



Universidade Estadual do Norte Fluminense Darcy Ribeiro

Curso: Ciência de Computação

Período: 3º **Turno:** Diurno

Disciplina: Estrutura de Dados

Professor: Fermín Alfredo Tang Montané

Atividade: Avaliação AM1

Data: 25/05/2021

Aluno: João Vítor Fernandes Dias

Matrícula: 00119110377

1. [1,5 Pontos] ilustre com um exemplo (diferente daquele usado em aula) a ideia do algoritmo que permite converter uma expressão infixa em posfixa usando uma pilha.

R:

<div>1</div> <div><div>INFIXA</div><div>A+B*C-D/E</div></div> <div><div>PILHA</div><div></div></div> <div><div>POSFIXA</div><div></div></div>	<div>2</div> <div><div>INFIXA</div><div>B*C-D/E</div></div> <div><div>PILHA</div><div>+</div></div> <div><div>POSFIXA</div><div>A</div></div>
<div>3</div> <div><div>INFIXA</div><div>C-D/E</div></div> <div><div>PILHA</div><div>*</div><div>+</div></div> <div><div>POSFIXA</div><div>AB</div></div>	<div>3</div> <div><div>INFIXA</div><div>.-D/E</div></div> <div><div>PILHA</div><div>*</div><div>+</div></div> <div><div>POSFIXA</div><div>ABC</div></div>
<div>5</div> <div><div>INFIXA</div><div>/E</div></div> <div><div>PILHA</div><div>-</div></div> <div><div>POSFIXA</div><div>ABC*+D</div></div>	<div>6</div> <div><div>INFIXA</div><div>E</div></div> <div><div>PILHA</div><div>/</div><div>-</div></div> <div><div>POSFIXA</div><div>ABC*+D</div></div>
<div>7</div> <div><div>INFIXA</div><div></div></div> <div><div>PILHA</div><div>/</div><div>-</div></div> <div><div>POSFIXA</div><div>ABC*+DE</div></div>	<div>8</div> <div><div>INFIXA</div><div>A+B*C-D/E</div></div> <div><div>PILHA</div><div></div></div> <div><div>POSFIXA</div><div>ABC*+DE/-</div></div>

2. **[2,5 Pontos]** duas pilhas são iguais se possuem os mesmos elementos, exatamente na mesma ordem. Considere que as pilhas P1 e P2 já existem e são passadas como parâmetro. Considere que as pilhas possuem elementos do tipo inteiro. Responda os seguintes pontos:

- i) **[0,8 Ponto]** descreva em palavras a ideia de um algoritmo para testar se duas pilhas P1 e P2 são iguais, usando operações do TAD Pilha, e preservando as pilhas originais;

R:

Para testarmos se duas pilhas (A e B) são iguais, primeiro iremos assumir que eles são iguais criando uma variável “SaoIguais” (que definirá ao final o estado da pilha, podendo ele ser 0 – Diferente, ou 1 – Igual) e definiremos seu valor inicial como 1. Também iremos criar uma variável temporária do tipo inteiro chamada “Temp” (que guardará temporariamente os elementos que estiverem sendo movidos) e uma pilha auxiliar chamada “Aux” (que armazenará temporariamente os elementos iguais em ambas as pilhas);

Começaremos comparando os contadores de cada pilha. Se os contadores forem diferentes, definiremos o valor de SaoIguais como 0, pois elas não são iguais.

Enquanto as duas pilhas não estiverem vazias e elas não estejam diferentes até então, repete-se o seguinte passo:

Compararemos o topo da pilha de ambos, se forem diferentes, a variável SaoIguais obtém o valor 0, se não forem diferentes, remove-se o elemento do topo de cada pilha e insere esse valor em uma pilha auxiliar;

Terminada essa repetição, outra repetição começa:

Enquanto a pilha auxiliar não estiver vazia, remove-se o elemento de seu topo e o insere tanto na pilha A quanto na pilha B.

Por fim, analisaremos o valor de “SaoIguais”. Se “SaoIguais” for igual a 1 será impresso na tela a mensagem “Sao iguais”, se “SaoIguais” não for igual a 1, será impresso na tela a mensagem “Sao diferentes”.

- ii) **[1,0 Ponto]** escreva o algoritmo correspondente;

