



Estudo de Alocação de Memória Visando Ampliar a Sobrevida de Sistemas Computacionais Defasados: Uma Abordagem para Promoção da Inclusão Digital em Escolas Públicas

Felipe Dias de Oliveira - UFPE - fdo@cin.ufpe.br,

ORCID: 0000-0002-0282-6717

André Barreto Rodrigues - UFPE - barretorodrigues16@gmail.com,

ORCID: 0000-0003-1007-2617

Ednaldo Dantas - UFPE - ednaldo.dantas.jr@gmail.com,

ORCID: 0000-0001-5029-7533

Anderson Fernandes Alencar - UFPE - anderson.alencar@ufape.edu.br,

ORCID: 0000-0002-1539-1775

Rodrigo Gusmão de Carvalho Rocha - UFPE - rodrigo.rocha@ufape.edu.br,

ORCID: 0000-0003-1993-5044

Igor Medeiros Vanderlei - UFPE - igor.vanderlei@ufape.edu.br,

ORCID: 0000-0002-1939-6979

Jean Carlos Teixeira de Araujo - UFPE - jean.teixeira@ufape.edu.br,

ORCID: 0000-0002-1688-4782

Resumo. O programa Um Computador por Aluno (UCA) foi uma iniciativa do Governo Federal que objetivava promover a inclusão digital para crianças de escolas públicas. Após mais de 10 anos do seu lançamento, esses computadores já se encontram demasiadamente defasados. Este artigo buscou avaliar a melhor configuração de memória *swap*, a fim de buscar o equilíbrio entre disponibilizar a maior quantidade de espaço em disco para instalação dos programas educacionais sem comprometer ainda mais o desempenho dos aplicativos em execução. Para atingir os objetivos, foi utilizada uma metodologia de avaliação de desempenho de sistemas em duas configurações de memória, onde concluiu-se que a configuração sem a partição de *swap* apresenta o desempenho ligeiramente inferior à configuração com 512MB de *swap*.

Palavras-chave: *Sistemas educacionais, avaliação de desempenho, ensino e aprendizagem.*

Memory Allocation Study Aiming to Extend the Survival of Outdated Computer Systems: An Approach to Promoting Digital Inclusion in Public Schools

Abstract. *The One Computer per Student (UCA) project was a Federal Government initiative that aimed to promote digital inclusion for children in public schools. However, more than 10 years after their launch, these computers are already outdated. This paper sought to evaluate the best swap memory configuration in order to find a balance between making the largest amount of disk space available for installing educational programs without further compromising the performance of running applications. To achieve the goals, a methodology for evaluating the performance of systems in two memory configurations was used, where it was concluded that the configuration without the swap partition presents a slightly lower performance than the configuration with 512MB of swap.*

Keywords: *Educational systems, performance evaluation, teaching and learning.*



1. Introdução

Os benefícios da utilização da tecnologia como ferramenta de auxílio ao processo de ensino e aprendizagem é um tema tratado quase como unanimidade entre os pesquisadores da área de Educação. Entretanto, do ponto de vista social, em localidades fortemente marcadas pela desigualdade social, como é o caso do Brasil, a tecnologia é apenas mais um instrumento que reforça e ajuda a perpetuar essa desigualdade. Nas escolas voltadas ao público das classes sociais A e B, em geral, existe um investimento para aquisição e manutenção dos modernos recursos tecnológicos e nos programas de formação continuada para que os professores possam tirar o melhor proveito desses recursos. Por outro lado, Silva *et al.* (2018) apontam “a carência enfrentada por grande parte das escolas públicas brasileiras não possuem laboratórios de ciências ou de informática”, o que deixa seus estudantes “ainda mais à margem do espaço laboral e profissional que essa cultura tecnológica configura” (LÖBLER *et al.*, 2012).

Buscando intervir nesta realidade do município de Garanhuns-PE, as atividades extensionistas ligadas à informática na educação, no contexto da Universidade Federal do Agreste de Pernambuco (ex UFRPE), datam de 2012, iniciadas por meio de projetos voltados à formação de professores(as) da rede municipal de Garanhuns-PE, seja para a construção de conteúdos digitais educacionais, produção e utilização de Objetos de Aprendizagem, uso do Linux Educacional, de laboratórios de informática ou para uso de tecnologias na Educação de Jovens e Adultos (SILVA *et al.*, 2020).

