

Análise de Algoritmo:

"VERIFICA ALGO"

Ryan Pimentel
Vicente Sampaio

```
VerificaAlgo (n: int);  
i, j, k, l: int;  
para l := 1 TO 10.000 faça  
    para i := 1 TO n-5 faça  
        para j := i+2 TO n/2 faça  
            para k := 1 TO n faça  
                {Inspezione elemento}
```

Função de custo e Complexidade:



$$\sum_{l=1}^{10000} \sum_{i=1}^{n-5} \sum_{j=i+2}^{n/2} \sum_{k=1}^n 1 =$$

(4) (3) (2) (1)

$$(1) \sum_{k=1}^n 1 = n$$

$$(2) \sum_{j=i+2}^{n/2} \sum_{k=1}^n 1 = \sum_{j=i+2}^{n/2} n = n \cdot \left(\frac{n}{2} - (i+2) + 1 \right) = n \cdot \left(\frac{n}{2} - i - 1 \right)$$

$$\begin{array}{cccc}
10000 & n-5 & n/2 & n \\
\sum_{l=1} & \sum_{i=1} & \sum_{j=i+2} & \sum_{k=1} 1 = \\
(4) & (3) & (2) & (1)
\end{array}$$

$$(3) \quad \sum_{i=1}^{n-5} \sum_{j=i+2}^{n/2} \sum_{k=1}^n 1 = \sum_{i=1}^{n-5} n \cdot \left(\frac{n}{2} - i - 1 \right) = n \sum_{i=1}^{n-5} \left(\frac{n}{2} - i - 1 \right)$$

$$= n \left[\sum_{i=1}^{n-5} \frac{n}{2} - \sum_{i=1}^{n-5} i - \sum_{i=1}^{n-5} 1 \right]$$

(3.3)
(3.2)
(3.1)

$$n \left[\sum_{i=1}^{n-5} \frac{n}{2} - \sum_{i=1}^{n-5} i - \sum_{i=1}^{n-5} 1 \right]$$

(3.3)
(3.2)
(3.1)

$$(3.1) \sum_{i=1}^{n-5} 1 = (n - 5)$$

$$(3.2) \sum_{i=1}^{n-5} i = \frac{(n - 5)(n - 4)}{2}$$

$$(3.3) \sum_{i=1}^{n-5} \frac{n}{2} = (n - 5) \cdot \frac{n}{2}$$

$$\begin{aligned}
(3) \quad & \sum_{i=1}^{n-5} \sum_{j=i+2}^{n/2} \sum_{k=1}^n 1 = n \left[\sum_{i=1}^{n-5} \frac{n}{2} - \sum_{i=1}^{n-5} i - \sum_{i=1}^{n-5} 1 \right] \\
&= n \left[(n-5) \cdot \frac{n}{2} - \frac{(n-5)(n-4)}{2} - (n-5) \right] \\
&= n(n-5) \left[\frac{n}{2} - \frac{n-4}{2} - 1 \right] = n(n-5) \left[\frac{n - (n-4)}{2} - 1 \right] \\
&= n(n-5) \left[\frac{4}{2} - 1 \right] = n(n-5) [2 - 1] \\
&= n(n-5) \cdot 1 = n(n-5)
\end{aligned}$$

