

Construção e Avaliação de uma Linha de Processos de Software para Jogos Sérios Digitais

Marcos Vinícius Bezerra
Benigno
PESC/COPPE/UFRJ
Rio de Janeiro, Brasil
benigno@cos.ufrj.br

Eldânae Nogueira Teixeira
IC/UFRJ
Rio de Janeiro, Brasil
eldany@ic.ufrj.br

Cláudia Maria Lima Werner
PESC/COPPE/UFRJ
Rio de Janeiro, Brasil
werner@cos.ufrj.br

Resumo

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 constrói e avalia uma Linha de Processos de Software para Jogos Sérios (LPrS-JS), fundamentada na metodologia OdysseyProcessReuse e no método ArchToDSSA, como mecanismo para gerenciar a variabilidade inerente ao domínio e elevar a sistematicidade do desenvolvimento. A linha foi construída a partir de nove processos de desenvolvimento de JS identificados na literatura acadêmica e é representada, em modelos de características na notação OdysseyProcess-FEX, por seis macrofases canônicas organizadas em torno de um vocabulário que integra papéis, artefatos, subatividades e variantes, com suas obrigatoriedades e opcionalidades. A avaliação, conduzida com seis especialistas por meio de *survey*, indicou resultados positivos nas dimensões de clareza, completude e utilidade prática, com unanimidade na percepção de 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 apontam para a necessidade de expansão do corpus de processos fundadores, de modo a incorporar contextos industriais e equipes com perfis artísticos.

Palavras-chave

Jogos Sérios, Linha de Processos de Software, Reutilização de Software, Modelagem de Variabilidades, Engenharia de Domínio

1 Introdução

A indústria de jogos digitais consolidou-se como um ramo dinâmico da economia global, com taxas de crescimento que superam as das indústrias cinematográfica e musical [14]. A aplicação desses sistemas expandiu-se para além do entretenimento, alcançando relevância em áreas como defesa, saúde e educação [3, 20]. Nesse movimento, destacam-se os *Jogos Sérios* (JS), aplicações cujo propósito transcende a diversão, voltando-se para a instrução, o treinamento ou a transmissão de informação [1, 20, 33]. Nesses jogos, a complexidade é intensificada pela necessidade de mesclar atividades lúdicas a objetivos pedagógicos ou terapêuticos, sob

a coordenação de equipes multidisciplinares que reúnem programadores, artistas, *designers* e especialistas de domínio [9, 23].

Apesar dessa necessidade, a literatura revela uma carência de processos padronizados e amplamente aceitos [9, 24]. Essa falta pode resultar em desenvolvimentos *ad hoc*, dispendiosos e propensos a erros [31]. Benigno et al. [9], O'Hagan et al. [23] 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 convergência em torno de um método único. A carência de padrões faz com que o conhecimento gerado em um projeto não seja transferido de forma estruturada para o próximo, podendo resultar em perda de retorno financeiro [18], bem como de capital intelectual [19].

A reutilização de software surge, nesse cenário, como prática para promover qualidade e reduzir esforços em novos projetos [18, 19], apoiada na tese de que processos de software são, em essência, software, sendo passíveis de descrição padronizada e reutilização sistemática [25]. Uma técnica que trabalha o reúso a partir das similaridades e variabilidades dos processos é a *Linha de Processos de Software* (LPrS): um conjunto de processos de um domínio que compartilham características comuns e são construídos a partir de ativos reutilizáveis [28, 31]. Ao representar as variabilidades de um domínio de forma explícita, uma LPrS permite derivar processos customizados com menor custo e tempo [28, 31], estratégia adequada ao cenário dos JS, no qual cada aplicação demanda adaptações, considerando seus objetivos educacionais, clínicos ou terapêuticos [20].

Nesse contexto, o trabalho investiga três questões de pesquisa (QPs):

- **QP1:** a representação do domínio de JS 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 JS é percebida como útil na prática e como ganho de produtividade na estruturação de novos processos?
- **QP3:** a derivação de processos específicos a partir de uma linha de processos para JS mostra-se viável em termos de custo-benefício, considerando o esforço exigido para sua configuração e derivação?

O resultado de maior alcance a que esta pesquisa se dirige é um passo em direção a um modelo conceitual de referência do domínio de JS, representado por meio de uma LPrS para Jogos Sérios (**LPrS-JS**). A proposta é utilizar a linha como mecanismo de organização de um domínio: uma estrutura que reúne, de forma abstrata e reutilizável, os elementos comuns e variáveis dos processos, a partir da qual processos específicos podem ser derivados.

A LPrS-JS apresentada neste trabalho foi construída a partir de nove processos da literatura e representada por modelos de características que organizam macrofases, papéis, artefatos, subatividades e variantes do domínio. Metodologicamente, a linha foi construída segundo a Análise do Domínio de Processos da *OdysseyProcessReuse* [30], apoiada no raciocínio do método *ArchToDSSA* [32].

Uma avaliação empírica com especialistas do domínio foi conduzida e fornece evidências sobre sua clareza, utilidade e viabilidade de derivação. Esse procedimento assumiu a forma de um *survey* exploratório de percepção com seis especialistas do domínio, voltado a aferir a clareza, a utilidade e a viabilidade percebidas dos artefatos da linha. Os resultados obtidos indicam clareza e completude satisfatórias e utilidade prática reconhecida, com viabilidade de derivação condicionalmente positiva em termos de custo-benefício; a principal contribuição é uma representação de referência do domínio, explícita, reutilizável e rastreável até suas fontes. Reconhecem-se, desde já, como limitações a origem acadêmica do corpus fundador, o caráter exploratório do painel reduzido e a sub-representação da dimensão artística do desenvolvimento de JS, retomadas na Seção 5.

O restante do artigo está organizado como segue. A Seção 2 apresenta a fundamentação e os trabalhos relacionados; a Seção 3, a construção e a apresentação da LPrS-JS; a Seção 4, a avaliação; a Seção 5, a discussão e as limitações; e a Seção 6, as conclusões.

