

**Pró-Reitoria Acadêmica**  
**Curso de (Ciência da Computação)**  
**Trabalho de Disciplina**

**Autor: Paulo Henrique Pereira Silva**  
**Orientador: Prof. João Robson Santos Martins**

## Introdução Teórica

A computação distribuída é um paradigma da ciência da computação que busca dividir uma tarefa entre vários computadores autônomos, que se comunicam e coordenam entre si através de uma rede para alcançar um objetivo comum. Cada nó do sistema possui sua própria memória e poder de processamento, o que diferencia esse modelo de sistemas paralelos com memória compartilhada. Segundo *Tanenbaum* (2007), um sistema distribuído é “uma coleção de computadores independentes que se apresenta ao usuário como um sistema único e coerente”.

No contexto do projeto desenvolvido, a computação distribuída é representada pela separação do processamento de busca em três servidores: o Servidor A, responsável pela orquestração, e os Servidores B e C, responsáveis pela execução da busca em partes diferentes do conjunto de dados.

A escalabilidade, conceito essencial na computação distribuída, refere-se à capacidade de um sistema se expandir de forma eficiente conforme aumenta a demanda. Neste projeto, se houvesse a necessidade de processar conjuntos ainda maiores, seria possível adicionar novos servidores sem grandes alterações na estrutura existente. Conforme destaca *Coulouris et al.* (2011), a escalabilidade de um sistema distribuído está diretamente relacionada à sua capacidade de manter o desempenho conforme cresce a carga de trabalho ou o número de usuários.

Já a tolerância a falhas diz respeito à habilidade do sistema continuar operando mesmo quando há falhas em parte dos seus componentes. Embora o projeto não implemente mecanismos formais de redundância, a arquitetura modular permite que servidores independentes sejam reiniciados ou substituídos com impacto mínimo. Ainda segundo os autores, a tolerância a falhas exige que partes do sistema possam falhar sem que o sistema como um todo pare de funcionar.

Entre as vantagens da computação distribuída estão a melhora no desempenho, a escalabilidade natural, e a possibilidade de reaproveitar recursos computacionais distribuídos geograficamente. Por outro lado, desafios como a sincronização, latência de comunicação e complexidade de depuração são algumas das desvantagens associadas.