A partir de 2016, também por meio de projetos de extensão, a partir de solicitação advinda da própria Secretaria Municipal de Educação, professores e estudantes envidaram esforços para promover a reativação dos laboratórios em desuso de todas as escolas do supracitado município. Contudo, esta reativação, além de demandar recuperação física de hardware e formação de professores, implicava na preparação de personalização GNU/Linux para utilização tanto em *netbooks* do extinto Programa “Um Computador por Aluno”, quanto para computadores de mesa, dado que o município não dispunha de licenças de outros sistemas operacionais, além de, na ocasião ter sido uma opção técnica do grupo envolvido por questões de desempenho. À personalização GNU/Linux para os *netbooks* foi dada o nome de edUCA, e para os computadores de mesa, SofiaEdu (SILVA *et al.*, 2018). Com a realização deste projeto, foi possível trazer de volta às salas de aula cerca de 367 *netbooks* UCA que estavam abandonados nos depósitos das escolas.

No ano de 2019, após cerca de três anos de desenvolvimento do edUCA, a equipe considerou necessária uma atualização por conta da quantidade de correções de segurança e atualização de grande parte dos programas nele disponibilizados. Neste processo, apesar da escolha de uma distribuição que consumisse pouco espaço em disco (Debian com interface gráfica LXDE), esses softwares atualizados aumentaram de tamanho consumido em disco, ocasionando problemas relacionados às funcionalidades.

Diante deste cenário, foi cogitada a possibilidade de remoção da partição *swap*, viabilizando a inserção dos aplicativos que não estavam mais sendo contemplados. Tomar essa decisão, contudo, implicava em um estudo minucioso para avaliar o desempenho do *netbook* diante de tal configuração. Os aplicativos rodariam com a mesma agilidade ou haveria insatisfações dos estudantes diante de suas experiências prévias com os *smartphones*? Como oferecer a melhor experiência com um recurso já obsoleto quanto ao seu hardware diante destas soluções contemporâneas?

Portanto, o objetivo deste artigo é avaliar diferentes configurações de memória para utilização dos *netbooks* do Programa Um Computador por Aluno (PEIXOTO,

2017). Devido à pouca disponibilidade de recursos computacionais nesses dispositivos, a incorreta configuração pode comprometer a experiência de uso, dificultando a eficiente utilização da informática no processo de ensino-aprendizagem. Este estudo possui o intuito de definir parâmetros que possam auxiliar na tomada de decisão de gestores sobre qual a melhor configuração sobre a utilização de memória a ser adotada nestes *netbooks* utilizados em algumas escolas municipais do país.

Este artigo está organizado da seguinte forma: A Seção 2 apresenta a metodologia adotada neste estudo. A Seção 3 apresenta o Estudo Experimental, composto pelo ambiente de testes e a análise dos resultados. Por fim, a Seção 4 apresenta as Considerações Finais, com algumas conclusões e possíveis trabalhos futuros.

2. Metodologia

A análise de desempenho requer um conhecimento íntimo do sistema sendo modelado e uma escolha cuidadosa de metodologias, cargas e ferramentas de análise (JAIN, 1991). Neste sentido, a **Figura 1** exhibe o fluxograma da metodologia adotada neste trabalho.

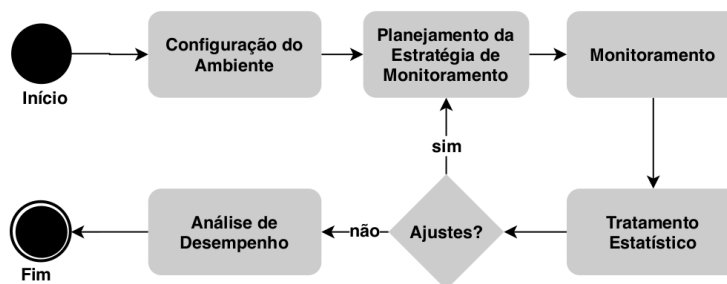


Figura 1: Metodologia adotada

Como primeiro passo, foi realizada a configuração do ambiente de testes. Inicialmente, os *netbooks* tiveram suas memórias *flash* formatadas. Após essa ação, houve a instalação do sistema edUCA. Durante o processo de instalação, as únicas configurações diferentes nos *netbooks* foram a configuração de memória, onde um deles possui memória *swap* e o outro não. O ambiente de testes foi preparado de maneira que não haja interferência de outros processos do sistema nos dados analisados. Salientamos a importância de ter um ambiente mais limpo possível para que não haja enviesamento nos testes.

