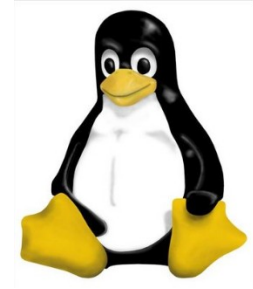


INSTITUTO FEDERAL
BAIANO



LP II – Estrutura de Dados

Filas Estáticas Sequenciais

Prof. José Honorato Ferreira Nunes

honoratonunes@softwarelivre.org

<http://softwarelivre.org/zenorato/honoratonunes>

Filas - conceitos

É uma lista linear em que a inserção é feita numa extremidade e a eliminação na outra.

Conhecida com estrutura FIFO (First In, First Out).

(a1, a2, a3, a4 , ... , an)	
eliminações	inserções
no início	no final

Filas - Exemplos

- Escalonamento de "Jobs": fila de processos aguardando os recursos do sistema operacional.
- Fila de pacotes a serem transmitidos numa rede de comutação de pacotes.
- Simulação: fila de caixa em banco.

Declaração da Estrutura Pilha

Para que possamos trabalhar com fila, devemos primeiramente definir a estrutura para declarar os tipos de dados que queremos armazenar.

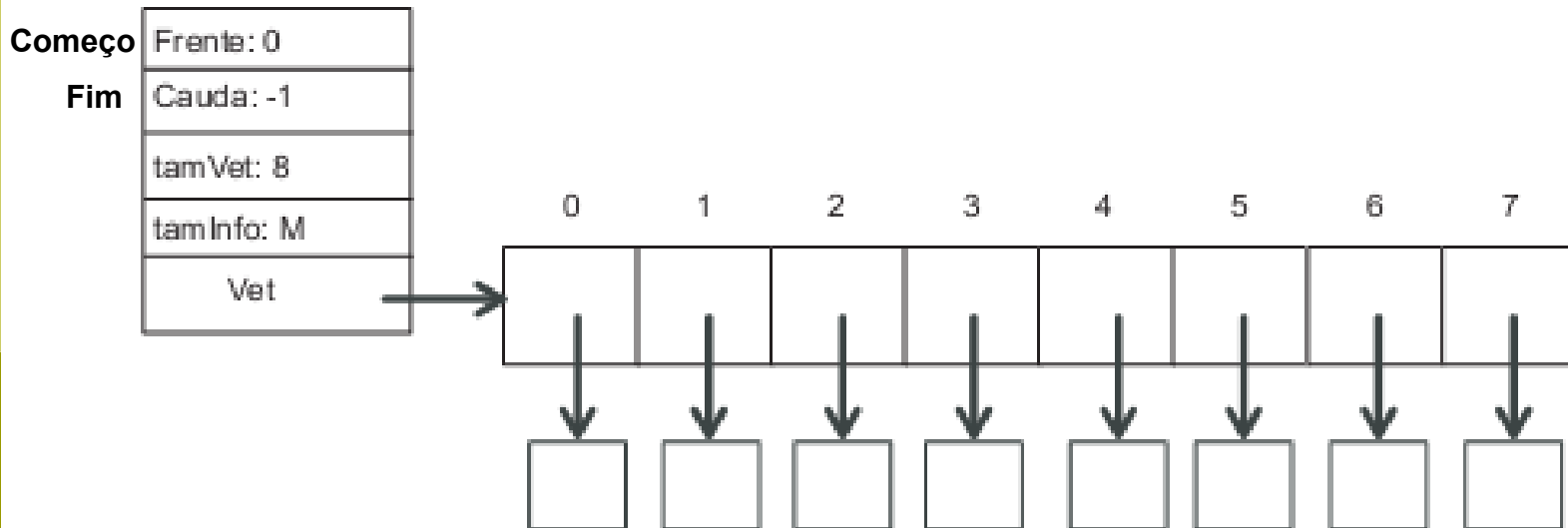
Devido a sua estrutura, será necessária a utilização de dois campos que armazenarão os índices do início e do final da fila e um vetor de elementos (onde serão armazenados os dados) com tamanho preestabelecido.

```
#define m 10
struct fila{
    int comeco, fim;
    char elem[m];
};
```

