

## **PRÁTICA 1 – Trabalho 2 Exercício 1: Tempo experimental**

**Nome:** Davi Gabriel Domingues

**Número USP:** 15447497

O trabalho consiste na implementação em C++ dos algoritmos prefixMedia1() e prefixMedia2(), os quais foram apresentados no pseudocódigo esquematizado como modelo para a utilização das respectivas funções, uma para cada algoritmo.

Para esse propósito, foi – se desenvolvido o programa a seguir:

```
1      #include <iostream>
2      #include <ctime>
3      using namespace std;
4
5      void PrefixMedia1(double X[], double A[], int n){
6          double a;
7          for (int i = 0; i < n; i++){
8              a = 0.0;
9              for (int j = 0; j <= i; j++){
10                 a = a + X[j];
11                 A[i] = a / (i + 1);
12             }
13         }
14
15      void PrefixMedia2(double X[], double A[], int n){
16          double s = 0.0; // soma parcial.
17          for (int i = 0; i < n; i++){
18              s = s + X[i];
19              A[i] = s / (i + 1);
20          }
21      }
22
```

```

23 int main() {
24     int N, numero_Execucoes = 10;
25     clock_t tempol, tempo2, tempo3, tempo4; // variáveis do tipo tempo, em clocks.
26     double tempoTotal, soma = 0.0;
27
28     cout<<"Informe o total de elementos de interesse/tamanho do vetor: ";
29     cin>>N;
30
31     double *X = new double[N];
32     double *A = new double[N];
33
34     for (int i = 0; i < N; i++)
35         X[i] = 1;
36
37     cout<<"\n";
38
39     cout<<"Algoritmo Prefix_media_1:"<<endl;
40     for (int i = 0; i < numero_Execucoes; i++){ // medição de tempo para o algoritmo 1.
41         cout<<"Execucao "<< i + 1 <<endl;
42         tempol = clock();
43         PrefixMedial(X, A, N);
44         tempo2 = clock();
45
46         tempoTotal = difftime(tempo2,tempol)/CLOCKS_PER_SEC;
47         soma += tempoTotal;
48         cout<<"Tempo = "<<tempoTotal<<" segundos"<<endl;
49         cout<<"\n";
50     }
51     cout<<"Tempo medio: "<<soma/numero_Execucoes<<endl;
52
53     cout<<"\n";
54

```

```

55     soma = 0.0;
56     cout<<"Algoritmo Prefix_media 2:"<<endl;
57     for (int i = 0; i < numero_Execucoes; i++){ // medição de tempo para o algoritmo 2.
58         cout<<"Execução "<< i + 1 <<endl;
59         tempo3 = clock();
60         PrefixMedia2(X, A, N);
61         tempo4 = clock();
62
63         tempoTotal = difftime(tempo4,tempo3)/CLOCKS_PER_SEC;
64         soma += tempoTotal;
65         cout<<"Tempo = "<<tempoTotal<<" segundos"<<endl;
66         cout<<"\n";
67     }
68     cout<<"Tempo medio: "<<soma/numero_Execucoes<<endl;
69
70     cout<<"\n";
71
72     delete A;
73     delete X;
74
75     return 0;
76 }
77

```

Sendo assim, observam – se as respostas dos itens a seguir:

- a) Considerando intervalos de valores distintos, pelo menos cinco, escolhidos, obteve – se a seguinte tabela representativa dos tempos de execução de cada algoritmo:

#### **Algoritmo 1:**

N	Tempo (s) 1°	Tempo (s) 2°	Tempo (s) 3°	Tempo (s) 4°	Tempo (s) 5°	Tempo (s) 6°	Tempo (s) 7°	Tempo (s) 8°	Tempo (s) 9°	Tempo (s) 10°	Tempo médio
5000	0,016	0,016	0,023	0,031	0,016	0,031	0,031	0,016	0,015	0,015	0,0236
10000	0,111	0,104	0,103	0,103	0,101	0,099	0,1	0,101	0,102	0,101	0,1021
15000	0,243	0,232	0,226	0,23	0,229	0,224	0,224	0,222	0,226	0,232	0,2288
20000	0,433	0,402	0,406	0,401	0,405	0,401	0,402	0,396	0,397	0,397	0,404
30000	0,938	0,895	0,888	0,904	0,907	0,914	0,91	0,901	0,919	0,907	0,9083

