

ESTRUTURA DE DADOS: FILAS

Prof. Jean Nunes

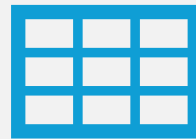
FILA



1 – Definição e Características



2 – Fila Dinâmica

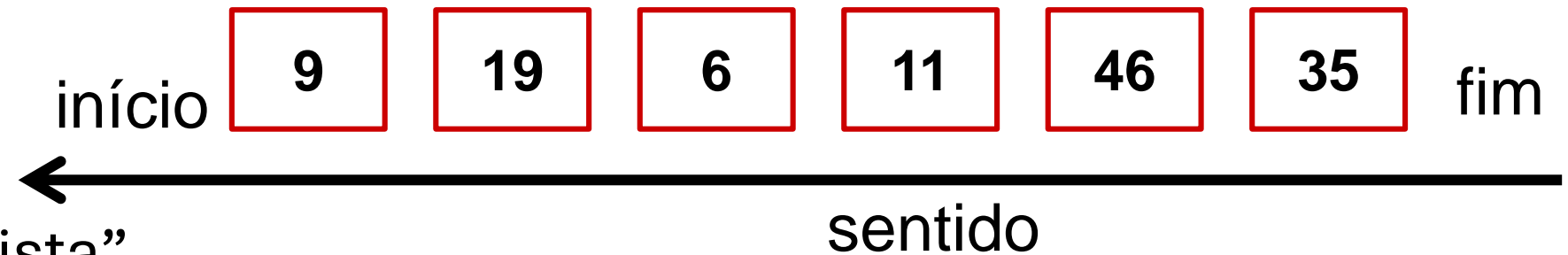


3 – Fila com Prioridades

FILA

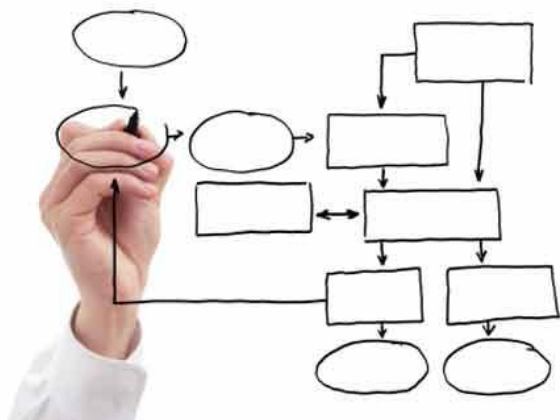
Uma estrutura do tipo “Fila” é uma sequência de elementos do mesmo tipo, como as “Listas”.

**First-in, First-out
(FIFO)**



- Tipo especial de “lista”.
- Inserções e remoções ocorrem nas extremidades.





Controle de fluxo

Atendimento Preferencial e Prioritário para:

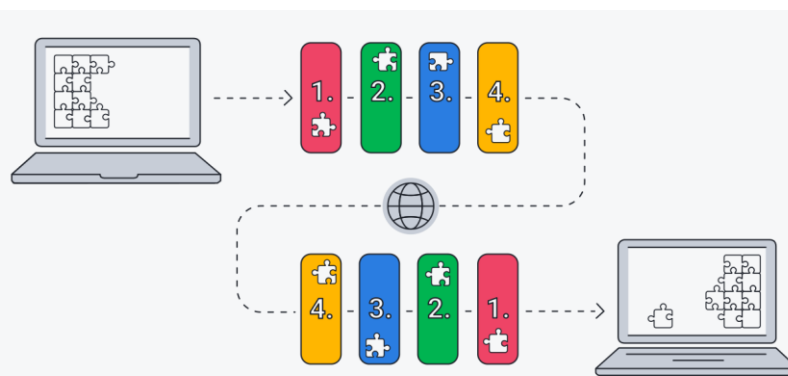


- idosos acima de 60 anos - Lei Federal 10.741/03;
- gestantes;
- pessoas com deficiência;
- pessoas com criança de colo.

Prioridades



Recursos compartilhados



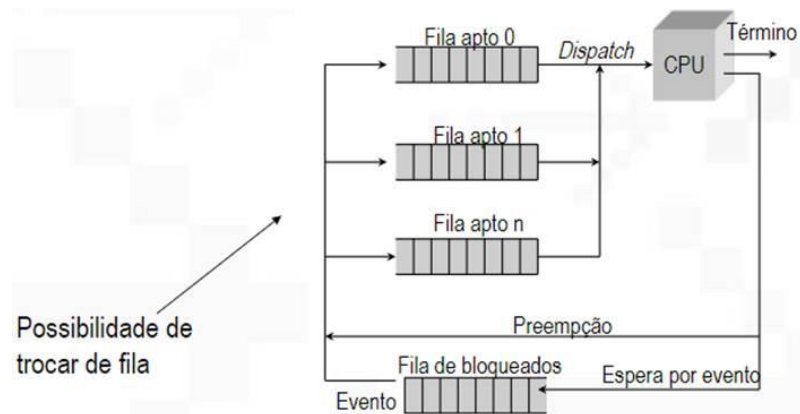
Protocolos de Internet (TCP/SMTP)

