

# **ULA LÁ!**

**Lucas Molinari, Enrique Susin, Pedro Abreu**

Instituto de Ciência e Tecnologia – Universidade Federal de São Paulo (UNIFESP) –  
São José dos Campos – SP – Brasil

l.molinari@unifesp.br, enrique.susin@unifesp.br, pedro.marx@unifesp.br

**Resumo.** Este trabalho apresenta o desenvolvimento de “ULA LÁ!”, um jogo de tabuleiro educativo baseado no jogo Saboteur, adaptado para ensinar conceitos fundamentais de Arquitetura e Organização de Computadores (AOC). O jogo simula o fluxo de dados e instruções através dos componentes de um sistema computacional, desde periféricos de entrada até a ULA. Os jogadores assumem papéis secretos de dados/instruções ou de gargalos de sistema, construindo colaborativamente caminhos através de barramentos, controladores e hierarquia de memória. Os jogadores devem conhecer as técnicas corretas para problemas específicos de arquitetura, incentivando o conhecimento dos conceitos de AOC. O protótipo visa proporcionar uma experiência lúdica e engajadora que reforce o aprendizado de conceitos como comunicação de E/S, DMA, interrupções, cache, deadlock e o gargalo de Von Neumann.

## **1. Introdução**

### **1.1. Contextualização**

A Arquitetura e Organização de Computadores é uma disciplina fundamental nos cursos de Computação, abordando conceitos essenciais sobre o funcionamento interno dos sistemas computacionais, desde a comunicação entre periféricos até o processamento na CPU. Entretanto, a natureza abstrata de muitos desses conceitos, especialmente relacionados a fluxo de dados, latência e comunicação hardware, frequentemente dificulta a compreensão por parte dos estudantes (PATTERSON; HENNESSY, 2017).

Metodologias ativas de aprendizagem, como jogos educativos, têm demonstrado eficácia significativa no engajamento e retenção de conhecimento em disciplinas técnicas. Os jogos proporcionam um ambiente onde os alunos podem experimentar conceitos abstratos de forma concreta e interativa, facilitando a construção de conhecimento significativo (SAVI; ULBRICHT, 2008).

### **1.2. Objetivos**

Este projeto desenvolve um jogo de tabuleiro que facilite o aprendizado de AOC de forma lúdica, com ênfase em dispositivos de E/S, barramentos e comunicação entre componentes. Os objetivos específicos são: simular fluxo de dados desde periféricos até a ULA; demonstrar problemas comuns (gargalos de E/S, deadlock, polling ineficiente); implementar mecânica que exija conhecimento técnico específico; e contextualizar problemas históricos como o gargalo de Von Neumann.

## **2. Fundamentação teórica**

### **2.1. Conceitos de Arquitetura e Organização de Computadores**

Hierarquia de Memória: Organizada em níveis com diferentes velocidades e capacidades, desde registradores até armazenamento secundário (STALLINGS, 2010). O princípio da localidade justifica memórias cache (L1, L2, L3). Cache miss força acesso à memória principal, mais lenta.

Barramentos: Vias de comunicação compartilhadas conectando componentes. USB, SATA e PCIe possuem características distintas de largura de banda e latência (TANENBAUM; AUSTIN, 2013). Bus contention ocorre quando múltiplos dispositivos competem simultaneamente, degradando desempenho.

Arquitetura de Von Neumann: Dados e instruções compartilham memória e barramento, criando o "gargalo de Von Neumann" (PATTERSON; HENNESSY, 2017). A arquitetura Harvard resolve parcialmente este problema com memórias separadas. Sistemas modernos usam arquitetura Harvard modificada nos caches L1.

Dispositivos de E/S: Comunicação entre periféricos e CPU ocorre por: polling (CPU verifica periodicamente o status), interrupções (dispositivo notifica via hardware) e DMA (transferências diretas sem envolvimento constante da CPU) (STALLINGS, 2010).

Deadlock: Ocorre quando processos ficam bloqueados indefinidamente esperando recursos uns dos outros (TANENBAUM; AUSTIN, 2013). O Algoritmo do Banqueiro verifica se alocações mantêm o sistema seguro. Outras soluções incluem ordenação de recursos e preempção.

ULA: Componente da CPU responsável por operações aritméticas e lógicas (PATTERSON; HENNESSY, 2017). Instruções devem chegar à ULA para serem processadas.

### **2.2. Trabalhos relacionados**

Savi e Ulbricht (2008) destacam que jogos educacionais promovem motivação e aprendizagem ativa. Silva et al. (2015) desenvolveram um simulador visual de pipeline MIPS, facilitando a compreensão de conceitos abstratos. Costa e Silva (2017) propuseram um jogo digital para organização de computadores com resultados positivos.

O diferencial deste trabalho está no formato físico (promove interação social direta), ênfase em periféricos e E/S, e mecânica inovadora exigindo correspondência correta entre problemas e soluções técnicas.

## **3. Materiais e métodos**

### **3.1. Desenvolvimento do protótipo**

Etapa 1: Definição de regras, balanceamento de papéis, mapeamento de cartas de problemas e soluções.

Etapa 2: Criação de cartas de componentes, sabotagem, solução e papéis secretos.

