

1. INTRODUÇÃO

Neste documento, são abordados conceitos fundamentais relacionados a sistemas operacionais, com foco em quatro temas inter relacionados: o conceito básico de processo, o pseudo paralelismo, as situações de corrida e o uso de semáforos para sincronização. Tais temas são essenciais para a compreensão do funcionamento de sistemas multitarefa e do controle de acesso a recursos compartilhados, garantindo a integridade dos dados, a ilusão de simultaneidade e a estabilidade do sistema. Além disso, serão apresentados exemplos práticos e comparações que ampliam a compreensão dos mecanismos envolvidos.

2. DESENVOLVIMENTO

2.1. Básico de Processo

Em sistemas operacionais, um processo pode ser definido como uma instância de um programa em execução. Enquanto o programa consiste em um conjunto de instruções armazenadas em disco, o processo é uma entidade ativa que reúne o código do programa, seu estado de execução atual e os recursos necessários para sua operação. A gerência de processos é uma função essencial do sistema operacional, envolvendo a criação, escalonamento, controle de concorrência, comunicação entre processos e manipulação dos estados de execução.

O escalonador de processos (scheduler) é o componente responsável por selecionar, entre os processos prontos, aquele que será executado, utilizando algoritmos que consideram critérios como tempo de resposta, throughput e uso eficiente dos recursos. O conceito de processo é, portanto, fundamental para viabilizar a execução simultânea de múltiplos programas (multitarefa) e para que o sistema responda de maneira organizada a eventos externos.

Como exemplo, considere um computador onde o usuário está escutando música, navegando na internet e executando uma planilha ao mesmo tempo. Cada uma dessas atividades está sendo executada como um processo distinto, gerenciado de forma eficiente pelo sistema operacional.

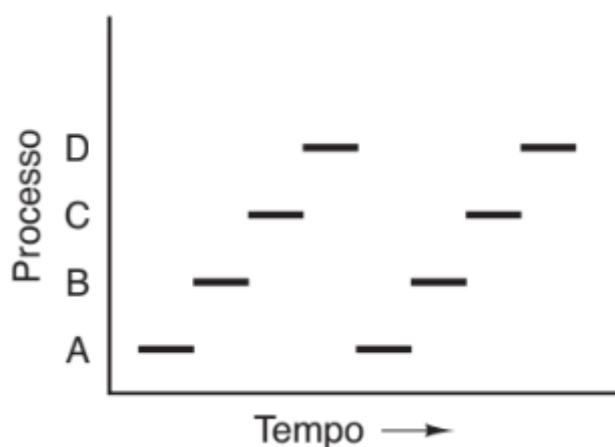
2.2. Pseudo Paralelismo

O pseudo paralelismo é um conceito fundamental em sistemas multitarefa, especialmente em sistemas com um único núcleo de processamento. Ele refere-se à capacidade do sistema operacional de alternar rapidamente entre diversos processos ou threads, criando a ilusão de que estão sendo executados simultaneamente. Na realidade, apenas um processo está ativo por vez, mas, devido à alta velocidade de alternância, os usuários percebem uma execução paralela.

Esse mecanismo é viabilizado pelo escalonador, que define a ordem e o tempo de execução de cada processo, garantindo uma distribuição justa e eficiente da CPU. O pseudo paralelismo permite melhor aproveitamento do processador e maior responsividade do sistema, sendo uma técnica essencial em ambientes de tempo compartilhado.

Diferentemente do paralelismo real, onde múltiplos núcleos permitem a execução simultânea de processos, o pseudo paralelismo depende inteiramente da alternância eficiente. Mesmo assim, ele é suficiente para garantir uma experiência fluida para o usuário em muitas aplicações do dia a dia.

Figura 1 - Demonstração de processos em pseudoparalelismo. Apenas um programa está ativo de cada vez.



2.3. Situações de Corrida

Em ambientes operacionais, as situações de corrida surgem quando dois ou mais processos ou threads acessam e modificam dados compartilhados sem a devida coordenação. Nesse cenário, o resultado final das operações pode variar conforme a ordem em que as instruções são executadas, levando a comportamentos imprevisíveis e inconsistentes. Tal fenômeno pode comprometer a integridade dos dados e afetar o funcionamento correto do sistema.

Um exemplo prático é o acesso simultâneo a uma conta bancária por dois caixas eletrônicos. Se ambos tentarem debitar valores ao mesmo tempo sem sincronização adequada, o saldo poderá ser calculado incorretamente, resultando em erros graves.

A identificação e prevenção de condições de corrida são, portanto, essenciais para o desenvolvimento de aplicações robustas e seguras.