FILA

Aplicações



Processamento de pedidos



CPU

FILA

Operações básicas



Criar fila



Inserir um elemento no **final da fila**



Excluir um elemento no **início da fila**



Acessar o elemento do **início da fila**

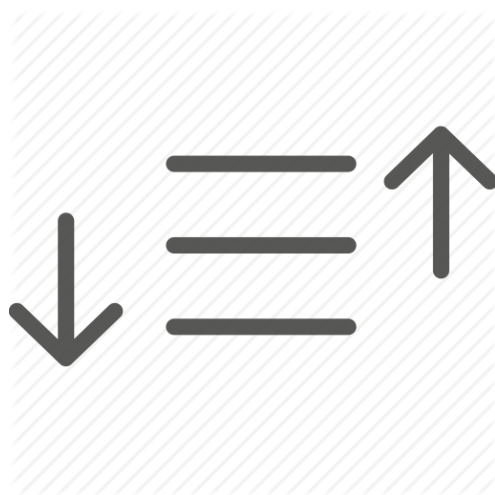


Destruir fila

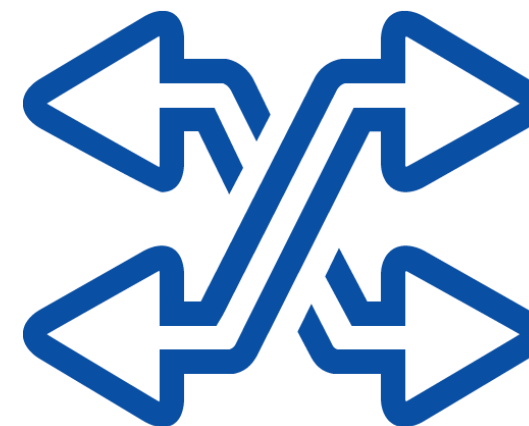


FILA

Essas operações dependem do tipo de alocação de memória usada



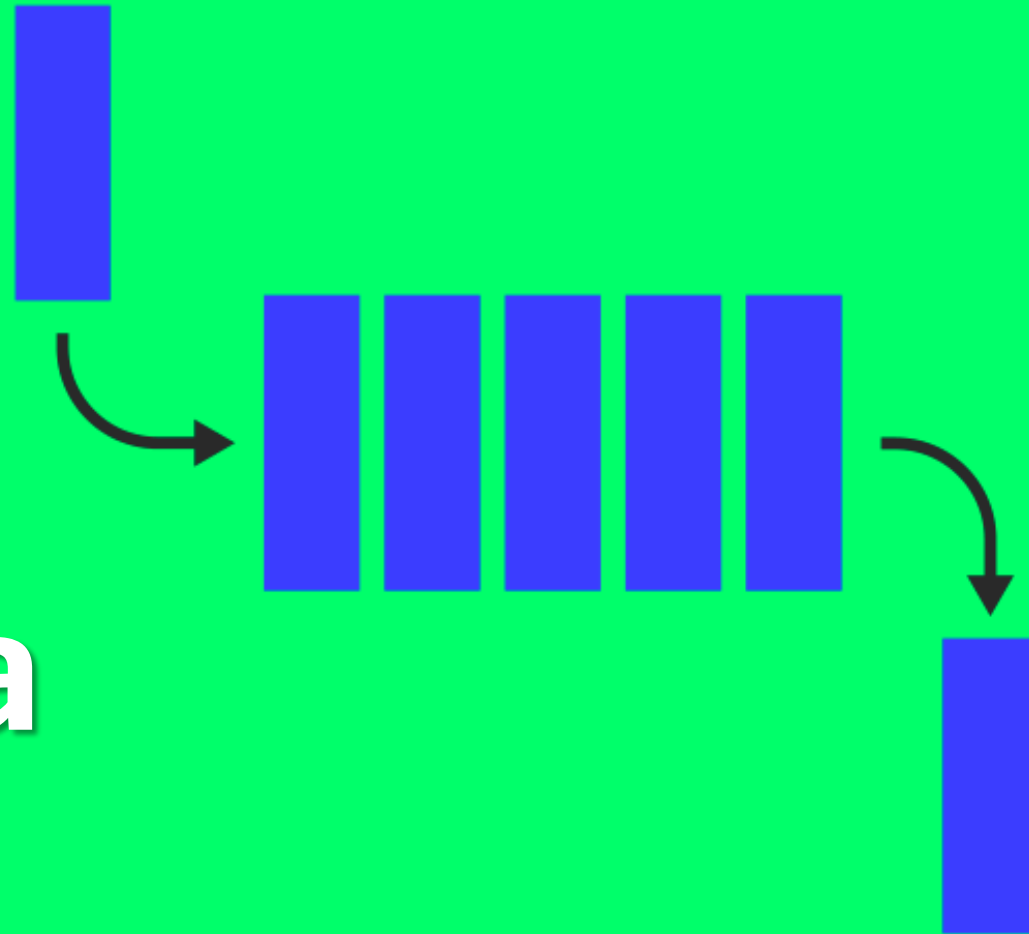
Estática



Dinâmica

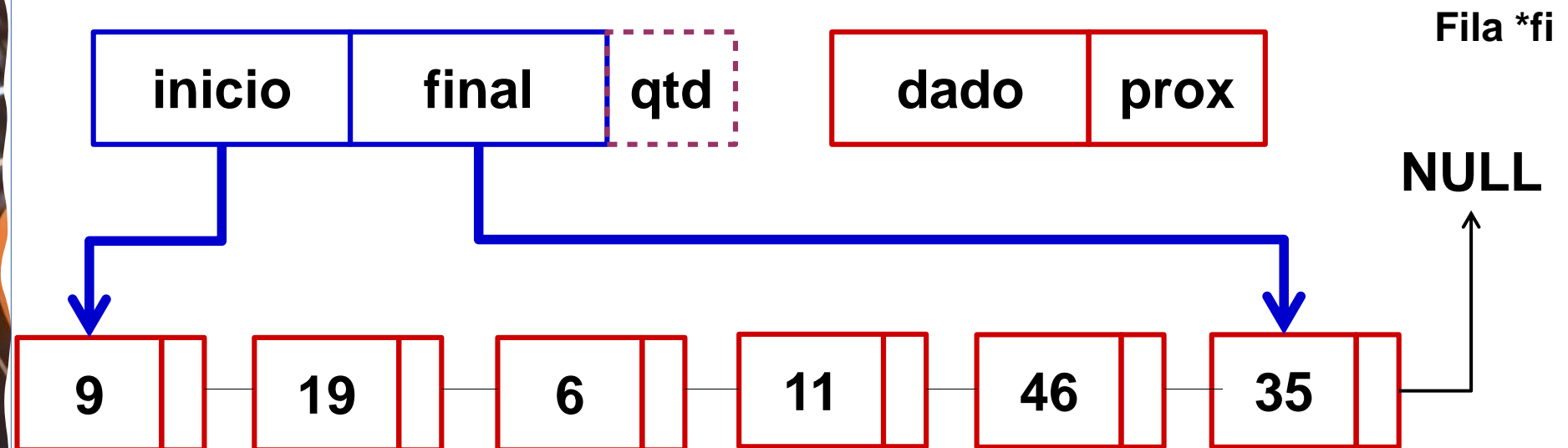
FILA

Dinâmica



FILE DINÂMICA

- Tipo de fila onde cada elemento aponta para o seu sucessor na “fila”.
- Usa um “nó descritor” para representar o início e o final da fila e uma indicação de final de fila (NULL).





FILE DINÂMICA

//Arquivo FilaDinamica.h

- Protótipos das funções
- O tipo de dado armazenado na lista
- O ponteiro “fila”

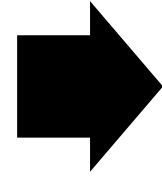
//Arquivo FilaDinamica.c

- O tipo de dado “fila”
- Implementa as suas funções

//main.c

- Interface com o usuário

FILA DINÂMICA



Implementando

//Arquivo FilaDinamica.h

```
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,media;
};
typedef struct fila Fila;
```

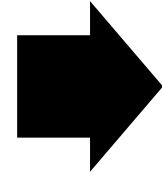
// FilaDinamica.c

```
#include <stdio.h>
#include <stdlib.h>
#include "FilaDinamica.h"

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

struct elemento{
    struct aluno dados;
    struct elemento *prox;
};
typedef struct elemento Elem;
```

FILA DINÂMICA



Implementando

// FilaDinamica.h

```
Fila* cria_fila();
```

//main.c

```
#include "FilaDinamica.h"
```

```
Fila *fi = cria_fila();
```

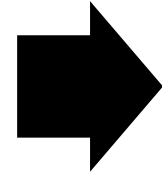
Fila *fi;



// FilaDinamica.c

```
Fila* cria_fila(){  
    Fila* fi = (Fila*) malloc(sizeof(Fila));  
    if(fi != NULL){  
        fi->final = NULL;  
        fi->inicio = NULL;  
    }  
    return fi;  
}
```

