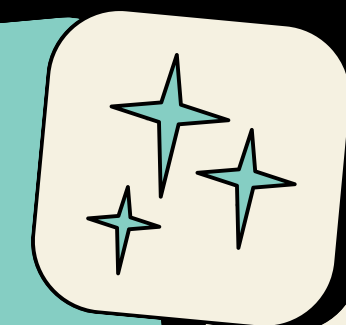


LOAD BALANCING

ALGORITHMS



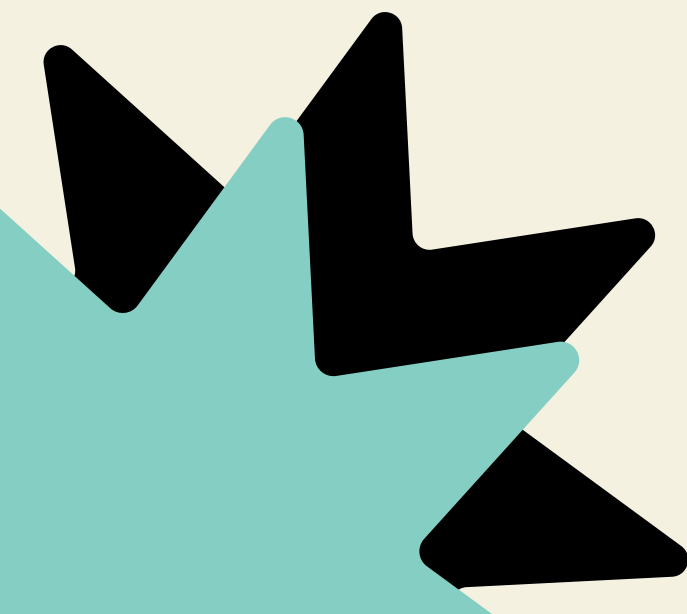
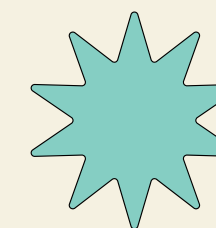
Agnes Bressan - 13677100

Carolina Elias - 13676687

Caroline Clapis - 13861923

Rauany Secci - 13721217

Rhayna Casado - 13676429



INTRODUÇÃO

- Os algoritmos de balanceamento de carga distribuem as requisições ou tarefas de acordo com critérios específicos, que variam conforme o tipo de algoritmo
- Evitar sobrecarga em qualquer recurso específico, melhorar o desempenho geral e garantir alta disponibilidade dos serviços.
- Podem ser implementados tanto no nível de hardware quanto no nível de software.



ETAPAS



1. Entrada de aquisição

Quando ocorre solicitação, essa requisição precisa ser encaminhada para um recurso que irá processá-la. O balanceador de carga atua como um intermediário, recebendo essa requisição e decidindo para onde enviá-la

2. Coleta de informações

O balanceador de carga coleta informações sobre os recursos disponíveis para tomar uma decisão

- Quantidade de conexões ativas em cada servidor
- Tempo de resposta recente dos servidores
- Capacidade dos servidores (CPU, memória)
- Localização do cliente em relação aos servidores

3. Escolha do recurso

Com base nos critérios do algoritmo, o balanceador escolhe o recurso mais adequado para processar a solicitação.

TIPOS

Diferentes formas de distribuir tarefas entre recursos disponíveis :

Balanceamento estático

- A alocação de tarefas é planejada previamente e não muda durante a execução.
- As decisões são tomadas com base em informações conhecidas antes da execução.

Exemplo: Round Robin

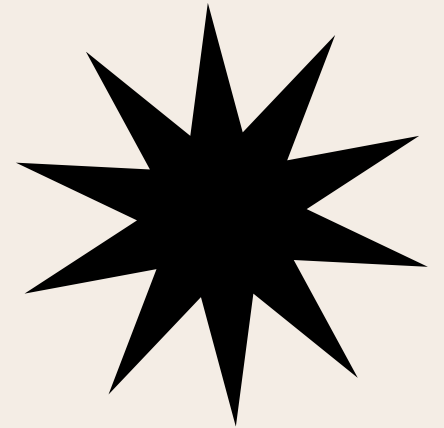
Balanceamento dinâmico

- As decisões são feitas em tempo de execução, considerando o estado atual do sistema.
- Flexível e adaptável a mudanças na carga de trabalho.



ROUND-ROBIN

ESTÁTICO



01



O que é?