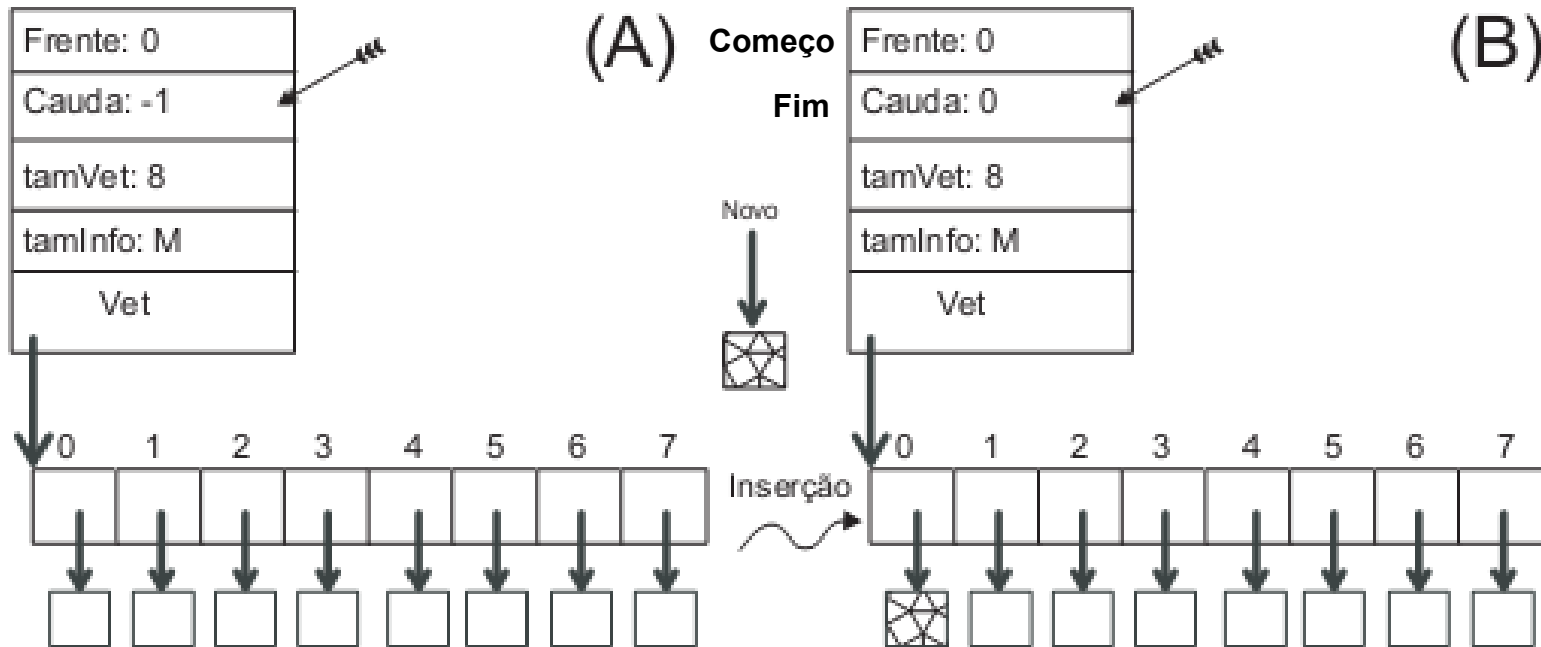
Estrutura FIFO – First In, First Out

- Primeiro elemento inserido é o primeiro a ser removido
- Filas permitem manipulações em duas extremidades: no início (ou frente) e no fim (ou cauda).
- Inserções são feitas na cauda (fim).
- Remoções são feitas na frente (começo).

