

WebJuvia: Simulador Web de Apoio ao Ensino de Gerência de Memória na Disciplina de Sistemas Operacionais

Evander Oliveira Silva
Universidade Federal do Sul e Sudeste
do Pará
evander.silva@unifesspa.edu.br

Warley Muricy Valente Junior
Universidade Federal do Sul e Sudeste
do Pará
wmvj@unifesspa.edu.br

João Victor Costa Carmona
Universidade Federal do Sul e Sudeste
do Pará
jvictor@unifesspa.edu.br

RESUMO

Durante o processo de ensino dos fundamentos e conceitos da disciplina de SO (Sistemas Operacionais), há uma série de dificuldades relacionadas a observação e análise dos eventos computacionais. Consequentemente o tipo de conteúdo utilizado nas aulas desta disciplina são, na maioria das vezes, teóricos, podendo comprometer o processo de aprendizado dos estudantes. Em contrapartida, o desenvolvimento de novos métodos de ensino mostram a importância da quebra do paradigma do ensino tradicional, trazendo para o ambiente educacional novas técnicas e ferramentas que visam auxiliar o processo ensino-aprendizagem. Partindo dessa premissa, o WebJuvia é uma aplicação web educativa desenvolvida com o propósito de simular o funcionamento dos principais algoritmos de alocação dinâmica de memória por meio de uma representação gráfica e interativa. A partir de uma pesquisa feita com vários discentes de cursos de graduação na área da computação, foi possível obter uma análise de suas experiências com a utilização de softwares educativos na disciplina de SO. Uma comparação qualitativa foi realizada do WebJuvia com as demais ferramentas relacionadas a temática, o que revelou ser uma solução promissora de apoio ao ensino do tópico de alocação de memória.

PALAVRAS-CHAVE

Sistemas Operacionais, Ensino-Aprendizagem, Simulador Web, Gerência de Memória

1 INTRODUÇÃO

A disciplina de SO (Sistemas Operacionais) está presente na grande maioria das grades curriculares dos cursos de graduação na área da computação, segundo [10] o objetivo dessa disciplina é apresentar a arquitetura de um sistema operacional, e entender como ele funciona de forma geral. A grande dificuldade no processo de ensino-aprendizagem nessa disciplina se encontra justamente na análise dos eventos computacionais, estes eventos ocorrem nas camadas mais baixas do nível hierárquico de um computador, no qual o sistema operacional pode estar interagindo com diferentes plataformas de hardware, que variam de acordo com o fabricante. O principal meio de ensino utilizado nas aulas de SO é a teoria, que nem sempre é suficiente para prover um bom entendimento aos

estudantes e nem para os motivar e engajar na disciplina, prejudicando o desempenho do processo de aprendizado.

Nesse contexto, a utilização de abordagens construtivistas de ensino visam auxiliar no processo de ensino-aprendizagem com foco principal no estudante, com um papel mais ativo no qual o mesmo busca o conhecimento, realiza a análise e reflexão, e é capaz de tirar suas próprias conclusões [6].

Essa abordagem desenvolve um espaço muito propício para a utilização de softwares educativos em sala de aula, principalmente simuladores, na qual o estudante pode programar o ambiente a ser simulado, conduzir o experimento, analisar as ações, observar o resultado e ter uma reflexão sobre a sua experiência. Quando o aluno consegue relacionar essa experiência com a teoria, toda essa informação fica mais simples de ser compreendida, o que colabora significativamente em um ganho no processo de aprendizagem. A área de SO ainda possui uma grande carência de simuladores, e essa situação é ainda mais agravada quando se observa a escassez de softwares educativos acessíveis aos estudantes [11].

Considerando os desafios expostos nos parágrafos anteriores, o objetivo desse artigo é apresentar o WebJuvia, um simulador web que visa auxiliar no processo de aprendizagem sobre gerência de memória na disciplina de Sistemas Operacionais, uma análise e comparação com trabalhos similares, e uma pesquisa feita com estudantes da área de computação sobre suas experiências com a utilização de softwares educativos nesta disciplina.

A estrutura deste artigo segue a seguinte ordem: uma apresentação de trabalhos relacionados e uma análise comparativa com a proposta na seção 2; uma pesquisa feita com os estudantes da área da computação sobre suas experiências com o uso de softwares educativos nas aulas de gerência de memória na disciplina de SO na seção 3; uma apresentação do WebJuvia como uma proposta de software educativo, sua aplicação no contexto de sala de aula invertida e a disponibilização da ferramenta na seção 4; detalhes sobre o desenvolvimento do software na seção 5; e a conclusão do trabalho na seção 6.

2 TRABALHOS RELACIONADOS

Esta seção tem como objetivo apresentar os trabalhos relacionados à proposta. Tais trabalhos serviram de contribuição para identificar os problemas e os requisitos mais genéricos. Estes trabalhos enfatizam as principais características que se mostram relevantes em um sistema educativo cujo o objetivo é prover o auxílio no aprendizado sobre gerência de memória.

O SOSim [9] e o SSOG [7] auxiliam no aprendizado apresentando conceitos de sistemas operacionais como a gerência de processos, gerência do processador e gerência de memória por meio de uma interface gráfica composta por janelas separadas para cada módulo. O

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'21, Abril 27–30, 2021, Jataí, Goiás, Brasil (On-line)

© 2021 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

SOsim particularmente é muito popular no meio acadêmico por ser complementar ao livro "Arquitetura de Sistemas Operacionais" [8], e sendo referência na maioria dos trabalhos relacionados a este tema. Por mais didático que estes sistemas possam ser, seus módulos de gerência de memória se apresentam apenas como uma visualização gráfica das consequências da manipulação do módulo de gerência de processos, oferecendo um acesso indireto e limitado, o que não deixa de condizer com a verdade, mas não disponibiliza outros meios de acesso para caso o discente queira focar apenas no módulo de gerenciamento de memória.

O TBC-SO/WEB [15], desenvolvido na linguagem de programação Java (*applets*), tem como objetivo auxiliar no aprendizado dos conceitos de gerência de memória e gerência de processador apresentando uma interface gráfica interativa. Por ter sido construído em Java e implementado na plataforma web por meio de *applets*, a própria tecnologia apresenta suas limitações, principalmente por possuir pouco espaço para processamento, tornando o carregamento e execução lentos, além de que o sistema não pode ter acesso a outros componentes utilitários do navegador, e requerer a máquina virtual Java para funcionar. Esta é semelhante ao OS Simulator [5], também desenvolvido em Java, mas que possui como objetivo principal, auxiliar no aprendizado mais especificamente de sistemas de arquivos.