2 Fundamentação e Trabalhos Relacionados

2.1 Processos para Jogos Sérios Digitais

Os jogos digitais são produtos de software, mas apresentam características que exigem métodos de construção distintos dos da Engenharia de Software convencional [9]. A gestão de requisitos caracteriza-se por volatilidade acentuada, com o *design* ocorrendo frequentemente de forma simultânea à implementação, o que favorece metodologias ágeis e híbridas [22, 23]. No subdomínio dos JS, que são aplicações de jogos que possuem o fim de transmissão de conhecimentos e habilidades [20], o processo incorpora camadas adicionais de complexidade em razão do propósito pedagógico ou terapêutico explícito, que deve coexistir com o componente de entretenimento e exige a integração de teorias de aprendizagem ao fluxo de engenharia [9, 20, 24]. A ausência de um padrão único dificulta a consolidação de práticas estáveis e amplia o risco de perda de informação entre projetos [9, 22].

2.2 Reutilização e Linhas de Processo de Software

A Engenharia de Linha de Produto de Software (LPS) é uma das abordagens mais consolidadas para a reutilização sistemática, fundamentando-se na construção de produtos a partir de uma base de ativos centrais e no gerenciamento explícito de similaridades e variabilidades [5, 10, 15, 28]. A abordagem de LPrS adapta esses fundamentos ao cenário de modelagem e administração de processos [7, 29–31], organizando-se em dois ciclos interdependentes: a Engenharia de Domínio de Processos de Software (desenvolvimento *para* reutilização) e a Engenharia de Aplicação (desenvolvimento *com* reutilização) [15, 28, 29].

2.3 *OdysseyProcessReuse* e *ArchToDSSA*

A metodologia *OdysseyProcessReuse* [30] funde os princípios das LPrS ao paradigma de Desenvolvimento Baseado em Componentes de Processos, organizando o ciclo de vida em Engenharia de Domínio e Engenharia de Aplicação. A modelagem utiliza a linguagem *OdysseyProcess-FEX* [29], que provê suporte à representação explícita de variabilidades e opcionalidades, permitindo demarcar itens como mandatórios ou opcionais, pontos de variação e suas variantes, além de regras de composição e cardinalidades. A Figura 1 apresenta os principais elementos gráficos dessa notação, empregados nos modelos de características discutidos ao longo deste artigo. A escolha dessa notação justifica-se por reunir, de forma integrada, elementos de processo com opcionalidades, pontos de variação, relações de dependência e exclusão mútua e fluxos de controle, capacidades pouco cobertas por outras notações [31]. O método *ArchToDSSA* [32], por sua vez, cria arquiteturas de referência a partir da análise comparativa de artefatos de um mesmo domínio, detectando opcionalidades e variabilidades com base em critérios de equivalência; o raciocínio de identificar convergências e divergências entre artefatos de um mesmo domínio embasa a análise conduzida neste trabalho.

3 Construção e Apresentação da LPrS-JS

A construção da LPrS-JS seguiu a Análise do Domínio de Processos de Software prevista pela *OdysseyProcessReuse* [30], adotando uma abordagem híbrida (*bottom-up* e *top-down*): a linha é derivada da análise de processos previamente documentados, complementada por necessidades do domínio. O procedimento, apoiado também no raciocínio do *ArchToDSSA* [32], percorreu as seguintes etapas: (a) delimitação do escopo; (b) captura de conhecimento; (c) extração dos elementos de processo (unidades de trabalho, papéis e artefatos); (d) definição de glossários; (e) construção de um *vocabulário canônico* que agrupa termos nativos de função semelhante sob termos canônicos, nas dimensões de macrofases, subatividades, papéis e artefatos; (f) análise de mandatoriedade; (g) identificação de variabilidades; e (h) modelagem em características no ferramental *Odyssey*¹ com a notação *OdysseyProcess-FEX* [29].

3.1 Seleção do Corpus de Processos

A captura de conhecimento teve como fonte primária a revisão terciária de Benigno et al. [9], escolhida pela abrangência, pela disponibilização da lista completa de trabalhos e pela atualidade. Cabe destacar que a busca restringiu-se ao subdomínio de Jogos Sérios, e não ao desenvolvimento de jogos em geral: a descida partiu de revisões secundárias especificamente voltadas a JS até os estudos primários propositores de processos para esse subdomínio. A partir dela, procedeu-se a uma descida sistemática das revisões secundárias para os estudos primários propositores de processos, organizada em quatro estágios: dos 22 estudos secundários analisados, 11 tratavam de JS, dos quais 6 foram retidos por citarem diretamente artigos propositores. A partir dessas revisões foram identificados 70 trabalhos primários potenciais, com 67 deles disponíveis. Após

¹Ferramental *Odyssey*, do Grupo de Reutilização de Software da COPPE/UFRJ: <http://reuse.cos.ufrj.br/infoprojeto/3>.

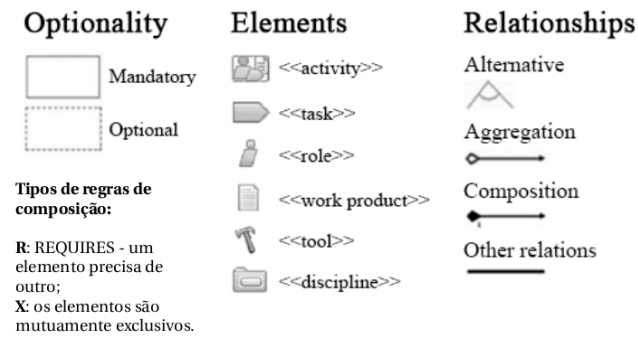


Figura 1: Principais elementos da notação OdysseyProcess-FEX [29]. Fonte: adaptado de Costa et al. [15].

um procedimento de avaliação de qualidade², 28 foram considerados aptos; e, por fim, 9 foram selecionados de forma intencional, orientados por diversidade de área de aplicação (educação formal, saúde e reabilitação física), variedade de nível de detalhamento documental e representatividade. Os processos fundadores e os apelidos atribuídos constam na Tabela 1.

