



ESTRUTURA DE DADOS

FILAS DINÂMICAS

Prof. Dr. Daniel Caetano

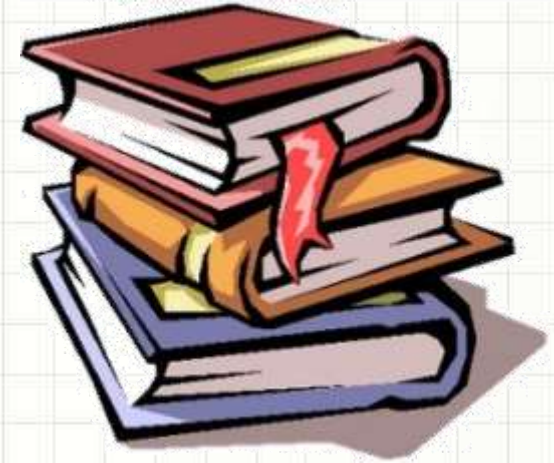
2012 - 2

Objetivos

- Compreender o conceito de fila encadeada
- Usar filas dinâmicas para aplicações
- Conhecer as vantagens e desvantagens das filas encadeadas
- **Atividade Estruturada!**



Material de Estudo



Material

Acesso ao Material

Apresentação

<http://www.caetano.eng.br/>
(Aula 12)

Material Didático

Estruturas de Dados (Parte 2) – Páginas 143 a 146

Online

C Completo e Total – Páginas 540 a 550



RECORDANDO...

Recordando...

- Vimos Pilhas Encadeadas
- Uso:
 - Inserção no Topo (Início)
 - Remoção do Topo (Início)
- E se continuarmos removendo do início, mas inserirmos no fim?
- Virou uma fila?



FILA ENCADEADA OU FILA DINÂMICA

Filas Encadeadas

- Sempre que usarmos uma lista encadeada
 - Inserirmos no fim... e
 - Removermos do início...
- Fila Dinâmica ou Encadeada
- Exemplo

Fila: Inicializando

fila

NULL

A diagram showing a variable named 'fila' on the left, with a blue arrow pointing to a light green rounded rectangle containing the text 'NULL'.

Quando criamos uma
fila, ela está vazia

```
no *fila = NULL;
```


Fila: Inserção

fila

NULL

Como criamos um elemento?

tmp

1

NULL

```
no *tmp;  
tmp = new no;  
tmp->valor = 1;  
tmp->prox = NULL
```

Fila: Inserção

fila

NULL

Como inseri-lo na fila?

tmp

1

NULL

```
if (fila == NULL) {  
  fila = tmp;  
}
```

Fila: Inserção

fila

NULL

Como inseri-lo na fila?

tmp

1

NULL

```
if (fila == NULL) {  
    fila = tmp;  
}
```

Fila: Inserção

fila

NULL

Como inseri-lo na fila?

tmp

1

NULL

```
if (fila == NULL) {  
    fila = tmp;  
}
```

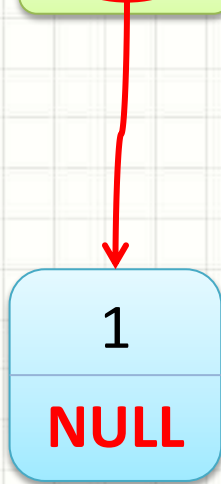
Fila: Inserção



Como inseri-lo na fila?

```
if (fila == NULL) {  
    fila = tmp;  
}
```

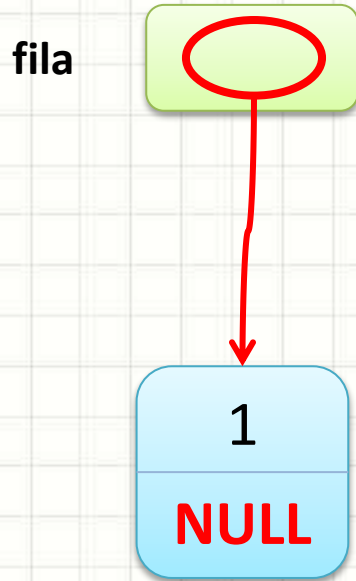

Fila: Inserção



Inserindo outro nó

```
if (fila != NULL) {  
    fim = fila;  
    while (fim->prox != NULL) {  
        fim = fim->prox;  
    }  
    fim->prox = tmp;  
}
```