A etapa seguinte, planejamento da estratégia de monitoramento, consistiu em selecionar quais métricas seriam utilizadas durante a avaliação de desempenho, e quais ferramentas seriam utilizadas para a coleta dos dados e para a geração de carga de trabalho. O monitoramento dos recursos deve ser realizado de tal forma que não interfira no resultado final, ou seja, a coleta de dados deve ser menos intrusiva possível em relação à exigência dos recursos do ambiente (JAIN, 1991). Devem-se utilizar técnicas que permitam monitorar os aspectos do sistema que se deseja observar, sem interferir em seu funcionamento (LILJA, 2000). Uma vez definido como o sistema seria monitorado, também foi definida a quantidade de amostras que seriam obtidas e quais os tratamentos estatísticos deveriam ser realizados.

Prosseguindo para a próxima etapa, o experimento foi executado e as métricas de interesse foram coletadas. A adoção de uma técnica de monitoramento adequada é muito importante, pois o monitoramento não deve ser intrusivo de modo a influenciar



nos resultados dos experimentos. Ao fim dessa etapa, arquivos de *logs* foram gerados com os dados de interesse que serviram de insumo para a próxima etapa.

Na etapa de tratamento estatístico, por sua vez, é o local no qual os dados são tratados para que seja possível aplicar técnicas estatísticas que serão utilizadas durante a análise, a exemplo da *Design of Experiments (DoE)* (MONTGOMERY, 2006). Durante a geração dessas estatísticas também é possível verificar a necessidade de ajustes durante a etapa de monitoramento, o que implicaria em ajustes na definição dos parâmetros, e execução de novos experimentos com o monitoramento adequado.

Por fim, tivemos a etapa de análise de desempenho. Nessa etapa, os resultados foram analisados a partir dos gráficos gerados na etapa anterior. Aqui é possível identificar qual estratégia de alocação de memória era a mais adequada para o propósito deste estudo.

3. Estudo Experimental

Para avaliar qual é a configuração de utilização de memória mais adequada para os *netbooks*, várias rodadas de testes foram realizadas. Esta seção apresenta as configurações adotadas no ambiente de testes e os resultados obtidos ao fim do tratamento estatístico.

3.1 Ambiente de Testes

Para a realização do estudo, o ambiente de testes foi composto por dois *netbooks* com mesma configuração de hardware: processador Intel Atom 1.6 GHz, 4GB de memória flash, 512 MB de memória DDR2. O sistema operacional utilizado foi o edUCA (kernel 4.9.0-8-amd64). O edUCA é uma personalização GNU/Linux que tem como base a distribuição Debian, sendo elaborada para ser implantada em escolas da rede municipal de ensino, mas que podem ser utilizadas em qualquer *netbook* do programa UCA por todo o país (SILVA *et al.*, 2018). Com o objetivo de ser uma distribuição leve e que atenda a especificidade destes *netbooks* presentes nas escolas municipais, a distribuição utiliza como interface gráfica o LXDE, que se trata de uma interface leve e rápida. Foram adicionados diversos softwares educacionais de acordo com os conteúdos previstos na Base Nacional Comum Curricular (BNCC), além de estarem compatíveis com os níveis de ensino e modalidade pretendidas. Também foram realizadas algumas alterações na interface gráfica, como: plano de fundo do usuário padrão, plano de fundo do Grub, excluídas as pastas Modelos e Público, reorganizadas e renomeadas no menu para os títulos e *software* corresponderem às áreas da BNCC, removidas extensões e ícones da barra inferior do LXDE (SILVA *et al.*, 2018).

A diferença entre as duas configurações testadas diz respeito à quantidade de memória e espaço em disco (na memória flash) disponível para uso. Os detalhes das configurações podem ser vistos na **Tabela 1**, que exhibe as diferenças entre as duas configurações. Na configuração #1, não há espaço de memória swap alocada para uso e, portanto, todo o espaço de disco é usado somente para o armazenamento (4 GB) e a memória total (memória RAM + swap) disponível para os softwares é de 512 MB. Já na configuração #2, há a reserva de 512 MB de espaço de disco para ser utilizada como memória swap, deixando 3,5 GB para ser utilizada para armazenamento. Nessa configuração, os softwares contarão com 1024 MB de memória total (RAM + swap).