A proposta do artigo [16], apresenta dois sistemas independentes, porém com interfaces gráficas muito semelhantes, um deles com o objetivo de simular o gerenciamento de processador, enquanto o outro simula o gerenciamento de memória. Esta abordagem dividida trás muita flexibilidade para o estudante, permitindo focar no que realmente agrega valor na hora de aprender, porém ambos os módulos (gerência de processador e de memória) poderiam estar disponíveis no mesmo sistema, e possuir execuções independentes para que o valor desse particionamento das funcionalidades não se perdesse.

O SigemVR [14] apresenta uma proposta nova entre os demais simuladores, pois consiste em um sistema de realidade virtual podendo ser acessado por meio de óculos de realidade virtual. Este dispõe de um módulo de gerenciamento de memória. Por mais que a tecnologia da realidade virtual esteja alcançando cada vez mais espaço na indústria e academia, ainda se trata de uma tecnologia com recursos muito inacessíveis para a maioria das instituições de ensino e consequentemente para estudantes.

O MOSS [12] é uma coleção de programas que apresentam diversos conceitos de sistema operacional. Este sistema possui uma interface pouco interativa ou comunicativa, com sérias falhas de usabilidade. O sistema é pouco didático e não oferece grande ajuda aos estudantes iniciantes na área de sistemas operacionais. Por outro lado, o VM Simulator [4] é um sistema inacabado que trás a proposta de simular as atividades presentes na memória virtual por meio de componentes gráficos. Este tem como objetivo apresentar a variabilidade de resultados na gerência de memória de acordo com o conceito utilizado. O autor dessa solução apresenta esse software como uma promessa de funcionalidades incluindo algumas prototipações gráficas sobre a gerência de memória, porém nada definido, portanto não há muitos detalhes a serem avaliados.

2.1 Análise Comparativa

Todos os trabalhos anteriormente mencionados foram analisados e comparados com o WebJuvia. O resultado é apresentado na Tabela 1 a seguir, levando em consideração as seguintes características:

- Objetivo geral (C1): define a principal contribuição da solução quanto aos tópicos de estudo na disciplina de SO.
- Plataforma de execução (C2): representa o ambiente e/ou tecnologia utilizada para execução da solução.
- Algoritmos de alocação (C3): soluções que implementam os principais algoritmos de alocação de memória: First-Fit, Worst-Fit, Best-Fit e Next-Fit.
- Apresentação do conceito teórico explicativo (C4): define se a solução possui descrição teórica do tópico de estudo.
- Responsividade e acessibilidade (C5): refere-se ao uso de tecnologias que permitem a adaptação da ferramenta em diferentes resoluções de telas e funções que tornam estas ferramentas acessíveis para discentes com alguma deficiência.

De acordo com a característica C1, as soluções propostas por [7, 9, 15, 16] visam o ensino abrangente de tópicos relacionados à gerenciamento de processos e memória, o que permite tanto ao professor quanto ao discente uma dinâmica de estudo de modo a correlacionar e compreender estes conceitos de forma mais clara. Porém, nestes casos, o tópico de gerência de memória é abordado de forma mais passiva, no qual a memória e seus recursos, se apresentam como uma consequência da utilização dos módulos de gerência de processos, isso inclui a ausência de algumas funcionalidades de manipulação do estado da memória que podem ser encontradas nos trabalhos citados. O [9], por exemplo, não informa ao usuário qual algoritmo de alocação está sendo utilizado, pois este trabalho não trata a alocação dinâmica de memória como uma funcionalidade. Outro exemplo é o [15], no qual o usuário não possui um controle do tempo em que o processo vai permanecer na memória.

Similar ao WebJuvia, as propostas de [4, 14] focam no ensino de gestão de memória, buscando apresentar por meio de recursos gráficos uma representação da memória e suas alterações a partir das ações do usuário. Porém, estes trabalhos não contemplam as funcionalidades de alocação dinâmica de memória, e nem citam se, e qual algoritmo está sendo utilizado no sistema. Além disso, o [4] se trata de um trabalho incompleto.

No que se refere a característica C2, a grande maioria das ferramentas estão disponíveis como arquivos executáveis a serem instalados no disco rígido de computadores desktops. Porém, este modo de execução exige constantes atualizações para garantir compatibilidade com o versionamento de sistemas operacionais modernos, o que dificulta a escalabilidade de instalação, uma vez que esta limitado a computadores, recurso este não presente na casa de maioria dos discentes de baixa renda. Para contornar as limitações anteriormente mencionadas, o WebJuvia é a única ferramenta projetada e desenvolvida para execução diretamente no navegador web, cujo tópico de ensino está relacionado à gerencia de memória. Por estar disponível na web, o WebJuvia pode ser acessado independente de plataforma e recurso computacional (computador, smartphone, tablet).

A característica C3 trata dos algoritmos de alocação dinâmica de memória utilizados durante a inserção dos processos. Na maioria dos trabalhos apresentados, esse assunto não é abordado ou sequer

Tabela 1: Tabela comparativa dos trabalhos relacionados com a proposta.

Solução	C1	C2	C3	C4	C5
[9]	Gerência de processos e de memória	Desktop	Next-Fit	X	X
[15]	Gerência de processos e de memória	Applet	First-Fit, Worst-Fit, Best-Fit e Next-Fit	✓	X
[16]	Gerência de processos e de memória	Desktop	First-Fit, Worst-Fit e Best-Fit	X	X
[14]	Gerência de memória	Head Mounted Display	Não informado	X	X
[7]	Gerência de processos e de memória	Desktop	Não informado	X	X
[12]	Sistema de arquivos e Gerência de Memória	Desktop	Não informado	X	X
[5]	Sistema de arquivos	Web	First-Fit, Worst-Fit e Best-Fit	X	X
[4]	Gerência de memória	Desktop	Não informado	X	X
WebJuvia	Gerência de memória	Web	First-Fit, Worst-Fit, Best-Fit e Next-Fit	✓	✓

considerado como uma funcionalidade. Dentre os que abordam, o [15] é o único que abrange todos os quatro principais algoritmos encontrados na literatura.

A característica C4 mostra que a maioria dos trabalhos relacionados não busca apresentar ao usuário a teoria em que o software se baseia, com exceção do [15] e do WebJuvia.

Por fim, de acordo com a característica C5, nenhum trabalho relacionado se preocupa em trazer recursos de responsividade e acessibilidade, tais características são essenciais para buscar a democratização do acesso à tecnologias educativas. Característica esta presente no WebJuvia.

Em resumo, por meio da tabela 1, é possível identificar características similares e diferentes do WebJuvia em relação as demais propostas. Portanto podemos destacar que o WebJuvia é o único simulador disponível na web capaz de auxiliar o aprendizado de gerência de memória através dos quatro principais algoritmos de alocação dinâmica de memória, com alta capacidade interação, apresentação de conceitos e suporte à acessibilidade para um público maior de estudantes de computação.

