UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Arthur Maia de Sousa

Simulador de Gerenciador de Processos para Sistemas Operacionais

Uberlândia, Brasil

2024

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Arthur Maia de Sousa

Simulador de Gerenciador de Processos para Sistemas Operacionais

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Thiago Pirola Ribeiro

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2024

É com grande satisfação que dedico este trabalho a todos os estudantes de ciência da computação, sistemas da informação e áreas afins, que têm paixão pela criação e implementação de soluções inovadoras em sistemas operacionais. Através deste simulador de gerenciador de processos, espero contribuir para a melhoria das práticas de gerenciamento de processos em sistemas operacionais e inspirar novas ideias e desenvolvimentos nessa área. Dedico também este trabalho aos meus amigos, familiares e professores, que me apoiaram em cada passo desta jornada acadêmica.

Agradecimentos

Agradeço meus pais, por me incentivarem a seguir meu sonho de me graduar na área de tecnologia e principalmente por me apoiarem durante o processo do desenvolvimento deste TCC.

Agradeço o meu orientador pela paciência e cordialidade em me repassar direcionamentos e documentos nos quais me ajudaram bastante no desenvolvimento deste TCC.

Agradeço, por fim, meus professores da Universidade Federal de Uberlândia, por serem mais do que responsáveis pelos meus conhecimentos em Sistemas da Informação e pelo meu gosto por esta área.

"Seu tempo é limitado, então, não o gaste vivendo a vida de outra pessoa"
(Steve Jobs)

Resumo

Com o objetivo de facilitar o entendimento dos conceitos da disciplina Sistemas Operacionais para os estudantes, este trabalho aborda o aperfeiçoamento de uma aplicação WEB desenvolvida que simula o funcionamento de um sistema operacional, por meios de processos e o gerenciamento destes. Sua funcionalidade é representada por uma interface intuitiva, na qual permite o usuário configurar seu processador fictício, além de criar, remover, iniciar e suspender processos, visualizando-os em listas atualizadas em tempo real, na qual são ordenadas de acordo do escalonamento escolhido.

O simulador permite configurar o processador da aplicação, sendo possível alterar a Fatia de Tempo, Tempo de espera de Entrada e Saída e seu tipo de escalonamento. Um processo pode ser criado, permitindo o usuário a escolher seu tipo, uma cor para facilitar sua visualização na lista de exibição, uma quantidade para facilitar criação em massa (para que o usuário não tenha o trabalho maçante de criar um processo por vez), além de outros campos que podem variar de acordo com o escalonamento selecionado. Além disso, é possível gerar um gráfico de processos finalizados, permitindo a análise do fluxo de vida do processo, desde sua criação até a sua finalização.

Os resultados obtidos com o desenvolvimento desta aplicação tendem a proporcionar aos estudantes uma compreensão mais clara e prática dos conceitos fundamentais de Sistemas Operacionais, focando especialmente em gerenciamento de processos. Pretende-se que esta aplicação não só auxilie no aprendizado teórico, como também incentive a experimentação e a compreensão prática dos conceitos abordados, além de promover uma maior interação e engajamento dos estudantes, permitindo-lhes explorar diferentes cenários do “sistema operacional” de forma dinâmica e educativa.

Palavras-chave: Sistemas Operacionais, Processos, Simulador, Gerenciamento, Escalonamento

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

CPU Unidade Central de Processamento - *Central Processing Unit*

ESORV Simulador de Ensino de SOs com Realidade Virtual

FCFS Primeiro a Chegar, Primeiro a ser Servido - *First-Come, First-Served*

FIFO Primeiro a Entrar, Primeiro a Sair - *First-In, First-Out*

I/O Entrada/Saída - *Input/Output*

PID Identificador de Processo - *Process Identifier*

SRTN Tempo Restante mais Curto em Seguida - *Shortest Remaining Time Next*

SWSO Simulador Web de Sistemas Operacionais

SSOG Simulador de Sistema Operacional Genérico

TBC-SO/WEB Treinamento Baseado em Computador para Sistemas Operacionais via Web

UFU Universidade Federal de Uberlândia

WTDCC Workshop de Teses e Dissertações em Ciência da Computação

Sumário

1 Introdução

A disciplina de Sistemas Operacionais, inserida nos cursos de tecnologia, geralmente possui uma abordagem mais teórica em comparação com outras disciplinas mais práticas. Devido a essa característica, é comum observar que os estudantes enfrentam certa dificuldade em assimilar o conteúdo apresentado nesta matéria.

A partir desta premissa, surge a ideia que, ao exercitar os conceitos ensinados na disciplina de forma prática, o uso de um simulador possa facilitar a absorção do conteúdo e até mesmo despertar o interesse do estudante acerca dos assuntos abordados.

Por conta disto, este trabalho propõe o aperfeiçoamento de uma aplicação que simula um gerenciador de processos dentro de um sistema operacional, dando uma representação abrangente e realista, permitindo ao usuário (no caso, o estudante da disciplina) visualizar e estudar detalhadamente os diferentes métodos de abordagem de escalonamento que podem envolver os processos.

Vale ressaltar que este trabalho dará continuidade na pesquisa de Freitas (2023), propondo a implementação de outros algoritmos de escalonamento de processos no sistema operacional, o "re-design" da interface do sistema e a possibilidade de gerar um gráfico do fluxo de execução dos processos que já finalizaram.

1.1 Objetivos

Este trabalho tem como principal objetivo a continuidade no desenvolvimento de uma aplicação WEB (FREITAS, 2023), que replica conceitos/funcionalidades referentes ao gerenciamento de processos de um sistema operacional, afim de facilitar o entendimento do conteúdo abordado na disciplina Sistemas Operacionais. Com isso, será possível permitir ao estudante visualizar e interagir com os conceitos ministrados em sala de aula, além de contar com uma interface intuitiva, facilitando a utilização da aplicação.

Referente a aplicação, se enquadram as funcionalidades de criar, editar e deletar processos; exibir os processos criados com atualização de dados em tempo real; configuração do processador da simulação e; gerar um gráfico do fluxo de execução dos processos.

Os objetivos específicos deste trabalho são:

- Analisar os princípios e algoritmos de gerenciamento de processos em sistemas operacionais, especialmente no escalonamento de processos implementando-os de forma visual/simulada;

- Realizar um levantamento de requisitos que já foram abordados na pesquisa de Freitas (2023) e, com isso, adicionar mais funcionalidades e características específicas de um simulador;
- Desenvolver uma aplicação com uma interface amigável e intuitiva, permitindo o usuário usufruir da aplicação da melhor forma possível;
- Permitir ao usuário a seleção de diferentes algoritmos de escalonamento, desde também diferentes configurações de processadores, tendo assim vários cenários e métricas de desempenho.

1.2 Justificativa

Este trabalho foi desenvolvido para ajudar os estudantes da disciplina Sistemas Operacionais que possuem dificuldades em abstrair os conceitos de processos dentro de um Sistema Operacional. Com a possibilidade do usuário (no caso, o estudante) executar várias funcionalidades dentro de um Sistema Operacional simulado, este verá, na prática, como processos de diferentes tipos e métodos de escalonamento funcionam.

Esta aplicação foca na "metodologia ativa", na qual insere o estudante em um ambiente realista, afim de incentivá-lo a aprender de forma autônoma e participativa, por meios de situações reais.

1.3 Organização da Monografia

O Capítulo 2 apresenta uma fundamentação teórica sobre o assunto, abordando conceitos referente a Sistemas Operacionais, gerenciamento e escalonamento de processos, além de abordar quais tecnologias foram usadas para o desenvolvimento da aplicação e apresentar uma comparação com outros projetos semelhantes que já foram desenvolvidos por outros autores.

O Capítulo 3 apresenta a estrutura de cada elemento trabalhado dentro do simulador, além de apresentar qual metodologia foi abordada no desenvolvimento da aplicação.

O Capítulo 4 apresenta uma explicação detalhada de cada funcionalidade do sistema, descrevendo um passo a passo intuitivo com imagens, para que o usuário possa entender como pode usufruir da melhor forma o sistema, além de abordar quais funcionalidades foram reaproveitas, removidas e adicionadas da pesquisa (FREITAS, 2023).

O Capítulo 5 apresenta a conclusão do trabalho, onde descreve as principais contribuições deste trabalho, além de abordar sugestões para futuras pesquisas.

2 Fundamentação Teórica

Neste capítulo será abordado o referencial teórico da aplicação desenvolvida neste trabalho, no qual consiste em um simulador de um sistema operacional, dando foco ao gerenciamento de processos. Para melhor entendimento, inicialmente será explicado o que é um simulador e sobre o assunto principal, que é Sistema Operacional.

Posteriormente, são destacados conceitos importantes sobre Sistema Operacional, incluindo características abordadas no simulador. Entre elas, são: o gerenciamento de processos; diferentes configurações de processador; e diferentes tipos de escalonamentos. Também serão abordados as tecnologias e ferramentas escolhidas para o desenvolvimento do simulador e o porquê de serem escolhidas.

A partir da explicação destas referencias teóricas e tecnológicas, é possível aprofundar o entendimento do funcionamento do simulador proposto neste trabalho e sua importância de estudo e análise em um ambiente controlado e simulado.

2.1 Simulador

Um simulador é uma aplicação de computador criada especialmente para imitar ou reproduzir o comportamento de um determinado sistema ou processo, permitindo aos usuários que o utilizam experienciar como este sistema ou processo se comportaria em diferentes cenários.

É uma ferramenta virtual que permite os usuários estudarem e experimentarem um sistema ou processo em um ambiente controlado, ajudando-os a compreender seu comportamento e aprimorar seu desempenho.

No contexto deste trabalho, o simulador seria responsável por reproduzir o comportamento de um sistema operacional, focando especialmente no gerenciamento dos processos dentro daquele ambiente.

Além disso, o simulador desenvolvido neste trabalho se trata de uma aplicação WEB, enfatizando a interatividade e acessibilidade deste. Trabalhos como Treinamento Baseado em Computador para Sistemas Operacionais via Web (TBC-SO/WEB) (REIS, 2009) e Simulador Web de Sistemas Operacionais (SWSO) (OLIVEIRA, 2015) também abordam esta estratégia, o que amplia o acesso e a facilidade para vários usuários.

2.2 Sistema Operacional

Um Sistema Operacional é um software que atua como um intermediário entre o hardware do computador e as aplicações/programas que são executados nele. Tem como principal função fornecer aos programas do usuário um modelo do computador melhor, simples e limpo, assim como lidar com o gerenciamento de vários recursos.

Em outras palavras, o Sistema Operacional serve para abstrair todas as funcionalidades que o hardware nos permite usar e, com isso, disponibiliza de programas que executam todas as funcionalidades do hardware, porém de uma forma mais amigável e com interfaces mais amigáveis (Figura 1). Essa abstração é o principal ponto para gerenciar toda a complexidade que envolve o hardware, transformando uma tarefa praticamente impossível em duas tarefas gerenciáveis.

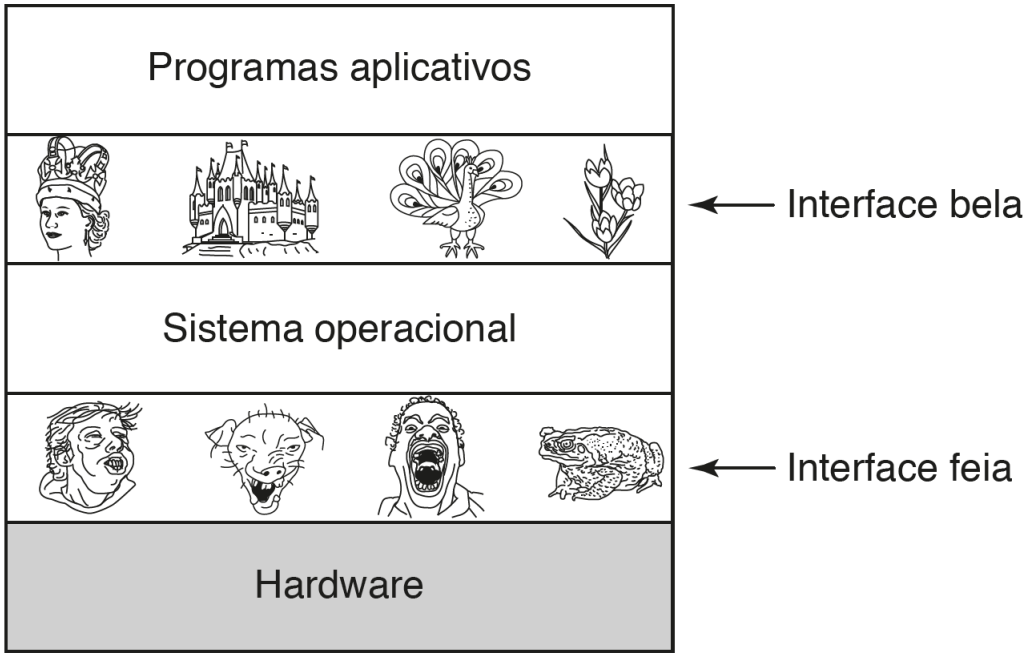


Figura 1 – Abstração do Sistema Operacional em cima do Hardware.

Fonte: Extraído de (TANENBAUM, 2010)

2.2.1 Processador

Conforme (TANENBAUM, 2010), o processador, mais conhecido como Unidade Central de Processamento (CPU) é o núcleo do computador. É responsável por buscar instruções da memória e executá-las. Seu ciclo básico consiste em buscar a primeira instrução da memória, decodificá-la (obtendo seu tipo e operandos), executá-la e assim por diante. Este ciclo é repetido até que o programa termine.

2.2.1.1 Configurações

Neste trabalho, é disponibilizado algumas configurações feitas diretamente ao processador, com o intuito de medir o desempenho do mesmo. Estas configurações, dependendo do valor imposto a eles, podem mudar toda a usabilidade e comportamento dos escalonamentos dos processos criados. Estas configurações são:

- Tempo de Espera de Entrada/Saída (I/O): estipula o tempo em que o processo do tipo I/O aguardará por uma Entrada/Saída;
- Fatia de Tempo: estipula o tempo em que o processador executará cada processo;
- Tipo de Escalonamento: altera a forma de escalonamento dos processos criados e que estão em execução na CPU.

2.3 Gerenciamento de Processos

O Gerenciamento de Processos consiste na representação e controle de processos pelo Sistema Operacional. Como o processador é compartilhado entre todas as aplicações que estão em execução, é necessário ter um gerenciamento dos processos que estão em execução, para que tanto o processador quanto os dispositivos de Entrada/Saída sejam utilizados de forma eficiente.

