CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U. nº 198, de 14/10/2016
AELBRA EDUCAÇÃO SUPERIOR - GRADUAÇÃO E PÓS-GRADUAÇÃO S.A.

Odilon Júnio Santos de Jesus

odilon@rede.ulbra.br

OSLIVE: módulo de simulação do mecanismo de segmentação

Palmas – TO

2021

Odilon Júnio Santos de Jesus

OSLIVE: módulo de simulação do mecanismo de segmentação

Projeto de Pesquisa elaborado e apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. Dr.M.e Madianita Bogo Marioti.

Palmas – TO

2021

Odilon Júnio Santos de Jesus

OSLIVE: módulo de simulação do mecanismo de segmentação

Projeto de Pesquisa elaborado e apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. Dr.M.e Madianita Bogo Marioti.

Aprovado em: _____/_____/_____

BANCA EXAMINADORA

_____

Prof. De M.e Madianita Bogo Marioti

Orientador

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. Esp. Fabio Castro Araujo

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. Esp. Fernanda Pereira Gomes

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2021

AGRADECIMENTOS

Primeiramente a Deus por nunca me abandonar e sempre me mostrar o caminho correto em minha vida, logo em seguida vem minha família por sempre me apoiar em minhas decisões e sempre ajudar quando necessário sempre estando ao meu lado, principalmente meus pais que auxiliaram no meu caráter e fazendo eu ser essa pessoa que hoje eu sou.

Agradeço aos professores que me auxiliaram no meu caminho, por todo ensinamento e oportunidades que eu recebi. Principalmente a minha orientadora que sempre me ajudou em tudo que precisei, auxiliando quando pedido.

Enfim, sou grato a todos que colaboraram com esse projeto.

RESUMO

JESUS, Odilon Júnio Santos de. **OSLIVE: MÓDULO DE SIMULAÇÃO DO MECANISMO DE SEGMENTAÇÃO.** 2021. 73 f. Trabalho de Estágio – Curso de Ciências da Computação, Centro Universitário Luterano de Palmas, Palmas/TO, 2021[1].

A disciplina de Sistema Operacionais é importante para os cursos relacionados à computação. A disciplina abrange o conhecimento dos alunos por possibilitar a compreensão de como funciona o sistema operacional. Com a disponibilidade de ferramentas online, o ensino fornece uma maneira prática de entender como funcionam os mecanismos do sistema operacional aprendidos na disciplina, e a pesquisa básica teórica pode ter um efeito claro. Este trabalho apresenta uma melhoria para o módulo de simulação do mecanismo de segmentação do OSLive e visa atualizar a interface do módulo simulação do mecanismo de segmentação padronizada com os outros módulos do OSLive para apresentar uma aparência agradável, aumentar a capacidade do módulo suportando uma simulação maior, com efeitos agradáveis para a utilização.

Palavras-chave: Sistema Operacionais. Gerenciamento de memória segmentação.

1

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

SO: Sistema Operacional

CEULP: Centro Universitário Luterano de Palmas

MMU: Memory Management Unit

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

JS: Javascript

SUMÁRIO

1 INTRODUÇÃO

Sistemas Operacionais é uma disciplina recomendada pelo MEC (2012) para os cursos da área de Computação do Brasil. Essa disciplina aborda conceitos relacionados ao funcionamento do Sistema Operacional (SO) que, segundo Oliveira (2001, p. 2), "procura tornar a utilização do computador mais eficiente e mais conveniente", fazendo a intermediação entre o *software* e o *hardware* para o usuário. O conteúdo da disciplina é dividido de acordo com as áreas de gerência dos SOs: gerência de processos, gerência de memória, sistema de arquivos e entrada e saída.

Para melhorar o aprendizado na disciplina Sistemas Operacionais foi desenvolvida dentro do Centro Universitário Luterano de Palmas (CEULP) uma ferramenta chamada OSLive. O OSLive é uma plataforma web que oferece recursos como simulação, resolução de exercícios, etc., e é um projeto do Grupo de Estudos em Novas Tecnologias para processos de Ensino de Aprendizagem (GENTE) do CEULP. O OSLive é composto por vários módulos, que abordam mecanismos das áreas de gerência dos SOs.

Um dos módulos já implementados é o da simulação do mecanismo de Gerência de Memória Segmentação. A Gerência de Memória é a área do SO responsável segundo Tanenbaum (2005, p. 139), "Sua função é manter o controle de quais partes da memória estão em uso e quais não estão", com isso pode se entender que a gerência de memória para faz a alocação e liberação de memória assim que requerido pelo programa a ser utilizado, como cada programa tem sua parte de memória reservada, não há conflito de dados por não reescrever dados que estão em reserva de outro programa.

A gerência de memória segmentação é um conteúdo da disciplina de Gerenciamento de Memória da matéria SO, com isso é apresentado que o esquema de segmentação divide o programa em blocos de segmentos. Segmentação de acordo com Oliveira (2001, p. 11) "O conceito de página, fundamental para a paginação, é uma criação do sistema operacional para facilitar a gerência da memória.".

O módulo de simulação do mecanismo de segmentação está no momento na sua versão 1 e precisa de alguns ajustes como adaptar a interface ao padrão dos outros módulos do OSLive, criar processo automaticamente, com valores aleatórios, permitir a exclusão de processos e aumentar o tamanho da memória física representada. Essas e outras melhorias foram realizadas como correção do semântico do código que inviabiliza a possibilidade de exclusão, com isso a criação de lacunas entre os espaços de memória foram possíveis e a modificação no código base para a possibilidade da execução direta em HTML.

Desta forma, este trabalho teve como objetivo realizar melhorias na versão 1 do Módulo de simulação do mecanismo Gerência de Memória por Segmentação do OSLive. Neste documento é apresentado todas as melhorias, sendo explicado a parte teórica do que foi atualizado, demonstrando como era a versão anterior, antes de ser atualizado. Após as explicações é demonstrado como foi o planejamento do projeto e em seguida a sua execução, a forma que o projeto se encontra hoje.