R:

```
void SaoIguais(*Pilha A, *Pilha B){
    int Temp;
    int SaoIguais=1;
    Pilha* Aux;
    CriaPilha(Aux);

    se (ContadorPilha(A) != ContadorPilha(B)){
        SaoIguais=0;
    }

    enquanto (!(PilhaVazia(A)||PilhaVazia(B))&&SaoIguais){
        se (TopoPilha(A)!=TopoPilha(B)) {
            SaoIguais=0;
        } senao {
            Temp=RemovePilha(A);
            Temp=RemovePilha(B);
            InserirPilha(Aux,Temp);
        }
    }
```

```

}

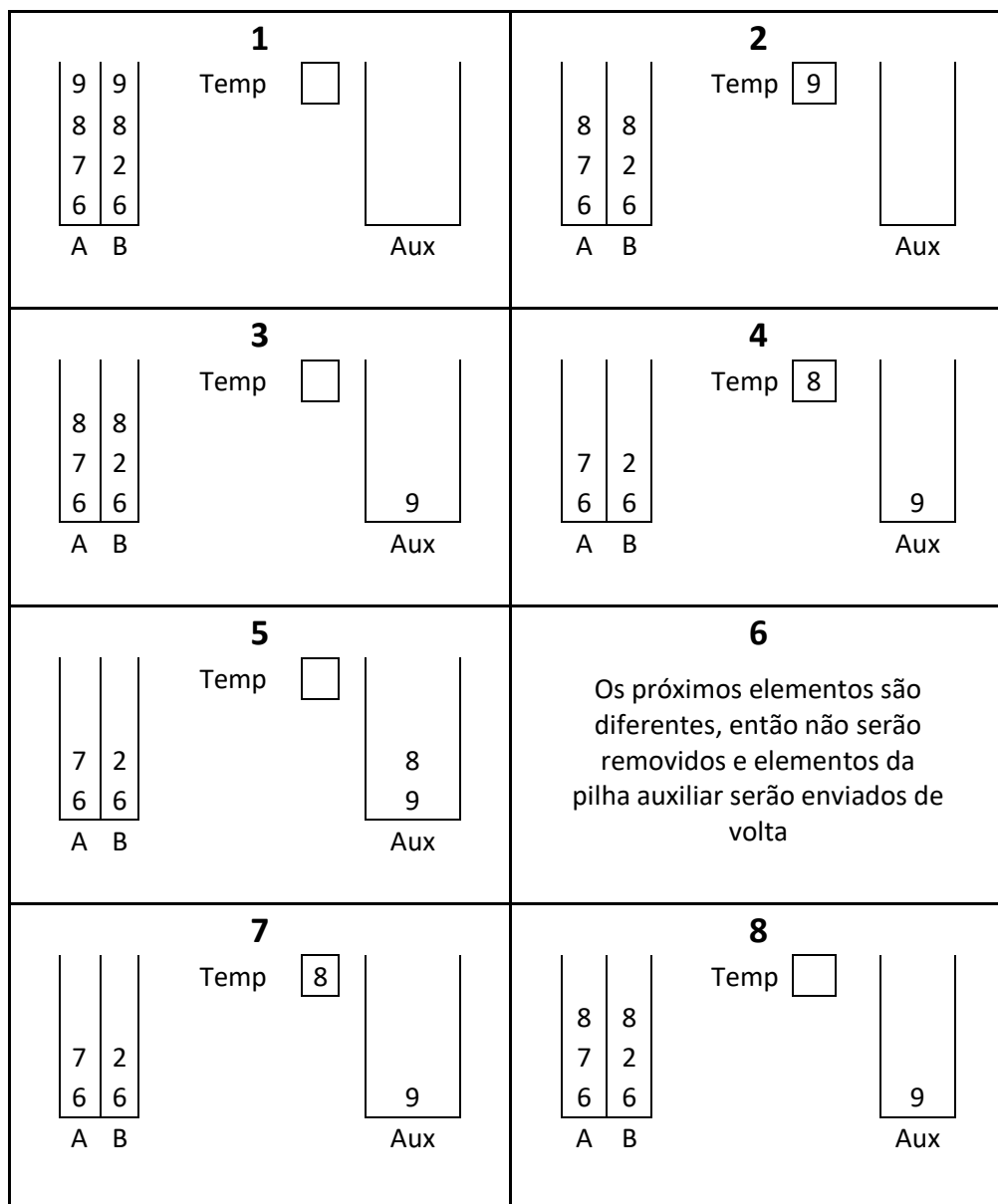
enquanto (!PilhaVazia(Aux)){
    Temp = RemovePilha(Aux);
    InserirPilha(A, Temp);
    InserirPilha(B, Temp);
}

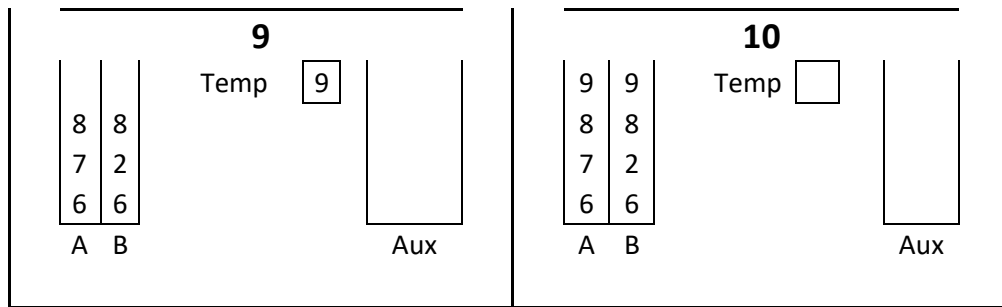
se (SaoIguais) {
    imprime("Sao Iguais\n");
} senao {
    imprime("Sao diferentes\n");
}
}

```

iii) **[0,7 Ponto]** ilustre seu algoritmo com um exemplo.

R:





3. [2,0 Pontos] considerando que o seguinte algoritmo é executado:

```

1. Q = createQueue()
2. enquanto (Não é Fim de arquivo) fazer
3.   ler número
4.   se (número diferente de 0)
5.     enqueue ( Q, número )
6.   se não
7.     x = queueRear( Q )
8.     enqueue ( Q, x )
9.   fim-se
10. fim-enquanto

```

Para processar a seguinte sequência de dados:

5, 7, 12, 4, 0, 4, 6, 8, 67, 34, 23, 5, 0, 44, 33, 22, 6, 0.

Responda:

i) [1,0 Pontos] qual seria o conteúdo da fila Q?

R: 5, 7, 12, 4, **4**, 4, 6, 8, 67, 34, 23, 5, **5**, 44, 33, 22, 6, **6**.

ii) [1,0 Pontos] explique o que o algoritmo faz.

R:

Cria uma fila Q

Enquanto não for o final do arquivo irá

Ler o próximo número presente no arquivo

Se esse número lido for diferente de 0

Irà adicionar esse número à fila Q

Se esse número lido não for diferente de 0, ou seja, se esse número lido for igual a 0

A variável X receberá o valor do último número que foi adicionado à fila Q

E o valor da variável X será adicionado ao fim da fila

Resumidamente:

Lê os números de um arquivo e adiciona eles à fila. Se o valor lido for 0, irá adicionar novamente o último número adicionado à fila.

4. [2,5 Pontos] responda os seguintes pontos:

i) [0,8 Ponto] descreva em palavras a ideia de um algoritmo que misture duas listas ordenadas em uma única lista, usando operações do TAD Lista;

R:

Uma função receberá 3 listas: uma que será a lista mesclada e as outras duas são as listas que serão mescladas. Haverá uma repetição até que todos os valores de A e B sejam lidos, essa repetição seguirá os seguintes passos: Primeiro será comparado entre os valores das listas A e B, o que tiver menor valor terá o seu valor copiado para um nó na lista F e a lista da qual foi tirado o elemento avançará para o próximo nó. Se o valor do nó de A for igual ao de B, o valor de A é copiado para F e tanto A quanto B avançam para o próximo nó. Até que todos os nós das duas listas já tenham sido percorridos.

ii) [1,0 Ponto] escreva o algoritmo correspondente;