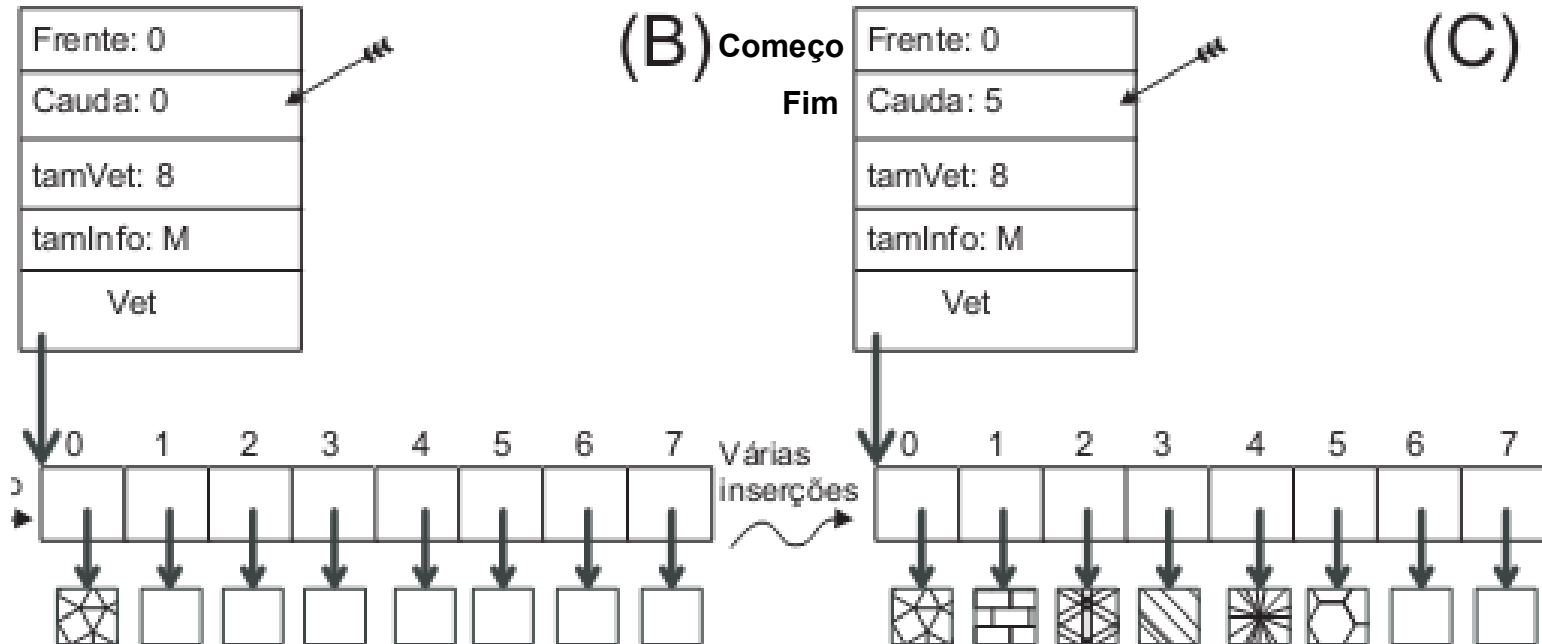
Criando uma Fila



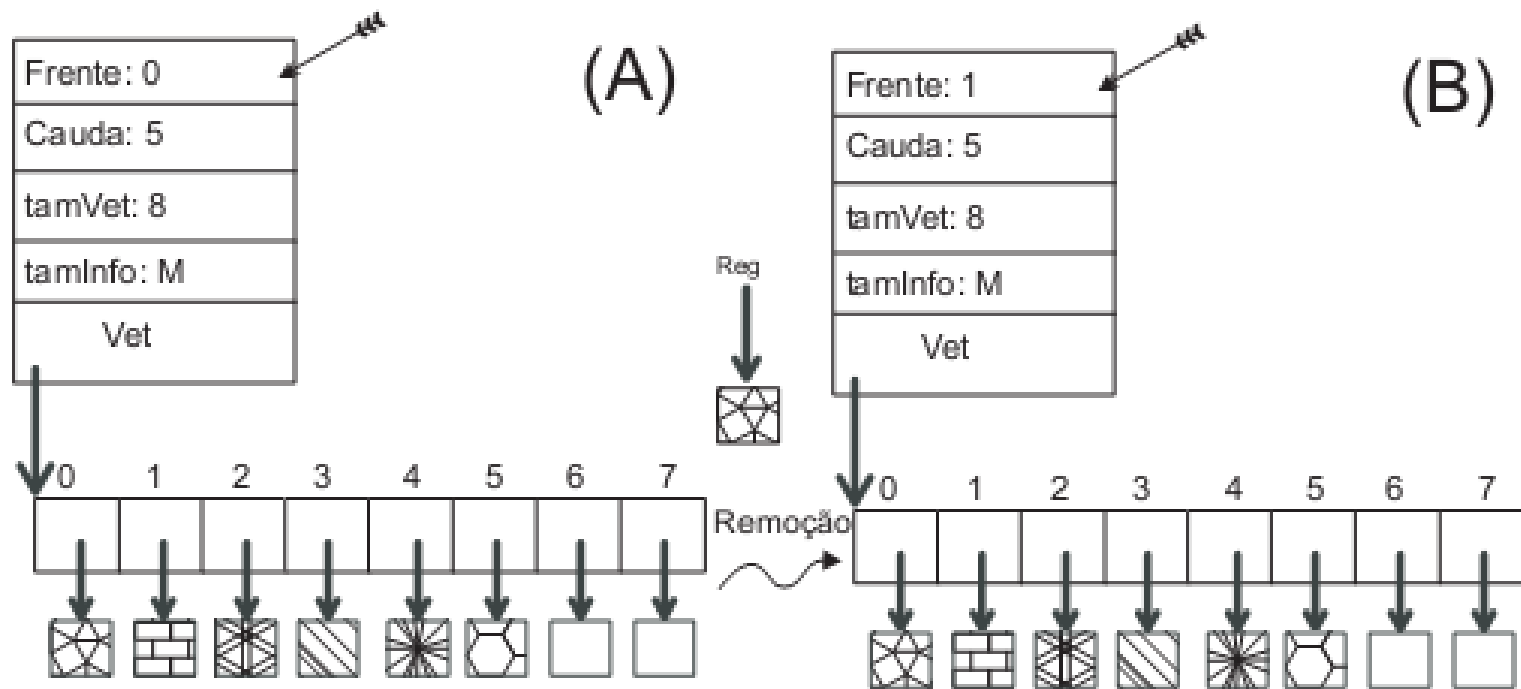
Inserindo na Fila



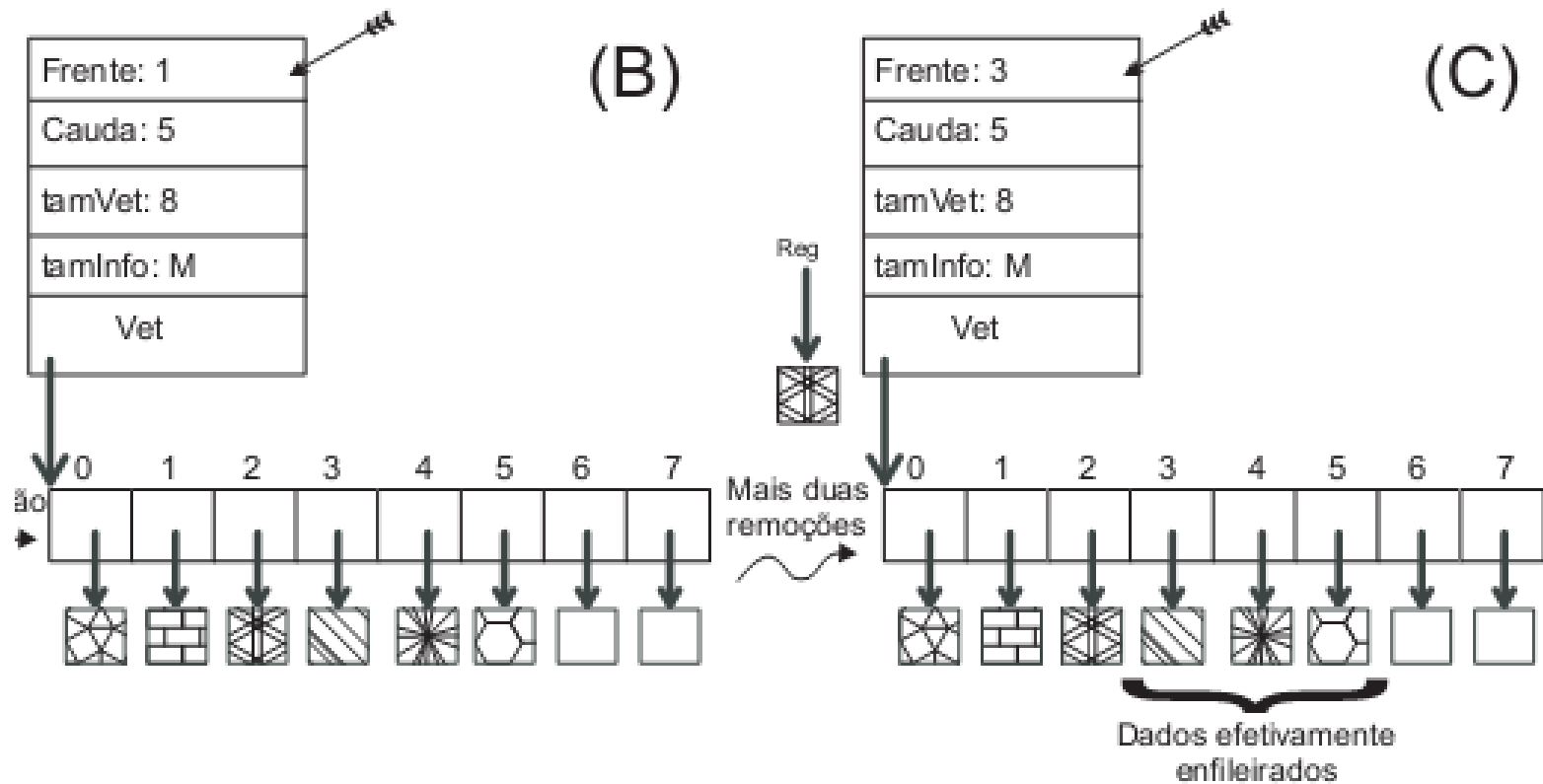
Inserindo na Fila



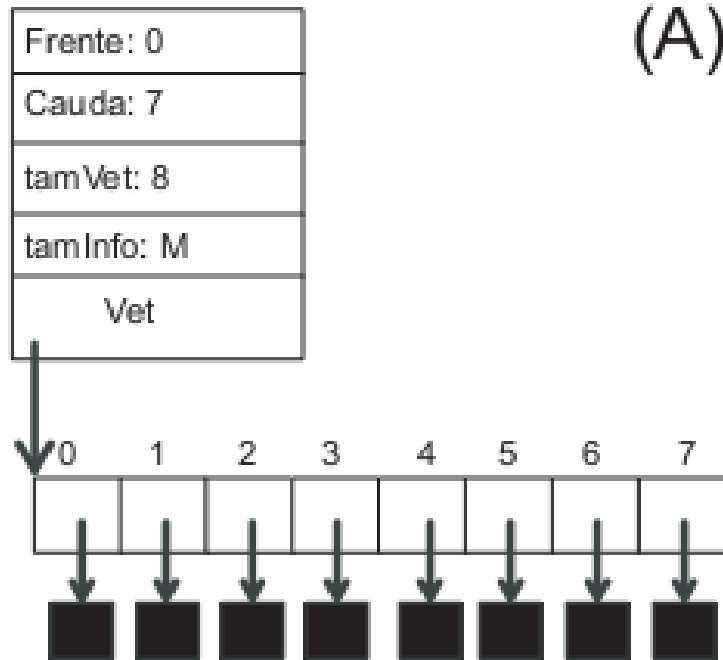
Remoção na Fila



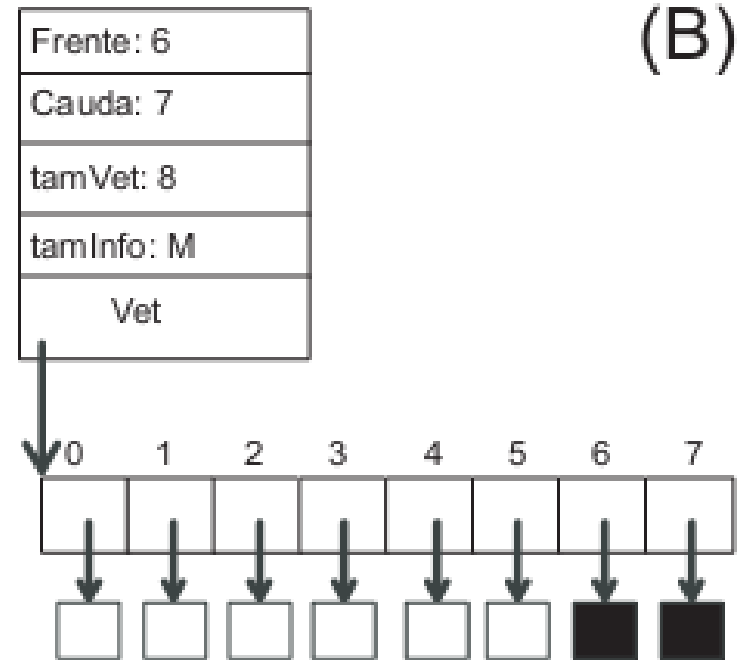
Remoção na Fila



Falso Cheio na Fila



Cheio



Falso Cheio

Operações Primitivas de Filas

- ✓ **Iniciar** – inicializa uma fila vazia;
- ✓ **Vazia** – testa se um fila está vazia;
- ✓ **Inserir** - insere um elemento no fim de uma fila;
- ✓ **Remove** - remove o elemento do início de uma fila, retornando o elemento removido;
- ✓ **Primeiro** - obtém o elemento do início de uma fila.

Primeiro passo é criar a Fila

Antes de começar a carregar a fila com dados, há a necessidade de declarar a estrutura e inicializá-la.

```
//criação da fila
```

```
#define m 10  
struct fila {  
    int comeco, fim;  