3 PESQUISA DE SATISFAÇÃO

Para motivar o desenvolvimento do WebJuvia, foi realizada uma pesquisa no formato de questionário aos alunos de graduação dos cursos de Ciência da Computação, Sistemas de Informação e Engenharia da Computação de diferentes universidades federais que possuem a disciplina de sistemas operacionais na grade curricular do curso. Este questionário foi amplamente divulgado na lista de e-mails institucionais durante um período de 4 meses, porém poucos alunos manifestaram interesse em respondê-lo. A seguir, as principais perguntas relacionadas às ferramentas disponíveis e utilizadas em conjunto com o nível de satisfação. Todo o questionário foi elaborado conforme recomendação de [1] para planejamento de avaliação de IHC (Interação Humano-Computador) que envolve uso de questionário para investigar alternativas de *design*, problemas de usabilidade, experiências e expectativas.

- Foi utilizado algum software educativo para o auxílio às aulas de gerenciamento de memória (Q1)?
- Qual software foi utilizado (Q2)?
- O quão útil você considera esse software (Q3)?
- O quão intuitivo você considera esse software (Q4)?
- O quão fácil você considera a utilização desse software (Q5)?
- O quão bem esse software realiza a função a qual se propõe fazer (Q6)?

O resultado é apresentado na Figura 1. Para as Figuras 1c, 1d, 1e, e 1f, a escala de pontuação varia entre 1 e 5, sendo que 1 representa

menor nível de satisfação e 5 maior nível de satisfação. O total de 36 alunos responderam o questionário, sendo que 8 alunos indicaram não terem iniciado ou concluído a disciplina de Sistemas Operacionais. Portanto, os 28 restantes responderam às perguntas listadas acima. Na pergunta Q1, conforme a Figura 1a, 15 alunos (correspondente a 53,6%), responderam não terem utilizado nenhuma ferramenta de apoio ao ensino do tópico de gerência de memória, enquanto 13 alunos responderam já terem tido experiência com alguma ferramenta de ensino para este tópico específico.

Desse conjunto de 13 alunos, 100% fizeram uso da ferramenta SOSim [9], por ser um simulador recomendado pelo autor do livro referência da disciplina de Sistemas Operacionais [10] da maioria das universidades. As demais ferramentas, MOSS [12] e TBC-SO/WEB [15], estão rotuladas no gráfico somente para destacar o fato de estarem disponíveis na web para uso, porém pelo resultado da pesquisa, nenhuma destas foram adotadas.

Nas Figuras 1c e 1f são apresentadas as respostas para as perguntas Q3 e Q6, respectivamente. Estas perguntas visam avaliar a experiência do aluno quanto a utilização do software. Podemos observar que em ambos os casos, houve maior concentração do número de discentes entre os níveis de satisfação 3 e 4, o que revela um nível de experiência razoável do uso do SOSim, indicando que mesmo sendo esta uma ferramenta recomendada, ainda carece de melhorias. As Figuras 1d e 1e ilustram os resultados para as perguntas Q4 e Q5, respectivamente. Estas perguntas visam avaliar a usabilidade e comunicabilidade do software. Os resultados indicam que houve uma concentração alta do número de discentes para os níveis de satisfação 1 até o 3 (veja a Figura 1d) e de 2 até 4 (veja a Figura 1e), ou seja, a usabilidade do SOSim é baixa e deficiente para com o propósito do simulador.

Portanto, mediante os resultados apresentados, é importante a proposta de novas soluções inovadoras que resolvam as problemáticas que foram reveladas após a aplicação deste questionário.

4 WEBJUVIA

O WebJuvia é um simulador web de apoio às aulas de gerência de memória na disciplina de sistemas operacionais. Esse software possui como objetivo auxiliar o processo de ensino-aprendizagem dos principais algoritmos de alocação dinâmica de memória por meio de um sistema simples e intuitivo que funciona nos principais navegadores web, incluindo dispositivos móveis. Outro detalhe é que o WebJuvia funciona totalmente do lado do cliente, ou seja, após o *download* da página, o sistema não depende mais de um servidor web para funcionar, e essa é uma característica muito significativa para os usuários que possuem uma baixa qualidade de internet

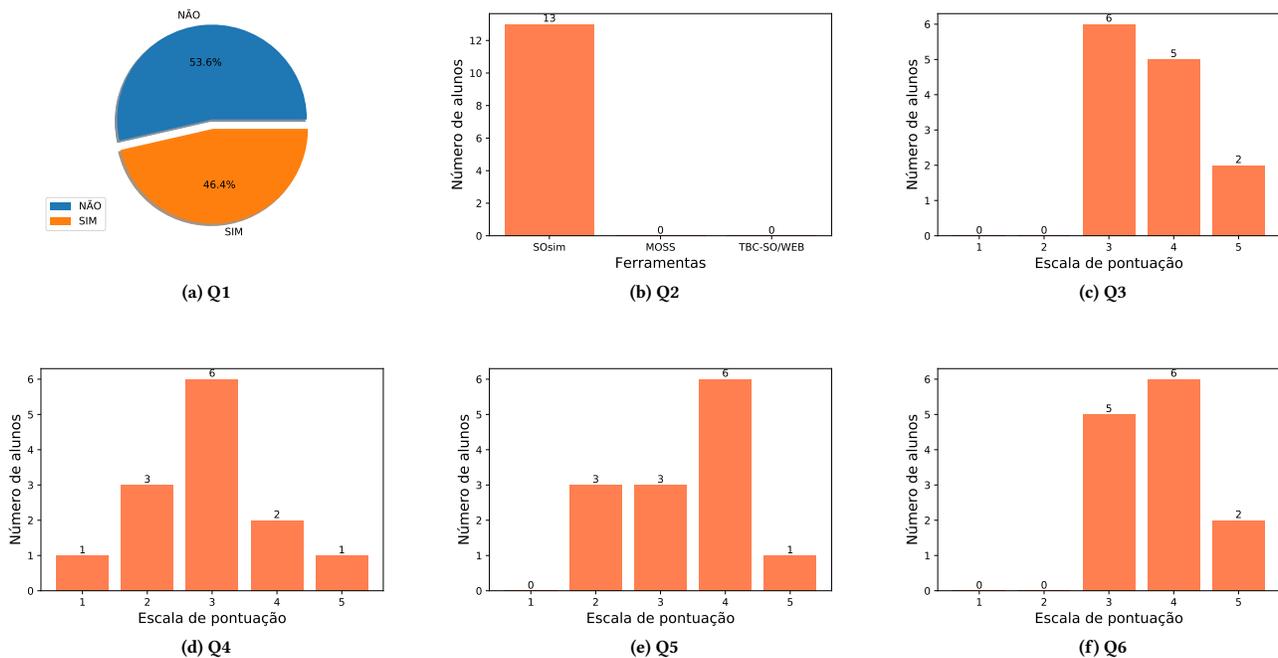


Figura 1: Resultado da pesquisa de satisfação de IHC.

e/ou conexões instáveis. Este simulador foi desenvolvido utilizando a linguagem de programação Javascript por meio do framework Vue.js, e sua interface foi construída com HTML5 e CSS3.