FILA DINÂMICA



Implementando

LIBERA FILA

// FilaDinamica.h

```
void libera_fila(Fila* fi);
```

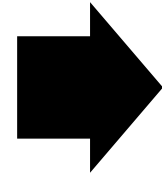
//main.c

```
libera_fila(fi);
```

//Arquivo FilaDinamica.c

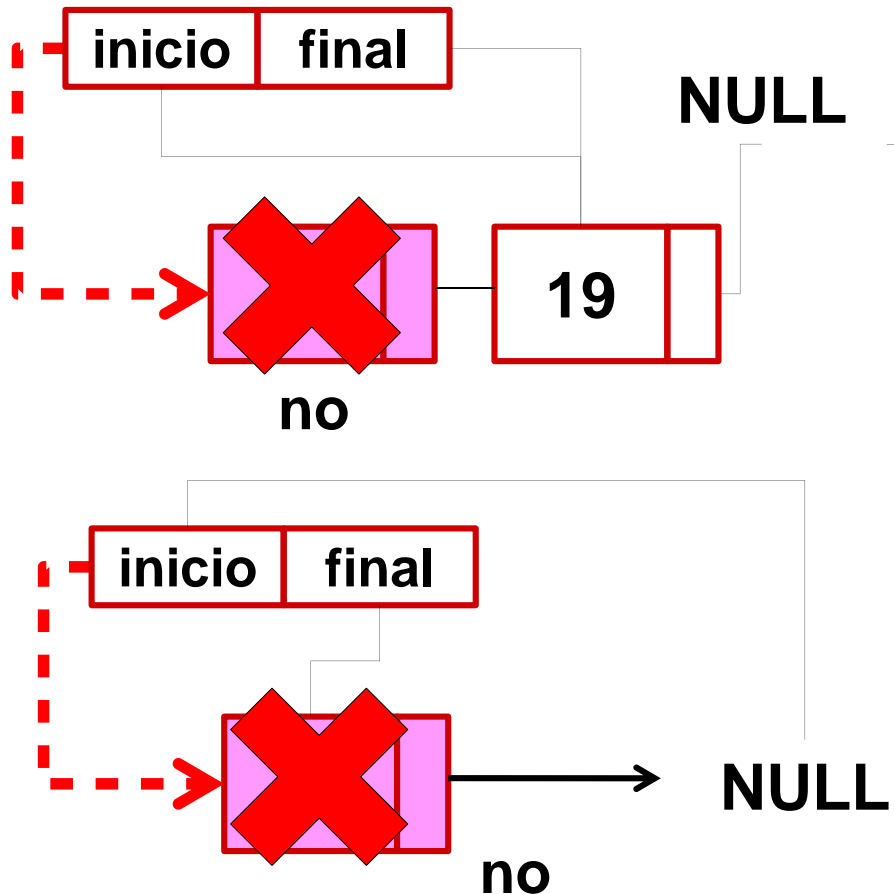
```
void libera_fila(Fila* fi){  
    if(fi != NULL){  
        Elem* no;  
        while(fi->inicio != NULL){  
            no = fi->inicio;  
            fi->inicio = fi->inicio->prox;  
            free(no);  
        }  
        free(fi);  
    }  
}
```

FILA DINÂMICA



Implementando

LIBERA FILA



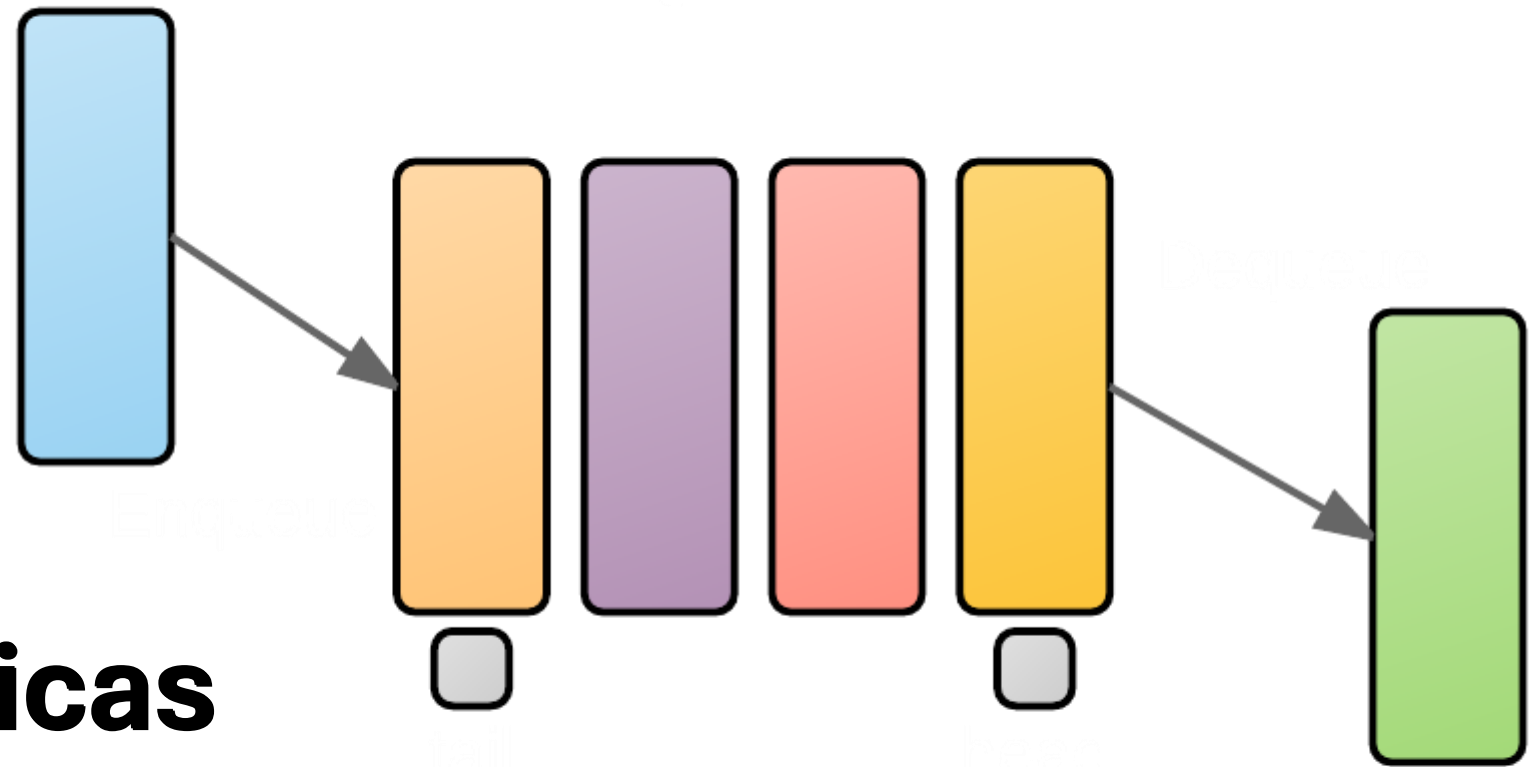
```
while(fi->inicio != NULL){  
    no = fi->inicio;  
    fi->inicio = fi->inicio->prox;  
    free(no);  
}  
free(fi);
```

