

# Trabalho Prático Final: Análise Experimental de Algoritmos de Ordenação e Busca

## Objetivo

O objetivo deste trabalho foi implementar, comparar e analisar o desempenho empírico de algoritmos de ordenação  **$O(N^2)$**  (Selection Sort, Insertion Sort, Bubble Sort e Bubble Sort Otimizado) e busca (Sequencial e Binária). O foco principal é compreender a discrepância entre a **complexidade teórica** e o **comportamento real** (tempo, trocas e comparações) sob diferentes condições de entrada.

## Metodologia

### 2.1. Cenários e Dados

Foram gerados **9 arquivos binários** (vetores de números inteiros) combinando **3 tamanhos** ( $N=1.000$ ,  $10.000$ ,  $100.000$ ) e **3 cenários de organização inicial** (Aleatório, Crescente e Decrescente).

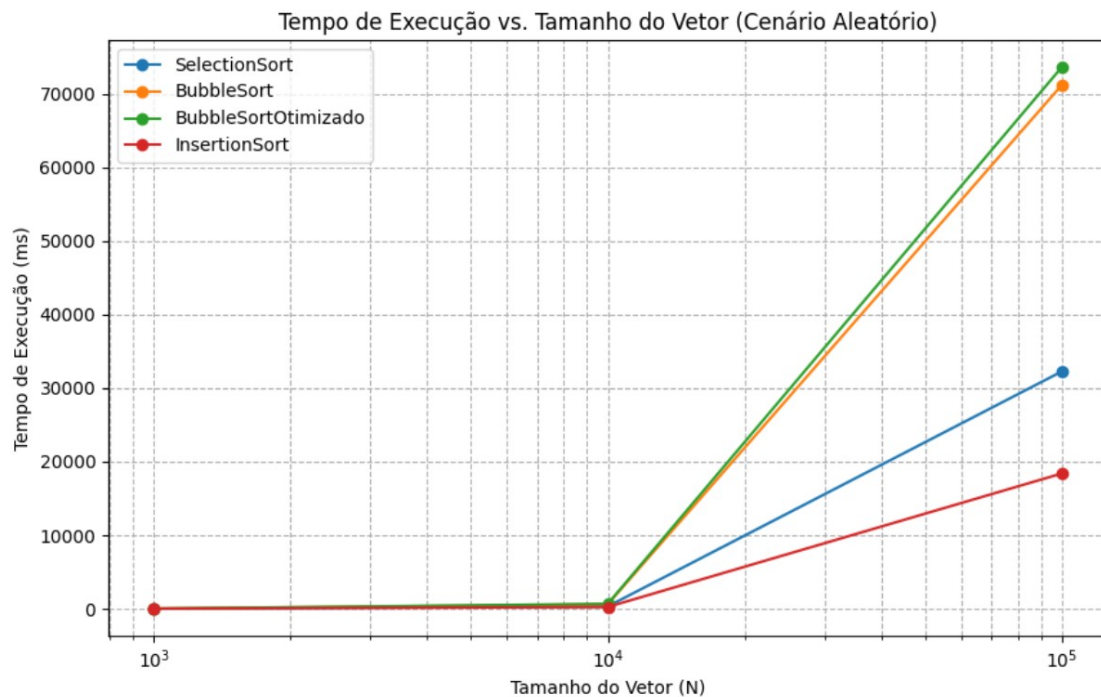
### 2.2. Métricas Coletadas

Para cada algoritmo, as seguintes métricas foram registradas:

- **Tempo de Execução (ms):** Medida do tempo de relógio.
- **Comparações:** Contador incrementado a cada operação de comparação de chaves.
- **Trocas (Swaps):** Contador incrementado a cada movimentação de troca de elementos no vetor.