O WebJuvia provê ao usuário uma simulação, por meio de recursos gráficos e com animação, do funcionamento dos principais algoritmos de alocação de memória presentes na literatura da disciplina de sistemas operacionais: First-Fit, Best-Fit, Worst-Fit e Next-Fit. E ele permite ter um controle direto dos processos na memória, incluindo sua ordem de inserção e exclusão, e o tempo que cada processo permanece na memória.

A Figura 2 apresenta a interface do WebJuvia em tela de computador desktop e em tela para dispositivos móveis. Os componentes presentes na interface são:

- (1) Adição de processos: este componente possui os principais meios de entrada de informações no sistema, tais como, o tamanho do processo e o algoritmo de alocação de memória.
- (2) Processos em memória: este componente lista todos os processos presentes na memória, incluindo algumas das suas informações, junto com o botão para excluir cada processo.
- (3) Opções: este componente disponibiliza algumas funcionalidades do sistema, como o recurso de animação, o botão de reiniciar a memória, o acesso à lista de comandos pelo teclado, e a escolha do tema do sistema.
- (4) Memória principal: este componente representa o estado atual da memória, ele é composto por 100 (cem) unidades, representando cada kilobyte presente na memória.
- (5) Memória conceitual: assim como a memória principal, este componente também representa o estado atual da memória, porém com informações diferentes, este componente possui

o desenho similar às ilustrações da memória apresentadas na literatura da disciplina de sistemas operacionais, seu objetivo é auxiliar o estudante a criar a relação entre o uso da ferramenta com o conteúdo teórico abordado nas aulas, além de prover um panorama mais simples e direto do estado atual da memória.

A principal informação contida neste sistema é o processo. Os processos são manipulados pelos algoritmos de alocação de memória, e são eles que são exibidos na memória principal e na lista de processos. Por meio da interface do WebJuvia, o usuário pode ter acesso à algumas informações sobre os processos. No componente Processos em Memória é possível identificar o processo por meio do emoji, que é o recurso gráfico utilizado para representá-lo. Esses caracteres são selecionados aleatoriamente pelo sistema durante cada inserção. Neste mesmo componente também é possível encontrar o tamanho do processo em kilobytes, e o algoritmo de alocação responsável por sua inserção. Na memória principal também é possível identificá-lo por meio dos emojis, mas além disso, também pode-se observar onde cada processo começa e onde termina a partir da numeração das unidades. Apesar de não ser possível identificá-los com facilidade na memória conceitual, é possível observar seus tamanhos e proporções a partir de uma visão panorâmica da memória.

Outra informação fundamental contida nesse sistema são as lacunas. As lacunas nada mais são que os espaços vazios da memória, ou seja, o conjunto de unidades sequenciais que não estão sendo ocupadas por processos em um determinado momento. Suas localizações e proporções são algumas das principais informações analisadas pelos algoritmos de alocação de memória, para que os mesmos

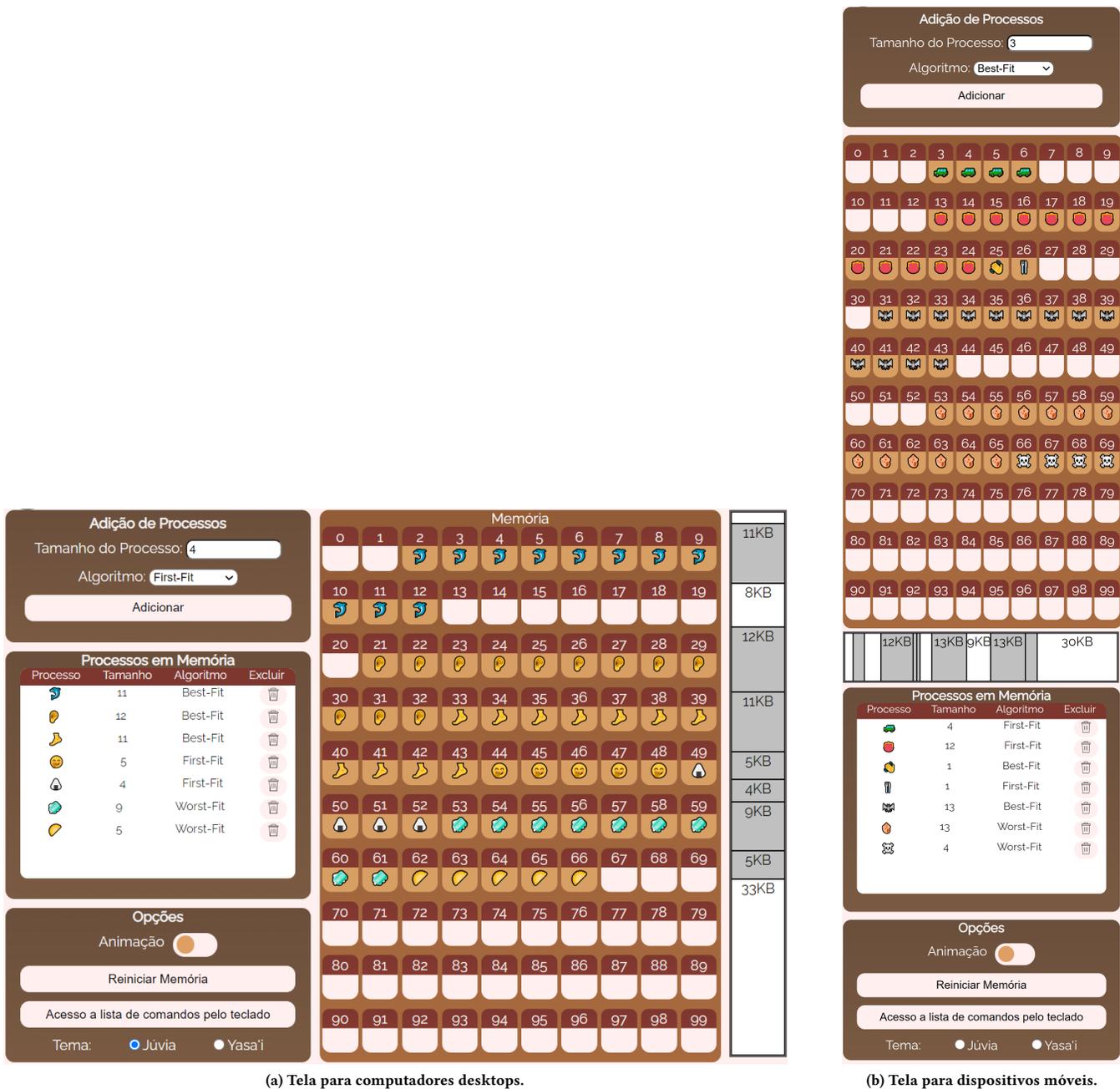


Figura 2: Tela do WebJuvia para desktop e mobile.