R:

```
void mesclarListas (LIST* Final, LIST* A, LIST* B) {
    if(!emptyList(A)&&emptyList(B)){
        NODE* Anode=A->head;
        NODE* Bnode=B->head;
        NODE* Fnode=Final->head;
        while((Anode!=NULL)&&(Bnode!=NULL)){
            if(Anode==NULL){
                addNode(F,Bnode->dataPtr);
                Bnode=Bnode->link
            } else if(Bnode==NULL){
                addNode(F,Anode->dataPtr);
                Anode=Anode->link
            } else {
                if (Anode->dataPtr<Bnode->dataPtr){
                    addNode(F,Anode->dataPtr);
                    Anode=Anode->link
                } else if(Bnode->dataPtr<Anode->dataPtr){
                    addNode(F,Bnode->dataPtr);
                    Bnode=Bnode->link
                } else if (Anode->dataPtr==Bnode->dataPtr){
                    addNode(F,Anode->dataPtr);
                    Anode=Anode->link
                    Bnode=Bnode->link
                }
            }
        }
    }
}
```

iii) [0,7 Ponto] ilustre seu algoritmo com um exemplo.

R:

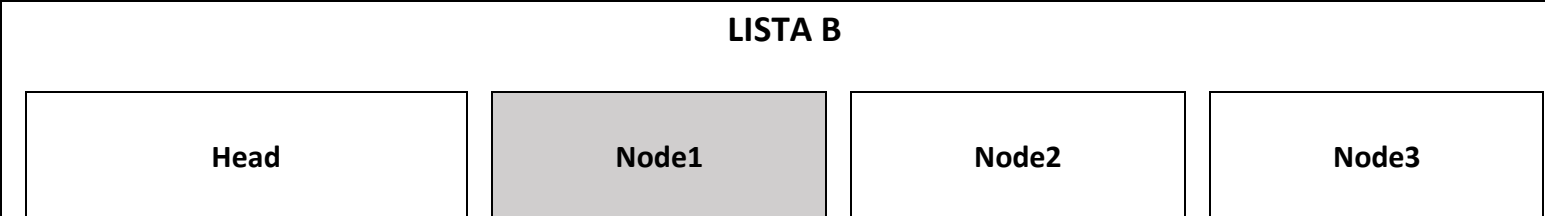
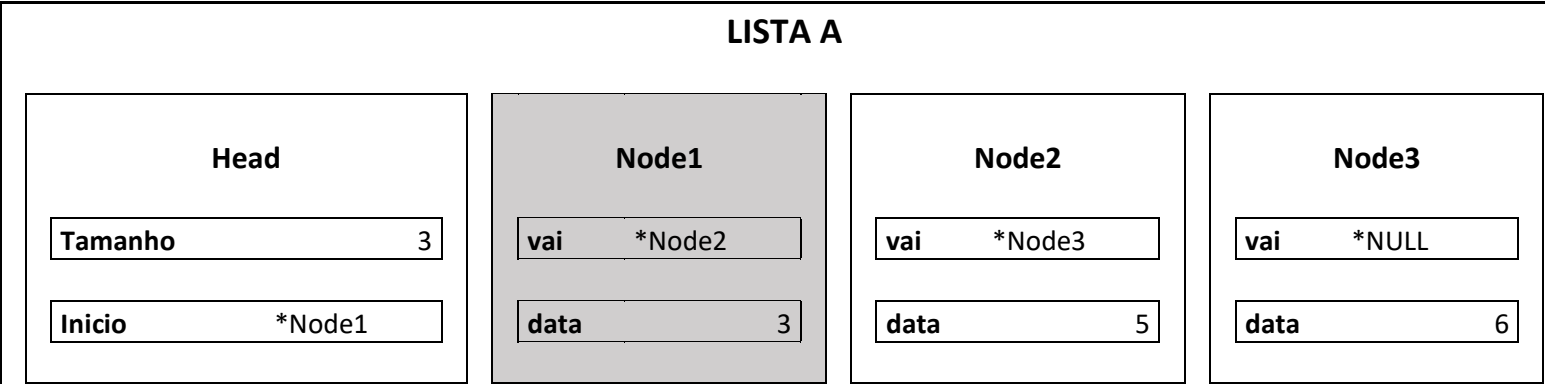
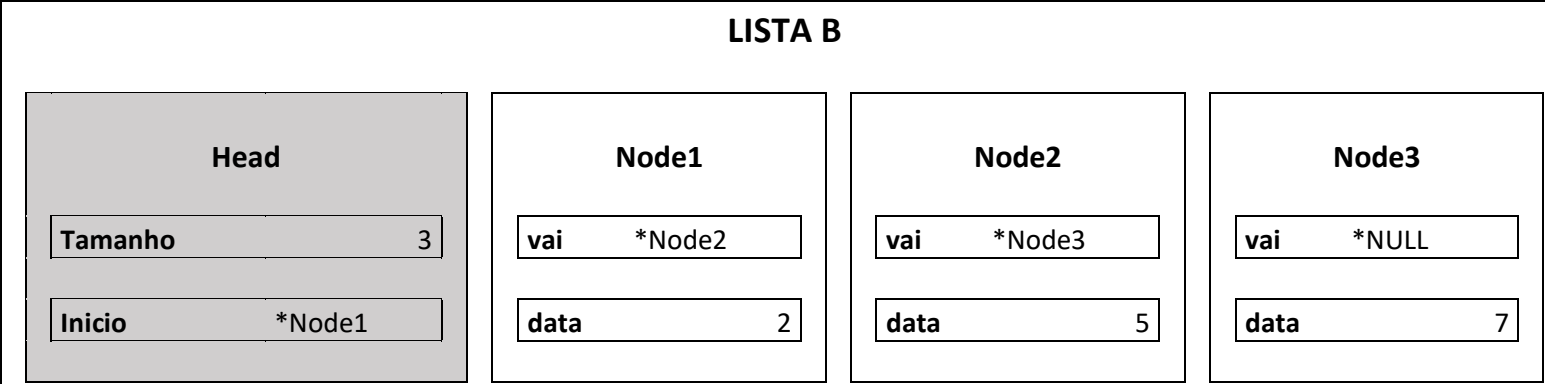
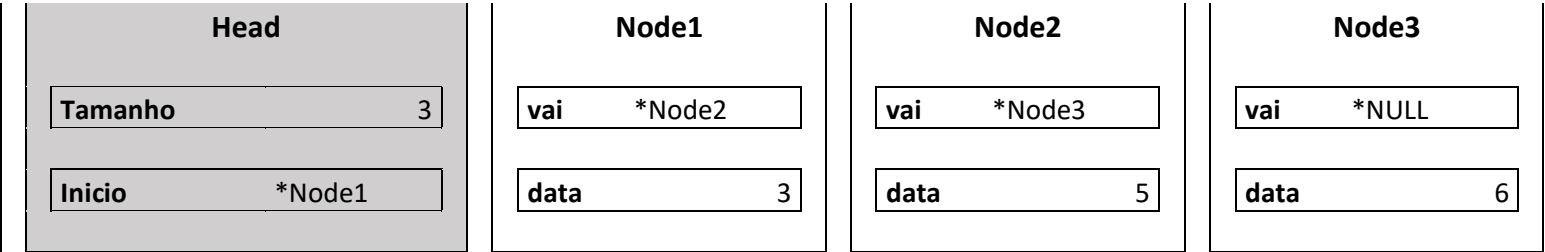
LISTA A

