UniCesumar
EDUCAÇÃO PRESENCIAL E A DISTÂNCIA

Estrutura de Dados I

#2022 Ressignificar Realizar Reinventar

AADD

Estrutura de Dados do Tipo Fila

Prof. Erinaldo Sanches Nascimento

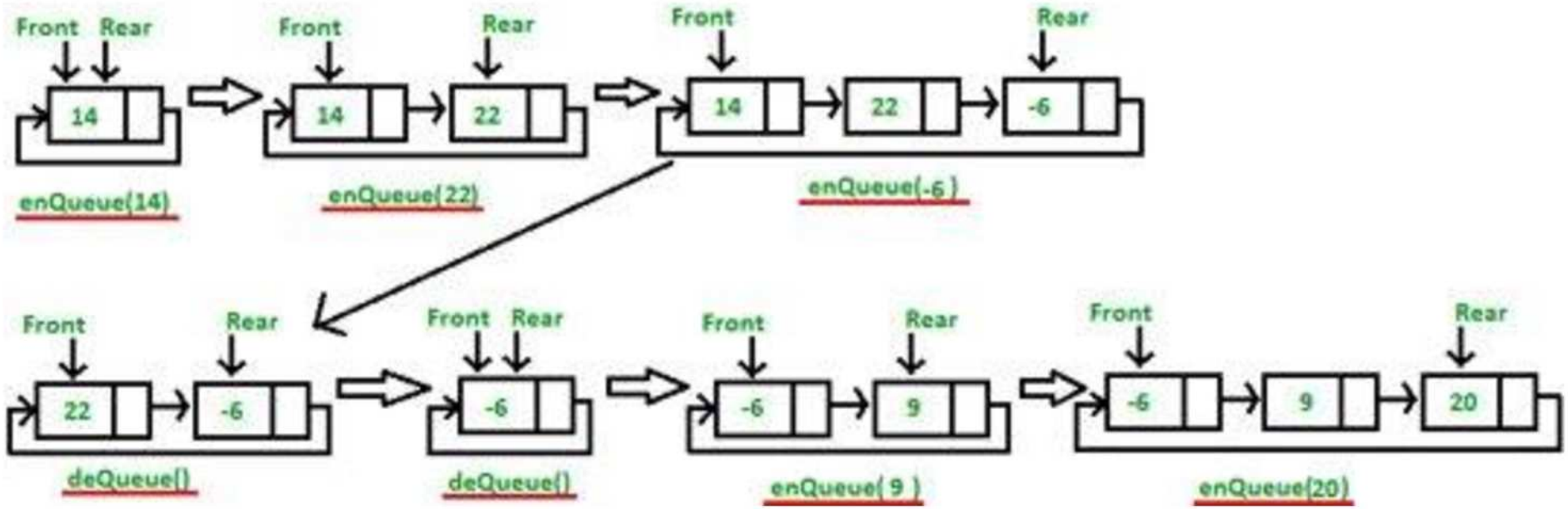
Fila é considerada FIFO (*First In First Out*)

Ascencio (2010, p. 191)

Unidade 2 e 3

Fila

1. Introdução
2. Lista Dinâmica como Fila
3. Fila Implementada com Vetor



Fila

Introdução

- Fila (*queue*) é uma estrutura de dados dinâmica que admite remoção e inserção de elementos.
- Mais especificamente, sempre que houver uma remoção, o elemento removido é o que está na estrutura há mais tempo.
- O primeiro objeto inserido na fila é também o primeiro a ser removido.
- Essa regra é conhecida por FIFO (*First-In-First-Out*).

Fila

Introdução

- Uma fila pode ser implementada:
  - Em um vetor (remover, inserir, *overflow*);
  - Distâncias em um grafo;
  - Varredura por níveis de uma árvore;
  - Implementação circular (adia o transbordamento);
  - Em vetor com redimensionamento (alocação dinâmica);
  - Em uma lista encadeada.

Fila

Lista Dinâmica como Fila

- Como administrar uma fila armazenada em uma lista encadeada?

```
5  typedef struct Fila{
6      int num;
7      struct Fila *prox;
8  }no;
```

- Esta lista terá uma célula-cabeça.

