Estudo Comparativo entre SOSim e OS Sim: Uma Análise Experimental de Simuladores de Sistemas Operacionais

Pâmela Braga, Nicolas Cipriano

[1]Universidade Federal de Pelotas (UFPel)
Pelotas – RS – Brasil

Resumo. *Este artigo apresenta um estudo comparativo entre dois simuladores de Sistemas Operacionais: SOSim e OS Sim. O objetivo é analisar aspectos de usabilidade, gerência de processos, políticas de escalonamento e mecanismos de memória. Foram realizados experimentos práticos em ambas as ferramentas para avaliar aderência teórica e capacidade de visualização. Os resultados indicam que o SOSim oferece maior profundidade técnica e alinhamento com conceitos acadêmicos, enquanto o OS Sim apresenta interface mais intuitiva e moderna. O estudo destaca vantagens, limitações e aplicações pedagógicas de cada simulador.*

1. Introdução

Simuladores de Sistemas Operacionais são ferramentas didáticas amplamente utilizadas no ensino de conceitos como escalonamento, estados de processos, paginação e gerenciamento de memória. Essas ferramentas permitem visualizar dinamicamente comportamentos que, teoricamente, são abstratos e de difícil compreensão sem experimentação prática [1].

Este trabalho apresenta um estudo comparativo entre o simulador clássico SOSim e o simulador alternativo OS Sim, com foco na análise de usabilidade, gerência de processos e gerência de memória. O objetivo é identificar as vantagens, limitações e adequações pedagógicas de cada ferramenta para o ensino de Sistemas Operacionais.

O SOSim [3] é um simulador clássico desenvolvido para Windows e Linux (via Wine), amplamente utilizado em disciplinas de graduação brasileiras. O OS Sim [4] é uma alternativa de código aberto, multiplataforma e desenvolvida em Java, disponível no SourceForge.

2. Visão Geral e Experiência do Usuário

2.1. SOSim

O SOSim apresenta uma interface técnica dividida em módulos independentes: console principal, gerência de processos, gerência do processador e gerência de memória. A ferramenta fornece visualização detalhada de estados, filas de execução, prioridades e *frames* de memória física.

Apesar da interface gráfica simples, o simulador oferece controle refinado de parâmetros como quantum, *clock* da CPU e prioridades de processos. O idioma da interface é o português, o que facilita o uso por estudantes brasileiros. A fila de prontos

é exibida como uma barra vertical numerada de 0 a 15, representando os níveis de prioridade, e o processo em execução é indicado por uma bolinha colorida no painel de Execução.

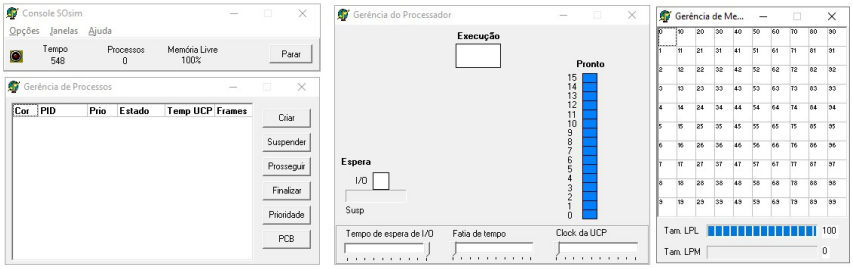


Figure 1. Interface principal do SOSim: janelas de Gerência de Processos, Gerência do Processador e Gerência de Memória.

2.2. OS Sim

O OS Sim apresenta uma interface gráfica moderna e organizada por módulos (*CPU*, *RAM*, *File System* e *Disk*). Sua navegação é mais intuitiva para iniciantes, com organização centralizada em menus e fluxo visual horizontal da *Ready Queue* que facilita a compreensão do ciclo de vida dos processos.