possam tomar sua decisão. Na interface do WebJuvia, as lacunas podem ser identificadas na memória principal como o conjunto de unidades próximas que não contém um emoji, e que não possuem seu interior colorido com a cor marrom. Na memória conceitual, as lacunas podem ser identificadas nas partições que não possuem a cor cinza, aliás, a memória conceitual é o melhor componente para analisar as lacunas, pois este é o componente que exibe suas

proporções de forma simples e direta, além de proporcionar ao usuário, toda a sua extensão sem “quebras de linha”, como geralmente acontece na memória principal. Saber analisar os processos e lacunas é uma habilidade crucial para tirar o máximo proveito da utilização do WebJuvia como ferramenta educativa, e possuir mais de um componente para prover essa análise é um fator que tem como objetivo tornar sua utilização mais eficiente.

O discente pode começar a utilizar o sistema de diferentes modos, ele pode começar adicionando algum processo, removendo algum processo que já esteja inserido, ou reiniciando a memória. Ao acessar o WebJuvia, alguns processos aleatórios, de tamanhos e algoritmos de alocação aleatórios já estão presentes na memória. Essa funcionalidade visa agilizar a utilização da ferramenta, pois, em vez de dispersar tempo inserindo e removendo processos, com o fim de obter lacunas de tamanhos diferentes em vários espaços da memória, esse cenário já está pronto assim que o sistema é inicializado, e consequentemente o usuário já pode iniciar a utilização do sistema observando as diferenças no comportamento de cada algoritmo de alocação de memória. Essa funcionalidade tem como objetivo agilizar o aproveitamento da ferramenta, e proporcionar o ambiente propício para o aprendizado de forma instantânea, mas caso o usuário queira começar a utilizar o sistema “do zero”, ele pode utilizar a funcionalidade de reiniciar a memória, que irá prover este ambiente desejado pelo usuário.

A Figura 3 apresenta o fluxo durante a adição de um processo. As informações iniciais do processo são inseridas no componente de adição de processos pelo usuário, e o restante da inserção fica por conta do sistema. Após clicar no botão adicionar, o sistema realiza uma verificação de validação do tamanho do processo, o valor inserido deve ser maior que zero, e menor ou igual a 100. Após essa verificação, o sistema analisa se há espaço na memória para a alocação do processo do tamanho especificado, e depois verifica se o algoritmo utilizado é o next-fit, pois neste caso é necessário haver um último processo inserido para que o algoritmo funcione. Após esses procedimentos, o algoritmo de alocação selecionado pelo usuário faz uma análise das lacunas da memória, e decide onde o processo deve ser armazenado. Essas alterações são refletidas no componente de processos em memória, memória principal e memória conceitual. Caso alguma das validações guie o fluxo para uma exceção, uma mensagem explicativa é exibida ao usuário e a inserção não é realizada.

O WebJuvia disponibiliza o recurso de animação, presente no componente de opções, que anima as unidades presentes na memória principal criando uma analogia do comportamento dos algoritmos de alocação. Essa funcionalidade é opcional, pois as animações podem levar alguns segundos para serem concluídas, então caso o aluno deseje que as adições de processos sejam refletidas instantaneamente na memória, ele pode desativar esse recurso. O recurso de animação não altera de forma alguma no processo de adição de processos apresentado na Figura 2, seu efeito é apenas estético, e possui a finalidade de auxiliar o usuário na análise do comportamento dos algoritmos de alocação de memória.

O *design* da interface foi desenvolvido seguindo algumas diretrizes apresentadas por [1], no qual o mesmo exemplifica como uma boa prática, a utilização de diferentes elementos em uma interface, organizados em conjuntos que representem suas finalidades, levando o usuário em direção ao seu objetivo com a utilização da ferramenta. As legendas apresentadas em cada componente tem como finalidade apresentar o objetivo de seus elementos. Alguns guias de estilo apresentado pelo autor também estão presentes na interface do WebJuvia, como a padronização dos botões e legendas e o uso de simbologias no *design* de ícones.

A memória principal possui um estilo mais familiar comparado aos trabalhos relacionados, com o conceito de *slots* ou unidades que

representam o tamanho da memória e a proporção dos processos, mas a forma como esses processos são representados na mesma é uma característica única do WebJuvia. O uso dos emojis como uma forma de representar os processos se trata de uma estratégia para aproximar a ferramenta do usuário final, como apresentado por [13], a interação no ciberespaço passa a ser mais afetiva com a utilização desses recursos, apesar da finalidade nessa ferramenta não ser de comunicação ou expressão, o reconhecimento desses caracteres tão populares, e a imprevisibilidade no qual eles são apresentados no WebJuvia, pode criar uma sensação de curiosidade ao usuário, tornando sua utilização mais interessante.

O design utilizado na memória conceitual é fundamentado nas ilustrações da memória apresentadas na literatura da disciplina de SO, como nos livros de [8] ou/e [17]. A tela para dispositivos móveis possui os componentes organizados seguindo a lógica da utilização da ferramenta, como já apresentado nesta seção. As cores utilizadas na interface do sistema são baseadas nas cores da Júvia (Castanha-do-Pará ou Castanha-do-Brasil), essa paleta de cores foi escolhida para gerar uma identidade brasileira para o sistema.

4.1 Aplicação Educacional

A disciplina de Sistemas Operacionais faz parte da grade curricular de grande maioria dos cursos de computação, por ser fundamental para a formação profissional dos discentes. A emenda desta disciplina aborda diversos conceitos e técnicas um tanto complexas, principalmente no que diz respeito a alocação dinâmica de memória. Tópico este, de difícil compreensão quando adota-se o modelo tradicional de ensino em sala de aula, no qual o professor segue sistematicamente a bibliografia e explica a teoria por meio de uma aula expositiva e com pouca interação.

Desta forma, o WebJuvia torna-se uma ferramenta de apoio ao processo de ensino-aprendizagem de gerência de memória que pode facilmente adequar-se às principais metodologias ativas de ensino, tais como sala de aula invertida [3] e aprendizado baseado em problemas [2].