Para cada um dos nove processos, foram extraídos elementos de unidades de trabalho, papel e artefato³, seguidos da construção dos glossários⁴ e do vocabulário canônico⁵. Um *termo canônico* é o rótulo padronizado que representa um conjunto de termos nativos semanticamente equivalentes, provenientes de processos distintos, unificando sob uma única denominação aquilo que cada processo nomeia à sua maneira. Esses elementos constituem a base sobre a qual se detalham, nas subseções seguintes, as macrofases, os papéis, os artefatos, as subatividades e as variabilidades da linha.

3.2 Macrofases Canônicas

As *macrofases* (MF) são grandes etapas do ciclo de vida, obtidas por agrupamento semântico das atividades e tarefas dos processos, de modo a ampliar a capacidade de generalização da linha. Sua identificação seguiu o princípio dos *pontos de inflexão* em que mudam simultaneamente o tipo de problema tratado, os artefatos gerados e os papéis predominantes. Da síntese, emergiram seis macrofases (Figura 2). A **MF1 – Problema & Objetivos** compreende a elicitação do problema, do contexto e dos *stakeholders* e a definição dos objetivos (educacionais, clínicos ou terapêuticos) e dos requisitos iniciais do jogo. A **MF2 – Design do Jogo & Prototipagem** traduz o que se deseja em como o jogo será, materializando-se em um documento de *design*, roteiro, modelos de narrativa, mecânicas e protótipos como produtos centrais. A **MF3 – Engenharia de Software** formaliza tecnicamente a solução por meio de requisitos rastreáveis, arquitetura e modelos de dados, aparecendo de forma explícita em alguns processos e implícita em outros. A **MF4 – Implementação & Publicação** reúne a codificação, a integração, o empacotamento e a publicação do jogo. A **MF5 – Uso em Campo**

Tabela 1: Processos fundadores da LPrS-JS e referências.

Apelido	Trabalho e referência
GAMED	GAMED: digital educational game development [4]
Reha	Reha@Stroke [6]
iPlus	iPlus: user-centered methodology [13]
eLuna	eLuna: co-design framework [11]
Meconesis	Serious games for auditory impairments [12]
Quartic	QUARTIC process model [16]
Urano	Designing Urano [27]
PROGame	PROGame: motor rehabilitation [2]
Exergame	Exergaming and rehabilitation [26]

& **Avaliação** corresponde ao uso do jogo por seus usuários, ao monitoramento e à análise de sua eficácia. A **MF6 – Feedback & Sustentação** consolida os resultados obtidos e planeja iterações e a manutenção do jogo. As macrofases MF3 e MF6 são opcionais; as demais, mandatórias.

3.3 Papéis, Artefatos e Subatividades Canônicos

Sobre o esqueleto das macrofases, os demais elementos tiveram seus termos nativos mapeados para termos canônicos. A dimensão de *papéis* (Figura 3) consolidou funções recorrentes como Facilitador (moderador), Decisor/Patrocinador, Especialista de Domínio (educador/terapeuta), Pesquisador/Analista, *Game Designer*, Usuários Finais/Playtesters, Desenvolvedor/Programador, Arquiteto/Designer de Software, *Release/Ops* e Manutenção/Suporte/PM. A dimensão de *artefatos* (Figura 4) reúne produtos de trabalho como a Especificação do problema/necessidades/objetivos, o GDD/Especificação de design de jogo, os Protótipos/artefatos de *gameplay*, os Requisitos detalhados, a Arquitetura & Design de software, os *Builds*/componentes executáveis, o Jogo publicado/implantado, os Resultados de avaliação e o *Backlog* de mudanças/feedback estruturado. Para as macrofases 1 e 2, foram também derivadas subatividades canônicas que sintetizam o fluxo de trabalho: por exemplo, na MF1, “Entender o problema”, “Identificar contexto e escopo”, “Identificar stakeholders”, “Descrever objetivos de aprendizagem” e “Identificar requisitos iniciais”.

3.4 Mandatoriedade e Variabilidades

A mandatoriedade de cada elemento canônico foi determinada por sua frequência de participação no conjunto de processos, adotando-se o limiar de 60%: um elemento presente em ao menos 6 dos 9 processos é classificado como mandatório; caso contrário, opcional⁶. Sobre essa base, os elementos foram organizados em pontos de variação obrigatórios (mandatórios) e opcionais. A definição de variabilidades foi conduzida para as subatividades das macrofases MF1 e MF2, em duas subfases: agrupamento (mapeamento do termo canônico de volta aos termos nativos, que representam variantes brutas) e definição de variantes (análise semântica das variantes brutas em variantes finais, com regras de composição, exclusão mútua e cardinalidades)⁷. A Figura 5 apresenta o modelo de características da MF1, com suas subatividades, variantes, cardinalidades e opcionalidades. Por exemplo, o ponto de variação “1.2 Identificar contexto

²Instrumento e resultados da avaliação de qualidade disponíveis em: <https://doi.org/10.5281/zenodo.19987818> e <https://doi.org/10.5281/zenodo.19987893>.

³Extração completa de elementos: <https://doi.org/10.5281/zenodo.19988020>.

⁴Glossários: <https://doi.org/10.5281/zenodo.19988264>.

⁵Vocabulário canônico: <https://doi.org/10.5281/zenodo.19988395>.

⁶Análise de mandatoriedade: <https://doi.org/10.5281/zenodo.20480392>.

⁷Definição de variabilidades: <https://doi.org/10.5281/zenodo.19988737>.