2.3.1 Processo

Conforme (STALLINGS, 2014), um processo é definido como uma instância em execução de um programa no sistema operacional, incluindo os valores atuais do contador do programa, registradores e variáveis.

É uma entidade dinâmica que representa a execução de um programa específico em um sistema operacional, com seu próprio contexto de execução e recursos associados.

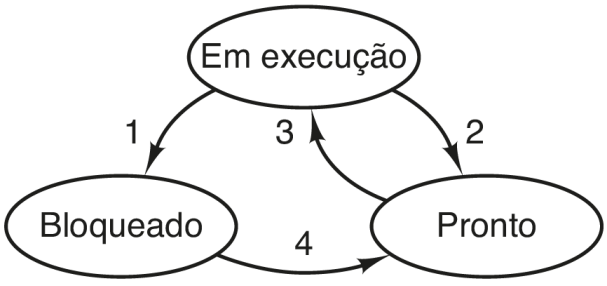
Durante seu ciclo de execução, este se passa por vários estados, sendo estes:

- Em execução: quando o processo está ativamente utilizando a CPU para executar suas instruções;
- Em I/O: quando o processo está ativamente engajado em operação de Entrada/Saída;
- Suspenso (ou Bloqueado): quando o processo não pode continuar sua execução até que algum evento externo aconteça;
- Pronto: quando o processo está pronto para executar mas segue aguardando a disponibilidade da CPU;

- Pronto para I/O: quando o processo está pronto para executar uma operação de Entrada/Saída, mas segue aguardando por acesso ao dispositivo ou algum recurso necessário;
- Finalizado: quando o processo finaliza sua execução.

Na Figura 2 é representado um diagrama de estados ilustrando o ciclo de vida de um processo de um sistema operacional.

- Transição 1: quando um processo no estado **Em execução** necessita de um recurso que não está disponível no momento (como Entrada/Saída, por exemplo), este será movido para o estado **Bloqueado** até que o recurso requerido esteja disponível.
- Transição 2: quando um processo no estado **Em execução** é interrompido pelo escalonador de processos para dar chance a outro processo de executar, este retorna para o estado **Pronto**.
- Transição 3: quando o escalonador de processos seleciona o processo no estado **Pronto** para ser o próximo a utilizar a CPU, este será movido para o estado **Em execução**.
- Transição 4: quando um processo no estado **Bloqueado** se torna disponível (como, por exemplo, a operação de Entrada/Saída foi contemplada), este processo retorna para o estado **Pronto**, aguardando a próxima oportunidade de ser executado pela CPU.



1. O processo é bloqueado aguardando uma entrada
2. O escalonador seleciona outro processo
3. O escalonador seleciona esse processo
4. A entrada torna-se disponível

Figura 2 – Ciclo de Vida de um processo.

Fonte: Extraído de (TANENBAUM, 2010)

2.3.1.1 Tipos de Processos

Tendo em vista que o processo é uma instância de um programa, quando um sistema operacional é executado, é criado uma série de processos. Destes inúmeros processos, existem processos de primeiro plano, destinados para os usuários interagirem e realizarem seus devidos propósitos. Existem também processos de segundo plano, que não são associados a um usuário, mas são necessários para que o sistema funcione de forma correta.

No simulador são abordados 3 tipos de processos, sendo permitido ao usuário definir seu tipo no momento da criação do processo:

- CPU-bound: tem longos períodos de uso da CPU e esporádicas esperas de I/O;
- I/O-bound: tem curtos períodos de uso da CPU e esperas de I/O frequentes;
- Ambos (CPU-bound e I/O-bound): mesclagem dos dois tipos.

2.3.1.2 Atributos

Como esta aplicação se trata de um simulador, há atributos que serão definidos para cada processo no momento da criação e estes serão de extrema importância para todo o processo de listagem e gerenciamento, sendo:

- Identificador de Processo (PID): número único usado para identificar um processo;
- Prioridade: prioridade em cima de outros processos que estão na fila de execução;
- Tempo de execução: tempo que foi definido para que o processo seja executado;
- Quantidade: quantidade de processos a serem criados;
- Tipo: tipo do processo (subseção 2.3.1.1).

2.4 Escalonamento

Para que os processos sejam executados pela CPU, deve-se ter uma forma de selecionar qual processo será executado, consecutivamente.

O Escalonamento de Processos é algo importante, pois garante um bom desempenho geral, tendo mais eficiência na utilização do processador. Por conta disso, acarreta um melhor tempo de resposta. Ainda é mais importante em sistemas de multiusuário ou multitarefas, os quais exigem a execução de processos para diferentes entidades, em momentos diferentes. Logo, é necessária uma distribuição justa do tempo do processador para cada processo, evitando que um processo esteja parado por conta de outro processo, sendo que ambos devem ser executados ao mesmo tempo.

Referente aos algoritmos de escalonamento descritos nesta seção, e que serão os utilizados no simulador, foram descritos e baseados no livro do Tanenbaum (2010):

- Escalonamento em Lote e,
- Escalonamento Interativo.

2.4.1 Escalonamento em Lote

O Escalonamento em Lote envolve o processamento de várias tarefas reunidas em um grupo, sendo uma maneira econômica de processar grandes quantidades de dados em um curto período. Por conta disso, há velocidade e economia, quando envolvemos custos operacionais.

2.4.1.1 Primeiro a chegar, primeiro a ser servido

Sendo um dos algoritmos de escalonamentos mais simples, o algoritmo Primeiro a Chegar, Primeiro a ser Servido (FCFS) é atribuído a CPU os processos na ordem em que foram requisitados, seguindo uma fila única de processos prontos. Ou seja, quando um processo é bloqueado, o próximo da fila é executado.

Por mais simples que seja, apresenta uma desvantagem quando se trata de processos limitados pela computação pela entrada/saída na mesma fila, pois processos limitados pela computação possuem um tempo pré-estabelecido para serem executados. Embora existem processos que dependem de entrada/saída, ou seja, depende de uma ação terceira, os próximos processos que não dependem estarão pausados. Isso pode acarretar um tempo de espera grande e sem necessidade.

No simulador, este tipo de escalonamento será abordado como Primeiro a Entrar, Primeiro a Sair (FIFO), ao invés de FCFS, por ser um jargão utilizado na área e para facilitar o entendimento do usuário.

2.4.1.2 Tempo Restante mais Curto em Seguida

Este algoritmo Tempo Restante mais Curto em Seguida (SRTN), envolvendo uma lista de vários processos com tempos pré-estabelecidos para serem executados e resolvidos, escolhe o processo com o tempo de execução restante mais curto para ser executado. Ao ser finalizado, é reavaliado entre todos os processos listados qual tem o tempo de execução restante mais curto e assim então, executá-lo.

Neste projeto, por questões de performance, foi implementado numa versão não preemptiva, aonde que o próximo processo a ser executado só será selecionado quando o processo atual finalizar, diferente da versão preemptiva que, enquanto um processo está

em execução, caso contenha outro processo com tempo de execução restante mais curto, o processo atual é suspenso.

2.4.2 Escalonamento Interativo

O Escalonamento Interativo envolve o escalonamento de processos priorizando a interatividade do usuário, dando ênfase em respostas rápidas e eficientes às suas interações, priorizando processos envolvidos em atividades de I/O ou processos que requerem a interação do usuário, sendo estes executados com maior frequência e tendo maior prioridades em cima de outros processos em execução.

Alguns algoritmos para Escalonamento Interativo são destaques e, dentre eles, são: o Escalonamento por Chaveamento Circular e o Escalonamento por Prioridades.

2.4.2.1 Escalonamento por Chaveamento Circular

Sendo um dos algoritmos de escalonamento mais antigos, o Escalonamento por Chaveamento Circular (também chamado de Escalonamento Circular), funciona designando uma fatia de tempo para um processo (subseção 2.2.1.1). Este intervalo serve de métrica para, caso o processo ainda esteja em execução após o final do intervalo, a CPU sofrerá uma preempção (ou seja, interrompido antes de concluir sua execução), e seguirá para a fila de pronto e o próximo processo será escalonado.

O diferencial deste algoritmo é o fato que o escalonamento acontece por conta do comprimento do intervalo, pois cada processo é chaveado. Este meio de chavear um processo para outro exige tempo para fazer sua administração, desde salvar e carregar registradores e mapas de memória, atualizar tabelas e listas, carregar e descarregar cache, entre outros.

2.4.2.2 Escalonamento por Prioridades

Diferente do algoritmo anterior, que julga a importância dos processos por meio do seu intervalo, este algoritmo define por níveis de prioridade. Ou seja, cada processo é designado uma prioridade e, a partir disso, o processo com prioridade mais alta é alocada para a execução. Após a finalização deste processo, o próximo com maior prioridade é executado.

Em casos de múltiplos processos, para evitar que processos de alta prioridade executem indefinidamente, o escalonador pode diminuir a prioridade do processo, ocorrendo um chaveamento de processo e seguindo para o próximo. Em contrapartida, pode ser definido para cada processo um intervalo de tempo máximo em que ele pode ser autorizado a executar e, caso este intervalo esgote, o processo seguinte com maior prioridade passa a ser executado.

2.5 Tecnologias

Para o desenvolvimento deste simulador e a implementação dos algoritmos citados, foram escolhidas tecnologias e linguagens de programações para seu desenvolvimento.

2.5.1 Javascript

Javascript é uma linguagem de programação leve, interpretada e baseada em objetos. Utilizada bastante em aplicações WEB, esta linguagem é baseada em protótipos, multi-paradigma e dinâmica, suportando estilos de orientação a objetos, imperativos e declarativos. Foi a linguagem escolhida pois, além de ter uma boa curva de aprendizagem, é altamente recomendada para navegadores, especialmente pelo fato do simulador desenvolvido ser uma aplicação WEB.

2.5.2 Angular

Angular é uma plataforma de aplicações WEB feita especialmente para desenvolver aplicações em diversas plataformas, mantido e desenvolvido pela Google.

Baseada em *Typescript*, é uma ferramenta potente, possuindo quase tudo que é preciso para o desenvolvimento de uma aplicação de forma nativa e tendo várias vantagens:

- Fácil aprendizagem, usabilidade e testabilidade;
- Suporte do Google e comunidade;
- Interface de Usuário declarativa;
- Consistência e reutilização do código.

2.5.3 Typescript

Typescript é uma linguagem de programação criada a partir do *Javascript*, adicionando funcionalidades que não são disponíveis nativamente ou que requer um esforço para utilização, como encapsulamento, herança, abstração e polimorfismo.

Com a utilização desta linguagem, é permitido ao desenvolvedor escrever linhas de código com a utilização de tipagem estática, orientação a objetos e uma escrita de código com sintaxe de fácil compreensão.

2.6 Estado da Arte

O Estado da Arte para o desenvolvimento de um Simulador de um Gerenciador de Processos para Sistemas Operacionais envolve o estudo em áreas de pesquisa que visam o entendimento de compostos necessários para a utilização da aplicação.

Este documento é um aprimoramento do estudo e aplicação desenvolvida por Freitas (2023). Por conta disso, muito da base que foi desenvolvido no documento anterior foi reaproveitado e readaptado para receber novos algoritmos, já que o simulador anterior usa apenas de um escalonamento interativo por prioridades.

Foram estudados e abordados vários temas para o entendimento e usabilidade do simulador desenvolvido, sendo:

- Utilização de simuladores na educação;
- Conceitos de Sistemas Operacionais;
- Algoritmos de escalonamento de processos;
- Configurações de desempenho de processador;
- Outros simuladores que abordam o mesmo assunto.

2.7 Trabalhos Correlatos

Os seguintes projetos citados são de simuladores de sistemas operacionais que possuem gerenciamento de processos e outros pontos que não foram abordados neste projeto. Estes projetos são muitos semelhantes ao que foi desenvolvido neste documento.

2.7.1 SOsim

SOsim é um software educacional para ser utilizado como ferramenta de apoio em aulas de sistemas operacionais. Ele permite visualizar os conceitos de multiprogramação, processo e suas mudanças de estado, gerência do processador e a gerência de memória virtual (MAIA, 2001).

Suas principais características são:

- Implementa o conceito de processo e escalonamento;
- Permite visualizar estruturas internas do sistema;
- Possui gerenciamento do processador (Figura 3);
- Possui gerenciamento de memória.

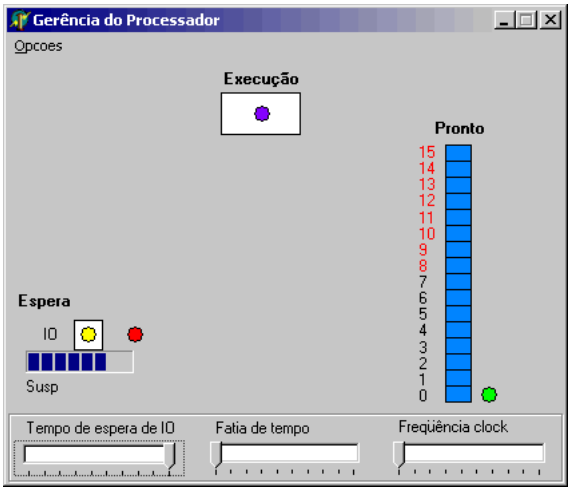


Figura 3 – Gerenciamento do Processador no SOsim.

Fonte: Extraído de (MAIA, 2001)

2.7.2 SWSO

O SWSO é uma ferramenta web criada para auxiliar o ensino-aprendizagem da disciplina de sistemas operacionais e visa simular o funcionamento de um Sistema Operacional, explorando os conceitos de gerência de disco, memória e processos de forma integrada e interativa (Figura 4) (OLIVEIRA, 2015).

Suas principais características são:

- Sistemas de Arquivos;
- Gerenciamento de Memória;
- Gerência de Processos.

2.7.3 ESORV

O Simulador de Ensino de SOs com Realidade Virtual (ESORV) é um software desenvolvido a partir dos princípios dos escalonadores de processos. Tendo suporte a Realidade Virtual, o sistema permite que o usuário se sinta imerso "em um computador", através de um ambiente imersivo e tridimensional que simula as técnicas de escalonamento de processos (Figura 5) (SCAMATI, 2017).

2.7.4 SSOG

O Simulador de Sistema Operacional Genérico (SSOG) é um software didático que aborda alguns conteúdos de Sistemas Operacionais, desde gerência de processos, processador, memória e animações. Possui uma interface gráfica simples, intuitiva e de fácil manuseio (Figura 6) (KIOKI, 2008).

Figura 4 – Criação de uma simulação no SWSO.

