

Quando o hardware ganha vida: antropomorfismo, narrativa e mecânica como estratégias de ensino em Operação Overclock

Title: When Hardware Comes to Life: Anthropomorphism, Narrative, and Mechanics as Teaching Strategies in Operation Overclock

Octacílio C. de Almeida¹, Socorro Vânia L. Alves¹, Alexandre Lopes de Sousa¹,
João Paulo N. da Costa¹

¹ Instituto de Engenharia e Geociências – Programa de Computação – Universidade Federal do Oeste do Pará (UFOPA)
Caixa Postal 168040-470 – Santarém – PA – Brasil

{octa.oa44, jncosta201}@gmail.com, socorro.alves@ufopa.edu.br,
alexandreلسousa@outlook.com

Abstract. Introduction: Computer Architecture and Organization courses present assimilation difficulties due to the invisibility of their physical processes, generating cognitive overload in purely expository teaching methods. **Objective:** This paper details the design and software architecture of the educational game Operação Overclock to make physical computer organization tangible. **Methodology or Steps:** Guided by Design Science Research (DSR) and Game-Based Learning (GBL), the artifact's development employs anthropomorphism to translate real hardware limitations — such as bus traffic and latency — into gameplay mechanics. **Results:** The project fills a gap in the literature by focusing on physical organization rather than programming logic, delivering a simulation model structurally coherent with classical theories for future evaluation.

Keywords Game-Based Learning, Design Science Research, Computer Organization, Educational Games, Anthropomorphism.

Resumo. Introdução: As disciplinas de Arquitetura e Organização de Computadores apresentam dificuldades de assimilação devido à invisibilidade de seus processos físicos, gerando sobrecarga cognitiva em métodos de ensino puramente expositivos. **Objetivo:** Este artigo detalha o projeto e a arquitetura de software do jogo educacional Operação Overclock para tangibilizar o ensino da organização física de computadores. **Metodologia ou Etapas:** Guiada por Design Science Research (DSR) e Game-Based Learning (GBL), o desenvolvimento do artefato usa o antropomorfismo para traduzir limitações reais de hardware – como tráfego de barramentos e latência – em mecânicas de jogo. **Resultados:** O projeto preenche uma lacuna na literatura ao focar na organização física em detrimento da lógica de programação, entregando um modelo de simulação estruturalmente coerente com teorias clássicas para futura avaliação.

Palavras-Chave Game-Based Learning, Design Science Research, Organização de Computadores, Jogos Educacionais, Antropomorfismo.

1. Introdução

A formação em Ciência da Computação exige que os estudantes compreendam não apenas como desenvolver *softwares*, mas também como as instruções lógicas são fisicamente executadas pela máquina. Nesse cenário, as disciplinas de Arquitetura e de Organização de Computadores desempenham um papel fundamental. No entanto, são frequentemente apontadas como áreas de maior dificuldade de aprendizado nos currículos de tecnologia, apresentando altos índices de reprovação e evasão.

Essa barreira educacional ocorre porque o funcionamento interno de um computador é, por natureza, invisível. Ao longo das décadas, os sistemas foram projetados com múltiplas camadas de abstração justamente para esconder do desenvolvedor a complexidade do *hardware*. Como consequência, quando os alunos são expostos a métodos de ensino puramente expositivos, eles enfrentam uma severa sobrecarga cognitiva. O estudante é exigido a criar modelos mentais complexos sobre o tráfego de dados, gargalos de processamento e alocação de memória sem ter nenhum referencial visual ou interativo para se apoiar.

Para mitigar esse problema, o uso de jogos digitais, embasado pelos princípios do *Game-Based Learning* (GBL), surge como uma ferramenta capaz de transformar conceitos abstratos em experiências práticas. Quando aliado a estratégias narrativas como o antropomorfismo — que atribui características, comportamentos e limitações humanas a entidades não humanas —, o *design* de um jogo pode materializar teorias complexas de *hardware* em situações familiares do cotidiano e criando a hipótese de que essa abordagem possua um forte potencial para facilitar a assimilação do conteúdo.

Nesse contexto, este artigo apresenta o projeto e a arquitetura de *software* do jogo educacional *Operação Overclock*. O artefato foi concebido com o objetivo de tangibilizar o ensino da organização física de computadores, com foco principal na arquitetura clássica de Von Neumann. Em vez de focar no ensino da lógica de programação e em códigos (camada de *software*), o jogo inova ao simular o esforço logístico e físico dos componentes internos da máquina.