Cada processo na fila de prontos é exibido como uma caixa colorida com nome, PID e prioridade, além de barrinhas de *burst* abaixo de cada caixa: quadradinhos vermelhos representam *bursts* de CPU e cinzas representam *bursts* de I/O. Isso permite identificar visualmente o tipo de processo (*CPU-bound* ou *I/O-bound*) antes mesmo do início da execução.

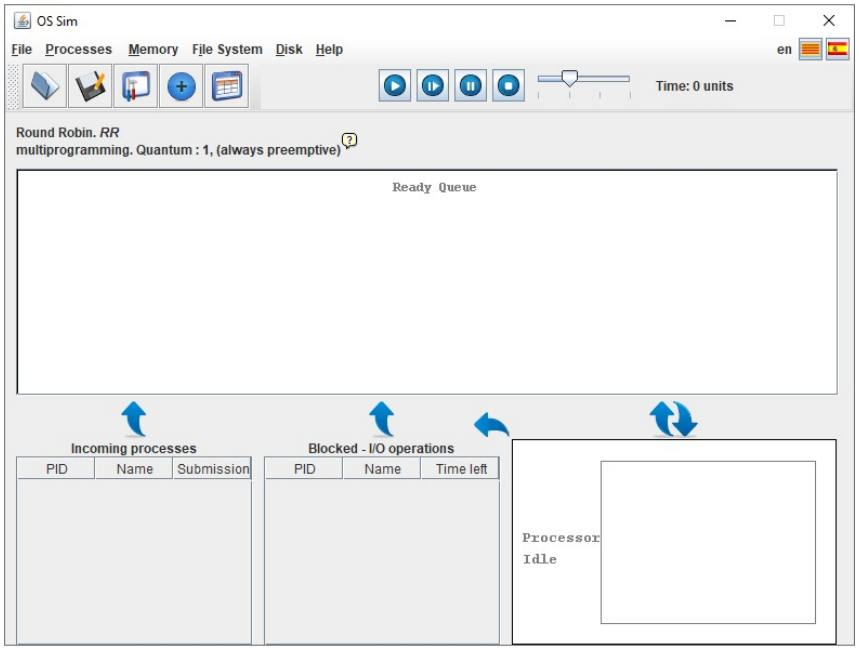


Figure 2. Interface do OS Sim: abas de ”Ready Queue”, ”Processor Idle”, ”Blocked = I/O Operation” e ”Incoming processes”.

2.3. Comparação de Usabilidade

Enquanto o OS Sim se destaca pela interface visual moderna e facilidade de uso, o SOSim apresenta maior profundidade técnica, sendo mais adequado para experimentação acadêmica avançada. O SOSim utiliza interface em português, o que reduz a barreira de entrada para estudantes brasileiros, enquanto o OS Sim opera em inglês e espanhol.

Table 1. Comparação de Usabilidade e Interface

Critério	SOSim	OS Sim
Idioma da interface	Português	Inglês/Espanhol
Plataforma	Windows/Linux (Wine)	Multiplataforma (Java)
Fila de prontos	Barra vertical por nível	Caixas horizontais
Processo em execução	Bolinha colorida	Caixa verde no painel
Código aberto	Não	Sim (SourceForge)
Curva de aprendizado	Moderada	Baixa (mais intuitivo)

3. Gerência de Processos

3.1. Criação de Processos: CPU-bound e I/O-bound

Em ambos os simuladores foram criados processos com características distintas de uso de CPU e I/O, seguindo o cenário da documentação:

- **P1 (I/O-bound):** 2 CPU $\rightarrow$ 2 I/O $\rightarrow$ 1 CPU
- **P2 (CPU-bound):** 4 CPU
- **P3 (misto):** 3 CPU $\rightarrow$ 2 I/O $\rightarrow$ 2 CPU
- **P4 (CPU-bound):** 3 CPU

Todos os processos foram submetidos no instante $t = 0$.