Fila

Lista Dinâmica como Fila

```
void inicia(no *f)
int vazia(no *f)
no *aloca()
void insere(no *f)
no *retira(no *f)
void exibe(no *f)
void libera(no *f)
```

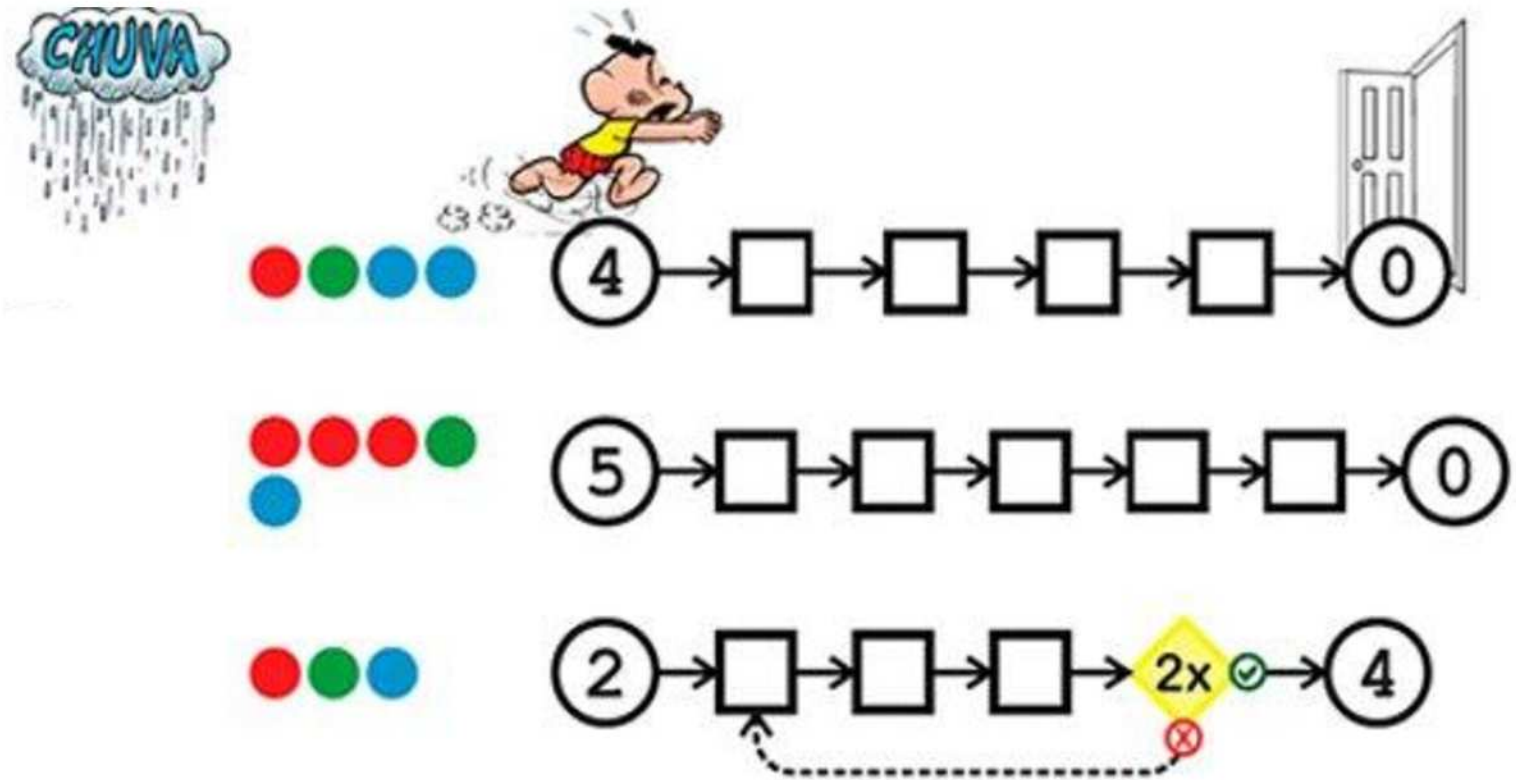
Vamos implementar?

```c
int main(void){
    no *f = (no *) malloc(sizeof(no));
    if(!f){
        printf("Sem memoria disponivel!\n");
        exit(1);
    }else{
        inicia(f);
    }
    insere(f);
    insere(f);
    insere(f);
    exibe(f);
    retira(f);
    exibe(f);
}
```

Vamos implementar?