- Algoritmo de balanceamento de carga que distribui as requisições de **forma sequencial e cíclica** entre os servidores disponíveis.

02



Como funciona?

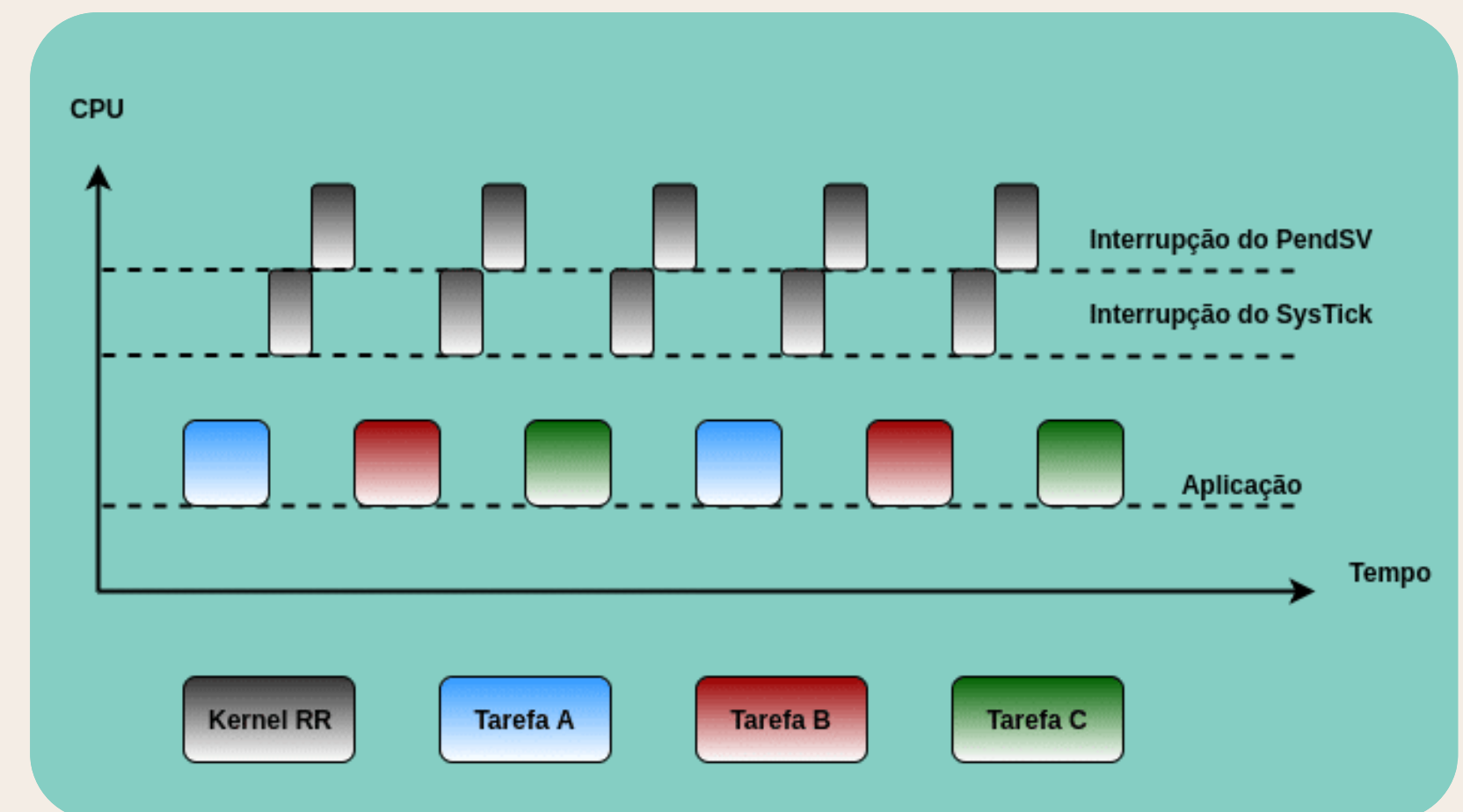
1. As requisições chegam a um intermediário.
2. Cada requisição é atribuída ao próximo servidor na lista.
3. Algoritmo retorna ao primeiro servidor e reinicia o ciclo.

