

Universidade Federal do Espírito Santo – Departamento de Informática
Estruturas de Dados (INF09292)
1º Trabalho Prático
Período: 2011/1
Profª Patrícia Dockhorn Costa
Email: pdcosta@inf.ufes.br

Data de Entrega: 18/05/2011

Grupos de 2 pessoas

Este trabalho tem como objetivo praticar o uso de tipos abstratos de dados e estruturas do tipo Lista.

Regras Importantes

- Não é tolerado plágio. Trabalhos copiados serão penalizados com zero.
- A data de entrega é inadiável. Para cada dia de atraso, é retirado um ponto da nota do trabalho.

Material a entregar

- Impresso: Documentação do trabalho, que deve conter:
 - Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa (em termos de módulos, arquivos, etc.).
 - Implementação: descrição da implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas (de preferência **com diagramas ilustrativos**), o funcionamento das principais funções utilizadas incluindo pré e pós condições, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Modularize o seu programa como discutido em sala de aula.
 - Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
 - Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.
- Por email (pdcosta@inf.ufes.br):
 - O assunto da mensagem deve ser ed201101CC:trab1:<nome1>:<nome2>
 - Por exemplo: ed201101CC:trab1:<joaosilva>:<mariacosta>
 - Documentação do trabalho (em formato PDF).
 - Todos os arquivos .c e .h criados (exigido código muito bem documentado!).
 - O makefile.
 - Favor nomear os arquivos da seguinte maneira: **containers.h**, **containers.c**, **navios.h**, **navios.c**, **porto.h**, **porto.c** e **simulaportos.c**.

