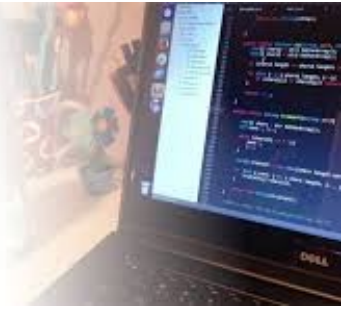
Tecnologia em Análise e Desenvolvimento de Sistemas - TADS

Estrutura de Dados

Prof. ***Luciano Vargas Gonçalves***
E-mail: luciano.goncalves@riogrande.ifrs.edu.br

tads
Análise e Desenvolvimento de Sistemas
INSTITUTO FEDERAL
Rio Grande do Sul
Campus Rio Grande



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

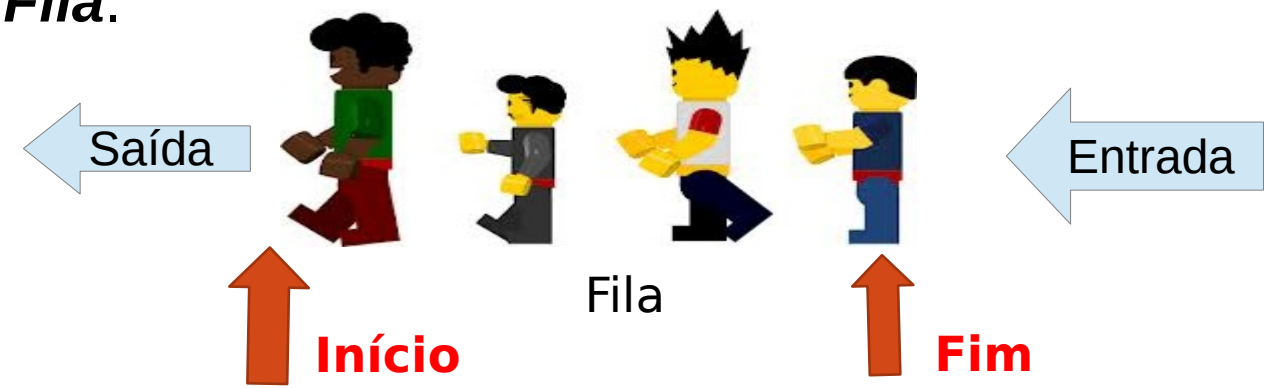
- Estrutura de dados para armazenar informações;
  - Semelhante a uma lista encadeada ***com restrições;***
    - ***A Fila só pode ser acessada pelos extremos (Início e Fim)***
      - ***Inserção no Fim e remoção no Início***
      - Exemplos:
        - Fila Banco
        - Fila Padaria
        - Fila Pedágio
        - Fila Posto médico

Fim

Início

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***.



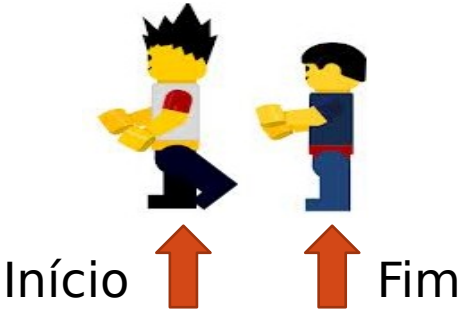
Fila

- 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.