Guiado pela metodologia de *Design Science Research* (DSR), este trabalho documenta o primeiro ciclo de pesquisa focado no projeto e na construção de uma Prova de Conceito (PoC). O intuito principal da presente etapa não se concentra na avaliação empírica de aprendizagem com usuários finais, mas sim em validar estruturalmente o modelo proposto. Busca-se demonstrar como as limitações reais da máquina, a exemplo do Gargalo de Von Neumann, podem ser traduzidas de forma coerente para mecânicas de jogo e obstáculos espaciais.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica, abordando a distinção entre arquitetura e organização, além dos conceitos de GBL e antropomorfismo. A Seção 3 detalha a escolha da metodologia DSR. A Seção 4 descreve o mapeamento de requisitos e a construção do artefato *Operação Overclock*. A Seção 5 apresenta a avaliação analítica da Prova de Conceito e, por fim, a Seção 6 traz as considerações finais e os direcionamentos para os próximos ciclos da pesquisa.

2. Fundamentação Teórica

2.1. Cognição, Metáforas e Antropomorfismo

A compreensão de sistemas abstratos e complexos, como a Arquitetura de Computadores, exige artifícios pedagógicos que facilitem a cognição. A Teoria da Metáfora Conceitual (TMC), apresentada por Lakoff e Johnson [Lakoff e Johnson 2002], propõe que a mente humana aprende ao mapear conceitos desconhecidos (Domínio Alvo) para experiências familiares e palpáveis (Domínio Fonte). Nesse contexto, destacam-se as metáforas ontológicas, essenciais na construção de artefatos educacionais por tratarem lógicas abstratas como entidades físicas. Quando a essas entidades virtuais são atribuídas características, emoções e personalidades, emerge o antropomorfismo.

Um exemplo notório da eficácia dessa estratégia é a animação *Divertida Mente* (Pixar), que personifica as emoções humanas (Domínio Alvo) por meio de personagens com trejeitos singulares (Domínio Fonte). Atribuir forma física a lógicas complexas facilita a compreensão do público, configurando uma tática que pode ser diretamente replicada no ensino de *hardware*.

Cognitivamente, o antropomorfismo apoia-se na Teoria da Mente (ToM), que é a capacidade inata do indivíduo de inferir e atribuir estados mentais aos outros [Premack e Woodruff 1978]. Segundo o antropólogo Stewart Guthrie [Guthrie 1993], o processo evolutivo moldou a cognição social humana de tal forma que desenvolvemos um viés de agência, buscando intenções e consciência até mesmo em seres inanimados.

O filósofo Daniel Dennett [Dennett 1987] aprofunda essa ideia por meio do conceito de Postura Intencional (*Intentional Stance*): tratar um sistema de alta complexidade como um agente com vontades próprias é a estratégia cognitiva mais rápida e econômica para o cérebro prever seu comportamento. É por essa razão que, rotineiramente, usuários inferem consciência a eletrônicos, classificando um computador lento como “preguiçoso” ou “teimoso”, em vez de analisar a falha técnica.

Em vez de lutar contra esse viés cognitivo do jogador, *Operação Overclock* o abraça como base de seu *design*. Ao antropomorfizar os componentes do computador, o jogo se apropria da inclinação natural humana de inferir estados mentais como uma ferramenta de ensino. Essa abordagem traduz as limitações de máquina em “defeitos de personagem”, potencializando a empatia e facilitando a compreensão didática do *hardware*.

2.2. Game-Based Learning e Domínios Semióticos

A transposição de conceitos complexos e abstratos para elementos interativos encontra forte sustentação no GBL. Sob a perspectiva de James Paul Gee [Gee 2003], os videogames operam como “domínios semióticos”, definidos como sistemas de práticas nos quais símbolos, imagens, artefatos e ações adquirem significados específicos dentro de um contexto compartilhado. Dessa forma, aprender a jogar implica desenvolver um novo letramento (*literacy*), que capacita o estudante a compreender, interpretar e interagir ativamente com a gramática de *design* própria daquele universo virtual.