Como exemplo de aplicação do WebJuvia, segue um passo-a-passo dentro do contexto de sala de aula invertida:

- (1) Antes da aula.
 - Passo 1: o professor disponibiliza material teórico de livro-texto da disciplina, vídeo aula, e conceitos sobre os algoritmos de alocação dinâmica de memória disponível no menu *Algoritmos* do WebJuvia.
 - Passo 2: o professor aplica uma atividade prática com o WebJuvia, cujo o desafio da utilização dos algoritmos é aproveitar a memória o máximo possível. Perguntas sugestivas podem ser feitas, tais como, qual algoritmo é o melhor para se utilizar na memória? E em processos pequenos, de 4 ou 6kb? E em processos maiores, como de 16 a 20kb? E levando em consideração a o processamento, qual algoritmo consome menos recurso, qual é o mais rápido? A rapidez do algoritmo compensa seu desempenho no aproveitamento da memória?.
- (2) Durante a aula.
 - Passo 3: o professor revisa os conceitos de gerência de memória já estudados e praticados pelos alunos. Em seguida, deixa em aberto um debate para que cada aluno defenda

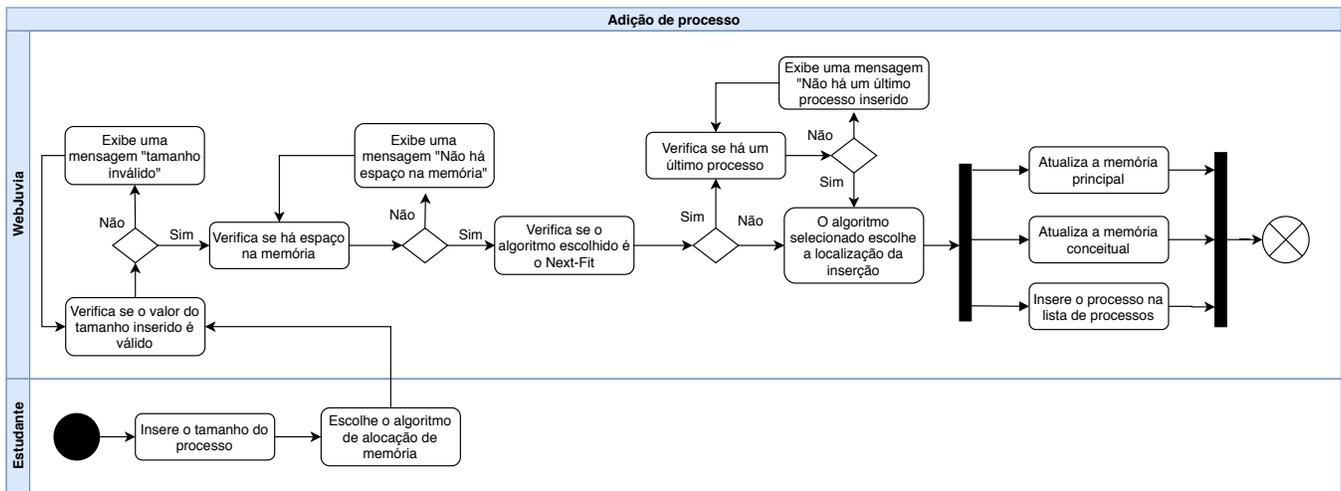


Figura 3: Fluxo de atividades do WebJuvia.

o comportamento de um algoritmo específico baseado na atividade prática aplicada antes da aula.

- Passo 4: cada aluno defende o comportamento de um algoritmo baseado na experiência que teve no uso do WebJuvia, ilustrando seu argumento a partir da tela da ferramenta.

4.2 Disponibilização da Ferramenta

O WebJuvia encontra-se disponível no link ¹. Por meio deste link é possível utilizar o sistema tanto na versão desktop quanto mobile, além de ter acesso à informações sobre os algoritmos de alocação de memória utilizados e informações gerais sobre o software. Um vídeo apresentando a ferramenta e explicando o seu funcionamento de forma prática também está disponível na página "Sobre" do site, mas também pode ser acessado por meio do link ². Neste vídeo é apresentado cada componente que compõe a interface do sistema, as informações básicas para a sua utilização, e uma explicação das suas principais funcionalidades.

O WebJuvia é uma ferramenta disponibilizada gratuitamente na internet, mas seu código-fonte é de caráter privado, pois o mesmo ainda se encontra em processo de melhoria e de implementação de novas funcionalidades.

5 DETALHES DE DESENVOLVIMENTO

Para representar a estrutura e os relacionamentos das classes do sistema, foi contruído o diagrama de classes da Figura 4.

Apesar da representação gráfica, a memória de fato está sendo representada em dois vetores diferentes, o array de processos, que possui todos os processos que estão presentes na memória, e o array de lacunas, que possui todas as lacunas da memória que não estão sendo ocupadas por processos. Esses dois vetores estão inseridos no Controller, a entidade principal do sistema, que controla os componentes do sistema e possui todas as informações necessárias para o seu funcionamento. É no Controller que as

funcionalidades principais do sistema são executadas, como o método adicionarProcesso que, a partir do tamanho do processo e o algoritmo de alocação de memória escolhidos pelo usuário, envia essas informações para o algoritmo correspondente para que o mesmo decida qual espaço da memória o processo deve ocupar. Outro método essencial para o sistema é o removerProcesso, que, como o nome já diz, remove do array de processos, o processo escolhido pelo usuário.

Além dessas funcionalidades principais, existem outros atributos e métodos implementados para garantir um bom funcionamento do sistema, como, ainda no Controller, o atributo last_process, que contém o último processo inserido, essa informação é útil para o funcionamento do algoritmo Next-Fit.

Na entidade Serviços há o método encontrarLacunas, este método também é fundamental para o funcionamento do sistema, uma vez que é ele que preenche o array de lacunas que se encontra no Controller, ele faz isso a partir da comparação entre a distância que os processos se encontram do início e do final da memória, e entre eles mesmos. Toda as vezes que um processo é adicionado ou removido, este método entra em ação, pois os algoritmos de alocação de memória precisam do array de lacunas atualizado para funcionar. Ainda em Serviços o método isLacuna auxilia o método encontrarLacunas para decidir se o espaço entre os processos são lacunas ou não. Para que o encontrarLacunas possa funcionar da maneira correta, o array de processos é sempre ordenado no método sorting, dessa forma o array "processos" está sempre organizado de acordo com a posição em que cada processo se encontra na memória. Esses métodos visam garantir a organização das estruturas de dados, para que os outros algoritmos possam ser implementados e executados sem uma grande complexidade.