03



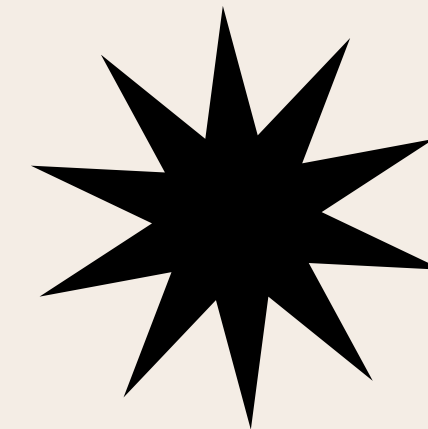
Vantagens e Desvantagens

- Simplicidade e Equidade
- Não considera o estado dos servidores.



LEAST CONNECTIONS

DINÂMICO



01



O que é?

- Distribui requisições com base no número atual de conexões ativas em cada servidor.
- Garante que a requisição seja enviada para o servidor menos ocupado no momento

02



Como funciona?

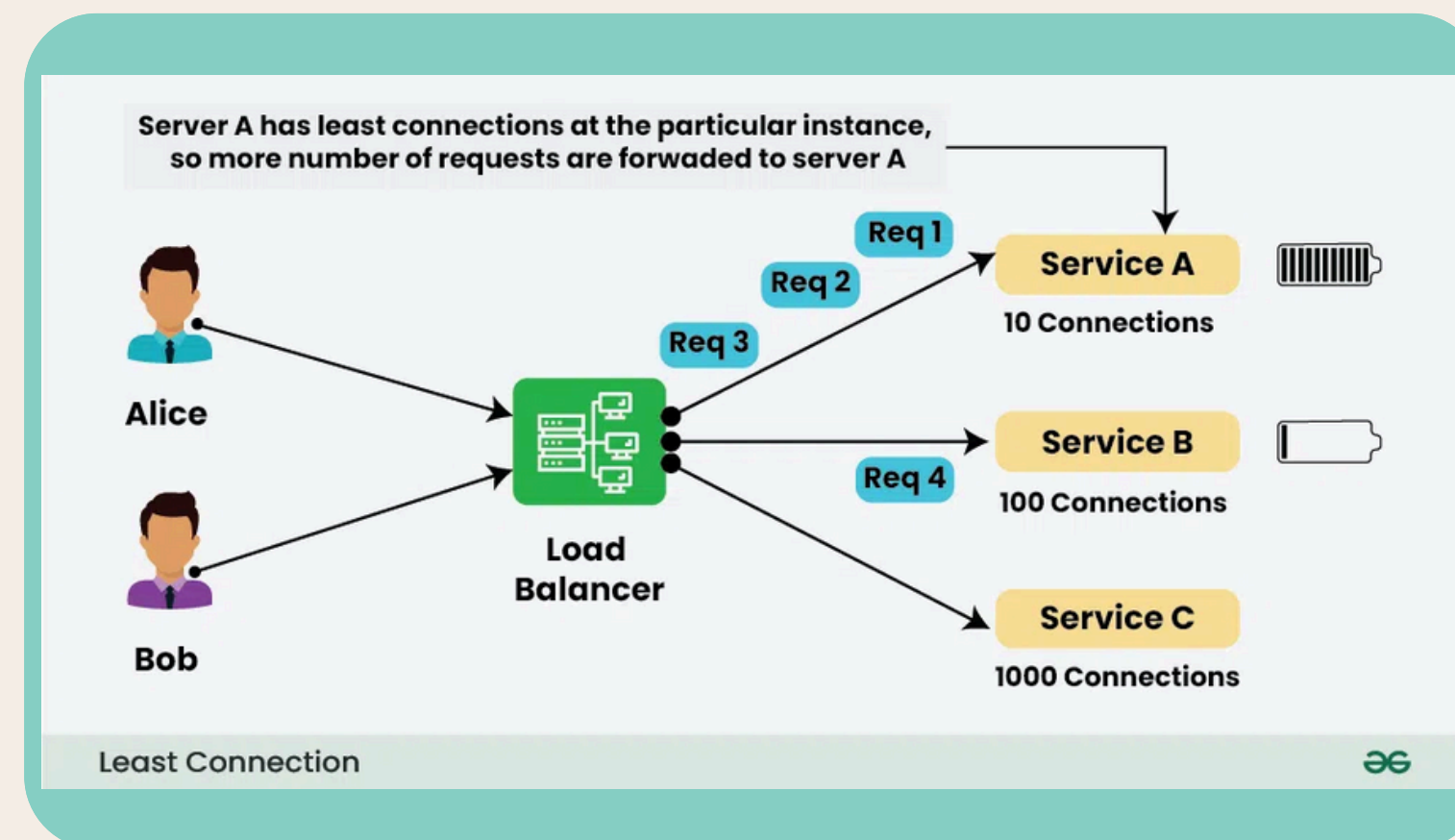
- Quando uma nova requisição chega, o balanceador de carga analisa todos os servidores disponíveis.
- Ele identifica o servidor com o menor número de conexões.
- A requisição é então direcionada para esse servidor