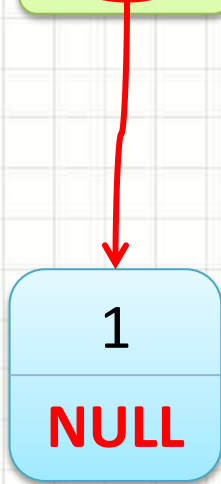
Fila: Inserção



Inserindo outro nó

```
if (fila != NULL) {  
    fim = fila;  
    while (fim->prox != NULL) {  
        fim = fim->prox;  
    }  
    fim->prox = tmp;  
}
```

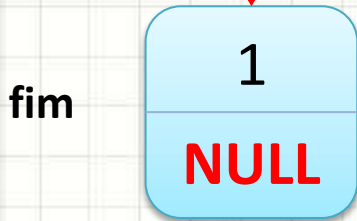
Fila: Inserção



Inserindo outro nó

```
if (fila != NULL) {  
    fim = fila;  
    while (fim->prox != NULL) {  
        fim = fim->prox;  
    }  
    fim->prox = tmp;  
}
```

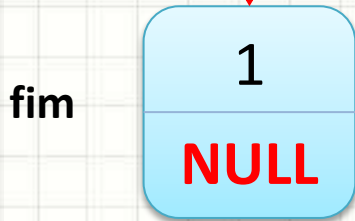
Fila: Inserção



Inserindo outro nó

```
if (fila != NULL) {  
    fim = fila;  
    while (fim->prox != NULL) {  
        fim = fim->prox;  
    }  
    fim->prox = tmp;  
}
```

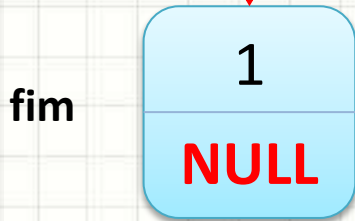
Fila: Inserção



Inserindo outro nó

```
if (fila != NULL) {  
    fim = fila;  
    while (fim->prox != NULL) {  
        fim = fim->prox;  
    }  
    fim->prox = tmp;  
}
```

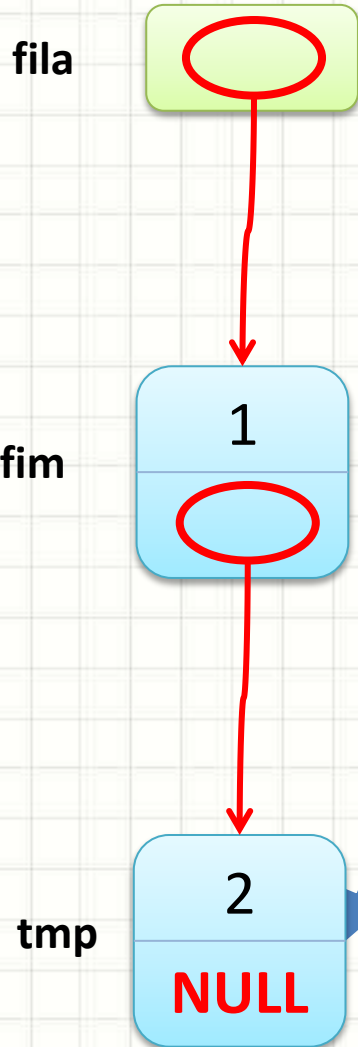

Fila: Inserção



Inserindo outro nó

```
if (fila != NULL) {  
    fim = fila;  
    while (fim->prox != NULL) {  
        fim = fim->prox;  
    }  
    fim->prox = tmp;  
}
```

