte forma: processos de software, documentados na literatura

[1] `https://teste-blond-ten.vercel.app/catalogo`

acadêmica, para a construção de jogos sérios digitais, excluindo, portanto, sistemas gamificados e jogos apenas de entretenimento. Tais documentações podem ser representadas em diferentes níveis de detalhamento e focos quanto ao *design* e quanto ao desenvolvimento.

Desse modo, a abordagem escolhida para a modelagem de domínio da linha foi a híbrida (*top-down e bottom-up*), em razão da utilização de descrições de processos encontrados na literatura acadêmica, já executados ou não.

A escolha da utilização de processos documentados na literatura acadêmica ocorreu em razão da maior facilidade de obtenção de insumos, além da área acadêmica ter-se mostrado interessada no desenvolvimento de JS com interesses que não são puramente o entretenimento (BENIGNO *et al.*, 2024).

5.2 Atividade: Captura de Conhecimento do Domínio

A captura de conhecimento do domínio teve como fonte primária a revisão terciária da literatura sobre processos de software para jogos digitais de BENIGNO et al. (2024). Essa revisão foi escolhida por três razões: a abrangência de seus resultados, que evidenciaram o interesse acadêmico pelo desenvolvimento de jogos sérios sob diferentes perspectivas; a disponibilização da lista completa de trabalhos analisados, o que permite a rastreabilidade da seleção; e a atualidade de seus dados, que cobrem um intervalo extenso de tempo.

Por se tratar de uma revisão terciária (uma revisão de estudos secundários), ela não fornece, diretamente, processos de desenvolvimento, mas um panorama das revisões existentes e dos estudos primários por elas agregados. Foi necessária, portanto, uma descida sistemática do nível das revisões secundárias para o nível dos estudos primários que propõem processos para jogos sérios. O percurso completo organiza-se em quatro estágios:

- Estágio 1: Identificação das revisões pertinentes. Dos 22 estudos secundários analisados em BENIGNO et al. (2024), 11 tratavam de jogos sérios. Esses 11 trabalhos foram recuperados e tiveram suas seções de resultados e discussão analisadas em busca de referências diretas a artigos propositores de processos de software para jogos sérios. Ao final, 6 revisões foram retidas; as 5 demais foram excluídas por não apresentarem, nas seções analisadas, referências diretas a artigos contendo documentação de processos;

- Estágio 2: Identificação e recuperação dos estudos primários. A partir das 6 revisões retidas, foram identificados 70 trabalhos primários potenciais, apresentados na Tabela 5.1 juntamente com a revisão de origem e a quantidade de

candidatos por revisão. A documentação com os trechos exatos em que cada artigo foi citado, bem como suas referências, está disponível *online*[2]. Os trabalhos foram recuperados por busca direta na internet; três trabalhos não foram localizados, resultando em 67 estudos primários efetivamente recuperados;

- Estágio 3: Avaliação de qualidade. Os 67 estudos primários recuperados foram lidos e submetidos a um instrumento de avaliação de qualidade desenvolvido para esta pesquisa, voltado a aferir a adequação de cada trabalho aos fins de construção do catálogo e da linha de processos. O instrumento opera em três fases. Na primeira, critérios de exclusão eliminam, sem pontuação, trabalhos cujo processo não esteja documentado ou esteja descrito apenas superficialmente, que não apresentem metodologia ou que sejam duplicados. Na segunda, admite-se o trabalho que defina claramente fases, atividades ou unidades de trabalho do processo (critério primário) ou que atenda a pelo menos dois critérios complementares: especificação de papéis e responsabilidades, documentação dos artefatos de cada fase, detalhamento do fluxo de trabalho e descrição de técnicas e ferramentas. Na terceira, os trabalhos admitidos recebem uma pontuação acumulada distribuída em três dimensões (documentação do processo, validação empírica e aspectos pedagógicos), sujeita a penalidades quando as etapas não são detalhadas (10) ou quando a abordagem é puramente técnica, sem considerações educacionais (15). A pontuação final classifica cada trabalho em alta (maior ou igual a 100 pontos), média (70 a 99) ou básica qualidade (40 a 69, ou atendimento aos requisitos mínimos de inclusão); trabalhos com menos de 40 pontos ou que não atingem os requisitos mínimos são classificados como inadequados e descartados. A descrição completa do instrumento[3] e os resultados[4] de sua aplicação estão disponíveis *online*. Após a avaliação, 28 dos 67 trabalhos foram classificados como aptos a compor o corpo de processos da LPrS-JS, conforme a Tabela 5.2;

- Estágio 4: Seleção do corpo fundador. Dentre as 28 publicações aptas, 9 foram selecionadas de forma intencional para compor o corpo fundador da linha. A seleção foi orientada por três critérios: diversidade de área de aplicação dos jogos sérios (educação formal, saúde e reabilitação física), variedade de nível de detalhamento documental, deliberadamente incluindo trabalhos de diferentes níveis de qualidade, de modo a expor o método à heterogeneidade real do domínio; e representatividade do conjunto. Os processos selecionados e os apelidos atribuídos a cada um encontram-se na Tabela 5.3.

[2] `https://doi.org/10.5281/zenodo.19987537`
[3] `https://doi.org/10.5281/zenodo.19987818`
[4] `https://doi.org/10.5281/zenodo.19987893`

Tabela 5.1: Trabalhos primários potenciais identificados.

ID em BE-NIGNO et al. 2024	**Referência**	**Nº de trabalhos primários potenciais**
A6	Approaches for serious game design: A systematic literature review - Avila-Pesántez, D., Rivera, L. A., & Alban, M. S. (2017)	11
A7	Developing games for non-leisure contexts: Identification of challenges and research gaps - Backlund, P., Engström, H., Berg Marklund, B., & Toftedahl, M. (2017)	8
A10	Developing theory-driven, evidence-based serious games for health: Framework based on research community insights - Verschueren, S., Buffel, C., & Stichele, G. V. (2019)	17
A12	Serious Games: Review of methodologies and Games engines for their development - Tomala-Gonzales, J., Guaman-Quinche, J., Guaman-Quinche, E., Chamba-Zaragocin, W., & Mendoza-Betancourt, S. (2020)	11
A15	Software Engineering Frameworks Used for Serious Games Development in Physical Rehabilitation: Systematic Review - Ambros-Antemate, J. F., Beristain-Colorado, M. D. P., Vargas-Trevino, M., Gutierrez-Gutierrez, J., Hernandez-Cruz, P. A., Gallegos-Velasco, I. B., & Moreno-Rodríguez, A. (2021)	8
A20	SCHEMA: A Process for the Creation and Evaluation of Serious Games-A Systematic Review towards Sustainability - Merino-Cajaraville, A., Reyes-de-Cózar, S., & Navazo-Ostúa, P. (2023)	15

A documentação, com os grifos de onde os artigos foram citados em cada uma das 6 revisões analisadas, bem como suas referências, encontra-se disponível em tabela disponível *online* [5].

Após essa análise, os trabalhos foram recuperados a partir de pesquisas na internet. Os artigos de IDs P15, P47 e P59, conforme a a Tabela anexa Coleta de Artigos, não foram encontrados - totalizando, portanto, 67 possíveis artigos propositores. Es-

[5] `https://doi.org/10.5281/zenodo.19987537`

ses 67 trabalhos foram lidos e avaliados frente a um procedimento de avaliação de qualidade desenvolvido e que funcionou como um critério de filtragem de artigos. A descrição do método de avaliação encontra-se em arquivo *online*[6]. O resultado de sua aplicação nos 67 artigos encontra-se, também, em planilha disponibilizada *online*[7].

Após a execução do procedimento de avaliação, 28 trabalhos foram considerados aptos para a composição do corpo de processos para a construção da LPrS-JS, conforme a Tabela 5.2.

Tabela 5.2: Avaliação de qualidade dos trabalhos primários selecionados.

Artigo	**Avaliação**
Towards a Methodology for Serious Games Design for Children with Auditory Impairments (2016)	ALTA QUALIDADE
GAMED: digital educational game development methodology (2015)	QUALIDADE MÉDIA
Design methodology for educational games based on graphical notations: Designing Urano (2017)	ALTA QUALIDADE
A serious game development process using competency approach. Case Study: Elementary School Math (2014)	QUALIDADE MÉDIA
An activity theory-based model for serious games analysis and conceptual design (2015)	QUALIDADE BÁSICA
EMERGO: A methodology and toolkit for efficient development of serious games in higher education (2008)	ALTA QUALIDADE
The QUARTIC Process Model for Developing Serious Games: 'Green My Place' Case Study (2014)	ALTA QUALIDADE
PROGame: A process framework for serious game development for motor rehabilitation therapy (2018)	ALTA QUALIDADE
A Multidisciplinary Approach To Serious Game Development In The Health Sector (2017)	QUALIDADE MÉDIA

Continua na próxima página

[6] https://doi.org/10.5281/zenodo.19987818
[7] https://doi.org/10.5281/zenodo.19987893

Tabela 5.2 – continuação da página anterior

Artigo	**Avaliação**
Hapop-Physio: a computer game to support cognitive therapies in children (2017)	QUALIDADE MÉDIA
Tools and Methods for Efficiently Designing Serious Games (2010)	QUALIDADE BÁSICA
Using Serious Games to Create Awareness on Visual Impairments (2015)	QUALIDADE BÁSICA
Development of Encantasya: War of the Four Kingdoms (2017)	QUALIDADE BÁSICA
Diseño Y Desarrollo De Un Videojuego Educativo Con Técnicas De Inteligencia Artificial Para La Plataforma Android Aplicando La Metodología Oohdm. Caso De Estudio: Laberinto En 3D (2014)	QUALIDADE BÁSICA
Chances for Serious Games in Rehabilitation of Stroke Patients on the Example of Utilizing the Wii Fit Balance Board (2013)	QUALIDADE BÁSICA
Exergaming and rehabilitation: A methodology for the design of effective and safe therapeutic exergames (2016)	ALTA QUALIDADE
Reha@Stroke - A Mobile Application to Support People Suffering from a Stroke Through Their Rehabilitation (2019)	QUALIDADE MÉDIA
Supporting Seniors Rehabilitation through Videogame Technology: A Distributed Approach (2012)	QUALIDADE MÉDIA
A design process for balanced educational video games with collaborative activities (2015)	QUALIDADE MÉDIA
A Framework to Design, Develop, and Evaluate Immersive and Collaborative Serious Games in Cultural Heritage (2017)	ALTA QUALIDADE
eLuna: A Co-Design Framework for Narrative Digital Game-Based Learning that Support STEAM (2022)	QUALIDADE MÉDIA

Continua na próxima página

Tabela 5.2 – continuação da página anterior

Artigo	**Avaliação**
Gamethinking: a roadmap to a Design Thinking-based model for Game Development education (2018)	QUALIDADE BÁSICA
Determinación de criterios generales para el diseño de juegos serios: modelo metodológico integrador (2021)	QUALIDADE BÁSICA
iPlus a User-Centered Methodology for Serious Games Design (2020)	ALTA QUALIDADE
Proposal of a Conceptual Model for Serious Games Design: A Case Study in Children With Learning Disabilities (2019)	ALTA QUALIDADE
Serious Game Development Model Based on the Game-Based Learning Foundation (2021)	QUALIDADE MÉDIA
Modelo de serious game para mejorar la atención en niños con trastorno por déficit de atención e hiperactividad (TDAH) (2018)	QUALIDADE BÁSICA
Towards a design approach for serious games (2016)	QUALIDADE MÉDIA

Dentre as 28 publicações aptas, 9 publicações foram escolhidas para compor a LPrS, abrangendo diferentes níveis de qualidade de documentação e diferentes áreas de atuação dos JS. Os artigos propositores escolhidos, bem como os *apelidos* definidos para seus processos, encontram-se na Tabela 5.3, bem como suas contextualizações e descrições encontram-se no Apêndice B.

Tabela 5.3: Relação de apelidos e referências dos trabalhos selecionados.

Apelido	**Trabalho e Referência**
GAMED	GAMED: digital educational game development methodology (ASLAN e BALCI, 2015)
Reha	Reha@Stroke - A Mobile Application to Support People Suffering from a Stroke Through Their Rehabilitation (BARANYI *et al.*, 2019)
iPlus	iPlus a User-Centered Methodology for Serious Games Design (CARRIÓN-TORO *et al.*, 2020)
eLuna	eLuna: A co-design framework for narrative digital game-based learning that supports STEAM (BREIEN e WASSON, 2022)
Meconesis	Towards a Methodology for Serious Games Design for Children with Auditory Impairments (CANO *et al.*, 2016)
Quartic	The QUARTIC Process Model for Developing Serious Games: 'Green My Place' Case Study (COWLEY, 2014)
Urano	Design methodology for educational games based on graphical notations: Designing Urano (PRIETO *et al.*, 2017)
PROGame	PROGame: A process framework for serious game development for motor rehabilitation therapy (ALCOVER *et al.*, 2018)
Exergame	Exergaming and rehabilitation: A methodology for the design of effective and safe therapeutic exergames (PIROVANO *et al.*, 2016)

5.3 Atividade: Extração de Elementos de Processo

Nesta etapa, para todos os nove processos selecionados, os elementos dos tipos unidades de trabalho (atividade e tarefa), papel e artefato foram extraídos. A extração ocorreu a partir da leitura das documentações disponíveis e dependeu do grau de detalhamento de cada publicação. Para alguns processos, em razão da falta de informações que comprometeriam o entendimento da completude do procedimento de construção, alguns elementos foram criados de modo indutivo, observando o fluxo das etapas de construção descritas nas publicações.

A extração dos elementos de processo foram representadas em uma tabela, com uma página para cada processo, conforme o modelo proposto. A extração completa dos elementos de processo encontra-se disponível em tabela *online*[8].

[8] `https://doi.org/10.5281/zenodo.19988020`

5.4 Atividade: Definição de Glossário

De posse dos elementos extraídos, os termos relevantes foram descritos e registrados em glossários de acordo com as publicações capturadas. Os glossários foram registrados em uma tabela, com uma página referindo-se a cada processo da seleção. A tabela de definições encontra-se disponível *online*[9].

5.5 Atividade: Definição de Vocabulário Canônico

Com base no glossário e na tabela de extração, as macrofases foram identificadas e o Vocabulário Canônico foi desenvolvido. A saída desta etapa encontra-se em uma tabela disponível *online*[10].

5.5.1 Definição de Macrofases

Em razão de as macrofases serem elementos criados a partir da junção de outros elementos de processo, não existindo em seus processos originais, esta subseção dedica-se a expor detalhes de suas definições.

As macrofases foram escolhidas a partir do princípio de identificação, nos processos analisados, dos "pontos de inflexão" em que mudam simultaneamente o tipo de problema tratado, os artefatos gerados e os papéis predominantes sob uma ótica de mais alto nível.

Inicialmente, foi realizada a leitura geral para a observação dos rótulos de fase e dos produtos citados nos processos, presentes no artefatos de Extração e de Glossário. Esse levantamento mostrou padrões: todos começam com algum tipo de elucidação do problema e dos objetivos; todos passam por uma etapa em que o jogo ganha forma (*design*/prototipagem), ainda que com granularidade variada; alguns distinguem um momento de Engenharia de Software (especificação/arquitetura/projeto/*design* de software) separado do momento de construção; e todos, por caminhos diferentes, chegam a um uso em campo com avaliação e a um retorno estruturado de *feedback*.

Essas regularidades surgem mesmo quando os termos mudam: "Preparation", "Identification + Pedagogical Objectives", "Research / Idea definition", "Problem formulation; Game idea generation; Requirements development", "Analysis", "Prephases 1-3 (Educational challenges; Type of game; Initial story & characters)", "Order 1 - Requirements", "1) Exercise definition" e "0) Project initiation" descrevem o mesmo gênero e indicam uma fronteira para uma primeira macrofase, por exemplo; o mesmo ocorre nas demais etapas.

[9] `https://doi.org/10.5281/zenodo.19988264`
[10] `https://doi.org/10.5281/zenodo.19988395`

No segundo movimento, os rótulos foram agrupados em eixos semânticos que fossem, ao mesmo tempo, moderados e expressivos. O critério de moderação limitou o número de macrofases ao menor conjunto capaz de percorrer todo o ciclo com mudanças claras de natureza do trabalho. O critério de expressividade exigiu que cada macrofase correspondesse a uma transformação observável nas entradas e saídas.

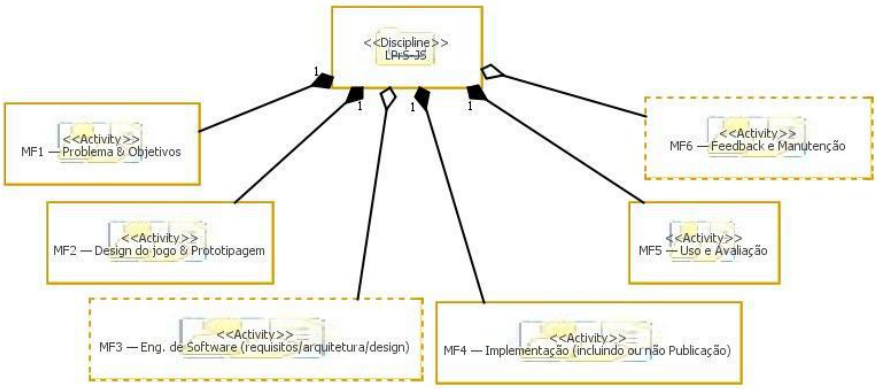
Dessa síntese, emergiram seis macrofases (Figura 5.1):

- **Problema & Objetivos**, ancorada na passagem de contexto e *stakeholders* para objetivos e requisitos iniciais;
- ***Design* do jogo & Prototipagem**, onde artefatos como GDD, roteiro, modelos de narrativa, mecânicas e protótipos tornam-se os produtos centrais;
- **Engenharia de Software (requisitos/arquitetura/*design*)**, que aparece explícita em alguns processos e, em outros, implícita nos artefatos;
- **Implementação & Publicação**, que concentra codificação, integração e empacotamento/implantação (foi mantida a fase "publicar" como atividade distinta, mas dentro da mesma macrofase, porque nem todas as fontes a descrevem como fase autônoma);
- **Uso em campo & Avaliação**, que inclui monitoramento, estudos com usuários/pacientes e análise de eficácia; e
- ***Feedback* & Sustentação**, que consolida resultados, refina o jogo e alimenta o *backlog*/planejamento de iterações seguintes.

Em alguns casos, a engenharia aparece diluída dentro do *design*; em outros, a publicação não se distingue da integração; em outros, ainda, a avaliação assume contornos clínicos.

Nas subseções seguintes, são apresentadas as macrofases definidas com mais detalhes.

Figura 5.1: Modelo de características com macrofases na notação *OdysseyProcess-FEX* (TEIXEIRA, 2011)



Fonte: Elaborado pelo autor (2026).

5.5.1.1. Do contexto à intenção técnica: MF1 (Problema & Objetivos)

O primeiro conjunto de atividades dá início ao entendimento do problema e a definição de objetivos. Aqui aparecem especificações de necessidade e escopo, perfis de público e *stakeholders*, bem como definições preliminares de restrições. No conjunto analisado, isso é materializado como: definições de objetivo terapêutico, "*client requirement*" ou *brief* inicial, listas de partes interessadas, necessidades e categorias de usuários, além de perfis/*roles* e metas pedagógicas/lúdicas. Esses artefatos são a causa da primeira fronteira: quando o problema sai do contexto e entra no espaço de objetivos, a macrofase MF1 é definida.

5.5.1.2. Do desejo de jogo ao jogo concebido: MF2 (*Design* do jogo & Prototipagem)

O segundo conjunto de atividades marca a passagem do "*o que se quer*" para "*como será o jogo*". É aqui que surgem os documentos de *design* e os protótipos iniciais. Em várias fontes, isso aparece como GDD e roteiro (com visão, mecânicas e narrativa), modelos de cenário, modelos de tarefa, diagramas de capítulos/cenas e cartões de cena, agenda pedagógica, "*story model*", "*play specification*" e o meta-game. Em alguns processos focados na saúde, a "virtualização do exercício" dá a forma técnica para que o jogo terapêutico exista; em focos diferentes, outros observam conteúdos, entidades e protótipos de interface (UI) que assumem esse papel. Essa família de artefatos cria a segunda fronteira: quando existe um *design* consensual e demonstrável (texto e prototipagem), a MF2 foi cumprida.

5.5.1.3. Do *design* ao compromisso de engenharia: MF3 (Engenharia de Software)

A terceira fase ocorre quando o *design* de jogo exige formalização técnica: requisitos rastreáveis, arquitetura e modelos de dados/componentes. Em alguns processos,

isso aparece em modelos de análise/usuário/pedagógicos e UML, como entidades instanciadas (dados, sociais, tarefas, mecânicas) e conexão entre *play* e *behavlets*, ou, como refino de requisitos e "*component designs*". Esses artefatos definem a fronteira da MF3: é passado do "conceito do jogo" para o "*blueprint implementável*".

5.5.1.4. Do *blueprint* ao executável integrado: MF4 (Implementação & Publicação)

Com a engenharia estabelecida, o ciclo prossegue para a construção e integração. Aqui o artefato característico é um *build* executável integrado (ou um conjunto de componentes integrados) - mesmo que ainda não "publicado" em larga escala. Isso pode aparecer como protótipos/mini-jogos e itens de teste integrados, módulos de monitoramento integrados ao jogo terapêutico e o próprio exergame ou, componentes executáveis. Quando o executável existe e passa a ser publicável/implantável, nasce a segunda metade de MF4, o gesto de publicação/implantação (que muitos processos não tratam como fase separada). O artefato decisivo aqui é o executável integrado/pronto para publicar, ele prossegue o ciclo para uso.

5.5.1.5. Do uso às evidências: MF5 (Uso em campo & Avaliação)

O jogo sai do laboratório e gera evidências. Os artefatos são resultados de avaliação, em alguns casos, marcadamente clínicos, em outros, heurísticos/de experiência do usuário (UX), e há processos que registram a avaliação por meio de um plano de próxima iteração derivado de testes, o monitoramento em tempo real e o *feedback* corretivo produzem evidências durante as sessões no caso de aplicações para a saúde. A fronteira da MF5 é, portanto, a presença de resultados e sua interpretação, o jogo está sendo usado e medido.

5.5.1.6. Das evidências ao planejamento de mudança: MF6 (*Feedback* & Sustentação)

Por fim, as evidências estabilizam-se em decisões de manutenção e evolução. O artefato que caracteriza isso é o *backlog* de mudanças (ou planejamento de próxima iteração). O plano de refino em uma nova iteração é explícito em alguns processos; em outros, o refino do jogo pós-testes cumpre esse papel; em outros, a consolidação de resultados finaliza o ciclo. Quando o *feedback* vira decisão operacional (lista priorizada, plano, refino documentado), a MF6 se cumpre e o processo fica pronto para reiniciar o ciclo.

5.5.2 Subtividades canônicas

Uma vez estabelecidas as macrofases, este passo busca detalhar a estrutura interna de cada uma delas por meio da identificação e derivação de suas subatividades canônicas. O mapeamento desta fase buscou agrupar as diferentes atividades, tarefas, práticas e ações encontradas nos processos analisados em subatividades mais está-

veis e de propósito claro dentro do ciclo de vida do processo para JS. O resultado é um conjunto de subatividades canônicas que sintetiza o fluxo de trabalho, cada qual ancorada em sua respectiva macrofase.

Neste processo, decidiu-se pela observação apenas das duas primeiras macrofases encontradas, em razão de tratar, em geral, do entendimento do problema e da concepção geral do jogo. A análise completa encontra-se na tabela , e as descrições de cada subatividade são apresentadas a seguir.

A **Entender o problema**, relacionada à *MF1 - Problema & Objetivos*, reúne tarefas associadas à identificação do problema educacional ou de saúde, bem como do seu contexto, estado da arte e significância. É responsável por estabelecer a delimitação teórica e a justificativa para o uso do jogo em um determinado domínio, garantindo que o projeto nasça de uma necessidade real e fundamentada.

A **Identificar contexto e escopo**, relacionada à *MF1 - Problema & Objetivos*, reúne termos ligados à delimitação do domínio de aplicação, compreender o público-alvo, situação dos aprendizes e as características gerais do jogo/terapia.

A **Identificar stakeholders**, relacionada à *MF1 - Problema & Objetivos*, consolida atividades focadas em mapear e descrever todas as partes interessadas, além do perfil demográfico e situacional dos usuários finais. O resultado dessa atividade é uma definição de quem usará o jogo (aprendizes, pacientes) e quem tomará as decisões corporativas ou clínicas sobre o projeto, alinhando expectativas e restrições.

A **Descrever objetivos de aprendizagem**, relacionada à *MF1 - Problema & Objetivos*, agrupa denominações referentes ao estabelecimento de metas curriculares, competências e objetivos comportamentais ou terapêuticos. Cabe a essa subatividade determinar e documentar o que se espera que o usuário aprenda, pratique ou alcance ao final da experiência, servindo como referência norteadora para o *design* subsequente.

A **Identificar requisitos iniciais**, relacionada à *MF1 - Problema & Objetivos*, corresponde aos termos associados à coleta de necessidades, restrições e premissas. Consiste na elicitação de necessidades práticas, restrições e requisitos iniciais do sistema e do projeto.

O **Elaborar *Design* do Jogo e Elementos Pedagógicos**, relacionado à *MF2 - Design & Prototipagem (iterativa)*, consolida as tarefas para a concepção da ideia do jogo, história, personagens, mecânicas, regras, fluxo (gameplay) e integração pedagógica/terapêutica.

A **Desenvolver Protótipos**, relacionada à *MF2 - Design & Prototipagem (iterativa)*, reúne os processos de criação de versões iniciais, mockups ou protótipos de interface e componentes de software para tangibilizar o design.

Os **Avaliar/Testar e Refinar Iterativamente**, relacionados à *MF2 - Design & Prototipagem*, correspondem à condução de testes práticos com os protótipos

construídos para avaliar usabilidade, mecânicas e nível de engajamento. Consiste em testar os protótipos e as ideias de design com usuários ou especialistas e aplicar ajustes conforme o feedback (ciclo iterativo).

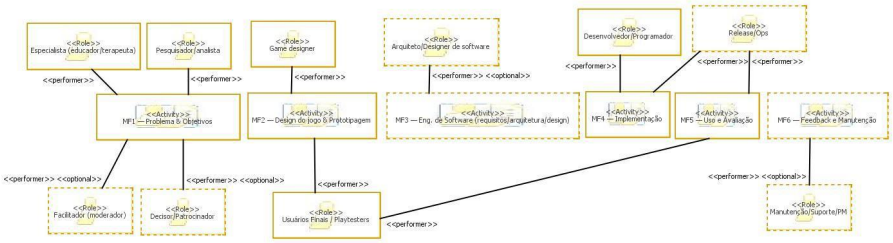
5.5.3 Definição de Vocabulário Canônico

Após a definição das macrofases, os outros elementos tiveram seus termos nativos mapeados para os termos canônicos. Nesta subseção, são apresentados os resultados desse mapeamento para as dimensões de papéis, artefatos, atividades e termos base, consolidando um vocabulário descritivo.

5.5.3.1 Papéis canônicos

O mapeamento de papéis buscou agrupar diferentes denominações encontradas nos processos analisados em funções mais estáveis, que aparecem recorrentemente na literatura e nos relatos documentados. O resultado é um conjunto de papéis canônicos que busca a sintetização destas funções, conforme apresetado no modelo de características da Figura 5.2.

Figura 5.2: Dimensão de papéis do modelo de características, apresentando suas opcionalidades e mandatoriedades.



Fonte: Elaborado pelo autor (2026).

