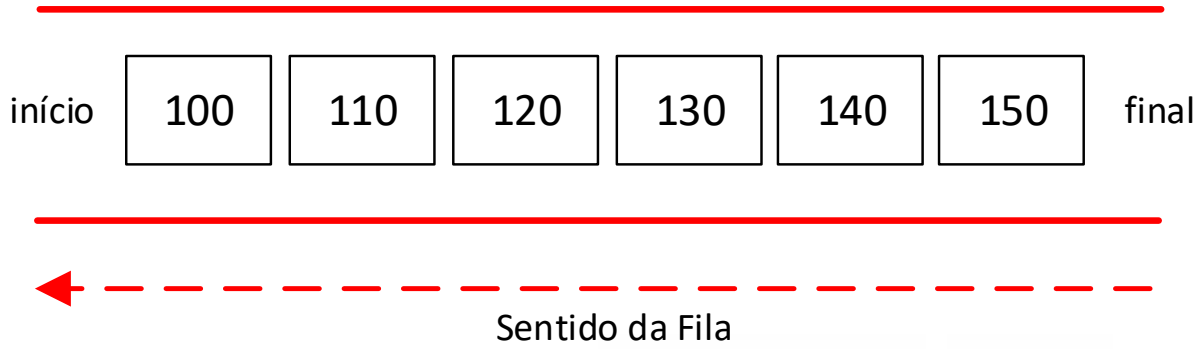
Estruturas de Dados 1

12 – Filas Dinâmicas

Antonio Angelo de Souza Tartaglia
angelot@ifsp.edu.br

Fila

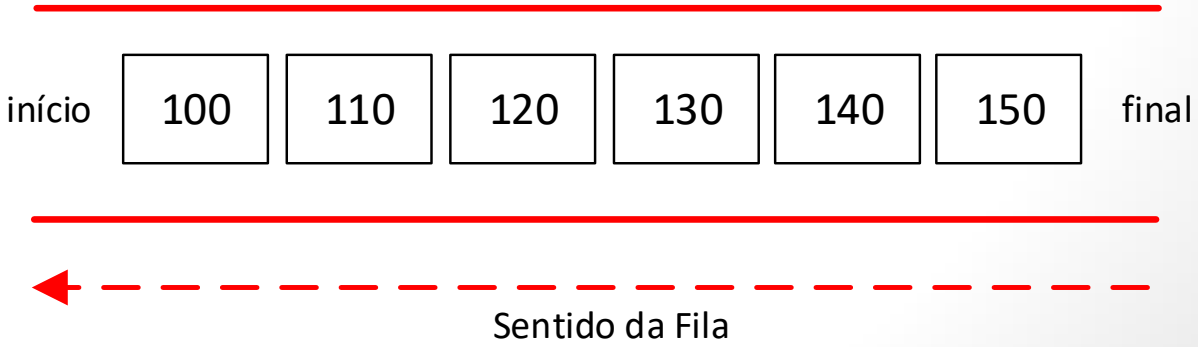
- Uma estrutura do tipo “Fila” é uma sequência de elementos do mesmo tipo, como as “Listas” que vimos anteriormente. Seus elementos possuem estrutura interna abstraída, ou seja, sua complexidade é arbitrária e não afeta o seu funcionamento. São estruturas do tipo “*FIFO*” (*first-in-first-out*).



Estrutura de Dados 1

Fila

- Uma fila é um tipo especial de “Lista”:
  - Inserções e exclusões de elementos ocorrem nas extremidades.
- Aplicações
  - Qualquer aplicação onde se necessite de:
    - Controle de fluxo;
    - Recursos compartilhados (impressoras, transações de bancos de dados, etc);



Estrutura de Dados 1

Fila

- Em uma fila podemos realizar as seguintes operações:
  - Criação da Fila;
  - Inserção de um elemento no final da Fila;
  - Remoção de um elemento no início da Fila;
  - Acesso ao elemento do início da Fila;
  - Destruição da Fila;
  - Etc.
- Essas operações dependem do tipo de alocação de memória utilizada:
  - Alocação estática;
  - Alocação Dinâmica.

Estrutura de Dados 1

Fila

- Alocação Estática:
  - O Espaço de memória é alocado no momento da compilação;
  - Exige a definição do número máximo de elementos da “Fila”;
  - Acesso sequencial: Elementos estão de forma consecutiva na memória.