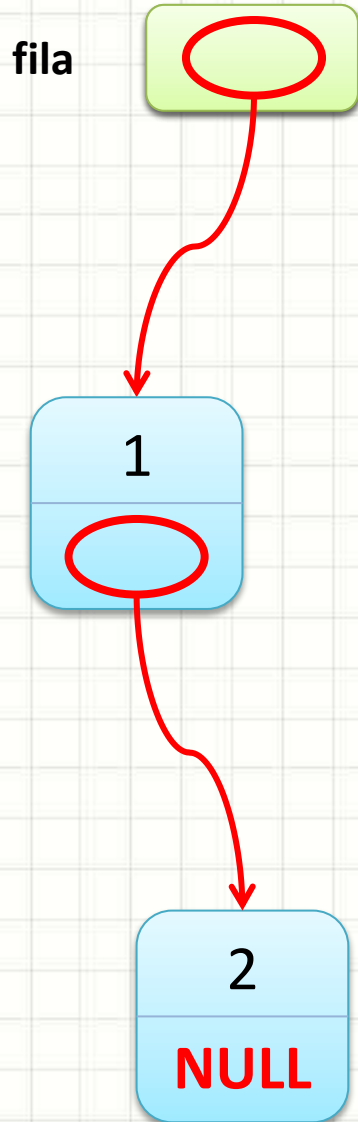
Fila: Inserção



Inserindo outro nó

```
if (fila != NULL) {  
    fim = fila;  
    while (fim->prox != NULL) {  
        fim = fim->prox;  
    }  
    fim->prox = tmp;  
}
```

