

Aula 4

- **Exercício 1.** Implemente uma fila em alocação dinâmica para armazenar tarefas que devem ser executadas na ordem em que chegam. A fila deve conter os seguintes **dados**: `id_tarefa` do tipo inteiro, `desc_tarefa` do tipo vetor de char. Implemente os seguintes **métodos**: (a) `enfileira()` para inserir nova tarefa; (b) `desenfileira()` para remover uma tarefa; (c) `estaVazia()` que retorna *true* se a fila está vazia e *false* se não está vazia; (d) `criaFila()` para criar o nó cabeça da fila. Entregue o programa no TIDIA.
- **Exercício 2.** Implemente uma pilha para checar se a sequência de parênteses e colchetes de uma expressão matemática está correta. Assim, por exemplo, a expressão `(([[]]))` está correta, mas a expressão `((()()))` não está. Implemente os seguintes **métodos**: (a) `empilha()` para inserir um parênteses na pilha; (b) `desempilha()` para remover um parênteses da pilha; (c) `estaVazia()` que retorna *true* se a pilha está vazia e *false* se não está vazia; (d) `criaPilha()` para criar a pilha; (d) `checaExpressao()` que verifica se uma expressão dada está correta. Entregue o programa no TIDIA.