O **Facilitador (moderador)** reúne termos relacionados à condução de *workshops*, dinâmicas de grupo e reuniões de co-criação. Esse papel é responsável por organizar as sessões e reuniões, explicitar objetivos, gerir o tempo, estimular a participação e registrar decisões e encaminhamentos. Sua atuação é central quando há diferentes *stakeholders* com formações distintas, pois ajuda a produzir saídas a partir de discussões coletivas.

O **Decisor/Patrocinador** consolida diferentes denominações para quem representa a instituição demandante (por exemplo, cliente, *product owner*, *decision maker*). Esse papel define e ajusta o escopo do projeto, aprova marcos intermediários, organiza prioridades e aceita ou rejeita entregas. Seu envolvimento deve reduzir o

risco de desalinhamento entre o produto desenvolvido e as expectativas institucionais ou organizacionais.

O **Especialista de Domínio (educador/terapeuta)** agrega termos que remetem a professores, terapeutas ou profissionais de saúde responsáveis pela definição dos objetivos pedagógicos ou terapêuticos e pela validação do conteúdo do jogo. Cabe a esse papel explicitar o que se espera que o público-alvo aprenda, pratique ou desenvolva, avaliar a adequação de narrativas, desafios e *feedbacks* e verificar a coerência do jogo com práticas reconhecidas na área.

O **Pesquisador/Analista** corresponde aos termos associados à construção de base teórica para o projeto. Esse papel realiza revisões de literatura, estudos de contexto, análises de soluções existentes e, quando necessário, coletas de dados. A partir desses insumos, contribui para a formulação do problema, a identificação de lacunas e o levantamento de requisitos, além de auxiliar na definição de protocolos de avaliação do jogo.

O ***Game designer*** agrupa denominações ligadas ao *design* da experiência de jogo. É o responsável por modelar mecânicas, regras, sistemas de progressão, *feedbacks* e elementos narrativos e estéticos, considerando tanto o engajamento quanto a intencionalidade do jogo sério. Sua atuação traduz objetivos de domínio em estruturas de jogo concretas, definindo como os jogadores interagem, avançam e recebem informações sobre seu desempenho.

O papel de **Usuários Finais/*Playtesters*** reúne denominações para o público-alvo que interage com protótipos e versões do jogo durante o desenvolvimento. Esses participantes fornecem *feedback* sobre compreensão das regras, engajamento, dificuldades de uso e percepção de utilidade. As evidências produzidas (observações, relatos, métricas de uso) são insumo para ajustes de design e priorização de melhorias.

O **Desenvolvedor/Programador** consolida termos relativos à implementação de software. É responsável por codificar funcionalidades, integrar componentes, tratar questões de desempenho e garantir que o jogo seja executável nas plataformas. Em jogos sérios, esse papel também implementa mecanismos de coleta de dados de uso, integrações com sistemas externos e funcionalidades de suporte à avaliação.

O **Arquiteto/*Designer* de software** agrupa denominações associadas à definição da estrutura técnica da solução. Esse papel decide como o sistema será decomposto em módulos, quais tecnologias serão utilizadas, como será feita a comunicação entre componentes e quais padrões de projeto serão adotados.

O papel ***Release*/Ops** sintetiza termos ligados ao empacotamento, distribuição e operação do jogo em ambiente de uso. Inclui a preparação de builds, configuração de ambientes (servidores, serviços em nuvem, laboratórios), publicação em canais apropriados e monitoramento de disponibilidade. Em contextos educacionais ou

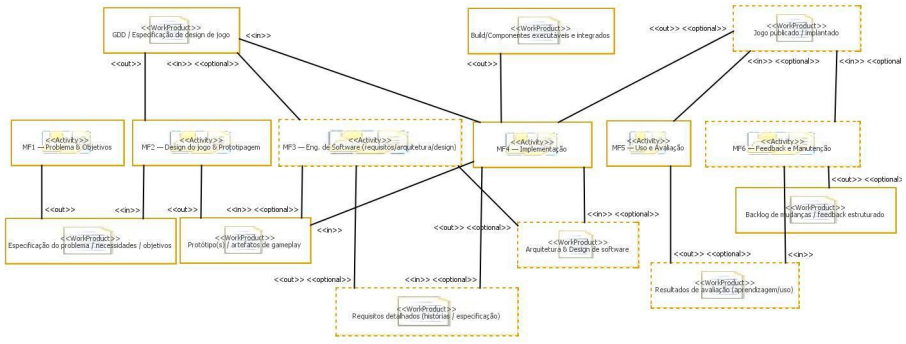
clínicos, essa função pode considerar calendários e rotinas institucionais.

Por fim, o papel **Manutenção/Suporte/PM** reúne denominações relacionadas à coordenação da evolução do produto após sua disponibilização inicial. Envolve organizar e priorizar o conjunto de mudanças estipulado, acompanhar incidentes e relatos de problemas, articular equipes técnicas e de campo e planejar ciclos de atualização, mantendo o jogo alinhado às necessidades do contexto de uso.

5.5.3.2 Artefatos canônicos

O mapeamento de artefatos concentrou termos nativos que se referiam a produtos de trabalho com funções semelhantes, ainda que nomeados de maneiras diferentes nos processos analisados. O resultado é um conjunto de artefatos canônicos que descreve o fluxo de informação e troca de dados ao longo dos processos. O modelo de características contendo os artefatos canônicos é apresentado na Figura 5.3.

Figura 5.3: Dimensão de artefatos do modelo de características, apresentando suas opcionalidades, mandatoriedades e relações de entrada e saída.



Fonte: Elaborado pelo autor (2026).

A **Lista de participantes** reúne todos os indivíduos que participam do processo de desenvolvimento: stakeholders, profissionais de domínio, representantes de usuários etc.

A **Especificação do problema/necessidades/objetivos** reúne documentos que descrevem o problema a ser enfrentado, o contexto de aplicação, o público-alvo, as principais necessidades identificadas e os objetivos de aprendizagem ou de impacto. Este artefato busca explicitar escopo, critérios de sucesso e restrições, servindo de referência para o alinhamento entre decisores, especialistas de domínio e equipe de desenvolvimento.

O **GDD/Especificação de *design* de jogo** consolida diferentes rótulos para documentos de *design* de jogo. Trata-se do documento em que se registram visão, mecânicas, fluxos de progressão, regras, *feedbacks*, narrativa, estética e principais elementos de interface. Nos jogos sérios, esse artefato também pode explicitar o vínculo entre objetivos de domínio e estruturas de jogo, indicando, por exemplo,

quais desafios estão associados a quais competências.

Os **Protótipos/artefatos de *gameplay*** agregam termos referentes a protótipos em papel, *wireframes*, *mockups*, versões jogáveis parciais e outros materiais usados para experimentar a experiência de jogo antes da implementação completa. Esses artefatos permitem testar ideias, identificar problemas de usabilidade e verificar a aderência entre a proposta de jogo e os objetivos educacionais ou terapêuticos.

O artefato de **Requisitos detalhados (histórias/especificação)** reúne representações estruturadas das funcionalidades e restrições do sistema, tais como histórias de usuário, casos de uso ou especificações formais. Contém requisitos funcionais e não funcionais rastreáveis, descrevendo comportamentos esperados, condições de qualidade, limitações de plataforma e necessidades de integração.

A **Arquitetura & *Design* de software** consolida documentos que descrevem a estrutura técnica da solução: módulos, camadas, componentes, interfaces e tecnologias adotadas. Esse artefato orienta a implementação, facilita a análise de impactos de mudanças e apoia a manutenção, agrupando diferentes formas de representação arquitetural.

Os ***Builds*/Componentes executáveis e integrados** representam versões executáveis do jogo em diferentes estágios (protótipos funcionais, versões de teste, *releases*). Esse artefato sintetiza o estado atual da integração entre componentes e serve como base para testes, demonstrações e implantação em ambientes de uso.

O **Jogo publicado/implantado** congrega denominações que se referem à versão do jogo efetivamente disponibilizada ao público-alvo, seja por meio de ambientes institucionais, plataformas de aprendizagem, lojas de aplicativos ou outros canais. Inclui a configuração do ambiente de execução e, quando pertinente, materiais de apoio à adoção.

Os **Resultados de avaliação (aprendizagem/uso)** reúnem relatórios, planilhas e demais registros que documentam evidências de eficácia, usabilidade e aceitação. Esse artefato reúne dados quantitativos e qualitativos coletados em estudos de uso, permitindo examinar se os objetivos definidos na especificação inicial foram atendidos.

Por fim, o ***Backlog* de mudanças/*feedback* estruturado** agrupa listas de demandas de correção, melhorias e extensões do jogo, oriundas de *feedback* de usuários, resultados de avaliação ou decisões de pesquisa. O mapeamento permite reunir registros de problemas em uma noção de *backlog* que orienta ciclos posteriores de desenvolvimento e manutenção.

5.6 Atividade: Definição de Mandatoriedade

Nesta etapa, os termos obtidos na fase de composição do vocabulário canônico foram agrupados de acordo com sua macrofase, e foi determinada a mandatoriedade dos termos relacionando a sua frequência de aparição dos termos entre os processos (disponível em tabela *online*[11]).

A mandatoriedade foi determinada se um item está presente em pelo menos 60% dos processos: sendo equivalente a ocorrência do termo em no mínimo 6 dos 9 processos. Os termos que estavam presentes em menos de 6 processos são considerados opcionais.

Toda a comparação é organizada em um conjunto de tabelas com quatro dimensões, cada uma com o seguinte objetivo:

1. **Mapa de macrofases:** guarda o alinhamento de cada processo às macrofases canônicas, além de uma coluna de *Descrição* onde contém detalhes daquela macrofase;

2. **Papéis por macrofase:** reúne, por papel canônico, os equivalentes encontrados em cada processo, além de uma coluna de *Descrição,* que descreve as atribuições daquele papel no processo;

3. **Artefatos por macrofase:** agrega, para cada artefato canônico, os artefatos citados por processo e *Descrição* que apresenta o sentido daquele artefato no quadro comparativo;

4. **Subatividades por macrofase:** apresenta, para cada macrofase canônica estudada, descrições com relação às tarefas realizadas na etapa.

5.7 Atividade: Definição de Variabilidades

Nesta etapa, as variabilidades da linha são identificadas e classificadas. Com base nos achados da mandatoriedade dos elementos, na etapa anterior, são definidos como pontos de variação obrigatórios os elementos mandatórios, e, os demais, opcionais.

Nesta execução, a etapa foi executada apenas para os elementos de subatividades. Os artefatos completos produzidos por essa fase encontram-se disponíveis em tabela *online*[12].

[11] `https://doi.org/10.5281/zenodo.20480392`
[12] `https://doi.org/10.5281/zenodo.19988737`

5.7.1 Agrupamento

Nesta subfase, o termo canônico é mapeado de volta para os termos nativos de cada processo fonte. Esses termos nativos representam as variantes brutas para cada termo canônico. Aqui, foi escolhido o trabalho apenas com as subatividades definidas anteriormente na execução deste método.

A Tabela 5.4 apresenta a estrutura do artefato gerado por este agrupamento, contendo o tipo de elemento, o termo canônico, a classificação de mandatoriedade, os processos que possuem o elemento e as variantes brutas originárias.

Tabela 5.4: Mapeamento detalhado dos elementos brutos para as subatividades canônicas.

Tipo	**Elemento Canônico**	**Class.**	**Processos**	**Elementos brutos**
Subatividade	1.1 Entender o problema	Mandatório	eLuna, iPlus, Reha, GAMED, MECONESIS, URANO, QUARTIC, EXERGAME, PROGame	• *Preparation (Summarize goals)* (eLuna) • Definir o problema geral (educacional/necessidade) (iPlus) • Revisão básica da literatura; Pesquisa (Reha) • *Problem formulation; Identify education problem; Identify state of the art* (GAMED) • *Observational studies; Interviews* (MECONESIS) • *Educational challenges* (URANO) • Receber do cliente objetivo comportamental ou fragmento de cenário (QUARTIC) • *Select therapy goal/domain* (EXERGAME) • *Identify need* (PROGame)

Continua na próxima página

Tabela 5.4 – continuação da página anterior

Tipo	**Elemento Canônico**	**Class.**	**Processos**	**Elementos brutos**
Subatividade	1.2 Identificar contexto e escopo	Mandatório	eLuna, GAMED, MECONESIS, URANO, EXERGAME, PROGame	• *Define content areas; Describe learner demographic/situation* (eLuna) • *Bound domain & collect data; Assess suitability* (GAMED) • *Analisis* (contexto implícito) (MECONESIS) • *Determine game characteristics; Type of game* (URANO) • *Select therapy domain* (EXERGAME) • *Select therapy* (PROGame)
Subatividade	1.3 Identificar *stakeholders*	Opcional	iPlus, GAMED, MECONESIS, PROGame	• Identificar participantes para as sessões (iPlus) • *Identify stakeholders & decision makers* (GAMED) • *Selecionar roles* e perfil do usuário (MECONESIS) • Identificar *stakeholders* e categorias de usuários (PROGame)

Continua na próxima página

Tabela 5.4 – continuação da página anterior

Tipo	**Elemento Canônico**	**Class.**	**Processos**	**Elementos brutos**
Subatividade	1.4 Descrever objetivos de aprendizagem	Mandatório	eLuna, iPlus, GAMED, MECONESIS, URANO, QUARTIC, EXERGAME	• *Define learning objectives* (eLuna) • Definir objetivos pedagógicos gerais e específicos (iPlus) • *Specify learning objectives* (GAMED) • *Define pedagogical/ludic strategies* (MECONESIS) • *Define competences & educational objectives* (URANO) • Receber do cliente objetivo comportamental ou fragmento de cenário (QUARTIC) • *Select therapy goal; Define parameters & metrics* (EXERGAME)
Subatividade	1.5 Identificar requisitos iniciais	Mandatório	iPlus, Reha, GAMED, QUARTIC, EXERGAME, PROGame	• Registrar necessidades e restrições (iPlus) • *Brainstorm*, Derivar requisitos iniciais (Reha) • *Elicit needs, objectives, constraints, assumptions* (GAMED) • Produzir *rough requirement* (QUARTIC) • *Decompose actions* (EXERGAME) • *Identify constraints; Initial requirements* (PROGame)

Continua na próxima página

Tabela 5.4 – continuação da página anterior

Tipo	**Elemento Canônico**	**Class.**	**Processos**	**Elementos brutos**
Subatividade	2.1 Elaborar *Design* do Jogo e Elementos Pedagógicos	Mandatório	eLuna, iPlus, GAMED, MECONESIS, URANO, QUARTIC, EXERGAME, PROGame	• *Co-Design; Co-Specification* (eLuna) • *Ludic Game Script* (iPlus) • *Game idea generation; Game design (spiral)* (GAMED) • *Pre-producción* (MECONESIS) • *pre-phase* 2 a *phase* 6 (URANO) • *Pedagogy schedule; Meta-game; Story model...* (QUARTIC) • *Virtualization (VE) + Game design* (EXERGAME) • *Modeling*: Desenhar conteúdos da aplicação... (PROGame)
Subatividade	2.2 Desenvolver Protótipos	Mandatório	eLuna, iPlus, Reha, GAMED, QUARTIC, EXERGAME, PROGame	• *Co-Design* (prototipagem implícita) (eLuna) • *GamePlay* (iPlus) • Desenvolvimento iterativo do protótipo (Reha) • *Game design (spiral)* (GAMED) • *Compose Behavlets* (QUARTIC) • *Virtualization* (EXERGAME) • Prototipar interface, Implementar componentes (PROGame)
Subatividade	2.3 Avaliar/Testar e Refinar Iterativamente	Opcional	Reha, GAMED, MECONESIS, EXERGAME	• Desenvolvimento iterativo (Reha) • *Game design (spiral)* [*spiral* implica teste] (GAMED) • Capturar interações no modelo de tarefas (CTT) (MECONESIS) • Testes iniciais e refinar (EXERGAME)

5.7.2 Definição de Variantes

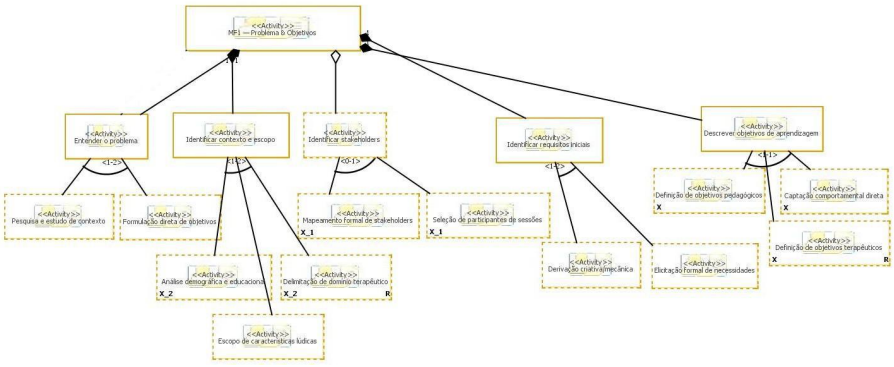
Nesta subfase, as variantes brutas de cada subatividade proveniente do artefato produzido na subfase anterior são avaliadas com base em sua descrição nos glossários, sendo realizadas análises semânticas para a identificação de propósitos e métodos de trabalho semelhantes nos conceitos desses termos brutos.

Os resultados dessa análise geram subagrupamentos de variantes brutas, formando as variantes finais (ou abstratas) da linha. No momento da derivação, o engenheiro de processo pode fazer a seleção apenas das abordagens coerentes para instanciar naquele ponto de variação (por exemplo, optar por um design focado em "*Co-Design*"ou focado em "Modelagem Formal", ou uma combinação dos dois, caso possível), escolhendo apenas as variantes coerentes para o projeto.

Essas variantes finais são documentadas em uma tabela com campos para o nome do agrupamento abstrato (a variante final), as variantes brutas originárias, as observações a respeito da análise semântica e as regras de uso, disponível em tabela *online*[13].

Para uma melhor visualização, as Figura 5.4 e 5.5 apresentam os modelos de características das macrofases 1 e 2, contendo as subatividades e suas possíveis variantes.

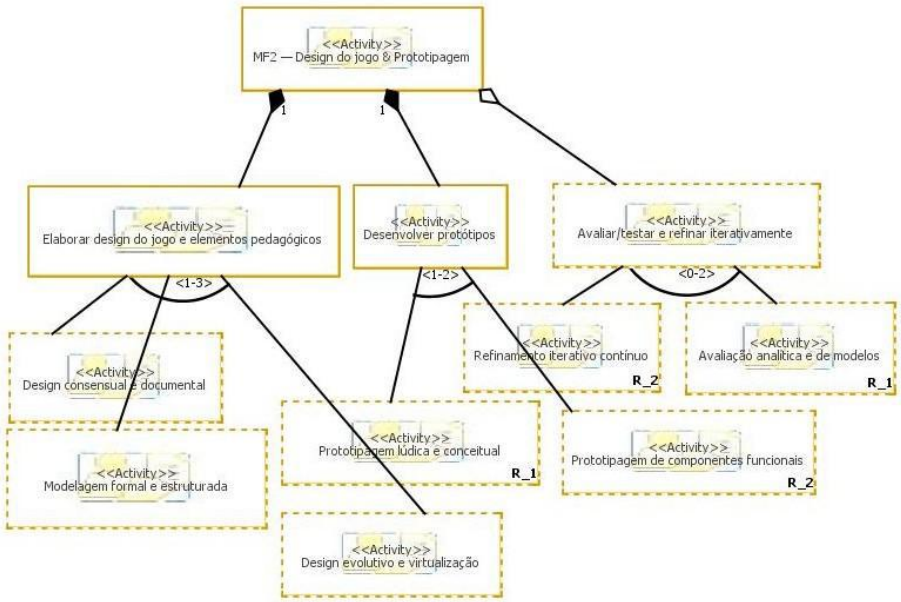
Figura 5.4: Dimensão da macrofase 1 do modelo de características, apresentando suas possíveis variantes, cardinalidades, opcionalidades e mandatoriedades.



Fonte: Elaborado pelo autor (2026).

[13] https://doi.org/10.5281/zenodo.19988737

Figura 5.5: Dimensão da macrofase 2 do modelo de características, apresentando suas possíveis variantes, cardinalidades, opcionalidades e mandatoriedades.



Fonte: Elaborado pelo autor (2026).

A seguir, são apresentados os pontos de variação e as suas possíveis variantes, bem como recomendações de uso e regras de composição, quando se aplicarem.

Ponto de variação: 1.1 Entender o problema

Variantes Obrigatórias (1..2):

1. Variante: Pesquisa e Estudo de Contexto - Foco em investigar o estado da arte e observar ativamente o ambiente - Usar se o problema ainda é abstrato e precisa de levantamento empírico prévio;

2. Variante: Formulação Direta de Objetivos - Foco na definição sumarizada e recebimento direto de metas - Usar se o cliente ou especialista já possui o problema ou a terapia bem delimitada.

Ponto de variação: 1.2 Identificar contexto e escopo

Variantes Obrigatórias (1..2):

1. Variante: Análise Demográfica e Educacional - Foco em descrever características do aluno e áreas de conteúdo - Usar em jogos educativos onde o perfil sociodemográfico do aprendiz dita a estratégia de design. *Mutuamente exclusiva com variante Delimitação de Domínio Terapêutico*;

2. Variante: Delimitação de Domínio Terapêutico - Foco estrito em selecionar a terapia ou limitação clínica abordada - Usar exclusivamente em exergames

ou jogos de reabilitação/saúde. *Mutuamente exclusiva com variante Análise Demográfica e Educacional. Requer seleção de variante Definição de Objetivos Terapêuticos*;

3. Variante: Escopo de Características Lúdicas - Foco em determinar características centrais do jogo de imediato - Usar se as restrições da plataforma ou do gênero do jogo impactarem as decisões iniciais do projeto.

Ponto de variação: 1.3 Identificar *stakeholders*

Variantes Opcionais (0..1):

1. Variante: Mapeamento Formal de *Stakeholders* - Foco na elicitação de clientes, decisores e perfis representativos - Usar em projetos formais e complexos que necessitam de rastreabilidade de aprovações. *Mutuamente exclusiva com variante Seleção de Participantes de Sessões*;

2. Variante: Seleção de Participantes de Sessões - Foco logístico em selecionar participantes para dinâmicas - Usar em metodologias participativas baseadas primariamente em sessões de co-design (workshops). *Mutuamente exclusiva com variante Mapeamento Formal de Stakeholders.*

Ponto de variação: 1.4 Descrever objetivos de aprendizagem

Variantes Obrigatórias (1..1):

1. Variante: Definição de Objetivos Pedagógicos - Foco em metas de aprendizagem, estratégias de ensino e habilidades - Usar em aplicações educacionais e escolas. *Mutuamente exclusiva com variante Captação Comportamental Direta*;

2. Variante: Definição de Objetivos Terapêuticos - Foco em parâmetros clínicos, movimento e segurança do paciente - Usar estritamente em jogos clínicos, físicos e de saúde. *Requer seleção de variante Delimitação de Domínio Terapêutico*;

3. Variante: Captação Comportamental Direta - Foco em absorver a mudança de comportamento requerida pelo cliente - Usar em projetos corporativos ou simulações comportamentais rápidas. *Mutuamente exclusiva com variante Definição de Objetivos Pedagógicos.*

Ponto de variação: 1.5 Identificar requisitos iniciais

Variantes Obrigatórias (1..2):

1. Variante: Elicitação Formal de Necessidades - Foco em documentar formalmente pressupostos e requisitos do software - Usar se o projeto demandar princípios clássicos de engenharia de software e contratos;

2. Variante: Derivação Criativa/Mecânica - Foco em derivar as restrições a partir de ideação ou análise biomecânica - Usar em projetos onde o requisito nasce de uma prática mais humanizada que deve ser virtualizada.

Ponto de variação: 2.1 Elaborar Design do Jogo e Elementos Pedagógicos
Variantes Obrigatórias (1..3):

1. Variante: Design Consensual e Documental - Foco em criar scripts, blueprints e GDDs mutuamente acordados em equipe - Usar para alinhar times mistos (professores e devs) através de documentação visual e compreensível;

2. Variante: Modelagem Formal e Estruturada - Foco na documentação técnica profunda, modelos de interface e árvores narrativas - Usar em arquiteturas complexas ou jogos que exigem amarração sistemática e profunda de dados;

3. Variante: Design Evolutivo e Virtualização - Foco em evoluir a ideia iterativamente ou basear o design na transposição virtual - Usar se a mecânica núcleo (ex: exercício) for prioritária e ditar a roupagem do design final.

Ponto de variação: 2.2 Desenvolver Protótipos
Variantes Obrigatórias (1..2):

1. Variante: Prototipagem de Componentes Funcionais - Foco na implementação rápida de código e componentes executáveis na engine - Usar se a equipe necessitar avaliar a viabilidade e funcionalidade técnica da simulação desde cedo no código. *Requer seleção de variante Refinamento iterativo contínuo*;

2. Variante: Prototipagem Lúdica e Conceitual - Foco em dinâmicas de papel, LEGO, mockups ou modelos de lógica abstrata - Usar para tangibilizar dinâmicas para stakeholders (sem custo de programação) antes do desenvolvimento pesado. *Requer seleção de variante Avaliação analítica e de modelos.*

Ponto de variação: 2.3 Avaliar/Testar e Refinar Iterativamente
Variantes Opcionais (0..2):

1. Variante: Refinamento Iterativo Contínuo - Foco em testar protótipos dinamicamente com usuários/especialistas e ajustar fisicamente - Usar como base de qualidade no desenvolvimento ágil ou na adaptação de jogos de saúde com acompanhamento prático. *Requer seleção de variante Prototipagem de Componentes Funcionais*;

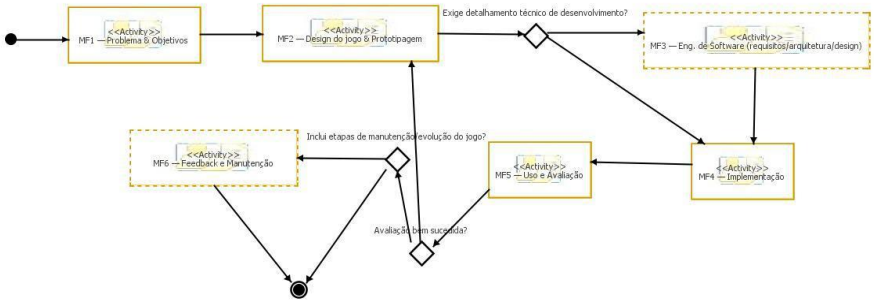
2. Variante: Avaliação Analítica e de Modelos - Foco em validar a usabilidade verificando a consistência lógica dos modelos estruturais criados - Usar para atestar a consistência formal antes de mobilizar recursos na geração de código compilado. *Requer seleção de variante Prototipagem Lúdica e Conceitual.*

5.8 Atividade: Modelagem

Nesta última etapa, com base nos artefatos de domínio identificados, foram construídos os Modelos de Características na ferramenta *Odyssey*, utilizando a linguagem *OdysseyProcess-FEX* (TEIXEIRA, 2011). Ao longo das seções anteriores, os Modelos de Características Estruturais foram apresentados, de acordo com cada etapa de construção da LPrS-JS.

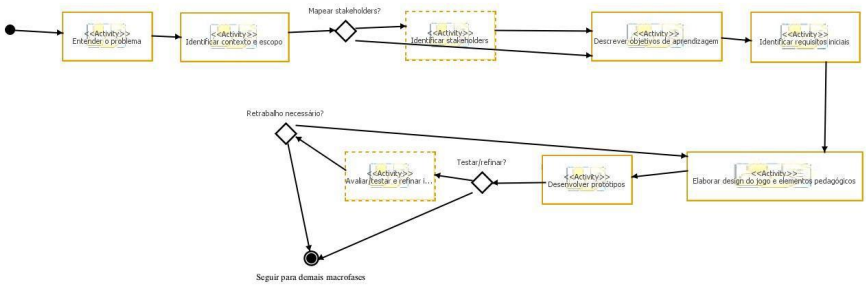
Os diagramas dos Modelos de Características Comportamentais relacionados às macrofases (Figura 5.6) e às subatividades relacionadas às macrofases 1 e 2 (Figura 5.7) são apresentados nas Figuras a seguir.

