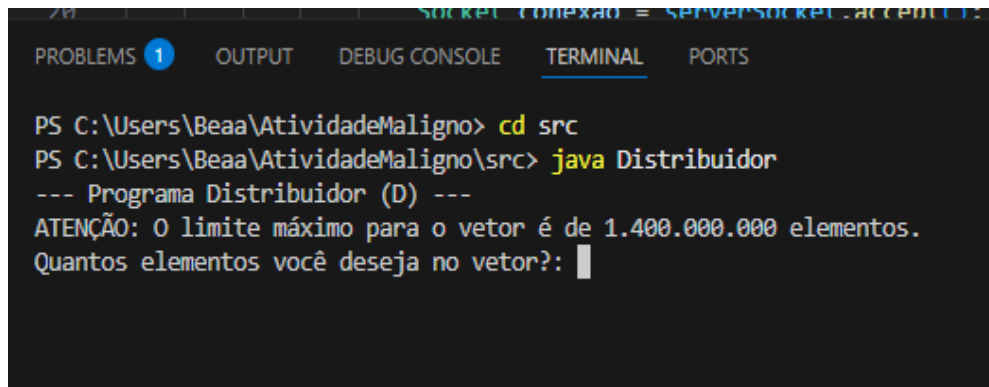


Trabalho Programação Paralela e Distribuída

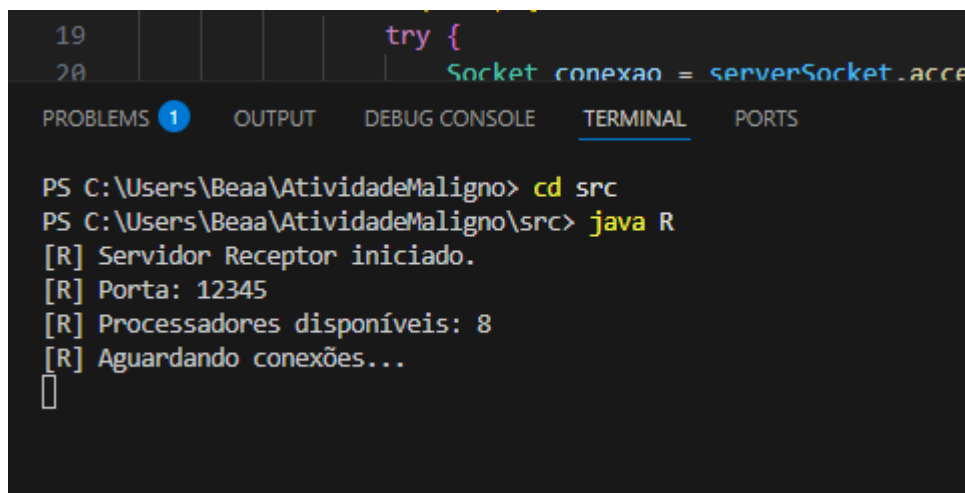
Grupo 2 - Beatriz Kamien Tehzy; Manoel Albino Coelho de Miranda Neto; Matheus Zanon Caritá

Prints dos testes



```
PS C:\Users\Beaa\AtividadeMaligno> cd src
PS C:\Users\Beaa\AtividadeMaligno\src> java Distribuidor
--- Programa Distribuidor (D) ---
ATENÇÃO: O limite máximo para o vetor é de 1.400.000.000 elementos.
Quantos elementos você deseja no vetor?:
```

Inicializando o distribuidor



```
PS C:\Users\Beaa\AtividadeMaligno> cd src
PS C:\Users\Beaa\AtividadeMaligno\src> java R
[R] Servidor Receptor iniciado.
[R] Porta: 12345
[R] Processadores disponíveis: 8
[R] Aguardando conexões...
```

