

## **Relatório do projeto de busca distribuída utilizando sockets**

1. A **Computação Distribuída** refere-se a um modelo no qual várias máquinas trabalham juntas para resolver problemas computacionais de forma coordenada. Essas máquinas, frequentemente chamadas “nós”, comunicam-se via redes, compartilhando dados e tarefas. A ideia principal é dividir a carga de trabalho para aumentar a eficiência, reduzir o tempo de execução e aproveitar a capacidade de processamento de múltiplos dispositivos.

De acordo com Tanenbaum e Steen (2007), “um sistema distribuído é um conjunto de computadores independentes que se apresentam ao usuário como um único sistema coerente.” Isso ilustra como a integração dos nós permite entregar uma experiência unificada, apesar da complexidade.

Sistemas distribuídos são amplamente utilizados em aplicações modernas, que envolvem grandes conjuntos de dados, justamente por permitir que esses dados sejam processados em paralelo e viabilizar alta disponibilidade por meio de redundância.

2. **Escalabilidade** é a capacidade de um sistema crescer em termos de desempenho ou capacidade, sem grandes modificações em sua arquitetura. No contexto da busca distribuída, a escalabilidade é crucial. A adição de novos servidores (nós) permite que o sistema lide com um maior número de consultas e volumes de dados sem impactar negativamente o desempenho.

2.1. **Tolerância a Falhas** refere-se à capacidade de um sistema continuar operando corretamente mesmo quando um ou mais componentes falham. Em sistemas distribuídos, essa propriedade é alcançada por meio de:

**Redundância:** replicação de dados em múltiplos servidores;

**Deteção e recuperação de falhas:** uso de monitoramento para identificar falhas e redirecionar tarefas automaticamente.

3. **Vantagens e desvantagens dessa arquitetura são:**

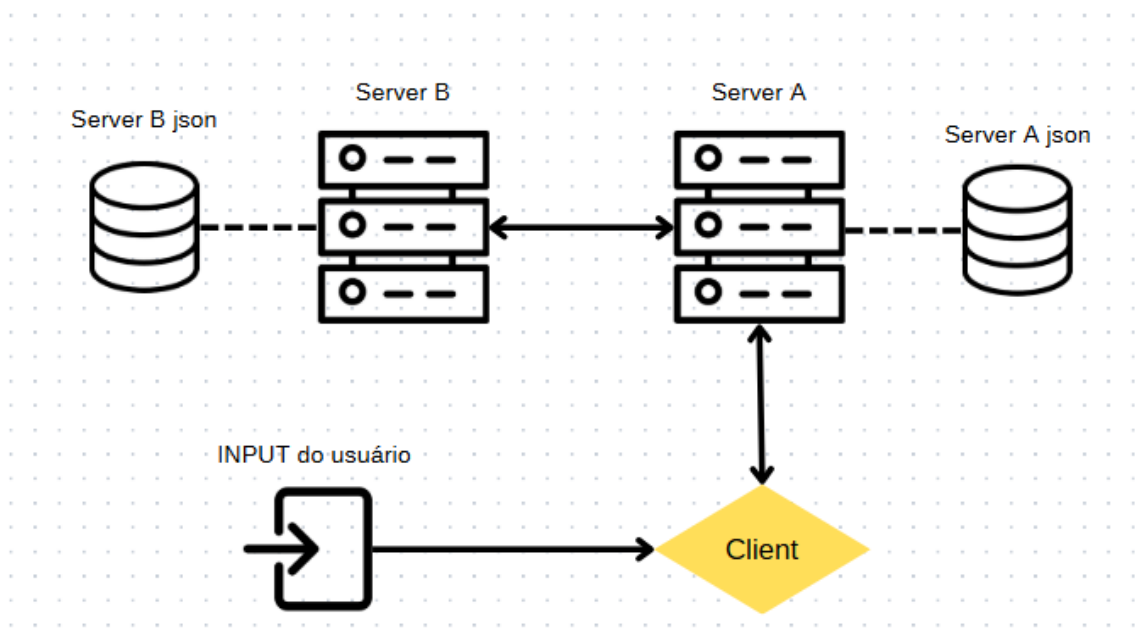
3.1. **Vantagens:**

- **Alta disponibilidade:** Propriedade do sistema de continuar operacional mesmo em caso de falha em um ou mais servidores.
- **Desempenho:** Consultas e tarefas podem ser executadas em paralelo através de múltiplos servidores. Reduzindo tempo de resposta.
- **Escalabilidade:** A capacidade do sistema pode ser aumentada facilmente adicionando novos servidores à rede, sem necessidade de reestruturação.
- **Flexibilidade:** Servidores podem ser distribuídos geograficamente, permitindo acesso rápido a dados localizados em diferentes localidades.

3.2. **Desvantagens:**

- **Complexidade:** A implementação, manutenção e monitoramento de um sistema distribuído exigem mais esforço, considerando comunicação, sincronização e consistência entre os servidores.
- **Latência:** A comunicação entre servidores e clientes através da rede pode resultar em atrasos, principalmente entre locais geograficamente distantes.
- **Custo:** A operação e manutenção de múltiplos servidores aumentam os custos com infraestrutura, energia, hardware etc.
- **Consistência:** Garantir que todos os servidores tenham os dados mais recentes pode ser um problema, que pode resultar em conflitos ou inconsistências.

Segue um exemplo simplificado do fluxo de comunicação:



4.0. O Formato dos Dados trafegados entre os servidores e o Client são representados em JSON, devido à sua simplicidade e suporte nativo em Java.

Exemplo de requisição:

```
{
  "query": "palavra-chave",
}
```

Exemplo de resposta:

```
{
  "resultados":
  [
    {"categoria": astronomia, "título": "palavra-chave"}
    (...)
  ]
}
```

- 4.1. A busca é realizada de forma distribuída, onde cada servidor:
1. Recebe uma query para consulta JSON.
  2. Realiza a busca local distribuída nos arquivos JSON armazenados.
  3. Retorna os resultados no terminal do ServerB->ServerA->ClientGUI

O servidor A agrega as respostas, concatenando as informações retornadas.

## **5. Referências Bibliográficas**

- Tanenbaum, A. S., & Steen, M. V. (2007). *Distributed Systems: Principles and Paradigms*. Pearson Education.
- Harold, E. R. (2013). *Java Network Programming*. O'Reilly Media.