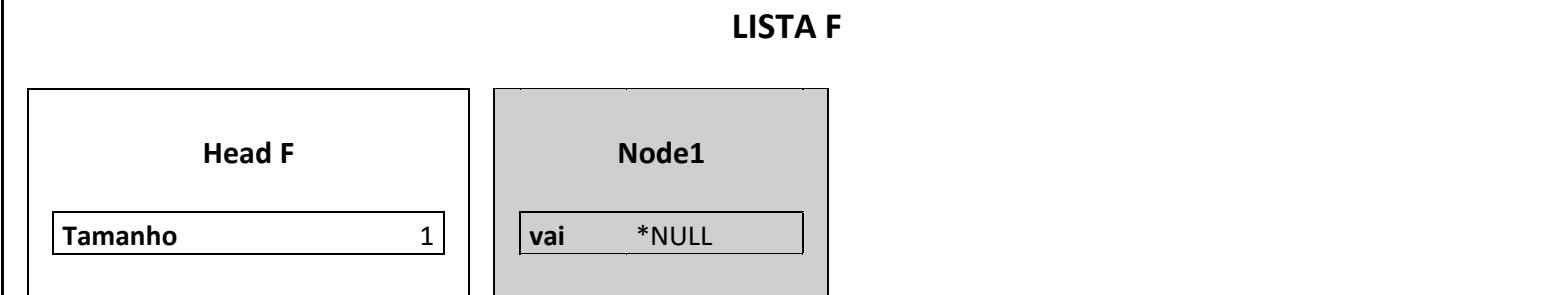
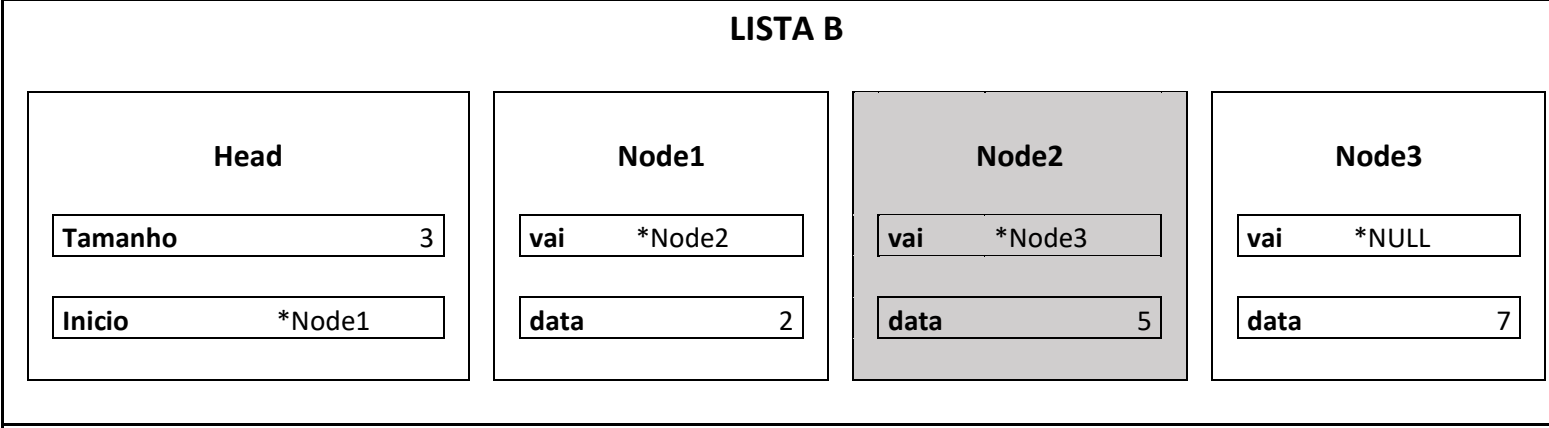
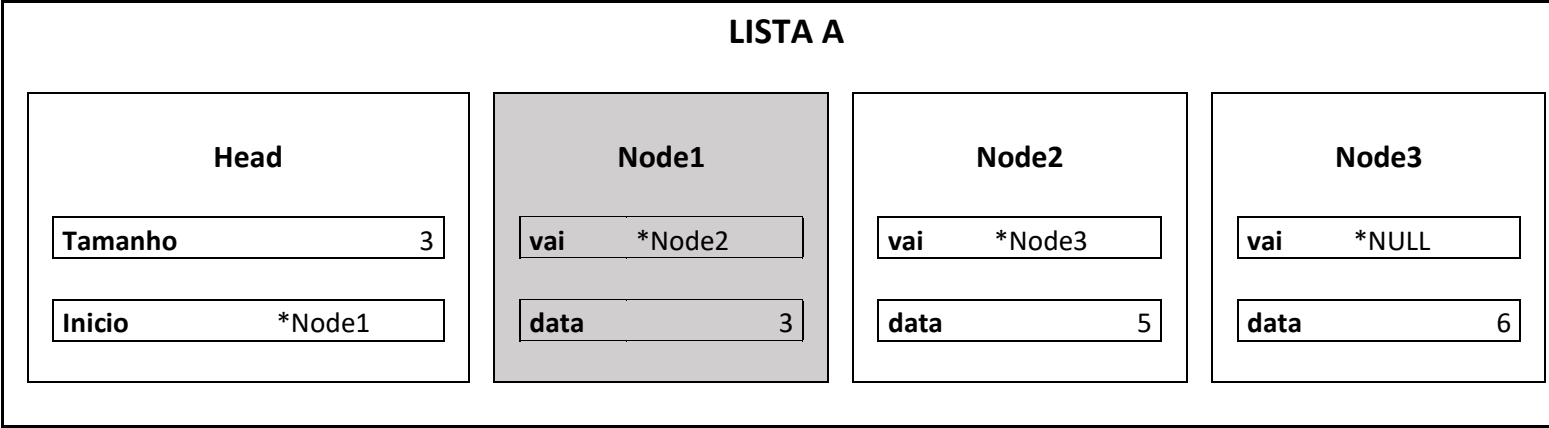
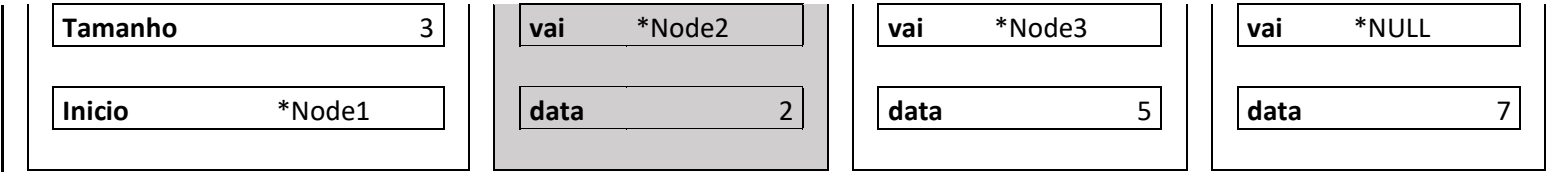
--