Inicializando o R

```

Quantos elementos você deseja no vetor?: 1000
Deseja ver o vetor grande na tela? (s/n): s
Deseja testar um número que não existe (111)? (s/n): s

Vetor gerado (início): [-83, -80, -52, -2, -92, 54, 95, 75, 97, 12, -8, -34, 9, -58, -35, 89, 99, -51, 77, -41, 90, 67, 43, -11, 67, 61, -88, 85, 13, -70, 73, -85, -81, 30, -87, -28, 6, -24, -10, 39, -4, -5, 45, 31, -96, -96, -85, -88, 35, 41, 43, 10, -86, 81, -90, 46, -96, 52, 22, 87, 36, -56, -18, 86, -36, -38, 83, -6, -61, -26, -74, 18, -28, 86, 25, -92, -46, 1, -13, 52, -74, 39, 82, -90, 56, -55, 16, -99, 37, 51, 77, 35, 4, 63, 32, 44, -70, 89, 28, 33]

[D] MODO DE TESTE: Procurando o número 111 (deve dar 0).

[D] O número escolhido para procurar é: 111
[D] O vetor será dividido entre 3 servidores R.
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Enviando Pedido para 127.0.0.1...
[D] Resposta recebida de 127.0.0.1: 0

```

Teste de número não existente (Distribuidor)

```

PS C:\Users\Beaa\AtividadeMaligno\src> java R
[R] Servidor Receptor iniciado.
[R] Porta: 12345
[R] Processadores disponíveis: 8
[R] Aguardando conexões...
[R] Conexão aceita do cliente: 127.0.0.1
[R] Pedido recebido do cliente 127.0.0.1
[R] Tamanho do vetor: 333
[R] Número procurado: 111
[R] Contagem realizada: 0
[R] Tempo de processamento: 0ms
[R] Resposta enviada para 127.0.0.1

```

Teste de número não existente R

```

PS C:\Users\Beaa\AtividadeMaligno> cd src
PS C:\Users\Beaa\AtividadeMaligno\src> java Distribuidor
--- Programa Distribuidor (D) ---
ATENÇÃO: O limite máximo para o vetor é de 1.400.000.000 elementos.
Quantos elementos você deseja no vetor?: 1000
Deseja ver o vetor grande na tela? (s/n): s
Deseja testar um número que não existe (111)? (s/n): n

Vetor gerado (início): [2, -66, -1, 37, -77, -7, 60, -55, 36, 34, 11, 57, -76, 82, -47, -10, 44, 42, -67, 88, 49, -60, 44, -75, -65, -13, 63, -34, 75, 60, -71, -96, -57, 58, 14, -38, -30, -29, 47, 12, -21, -39, -53, 70, 31, -23, 5, -18, 31, 97, -65, -10, -78, 4, 100, -64, -60, -97, -73, -21, -12, 4, 60, -53, -52, 16, 20, -2, 48, -68, -22, -95, 63, 97, -5, 68, -42, 15, 5, -13, -57, 77, -81, -54, 53, 61, 23, 96, 39, -17, 73, -65, 28, 4, 6, 90, -27, -34, -82, 46]

[D] O número escolhido para procurar é: -93
[D] O vetor será dividido entre 3 servidores R.
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Enviando Pedido para 127.0.0.1...
[D] Resposta recebida de 127.0.0.1: 2

```

Teste de número existente Distribuidor

```

PS C:\Users\Beaa\AtividadeMaligno> cd src
PS C:\Users\Beaa\AtividadeMaligno\src> java R
[R] Servidor Receptor iniciado.
[R] Porta: 12345
[R] Processadores disponíveis: 8
[R] Aguardando conexões...
[R] Conexão aceita do cliente: 127.0.0.1
[R] Pedido recebido do cliente 127.0.0.1
[R] Tamanho do vetor: 333
[R] Número procurado: -93
[R] Contagem realizada: 2
[R] Tempo de processamento: 0ms
[R] Resposta enviada para 127.0.0.1

```

Teste de número existente R

```

PS C:\Users\Beaa\AtividadeMaligno> cd src
PS C:\Users\Beaa\AtividadeMaligno\src> java ContagemSequencial
--- Programa de Contagem SEQUENCIAL ---
Quantos elementos você deseja no vetor? (Máx: 1.400.000.000): 5000
[S] O número escolhido para procurar é: -55
[S] Iniciando contagem sequencial em um único bloco...

--- RESULTADO SEQUENCIAL ---
[S] Contagem Final: 23 ocorrências do número -55
[S] Tempo total de execução (Sequencial): 0,003 segundos
PS C:\Users\Beaa\AtividadeMaligno\src>