$$\sum_{l=1}^{10000} \sum_{i=1}^{n-5} \sum_{j=i+2}^{n/2} \sum_{k=1}^n 1 =$$

$$= 10000 \cdot n(n-5)$$

$$= 10000n^2 - 50000n$$

$$T(n) = 10000n^2 - 50000n = O(n^2)$$

GitHub

https://github.com/saggita1/RyanPimentelVicenteSampaio_ws_AA_RR_2025.git

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <time.h>
4
5  double verifica_algo(int n, long long int *cont){
6      int i,j,k,l;
7      *cont = 0;
8
9
10     clock_t start = clock();
11     for (l = 1; l <= 10000; l++)
12     {
13         for (i = 1; i <= n-5; i++)
14         {
15             for (j = i+2; j <= n/2; j++)
16             {
17                 for (k = 1; k <= n; k++)
18                 {
19                     (*cont)++;
20                 }
21             }
22         }
23     }
24
25
26     clock_t end = clock();
27     double time = (double)(end - start) / CLOCKS_PER_SEC;
28
29     return time;
30 }
31
32
```

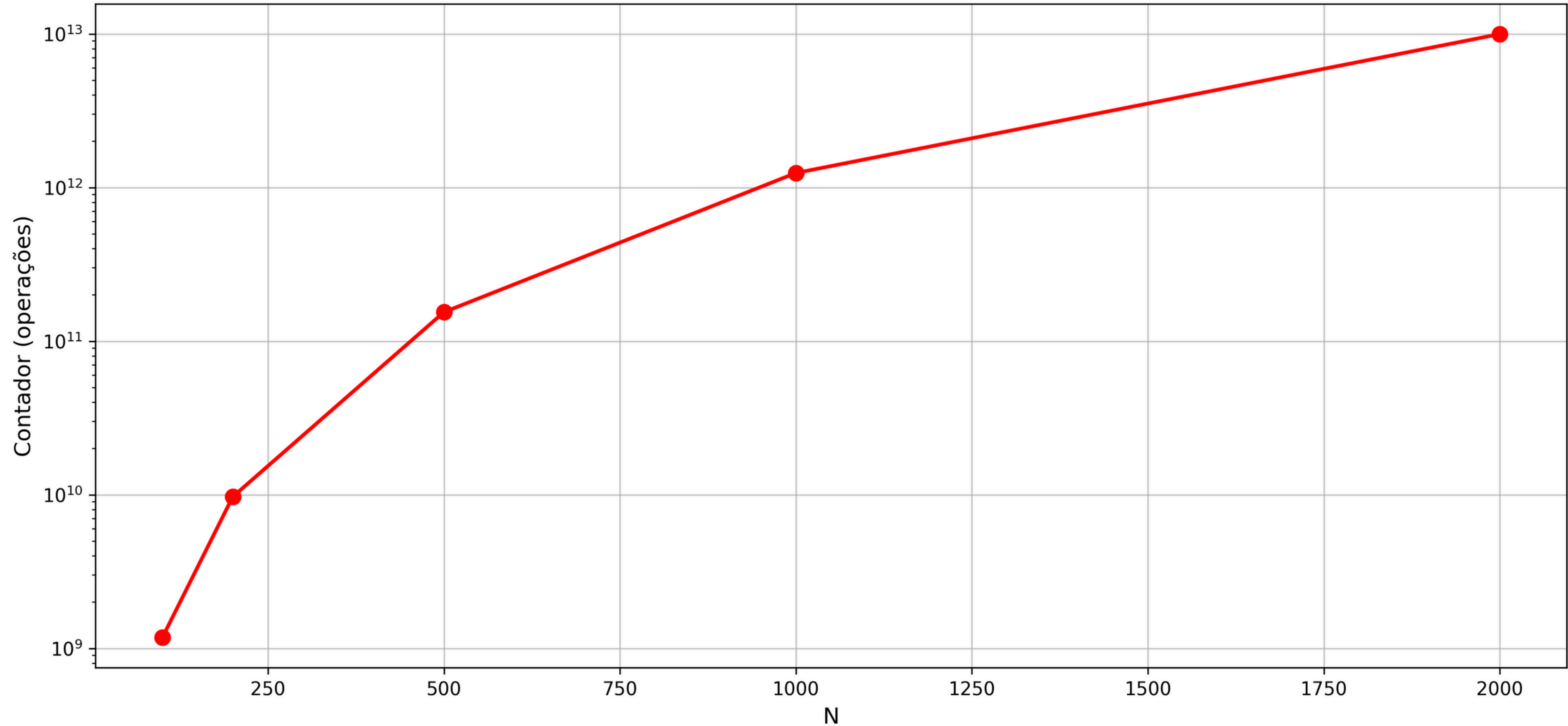
```
VerificaAlgo (n: int);
i, j, k, l: int;