**Tabela 1: Detalhamento UCA**

Configuração	Armazenamento	Swap	Mem. Total (RAM+Swap)
#1	4 GB	0 MB	512 MB
#2	3,5 GB	512 MB	1024 MB

Ao realizar um estudo para compreender o processo de utilização de memória em diferentes configurações, é necessário levar em conta a complexidade dos softwares que serão executados e a dinamicidade no processo de alocação/desalocação de memória, que é gerenciada pelo sistema operacional. Sendo assim, como não sabíamos qual seria o comportamento dos programas que estariam em execução ao longo da utilização dos *netbooks* pelos alunos, testamos diferentes possibilidades de tamanhos de escrita e leitura de memória (i.e., a quantidade total de memória), considerando os diferentes tamanhos de blocos (i.e., a unidade básica de memória que será lida/escrita para se alcançar a quantidade total de memória). Dessa forma, simulamos o comportamento do sistema em um mundo real, abrangendo diversas possibilidades. A **Tabela 2** exhibe os diferentes valores que foram utilizados durante a execução dos experimentos.

Tabela 2: Parâmetros de teste

Parâmetro	Valor
Tam. Total (MB)	1, 8, 32, 128, 512, 2024
Tam. do Bloco (KB)	1, 4, 8, 16, 32, 64, 128, 256, 512, 1024
Operação	Leitura, escrita

Foram realizadas combinações desses parâmetros, de forma a obter métricas para diferentes cenários de uso. Os valores de tamanho total de memória foram adotados de forma a simular aplicações que necessitam de pouca quantidade de memória (e.g., aplicações que utilizem menos de 32 MB) ou muita (e.g., aplicações que utilizem mais que 128 MB). Já para simular combinações com os diferentes tipos de dados (i.e., inteiros, *floats*, *doubles*, *chars*, *strings*, objetos, etc.), os diferentes valores de tamanhos de blocos foram adotados de forma a avaliar estruturas de dados simples (1KB) até outras mais complexas (1024 KB).

O principal indicador utilizado foi o tempo total para realização de leitura ou escrita de um espaço de memória. A escolha dessa métrica se deu para que fosse possível visualizar como as operações com diferentes tamanhos de memória podem afetar o desempenho do notebook. Nessa etapa foi adotado o *benchmark Sysbench*, que é uma suíte de testes que permite a obtenção de diversas métricas (como operações de entrada e saída em disco, CPU, etc.), dentre as quais as que são relacionadas à memória, que é o objeto deste estudo. A escolha deste *benchmark* se deu por sua simplicidade, facilidade na realização dos testes e possibilidades de ajustes de parâmetros.

Todos os testes foram executados 32 vezes, resultando em 32 amostras que foram salvas em arquivos de texto pelo *sysbench*. Scripts na linguagem *Bash* foram utilizados para extrair e formatar os dados para um padrão que pudesse ser utilizado em programas que permitam a realização de tratamento estatístico. Na etapa de tratamento dos dados extraídos, foi realizada a ordenação dos dados, o que permitiu a identificação e exclusão dos *outliers* (o menor e o maior resultado). Ao fim desta ação, 30 resultados



de cada teste foram utilizados para calcular a média, desvio padrão, variância e intervalo de confiança T de 95%.

3.2 Resultados e Discussão

Nesta subseção são apresentados os resultados obtidos a partir deste estudo experimental. A **Figura 2** exibe os resultados dos tempos obtidos para operações escrita e leitura de 1MB, 8MB e 32MB de memória.