```

Comparação sequencial com vetor de 5000 - Contagem Sequencial

```

PS C:\Users\Beaa\AtividadeMaligno> cd src
PS C:\Users\Beaa\AtividadeMaligno\src> java Distribuidor
--- Programa Distribuidor (D) ---
ATENÇÃO: O limite máximo para o vetor é de 1.400.000.000 elementos.
Quantos elementos você deseja no vetor?: 5000
Deseja ver o vetor grande na tela? (s/n): s
Deseja testar um número que não existe (111)? (s/n): n

Vetor gerado (início): [77, -6, 7, -10, 70, 99, -35, 81, 76, -37, 91, -16, 40, -62, -46, -21, 99, 74, 73, -71, 15, 64, -4, 13, 47, 76, -59,
-36, -74, -5, -43, 0, -95, 1, 11, -3, -86, -7, -6, -71, -72, -38, -48, -15, -5, 56, -36, 34, -57, 43, 89, -49, 82, -40, 88, -39, 45, -90,
92, -9, -7, 29, -21, -58, 92, 76, 57, -23, -52, -43, 76, 82, -21, -76, 2, -58, 85, 64, -89, -24, 18, -54, 45, -66, 9, -1, -48, 84, 55, -
7, 26, 23, 13, -16, -66, -65, 7, 76, -72, -40]

[D] O número escolhido para procurar é: 31
[D] O vetor será dividido entre 3 servidores R.
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Enviando Pedido para 127.0.0.1...
[D] Resposta recebida de 127.0.0.1: 7

```

Comparação sequencial com vetor de 5000 - Distribuidor

```

PS C:\Users\Beaa\AtividadeMaligno> cd src
PS C:\Users\Beaa\AtividadeMaligno\src> java R
[R] Servidor Receptor iniciado.
[R] Porta: 12345
[R] Processadores disponíveis: 8
[R] Aguardando conexões...
[R] Conexão aceita do cliente: 127.0.0.1
[R] Pedido recebido do cliente 127.0.0.1
[R] Tamanho do vetor: 1666
[R] Número procurado: 31
[R] Processando com 8 threads
[R] Contagem realizada: 7
[R] Tempo de processamento: 9ms
[R] Resposta enviada para 127.0.0.1

```

Comparação sequencial com vetor de 5000 - R

```

PS C:\Users\Beaa\AtividadeMaligno> cd src
PS C:\Users\Beaa\AtividadeMaligno\src> java ContagemSequencial
--- Programa de Contagem SEQUENCIAL ---
Quantos elementos você deseja no vetor? (Máx: 1.400.000.000): 400000000
[S] O número escolhido para procurar é: -62
[S] Iniciando contagem sequencial em um único bloco...

--- RESULTADO SEQUENCIAL ---
[S] Contagem Final: 1989786 ocorrências do número -62
[S] Tempo total de execução (Sequencial): 0,160 segundos
PS C:\Users\Beaa\AtividadeMaligno\src>

```

Comparação sequencial com vetor de 400000000 - Contagem Sequencial

```

PS C:\Users\Beaa\AtividadeMaligno> cd src
PS C:\Users\Beaa\AtividadeMaligno\src> java Distribuidor
--- Programa Distribuidor (D) ---
ATENÇÃO: O limite máximo para o vetor é de 1.400.000.000 elementos.
Quantos elementos você deseja no vetor?: 400000000
Deseja ver o vetor grande na tela? (s/n): s
Deseja testar um número que não existe (111)? (s/n): n

Vetor gerado (início): [31, 56, 53, -49, 15, 39, 84, -59, -57, 3, 80, -64, -66, 15, 74, 31, 81, -11, -12, -70, -87, 25, 58, -28, 39, -40,
13, 86, -25, -55, 29, -40, -64, -14, -80, 93, 14, -54, 42, -40, 45, -86, -79, -9, 87, -36, -60, -35, -51, -24, 73, -53, -22, -77, 76, 18,
-52, 66, -41, -75, -39, 68, 25, -10, 25, 75, -71, 58, -68, 0, 10, 99, -12, 95, -29, -60, -77, 80, -67, 49, 35, -13, 12, -55, 78, 76, 31, -
96, 16, 16, 25, 84, 0, 39, -47, 47, -38, -57, 9, 24]

[D] O número escolhido para procurar é: -70
[D] O vetor será dividido entre 3 servidores R.
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Enviando Pedido para 127.0.0.1...
[D] Resposta recebida de 127.0.0.1: 663715

