



# IMPLEMENTAÇÃO DE FILAS EM JAVASCRIPT

ESTRUTURA DE DADOS

CST em Desenvolvimento de Software Multiplataforma



PROF. Me. TIAGO A. SILVA



# LIVRO DE REFERÊNCIA DA DISCIPLINA

- **BIBLIOGRAFIA BÁSICA:**

- GRONER, Loiane. **Estrutura de dados e algoritmos com JavaScript**: escreva um código JavaScript complexo e eficaz usando a mais recente ECMAScript. **São Paulo: Novatec Editora, 2019.**

- **NESTA AULA:**

- **Capítulo 5** – Filas e Deques



# PARA SOBREVIVER AO JAVASCRIPT

Non-zero value



null



0



undefined



# FILA

*Exemplo de Implementação em  
JavaScript, sem o uso de funções  
nativas*

# IMPLEMENTAÇÃO DE UMA FILA

- Estrutura interna (**items**): Usamos um objeto para armazenar os elementos da fila. As chaves do objeto representam as posições dos elementos na fila.
  - **enqueue(elemento)**: Adiciona um novo elemento ao final da fila. O índice do fim da fila (**this.fim**) é incrementado cada vez que um novo elemento é inserido.
  - **dequeue()**: Remove o primeiro elemento da fila. O índice do início (**this.inicio**) é incrementado após a remoção de um elemento, e o item correspondente é excluído do objeto.
  - **front()**: Retorna o elemento na frente da fila sem removê-lo.
  - **isEmpty()**: Verifica se a fila está vazia, comparando os índices de início e fim.
  - **size()**: Calcula o número de elementos na fila subtraindo o índice do início do índice do fim.
  - **clear()**: Limpa completamente a fila, reiniciando os valores.
- Essa implementação manual simula o comportamento de uma fila usando apenas um objeto como armazenamento interno e gerenciando manualmente os índices para garantir o funcionamento correto da fila sem o uso de funções nativas como **push()** e **shift()**.

```
1  class Fila {
2      constructor()
3      {
4          this.items = {}; // Usamos um objeto para armazenar os itens
5          this.inicio = 0; // Representa o índice do início da fila
6          this.fim = 0;    // Representa o índice do fim da fila
7      }
8
9      // Adiciona um elemento ao final da fila (enqueue)
10     enqueue(elemento)
11     {
12         this.items[this.fim] = elemento; // Coloca o elemento no fim da fila
13         this.fim++; // Incrementa o índice do fim da fila
14     }
```

```
15
16 // Remove e retorna o primeiro elemento da fila (dequeue)
17 dequeue()
18 {
19     if (this.isEmpty()) {
20         return undefined; // Se a fila estiver vazia, retorna undefined
21     }
22
23     const item = this.items[this.inicio]; // Obtém o primeiro elemento
24     delete this.items[this.inicio]; // Remove o item do início da fila
25     this.inicio++; // Move o índice do início para o próximo item
26
27     // Quando o início e o fim estiverem alinhados, redefine a fila
28     if (this.inicio === this.fim) {
29         this.inicio = 0;
30         this.fim = 0;
31     }
32
33     return item; // Retorna o item removido
34 }
35
36 // Retorna o primeiro elemento da fila sem removê-lo (peek)
37 front()
38 {
39     if (this.isEmpty()) {
40         return undefined; // Se a fila estiver vazia, retorna undefined
41     }
42     return this.items[this.inicio]; // Retorna o primeiro elemento
43 }
```

```
44 // Verifica se a fila está vazia
```

```
45 isEmpty()
```

```
46 {
```

```
47     return this.fim === this.inicio; // Verifica se os índices estão iguais
```

```
48 }
```

```
49  
50 // Retorna o tamanho da fila
```

```
51 size()
```

```
52 {
```

```
53     return this.fim - this.inicio; // Calcula a diferença entre fim e início
```

```
54 }
```

```
55  
56 // Limpa a fila
```

```
57 clear()
```

```
58 {
```

```
59     this.items = {};
```

```
60     this.inicio = 0;
```

```
61     this.fim = 0;
```

```
62 }
```

```
63  
64 }
```



# FILA – COMO USAR

```
65
66 // Exemplo de uso
67 let minha_variavel = new Fila();
68
69 minha_variavel.enqueue("Cliente 1");
70 minha_variavel.enqueue("Cliente 2");
71 minha_variavel.enqueue("Cliente 3");
72
73 console.log(minha_variavel.front()); // Saída: "Cliente 1"
74
75 console.log(minha_variavel.dequeue()); // Saída: "Cliente 1"
76 console.log(minha_variavel.dequeue()); // Saída: "Cliente 2"
77
78 minha_variavel.enqueue("Cliente 4");
79
80 console.log(minha_variavel.size()); // Saída: 2 (Cliente 3 e Cliente 4 ainda estão na fila)
81 console.log(minha_variavel.front()); // Saída: "Cliente 3"
```

## EXERCÍCIOS

*Use a classe implementada acima para  
resolver os exercícios*

# EXERCÍCIO 1

- Crie um sistema de atendimento onde clientes entram na fila e são atendidos na ordem de chegada. A cada iteração, um cliente deve ser removido da fila e exibido no console. Novos clientes podem ser adicionados aleatoriamente.
- **Desafio Extra:** Simule diferentes tempos de atendimento para cada cliente, usando `setTimeout()`.

## EXERCÍCIO 2

- Simule uma fila de impressão onde diferentes documentos são adicionados à fila. Cada documento deve ter um nome e um tamanho em páginas. Ao processar a fila, exiba no console qual documento está sendo impresso e remova-o da fila após a "impressão".
- **Desafio Extra:** Limite a capacidade da fila (ex: 5 documentos) e exiba uma mensagem quando a fila estiver cheia.

## EXERCÍCIO 3

- Imagine um brinquedo que comporta apenas um passageiro por vez. Os visitantes entram na fila e são chamados um por um para embarcar. Crie um sistema que adicione visitantes à fila e simule o embarque e desembarque.
- **Desafio Extra:** Dê um tempo fixo para cada embarque usando **setTimeout()** e adicione um sistema de prioridade para visitantes VIP.

## EXERCÍCIO 4

- Simule um restaurante onde os pedidos são adicionados a uma fila e preparados na ordem correta. Cada pedido deve ter um número e um nome (ex: "Pedido 1: Hambúrguer"). À medida que são preparados, os pedidos devem ser removidos da fila.
- **Desafio Extra:** Implemente um sistema onde os pedidos demoram tempos diferentes para serem preparados.

## EXERCÍCIO 5

- Simule um caixa de supermercado onde os clientes entram na fila e passam pelo caixa um por um. Cada cliente pode ter um número aleatório de produtos (entre 1 e 10). O tempo de atendimento deve ser proporcional ao número de produtos.
- **Desafio Extra:** Crie várias filas e distribua os clientes entre elas para otimizar o tempo de atendimento.

# OBRIGADO!

- Encontre este **material on-line** em:
  - Slides: Plataforma Microsoft Teams
- Em caso de **dúvidas**, entre em contato:
  - **Prof. Tiago:** [tiago.silva238@fatec.sp.gov.br](mailto:tiago.silva238@fatec.sp.gov.br)