03



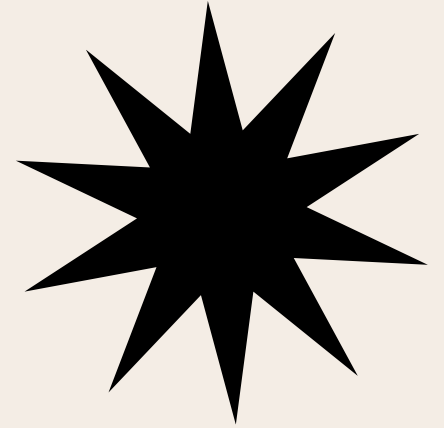
Vantagens e Desvantagens

- Eficiência em cargas variáveis.
- Maior complexidade e depende de informações mais precisas.



WEIGHTED ROUND-ROBIN

ESTÁTICO



01



O que é?

- Algoritmo de balanceamento que considera pesos atribuídos aos servidores.

02



Como funciona?

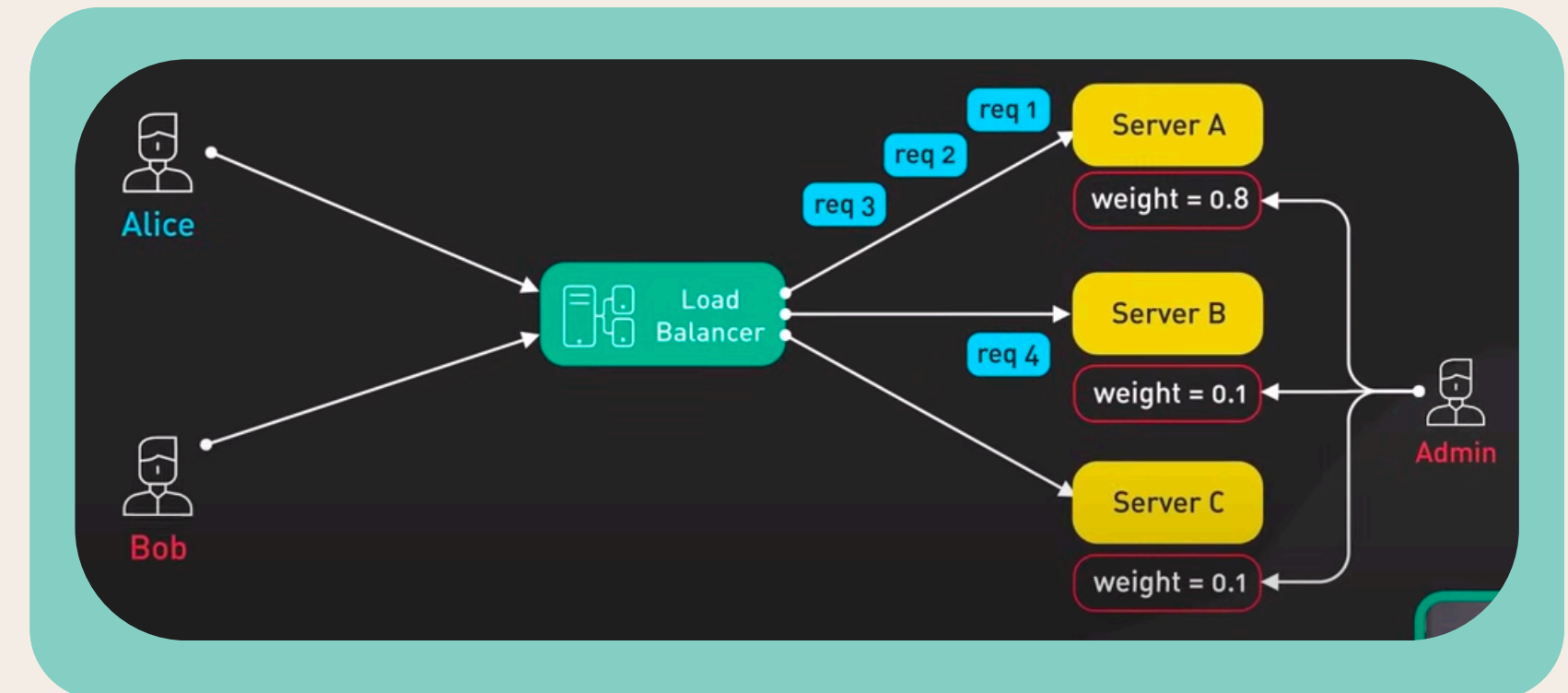
- Atribuição de pesos aos servidores.
- Distribuição de requisições proporcional aos pesos.
- Rotação cíclica para atender às requisições.