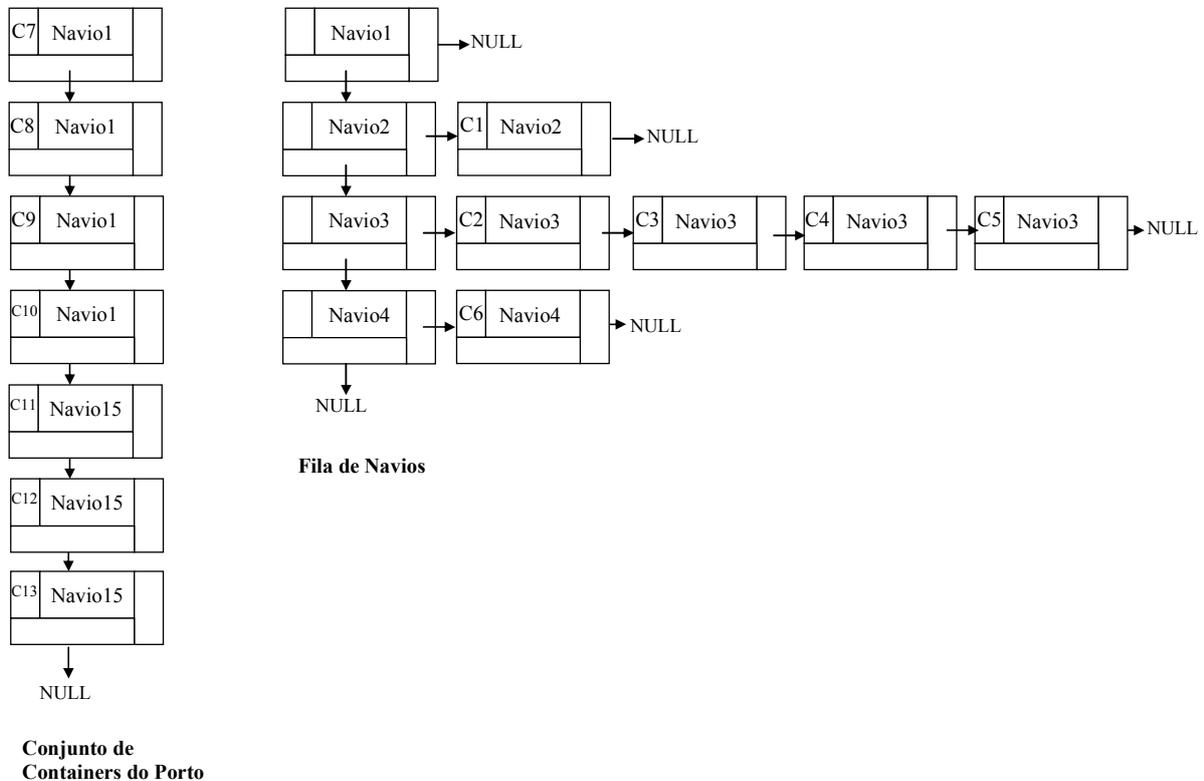
Sistema Portuário

Um sistema portuário tem a função de gerenciar as atividades realizadas em um porto. Neste porto, containers podem chegar por via terrestre ou via marítma. Quando chegam por via terrestre, eles são adicionados ao conjunto de containers do porto, que ficam disponíveis para serem carregados em navios. Containers chegam por via marítma através da chegada de navios, a serem descarregados. À medida que chegam, os navios entram em uma fila de navios e esperam para serem *processados*. Processar um navio significa *carregá-lo*, no caso do navio estar vazio, ou *esvaziá-lo*, caso o navio tenha algum container. Navios são carregados com os containers disponíveis no conjunto de containers do porto (esses containers são designados para um determinado navio). Os containers retirados de um navio são imediatamente retirados do

porto. Assim que os navios forem processados, eles são desconsiderados pelo sistema portuário. A figura a seguir ilustra o funcionamento do porto.



No porto mostrado na figura acima, há 3 navios na fila esperando para serem processados, um conjunto de containers disponíveis para serem carregados em navios e alguns caminhões e trens carregando o porto com containers. Esse sistema portuário pode ser implementado com um conjunto de listas encadeadas: basicamente, implementa-se uma lista de Navios, na qual cada célula contém o nome do navio (identificador único), bem como uma lista de containers que este navio eventualmente estiver transportando. Uma outra lista pode ser utilizada para armazenar o conjunto de containers do porto, na qual cada célula representa um container, que deve especificar o identificador único do container, e o identificador do navio para o qual ele será carregado. Isso pode ser observado na seguinte figura:



Como pode ser visto na figura, no conjunto de containers há 4 containers (08, 09, 10 e 11) destinados para o navio 1 (identificador único “navio1”). Desta forma, quando o navio 1 for processado, estes containers serão carregados neste navio. Os outros containers se referem aos navios 15 e 16, que ainda não chegaram ao porto. Quando for a vez do navio 2, o container c2 será descarregado no navio (célula retirada) e este navio sairá da lista de navios.

