1. Justificativa do Design

O projeto utiliza uma **Lista Encadeada Simples** para as filas de processos. Essa escolha foi feita para respeitar a "Regra Fundamental" do projeto, que exige a implementação de uma estrutura de dados do zero. Essa abordagem demonstra o entendimento de como as listas funcionam internamente, manipulando os ponteiros e referências de cada nó.

As operações de **adicionar no final** e **remover do início**, essenciais para o escalonador, são muito eficientes nesta estrutura, pois são realizadas em tempo constante, **O(1)**.

2. Análise da Lógica de Anti-Inanição

Ela serve para que processos de baixa prioridade não fiquem esquecidos. A cada 5 processos de alta prioridade, é obrigatório que a CPU execute um processo de média ou baixa prioridade. Após essa execução, o contador é zerado

Essa regra evita que o fluxo constante seja monopolizado por processos de alta prioridade, permitindo que todos sejam executados.

3. Análise do Bloqueio

Quando o escalonador seleciona um processo, ele verifica se esse processo precisa do recurso "DISCO". Se for verdadeiro, ele não é executado. Em vez disso, ele é:

- Retirado da lista de prioridade.
- Adicionado no final da lista de bloqueados.

Com isso, a CPU fica livre para escolher e executar outro processo, enquanto o processo bloqueado aguarda a sua vez.

No início de cada ciclo da CPU, o sistema verifica se há processos na lista de bloqueados. Se houver, o método desbloquear é chamado. Ele executa da seguinte forma:

- O processo mais antigo é removido.
- Esse processo é reinserido no final de sua fila de prioridades original.

Esse ciclo garante que o processo retorne para sua fila original, com sua prioridade intacta, e que todos os processos bloqueados tenham a chance de serem executados pela CPU.

4. Ponto Fraco e Melhoria

O principal ponto fraco é a ineficiência de buscar ou mover um processo que esteja no meio de uma das filas, já que a operação teria uma complexidade de tempo **O(n)**.

Uma melhoria teórica seria usar uma estrutura de dados como um **min-heap** para gerenciar as prioridades, que é mais eficiente para essa tarefa. No entanto, essa abordagem violaria a "Regra Fundamental" do projeto.