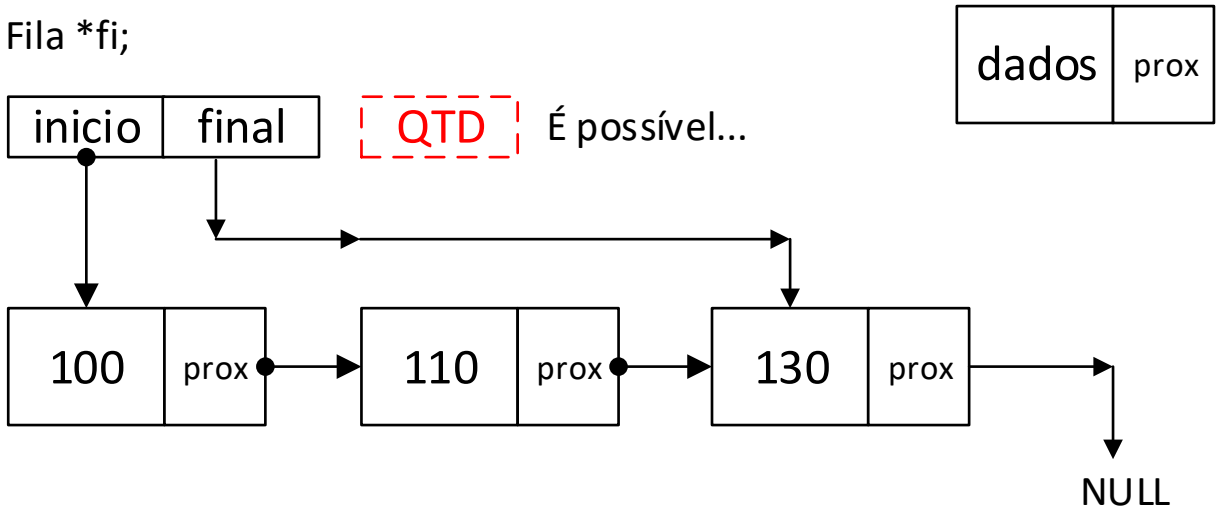
Estrutura de Dados 1

Fila

- Alocação Dinâmica:
  - O espaço de memória é alocado em tempo de execução;
  - A fila cresce a medida que novos elementos são armazenados, e diminui a medida que elementos são removidos;
  - Acesso encadeado: Cada elemento pode estar em uma área distinta da memória. Para acessar um elemento, é preciso percorrer todos os seus antecessores na “Fila”.

Fila Dinâmica - Implementação

- Em uma Fila Dinâmica cada elemento aponta para o seu sucessor na Fila;
- Este tipo de Fila usa um nó “descritor” para representar o início e o final da fila, e uma indicação especial de final de Fila:



Fila Dinâmica - Implementação

- **filaD.h**
  - Os protótipos das funções que manipulam os dados no tipo “Fila” (arquivo “.c”);
  - O tipo de dado que será armazenado na Fila;
  - O ponteiro Fila.
- **filaD.c**
  - O tipo de dados “Fila”;
  - Implementar as funções que manipulam exclusivamente o tipo de dado “Fila”.

Fila Dinâmica - Implementação

- Variáveis para utilização no programa
  - A variável x será utilizada para o retorno de informações de execução das funções de manipulação do tipo de dado Fila.
  - As variáveis `al1`, `al2` e `al3` serão inseridas na Fila. A variável `al` será utilizada para o retorno de dados da função consulta.

```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "filaD.h"

int main()
{
    int x; //para os codigos de erro
    ALUNO al, al1, al2, al3;
    al1.matricula = 100;
    al1.n1 = 8.3;
    al1.n2 = 8.4;
    al1.n3 = 8.5;

    al2.matricula = 110;
    al2.n1 = 7.3;
    al2.n2 = 7.4;
    al2.n3 = 7.5;

    al3.matricula = 120;
    al3.n1 = 6.3;
    al3.n2 = 6.4;
    al3.n3 = 6.5;
```