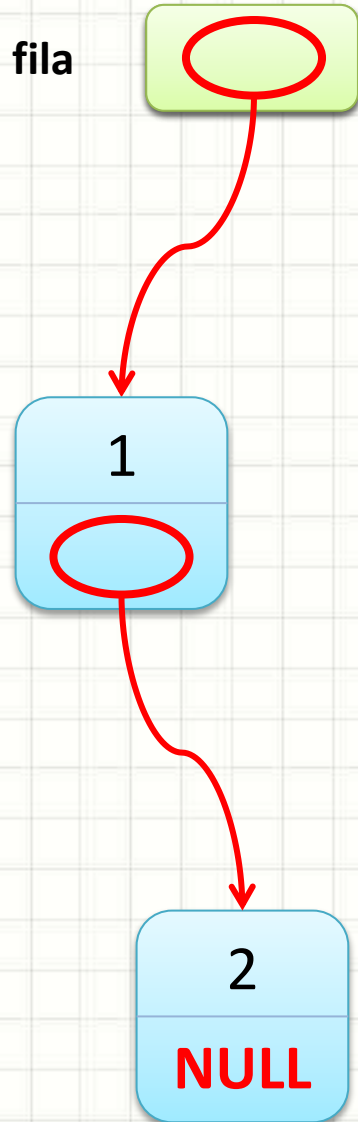
Fila: Remoção



Removendo um nó

```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

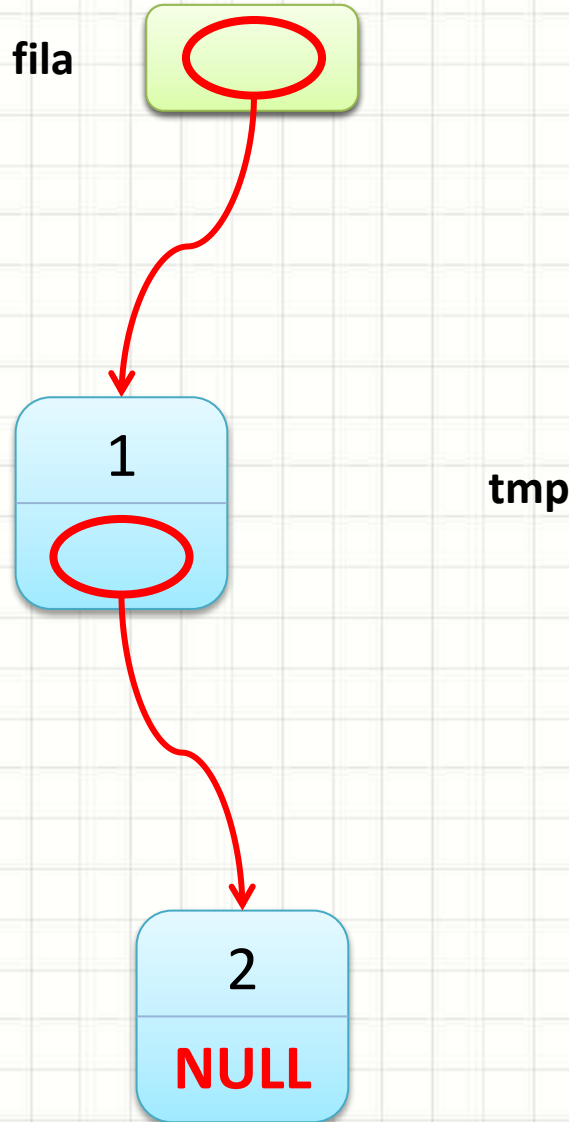
Fila: Remoção



Removendo um nó

