



```
-  
-      SENAO:  
-      adicional = 0;  
- listaauxiliar <- listaauxiliar.inverterordem();  
- listaauxiliar <- string(listaauxiliar);  
- ESCRIVA (listaauxiliar);  
- finalgoritmo;
```

### **- 1.2 Pseudocódigo do algoritmo de multiplicação (método Russo);**

```
- inicio;  
- LEIA (entrada1, entrada2);  
- soma <- 0;  
- ENQUANTO entrada1 >= 1:  
-   SE entrada1 restodadivisaopor 2 != 0:  
-     entrada1 <- entrada1 - 1;  
-     soma <- soma + entrada2;  
-   entrada1 <- entrada1 / 2;  
-   entrada2 <- entrada2 * 2;  
- ESCRIVA (soma)  
- finalgoritmo;
```

## **2. ESPECIFICAÇÕES DO COMPUTADOR ESCOLHIDO.**

O computador usado para a execução dos algoritmos objetos de estudo nesse trabalho possui as seguintes configurações:

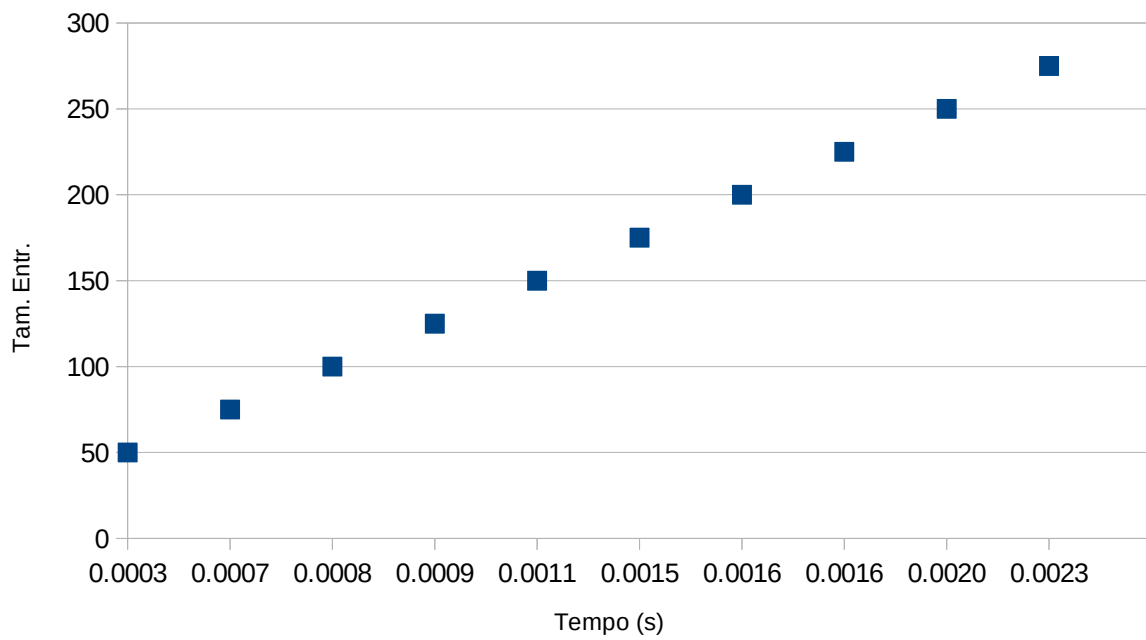
- Quanto ao processador, Intel(R) Core(TM) i5-6400 CPU @ 2.70 GHz ~ 2.71 GHz

- Referente à quantidade de memória principal, esse modelo possui 8,00 GB de memória RAM.

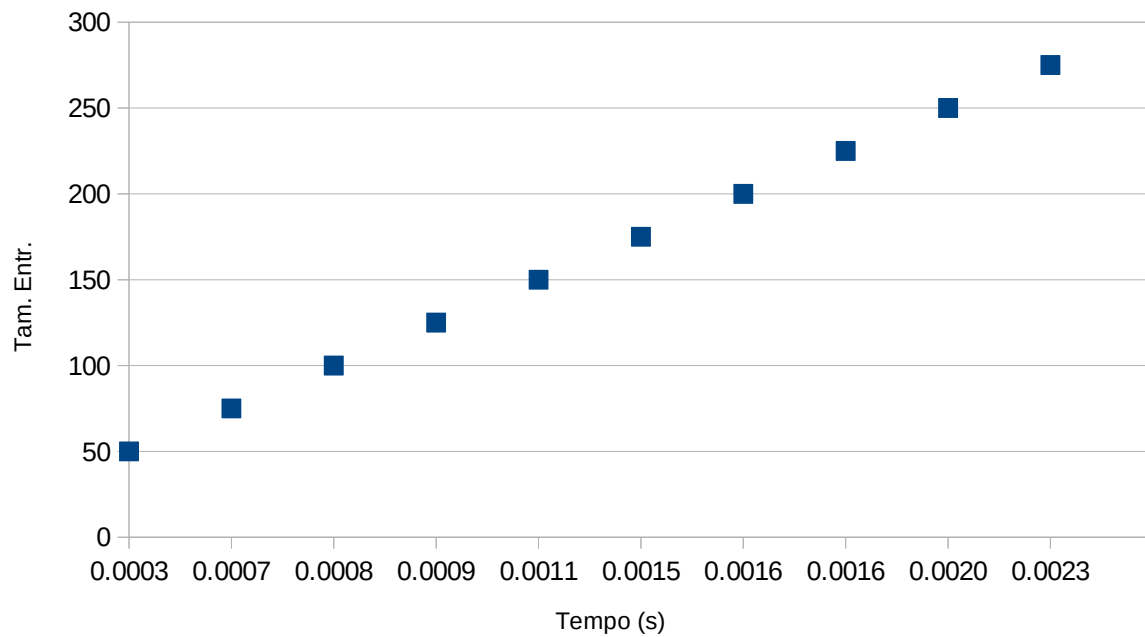
- Tratando-se do sistema operacional, Windows baseado em x64, sistema operacional de 64 bits, Versão 10 pro.

### 3. GRÁFICOS DOS ALGORITMOS.

#### 3.1 Gráfico do algoritmo de soma;



#### 3.2 Gráfico do algoritmo de multiplicação;



#### **4. ANÁLISE DOS RESULTADOS.**

De acordo com os resultados obtidos pelos testes, observa-se que para as mesmas entradas nos dois algoritmos com tamanho de 200 dígitos obteve-se com algoritmo de soma um tempo de execução de 0.000385 segundos e pelo de multiplicação 0.000417 segundos. Diferença mínima entre os tempos porém com entradas maiores essa diferença mantém-se como demonstrado nos gráficos. Os gráficos apresentam funções constantes para os algoritmos em questão refletindo exatamente como o esperado diante do tamanho da entrada. O resultado do algoritmo de soma no melhor caso resulta em no mesmo tamanho da entrada, ou seja, se a entrada for  $n$  o resultado será  $n$ , e no pior caso com entrada  $n$  a saída será  $n + 1$ . Para o algoritmo de multiplicação, no melhor caso com entrada de tamanho  $n$  resultará em  $n$  e no pior caso resultará em  $2n$ .

#### **5. CONCLUSÕES FINAIS.**

Os algoritmos estudados apresentam um tempo de execução linear que demonstra que eles são otimizados sempre com o tempo de execução variando de maneira equivalente de acordo com o tamanho das entradas.