Exercícios extra

Lista Dinâmica como Fila



1. Implemente a função inicia a fila.

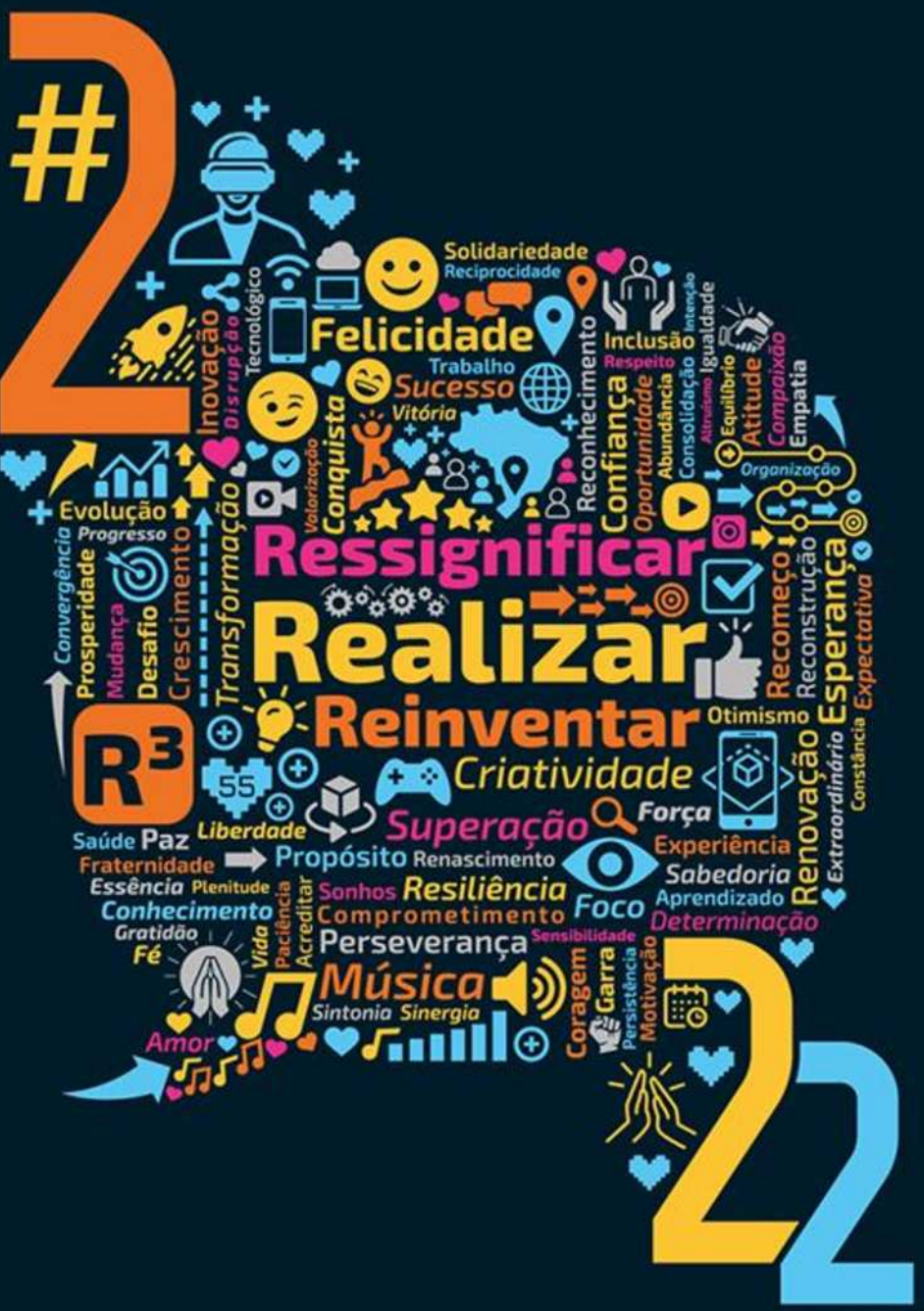
2. Implemente uma função de busca.

3. Implemente a função libera.

4. Imagine um banco. Implemente um sistema bancário de fila, que considere clientes prioritários. A cada 2 clientes sem necessidades prioritárias de chamar um cliente prioritário, até o máximo de 100 atendimentos diários.

Material Complementar

1. Ascencio, A. F. G. **Estrutura de dados:** algoritmos, análise da complexidade e implementações em Java e C/C++. São Paulo: Pearson Prentice Hall, 2010.
2. Tenenbaum, A. M. **Estruturas de dados usando C.** São Paulo: MAKRON Books, 1995.
3. Deitel, P.; Deitel, H. **Java:** Como programar. São Paulo: Pearson Education do Brasil, 2017.


#2022
Realizar
Ressignificar
Reinventar
R3
Solidariedade
Reciprocidade
Felicidade
Trabalho
Sucesso
Vitória
Inclusão
Respeito
Intenção
Igualdade
Tecnológico
Inovação
Disrupção
Reconhecimento
Confiança
Oportunidade
Abundância
Consolidação
Altruísmo
Equilíbrio
Atitude
Compaixão
Empatia
Organização
Valorização
Conquista
Evolução
Progresso
Convergência
Prosperidade
Mudança
Desafio
Crescimento
Transformação
Recomeço
Reconstrução
Esperança
Expectativa
Extraordinário
Constância
Otimismo
Criatividade
Renovação
Força
Superação
Experiência
Sabedoria
Aprendizado
Foco
Determinação
Sensibilidade
Liberdade
Saúde
Paz
Fraternidade
Propósito
Renascimento
Essência
Plenitude
Sonhos
Resiliência
Comprometimento
Perseverança
Conhecimento
Gratidão
Fé
Vida
Paciência
Acreditar
Música
Sintonia
Sinergia
Amor
Coragem
Garra
Persistência
Motivação
2022
#2022
AADD
UniCesumar
EDUCAÇÃO PRESENCIAL E A DISTÂNCIA