No ensino de ciências tecnológicas, a construção de um domínio semiótico lúdico atua como um tradutor estrutural. Ao materializar dinâmicas invisíveis em um conjunto de signos espaciais e mecânicas tangíveis, o jogo possibilita que o aluno

assimile empiricamente regras e restrições complexas. Através da manipulação dessas representações visuais e do *feedback* imediato inerente ao *gameplay*, o estudante é estimulado a testar hipóteses e a construir modelos mentais sólidos, mitigando a sobrecarga cognitiva frequentemente associada a abordagens centradas na transmissão de conteúdo..

3. Metodologia: Design Science Research

A presente pesquisa, de natureza aplicada, surge como uma solução para um problema prático: a alta abstração e complexidade de conceitos relacionados à Arquitetura de Computadores.

Na área de Sistemas de Informação existem dois paradigmas que frequentemente guiam as pesquisas: a ciência comportamental (*behavioral science*) e a *design science*. A primeira foca no comportamento humano e/ou organizacional e na sua verificação e previsão. Contudo, a ciência do *design* se difere por buscar formas de melhorar as capacidades humanas e organizacionais por meio dos chamados artefatos, que são elementos de natureza inovadora [Hevner et al. 2004]. Alinhada à busca por elementos com as características apresentadas, a metodologia adotada neste trabalho é a DSR justamente por se tratar de um paradigma voltado à resolução de problemas [Hevner et al. 2004]. Nesse contexto, o artefato projetado e desenvolvido por esta pesquisa consiste no jogo *Operação Overclock*.

Neste artigo, documenta-se o primeiro ciclo da DSR voltado à construção de uma Prova de Conceito (PoC) para validar a tradução do fluxo estrutural da arquitetura de Von Neumann para as mecânicas dentro do jogo. Neste cenário, protótipos são essenciais no ciclo de um projeto, visto que, por meio deles, podemos demonstrar conceitos, funcionalidades e compreender melhor o próprio problema e as soluções possíveis antes que grandes esforços sejam despendidos para seu desenvolvimento [Sommerville 2019].

4. O Artefato: Design e Construção de Operação Overclock

Para garantir a correta representação dos princípios técnicos abordados, o processo de levantamento de requisitos foi executado. Os requisitos são funcionalidades e restrições que o software deve disponibilizar para seus usuários a fim de atender suas necessidades [Sommerville 2019]. No contexto de *Operação Overclock*, o requisito mais importante é a representação de forma adaptada da arquitetura definida por Von Neumann para a interação entre elementos dentro do jogo. Contudo, essa representação técnica deve ser alcançada sem perder a ludicidade, garantindo que a experiência do jogador não seja afetada negativamente e o propósito educacional seja mantido.

4.1. Levantamento de Requisitos e Mapeamento Técnico

O levantamento de requisitos foi conduzido com o objetivo central de garantir que os princípios fundamentais da arquitetura de computadores fossem transpostos com rigor técnico para o ambiente virtual. Para estruturar essa base, a pesquisa apoiou-se no modelo arquitetural clássico de Von Neumann, identificando os componentes físicos primários responsáveis pelo ciclo de processamento de dados: o processador (representado pela Unidade Central de Processamento - CPU), a memória principal, as vias de comunicação (barramentos) e a Unidade Lógica e Aritmética (ULA).

Tabela 1. Mapeamento de componentes com suas respectivas metáforas.

Componente Físico/Conceito	Função Real	Metáfora no Jogo	Mecânica e Ação do Jogador
Sistema Operacional	Gerar instruções e receber os resultados do processamento	A Cientista	NPC que emite as tarefas e aguarda o retorno.
Barramento	Via de comunicação e tráfego de dados	Esteiras de Entrada e Saída	Elemento logístico de entrega das tarefas para o processador.
CPU	Orquestrar o ciclo de instrução	O Operário	Personagem controlado pelo jogador que se move entre as “salas”.
Ciclo de busca	Buscar a informação e interpretá-la	Coleta e Leitura da Caixa	Ação de pegar a caixa da esteira, abrir, ver a descrição para entender a tarefa.
Memória	Alocar dados temporariamente para a instrução	Sala de Memória	Ambiente onde o jogador deposita fisicamente os dados (ex: números e operações).
Unidade Lógica e Aritmética (ULA)	Executar cálculos e operações lógicas	Sala de Cálculo (Puzzle Peças de Tetris)	Minijogo no qual o jogador posiciona as peças de Tetris corretamente para representa a execução da tarefa. Cada peça representa uma variável que foi colocada na memória.