Figura 5.6: Diagrama do Modelo de Características comportamental de alto nível considerando apenas as macrofases definidas.



Fonte: Elaborado pelo autor (2026).

Figura 5.7: Diagrama do Modelo de Características comportamental de alto mais baixo, considerando apenas as subatividades das macrofases 1 e 2.



Fonte: Elaborado pelo autor (2026).

Capítulo 6

Instanciação

A LPrS-JS documentada no Capítulo 5 foi construída a partir da análise de diferentes processos descritos na literatura, baseando-se no método proposto no Capítulo 4. O resultado é uma LPrS representada por modelos de características que organizam macrofases, papéis e produtos de trabalho recorrentes no desenvolvimento de JS digitais. A viabilidade dessa linha pode ser verificada na medida em que é possível instanciar novos processos específicos representantes de métodos de desenvolvimento de jogos sérios digitais descritos na literatura.

Este capítulo descreve como a LPrS-JS pode ser instanciada para um processo documentado na literatura acadêmica. Aqui, o foco recai sobre o procedimento de instanciação, em quais evidências textuais são usadas para justificar a seleção de características e como o resultado é documentado. A partir desse procedimento, um processo descrito na literatura é tomado como exemplo e transformado em instância da LPrS-JS construída, de modo a permitir a comparação de abordagens, a identificação de padrões e a observação de possíveis lacunas da linha.

6.1 Procedimento geral de instanciação da LPrS

Conforme descrito no Capítulo anterior, a partir de um procedimento de criação de uma LPrS-JS, foram construídos modelos de características que organizam macrofases, papéis e produtos de trabalho em uma notação comum. A instanciação consiste em selecionar, a partir desse modelo, um subconjunto de características que represente um determinado processo-alvo descrito na literatura.

O procedimento seguido pode ser descrito em dois passos. O primeiro passo consiste em delimitar o escopo da instanciação, definindo qual processo será considerado como alvo e qual recorte desse processo interessa à LPrS, bem como a descrição textual do processo.

O segundo passo foca em papéis, artefatos, macrofases, subatividades e suas variantes. Ao ler analiticamente o documento, no nível de cada elemento, são pro-

curadas no texto menções a elementos descritos na linha. Essas funções são então associadas aos elementos canônicos definidos na LPrS-JS, com uma justificativa de seleção.

O resultado é uma instância da LPrS-JS que representa o processo descrito na fonte analisada. Essa instância serve tanto para fins de comparação entre processos quanto para avaliar a cobertura da linha frente a diferentes práticas de desenvolvimento de JS.

6.2 Instanciação

Esta seção apresenta a instanciação da LPrS-JS, a partir de um artigo científico publicado no Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames). O artigo trabalhado, de CLASSE *et al.* (2017), apresenta o jogo Desaparecidos RJ, um *text game adventure* construído a partir do processo de descoberta de paradeiros da Polícia Civil do Estado do Rio de Janeiro.

O processo documentado no artigo não foi escolhido a partir da seleção inicial de potenciais publicações propositoras, mas de forma *ad hoc*. Isso ocorreu em razão da necessidade de diversificar o escopo da instanciação por meio de um caso mais próximo ao real, com a construção de um jogo concreto, e que traga uma perspectiva externa e complementar à seleção inicial, focando no enriquecimento da validação do modelo com dados práticos não previstos na amostra anterior.

A derivação é conduzida conforme o procedimento de instanciação descrito na seção anterior, percorrendo macrofases, subatividades, variantes, papéis e artefatos canônicos da LPrS-JS. Para cada elemento, indica-se o *status* de ativação com base em evidências das documentações dos respectivos artigos. A análise visa verificar a cobertura da linha frente a contextos de desenvolvimento distintos, bem como identificar lacunas da LPrS-JS quando confrontada com práticas que escapam ao perfil dos processos originários.

6.2.1 Descrição do processo

O artigo apresenta o desenvolvimento do jogo Desaparecidos RJ (CLASSE *et al.*, 2017), um *text game adventure* construído a partir do processo de descoberta de paradeiros da Delegacia de Descoberta de Paradeiros (DDPA) da Polícia Civil do Estado do Rio de Janeiro. O objetivo é investigar a viabilidade de usar jogos digitais para transmitir aos cidadãos o funcionamento e os valores envolvidos no processo de prestação de serviços públicos, permitindo que eles vivenciem o serviço de forma lúdica e compreendam as etapas e responsabilidades de cada ator.

O recorte do jogo incide sobre o subprocesso de criação de Registro de Ocorrência,

no qual o jogador assume o papel de um policial ouvinte que deve coletar o máximo de informações sobre um caso de desaparecimento infantil, registrar a ocorrência no sistema ROWeb e enviá-la à DDPA. O jogo incorpora mecânicas de tempo, inventário e pontuação, além de uma mecânica de angústia da mãe comunicante que introduz fatores de insucesso. A plataforma adotada é o *Quest Text Adventure*.

O artigo também propõe, como contribuição metodológica, uma primeira versão de um *checklist* de *design* de jogos digitais baseados em processos de prestação de serviços públicos, listando as etapas seguidas durante o estudo de caso como procedimento replicável.

6.2.2 Fases e Condução do Processo no Artigo

O processo de desenvolvimento do jogo Desaparecidos RJ é organizado em seis etapas práticas de construção. Em nível mais alto, o artigo sintetiza o método utilizado em nove passos: selecionar o contexto do processo, levantar o processo, documentar e modelar o processo, validar o processo, pensar e projetar o jogo, desenvolver o protótipo, avaliar o protótipo, redefinir o protótipo e validar o jogo.

A seção de Levantamento e Modelagem do Processo descreve o contato com a DDPA por meio de um investigador, a coleta do modelo do processo em notação BPMN e o fornecimento de casos reais pela delegacia como subsídios narrativos. O Projeto do Jogo apresenta a proposta narrativa do caso de desaparecimento infantil, bem como é apresentada uma tabela que mapeia elementos do modelo de processo para elementos do gênero aventura, constituindo o principal artefato de *design* do jogo. O Desenvolvimento do Protótipo documenta a implementação da primeira versão no Quest Text Adventure, com mecânicas de perguntas e respostas, inventário e sistema de pontuação por tempo e ações corretas.

A fase da Avaliação do Protótipo narra a reunião com o conselho do projeto, que identificou o jogo como excessivamente determinístico, sem fatores de insucesso. A Revisão do Projeto do Protótipo diz respeito ao *redesign* com a introdução da mecânica de angústia da mãe, que aumenta com o tempo e, ao atingir o nível máximo, encerra o jogo sem solução. A Validação do Jogo com a Instituição Pública descreve a demonstração do protótipo revisado aos membros da DDPA, que confirmaram a representação fiel do processo. O artigo reconhece que a validação com cidadãos não foi realizada no âmbito deste estudo.

O processo descrito é marcado por uma orientação dupla, como estudo exploratório de viabilidade e como primeiro passo de um método de *design* baseado em modelos de processos. Essa característica impacta o nível de formalização dos artefatos produzidos, que são em grande parte narrativos ou em forma de tabela, sem estruturação em documentos formais de engenharia de software.

6.2.3 Derivação da LPrS-JS

A Tabela 6.1 apresenta a derivação dos elementos canônicos da LPrS-JS para o processo descrito no artigo, com indicação do *status* de ativação e justificativa baseada no texto da publicação.

Tabela 6.1: Elementos canônicos analisados na instância

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Macrofase	MF1 - Problema & Objetivos (Mandatória)	Ativo	Identificação do distanciamento entre cidadãos e processos de serviços públicos; motivação pela escolha do serviço de descoberta de paradeiros como processo pouco conhecido.	O problema cívico-educacional é formulado com base em dados empíricos e revisão de literatura, e os objetivos de conscientização do cidadão são definidos antes do *design*. Todos os elementos constitutivos desta macrofase estão presentes.
Macrofase	MF2 - *Design* do Jogo & Prototipagem (Mandatória)	Ativo	Seção 6.2.2 (Projeto do Jogo): seleção do caso narrativo, definição do papel do jogador (policial ouvinte), mapeamento de elementos de processo para elementos de jogo. Seções 6.2.3 e 6.2.5: desenvolvimento e revisão iterativa de protótipos.	O *design* é formalizado em um mapeamento processo-jogo e o protótipo é revisado após avaliação interna, configurando um ciclo de *design* e prototipagem iterativa explícito.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Macrofase	MF3 - Engenharia de Software (Opcional)	Não ativo	A plataforma Quest Text Adventure é descrita como escolha tecnológica, mas não há documentação de arquitetura, requisitos rastreáveis ou *design* de software formal como fase autônoma.	A macrofase é opcional na LPrS-JS. A ferramenta Quest impõe uma estrutura pré-definida de *text game adventure*, tornando o *design* técnico absorvido pela escolha de plataforma sem necessidade de fase de Engenharia de Software autônoma.
Macrofase	MF4 - Implementação & Publicação (Mandatória)	Ativo	Seção 6.2.3 e 6.2.5: duas versões do protótipo implementadas na plataforma Quest Text Adventure, com mecânica de tempo, inventário, pontuação e mecânica de angústia acrescentada na revisão.	Um *build* funcional foi produzido em duas iterações. A publicação ao público cidadão não ocorreu no escopo do artigo, mas a implementação do protótipo cumpre o critério de existência de executável integrado.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Macrofase	MF5 - Uso em Campo & Avaliação (Mandatória)	Ativo	Seção 6.2.4: avaliação interna com o conselho do projeto (identificou jogo determinístico). Seção 6.2.6: validação com membros da DDPA, confirmando representação fiel do processo. Avaliação com cidadãos prevista como trabalho futuro.	A fase de uso e avaliação ocorreu em dois momentos documentados: avaliação interna e validação institucional. Embora o estudo com cidadãos não tenha sido realizado, a validação com a instituição prestadora do serviço cumpre o escopo desta macrofase.
Macrofase	MF6 - *Feedback* & Sustentação (Opcional)	Ativo	Seção 6.2.6 e Seção 8 (Conclusões e Trabalhos Futuros): sugestões da DDPA (filas de espera, casos não configuráveis como ocorrência), planos de versão gráfica (RPG), avaliação com cidadãos e desenvolvimento de software de suporte ao mapeamento.	A macrofase é opcional na LPrS-JS. O artigo documenta sugestões institucionais de novos elementos e define encaminhamentos de melhoria, funcionando como um *backlog* de mudanças estruturado proveniente da validação com a DDPA.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Subatividade	1.1 Entender o Problema (Mandatória)	Ativo	Seção 4: distanciamento cidadão-instituição pública como problema; Seção 6.1: o serviço de descoberta de paradeiros como desconhecido da população. Dados do fundamentam.	O problema é identificado com base em dados empíricos e revisão bibliográfica em democracia digital e gestão de processos, cumprindo o requisito desta subatividade mandatória.
Subatividade	1.2 Identificar Contexto e Escopo (Mandatória)	Ativo	Seção 6.1: escopo delimitado ao subprocesso de criação de Registro de Ocorrência da DDPA do Estado do Rio de Janeiro. Público-alvo: cidadãos, com foco especial em populações vulneráveis.	O domínio de aplicação e o recorte do processo são definidos antes do *design*, atendendo ao critério desta subatividade mandatória.
Subatividade	1.3 Identificar *Stakeholders* (Opcional)	Ativo	Seção 6.2.1: contato formal com a DDPA por meio do investigador participante como *stakeholder*-chave. Conselho do projeto como partes internas. Cidadãos como público-alvo.	Esta subatividade é opcional na LPrS-JS. O artigo descreve engajamento formal com a instituição detentora do processo, o que constitui uma atividade estruturada de identificação e contato com *stakeholders*, justificando sua ativação.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Subatividade	1.4 Descrever Objetivos de Aprendizagem (Mandatória)	Ativo	Objetivos de conscientização e compreensão cívica: o cidadão deve entender o processo do Registro de Ocorrência, compreender a importância das informações fornecidas e desenvolver empatia com a instituição prestadora do serviço.	Os objetivos de aprendizagem (cívica e comportamental) são definidos no início do artigo e orientam todo o *design* do jogo, cumprindo o requisito desta subatividade mandatória.
Subatividade	1.5 Identificar Requisitos Iniciais (Mandatória)	Ativo	Seção 6.2.1: o processo real da DDPA e os casos reais fornecidos constituem os requisitos de fidelidade do jogo. A ordem das etapas do processo de criação de registro de ocorrência define as regras estruturais do jogo.	Os requisitos são extraídos diretamente do modelo do processo real e dos casos narrativos fornecidos pela instituição, configurando uma elicitação formal de necessidades a partir do processo de negócio.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Subatividade	2.1 Elaborar *Design* do Jogo e Elementos Pedagógicos (Mandatória)	Ativo	Seção 6.2.2: Tabela mapeia elementos do modelo de processo para elementos do jogo de aventura. Narrativa de desaparecimento infantil selecionada e descrita.	O *design* é formalizado em um mapeamento processo-jogo que traduz objetivos de domínio em estruturas lúdicas, cumprindo o escopo desta subatividade mandatória.
Subatividade	2.2 Desenvolver Protótipos (Mandatória)	Ativo	Seção 6.2.3: primeira versão do protótipo no Quest Text Adventure. Seção 6.2.5: revisão com inclusão da mecânica de angústia da mãe e reclamações aleatórias para criar fatores de insucesso.	Duas versões do protótipo foram produzidas e avaliadas, configurando um processo de prototipagem iterativa explícito e documentado.
Subatividade	2.3 Avaliar/Testar e Refinar Iterativamente (Opcional)	Ativo	Seção 6.2.4: avaliação interna identificou jogo determinístico. Seção 6.2.5: *redesign* com inserção de mecânica de angústia e reclamações como resposta direta à avaliação.	Esta subatividade é opcional na LPrS-JS. O ciclo de avaliação interna seguido de refinamento de protótipo (adição de fatores de insucesso) é documentado, justificando a ativação desta subatividade.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Variante	1.1 - Pesquisa e Estudo de Contexto	Ativo	Revisão bibliográfica em democracia digital, gestão de processos de negócio e jogos digitais. Dados sobre ocorrências são usados como evidência do problema.	O estado da arte é levantado, configurando a variante de pesquisa e estudo de contexto como abordagem central de entendimento do problema.
Variante	1.2 - Análise Demográfica e Educacional (Contexto Cívico)	Ativo	O contexto cívico do cidadão fluminense como público-alvo, com ênfase em populações vulneráveis (baixa renda, baixa instrução) mais afetadas pelo desconhecimento do serviço.	Embora o contexto seja cívico e não estritamente escolar, a variante de Análise Demográfica e Educacional é a mais próxima do perfil do público descrito. A variante de Domínio Terapêutico é mutuamente exclusiva e não se aplica.
Variante	1.3 - Mapeamento Formal de *Stakeholders*	Ativo	Contato formal com a DDPA. Conselho do projeto como *stakeholder* interno.	O engajamento formal com a instituição detentora do processo configura um mapeamento de partes interessadas com impacto direto no escopo e no *design* do jogo.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Variante	1.4 - Definição de Objetivos Pedagógicos e Competências	Ativo	Objetivos de conscientização sobre o processo do Registro de Ocorrência, importância das informações fornecidas e empatia entre cidadão e policial.	A abordagem se aproxima dos objetivos pedagógicos, ainda que no contexto de educação cívica. A variante terapêutica e a captação comportamental direta não se aplicam a este domínio de serviço público.
Variante	1.5 - Elicitação Formal de Necessidades	Ativo	O modelo do processo da DDPA e os casos reais fornecidos pela delegacia definem formalmente as regras e restrições que o jogo deve respeitar.	Os requisitos são extraídos de documentação institucional formal (modelo BPMN do processo e casos narrativos reais), configurando uma elicitação formal de necessidades.
Variante	2.1 - *Design* Consensual e Documental	Ativo	Tabela como *blueprint* de mapeamento processo-jogo, narrativa do caso de desaparecimento infantil como roteiro consensual, discutida e validada com o conselho do projeto.	A Tabela funciona como documento de *design* consensual que integra os elementos do processo e os elementos lúdicos, alinhando a equipe antes da implementação.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Variante	2.2 - Prototipagem de Componentes Funcionais	Ativo	Protótipo implementado diretamente na plataforma Quest Text Adventure com inventário, sistema de pontuação, mecanismo de tempo e mecânica de angústia.	A equipe optou por implementar os componentes funcionais diretamente na plataforma de produção, avaliando a viabilidade técnica das mecânicas por meio de execução real desde a primeira versão.
Variante	2.3 - Refinamento Iterativo Contínuo	Ativo	Seção 6.2.5: *redesign* com adição de mecânica de angústia e reclamações como resposta à avaliação interna que identificou o jogo como determinístico.	O ciclo de avaliação e refinamento é documentado: a avaliação (Seção 6.2.4) diretamente motivou mudanças de *design* (Seção 6.2.5), configurando refinamento iterativo contínuo.
Papel	Pesquisador / Analista	Ativo	Equipe descrita no artigo: revisão de literatura em democracia digital e gestão de processos, análise de trabalhos relacionados.	A revisão bibliográfica e o estudo de contexto são realizados pela equipe acadêmica, configurando o papel de pesquisador no processo.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Papel	Especialista de Domínio	Ativo	Investigador e membros da DDPA: fornecimento do modelo do processo, casos narrativos reais e validação da representação do serviço no jogo.	Os membros da DDPA atuam como especialistas do domínio de serviço público, fornecendo o conhecimento técnico-institucional que orienta o *design* do jogo.
Papel	Desenvolvedor / Programador	Ativo	Equipe de pesquisa: implementação do protótipo em Quest Text Adventure com lógica narrativa, inventário e sistema de *feedback*.	A implementação do protótipo é realizada pela equipe, que acumula os papéis de *design* e desenvolvimento, como é comum em projetos acadêmicos de pequena escala.
Papel	Game Designer	Ativo	Equipe de pesquisa: mapeamento processo-jogo, definição do papel do jogador, seleção do caso narrativo, *design* de mecânicas e sistema de pontuação.	O *design* da experiência lúdica e dos mecanismos de jogo é documentado na Seção 6.2.2, configurando este papel.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Papel	Decisor / Patrocinador	Ativo (parcial)	A DDPA atua como detentora do processo e validadora do produto, cumprindo parcialmente a função de decisor institucional.	O papel é opcional na LPrS-JS. A DDPA exerce função de decisor ao validar a fidelidade do processo retratado, mas não há menção a um patrocinador formal com responsabilidade de escopo e financiamento.
Papel	Usuários Finais/*Playtesters*	Ativo (parcial)	Conselho do projeto (avaliação interna, Seção 6.2.4) e membros da DDPA (validação institucional, Seção 6.2.6). Cidadãos como público-alvo final, mas não avaliaram o jogo no âmbito do artigo.	O papel é parcialmente cumprido pela avaliação interna e institucional. A ausência de testes com cidadãos é reconhecida pelos autores como etapa futura.
Papel	Facilitador / Moderador	Não ativo	Não há descrição de um papel de facilitador em dinâmicas de co-design ou sessões estruturadas.	O papel é opcional na LPrS-JS. O processo não utiliza dinâmicas participativas que demandem facilitação formal.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Papel	Arquiteto/*Designer* de Software	Não ativo	Não há documentação de arquitetura de software; a estrutura do jogo é determinada pela plataforma Quest Text Adventure.	O papel é opcional na LPrS-JS. A plataforma escolhida impõe a arquitetura do *text game adventure*, dispensando um arquiteto autônomo para o escopo do protótipo.
Papel	*Release* / Ops	Não ativo	Não há atividade de empacotamento, distribuição ou operação do jogo descrita no artigo.	O papel é opcional na LPrS-JS. O protótipo permanece em ambiente de desenvolvimento e validação, sem distribuição ao público no recorte do artigo.
Papel	Manutenção / Suporte / PM	Não ativo	Não há descrição de atividades de gerenciamento de projeto ou manutenção pós-validação.	O papel é opcional na LPrS-JS. O escopo do artigo encerra-se na validação institucional.
Artefato	Especificação do Problema / Necessidades / Objetivos	Ativo	Seção 4: problema de distanciamento cidadão-instituição. Seção 6.1: motivação pela escolha do serviço e objetivos de conscientização.	O problema e os objetivos são documentados de forma narrativa no artigo, servindo como especificação de necessidades do jogo.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Artefato	GDD / Especificação de *Design* de Jogo	Ativo (parcial)	Tabela de mapeamento processo-jogo e descrição do caso narrativo cumprem parcialmente a função do GDD. Não há um documento de *design* do jogo nomeado como tal.	A Tabela constitui um artefato de *design* estruturado que mapeia elementos canônicos do processo para elementos do jogo, funcionando como a especificação de design mais próxima de um GDD.
Artefato	Protótipos / Artefatos de *Gameplay*	Ativo	Duas versões do protótipo em Quest Text Adventure: versão inicial e versão revisada com mecânica de angústia.	Os protótipos são produzidos, avaliados e documentados no artigo com capturas de tela ilustrativas das telas de coleta, inventário e encerramento do jogo.
Artefato	Requisitos Detalhados	Ativo (parcial)	O modelo do processo e os casos reais fornecidos pela DDPA funcionam como especificação de requisitos de fidelidade do jogo ao processo real.	Os requisitos são derivados do modelo de processo e da documentação institucional, mas não estão formalizados em um artefato de requisitos de software convencional.
Artefato	*Builds* / Componentes Executáveis	Ativo	Duas versões do protótipo implementadas e executadas na plataforma Quest Text Adventure.	Os *builds* existem e foram usados nas avaliações interna e institucional.

Continua na próxima página

Tabela 6.1 – continuação da página anterior

Tipo	**Elemento Canônico**	**Status**	**Correspondência no Artigo**	**Justificativa**
Artefato	Resultados de Avaliação	Ativo	Seção 6.2.4 (avaliação interna) e Seção 6.2.6 (validação com DDPA): conclusões sobre fidelidade do processo e sugestões de novos elementos.	Os resultados das duas avaliações são documentados e usados para orientar a revisão do protótipo e os trabalhos futuros.
Artefato	*Backlog* de Mudanças / *Feedback* Estruturado	Ativo	Sugestões da DDPA (filas de espera, casos não configuráveis como ocorrência, nova versão gráfica) registradas como encaminhamentos de melhoria.	O conjunto de sugestões institucionais e de pesquisa documentado nas conclusões funciona como um *backlog* estruturado para a próxima iteração.
Artefato	Lista de Participantes	Não ativo	Os participantes das avaliações são mencionados nominalmente ou de forma agregada, sem um documento formal de partes interessadas.	Artefato não produzido formalmente. Sua ausência não constitui lacuna dado o escopo reduzido do projeto.
Artefato	Arquitetura & *Design* de Software	Não ativo	Não há documentação de arquitetura de software; a estrutura é absorvida pela plataforma Quest Text Adventure.	A plataforma impõe uma arquitetura predefinida, tornando desnecessária a elaboração de um documento de arquitetura autônomo.

6.2.4 Discussão: Pontos Fortes e Pontos de Atenção da LPrS-JS Frente ao Processo

A derivação da LPrS-JS para o artigo analisado permite observar um conjunto de padrões que evidenciam tanto a robustez da linha quanto seus limites de cobertura frente a contextos de desenvolvimento distintos.

Entre os pontos fortes da LPrS-JS frente a esta derivação, destaca-se que a linha foi capaz de cobrir todas as seis macrofases canônicas, com as quatro mandatórias plenamente ativadas e as duas opcionais (MF3 e MF6) adequadamente tratadas. O ciclo iterativo de *design* e refinamento encontrou correspondência na subatividade 2.3 Avaliar/Testar e Refinar Iterativamente, demonstrando que a linha modela com precisão o padrão de avaliação interna seguida de *redesign*. A subatividade 1.3 Identificar *Stakeholders*, opcional na LPrS-JS, foi ativada neste caso em razão do engajamento formal com a DDPA, o que confirma a pertinência de sua opcionalidade: em jogos baseados em domínios institucionais, este elemento é mais relevante do que em projetos educacionais de escopo fechado.

O ponto de variação 1.5, com a variante de Elicitação Formal de Necessidades, encontrou um correspondente específico neste processo: o modelo BPMN do processo real como fonte de requisitos do jogo, uma situação incomum nos processos fundadores da LPrS-JS e não prevista explicitamente nas variantes da linha. Essa característica, por ser pontual, poderia provocar uma adaptação especificamente nesse ponto na instância, de modo a suportar esse tipo de insumo de *design*.

A Tabela presente no artigo, que mapeia elementos do processo para elementos do jogo de aventura, também não possui correspondente canônico direto na LPrS-JS além do GDD genérico. Essa especificidade metodológica do *design* baseado em processos não é capturada pela linha em seu estado atual, o que pode representar uma lacuna frente a projetos desta natureza, o que poderia ser coberto caso fossem derivadas variantes para o artefato de descrição do jogo.

O ponto de atenção mais relevante da LPrS-JS nesta derivação foi a ausência de validação com cidadãos no âmbito do artigo, que deixa a dimensão obrigatória de Uso em Campo (MF5) parcialmente verificada, evidenciando uma limitação do estudo que, no plano da derivação, se traduz em uma ativação parcial dos artefatos de resultados de avaliação.

Cabe registrar uma ressalva quanto ao alcance desta prova de conceito. Os pontos de atenção identificados nesta derivação (a ausência de uma variante para insumos externos de *design*, como o modelo BPMN utilizado como fonte de requisitos, e a inexistência de um artefato canônico específico para o mapeamento processo-jogo) constituem candidatos à evolução da linha, mas não foram incorporados ao modelo de características no âmbito deste trabalho. A realimentação do domínio a partir

da instanciação, não foi exercida aqui; os modelos de características, portanto, não foram evoluídos após a derivação. Uma única instanciação revela pontos candidatos à evolução, e não necessidades de variabilidade consolidadas: propagar ao domínio todo desvio observado em uma instância isolada poderia causar o superajustar da linha a um caso particular, comprometendo sua generalidade. A evolução do modelo de características em nível de domínio deve ser conduzida quando uma mesma necessidade de variação se mostra recorrente ao longo de múltiplas instâncias e processos, evidência que só se acumula no longo prazo. Por essa razão, os achados desta derivação são tratados como insumos para futuras iterações da linha, e não como motivadores de alterações imediatas.

Capítulo 7

Avaliação da LPrS-JS

Este capítulo descreve o processo de avaliação da LPrS-JS proposta neste trabalho. O objetivo da avaliação foi verificar se os artefatos gerados são compreensíveis, úteis e viáveis na perspectiva de especialistas, complementando a verificação por instanciação realizada no Capítulo 6. Enquanto a verificação demonstrou a capacidade de derivação da linha, a avaliação busca evidências sobre sua qualidade percebida e sua aplicabilidade prática.

A avaliação está estruturada em torno das três questões de pesquisa definidas no Capítulo 1: QP1, sobre clareza e completude; QP2, sobre utilidade prática e produtividade percebida; e QP3, sobre viabilidade considerando o esforço de configuração e derivação. As subseções seguintes descrevem o método de avaliação adotado, os materiais disponibilizados, o instrumento utilizado, o procedimento de coleta e a análise dos resultados.