*fi

inicio	final
--------	-------

FILA DINÂMICA

Informações básicas



Tamanho

Fila vazia

FILA DINÂMICA

Informações básicas

Tamanho

//FilaDinamica.h

```
int tamanho_fila(Fila* fi);
```

//main.c

```
int x = tamanho_fila(fi);
```

Se quiser, posso inserir o campo quantidade e incrementá-lo ao inserir um novo elemento.

//FilaDinamica.c

```
int tamanho_fila(Fila* fi){  
    if(fi == NULL) return 0;  
    int cont = 0;  
    Elem* no = fi->inicio;  
    while(no != NULL){  
        cont++;  
        no = no->prox;  
    }  
    return cont;  
}
```


FILA DINÂMICA

Informações básicas

Fila vazia

// FilaDinamica.h
int fila_vazia(Fila* fi);

//main.c
if(fila_vazia(fi))

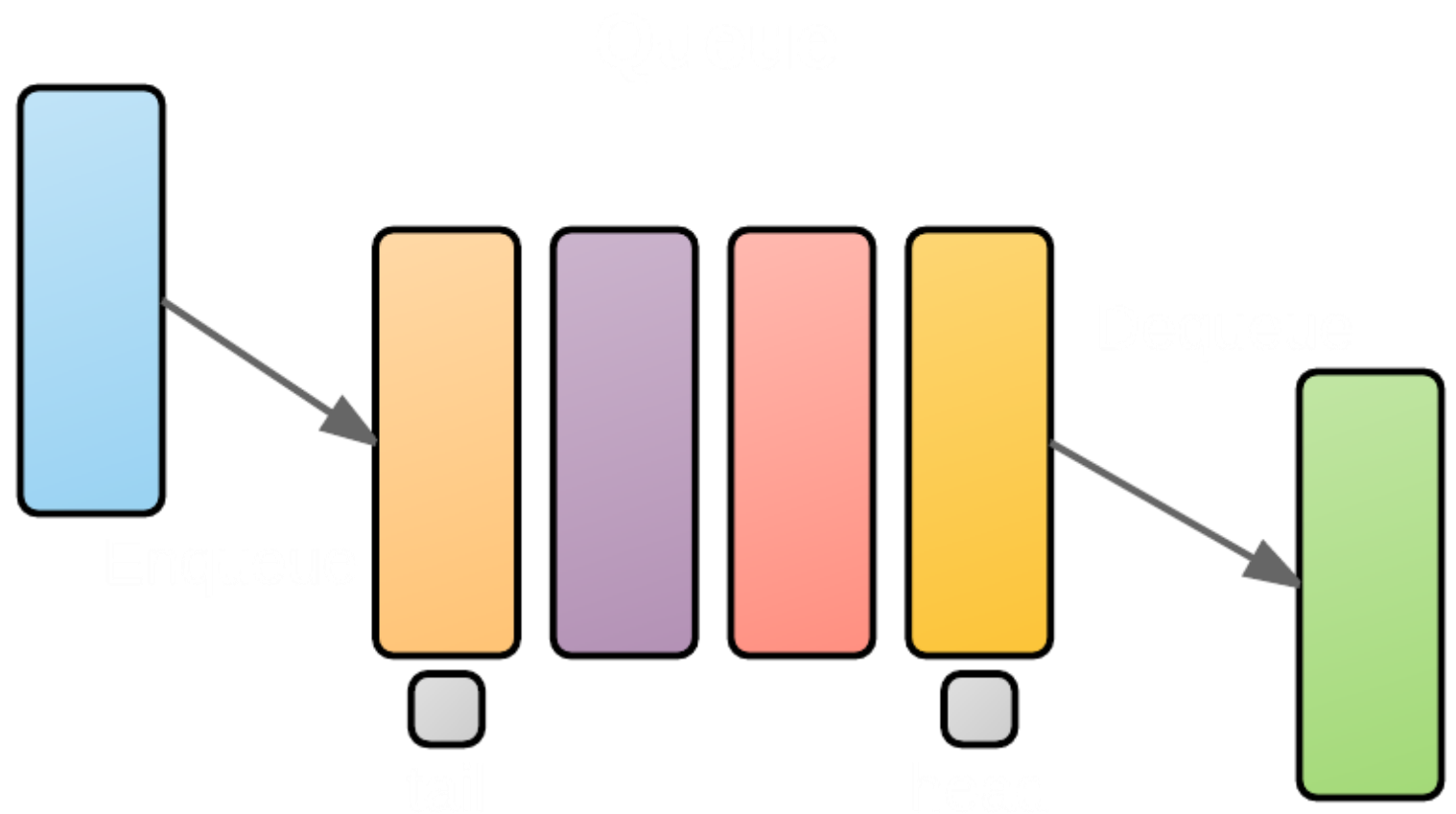
Fila *fi;



