

# **Relatório implementação de fila - Trabalho Avaliativo**

**Alunos:**

**Eduardo Lemes CT3037568**

**Guilherme Batista de Souza CT3037274**

- **Qual estrutura foi escolhida**

A estrutura escolhida para a implementação foi uma Fila de Prioridade usando uma Lista Encadeada. A fila de prioridade é um tipo de estrutura de dados onde os elementos são ordenados de acordo com uma chave de prioridade.

- **Onde essa estrutura é útil (aplicações reais)**

Sistemas de Gerenciamento de Processos (Escalonamento de CPU): A fila de prioridade é usada para gerenciar os processos em um sistema operacional, onde os processos com maior prioridade são executados primeiro.

Sistemas de Atendimento ao Cliente no Banco Físico: A fila de prioridade é utilizada para organizar o atendimento aos clientes em uma agência bancária, onde clientes com necessidades especiais, como idosos ou gestantes, são atendidos primeiro. Isso garante que aqueles com maior urgência ou vulnerabilidade sejam priorizados, enquanto os demais clientes podem aguardar por mais tempo.

- **Explicação breve das funções implementadas**

Função `inserir(Noh** inicio, int dado, int prioridade)`:

Esta função insere um novo elemento na fila de prioridade. O nó é inserido de forma ordenada com base na sua prioridade, de modo que os elementos com maiores prioridades (menor valor de prioridade) fiquem à frente na lista.

O nó é inserido no início caso a lista esteja vazia ou a prioridade seja a maior. Caso contrário, ele é inserido na posição correta em relação à prioridade, mantendo a ordem crescente de prioridade.

Função `remover(Noh** inicio)`:

Esta função remove o nó da fila de prioridade com a maior prioridade, ou seja, o nó com o menor valor de prioridade.

A remoção é feita encontrando o nó de maior prioridade e ajustando os

ponteiros para desconectar esse nó da lista encadeada. O valor do nó removido é retornado.

Função imprimirLista(Noh\* inicio):

Esta função percorre a lista encadeada e imprime os dados e prioridades de todos os nós da fila de prioridade.

Ela é útil para visualizar o estado da fila após cada operação de inserção ou remoção.

- **Principais dificuldades encontradas**

A principal dificuldade na implementação foi a remoção dos elementos, pois era necessário identificar e localizar o nó com a maior prioridade. Isso envolvia percorrer a lista e comparar as prioridades de cada nó, o que exigiu cuidado para manter a integridade da estrutura durante a remoção. Foi preciso ajustar os ponteiros corretamente para evitar a perda de dados. Além disso, era importante garantir que a lista continuasse ordenada após a remoção e que a memória fosse liberada corretamente para evitar vazamentos.