    char elem[m];  
};
```

Inicialização da Fila

O processo de inicialização é realizado por uma função de `iniciarFila` que atribui o valor: **-1** a variável ***fim*** e **0** a variável ***comeco***.

```
void iniciaFila(struct fila *F){  
    F->comeco = 0;  
    F->fim = -1;  
}
```

INSERIR - Inserção na Fila

Essa função recebe dois parâmetros: a fila e o valor a ser armazenado.

A cada inserção, o novo endereço será calculado somando **um** na variável ***fim***.

Devemos verificar se ainda há espaço no vetor para mais um elemento!

```
void inserir(struct fila *F, char x){
    if (F->fim == m-1)
        printf("Fila Cheia");
    else{
        F->fim++;
        F->elem[F->fim]=x;
    }
}
```

REMOVER – Retirada na Fila

Essa função elimina o elemento que está no início da Fila, obedecendo o princípio que rege as filas - FIFO.

Antes de remover devemos verificar se a fila está vazia.

```
void retirar(struct fila *F){  
    if (F->fim < F->comeco)  
        printf("Fila Vazia");  
    else  
        F->comeco++;  
}
```


PRIMEIRO – consulta o primeiro elemento da Fila

A função ***primeiro*** retorna o valor da variável no início da fila.

```
char primeiro (struct fila *F){
    char x;
    if (F->fim < F->comeco)
        printf("Fila Vazia");
    else{
        x = F->elem[F->comeco];
        return(x);
    }
}
```

Atividade

Faça um programa que, utilizando as funções criadas nos exemplos dessa aula crie uma fila F e exiba o seguinte menu de opções:

EDITOR DE FILA

- 1 - INSERIR
- 2 - REMOVER
- 3 - EXIBIR ELEMENTO DO INICIO
- 4 - EXIBIR A FILA
- 5 - ESVAZIAR A FILA
- 6 - SAIR

DIGITE SUA OPÇÃO:

Bibliografia Básica

- ❑ SILVA, Osmar Quirino da. **Estruturas de Dados e Algoritmos Usando C – Fundamentos e Aplicações.** Rio de Janeiro, Ed: Ciência Moderna, 2007.
- ❑ MANZANO, Wilson Y. Yamaturni-São Paulo-SP. **Lógica estruturada para programação de computadores,** Ed. Érica 1997 e 2001.
- ❑ MORAES, Celso Roberto. **Estruturas de Dados e Algoritmos.** Ed. Érica, São Paulo.
- ❑ LOPES, Anita. **Introdução à programação.** Rio de Janeiro: Campus, 2002.

Bibliografia Complementar

- ❑ BENEDUZZI, Humberto M. e METZ, João A. **Lógica e Linguagem de Programação – Introdução ao Desenvolvimento de Software** (1ª edição). Editora do Livro Técnico, 2010
- ❑ FORBELLONE, A. L. V. e Eberspacher, H. F. **Lógica de Programação - a Construção de Algoritmos e Estruturas de Dados** (3ª edição). Pearson, 2005
- ❑ CORMEN, Thomas H. et. al. **Algoritmos: Teoria e Prática**. Editora Campus, 2002.
- ❑ ZIVIANI, Nivio. **Projeto de Algoritmos**. Editora Nova Fronteira, 2004.
- ❑ SEBESTA, Robert W. **Conceitos de Linguagens de Programação**. Bookman, 2001.