03



Vantagens e Desvantagens

- Eficiência e simplicidade.
- Necessidade de ajustar pesos manualmente.



CONSISTENT HASHING

ESTÁTICO/DINÂMICO

01



O que é?

- Algoritmo projetado para resolver o problema de distribuição eficiente de dados entre vários servidores.

02



Como funciona?

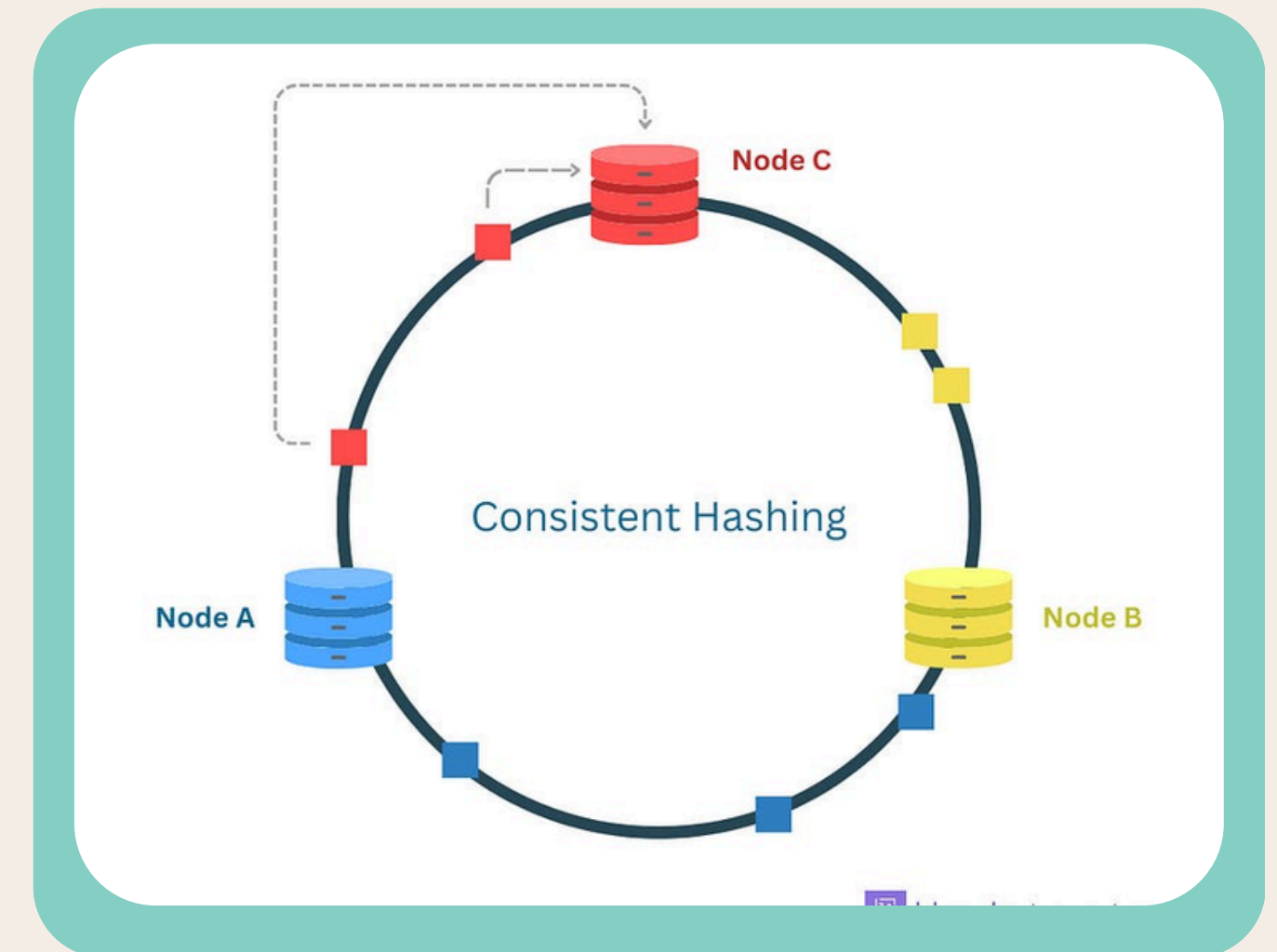
- Hash Circular.
- Distribuição de dados.
- Adicionando e Removendo nós.