2 REFERENCIAL TEÓRICO

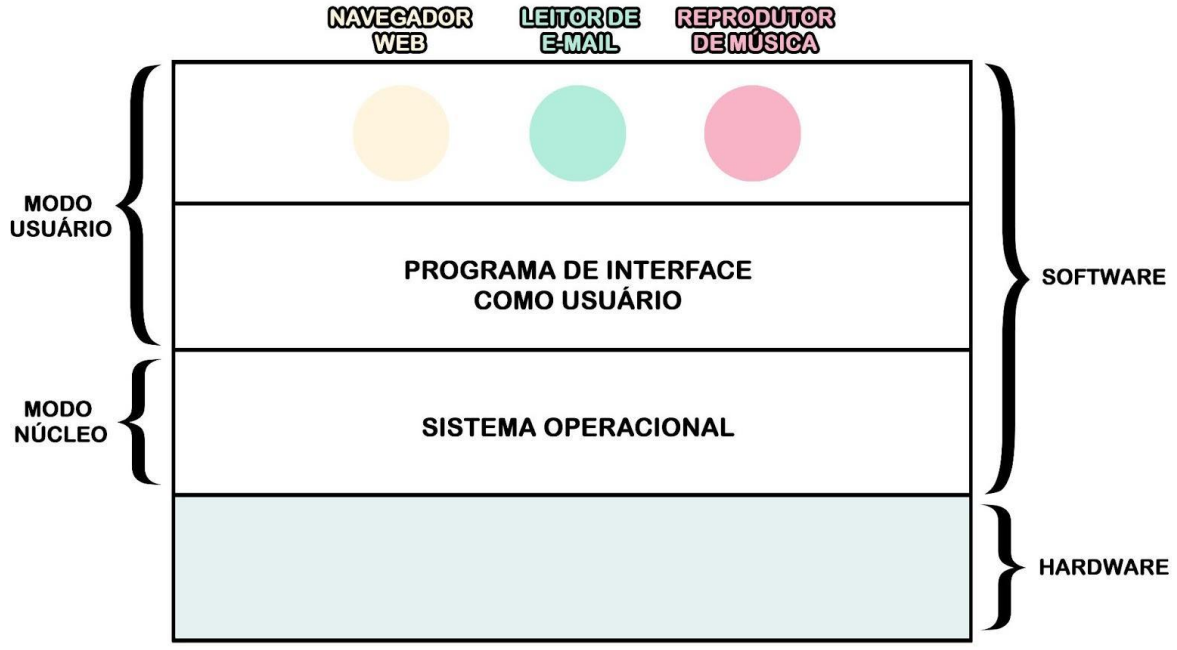
Nesta seção será falado de uma forma geral sobre as áreas do sistema operacional: gerência de memória; entrada e saída; gerenciamento de processos; gerência de arquivos. A versão 1 do módulo de simulação do mecanismo de gerenciamento de memória segmentação também será apresentada, com visualização de como será modificada o projeto.

2.1 SISTEMA OPERACIONAL

O sistema operacional, de acordo com Tanenbaum (2015, p. 1), "fornece aos programas do usuário um modelo do computador melhor, mais simples e mais limpo, assim como lidar com o gerenciamento de todos os recursos". O sistema operacional é o responsável por gerenciar os recursos que o *hardware* entrega, para fornecer aos *softwares* uma forma de funcionar adequadamente.

Os componentes do *hardware*, de acordo com Tanenbaum (2015, p.1), "chips, placas, discos, um teclado, um monitor e objetos físicos similares", o *hardware* unido faz o *software* funcionar. Como mostrado na figura 1, apresenta uma base de todos os componentes *hardware* e *software* que fazem parte do sistema computacional.

Figura 1. Onde o sistema operacional se encaixa.



Fonte: Adaptado de Tanenbaum, Andrew S, 2015

Pode ser observado na figura 1, o SO é uma camada que fica entre o modo usuário e o *hardware*, no *hardware* tem toda a parte física (teclado, mouse, chips, placas, HDD, etc). O SO é a ligação entre o *hardware* com o usuário, é responsável pela execução de tudo que o usuário quer executar. No modo usuário é composto por todo *software* que é necessário para a utilização do seu usuário, cada usuário tem sua preferência, mas diferente do SO não tem livre acesso ao *hardware*. A execução de todos os *softwares* no computador é gerenciada pelo SO com o intuito de que toda execução seja facilmente entendida pelo usuário.

- **Gerência de memória**: de uma forma segura o SO é responsável por liberar e alocar memória para o processo. Se o processo estiver a utilizar uma memória, o SO garante que outro processo não utiliza a área já ocupada, nesse gerenciamento evita erros no sistema, travamentos ou o funcionamento imperfeito desse processo;
- **Gerência de entrada e saída**: todos os dispositivos que estão a ser utilizados pelo computador são gerenciados pelo SO, toda ação realizada a pedido dos processos é controlada e executada pelo SO;
- **Gerência de processo**: um processo é programado para executar certas funções, um programa no computador pode ter vários processos, cada processo tem sua função. Como o SO tem vários processos sendo executado ao mesmo tempo, é preciso fazer um gerenciamento;
- **Gerência de arquivos**: é o gerenciamento de todo o armazenamento de dados do SO, todo arquivo que é utilizado pelo usuário (música, documento, etc). O gerenciamento de arquivos é mais fácil de ser entendido por usuários mais comuns, pois é utilizado a todo momento.

O objetivo deste trabalho é a simulação do mecanismo de segmentação, que é um mecanismo que faz parte da área de gerência de memória, por isso, as próximas seções abordam esse mecanismo de forma mais aprofundada

2.2 GERÊNCIA DE MEMÓRIA

Na multiprogramação muitos processos são executados simultaneamente e, para serem executados mais rapidamente, são carregados para a memória física (RAM), pois quando tem a necessidade de buscar no disco rígido para finalizar o processo o tempo de execução se eleva muito. O gerenciamento de memória do SO é necessário para que os diversos processos que estão compartilhando a memória sejam executados de forma segura e eficiente.

De acordo com Oliveira (2001, p8), “é função da gerência de memória do sistema operacional prover os mecanismos necessários para que os diversos processos compartilhem a memória de forma segura e eficiente”, no caso, otimiza o uso da memória de uma forma que os dados de um processo que estão armazenados em uma área de memória física não sejam sobrepostos por dados de outro processo.

O processo tem uma memória lógica que o processo é capaz de acessar, os endereços lógicos são manipulados pelos processos. O espaço de endereçamento lógico faz a junção de todos os endereços lógicos formados pelo processo, cada processo tem um espaço de endereçamento lógico próprio.

A memória física funciona em nível de máquina, no caso é a parte de circuitos e componentes eletrônicos do computador, o espaço de endereço físico consiste em todos os endereços circuitos aceitáveis, com isso todos os endereços físicos são formados por uma posição real na memória.

A memória lógica e memória física são enxergadas de forma diferente, pois o processo somente enxerga a memória lógica enquanto o *hardware* somente enxerga a memória física, com isso gera-se a necessidade de algum conversor de endereço de memória lógico para endereço de memória físico. A Unidade de gerência de memória (Memory Management Unit, MMU), de acordo com Tanenbaum (2015, p.125), é a responsável por mapear os endereços lógicos em endereços de memória física”.

O processo que está sendo executado no processador que for utilizar a memória tem a necessidade de traduzir o endereço lógico para o endereço físico, para ser possível descobrir o local em que está alocado na memória física. O processo passa o endereço lógico e o processador solicita o mapeamento deste endereço lógico para o endereço físico, o que é feito pela MMU O processo é capaz de acessar a memória física após a tradução do endereço lógico.

Há vários mecanismos de gerência de memória, capazes de determinar a forma que será feita a utilização da memória física pelos processos. Como o objetivo desse trabalho é o mecanismo de segmentação, assim terá uma melhor explicação na seção seguinte.