A partir disso, o ambiente está pronto para o funcionamento dos algoritmos de alocação de memória. Na entidade Algoritmos se encontram todos os algoritmos que o sistema pode simular, o FirstFit, BestFit, WorstFit e o NextFit. Todos eles funcionam de acordo como é apresentado na literatura, a partir do tamanho do processo (e da posição do último processo alocado, no caso do

¹<https://modest-cray-5bf6fc.netlify.app/>

²<https://youtu.be/dkb4OqzMsJg>

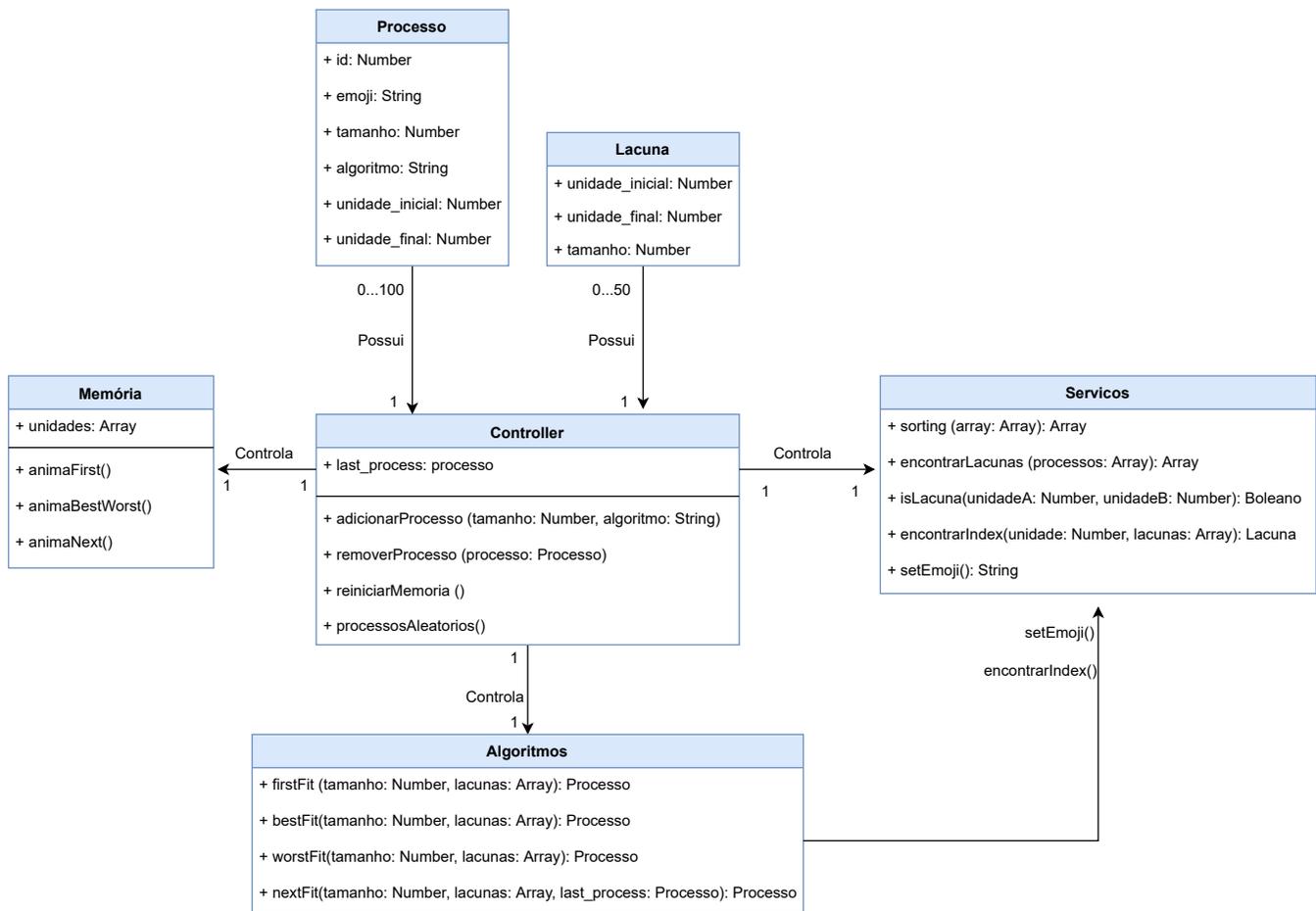


Figura 4: Diagrama de classes do WebJuvia.

Next-Fit), verificam as lacunas disponíveis na memória e decidem onde o processo vai estar, no caso do First-Fit é na primeira lacuna que couber o processo, o Best-Fit escolhe a lacuna que menos desperdiçar espaço, ao contrário do Worst-Fit que escolhe a que mais desperdiça. O Next-Fit funciona de forma muito similar ao First-Fit, a diferença é que ele começa a procurar a partir da localização do último processo inserido, e, para saber em qual lacuna ele deve começar a procurar, o método `encontrarIndex`, na entidade Serviços, retorna a lacuna em questão. Além de retornar um processo, algo que todos esses algoritmos possuem em comum é a utilização do método `setEmoji`, na entidade Serviços, esse método retorna um emoji aleatório para representar o processo graficamente na interface do sistema. Esses métodos e atributos já são o suficiente para o sistema funcionar plenamente, mas ainda existem outras funcionalidades que buscam prover uma experiência ainda melhor para o usuário.

No Controller, o método `reiniciarMemoria` “zera” todo o sistema, desde os vetores de processos e lacunas até a forma como a memória está sendo representada na interface do sistema, esse método é útil quando se tem vários processos na memória e é necessário remover todos, ou simplesmente para começar a usar o

sistema com a memória vazia. A diferença deste método para a ação de usar o método `removerProcesso` em todos os processos, é que, ao reiniciar a memória, o atributo `last_process` também é reiniciado, enquanto, quando simplesmente os processos são removidos, este atributo ainda irá armazenar as informações do último processo inserido. Este método também se difere de atualizar a página por dois motivos: primeiro que, reiniciar a memória não requer o *download* e renderização da página web novamente, e, em segundo, quando a página web é baixada ou atualizada a memória não está vazia, ela já possui alguns processos inseridos graças ao método `processosAleatorios`. Ainda no Controller, o método `processosAleatorios` insere alguns processos de tamanhos aleatórios na memória, de forma que, quando o usuário possui contato com o sistema, o simulador já apresenta um ambiente propício para diferenciar o comportamento dos algoritmos de alocação de memória, dispensando a necessidade de o usuário inserir e remover vários processos afim de obter lacunas diferentes. Além desses métodos, o sistema ainda oferece outras funcionalidades que buscam prover uma experiência visual do funcionamento dos algoritmos de alocação de memória, e como eles afetam o estado da memória. Na entidade Memória há o atributo “unidades”, este atributo contém

todos os elementos HTML que representam as unidades da memória, quando um processo é adicionado ou removido, é nesse array que os processos e lacunas são refletidos, exemplificando para o usuário como suas ações na utilização do sistema impactaram no estado da memória.