Fonte: Extraído de (OLIVEIRA, 2015)

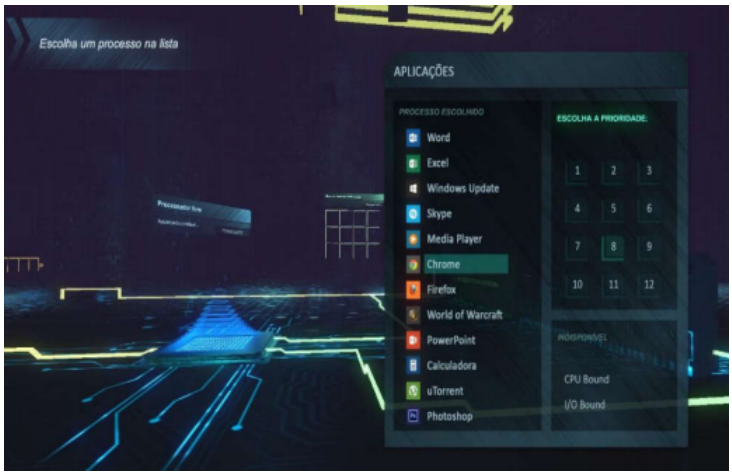


Figura 5 – Ambiente virtual do simulador ESORV.

Fonte: Extraído de (SCAMATI, 2017)

2.7.5 TBC-SO/WEB

O TBC-SO/WEB é um software educativo e ferramenta de ensino das políticas de gerência de processos e de gerência de memória em sistemas operacionais (REIS, 2009).

Suas principais características são:

- Gerenciamento de Processos (Figura 7);
- Gerenciamento de Memória;

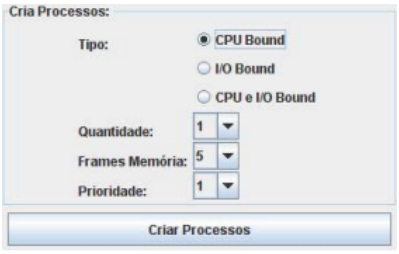


Figura 6 – Criação de Processos no SSOG.

Fonte: Extraído de (KIOKI, 2008)

- Gerenciamento de Dispositivos de Entrada/Saída;
- Gerenciamento de Sistema de Arquivos.

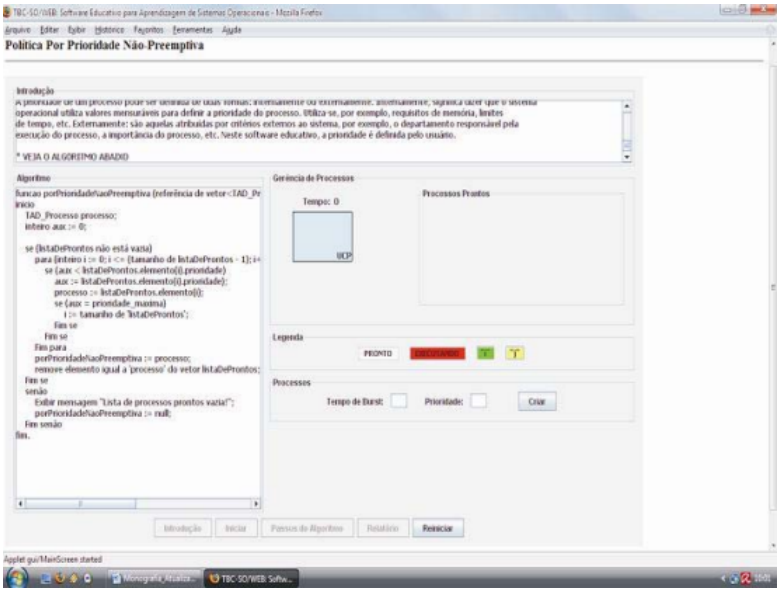


Figura 7 – Gerência de Processos no TBC-SO/WEB.

Fonte: Extraído de (REIS, 2009)

2.7.6 Comparativo com os outros trabalhos

Na Tabela 1 é possível visualizar, de uma forma clara e concisa, quais funcionalidades estão presentes em diferentes sistemas educacionais de simulação de sistemas operacionais, incluindo o SimulateOS. É notável que sistemas como SWSO e TBC-SO/WEB oferecem um amplo leque de funcionalidades, cobrindo todos os aspectos avaliados, já outros com um foco mais restrito. O SimulateOS, por exemplo, se destaca por focar especialmente em gerência de processos e de processador e ser uma aplicação web, permitindo uma maior acessibilidade e interatividade para o usuário. Em contraste, sistemas como o ESORV focam apenas no gerenciamento de processos, o que pode limitar sua aplicação

em ambientes de ensino mais dinâmicos e interativos. Por fim, o principal ponto que o SimulateOS se difere dos outros trabalhos é sua interface simples e intuitiva, o que favorece a usabilidade e experiência do usuário.

Tabela 1 – Tabela com comparações do SimulateOS com outras aplicações semelhantes.

Sistemas/Características	Gerência de Processos	Gerência de Processador	Gerência de Memória	Gerência de Sistema de Arquivos	Sistema WEB
SimulateOS	X	X			X
SOsim	X	X	X		
SWSO	X	X	X	X	X
ESORV	X				
SSOG	X	X	X		
TBC-SO/WEB	X		X	X	X

3 Estruturação do software

Neste tópico, é explicado a estrutura de cada elemento trabalhado dentro deste simulador. São eles: processo, processador e registros. Além disso, é abordado a metodologia aplicada no desenvolvimento da aplicação.

3.1 Processo

O processo, elemento principal da nossa aplicação, é composto pelos seguintes campos:

- Identificador de Processo (PID) (`id`): o identificador é único para cada processo. Por ser um simulador, é criado um número randômico para cada processo criado;
- Prioridade (`priority`): um valor numérico para identificar qual o nível de prioridade do processo (usado apenas pelo algoritmo de escalonamento "Circular com Prioridades");
- Cor (`color`): a cor do processo. Importante para ajudar a identificar o processo dentro da lista de processos;
- Tipo (`type`): o tipo do processo (subseção 2.3.1.1);
- Estado (`state`): o estado que o processo está atualmente (subseção 2.3.1);
- Tempo de CPU (`cpuTime`): um valor numérico usado para controlar o tempo que um processo esteve na CPU;
- Tempo de Criação (`timeCreated`): tempo do processador no momento que foi criado o processo. Este dado será importante para gerar os registros;
- Unidade de tempo (`processTimeToFinish`): quantidade de unidades de tempo que o processo ficará em CPU para ser finalizado. Principal atributo para controlar o fluxo de execução e finalização de um processo;
- Tempo de Execução (`executingTime`): atributo criado para controlar o tempo que um processo está sendo executado a partir da fatia de tempo definida no processador (subseção 2.2.1.1);
- Atual Tipo (`currentType`): o tipo atual do processo. Este dado é importante quando um processo é do tipo "CPU-bound e I/O-bound".

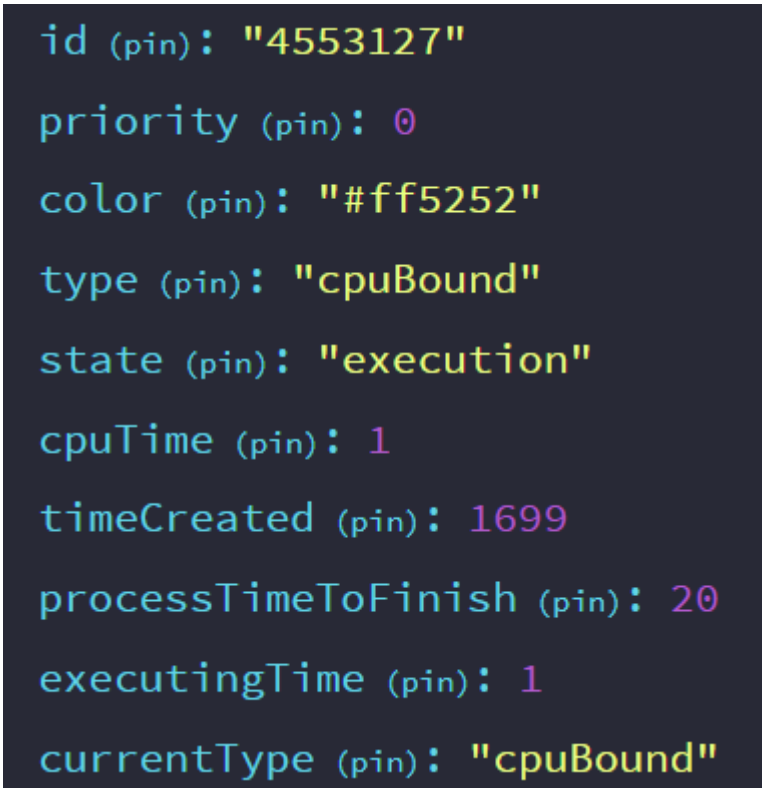


Figura 8 – Campos do processo dentro da aplicação.

3.2 Estrutura do registro

O registro, elemento criado especialmente para o recurso de gerar gráficos contendo o fluxo de vida de um processo dentro de uma CPU, é composto pelos seguintes campos:

- ID (`id`): identificador único do registro. Criado apenas para garantir não duplicidade de registros;
- Processo (`process`): o processo, no momento em que foi gerado o registro. Nele, conterá todas as informações do processo, importantes para gerar o gráfico;
- Tempo Atual (`currentTime`): o tempo atual do processador, no momento em que foi gerado o registro.

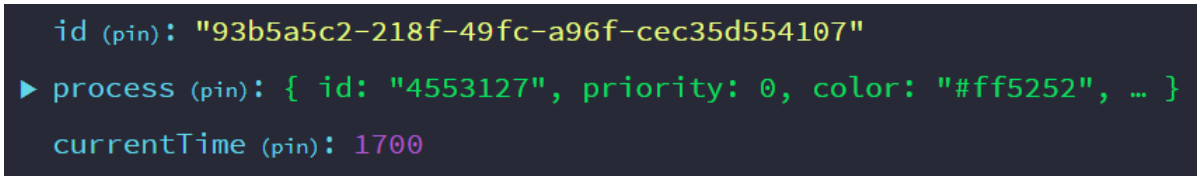


Figura 9 – Campos do registro dentro da aplicação.

3.3 Processador

O processador, elemento responsável para controlar o escalonamento dos processos, é composto pelos seguintes campos:

- Dados (`data`): lista dos processos criados desde que o processador foi iniciado;
- Cores (`colors`): possíveis cores que um processo pode ter;
- Temporizador (`timer`): um contador de tempo, aonde é incrementado ao longo do escalonamento dos processos;
- Tempo de Espera I/O (`ioWaitTime`): o tempo de espera para processos de tipo I/O-bound antes de finalizarem ou retornarem ao estado de "Pronto";
- Fatia de Tempo (`timeSlice`): tamanho da fatia de tempo para o escalonamento dos processos;
- Tipo de Escalonamento (`scalingType`): o tipo de escalonamento dos processos;
- Colunas visíveis (`displayedColumns`): uma lista de colunas, no qual é usado para controlar quais campos serão mostrados nas listagens dos processos.

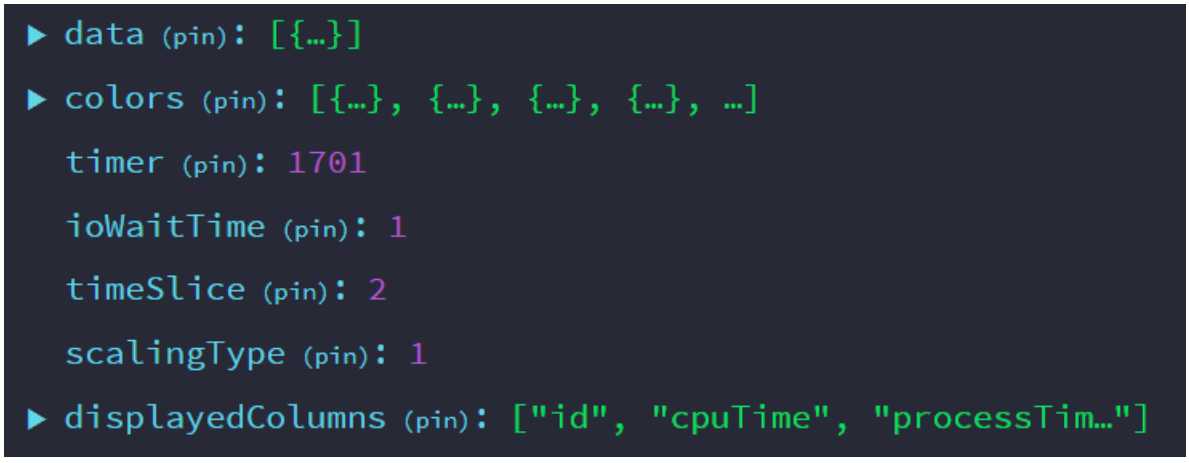


Figura 10 – Campos do processador dentro da aplicação.

3.4 Depuração

Para o desenvolvimento desta aplicação, foi necessário utilizar ferramentas para monitorar o estado da aplicação e facilitar o processo de depuração, no qual desempenhou algo essencial na identificação e correção de erros, garantindo eficiência e estabilidade do sistema desenvolvido.

Para isso, foi instalado a extensão de navegador “Redux DevTools” (BIEREMA, 2015), no qual permitiu identificar os valores das entidades citadas anteriormente.