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

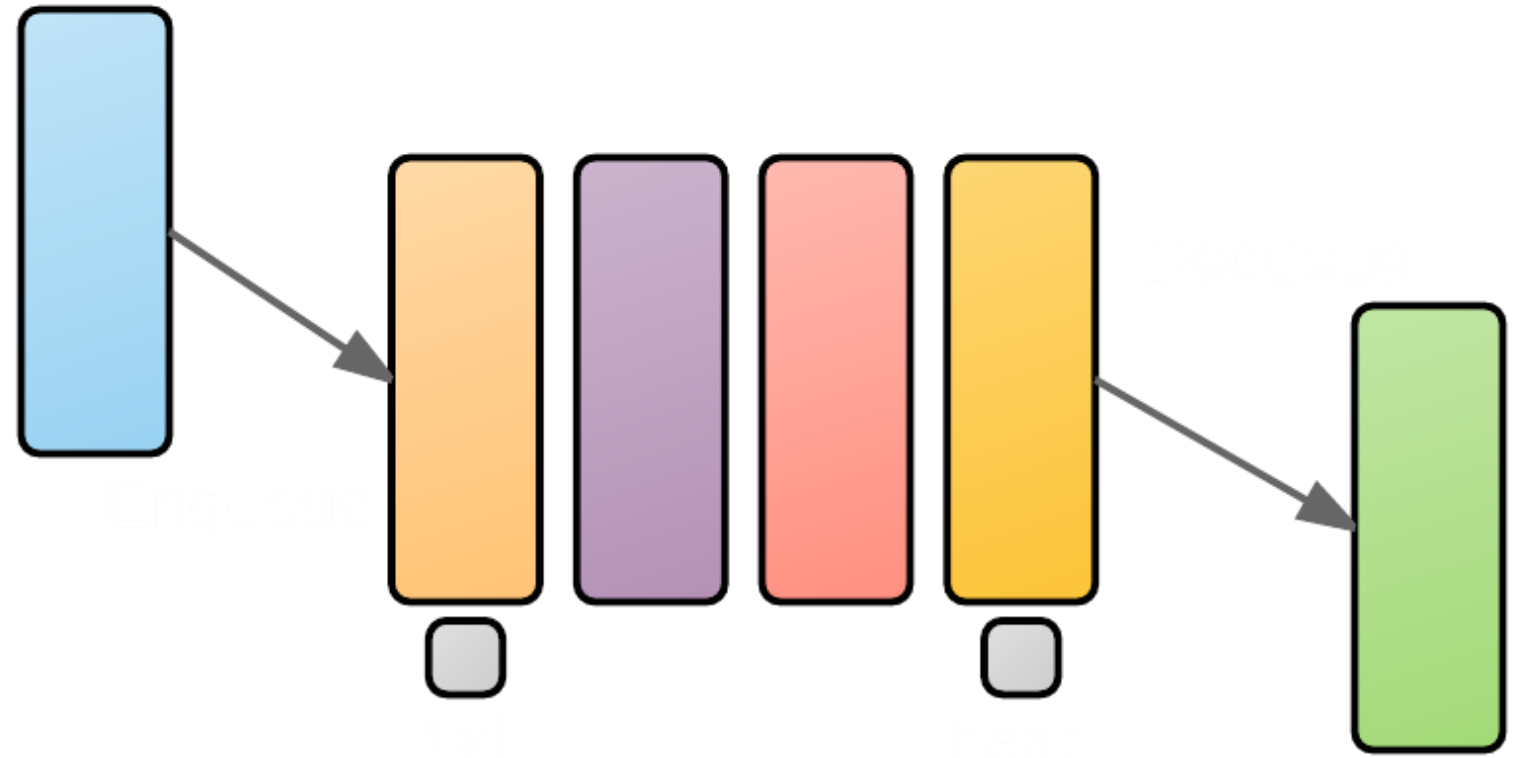
FILA DINÂMICA

Inserção



FILA DINÂMICA

Inserção



- Na “fila” a inserção é sempre no seu final, ou em fila vazia.
- Não se pode inserir em fila cheia.

