

Tecnologia em Análise e Desenvolvimento de Sistemas - TADS

Estrutura de Dados

Prof. Luciano Vargas Gonçalves

E-mail: luciano.goncalves@riogrande.ifrs.edu.br



Sumário

Estrutura de Dados

- Introdução a estrutura de Dados - FILA

Sumário

- **Estruturas de Dados**

- **Dinâmicas:**

- O tamanho se altera com a necessidade;
 - Cresce ou Decresce
 - Simplesmente Encadeadas
 - Listas
 - **Filas**
 - Pilhas
 - Duplamente Encadeadas

Filas

- Estrutura de dados para armazenar informações;
 - Semelhante a uma lista, mas com restrições;
 - ***A Fila só pode ser acessada pelos extremos (Início e Fim)***
 - Exemplos:
 - Fila Banco
 - Fila Padaria
 - Fila Pedágio
 - Fila Posto médico

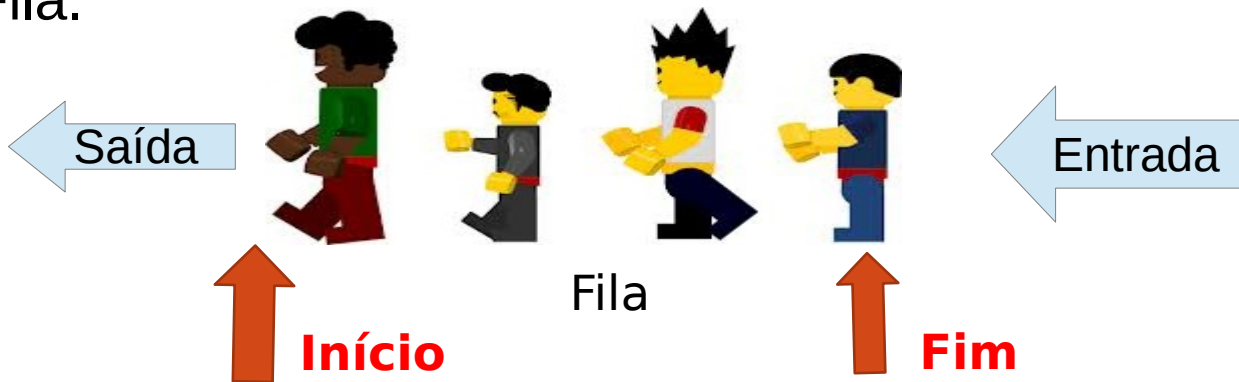


Início

Fim

Fila

- Quais Restrições??
 - A fila é acessada pelos extremos:
 - Início e Fim da Fila;
 - **Início** é ponto de partida da fila e a porta de saída da fila;
 - **Fim** é o ponto mais distante do início da Fila, porta de entrada na Fila.



Fila

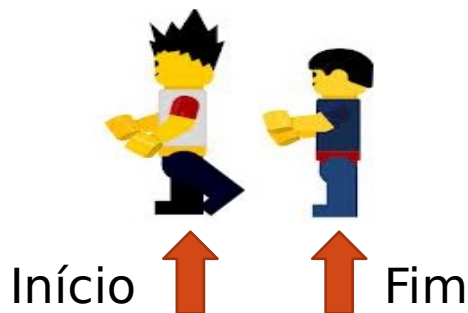
- Quais Restrições??
 - A fila é acessada pelos extremos:
 - Início e Fim da Fila;
 - **Início** porta de saída da fila;
 - **Fim** porta de entrada na Fila.



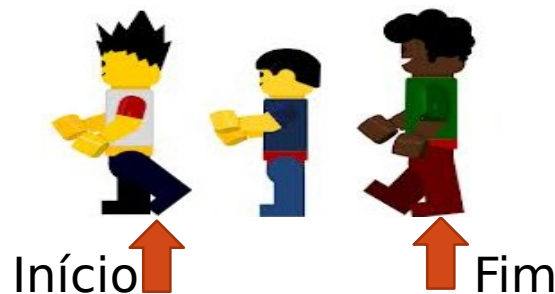
Fila – Operação Enfileirar

- **ENFILEIRAR (enqueuei) -**

- Ocorre ao inserir um novo elemento na Fila, o apontador **(Fim)** é deslocado para o novo elemento.