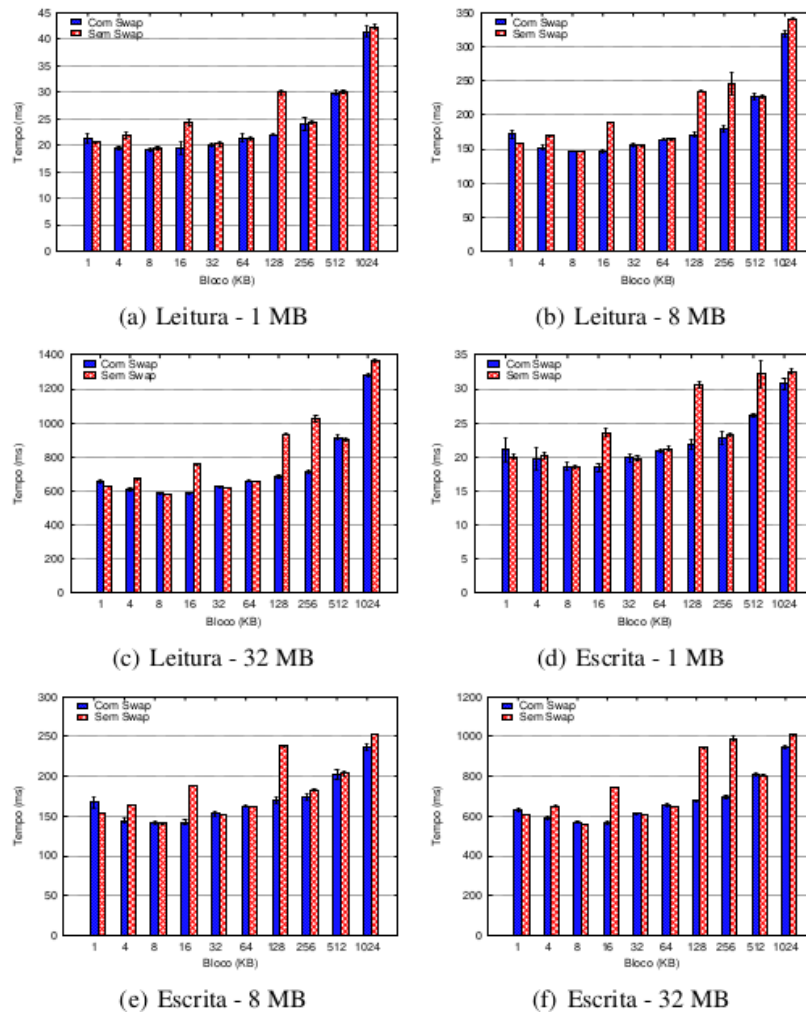


Figura 2: Operações de Leitura e escrita - pouca utilização de memória

O teste de 1MB exemplifica uma aplicação muito pequena, que ocupa um pequeno espaço em memória. Na operação de leitura (Figura 2(a)), houve diferenças significativas quando os blocos foram de 4KB (12%), 16KB (24.38%) e 128KB (36%), indicando que nesses casos o ambiente com memória *swap* se saiu melhor. Os percentuais em parênteses indicam a diferença entre as configurações. Os demais tamanhos de blocos apresentaram resultados equivalentes. Já na operação de escrita (Figura 2(d)), houve diferenças significativas quando os blocos foram de 16KB (28%), 128KB (40%) e 512KB (23%) indicando um menor tempo para escrever um total de 1MB para a Configuração #2. Para o teste com 8MB de memória, a Configuração #1 alcançou melhor desempenho quando o bloco foi de 1KB (leitura 7.6% e a escrita 8%). Já quando os

tamanhos dos blocos de leitura foram 4KB (12%), 16KB (28%), 128KB (37%), 256KB (37.7%) e 1024KB (6.7%) a configuração que possui memória *swap* obteve melhor resultado. O mesmo ocorreu quando as operações foram de escrita, 4 KB (13%), 16 KB (32%), 128 KB (39.7%), 256 KB (5%) e 1024 KB (6.8%). Os demais blocos não apresentaram variações significativas em ambas operações.

Quando avaliada a leitura ou escrita de 32MB de memória, percebemos um comportamento similar ao visto anteriormente quando avaliado um tamanho de memória de 8MB.

Até aqui foram observados os tempos de leitura e escrita para diferentes tamanhos de memória (1MB, 8MB e 32MB), que podem ser interpretadas como aplicações onde há baixa utilização desse recurso. É possível verificar que, para esses casos, quando se considera os diferentes tamanhos de blocos de memória, geralmente não é visível diferenças significativas. Quando essas diferenças existem dando vantagem à Configuração #2, o tempo absoluto para a conclusão da operação é da ordem de milissegundos, o que em uma situação de uso real talvez seja imperceptível para o usuário.

Considerando aplicações que demandem mais utilização de memória, é possível observar que essas diferenças se tornam mais relevantes. A **Figura 3** apresenta os resultados dos tempos obtidos nas operações de leitura e escrita de 128MB, 512MB e 2GB.