No contexto do desenvolvimento de um artefato educacional, os requisitos transcendem as funcionalidades tradicionais de *software*. Eles atuam como restrições de *design* que obrigam o jogo a simular as reais limitações da máquina. Sendo assim, o mapeamento técnico consistiu em definir como o comportamento de cada um desses componentes de *hardware* ditaria as regras e as mecânicas (*gameplay*) do projeto. A premissa estabelecida foi a de que o jogador não deveria apenas visualizar esses elementos, mas ser submetido de forma prática às suas limitações físicas de operação, como o tempo gasto no tráfego de dados e a necessidade de alocação prévia de informações antes da execução de um cálculo.

A partir dessa definição formal dos requisitos de *hardware*, o processo avançou para a etapa de tradução lúdica. Cada componente técnico foi sistematicamente pareado com uma metáfora do mundo real — focada na logística de um ambiente de trabalho —, assegurando que o rigor teórico permanecesse intacto sob a camada de entretenimento. O resultado desse mapeamento rigoroso entre a função real do componente e a sua respectiva mecânica de simulação no jogo está consolidado na Tabela 1.

4.2. Tradução Comportamental

Para trazer os elementos descritos na tabela para dentro do jogo, adotou-se a estratégia do antropomorfismo. A empatia por agentes virtuais pode impactar positivamente sua aceitação pelos jogadores e potencializar seus benefícios educacionais [Tsumura e Yamada 2023]. Apoiando-se nas Metáforas Conceituais [Lakoff e Johnson 2002] e no viés de agência humano, o *design* do jogo transforma

as limitações de *hardware* em fatores narrativos, especificamente no que a literatura de roteiro categoriza como *character flaws* (defeitos de personagem). Segundo Vogler [Vogler 2006, p. 56], a inclusão de vulnerabilidades, como fraquezas e limitações, é fundamental para humanizar essas figuras. O autor defende que esses defeitos são responsáveis por tornar qualquer personagem “mais real e atraente” para o público, favorecendo a imersão na experiência e gerando uma conexão empática.

Como exemplo, o conhecido “Gargalo de Von Neumann” apresentado por Tanenbaum e Austin [Tanenbaum e Austin 2013] como a limitação na taxa de transferência de dados, é traduzido para o jogo por meio das restrições físicas do personagem CPU (representado como um operário). Segundo os autores, as instruções precisam passar sequencialmente por essa via de comunicação. Logo, o jogo busca refletir essa realidade na movimentação do avatar: o fato de ele precisar caminhar até uma esteira (barramento), conseguir carregar apenas uma caixa por vez e ter que se deslocar fisicamente pelas salas de memória e de cálculo. Essa transposição modela o jogo como um sistema cibernético formal. Conforme postulado por Salen e Zimmerman [Salen e Zimmerman 2012], as regras estabelecem ciclos de *feedback* que regulam o fluxo da interação. Além disso, sob a lente de Jesper Juul [Juul 2005], as regras e restrições são os elementos estruturantes que separam o jogador de seu objetivo. No contexto de *Operação Overclock*, essa restrição logística atua como um ciclo de *feedback* negativo que impede o progresso trivial, traduzindo o Gargalo de Von Neumann em uma limitação física do avatar que gera um desafio lúdico e instiga a resolução de problemas.

Como a metodologia DSR empregada é baseada em ciclos iterativos de construção e avaliação, o planejamento do artefato engloba um modelo mental mais amplo do que o implementado na Prova de Conceito (PoC) inicial. Neste primeiro ciclo de *design* (Ciclo 1), o foco recaiu sobre a validação mecânica do fluxo principal de processamento (*fetch-decode-execute*). Por conseguinte, somente a CPU e o Sistema Operacional foram priorizados para estabelecer a interação fundamental de antropomorfização. O Sistema Operacional foi intencionalmente personificado por uma figura feminina (*A Cientista*), uma escolha de *design* que busca fomentar a representatividade e a inclusão de mulheres nas ciências da computação, contrapondo estereótipos históricos da área. A Memória e a Unidade Lógica e Aritmética (ULA) foram representadas temporariamente como ambientes interativos (salas) para priorizar a validação da alocação dos dados. Contudo, para guiar as expansões do próximo ciclo da DSR, estabeleceu-se um mapeamento prospectivo de traços de personalidade e *character flaws* para as futuras entidades computacionais, visando garantir a coesão da metáfora, conforme estruturado na Tabela 2.