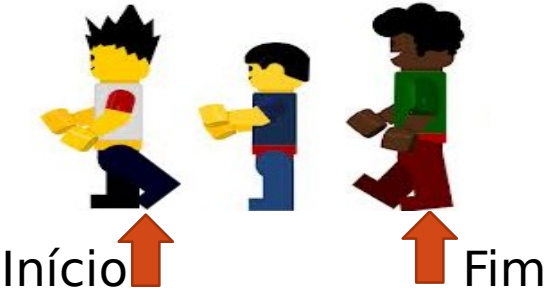
Fila Banco, Supermercado

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

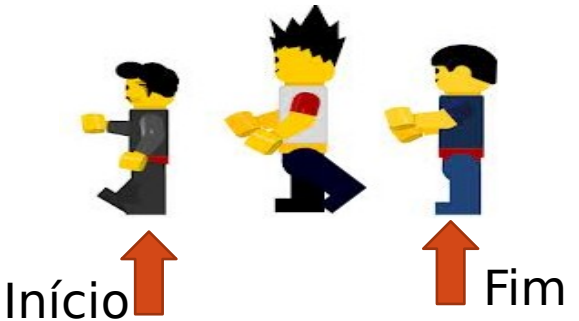


Após a inserção

Fila – Operação Desenfileirar

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

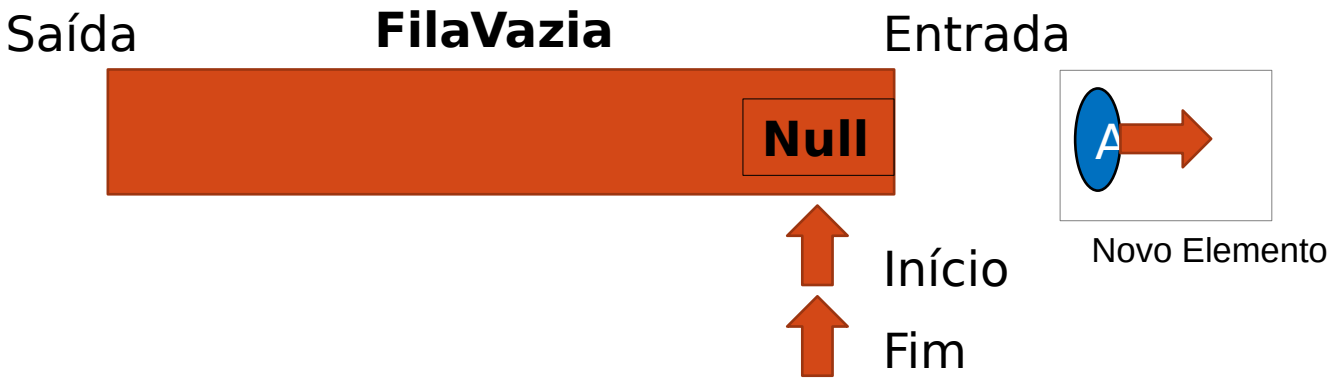
Antes da remoção



Após a remoção

