



Ciência da Computação  
Algoritmos e Estrutura de Dados 1

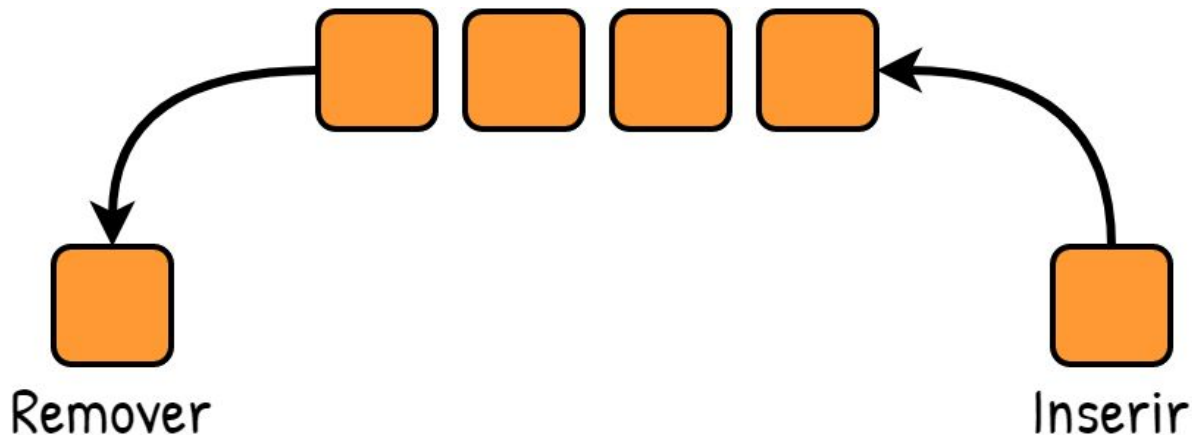
# Fila

---

# Definição

Lista linear em que a operação de **inserção** ocorre em uma extremidade e a **remoção** na outra extremidade

**FIFO**: First In First Out



# Aplicações

- Recursos compartilhados
- Algoritmos em grafos

# TAD Fila

## Dados

```
typedef int TipoElemento;  
typedef struct fila Fila;
```

## Operações

```
Fila* fila_criar();  
void fila_destruir(Fila** enderecoFila);
```

→ CRIAR E DESTRUIR A FILA

```
bool fila_inserir(Fila* f, TipoElemento elemento);  
bool fila_remover(Fila* f, TipoElemento* saida);
```

→ INSERIR E REMOVER UM ELEMENTO NA FILA

```
bool fila_primeiro(Fila* f, TipoElemento* saida);
```

→ ACESSA E DEVOLVE O ELEMENTO QUE ESTÁ NO TOPO DA FILA

```
bool fila_vazia(Fila* f);
```

→ VERIFICA SE A FILA ESTÁ VAZIA

```
void fila_toString(Fila* f, char* str);
```

→ DEVOLVE UMA STRING QUE REPRESENTA A FILA

# TAD Fila

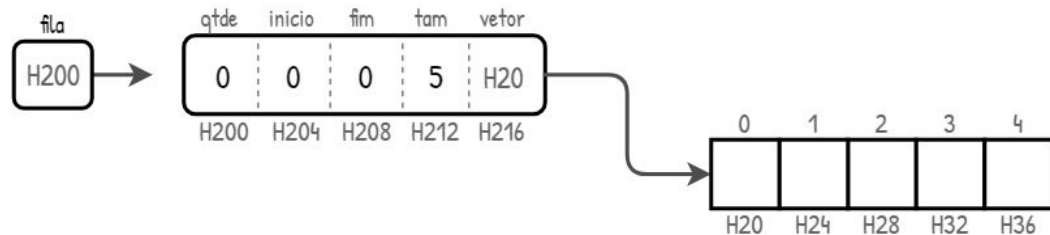
```
typedef int TipoElemento;
/*****
* DADOS
*****/
typedef struct fila Fila;

/*****
* PROTÓTIPOS
*****/
Fila* fila_criar();
void fila_destruir(Fila** enderecoFila);
bool fila_inserir(Fila* f, TipoElemento elemento);
bool fila_remove(Fila* f, TipoElemento* saida);
bool fila_primeiro(Fila* f, TipoElemento* saida);
bool fila_vazia(Fila* f);
void fila_toString(Fila* f, char* saida);
```

# Organização dos dados na memória

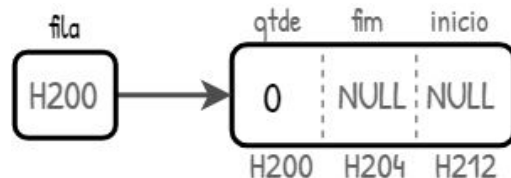
## Contígua

```
struct fila {  
    TipoElemento* vetor;  
    int tam;  
    int inicio;  
    int fim;  
    int qtde;  
};
```



