

## Lista de TAD Fila

- 1) Implemente o TAD Fila na forma de um *wrapper* para uma lista encadeada dinâmica (ver dica abaixo)

Dica para os exercícios 1

- Crie um arquivo .h com a definição da struct aluno (aluno.h)
- Altere seus códigos de listas de forma a retirar a definição da struct aluno do arquivo .h
- Inclua no .h de suas listas um comando #include aluno.h
- Inclua no .h de sua fila um comando #include aluno.h
- No arquivo .c da fila faça um include do .h da lista. Não o inclua no .h pois dessa forma as funções da lista ficariam visíveis no main.
- No main faça o include do .h da fila

- 2) Implemente o TAD Fila utilizando alocação dinâmica. Dica: muito código será semelhante ao de lista encadeada. Não coloque a definição da struct aluno no .h da fila.
- 3) Altere o TAD Fila para que suporte filas de tamanhos variáveis, mas que ainda utilize vetores para armazenar os elementos da fila.

Ao criar a fila, passe como parâmetro um tamanho inicial máximo de elementos da fila

Ex: 100 → li = queue\_create(100);

Caso o usuário tente inserir mais elementos do que a fila suporta, o TAD deve automaticamente realocar mais memória, adicionando a mesma quantidade inicial de elementos. Ou seja, se ao criar o TAD o usuário alocou inicialmente 100 elementos, quando o usuário tentar inserir o 101º elemento o TAD deverá alocar antes mais 100 elementos, totalizando 200 elementos no máximo. Se o usuário posteriormente tentar inserir o 201º elemento, mais 100 elementos serão alocados, totalizando 300 no máximo.

Ao remover elementos a memória já previamente alocada deve ser mantida. No entanto, crie uma função chamada compactar\_lista, que reduza o tamanho do vetor da fila para o menor tamanho possível que seja múltiplo do tamanho inicial da fila.

Exemplo:

Fila com 1000 posições máximas

Atualmente com 560 elementos.

Tamanho inicial: 100

Ao reduzir a fila ficará com no máximo 600 posições alocadas.

$\text{Ceil}(560/100) * 100$

Obs: exercício semelhante já foi feito utilizando listas. No entanto, observe que para este exercício, que é um TAD Fila, ao aumentar o tamanho do vetor pode ser que seja necessário

deslocar alguns elementos (por exemplo, se o índice para o último elemento é menor que o índice do primeiro elemento). O mesmo vale para a compactação do vetor.