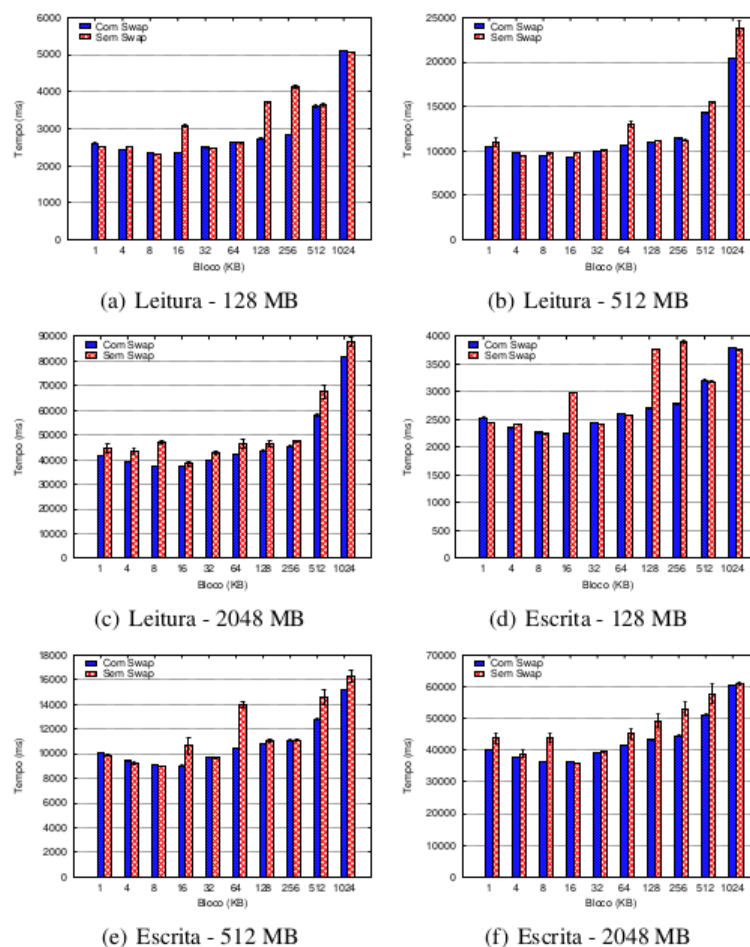


Figura 3: Operações de Leitura e escrita - alta utilização de memória

Ao se avaliar o caso de teste em que 128MB são utilizados, temos que no processo de leitura (Figura 3(a)), a Configuração #1 atingiu melhores tempos quando os blocos foram 1KB (3.5%), quando os blocos assumiram os valores 8KB, 32KB, 64KB e 1024KB a diferença de desempenho para a Configuração #2 foi de aproximadamente 1%. A Configuração #2 se saiu melhor quando os blocos foram 4KB (3.51%, 2.6%), 16KB (31%, 32%), 128KB (36%, 39.4%) e 256KB (45%, 40%), onde os percentuais em parênteses indicam a diferença para o desempenho da Configuração #1 nas operações de leitura e escrita.

Para os testes realizados com operações de leitura e escrita em uma memória total de 512 MB, a Configuração #1 se saiu melhor na operação de leitura quando os blocos foram 4 KB, 256 KB, e na operação de escrita, quando os blocos tiveram valores de 1 KB e 4 KB. Já a Configuração #2 obteve melhor desempenho quando os tamanhos dos blocos foram de 1 KB, 8 KB, 16 KB, 32 KB, 64 KB, 128 KB, 512 KB e 1024 KB, quando a operação realizada foi de leitura. Quando a operação realizada foi escrita, essa configuração se saiu melhor quando os tamanhos dos blocos foram 16 KB, 64 KB, 128 KB, 512 KB, 1024 KB.

Para os testes onde o total de memória gerenciada assume o valor de 2048MB, as diferenças mais significativas na operação de leitura foram na operação de leitura quando o bloco utilizado foi de 64KB e 1024KB, onde a Configuração #2 apresentou melhor desempenho. Já na operação de escrita, as maiores diferenças foram quando os blocos assumiram os valores de 16 KB, 64 KB, 512 KB, e 1024 KB.

