

## **RELATÓRIO DO SEGUNDO EXERCÍCIO PROGRAMA DE PROGRAMAÇÃO CONCORRENTE**

**Nomes:** Karina Suemi Awoki      **nºUSP:** 7572102  
Mateus Barros Rodrigues      7991037

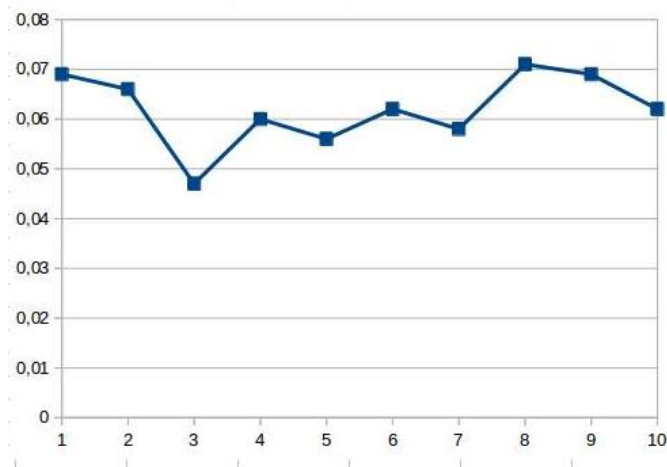
## Configurações da máquina:

- Arquitetura: x86\_64
- Modo(s) operacional da CPU: 32-bit, 64-bit
- Ordem dos bytes: Little Endian
- CPU(s): 8
- Lista de CPU(s) on-line: 0-7
- Thread(s) per núcleo 2
- Núcleo(s) por soquete: 4
- Soquete(s): 1
- Nó(s) de NUMA: 1
- ID de fornecedor: GenuineIntel
- Família da CPU: 6
- Modelo: 42
- Nome do modelo: Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz
- Step: 7
- CPU MHz: 950.000
- CPU MHz máx.: 2900,0000
- CPU MHz mín.: 800,0000
- BogomIPS: 3992.19
- Virtualização: VT-x
- cache de L1d: 32K
- cache de L1i: 32K
- cache de L2: 256K
- cache de L3: 6144K
- CPU(s) de nó0 NUMA: 0-7

## Entradas:

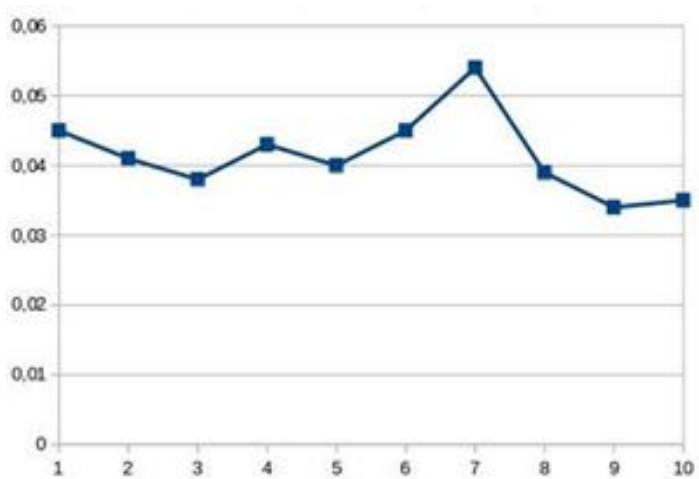
- As informações dos gráficos a seguir representam os valores de **vez em que foi rodado vs tempo de execução** em segundos.
- Os tempos usados nos gráficos foram determinados a partir do comando “*time*” no terminal.
- Os gráficos a seguir foram obtidos com os seguintes valores:
  - $f = 10^{-1000}$
  - $x = 3,14$
- Foram executados 10 testes em cada quantidade de threads (sequencial, 3 cores, 5 cores e 8 cores)

## 1. Sequencial:



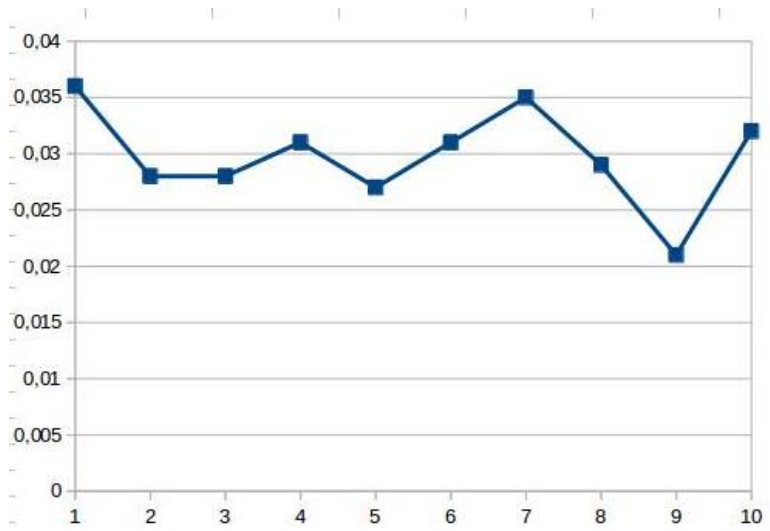
- **média:** 0,062
- **desvio padrão:** 0,0068992753
- **intervalo de confiança(90%):** (0,054645373; 0,069354627)