Nesse trabalho, você deverá implementar essa estrutura. No site da disciplina pode ser encontrada a especificação do TAD (porto.h), que define os seguintes tipos opacos e operações:

```
//Tipo que define um porto
typedef struct porto Porto;
//Tipo que define um navio
typedef struct navio Navio;
//Tipo que define um container
typedef struct container Container;

/*Cria um novo porto com um identificador id (único no sistema)*/
Porto* CriaPorto(char* id);

/*Cria um novo navio com um identificador id. Este navio ainda nao esta' em nenhum
porto */
Navio* CriaNavio(Porto* p, char* id);

/*Insere um container em um conjunto de containers disponiveis em um porto. Vale
ressaltar que esta funcao nao carrega o container em navio */
Container* InsererContainerPorto (Porto* p, char* id_navio, char* id_container);

/*Retira um container de um determinado porto (que nao esteja carregado em navio)*/
Container* RetiraContainerPorto (Porto* p, char* id_container);

/*Anuncia a chegada de um navio (carregado ou vazio) em um dado porto. O navio vai para
a ultima posicao da fila */
void ChegaNavio (Porto* p, Navio* n);

/*Anuncia a chegada de um navio prioritario (carregado ou vazio) em um dado porto. O
navio vai para a primeira posicao da fila */
void ChegaNavioPrioritario (Porto* p, Navio* n);

/*Funcao que processa o primeiro navio da lista de navios do porto. Processar significa
carregar (caso o navio esteja vazio) ou descarregar (caso o navio esteja cheio). Assim
que o navio for processado, ele deve deixar o porto (ou seja, deve ser excluido da
lista de navios). A função retorna o navio retirado, com ou sem containers */
Navio* ProcessaNavio(Porto* p);

/*Imprime todos os containers carregados e um determinado navio, no seguinte formato
(para o exemplo da figura, considere que pediu-se para imprimir informações do navio
3):

Navio: navio3
Containers: C2, C3, C4, C5 */
void ImprimeContainersNavio (Porto* p, Navio* n);

/*Imprime todos os containers disponiveis em um porto (ou seja, nao carregados em
navios). Para o exemplo da figura, deve-se imprimir os containers, no seguinte formato:

Containers do Porto vitoria: C7(navio1), C8(navio1), C9(navio1), C10(navio1),
C11(navio15), C12(navio15), C13(navio15) */
void ImprimeContainersPorto (Porto* p);
```

```

/*Imprime todos os navios de um determinado porto (e seus containers), no seguinte
formato:

Navio: navio1
Containers:

Navio: navio2
Containers: C1

Navio: navio3
Containers: C2, C3, C4, C5

Navio: navio4
Containers: C6 */
void ImprimeNaviosPorto (Porto* p);

/*Imprime todas as informacoes referentes a um porto (lista de navios e containers,
lista de containers disponiveis), no seguinte formato:

Navios do Porto vitoria:
Navio: navio1
Containers:

Navio: navio2
Containers: C1

Navio: navio3
Containers: C2, C3, C4, C5

Navio: navio4
Containers: C6

Containers do Porto vitoria: C7(navio1), C8(navio1), C9(navio1), C10(navio1),
C11(navio15), C12(navio15), C13(navio15)*;
void ImprimePorto (Porto* p);

/* Exclui toda a estrutura do porto, liberando a memoria alocada para todas as
estruturas de dados utilizadas */
void ExcluirPorto (Porto* p);

```

Dicas para implementação do arquivo porto.c (implementação do TAD porto.h)

A estrutura porto deverá conter um campo com o nome do porto (que o identifica unicamente), e campos que definem as listas de containers e navios. Seria algo do tipo (apenas exemplo):

```

struct porto{
    char* id;
    lista_containers* containers;
    lista_navios* navios;
}

```

Para implementar estas listas, são necessários dois TAD's (a serem importados no arquivo porto.c): uma lista de containers (containers.h, containers.c) e uma lista de navios (navios.h, navios.c). Estes tads não serão fornecidos pela professora, portanto, o aluno deve definir os tipos e as operações a serem especificados por estes tads. Notem que um item da lista de navios contém um ponteiro para uma lista de containers. Isso significa que o Tad navios deve incluir o Tad containers.

O Programa Testador (simulaportos.c)

O programa testador deverá ser capaz de ler as instruções do arquivo texto de entrada e realizar as devidas operações no tad porto. O programa testador precisa manter uma lista de portos, que poderá ser definida por um vetor de ponteiros para portos, visto que no máximo 20 portos serão incluídos no sistema portuário.

O seu programa (**simulaportos**) deverá ler os dados de entrada a partir de um arquivo, cujo nome é passado como parâmetro na linha de comando (faz parte do trabalho descobrir como manipular arquivos e strings em C). Exemplo de execução do programa a partir da linha de comando:

```
simulaportos entrada.txt
```

O arquivo de entrada é basicamente uma lista de comandos (um por linha) em formato texto. O último comando é a palavra FIM, que indica o final do arquivo. O formato a ser usado é exemplificado abaixo:

Exemplo de arquivo de entrada

```
CRIAPORTO vitoria
CRIAPORTO roterda
CRIAPORTO santos
INLUICONTAINERPORTO vitoria c1 n1
INLUICONTAINERPORTO vitoria c2 n1
INLUICONTAINERPORTO vitoria c3 n1
INLUICONTAINERPORTO vitoria c4 n2
INLUICONTAINERPORTO vitoria c5 n3
INLUICONTAINERPORTO vitoria c6 n2
INLUICONTAINERPORTO roterda c7 n1
INLUICONTAINERPORTO roterda c8 n2
INLUICONTAINERPORTO roterda c9 n3
INLUICONTAINERPORTO roterda c10 n4
INLUICONTAINERPORTO santos c11 n4
INLUICONTAINERPORTO santos c12 n4
INLUICONTAINERPORTO santos c13 n5
INLUICONTAINERPORTO santos c14 n6
EXLUICONTAINERPORTO santos c14
IMPRIMECONTAINERSPORTO santos
IMPRIMECONTAINERSPORTO vitoria
IMPRIMECONTAINERSPORTO roterda
CRIANAVIO n1 vitoria
CRIANAVIO n2 vitoria
CRIANAVIO n3 roterda
CRIANAVIO n4 santos
CRIANAVIO n5 santos
CRIANAVIO n6 santos
CRIANAVIO n7 roterda
CRIANAVIO n8 santos
CRIANAVIO n9 vitoria
IMPRIMENAVIOSPORTO vitoria
IMPRIMENAVIOSPORTO santos
IMPRIMENAVIOSPORTO roterda
```

```
TRANSPORTA vitoria roterda
TRANSPORTA roterda vitoria
TRANSPORTA santos vitoria
TRANSPORTA vitoria santos
INCLUICONTAINERPORTO vitoria c15 n8
INCLUICONTAINERPORTO santos c16 n8
INCLUICONTAINERPORTO santos c17 n8
TRANSPORTA vitoria santos
TRANSPORTA vitoria santos
TRANSPORTAPRIORITARIO vitoria santos
TRANSPORTA santos vitoria
TRANSPORTAPRIORITARIO roterda vitoria
TRANSPORTA santos roterda
IMPRIMEPORTO vitoria
IMPRIMEPORTO roterda
IMPRIMEPORTO santos
EXCLUIPORTO vitoria
EXCLUIPORTO santos
EXCLUIPORTO roterda
FIM
```

É garantido que o arquivo de entrada não inclui dados inconsistentes (identificadores incorretos, duplicados, etc.)

Os comandos de impressão de dados (IMPRIMECONTAINERSPORTO, IMPRIMENAVIOSPORTO e IMPRIMEPORTO), imprimem os dados com o formato especificado no tad porto.

Os comandos CRIAPORTO, INCLUICONTAINERPORTO, EXCLUICONTAINERPORTO, CRIANAVIO e EXCLUIPORTO possuem uma correspondência direta com as funções do tad porto. Portanto, para interpretá-los, vide documentação do arquivo porto.h.

O comando (TRANSPORTA porto1 porto2) faz com que seja processado o primeiro navio da fila de navios do porto1 e que este chegue no porto2 e entre na última posição da fila de navios.

O comando (TRANSPORTAPRIORITARIO porto1 porto2) faz com que seja processado o primeiro navio da fila de navios do porto1 e que este chegue no porto2 e entre na primeira posição da fila de navios.

Considerando o arquivo de entrada dado acima, espera-se o seguinte no arquivo de saída:

Arquivo de saída para o arquivo entrada.txt

```
Containers do Porto santos: C11(navio4), C12(navio4), C13(navio5)

Containers do Porto vitoria: C1(navio1), C2(navio1), C3(navio1),
c4(navio2), c5(navio3), c6(navio2)

Containers do Porto roterda: C7(navio1), C8(navio2), C9(navio3),
c10(navio4)

Navios do Porto vitoria
Navio: navio1
Containers:
```

Navio: navio2
Containers:
Navio: navio9
Containers:

Navios do Porto santos
Navio: navio4
Containers:
Navio: navio5
Containers:
Navio: navio6
Containers:
Navio: navio8
Containers:

Navios do Porto roterda
Navio: navio3
Containers:
Navio: navio7
Containers:

Navios do Porto vitoria
Navio: navio7
Containers:
Navio: navio4
Containers:

Containers do Porto vitoria: C5(navio3)

Navios do Porto roterda
Navio: navio1
Containers: c1, c2, c3
Navio: navio5
Containers: c13

Containers do Porto roterda: C7(navio1), c8(navio2), c10(navio4)

Navios do Porto santos
Navio: navio6
Containers:
Navio: navio8
Containers:
Navio: navio2
Containers: c4, c6
Navio: navio9
Containers:
Navio: navio3
Containers:

Containers do Porto santos: C16(navio8), c17(navio8)

BOM TRABALHO!