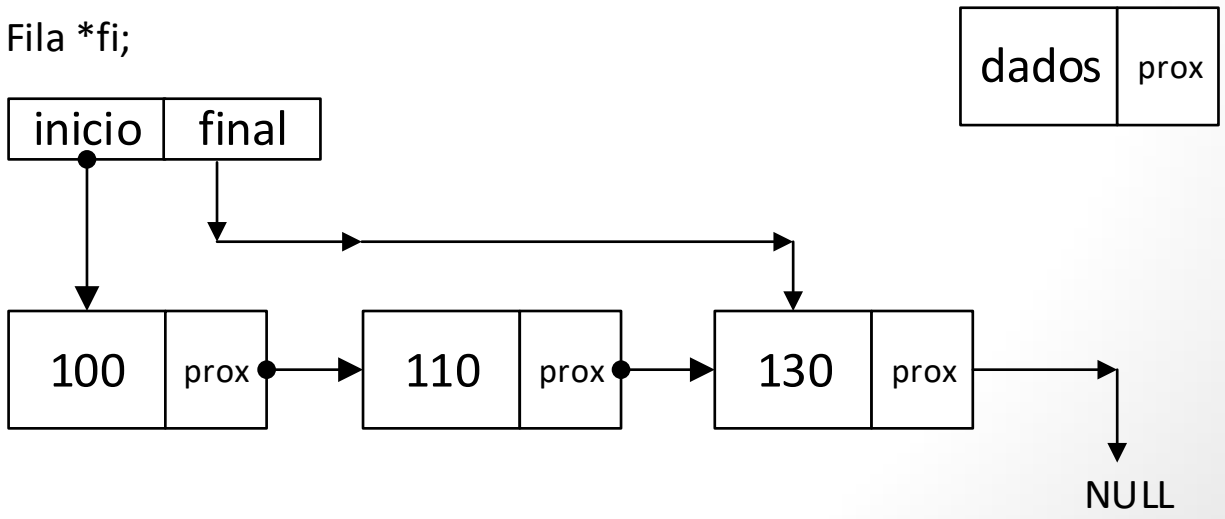
Fila Dinâmica - Implementação

```c
//Arquivo filaD.c
#include <stdio.h>
#include <stdlib.h>
#include "filaD.h"

struct fila{
    struct elemento *inicio;
    struct elemento *fim;
};

struct elemento{
    ALUNO dados;
    struct elemento *prox;
};

typedef struct elemento Elem;
```

Apenas para não digitar muito a todo instante...

```c
//Arquivo main.c
Fila *fi; //ponteiro para o no descritor
```

```c
//Arquivo filaD.h
typedef struct aluno{
    int matricula;
    float n1, n2, n3;
}ALUNO;

typedef struct fila Fila;
```

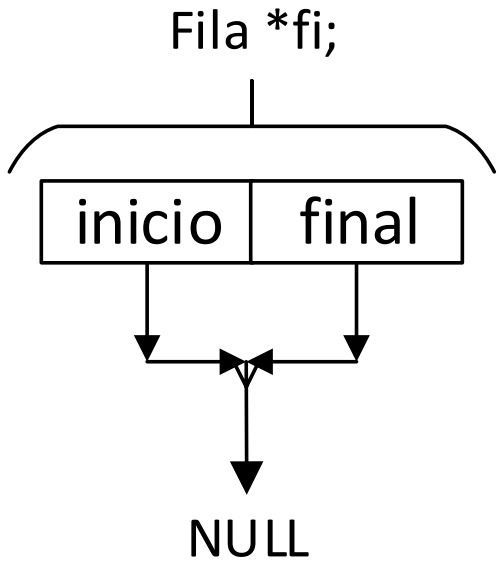


Fila Dinâmica - Criando a Fila

```c
//Arquivo filaD.h
Fila *cria_fila();
```

```c
//Arquivo filaD.c
Fila *cria_fila(){
    Fila *fi = (Fila*) malloc(sizeof(Fila));
    if(fi != NULL){
        fi->fim = NULL;
        fi->inicio = NULL;
    }
    return fi;
}
```

```c
//Arquivo main.c
fi = cria_fila();
```