```

Comparação sequencial com vetor de 400000000 - Distribuidor

```

PS C:\Users\Beaa\AtividadeMaligno> cd src
PS C:\Users\Beaa\AtividadeMaligno\src> java R
[R] Servidor Receptor iniciado.
[R] Porta: 12345
[R] Processadores disponíveis: 8
[R] Aguardando conexões...
[R] Conexão aceita do cliente: 127.0.0.1
[R] Pedido recebido do cliente 127.0.0.1
[R] Tamanho do vetor: 13333333
[R] Número procurado: -70
[R] Processando com 8 threads
[R] Contagem realizada: 663715
[R] Tempo de processamento: 42ms
[R] Resposta enviada para 127.0.0.1

```

Comparação sequencial com vetor de 400000000 - R

Teste com dois Clientes:

R:

```

→ AtividadeMaligno git:(main) ✗ java -cp bin R
[R] Servidor Receptor iniciado.
[R] Porta: 12345
[R] Processadores disponíveis: 20
[R] Aguardando conexões...
[R] Conexão aceita do cliente: 127.0.0.1
[R] Pedido recebido do cliente 127.0.0.1
[R] Tamanho do vetor: 3000000
[R] Número procurado: 83
[R] Processando com 20 threads
[R] Contagem realizada: 15012
[R] Tempo de processamento: 69ms
[R] Resposta enviada para 127.0.0.1
[R] Conexão encerrada pelo cliente 127.0.0.1
[R] Conexão fechada com 127.0.0.1
[R] Conexão aceita do cliente: 127.0.0.1
[R] Conexão encerrada pelo cliente 127.0.0.1
[R] Conexão fechada com 127.0.0.1
[R] Conexão aceita do cliente: 127.0.0.1
[R] Conexão encerrada pelo cliente 127.0.0.1
[R] Conexão fechada com 127.0.0.1
[R] Conexão aceita do cliente: 127.0.0.1
[R] Pedido recebido do cliente 127.0.0.1
[R] Tamanho do vetor: 1666666
[R] Número procurado: -45
[R] Processando com 20 threads
[R] Contagem realizada: 8283
[R] Tempo de processamento: 5ms
[R] Resposta enviada para 127.0.0.1

```

Distribuidor A:

```
AtividadeMaligno git:(main) ✗ java -cp bin Distribuidor

... Programa Distribuidor ...
ATENÇÃO: O limite máximo para o vetor é de 1,400,000,000 elementos.
Quantos elementos você deseja no vetor?: 9000000
Deseja ver o vetor grande na tela? (s/n): s
Deseja testar um número que não existe (111)? (s/n): n

Vetor gerado (início): [60, 41, -39, 96, 32, 13, 91, -24, -40, -19, 45, 14, -34, -13, -85, 60, -33, 11, -63, 40, -11, -57, -3,
38, 9, 26, 79, -30, 34, -30, -10, 83, -36, -69, 61, 17, -2, -16, 87, 95, 75, 86, 16, -17, 85, -63, -53, -49, -17, 42, 19, 50, 7
, 80, -14, -73, -70, -67, -83, 70, -68, -90, -97, -91, 93, 74, 18, -78, -50, -57, 44, 10, 88, 19, -79, -77, 32, 29, -3, 30, -79
-28, 55, -72, 70, 74, -93, -59, -88, 60, 84, 55, -22, -19, 1, 15, -25, 78, 49, -70]

[D] O número escolhido para procurar é: 83
[D] O vetor será dividido entre 3 servidores R.
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Enviando Pedido para 127.0.0.1...
[D] Resposta recebida de 127.0.0.1: 15012
^C
AtividadeMaligno git:(main) ✗
```

