

# Trabalho 1 de Estrutura de dados: Simulação de Porto

---

Trabalho realizado por Gustavo Teixeira Magalhães. Esse documento se refere à como usar o makefile e compilar o arquivo de forma mais eficiente.

## Tutorial Make:

---

Para compilar e rodar o programa com o makefile, entre na linha de comando (Bash do linux ou o interpretador do Cygwin) e digite os seguintes comandos:

- `make` para apenas compilar o trabalho.
- `make run` para compilar e rodar o trabalho.
- `make clean` para limpar arquivos previamente compilados.
- `make cleanmake` para recompilar o arquivo, removendo a compilação anterior.
- `make cleanrun` para recompilar, limpar e rodar a nova compilação.

## Estrutura dos Arquivos:

---

Os arquivos código fonte estão na pasta `src`, o arquivo compilado fica na pasta `obj` e as documentações (fora este README) ficam na pasta `DOCS`.

## Sobre a implementação

---

O trabalho visa a implementação de um simulador portuário por meio de estruturas clássicas de dados (pilha e fila), visando o entendimento e prática no uso dessas estruturas.

## Estruturas Escolhidas

---

Para controlar as operações do porto, resolvi utilizar a implementação dinâmica das estruturas pilha e fila, devido à facilidade da implementação, eficiência de memória e limpeza de código. As estruturas são, especificamente:

- Uma fila de atracamento (que contém os navios);
- Quatro pilhas de containers por navio;
- Cinco pilhas de containeres pra travessa (que contém os containers descarregados pelos navios);

As imagens abaixo esclarecem melhor as estruturas:

□

*Anexo1: Estruturas da pilha e fila*

□

*Anexo2: Estrutura da travessa e suas pilhas de container*

## Outras Estruturas

---

Além das estruturas encapsuladas nos processos e objetos principais, também temos outra estrutura separada, para manutenção e controle de tempo:

□

## Contadores de tempo

---

Como pode ser observado na imagem acima, existem duas variáveis que controlam o tempo de execução da simulação:

- **dia:** Unidade de tempo que conta quantas iterações da simulação ocorreram até agora.
- **UT:** Unidade de tempo que conta cada ação que é feita (atracar, descarregar, armazenar, enfileirar).

## A simulação

---

O usuário entra no programa e lhe é perguntado quantos dias do porto ele quer simular, ou seja, quantas iterações da simulação ele quer ver. Após a decisão do usuário, o programa procede com a execução e segue um algoritmo padrão, reportando um relatório no final de cada dia. Esses processos são, segundo a ordem de execução:

### Atracar

---

No começo de cada dia, de 1 a 3 navios são enfileirados na fila de atracamento. O numero que é enfileirado é escolhido aleatoriamente.

Ao enfileirar um navio, ele é criado, recebe um id (baseado na quantidade de navios na pilha), suas pilhas de containers são empilhadas (número aleatório entre 4 e 16 containeres, mínimo de um por pilha) e por fim, o navio é colocado na fila.

### Desembarcar

---

Após atracar todos os navios do dia, começa o processo de desembarque, no qual todos os navios na fila descarregam seus containeres na travessa. Esse é o processo:

Um navio é desenfileirado, suas pilhas são desempilhadas, e seus containers são empilhados nas pilhas da travessa.

### Armazenar

---

Os containers nas pilhas da travessa são levadas por guias e encaminhadas para armazenamento (abstração). Na prática, as pilhas da travessa apenas são desempilhadas.

### Relatório

---

No final do dia é feito um relatório, que mostra a fila de navios, as travessas e o tempo decorrido (dias e UT).

### Reset

---

Se a fila estiver vazia, ela é resetada, para evitar id's muito grandes dos navios.

## Referências e Tecnologias

---

- Todas as estruturas de dado foram feitas usando como base os slides do professor Nivio Ziviani.
- As imagens foram feitas usando o microsoft paint.
- O programa foi codificado na linguagem C, usando o compilador gcc por meio do WSL (Windows Subsystems for Linux).
- Foi também criado, um *makefile* para compilação mais simplificada.
- Foram usados git e github para o controle de versão e repositório do projeto.