FILA DINÂMICA

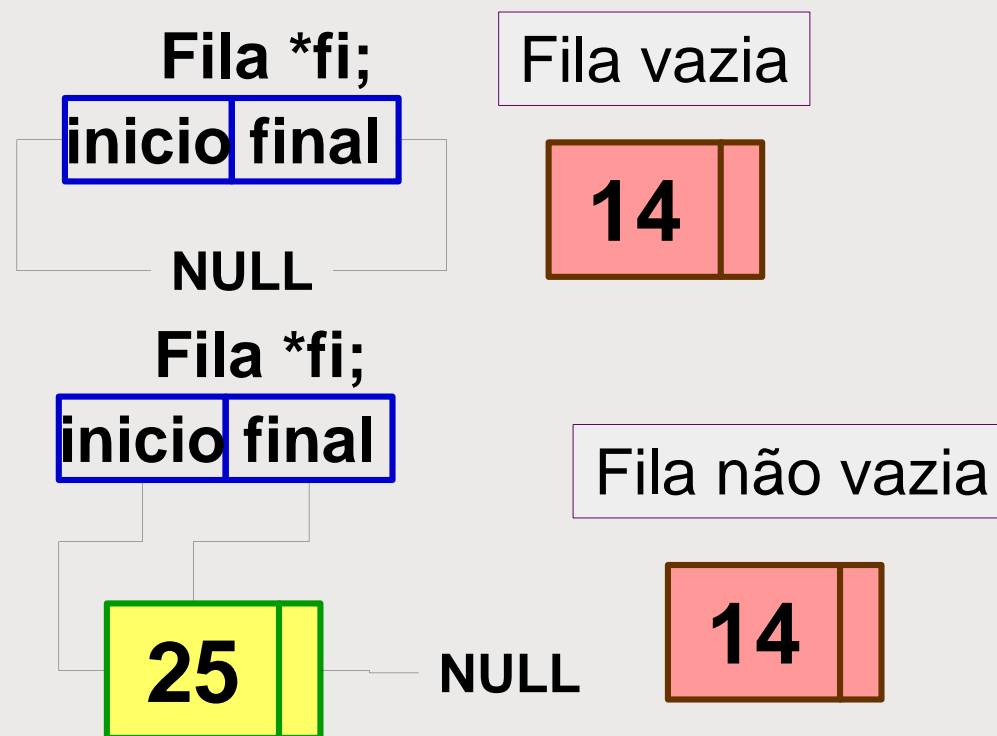
Inserção

//Arquivo FilaDinamica.h

```
int insere_fila(Fila* fi, struct aluno al);
```

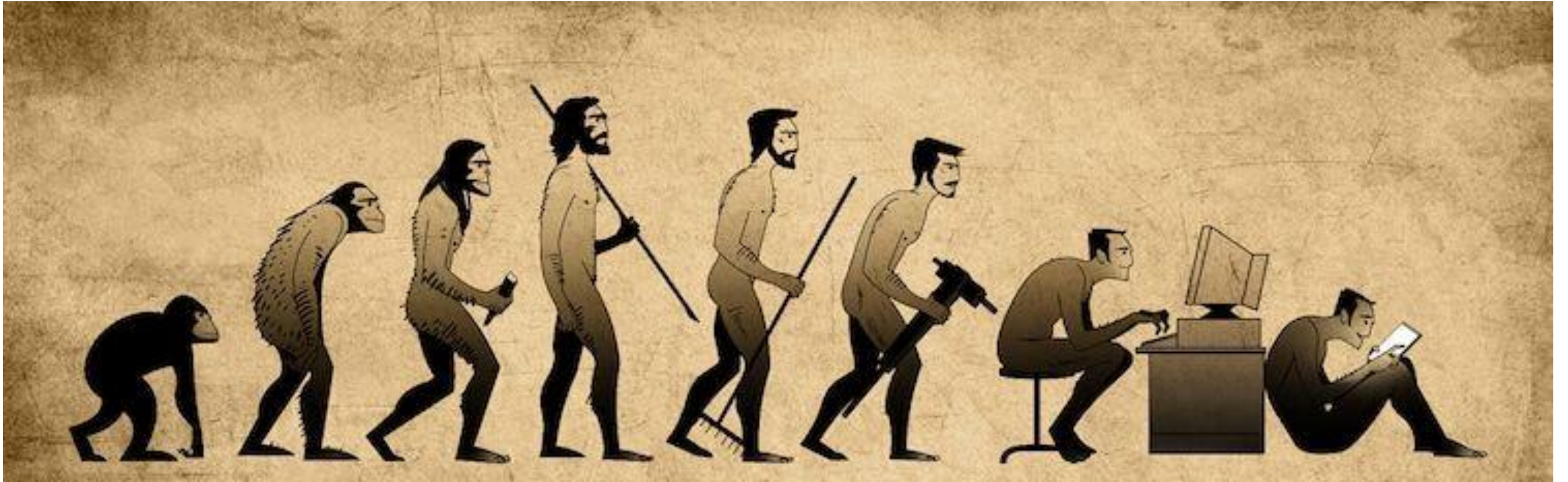
//main.c

```
int x = insere_fila(fi, <dados_aluno>);
```