7.1 Método de Avaliação

A avaliação do artefato é uma etapa necessária em pesquisas que contemplam a construção de artefatos para demonstrar sua utilidade, qualidade e eficácia (HEVNER *et al.*, 2004). Para cumprir esse objetivo, a estratégia de avaliação adotada neste trabalho é a avaliação por especialistas, uma modalidade que permite verificar a aplicabilidade do artefato proposto. Nessa modalidade, profissionais com experiência no(s) domínio(s) de interesse interagem com o artefato e fornecem julgamentos sobre suas características, identificando pontos fortes, limitações e oportunidades de melhoria.

Essa escolha justifica-se pela natureza do artefato construído. A LPrS-JS é um instrumento de apoio à tomada de decisão em processos de desenvolvimento de jogos sérios digitais, e sua qualidade não pode ser avaliada apenas por critérios formais de corretude estrutural, dependendo do reconhecimento da utilidade por parte de quem atuaria como seu usuário. A avaliação por especialistas permite analisar essa

dimensão por meio da captura de percepções quantitativas e qualitativas sobre o artefato.

Cabe delimitar, desde já, o alcance pretendido por esta avaliação. Trata-se de um estudo exploratório de percepção, conduzido com um painel reduzido de especialistas, cujo propósito é coletar julgamentos qualificados sobre a clareza, a utilidade e a viabilidade dos artefatos, e não estimar parâmetros de uma população ou testar hipóteses. Os dados quantitativos coletados têm, portanto, função descritiva, resumindo o posicionamento do painel e orientando a leitura das respostas discursivas, que constituem o principal substrato interpretativo da avaliação. Essa delimitação é coerente com a avaliação por especialistas prevista na estratégia adotada, voltada a evidências de utilidade e qualidade percebidas.

7.2 Perfil dos Participantes

Os participantes da avaliação foram selecionados a partir de critérios de perfil que buscaram garantir diversidade de perspectivas sobre os domínios avaliados. Foram convidados profissionais e pesquisadores que atendessem a ao menos um dos seguintes critérios:

1. experiência em desenvolvimento de jogos digitais ou jogos sérios, em qualquer papel (técnico, pedagógico, terapêutico, pesquisador ou de gestão);

2. atuação em pesquisa ou ensino na área de Engenharia de Software, com ênfase em processos de software e/ou reutilização; ou

3. familiaridade com metodologias de desenvolvimento de jogos sérios, adquirida por meio de pesquisa acadêmica ou prática profissional.

O recrutamento foi realizado por convite direto, a partir de contatos acadêmicos e profissionais, não sendo adotado nenhum critério de exclusão relacionado à área de formação ou ao nível de familiaridade com a metodologia *OdysseyProcessReuse*. Dessa maneira, busca-se observar se a linha proposta é compreensível e útil também para profissionais sem experiência prévia com linhas de processos, e a inclusão de participantes com diferentes graus de familiaridade com o referencial teórico permite avaliar essa dimensão em um espectro maior.

7.3 Materiais Disponibilizados - Catálogo

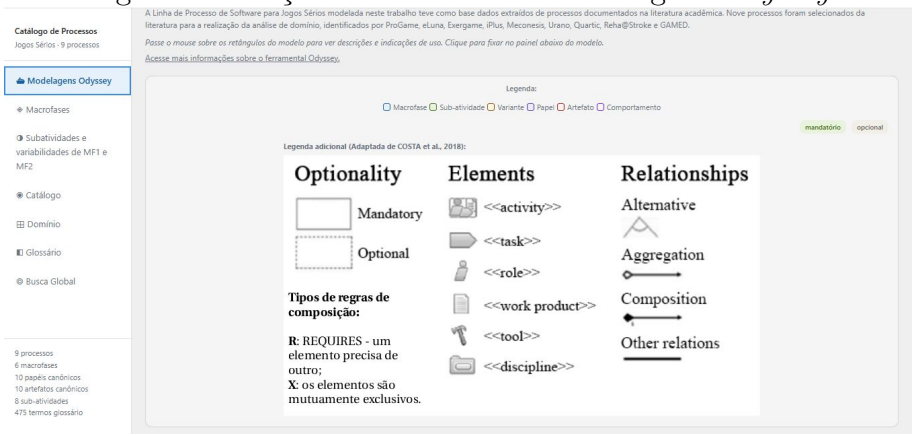
Para basear as respostas dos participantes em uma interação concreta com os artefatos produzidos de forma facilitada, foi construída e disponibilizada uma página *web*

contendo um Catálogo de Processos para jogos sérios. Nele, encontram-se disponíveis as modelagens de características, produto da construção da LPrS-JS relatada no Capítulo 5, além dos artefatos produzidos durante a sua construção. As subseções seguintes apresentam cada visão presente no Catálogo[1].

7.3.1 Visão Modelagens *Odyssey*

Nesta primeira visão, são apresentadas as modelagens de características que representam a LPrS-JS. A seção inicial da visão contém uma legenda dos itens que podem ser utilizados de acordo com a notação *OdysseyProcess-FEX* (TEIXEIRA, 2011), além de uma introdução contextualizando o usuário, conforme apresentado na Figura 7.1.

Figura 7.1: Seção inicial da visão Modelagens *Odyssey*.



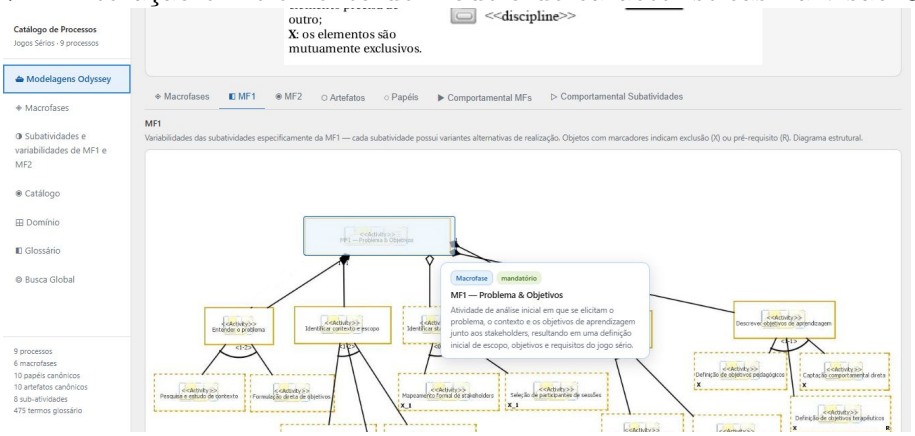
Fonte: Elaborado pelo autor (2026).

Cada modelagem foi disponibilizada em uma aba, contendo as modelagens de características estrutural das macrofases, das subatividades da macrofase 1 e da macrofase 2, dos artefatos e dos papéis. Também estão disponíveis os modelos de características comportamentais relativos às macrofases 1 e 2 (utilizando suas subatividades), além de um modelo de mais alto nível, com o fluxo comportamental incluindo todas as macrofases.

Cada modelo disponibilizado teve seus elementos mapeados, possibilitando, ao passar o cursor, observar a descrição do item, conforme apresentado na Figura 7.2.

[1] https://teste-blond-ten.vercel.app/catalogo

Figura 7.2: Interação em elemento de modelo de características na visão *Odyssey*.

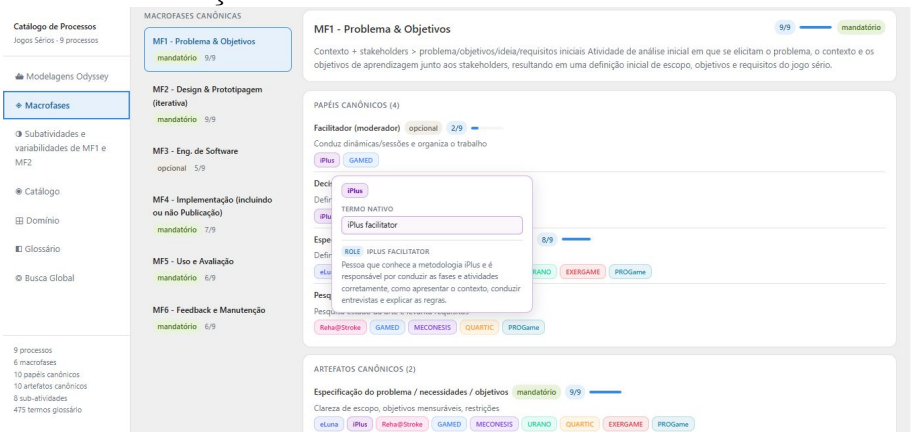


Fonte: Elaborado pelo autor (2026).

7.3.2 Visão Macrofases

A visão Macrofases apresenta, por macrofase da linha, sua descrição, papéis, artefatos e subatividades canônicos que participam da fase. Em cada item, há uma descrição do item canônico, bem como se o item é mandatório ou opcional, em quantos dos 9 processos analisados ele está presente, e o nome de cada processo que possui tal elemento. Ao passar o cursor no nome do processo, é possível ver o elemento nativo que deu origem àquele item canônico, conforme apresentado na Figura 7.3.

Figura 7.3: Visão Macrofases, com o cursor do usuário sob o rótulo do processo iPlus, exibindo a descrição do item nativo.



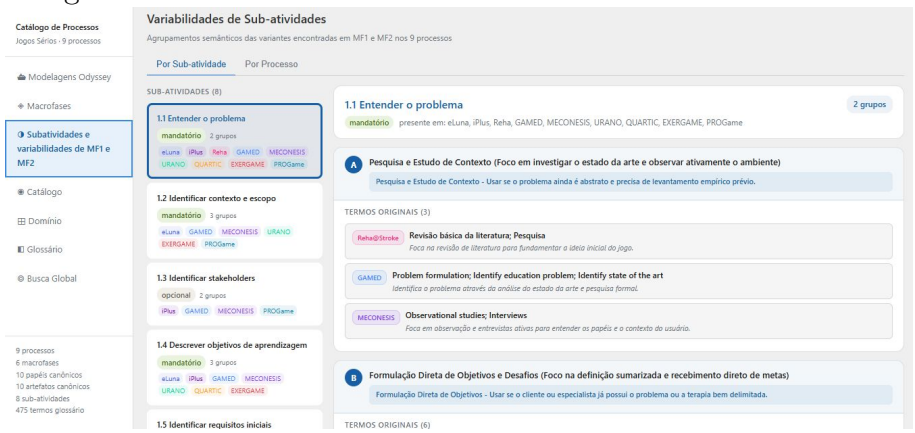
Fonte: Elaborado pelo autor (2026).

7.3.3 Visão Subatividades e Variabilidades de MF1 e MF2

Na visão Subatividades e Variabilidades de MF1 e MF2, são apresentadas duas abas. Na aba "Por Subatividade", conforme a Figura 7.4, é apresentada cada subatividade

relacionada às macrofases 1 e 2, bem como se ela é mandatória ou opcional, além das variabilidades associadas a elas, com os termos que deram origem à variabilidade em questão expressos.

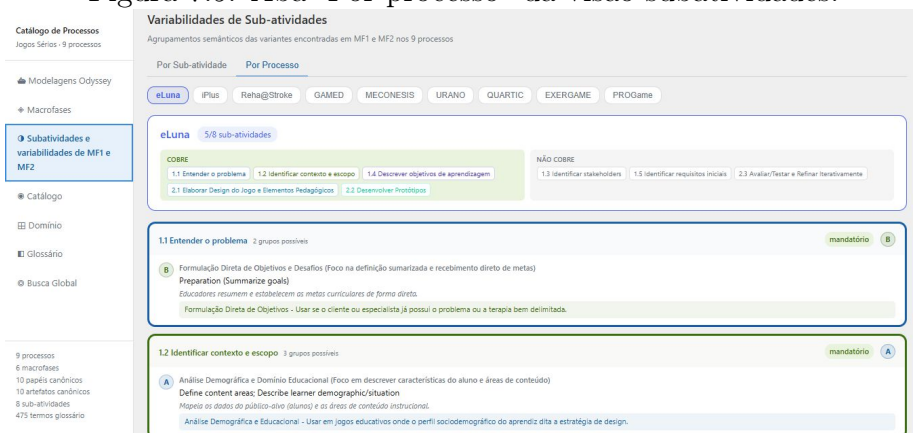
Figura 7.4: Aba “Por Subatividade” da visão de subatividades.



Fonte: Elaborado pelo autor (2026).

Na aba “Por Processo”, conforme aprentado na Figura 7.5, são exibidas quais subatividades são ativadas por processo de origem analisado e suas variabilidades relacionadas.

Figura 7.5: Aba “Por processo” da visão subatividades.

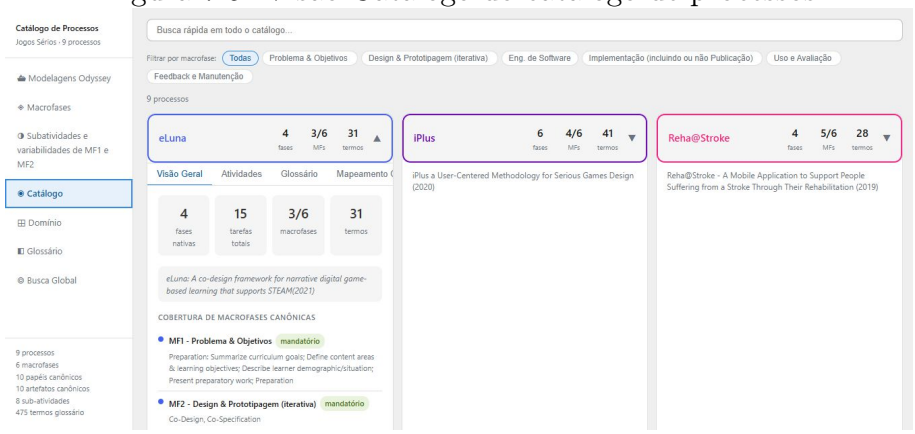


Fonte: Elaborado pelo autor (2026).

7.3.4 Visão Catálogo

Nessa visão, são exibidas informações gerais de cada processo origem analisado, bem como cobertura de macrofases, a publicação que originou o processo, os termos nativos extraídos dos processos por atividade, um glossário com a descrição de cada termo extraído do processo original e o mapeamento canônico de cada item (se ele está presente nesse processo). A visão é apresentada na Figura 7.6.

Figura 7.6: Visão Catálogo do catálogo de processos.

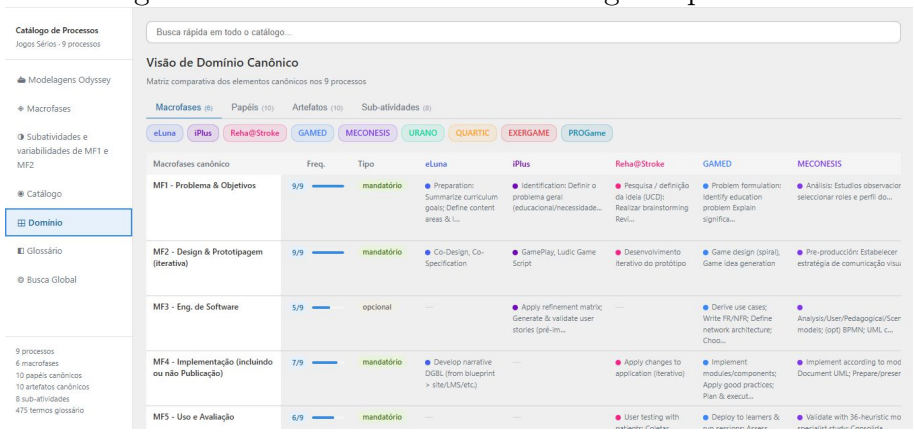


Fonte: Elaborado pelo autor (2026).

7.3.5 Visão Domínio

Nesta visão, conforme a Figura 7.7, são apresentados os elementos canônicos das dimensões de macrofases, papéis, artefatos e subatividades, mapeando em uma tabela os itens que contribuíram para a criação de cada item.

Figura 7.7: Visão Domínio do catálogo de processos.

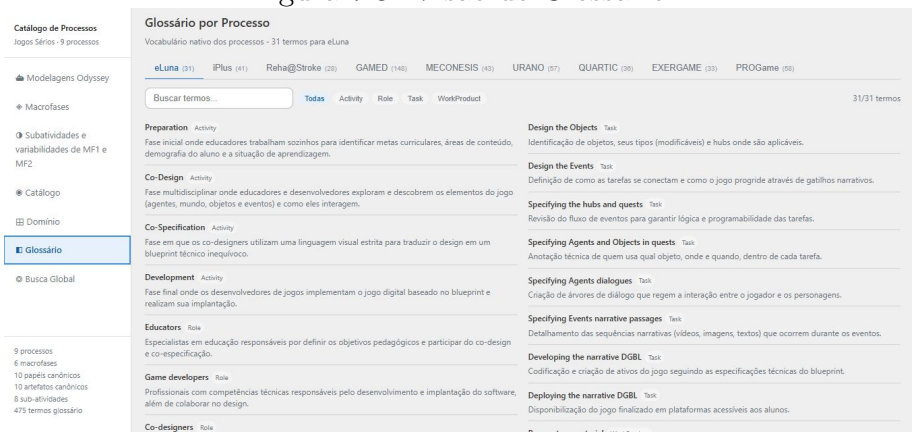


Fonte: Elaborado pelo autor (2026).

7.3.6 Visão Glossário

Na visão glossário, Figura 7.8, são apresentadas as descrições dos termos nativos extraídos de cada processo originário. Também é possível realizar buscas textuais dos termos no campo presente na visão.

Figura 7.8: Visão do Glossário.



Fonte: Elaborado pelo autor (2026).

7.3.7 Visão Busca Global

Nessa visão, Figura 7.9, é disponibilizado um campo de busca textual que possibilita a busca de termos por todo o catálogo.

Figura 7.9: Visão de Busca Geral.

Fonte: Elaborado pelo autor (2026).

7.4 Instrumento de Avaliação e Avaliação Piloto

O instrumento de avaliação foi desenvolvido na forma de questionário estruturado, disponibilizado por meio da plataforma *Google Forms*. Sua estrutura organiza-se em cinco seções, sendo a primeira dedicada à caracterização do respondente e as demais diretamente vinculadas às questões de pesquisa.

A **Seção 1 - Caracterização do Respondente** coleta informações sobre o perfil dos participantes, como nível e área de formação, experiência com desenvolvimento de software e jogos sérios, e grau de familiaridade com linhas de produtos e

processos. Essas informações permitem contextualizar as respostas e verificar se há variações de percepção associadas ao perfil dos participantes.

A **Seção 2 - Clareza e Completude** é composta por itens em escala *Likert* (LIKERT, 1932) de cinco pontos e duas questões abertas, referindo-se a QP1. Os itens avaliam a clareza das macrofases, dos papéis, dos artefatos e das subatividades canônicas, a compreensão do modelo de características e a capacidade da linha de estabelecer rastreabilidade entre objetivos do domínio e elementos técnicos do processo. As questões abertas investigam possíveis ausências ou inadequações nos elementos propostos. Nesta seção, também foi fornecido o link para o catálogo e um vídeo explicando a sua utilização.

A **Seção 3 - Utilidade Prática e Produtividade Percebida** contém itens *Likert* (LIKERT, 1932) e duas questões abertas, respondendo à QP2. Os itens avaliam se a linha seria útil como ponto de partida para novos projetos, se reduziria o esforço de definição de processos, se apoiaria a comunicação multidisciplinar de profissionais e se os exemplos de instanciação demonstram sua aplicabilidade de forma convincente. As questões abertas buscam compreender situações de uso e limitações percebidas.

A **Seção 4 - Viabilidade de Configuração e Derivação** apresenta cinco itens *Likert* (LIKERT, 1932) e duas questões abertas, voltados à QP3. Os itens investigam a compreensibilidade das regras de derivação, a autonomia do usuário para derivar processos sem apoio dos criadores da linha, a suficiência das variantes disponíveis e a extensibilidade da linha para incorporação de novos processos. As questões abertas exploram obstáculos à adoção e recursos de apoio desejáveis.

A **Seção 5 - Avaliação Global** observa uma avaliação geral da qualidade da LPrS-JS por meio de um item em escala de cinco pontos, complementado por três questões abertas sobre principais pontos fortes, pontos a melhorar e sugestões adicionais.

A escala *Likert* (LIKERT, 1932) adotada varia de 1, discordo totalmente, a 5, concordo totalmente, com ponto neutro em 3. As questões *Likert* (LIKERT, 1932) foram configuradas como obrigatórias; as questões abertas, como opcionais. O instrumento completo encontra-se disponível no Apêndice A.

7.5 Procedimento de Coleta de Dados e Avaliação Piloto

A coleta de dados foi realizada de forma assíncrona e remota. Os participantes selecionados receberam uma mensagem de convite contendo uma descrição da pesquisa e o *link* de acesso ao instrumento, que continha a proposta de aceite com relação a

participação da pesquisa e os demais detalhes para seu preenchimento.

7.5.1 Piloto

Foi realizada uma avaliação piloto com um participante, de modo remoto, antes do envio do convite aos demais especialistas. O respondente é pós-doutor com formação principal em Engenharia de Software, possui entre cinco e dez anos de experiência em desenvolvimento de software e em jogos digitais, e atuou como membro técnico (programador e *designer*) em projetos de jogos sérios, tendo participado de quatro a seis projetos do gênero; declara, ainda, experiência consolidada com Linhas de Produtos e Linhas de Processos de Software. Esse perfil, de especialista técnico que reúne domínio tanto do desenvolvimento de jogos sérios quanto do referencial de linhas de processos, é adequado à condução do piloto, por cobrir as duas frentes de conhecimento que o instrumento busca avaliar. O respondente preencheu, em cerca de uma hora, a avaliação de forma assistida, por compartilhamento de tela e de áudio. Como os ajustes decorrentes do piloto recaíram sobre os materiais de apoio e as instruções, e não sobre a substância dos itens avaliados, seus dados foram incorporados ao conjunto de resultados, correspondendo ao respondente identificado como R1 na Seção 7.6.

7.5.1.1 Ocorrências e Ações Tomadas

Durante a aplicação piloto do formulário, foram identificadas 5 ocorrências, das quais 4 resultaram em ajustes e 1 foi descartada:

1. **Dúvida sobre linha de processo vs. linha de produto (Identificação)**: O participante questionou se era necessário conhecer o conceito de linha de processo para responder. A pergunta foi considerada clara no formulário e nenhuma alteração foi realizada;

2. **Dificuldade em localizar as regras de composição na legenda**: O participante sugeriu inserir as regras diretamente na imagem de legenda da visão *Odyssey* do catálogo. A sugestão foi acatada e implementada;

3. **Dúvida sobre expansão das macrofases em subatividades**: O participante questionou se as demais macrofases haviam sido expandidas. Foram adicionadas ênfases nas descrições de cada modelagem nos artefatos, indicando explicitamente quais macrofases foram expandidas.

4. **Dúvida sobre escopo dos artefatos (MFs 1 e 2)**: Mesma natureza da ocorrência anterior. Resolvida em conjunto com o item 3, reforçando as descrições das modelagens.

5. **Rótulos dos nomes de processos pouco compreensíveis**: O participante não identificou de imediato o que representavam os nomes dos processos no catálogo. Foi adicionado um parágrafo explicativo no catálogo e no formulário, definindo os nomes dos processos de software que originaram a linha e enfatizando sua base acadêmica.

6. **Necessidade de baixar o vídeo de instrução**: O participante precisou fazer download do vídeo. O arquivo foi publicado no YouTube e o link foi atualizado no formulário.

7.6 Resultados

Os dados coletados foram analisados por meio de duas abordagens, correspondentes aos tipos de questões presentes no instrumento.

Para os itens quantitativos em escala *Likert*, foram calculadas medidas de tendência central (média e mediana) e de dispersão (desvio padrão) para cada item individualmente e para cada seção como um conjunto. A mediana foi adotada como medida principal de síntese, por ser mais robusta em escalas ordinais com número reduzido de respondentes (FIELD, 2009). Os resultados são apresentados por meio de tabelas de frequência e gráficos de distribuição, permitindo identificar padrões de concordância e discordância para cada dimensão avaliada.

As questões abertas foram analisadas individualmente, observando sempre o perfil do usuário respondente. As respostas foram lidas e analisadas.

Os resultados da análise são apresentados nas subseções seguintes, de modo a manter a rastreabilidade entre os dados coletados e os objetivos avaliativos que os motivaram.

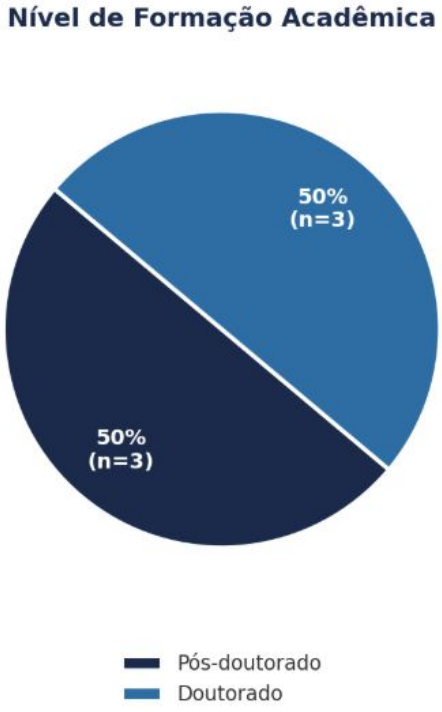
7.6.1 Perfil dos Participantes

Os seis especialistas foram recrutados conforme os critérios definidos na Seção 7.2, considerando experiência em desenvolvimento de jogos digitais ou jogos sérios, atuação em pesquisa ou ensino em Engenharia de Software, ou familiaridade com metodologias de desenvolvimento de JS. A Tabela 7.1 e as Figuras 7.10 a 7.14 sintetizam o perfil do painel de especialistas.

Tabela 7.1: Perfil dos participantes da avaliação.

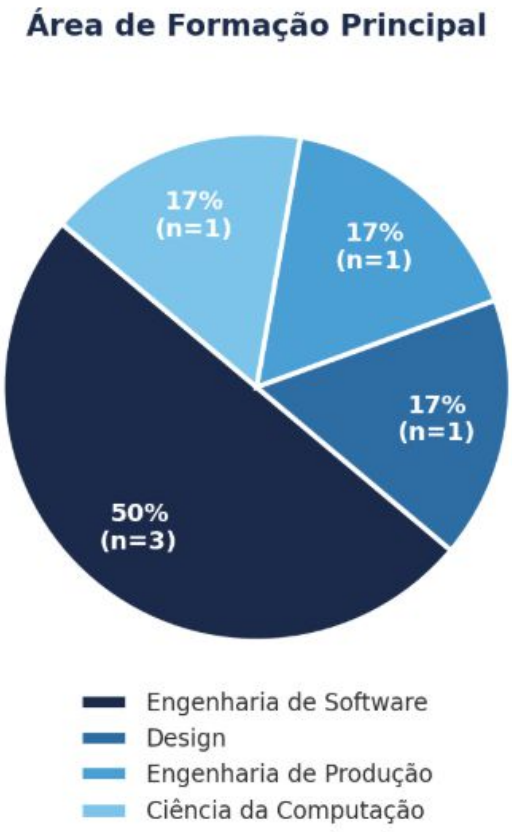
Resp.	**Formação**	**Área**	**Exp. SW**	**Exp. Jogos**	**Papel em JS**	**Fam. LPS/LPrS**
R1	Pós-doutorado	Eng. Software	5–10 anos	5–10 anos	Técnico (prog./designer)	Consolidada
R2	Doutorado	Design	>10 anos	>10 anos	Gestão / Pesquisa	Consolidada
R3	Doutorado	Eng. Produção	<2 anos	>10 anos	Técnico + Espec. de Dom. + Gestão	Similares
R4	Pós-doutorado	Ciência Comp.	<2 anos	<2 anos	Não participou	Consolidada
R5	Pós-doutorado	Eng. Software	>10 anos	>10 anos	Gestão / Pesquisa	Consolidada
R6	Doutorado	Eng. Software	>10 anos	>10 anos	Gestão / Pesquisa	Teórica

Figura 7.10: Nível de formação acadêmica dos participantes.