No SOSim, a distinção entre processos *CPU-bound* e *I/O-bound* é feita já no momento da criação: a janela Criação de Processo oferece um menu *dropdown* com tipos pré-definidos, como *CPU-bound*, *I/O-bound (disco)*, *I/O-bound (fita)*, *I/O-bound (terminal)*, *CPU e I/O-bound (disco)* e *CPU e I/O-bound (fita)*. O usuário seleciona o tipo desejado, define a prioridade e o limite de *frames*, e o simulador configura automaticamente o comportamento do processo. Durante a execução, a janela de Gerência de Processos exibe a coluna Estado, que indica em tempo real se o processo está em *Execução*, *Pronto* ou *I/O*, e processos *I/O-bound* aparecem na seção Espera/I/O representados por bolinhas coloridas.

No OS Sim, a criação é mais direta e flexível: o usuário define apenas o nome e a prioridade do processo e, em seguida, configura cada *burst* individualmente, marcando-o como CPU ou I/O. Não há tipos pré-definidos – o perfil do processo emerge da sequência de *bursts* configurada. Essa abordagem torna a natureza do processo (*CPU-bound* ou *I/O-bound*) imediatamente visível na *Ready Queue* já antes da execução, através das barrinhas exibidas abaixo de cada caixa de processo. Quando um processo entra em I/O durante a simulação, é movido para a tabela *Blocked – I/O operations*.

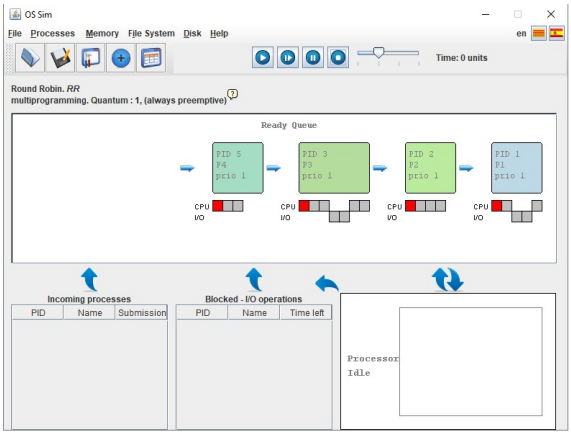


Figure 3. Janela de criação de processo no OS Sim.

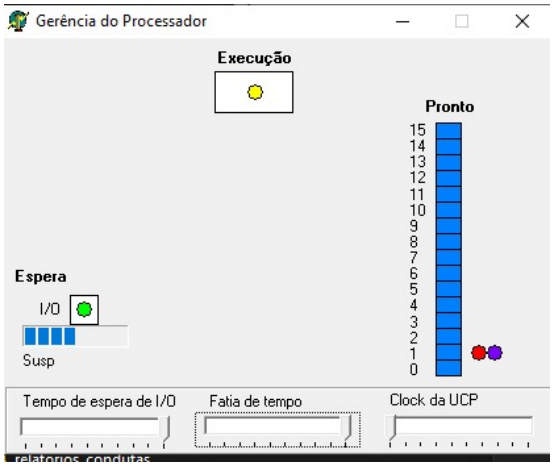
3.2. Estados do Processo

No SOSim, é possível visualizar claramente os estados clássicos do processo [2]:

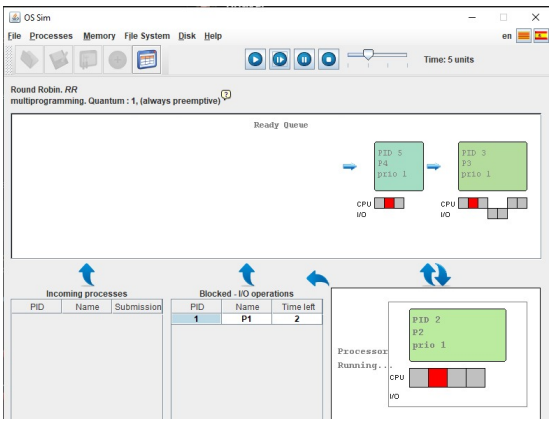
- **Pronto** – processo na fila aguardando a CPU
- **Executando** – bolinha colorida no painel de Execução
- **Bloqueado (I/O)** – bolinha no painel Espera/I/O com barra de progresso
- **Suspenso** – processo explicitamente suspenso pelo usuário