Distribuidor B:

```
AtividadeMaligno git:(main) ✗ java -cp bin Distribuidor

... Programa Distribuidor ...
ATENÇÃO: O limite máximo para o vetor é de 1,400,000,000 elementos.
Quantos elementos você deseja no vetor?: 5000000
Deseja ver o vetor grande na tela? (s/n): s
Deseja testar um número que não existe (111)? (s/n): n

Vetor gerado (início): [-43, 10, 60, -58, -15, -24, -49, 90, -11, -44, 73, -99, -44, 46, 99, -87, -16, 66, -27, 53, 40, -30, 96,
-81, -88, 5, -82, 98, 46, -87, -72, -24, 76, 19, 22, 29, 72, 48, -54, 3, -93, 34, -1, 99, -12, 15, 64, 94, 63, -84, 64, -39, -7
5, -70, 65, -31, -45, -80, -86, -20, -1, -95, -73, -92, -96, 31, 38, -32, 5, -96, -66, -70, -30, -48, 44, 78, 51, 7, 22, 71, -66
, -67, -11, -18, -72, 5, 28, 34, -11, 78, 87, -17, -82, -91, -23, -44, -82, 45, 58, 78]

[D] O número escolhido para procurar é: -45
[D] O vetor será dividido entre 3 servidores R.
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Thread para 127.0.0.1 iniciando conexão...
[D] Enviando Pedido para 127.0.0.1...
[D] Resposta recebida de 127.0.0.1: 8283
```

Relato breve (até 10 linhas) sobre os testes realizados.

Com o intuito de provar a capacidade do programa de executar várias funções, vários testes foram feitos, como por exemplo o teste de conexão com múltiplos clientes. Nesse teste, o servidor foi capaz de cuidar de cada conexão individualmente, sem a existência de concorrência entre os múltiplos clientes. Além disso, testes básicos como testes de vetores enormes, comparações sequenciais, números inexistentes foram feitos. Após isso, o programa foi capaz de realizar cada teste e prover os resultados esperados.

Relatos pessoais de desenvolvimento da equipe:

Beatriz Kamien Tehzy: Dentre os desafios enfrentados no projeto, dou ênfase na fase de integração do código, que foi onde realmente juntamos as peças de todo o time. Essa parte exigiu uma dedicação extra, especialmente no debugging durante os testes, que foi fundamental para conseguirmos aplicar os patches de correção e garantir que o sistema ficasse robusto. Achei super interessante e um ótimo aprendizado a forma como aplicamos a divisão de tarefas utilizando o conceito de paralelismo. Ver a performance melhorar com essa otimização foi bem gratificante, e o fato de termos usado Java facilitou muito, já que é uma linguagem que adoro trabalhar. O projeto era complexo, não era nada simples, mas era exatamente esse o desafio. Ter conseguido entregar e fazer tudo funcionar perfeitamente me deu uma sensação de competência muito grande. Fiquei feliz e com o sentimento de que evolui muito.

Manoel Albino Coelho de Miranda Neto: A implementação da comunicação básica com Sockets foi um ótimo exercício sobre arquitetura cliente-servidor. No entanto, a verdadeira complexidade surgiu ao introduzir Threads para gerenciar a concorrência. Garantir que múltiplas tarefas pudessem executar em paralelo, acessando recursos de forma segura e sem conflitos, exigiu um estudo aprofundado sobre sincronização. Foi um aprendizado intenso que solidificou meus conhecimentos sobre os desafios práticos da programação concorrente e da comunicação em rede.

Matheus Zanon Caritá: O desenvolvimento deste projeto nos trouxe vários desafios, posso citar como sendo os principais: o tempo para a realização do projeto, tivemos que manejar bem as tarefas para cada um, caso contrário não teríamos tempo para acabar o projeto; a integração do trabalho que cada um fez, que as vezes gerou certa incompatibilidade devido a por exemplo, nomes de variáveis; a interpretação do enunciado em geral nos trouxe certas confusões, como por exemplo na obtenção dos IPs dos computadores através do IP config, em que não sabíamos ao certo no momento qual era o IP que deveria ser utilizado para deixar hard-coded no projeto. Por fim, os testes foram muito úteis para eliminar dúvidas e até mesmo erros escondidos do projeto, como por exemplo os servidores aceitarem mais de uma máquina ao mesmo tempo.