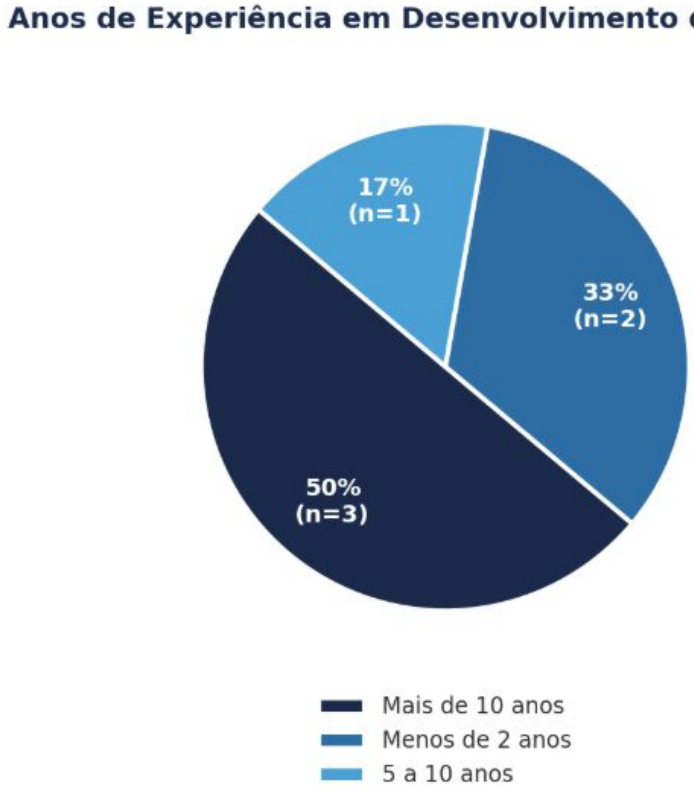
Fonte: Elaborado pelo autor (2026).

Figura 7.11: Área de formação principal.



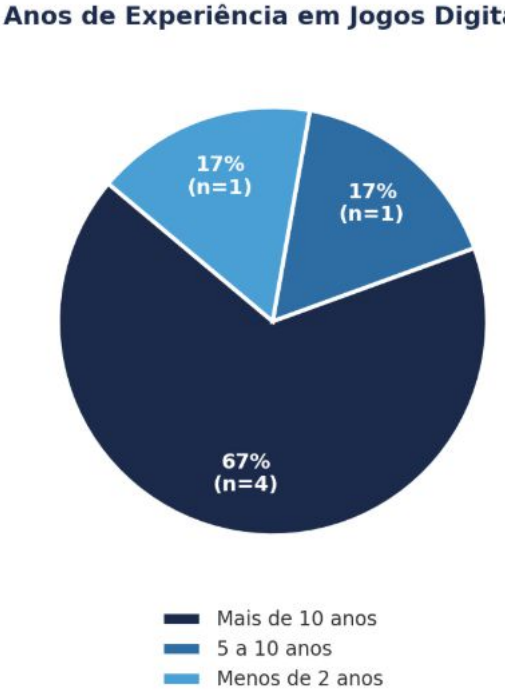
Fonte: Elaborado pelo autor (2026).

Figura 7.12: Anos de experiência em desenvolvimento de software.



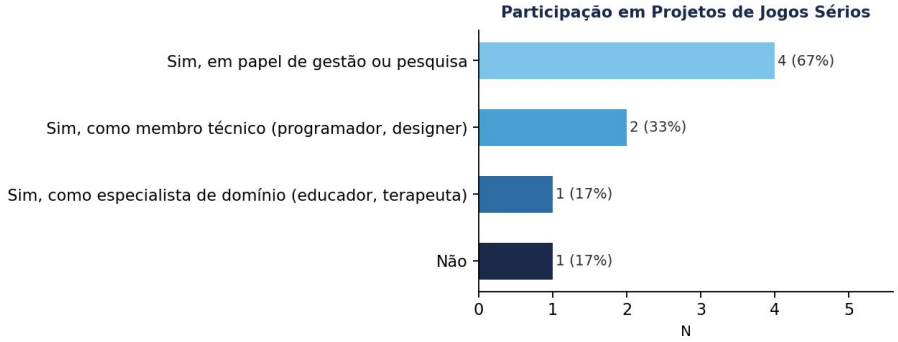
Fonte: Elaborado pelo autor (2026).

Figura 7.13: Anos de experiência em jogos digitais.



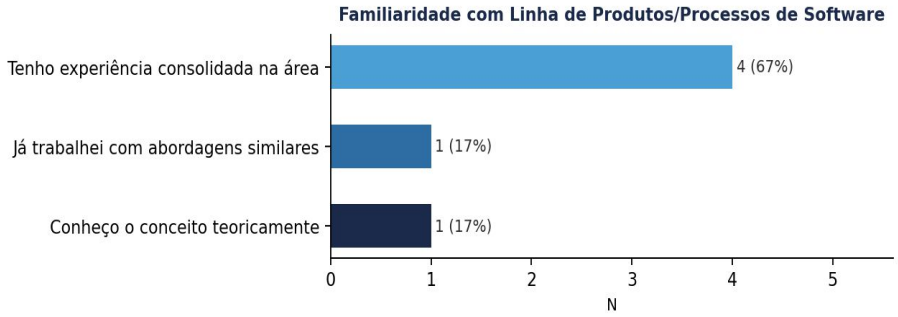
Fonte: Elaborado pelo autor (2026).

Figura 7.14: Participação em projetos de jogos sérios (multi-seleção).



Fonte: Elaborado pelo autor (2026).

Figura 7.15: Grau de familiaridade com LPS ou LPrS.



Fonte: Elaborado pelo autor (2026).

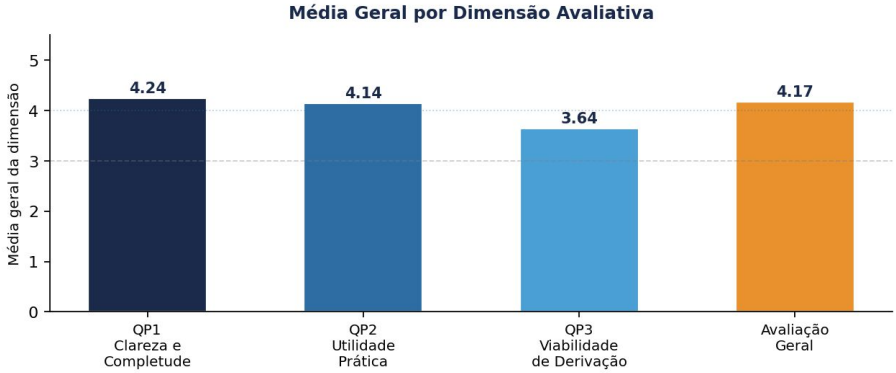
Todos os seis participantes possuem doutorado, e metade deles (três) conta também com pós-doutorado. A formação distribui-se entre Engenharia de Software (3), Design (1), Engenharia de Produção (1) e Ciência da Computação (1). Cinco dos seis possuem mais de cinco anos de experiência com jogos digitais, e quatro já utilizaram algum modelo de processo de desenvolvimento de jogos sérios. O grau de familiaridade com LPS/LPrS varia de conhecimento teórico (R6) a experiência consolidada (R1, R2, R4, R5), criando a experiência necessária para observar se a LPrS-JS é acessível a diferentes perfis. O único respondente sem experiência em jogos sérios (R4) é aquele com familiaridade consolidada em LPS/LPrS-JS, posicionamento que permite isolar a contribuição de cada tipo de *expertise* para as avaliações.

7.6.2 Achados Quantitativos

Esta seção apresenta e discute os resultados quantitativos organizados pelas três questões de pesquisa. Para cada dimensão, uma figura com a distribuição das respostas *Likert* permite visualizar o posicionamento dos respondentes. Uma segunda

figura mostra as médias por item, representando o desvio-padrão. A Figura 7.16 apresenta uma visão das médias por dimensão de avaliação.

Figura 7.16: Média geral por dimensão avaliativa.



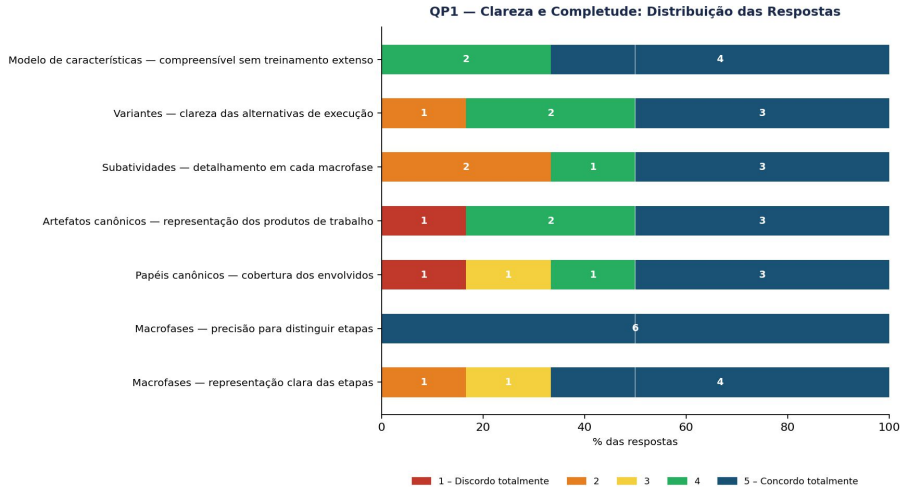
Fonte: Elaborado pelo autor (2026).

A Figura 7.16 resume, de forma descritiva, o posicionamento do painel por dimensão. As respostas concentram-se na faixa superior da escala (notas 4 e 5) nas dimensões de clareza e completude (QP1), utilidade prática (QP2) e na avaliação geral. A viabilidade de configuração e derivação (QP3) reúne as respostas mais cautelosas e dispersas, ainda assim majoritariamente situadas acima do ponto neutro (nota 3). As médias por dimensão exibidas na figura têm caráter apenas descritivo e não devem ser lidas como estimativas; este panorama orienta a leitura item a item das seções seguintes.

QP1 - Clareza e Completude

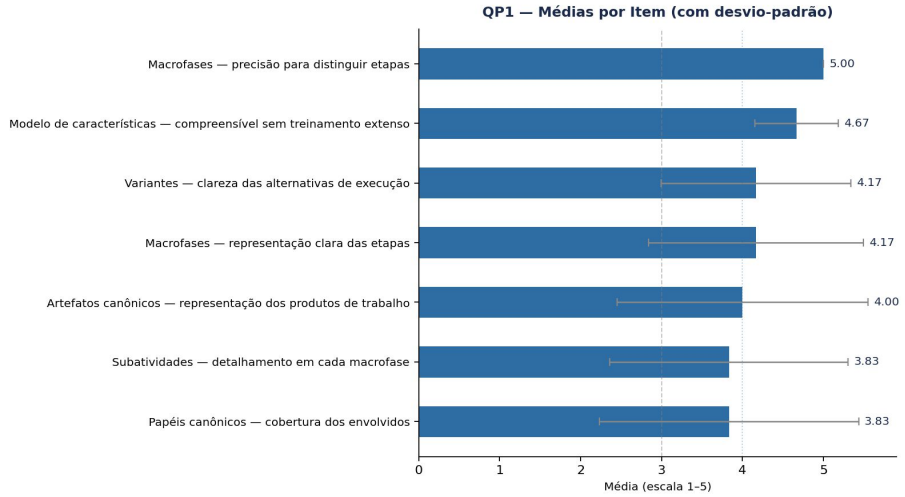
A QP1 investiga se a representação do domínio de jogos sérios por meio de uma linha de processos descreve o desenvolvimento desse tipo de sistema com clareza e completude. A Seção 2 do instrumento operacionaliza essa questão por meio dos elementos da LPrS-JS, com itens *Likert* LIKERT (1932) sobre a clareza dos elementos canônicos e a compreensibilidade do modelo de características.

Figura 7.17: QP1 - Distribuição das respostas *Likert* por item (porcentagem de respondentes).



Fonte: Elaborado pelo autor (2026).

Figura 7.18: QP1 - Médias por item com desvio-padrão. Itens ordenados por média crescente.



Fonte: Elaborado pelo autor (2026).

O item de maior consenso é a precisão das macrofases para distinguir etapas: os seis respondentes atribuíram nota 5 (mediana 5,0), configurando unanimidade entre os participantes. Na Figura 7.17, esse item aparece como uma barra inteiramente preenchida (nota 5), tornando visível a convergência. Esse resultado valida a estratégia de agrupamento adotada, que consolidou atividades dos nove processos originários em macrofases a partir de equivalências distintas.

Em contraste, o item sobre a representação geral das macrofases como etapas claras apresenta a distribuição mais heterogênea desta dimensão, com quatro respondentes em nota 5, mas dois em notas 2 e 3. A interpretação mais coerente com as

respostas discursivas é que as notas menores refletem a ausência de atividades vinculadas à dimensão criativa do *design* de jogos, lacunas que correspondem à ameaça de validade de construto reconhecida, relacionada ao *corpus* de processos originários de fonte acadêmica que pode não capturar atividades típicas de equipes que incluem artistas.

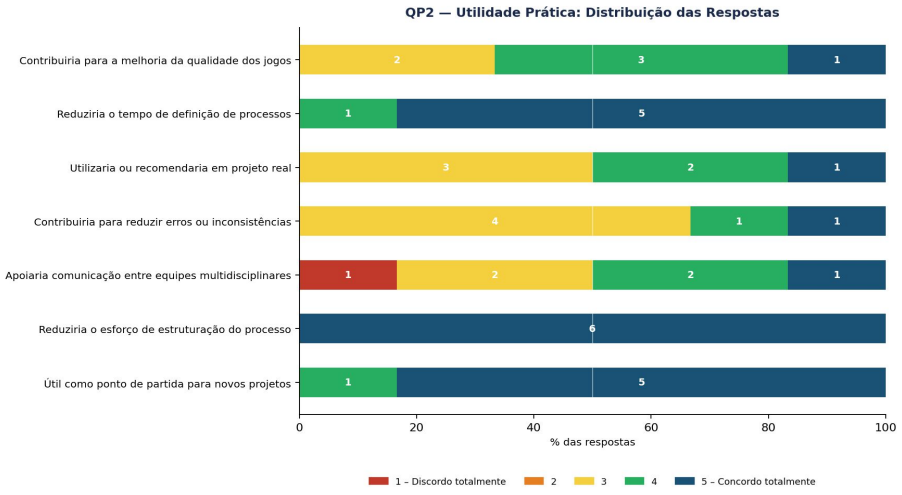
Papéis e artefatos canônicos apresentam as respostas mais dispersas da dimensão, percorrendo toda a amplitude da escala: há respondentes que percebem a cobertura como plena (nota 5) e ao menos um que a percebe como insuficiente (nota 1), como evidencia a Figura 7.18. As subatividades seguem o mesmo padrão misto, com respostas distribuídas ao longo da escala e sem concentração em um único ponto. Essa polarização também pode ser explicada pela limitação dos processos selecionados, sendo acadêmicos e não cobrindo com completude as possibilidades do que se espera de um processo para jogos padrão.

O modelo de características obteve mediana 5,0, com as respostas concentradas quase inteiramente nas notas 4 e 5 (Figura 7.17), confirmando que a representação visual da linha com a notação *OdysseyProcess-FEX* (TEIXEIRA, 2011) é acessível a especialistas mesmo sem formação prévia na metodologia. Esse resultado é relevante porque a compreensibilidade do modelo é condição necessária para a viabilidade da derivação sem auxílio dos criadores da linha (QP3).

QP2 - Utilidade Prática e Produtividade Percebida

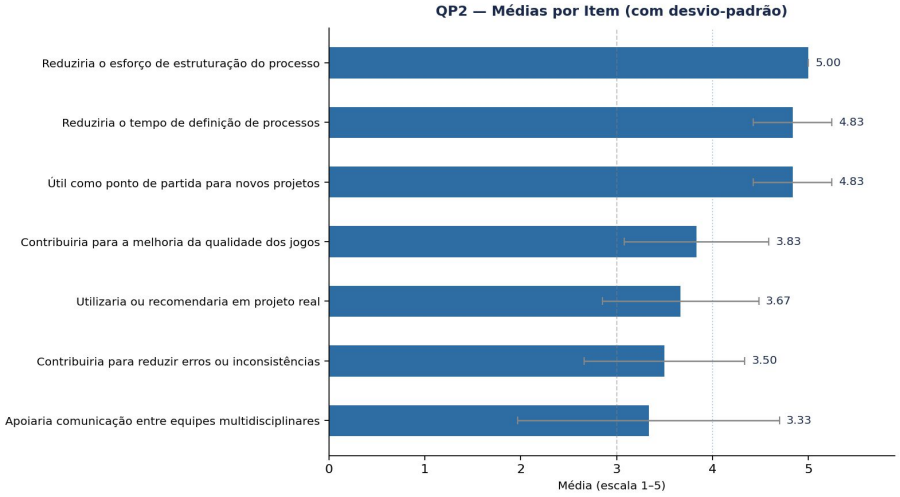
A QP2 investiga se a adoção de uma linha de processos para o domínio de jogos sérios oferece utilidade prática e é percebida como ganho de produtividade. A Seção 3 do instrumento operacionaliza essa questão, com base na LPrS-JS, por meio de itens *Likert* LIKERT (1932), complementados por dois itens da Seção 5 sobre redução de tempo e melhoria de qualidade.

Figura 7.19: QP2 - Distribuição das respostas *Likert* por item (% de respondentes).



Fonte: Elaborado pelo autor (2026).

Figura 7.20: QP2 - Médias por item com desvio-padrão.



Fonte: Elaborado pelo autor (2026).

A dimensão de utilidade apresenta o contraste mais evidente do instrumento. Na Figura 7.19, três itens reúnem todas as respostas nas notas 4 e 5: "Reduziria o esforço de estruturação", em que os seis respondentes atribuíram nota máxima (unanimidade); e "LPrS-JS útil como ponto de partida" e "Reduziria o tempo de definição de processos", ambos com cinco respondentes na nota 5 e um na nota 4 (mediana 5,0). Esse núcleo de consenso responde afirmativamente à QP2 e está relacionado às vantagens da área de reuso de artefatos de software; a reutilização de processos, ao encapsular variabilidades do domínio em componentes modulares, pode reduzir o esforço de definição em novos projetos (ROMBACH, 2006; TEIXEIRA *et al.*, 2019).

"Utilizaria ou recomendaria em projeto real" obteve mediana 3,5, com três dos seis respondentes na nota 3. Na Figura 7.19, essa barra apresenta segmentos em tons amarelo e verde em um distribuição mais equilibrada, correspondendo a uma representação visual de uma avaliação neutra para positiva com reservas. Essa *lacuna entre utilidade percebida e intenção de adoção* não é contraditória, os mesmos especialistas que reconhecem unanimemente que a linha economizaria esforço não demonstram a mesma convicção sobre seu uso em contexto real, o que sugere barreiras de adoção além do valor do artefato.

"Apoiaria a comunicação entre equipes multidisciplinares" obteve mediana 3,5 e foi o item mais disperso da dimensão, com respostas espalhadas por toda a escala (Figura 7.19). Na Figura 7.19, a barra correspondente é colorida, possuindo segmentos vermelho, amarelo, verde e azul. Essa dispersão reflete dúvida entre os respondentes. Esse resultado sinaliza que a adoção da linha não apresenta ferramentas de suporte comunicativo de profissionais de diferentes áreas com clareza.

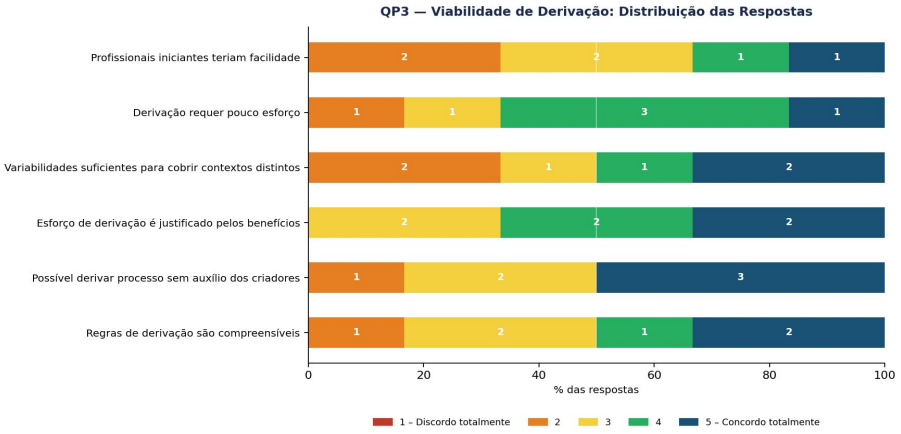
"Contribuiria para reduzir erros recorrentes" concentrou quatro dos seis respon-

dentes na nota 3 (mediana 3,0), padrão de convergência para a neutralidade distinto da dispersão observada nos demais itens. Esse padrão de *cautela* difere da dispersão observada nos demais itens, os especialistas convergem para a neutralidade. A interpretação mais provável, considerando as respostas discursivas de R5, é que a inferência da linha de processo para o produto final é considerada plausível, porém distante para ser avaliada com convicção a partir de uma interação com o catálogo disponibilizado.

QP3 - Viabilidade de Configuração e Derivação

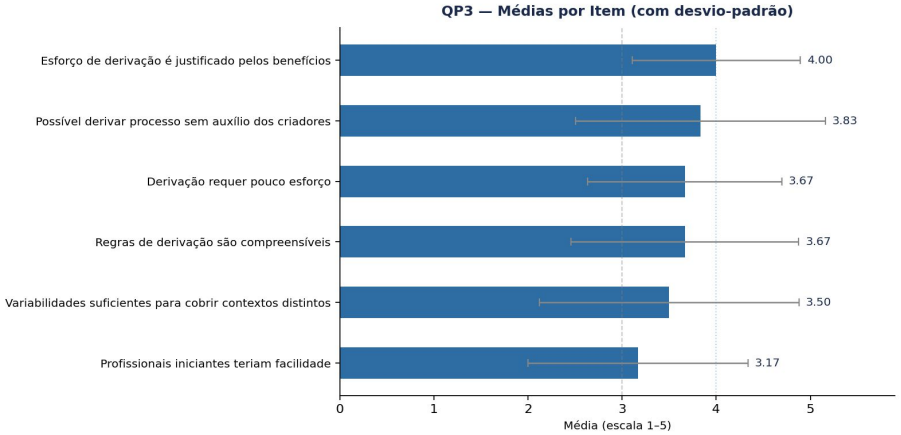
A QP3 investiga se a derivação de processos específicos a partir de uma linha de processos para jogos sérios é viável considerando o esforço exigido para sua configuração e derivação. Avaliada por meio da LPrS-JS, esta é a dimensão com as avaliações mais baixas do instrumento.

Figura 7.21: QP3 - Distribuição das respostas *Likert* por item (% de respondentes).



Fonte: Elaborado pelo autor (2026).

Figura 7.22: QP3 - Médias por item com desvio-padrão.



Fonte: Elaborado pelo autor (2026).

"Profissionais iniciantes teriam facilidade" obteve a mediana mais baixa de todo o instrumento (3,0) e o posicionamento mais reservado: dois respondentes na nota 2 e dois na nota 3 (Figura 7.21), único item da dimensão com presença expressiva de notas 2. Esse resultado não configura necessariamente uma falha do artefato; o dado indica, porém, que a adoção em contextos de equipes com baixa maturidade em Engenharia de Software exigiria esforços extras, talvez, através de um instrumento de apoio.

"Regras de derivação são compreensíveis" obteve mediana 3,5, com distribuição trimodal (Figura 7.21): parte do painel considera as regras compreensíveis, enquanto outra parte as percebe como insuficientemente explícitas. Esse padrão não é capturado por nenhuma medida de tendência central, apenas a distribuição visual expõe o padrão. Ela é coerente com a ameaça de validade interna, que reconhece que as decisões interpretativas de derivação não podem ser inteiramente dissociadas do conhecimento tácito dos pesquisadores. Além disso, um respondente apontou a dificuldade de identificação das regras de derivação no catálogo fornecido, o que pode configurar, também, falta de clareza no material de consulta.

"Esforço de derivação é justificado pelos benefícios" obteve mediana 4,0, o melhor resultado desta dimensão e dado central para a QP3. Mesmo os respondentes mais críticos à facilidade de uso reconhecem que o retorno compensa o investimento. Na Figura 7.22, esse item ocupa a posição mais à direita no gráfico de médias, reforçando visualmente que a percepção de custo-benefício é mais favorável do que a de facilidade de uso.

"Variabilidades suficientes para cobrir diferentes contextos" obteve mediana 3,5 e foi um dos itens mais dispersos do instrumento: dois respondentes na nota 2, um na 3, um na 4 e dois na 5 (Figura 7.21). Esse resultado precisa ser observado sob a perspectiva do critério de mandatoriedade disponíveis; além do conjunto limitado de processos selecionados. Especialistas cujos contextos de trabalho incluem domínios não representados nos nove processos analisados (exergames, aplicações terapêuticas especializadas) tendem a perceber a cobertura como insuficiente. A percepção de insuficiência pode, portanto, refletir tanto uma limitação real quanto a extrapolação do escopo da linha.

7.6.2.1 Avaliação Geral da Qualidade

A Seção 5 do instrumento observa uma avaliação global da LPrS-JS em um único item.

Figura 7.23: Distribuição das respostas na avaliação geral de qualidade.

Fonte: Elaborado pelo autor (2026).

A Figura 7.23 sintetiza o posicionamento do painel: um respondente na nota 3, três na nota 4 e dois na nota 5, resultando em mediana 4,0. A ausência de notas abaixo de 3, mesmo entre os respondentes que identificaram limitações estruturais relevantes, é importante. Ela indica que as críticas formuladas pelo painel configuram oportunidades de melhoria de um artefato cuja qualidade é reconhecida, e não objeções à validade da contribuição da linha como um todo. Esse resultado é coerente com o enquadramento da pesquisa em DSR como ciclo de construção, avaliação e refinamento, cujo objetivo não é produzir um artefato definitivo, mas identificar a trajetória de aprimoramento.

7.6.3 Triangulação: Perfil, Dados Quantitativos e Respostas Discursivas

Esta seção triangula os resultados quantitativos com as respostas discursivas e o perfil dos respondentes. A triangulação permite identificar padrões interpretativos que o dado numérico isolado não captura e avaliar a coerência das avaliações de cada especialista. O procedimento é organizado por respondente, com análise individual seguida de síntese.

R1 - Especialista técnico em desenvolvimento de jogos sérios

Perfil: Pós-doutorado, Eng. Software, 5-10 anos em software e jogos, membro técnico em JS, experiência consolidada em LPS/LPrS.

R1 é o único participante com experiência como programador/*designer* em projetos de jogos sérios, o que direciona suas críticas à prática de desenvolvimento que os demais não compartilham. Suas avaliações mais baixas concentram-se em papéis canônicos, subatividades e insuficiência das variabilidades, padrão coerente com a percepção de ausência de elementos do desenvolvimento técnico-criativo. Nas respostas discursivas, R1 identificou a ausência de papéis para arte visual e sonora, falta de atividades para *design* de personagens e narrativa, e inexistência de atividades sobre ciclo de engajamento. Essas lacunas são explicáveis pelo *corpus* de processos-fonte de origem acadêmica, conforme reconhecido na ameaça de validade externa.