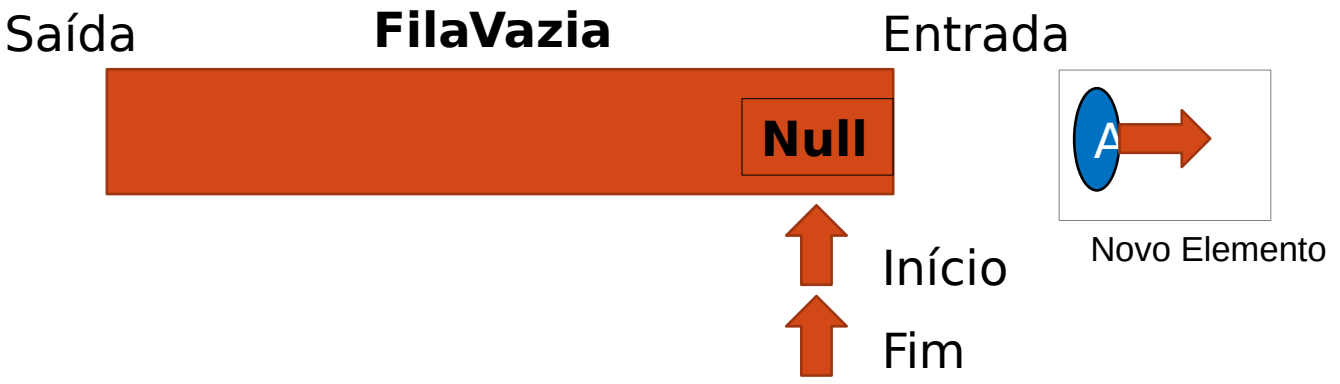
Fila

- 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).**



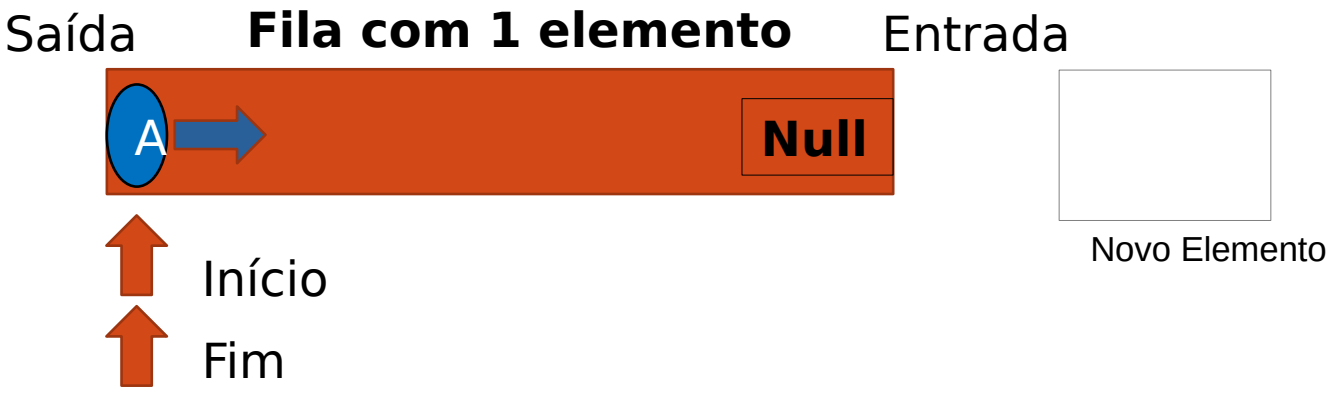
Fila – Operação Enfileirar

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



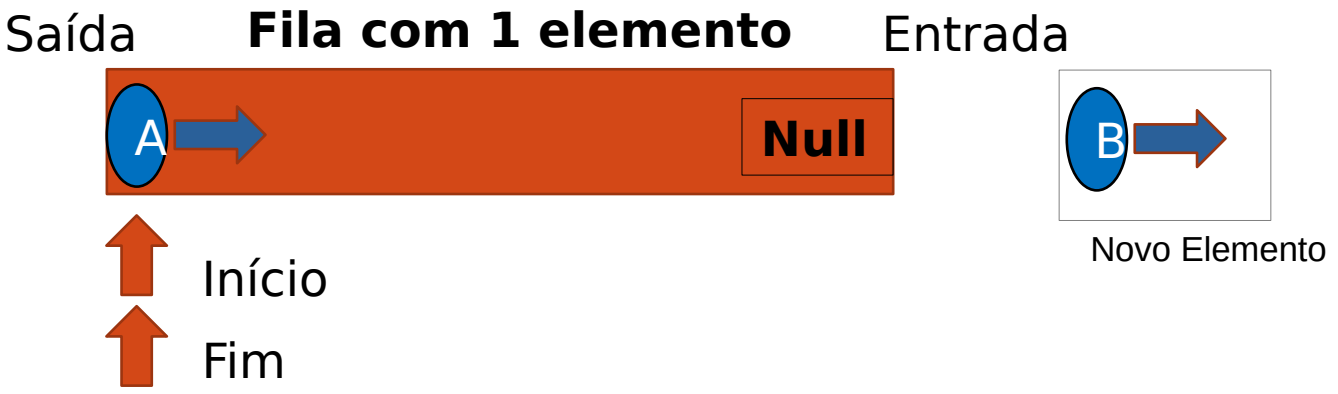
Fila – Operação Enfileirar

- Novo Elemento (A) inserido no fim;