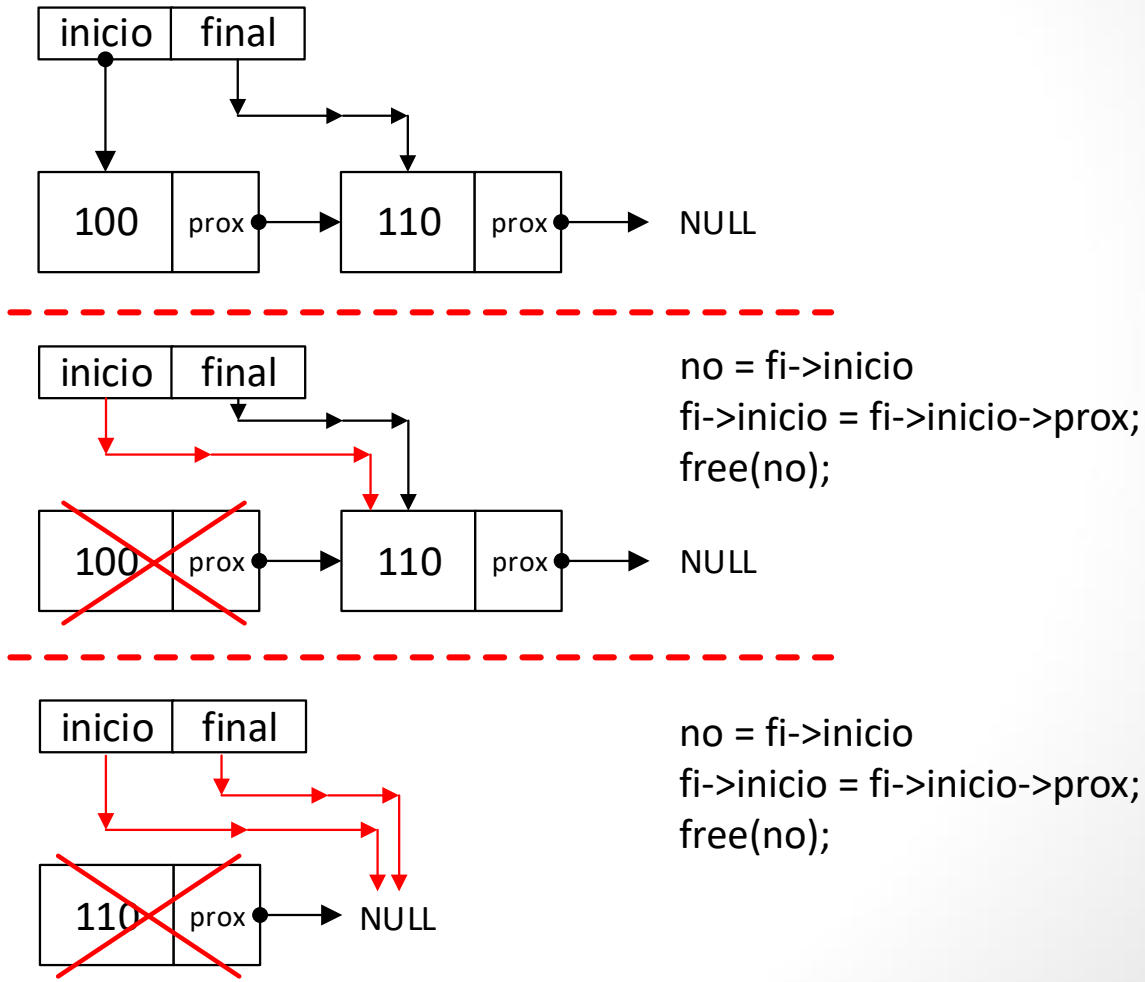


Fila Dinâmica - Destruindo a Fila

```c
//Arquivo filaD.h
void destroi_fila(Fila *fi);
```

```c
//Arquivo filaD.c
void destroi_fila(Fila *fi){
    if(fi == NULL){
        Elem *no;
        while(fi->inicio != NULL){
            no = fi->inicio;
            fi->inicio = fi->inicio->prox;
            free(no);
        }
        free(fi);
    }
}
```

```c
//Arquivo main.c
destroi_fila(fi);
```



Fila Dinâmica Fila

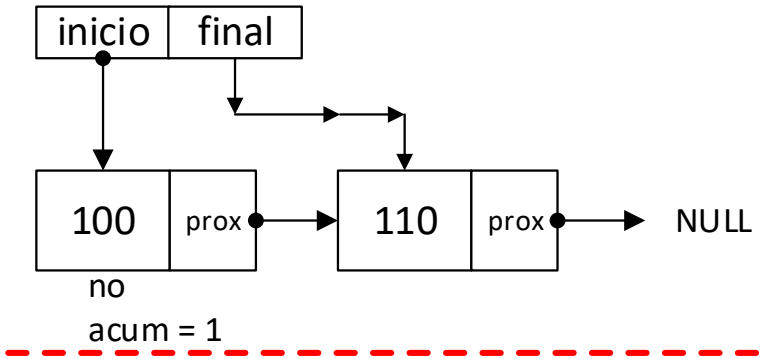
- Informações básicas sobre a Fila:
  - Tamanho;
  - Se está cheia;
  - Se está vazia.

```c
//Arquivo filaD.h
int tamanho_fila(Fila *fi);
```

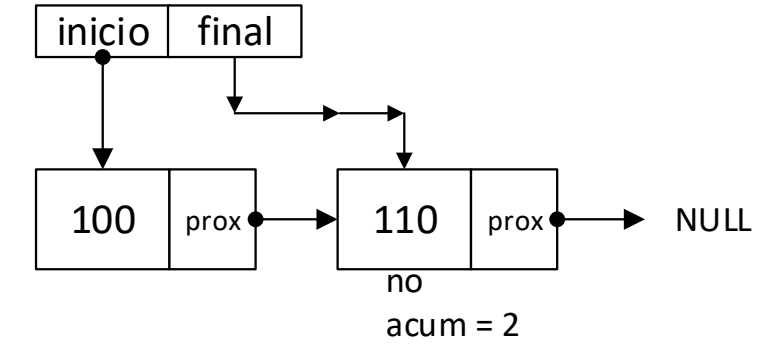
```c
//Arquivo main.c
x = tamanho_fila(fi);
printf("\nO tamanho da fila e: %d", x);
```

```c
//Arquivo filaD.c
int tamanho_fila(Fila *fi){
    if(fi == NULL){
        return 0;
    }
    int acum = 0;
    Elem *no = fi->inicio;
    while(no != NULL){
        acum++;
        no = no->prox;
    }
    return acum;
}
```