Quando o tamanho da memória lida ou escrita é maior (512 MB, 2048 MB), vemos que as diferenças de tempo de leitura e escrita ficam mais evidentes. Desta forma, é mais provável que os usuários sejam mais afetados nesses cenários.

Desta forma, para que os gestores possam ter uma forma mais objetiva de visualizar quais as características que possuem maior relevância para suas decisões na adoção dos diferentes parâmetros que foram analisados, uma análise DoE foi aplicada. Ao realizar a análise, é desejável que a configuração a ser definida tenha o melhor desempenho, ou seja, o menor tempo para realizar as operações. Em ordem de prover uma melhor visualização de quais os principais fatores influenciam no desempenho da utilização de memória, um diagrama de efeitos de Pareto foi implementado, considerando um intervalo de confiança de 95% ($\alpha = 0,05$). A **Figura 4** exibe o diagrama obtido.

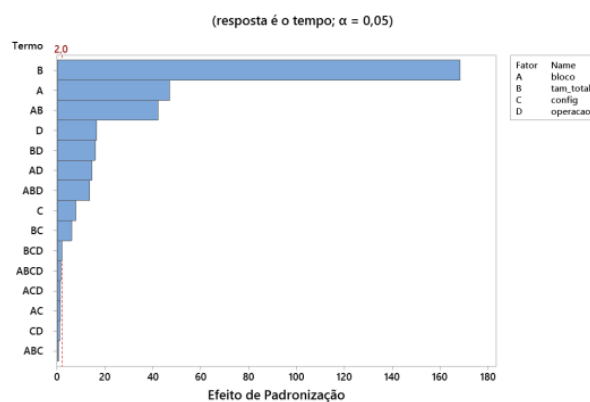
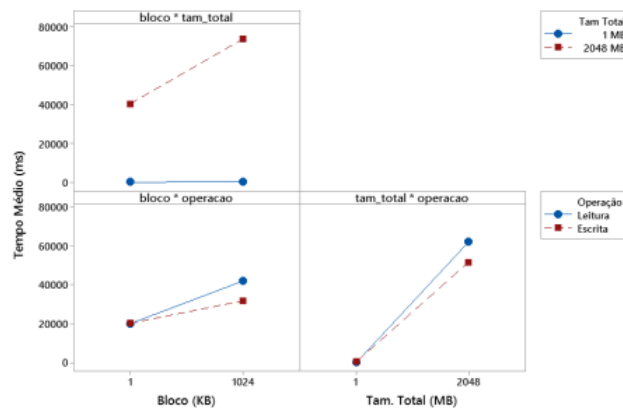


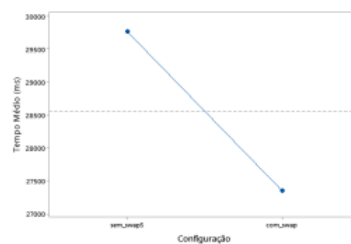
Figura 4: Pareto para os efeitos de padronização

Foram identificados que os principais fatores que estão influenciando na variável tempo de forma mais significativa são B, A, AB, D, BD, AD, ABD, C, BC e

BCD. Nas **Figuras 5(a) e 5(b)** são exibidas as interações ABD e os efeitos principais para o fator C. A partir destes gráficos é possível concluir que os melhores parâmetros utilizados para otimizar o desempenho (minimizar o tempo) são computadores com *swap* (Configuração #2), o tamanho dos blocos (1K), o tamanho total (1MB) e a operação realizada (escrita).



(a) Interação ABD para o tempo



(b) Efeitos principais

Figura 5: Interações e efeitos principais

4. Considerações Finais

A partir das experiências vividas no contexto dos projetos de extensão realizados, percebemos que muitas crianças não possuem acesso a computadores de mesa em suas casas, e raramente tem contato com *notebooks/netbooks*. Já o contato com os *smartphones*, especialmente por meio dos aparelhos de seus pais, avós ou responsáveis, é uma constante. Isto se evidenciou nas atividades desenvolvidas em uma creche, ao percebermos a quantidade de crianças que, no primeiro contato com o UCA (Programa Um Computador por Aluno), tentavam “cliquear” em elementos visuais da tela ou até movê-los usando o dedo indicador. Este indício nos colocou diante de uma série de dilemas, mas especialmente, de pensarmos um sistema que pudesse ser rápido, ágil, a exemplo da maioria dos *smartphones* que, após o toque, de modo quase instantâneo, exhibe em tela páginas, aplicativos e jogos.