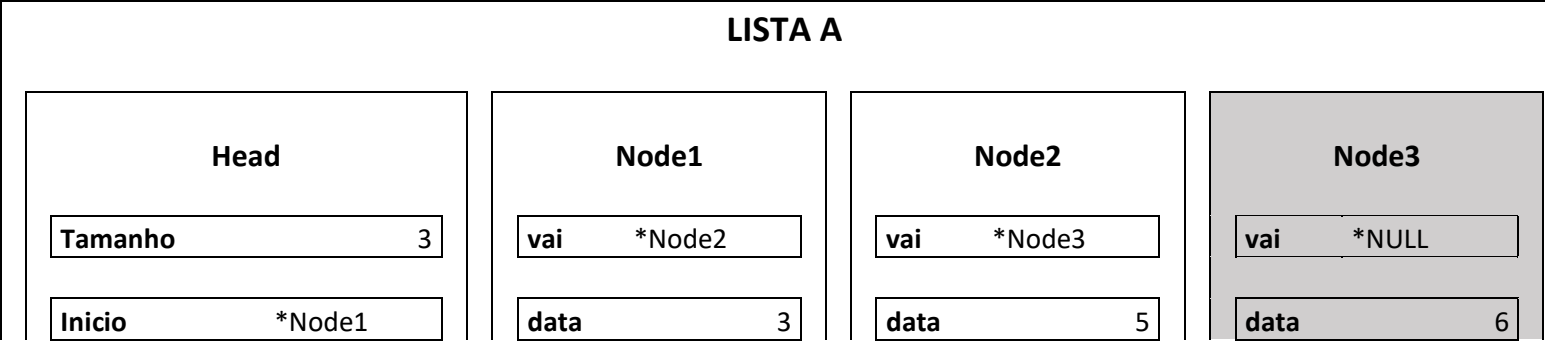
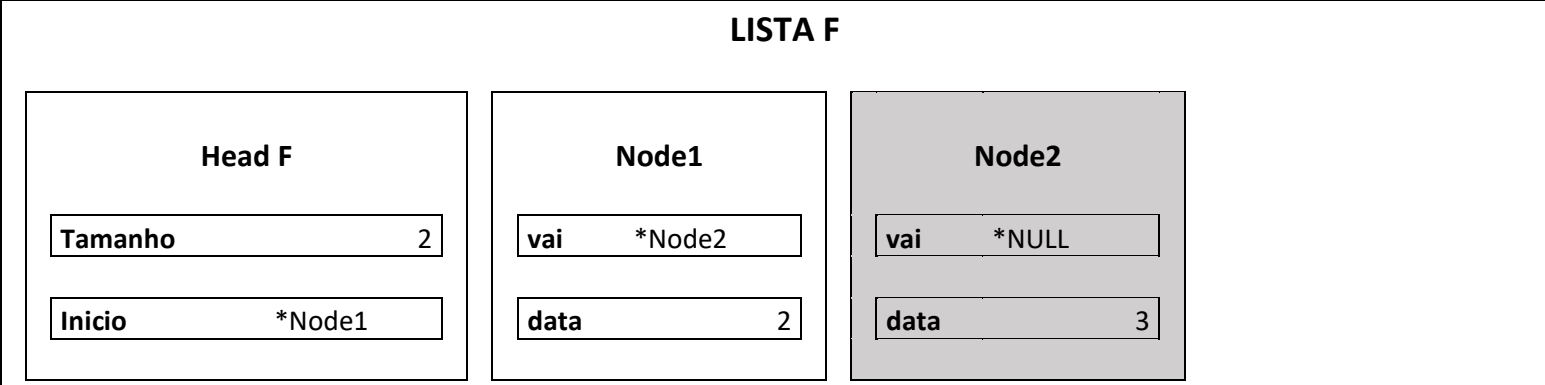
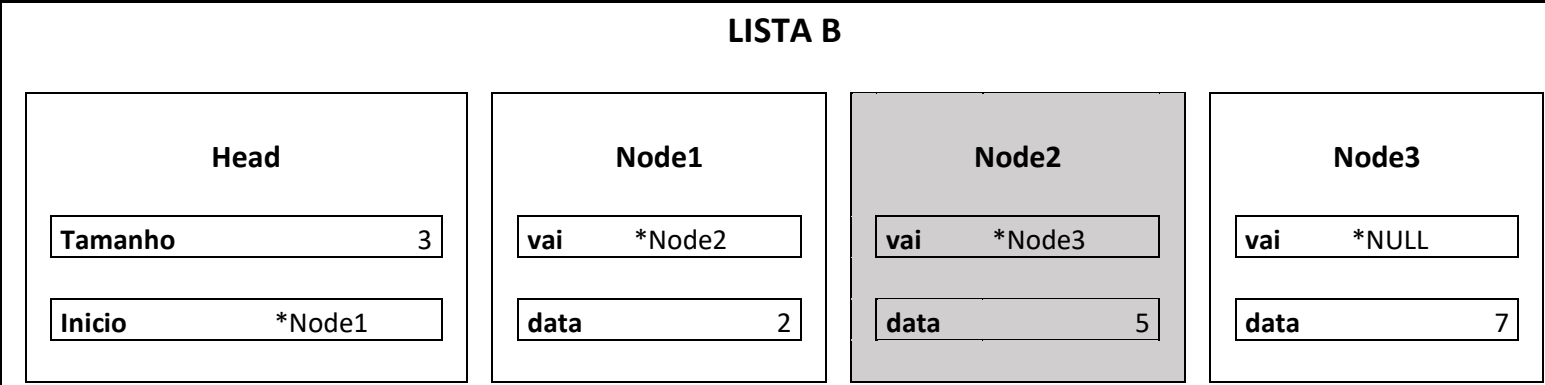
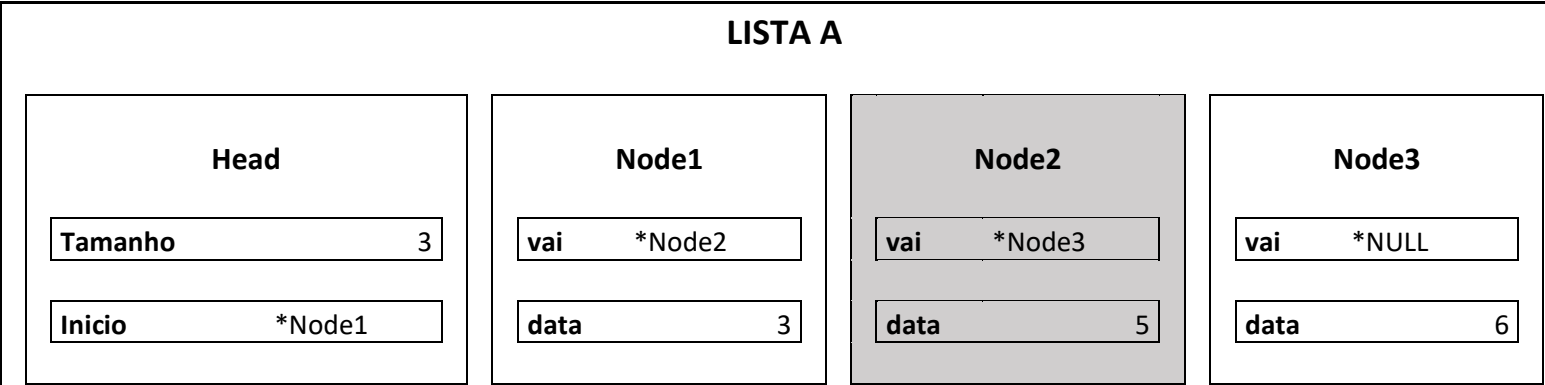
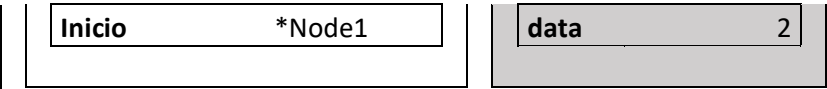
--