2.2.1 Segmentação

Segundo Oliveira (2001, p. 11), é possível orientar a gerência de memória para suportar diretamente o conceito de segmento. O endereçamento da memória lógica passa a ser endereçada pelo número do segmento e o deslocamento desde o início de seu segmento. Ao carregar, cada segmento é copiado para a memória física e uma tabela de segmento é

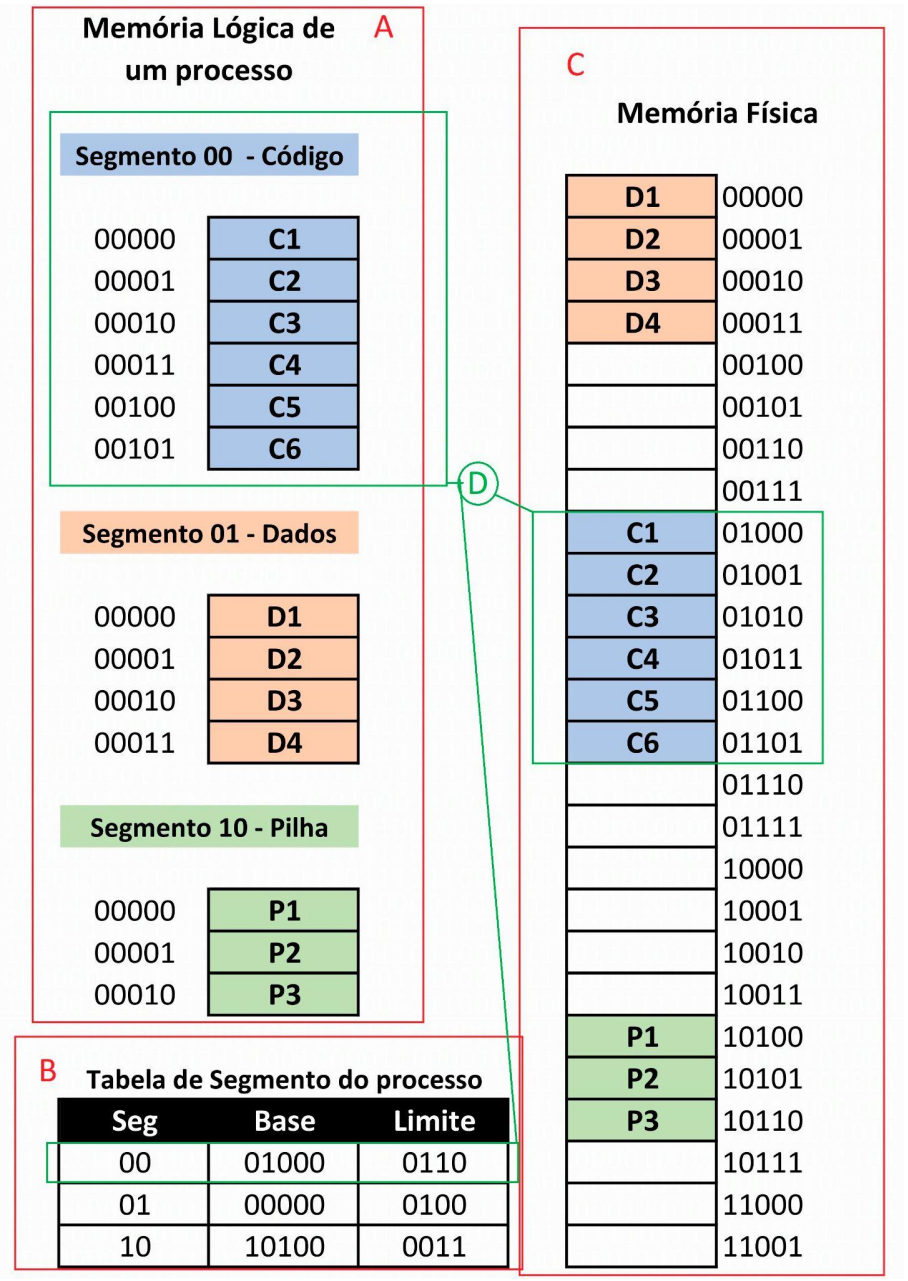
construída, que informa a cada segmento em que endereço de memória física ele está colocado e qual o seu tamanho.

Os processos são divididos em endereçamento logicamente independentes, a entrada na tabela de segmento tem o endereço do segmento da memória física, com informações do seus próprios endereços base e o seu limite. Esse mecanismo auxilia no compartilhamento, pois como cada segmento é representado por uma parte específica do programa, se essa parte for de uso comum podem ser utilizados de forma compartilhada.

A segmentação não apresenta fragmentação interna, pois a quantidade de memória exata é alocada para cada segmento. Mas como existem várias áreas de diferentes tamanhos acaba gerando fragmentações externas, pois quando o processo é finalizado, tem a possibilidade de gerar lacunas entre os processos, que acaba sem a possibilidade de ser utilizado por processos maiores.

A figura 2 mostra o funcionamento da alocação de memória na segmentação. Na área A é destacada a memória lógica de um processo, no caso cada segmento da memória lógica de um processo é, na verdade, armazenada na sua devida posição na memória física como visto na área C. A tabela de segmento do processo na área B indica a base do segmento, que é o início do seu espaço de endereçamento, ou seja, o primeiro *byte*, e o limite, que se refere ao tamanho do seu espaço de endereçamento, com isso mostra o mapeamento entre os endereços lógico e físico.

Figura 2. Alocação de memória de segmentação.



Fonte: Adaptado de Oliveira, 2001

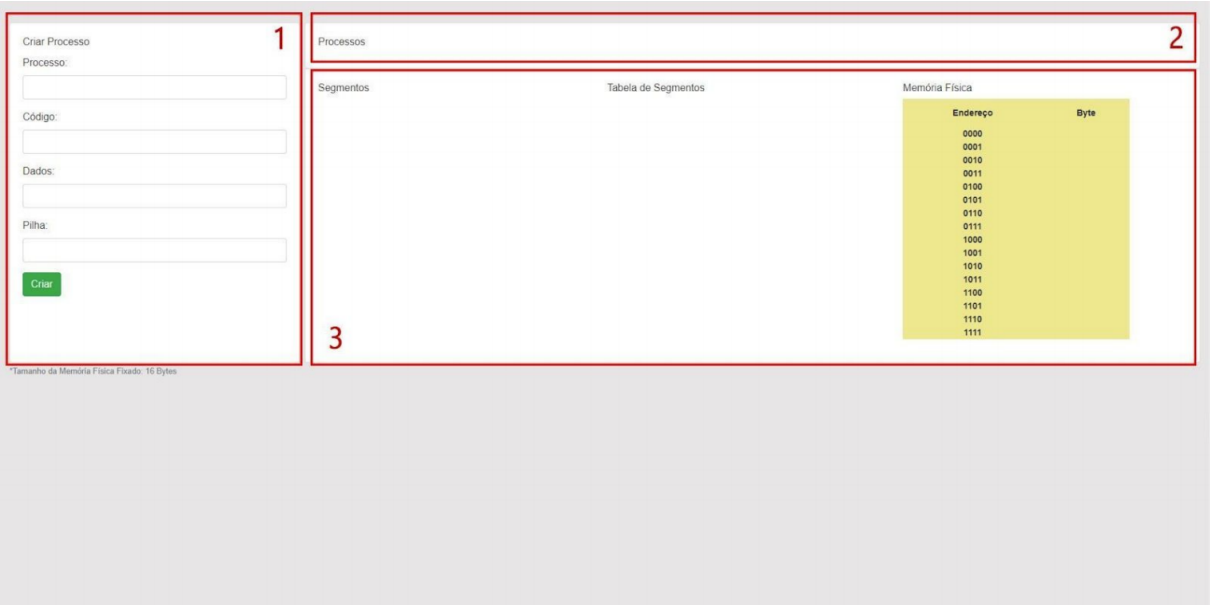
Por exemplo, na situação apresentada na figura 2, na área D, mostra como o local de armazenamento da memória, o segmento 00 do processo (area A) inicia no endereço 01000 na memória física (área D), valor que é indicado como base na tabela de segmentos para o segmento 00 (área B). O indica que o processo ocupa do endereço 01000 até o endereço imediatamente anterior ao 01000+0110, com isso é visto na memória física (área D) a posição que se localiza o segmento 00.