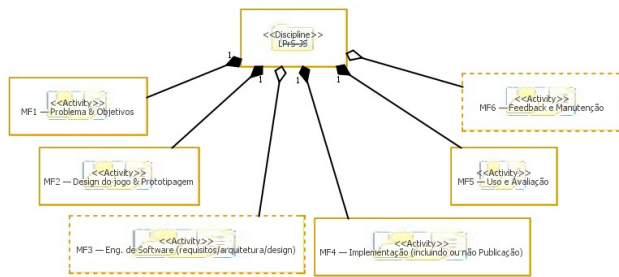


Figura 2: Modelo de características com as seis macrofases canônicas na notação OdysseyProcess-FEX [29]. Modelos em alta resolução disponíveis no material complementar.

e escopo” oferece variantes mutuamente exclusivas para contextos educacionais (“Análise Demográfica e Educacional”) e clínicos (“Delimitação de Domínio Terapêutico”), esta última exigindo a variante correspondente de objetivos terapêuticos. São essas variantes específicas, como a delimitação de domínio terapêutico e os objetivos de aprendizagem ou terapêuticos, que distinguem, sobretudo, os JS do desenvolvimento de jogos de puro entretenimento, ao acoplar o propósito pedagógico ou clínico ao fluxo de engenharia; a extensão da linha a jogos tradicionais, pela supressão dessas variantes, permanece como possibilidade a investigar. De forma análoga, a macrofase MF2 (Figura 6) organiza as subatividades de *design* e prototipagem e distingue variantes de *design* consensual e documental, modelagem formal e estruturada, e *design* evolutivo e virtualização, com regras de composição que vinculam, por exemplo, a prototipagem de componentes funcionais ao refinamento iterativo contínuo.

A LPrS-JS resultante⁸ é, assim, uma representação de referência do domínio: um conjunto de modelos de características que registra, de forma rastreável até os processos fundadores, as macrofases, os papéis, os artefatos e as subatividades do desenvolvimento de JS, com suas obrigatoriedades, opcionalidades e variantes. Cabe observar que, por essa natureza, a LPrS-JS não constitui um processo rígido a ser seguido, mas um ativo de reúso que organiza a variabilidade do domínio: dela podem ser derivados fluxos concretos — inclusive ágeis e híbridos — ajustados a cada projeto. É essa explicitação de macrofases, papéis e artefatos que favorece a rastreabilidade e o reaproveitamento de conhecimento entre projetos.

4 Avaliação com Especialistas

A avaliação teve por objetivo verificar se os artefatos que compõem a linha são compreensíveis, úteis e viáveis na perspectiva de especialistas. Trata-se, portanto, de uma avaliação da percepção dos especialistas sobre os artefatos da linha, e não de uma validação por instanciamento ou uso da linha em um projeto real de desenvolvimento — o que se reserva a trabalhos futuros (Seção 6). Trata-se de um estudo exploratório de percepção, conduzido com um painel reduzido, cujos dados quantitativos têm função descritiva e orientam a leitura das respostas discursivas. Para fundamentar as respostas em uma interação concreta, os artefatos da linha (modelos de características,

vocabulário canônico e glossário) foram disponibilizados aos participantes por meio do *Catálogo de Processos*, uma ferramenta web construída especificamente para apoiar a avaliação⁹, que reúne as modelagens em uma interface navegável e não exige familiaridade com o ferramental *Odyssey*.

O instrumento, um questionário estruturado com itens em escala Likert [21] de cinco pontos e questões abertas, foi organizado em seções vinculadas às três QPs, após uma avaliação piloto assistida. Participaram seis especialistas: todos com doutorado (metade também com pós-doutorado), com formação em Engenharia de Software (3), Design (1), Engenharia de Produção (1) e Ciência da Computação (1); cinco possuem mais de cinco anos de experiência com jogos digitais e o grau de familiaridade com LPS/LPrS varia de teórico a consolidado. A análise adotou a mediana como medida principal de síntese, por sua robustez em escalas ordinais com poucos respondentes [17], complementada por triangulação com o perfil e as respostas discursivas. A Figura 7 resume o posicionamento do painel por dimensão.

QP1 – Clareza e completude. O item de maior consenso foi a precisão das macrofases para distinguir etapas, com unanimidade na nota máxima (mediana 5,0), validando a estratégia de agrupamento que consolidou atividades dos nove processos em macrofases canônicas. O modelo de características obteve mediana 5,0, confirmando que a representação em *OdysseyProcess-FEX* [29] é acessível mesmo a especialistas sem formação prévia na metodologia. Em contraste, papéis, artefatos e subatividades apresentaram as respostas mais dispersas, refletindo a ausência de atividades ligadas à dimensão criativa do *design* de jogos, coerente com a ameaça de validade de construto associada ao corpus de origem acadêmica.

QP2 – Utilidade prática. A dimensão registrou o resultado mais consistente: o item “reduziria o esforço de estruturação” obteve unanimidade na nota máxima, e “útil como ponto de partida” e “reduziria o tempo de definição” obtiveram mediana 5,0. Esse núcleo de consenso responde afirmativamente à QP2 e alinha-se às vantagens da reutilização de processos [28, 31]. A intenção de adoção em contexto real, porém, foi moderada (mediana 3,5), sinalizando barreiras de adoção para além do valor do artefato, como o custo de aprendizagem da notação e a ausência de ferramental de derivação automatizado.

QP3 – Viabilidade de configuração e derivação. Foi a dimensão com avaliações mais cautelosas, ainda que majoritariamente acima do ponto neutro. O item sobre a facilidade para profissionais iniciantes obteve a menor mediana (3,0), indicando que a linha é mais acessível a engenheiros de processos com formação em LPrS. O item central da QP3, “o esforço de derivação é justificado pelos benefícios”, obteve mediana 4,0, o melhor resultado da dimensão; mesmo os respondentes mais críticos à facilidade de uso reconhecem que o retorno compensa o investimento.

A avaliação global da linha obteve mediana 4,0, sem notas abaixo de 3, indicando que as críticas configuram oportunidades de melhoria de um artefato cuja qualidade é reconhecida.