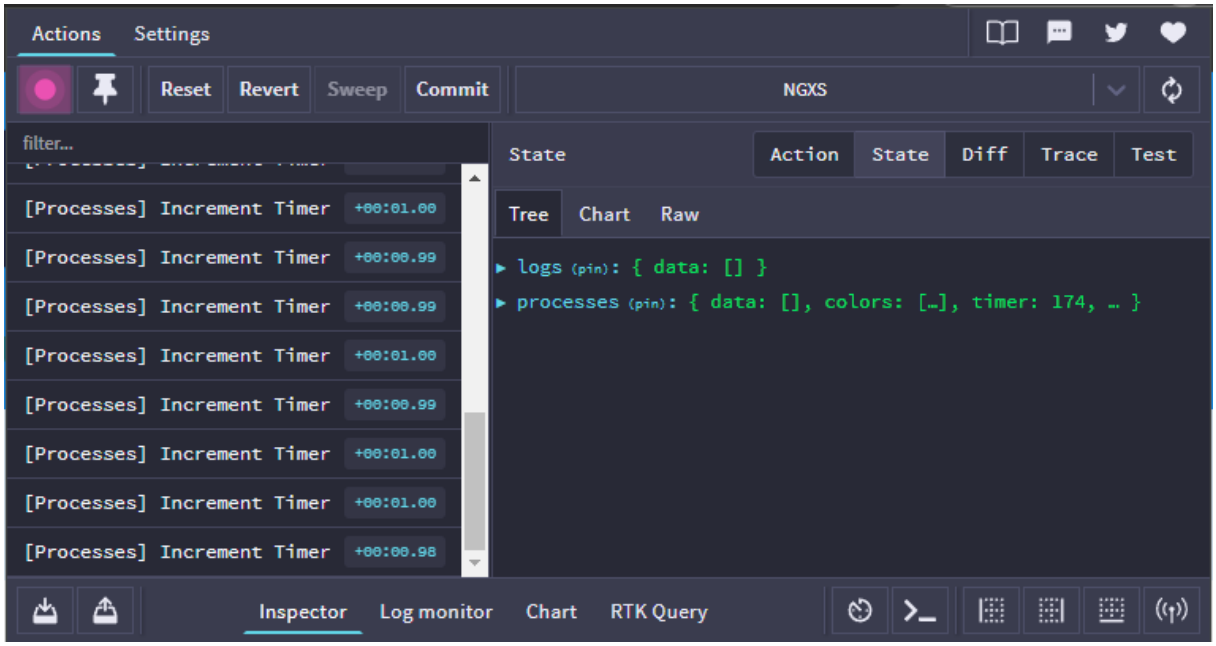


Figura 11 – Extensão usada para depurar alterações de estado do aplicativo.

3.5 Metodologia

O desenvolvimento e aperfeiçoamento deste simulador foi realizado utilizando uma abordagem ágil de desenvolvimento de software, que é a metodologia ágil SCRUM. Por mais que esta metodologia seja recomendada para trabalhos em equipe, foi feita a escolha dessa metodologia pelo fato de que ela permite uma maior flexibilidade e adaptação às mudanças, além de ser muito comum a utilização dela em empresas de desenvolvimento de software.

Para implementação do simulador, foi utilizado o *Angular*, uma plataforma de desenvolvimento de aplicações web, que oferece diversas ferramentas e bibliotecas para a criação de interfaces responsivas e dinâmicas. Esta plataforma é movida pela linguagem de programação *Javascript* e *Typescript*.

A validação do simulador foi realizada através de testes de Simulação de Cenários, que foram executados tanto por mim quanto pelo meu orientador. Os resultados foram analisados e utilizados para identificar possíveis erros e aprimoramentos no simulador.

É importante destacar que a escolha dessas metodologias foi feita com base nos objetivos do projeto e nas necessidades do ambiente em que ele seria desenvolvido. Além disso, foram consideradas as limitações e possíveis fontes de viés da metodologia escolhida, buscando minimizar possíveis problemas e garantir a qualidade e a confiabilidade do simulador.

Por fim, é importante ressaltar que a utilização do modelo ágil SCRUM em conjunto com a tecnologia *Angular* e os testes de validação foram fundamentais para o sucesso do projeto, garantindo um desenvolvimento mais eficiente e colaborativo e um produto final de alta qualidade.

4 Experimentos e Análise dos Resultados

Neste capítulo serão apresentados experimentos realizados na aplicação, visando demonstrar suas funcionalidades. Para facilitar o entendimento dos experimentos, a partir de cada funcionalidade experimentada, seguirá de uma breve explicação, seguida de imagens do sistema e da funcionalidade em questão.

Lembrando que, como este trabalho é baseado em outro projeto (FREITAS, 2023), muitos tópicos e figuras serão semelhantes. Serão notados algumas diferenças em telas já desenvolvidas, nas quais serão detalhadas no final deste capítulo.

4.1 O Sistema

A aplicação consiste em um simulador de gerenciamento de processos de um Sistema Operacional. É possível criar, editar, suspender e excluir processos, além de configurar o "processador" e até mesmo gerar um gráfico para visualizar o ciclo de vida do processo.

Para fins de uso da aplicação, está foi prontamente disponibilizada no site do SimulateOS[1] pelo meu orientador (Figura 12). Além disso, para fins de estudo e/ou continuação da pesquisa, foi disponibilizado o repositório público do código da aplicação[2].

4.2 Processador

Neste simulador, o processador é uma unidade de execução fictícia responsável por executar os processos. Nesta aplicação, é possível efetuar algumas configurações do processador, como:

- Escalonamento (seção 2.4): responsável por definir qual tipo de escalonamento será aplicado no gerenciamento dos processos. Seu valor padrão é o escalonamento "Circular";
- Tempo de Espera de I/O: responsável por definir quantas unidades de tempo um processo do tipo "I/O-bound" permanecerá em I/O. Valor padrão é 1, sendo $1 \leq$ Tempo de Espera de I/O ≤ 10;

[1] Endereço para acessar o simulador: https://www.facom.ufu.br/~thiago/simulateOS/
[2] Endereço do repositório: https://github.com/arthurmaia/TCC

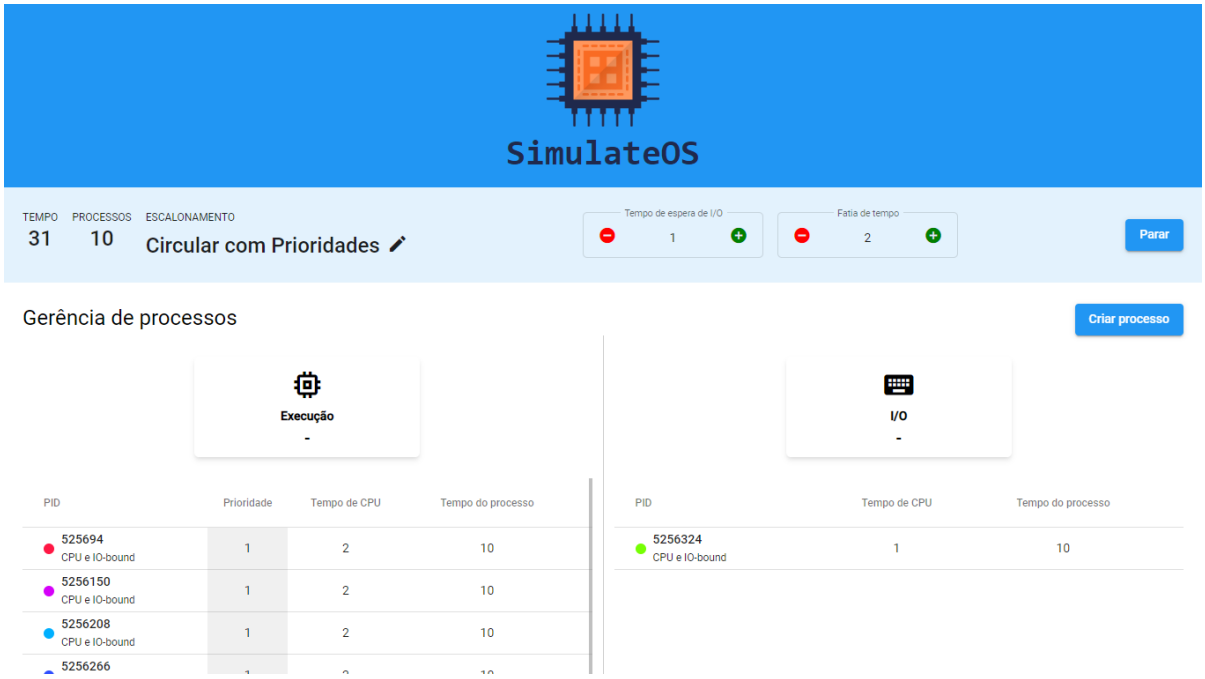


Figura 12 – Página inicial da aplicação.

- Fatia de Tempo: responsável por definir quantas unidades de tempo um processo ficará em execução no processador. Valor padrão é 2, sendo $2 \leq$ Fatia de Tempo ≤ 10.

No cabeçalho da página, logo abaixo da logo da aplicação, está uma barra contendo informações sobre o processador (Figura 13):

- `Tempo`: o tempo decorrido desde a inicialização do processador. Este tempo define a quantidade de unidades de tempo que se passou;
- `Processos`: contador de processos não finalizados;
- `Escalonamento`: o tipo de escalonamento atual do processador com um ícone que, ao clicar, é aberto uma janela, permitindo o estudante a selecionar qual tipo de escalonamento desejado;
- `Tempo de Espera de I/O`: o valor atual do tempo de espera, com botões de ações para incrementar e decrementar o valor;
- `Fatia de tempo`: o valor atual da fatia de tempo, com botões de ações para incrementar e decrementar o valor;
- Botão `Parar`: um botão que, ao clicar, reinicia o tempo, apagando todos os processos já criados, além de retornar as configurações do processador para seus valores padrões.

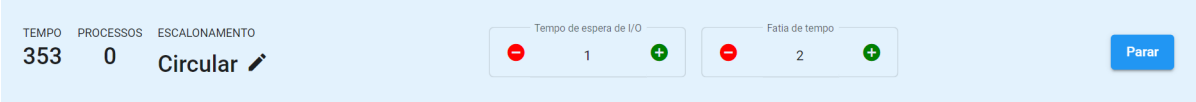


Figura 13 – Cabeçalho contendo informações e campos/botões de ações do processador.

Ao clicar no ícone (lápis) para editar o tipo de escalonamento, é apresentada uma janela (Figura 14) contendo um texto informando que, ao trocar o tipo de escalonamento, todos os processos serão apagados e reiniciados (semelhante à ação do botão `Parar`), seguido por um campo selecionável, contendo todos os possíveis tipos de escalonamento suportados pela aplicação (Figura 15). Selecionando o tipo de escalonamento e clicando no botão `Ok`, a janela fechará, atualizando o tipo de escalonamento na barra do processador e todo o progresso feito na aplicação será reiniciado. Caso o estudante não queira atualizar o tipo de escalonamento, ou talvez tenha clicado por engano, pode clicar no botão `Cancelar`, no qual fechará a janela e nenhuma alteração será feita.

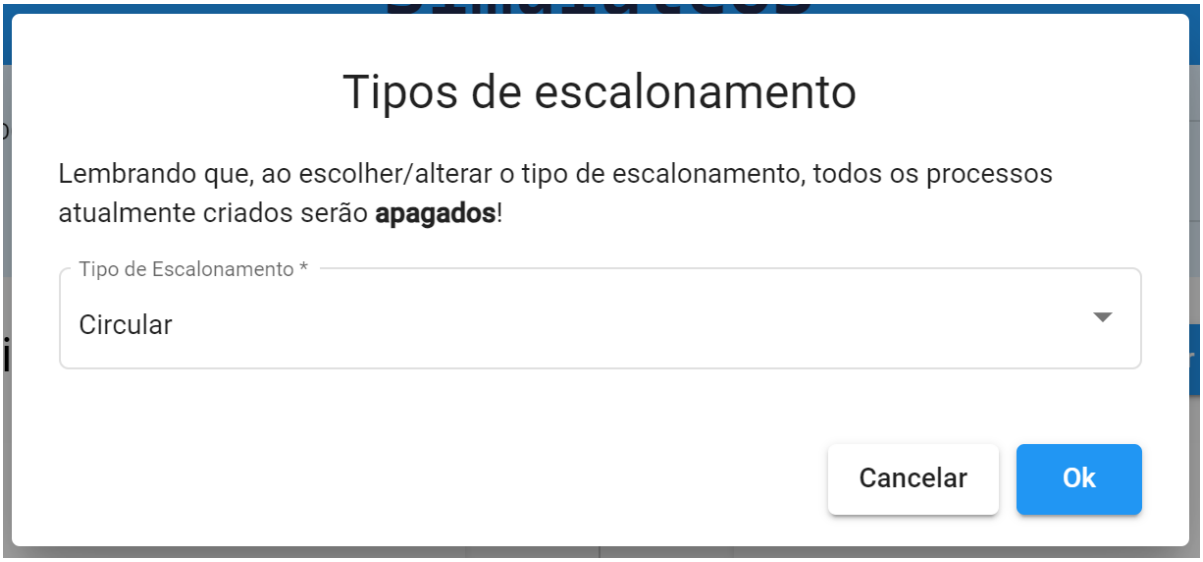


Figura 14 – Janela que permite alterar o tipo de escalonamento do processador.

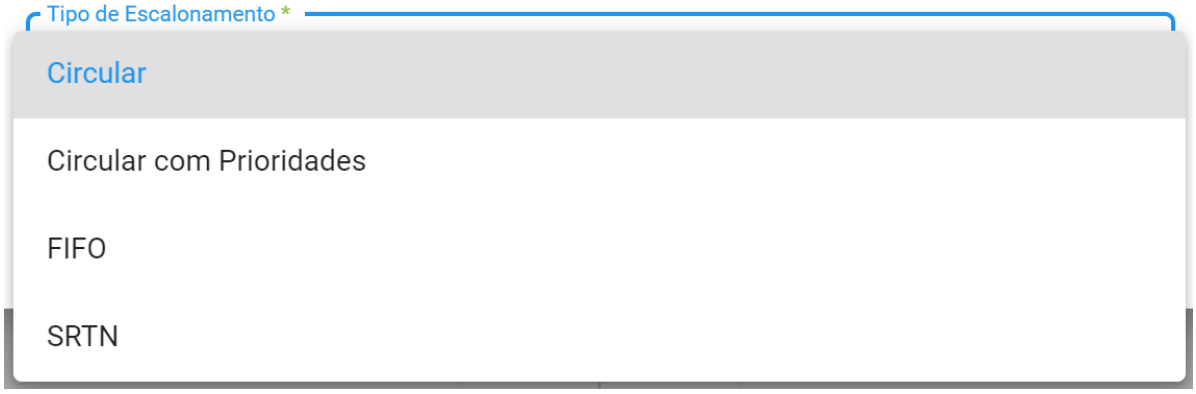


Figura 15 – Campo de seleção aonde contém todos os tipos de escalonamento disponíveis.

4.3 Processo

O processo, elemento principal da nossa aplicação, é uma entidade que representa uma tarefa em execução neste "sistema operacional" fictício. Pode-se criar, editar, suspender e apagar um processo. Possui diversos tipos (subseção 2.3.1.1) e estados (subseção 2.3.1), tendo comportamentos diferentes a partir do tipo de escalonamento e outras configurações do processador.

4.3.0.1 Criação

Para criar um processo, é necessário clicar no botão `Criar processo`, abrindo uma janela com um formulário (Figura 16). Este formulário mudará de acordo com o tipo de escalonamento selecionado. Lembrando que este botão será bloqueado caso a quantidade de processos não finalizados ultrapasse o valor permitido e só será liberado caso algum processo ativo seja suspendido, excluído ou finalizado.