2.3 VERSÃO 1 DA FERRAMENTA

Já foi implementado um módulo de simulação da segmentação para o OSLive. Essa primeira versão foi apresentada no trabalho “implementação do módulo de simulação do mecanismo de segmentação do OSLive” (MARTINS, 2019). Essa seção apresenta a versão 1 do módulo, apresentando o seu funcionamento básico e as fragilidades encontradas.

A figura 3 apresenta a tela inicial antes da utilização do módulo.

Figura 3. Apresentação do Ambiente



Fonte: Implementação do módulo de simulação do mecanismo de segmentação do OSLive, Martins, 2019

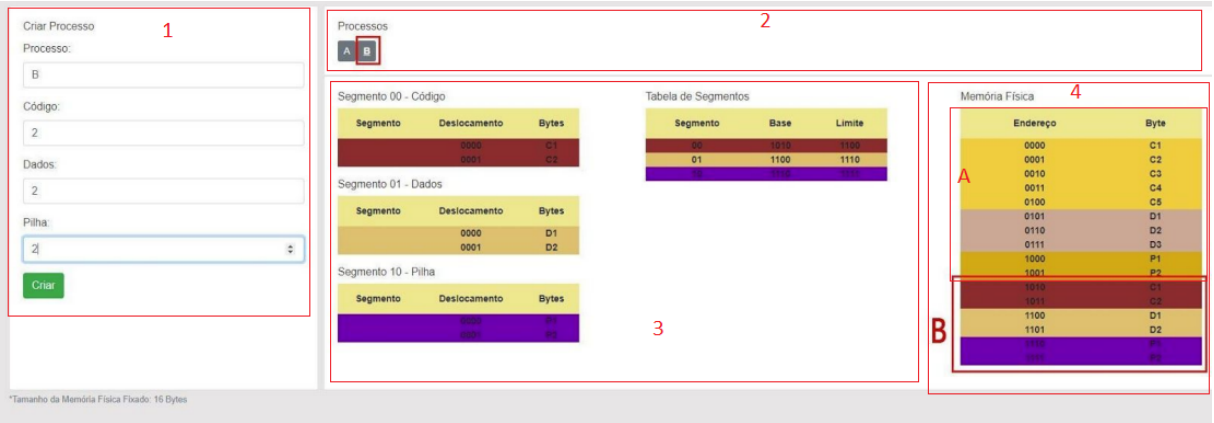
Conforme os destaques da figura 3, para a realização da simulação, são disponibilizadas três áreas recebendo valor inteiro e simbolizam os segmentos pré-definidos:

- A área 1 - Criar processo: onde se encontra os campos para a adição das informações dos processos que serão criados, é pedido um Processo na *textbox*, com a possibilidade de escolha de qualquer nome pelo usuário; um tamanho dos segmentos de código, dados e pilha, com limite total de 16 *Bytes*;
- A área 2 - Processos: em ordem de criação, é mostrado ao usuário os nomes dos processos, no momento em que o usuário selecionar um dos processos, todos os dados deste processo serão exibidos na área 3.

- A área 3 - Segmentos: é mostrado a impressão do processo selecionado na área 2 e é onde ocorre os processos dos dados, com a tabela de segmentos mostrando a ligação da memória lógica com a memória física.

A figura 4 apresenta a tela da simulação após a criação de 2 processos: processos A com 10 *bytes* e o processo B com 6 *bytes*. Na figura, o processo B foi selecionado na área 2, com isso é demonstrado na área 3.

Figura 4. Tela da aplicação após a criação de dois processos



Fonte: Adaptado da Implementação do módulo de simulação do mecanismo de segmentação do OSLive, Martins, 2019

Conforme destacado na figura 4, as áreas 2 a 4 foram preenchidas após a criação dos processos na área 1:

- A área 1 Criar Processo: mostra os dados com que cada *textbox* foi preenchido para a criação do processo B;
- A área 2 Processos: apresenta os processos listados, com a seleção no processo B;
- A área 3 Segmento: é apresentada a memória lógica, composta por segmentos gera tabela com divisão em seus respectivos campos, na adição do próximo como exemplo o processo B com 2 *bytes* de espaço, o processo automaticamente atualiza e mostra os dados do novo processo que está a chegar. É localizada a tabela de segmentos que mostra o início do segmento com o seu limite;
- A área 4 Memória Física: representa onde estão armazenados os segmentos dos processos na memória Física, as cores geradas no processo são diferentes para cada processo e segmentos, assim se torna mais fácil a identificação de cada segmento e processo.

A versão 1 do módulo foram encontradas algumas fragilidades que podem ser feito ajustes para melhorar a apresentação da simulação e a compreensão do usuário:

- interface diferente dos outros módulos: a interface padrão ajudará os usuários que já conhecem os outros módulos da ferramenta do OSLive utilizar com muito mais facilidade. Além disso, a forma aleatória de escolha de cores, imprime cores muitos escuras para visualização do processo, assim gera dificuldade na visualização das informações da simulação;
- não tem possibilidade de geração automática de processos com valores aleatórios: com possibilidade da adição automática de processos, assim gera uma agilidade para criar exemplos rápidos;
- não tem possibilidade de remoção de processos adicionados: com a função de exclusão de processos o usuário teria a possibilidade de visualizar o que acontece na memória física quando um processo deixa o sistema e libera a memória física, o que é importante para a compreensão do mecanismo;
- não tem possibilidade de simulações maiores do que 16 *bytes*: o aumento do espaço possibilita simulações com mais dados e processos, aumentando as possibilidades de configuração para o usuário.

3 METODOLOGIA

Este trabalho tem como objetivo uma atualização de uma ferramenta web, Módulo de simulação do mecanismo Gerência de Memória por Segmentação, que será adicionado ao OSLive.