```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

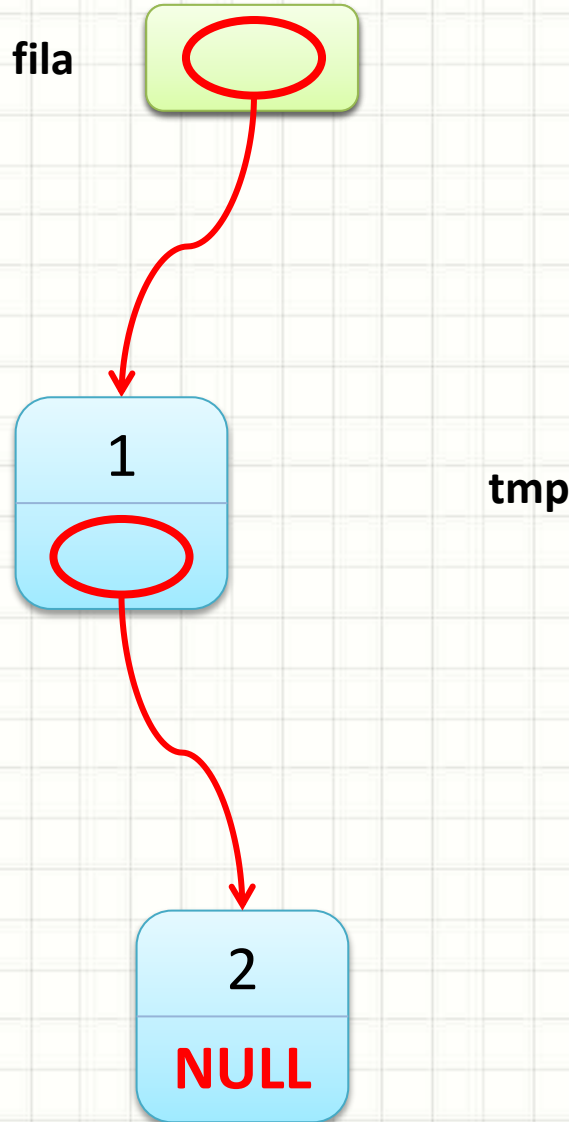
Fila: Remoção



Removendo um nó

```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

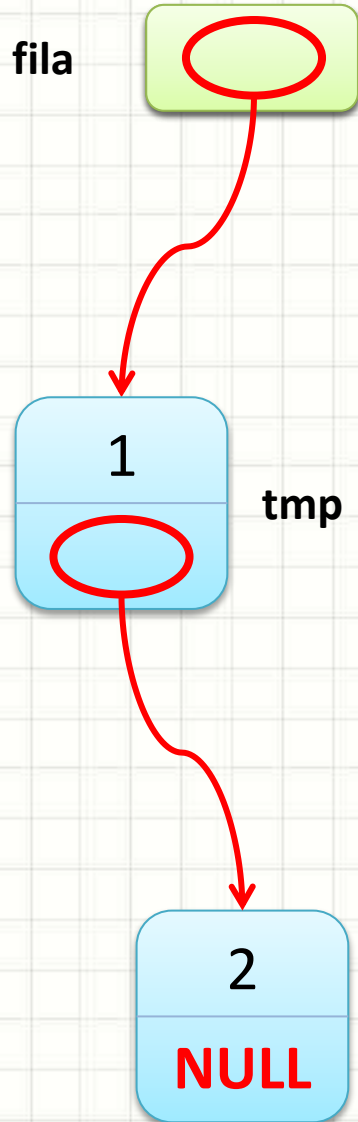

Fila: Remoção



Removendo um nó

```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

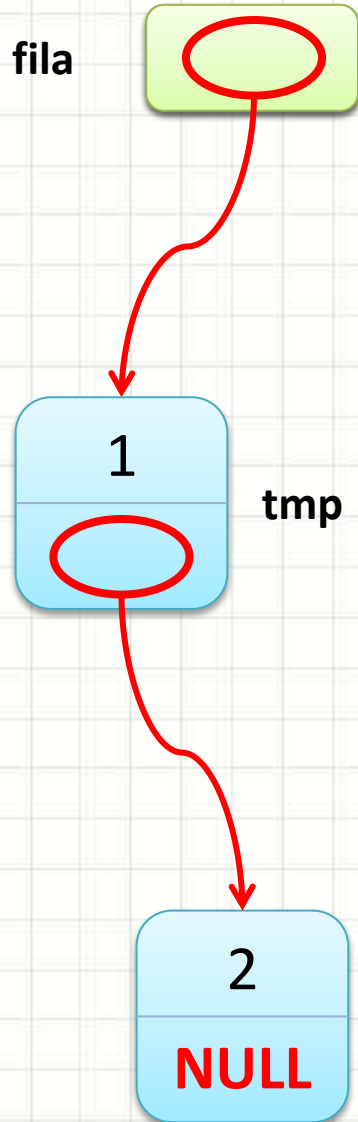
Fila: Remoção



Removendo um nó

```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

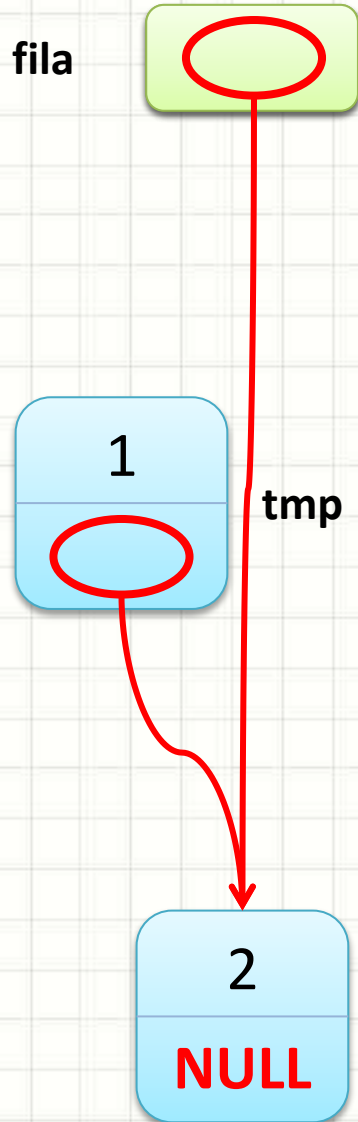
Fila: Remoção



Removendo um nó