para l := 1 TO 10.000 faça
    para i := 1 TO n-5 faça
        para j := i+2 TO n/2 faça
            para k := 1 TO n faça
                {Inspeção elemento}
```

GitHub

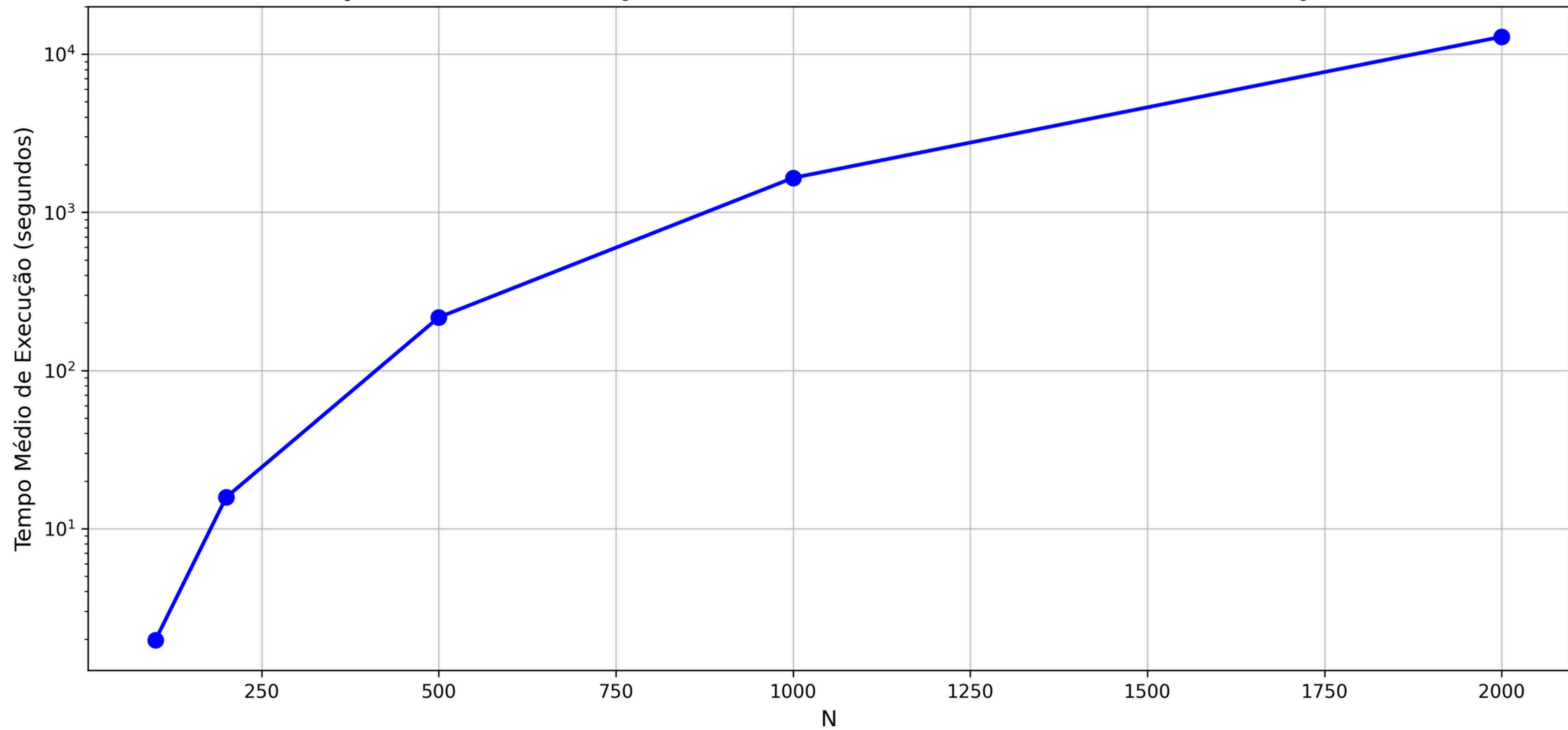
https://github.com/saggita1/RyanPimentelVicenteSampaio_ws_AA_RR_2025.git

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <time.h>
4
5  double verifica_algo(int n, long long int *cont){
6      int i,j,k,l;
7      *cont = 0;
8
9      clock_t start = clock();
10     for (l = 1; l <= 1; l++) {
11         for (i = 1; i <= 1; i++) {
12             for (j = 1; j <= 1; j++) {
13                 for (k = 1; k <= 1; k++) {
14                     *cont++;
15                 }
16             }
17         }
18     }
19
20     clock_t end = clock();
21     double time = (double)(end - start) / CLOCKS_PER_SEC;
22
23     return time;
24 }
25
```

Número de Operações vs Tamanho de Entrada



Tempo Médio de Execução vs Tamanho de Entrada (Média de 7 execuções)



OBRIGADO!