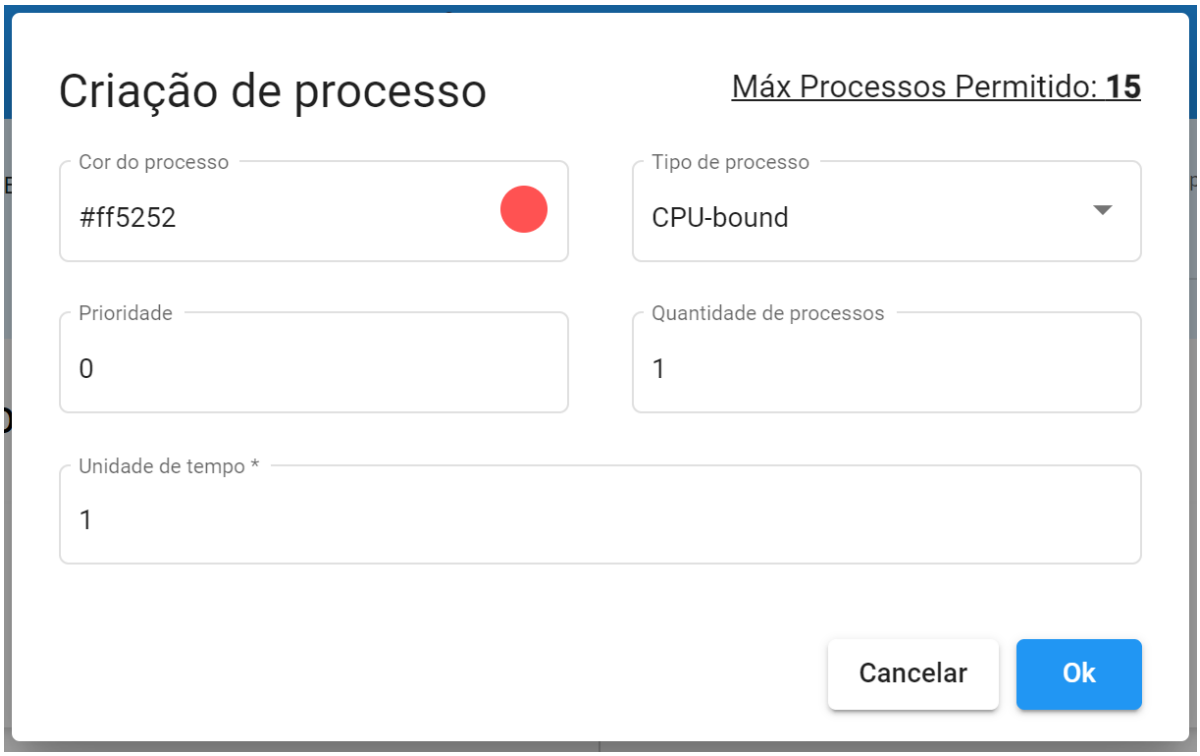


Figura 16 – Janela para a criação de um processo.

Nessa janela para a criação de um processo (Figura 16), alguns campos são necessários para a criação, sendo:

- `Cor do processo` (Figura 17): campo para selecionar uma cor para o processo, para facilitar a visualização dos processos na listagem (ideal quando será criado apenas um processo, caso seja criado mais de um, serão cores aleatórias);

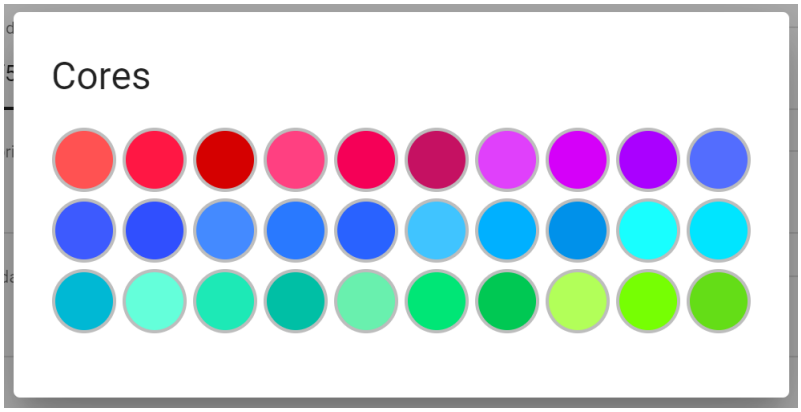


Figura 17 – Janela listando as possíveis cores para o processo.

- `Tipo de processo` (Figura 18): campo aonde é selecionado o tipo do processo (subseção 2.3.1.1);

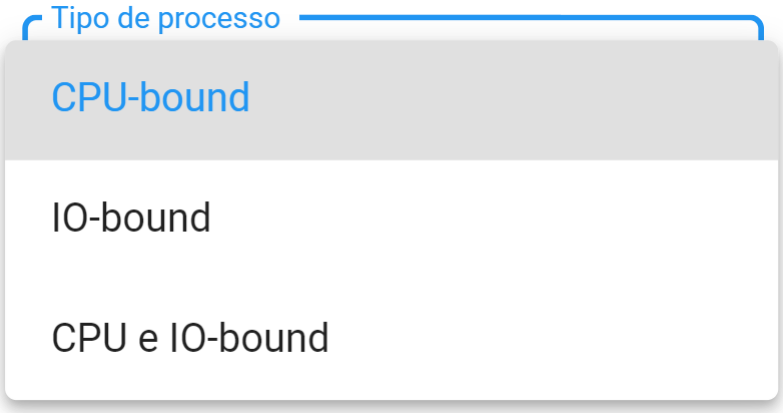


Figura 18 – Campo de seleção para o tipo do processo (expandido).

- `Prioridade`: campo numérico para inserir o nível da prioridade do processo (este campo aparecerá apenas quando o tipo de escalonamento selecionado for "Circular com Prioridades"). Por padrão, já vem com valor 0;
- `Quantidade de processos`: campo numérico aonde é inserido a quantidade de processos que deseja criar de uma vez (respeitando a quantidade máxima permitida, informado no canto superior direito da janela). Por padrão, já vem com valor 1;
- `Unidade de tempo`: campo numérico aonde é inserido a quantidade de unidade de tempos necessário para o processo finalizar. Por padrão, já vem com valor 1.

Preenchendo os campos explicados anteriormente e clicando no botão `Ok`, a janela será fechada e os processos serão criados.

4.3.0.2 Listagem

É possível visualizar todos os processos criados, permitindo o estudante monitorar e interagir com estes. Ações como suspender, retomar, excluir e editar processos são possíveis e, são instrumentos essenciais para simular cenários de teste e analisar o comportamento dos processos no fluxo de escalonamento.

Há 3 tipos de listagens presentes na tela (Figura 19):

- Processos ativos (demarcada em vermelho): listagem dos processos que se encontram no estado "Pronto";
- Processos I/O (demarcada em azul): listagem dos processos do tipo "I/O-bound" ou "CPU-bound e I/O-bound" que se encontram no estado "Pronto I/O";
- Processos finalizados ou suspensos (demarcada em verde): listagem dos processos que foram finalizados (por tempo ou exclusão) ou suspensos.

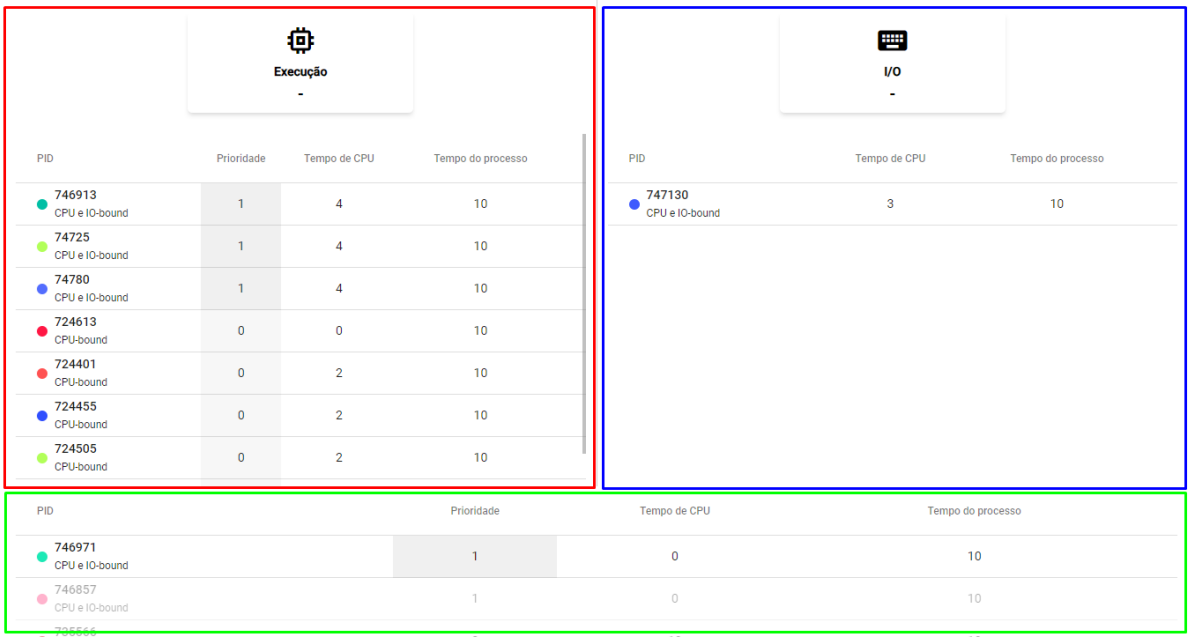


Figura 19 – Todas as listagens da aplicação demarcadas em cores diferentes.

A listagem dos processos ativos (Figura 20) contêm um campo informando qual processo está em execução no momento, seguido de uma tabela com os processos que estão aguardando a serem executados pelo processador. Nesta listagem, pode conter processos dos 3 tipos (subseção 2.3.1.1).

A listagem de processos I/O (Figura 21) contêm um campo informando qual processo do tipo "I/O-bound" está aguardando a ação, seguido de uma tabela com os processos que estão na fila para aguardar uma ação.

Figura 20 – Listagem de todos os processos que estão aguardando execução em formato de tabela.

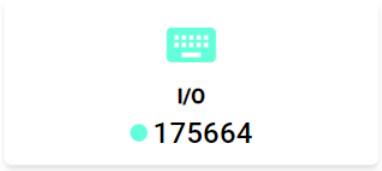
A listagem de processos finalizados ou suspensos (Figura 22) contêm uma tabela com todos os processos que foram suspensos ou finalizados. É possível diferenciar um processo do outro pois o processo finalizado estará com um efeito "borrado". Já o processo suspenso não estará com o efeito, além de ser possível clicar nele e reativá-lo.

4.3.0.3 Edição

A partir da listagem dos processos, é possível editar um processo (Figura 23). Isso varia de acordo com o seu estado e também o tipo de escalonamento selecionado no momento. É possível editar processos que estão com estado "Pronto", "Pronto I/O" e "Suspenso". Além disso, dos tipos de escalonamentos citados (seção 2.4), apenas o "Circular com prioridades" possui um campo editável, que é a própria prioridade do processo. Outros campos não podem ser editados, pois interferem diretamente na execução do processo e no fluxo de escalonamento.

É possível suspender e retomar o processo, porém apenas em processos com estado "Pronto" e "Pronto I/O". Basta clicar no ícone presente dentro da janela de edição, conforme Figura 24 e Figura 25.

Caso queira finalizar o processo, ainda dentro da janela de edição, basta clicar no ícone no canto superior direito (lixeira), ao lado do ícone de suspender/retomar processo (pause). Ao clicar, a janela automaticamente fechará e o processo será movido para a



PID	Tempo de CPU	Tempo do processo
1756116 IO-bound	1	10
1756177 IO-bound	1	10
1756239 IO-bound	1	10
1756301 IO-bound	1	10
1756365 IO-bound	1	10
1755851 IO-bound	7	10
1755902 IO-bound	7	10

Figura 21 – Listagem de todos os processos I/O que estão aguardando uma Entrada/Saída em formato de tabela.

PID	Prioridade	Tempo de CPU	Tempo do processo
1044900 CPU-bound	0	14	20
1050475 IO-bound	0	12	20
3933719 CPU-bound	1	10	10
3933775 CPU-bound	1	10	10
3933832 CPU-bound	1	10	10
3925719 CPU-bound	0	10	10
3925775 CPU-bound	0	10	10
717738 CPU e IO-bound	0	0	20
717789 CPU e IO-bound	0	0	20

Figura 22 – Listagem de todos os processos suspensos e finalizados em formato de tabela.

listagem de processos suspensos/finalizados.

4.3.0.4 Exclusão

É possível excluir apenas um processo ou todos de uma só vez. Caso queira excluir um processo, clicando nele, abrirá a janela de edição e, ao clicar no ícone no canto superior direito (lixeira), este será excluído. No caso da exclusão de todos os processos, basta clicar no botão `Parar` na aba de configurações do Processador. Isto fará com que o sistema seja

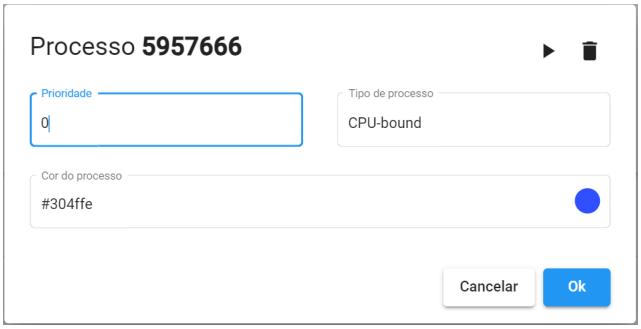


Figura 23 – Janela de edição contendo o campo para editar a prioridade.

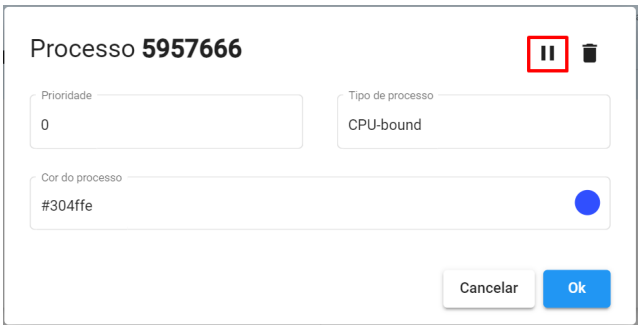


Figura 24 – Janela de edição contendo o botão para suspender o processo.