2.4. Semáforo

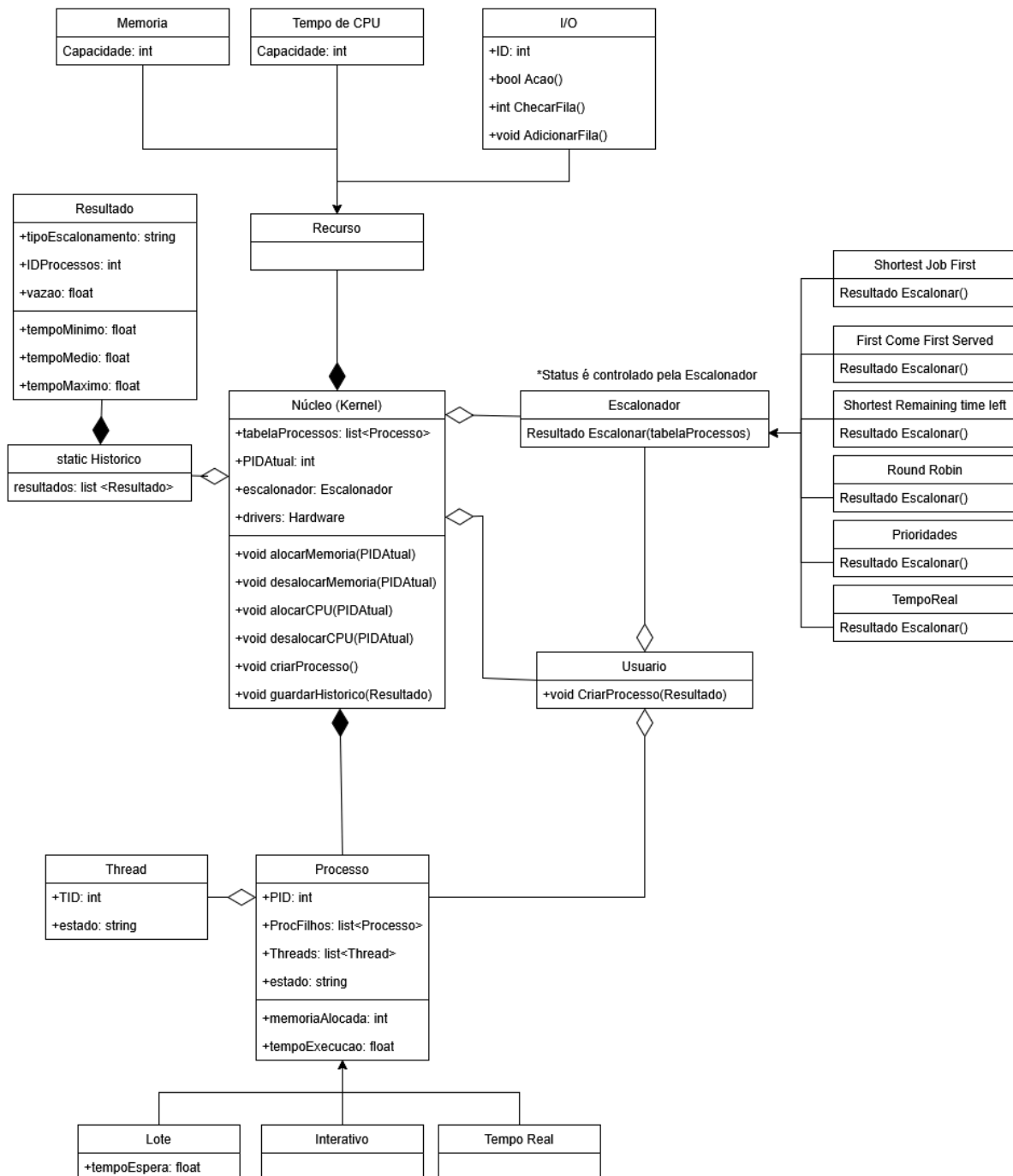
Para mitigar os problemas decorrentes de acessos não controlados, os semáforos são empregados como mecanismos de sincronização. Em sistemas operacionais, um semáforo consiste em uma variável inteira associada a um conjunto de operações atômicas que garantem sua manipulação segura. As principais operações são:

- **Wait** (P ou *down*): Decrementa o valor do semáforo. Se o resultado for negativo, o processo ou *thread* que realizou a operação é bloqueado até que outro processo incremente o semáforo.
- **Signal** (V ou *up*): Incrementa o valor do semáforo e, caso haja processos ou *threads* bloqueados, libera um deles para que prossiga a execução.

