



Algoritmos Computacionais e Estruturas de Dados

Prof. Me. Felipe Borges

Prof. Felipe Borges

Doutorando em Sistemas de Potência – UFMA – Brasil

Mestre em Sistemas de Potência – UFMA – Brasil

MBA em Qualidade e Produtividade – FAENE – Brasil

Graduado em Engenharia Elétrica – IFMA – Brasil

Graduado em Engenharia Elétrica – Fontys – Holanda

Técnico em Eletrotécnica – IFMA – Brasil

Projetos e Instalações Elétricas – Engenharia – Banco do Brasil

Desenvolvimento e Gestão de Projetos – Frencken Engineering BV

Fila: Conceito

Conceito base: O primeiro a entrar é o primeiro a sair. Em Inglês (Fifo): **F**irst **I**n **F**irst **O**ut

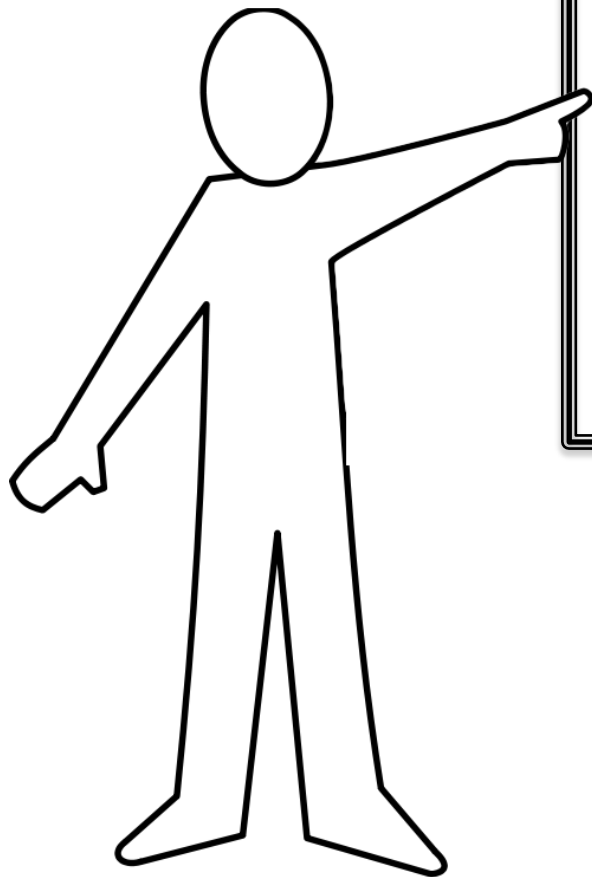


- Fila de banco
- Fila de pacientes
- Fila do supermercado

O Tipo Abstrato Fila

O Tipo Abstrato Fila

Ou seja, as operações que descrevem o comportamento da Fila.



Quais seriam estas
operações, são capazes de
enumerá-las ?