Outro ponto complicado de elaborar dentro de uma linha de processo: a parte

> *criativa do game designer e o fluxo de criação de ideias muitas vezes não segue um processo muito bem definido. (R1)*

A crítica à opcionalidade de *feedback* e manutenção questiona o critério de mandatoriedade; se esses elementos aparecem como opcionais, estão presentes em menos de cinco dos nove processos-fonte, o que R1, com base em experiência prática, considera não correspondente à realidade. Esse dado qualitativo é indicador de que o limiar de mandatoriedade pode ter sido calibrado de forma conservadora para o subdomínio de jogos sérios. A sugestão de referenciar a Tétrade Elementar de Schell (SCHELL, 2014) para modelar os componentes do jogo (mecânicas, estética, narrativa, tecnologia) oferece um caminho teórico para futuras iterações.

R2 - *Designer* com visão analítica e experiência consolidada

Perfil: Doutorado, *Design*, >10 anos em software e jogos, gestão/pesquisa, experiência consolidada em LPS/LPrS.

R2 foi o respondente mais positivo do painel, sem nenhuma nota abaixo de 3. Seu perfil em *Design* pode explicar a receptividade ao modelo de características. Um *designer*, em geral, valoriza a capacidade de mapear e remixar alternativas de forma visual e analítica, funções que o modelo de características desempenha. Sua resposta mais interessante é que a linha lhe apresentou elementos que pretende incorporar ao próprio processo, indicando utilidade prática imediata que vai além da avaliação abstrata do artefato.

> *Essa estruturação e mapeamento dos modelos permite comparação analítica, acesso rápido, e base para hackear / subverter / remixar os modelos. (R2)*

A única ressalva de R2 seria o volume de informação para absorver de imediato, e sua sugestão de um questionário de mapeamento inicial para direcionar o uso converge com apontamentos dos demais respondentes. Para R2, trata-se de mediação de acesso, não de limitação estrutural da linha, sugerindo que parte das barreiras de adoção pode ser resolvida por artefatos de suporte sem alterações na LPrS-JS.

R3 - Especialista em produção com experiência ampla no domínio

Perfil: Doutorado, Eng. Produção, >10 anos em jogos, papel técnico + domínio + gestão, abordagens similares a LPS/LPrS, <2 anos em software formal.

R3 possui o perfil mais multidimensional do painel, com experiência simultânea nos três papéis possíveis em JS, e produziu as respostas discursivas mais elaboradas. Suas avaliações neutras em comunicação multidisciplinar, derivação autônoma e iniciantes refletem o diagnóstico de um especialista que conhece os desafios de implementação em equipes reais. Sua crítica central relaciona-se a tensão entre o formalismo da notação utilizada e as práticas ágeis da indústria. Ele nomeia o problema como "abismo semântico" entre a modelagem acadêmica e a operação diária

com *sprints* e Kanban, e propõe como solução a evolução do catálogo estático para um assistente de derivação automatizado (*wizard*) com exportação para ferramentas como Jira e Trello.

> *Infelizmente, jogos devem emergir do caos. Ferramentas processuais no game dev precisam ser quase invisíveis ou altamente flexíveis para não estragarem a moral e a criatividade da equipe. (R3)*

R3 também identifica o isolamento rígido entre *Design* do Jogo e Engenharia de Software como indução de antipadrão *waterfall* (cascata), crítica relevante dado que jogos requerem prototipagem paralela de mecânicas e código. Do ponto de vista de QP3, R3 reconhece o maior potencial da linha na pré-produção como "contrato de arquitetura", o que constitui resposta afirmativa à questão de viabilidade quando o escopo de uso é delimitado à fase de planejamento.

R4 - Especialista em LPS/LPrS sem experiência em jogos sérios

Perfil: Pós-doutorado, Ciência da Computação, <2 anos em software e jogos, sem participação em JS, experiência consolidada em LPS/LPrS.

R4 é o participante que mais diretamente testa o critério de recrutamento do painel, verificando se a linha é compreensível para profissionais sem experiência no domínio de jogos sérios. Suas avaliações são internamente coerentes com a limitação autodeclarada, com notas razoáveis nos itens de estrutura (nos quais seu expertise em LPS/LPrs o habilita) e notas neutras nos itens que exigiriam experiência com o domínio.

> *Não tenho nenhuma experiência com desenvolvimento de jogos sérios. Então, não sou capaz de avaliar se as macrofases, papéis e etc. correspondem ao que se espera desse tipo de domínio. (R4)*

A contribuição mais relevante de R4 está na identificação da *barreira cultural* como principal obstáculo à adoção; sendo especialista não técnica em JS, mas de familiaridade dos profissionais com a linguagem de LPS/LPrS. Esse resultado é consistente com a ameaça de validade de construto e sugere que a compreensibilidade da LPrS-JS, positivamente avaliada pelo painel especialista, pode ser significativamente menor para equipes de desenvolvimento sem formação em processos de software.

R5 - Especialista em ES com contato instrumental com jogos sérios

Perfil: Pós-doutorado, Eng. Software, >10 anos em software e jogos, gestão/-pesquisa em JS, experiência consolidada em LPS/LPrS.

R5 orientou projetos de jogos sérios como especialista em ES, sem assumir o papel técnico primário no domínio. Esse posicionamento produziu avaliações altas nos itens de utilidade (nota 5 em utilidade como ponto de partida, redução de esforço e tempo) e cautelosas nos de impacto indireto (nota 3 em comunicação multidisciplinar

e redução de erros). Nas respostas discursivas, R5 explicitou a razão da distinção, a LPrS codifica o conhecimento de nove processos e permite gerar processos específicos com alguma facilidade, sendo uma afirmação direta sobre o valor instrumental. Mas a inferência de que isso se traduz em melhor comunicação ou menor número de erros no produto final não é possível afirmar.

A contribuição técnica mais diagnóstica de R5 é a de maior relevância para a revisão imediata da LPrS-JS foi a de que ele não encontrou as regras de derivação explicitadas no material, mesmo após revisitar o vídeo de apresentação.

> *Vi as marcações R e X no diagrama, mas senti falta de regras explícitas. Por exemplo, R indica dependência, mas isso não é necessariamente bidirecional. (R5)*

Essa percepção é uma lacuna operacional e não necessariamente uma inconsistência estrutural. As regras de composição implícitas para os autores da linha não foram percebidas como explícitas por um especialista externo em LPrS-JS. R5 propõe um conjunto de questões que codifiquem as regras de derivação em alto nível, e ferramental que automatize a verificação de consistência na seleção de variantes.

R6 - Especialista em ES com conhecimento teórico de LPS

Perfil: Doutorado, Eng. Software, >10 anos em software e jogos, gestão/pesquisa, conhecimento teórico de LPS/LPrS.

R6 combina experiência em ES e jogos com familiaridade apenas teórica em LPS/LPrS, posicionando-se como especialista que domina o domínio mas enfrenta a barreira de entrada no formalismo da linha. Atribuiu nota 1 em comunicação multidisciplinar e nota 2 em iniciantes, as avaliações mais baixas do painel nestes itens, com justificativas nas respostas discursivas.

A crítica com mais ênfase de R6 é a ausência sistemática de profissionais de arte (arte conceitual, personagens, animação, som, música) como papéis reconhecidos na LPrS-JS. Além de constatar a ausência, R6 argumenta que esses profissionais *geram restrições e criam possibilidades* ao longo do desenvolvimento, perspectiva que aponta para uma lacuna de modelagem nas dependências entre papéis e artefatos, e não apenas em um item de lista.

> *Roles devem ser tratados mais detalhadamente, desenvolvendo mais a ideia de capacidades do que de papéis formais. (R6)*

A proposta de uma dimensão em paralelo à estrutura atual, na qual a linha seria consultada quando um problema emergisse, e não apenas no planejamento, é a principal sugestão do respondente. Ela é coerente e poderia ampliar a aplicabilidade da LPrS-JS em contextos diversos de desenvolvimento, como comportamentos iterativos.

7.6.4 Síntese da Análise

Com a análise individual cruzada com o referencial teórico e os dados quantitativos, foi possível identificar quatro padrões temáticos que estruturam os achados da avaliação do painel de especialistas.

Tema 1 - Valor instrumental consensual, intenção de adoção moderada

O item de unanimidade absoluta, redução do esforço de estruturação, é reforçado discursivamente por todos os respondentes, incluindo os mais críticos à adoção prática. A dissociação com a intenção de uso (mediana 3,5) não contradiz esse resultado, mas contextualiza a existência de barreiras de adoção além do valor do artefato, como custo de aprendizagem da notação, ausência de ferramental de derivação automatizado na proposta e incompatibilidade percebida com fluxos ágeis.

Tema 2 - Convergência independente sobre papéis artísticos

R1 (especialista técnico) e R6 (especialista em gestão/pesquisa) identificaram, de forma independente, a ausência de profissionais de arte e de atividades da dimensão criativa como a principal lacuna de cobertura. Essa convergência entre perfis distintos fortalece a relevância do achado, demonstrando caráter estrutural. Na perspectiva da ameaça de validade de construto, o resultado indica que o *corpus* de processos originários, embora diversificado em domínio de aplicação, não captura adequadamente equipes que incluem artistas, uma limitação relacionada à origem acadêmica dos processos analisados.

Tema 3 - Demanda convergente por derivação assistida

R3 e R5, partindo de perspectivas distintas, atrito com práticas ágeis (R3) e ausência percebida de regras explícitas (R5), propuseram independentemente a mesma solução: um mecanismo de assistência à derivação. A viabilidade de derivação autônoma está condicionada à disponibilidade de suporte explícito ao processo de derivação, não inteiramente substituível pela documentação atual. A implementação de derivação assistida é, portanto, a recomendação com maior suporte do painel. No entanto, o ferramental *Odyssey*, atualmente, já possui funcionalidades de apoio a derivação; mas esse não foi disponibilizado aos respondentes em razão da complexidade do uso da ferramenta de modelagem por um indivíduo não treinado.

Tema 4 - Dependência de perfil na percepção de viabilidade

A percepção de viabilidade (QP3) é a dimensão com maior variação associada ao perfil. Especialistas com experiência consolidada em LPS/LPrS (R1, R2, R5) avaliam a derivação autônoma como mais viável do que aqueles com *background* predominantemente em jogos (R3, R6) ou sem experiência no domínio (R4). Essa dependência de perfil, prevista no protocolo de recrutamento do painel, tem implicação prática de que a LPrS-JS é mais imediatamente utilizável por engenheiros de processos com formação em LPrS, e sua disseminação para equipes de desenvolvi-

mento de jogos sem esse conhecimento requer programa de capacitação ou artefatos de mediação complementares. Apesar disso, dentre os respondentes com menos experiência no ramo de software, foi observado um entendimento satisfatório dos artefatos.

Cabe explicitar que a LPrS-JS não foi refinada após a avaliação com especialistas. De modo coerente com o enquadramento da pesquisa, que percorre o ciclo de construção, avaliação e refinamento uma única vez, a avaliação teve por objetivo identificar a trajetória de aprimoramento da linha, e não produzir um artefato definitivo; o refinamento efetivo é tratado como trabalho futuro (Seção 8.4). Sugestões pontuais, formuladas por um único respondente, como a adoção da Tétrade Elementar de Schell (SCHELL, 2014), o tratamento de papéis como capacidades ou a representação explícita de fluxos ágeis, permanecem como candidatas, condicionadas à confirmação de sua recorrência. Já os achados convergentes, identificados de forma independente por respondentes de perfis distintos, oferecem base mais sólida para retroalimentar a linha. O mais robusto é a ausência de papéis artísticos e de subatividades ligadas à dimensão criativa do *design* de jogos, apontada de forma independente por R1 e R6 e refletida quantitativamente nos itens de maior dispersão da avaliação (papéis e variabilidades). Por ter caráter estrutural e recorrente, essa lacuna constitui o principal candidato a um refinamento futuro do modelo de características, na direção indicada na Seção 8.4.

Capítulo 8

Conclusão e Trabalhos Futuros

Este capítulo encerra a pesquisa sintetizando os resultados obtidos, respondendo às questões de pesquisa formuladas na introdução e discutindo as limitações do trabalho. São também apresentadas as perspectivas para investigações futuras, que emergem tanto das lacunas identificadas durante a construção e avaliação da LPrS-JS quanto das sugestões registradas pelo painel de especialistas.

8.1 Síntese das Contribuições

A motivação central desta pesquisa residia na ausência de uma infraestrutura sistemática e reutilizável para o desenvolvimento de jogos sérios digitais. A literatura evidenciava que, embora o domínio dos JS apresentasse crescimento expressivo em termos de relevância socioeconômica e educacional, a produção metodológica permanecia fragmentada, dependente de adaptações *ad hoc* e incapaz de promover a transferência estruturada de conhecimento entre projetos (BENIGNO *et al.*, 2024; OLIVEIRA *et al.*, 2020). Diante dessa lacuna, o trabalho propôs a construção e a avaliação de uma Linha de Processos de Software para Jogos Sérios (LPrS-JS), fundamentada na metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016) e no método *ArchToDSSA* (VASCONCELOS e WERNER, 2011), como mecanismo para gerenciar a variabilidade inerente ao domínio e elevar a sistematicidade do desenvolvimento.

A primeira contribuição do trabalho é o próprio método de análise comparativa de processos heterogêneos, apresentado no Capítulo 4. Esse método operacionaliza a etapa de Análise do Domínio de Processos de Software prevista pela *Odyssey-ProcessReuse* (TEIXEIRA, 2016) para o contexto específico de fontes acadêmicas com granularidades, nomenclaturas e estruturas distintas. O procedimento percorre de forma sequencial as etapas de delimitação de escopo, captura de conhecimento, extração de elementos, definição de glossário, construção do vocabulário canônico, análise de mandatoriedade e identificação de variabilidades, culminando na mo-

delagem em notação *OdysseyProcess-FEX* (TEIXEIRA, 2011). A explicitação de heurísticas e de critérios de decisão em cada etapa é um aspecto relevante no método desenvolvido, reduzindo a dependência do julgamento tácito dos pesquisadores, mas não eliminando, pelo uso de mecanismos como o registro de notas e observações nos artefatos intermediários e o uso de limiares quantitativos de mandatoriedade.

A segunda contribuição consiste na própria LPrS-JS, construída no Capítulo 5, a partir da aplicação do método sobre nove processos de desenvolvimento de JS identificados na literatura acadêmica. A linha representa o domínio por meio de seis macrofases canônicas organizadas em torno de um vocabulário canônico que integra papéis, artefatos, subatividades e variantes. A modelagem no ferramental *Odyssey*, com uso da notação *OdysseyProcess-FEX* (TEIXEIRA, 2011), proporciona a rastreabilidade entre os elementos canônicos pelos modelos de características, preservando a inteligibilidade do conhecimento construído.

A terceira contribuição é a verificação por instanciação, descrita no Capítulo 6. A derivação de um processo a partir da LPrS-JS, tomando como referência um processo de JS documentado na literatura acadêmica, demonstrou a capacidade de cobertura da linha sobre um contexto não incluído entre seus processos fundadores. As seis macrofases foram avaliadas, com as quatro mandatórias plenamente ativadas e as duas opcionais tratadas de modo adequado ao contexto do processo de referência. A análise da derivação frente à linha identificou tanto a robustez da LPrS-JS quanto uma lacuna estrutural nesse cenário, sendo a ausência de um artefato canônico que represente insumos externos de *design*, necessidade geralmente vista em projetos orientados a domínios institucionais.

A quarta contribuição é o Catálogo de Processos desenvolvido para a avaliação, descrito no Capítulo 7. Esse sistema *web* reúne as modelagens de características, os artefatos intermediários, o vocabulário canônico e o glossário em uma interface navegável que permite a interação com os produtos da linha sem exigir familiaridade com o ferramental *Odyssey*. O catálogo constitui, em si, um ativo reutilizável para a comunidade de pesquisa em Engenharia de Software voltada ao desenvolvimento de JS.

Em conjunto, essas contribuições respondem ao objetivo geral da pesquisa, a sistematização e o amadurecimento do desenvolvimento de JS por meio da Engenharia de Linha de Processos de Software, e oferecem à comunidade um conjunto de artefatos, método, linha e catálogo, que pode ser adotado, estendido e avaliado em contextos organizacionais e acadêmicos futuros.

8.2 Respostas às Questões de Pesquisa

As três questões de pesquisa formuladas no Capítulo 1 são respondidas a seguir, com base nos dados coletados na verificação por instanciação e na avaliação por especialistas, interpretados como evidência sobre a abordagem de linha de processos a partir da LPrS-JS, sua instância representativa.

QP1 - A representação do domínio de jogos sérios por meio de uma linha de processos descreve o desenvolvimento desse tipo de sistema com clareza e completude?

A evidência quantitativa indica cobertura e clareza satisfatórias. O item de maior consenso em todo o instrumento foi a precisão das macrofases para distinguir etapas do processo, com unanimidade entre os respondentes. O modelo de características, representado pela notação *OdysseyProcess-FEX* (TEIXEIRA, 2011), foi considerado compreensível mesmo por participantes sem formação prévia na metodologia, o que é condição necessária para a rastreabilidade entre objetivos do domínio e elementos técnicos do processo. A instanciação apresentada no Capítulo 6 demonstrou que a linha é capaz de cobrir um processo inédito com coerência, ativando os elementos mandatórios e oferecendo as variantes adequadas ao contexto analisado.

A dimensão qualitativa, no entanto, dá perspectiva a esse achado. Dois respondentes com experiência prática em equipes de desenvolvimento, identificaram de forma convergente e independente a ausência de papéis artísticos e de subatividades relacionadas à dimensão criativa do *design* de jogos. Essa lacuna de cobertura é coerente com a ameaça de validade de construto reconhecida no Capítulo 3: o corpus de processos analisados, de origem acadêmica, não representa adequadamente equipes que incluem profissionais de arte, limitação que a linha herda de seus processos originários. A resposta à QP1 é, portanto, positiva com ressalvas estruturais identificadas e passíveis de ajustes em evoluções futuras.

QP2 - A adoção de uma linha de processos para o domínio de jogos sérios oferece utilidade prática e é percebida como ganho de produtividade pelos profissionais envolvidos no desenvolvimento?

A dimensão de utilidade registrou o resultado mais consistente da avaliação. O item que verificou se a linha reduziria o esforço de estruturação de um novo processo foi o único a atingir unanimidade em toda a avaliação, com todos os respondentes atribuindo nota máxima. A percepção de redução de esforço e de tempo foi reforçada pela concentração das respostas na nota máxima em ambos os itens relacionados (cinco respondentes na nota 5 e um na nota 4; mediana 5,0), constituindo um núcleo de consenso sobre a utilidade instrumental da linha. Um dos respondentes, especialista com experiência em orientação de projetos de jogos sérios, explicitou o mecanismo: a LPrS-JS codifica o conhecimento de nove processos e permite gerar

processos específicos com esforço reduzido, o que é o fundamento da vantagem de produtividade no campo da reutilização.

A intenção de adoção em contexto real apresentou resultado moderado, com mediana 3,5 e três respondentes em nota 3. Essa lacuna entre utilidade percebida e intenção de adoção não contradiz a resposta desta questão de pesquisa, mas ela sinaliza que barreiras de adoção além do valor do artefato, como custo de aprendizagem da notação, ausência de ferramental de derivação automatizado e incompatibilidade percebida com fluxos ágeis; precisam ser consideradas para que a utilidade instrumental converta-se em adoção efetiva. A resposta à QP2 apresentou um saldo positivo, a linha sendo percebida como ganho de produtividade, mas sua adoção ampla requer artefatos de mediação e programas de capacitação complementares.

QP3 - A derivação de processos específicos a partir de uma linha de processos para jogos sérios mostra-se viável considerando o esforço exigido para sua configuração e derivação?

A viabilidade de configuração e derivação foi a dimensão com as avaliações mais cautelosas, ainda que majoritariamente situadas acima do ponto neutro. O item que investigou a facilidade para profissionais iniciantes obteve a mediana mais baixa de todo o questionário (3,0), indicando que a linha, em seu estado atual, é mais acessível a engenheiros de processos com formação em LPrS do que a equipes de desenvolvimento sem esse perfil. Esse resultado é coerente com a natureza do artefato, linhas de processos de software são instrumentos de especialistas em processos, e não guias operacionais para desenvolvedores.

O item mais favorável da dimensão, porém, fornece a resposta central a esta questão. O esforço de derivação foi considerado justificado pelos benefícios obtidos, mesmo pelos respondentes mais críticos à facilidade de uso. Um dos respondentes identificou a ausência de regras de derivação explicitamente documentadas no catálogo como a principal barreira à autonomia do usuário, lacuna operacional que não implica inconsistência estrutural da linha, mas indica que os mecanismos de derivação, embora presentes no ferramental *Odyssey*, precisam ser tornados mais acessíveis nos materiais de apoio. A resposta à QP3 é condicionalmente positiva, sendo a viabilidade da derivação reconhecida em termos de custo-benefício, mas estando condicionada à disponibilidade de suporte ao processo de derivação, seja por meio de guias, de ferramental assistido ou de capacitação prévia.

8.3 Limitações e Ameaças à Validade

As limitações desta pesquisa decorrem, em parte, das ameaças à validade discutidas no Capítulo 3 e, em parte, de restrições práticas do percurso metodológico adotado. Sua identificação é relevante para que os resultados sejam interpretados corretamente

e para que estudos futuros possam partir de uma posição mais informada sobre os limites do trabalho.

No plano da validade interna, a dimensão interpretativa da análise de processos constitui a principal limitação. As decisões de agrupamento semântico realizadas nas etapas de definição do vocabulário canônico e de identificação de macrofases não são inteiramente dissociáveis da perspectiva dos pesquisadores envolvidos, ainda que o método proposto reduza essa dependência por meio de heurísticas e critérios quantitativos de mandatoriedade. O limiar de 60% adotado para a classificação de um elemento como mandatório é uma escolha metodológica que influencia diretamente o escopo da linha: como comentado, limiares mais restritivos produziriam um núcleo mais compacto, enquanto limiares mais permissivos incorporariam elementos presentes em contextos específicos ao conjunto obrigatório. Esse ponto foi reconhecido como ameaça de validade de construto e não dispõe de solução universal, sua calibração é uma decisão que deve ser revisitada à medida que o corpus de processos analisados se expanda.

No plano da validade de construto, a origem acadêmica dos nove processos fundadores constitui a limitação mais relevante. Conforme apontado por respondentes na etapa de avaliação, equipes de desenvolvimento que incluem profissionais de arte não são adequadamente representadas nos processos analisados, o que se reflete na ausência de papéis artísticos e subatividades criativas no modelo da linha. Essa limitação não invalida a contribuição da LPrS-JS, mas restringe sua completude a contextos de desenvolvimento próximos ao perfil acadêmico dos processos originários.

No plano da validade externa, os resultados obtidos a partir de processos documentados na literatura acadêmica podem ter generalização limitada para contextos industriais, nos quais os processos costumam ser menos formalizados, mais dependentes de conhecimento tácito das equipes e mais sensíveis às práticas ágeis correntes.

No plano da validade de conclusão, a avaliação foi conduzida como um estudo exploratório de percepção com um painel reduzido de seis especialistas. Os dados quantitativos têm função estritamente descritiva, resumindo o posicionamento do painel e não sustentam inferências nem generalização para uma população. Sua leitura privilegiou a distribuição das respostas, e os achados foram ancorados na triangulação com as respostas discursivas e o perfil de cada respondente. Por essa razão, os resultados são apresentados como indícios a serem confirmados em estudos de maior porte, e não como conclusões definitivas.

8.4 Trabalhos Futuros

Os achados desta pesquisa, combinados às lacunas identificadas durante a construção, verificação e avaliação da LPrS-JS, e sistematizados com base nos resultados discursivos e quantitativos do painel de especialistas, apontam para um conjunto de direções de investigação futura organizado em quatro eixos complementares de expansão e manutenção, evolução da modelagem, desenvolvimento e integração de ferramental de apoio e avaliação de impacto em contextos reais.

As direções apresentadas a seguir distinguem-se pelo grau de representação recebido na avaliação. Recebem prioridade os encaminhamentos sustentados por achados convergentes (identificados de forma independente por respondentes de perfis distintos e, quando possível, influenciados pelos itens quantitativos), pois a convergência reduz o risco de que a direção reflita a perspectiva isolada de um participante. É o caso da ampliação do corpus para cobrir papéis e atividades da dimensão criativa, lacuna apontada por R1 e R6, e da derivação assistida, demanda formulada de modo independente por R3 e R5. Encaminhamentos baseados na contribuição de um único respondente, como a modelagem de papéis como capacidades (R6) ou a representação explícita de fluxos ágeis (R3), são apresentados como candidatos, cuja incorporação depende da confirmação de sua recorrência. Por fim, direções que extrapolam o escopo avaliado, como a avaliação da eficácia pedagógica dos jogos derivados, são tratadas como extensões exploratórias de longo prazo, e não como decorrência direta dos achados.

8.4.1 Expansão e Manutenção da Linha

Ampliação do corpus de processos originários. A adição de novos processos à LPrS-JS é a recomendação de maior impacto imediato sobre a completude da linha. A avaliação sinalizou, por meio da maior dispersão das respostas do instrumento nos itens de papéis canônicos e variabilidades, que parte dos especialistas percebe a cobertura atual como insuficiente para determinados contextos. Uma das causas mais prováveis para essa percepção é a origem predominantemente acadêmica dos nove processos originários, que tende a sub-representar equipes com profissionais de arte. A inclusão de processos originados em contextos industriais e de relatos de casos reais permitiria redistribuir as frequências de ocorrência dos elementos e recalibrar a linha e os elementos canônicos. Adicionalmente, a incorporação de processos voltados a subdomínios não representados no corpus atual, como aplicações de treinamento militar ou jogos para reabilitação em frentes especializadas, ampliaria o espectro de variantes disponíveis para a derivação. O método proposto no Capítulo 4 foi concebido com um mecanismo de iteração que suporta a inclusão incremental de novos processos sem demandar a reconstrução da linha, o que facilita

sua expansão progressiva por outros pesquisadores.

***Survey* de práticas reais de desenvolvimento de jogos sérios.** A triangulação entre o perfil dos respondentes e suas avaliações discursivas revelou um conflito estrutural entre o formalismo acadêmico da LPrS-JS e as práticas operacionais da indústria de jogos. Essa tensão não pode ser inteiramente resolvida pela ampliação do corpus bibliográfico, exigindo a coleta da visão de profissionais atuantes, incluindo elementos do processo que geralmente são documentados na literatura, como fluxos de prototipagem criativa, dinâmicas de equipe multidisciplinar e mecanismos informais de validação pedagógica. A condução de um *survey* estruturado com profissionais da indústria permitiria identificar quais atividades, papéis e artefatos são percebidos como essenciais na prática, complementando o corpus bibliográfico com evidências de casos reais.

Evolução do catálogo de processos. Como uma forma de evoluir a infraestrutura de suporte à linha, é sugerida a implementação de uma versão interativa do catálogo de processos desenvolvido neste trabalho, de modo a permitir a engenheiros de software e a pesquisadores a inclusão e a visualização otimizada de seu domínio, facilitando a manutenção e a utilização continuada da LPrS-JS. Além disso, é sugerida a inclusão de uma funcionalidade que permita a comunicação dos dados entre o catálogo e o ferramental *Odyssey*, de modo a amplificar o potencial de uso da linha com a associação das ferramentas em um ambiente integrado. Essa evolução posicionaria o catálogo como ponto de entrada para o ecossistema de construção, derivação e manutenção de linhas de processos.

8.4.2 Evolução da Modelagem

Modelagem de papéis como capacidades. A proposta de um dos respondentes da avaliação de tratar os papéis da linha como capacidades, em vez de posições formais, oferece um caminho teórico para mitigar a limitação de representação de profissionais de arte e de papéis híbridos comuns em projetos acadêmicos de pequena escala. Essa reformulação permitiria que a linha capturasse, com maior fidelidade, a sobreposição de funções típica de equipes multidisciplinares de JS, tornando as instruções de derivação mais aderentes às configurações reais de projetos.