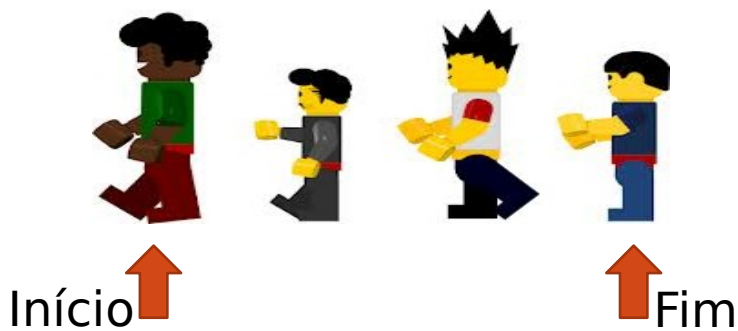
Antes da inserção



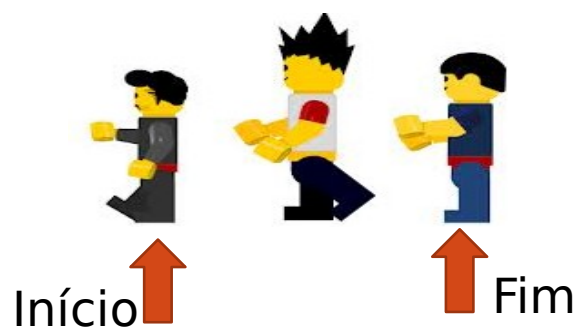
Após a inserção

Fila – Operação Desenfileirar

- **Desenfileirar (dequeue)**
 - Ocorre ao retirarmos um elemento da fila, o apontador (**início**) é deslocado para o elemento anterior.



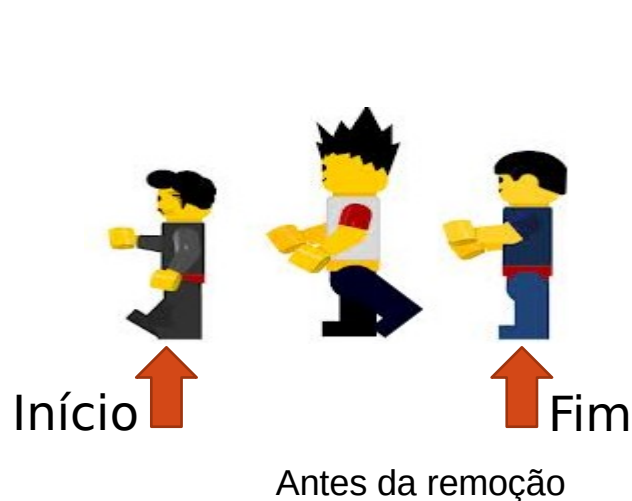
Antes da remoção



Após a remoção

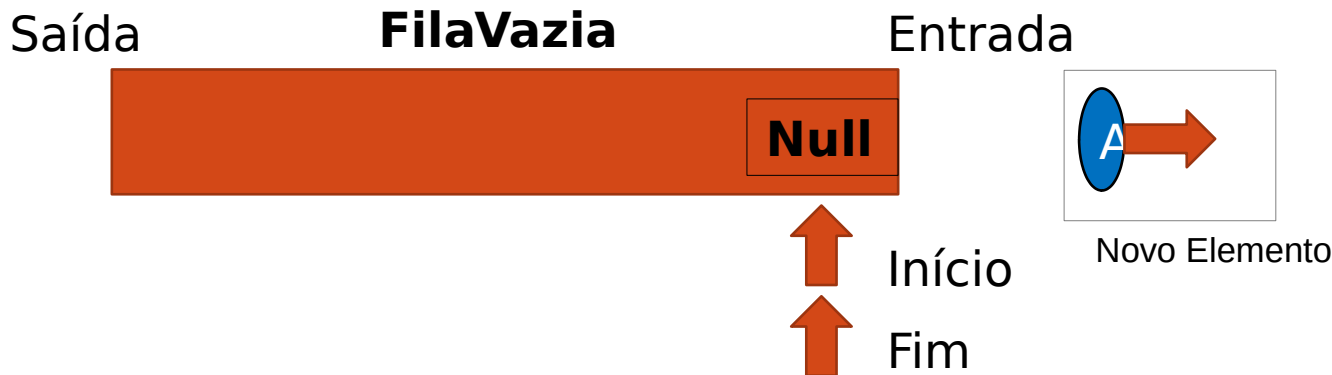
Fila – Operação Desenfileirar

- **Ao retiramos** um elemento do início da fila, o apontador (**início**) é deslocado para o elemento anterior.