```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

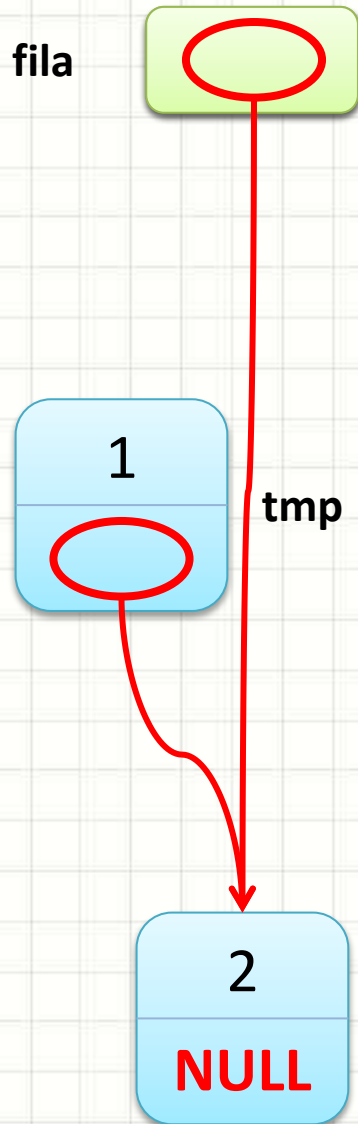
Fila: Remoção



Removendo um nó

```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

Fila: Remoção



Removendo um nó