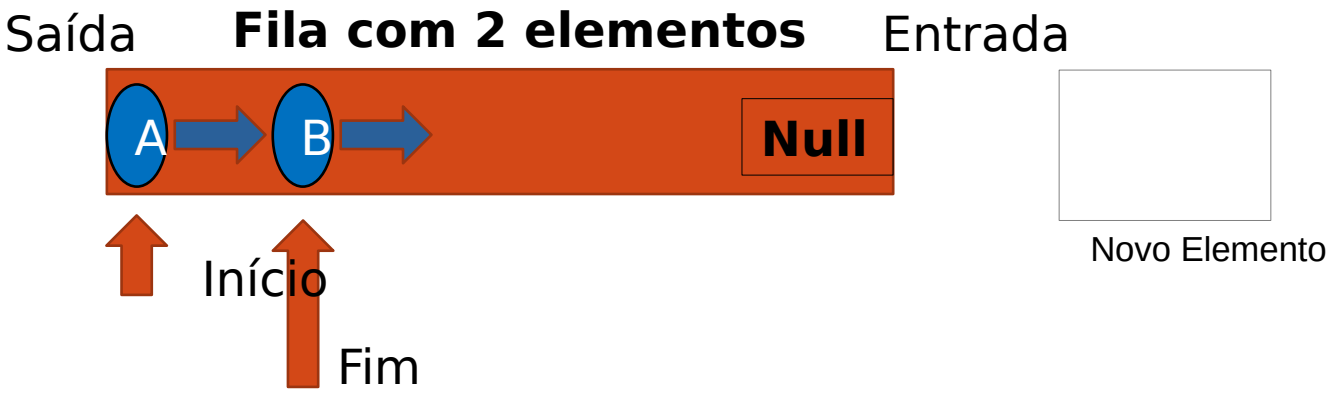
Fila – Operação Enfileirar

- Novo Elemento (B) inserido no fim;



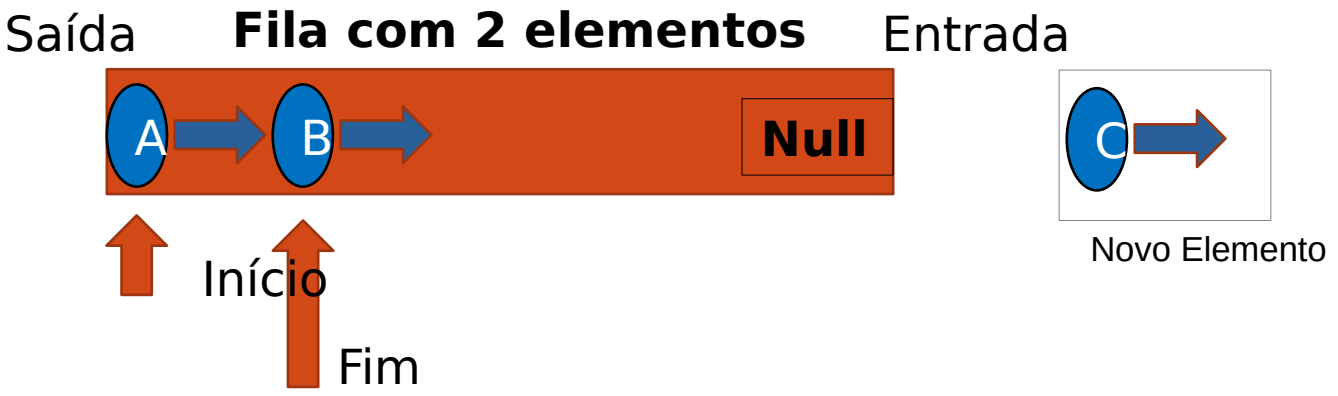
Fila – Operação Enfileirar

- Novo Elemento (B) inserido no fim;