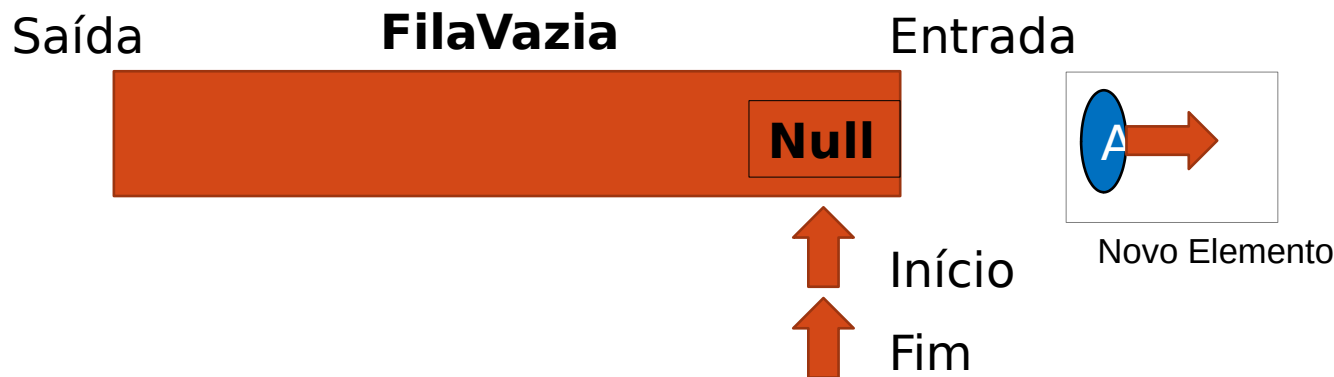
Fila

- Nas Filas os elementos são adicionados na última posição e removidos da primeira posição. Este comportamento é conhecido como:
 - FIFO (First In, First Out)**, do inglês,
 - PEPS (Primeiro a Entra é o Primeiro a Sair)**.



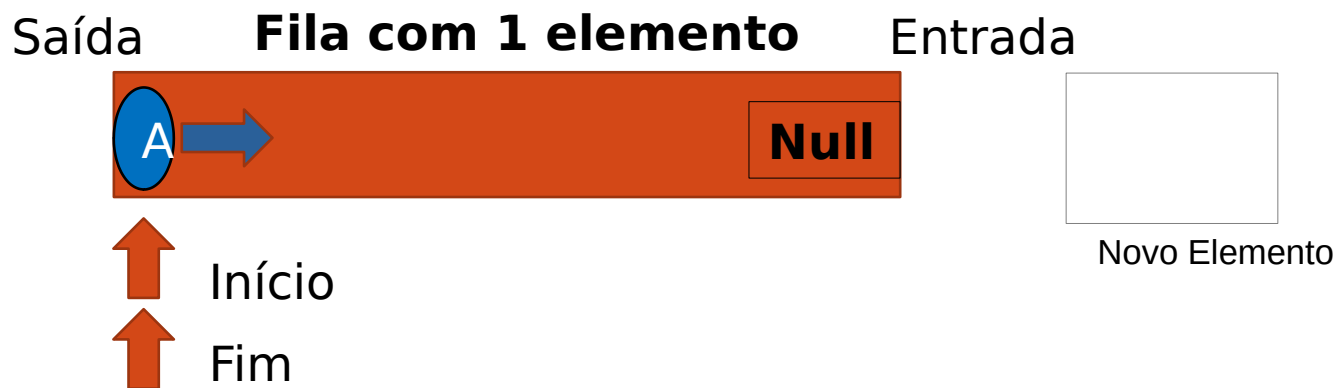
Fila – Operação Enfileirar

- Nas Filas os elementos são adicionados na última posição (FIM). O ponteiro Fim se desloca para indicar o último elemento inserido.



