

# Listas Lineares

Prof. Flavio B. Gonzaga  
[flavio.gonzaga@unifal-mg.edu.br](mailto:flavio.gonzaga@unifal-mg.edu.br)  
Universidade Federal de Alfenas  
UNIFAL-MG

# Sumário

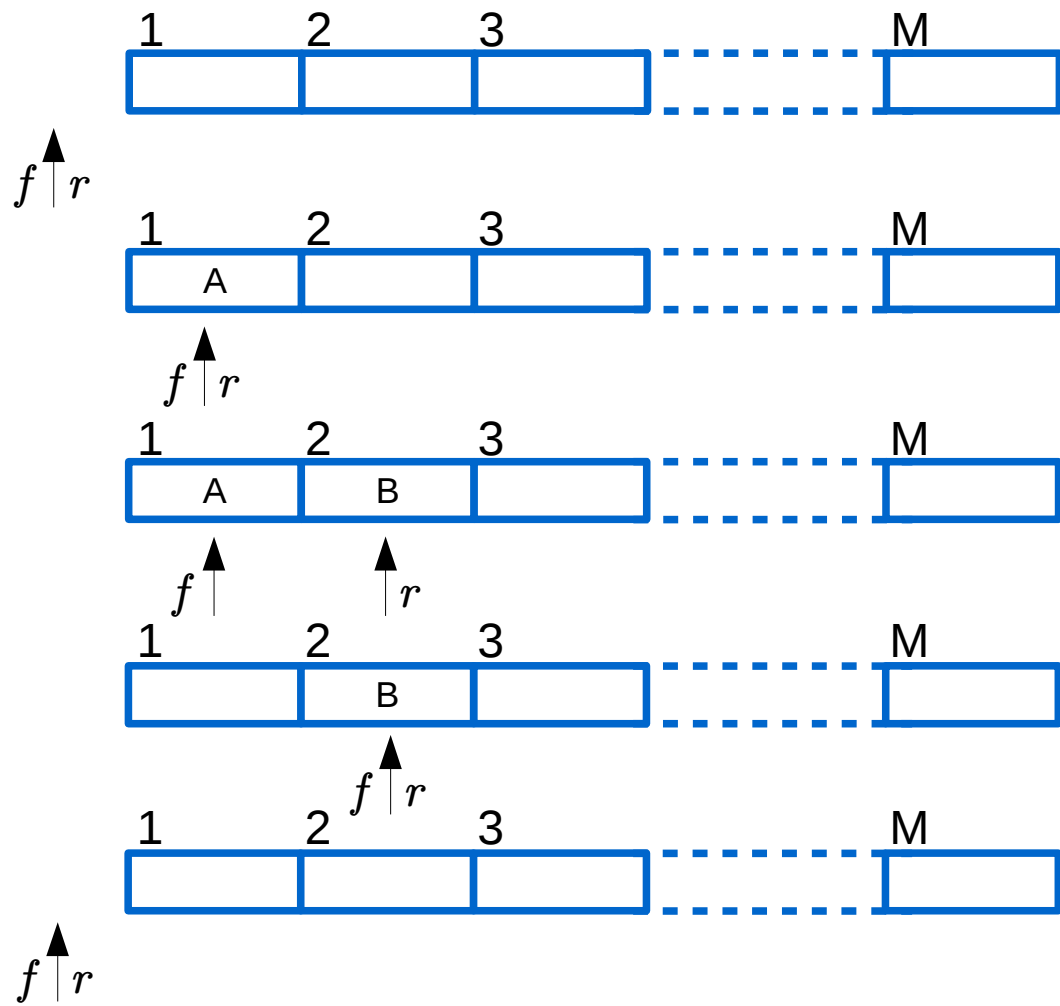
- Listas lineares
- Alocação sequencial
  - Filas
    - Inserção;
    - Remoção;

# Filas

- Filas exigem uma implementação um pouco mais elaborada;
- São necessários dois ponteiros: início de fila ( $f$ ) e retaguarda ( $r$ );
- Para adição de um elemento, move-se o ponteiro  $r$ ;
- Para a retirada de um elemento, move-se o ponteiro  $f$ ;
- A situação de fila vazia é representada por  $f = r = 0$ ;

- Na implementação de filas, faz-se necessário o armazenamento das localizações do *inicio\_de\_fila* ( $f$ ) e *retaguarda* ( $r$ ) da mesma;

## Filas



# Filas

- À medida que operações de inserção (e remoção) são executadas, os ponteiros se movimentam para a direita na memória;
- Tal ação gera a falsa impressão de esgotamento de posições disponíveis na memória;
- Para eliminar esse problema, consideram-se os  $M$  nós alocados como se estivessem em círculo, onde após  $F[M]$  retorna-se para  $F[1]$ ;

# Inserção

- Retorna:
  - $-1$  no caso de fila cheia;
  - A posição de inserção caso contrário;

função *insere*(*no*)

*insere*  $:= -1$

*prov*  $:= r \bmod M + 1$

se *prov*  $\neq f$  então

*r*  $:= prov$

*F*[*r*]  $:= no$

*insere*  $:= r$

se *f* = 0 então

*f*  $:= 1$

# Remoção

- Retorna:

- *nulo* no caso de fila vazia;
- o *nó* caso contrário;

função *remove*( )

*remove* := *nulo*

se  $f \neq 0$  então

*remove* :=  $F[f]$

se  $f = r$  então

$f := r := 0$

senão  $f := f \bmod M + 1$

# Referências Bibliográficas

- Estruturas de Dados e Seus Algoritmos. Szwarcfiter J. L.; Markenzon L.. 3a Edição. Editora LTC. 2010.
- Estruturas De Dados Usando C. Tenenbaum A. M.; Langsam Y.; Augenstein M. J.. 1a Edição. Editora Pearson. 1995.
- Introdução a Estruturas de Dados: Com Técnicas de Programação em C. Celes W.; Cerqueira R.; Rangel J.. 2a Edição. Editora Elsevier. 2017.