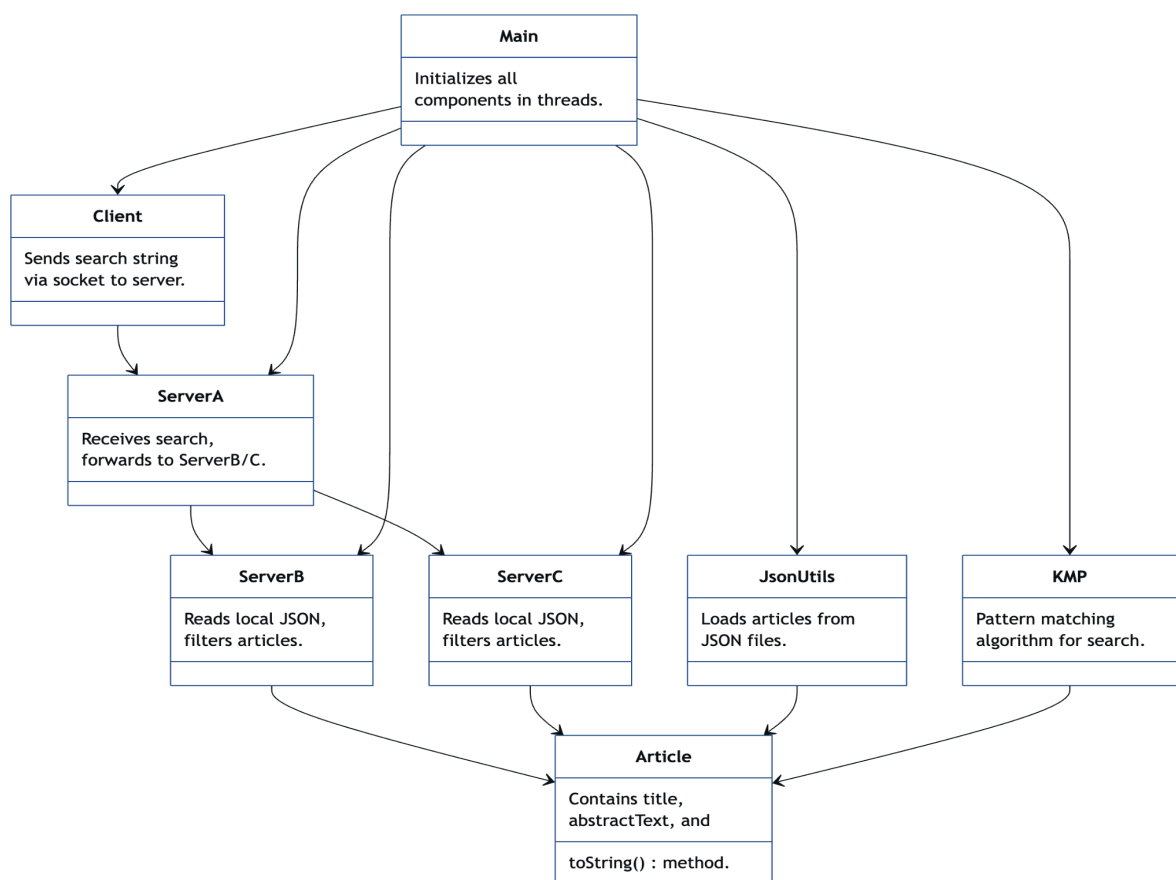
---

## Arquitetura da Solução

A arquitetura é composta por três servidores e um cliente, que se comunicam via sockets TCP. O Servidor A recebe uma string de busca do Cliente, encaminha a solicitação para os Servidores B e C (que processam arquivos JSON distintos), e consolida os resultados recebidos para retornar ao Cliente.

Fluxo de comunicação:

- Cliente → Servidor A → {Servidor B + Servidor C} → Servidor A → Cliente



O dado trafegado consiste em uma string (o termo de busca) enviada do Cliente ao Servidor A, que a repassa para os demais. Os servidores B e C retornam uma lista de artigos que contenham o termo no título ou resumo, além de informar quantas vezes o termo aparece em seus textos.

O formato do retorno é uma string formatada com título e resumo, seguida de uma linha com a contagem de ocorrências. O Servidor A soma esses dados e envia ao cliente, que salva o resultado no arquivo *'resultados.txt'*. A leitura dos arquivos no formato **JSON** foi

feita com a biblioteca `org.json`, amplamente utilizada para manipulação de dados em Java (JSON.org, 2025).

O algoritmo de busca utilizado foi o ***Knuth-Morris-Pratt (KMP)***, conhecido por sua eficiência em encontrar substrings dentro de textos longos. A decisão pelo KMP foi tomada devido à sua complexidade linear  $O(n + m)$ , que garante bom desempenho mesmo com grandes volumes de dados, como os encontrados nos arquivos **JSON**. O algoritmo foi originalmente proposto por Knuth, Morris e Pratt (1977) para resolver o problema de busca de padrões com mínimo de retrocessos no texto analisado.

---

**Brasília  
2025**

## **REFERÊNCIAS**

TANENBAUM, A. S.; VAN STEEN, M. Sistemas Distribuídos: princípios e paradigmas. 2. ed. São Paulo: Pearson, 2007.

GeeksForGeeks. KMP (Knuth Morris Pratt) Pattern Searching. Disponível em: <https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/>. Acesso em: 15 jun. 2025.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. Distributed Systems: Concepts and Design. 5. ed. Pearson, 2011.

COMER, D. E. Internetworking with TCP/IP – Principles, Protocols, and Architecture. Vol. 1. 6. ed. Pearson, 2013.

KNUTH, D. E.; MORRIS JR., J. H.; PRATT, V. R. Fast pattern matching in strings. SIAM Journal on Computing, v. 6, n. 2, p. 323–350, 1977.

JSON.org. The Java JSON Library (org.json). Disponível em: <https://www.json.org/json-en.html>. Acesso em: 15 jun. 2025.

---