3.1 MATERIAIS

Os materiais que foram utilizados para o desenvolvimento da atualização, tanto para padronização do modelo do OSLive como para as melhorias na simulação, foram:

- Bootstrap: framework gratuito para desenvolvimento HTML, CSS e JS (BOOTSTRAP, 2021). Foi utilizado para as atualizações de interface da aplicação;
- Vue.JS: framework JavaScript de código aberto para desenvolvimento de interfaces de usuário e aplicações (VUEJS, 2021). Foi utilizado para as atualizações da aplicação, modificação base do código e ajustes para funcionalidade em HTML;

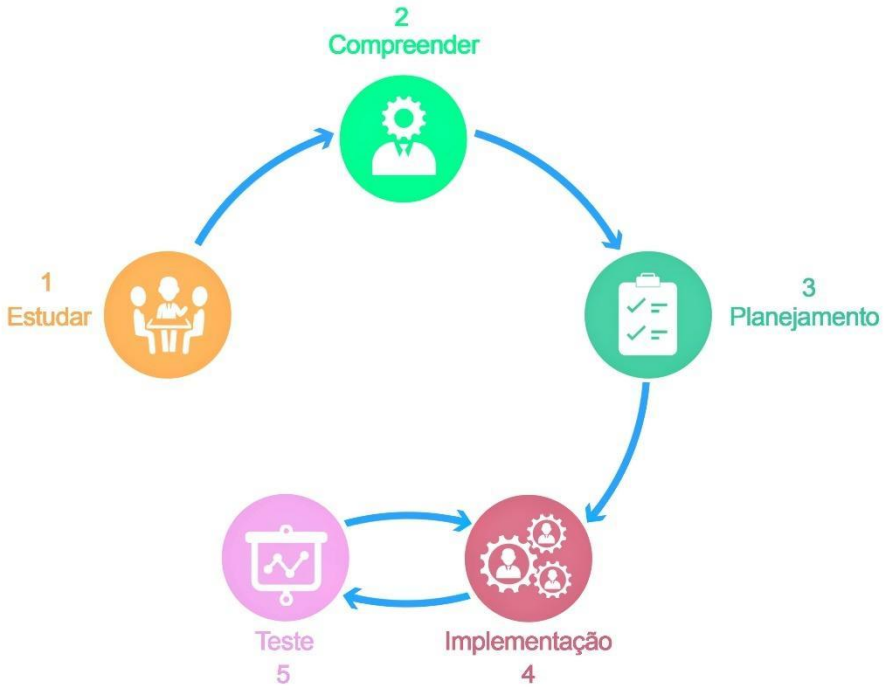
As ferramentas foram utilizadas como base do desenvolvimento, tendo o VUE.JS como a base da aplicação, como as funções para o funcionamento e foi a parte lógica do projeto gerando a possibilidade da execução das simulações.

O Bootstrap foi utilizado na parte visual da aplicação, com a responsabilidade da beleza e padronização da interface do OSLive, com o objetivo de deixar a estadia mais agradável.

3.2 MÉTODOS

A forma da organização, desenvolvimento e execução do trabalho ocorreu de acordo com a sequência apresentada na figura 5.

Figura 5. Fluxo de atividades



No começo, na fase 1, foi necessário realizar uma revisão na disciplina de SO, com foco em gerência de memória, a fim de compreender completamente o funcionamento e definir como será a sequência de ações durante a execução, para servir de base para a implementação. Essa realização foi feita com livros didáticos de SO e matérias da aula de SO. Além disso, foram realizadas reuniões com a professora Madianita Bogo, que ministra a disciplina de SO no CEULP/ULBRA, para a orientação e esclarecimento de dúvidas.

Em seguida na fase 2, foi estudada a base do código da versão 1, com o intuito de verificar onde e como pode ser mudado para melhoria e padronização, além disso, com o estudo do código foram verificados erros que foram corrigidos na fase 4.

Na fase 3 foi realizada a definição da lógica do algoritmo, a interface e os detalhes da simulação. Com isso foi trabalhada a parte mais técnica do processo.

Na próxima etapa, a fase 4, foi realizada a implementação do que foi definido anteriormente, com a atualização do código da interface da versão 1 e com a adição das novas funcionalidades.

Na fase 5 foram realizados os testes sobre a modificação da interface, implementação das novas funcionalidades e correção dos erros encontrados na versão 1, o que foi feito em paralelo à fase 4, de implementação. A verificação de todas as funcionalidades foi feita com o auxílio da professora da disciplina Sistemas Operacionais do CEULP, analisando se está tudo em perfeito funcionamento de acordo com o que foi proposto. Além disso, foram realizados

testes para buscar possíveis erros nas novas funcionalidades, visando garantir que estivesse tudo com o funcionamento correto. Assim, foram aplicadas as correções dos erros encontrados e novos testes foram executados até alcançar o resultado esperado.

4 RESULTADOS E DISCUSSÃO

A apresentação do desenvolvimento da ferramenta de simulação do mecanismo de segmentação é mostrada nesta seção, que abordará detalhamentos sobre o funcionamento da ferramenta e explicações sobre o processo de execução das simulações. A simulação ocorre quando o usuário adiciona processo, inserindo as informações manualmente ou escolhendo o método de preenchimento automático, que mostrará automaticamente as informações geradas aleatoriamente na interface.

4.1 MELHORIAS IMPLANTADAS

Antes de começar a explicação do passo a passo das melhorias aplicadas, a seção 2.3 "versão 1 da ferramenta" apresenta a forma que estava a ferramenta antes das atualizações realizadas. A ferramenta anterior será citada como versão 1.

Antes de iniciar a implementação da nova versão, foi estudado o código da versão 1, para encontrar as necessidades de melhoria. Como este módulo será implantado juntamente com outros módulos já existentes, uma certa semelhança de aparência se torna necessária, para facilitar a utilização do OSLive, além de melhorias deixando o módulo com novas funcionalidades.

A interface da versão 1 é diferente do padrão dos outros módulos do OSLive, a padronização é importante para facilitar a visualização e o uso por parte dos alunos. Então, a primeira melhoria foi ajustar a interface, ajustando o formato anterior e colocando com a mesma aparência dos outros módulos do OSLive.

Na versão 1 é representada uma memória física com apenas 16 *Bytes*, com isso a quantidade e tamanhos de processos que podem ser criados para a simulação fica limitado. Após estudar a forma da criação, foi verificado que a implementação era em binário com espaços de quatro dígitos, com isso foi necessário a expansão para cinco dígitos possibilitando a criação da memória de 32 Bytes.

A versão 1 do simulador possibilita a utilização de somente a função de inclusão de processos e é importante possibilitar a exclusão de processos, para que o aluno compreenda melhor o funcionamento do mecanismo de segmentação. Para que essa possibilidade fosse feita, teve correção do código e foi adicionado um botão de exclusão do processo para que o usuário possa remover um processo criado.

Como na versão 1 não há a possibilidade de processos serem gerados automaticamente, o usuário sempre tem a necessidade de inserir todos os processos manualmente. Para resolver esse problema, foi criada uma variável randômica, que escolhe

uma letra do alfabeto como nome do processo e aleatoriamente valores de 1 a 4 para o preenchimento dos valores código, dados e pilha.