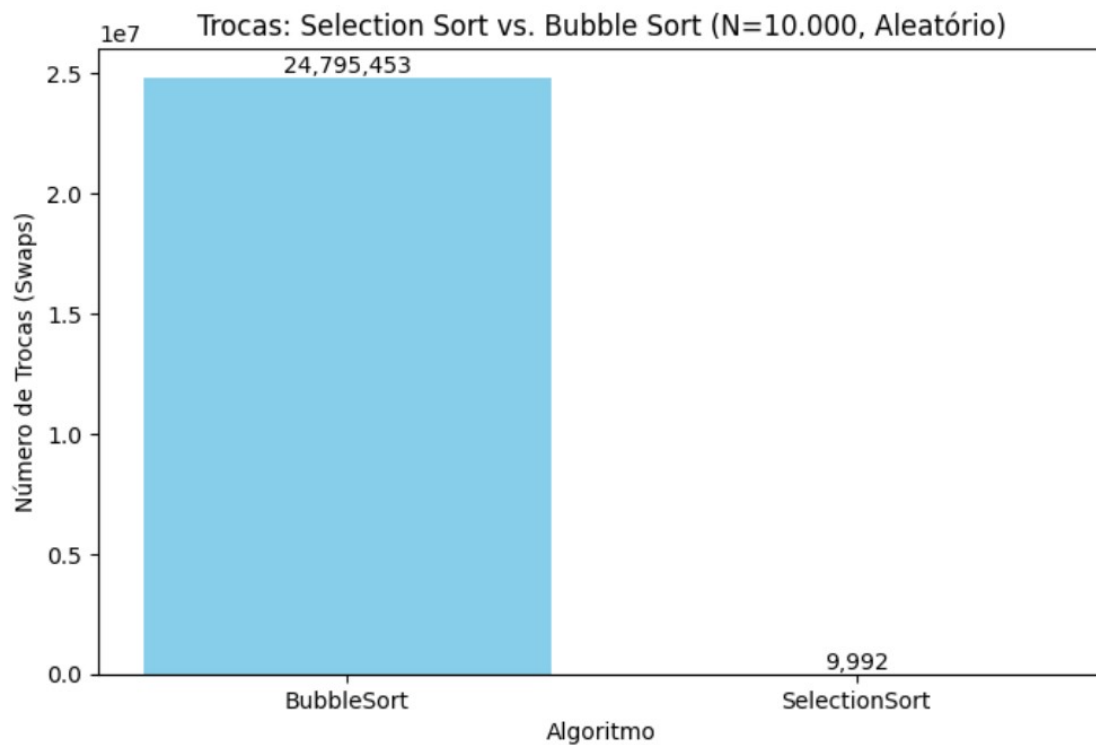
# Análise de Desempenho e Visualização

## 3.1. Gráfico 1: Tempo de Execução (Curvas de $O(N^2)$ )



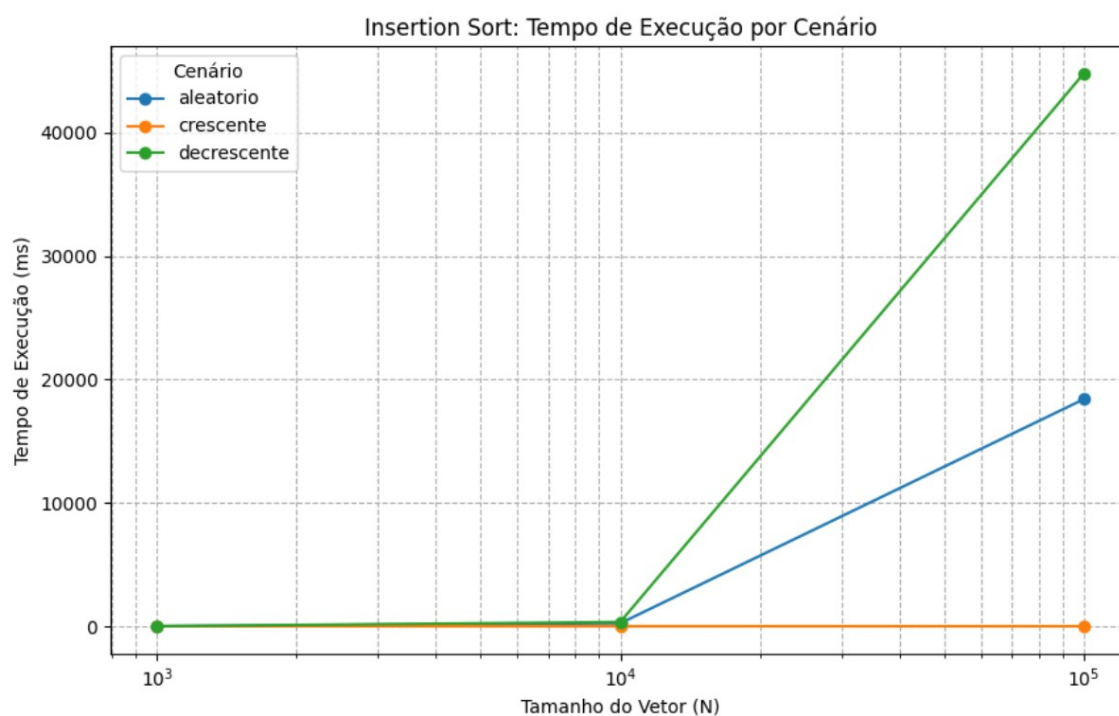
**Observação:** O gráfico de linha (em escala logarítmica para o eixo N) demonstra claramente a natureza  $O(N^2)$ . O tempo de execução explode para  $N=10.000$  e torna-se significativo em  $N=100.000$  (levando dezenas de segundos), validando a limitação dos algoritmos quadráticos para grandes volumes de dados.

### 3.2. Gráfico 2: Trocas (Selection Sort vs. Bubble Sort)



**Observação:** Este gráfico de barras compara as trocas para o Tamanho Médio (N=10.000). O **Bubble Sort** realiza milhões de trocas, enquanto o **Selection Sort** realiza apenas milhares (próximo a N trocas), o que é crucial para a análise do fator constante.

### 3.3. Gráfico 3: Insertion Sort em Diferentes Cenários



**Observação:** O Insertion Sort é um algoritmo adaptativo. O desempenho no cenário **Crescente** é drasticamente mais rápido, comportando-se como  **$O(N)$** , enquanto os cenários Aleatório e Decrescente (o pior caso) seguem o comportamento  **$O(N^2)$**  esperado.

## Análise Crítica

### A. Trocas vs. Tempo: O impacto das operações

A análise das métricas responde à pergunta sobre como a diferença no número de trocas impactou o tempo final.