Etapa 3: Design gráfico, impressão e montagem do protótipo, criação do manual.

Etapa 4: Testes internos e com outros alunos, ajustes de balanceamento.

### **3.2. Materiais necessários**

- 60-80 cartas de caminho (componentes)
- 20-30 cartas de sabotagem (problemas)
- 20-30 cartas de solução
- 8-10 cartas de papel secreto
- 3 cartas finais (destinos)

### **3.3. Descrição do protótipo**

Jogadores: 4 a 8 participantes

Papéis: Dados/Instruções (maioria - representam informações de periféricos ou instruções como LOAD, STORE, ADD) e Gargalos (1-2 jogadores - componentes problemáticos como HD mecânico, USB 2.0 defeituoso, placa de rede 10Mbps).

Estrutura: Carta inicial (periféricos) e três cartas de destino viradas (duas falhas e uma ULA funcional).

Construção: Jogadores constroem caminho com cartas de: periféricos → barramentos (USB, PCIe, SATA) → controladores (DMA, I/O, Interrupt) → memória (Cache, RAM) → ULA.

Sistema de Problemas e Soluções: Cartas de sabotagem representam problemas reais. Para remover, jogador deve usar solução correta:

- Gargalo de Von Neumann → Arquitetura Harvard
- I/O Wait → DMA ou Buffering
- Device Not Ready → Ready Signal/Interrupt
- Polling Ineficiente → Sistema de Interrupções
- Bus Contention → Arbitragem de Barramento
- Buffer Overflow → Controle de Fluxo
- Deadlock (2+ jogadores) → Algoritmo do Banqueiro

Utilizar uma solução incorreta resulta em perda da carta sem efeito.

Vitória: Dados/Instruções vencem chegando à ULA, enquanto Gargalos vencem impedindo ou levando a falhas.

### **3.4. Avaliação**

Testes internos: Jogabilidade básica (3-5 partidas).

Testes com pares: Sessão com outros alunos, feedback sobre clareza.

Avaliação educacional: Questionário pré/pós-jogo comparando aprendizado.

Métricas: Tempo de partida (30-45min ideal), compreensão de regras, taxa de acerto problema-solução, engajamento (Likert 1-5), delta de aprendizado.

#### **4. Resultados esperados**

Espera-se que "ULA LÁ!" proporcione:

Compreensão de Conceitos: Melhor entendimento sobre hierarquia de memória, barramentos, métodos de E/S (polling, interrupção, DMA), gargalo de Von Neumann, deadlock e papel da ULA.

Conhecimento Técnico Preciso: A correspondência obrigatória entre problemas e soluções específicas força compreensão exata de qual técnica resolve cada problema.

Engajamento: Natureza lúdica e competitiva com elementos de dedução social deve aumentar o interesse pela disciplina.

Aprendizagem Ativa: Decisões estratégicas baseadas em conhecimento de AOC promovem aprendizagem significativa através da experiência prática.

#### **5. Conclusão**

A proposta do jogo combina mecânicas estabelecidas (Saboteur) com conceitos fundamentais de AOC, equilibrando diversão e rigor técnico.

A fundamentação demonstrou centralidade dos conceitos abordados. A ênfase em E/S e comunicação entre componentes preenche lacuna comum em abordagens tradicionais. A mecânica de correspondência obrigatória entre problemas e soluções exige conhecimento técnico preciso.

A metodologia busca garantir protótipo funcional através de ciclos iterativos. Os resultados esperados indicam contribuição significativa ao aprendizado: tornando conceitos concretos, promovendo aprendizagem ativa, facilitando identificação de lacunas e aumentando engajamento.

Como trabalhos futuros: expandir com conceitos avançados (pipeline, cache coherence), criar guia do professor, avaliar quantitativamente o impacto educacional e explorar uma possível versão digital.

O projeto representa contribuição às metodologias ativas no ensino de Computação, demonstrando que é possível aliar rigor conceitual, precisão técnica, diversão e aprendizagem significativa.

#### **7. Referências bibliográficas**

Costa, T. K. L. and Silva, M. A. A. (2017) "Jogo digital educacional para o ensino de organização de computadores", In: Anais do XXIII Workshop de Informática na Escola, Porto Alegre, SBC, p. 91-100.

Patterson, D. A. and Hennessy, J. L. (2017), Computer Organization and Design: The Hardware/Software Interface, 5th edition, Elsevier, Rio de Janeiro.

Savi, R. and Ulbricht, V. R. (2008) "Jogos digitais educacionais: benefícios e desafios", RENOTE - Revista Novas Tecnologias na Educação, v. 6, n. 1, p. 1-10.

Silva, J. C. et al. (2015) "Simulador visual de pipeline MIPS como ferramenta de apoio ao ensino de Arquitetura de Computadores", In: Anais do XXIII Workshop sobre Educação em Computação, Porto Alegre, SBC, p. 123-132.

Stallings, W. (2010), Computer Organization and Architecture, 8th edition, Pearson Prentice Hall, São Paulo.

Tanenbaum, A. S. and Austin, T. (2013), Structured Computer Organization, 6th edition, Pearson, São Paulo.