A triangulação com o perfil e as respostas discursivas evidenciou quatro temas: valor instrumental consensual com intenção de adoção moderada; convergência independente, por respondentes de perfis distintos, sobre a ausência de papéis artísticos e de atividades

⁸LPrS-JS completa disponível em <https://doi.org/10.5281/zenodo.21267137>

⁹Catálogo de Processos da LPrS-JS: <https://teste-blond-ten.vercel.app/catalogo>.

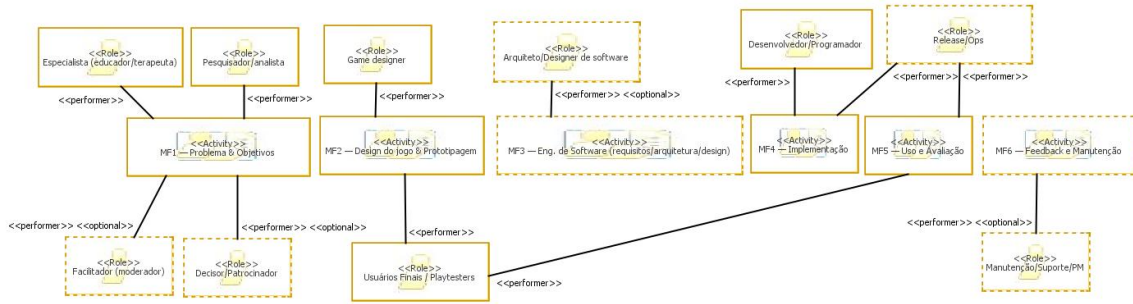


Figura 3: Dimensão de papéis do modelo de características, com suas opcionalidades e mandatoriedades. Modelos em alta resolução disponíveis no material complementar.

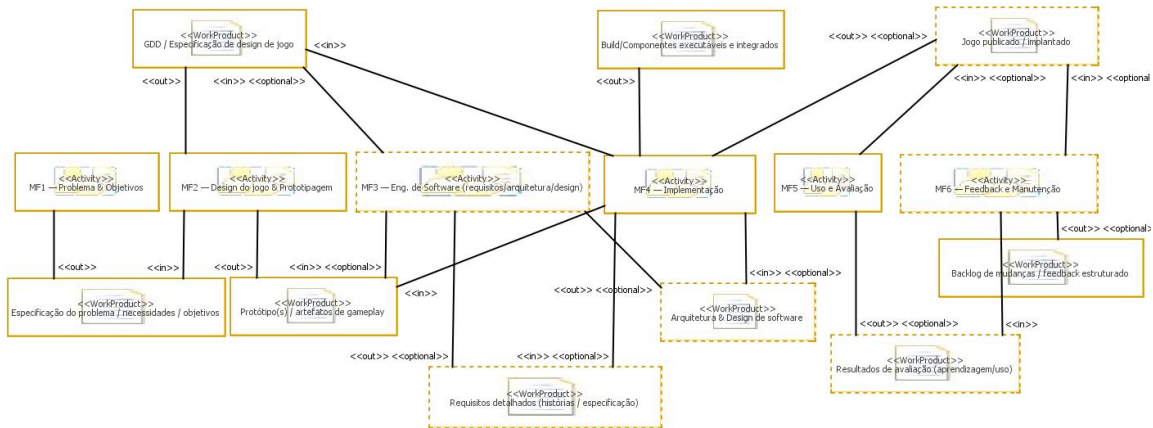


Figura 4: Dimensão de artefatos do modelo de características, com suas opcionalidades, mandatoriedades e relações de entrada e saída. Modelos em alta resolução disponíveis no material complementar.

da dimensão criativa; demanda convergente por derivação assistida; e dependência do perfil na percepção de viabilidade. Por seu caráter estrutural e recorrente, a ausência de papéis artísticos constitui o principal candidato a refinamento futuro do modelo.

5 Discussão e Limitações

Os resultados sustentam que a abordagem de linha de processos representa o domínio de JS com clareza satisfatória (QP1) e utilidade prática reconhecida (QP2), sendo a viabilidade de derivação condicionalmente positiva em termos de custo-benefício (QP3). Uma análise das três dimensões revela um padrão coerente: os itens estruturais, que dependem diretamente do modo como a linha organiza o domínio, a distinção entre macrofases e a legibilidade do modelo de características, concentraram os julgamentos mais elevados, ao passo que os itens relacionados à adoção em contexto

real e à facilidade de uso por iniciantes reuniram as respostas mais cautelosas. Essa separação sugere que a contribuição central, a representação explícita e reutilizável do domínio — foi reconhecida de forma consistente, enquanto as ressalvas concentram-se, em geral, em fatores externos ao artefato, como a curva de aprendizado da notação e a inexistência de suporte automatizado à derivação. Cabe distinguir a contribuição do artefato de suas partes mais reconhecíveis: as seis macrofases, isoladamente, aproximam-se do fluxo geral de qualquer desenvolvimento de jogo e não constituem, por si sós, a novidade da proposta. A contribuição da LPrS-JS está em explicitar, de forma sistemática e rastreável até os processos fundadores, as variabilidades, o vocabulário canônico e as obrigadoriedades e opcionalidades do domínio — precisamente aquilo que hoje permanece tácito e *ad hoc*. Distingue-se, ainda, de ativos técnicos como *game engines* e *frameworks*, que apoiam a construção do