```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```


Fila: Remoção

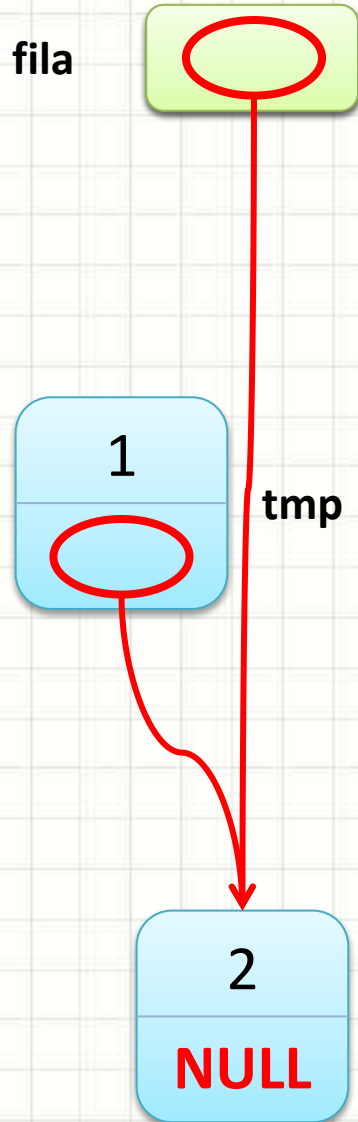


Removendo um nó

1

```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

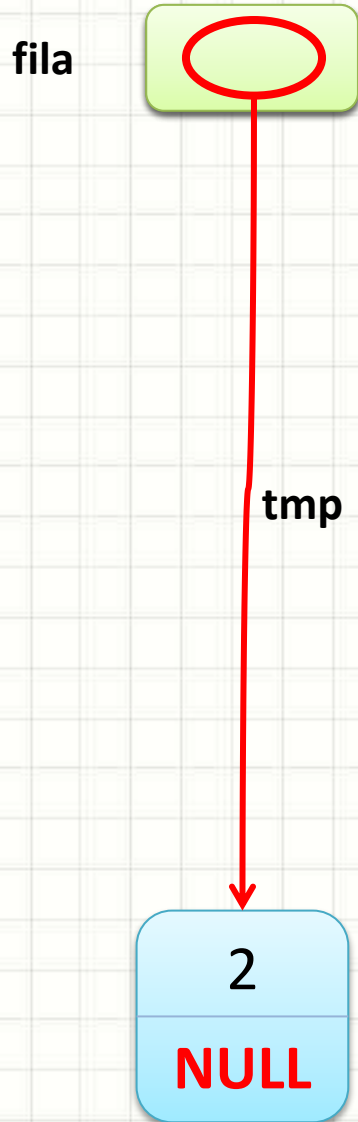
Fila: Remoção



Removendo um nó

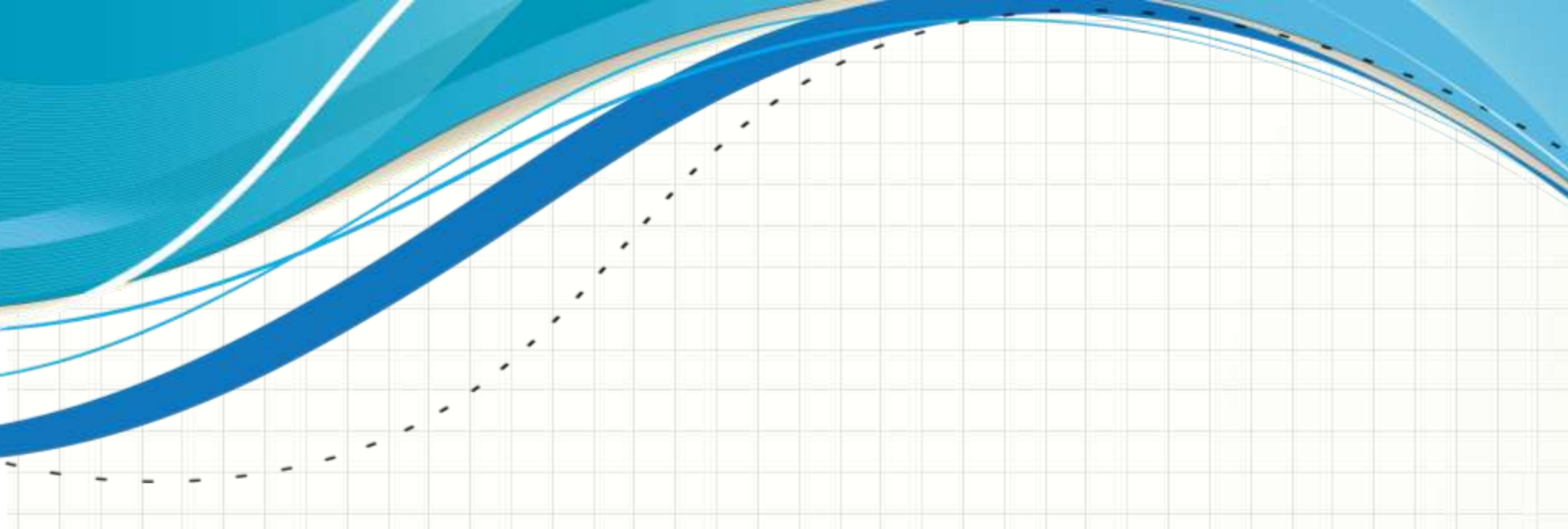
```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