03



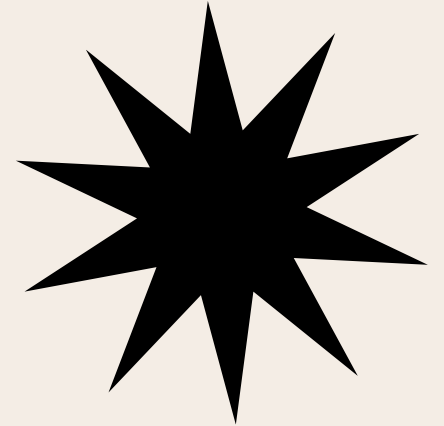
Vantagens e Desvantagens

- Escalabilidade e Redistribuição Mínima.
- Distribuição Desigual e Dependência da Função Hash.



RANDOMIZED ALGORITHMS

DINÂMICO



01



O que é?

- Algoritmos que incorporam aleatoriedade em seus processos para resolver problemas.

02



Como funciona?

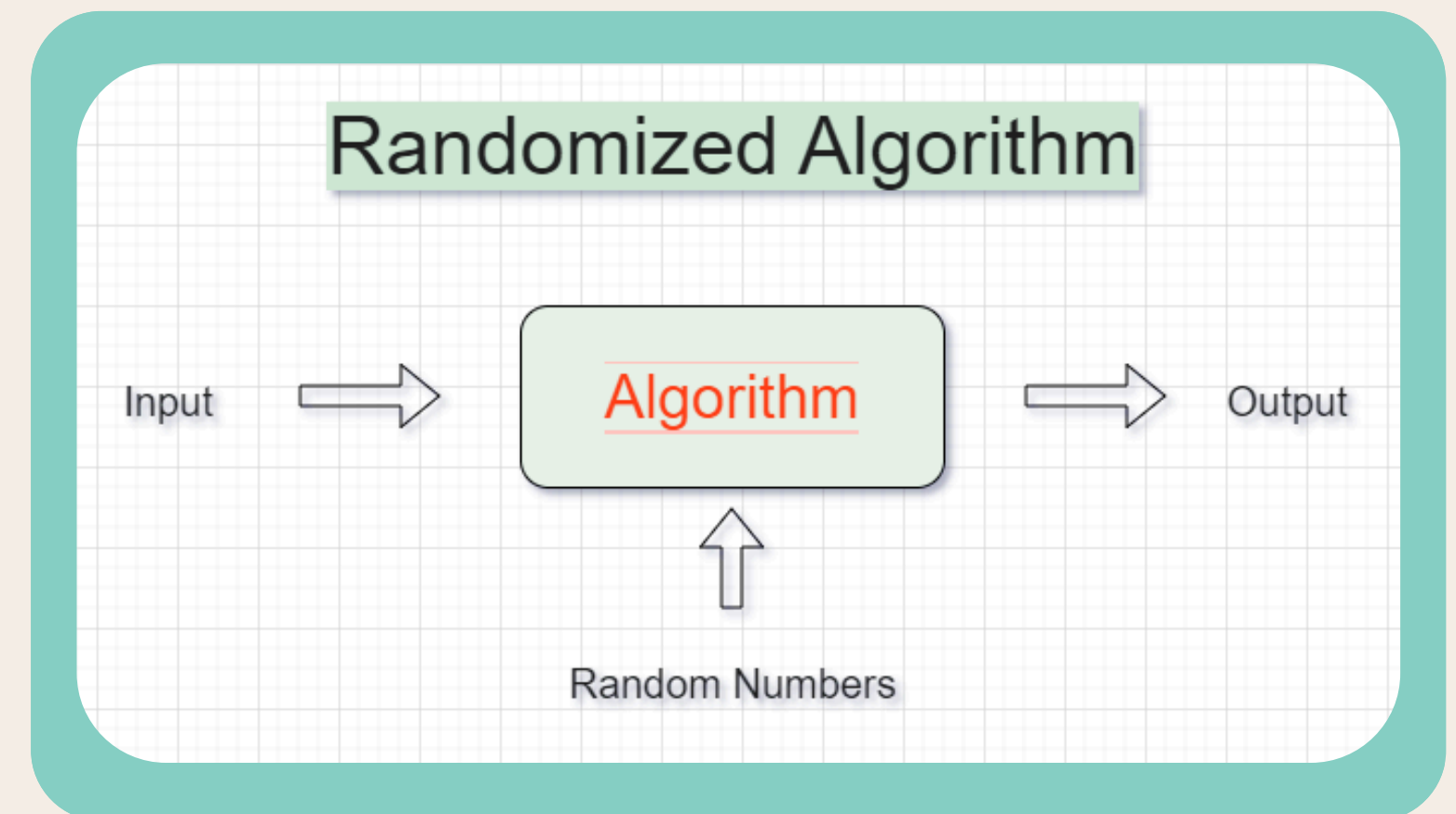
- Escolher um servidor aleatoriamente para processar uma requisição.
- Selecionar um caminho ou estratégia em problemas de otimização.

