

Sistema de controle de carregamento portuário

Contextualização do problema

Um porto recebe por dia uma quantidade muito grande de containers, e estes são empilhados um sob o outro. Além disso, existe também uma fila de navios aguardando a oportunidade de ancoragem, e por sua vez de carregamento. Cada navio suporta um número X de toneladas, container pode ser: i) grande (80 toneladas); ii) médio (50 toneladas); ou iii) pequeno (35 toneladas).

Deve-se desenvolver uma solução para o gerenciamento do porto utilizando os **CONCEITOS de pilha e fila**. Portanto, deve-se desenvolver uma solução que contenha uma TAD para fila (navios) e uma TAD para pilha (containers).

- Estruturas:
 - Container
 - Tamanho
 - Tipo do Produto
 - É perecível?
 - É corrosivo?
 - Navio
 - Capacidade em toneladas *//máximo de 125 mínimo de 25 deve-se usar um rand() para preenchimento obedecendo o intervalo acima.*
 - Suporta cargas perecíveis?
 - Suporta cargas corrosivas? *//se suporta corrosivo não pode carregar perecível e vice-versa.*
 - Containers * (inserir o(s) container(s) desde navio)

A aplicação deverá realizar as seguintes funcionalidades

0 – Criar estruturas

1 – Cadastro

- Container
- Navio

2 – Remover

- Container
- Navio

3 – Impressão

- Todos os containers
- Todos os navios
- Containers de um determinado navio

4 – Carregamento

- Nessa etapa será feita o carregamento dos navios, entretanto, alguns fatores devem ser observados:
 - O tamanho do container;

- A capacidade do navio;
- A prioridade para o carregamento é dos **containers**, uma vez que, podem existir produtos perecíveis. Algumas questões pertinentes:
 - Se a prioridade de saída da pilha é um container que não seja perecível?
 - O que fazer para não perder os containers que foram desempilhados e não foram embarcados?
 - As limitações de carregamento de navios e containers não podem ser ignoradas.
 - As cargas perecíveis não podem ir junto com cargas corrosivas.
 - Nem todos os navios estão aptos ao carregamento de cargas perecíveis e corrosíveis.
 - Portanto, deve-se respeitar as limitações impostas. Entretanto, o usuário deverá sempre ser informado da situação para que o carregamento não seja suspenso por motivos de indisponibilidade de navios ou containers.
- A medida em que, os containers forem sendo embarcados as **posições de memória associadas no carregamento, isto é, tanto container quanto navios**, deveram ser liberadas.
- Quando a pilha de containers acabar ou a fila de navios se esvaziar, o carregamento deverá ser suspenso e a função 3 deverá ser executada imediatamente, apenas com a(s) opção(es) que ainda restar(em), isto é, navios ou containers.

5 – Score

Ao final do carregamento deve-se ter um **score geral**, este por sua vez, será computado da seguinte maneira:

- Para cada tonelada carregada + 2 pontos.
- Para cada espaço livre nos navios, - 0,5 pontos.
- Outras penalidades
 - Desempilhar: -5% (convertidos em pontos) de acordo com o tamanho do container;
 - Desenfileirar: -5% (convertidos em pontos) de acordo com a capacidade do Navio;
- **As propriedades de Fila e Pilha devem ser respeitadas nessa aplicação.**

6 – Liberar memória (SAIR)