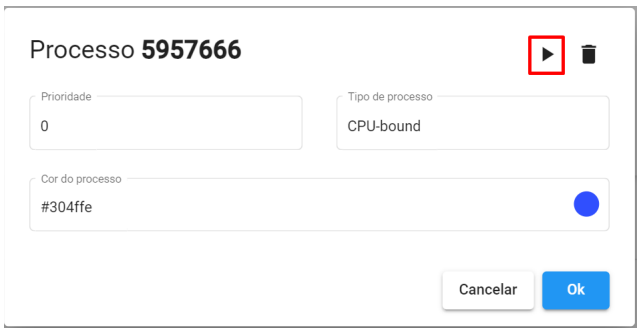


Figura 25 – Janela de edição contendo o botão para retomar o processo.

reiniciado, apagando todos os processos em tela.

4.3.0.5 Gráfico

A aplicação permite gerar um gráfico do fluxo de execução dos processos do tipo "CPU-bound" que já finalizaram. Isso permite ao estudante analisar o ciclo de tempo de vida do processo, tendo métricas fundamentais para avaliar a eficiência do escalonamento e a rapidez com que os processos foram atendidos, dependendo do tipo do escalonamento escolhido. Para isso, basta clicar no botão `Gerar gráfico de tempo de vida` (Figura 27). Lembrando que este botão só aparecerá caso contenha algum processo finalizado.

Ao clicar no botão, abrirá uma janela contendo uma tabela listando os processos finalizados, permitindo ao estudante selecionar quais processos deseja analisar o tempo de vida. O estudante pode selecionar manualmente ou, caso queira analisar todos os possíveis

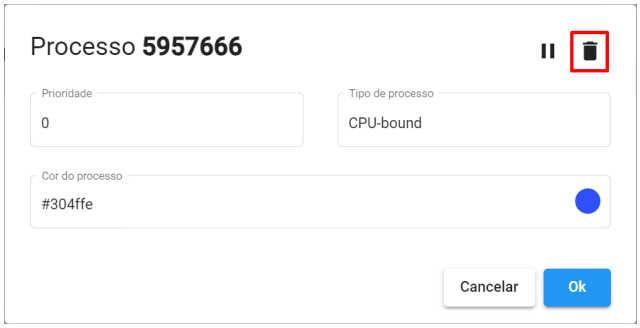


Figura 26 – Janela de edição contendo o botão para finalizar o processo.

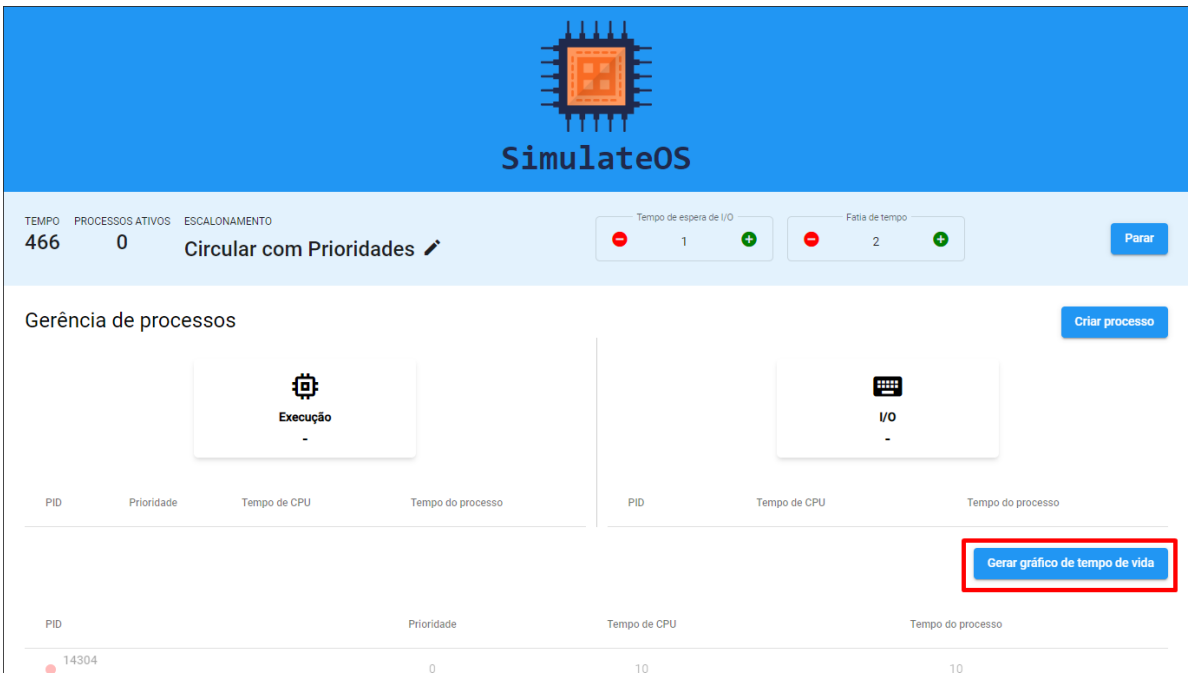


Figura 27 – Botão para abrir janela, permitindo gerar o gráfico.

processos, há um campo de seleção no cabeçalho da tabela que, ao selecioná-lo, todos os processos serão selecionados (Figura 28).

Selecionando os processos, um botão `Gerar gráfico` abaixo da tabela será liberado e, clicando-o, aparecerá logo abaixo dele, um gráfico contendo os processos, exibindo: PID, cor e o tempo que foi executado (Figura 29).

Como há muito conteúdo para uma janela pequena, há um botão no canto superior direito da janela para expandir, preenchendo a tela inteira, facilitando a visualização do gráfico. Além disso, o gráfico contém botões nos quais permitem aumentar e diminuir a visualização dos processos, além de exportar o conteúdo do gráfico em arquivos *.svg* e *.png* (Figura 30).

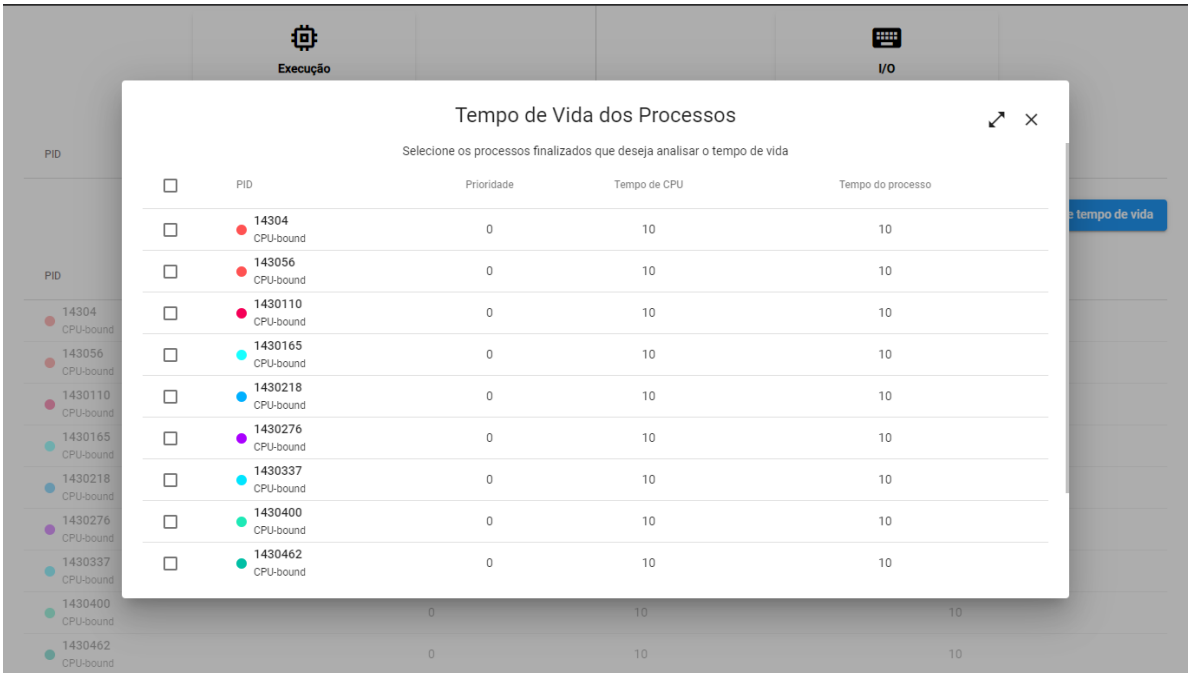


Figura 28 – Lista de seleção dos processos finalizados para gerar o gráfico.

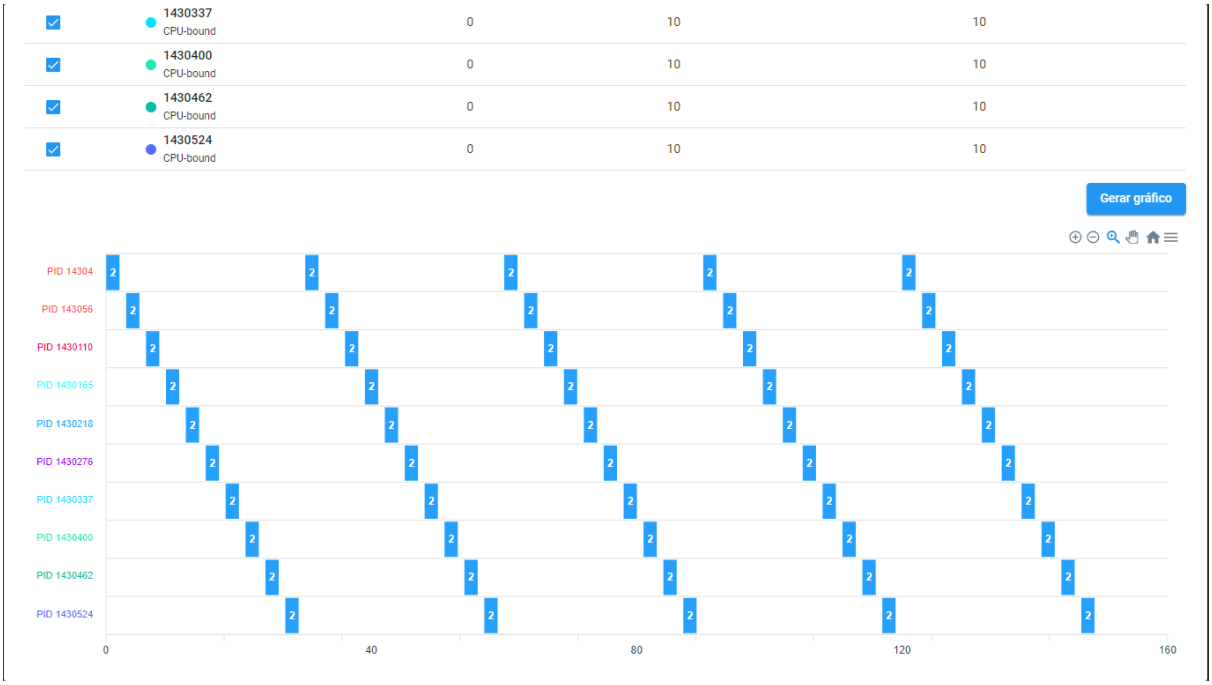


Figura 29 – Gráfico gerado a partir dos processos selecionados.

4.3.0.6 Escalonamento

Para cada tipo de processo - **CPU-bound**, **I/O-bound** e **CPU e I/O-bound** (subseção 2.3.1.1) - o fluxo de escalonamento é gerenciado de maneira distinta, refletindo as características específicas de cada tipo:

- Processo CPU-bound:

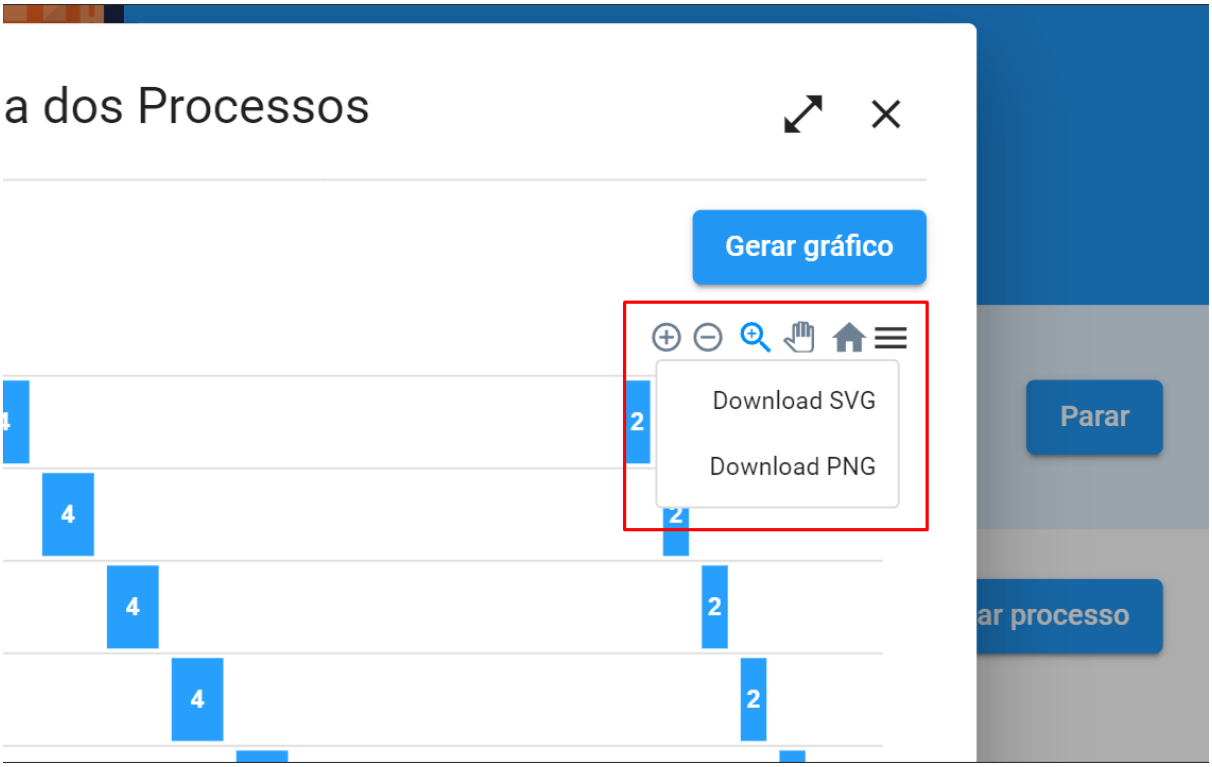


Figura 30 – Botões de ação do gráfico demarcados em vermelho.

