

Projeto Final

Crie um projeto (**em Java 17**) para representar um sistema de busca distribuído utilizando *sockets*. O sistema deverá ser formado por três servidores (A, B e C). Dois dos três servidores serão **responsáveis por uma busca em metade de um dado de artigos científicos** do arXiv (<https://paperswithcode.com/dataset/arxiv-10>).

Um dos servidores (A) deverá receber a solicitação de busca do cliente, **contendo uma substring de um possível título ou introdução de um artigo**, enviar ao outros dois servidores (B e C) e em seguida ambos devem realizar a busca no dado designado a cada um.

Após a busca, o servidor A deve enviar ao cliente o resultado da busca.

Observações:

- Os arquivos JSON com os dados será divulgado previamente pelo professor;
- A comunicação entre cliente e servidor deve ser feita **obrigatoriamente** por meio de *sockets*;
- O algoritmo de busca em cada servidor deverá ser escolhido, **justificado e explicado** pelo grupo. **A string enviada pelo cliente deve ser buscada como uma substring, ou seja, amostras que contenham a string enviada no título ou na introdução devem ser retornados.** Algumas opções de algoritmos são listadas aqui: <https://www.geeksforgeeks.org/pattern-searching/>

Critérios de avaliação

O projeto vale no máximo 3 pontos na nota do bimestre (N2), sendo que a avaliação será baseada em dos tipos de critérios:

Coletivos (2.0 pontos):

- **Relatório (0.5):** relatório, no formato A4, com espaçamento 1,5 linhas e fonte Times New Roman 12, contendo no mínimo:
 - Explicação teórica, com no mínimo uma página e uma citação de referência bibliográfica (artigo científico ou livro), explicando os seguintes tópicos:
 - O que é computação distribuída;
 - Os conceitos de escalabilidade e tolerância a falhas e como eles se aplicam ao projeto.
 - Vantagens e desvantagens de utilizar essa arquitetura
 - Arquitetura da solução, contendo
 - Diagrama explicando comunicação durante uma busca entre cliente e servidores;
 - Formato do dado (**escolhido pelo grupo**) trafegado entre servidor A, B e C e servidor A e cliente;
 - **Explicação e justificativa do algoritmo de busca utilizado.**
 - Listagem das referências bibliográficas utilizadas, ao fim.
- **Criação do servidores: 0.5**
 - Correta implementação, usando as classes necessárias e realizando operações de entrada e saída corretamente.
 - Correta implementação da busca
 - Correta implementação da comunicação com sockets
- **Criação do cliente: 0.5**
 - Correta implementação, usando as classes necessárias e realizando operações de entrada e saída corretamente.
 - Correta implementação da comunicação com sockets
- **Estruturação e organização do código: 0.4;**
 - Divisão do projeto em classes;
 - Modularização do código, utilizando métodos sempre que possível;
 - Organização/clareza do código (nomes significativos de variáveis, indentação, etc.);
- **Organização do GitHub (README, comentários e tamanho dos commits, etc.) (0.1);**

Individuais (1.0 ponto):

- Participação no desenvolvimento do projeto (quantidade e qualidade dos commits e proporção de commits feitos por cada membro) (1);

Observações

- **O trabalho deve ser feito em grupo, mas qualquer tipo de plágio/cola será penalizado (o projeto receberá nota 0).**
- **Projetos com erro de sintaxe ou que não possam ser executados irão receber nota 0 também**
- A entrega será dia 15/06, até às 23:59, pelo AVA.
- O envio deve conter apenas um link para o repositório (que deve estar público) com o código do projeto.
 - **NÃO ENVIEM UM ZIP OU COLOQUEM O PROJETO ZIPADO NO GITHUB!**

Material de apoio

Java

- Trabalhando com JSON em Java: o pacote org.json. Disponível em: <https://www.devmedia.com.br/trabalhando-com-json-em-java-o-pacote-org-json/25480>
- Introdução ao Maven. Disponível em: <https://www.devmedia.com.br/introducao-ao-maven/25128>

Ferramentas de busca de referências

- <https://pergamum.ucb.br/>
- <https://arxiv.org/>
- <https://scholar.google.com.br/?hl=pt>