As transições são exibidas dinamicamente, permitindo observar o comportamento do escalonador. O SOSim ainda oferece o botão PCB para visualizar o Bloco de Controle do Processo, recurso pedagógico importante que não está disponível no OS Sim.

No OS Sim, os estados são inferidos pela posição visual do processo nos painéis: *Ready Queue* (Pronto), painel *Processor Running* (Executando) e tabela *Blocked – I/O* (Bloqueado). Embora não exiba um diagrama gráfico de transição de estados, o fluxo visual horizontal torna as transições intuitivas para iniciantes.



(a) SOSim: processo bloqueado (bolinha no painel Espera/I/O)



(b) OS Sim ($t = 5$): P1 bloqueado em I/O com *Time left = 2*

Figure 4. Comparação da visualização do estado Bloqueado (I/O) nos dois simuladores.

3.3. Escalonamento Round Robin

O SOSim permite configurar fatia de tempo (quantum), *clock* da UCP e prioridades. A barra animada de Fatia de tempo é um recurso altamente didático: é possível observar o quantum sendo consumido em tempo real até zerar e o processo ser preemptado, retornando ao final da fila de prontos.

No OS Sim, o Round Robin foi configurado com quantum igual a 1 unidade, mas é possível utilizar outros valores. O funcionamento é correto e verificável – os processos se alternam na CPU a cada *tick* –, porém o quantum é exibido apenas como valor estático no topo da tela, sem animação visual do seu consumo. Isso torna a visualização do mecanismo menos imediata do que no SOSim.

Ambas as ferramentas executam corretamente o escalonamento Round Robin com multiprogramação, permitindo que processos em I/O liberem a CPU para os demais.

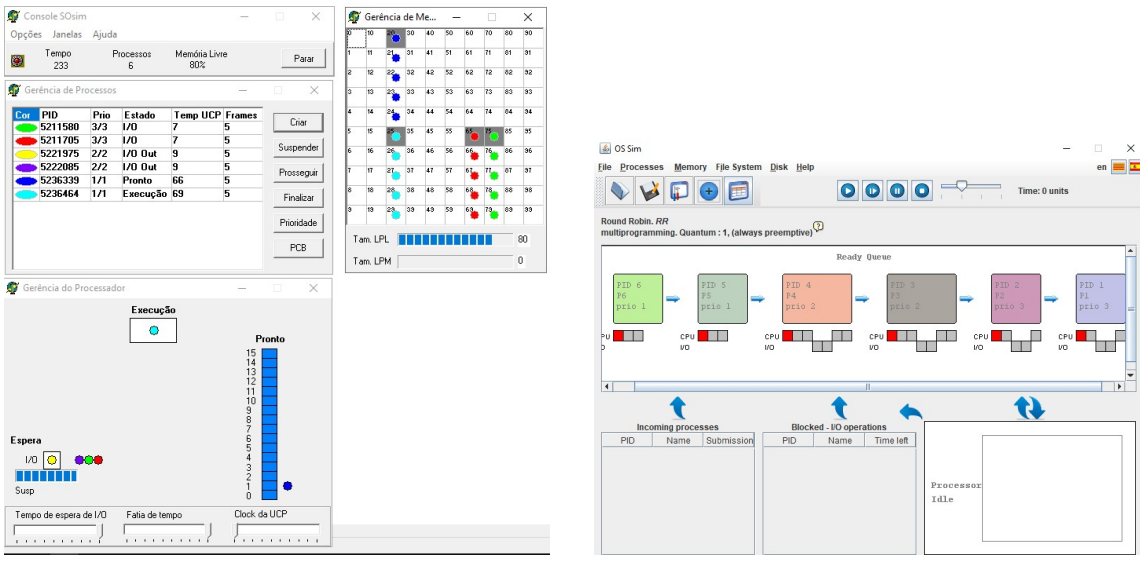
3.4. Experimento de Prioridades

No SOSim foram configurados 6 processos com diferentes prioridades:

- 2 processos com Prioridade 3 (I/O-bound)
- 2 processos com Prioridade 2 (mistos)
- 2 processos com Prioridade 1 (CPU-bound)

O SOSim permite definir e modificar prioridades dinamicamente, possibilitando análise comparativa entre diferentes configurações. A fila de prontos exibe os processos por nível de prioridade (0 a 15), tornando imediatamente visível quais processos terão preferência de execução.

O OS Sim suporta prioridades, exibindo o campo `prio` em cada caixa de processo. Ao inverter as prioridades nos dois simuladores, ambos demonstraram comportamento consistente: processos de maior prioridade passaram a ser escalonados antes.



(a) SOSim (b) OS Sim

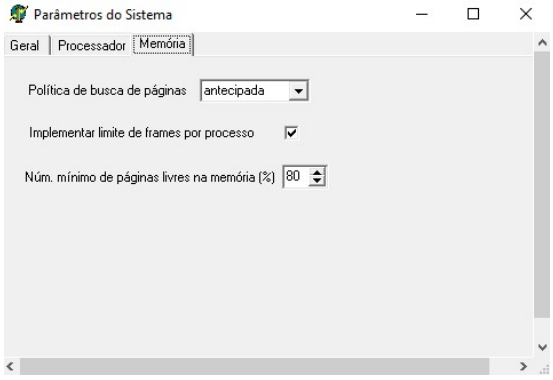
Figure 5. Processos configurados para analise do impacto da prioridade.

4. Gerência de Memória

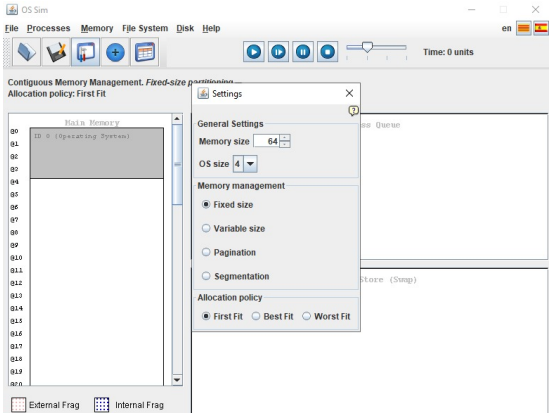
4.1. Políticas de Busca

O SOSim oferece controle explícito sobre a política de busca de páginas, configurável na janela Parâmetros do Sistema, aba Memória (Figura 6a). O usuário pode alternar entre paginação antecipada (*prefetching*) e paginação por demanda (*demand paging*), além de definir o número mínimo de páginas livres na memória e habilitar o limite de *frames* por processo.

O OS Sim também oferece configuração de gerência de memória, porém através de uma janela de Settings independente (Figura 6b). O simulador suporta quatro modelos de gerenciamento: *Fixed size*, *Variable size*, *Pagination* e *Segmentation*. Para a política de alocação, é possível escolher entre *First Fit*, *Best Fit* e *Worst Fit*. Embora esses recursos sejam expressivos, há uma limitação importante: a aba de memória opera de forma completamente independente da aba de processos, de modo que os processos criados na gerência de processos não são refletidos automaticamente na gerência de memória, sendo necessário cadastrá-los separadamente. Isso reduz a coerência da simulação e diverge do funcionamento integrado esperado em um sistema operacional real.



(a) SOSim: política de busca configurável entre paginação antecipada e por demanda.