  - Quando um processo do tipo **CPU-bound** é criado, ele é colocado na fila de execução, no estado **Pronto**;
  - Após chegar sua vez no escalonamento, ele avança para o estado **Em execução**, onde permanecerá até que a `Fatia de Tempo` definida nas configurações do processador tenha terminado ou até que a quantidade de unidade de tempo executada atinja o seu limite;
  - É importante ressaltar que este processo pode ser **Suspenso** ou **Finalizado** manualmente a qualquer momento, conforme necessário.

- Processo I/O-bound:
  - Ao ser criado, um processo do tipo **I/O-bound** também entra na fila de execução, no estado **Pronto**;
  - Quando chega sua vez, o processo é movido para o estado **Em execução** e permanece lá por um período de tempo pré-definido, que é **1 (uma) unidade de tempo**;
  - Se o processo não for concluído durante esse período, ele é movido para o estado **Pronto I/O** e entra em uma fila de espera específica para processos **I/O-bound**;
  - Após aguardar na fila, o processo é movido para o estado **Em I/O** por um tempo determinado pela configuração `Tempo de Espera I/O` do processador;

  - Se o processo não for concluído após esse período, ele retorna ao estado **Pronto** e segue o fluxo de escalonamento novamente;
  - Assim como nos outros tipos, o processo **I/O-bound** também pode ser **Suspenso** ou **Finalizado** manualmente durante sua execução.

- Processo CPU e I/O-bound:
  - O processo **CPU e I/O-bound** segue uma lógica semelhante aos outros tipos, com a diferença de que, a cada execução, é sorteado se ele atuará como **CPU-bound** ou **I/O-bound**;
  - Portanto, o processo pode alternar entre os estados **Em execução** e **Pronto I/O** conforme determinado pelo sorteio;
  - Da mesma forma que nos demais tipos, o processo **CPU e I/O-bound** também pode ser **Suspenso** ou **Finalizado** manualmente durante sua execução.

4.4 Modificações

Para analisar melhor os resultados, faz-se necessário analisar as alterações e adições referente ao projeto anterior (FREITAS, 2023). Portanto, nesta seção serão listados e explicados os pontos e funcionalidades que foram modificadas.

- **Alterado: Os campos das configurações do processador.** Anteriormente os campos eram em formato de controle deslizante (Figura 31). Este formato, em questões de experiência do usuário, não é muito efetivo. Por conta disso, foi alterado para um campo convencional, facilitando a visualização do valor da configuração, além de facilitar a alteração deste, por meio de botões de ações de incremento e decremento (Figura 32).

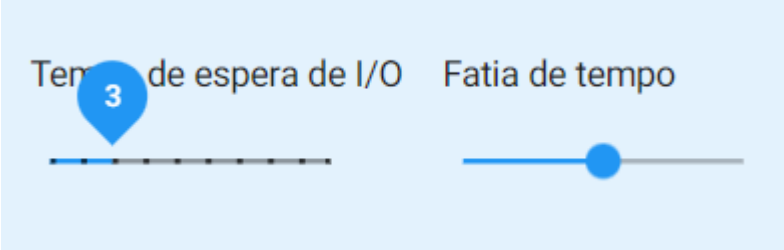


Figura 31 – Campos para alterar os valores das configurações.

Fonte: Extraído de (FREITAS, 2023)

- **Remoção da configuração "clock da CPU".** Esta configuração foi removida por questões de desempenho e por acarretar conflitos, dependendo do tipo de escalonamento selecionado (Figura 33). Futuramente, pretende-se implementar tal

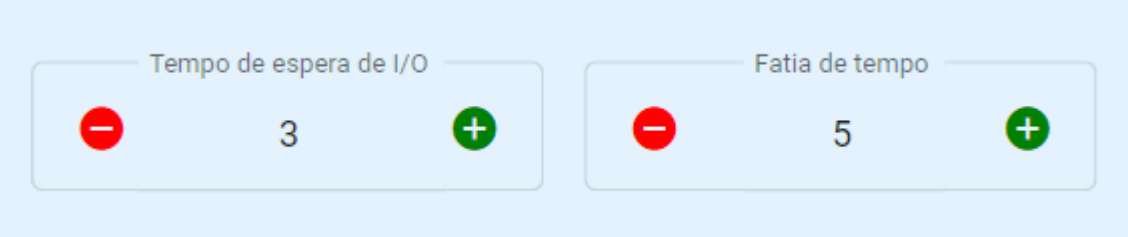


Figura 32 – Novos campos para alterar os valores das configurações.

funcionalidade para que se trabalhe com simulações mais longas em menor tempo de visualização.

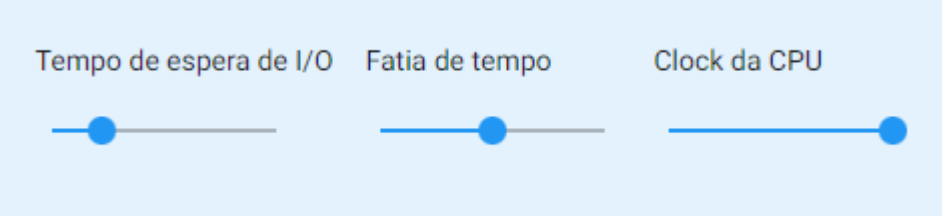


Figura 33 – Configurações do processador contendo o "Clock da CPU".

Fonte: Extraído de (FREITAS, 2023)

- **Redimensionamento da aplicação inteira.** Esta alteração foi feita pois, anteriormente, a aplicação possuía uma largura pre-definida (Figura 34), visando funcionar apenas em dispositivos com menor resolução (tablet, notebooks). Por consequência, ocupava bastante espaço necessário para as novas funcionalidades adicionadas. Logo, foi removido esta largura pre-definida (Figura 35).

- **Adicionado: Novos tipos de escalonamento**. A aplicação anterior possuía apenas o tipo de escalonamento "Circular com Prioridades". Com a adição de novos tipos, o estudante pode experienciar diferentes cenários. Por conta disso, foi adicionado uma nova configuração no "Processador" (Figura 36) e criado uma janela para alterar o tipo de escalonamento atual (Figura 37).

- **Adicionado: Campo "Unidade de Tempo" e "Limite máximo de processos".** Agora é possível, no fluxo de criação de processos, definir em quantas unidades de tempo um processo será finalizado. Este campo é de grande importância para analisar o fluxo de execução dos processos. Além disso, foi adicionado um campo informativo no canto superior direito da janela de criação de processo, informando quantos processos pode-se criar naquele momento (Figura 38). Este campo foi adicionado, pois anteriormente, não era intuitivo para o estudante identificar a quantidade de processos que poderia criar naquele momento.

- **Adicionado: Funcionalidade de gerar gráfico.** Com a adição de novos tipos de escalonamento, para facilitar a visualização e análise do comportamento de cada

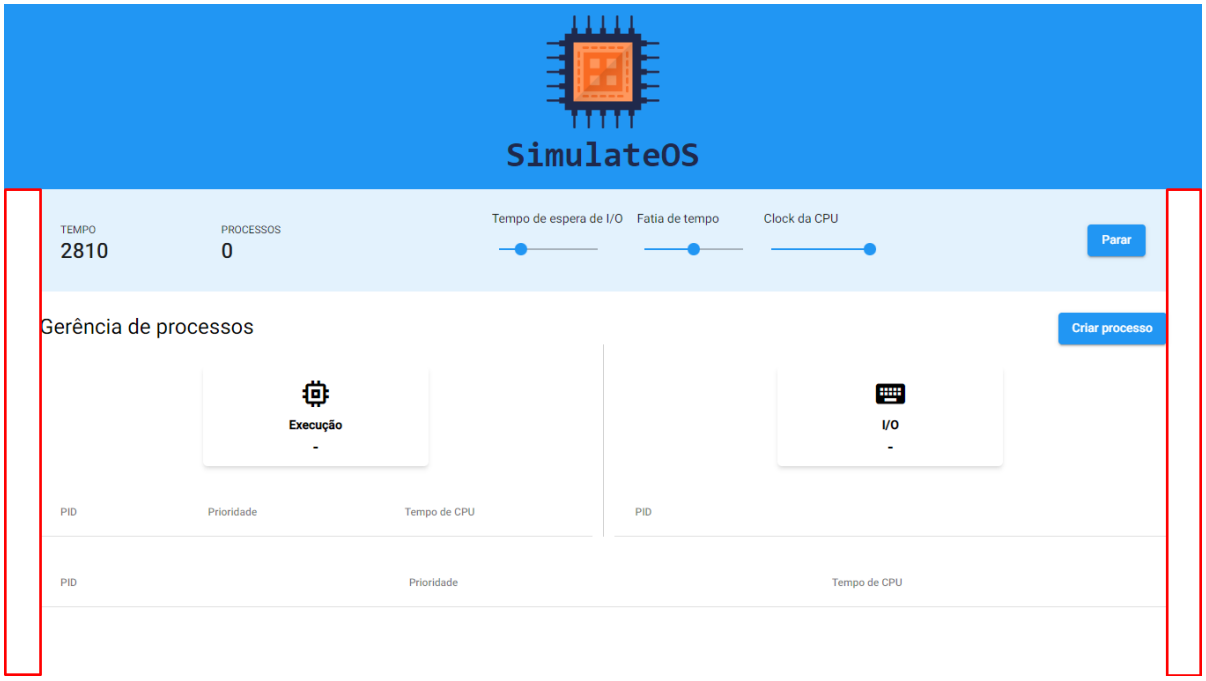


Figura 34 – Antiga interface da aplicação com espaço removido demarcado em vermelho.

Fonte: Extraído de (FREITAS, 2023)

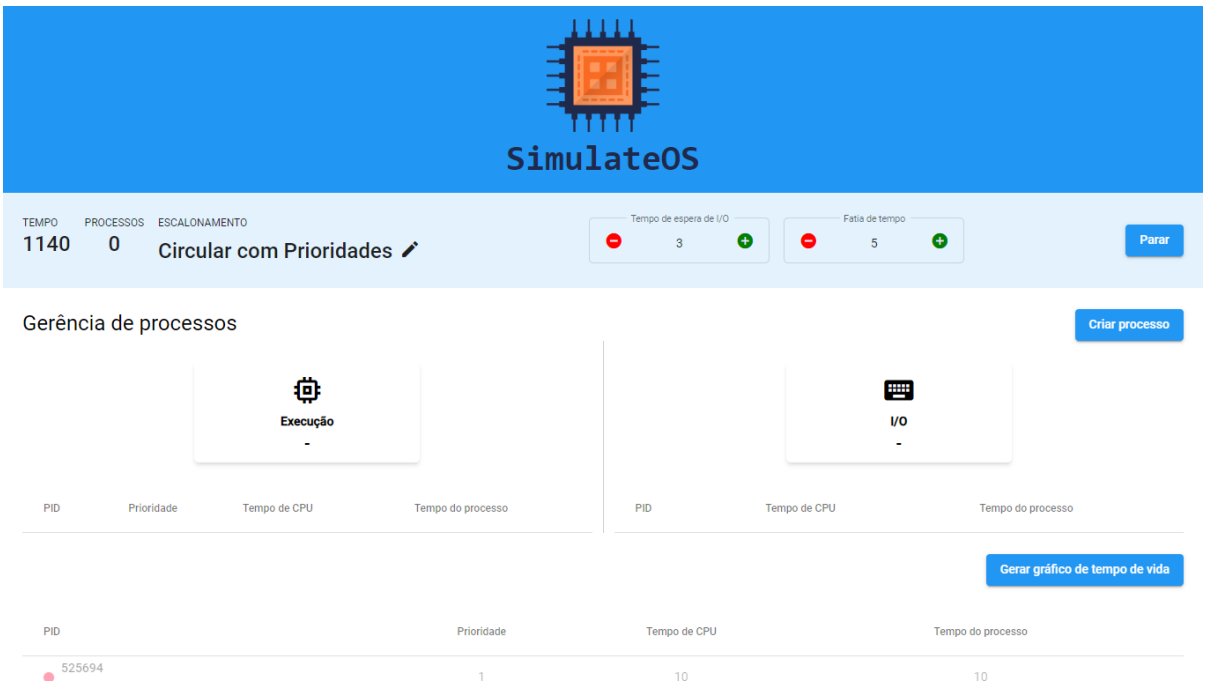


Figura 35 – Interface da aplicação após reestruturação.

processo em determinado momento, foi adicionado uma funcionalidade para gerar um gráfico contendo o fluxo de execução dos processos. Com isso, também foi criada uma janela que lista os processos que já finalizaram, permitindo que o estudante possa selecionar quais processos deseja visualizar e, por fim, gerar o gráfico (Figura 39).

Figura 36 – Enfase nos novos campos das configurações do processador.

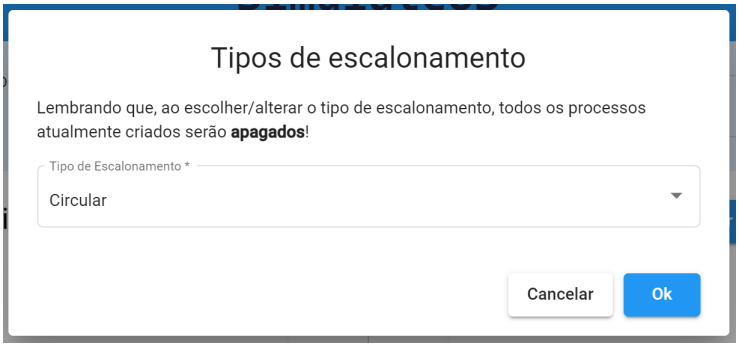


Figura 37 – Janela que permite alterar o tipo de escalonamento do processador.