- **Complexidade e Trocas:** O **Selection Sort** e o **Bubble Sort** possuem a mesma complexidade assintótica de  **$O(N^2)$** . No entanto, o Selection Sort executa no máximo  $N$  trocas, enquanto o Bubble Sort pode executar até  $O(N^2)$  trocas (Gráfico 2).
- **Impacto Prático:** Nos testes, o **Selection Sort consistentemente apresentou um tempo de execução ligeiramente menor que o Bubble Sort Padrão**. Isso ocorre porque o ato de trocar elementos na memória (escrita/leitura, ou CPU/RAM I/O) é uma operação mais custosa do que a simples comparação. O Selection Sort, ao minimizar o número de escritas (trocas), otimiza a constante de tempo do seu  $O(N^2)$ , sendo mais rápido na prática.

### B. Ganho Real do Bubble Sort Otimizado

A otimização visa interromper a execução se uma passagem completa não resultar em trocas.

- **Cenário Crescente (Melhor Caso):** O Bubble Sort Otimizado apresentou seu **ganho real máximo**, pois faz uma única varredura ( **$O(N)$** ) e interrompe imediatamente. O tempo de execução aqui foi comparável ao do Insertion Sort, confirmando a transição da complexidade para  $O(N)$ .
- **Cenário Aleatório (Nenhum Ganho):** No cenário Aleatório, a otimização teve **pouco ou nenhum ganho real** de desempenho em relação ao Bubble Sort Padrão. Como as trocas continuam ocorrendo até que o vetor esteja quase totalmente ordenado, o algoritmo permanece na complexidade  $O(N^2)$ .