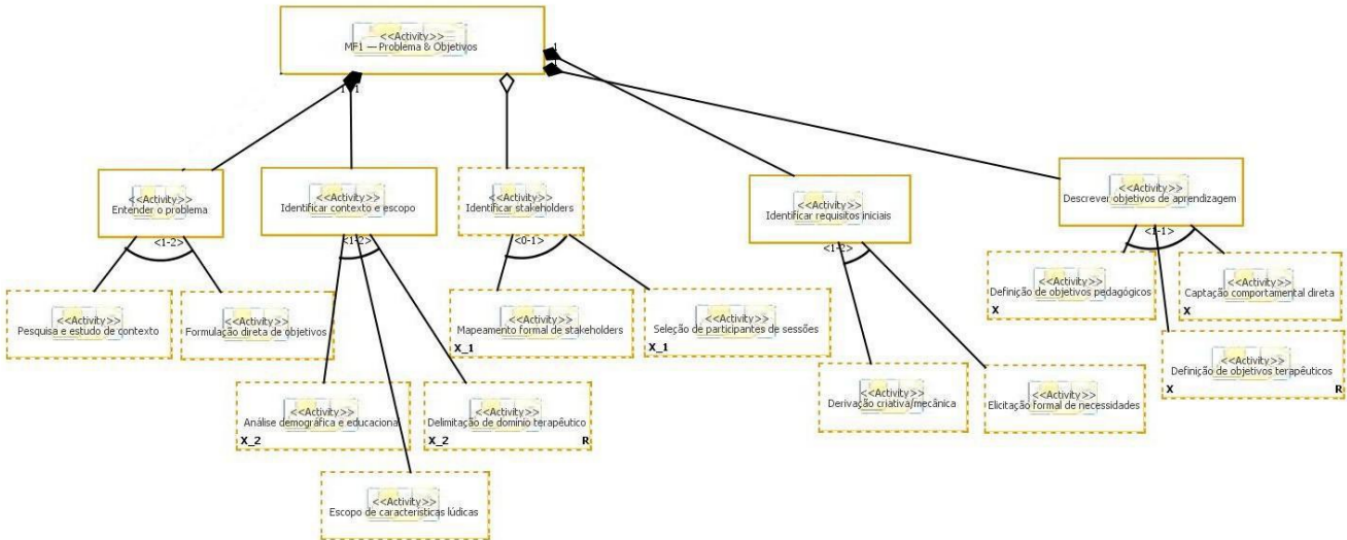


Figura 5: Modelo de características da macrofase MF1, com subatividades, variantes, cardinalidades e opcionalidades. Modelos em alta resolução disponíveis no material complementar.

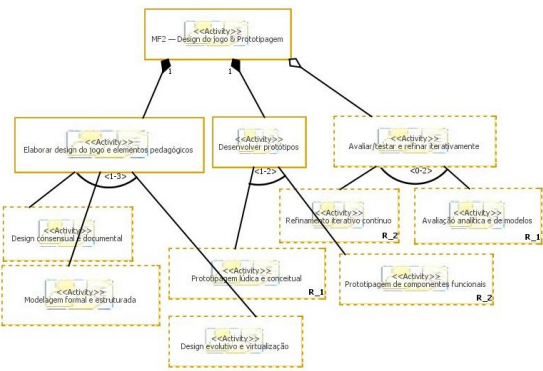


Figura 6: Modelo de características da macrofase MF2, com subatividades, variantes, cardinalidades e opcionalidades. Modelos em alta resolução disponíveis no material complementar.

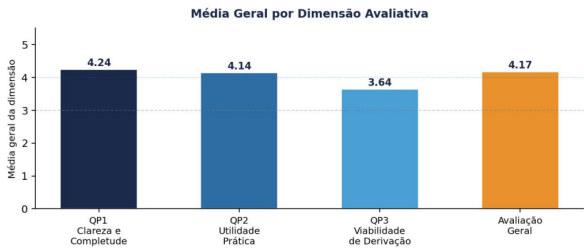


Figura 7: Média geral por dimensão avaliativa (caráter descritivo). Fonte: [8].

produto mas não modelam o processo de desenvolvimento nem a sua variabilidade.

A convergência entre o valor instrumental consensual (QP2) e a intenção de adoção apenas moderada indica que a percepção de utilidade não se traduz automaticamente em disposição de uso. Reduzir a distância entre uma e outra depende em parte da diminuição do custo de sua operação, o que reforça a prioridade atribuída, entre os trabalhos futuros, ao desenvolvimento de mecanismos de derivação assistida. Já a dependência do perfil na percepção de viabilidade, com a linha considerada mais acessível a engenheiros de processos familiarizados com LPrS, sugere que a adoção pode ser favorecida tanto por materiais de apoio e capacitação prévia quanto por interfaces que ocultem a complexidade da notação subjacente, como o próprio Catálogo de Processos utilizado na avaliação.

O achado mais consistente, por ter emergido de forma independente entre respondentes de perfis distintos, foi a ausência de papéis artísticos e de atividades ligadas à dimensão criativa do *design* de jogos. Por seu caráter estrutural, esse achado representa um indicador de cobertura, apontando que o corpus fundador, de origem acadêmica, sub-representa a face artística do desenvolvimento de JS, e delimita a principal direção de refinamento da linha. Ainda assim, a avaliação global positiva, sem notas abaixo do ponto neutro, indica que tais reservas se configuram como oportunidades de evolução do artefato.

Esses resultados devem ser lidos à perspectiva das ameaças à validade específicas da construção e da avaliação da linha. No plano da validade interna, a análise de domínio possui uma dimensão interpretativa irreduzível, o alinhamento de processos concebidos com objetivos e públicos distintos; mitigada pelo registro explícito das decisões em campos de notas ao longo da construção e pela adoção de critérios quantitativos, como o limiar de 60% para a mandatoriedade, cuja calibração, no entanto, influencia o escopo do que a linha considera comum ou variável. No plano da validade

de construto, a origem acadêmica dos nove processos fundadores é a limitação mais relevante, refletindo-se na sub-representação de equipes, exatamente o achado convergente da avaliação sobre a inclusão de profissionais de arte. No plano da validade externa, a generalização para contextos industriais — entendidos aqui como ambientes profissionais e comerciais de desenvolvimento de jogos (estúdios da indústria), e não como um setor de aplicação dos jogos — é limitada, uma vez que a linha foi construída a partir de relatos acadêmicos. Soma-se a isso o perfil predominantemente acadêmico do painel, que pode não refletir integralmente a percepção de profissionais atuantes no mercado; a avaliação com desenvolvedores da indústria é apontada como trabalho futuro. Por fim, no plano da validade de conclusão, o caráter exploratório do estudo e o número reduzido de participantes impõem que os dados quantitativos sejam tratados como estritamente descritivos, sem inferência para uma população; a ênfase na distribuição das respostas e a triangulação com o perfil e os dados qualitativos visam mitigar essa limitação.