Ainda na Memória, existem os recursos de animação do sistema, estes são: animaFirst, animaBestWorst e o animaNext. Estes métodos são responsáveis por aplicar efeitos de animação nos elementos da interface para criar uma breve analogia sobre como os algoritmos percorrem a memória e como eles fazem sua tomada de decisão.

6 CONCLUSÃO

Nesse artigo foi apresentado o desenvolvimento, arquitetura e implementação do WebJuvia, um simulador que pode ser útil no auxílio nas aulas de gerência de memória, possibilitando que os alunos obtenham um conhecimento mais prático sobre a disciplina. Por meio da pesquisa feita com os diversos discentes de cursos da área de computação, pode-se concluir uma real necessidade de alternativas de ferramentas educativas para a utilização nas aulas de SO, pois, segundo os resultados obtidos, a utilização de softwares educativos nas aulas dessa disciplina não é uma realidade para grande parte dos estudantes, e nos casos de utilização, há uma baixa qualidade no nível de usabilidade. Essa situação é ainda mais acentuada quando se analisa a disponibilidade das ferramentas desenvolvidas, pois ainda que existam poucas opções deste tipo de software, a maioria delas não está disponível na internet, impossibilitando o seu acesso. Por meio da análise comparativa com softwares similares, o WebJuvia pode se destacar como uma ferramenta simples, inovadora e acessível, que possui um foco específico na área de alocação dinâmica de memória, e que se preocupa em prover um bom nível de interação com o usuário.

Apesar de promissora, o WebJuvia é uma ferramenta que ainda apresenta algumas limitações. O objetivo inicial do WebJuvia era simular os principais algoritmos de alocação de memória, e este objetivo foi contemplado, mas sua idealização inicial incluía o Quick-Fit dentre os algoritmos apresentados. Este método de alocação acabou não sendo implementado devido a uma grande carência de informações técnicas sobre o seu funcionamento. Outra limitação encontra-se na utilização do WebJuvia em determinados navegadores web menos robustos: navegadores desatualizados e/ou menos robustos podem não renderizar alguns dos emojis utilizados na ferramenta, o que pode gerar um incômodo na comunicabilidade, portanto recomenda-se a utilização do WebJuvia nos principais navegadores da internet.

Como trabalhos futuros, pretende-se implementar o Quick-Fit entre os algoritmos de alocação dinâmica de memória, desta forma tornando o WebJuvia uma ferramenta ainda mais completa. Atualmente o simulador possui uma funcionalidade de inserção aleatória de processos na memória, mas futuramente pretende-se incluir a funcionalidade de inserções pré-determinadas de processos, para que os educadores possam simular os algoritmos, junto aos alunos, a partir de um mesmo cenário de estado da memória, oferecendo novas possibilidades para o uso da ferramenta.

REFERÊNCIAS

- [1] Simone Barbosa and Bruno Silva. 2010. *Interação humano-computador*. Vol. 1. Elsevier Brasil.
- [2] Howard S Barrows, Robyn M Tamblyn, et al. 1980. *Problem-based learning: An approach to medical education*. Vol. 1. Springer Publishing Company.
- [3] Jonathan Bergmann and Aaron Sams. 2012. How the flipped classroom is radically transforming learning. *The Daily Riff* 2012 (2012), 1–3.
- [4] Tiago Davi Neves de Sousa and Cleyton Caetano de Souza. 2012. VM Simulator: Uma proposta de simulador para apoiar o ensino e a aprendizagem de Gerência de Memória Virtual. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 1.
- [5] Renê NS Gadelha, Ryan Ribeiro de Azevedo, Hilário TA de Oliveira, Tiago D Neves, Cleyton C Souza, and Edilson Leite da Silva. 2010. OS Simulator: Um Simulador de Sistema de Arquivos para Apoiar o Ensino/Aprendizagem de Sistemas Operacionais. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 1.
- [6] Alexandre Grotta and Edmir PV Prado. 2018. Um ensaio sobre a experiência educacional na programação de computadores: a abordagem tradicional versus a aprendizagem baseada em projetos. In *Anais do XXVI Workshop sobre Educação em Computação*. SBC.
- [7] Emily Yuriko Kioki, Patrick Piazza Santiago, and Afonso Celso Soares. 2008. Um simulador didático como ferramenta de apoio ao ensino da disciplina de sistemas operacionais. *INICIA* 37 (2008), 40.
- [8] Francis Berenger Machado and Luiz Paulo Maia. 2004. *Arquitetura de sistemas operacionais*. Vol. 4. LTC.
- [9] Luiz Paulo Maia. 2001. Sosim: Simulador para o ensino de sistemas operacionais. *Rio de Janeiro* (2001).
- [10] Luiz Paulo MAIA and Francis Berenger MACHADO. 2007. *Arquitetura de Sistemas Operacionais*.
- [11] Lucinéia Souza Maia, Marcelo Augusto Santos Turine, Hércules da Costa Sandim, and Débora Maria Barroso Paiva. 2010. Um modelo para o Desenvolvimento de Aplicações Web Acessíveis. In *Anais Principais do XVI Simpósio Brasileiro de Sistemas Multimídia e Web*. SBC, 235–242.
- [12] R Ontko. 2001. MOSS memory management simulator user guide. Retrieved June 22 (2001), 2010. http://www.ontko.com/moss/memory/user_guide.htm
- [13] Ana Lorena Nascimento PAIVA and Ronaldo BISPO. 2017. Emojis, as emoções representadas graficamente no ciberespaço. In *Intercom-XIX Congresso de Ciências da Comunicação na Região Nordeste*.
- [14] Universitário Campo Limpo Paulista, Luiz Felipe dos Santos Freitas, and Marcelo de Paiva Guimarães. 2018. SigemVR: Um Simulador Imersivo e Interativo de Gerência de Memória. (2018).
- [15] FABRÍCIO PEREIRA REIS. 2009. *TBC-SO/WEB: SOFTWARE EDUCATIVO PARA APRENDIZAGEM DE GERÊNCIA DE PROCESSOS E DE GERÊNCIA DE MEMÓRIA EM SISTEMAS OPERACIONAIS*. Ph.D. Dissertation. Universidade Federal de Lavras.
- [16] Thiago Pirola Ribeiro, Rafael Lucas Bernardes Lima, and Edgard Araujo Lobo. 2014. Simuladores de Gerência de Memória e Processador para Auxílio às Aulas Teóricas de Sistemas Operacionais. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 25. 1028.
- [17] Andrew S Tanenbaum and Herbert Bos. 2015. *Modern operating systems*. Vol. 4. Pearson Education.