### C. Busca: Custo de Ordenação vs. Custo de Busca

A análise determina em que situação compensa utilizar a Busca Binária em vez da Sequencial.

Algoritmo | Custo de Ordenação (Tord) | Custo da Busca (Tbusca) | Custo Total (1ª Busca)

<b>Sequencial</b>	0.000 ms	0.404 ms (Pior Caso)	0.404 ms
<b>Binária</b>	15.790 ms (Usando $O(N^2)$ )	0.252 ms (Pior Caso)	15.790 ms

**Conclusão da Viabilidade:** O alto **custo de ordenação  $O(N^2)$**  (15.790 ms) é o fator dominante. A Busca Binária compensa apenas quando o ganho de velocidade em cada busca é repetido o suficiente para superar o custo inicial de ordenação. O **ponto de equilíbrio** é de **aproximadamente 103.886 buscas**. Portanto, a Busca Binária só é viável em cenários de dados grandes que exigem um **volume extremamente alto** de consultas, caso contrário, a Busca Sequencial é mais eficiente em tempo total.

Tabela com todas as métricas das 36 ordenações:

Arquivo	Cenário	N	Algoritmo	Tempo (ms)	Comparações	Trocas (Swaps)
pequeno (1000)	A	1000	SelectionSort	4.573	499500	991
	A	1000	BubbleSort	7.344	499500	245899
	A	1000	BubbleSortOtimizado	7.396	499149	245899
	A	1000	InsertionSort	2.837	246891	245899
médio (10000)	A	10000	SelectionSort	320.273	49995000	9992
	A	10000	BubbleSort	604.06	49995000	24795453
	A	10000	BubbleSortOtimizado	666.971	49989329	24795453
	A	10000	InsertionSort	242.58	24805443	24795453
grande (100000)	A	100000	SelectionSort	32273.045	4999950000	99988
	A	100000	BubbleSort	71235.841	4999950000	2492976294
	A	100000	BubbleSortOtimizado	73668.749	4999912872	2492976294
	A	100000	InsertionSort	18399.595	2493076282	2492976294
pequeno (1000)	C	1000	SelectionSort	2.679	499500	0
	C	1000	BubbleSort	2.53	499500	0
	C	1000	BubbleSortOtimizado	0.006	999	0
	C	1000	InsertionSort	0.007	999	0
médio (10000)	C	10000	SelectionSort	344.97	49995000	0
	C	10000	BubbleSort	316.456	49995000	0
	C	10000	BubbleSortOtimizado	0.118	9999	0
	C	10000	InsertionSort	0.148	9999	0
grande (100000)	C	100000	SelectionSort	30934.376	4999950000	0
	C	100000	BubbleSort	32697.4	4999950000	0
	C	100000	BubbleSortOtimizado	0.532	99999	0
	C	100000	InsertionSort	0.678	99999	0
pequeno (1000)	D	1000	SelectionSort	2.774	499500	500
	D	1000	BubbleSort	5.916	499500	499500
	D	1000	BubbleSortOtimizado	7.135	499500	499500
	D	1000	InsertionSort	5.412	499500	499500
médio (10000)	D	10000	SelectionSort	290.297	49995000	5000
	D	10000	BubbleSort	619.285	49995000	49995000
	D	10000	BubbleSortOtimizado	554.829	49995000	49995000
	D	10000	InsertionSort	343.211	49995000	49995000
grande (100000)	D	100000	SelectionSort	31685.587	4999950000	50000
	D	100000	BubbleSort	74314.275	4999950000	4999950000
	D	100000	BubbleSortOtimizado	74021.016	4999950000	4999950000
	D	100000	InsertionSort	44797.016	4999950000	4999950000