--

--







--	--	--	--

LISTA B

Head	Node1	Node2	Node3
<div>Tamanho3</div> <div>Inicio*Node1</div>	<div>vai*Node2</div> <div>data2</div>	<div>vai*Node3</div> <div>data5</div>	<div>vai*NULL</div> <div>data7</div>

LISTA F

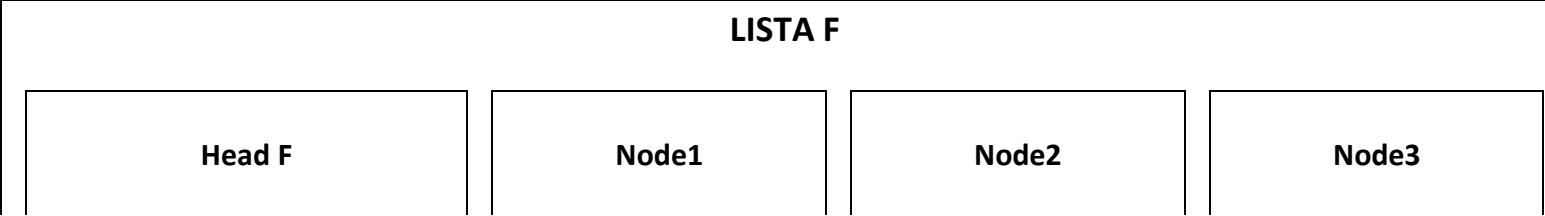