6 Conclusão e Trabalhos Futuros

Este trabalho apresentou a construção e a avaliação de uma Linha de Processos de Software para Jogos Sérios (LPrS-JS). A linha, construída a partir de nove processos da literatura acadêmica e representada por seis macrofases canônicas organizadas em torno de um vocabulário que integra papéis, artefatos, subatividades e variantes, oferece uma representação de referência do domínio, rastreável até suas fontes. A avaliação com seis especialistas indicou clareza e completude satisfatórias, utilidade prática reconhecida, com unanimidade quanto à redução do esforço de estruturação de novos processos, e viabilidade de derivação positiva em termos de custo-benefício, ainda que condicionada a materiais de apoio e capacitação prévia. Reitera-se, contudo, o caráter exploratório da avaliação, o tamanho reduzido e o perfil predominantemente acadêmico do painel, e a origem acadêmica do corpus fundador, que recomendam a leitura das evidências como preliminares.

Como trabalhos futuros, destaca-se, com prioridade por seu caráter convergente na avaliação, a ampliação do corpus de processos originários, de modo a incorporar contextos industriais e equipes com perfis artísticos, redistribuindo as frequências de ocorrência e recalibrando os elementos canônicos, o que endereçaria a principal lacuna de cobertura identificada. Somam-se a isso o desenvolvimento de mecanismos de derivação assistida e, como extensão de mais longo prazo, a avaliação de impacto em contextos reais de desenvolvimento. A descrição detalhada do método de análise empregado na construção da linha e a sua verificação por instanciamento, complementares ao recorte aqui apresentado, são desenvolvidas na dissertação que originou este artigo [8]. Ao sistematizar o conhecimento disperso na literatura em uma infraestrutura de reutilização, este trabalho representa um passo em direção ao amadurecimento metodológico do domínio.

Disponibilidade de Artefatos

Os artefatos que sustentam a construção da linha estão publicamente disponíveis na plataforma Zenodo: extração de elementos (<https://doi.org/10.5281/zenodo.19988020>), glossários (<https://doi.org/10.5281/zenodo.19988264>), vocabulário canônico

(<https://doi.org/10.5281/zenodo.19988395>), análise de mandatoriedade (<https://doi.org/10.5281/zenodo.20480392>), definição de variabilidades (<https://doi.org/10.5281/zenodo.19988737>), além do instrumento (<https://doi.org/10.5281/zenodo.19987818>) e dos resultados (<https://doi.org/10.5281/zenodo.19987893>) da avaliação de qualidade dos processos. O Catálogo de Processos utilizado na avaliação está disponível em <https://teste-blond-ten.vercel.app/catalogo> e o ferramental Odyssey em <http://reuse.cos.ufrj.br/infoprojeto/3>. As versões em alta resolução dos modelos de características estão disponíveis no material complementar em <https://doi.org/10.5281/zenodo.21172845>. O modelo de características em um diagrama único está disponível em <https://doi.org/10.5281/zenodo.21267137>.

Agradecimentos

Em conformidade com as políticas de transparência sobre o uso de inteligência artificial, declara-se que a concepção da pesquisa, a definição do método e das decisões metodológicas, a seleção e a análise dos processos, a construção da LPrS-JS e de todos os seus artefatos, a condução e a análise da avaliação e a formulação das conclusões foram realizadas pelos autores. A ferramenta de Inteligência Artificial generativa *Claude* (modelo *Sonnet*), da *Anthropic*, foi utilizada como apoio editorial, para organização textual, em artefatos, na melhoria da estrutura da redação, no refinamento da apresentação das ideias e na revisão da clareza e da coesão.

Referências

- [1] Clark C. Abt. 1970. *Serious Games*. Viking Press, New York.
- [2] E. Alcover, A. Jaume-i Capó, and G. Moyà Alcover. 2018. PROGame: A process framework for serious game development for motor rehabilitation therapy. *PLOS ONE* 13, 5 (2018), e0197383. doi:10.1371/journal.pone.0197383
- [3] Saiqa Aleem, Luiz Fernando Capretz, and Faheem Ahmed. 2016. Game development software engineering process life cycle: a systematic review. *Journal of Software Engineering Research and Development* 4, 1 (2016). doi:10.1186/s40411-016-0032-7
- [4] Serdar Aslan and Osman Balci. 2015. GAMED: Digital educational game development methodology. *SIMULATION* 91 (2015), 307–319. doi:10.1177/0037549715572673
- [5] Felix Bachmann and Paul C. Clements. 2005. *Variability in Software Product Lines*. doi:10.21236/ada450337
- [6] R. Baranyi, P. Czech, F. Walcher, et al. 2019. Reha@Stroke – A Mobile Application to Support People Suffering from a Stroke Through Their Rehabilitation. In *2019 IEEE 7th International Conference on Serious Games and Applications for Health (SeGAH)*. IEEE, 1–8. doi:10.1109/SeGAH.2019.8882447
- [7] A. S. Barreto, L. G. P. Murta, and A. R. C. da Rocha. 2011. Software Process Definition: a Reuse-based Approach. *Journal of Universal Computer Science* 17, 13 (2011), 1765–1799.
- [8] Marcos Vinícius Bezerra Benigno. 2026. *Linha de Processos para Jogos Sérios Digitais*. Dissertação de Mestrado. Programa de Engenharia de Sistemas e Computação (PESC), COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro.
- [9] Marcos Vinícius Bezerra Benigno, Eldāne Nogueira Teixeira, and Cláudia Maria Lima Werner. 2024. Processos de Software para Jogos Digitais: uma Revisão Terciária da Literatura. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação (SBIE 2024)*. Sociedade Brasileira de Computação (SBC), 665–680. doi:10.5753/sbie.2024.242237
- [10] Jan Bosch, Gert Florijn, Danny Greefhorst, Juha Kuusela, J. Henk Obbink, and Klaus Pohl. 2002. *Variability Issues in Software Product Lines*. Springer Berlin Heidelberg, 13–21. doi:10.1007/3-540-47833-7_3
- [11] Fredrik Breien and Barbara Wasson. 2022. eLuna: A Co-Design Framework for Narrative Digital Game-Based Learning that Support STEAM. *Frontiers in Education* 6 (2022). doi:10.3389/educ.2021.775746
- [12] S. Cano, J. Muñoz Arteaga, C. Collazos, et al. 2016. Towards a Methodology for Serious Games Design for Children with Auditory Impairments. *IEEE Latin America Transactions* 14, 5 (2016). doi:10.1109/TLA.2016.7530453
- [13] M. Carrión-Toro, M. Santórum G., P. Acosta-Vargas, et al. 2020. iPlus a User-Centered Methodology for Serious Games Design. *Applied Sciences* 10, 24 (2020), 9007. doi:10.3390/app10249007