Por meio dessas operações, os semáforos asseguram que o acesso a regiões críticas seja feito de forma exclusiva, evitando que múltiplas entidades acessem e

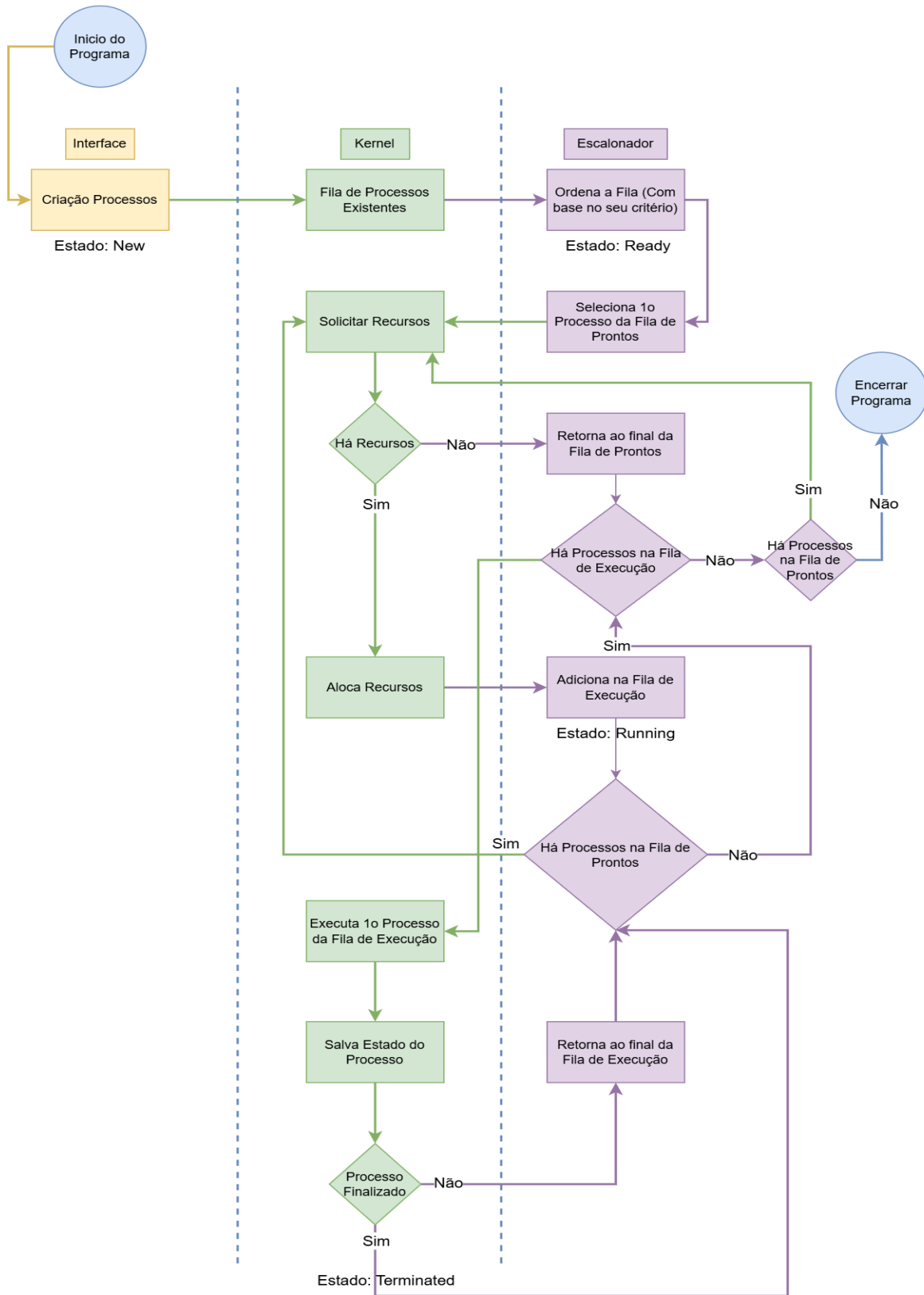
modifiquem simultaneamente os mesmos dados. Assim, os semáforos são ferramentas essenciais para a coordenação e o gerenciamento de processos, promovendo a distribuição ordenada e segura dos recursos do sistema.

Figura 2 - UML como representação lógica do SO em desenvolvimento.



Fonte: Autores, desenvolvido com a ferramenta [DRAW.IO](https://draw.io).

Figura 3 - Fluxograma representando a lógica de processo em desenvolvimento.



Fonte: Autores, desenvolvido com a ferramenta [DRAW.IO](https://draw.io).

3. CONCLUSÃO

A compreensão dos processos e pseudo paralelismo, das situações de corrida e dos semáforos é crucial para o desenvolvimento e a manutenção de sistemas operacionais modernos. Enquanto o conceito de processo possibilita a execução simultânea de programas, as condições de corrida evidenciam os riscos inerentes ao acesso descoordenado a recursos compartilhados. Nesse contexto, os semáforos surgem como um mecanismo eficaz para garantir a integridade dos dados e a estabilidade do sistema, permitindo a sincronização adequada entre processos e threads.

O pseudo paralelismo permite simular a execução simultânea em sistemas com núcleos únicos, oferecendo uma experiência fluida ao usuário. Em conjunto, esses fundamentos são indispensáveis para a construção de sistemas robustos e eficientes, onde a coordenação, a ilusão de simultaneidade e o controle dos recursos desempenham um papel determinante na confiabilidade das operações. A inclusão de exemplos e comparações neste relatório reforça a aplicabilidade prática e a relevância desses conceitos no desenvolvimento de software confiável.