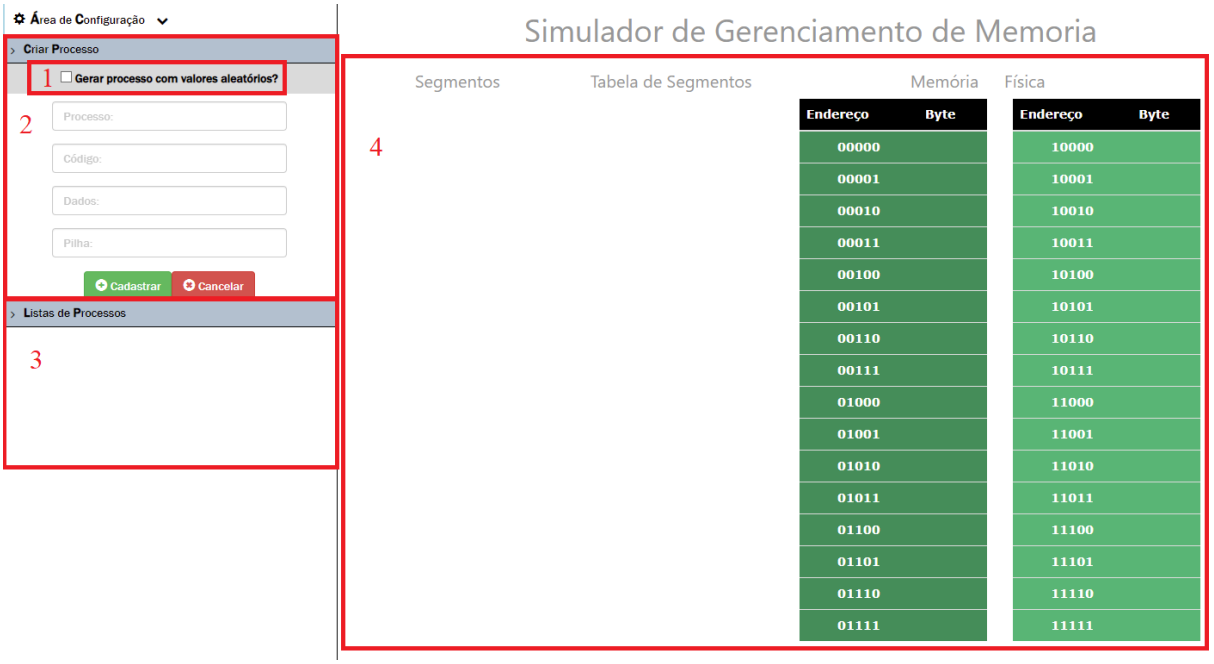
Como a versão 1 necessitava de instalação na máquina para a utilização do código, dificultava para disponibilizar a ferramenta online. Para resolver esse problema, foi necessário modificar a base do código, diminuindo e mudando a forma inicial, para ter a leitura de forma correta na criação do HTML.

Durante os testes e a implementação das melhorias realizadas, foram encontrados erros na execução da versão 1, estes erros atrapalhavam na implementação das novas funcionalidades. Os erros foram corrigidos durante a implementação da nova versão.

4.2 AMBIENTE DA SIMULAÇÃO

A tela inicial do ambiente da simulação do mecanismo de segmentação é apresentado na Figura 6, tela que é visualizada pelo usuário no momento em que acessa o sistema, a interface segue o padrão dos módulos disponíveis no OSLive. Na interface do OSLive, no lado esquerdo é apresentada a Área de Configurações contendo: entradas de dados, visualizações do processo, cadastro de novos processos e opção de criação de processo aleatório e no lado direito são apresentadas as simulações.

Figura 6. Tela inicial



A tela inicial do ambiente de simulação do mecanismo de segmentação é mostrada na figura 6, no momento de inicialização, sem nenhum dado inserido. Na tela, é possível a

visualização de "Criar Processos" (Figura 6 - 1 e 2), "Listas de Processos" (Figura 6 - 3) e o resultado da simulação na área “Simulador de Gerenciamento de Memória” (Figura 6 - 4).

Na primeira área, criar os processos (Figura 6 - 1 e 2), se o usuário preferir adicionar processos automaticamente, é necessário selecionar o *checkbox* “Gerar processo com valores aleatórios?" (1) habilitando o botão Gerar Processo, a cada clique neste botão cria-se um processo com valores aleatórios. Para criar os processos manualmente (2), o usuário deve entrar com as informações de um processo (código, dados e pilha) e, após inserir as informações, o usuário deve selecionar a opção “cadastrar”. Nos dois casos, o processo é inserido na área “Lista de Processos” (3) e na área de apresentação das memórias lógica e física dos processos (4), apresentando a situação após a criação.

Após o cadastro de processos, de forma aleatória ou manual, os processos apresentados na área “Lista de Processos” (3) terão as opções de exclusão do processo e selecionar o processo desejado, sendo que as informações da memória lógica do processo selecionado serão apresentadas na área da simulação (4). A área da simulação apresentará as seguintes informações:

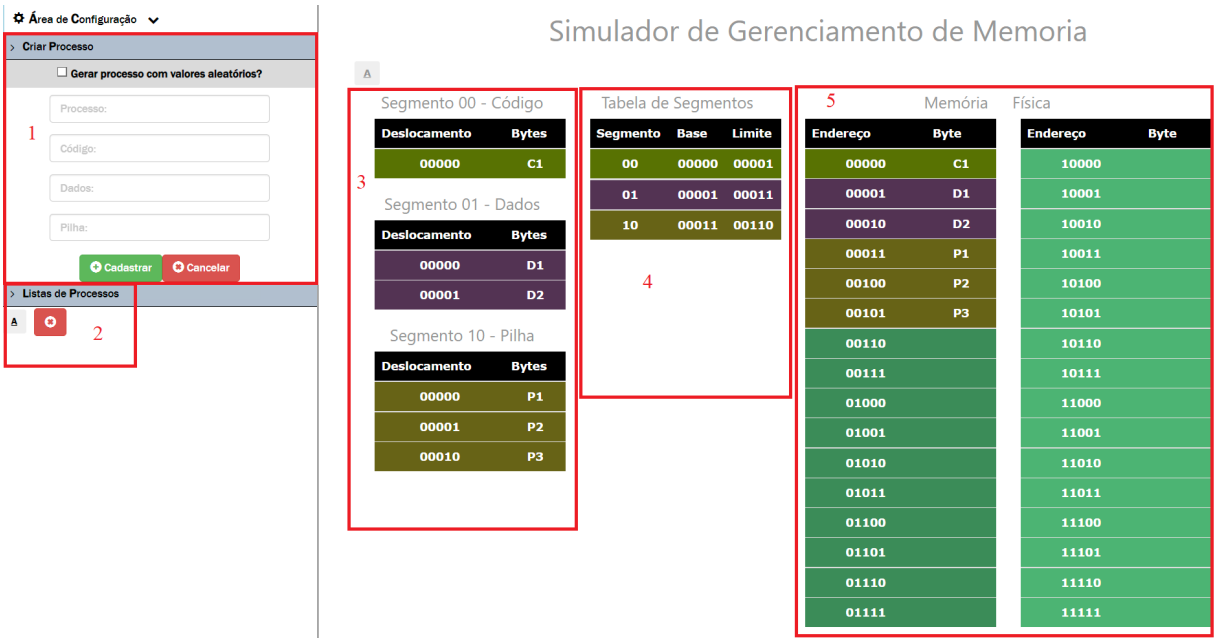
- a memória lógica do processo selecionado, com os segmentos com código, dados e pilha, mostrando em cada byte o seu deslocamento;
- informações da tabela de segmentos do processo selecionado, apresentando para cada segmento a sua base com o limite;
- a memória física, que apresenta os quadros (páginas físicas) livres e ocupados.