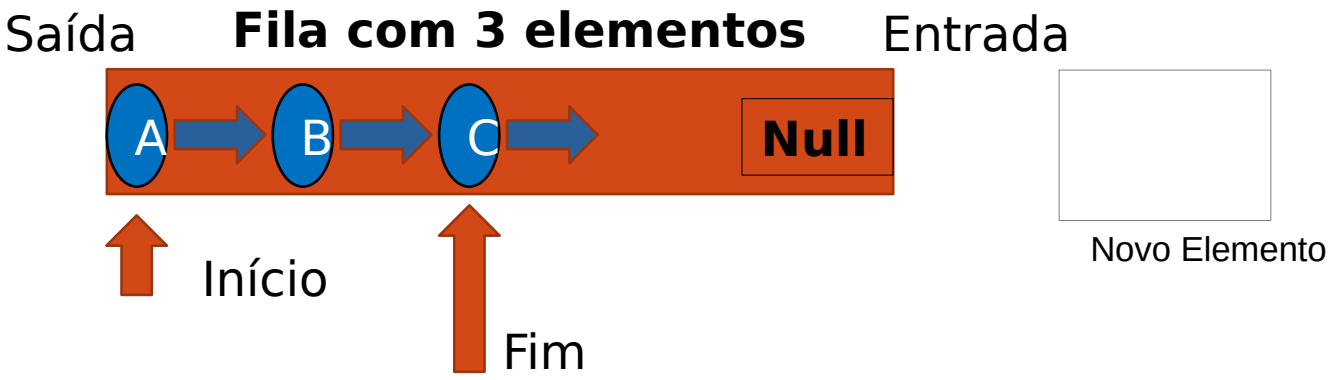
Fila – Operação Enfileirar

- Novo Elemento (C) inserido no fim;



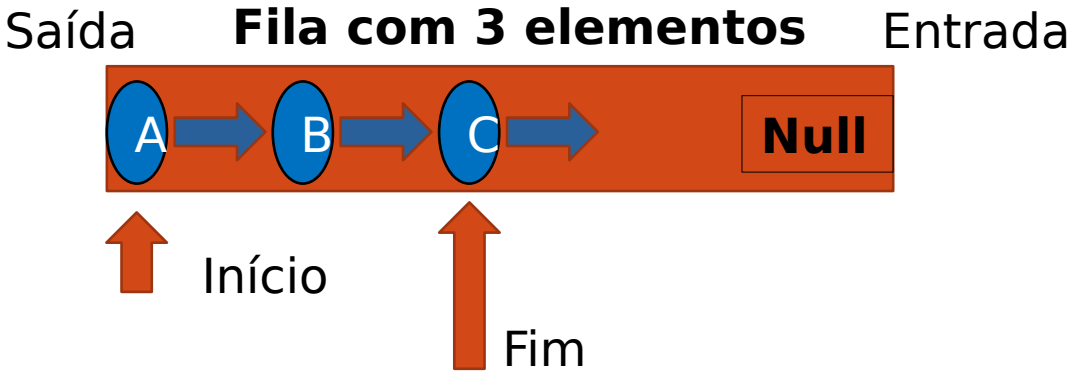
Fila – Operação Enfileirar

- Novo Elemento (C) inserido no fim;