1. criarFila

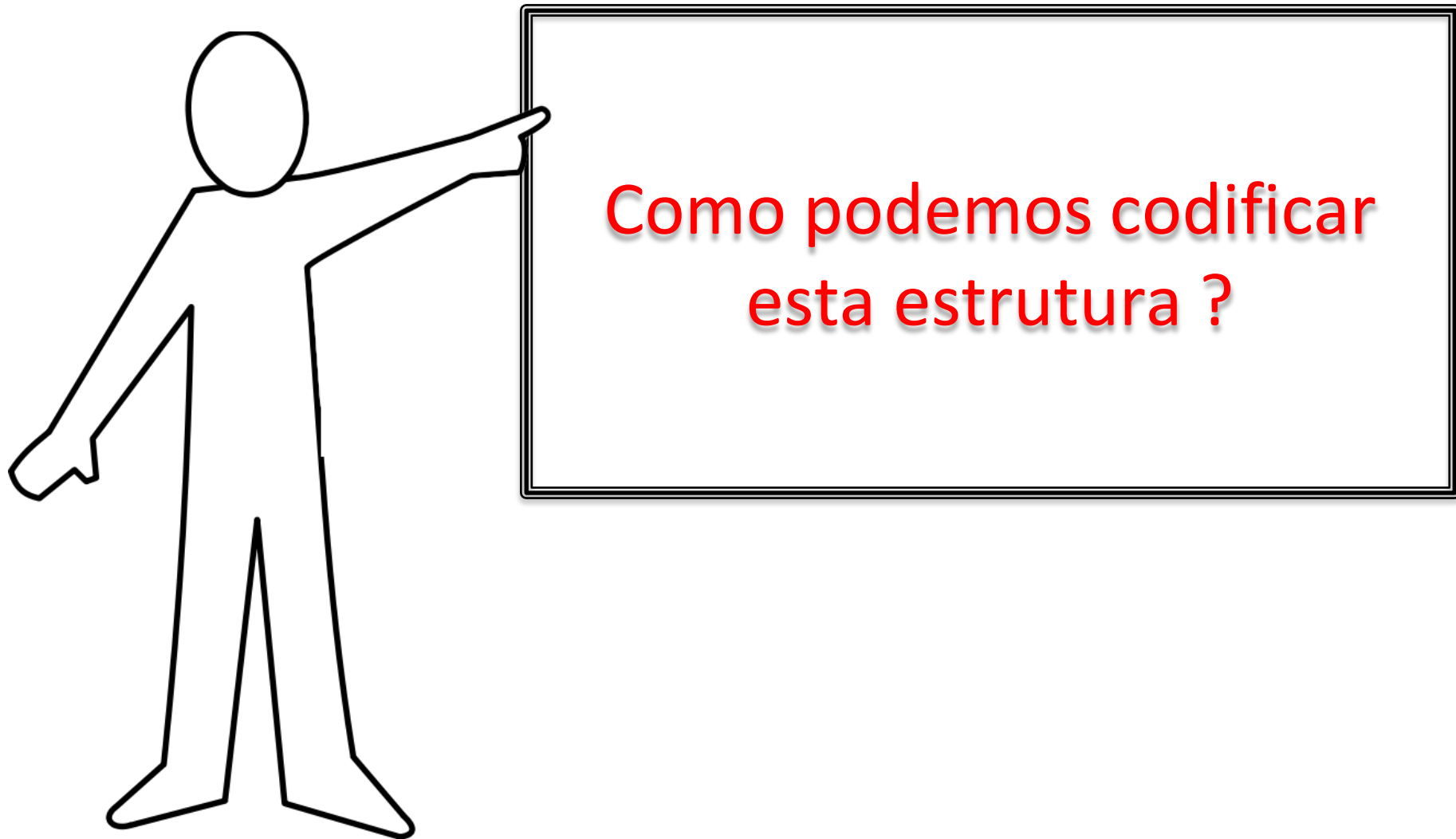
1. criarFila

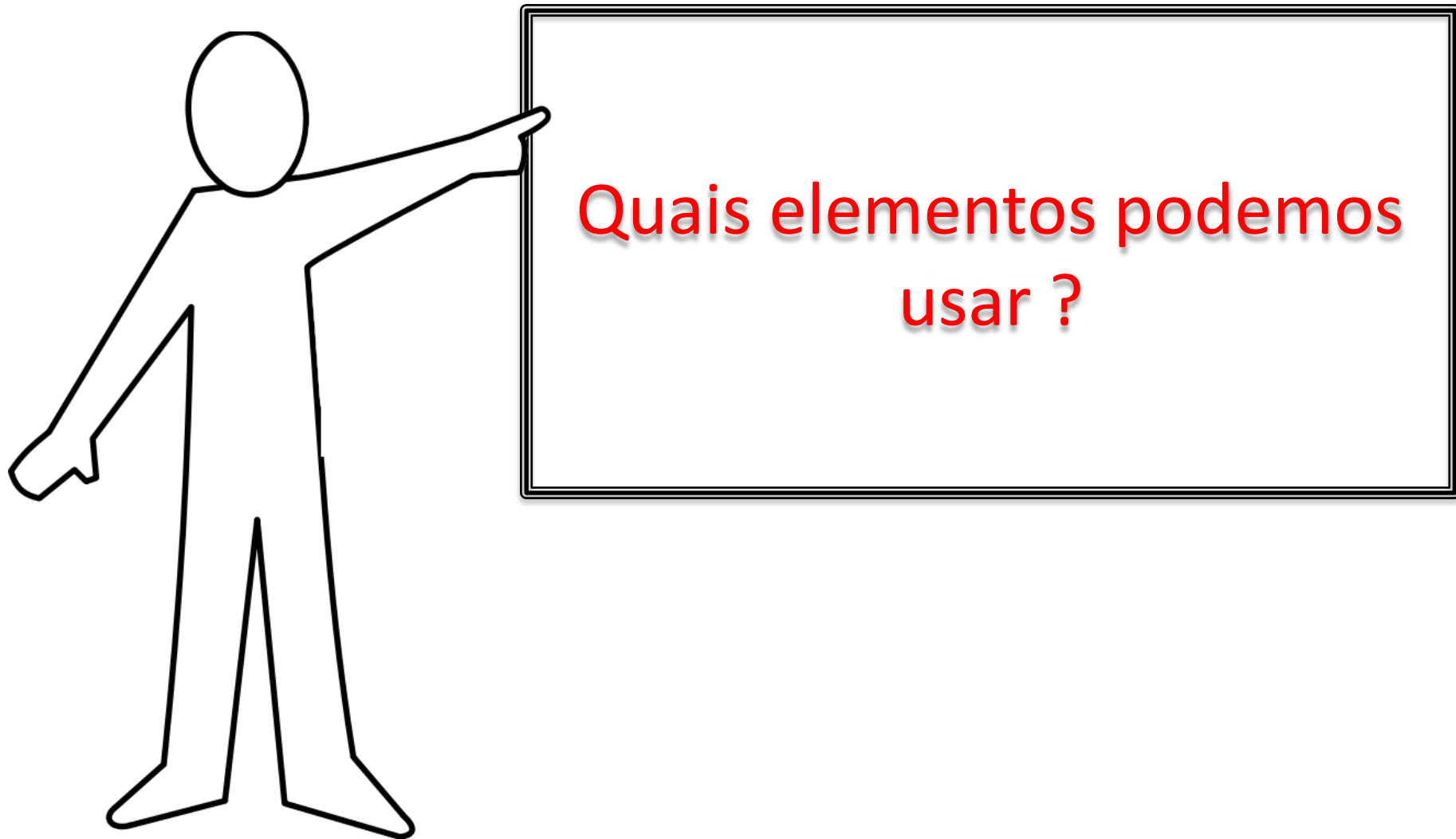
2. enfileirar

1. criarFila

2. enfileirar

3. desinfileirar





Fila

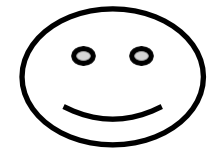
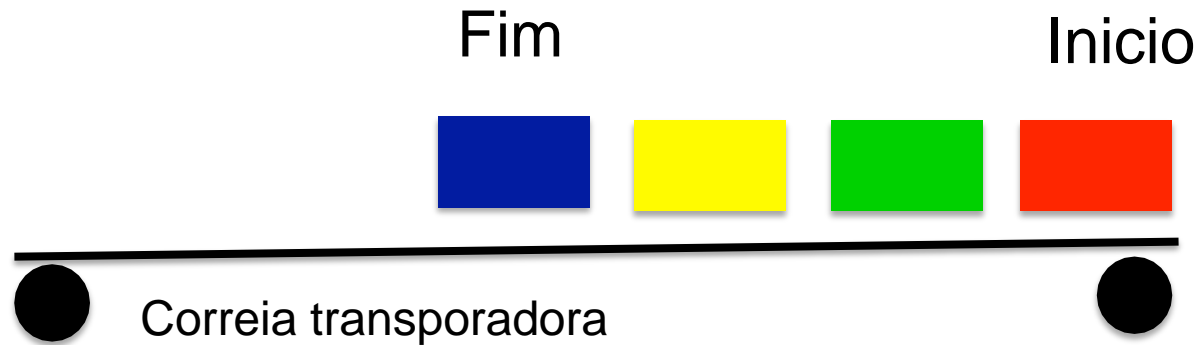
- Novamente, uma possibilidade é usar vetores, dado que estes permitem armazenar uma coleção de dados.
- Uma fila que usa vetor como estrutura básica é chamada de fila estática.
- Na Unidade II, veremos como codificar uma fila dinâmica usando listas encadeadas.



Estava pensando, e observei que na pilha a entrada e saída é pelo mesmo “lado”.
Porém na fila é diferente, quem chega vai para fim da fila e sai quem está no início da fila.