O tamanho máximo da memória física foi definido com 32 Bytes, sendo o dobro da versão 1, possibilitando a criação de processos com maior quantidade de *Bytes* ou mais processos.

4.3 EXECUÇÃO DA SIMULAÇÃO

A funcionalidade da ferramenta é demonstrada nesta subseção, dependendo das informações inseridas, a simulação ficará de forma diferente, dificultando a repetição exata do mesmo teste por causa das cores aleatórias geradas. Para demonstração foi criado um novo processo chamado "**A"** definindo os valores do segmento **Código** com 1 Byte, o segmento **Dados** com 2 bytes e o segmento **Pilha** com 3 bytes, assim a ferramenta realizará a produção das tabelas de acordo com o valor escolhido para cada segmento, criando a tabela de segmentos mostrando a posição de onde está localizado na memória física melhorando a visualização por ser separado por cores.

Figura 7. Processo A

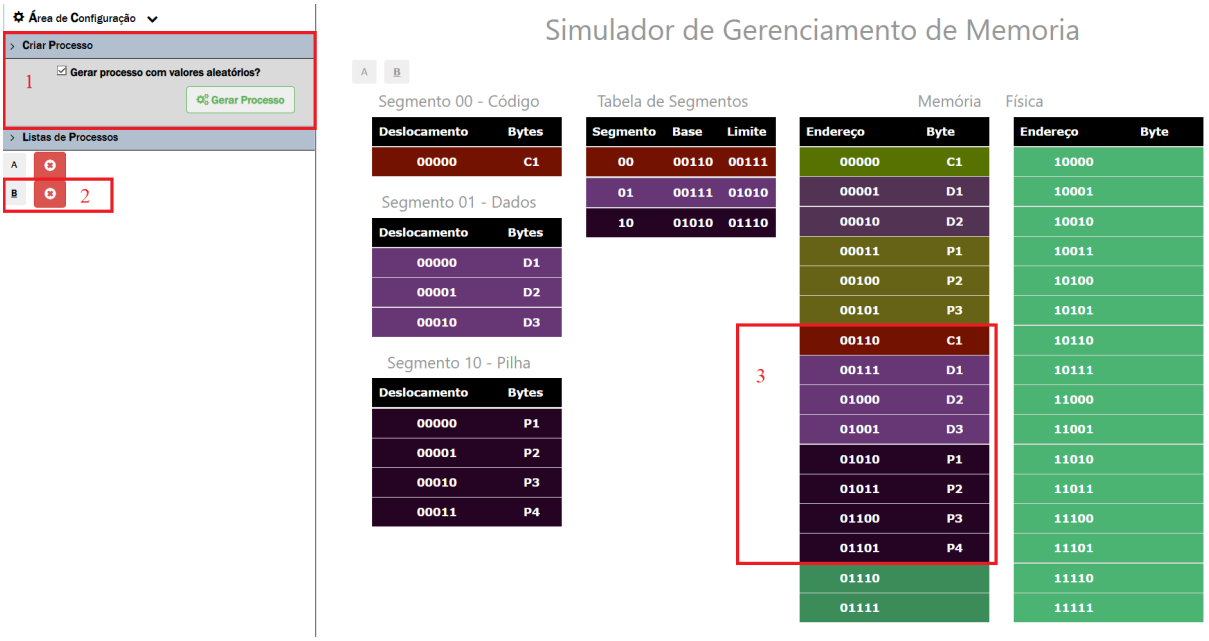


A figura 7 apresenta a tela após a inserção manual do processo A na área “Criar Processos” (Figura 7 - 1), com os valores Código 1, Dados 2 e Pilha 3 demonstrada na área “Segmentos” (Figura 7 - 3):

- em lista de processos (Figura 7 - 2) mostra o processo A, quando criado o processo é selecionado automaticamente, ficando em negrito e sublinhado, um X aparece na frente do processo possibilitando a exclusão do mesmo;
- em segmentos (Figura 7 - 3) fica localizado os segmento do processo A, mostrando o deslocamento com o respectivo *Bytes,* as cores de cada segmento do processo A é diferente, facilitando a visualização do processo;
- em tabela de segmento (Figura 7 - 4) mostra o resultado da tabela de segmentos do processo A, respeitando a cor já escolhida automaticamente, criando a base e limite de cada segmento;
- em memória física (Figura 7 - 5) é possível identificar a posição do processo A na memória física, com as cores sendo a mesma do segmento deixando destacado e mais fácil a ligação da tabela e do segmento com a memória física.

A figura 8 apresenta o resultado da simulação após inserir um processo B utilizando o gerador aleatório de processo (Figura 8 - 1), o gerador automaticamente adiciona o processo com Código 1, Dados 3 e Pilha 4, a situação anterior, já existia um processo A de tamanho Código 1, Dados 2 e Pilha 3.

Figura 8. Processo B Aleatório

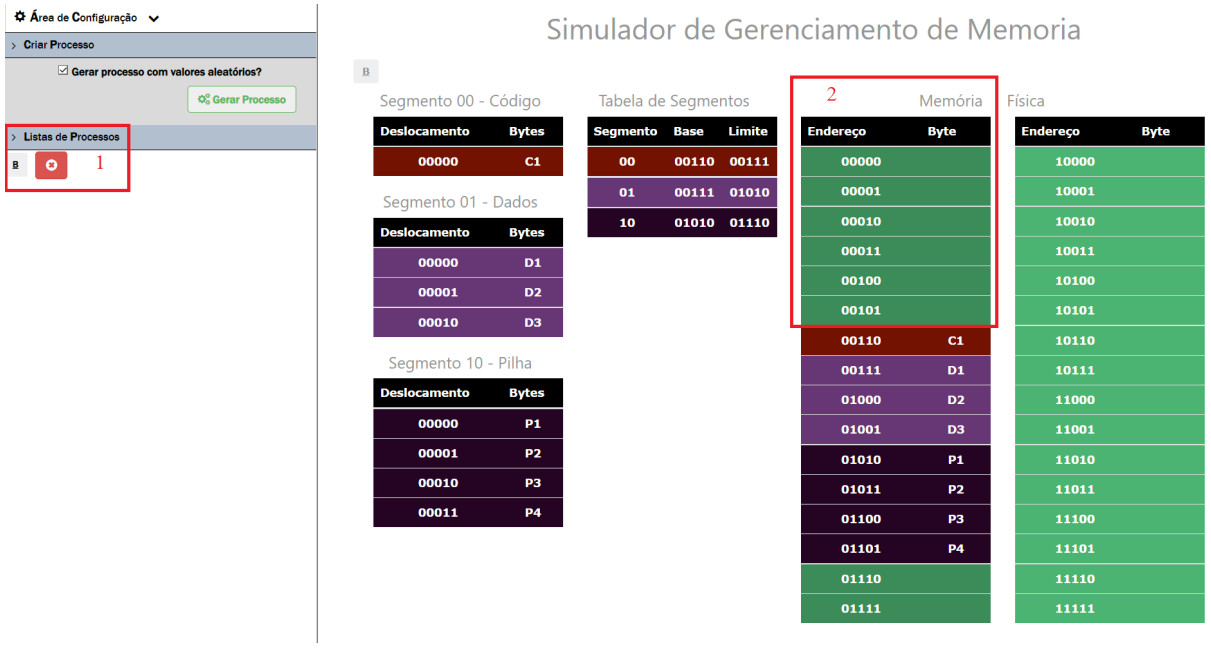


A situação apresentada na figura 8 é a seguinte:

- em lista de processos (Figura 8 - 2) destaca a seleção no processo B logo após sua adição;
- em memória física (Figura 8 - 3) demonstra a posição dos segmentos no endereço da memória física do processo B com o seu respectivo *Byte*.