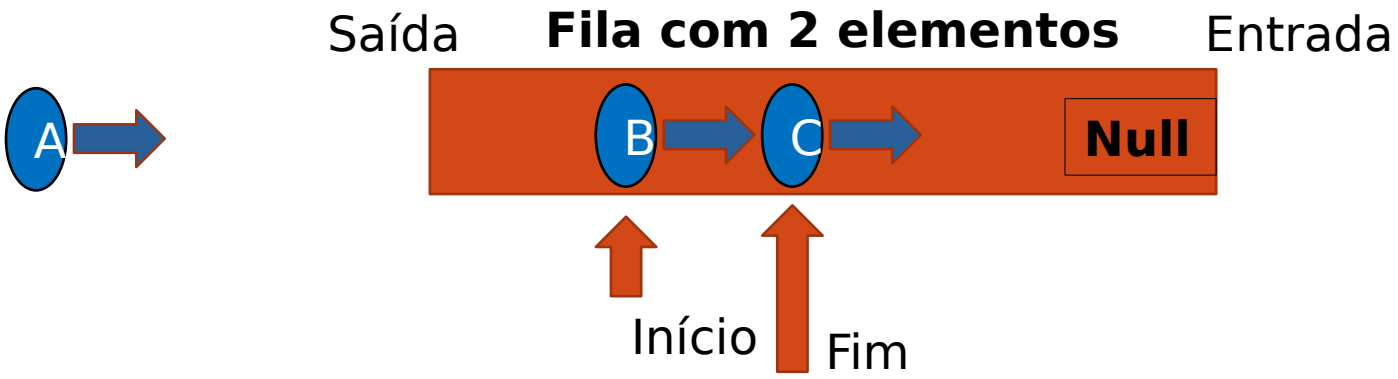
Fila – Operação Desenfileirar

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



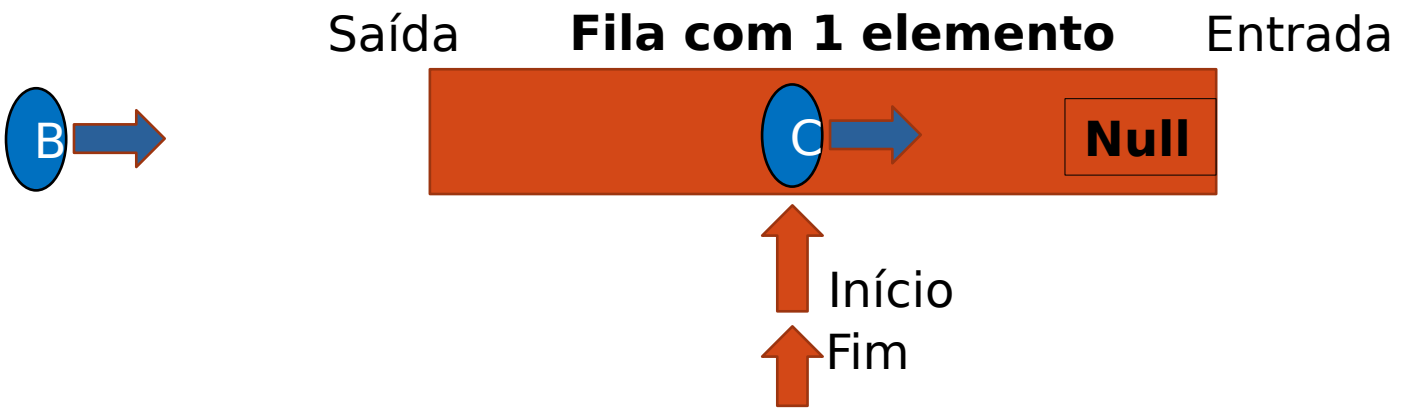
Fila – Operação Desenfileirar

- Desenfileirar:
  - Elemento (A)



