

## ESTRUTURAS DE DADOS 1

# Trabalho 1

## Filas

Prof. John Lenon C. Gardenghi  
17 de outubro de 2019

**Instruções**

1. O presente trabalho vale NOTA, e comporá a média final de trabalhos.
2. O trabalho poderá ser feito individualmente ou em **dupla**.
3. Compacte sua solução em um arquivo com nome

`nome_matricula_nome_matricula.zip`

Esse arquivo deve conter:

- (a) Seu fonte com a função `main`.
- (b) Os pares das 3 bibliotecas implementadas.

Portanto, o arquivo compactado deve conter **7 arquivos** sendo 4 arquivos `.c` e 3 arquivos `.h`. Nomeie-os como preferir, mas lembre-se que os pares `.c` e `.h` que definem uma biblioteca **devem possuir mesmo nome**.

4. Submeta o arquivo compactado na página da disciplina na plataforma Aprender.
5. Atente-se ao prazo de submissão. A data limite para submissão é dia **31 de outubro de 2019, às 23:55**. Depois deste prazo, **as submissões estão bloqueadas**.

O objetivo do presente trabalho é utilizar os conceitos vistos de fila para um entendimento aprofundado da estrutura. Para tanto,

(a) Implemente 3 bibliotecas de filas de inteiro, a saber,

1. Fila de **vetor com tamanho fixo**.
2. Fila **circular com redimensionamento**.
3. Fila de **lista encadeada circular**.

As 3 bibliotecas devem oferecer as seguintes funções, atendendo aos seguintes requisitos:

1. **Criação da fila**

Deve inicializar as estruturas necessárias a fila e os indicadores.

## 2. Inserção na fila

Deve inserir um novo elemento na fila, retornando 1 se o elemento foi inserido com sucesso, 0 caso contrário.

## 3. Remoção da fila

Deve remover um elemento da fila, e retornar 1 caso a remoção tenha sido bem sucedida (neste caso, retornando também o elemento removido) ou 0 caso contrário.

## 4. Verificação se a fila está cheia

Deve retornar 1 se a fila estiver cheia, 0 caso contrário.

## 5. Verificação se a fila está vazia

Deve retornar 1 se a fila estiver vazia, 0 caso contrário.

## 6. Tamanho da fila

Deve retornar quantos elementos há na fila.

## 7. Impressão da fila

Deve imprimir a fila no seguinte formato:

```
-----  
| xxx | xxx | xxx | xxx | xxx | xxx | xxx | xxx | xxx | xxx |  
-----  
      p                u
```

Note que, para a implementação com vetor, é necessário *imprimir o vetor fila inteiro*.

(b) Implemente um arquivo-fonte com uma função `main` que ofereça as seguintes opções:

1. Inserir `n` elementos na fila.
2. Remover `n` elementos na fila.
3. Imprimir a fila.
4. Reiniciar a fila.
5. Sair.

(c) Nas operações, respectivamente,

1. Pergunte ao usuário quantos elementos devem ser inseridos exibindo a mensagem “Digite quantos elementos você deseja inserir: ”, ou algo do tipo. **Garanta que o elemento inserido está entre 0 e 999.** *Cuidado: À medida que for lendo os elementos, vá inserindo-os na fila. Se nem todos os `n` elementos couberem na fila, exiba a seguinte mensagem: `x elementos foram inseridos na fila, mas n-x não couberam`.*
2. Pergunte ao usuário quantos elementos devem ser removidos exibindo a mensagem “Digite quantos elementos você deseja remover: ”, ou algo do tipo. Para cada remoção, imprima algo do tipo “Elemento removido: xxx”. *Cuidado: Remova elementos enquanto houver na fila. Caso a fila não possua mais elementos, remova o máximo possível e ao final avise ao usuário que não é possível mais remover pois a fila ficou vazia.*
3. Use a função de impressão da fila presente na biblioteca.
4. Reinicie a fila como se o programa estivesse sendo executado do zero.

(d) Observações

1. Sua implementação deve ser de tal forma que, mudando o **#include** de uma biblioteca para outra baste para executá-la. Essa mudança será feita, na correção, de forma **manual**. Por isso, *as implementações das 3 bibliotecas devem seguir o mesmo padrão para atenderem às chamadas na main*.
2. A função de impressão da fila **não faz parte da estrutura de dados**. Está sendo usada para este trabalho apenas para que vocês vejam, na tela, o que está acontecendo com a fila ao longo das operações.
3. Nas bibliotecas que implementam fila com vetor, defina o tamanho (inicial) do vetor como 5. Assim é possível verificar, na tela, o que acontece.
4. Teste bem seu programa *com as 3 bibliotecas implementadas*. Certifique-se que seu programa não será encerrado de forma inesperada por executar alguma operação indevida.

Bons estudos!

Prof. John Lenon C. Gardenghi  
john.gardenghi@unb.br  
Sala 22 - UED