A figura 9 apresenta o resultado da simulação após remoção do processo A da situação anterior, apresentada na figura 8.

Figura 9. Remoção do processo A

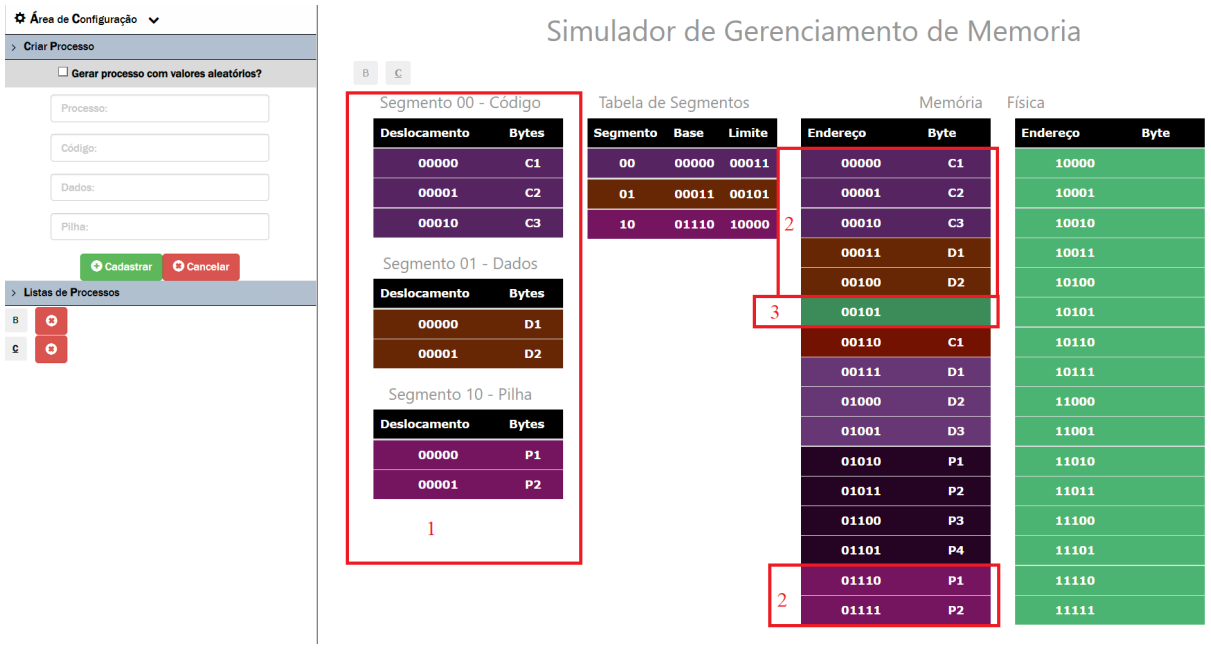


A situação apresentada na figura 9 é a seguinte:

- em lista de processo (figura 9 - 1) não apresenta mais o processo A, assim não há a possibilidade de selecioná-lo, além do processo B ocupar a posição do processo A;
- a memória física (figura 9 - 2) possibilita visualizar que o conteúdo do processo A foi removido da memória física, os espaços que o processo A ocupavam ficaram como área livre possibilitando a adição de novos processos.

A figura 10 apresenta o resultado da simulação após inserção de um novo processo na situação apresentada na figura 9.

Figura 10. Processo C



A situação apresentada na figura 10 corresponde à adição de um novo processo C com os dados Código 3 *Bytes*, Dados 2 *Bytes* e Pilha 2 *Bytes*:

- Em memória lógica (figura 10 - 1) mostra todos os segmentos do processo C com suas respectivas cores;
- Em memória física (figura 10 - 2) visualiza a adição do processo C na memória física. Na memória física (figura 10 - 3) como não tem a possibilidade de adição de 2 *Bytes* em uma vaga de 1 *Byte* de memória, demonstra uma lacuna.

Diante do que foi apresentado sobre a ferramenta, verifica-se que os alunos podem simular algumas situações apresentadas em aula para melhorar a compreensão, também pode usar esta ferramenta para criar diferentes situações. Com isso, os alunos poderão fazer casos específicos em que está tendo dificuldades para que suas dúvidas possam ser abordadas e melhorar o progresso acadêmico.

5 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo apresentar o módulo de simulação do mecanismo de segmentação para o OSLive, aplicando melhorias na sua versão anterior, com adição de novas funcionalidades. Mostrando a nova interface sendo possível comparar com a versão anterior da ferramenta.

As melhorias implantadas na execução deste projeto aumentaram as possibilidades de simulação ao ambiente de simulação do mecanismo de segmentação, padronizando a interface dos módulos ao ambiente OSLive e possibilitando a execução em qualquer máquina somente executando o HTML da ferramenta.

A interface apresentada nessa nova versão, facilita o manuseio da ferramenta por estar seguindo os padrões do OSLive, com o intuito de facilitar o aprendizado do aluno. O desenvolvimento da ferramenta foi realizado de forma fiel ao conteúdo didático, de uma forma atrativa.

Em relação a trabalhos futuros, este trabalho fornece a opção de adicionar exercícios. Utilizando essa ferramenta como base, a criação de exercícios pode ser adicionada com facilidade, adicionando somente a opção de não demonstrar a resposta, com espaço para o usuário colocar suas respostas, assim fazendo a comparação da impressão desta ferramenta com a resposta do usuário.

REFERÊNCIAS

BOOTSTRAP. **Introduction · Bootstrap**. Disponível em:<https://getbootstrap.com/docs/4.1/getting-started/introduction/>. Acesso em: 28 jun. 2021.

JQUERY. **Jquery**. Disponível em: <https://jquery.com>. acesso em: 28 jun. 2021.

MARTINS, Tallyson Ruiter Andrade. **Implementação do módulo de simulação do mecanismo de segmentação do OSLive (TCC)**. Palmas: Ed. ULBRA, 2019.

MEC, MINISTÉRIO DA EDUCAÇÃO. Diretrizes Curriculares Nacionais para os cursos de graduação em Computação. Brasília: [s.n.]. Disponível em: .Acesso em: 06 de abril de 2021.

OLIVEIRA, Rômulo Silva de; DA SILVA CARISSIMI, Alexandre; TOSCANI, Simão Sirineo. **Sistemas Operacionais**. 2001

TANENBAUM, A. S.; BOS, H. **Sistemas Operacionais Modernos**. 4. ed. São Paulo: Pearson Education do Brasil, 2016.

VUEJS. **Vue.js**. Disponível em: < https://vuejs.org>. acesso em: 28 jun. 2021.