//Arquivo FilaDinamica.c

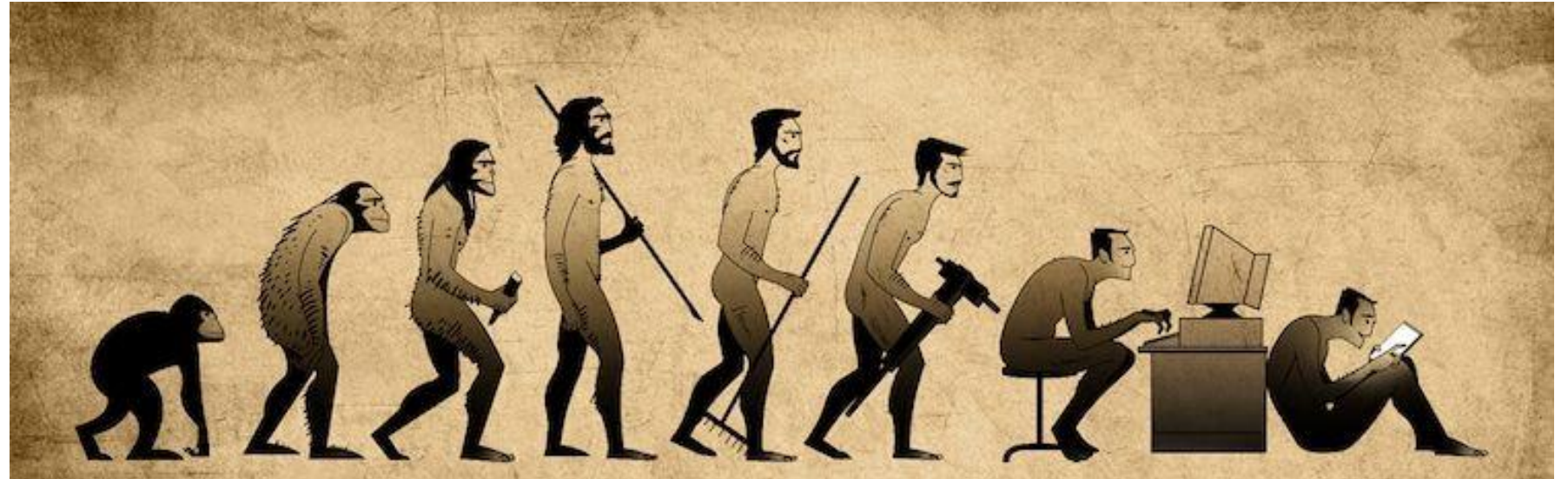
```
int insere_fila(Fila* fi, struct aluno al)
{
    if(fi == NULL) return 0;
    Elem* no = (Elem*) malloc(sizeof(Elem));
    if(no == NULL) return 0;
    no->dados = al; // qualquer tipo de dado
    no->prox = NULL;
    if (fi->final == NULL) // vazia
        fi->inicio = no;
    else fi->final->prox = no;
    fi->final = no;
    fi->qtd++;
    return 1;
}
```



FILA DINÂMICA

Remoção

FILA DINÂMICA Remoção



- Na “fila” a remoção é sempre no seu início.
- Não se pode remover em fila vazia.

Arquivo FilaDinamica.c

```
int remove_fila(Fila* fi)
{
    if(fi == NULL) return 0;
    if(fi->inicio == NULL) return 0; // vazia
    Elem *no = fi->inicio;
    fi->inicio = fi->inicio->prox;
    no->prox = NULL;
    if (fi->inicio == NULL) // ficou vazia
        fi->final = NULL;
    free(no);
    fi->qtd--;
    return 1;
}
```

Fila não fica vazia



NULL

fica vazia



NULL

1. FILAS E LISTAS (AP1)

a. Crie um programa em C utilizando a estrutura de dados FILA para gerenciar filas de espera de transplantes dos seguintes tipos:

- CORAÇÃO
- CÓRNEA
- MEDULA.

b. Usando a aplicação acima, adapte uma LISTA DINÂMICA ENCADEADA para funcionar como FILA, em que a inserção é feita no final e a remoção é feita no início (**ATENÇÃO: na LISTA não deve ter ponteiro para o final**).

c. Faça demonstração/simulação de 100 inserções e 40 remoções aleatórias na FILA e na LISTA (a FILA/LISTA escolhida também deverá ser aleatória).

d. **Guarde os tempos de inserção/remoção em LISTAS.**

e. **Calcule os tempos médios de inserção/remoção e compare a eficiência das duas estruturas.**

