

## Experimentos com algoritmos de ordenação.

Especificações do computador:

1. **Processador:** Intel Core i3 2350M
  - 1.1. Frequência do processador: 2.30 GHZ
2. **Memoria**
  - 2.1. **Tipo:** DDR3 com **6GB**
  - 2.2. **Frequência:** 665.1 MHz
3. **HD: Western Digital**
  - 3.1. **Transferência:** SATA-II 3.0Gb/s
  - 3.2. **Características:** S.M.A.R.T., APM, NCQ
  - 3.3. **Capacidade:** 149 GB
  - 3.4. **5400 RPM**

Link do Github com seus códigos e dados:

[https://github.com/VicenteDanilo/Algoritmos\\_de\\_Ordenacao](https://github.com/VicenteDanilo/Algoritmos_de_Ordenacao)

Ordenação **Bubble Sort1:**

Para uma entrada de **100** dados:

- A quantidade de **comparações**  
Com 100 dados: **10100 comparações**  
Com 10 mil dados: **999000 comparações**  
Com 100 mil dados: **81027002 comparações**
- A quantidade de **trocas**  
Com 100 dados: **2724 trocas**  
Com 10 mil dados: **210749 trocas**  
Com 100 mil dados: **19870632 trocas**
- O tempo de execução (em **segundos**)  
Com 100 dados: **0.1169424057006836 segundos**  
Com 10 mil dados: **1.7010843753814697 segundos**  
Com 100 mil dados: **139.98343873023987 segundos**

- O uso da **CPU**  
Com 100 dados: **20.0 %**  
Com 10 mil dados: **25.8 %**  
Com 100 mil dados: **27.1 %**
- O uso de **RAM**  
Com 100 dados: **43.9 %**  
Com 10 mil dados: **44.2 %**  
Com 100 mil dados: **44.5 %**

#### Ordenação **Bubble Sort2:**

- A quantidade de **comparações**  
Com 100 dados: **5252 comparações**  
Com 10 mil dados: **501500 comparações**  
Com 100 mil dados: **40531505 comparações**
- A quantidade de **trocas**  
Com 100 dados: **2084 trocas**  
Com 10 mil dados: **158109 trocas**  
Com 100 mil dados: **14905230 trocas**
- O tempo de execução (em **segundos**)  
Com 100 dados: **0.03634381294250488 segundos**  
Com 10 mil dados: **1.0215890407562256 segundos**  
Com 100 mil dados: **71.9771671295166 segundos**
- O uso da **CPU**  
Com 100 dados: **25.0 %**  
Com 10 mil dados: **45.7 %**  
Com 100 mil dados: **25.9 %**

- O uso de **RAM**  
Com 100 dados: **44.1 %**  
Com 10 mil dados: **44.1 %**  
Com 100 mil dados: **44.2 %**

#### Ordenação **Bubble Sort3**:

- A quantidade de comparações  
Com 100 dados: **10000 comparações**  
Com 10 mil dados: **843156 comparações**  
Com 100 mil dados: **80342926 comparações**
- A quantidade de trocas  
Com 100 dados: **2724 trocas**  
Com 10 mil dados: **210749 trocas**  
Com 100 mil dados: **19870632 trocas**
- O tempo de execução (em segundos)  
Com 100 dados: **0.0794215202331543 segundos**  
Com 10 mil dados: **1.4612314701080322 segundos**  
Com 100 mil dados: **138.1727409362793 segundos**
- O uso da **CPU**  
Com 100 dados: **15.8 %**  
Com 10 mil dados: **30.3 %**  
Com 100 mil dados: **27.5 %**
- O uso de **RAM**  
Com 100 dados: **44.6 %**  
Com 10 mil dados: **44.4 %**  
Com 100 mil dados: **44.6 %**

#### Ordenação **Insertion Sort**:

- A quantidade de **comparações**  
Com 100 dados: **100 comparações**  
Com 10 mil dados: **999 comparações**  
Com 100 mil dados: **9001 comparações**
- A quantidade de trocas  
Com 100 dados: **2724 trocas**  
Com 10 mil dados: **210749 trocas**  
Com 100 mil dados: **19870632 trocas**
- O tempo de **execução (em segundos)**  
Com 100 dados: **0.027981281280517578 segundos**  
Com 10 mil dados: **0.3168046474456787 segundos**  
Com 100 mil dados: **25.960098266601562 segundos**
- O uso da **CPU**  
Com 100 dados: **25.0 %**  
Com 10 mil dados: **38.2 %**  
Com 100 mil dados: **26.6 %**
- O uso de **RAM**  
Com 100 dados: **44.5 %**  
Com 10 mil dados: **45.0 %**  
Com 100 mil dados: **44.7 %**

#### Ordenação **Selection Sort**:

- A quantidade de **comparações**  
Com 100 dados: 100 **comparações**  
Com 10 mil dados: 999 **comparações**  
Com 100 mil dados: 9001

- A quantidade de **trocas**  
Com 100 dados: **95 trocas**  
Com 10 mil dados: **985 trocas**  
Com 100 mil dados: **8986 trocas**
- O tempo de execução (em segundos)  
Com 100 dados: **0.04297351837158203 segundos**  
Com 10 mil dados: **0.30980992317199707 segundos**  
Com 100 mil dados: **21.79980492591858 segundos**
- O uso da **CPU**  
Com 100 dados: **75.0 %**  
Com 10 mil dados: **28.4 %**  
Com 100 mil dados: **25.8 %**
- O uso de **RAM**  
Com 100 dados: **45.4 %**  
Com 10 mil dados: **45.5 %**  
Com 100 mil dados: **45.5 %**

Tabela de relação

Qntd de trocas					
	Bubble Sort1	Bubble Sort2	Bubble Sort3	Insertion Sort	Selection Sort
100	10100	2084	2724	2724	95
10 mil	999000	158109	210749	210749	985
100 mil	81027002	14905230	19870632	19870632	8986
Uso de CPU					
	Bubble Sort1	Bubble Sort2	Bubble Sort3	Insertion Sort	Selection Sort
100	20.0 %	25.0 %	15.8 %	25.0 %	75.0 %
10 mil	25.8 %	45.7 %	30.3 %	38.2 %	28.4 %
100 mil	27.1 %	25.9 %	27.5 %	26.6 %	25.8 %
Uso de RAM					
	Bubble Sort1	Bubble Sort2	Bubble Sort3	Insertion Sort	Selection Sort
100	43.9 %	44.1 %	44.6 %	44.5 %	45.4 %
10 mil	44.2 %	44.1 %	44.4 %	45.0 %	45.5 %
100 mil	44.5 %	44.2 %	44.6 %	44.7 %	45.5 %

Algoritmo com menor quantidade de trocas para:

- **Conjunto com 100 dados:** Selection Sort
- **Conjunto com 10.000 dados:** Selection Sort
- **Conjunto com 100.000 dados:** Selection Sort

Algoritmo com menor uso de CPU para:

- **Conjunto com 100 dados:** Bubble Sort3
- **Conjunto com 10.000 dados:** Bubble Sort1
- **Conjunto com 100.000 dados:** Selection Sort

Algoritmo com menor uso de RAM para:

- **Conjunto com 100 dados:** Bubble Sort2
- **Conjunto com 10.000 dados:** Bubble Sort2
- **Conjunto com 100.000 dados:** Bubble Sort2