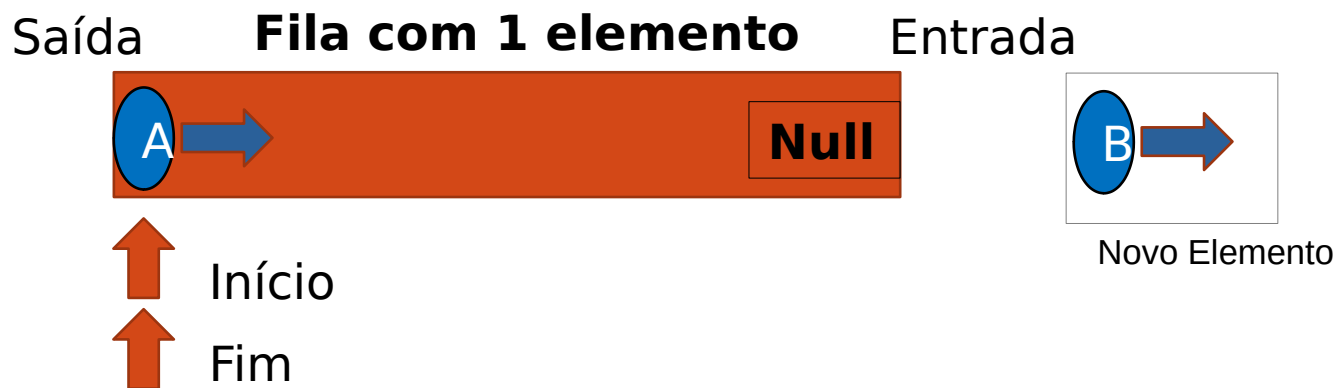
Fila – Operação Enfileirar

- Novo Elemento (A) inserido no fim;



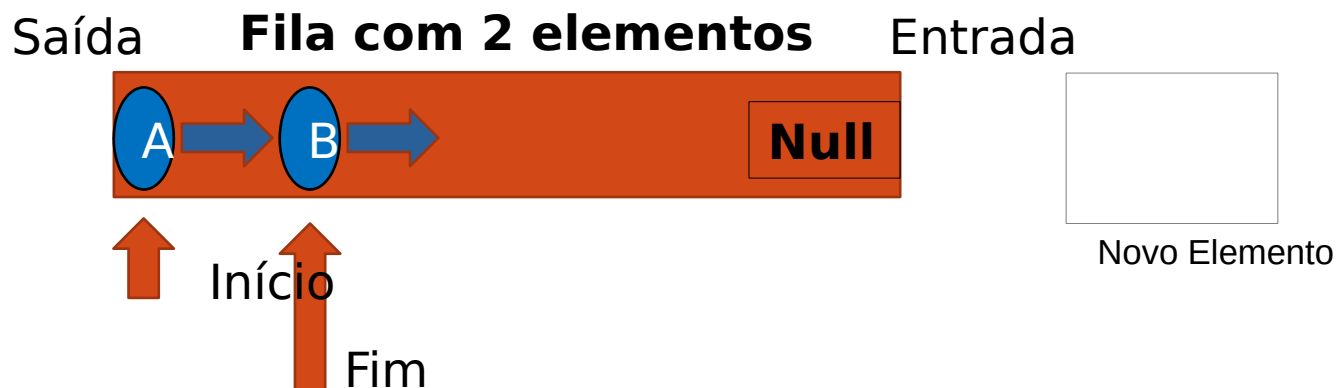
Fila – Operação Enfileirar

- Novo Elemento (B) inserido no fim;