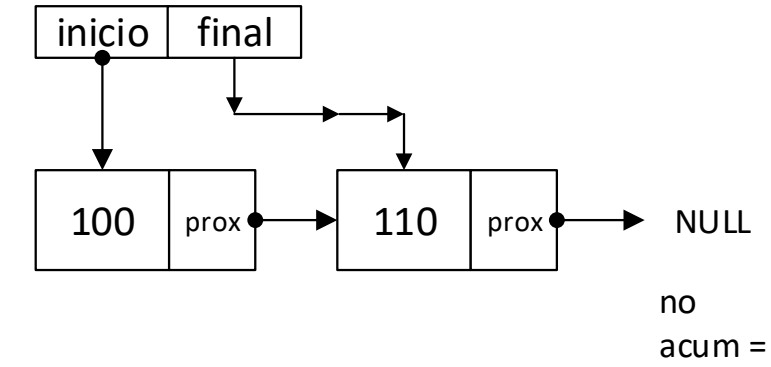
Fila Dinâmica - Tamanho




```
acum++;
no = no->prox;
```




```
acum++;
no = no->prox;
```




```
Fim:
No = NULL
```

Fila Dinâmica - Fila Cheia

- Novamente, em estruturas alocadas dinamicamente não faz sentido o conceito de “estruturas cheias”. Mantêm-se a função por uma questão de padronização.

```c
//Arquivo filaD.h
int fila_cheia(Fila *fi);
```

```c
//Arquivo filaD.c
int fila_cheia(Fila *fi){
    return 0;
}
```

```c
//Arquivo main.c
x = fila_cheia(fi);
if(x){
    printf("\nA fila está cheia!");
}else{
    printf("\nA fila não está cheia.");
}
```

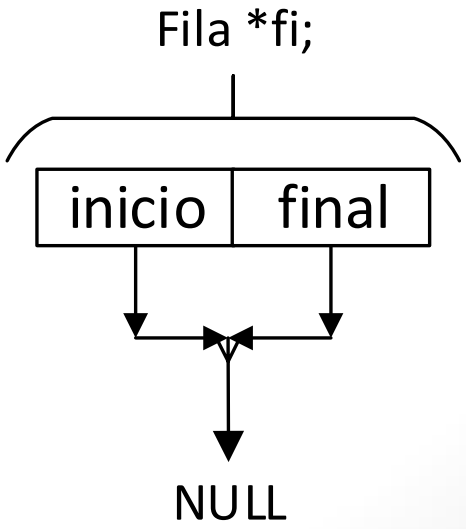
Fila Dinâmica - Fila Vazia

```c
//Arquivo filaD.h
int fila_vazia(Fila *fi);
```

```c
//Arquivo main.c
x = fila_vazia(fi);
if(x){
    printf("\nA fila está vazia!");
}else{
    printf("\nA fila não está vazia.");
}
```

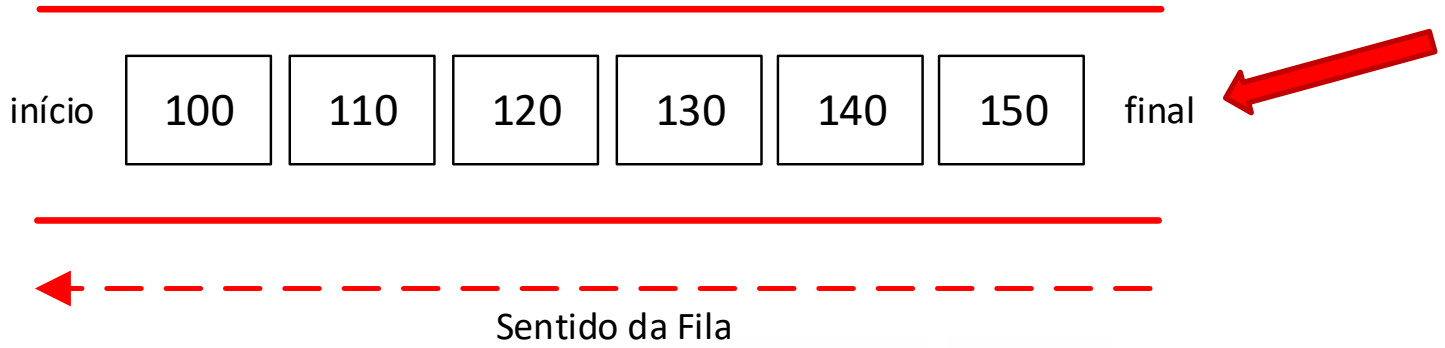
```c
//Arquivo filaD.c
int fila_vazia(Fila *fi){
    if(fi == NULL){
        return 1;
    }
    if(fi->inicio == NULL){
        return 1;
    }
    return 0;
}
```

