

# Trabalho de AEDS: List

Por: David Francisco Cordeiro Neto, João Pedro de Almeida Dupim e Pedro Henrique Santos

# Sumário

01

**Introdução**

02

**Pontos  
Principais**

03

**Como Utilizar**

04

**Exemplo  
Prático**



# 01

---

## Introdução ao List

A biblioteca `<list>` é um componente fundamental da linguagem de programação C++, oferecendo uma estrutura de dados flexível e poderosa para manipulação de listas duplamente encadeadas.

Através da classe `list`, essa biblioteca fornece um conjunto de funcionalidades eficientes para inserir, remover e acessar elementos em uma lista, garantindo operações de tempo constante para inserção e remoção no início ou no final da lista, bem como iteração eficiente através de seus elementos.





---

02

Pontos Principais

A `<list>` é uma estrutura de dados valiosa em C++, proporcionando uma gama de operações eficientes e flexíveis para manipulação de coleções de elementos, especialmente quando a necessidade de inserção e remoção dinâmica de dados é prioritária.

Embora as listas encadeadas possam consumir mais memória do que arrays, a `<list>` do C++ oferece um desempenho estável para inserção e remoção, tornando-a uma escolha vantajosa para operações frequentes de modificação, onde o tempo constante é fundamental.

A `<list>` implementa uma lista duplamente encadeada, permitindo a inserção e remoção eficientes de elementos no início, final ou em posições específicas da lista. Isso resulta em operações rápidas de inserção e remoção, mesmo para grandes conjuntos de dados.



# Principais Operações oferecidos pela biblioteca

## **push\_back** (value)

Adiciona um elemento no **final** da lista.

## **push\_front** (value)

Adiciona um elemento no **início** da lista.

## **erase** (position)

Remove o elemento na posição especificada.

## **clear**

Remove todos os elementos da lista.

## **sort**

Ordena os elementos na lista.

## **reverse**

Inverte a ordem dos elementos na lista.

# 03

---

## Como Utilizar



---

Para utilizar a biblioteca `<list>` em C++, comece incluindo o cabeçalho `<list>`, declare uma lista usando `list<Tipo> nomeDaLista;`, onde “Tipo” representa o tipo de dado a ser armazenado. Após a declaração, utilize os métodos fornecidos pela lista, como `push_back`, `push_front`, `pop_back`, `pop_front`, `insert` e `erase`, para adicionar, remover e manipular elementos na lista.

```
#include <iostream>
#include <list>
```

A `<list>` fornece uma maneira versátil e eficiente de gerenciar coleções de elementos dinâmicos, dessa forma os métodos `front`, `back`, `begin`, `end`, `size` e `empty` permitem acessar informações sobre os elementos e o estado atual da lista.

---

# 04

---

## Exemplo Prático