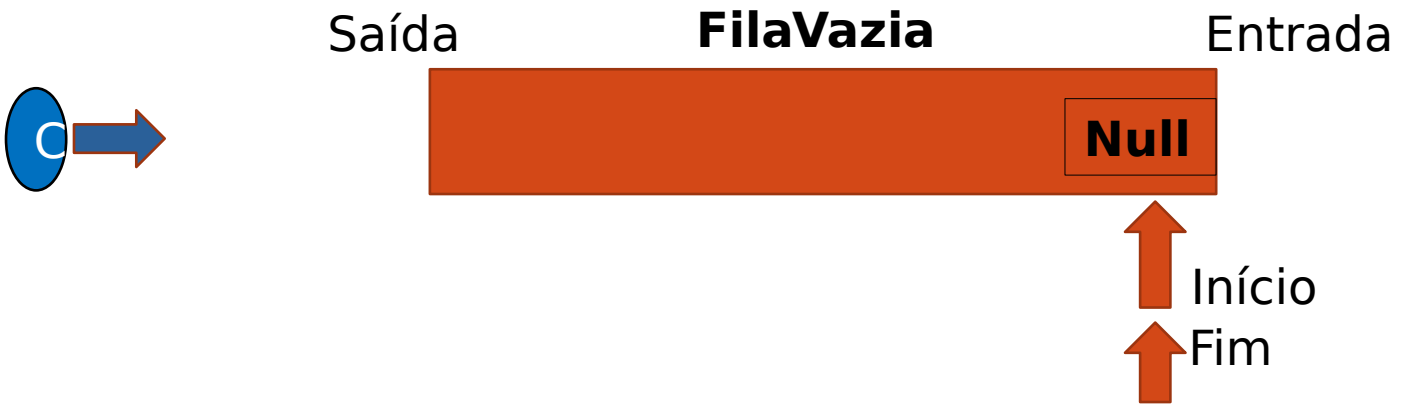
Fila – Operação Desenfileirar

- Desenfileirar:
  - Elemento (B)



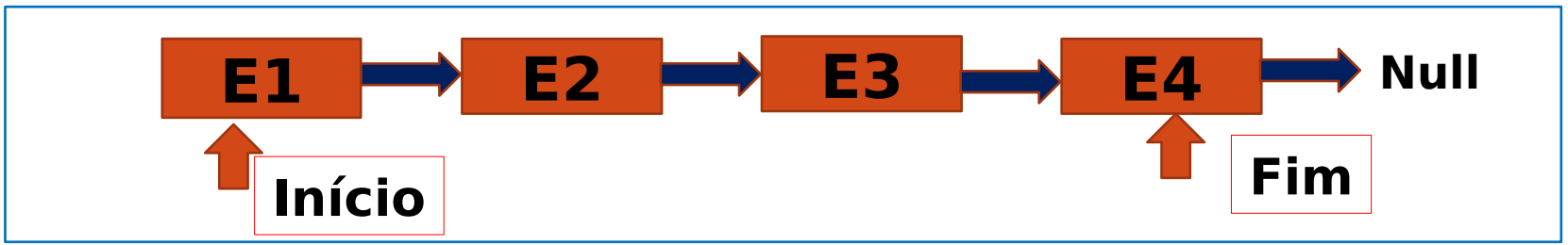
Fila – Operação Desenfileirar

- Desenfileirar:
  - Elemento (C)



Fila

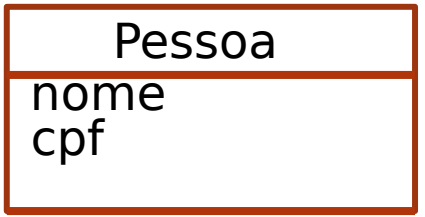
- Fila nada mais é do que lista linear com restrições:
  - Insere no Fim (Enfileirar)
  - Remove no Início (Desenfileirar)



FILA - FIFO

Fila - Elemento

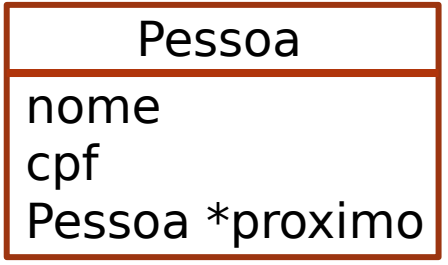
- Um fila pode armazenar qualquer objeto.
  - Exemplo: Pessoas, carros e etc;