Não é necessário verificar o final da lista



Fila Dinâmica - Inserção

- Em uma Fila a inserção é sempre no seu final;
- Também existe o caso onde a inserção é feita em uma fila que está vazia:



Fila Dinâmica - Inserção

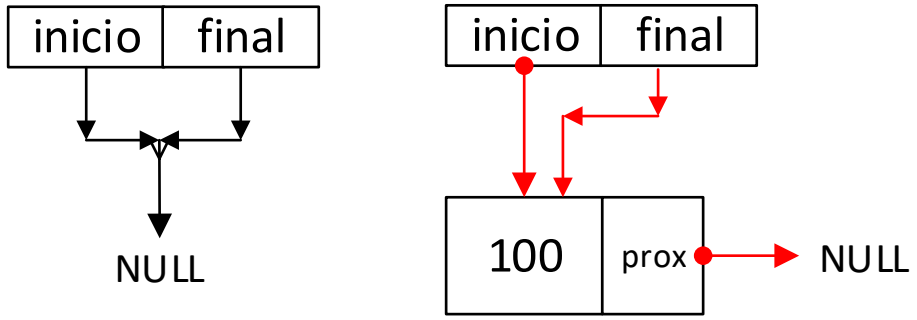
```c
//Arquivo main.c
x = insere_fila(fi, al1);
if(x){
    printf("\nElemento inserido com sucesso!");
}else{
    printf("\nErro, elemento nao inserido.");
}
x = insere_fila(fi, al2);
if(x){
    printf("\nElemento inserido com sucesso!");
}else{
    printf("\nErro, elemento nao inserido.");
}
x = insere_fila(fi, al3);
if(x){
    printf("\nElemento inserido com sucesso!");
}else{
    printf("\nErro, elemento nao inserido.");
}
```

```c
//Arquivo filaD.h
int insere_fila(Fila *fi, ALUNO al);
```

Fila Dinâmica - Inserção

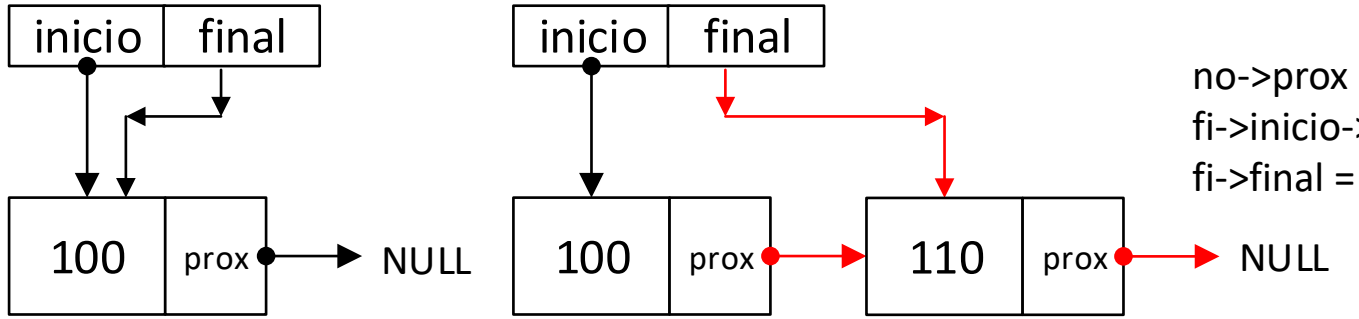
```c
//Arquivo filaD.c
int insere_fila(Fila *fi, ALUNO al){
    if(fi == NULL){
        return 0;
    }
    Elem *no = (Elem*) malloc(sizeof(Elem));
    if(no == NULL){
        return 0;
    }
    no->dados = al;
    no->prox = NULL;
    if(fi->fim == NULL){//fila vazia
        fi->inicio = no;
    }else{ //insere no final da fila
        fi->fim->prox = no;
    }
    fi->fim = no; //no passa a ser novo final da fila
    return 1;
}
```

Fila Dinâmica - Inserção



Inserção em Fila vazia:

```
no->prox = NULL;
fi->inicio = no;
fi->final = no;
```