## 2. 3 CORES:



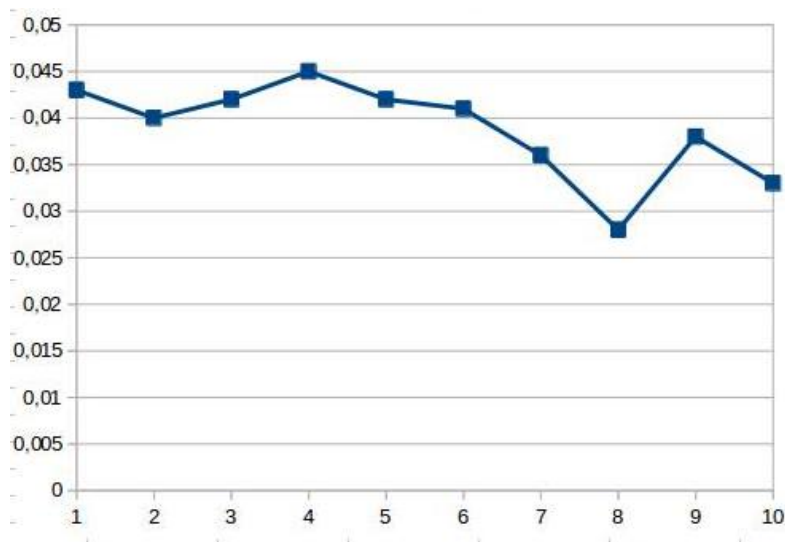
- **média:** 0,0414
- **desvio padrão:** 0,0054990908
- **intervalo de confiança(90%):** (0,035537969; 0,047262031)

### 3. 5 CORES:



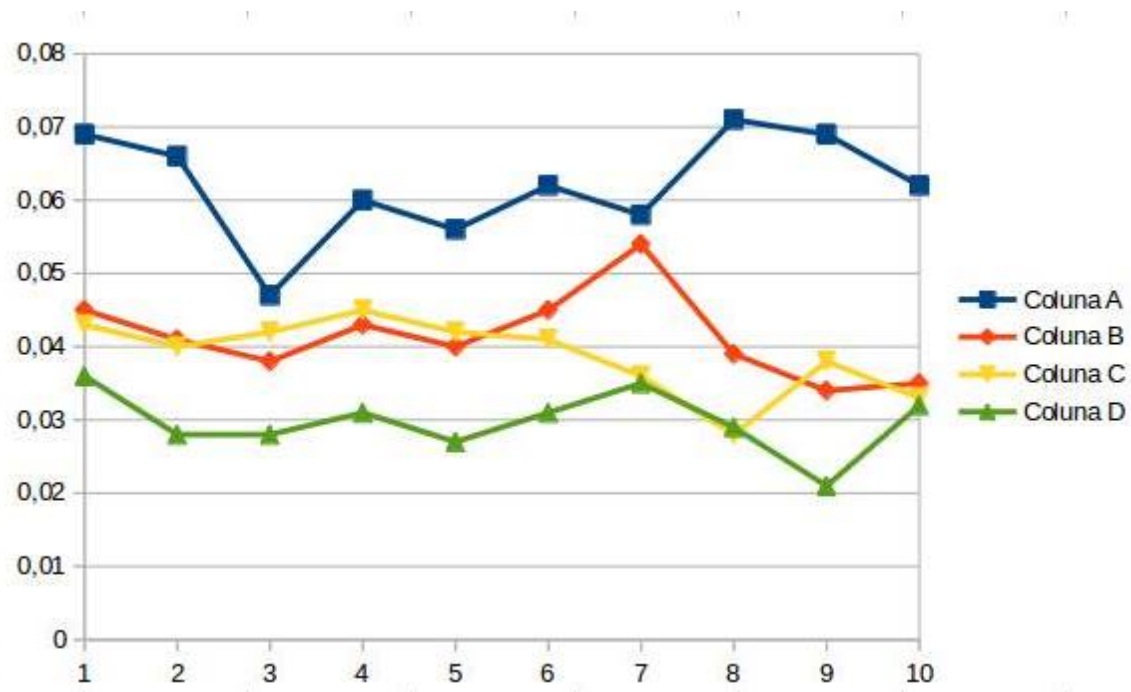
- **média:** 0,0388
- **desvio padrão:** 0,0049152823
- **intervalo de confiança(90%):** (0,0335603091; 0,0440396909)

### 4. 8 CORES:



- **média:** 0,0298
- **desvio padrão:** 0,004069398
- **intervalo de confiança(90%):** (0,0254620217; 0,0341379783)

## 5. Comparando:



Legenda:

- Coluna A = Sequencial
- Coluna B = 3 CORES
- Coluna C = 5 CORES
- Coluna D = 8 CORES

## Comentários sobre os resultados:

Em geral, os resultados para esses parâmetros foram condizentes com o esperado, o sequencial foi o mais devagar, para 3 e 5 threads tivemos tempos parecidos, mas a média de velocidade e o desvio padrão ainda foram melhores para 5 threads, e com 8 threads tivemos tempos melhores em quase todas as execuções.

## Comentários sobre a GMP:

A GMP internamente usa [Palavras](#) como elemento mais básico. Ela usa algoritmos diferentes dependendo do tamanho do número, pois geralmente algoritmos usados em números grandes não são eficientes para números pequenos. Ela usa de código assembly altamente otimizado nos laços internos mais importantes, especializados de acordo com o processador.