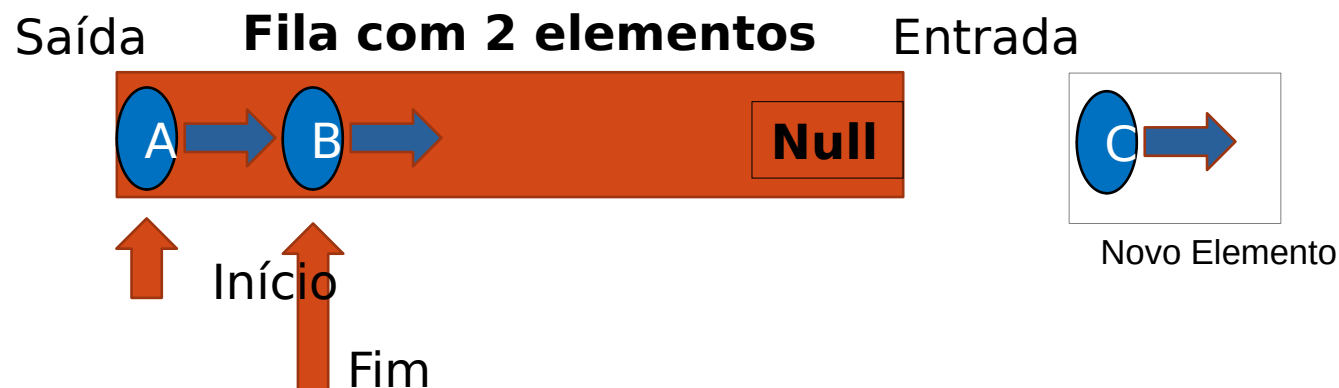
Fila – Operação Enfileirar

- Novo Elemento (B) inserido no fim;



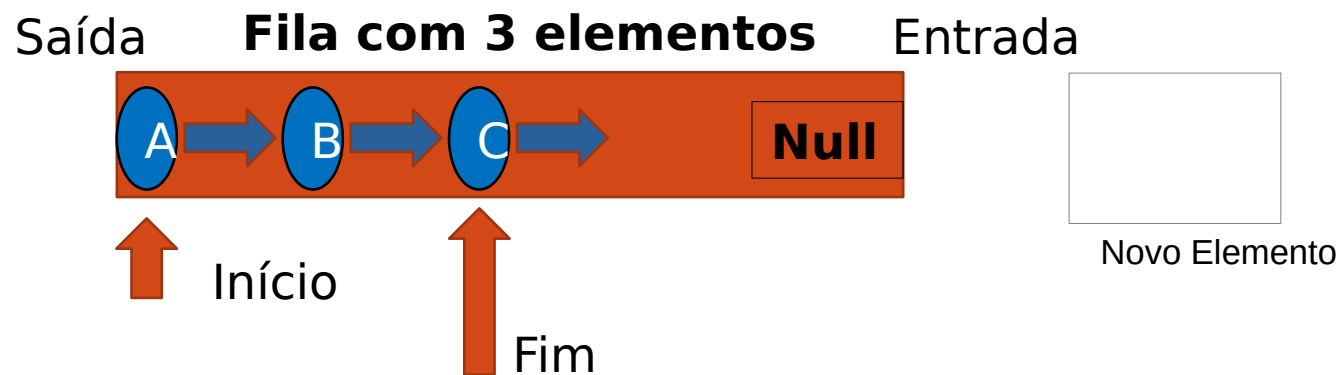
Fila – Operação Enfileirar

- Novo Elemento (C) inserido no fim;



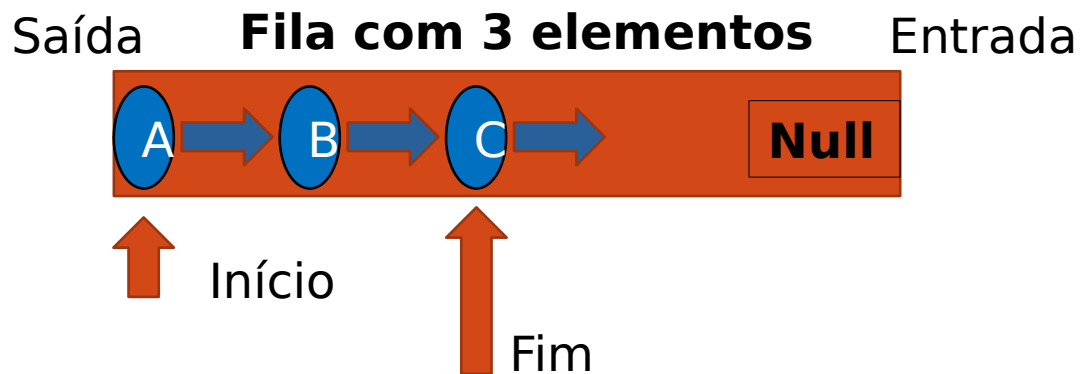
Fila – Operação Enfileirar

- Novo Elemento (C) inserido no fim;