Fila: Remoção



Removendo um nó

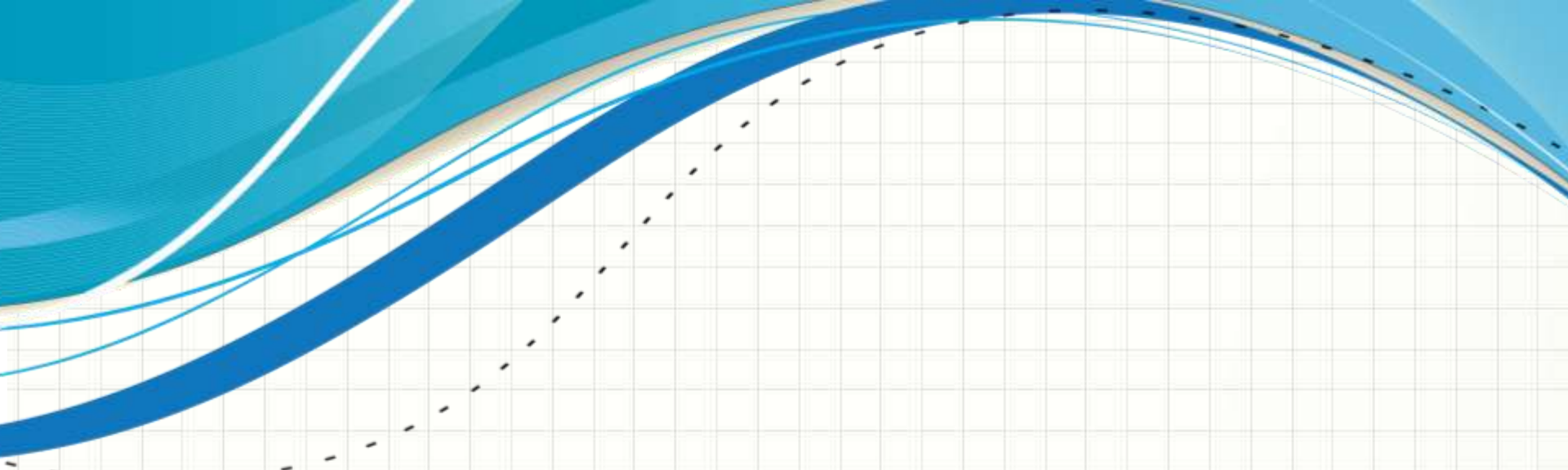
```
no *tmp;  
tmp = fila;  
fila = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```



DISCUSSÃO: QUALIDADE DA IMPLEMENTAÇÃO

Qualidade da Implementação

- Qual o problema dessa implementação?
 - Pense na inserção e remoção...
 - Há como melhorar?



PRÁTICA:
CONSTRUINDO UMA
FILA ENCADEADA

Construindo uma Pilha Encadeada

- Acompanhe o professor na construção...



EXERCÍCIOS DE FIXAÇÃO

Exercícios

- Como base em “ex1.cpp” e “filaaluno.cpp”:
 - Modifique ex1.cpp para que implemente uma fila de alunos
- Em grupos de 4 alunos, debata as vantagens e desvantagens do uso de encadeamento;
 - Elabore uma lista de vantagens e desvantagens comparando as estruturas de dados contíguas com suas versões encadeadas: pilhas, filas e listas

Vantagens e Desvantagens

	Vantagens	Desvantagens
Encadeados Em geral		
Pilhas Encadeadas		
Filas Encadeadas		
Listas Encadeadas		

Vantagens e Desvantagens

	Vantagens	Desvantagens
Encadeados Em geral	<ul style="list-style-type: none">- Menor consumo de memória- Menor processamento em tarefas complexas- Tamanho máximo limitado pela memória do equipamento- Protótipos mais simples	<ul style="list-style-type: none">- Maior complexidade de programação- Uso de ponteiros- Maior processamento em tarefas simples- Consumo de processamento para alocação de memória
Pilhas Encadeadas	<ul style="list-style-type: none">- Apenas um apontador	
Filas Encadeadas	<ul style="list-style-type: none">- Não exige implementação circular- Apenas dois apontadores	<ul style="list-style-type: none">- Se não for usado apontador de fim, custo para inserção cresce com o tamanho da fila
Listas Encadeadas	<ul style="list-style-type: none">- Inserção ordenada e ordenação mais eficientes- Apenas um apontador	<ul style="list-style-type: none">- Acesso aleatório prejudicado (busca)



CONCLUSÕES

Resumo

- As listas encadeadas podem ser usadas como filas
- É mais interessante criar uma estrutura específica, com referências para início e fim
- Adaptar programas que usam filas contíguas para usar filas encadeadas é simples

- **TAREFA**
 - Atividade Estruturada!

Próxima Aula



THE END!



PERGUNTAS?



**BOM DESCANSO
A TODOS!**