Dados Pessoa

Fila - Elemento

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



Dados Pessoa

Fila

- **Interface da Fila (operações)**
  - Enfileirar Pessoa;
    - Adiciona uma Pessoa na fila (fim da fila);
  - Desenfileirar Pessoa;
    - Remove uma Pessoa da fila (início da 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
  - Dados de um cliente bancário
    - Nome, operação, valor
  - Dados da fila;
    - Estrutura
    - Ponteiros
      - Início e Fim;

```c
typedef struct cliente{   //dados do cliente bancário
    char nome [20];
    int operacao;
    float valor;
    int id;
    struct cliente *proximo;
}Cliente;
```

```c
typedef struct fila{   //atributos de Fila
    Cliente *inicio; //inicio
    Cliente *fim;
    int qtd;
}Fila;
```

Fila - Interface

- **Interface da Fila (operações)**
  - Enfileirar pessoa;
    - **void enfileirar (Fila *f, Cliente *cl);**
  - Desenfileirar Pessoa;
    - **Cliente * 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)**

```
Cliente* cadastraNovoCliente(char nome [] , int op, double vl, int id ){…

void enfileirar(Fila *fl, Cliente *cl){ …

Cliente* desenfileirar(Fila *fl){  …

void mostraCliente(Cliente cl){…

void mostraFila(Fila *fl){…

void apagaCliente(Cliente *cl){…

void apagaFila(Fila *fl){…
```

Fila – Inserção

- **ENFILEIRAR (enqueuei)** - Função de inserção na Fila

```
void enfileirar(Fila *fl, Cliente *cl){
    if(fl->inicio == NULL){ //Fila Vazia
        fl->inicio = cl;
    }else{              //Fila com elementos
        fl->fim->proximo = cl;
    }
    cl->proximo = NULL;
    fl->fim = cl;
    fl->qtd++;
}
```

Fila – Remoção

- **DESENFILEIRAR (denqueuei) -** Função de remoção na Fila

```
Cliente* desenfileirar(Fila *fl){
    Cliente *aux = fl->inicio;
    if(aux == NULL)   //Fila vazia
        printf("\nErro - Fila Vazia\n");
    else{       //Lista com elementos
        fl->inicio = aux->proximo;
        fl->qtd--;
        aux->proximo = NULL;
        if(fl->inicio == NULL)
            fl->fim = 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 Opçao:2
```

Exercício 1

- Implemente um sistema para distribuir fichas aos clientes em um atendimento bancário;
  - Implemente a fila de clientes prioritários;
    - Tem operações de (saque ou depósito)
  - Implemente a fila de clientes gerais;
    - Tem operações de (saque ou depósito)

Exercício 2

- Implemente um sistema para distribuir fichas aos clientes em um atendimento bancário;

  Atendimento pelo Gerente

  - Implemente a fila de clientes prioritários;
  - Implemente a fila de clientes gerais;

Exercício 3

- Implemente um sistema para distribuir fichas aos clientes em um atendimento bancário;

  Crie um quadro com os números dos últimos clientes atendidos:

  - Atendimento ao Caixa:
    - Prioritário Número:
    - Geral Número:
  - Atendimento Gerente
    - Prioritário Número:
    - Geral Número:

Exercício 4

- Implemente um sistema para distribuir fichas aos clientes em um atendimento bancário;

  Implemente um sistema para chamar os clientes de cada fila (Caixa e Gerente);

  - Sempre iniciar chamando o cliente prioritário
    - Após 2 chamados para clientes prioritários, chamar um cliente geral
    - Voltar a chamar um novo cliente prioritário, após um cliente geral;

Exercício 5

- Implemente duas filas de LOGs, uma para cada atendimento Caixa ou Gerente;
- Simule um caixa de banco, os clientes podem realizar duas operações (Depósito ou Saque)
  - O caixa tem um saldo inicial, caso o valor seja zerado, informar o gerente da falta de dinheiro em caixa;

Exercício

- Um algoritmo ao ser compilado e executado dará origem a um processo, este tem um 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 comando 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