(b) OS Sim: Settings com suporte a Fixed size, Variable size, Pagination, Segmentation e políticas First/Best/Worst Fit.

Figure 6. Configurações de gerência de memória nos dois simuladores.

4.2. Page Faults

No SOSim, a atividade de paginação é acompanhada pela janela Arquivo de Paginação, exibida separadamente da janela de memória física (Figura 7). Páginas marcadas com "R" (*referenced*) indicam acessos recentes, permitindo inferir a frequência de substituições sob carga. Com 6 processos ativos e memória livre em 83%, o escalonador de memória atuou para manter o limiar mínimo configurado de 80% de páginas livres. Embora o SOSim não exiba um contador numérico de *page faults*, o padrão de substituição observado na janela de paginação permite identificar a ocorrência de falhas de página durante a execução.

O OS Sim oferece o modelo de *Pagination* nas configurações de memória, porém,

como o módulo de memória opera de forma independente da aba de processos, não é possível observar *page faults* integrados à execução dos processos escalonados. Não há contador, log ou indicador de falhas de página durante a simulação de processos.

4.3. Análise: Logs e Desempenho da Memória

O SOSim oferece visualização em tempo real da memória física através do *grid* de *frames* coloridos por processo e do Arquivo de Paginação, que exibe as páginas em *swap* com marcação "R". Embora não gere logs exportáveis ou gráficos de desempenho, a visualização simultânea das duas janelas permite acompanhar o comportamento da memória de forma detalhada durante toda a execução.

O OS Sim não oferece logs ou gráficos de desempenho de memória. Sua interface exibe apenas o estado atual das partições, com legenda distinguindo Fragmentação Externa e Fragmentação Interna, conceitos relevantes nos modelos de tamanho fixo e variável. Apesar da riqueza de modelos configuráveis (incluindo paginação e segmentação), a ausência de integração com a aba de processos impede uma análise completa do comportamento da memória em conjunto com o escalonamento, o que é uma limitação pedagógica significativa em relação ao SOSim.

4.4. Estruturas de Controle: PCB e Frames

O SOSim disponibiliza duas janelas complementares para análise das estruturas de controle (Figura 7): a Gerência de Memória apresenta os *frames* físicos numerados e identificados por cor de processo, com indicadores de tamanho de LPL (*lista de páginas livres*) e LPM (*lista de páginas da memória*); o Arquivo de Paginação complementa essa visão exibindo as páginas em *swap*. Adicionalmente, o botão PCB na janela de Gerência de Processos permite inspecionar individualmente o Bloco de Controle de cada processo, incluindo informações de mapeamento de memória.

O OS Sim não dispõe de visualização de *frames* numerados, tabela de páginas ou equivalente ao PCB. Uma limitação adicional é que o módulo de Memory opera de forma independente da aba de processos: os processos criados na gerência de processos não são refletidos automaticamente na gerência de memória, sendo necessário cadastrá-los separadamente neste módulo. O SOSim, por outro lado, integra completamente o ciclo de vida dos processos à alocação de memória, refletindo com maior fidelidade o funcionamento de um sistema operacional real.