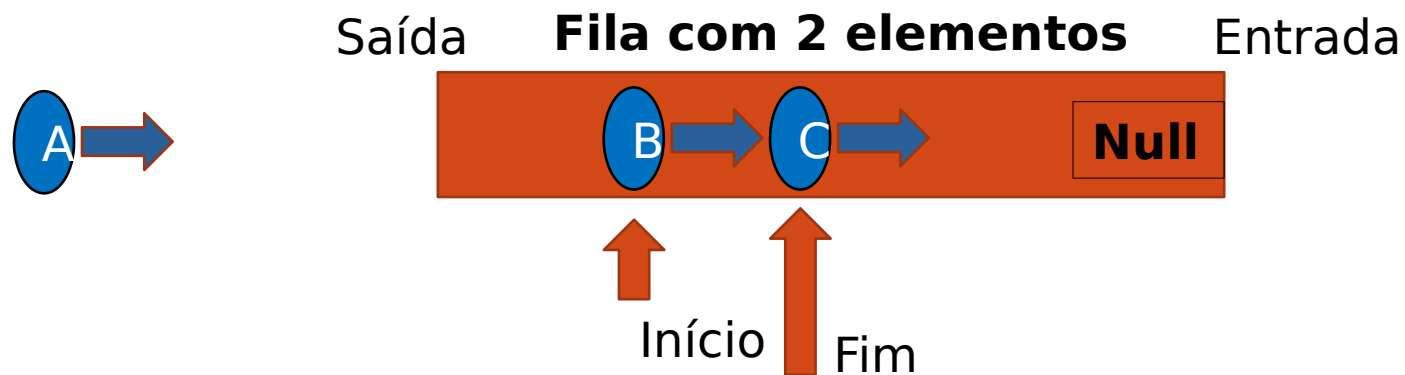
Fila – Operação Desenfileirar

- Desenfileirar:
 - Nas Filas os elementos são removidos na primeira posição (INÍCIO). O ponteiro Início se desloca e passa a indicar o elemento anterior.



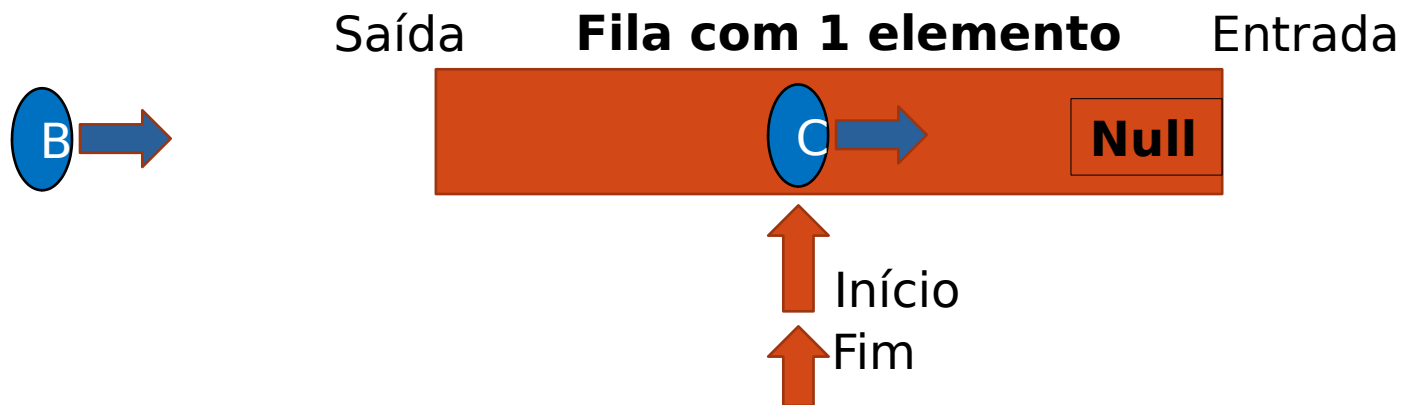
Fila – Operação Desenfileirar

- Desenfileirar:
 - Elemento (A)



