

Universidade Federal de Mato Grosso  
ICET - Instituto de Ciências Exatas e da Terra  
Departamento de Ciência da Computação  
Graduação em Ciência da Computação

Murilo Becker  
Raissa Cavalcanti

## **Análise de lista estática x lista dinâmica**

Trabalho de Estrutura de  
Dados I

Barra do Garças, 10  
de Março de 2025

Murilo Becker  
Raissa Cavalcanti

**Análise de lista estática x  
lista dinâmica**

Segundo trabalho da disciplina de Estrutura  
de Dados I  
Orientador: Professor Ivairton Monteiro  
Santos

Barra do Garças 10 de  
março de 2025

## **Resumo**

Este documento apresenta uma análise comparativa entre listas estáticas e dinâmicas no contexto de estrutura de dados. O estudo avalia o custo computacional dessas abordagens, considerando o tempo de processamento e a ocupação de memória para diferentes tipos de inserção de elementos.

## **Sumário**

<b>Introdução .....</b>	<b>5</b>
<b>Estrutura.....</b>	<b>5</b>
<b>Fluxo.....</b>	<b>5</b>
<b>Análise dos Resultados.....</b>	<b>6</b>
<b>Conclusão ... .....</b>	<b>7</b>
<b>Conclusão ... .....</b>	<b>8</b>

## 1 Introdução

O objetivo deste trabalho é implementar e analisar o desempenho de uma estrutura de dados estática e duas dinâmicas (simplesmente encadeada e duplamente encadeada). A avaliação é realizada por meio da medição do tempo de inserção e do consumo de memória.

### 1.1 Estrutura

Seguindo sob os requisitos proposto pelo professor, dividimos nosso trabalho nas seguintes etapas: Geração dos dados de entrada, execução dos testes e comparação dos resultados.

### 1.2 Fluxo

A implementação foi realizada na linguagem C, utilizando o ambiente VS Code. O experimento seguiu os seguintes passos:

- No caso da lista estática, foi criada uma estrutura capaz de armazenar 10.000 registros.
- Para as listas dinâmicas, foram utilizadas tanto a abordagem de lista encadeada simples quanto a duplamente encadeada.
- Foram gerados aleatoriamente 10.000 números inteiros, distribuídos da seguinte forma:
  - Aproximadamente 1/3 inserido no início da lista.
  - 1/3 inserido no fim da lista.
  - 1/3 inserido em uma posição aleatória.

Cada teste foi executado 10 vezes para garantir a confiabilidade dos resultados.

Comparação dos Resultados: Registrarmos os resultados obtidos, para cada um dos testes realizados, e por fim realizamos a comparação entre os tipos.

## 2. Análise dos Resultados

Estrutura	Qtd. Ins. Início	Qtd. Ins. Fim	Qtd. Ins. Posição	Menor tempo	Maior tempo	Tempo Médio
Lista estática com capacidade de 10.000	3.333	3.334	3.333	0.120695	0.124173	0,1223343
Lista dinâmica com encadeamen to (simples)	3.333	3.334	3.333	0.312806	0.338701	0,3355867
Lista dinâmica duplamente encadeada	3.333	3.334	3.333	0.407038	0.421731	0,4141668

### **3. Conclusão**

#### **A respeito da Lista Estática:**

A implementação envolveu sete funções principais, incluindo a inicialização da estrutura, verificação de capacidade, inserção em diferentes posições e impressão dos elementos. A principal vantagem observada foi a velocidade, já que o acesso a um vetor é mais eficiente que manipulações de ponteiros.

#### **A respeito das Listas Encadeadas:**

##### **Lista Encadeada Simples**

Essa implementação incluiu cinco funções principais, responsáveis pela inicialização da lista, inserção (no início, fim e posição aleatória) e impressão. Embora o consumo de memória tenha sido maior, a flexibilidade na manipulação dos elementos é uma vantagem.

##### **Lista Duplamente Encadeada**

A implementação seguiu a mesma estrutura da lista encadeada simples, mas incluiu ponteiros para os nós anteriores. Isso possibilita percorrer a lista em ambas as direções, tornando algumas operações mais eficientes. No entanto, o custo de memória foi ainda maior.

## Conclusões Finais:

A análise comparativa mostrou que:

- A lista estática é a mais eficiente em termos de velocidade.
- A lista encadeada simples tem melhor uso de memória em relação à duplamente encadeada.
- A lista duplamente encadeada permite percorrer em ambas as direções, tornando-se mais versátil, mas com maior consumo de memória.

## Medida do Consumo de Memória

Para avaliar a ocupação da memória, utilizamos a ferramenta Valgrind, que forneceu os seguintes resultados:

<b>Lista Estática</b>	<b>42kb</b>
<b>Lista Encadeada Simples</b>	<b>162kb</b>
<b>Lista Duplamente Encadeada</b>	<b>242kb</b>

Dessa forma, a escolha da estrutura de dados mais adequada depende do contexto de uso. Se a prioridade for eficiência e menor consumo de memória, a lista estática é a melhor opção. Por outro lado, se a aplicação exige flexibilidade na manipulação de dados, as listas dinâmicas são mais indicadas.