

Estrutura de Dados - MC202 A

1º Semestre de 2018

Tiago de Paula Alves - R.A.: 187679

3 de abril de 2018

Lista 1 - Exercício 1

```
1  /* Garante que a fila inicie vazia. */
2  void fila_criar(Fila *fila) {
3      fila->tamanho = 0;
4      fila->inicio = fila->fim = NULL;
5      /* o início é igual ao fim que é igual a NULL */
6  }
7
8  /* Garante que não ocorra vazamento da memória. */
9  void fila_destruir(Fila *fila) {
10
11      /* percorre a fila */
12      No *ptr = fila->inicio;
13      while(ptr != NULL) {
14          /* liberando a memória de cada nó */
15          free(ptr);
16          ptr = ptr->proximo;
17      }
18
19      /* zera a lista para evitar erros */
20      fila_criar(fila);
21  }
22
23  /* Adiciona um novo registro na fila. */
24  void fila_enfileirar(Fila *fila , Registro registro) {
25      /* novo nó, já inicializado com zeros */
26      No *novo = (No *) calloc(1, sizeof(No));
27      novo->registro = registro;
28
29      /* insere o nó */
30      if (fila->inicio == NULL) {
31          fila->inicio = novo;
32      } else {
33          fila->fim->proximo = novo;
34      }
35
36      /* atualiza o ponteiro para o final */
37      fila->fim = novo;
38      /* e o tamanho */
39      fila->tamanho++;
40  }
41
42  /* Remove o elemento mais antigo da fila. */
43  Registro fila_desenfileirar(Fila *fila) {
44      /* registro que deve ser retornado */
45      Registro resposta;
```

```

46     /* ponteiro para o nó que deve ser removido */
47     No *ptr = fila->inicio;
48
49     /* pega o registro */
50     resposta = ptr->registro;
51     /* remove o nó da fila */
52     fila->inicio = fila->inicio->proximo;
53     fila->tamanho--;
54     /* e libera a memória */
55     free(ptr);
56
57     return resposta;
58 }
59
60 /* Pega o elemento mais antigo da fila, mas sem removê-lo. */
61 Registro fila_pegar_proximo(Fila *fila) {
62     return fila->inicio->registro;
63 }
64
65 /* Tamanho da fila */
66 int fila_tamanho(Fila *fila) {
67     return fila->tamanho;
68 }

```