- [14] J. Chueca, J. Verón, J. Font, 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* 165 (2024), 107330. doi:10.1016/j.infsof.2023.107330
- [15] D. M. Costa, Eldānae Nogueira Teixeira, and Cláudia Maria Lima Werner. 2018. Software Process Definition using Process Lines: A Systematic Literature Review. In *2018 XLIV Latin American Computer Conference (CLEI)*. IEEE, 110–119. doi:10.1109/clei.2018.00022
- [16] Ben Cowley. 2014. The QUARTIC Process Model for Developing Serious Games: 'Green My Place' Case Study. In *Serious Games Development and Applications*. 143–172. doi:10.1007/978-1-4939-0965-0_8
- [17] Andy Field. 2009. *Discovering Statistics Using SPSS*. SAGE Publications Ltd.
- [18] William Frakes and Kyo Kang. 2005. Software reuse research: status and future. *IEEE Transactions on Software Engineering* 31, 7 (2005), 529–536. doi:10.1109/tse.2005.85
- [19] Yongbeom Kim and Edward A. Stohr. 1998. Software Reuse: Survey and Research Directions. *Journal of Management Information Systems* 14, 4 (March 1998), 113–147. doi:10.1080/07421222.1998.11518188
- [20] Fedwa Laamarti, Mohamad Eid, and Abdulmotaleb El Saddik. 2014. An Overview of Serious Games. *International Journal of Computer Games Technology* 2014 (2014), 1–15. doi:10.1155/2014/358152
- [21] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology* 140 (1932), 5–55.
- [22] Emerson Murphy-Hill, Thomas Zimmermann, and Nachiappan Nagappan. 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)*. ACM, 1–11. doi:10.1145/2568225.2568226
- [23] A. O. O'Hagan, G. Coleman, and R. V. O'Connor. 2014. Software Development Processes for Games: A Systematic Literature Review. In *Systems, Software and Services Process Improvement*. Springer Berlin Heidelberg, 182–193. doi:10.1007/978-3-662-43896-1_16
- [24] P. Oliveira, L. Fontoura, and R. Medina. 2020. Metodologias Usadas no Desenvolvimento de Jogos Eletrônicos Educacionais: Uma Revisão da Literatura. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação (SBIE)*. SBC, Porto Alegre, RS, Brasil, 542–551. doi:10.5753/cbie.sbie.2020.542
- [25] Leon J. Osterweil. 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)*. ACM Press, 540–548. doi:10.1145/253228.253440
- [26] M. Pirovano, E. Surer, R. Mainetti, et al. 2016. Exergaming and rehabilitation: A methodology for the design of effective and safe therapeutic exergames. *Entertainment Computing* (2016). doi:10.1016/j.entcom.2015.10.002
- [27] R. Prieto, J. López-Arcos, N. Medina-Medina, et al. 2017. Design methodology for educational games based on graphical notations: Designing Urano. *Entertainment Computing* 18 (2017). doi:10.1016/j.entcom.2016.08.005
- [28] Dieter Rombach. 2006. Integrated Software Process and Product Lines. In *Unifying the Software Process Spectrum*. Springer Berlin Heidelberg, 83–90. doi:10.1007/11608035_9
- [29] Eldānae Nogueira Teixeira. 2011. *OdysseyProcess-FEX: Uma Abordagem para Modelagem de Variabilidades de Linha de Processos de Software*. Master's thesis. COPPE/UFRJ, Rio de Janeiro.
- [30] Eldānae Nogueira Teixeira. 2016. *OdysseyProcessReuse: Uma Metodologia para Engenharia de Linha de Processos de Software Baseada em Componentes*. Ph. D. Dissertation. COPPE/UFRJ, Rio de Janeiro.
- [31] Eldānae Nogueira Teixeira, F. A. Aleixo, F. D. D. S. Amâncio, et al. 2019. Software process line as an approach to support software process reuse: A systematic literature review. *Information and Software Technology* 116 (2019), 106175. doi:10.1016/j.infsof.2019.08.007
- [32] Aline Vasconcelos and Cláudia Werner. 2011. Evaluating reuse and program understanding in ArchMine architecture recovery approach. *Information Sciences* 181, 13 (2011), 2761–2786. doi:10.1016/j.ins.2010.05.024
- [33] Phil Wilkinson. 2016. *A Brief History of Serious Games*. Springer International Publishing, 17–41. doi:10.1007/978-3-319-46152-6_2