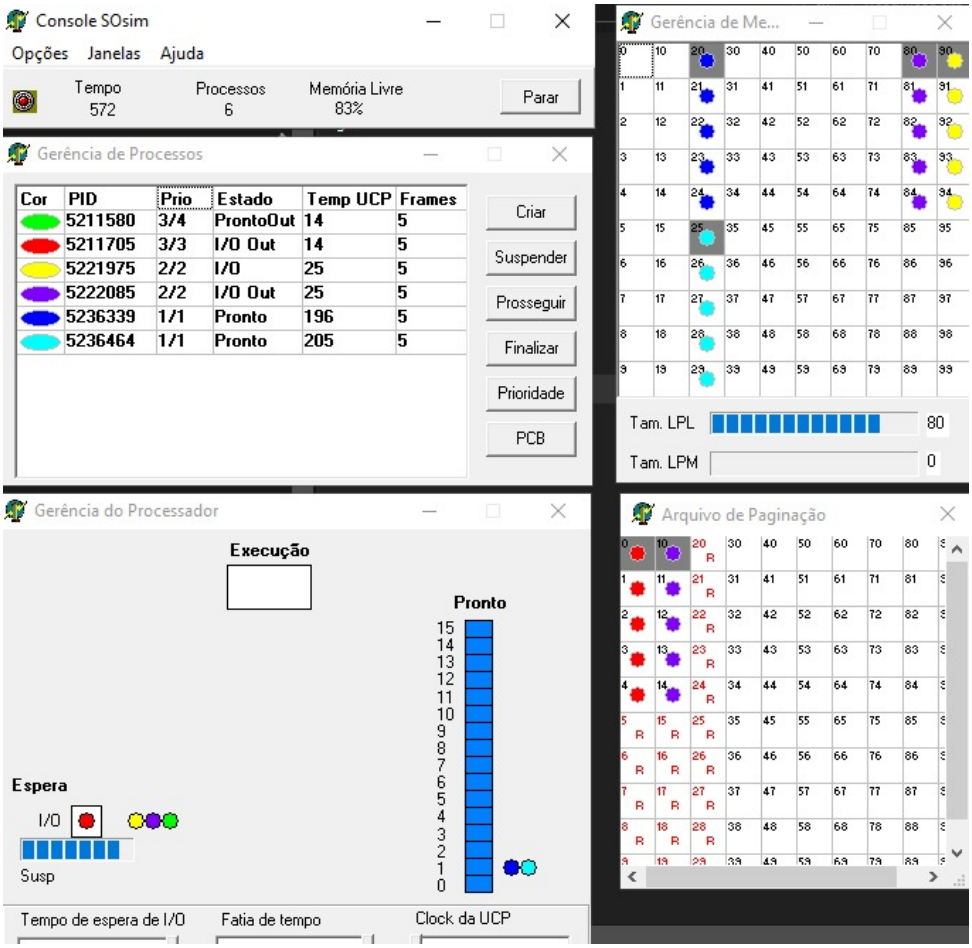


Figure 7. SOSim com 6 processos ativos: Gerência de Memória (*frames* coloridos por processo), Arquivo de Paginação (páginas com marcação "R") e Gerência de Processos (estados, prioridades e tempo de CPU). Memória livre: 83%.

5. Conclusão

O estudo comparativo evidencia que as ferramentas possuem propostas pedagógicas distintas e complementares. O OS Sim é adequado para a introdução aos conceitos básicos de Sistemas Operacionais, devido à sua interface moderna e intuitiva, com visualização imediata do tipo de processo (*CPU-bound* vs *I/O-bound*) e fluxo visual da *Ready Queue*.

O SOSim demonstra maior fidelidade teórica e capacidade de experimentação acadêmica, sendo mais indicado para aprofundamento conceitual e análises detalhadas de escalonamento, estados de processo e gerência de memória. Recursos como o botão PCB, a barra animada de quantum e a visualização detalhada de *frames* são diferenciais significativos para o estudo avançado.

Conclui-se que as ferramentas são complementares: o OS Sim pode ser utilizado em estágios iniciais do aprendizado para construir intuição visual sobre os conceitos, enquanto o SOSim se mostra mais apropriado para estudo avançado e experimentação prática com maior rigor acadêmico.

References

[1] MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**. 5. ed. Rio de Janeiro: LTC, 2013.

[2] SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Fundamentos de Sistemas Operacionais**. 9. ed. Rio de Janeiro: LTC, 2015.

[3] SOSim – Simulador de Sistemas Operacionais. Disponível em: `http://www.din.uem.br/~ia/sosim`. Acesso em: fev. 2026.

[4] OS Sim – Operating System Simulator. Disponível em: `https://sourceforge.net/projects/oscsimulator`. Acesso em: fev. 2026.