4.3. Construção Narrativa e Integração de Mecânicas

A estrutura narrativa de *Operação Overclock* foi concebida para conduzir o jogador através da evolução histórica do *hardware*. O escopo global do projeto prevê a futura implementação de um estágio tutorial ambientado na era do ENIAC (*Electronic Numerical Integrator and Computer*). Essa fase demonstrará como a programação inicial exigia o manuseio físico de cabos e válvulas, evidenciando a extrema complexidade logística dos primórdios da computação. Contudo, em alinhamento com os objetivos do primeiro ciclo da DSR documentado neste artigo, a Prova de Conceito (PoC) concentra-se exclusivamente na “Fase 1”. Este cenário modela a consolidação da arquitetura de Von

Tabela 2. Mapeamento de personagens, suas personalidades e defeitos.

Componente Físico/Conceito	Personagem	Personalidade e <i>Flaws</i>	Justificativa da Metáfora
Sistema Operacional	A Cientista	Autoritário e pragmático. Focado no resultado final e alheio a esforço logístico interno.	NPC que emite as tarefas e aguarda o retorno.
CPU	O Operário	Empenhado, porém, fisicamente limitado.	Representa o Gargalo de Von Neumann.
Memória	A Arquivista de Memória Curta	Hiperativa e rápida, mas esquece fácil se distraída.	A memória RAM é rápida, mas perde os dados quando o computador desliga (volátil).
Unidade Lógica e Aritmética (ULA)	O Matemático	Age sempre com precisão e com os dados alinhados perfeitamente.	Representa a execução lógica, que deve estar com os dados bem estruturados para que os cálculos sejam feitos.
HD	O Estoquista Lento	Guarda permanentemente tudo que lhe passam, mas demora muito para executar.	Representa a memória capaz de gravar dados sem os perder ao desligar o computador (não volátil), mas possui uma latência muito alta.

Neumann, estabelecendo o ambiente de trabalho contemporâneo do avatar (Figura 1).

**Figura 1. Área principal do jogo**

No ambiente desenvolvido, o ciclo principal de interação (*core loop*) foi arquitetado para simular de forma lúdica o clássico ciclo de instrução do processador (*fetch-decode-execute*). A dinâmica tem início quando o Sistema Operacional, personificado pela personagem “Cientista”, emite uma requisição computacional urgente. A tarefa é depositada em uma esteira de entrada, que atua como a metáfora visual para o barramento do sistema. Ao assumir o controle do avatar (CPU), o jogador deve se deslocar fisicamente até a esteira para coletar a caixa de instrução, materializando espacialmente a etapa de busca (*fetch*) (Figura 2). Após a coleta, o personagem precisa dirigir-se à “Sala de Memória” para organizar os dados e os operadores matemáticos exigidos pela tarefa,

preparando os insumos lógicos necessários para a próxima etapa.



Figura 2. Interação com a missão.

A etapa culminante do ciclo ocorre na “Sala de Cálculo”, ambiente que representa a Unidade Lógica e Aritmética (ULA). Nesse espaço, o jogador visualiza os valores numéricos e os operadores previamente alocados na memória, que são representados por blocos geométricos semelhantes às peças do jogo *Tetris* (Figura 3). A mecânica exige que o jogador posicione e rotacione as peças corretamente para validar o cálculo. Essa exigência de encaixe lúdico traduz uma limitação física real: a necessidade técnica de que os dados estejam perfeitamente estruturados e alinhados nos registradores internos do processador para que a ULA consiga executar a operação aritmética (*execute*).