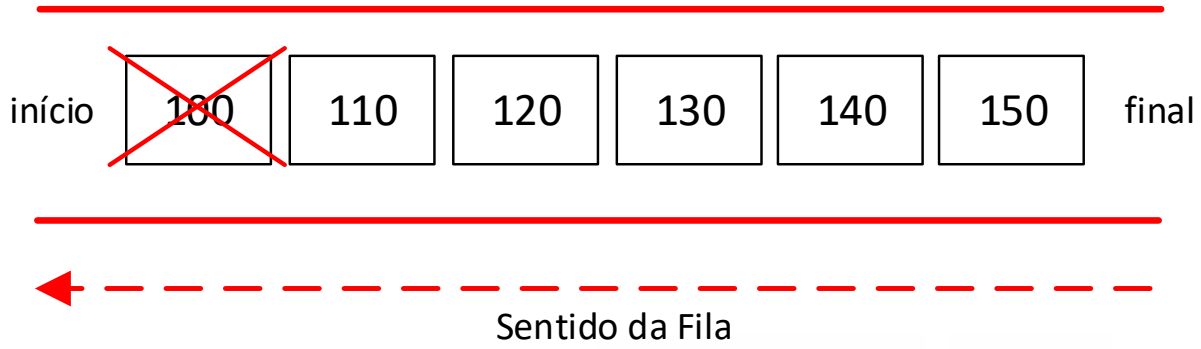


Inserção em Fila não vazia:

```
no->prox = NULL;
fi->inicio->prox = no;
fi->final = no;
```

Fila Dinâmica - Remoção

- Em uma Fila a remoção é sempre no seu início.
- Cuidado: não se pode remover de uma Fila vazia!



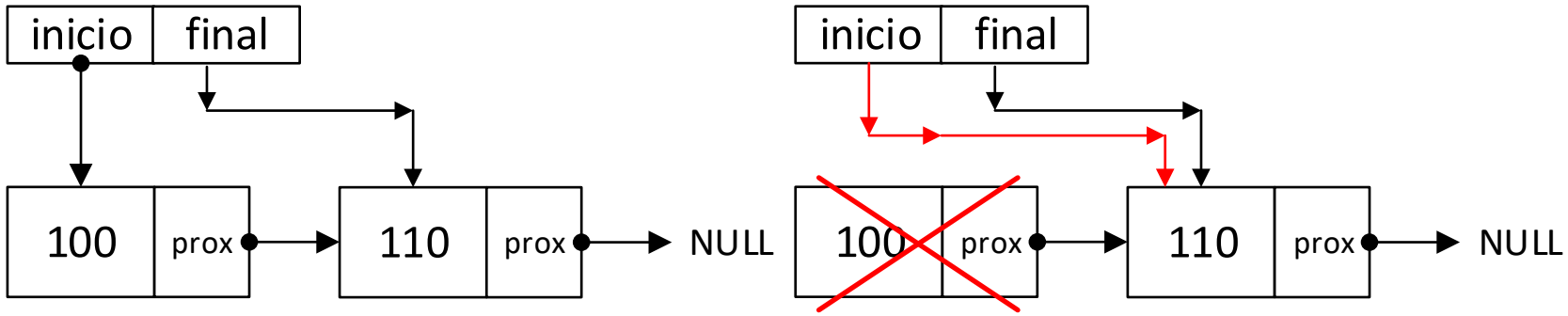
Fila Dinâmica - Remoção

```c
//Arquivo filaD.h
int remove_fila(Fila *fi);
```

```c
//Arquivo main.c
x = remove_fila(fi);
if(x){
    printf("\nElemento removido com sucesso!");
}else{
    printf("\nErro, elemento nao removido.");
}
```

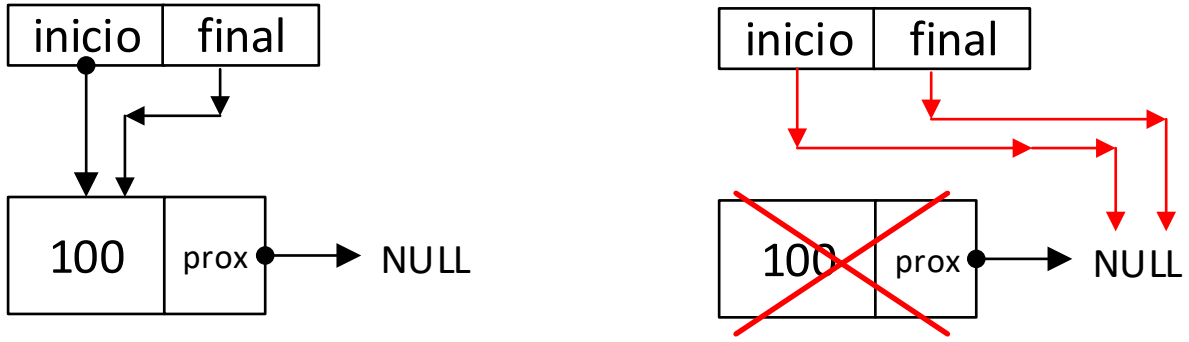
```c
//Arquivo filaD.c
int remove_fila(Fila *fi){
    if(fi == NULL){
        return 0;
    }
    if(fi->inicio == NULL){ //fila vazia
        return 0;
    }
    Elem *no = fi->inicio;
    fi->inicio = fi->inicio->prox;
    if(fi->inicio == NULL){//fila ficou vazia
        fi->fim = NULL;
    }
    free(no);
    return 1;
}
```