Verificação e validação do modelo de características. A avaliação conduzida neste trabalho contemplou a verificação da capacidade de derivação da linha por meio de uma instanciação (Capítulo 6) e a avaliação da qualidade percebida dos artefatos por especialistas (Capítulo 7), mas não submeteu o modelo de características resultante a um procedimento dedicado de verificação e validação. Como direção futura, recomenda-se a aplicação de técnicas de análise de modelos de características para verificar a corretude interna da LPrS-JS, identificando eventuais anomalias. À

verificação interna deve seguir-se uma etapa de validação, na qual um conjunto de configurações derivadas do modelo seja confrontado sistematicamente com processos reais ou esperados do domínio, de modo a confirmar que as instâncias geradas são internamente válidas, coerentes e úteis enquanto processos de desenvolvimento de jogos sérios. Essa atividade ganha relevância à medida que o corpus de processos e o número de variantes se expandem, pois o crescimento do espaço de configurações aumenta o risco de anomalias e torna a inspeção manual progressivamente inviável; sua automação, possivelmente integrada ao ferramental *Odyssey*, complementaria a verificação por instanciação e a avaliação por especialistas, fechando o ciclo de construção, avaliação e refinamento previsto pela *Design Science Research*.

8.4.3 Ferramental e Apoio à Adoção

Derivação assistida - evolução do *Odyssey*. A demanda por um mecanismo de assistência à derivação constitui a recomendação com maior suporte do painel de especialistas. A viabilidade de derivação autônoma foi comprometida pela ausência de regras de composição explicitamente documentadas: as marcações R (dependência) e X (exclusão mútua) no diagrama do modelo de características foram reconhecidas, mas sua semântica de composição não foi percebida como suficientemente clara para guiar decisões autônomas de derivação. O *Odyssey*, ferramenta de apoio à instanciação de processos da metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016), representa o mecanismo mais adequado para mitigar essa lacuna, sendo evoluído para incorporar um fluxo guiado de derivação, análogo a um assistente (*wizard*), ele poderia conduzir o usuário pelas decisões de seleção de variantes e registrar as decisões tomadas com suas justificativas. A implementação desse mecanismo como extensão do ferramental *Odyssey* ou como módulo integrado ao catálogo web construído neste trabalho permitiria guiar o usuário da linha ao longo das escolhas de variantes, ampliando o potencial de adoção prática.

Modelagem da dimensão ágil do fluxo de desenvolvimento. Um respondente identificou o fluxo de macrofases como um antipadrão *waterfall*: em projetos reais de jogos, a prototipagem de mecânicas e o desenvolvimento de código ocorrem de forma paralela e iterativa, e não sequencial. Para mitigar esse comportamento, sugere-se a proposição de evoluções nas notações comportamentais da linguagem *OdysseyProcess-FEX* (TEIXEIRA, 2011), de modo a fornecer uma melhor representação de fluxos que, em sua natureza, não correspondem ao padrão em cascata.

Redução da barreira cultural de adoção. O item sobre apoio à comunicação entre equipes multidisciplinares foi o de maior dispersão na dimensão de utilidade e o que mais expõe a tensão entre o valor técnico da linha para engenheiros de processos e sua acessibilidade para profissionais de outras áreas. Um dos respondentes nomeou

esse obstáculo como "barreira cultural", não uma limitação técnica do artefato, mas de familiaridade dos profissionais com a linguagem de linhas de processos. O refinamento sugerido consiste em desenvolver materiais de mediação que traduzam os conceitos e a notação da LPrS-JS para linguagens acessíveis a pedagogos, terapeutas, artistas e gestores; o público que compõe a equipe multidisciplinar típica de um projeto de jogo sério. Isso pode incluir guias visuais simplificados, glossários de domínio cruzado e um questionário que direcione o usuário para as partes da linha mais relevantes ao seu papel no projeto. Se implementado, esse refinamento ampliaria o alcance da LPrS-JS para além da comunidade de Engenharia de Software.

8.4.4 Avaliação de Impacto

Avaliação de impacto em desenvolvimento comparado. A avaliação conduzida neste trabalho verificou a qualidade percebida da LPrS-JS por especialistas, mas não avaliou seu impacto efetivo sobre o produto final ou sobre o desempenho das equipes que a adotam. Um estudo comparado que acompanhe equipes de desenvolvimento de JS utilizando a linha e equipes que adotam abordagens *ad-hoc* permitiria verificar se a redução de esforço percebida se traduz em benefícios mensuráveis, como tempo de definição de processo, incidência de erros de rastreabilidade pedagógica ou satisfação da equipe.

Avaliação da eficácia pedagógica dos jogos derivados. Além do impacto no processo de desenvolvimento, uma direção de pesquisa relevante é a avaliação da qualidade pedagógica dos jogos sérios produzidos com apoio de processos derivados da LPrS-JS em comparação com produtos desenvolvidos sem o uso de processos derivados da linha. Essa investigação permitiria verificar se o uso da técnica também se traduz em resultados de aprendizagem mensuráveis nos usuários finais, contribuindo para a consolidação da relação entre qualidade de processo e qualidade do produto no domínio dos JS.

Os refinamentos apresentados não são independentes entre si, sendo a ampliação do corpus e o *survey* de práticas reais alimentam a revisão das variabilidades e papéis; a evolução do *Odyssey* constitui um componente de um ecossistema de apoio à derivação; e os materiais de mediação cultural dependem, em parte, da evolução do catálogo como ponto de acesso unificado. As direções apresentadas não são mutuamente excludentes e podem ser executadas de forma paralela por grupos de pesquisa distintos. O método, a linha e o catálogo disponibilizados constituem uma base sobre a qual essas extensões podem ser construídas de forma incremental, preservando a rastreabilidade com os artefatos originais e permitindo que o conhecimento acumulado neste trabalho seja ampliado e refinado pela comunidade de Engenharia de Software voltada ao desenvolvimento de jogos sérios.

8.5 Considerações finais

Ao sistematizar o conhecimento disperso na literatura em uma infraestrutura de reutilização explícita, este trabalho representa um passo em direção ao amadurecimento metodológico do domínio, contribuindo para que o desenvolvimento de jogos sérios digitais avance de uma prática predominantemente *ad-hoc* para uma disciplina com bases mais sólidas e passíveis de evolução sistemática com apoio da reutilização.

Referências Bibliográficas

ABRAHAMSSON, P., SALO, O., RONKAINEN, J., et al., 2002, "Agile Software Development Methods: Review and Analysis", *Proc. Espoo 2002*, (01), pp. 3–107. doi: 10.48550/arXiv.1709.08439.

ABT, C. C., 1970, *Serious games*. New York : Viking Press. Disponível em: <https://hal.science/hal-00702964>. Second edition in 1987 by University Press of America : https://rowman.com/ISBN/9780819161475 Extracts from the 2nd edition available on Google Books (see link below).

ADENOWO, A., ADENOWO, B., 2020, "Software Engineering Methodologies: A Review of the Waterfall Model and Object- Oriented Approach", *International Journal of Scientific and Engineering Research*, v. 4 (09), pp. 427–434.

ALCOVER, E., JAUME-I CAPÓ, A., MOYÀ ALCOVER, G., 2018, "PROGame: A process framework for serious game development for motor rehabilitation therapy", *PLOS ONE*, v. 13 (05), pp. e0197383. doi: 10.1371/journal.pone.0197383.

ALEEM, S., CAPRETZ, L. F., AHMED, F., 2016, "Game development software engineering process life cycle: a systematic review", *Journal of Software Engineering Research and Development*, v. 4, n. 1 (nov.). ISSN: 2195-1721. doi: 10.1186/s40411-016-0032-7. Disponível em: <http://dx.doi.org/10.1186/s40411-016-0032-7>.

ALSAQQA, S., SAWALHA, S., ABDEL-NABI, H., 2020, "Agile Software Development: Methodologies and Trends", *International Journal of Interactive Mobile Technologies (iJIM)*, v. 14, n. 11, pp. 246. ISSN: 1865-7923. doi: 10.3991/ijim.v14i11.13269. Disponível em: <http://dx.doi.org/10.3991/ijim.v14i11.13269>.

ALVAREZ, J., DJAOUTI, D., 2011, "An introduction to Serious Game Definitions and Concepts", *In Proceedings of the Serious Games Simulation for Risks Management Workshop*, (01), pp. 11–15.

AND PRABOWO, H., GAOL, F. L., HIDAYANTO, A. N., 2022, "Comparison of the System Development Life Cycle and Prototype Model for Software Engineering", *International Journal of Emerging Technology and Advanced Engineering*, v. 12, n. 4 (abr.), pp. 155–162. ISSN: 2250-2459. doi: 10.46338/ijetae0422_19. Disponível em: <http://dx.doi.org/10.46338/ijetae0422_19>.

ARJORANTA, J., 2019, "How to Define Games and Why We Need to", *The Computer Games Journal*, v. 8, n. 3-4 (maio), pp. 109–120. ISSN: 2052-773X. doi: 10.1007/s40869-019-00080-6. Disponível em: <http://dx.doi.org/10.1007/s40869-019-00080-6>.

ASLAN, S., BALCI, O., 2015, "GAMED: Digital educational game development methodology", *SIMULATION*, v. 91 (04), pp. 307–319. doi: 10.1177/0037549715572673.

BABAR, M. A., CHEN, L., SHULL, F., 2010, "Managing Variability in Software Product Lines", *IEEE Software*, v. 27, n. 3, pp. 29–39.

BACHMANN, F., CLEMENTS, P. C., 2005, *Variability in Software Product Lines*. doi: 10.21236/ada450337. Disponível em: <http://dx.doi.org/10.21236/ada450337>.

BARANYI, R., CZECH, P., WALCHER, F., et al., 2019, "Reha@Stroke - A Mobile Application to Support People Suffering from a Stroke Through Their Rehabilitation". pp. 1–8, 08. doi: 10.1109/SeGAH.2019.8882447.

BARRETO, A. S., MURTA, L. G. P., DA ROCHA, A. R. C., 2011, "Software Process Definition: a Reuse-based Approach", *Journal of Universal Computer Science*, v. 17, n. 13, pp. 1765–1799.

BECK, K., BEEDLE, M., BENNEKUM, A. V., et al., 2001. "Manifesto para Desenvolvimento Ágil de Software". Disponível em: <http://agilemanifesto.org/>.

BECK, K., 1999, "Extreme programming: a discipline of software development (invited paper) (abstract only)", *ACM SIGSOFT Software Engineering Notes*, v. 24, n. 6 (out.), pp. 1. ISSN: 0163-5948. doi: 10.1145/318774.318778. Disponível em: <http://dx.doi.org/10.1145/318774.318778>.

BENIGNO, M. V. B., TEIXEIRA, E. N., WERNER, C., 2024, "Processos de Software para Jogos Digitais: uma Revisão Terciária da Literatura". In: *Anais do XXXV Simpósio Brasileiro de Informática na Educação*

(SBIE 2024), SBIE 2024, p. 665–680. Sociedade Brasileira de Computação - SBC, nov. doi: 10.5753/sbie.2024.242237. Disponível em: <http://dx.doi.org/10.5753/sbie.2024.242237>.

BEYNON-DAVIES, P., CARNE, C., MACKAY, H., et al., 1999, "Rapid application development (RAD): an empirical review", *European Journal of Information Systems*, v. 8, n. 3, pp. 211–223. ISSN: 1476-9344. doi: 10.1057/palgrave.ejis.3000325. Disponível em: <http://dx.doi.org/10.1057/palgrave.ejis.3000325>.

BOSCH, J., FLORIJN, G., GREEFHORST, D., et al., 2002, "Variability Issues in Software Product Lines". In: *Software Product-Family Engineering*, p. 13–21, Springer Berlin Heidelberg. ISBN: 9783540478331. doi: 10.1007/3-540-47833-7_3. Disponível em: <http://dx.doi.org/10.1007/3-540-47833-7_3>.

BREIEN, F., WASSON, B., 2022, "eLuna: A Co-Design Framework for Narrative Digital Game-Based Learning that Support STEAM", *Frontiers in Education*, v. 6 (01). doi: 10.3389/feduc.2021.775746.

BREUER, J. S., BENTE, G., 2010, "Why so serious? On the relation of serious games and learning", *Eludamos: Journal for Computer Game Culture*, v. 4, n. 1 (abr.), pp. 7–24. ISSN: 1866-6124. doi: 10.7557/23.6111. Disponível em: <http://dx.doi.org/10.7557/23.6111>.

CAILLOIS, R., 2001, *Man, Play and Games*. Urbana and Chicago, University of Illinois Press. ISBN: 9780252070334. ASIN: B00FBT0F6E.

CANO, S., MUÑOZ-ARTEAGA, J., COLLAZOS, C., et al., 2016, "Towards a Methodology for Serious Games Design for Children with Auditory Impairments", *IEEE Latin America Transactions*, v. 14 (05). doi: 10.1109/TLA.2016.7530453.

CARRIÓN-TORO, M., SANTÓRUM G., M., ACOSTA-VARGAS, P., et al., 2020, "iPlus a User-Centered Methodology for Serious Games Design", *Applied Sciences*, v. 10 (12), pp. 9007. doi: 10.3390/app10249007.

CHUECA, J., VERÓN, J., FONT, J., et al., 2024, "The consolidation of game software engineering: A systematic literature review of software engineering for industry-scale computer games", *Information and Software Technology*, v. 165 (jan.), pp. 107330. ISSN: 0950-5849. doi: 10.1016/j.infsof.2023.107330. Disponível em: <http://dx.doi.org/10.1016/j.infsof.2023.107330>.

CLASSE, T. M. D., ARAUJO, R. M. D., XEXÉO, G. B., 2017, "Desaparecidos RJ – Um Jogo Digital para o Entendimento de Processos de Prestação de Serviços Públicos". In: *Anais do XVI Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pp. 772–781, Curitiba. SBC.

CLEMENTS, P., NORTHROP, L., 2002, *Software Product Lines: Practices and Patterns*. SEI Series in Software Engineering. Addison-Wesley. ISBN: 9780201703320.

COSTA, D. M., TEIXEIRA, E. N., WERNER, C. M. L., 2018, "Software Process Definition using Process Lines: A Systematic Literature Review". In: *2018 XLIV Latin American Computer Conference (CLEI)*, p. 110–119. IEEE, out. doi: 10.1109/clei.2018.00022. Disponível em: <http://dx.doi.org/10.1109/clei.2018.00022>.

COWLEY, B., 2014, "The QUARTIC process model for developing serious games: 'green my place' case study". pp. 143–172, 03. ISBN: 978-1-4939-0964-3. doi: 10.1007/978-1-4939-0965-0_8.

DE CARVALHO, D. D., CHAGAS, L. F., LIMA, A. M., et al., 2014, "Software Process Lines: A Systematic Literature Review". In: *Software Process Improvement and Capability Determination*, p. 118–130, Springer International Publishing. ISBN: 9783319130361. doi: 10.1007/978-3-319-13036-1_11. Disponível em: <http://dx.doi.org/10.1007/978-3-319-13036-1_11>.

DE GLORIA, A., BELLOTTI, F., BERTA, R., 2014, "Serious Games for education and training", *International Journal of Serious Games*, v. 1, n. 1 (fev.). ISSN: 2384-8766. doi: 10.17083/ijsg.v1i1.11. Disponível em: <http://dx.doi.org/10.17083/ijsg.v1i1.11>.

DINGSØYR, T., NERUR, S., BALIJEPALLY, V., et al., 2012, "A decade of agile methodologies: Towards explaining agile software development", *Journal of Systems and Software*, v. 85, n. 6, pp. 1213–1221. ISSN: 0164-1212. doi: 10.1016/j.jss.2012.02.033. Disponível em: <http://dx.doi.org/10.1016/j.jss.2012.02.033>.

FIELD, A., 2009, *Discovering Statistics Using SPSS*. SAGE Publications Ltd. ISBN: 9781847879073.

FRAKES, W., KANG, K., 2005, "Software reuse research: status and future", *IEEE Transactions on Software Engineering*, v. 31, n. 7, pp. 529–536.

ISSN: 0098-5589. doi: 10.1109/tse.2005.85. Disponível em: <http://dx.doi.org/10.1109/tse.2005.85>.

FRAKES, W., TERRY, C., 1996, "Software reuse: metrics and models", *ACM Computing Surveys*, v. 28, n. 2, pp. 415–435. ISSN: 1557-7341. doi: 10.1145/234528.234531. Disponível em: <http://dx.doi.org/10.1145/234528.234531>.

FREITAS, H., OLIVEIRA, M., SACCOL, A. Z., et al., 2000, "O método de pesquisa survey", *Revista de Administração*, v. 35, n. 3, pp. 105–112.

FUGGETTA, A., 2000, "Software process: a roadmap". In: *Proceedings of the Conference on The Future of Software Engineering*, ICSE00, p. 25–34. ACM, maio. doi: 10.1145/336512.336521. Disponível em: <http://dx.doi.org/10.1145/336512.336521>.

FUGGETTA, A., DI NITTO, E., 2014, "Software process". In: *Future of Software Engineering Proceedings*, ICSE '14, p. 1–12. ACM, maio. doi: 10.1145/2593882.2593883. Disponível em: <http://dx.doi.org/10.1145/2593882.2593883>.

GIL, A. C., 2002, *Como elaborar projetos de pesquisa*. 4 ed. São Paulo, Atlas.

HEVNER, A. R., MARCH, S. T., PARK, J., et al., 2004, "Design Science in Information Systems Research1", *MIS Quarterly*, v. 28, n. 1 (mar.), pp. 75–106. ISSN: 2162-9730. doi: 10.2307/25148625. Disponível em: <http://dx.doi.org/10.2307/25148625>.

HUIZINGA, J., 1955, *Homo Ludens: A Study of the Play-Element in Culture*. N. v. 911;v. 1949, Beacon paperbacks ; 15 : Sociology. Beacon Press. ISBN: 9780807046814. Disponível em: <https://books.google.com.br/books?id=5QlmAAAAMAAJ>.

J., J. E., LINSTONE, H. A., TUROFF, M., 1976, "The Delphi Method: Techniques and Applications", *Technometrics*, v. 18, n. 3 (ago.), pp. 363. ISSN: 0040-1706. doi: 10.2307/1268751. Disponível em: <http://dx.doi.org/10.2307/1268751>.

KIM, Y., STOHR, E. A., 1998, "Software Reuse: Survey and Research Directions", *Journal of Management Information Systems*, v. 14, n. 4 (mar.), pp. 113–147. ISSN: 1557-928X. doi: 10.1080/07421222.1998.11518188. Disponível em: <http://dx.doi.org/10.1080/07421222.1998.11518188>.

KRUEGER, C. W., 1992, "Software reuse", *ACM Computing Surveys*, v. 24, n. 2, pp. 131–183. ISSN: 1557-7341. doi: 10.1145/130844.130856. Disponível em: <http://dx.doi.org/10.1145/130844.130856>.

KRUEGER, C., 2002, "Easing the Transition to Software Mass Customization". In: van der Linden, F. (Ed.), *Software Product-Family Engineering*, pp. 282–293, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN: 978-3-540-47833-1.

KUHRMANN, M., FERNÁNDEZ, D. M., TERNITÉ, T., 2014, "Realizing software process lines: insights and experiences". In: *Proceedings of the 2014 International Conference on Software and System Process*, ICSSP '14, p. 99–108. ACM, maio. doi: 10.1145/2600821.2600833. Disponível em: <http://dx.doi.org/10.1145/2600821.2600833>.

LAAMARTI, F., EID, M., EL SADDIK, A., 2014, "An Overview of Serious Games", *International Journal of Computer Games Technology*, v. 2014, pp. 1–15. ISSN: 1687-7055. doi: 10.1155/2014/358152. Disponível em: <http://dx.doi.org/10.1155/2014/358152>.

LANDERS, R. N., CALLAN, R. C., 2011, "Casual Social Games as Serious Games: The Psychology of Gamification in Undergraduate Education and Employee Training". In: *Serious Games and Edutainment Applications*, p. 399–423, Springer London. ISBN: 9781447121619. doi: 10.1007/978-1-4471-2161-9_20. Disponível em: <http://dx.doi.org/10.1007/978-1-4471-2161-9_20>.

LIKERT, R., 1932, "A technique for the measurement of attitudes", *Archives of Psychology*, , n. 140, pp. 5–55.

MARTIN, J., 1991, *Rapid Application Development*. The James Martin productivity series. Macmillan Publishing Company. ISBN: 9780023767753. Disponível em: <https://books.google.com.br/books?id=o6FQAAAAMAAJ>.

MATHARU, G. S., MISHRA, A., SINGH, H., et al., 2015, "Empirical Study of Agile Software Development Methodologies: A Comparative Analysis", *ACM SIGSOFT Software Engineering Notes*, v. 40, n. 1 (fev.), pp. 1–6. ISSN: 0163-5948. doi: 10.1145/2693208.2693233. Disponível em: <http://dx.doi.org/10.1145/2693208.2693233>.

MORETTIN, P. A., BUSSAB, W. D. O., 2010, *Estatística Básica*. 6 ed. São Paulo, Saraiva. ISBN: 978-85-02-08177-2.

MOUAHEB, H., FAHLI, A., MOUSSETAD, M., et al., 2012, "The Serious Game: What Educational Benefits?" *Procedia - Social and Behavioral Sciences*, v. 46, pp. 5502–5508. ISSN: 1877-0428. doi: 10.1016/j.sbspro.2012.06.465. Disponível em: <http://dx.doi.org/10.1016/j.sbspro.2012.06.465>.

MURPHY-HILL, E., ZIMMERMANN, T., NAGAPPAN, N., 2014, "Cowboys, ankle sprains, and keepers of quality: how is video game development different from software development?" In: *Proceedings of the 36th International Conference on Software Engineering*, ICSE '14, p. 1–11. ACM, maio. doi: 10.1145/2568225.2568226. Disponível em: <http://dx.doi.org/10.1145/2568225.2568226>.

OBJECT MANAGEMENT GROUP, 2008, *Software & Systems Process Engineering Metamodel Specification, Version 2.0*. Formal Specification formal/08-04-01, Object Management Group, April. Disponível em: <https://www.omg.org/spec/SPEM/2.0/PDF>.

OLIVEIRA, P., FONTOURA, L., MEDINA, R., 2020, "Metodologias Usadas no Desenvolvimento de Jogos Eletrônicos Educacionais: Uma Revisão da Literatura". In: *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pp. 542–551, Porto Alegre, RS, Brasil. SBC. doi: 10.5753/cbie.sbie.2020.542. Disponível em: <https://sol.sbc.org.br/index.php/sbie/article/view/12810>.

OLIVEIRA JUNIOR, E. A., PAZIN, M. G., GIMENES, I. M. S., et al., 2013, "SMartySPEM: A SPEM-Based Approach for Variability Management in Software Process Lines". In: Heidrich, J., Oivo, M., Jedlitschka, A., et al. (Eds.), *Product-Focused Software Process Improvement*, pp. 169–183, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN: 978-3-642-39259-7.

OSTERWEIL, L. J., 1997, "Software processes are software too, revisited: an invited talk on the most influential paper of ICSE 9". In: *Proceedings of the 19th international conference on Software engineering - ICSE '97*, ICSE '97, p. 540–548. ACM Press. doi: 10.1145/253228.253440. Disponível em: <http://dx.doi.org/10.1145/253228.253440>.

O'HAGAN, A. O., COLEMAN, G., O'CONNOR, R. V., 2014, "Software Development Processes for Games: A Systematic Literature Review". In: *Systems, Software and Services Process Improvement*, p. 182–193, Springer Berlin Heidelberg. ISBN: 9783662438961. doi: 10.1007/978-3-662-43896-1_16.

Disponível em: <http://dx.doi.org/10.1007/978-3-662-43896-1_16>.

PAZIN, M. G., ALLIAN, A. P., OLIVEIRAJR, E., 2018, "Empirical study on software process variability modelling with SMartySPEM and vSPEM", *IET Software*, v. 12, n. 6, pp. 536–546. doi: https://doi.org/10.1049/iet-sen.2017.0061. Disponível em: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-sen.2017.0061>.

PIROVANO, M., SURER, E., MAINETTI, R., et al., 2016, "Exergaming and rehabilitation: A methodology for the design of effective and safe therapeutic exergames", *Entertainment Computing*, (05). doi: 10.1016/j.entcom.2015.10.002.

PRIETO, R., LÓPEZ-ARCOS, J., MEDINA-MEDINA, N., et al., 2017, "Design methodology for educational games based on graphical notations: Designing Urano", *Entertainment Computing*, v. 18 (01). doi: 10.1016/j.entcom.2016.08.005.

ROMBACH, D., 2006, "Integrated Software Process and Product Lines". In: *Unifying the Software Process Spectrum*, p. 83–90, Springer Berlin Heidelberg. ISBN: 9783540324508. doi: 10.1007/11608035_9. Disponível em: <http://dx.doi.org/10.1007/11608035_9>.

ROYCE, W. W., 2021, "Managing the Development of Large Software Systems (1970)". In: *Ideas That Created the Future*, p. 321–332, The MIT Press, fev. ISBN: 9780262363174. doi: 10.7551/mitpress/12274.003.0035. Disponível em: <http://dx.doi.org/10.7551/mitpress/12274.003.0035>.

SALAZAR, M. G., MITRE, H. A., OLALDE, C. L., et al., 2012, "Proposal of Game Design Document from software engineering requirements perspective". In: *2012 17th International Conference on Computer Games (CGAMES)*, p. 81–85. IEEE. doi: 10.1109/cgames.2012.6314556. Disponível em: <http://dx.doi.org/10.1109/cgames.2012.6314556>.

SALEN TEKINBAŞ, K., ZIMMERMAN, E., 2003, *Rules of Play: Game Design Fundamentals*. Cambridge, The MIT Press. ISBN: 9780262240451.

SCHELL, J., 2014, *The Art of Game Design: A Book of Lenses*. CRC Press. ISBN: 9781466598645.

SCHWABER, K., 2004, *Agile Project Management with Scrum*. Redmond, Microsoft Press. ISBN: 9780735619937.

SILVA, F. A., DA, P., MOTA, P., et al., 2011, "Linhas de Produtos de Software: Uma tendência da indústria". pp. 7–31, Sociedade Brasileira de Computação, 11. ISBN: 978-85-7669-258-4.

SIMMONDS, J., BASTARRICA, M. C., 2011, *Modeling Variability in Software Process Lines*. Technical report, Universidad de Chile, Departamento de Ciencias de la Computación, Santiago, Chile.

SOMMERVILLE, I., 1996, "Software process models", *ACM Computing Surveys*, v. 28, n. 1 (mar.), pp. 269–271. ISSN: 1557-7341. doi: 10.1145/234313.234420. Disponível em: <http://dx.doi.org/10.1145/234313.234420>.

STANDISH, T. A., 1984, "An Essay on Software Reuse", *IEEE Transactions on Software Engineering*, v. SE-10, n. 5, pp. 494–497. ISSN: 0098-5589. doi: 10.1109/tse.1984.5010272. Disponível em: <http://dx.doi.org/10.1109/tse.1984.5010272>.

STENROS, J., 2016, "The Game Definition Game: A Review", *Games and Culture*, v. 12, n. 6, pp. 499–520. ISSN: 1555-4139. doi: 10.1177/1555412016655679. Disponível em: <http://dx.doi.org/10.1177/1555412016655679>.

TAVINOR, G., 2008, "Definition of Videogames", *Contemporary Aesthetics*, v. 6 (01).

TEIXEIRA, E. N., 2011, *OdysseyProcess-FEX: Uma Abordagem para Modelagem de Variabilidades de Linha de Processos de Software*. Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro.