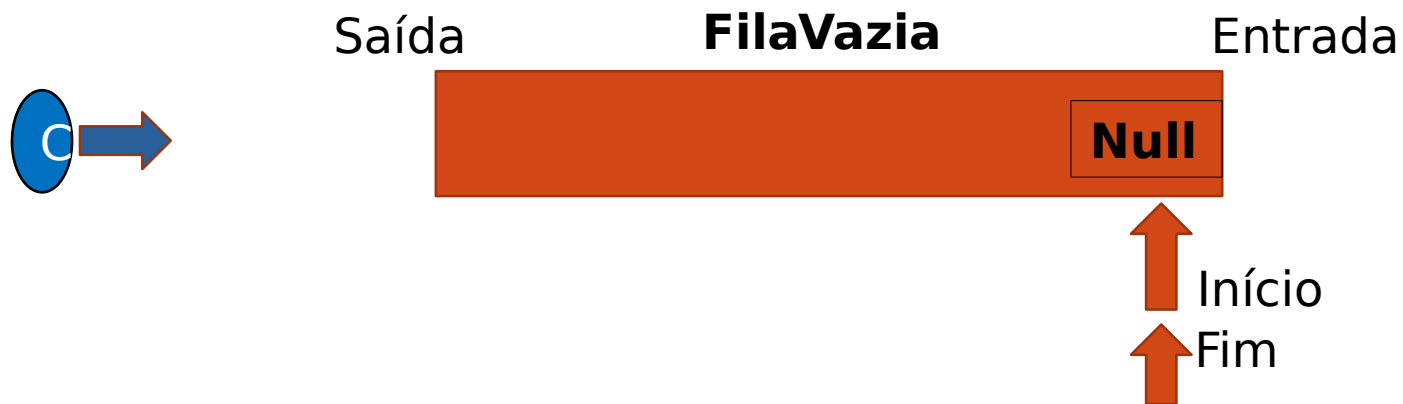
Fila – Operação Desenfileirar

- Desenfileirar:
 - Elemento (B)



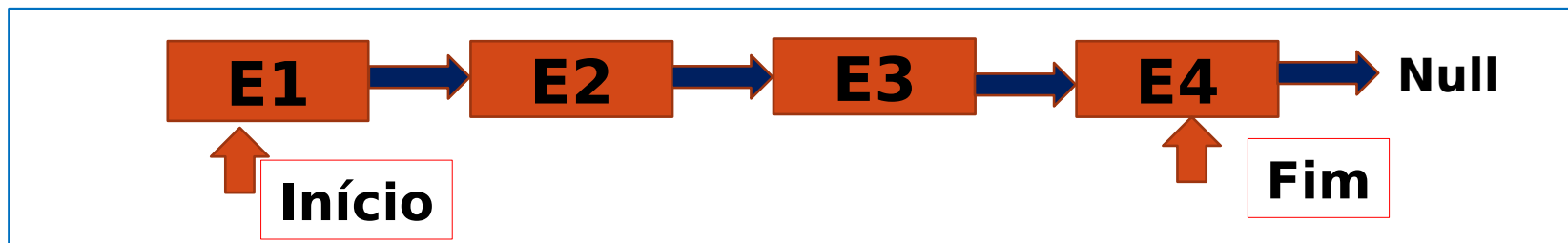
Fila – Operação Desenfileirar

- Desenfileirar:
 - Elemento (C)



Fila

- Fila nada mais é do que lista linear com restrições:
 - Insere no Fim (Enfileirar)
 - Remove no Início (Desenfileirar)



FILA - FIFO

Fila - Elemento

- Um fila pode armazenar qualquer objeto.
 - Exemplo: Pessoas, carros e etc;

Pessoa
nome cpf

Dados Pessoa

Fila - Elemento

- Encadeamento:
 - Os elementos têm uma referência para o próximo elemento
“próximo da fila” (tipo Pessoa)

Pessoa
nome cpf Pessoa *proximo

Dados Pessoa

Fila

- **Interface da Fila (operações)**

- Enfileirar Pessoa;
 - Adiciona uma Pessoa na fila (final fila);
- Desenfileirar Pessoa;
 - Remove uma Pessoa da fila (início fila);
- Fila vazia
 - Retorna (True) se a fila não tiver elementos armazenados
 - Retorna (False) se a fila tiver elementos armazenados
- Exibe Fila
 - Mostra todos elementos da fila, começando pelo Início da fila até chegar no Fim;