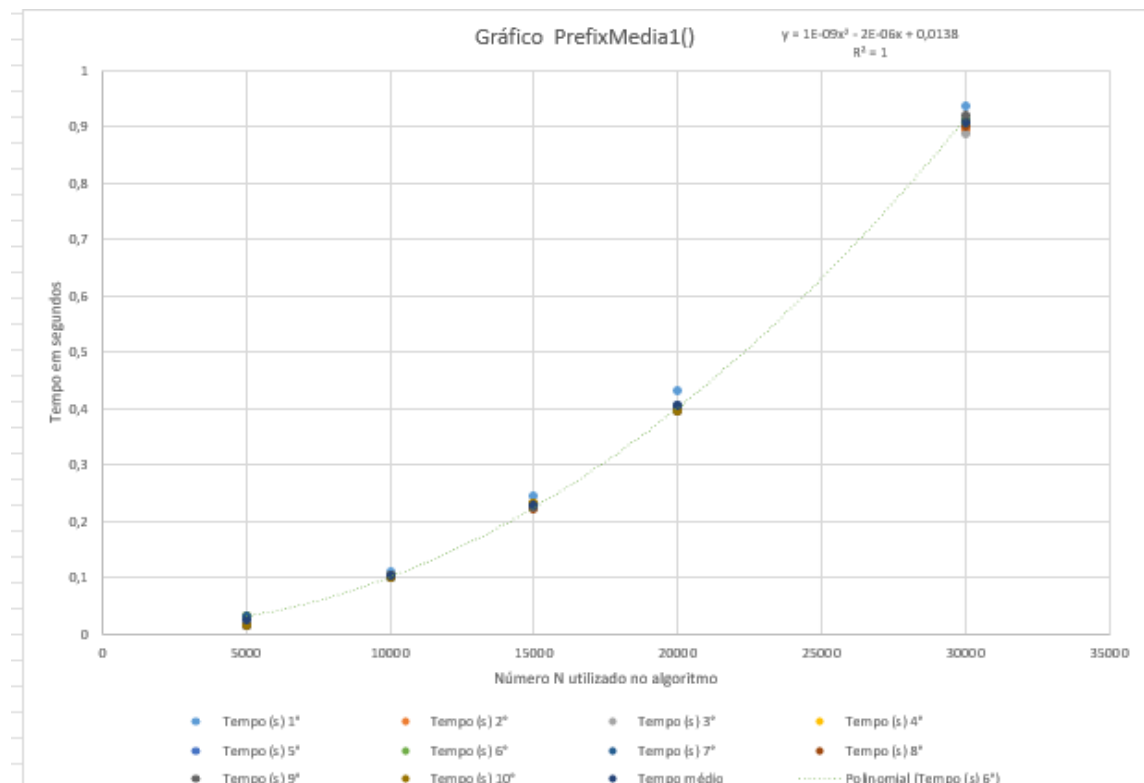
## Algoritmo 2:

N	Tempo (s) 1º	Tempo (s) 2º	Tempo (s) 3º	Tempo (s) 4º	Tempo (s) 5º	Tempo (s) 6º	Tempo (s) 7º	Tempo (s) 8º	Tempo (s) 9º	Tempo (s) 10º	Tempo médio
500000	0,001	0,001	0,001	0,001	0,002	0,001	0,002	0,002	0,001	0,002	0,0014
1000000	0,004	0,004	0,003	0,003	0,004	0,003	0,003	0,004	0,004	0,003	0,0036
3000000	0,012	0,011	0,01	0,009	0,011	0,01	0,01	0,007	0,01	0,008	0,0097
12000000	0,038	0,033	0,033	0,032	0,032	0,032	0,03	0,03	0,031	0,031	0,0321
60000000	0,2	0,153	0,153	0,172	0,177	0,172	0,152	0,158	0,153	0,154	0,1624

- b) A partir dos dados verificados no item “a)”, pode – se gerar um modelo de aproximação da variação dos dados, analisados nas tabelas acima, em um gráfico de n x tempo de execução, tal que n seja o número total de elementos presentes no vetor a ser analisado, ou seja, o “tamanho” do vetor (array de números, neste caso, de tipo double).

Dessa maneira, observam – se os gráficos a seguir:

### Gráfico do algoritmo prefixMedia1():



### Gráfico do algoritmo prefixMedia2():