03



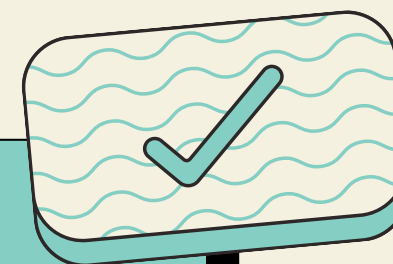
Vantagens e Desvantagens

- Simplicidade e Eficiência.
- Não Determinísticos e Impacto da má aleatoriedade.





CASOS DE USO



SERVIDORES WEB

Em sistemas web distribuídos, os servidores atendem requisições de usuários simultaneamente. O Load Balancer atua como intermediário, distribuindo essas requisições entre servidores de forma equilibrada.

Evita que um servidor fique sobrecarregado enquanto outros permanecem ociosos, mantendo a performance e evitando falhas.

CLOUD COMPUTING

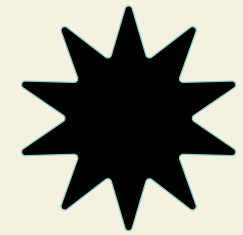
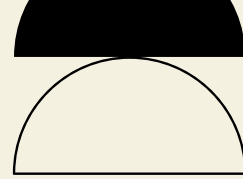
No ambiente de cloud computing, as cargas de trabalho são distribuídas dinamicamente entre máquinas virtuais ou contêineres. Os algoritmos monitoram constantemente métricas como uso de CPU, memória e armazenamento. Quando um recurso está sobrecarregado, a carga é redistribuída automaticamente.

Garante a escalabilidade, ou seja, o sistema cresce conforme a demanda e maximiza o uso dos recursos, reduzindo custos

IA E PROCESSAMENTO DISTRIBUÍDO

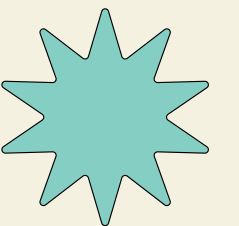
Em aplicações de Machine Learning e IA, tarefas complexas são divididas e distribuídas entre diferentes GPUs ou nós de processamento. Algoritmos de balanceamento garantem que os cálculos sejam realizados de forma paralela e eficiente.

Durante o treinamento de modelos grandes as tarefas são balanceadas entre vários dispositivos de hardware. Isso acelera o processamento e otimiza o uso dos recursos.