Sob a ótica do *game design*, esse quebra-cabeça ilustra como restrições operacionais rígidas dão origem ao que Salen e Zimmerman [Salen e Zimmerman 2012] definem como interação lúdica significativa (*meaningful play*). O engajamento emerge do fato de que as ações possuem consequências claras e sistêmicas: o encaixe geométrico fornece um retorno visual imediato. Essa resposta responsiva do sistema proporciona o que Murray [Murray 2003] define como o prazer da agência — a capacidade do jogador de realizar ações significativas e ver os resultados de suas escolhas modelando o universo virtual. Com o quebra-cabeça resolvido, a tarefa é concluída. O ciclo computacional encerra-se com o avatar transportando o resultado até uma esteira de saída, entregando a operação processada de volta ao seu superior.



Figura 3. Sala de cálculo.

4.4. Aspectos Técnicos da Prova de Conceito (PoC)

A Prova de Conceito (PoC) foi desenvolvida utilizando a engine *Unity*, amplamente empregada no desenvolvimento de jogos digitais, devido ao seu suporte multiplataforma e à sua arquitetura baseada em componentes. A implementação foi realizada utilizando

a linguagem C#, nativamente integrada à engine, permitindo a construção de sistemas interativos por meio de scripts orientados a objetos.

O desenvolvimento seguiu o paradigma de programação orientada a objetos (POO), promovendo modularidade, encapsulamento e reutilização de código. A arquitetura do sistema foi organizada com base no modelo de componentes da *Unity*, no qual comportamentos são implementados como *scripts* associados aos objetos da cena, possibilitando uma separação clara entre lógica de jogo, controle de personagens e interação com o ambiente (Figura 1). No contexto da PoC, essa estrutura foi utilizada para implementar as mecânicas centrais descritas anteriormente, como o controle do personagem CPU, a interação com os elementos do cenário (esteiras e salas) e a execução dos *puzzles* que representam as operações da Unidade Lógica e Aritmética (ULA) (Figura 2). A lógica do ciclo de instrução (*fetch-decode-execute*) foi modelada por meio de estados e transições controladas por *scripts*, garantindo a correspondência entre o comportamento do jogador e o fluxo computacional representado.

Além disso, a abordagem adotada favorece a rápida prototipagem e iteração, aspectos fundamentais no contexto da DSR, permitindo validar progressivamente as mecânicas e sua aderência aos conceitos teóricos. Dessa forma, a estrutura técnica implementada não apenas viabiliza o funcionamento do jogo, mas também sustenta a tradução dos conceitos abstratos da arquitetura de computadores em interações lúdicas e compreensíveis.

5. Avaliação Analítica

A avaliação do artefato priorizou aspectos relativos à estrutura e às teorias envolvidas antes de testes com usuários reais. Como abordado por [Hevner et al. 2004], a avaliação durante a pesquisa pode empregar métodos analíticos. Dentre eles, existe a Análise de Arquitetura, a qual avalia o grau de fidelidade do artefato em conformidade a determinada arquitetura de referência. Sob essa ótica, o método Analítico foi escolhido para aplicação na Prova de Conceito (PoC) de *Operação Overclock*, atestando sua coerência quanto à tradução dos conceitos teóricos para elementos e mecânicas presentes no jogo.

A análise arquitetural ratifica que a representação dos conceitos apresenta aderência ao modelo teórico. O Ciclo de Instrução encontra-se representado organicamente ao andamento do *gameplay* da fase. A restrição atribuída ao personagem CPU, de caminhar até a esteira (barramento) e às salas interativas (memória e ULA) com apenas uma caixa (instrução), representa o gargalo de Von Neumann fielmente à teoria. Foi possível constatar que essa limitação do avatar justifica as regras do jogo, fazendo com que um conceito abstrato se transforme em um desafio logístico familiar. As mecânicas referentes à execução de tarefas também passaram pelo processo de avaliação. A metáfora designada demonstrou-se como uma opção analiticamente viável para mostrar ao jogador que os dados (peças) precisam ser organizados corretamente para que a tarefa seja executada, assim como operações lógicas e aritméticas precisam estar devidamente estruturadas no processador para que sejam executadas.

Contudo, reconhece-se que a presente avaliação possui uma natureza estritamente endógena, configurando-se neste momento como uma descrição interpretativa do *design* pelos próprios autores. Por se tratar do primeiro ciclo metodológico da DSR, a Prova de Conceito (PoC) ainda carece da aplicação de protocolos formais e critérios objetivos

de validação externa. A condução de avaliações rigorosas — tais como inspeção por especialistas, análise heurística e testes empíricos de jogabilidade com o público-alvo — constituem etapas metodológicas indispensáveis que serão operacionalizadas nos próximos ciclos da pesquisa.

