



**Universidade Federal de Uberlândia**  
**Curso de Graduação em Gestão da Informação**  
**3º Trabalho de Estruturas de Dados I – Prof. Daniel A. Furtado**  
**Trabalho Individual – Filas**

**INTRUÇÕES GERAIS – LER ATENTAMENTE**

- Esta atividade deve ser realizada individualmente;
- Este trabalho deve ser feito utilizando a linguagem Python. Trabalhos implementados utilizando outras linguagens serão anulados;
- **Todo o código deve ser colocado em um único arquivo**, pois apenas um arquivo (.py) deve ser entregue. Não crie arquivos adicionais para as classes (o único arquivo a ser entregue deve conter todas as definições de classes, assim como o programa principal).
- **Não compactar** os arquivos no momento da entrega;
- Os recursos adequados da linguagem devem ser utilizados;
- A eficiência, clareza e coerência do código também serão avaliadas;
- Esteja atento às **observações sobre plágio** apresentadas no final deste documento. Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet, de trabalhos de colegas de classe ou de semestres anteriores serão anulados;
- As implementações não devem conter qualquer conteúdo de caráter imoral, desrespeitoso, pornográfico, discurso de ódio, desacato, etc.;
- O trabalho deve ser entregue até a data definida pelo professor em aula síncrona, pelo Sistema de Aplicação de Testes (SAAT);
- Trabalhos enviados por e-mail ou por outros meios digitais (como o MS Teams) **não serão considerados** (veja instruções no final);

### **Parte 1**

---

Implementar por completo o TAD Fila em *array* (com lógica circular) utilizando as definições de classe e métodos apresentadas nos slides de aula.

Acrescentar um método de nome **ShowItems** para exibição de todos os elementos da fila, do primeiro ao último, sem remover os elementos e sem fazer qualquer alteração na Fila.

Criar um código principal para exemplificar o uso da fila. Inserir elementos, remover, mostrar etc. Estudar em detalhes os métodos e o funcionamento geral da fila

### **Parte 2**

---

Uma **Deque** é uma estrutura de dados linear que permite a inserção e a remoção de elementos nas duas extremidades. Por ter essa característica, uma Deque pode ser utilizada para atuar tanto como uma pilha quanto como uma fila, desde que apenas as operações em questão sejam utilizadas.

Utilize a lógica de *array* circular proposta nos slides de aula (Módulo 3 - Filas) em conjunto com os “ponteiros” **Início** e **Final** para implementar a estrutura **Deque** como um tipo abstrato de dados. A classe deve ter os seguintes métodos públicos:

- **AddRear** – para adicionar um novo elemento no final da estrutura
- **RemoveRear** – para remover e **retornar** o elemento do final da estrutura
- **AddFront** – para adicionar um novo elemento no início da estrutura
- **RemoveFront** – para remover e **retornar** o elemento do início da estrutura

**Dicas:** na implementação das operações **AddRear** e **RemoveFront**, os ponteiros precisam avançar no sentido **horário** utilizando incremento com módulo, de forma similar às operações *Enqueue* e *Dequeue*. Nas operações **AddFront** e **RemoveRear**, os ponteiros precisam avançar no sentido **anti-horário**. Neste caso, quando atingirem a posição 0, a próxima atualização anti-horária deve coloca-los na posição **n-1** do *array*, onde **n** é o tamanho do *array*.

Após implementação da classe referente ao TAD Deque, mostre, no programa principal, um exemplo utilizando o TAD e seus métodos.

**OBS:** Ao término do trabalho o programa principal deve conter todos os trechos de código de teste, na sequência das partes do exercício. Ao ser executado, primeiramente o código deve mostrar os testes relacionados ao TAD Deque. Em seguida, deve mostrar os testes relacionados ao TAD Fila em lista ligada.

## Entrega

Um único arquivo (.py) contendo todas as definições de classes e também o programa principal deve ser enviado pelo Sistema de Aplicação de Testes (SAAT) até a data limite indicada pelo professor. Não envie arquivos compactados ou outros arquivos do editor ou da IDE de desenvolvimento que não sejam o arquivo de código principal.

## Sobre Eventuais Plágios

Este é um trabalho individual. Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.