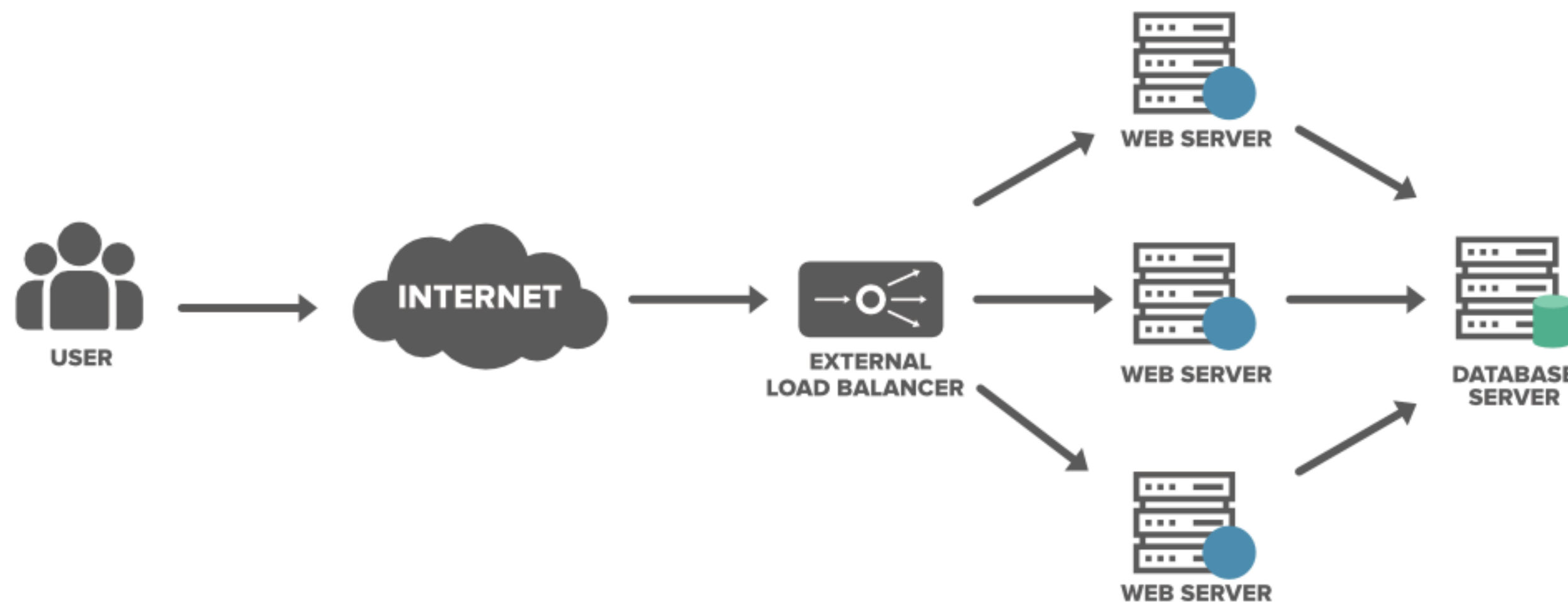
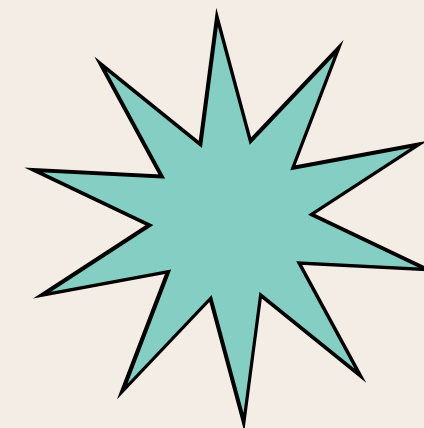


CONCLUSÃO

Algoritmos de balanceamento são essenciais em sistemas distribuídos. Eles trazem melhorias significativas na performance, na disponibilidade na escalabilidade e na segurança das aplicações

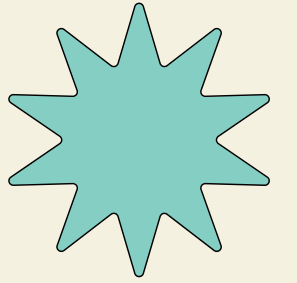
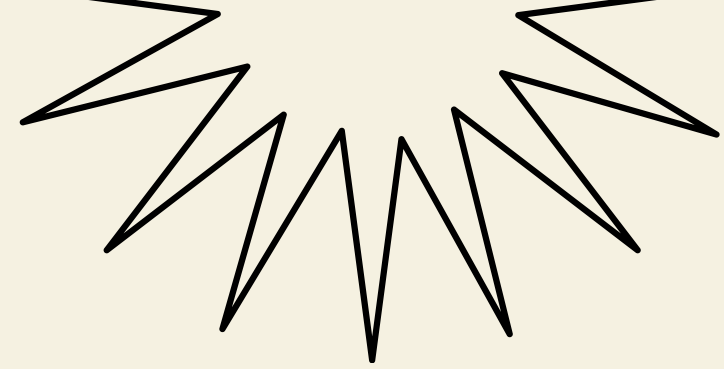


IMPLEMENTAÇÃO



ROUND-ROBIN EM SISTEMA DISTRIBUÍDO

- Comunicação em Rede (TCP via sockets)
- Balanceador como intermediário entre cliente e servidor



OBRIGADA