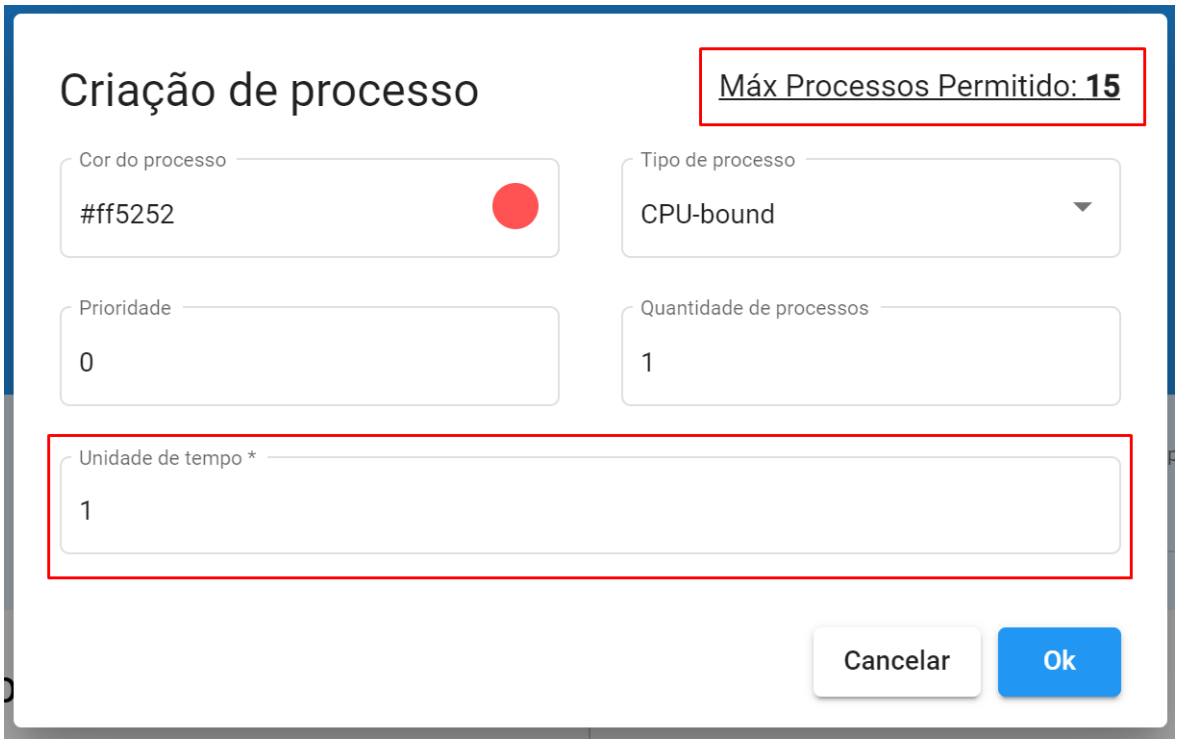


Figura 38 – Janela de criação de um processo com os novos campos destacados em vermelho.

- **Adicionado: "Processos Finalizados" na listagem.** Anteriormente, não era importante mostrar ou usufruir dos processos que já finalizaram. Porém, com a adição da funcionalidade de gerar um gráfico, foi adicionado, juntamente com os processos suspensos, os processos finalizados. Para conseguir diferenciá-los, optou-se por criar uma diferença na estilização (Figura 40).

- **Alterado: Comportamento da edição de um processo.** Devido a adição de novos tipos de escalonamento, caso algum campo seja alterado durante o fluxo de

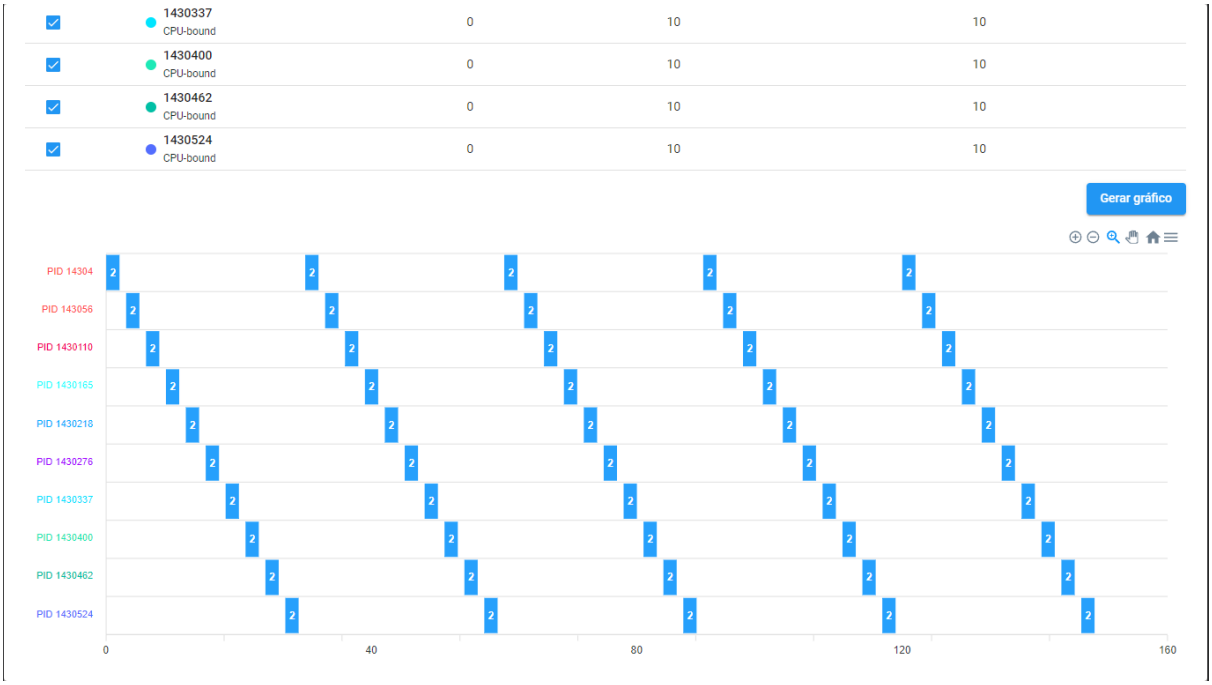


Figura 39 – Janela contendo um gráfico gerado a partir dos processos selecionados.

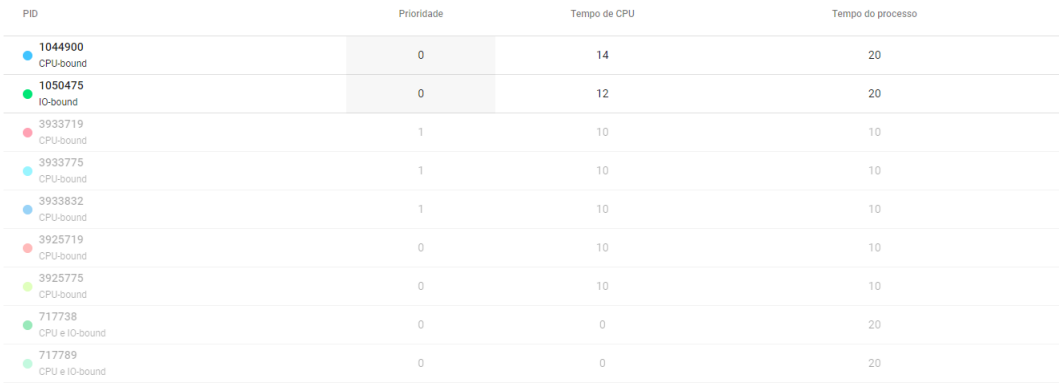


Figura 40 – Listagem de todos os processos suspensos e finalizados.

execução, poderia acarretar conflitos e/ou divergências na geração do gráfico. Por conta disso, apenas o campo “Prioridade” (em casos do escalonamento atual ser “Circular com Prioridades”) pode ser editado (Figura 41).

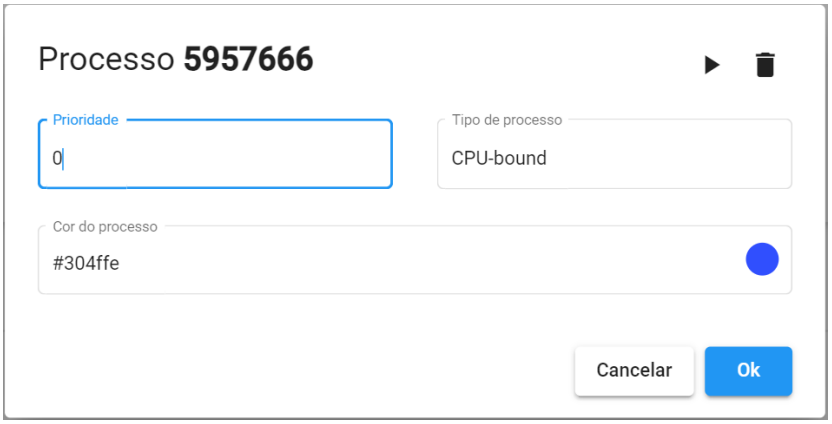


Figura 41 – Janela de edição contendo o campo para editar a prioridade.

5 Conclusão

Os conceitos de Sistemas Operacionais são de extrema importância para estudantes de cursos de tecnologia. Este trabalho teve como objetivo aprimorar um simulador de gerenciamento de processos com o desenvolvimento já iniciado por Freitas (2023).

A aplicação permitia o estudante criar experiências em um ambiente simulado, facilitando o entendimento de como os processos funcionam em um sistema operacional. Com o desenvolvimento deste trabalho, pôde-se melhorar diversas funcionalidades já criadas, além de adicionar novas, tudo com o real propósito de trazer mais clareza ao usuário (estudante).

A adição de novos tipos de escalonamento, possibilitando o usuário a criar diferentes cenários, além da possibilidade de gerar gráficos envolvendo o fluxo de execução de cada processo, foram de extrema importância para entender melhor como os processos são escalonados.

Comparando-o com outros trabalhos semelhantes a esse, são visíveis as inúmeras possibilidades e novas funcionalidades que podem ser adicionadas a esta aplicação, mas que infelizmente não foram possíveis serem implementadas neste trabalho e serão listados em **Trabalhos Futuros**.

5.1 Principais Contribuições

O presente trabalho representa uma contribuição significativa para a área de educação em tecnologia, particularmente no ensino de Sistemas Operacionais. Ao observar as dificuldades que os estudantes possuem ao absorver os conceitos teóricos desta disciplina, foi identificado a oportunidade de aprimorar ainda mais a aplicação.

Uma das principais contribuições deste trabalho é o desenvolvimento e aperfeiçoamento de um simulador de gerenciamento de processos em sistemas operacionais. Este simulador não apenas replica os conceitos e funcionalidades relacionados ao gerenciamento de processos, mas também oferece uma experiência prática e imersiva aos usuários, permitindo-lhes explorar e compreender os diferentes métodos de escalonamento de processos de forma interativa.

Além disso, a validação do simulador por meio de testes de simulação de cenários proporcionou percepções valiosas para identificar possíveis melhorias e aprimoramentos no simulador, garantindo sua qualidade e eficácia na promoção da aprendizagem dos estudantes.

5.2 Trabalhos Futuros

Por questões de tempo e por infraestrutura da aplicação, infelizmente alguns pontos não foram possíveis de serem desenvolvidos. São eles:

- Adicionar novos tipos de escalonamentos à aplicação, como por exemplo múltiplas filas, escalonamento por loteria e escalonamento garantido;
- Remodelar a infraestrutura do código, a ponto de ser possível retornar a configuração "Clock da CPU" sem acarretar conflitos;
- Atualizar a versão da aplicação *Angular*, para assim usufruir de bibliotecas de terceiros e obter mais funcionalidades, como foi o exemplo do *.csv* do gráfico, que foi removido propositalmente devido a versão da biblioteca que continha um erro e, apenas uma atualização completa da aplicação possibilitaria resolver;
- Aperfeiçoar a responsividade da aplicação para melhorar a experiência do usuário em um smartphone.

5.3 Contribuições em Produção Bibliográfica

Durante o desenvolvimento do trabalho, os resultados parciais foram publicados no evento XVII Workshop de Teses e Dissertações em Ciência da Computação (WTDCC) 2023[1], evento acadêmico na Universidade Federal de Uberlândia (UFU). Além do artigo criado, foi também apresentada a ideia e a aplicação por meio de textos e imagens em um "banner" para outros estudantes.

Pretende-se, em breve, submeter o trabalho completo para evento nacional na área.

[1] https://techweek.facom.ufu.br/WTDCC2023

Referências

BIEREMA, N. **Redux DevTools - uma extensão de navegador para depurar alterações de estado do aplicativo**. ReduxJS: [s.n.], 2015. Chrome Web Store. Disponível em: <https://chromewebstore.google.com/detail/redux-devtools/lmhkpmbekcpmknklioeibfkpmmfibljd?pli=1>. Citado na página 30.

FREITAS, G. M. B. de. **Simulador de gerência de processos para sistemas operacionais**. 39 p. Monografia (Graduação em Sistemas de Informação) — Universidade Federal de Uberlândia, Uberlândia, 2023. Disponível em: <https://repositorio.ufu.br/handle/123456789/37373>. Citado 8 vezes nas páginas 12, 13, 22, 32, 44, 45, 46 e 50.

KIOKI, E. Y. **Um Simulador Didático como ferramenta de apoio ao ensino da disciplina de Sistemas Operacionais**. 8 p. Monografia (Pesquisa Científica) — Fundação de Amparo à Pesquisa do Estado de Minas Gerais, Horto Florestal, 2008. Disponível em: <https://fai-mg.br/portal/download/revista_inicia_2008/pub_dw_artigo_simulador.pdf>. Citado 2 vezes nas páginas 23 e 25.

MAIA, L. P. **SOsim: Simulador para o Ensino de Sistemas Operacionais**. NCE/UFRJ: [s.n.], 2001. Auto-publicado. Disponível em: <http://www.training.com.br/sosim/>. Citado 2 vezes nas páginas 22 e 23.

OLIVEIRA, R. A. **SWSO - Simulador Web de Sistemas Operacionais**. 18 p. Monografia (Graduação em Análise e Desenvolvimento de Sistemas) — Instituto Federal da Bahia, Salvador, 2015. Disponível em: <https://labrasoft.ifba.edu.br/publicacoes/tcc/swso-simulador-web-de-sistemas.pdf>. Citado 3 vezes nas páginas 14, 23 e 24.

REIS, F. P. **TBC-SO/WEB: Software Educativo para Aprendizagem de Gerência de Processos e de Gerência de Memória em Sistemas Operacionais**. UFLA: [s.n.], 2009. Auto-publicado. Disponível em: <http://algol.dcc.ufla.br/~heitor/Projetos/TBC_SO_WEB/tbc_so_web.html>. Citado 3 vezes nas páginas 14, 24 e 25.

SCAMATI, V. **Um Simulador para o Ensino de Sistemas Operacionais com a Tecnologia da Realidade Virtual (ESORV)**. 115 p. Monografia (Mestrado em Ciência da Computação) — Centro Universitário Campo Limpo Paulista, Campo Limpo Paulista, 2017. Disponível em: <https://www.cc.faccamp.br/Dissertacoes/VagnerScamati.pdf>. Citado 2 vezes nas páginas 23 e 24.

STALLINGS, W. **Operating Systems: Internals and Design Principles**. Pearson: [s.n.], 2014. Citado na página 16.

TANENBAUM, A. **Sistemas Operacionais Modernos**. 4. ed. Pearson: [s.n.], 2010. Citado 3 vezes nas páginas 15, 17 e 19.