Fila

- Estruturas em C para armazenar
 - Fila de pessoas em um banco

```
1  typedef struct dados{    //dados do cliente bancário
2      char nome [20];
3      int operacao;
4      float valor;
5  }Dados;
6
7  typedef struct elemento{    //elemento de Fila
8      struct elemento *proximo;
9      Dados dados;
10     int id;
11 }Elemento;
12
13
14 typedef struct fila{    //atributos de Fila
15     Elemento *primeiro;
16     Elemento *ultimo;
17     int qtd;
18 }Fila;
--
```

Fila - Interface

- **Interface da Fila (operações)**

- Enfileirar pessoa;
 - **void enfileirar (Fila *f, Efila *ef);**
- Desenfileirar Pessoa;
 - **Efila * desenfileirar (Fila *f);**
- Fila vazia
 - Retorna (True) se a fila não tiver elementos armazenados;
 - Retorna (False) se a fila tiver elementos armazenados;
- Exibe Fila
 - Mostra todos elementos da fila, começando pelo primeiro da fila até chegar ao final;
 - void mostraFila(Fila *f);

Fila - funções

- Interface da Fila (Pessoas)

- ```
Fila * criaFila();

Dados preencheElemento(char nome [] , int op, double vl);

Elemento * criaElemento(Dados dl, int id);

void enfileirar(Fila *p, Elemento *e); //insereNoFim()

Elemento * desenfileirar(Fila *p); //removeNoInicio()

void mostraFila(Fila *p);

void apagaFila(Fila *p);

void apagaElemento(Elemento *ef);

void mostraElemento(Elemento *ef);
```

# Fila – Inserção

- Função de inserção na Fila
  - Enfileirar (Fila \*f, Elemento \*e)

```
void enfileirar(Fila *f, Elemento *e){ //insereNoFim()
 if(f->primeiro == NULL){ //Fila Vazia
 e->proximo = NULL;
 f->primeiro = e;
 }else{ //Fila com elementos
 e->proximo = NULL;
 f->ultimo->proximo = e;
 }
 f->ultimo = e;
 f->qtd++;
}
```

# Fila – Remoção

- Função de remoção na Fila
  - Elemento \* Desenfileirar (Fila \*f)

```
Elemento * desenfileirar(Fila *f){ //removeNoInicio()
 Elemento *aux = f->primeiro;
 if(aux == NULL) //Fila vazia
 printf("\nErro - Fila Vazia\n");
 else{ //Lista com elementos
 f->primeiro = aux->proximo;
 f->qtd--;
 aux->proximo = NULL;
 if(f->primeiro == NULL)
 f->ultimo = NULL;
 }
 return aux;
}
```

# Execução Fila

- Programa Main()
  - Menu de operações na Fila:

```
Informe uma Opção:
-- 1 - para Insere:
-- 2 - para Remove:
-- 3 - MostraFila:
-- 4 - Apaga Fila:
-- 0 - para Sair do Programa:
```

```
Informe sua Opção:2
```

# Exercício

- Implemente um sistema para distribuir fichas aos clientes em um atendimento bancário;
  - Caixa:
    - Implemente a fila de prioritários;
    - Implemente a fila de atendimentos ao caixa;
  - Gerente
    - Implemente a fila de prioritários;
    - Implemente a fila de atendimento ao gerente;
- Sumule o atendimento de vários clientes pelo gerente e pelo caixa;
  - Apresente em uma tela a chamada dos clientes de modo aleatório para o caixa e para o gerente;

# Exercício

- Um algoritmo ao ser compilado e executado dará origem a um processo, este tem uma fila para gerenciar a execução de tarefas do programa.
- Implemente uma fila para armazenar os comandos e as variáveis que serão alteradas na execução do comando;
- Faça um parser para ler um pequeno programa em Neander, e os insira em Fila;
- Após realize a execução dos comandos que estão armazenados na fila, verifique o resultado.
  - Exemplo; para somar dois números em Neander

| simbólico | comentário                                               |
|-----------|----------------------------------------------------------|
| LDA 128   | Acumulador (AC) recebe o conteúdo da posição 128         |
| ADD 129   | Conteúdo de AC é somado ao conteúdo da posição 129       |
| ADD 130   | Conteúdo de AC é somado ao conteúdo da posição 130       |
| STA 131   | Conteúdo de AC é armazenado (copiado) para a posição 131 |
| HLT       | Processador pára                                         |