Fila Dinâmica - Remoção



Remoção em Fila não vazia:

```
*no = fi->inicio;
fi->inicio = fi->inicio->prox;
free(no);
```

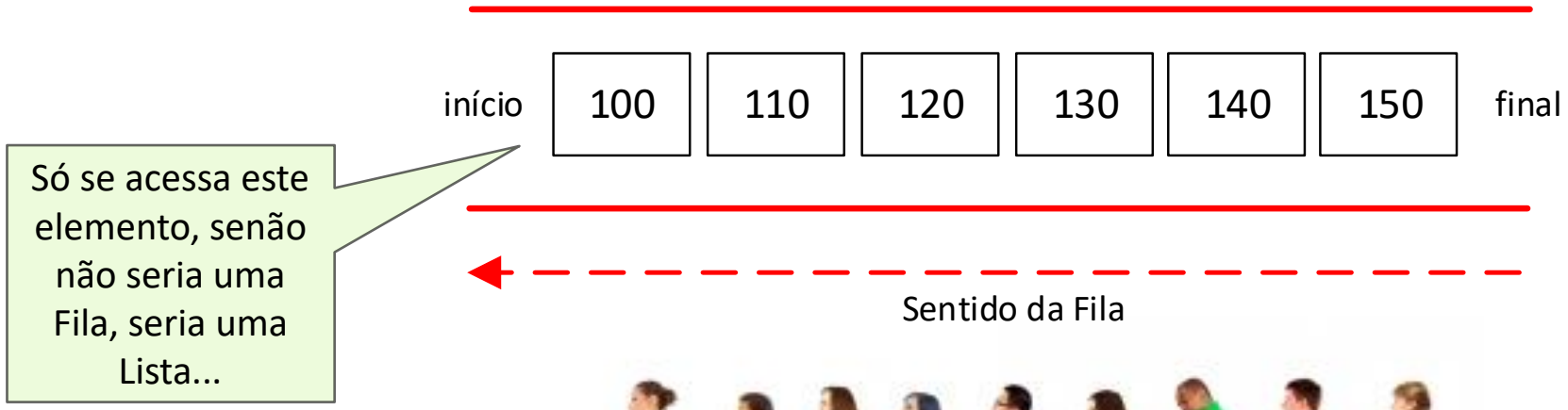


Fila fica vazia:

```
*no = fi->inicio;
fi->inicio = fi->inicio->prox;
fi->final = NULL;
free(no);
```

Fila Dinâmica - Consulta

- Em uma Fila a consulta se dá apenas ao elemento que está no seu início:

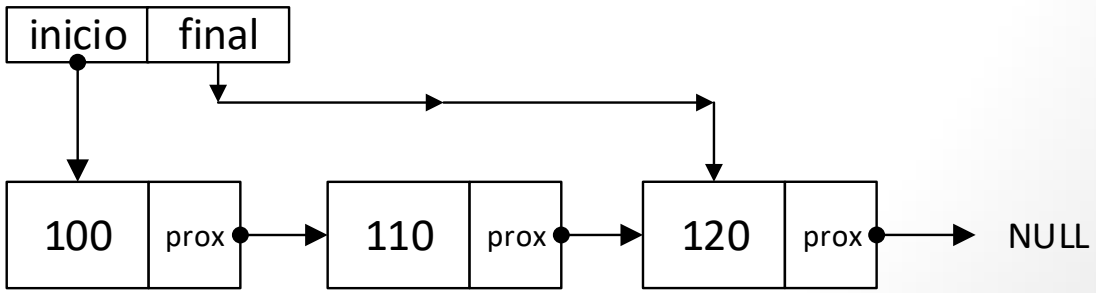


Fila Dinâmica - Consulta

```c
//Arquivo filaD.h
int consulta_fila(Fila *fi, ALUNO *al);
```

```c
//Arquivo filaD.c
int consulta_fila(Fila *fi, ALUNO *al){
    if(fi == NULL){
        return 0;
    }
    if(fi->inicio == NULL){//fila vazia
        return 0;
    }
    *al = fi->inicio->dados;
    return 1;
}
```

```c
//Arquivo main.c
x = consulta_fila(fi, &al);
if(x){
    printf("\nConsulta realizada com sucesso:");
    printf("\nMatricula: %d", al.matricula);
    printf("\nNota 1:    %.2f", al.n1);
    printf("\nNota 2:    %.2f", al.n2);
    printf("\nNota 3:    %.2f", al.n3);
}else{
    printf("\nErro, consulta nao realizada.");
}
```



*al = fi->inicio->dados;

Estrutura de Dados 1

Atividade 1

- Entregue os arquivos no Moodle como “atividade 1 - Fila”