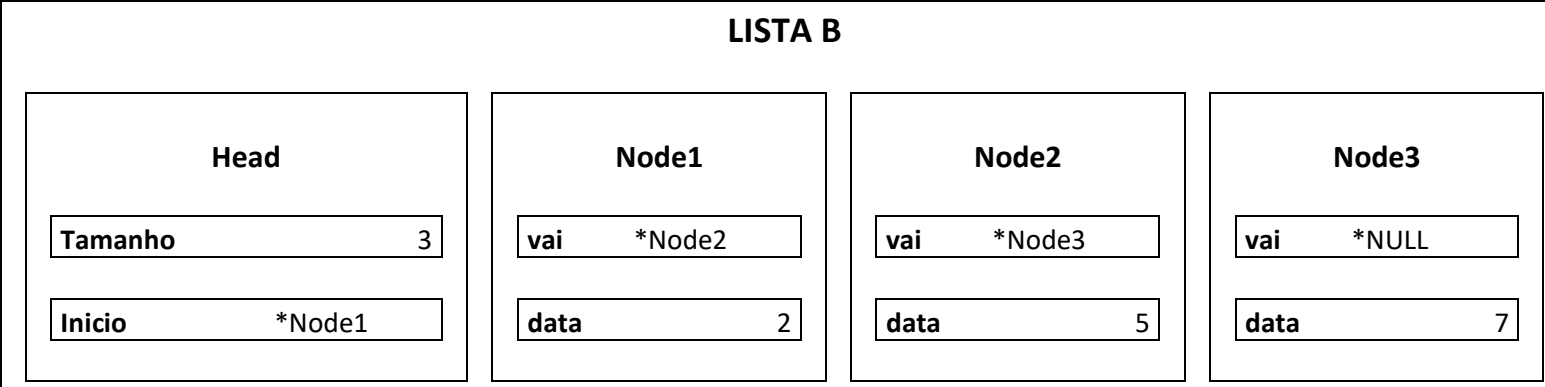
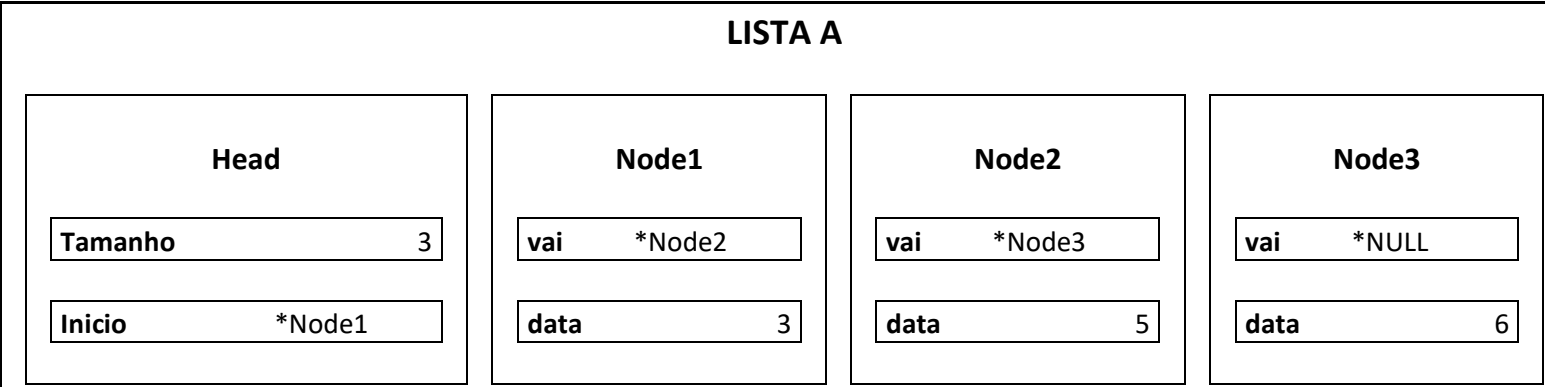
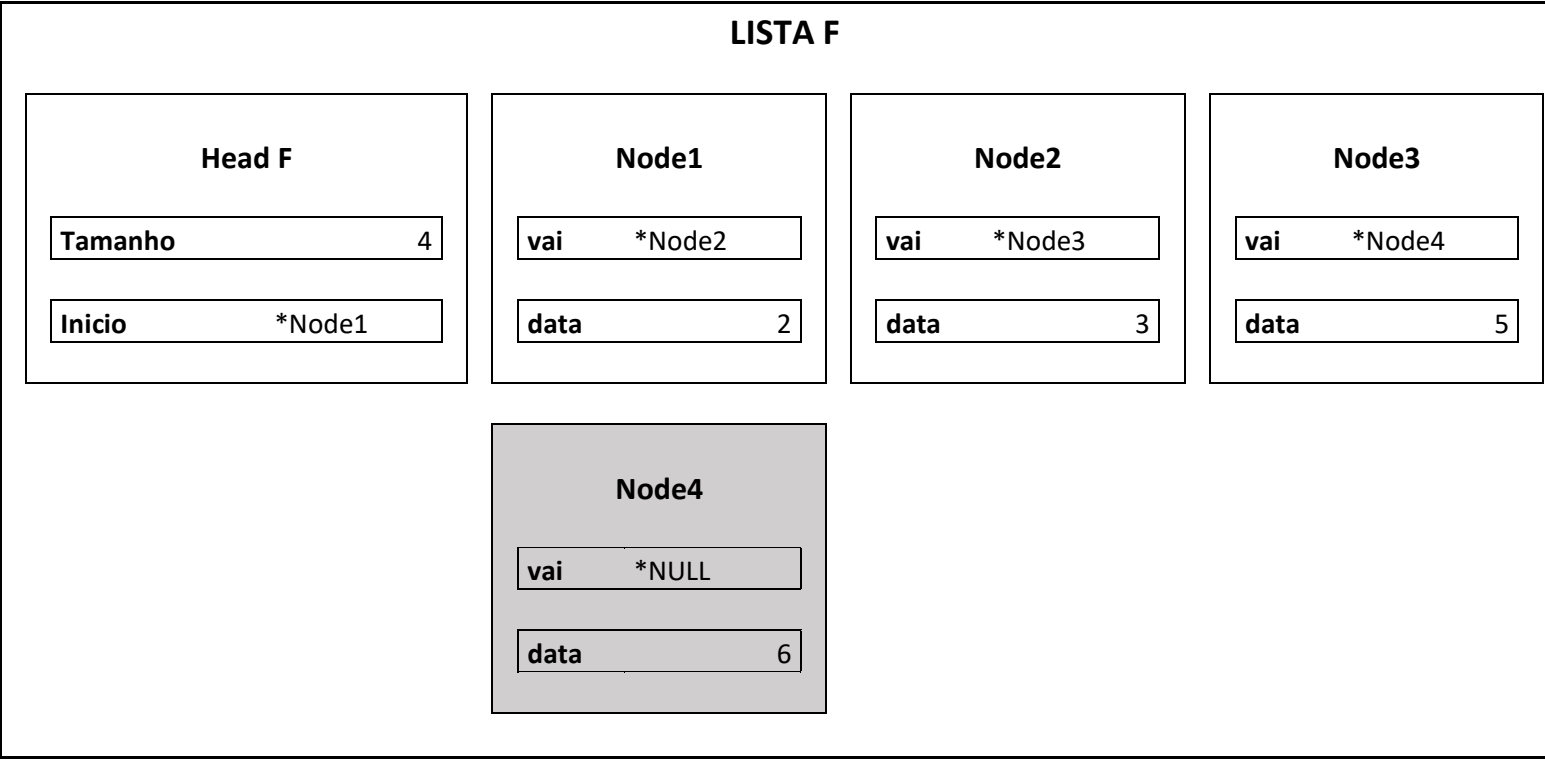
Head F	Node1	Node2	Node3
<div>Tamanho3</div> <div>Inicio*Node1</div>	<div>vai*Node2</div> <div>data2</div>	<div>vai*Node3</div> <div>data3</div>	<div>vai*NULL</div> <div>data5</div>

