

# Estruturas de Dados Básicas

## - Fila

Estruturas de Dados



**Universidade de Brasília**

Departamento de Ciência da Computação

# Conceito de Fila

Uma fila é uma estrutura de dados simples usada para armazenar dados.

A principal diferença da fila para outras estruturas semelhantes é a **ordem** de entrada e saída dos elementos/nós da fila.

# Definição de Fila

Uma fila é uma lista ordenada em que a deleção é feita em uma extremidade e a inserção é feita na outra extremidade. O **primeiro** elemento inserido é o primeiro a ser excluído

FIRST IN FIRST OUT (FIFO)

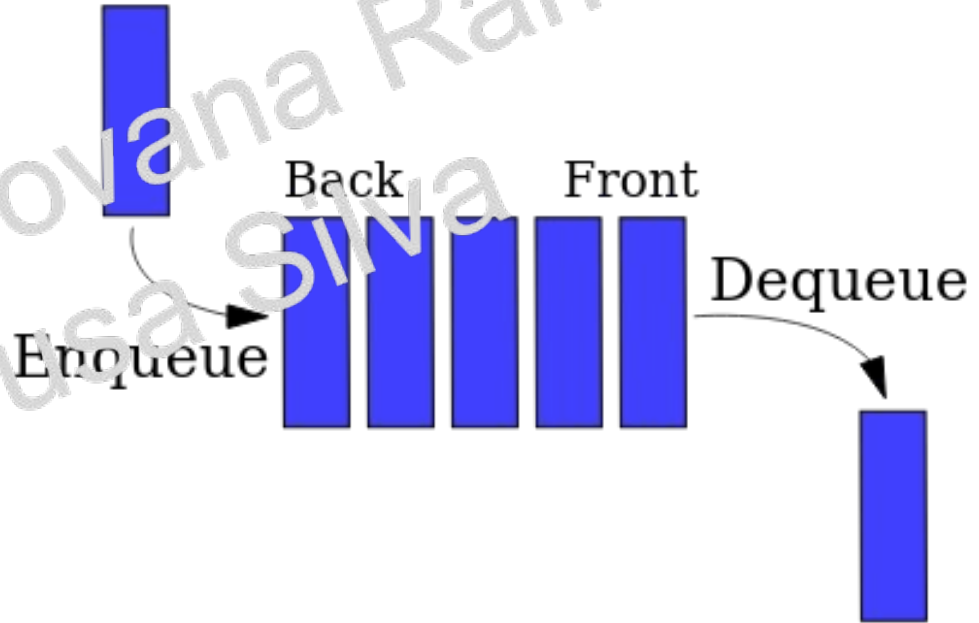
LAST IN LAST OUT (LILO)

# Definição de Fila



# Operações de Fila

- enqueue (enfileirar):
  - inserir na fila
- dequeue (desenfileirar):
  - remover da fila



# Implementação

- Array
- Lista encadeada

Profa. Geovana Ramos  
Sousa Silva

# Implementação como Array

```
class Fila:
```

```
    def __init__(self):  
        self.items = []
```

```
    def enqueue(self, item):  
        self.items.append(item)
```

```
    def dequeue(self):  
        return self.items.pop(0)
```

# Implementação como Lista Encadeada

Considerando as operações abaixo, como implementar uma fila?

- Inserir no início
- Inserir no meio
- Inserir no fim
- Deletar do início
- Deletar do meio
- Deletar do fim



# Implementação como Lista Encadeada

Considerando as operações abaixo, como implementar uma fila?

- Inserir no início
- Inserir no meio
- Inserir no fim
- Deletar do início
- Deletar do meio
- Deletar do fim

# Implementação como Lista Encadeada

Considerando as operações abaixo, como implementar uma fila?

- Inserir no início
- Inserir no meio
- Inserir no fim
- Deletar do início
- Deletar do meio
- Deletar do fim

# Implementação como Lista Encadeada

```
class Fila:
```

```
    def __init__(self):
```

```
        self.cabeca = None
```

```
        self.cauda = None
```

```
    def enqueue(self, valor):
```

```
        novo_no = No(valor)
```

```
        if self.cabeca is None:
```

```
            self.cabeca = self.cauda = novo_no
```

```
        else:
```

```
            self.cauda.proximo = novo_no
```

```
            self.cauda = novo_no
```

```
    def dequeue(self):
```

```
        removido = self.cabeca.valor
```

```
        if self.cabeca == self.cauda:
```

```
            self.cabeca = self.cauda = None
```

```
        else:
```

```
            self.cabeca = self.cabeca.proximo
```

```
        return removido
```

# Para casa

Usando as classes `ArrayFila` e `Fila`:

- Faça um método que imprima a fila.
- Faça um método que verifique se a fila está vazia.
- Faça um método (`peek`) que retorne o primeiro elemento da fila (sem deletar).
- Faça um método que inverta a fila usando *dequeue* e *enqueue* e uma pilha auxiliar.
- Adicione tratamento de erros nos métodos *enqueue* e *dequeue* para fila vazia e fila cheia. Defina você mesmo um tamanho máximo.

# Bibliografia