Por outro lado, era preciso proporcionar aos professores acesso às mais ferramentas para a sua prática pedagógica, especialmente no ensino de conteúdos e habilidades, no contexto da educação infantil e dos anos iniciais do Ensino Fundamental, de modo a viabilizar aos estudantes instrumentos pedagógicos multimidiáticos e lúdicos visando a aprendizagem significativa de conteúdos e habilidades. Se por um lado, a configuração que utiliza memória *swap*, apresentou desempenho superior em quase todos os testes realizados, a não utilização desta partição possibilita a inclusão de mais softwares educacionais em suas versões atualizadas.



Assim, chegou-se à conclusão que as máquinas do UCA já ultrapassaram o limite de sua vida útil, sendo necessários mais investimentos para equipar as escolas públicas com equipamentos minimamente viáveis. Entretanto, enquanto esse investimento não torna-se realidade, continuaremos trabalhando com os *netbooks* UCA na configuração #2, visto que o contato com a tecnologia, mesmo em situação precária, pode trazer benefícios aos estudantes, como: O desenvolvimento da coordenação motora fina desenvolvida no trato com o *touchpad* e o teclado do *netbook*; A apropriação paulatina da linguagem alfabética pelo contato com as letras dispostas visualmente no teclado, e posteriormente, em sua materialização, também visual, na tela do *netbook*; O acesso a softwares educativos gratuitos e livres visto que há desconhecimento tanto de professores(as) e estudantes acerca da existência de tais soluções; A promoção do processo de inclusão das crianças, com a possibilidade da sua apropriação como instrumento de futura qualificação profissional, permitindo a este, as mesmas condições de acesso a tecnologia, que estudantes de redes privadas de ensino.

Referências Bibliográficas

- BOULIC, R.; RENAULT, O. 3d hierarchies for animation. In: Magnenat-Thalmann, N.; Thalmann, D. (Ed.). **New Trends in Animation and Visualization**. [S.l.]: John Wiley & Sons Ltd., 1991.
- JAIN, R. **The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling**. New York: Wiley Computer Publishing, John Wiley & Sons, Inc., 1991. ISBN 0471503363.
- LILJA, D. J. **Measuring Computer Performance: A Practitioner's Guide**. New York: Cambridge University Press, 2000. ISBN 0-521-64105-5.
- LÖBLER, M. L.; LÖBLER, L. M. B.; NISHI, J. M. Os laboratórios de informática em escolas públicas e sua relação com o desempenho escolar. **Revista RENOTE - Novas Tecnologias na Educação**, Universidade Federal do Rio Grande do Sul, v. 10, n. 3, dez 2012.
- MONTGOMERY, D. C. **Design and Analysis of Experiments**. [S.l.]: John Wiley & Sons, 2006. ISBN 0470088109.
- PEIXOTO, A. E. E J. Programa um computador por aluno: o acesso às tecnologias digitais como estratégia para a redução das desigualdades sociais. **Ensaio: Avaliação e Políticas Públicas em Educação**, v. 25, n. 95, 2017.
- SILVA, I. N. DA; SILVA, K. C. N. DA; LOTTHAMMER, K. S.; SILVA, J. B. DA SILVA BENTO DA; BILESSIMO, S. M. S. Inclusão digital em escolas públicas através de tecnologias inovadoras de baixo custo no ensino de disciplinas STEM. **Revista RENOTE - Novas Tecnologias na Educação**, Universidade Federal do Rio Grande do Sul, v. 15, n. 2, jan 2018.
- SILVA, J.; SOUZA, S.; VANDERLEI, I.; ANDRADE, M.; ALENCAR, A. Informática na educação e formação de professores(as) da rede municipal de Garanhuns-PE: Uma experiência extensionista. **International Journal Education and Teaching (PDVL)**. ISSN 2595-2498, v. 3, n. 2, p. 50 – 66, 2020.
- SILVA, J. DE N.; SOUZA, S. A. G.; VANDERLEI, I. M.; ANDRADE, M. J. P. DE; ALENCAR, A. F. DE. SofiaEdu: o desenvolvimento de uma distribuição gnu/linux educacional para os computadores das escolas públicas de Garanhuns-PE. **Congresso Nacional de Educação (CONEDU)**, 2018.