TEIXEIRA, E. N., 2016, *OdysseyProcessReuse: Uma Metodologia para Engenharia de Linha de Processos de Software Baseada em Componentes*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro.

TEIXEIRA, E. N., ALEIXO, F. A., AMÂNCIO, F. D. D. S., et al., 2019, "Software process line as an approach to support software process reuse: A systematic literature review", *Information and Software Technology*, v. 116 (dez.), pp. 106175. ISSN: 0950-5849. doi: 10.1016/j.infsof.2019.08.007. Disponível em: <http://dx.doi.org/10.1016/j.infsof.2019.08.007>.

VASCONCELOS, A., WERNER, C., 2011, "Evaluating reuse and program understanding in ArchMine architecture recovery approach", *Information Sciences*, v. 181, n. 13, pp. 2761–2786. ISSN: 0020-0255. doi: 10.1016/j.

ins.2010.05.024. Disponível em: <http://dx.doi.org/10.1016/j.ins.2010.05.024>.

WASHIZAKI, H., 2006, "Building Software Process Line Architectures from Bottom Up". In: *Product-Focused Software Process Improvement*, p. 415–421, Springer Berlin Heidelberg. ISBN: 9783540346838. doi: 10.1007/11767718_37. Disponível em: <http://dx.doi.org/10.1007/11767718_37>.

WHITSON, J. R., 2019, "The New Spirit of Capitalism in the Game Industry", *Television amp; New Media*, v. 20, n. 8 (maio), pp. 789–801. ISSN: 1552-8316. doi: 10.1177/1527476419851086. Disponível em: <http://dx.doi.org/10.1177/1527476419851086>.

WILKINSON, P., 2016, "A Brief History of Serious Games". In: *Entertainment Computing and Serious Games*, p. 17–41, Springer International Publishing. ISBN: 9783319461526. doi: 10.1007/978-3-319-46152-6_2. Disponível em: <http://dx.doi.org/10.1007/978-3-319-46152-6_2>.

XEXÉO, G., MANGELI, E., SILVA, F., et al., 2021, "Games as Information Systems". In: *XVII Brazilian Symposium on Information Systems*, SBSI 2021, p. 1–8. ACM. doi: 10.1145/3466933.3466961. Disponível em: <http://dx.doi.org/10.1145/3466933.3466961>.

Apêndice A

Instrumento de Avaliação da LPrS-JS

Este questionário tem por objetivo avaliar a qualidade e a aplicabilidade da LPrS-JS, principal artefato construído nesta pesquisa. O instrumento está organizado em cinco seções: caracterização do respondente, clareza e completude (QP1), utilidade prática e produtividade percebida (QP2), viabilidade de configuração e derivação (QP3), e avaliação global. Na Tabela A.1 são apresentados os campos do formulário disponibilizado aos participantes, onde a escala definida para as respostas do tipo *Likert* (LIKERT, 1932) foi 1, discordo totalmente, 2, discordo, 3, neutro, 4, concordo e 5, concordo totalmente. ncordo e 5, concordo totalmente.

Tabela A.1: Questionário de avaliação da LPrS-JS com especialistas.

Seção	**Pergunta**	**Tipo / Opções de Resposta**
1 - Caracterização do Respondente	Qual é o seu nível de formação acadêmica?	Graduação / Especialização / Mestrado / Doutorado / Pós-doutorado
	Qual é a sua área de formação principal?	Ciência da Computação / Engenharia de Software / Educação / Saúde / Design / Outra (especificar)
	Há quantos anos você atua com desenvolvimento de software?	Menos de 2 anos / 2 a 5 anos / 5 a 10 anos / Mais de 10 anos
	Há quantos anos você atua com jogos digitais?	Menos de 2 anos / 2 a 5 anos / 5 a 10 anos / Mais de 10 anos

Continua na próxima página

Tabela A.1 – continuação da página anterior

Seção	**Pergunta**	**Tipo / Opções de Resposta**
	Você já participou do desenvolvimento de algum jogo sério digital?	Sim, como membro técnico / Sim, como especialista de domínio / Sim, em papel de gestão ou pesquisa / Não
	Quantos projetos de jogos sérios você já esteve envolvido?	1 / 2 a 3 / 4 a 6 / Mais de 6 / Não se aplica
	Você já utilizou algum modelo de processo de desenvolvimento de jogos sérios?	Sim / Não
	Você já definiu algum modelo de processo de desenvolvimento de jogos sérios?	Sim / Não
	Você tem familiaridade com o conceito de Linha de Produtos ou Linha de Processos de Software?	Nenhuma familiaridade / Conheço o conceito teoricamente / Já trabalhei com abordagens similares / Tenho experiência consolidada na área
	Você tem familiaridade com a metodologia OdysseyProcessReuse?	Nenhuma / Já ouvi falar / Conheço e já li sobre / Já trabalhei com ela
2 - Clareza e Completude da LPrS-JS (QP1)	As macrofases definidas na LPrS-JS representam de forma clara as grandes etapas do desenvolvimento de um jogo sério.	Likert 1-5
	As definições das macrofases são suficientemente precisas para distinguir uma etapa da outra.	Likert 1-5
	Os papéis canônicos definidos na linha cobrem os principais envolvidos no desenvolvimento de um jogo sério.	Likert 1-5
	Os artefatos canônicos representam adequadamente os produtos de trabalho gerados ao longo do processo.	Likert 1-5

Continua na próxima página

Tabela A.1 – continuação da página anterior

Seção	Pergunta	Tipo / Opções de Resposta
	As subatividades canônicas descrevem com clareza o trabalho realizado em cada macrofase.	Likert 1-5
	As variantes das subatividades definidas descrevem com clareza as alterantivas ou possibilidades de execução da subatividade proposta?	Likert 1-5
	A classificação dos elementos em mandatórios e opcionais reflete adequadamente o que é essencial em um processo de desenvolvimento de jogo sério.	Likert 1-5
	O modelo de características é compreensível sem necessidade de treinamento especializado extenso.	Likert 1-5
	Na sua opinião, há algum elemento relevante ao desenvolvimento de jogos sérios que está ausente na linha?	Resposta aberta
	Há algum elemento (macrofase | atividade | artefato | papel) presente na linha que você considera desnecessário ou inadequado para o domínio?	Resposta aberta
3 - Utilidade Prática e Produtividade Percebida (QP2)	A LPrS-JS seria útil como ponto de partida para a definição de um processo em um novo projeto de jogo sério.	Likert 1-5
	O uso da linha reduziria o esforço necessário para estruturar um processo em comparação com a definição do zero.	Likert 1-5
	A linha apoiaria a comunicação entre membros de equipes multidisciplinares (programadores, designers, educadores, terapeutas).	Likert 1-5
	A linha contribuiria para reduzir erros ou inconsistências recorrentes no desenvolvimento de jogos sérios.	Likert 1-5

Continua na próxima página

Tabela A.1 – continuação da página anterior

Seção	Pergunta	Tipo / Opções de Resposta
	A linha contribuiria para reduzir o tempo da definição de processos para o desenvolvimento de jogos sérios.	Likert 1-5
	Você utilizaria ou recomendaria o uso desta linha em um projeto real de desenvolvimento de jogo sério.	Likert 1-5
	A linha contribuiria para a melhoria da qualidade dos jogos sérios digitais desenvolvidos?	Likert 1-5
	Profissionais iniciantes teriam facilidade para utilizar a linha de processos proposta.	Likert 1-5
	Em qual fase ou situação de um projeto você enxerga maior potencial de uso da LPrS-JS?	Resposta aberta
	Que limitações você identificaria no uso prático da linha em um contexto real de desenvolvimento?	Resposta aberta
4 - Viabilidade de Configuração e Derivação (QP3)	As regras de derivação da linha (escolha de variantes, elementos mandatórios e opcionais) são compreensíveis.	Likert 1-5
	Seria possível derivar um novo processo a partir da linha sem auxílio direto de seus criadores.	Likert 1-5
	O esforço necessário para derivar um processo a partir da linha é justificado pelos benefícios obtidos.	Likert 1-5
	O conjunto de variabilidades disponível para cada subatividade é suficiente para cobrir diferentes contextos de desenvolvimento de jogos sérios.	Likert 1-5
	A estrutura da linha permite que novos processos sejam derivados com pouco esforço?	Likert 1-5

Continua na próxima página

Tabela A.1 – continuação da página anterior

Seção	**Pergunta**	**Tipo / Opções de Resposta**
	Qual seria, na sua avaliação, o maior obstáculo para a adoção da LPrS-JS por uma equipe de desenvolvimento?	Resposta aberta
	Que recursos ou materiais de apoio tornariam a derivação de processos a partir da linha mais fácil?	Resposta aberta
5 - Avaliação Global	De forma geral, como você avalia a qualidade da LPrS-JS proposta?	Escala 1-5 (1 = Muito baixa / 3 = Adequada / 5 = Muito alta)
	Qual é o principal ponto forte da LPrS-JS, na sua opinião?	Resposta aberta
	Qual é o principal ponto a ser melhorado na LPrS-JS?	Resposta aberta
	Você tem alguma sugestão adicional para o aprimoramento da linha?	Resposta aberta

Apêndice B

Processos do Domínio de Jogos Sérios

Este apêndice apresenta uma descrição sucinta dos nove processos de desenvolvimento de Jogos Sérios (JS) que compuseram o corpo fundador da Linha de Processos de Software para Jogos Sérios (LPrS-JS), relacionados na Tabela 5.3. Para cada processo, são caracterizados sua origem e domínio de aplicação, a estrutura de fases, papéis e atividades que o constitui, a forma como articula a dupla natureza dos JS, a finalidade pedagógica, clínica ou terapêutica e os elementos lúdicos; e, quando reportados pelos autores, os estudos de caso e os procedimentos de avaliação empregados. As descrições preservam os apelidos adotados ao longo do texto e seguem a ordem da Tabela 5.3.

B.1 GAMED (ASLAN e BALCI, 2015)

O GAMED (*diGital educAtional gaMe dEvelopment methoDology*) é uma metodologia para o desenvolvimento de jogos educacionais digitais concebida como um corpo de métodos e regras embutidos em um ciclo de vida que abrange desde a concepção até a conclusão do produto. O ciclo de vida organiza-se em quatro fases: projeto do jogo, projeto do software do jogo, implementação e publicação, e aprendizagem baseada em jogo com retroalimentação; subdivididas em cerca de doze processos sequenciais, ainda que de natureza iterativa, entre os quais a formulação do problema, a geração da ideia do jogo, o desenvolvimento de requisitos, a arquitetura, o projeto de software, a programação, a integração, a publicação e a manutenção.

A metodologia distingue-se pela ênfase na garantia de qualidade conduzida de forma contínua e integrada a cada tarefa do ciclo de vida, estruturada em torno de quatro perspectivas, os chamados *quatro Ps (Product, Process, People e Project)*, e operacionalizada por um conjunto de indicadores de qualidade, como aceitabilidade,

clareza, eficácia, engajamento e usabilidade. A etapa de projeto do jogo adota uma estratégia de *design* em espiral, na qual prototipagem, *playtesting*, avaliação e análise de risco são repetidos até que o artefato atinja qualidade considerada satisfatória.

Por sua origem na Engenharia de Software, o GAMED oferece um arcabouço modular e estruturado que busca reduzir o risco de falha e orientar tanto desenvolvedores quanto gerentes de projeto. Os autores destacam que a metodologia é aplicável a projetos de diferentes portes, podendo ser adaptada e empregada também para o desenvolvimento de outros tipos de jogos além dos educacionais.

B.2 Reha (BARANYI et al., 2019)

O Reha (*Reha@Stroke*) corresponde ao processo de desenvolvimento de um jogo sério móvel voltado à reabilitação de pacientes que sofreram Acidente Vascular Cerebral (AVC). Sua principal característica é dispensar dispositivos ou sensores adicionais, valendo-se exclusivamente dos sensores integrados de um aparelho móvel, de modo a oferecer uma solução de reabilitação domiciliar de fácil acesso, complementar à terapia convencional.

O processo apoia-se em uma abordagem de *Design* Centrado no Usuário (UCD) e é conduzido em conjunto com um terapeuta. Inicia-se por uma fase de pesquisa, com *brainstorming* e revisão da literatura, da qual derivam a ideia do jogo e os requisitos iniciais; segue-se o desenvolvimento iterativo de um protótipo, refinado a partir da retroalimentação do terapeuta até que um estado satisfatório seja alcançado. O jogo organiza-se em três categorias de exercícios: movimento, toque e gesto; cada uma com níveis de dificuldade progressivos e recompensas associadas, além de uma rotina diagnóstica inicial que permite ao terapeuta calibrar a dificuldade para cada paciente.

A avaliação do protótipo foi conduzida com treze pacientes em reabilitação, que forneceram retorno qualitativo sobre aspectos como navegação, sensibilidade do controle por movimento e clareza das instruções, resultando em ajustes posteriores na aplicação. Os próprios autores reconhecem como limitações o número reduzido de participantes e a ausência de avaliação de longo prazo.

B.3 eLuna (BREIEN e WASSON, 2022)

O eLuna é um arcabouço de *co-design* para a Aprendizagem Baseada em Jogos Digitais (DGBL) de natureza narrativa, com ênfase no suporte à educação STEAM. Fundamenta-se nas categorias do *extended Ludo Narrative Variable Model* identificadas como associadas a efeitos positivos sobre engajamento, motivação e aprendizagem; mundos do tipo *hubshaped quest landscape*, objetos modificáveis, agentes

consistentes e fundamentados, e eventos do tipo *interchangeable fixed kernels*.

O arcabouço compõe-se de dois elementos. O primeiro é um método de *co-design* em quatro fases: uma fase preparatória, conduzida isoladamente pelos educadores, na qual se definem objetivos curriculares, de aprendizagem, o perfil do aprendiz e a situação de aprendizagem; uma fase de *co-design*, conduzida por educadores e desenvolvedores em conjunto; uma fase de co-especificação, que resulta em um *blueprint*; e uma fase de desenvolvimento, a cargo dos desenvolvedores. O segundo elemento é uma linguagem visual, baseada em ícones, cores e descrições textuais, empregada de forma flexível na fase de *design* e de forma estrita na fase de co-especificação, organizada em torno das ontologias de agentes, mundo, objetos e eventos.

A linguagem visual atua como um objeto de fronteira que viabiliza a colaboração entre educadores e desenvolvedores, permitindo que os primeiros visualizem o conteúdo de aprendizagem de forma organizada e que os segundos interpretem o *blueprint* de maneira diretamente implementável. O uso do arcabouço é ilustrado pelos autores por meio da co-concepção de um jogo narrativo intitulado *Idun's Apples*, relacionado a metas de sustentabilidade.

B.4 Meconesis (CANO et al., 2016)

O Meconesis (*MEtodología para la CONcepción de juEgos Serios para nIñoS con discapacidad auditiva*) é uma metodologia para a concepção de jogos sérios voltados a crianças com deficiência auditiva, desenvolvida sob a filosofia de *Design* Centrado no Usuário (ISO 13407) e com influência da Interação Humano-Computador. A metodologia estrutura-se em quatro fases: análise, pré-produção, produção e pós-produção; apoiando-se no Processo Unificado de desenvolvimento de software e em notações como CTT, para a modelagem de tarefas; UML, para a modelagem de classes; IMS-LD, para a descrição de cenários pedagógicos; e BPMN, para a descrição de processos.

A metodologia articula-se em torno de seis modelos: de análise, de usuário, de objetivos pedagógicos, de tarefas, de cenários e de validação. O modelo de análise, com enfoque na experiência do usuário, identifica aspectos de *design*, jogo, educação e desenvolvimento; o modelo de objetivos pedagógicos, baseado em competências, envolve diretamente os docentes na definição das metas a alcançar. Para articular as dimensões pedagógica e lúdica, a metodologia propõe estratégias de comunicação que associam mecânicas de aprendizagem a mecânicas de jogo.

A fase de pós-produção contempla um modelo de validação composto por trinta e seis heurísticas agrupadas em quatro facetas, derivado de trabalhos anteriores sobre facetas de jogos sérios. A metodologia é validada por meio de um estudo de caso

único com o jogo ABC-Español, conduzido junto a crianças com deficiência auditiva e docentes do programa USAER, em Aguascalientes (México).

B.5 Urano (PRIETO et al., 2017)

O Urano corresponde a uma metodologia de *design* de jogos educacionais baseada em notações gráficas, que adota a narrativa como núcleo organizador do jogo, estruturada em capítulos e cenas aos quais os demais elementos (cenários, personagens, desafios lúdicos e educacionais) são progressivamente agregados. As notações gráficas, complementadas por fichas descritivas, têm como finalidade facilitar a comunicação no interior de equipes multidisciplinares e podem ser interpretadas diretamente por desenvolvedores não envolvidos na concepção, compondo, em conjunto, o *Game Design Document* (GDD).

A metodologia compreende três pré-fases: *design* dos desafios educacionais (competências e objetivos definidos pela equipe de educadores), *design* do tipo de jogo e *design* inicial da história e dos personagens principais; seguidas de seis fases iterativas e interrelacionadas: *design* de capítulos; *design* de cenas (incluindo cenário, personagens, diálogos e desafios); *design* educacional (identificação e rotulação dos desafios e avaliação); *design* emocional (rotulação das emoções pretendidas); *design* da adaptação; e *design* da colaboração. As fases de terceira a sexta são executadas para cada cena.

A proposta combina uma estratégia inicial *top-down*, na qual se estabelece a concepção geral do jogo, com uma estratégia *bottom-up* subsequente, em que cada detalhe da história interativa é especificado. A metodologia incorpora, ainda, técnicas de avaliação da própria história, tanto pela análise de sua estrutura narrativa, sob a perspectiva de cânones clássicos como a Jornada do Herói, quanto pela análise da intensidade emocional, por meio de curvas de interesse. Sua aplicação é ilustrada pelo jogo *Urano: Invasión de los ladrones de planetas*, uma aventura bidimensional voltada à prática da compreensão leitora.

B.6 PROGame (ALCOVER et al., 2018)

O PROGame é um *framework* de processo para o desenvolvimento de jogos sérios voltados à terapia de reabilitação motora. Sua concepção apoia-se em dois fundamentos: as semelhanças entre o desenvolvimento de jogos sérios e o de aplicações *web*, ambos distintos dos sistemas de software convencionais; e a analogia entre o desenvolvimento de um jogo sério para reabilitação e um ensaio clínico para novos medicamentos. Desse modo, o *framework* combina três abordagens, o Scrum, para a

gestão ágil do projeto; um modelo de desenvolvimento de aplicações web; e as fases de um ensaio clínico.

O processo organiza-se em um fluxo iterativo de duas dimensões. A primeira compreende uma atividade de iniciação do projeto, um fluxo iterativo de quatro atividades básicas de desenvolvimento (planejamento e controle, modelagem, construção e avaliação) e um estudo clínico final. A segunda dimensão estrutura três incrementos, derivados das fases centrais do ensaio clínico: o mecanismo de interação, os elementos de interação e o jogo sério propriamente dito. A seleção de uma terapia validada e a participação de fisioterapeutas na validação de cada incremento são tratadas como atividades centrais para assegurar a eficácia e a segurança terapêutica.

A validação do *framework* deu-se por meio do desenvolvimento de um jogo sério para a melhoria do equilíbrio e do controle postural de adultos com paralisia cerebral, conduzido em parceria com uma associação especializada. A avaliação clínica, apoiada em instrumentos padronizados como a Escala de Equilíbrio de Berg, o *Functional Reach Test* e o teste de Tinetti, indicou melhora funcional dos participantes, e o jogo resultante passou a ser explorado por organizações no âmbito de contratos de transferência tecnológica.

B.7 Exergame (PIROVANO et al., 2016)

O Exergame corresponde a uma metodologia para o *design* de *exergames* terapêuticos eficazes e seguros, fundamentada em uma definição abrangente segundo a qual um *exergame* é um exercício com um jogo construído em sua estrutura, e um *exergame* terapêutico é aquele que contempla todos os objetivos primários e secundários definidos para o exercício. A distinção entre objetivos primários (os movimentos a serem eliciados) e objetivos secundários (a forma como tais movimentos devem ser corretamente executados) é central para garantir a validade terapêutica, evitando compensações inadequadas e posturas que tornariam o exercício ineficaz ou perigoso.

A metodologia organiza-se em um procedimento de quatro etapas. A primeira é a definição do exercício, conduzida em estreita colaboração com os terapeutas, na qual se estabelecem os objetivos primários e secundários, bem como parâmetros de qualidade e de avaliação do desempenho. A segunda é a virtualização, em que os objetivos primários são implementados em um exercício virtual com requisitos de rastreamento (entrada) e de retroalimentação (saída) e a escolha dos dispositivos adequados. A terceira é o *design* do jogo, na qual se acrescentam os elementos lúdicos sem interferir na correta execução do exercício. A quarta é o tratamento dos objetivos secundários, por meio de um módulo de monitoramento que identifica movimentos incorretos em tempo real e fornece retorno imediato ao paciente.

A separação entre exercício e jogo delimita os papéis das equipes clínica e de Tecnologia da Informação e Comunicação ao longo do desenvolvimento e favorece a modularidade e o reúso, estabelecendo uma relação de muitos-para-muitos entre exercícios e *exergames*. A metodologia é ilustrada por meio de *exergames* para reabilitação de equilíbrio e postura e para reabilitação, implementados e testados com pacientes pós-AVC em treinamento domiciliar autônomo.

B.8 QUARTIC (COWLEY, 2014)

O QUARTIC (*QUality Assuring Recursive TEL Instruction co-Creation*) é um modelo de processo em espiral recursiva voltado ao desenvolvimento de jogos sérios, proposto com o objetivo de conciliar as práticas provenientes do *design* de jogos de entretenimento com as exigências pedagógicas do software educacional. O modelo originou-se da necessidade prática observada em dois projetos europeus de jogos sérios, nos quais a ausência de um processo claro para integrar a expertise heterogênea de especialistas em pedagogia e em design de jogos resultava em retrabalho. Sua premissa é a de que uma experiência de aprendizagem mediada por tecnologia pode ser construída pela correlação iterativa entre objetivos (a lição a ser aprendida ou o comportamento a ser modificado) e meios (os cenários e as mecânicas de jogo que viabilizam o ensino), conduzidos em dois fluxos paralelos: o fluxo de história (*story stream*), focado no desenvolvimento do jogo, e o fluxo de comportamento (*behaviour stream*), focado na modelagem pedagógica.

O processo organiza-se em quatro ordens de complexidade crescente, percorridas em ambos os fluxos antes do avanço à ordem seguinte. Na primeira ordem, de Requisitos, parte-se de um requisito do cliente, formulado como meta comportamental ou como fragmento de cenário, do qual se derivam o *game design brief* e o objetivo pedagógico. A segunda ordem, de Mapeamento, traduz essas descrições gerais em especificações mais precisas, produzindo o modelo de história e os cenários, de um lado, e o cronograma pedagógico e os comportamentos e competências, de outro. A terceira ordem, de Contextualização, aproxima o conteúdo sério e o lúdico, definindo o fluxo do jogo e suas tarefas e desafios, bem como os *Behavlets* e seus recursos e ações. A quarta ordem, de Modelagem, instancia as mecânicas de jogo e as mecânicas pedagógicas que conduzem à implementação do jogo.

A utilização do modelo é regida por três princípios: seguir as etapas uma de cada vez, observar os requisitos originais do cliente como pontos de verificação e concluir ambos os fluxos de uma ordem antes de avançar à ordem seguinte. O caráter recursivo manifesta-se na necessidade de retornar a uma ordem anterior e atravessar para o fluxo paralelo, de modo a alimentar a geração de conteúdo com os resultados já obtidos. O modelo dialoga com referenciais consolidados, sendo conce-

bido para se sobrepor a práticas ágeis existentes, como o Scrum, e não substituí-las. Um conceito de apoio relevante é o *Behavlet*, unidade mínima de transformação de comportamento estruturada em torno de um trio recurso-ação-resultado, complementada por modificadores de contexto, por um instrumento de consolidação e por um peso de importância.

A aplicação do QUARTIC é ilustrada pelo caso do *Green My Place*, jogo sério voltado à transformação de comportamento em eficiência energética, desenvolvido no âmbito do projeto SAVE ENERGY para cinco edifícios públicos em cinco cidades europeias. COWLEY (2014) reconhece a inviabilidade de submeter o próprio modelo a um experimento controlado, dada a quantidade de variáveis que afeta cada projeto de desenvolvimento, e opta por avaliá-lo de forma indireta, pelos jogos produzidos, observando o valor educacional a partir das visitas às páginas de aprendizagem motivadas pelo jogo e a energia economizada nos edifícios-piloto. Entre as limitações apontadas, destacam-se a baixa especificidade quanto ao encerramento e ao reinício das iterações e o risco de restringir o processo criativo do design, restrição que o autor considera, no contexto dos jogos sérios, desejável para conter o excesso de funcionalidades e assegurar um patamar mínimo de funcionalidade pedagógica.

B.9 iPlus (CARRIÓN-TORO et al., 2020)

O iPlus é uma metodologia de design de jogos sérios de caráter participativo, flexível e centrado no usuário proposta para responder à carência de métodos que incorporem, de forma equilibrada, tanto as mecânicas de entretenimento quanto o componente sério. Sua construção apoiou-se em protocolos experimentais conduzidos de modo participativo, o que permitiu validar e adaptar progressivamente suas etapas a partir do consenso entre especialistas e usuários. A metodologia estrutura-se por meio de um diagrama, organizando seus elementos em Método (a sequência de fases), Participantes, Ferramentas (modelos, teorias e técnicas), Materiais e Recursos, e Artefatos.

O processo de design desenvolve-se em cinco fases. A fase de Identificação define o problema geral e identifica os participantes envolvidos. A fase de Objetivos Pedagógicos estabelece, de forma participativa e consensual, os objetivos geral e específicos, sob a orientação de um especialista em pedagogia. A fase de Roteiro Lúdico (*Ludic Game Script*) produz o GDD, reunindo narrativa, conteúdo de aprendizagem, personagens, regras, mundos de jogo, elementos multimídia e técnicas de gamificação. A fase de *GamePlay* especifica as funções e ações que o jogador realiza, o gênero do jogo e os termos-chave que o nomeiam. Por fim, a fase de Refinamento valida os requisitos quanto a características de qualidade derivadas de normas ISO, gerando as histórias de usuário (*user stories*).

A metodologia prevê a participação ativa de um conjunto amplo de partes interessadas, como facilitador, *product owner*, usuários finais, especialistas em pedagogia, especialista no domínio, desenvolvedores de software, designers de jogos, psicólogos e jogadores; e recorre a ferramentas como a Taxonomia de Bloom, a Teoria das Inteligências Múltiplas, o *brainstorming*, o diagrama de afinidades, a gamificação e os blocos de *GamePlay*. Seus conceitos e relações são formalizados em um metamodelo, fundamentado na definição de jogo sério de Zyda, composto por história, arte e software, acrescido do componente pedagógico, ao qual o iPlus adiciona as noções de gamificação, *gameplay*, gênero e participante. Por produzir histórias de usuário como artefato final, a metodologia integra-se a abordagens ágeis de desenvolvimento, como o Scrum.

A metodologia foi ilustrada e validada por meio de diversos estudos de caso, com destaque para o *Educaplay*, jogo sério destinado a reforçar aptidões acadêmicas de estudantes que se preparam para o exame de ingresso ao ensino superior. Nesse caso, a avaliação de usabilidade foi conduzida com os questionários aplicados a jogadores e a administradores, registrando índices satisfatórios. CARRIÓN-TORO et al. (2020) reportam outras aplicações da metodologia, como jogos para o ensino de engenharia, para terapia recreativa em realidade virtual, para inclusão laboral de pessoas com deficiência intelectual e para a alfabetização digital básica de adultos.