Por fim, conclui-se que a PoC cumpre seu papel ao ocultar a complexidade do domínio alvo (*hardware*) relacionado com a familiaridade do domínio fonte (ambiente de trabalho). A conformidade estrutural evidenciada por esta análise confirma a viabilidade necessária para que novas iterações da DSR sejam executadas para continuidade de aprimoramentos no artefato apresentado, dando prosseguimento à antropomorfização dos demais conceitos e componentes e posterior avaliação por alunos em ambiente educacional.

6. Considerações Finais e Trabalhos Futuros

O presente trabalho documentou o processo de *design* e a arquitetura de *software* do jogo educacional *Operação Overclock*, criado para reduzir a alta abstração no ensino de Arquitetura e Organização de Computadores. Ao unir os princípios do GBL à técnica narrativa do antropomorfismo, a pesquisa demonstrou como conceitos físicos invisíveis ao programador podem se transformar em mecânicas lúdicas e desafios logísticos espaciais.

Guiada pela DSR, a execução do primeiro ciclo focou na construção e na validação analítica de uma PoC. A análise indica que o artefato apresenta viabilidade estrutural ao converter restrições técnicas de *hardware* em *character flaws*. Essa abordagem levanta a hipótese de que seja possível utilizar vieses cognitivos inatos, ancorados na Teoria da Mente, para transformar a máquina em um sistema tangível e empático. Dessa forma, o projeto preenche uma lacuna metodológica ao propor o estudo da organização física em vez do foco tradicional na escrita de códigos.

Como trabalhos futuros, os próximos ciclos da DSR terão como foco a expansão mecânica e narrativa do jogo. O objetivo será implementar as entidades antropomorfizadas que já foram mapeadas nesta pesquisa. Ambientes que na atual PoC funcionam apenas como salas interativas passarão a ser controlados por personagens próprios: a ULA assumirá a identidade de um matemático rigoroso, enquanto a Memória RAM será representada por uma arquivista hiperativa que esquece com facilidade. A entrada dessas novas figuras vai consolidar a metáfora do processamento de dados como uma interação social e logística complexa.

Por fim, após o amadurecimento estrutural e a inclusão de todo o elenco de personagens, os passos seguintes da pesquisa focarão na avaliação empírica do jogo. Planeja-se aplicar o artefato com estudantes de graduação utilizando instrumentos de análise cognitiva e observação de uso, com a finalidade de medir seus impactos reais, verificar o grau de imersão e validar cientificamente se a ferramenta é eficaz na redução da sobrecarga cognitiva durante o aprendizado da infraestrutura de computadores.

Referências

Dennett, D. C. (1987). *The intentional stance*. MIT press.

Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. Palgrave Macmillan.

- Guthrie, S. (1993). *Faces in the clouds: A new theory of religion*. Oxford University Press.
- Hevner, A. R., March, S. T., Park, J., e Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–106.
- Juul, J. (2005). *Half-real: Video games between real rules and fictional worlds*. MIT press, Cambridge.
- Lakoff, G. e Johnson, M. (2002). *Metáforas da vida cotidiana*. Mercado das Letras, Campinas, SP.
- Murray, J. H. (2003). *Hamlet no holodeck: o futuro da narrativa no ciberespaço*. Unesp/Itaú Cultural, São Paulo.
- Premack, D. e Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526.
- Salen, K. e Zimmerman, E. (2012). *Regras do jogo: fundamentos do design de jogos, volume 1: principais conceitos*. Editora Edgard Blucher, São Paulo. Tradução de Edson Furmankiewicz.
- Sommerville, I. (2019). *Engenharia de Software*. Pearson Education do Brasil, São Paulo, 10 edition.
- Tanenbaum, A. S. e Austin, T. (2013). *Organização Estruturada de Computadores*. Pearson, São Paulo, 6 edition.
- Tsumura, T. e Yamada, S. (2023). Influence of anthropomorphic agent on human empathy through games. *IEEE Access*, 11:40412–40429.
- Vogler, C. (2006). *A Jornada do Escritor: Estruturas Míticas para Escritores*. Nova Fronteira, Rio de Janeiro.