## Encadeada

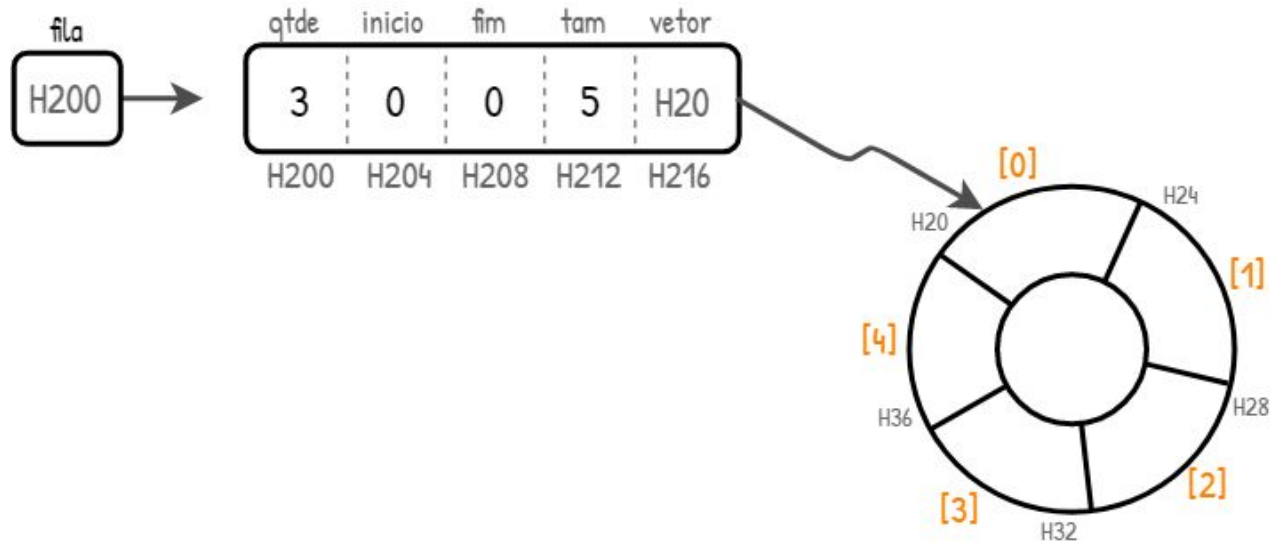
```
typedef struct no{  
    int dado;  
    struct no* prox;  
}No;  
  
struct fila {  
    No* inicio;  
    No* fim;  
    int qtde;  
};
```



# Organização Contígua

## Especificação da struct

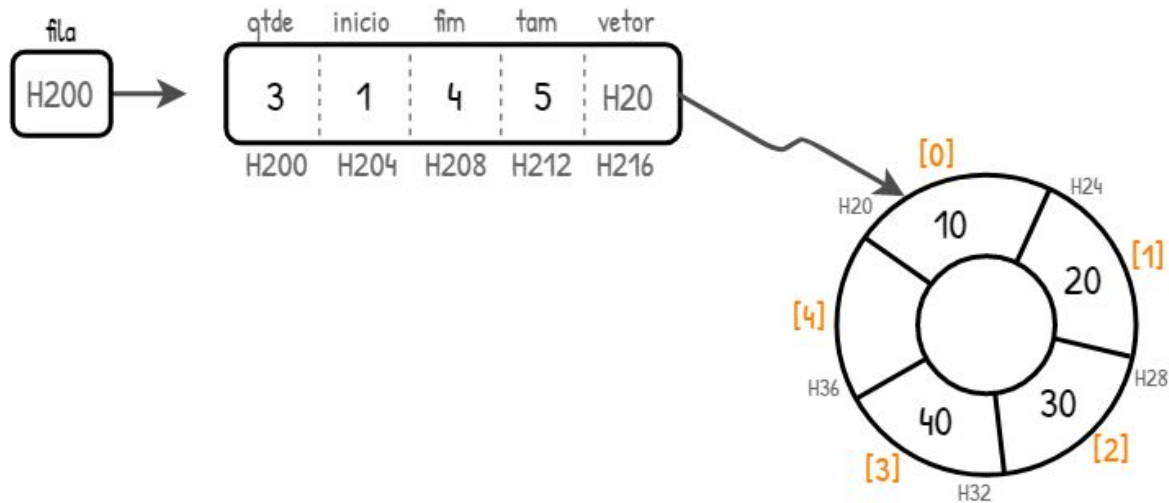
```
struct fila {  
    TipoElemento* vetor;  
    int tam;  
    int inicio;  
    int fim;  
    int qtde;  
};
```



# Organização Contígua

```
#include "fila.h"
/*****
* DADOS
*****/
struct fila {
    TipoElemento* vetor;
    int tam;
    int inicio;
    int fim;
    int qtde;
};

/*****
* IMPLEMENTAÇÃO
*****/
Fila* fila_criar();
void fila_destruir(Fila** enderecoFila);
bool fila_inserir(Fila* f, TipoElemento elemento);
bool fila_remover(Fila* f, TipoElemento* saida);
bool fila_primeiro(Fila* f, TipoElemento* saida);
bool fila_vazia(Fila* f);
void fila_toString(Fila* f, char* saida);
```



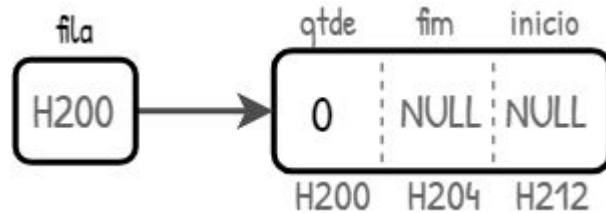


# Organização Encadeada

## Especificação da struct

```
typedef struct no{
    int dado;
    struct no* prox;
}No;

struct fila {
    No* inicio;
    No* fim;
    int qtde;
};
```



# Organização Encadeada

```
#include "fila.h"
/*****
 * DADOS
 *****/
typedef struct no{
    int dado;
    struct no* prox;
}No;
struct fila {
    No* inicio;
    No* fim;
    int qtde;
};
/*****
 * IMPLEMENTAÇÃO
 *****/
Fila* fila_criar();
void fila_destruir(Fila** enderecoFila);
bool fila_inserir(Fila* f, TipoElemento elemento);
bool fila_remover(Fila* f, TipoElemento* saida);
bool fila_primeiro(Fila* f, TipoElemento* saida);
bool fila_vazia(Fila* f);
void fila_toString(Fila* f, char* saida);
```