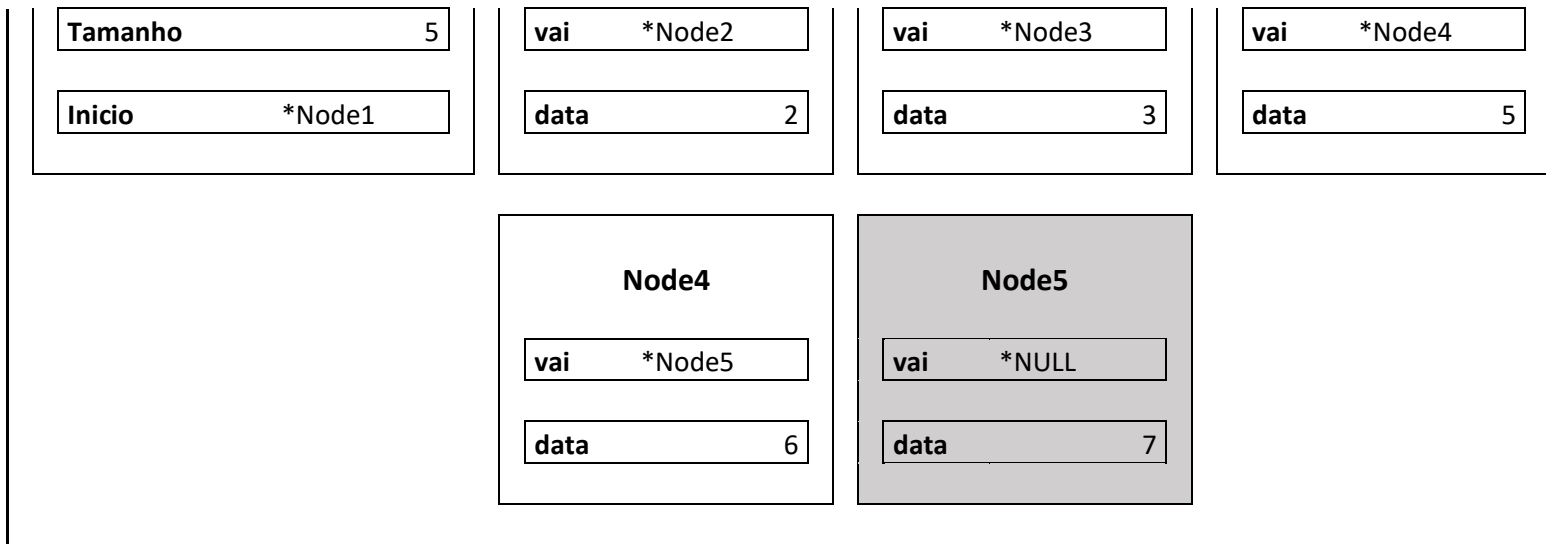
LISTA A

Head	Node1	Node2	Node3
<div>Tamanho3</div> <div>Inicio*Node1</div>	<div>vai*Node2</div> <div>data3</div>	<div>vai*Node3</div> <div>data5</div>	<div>vai*NULL</div> <div>data6</div>

LISTA B

Head	Node1	Node2	Node3
<div>Tamanho3</div> <div>Inicio*Node1</div>	<div>vai*Node2</div> <div>data2</div>	<div>vai*Node3</div> <div>data5</div>	<div>vai*NULL</div> <div>data7</div>

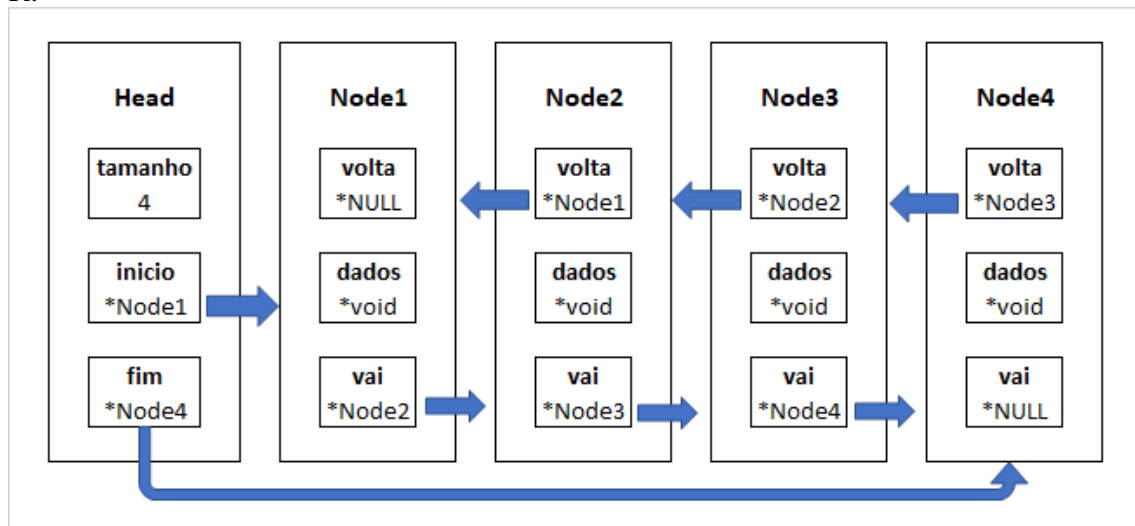




5. [1,5 Pontos] defina o conceito de lista duplamente encadeada.

i) Ilustre graficamente.

R:



ii) Cite uma vantagem e uma desvantagem do uso de listas duplamente encadeadas com relação as listas encadeadas simples.

R:

Uma das vantagens do uso de listas duplamente encadeadas em relação às listas encadeadas simples é a maior mobilidade de navegação interna, podendo ela avançar ou retroceder entre os seus elementos.

Uma das desvantagens do uso de listas duplamente encadeadas em relação às listas encadeadas simples o aumento da memória necessária por causa do aumento na quantidade de ponteiros presentes em cada elemento da lista.