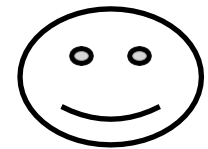
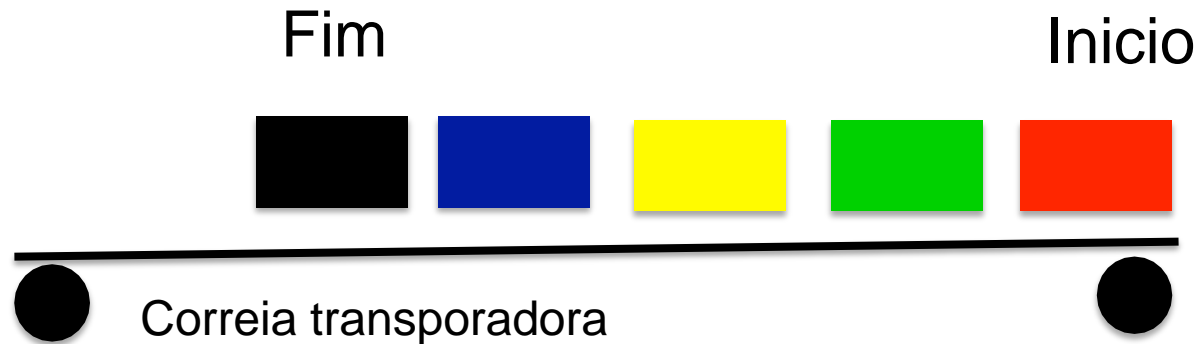


Imaginando, um trabalho em série, onde um conjunto de peças percorre uma correia até ela ir para empacotamento teríamos algo assim:



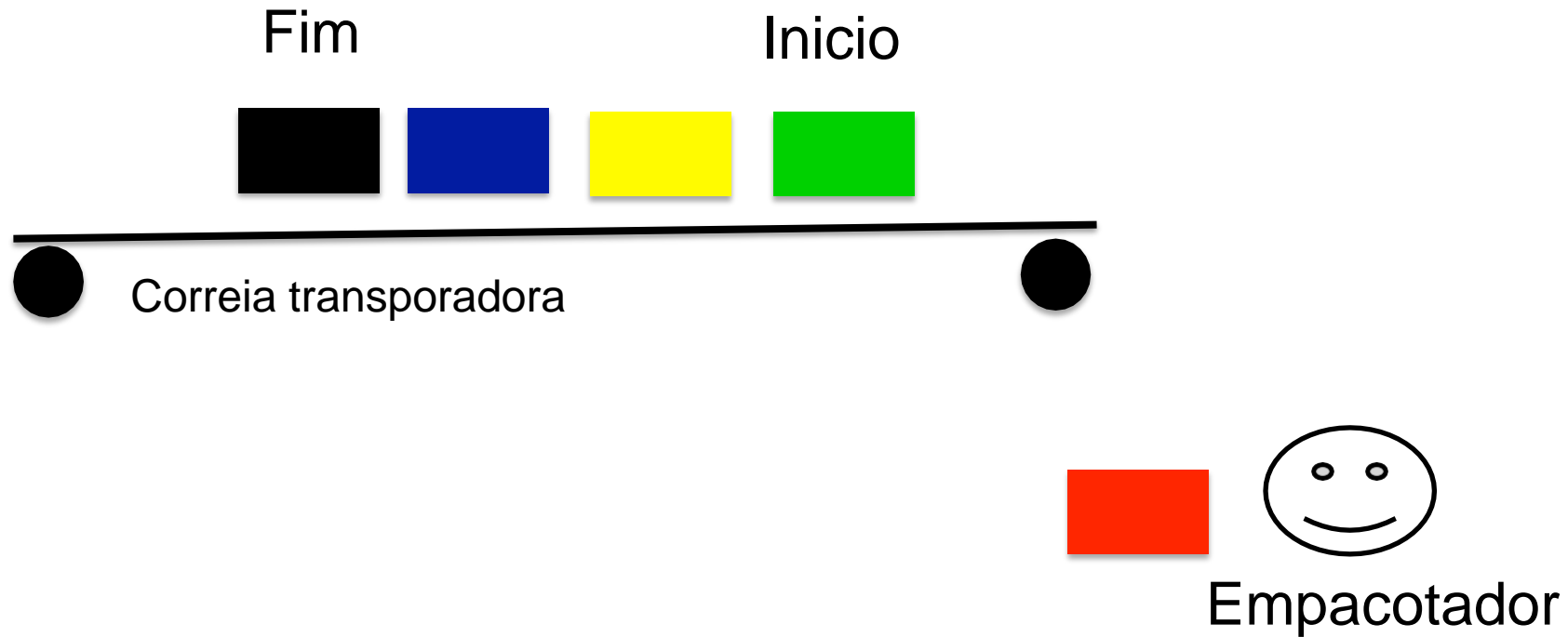
Empacotador

Quando chega um novo produto, ele é colocado no fim da fila.

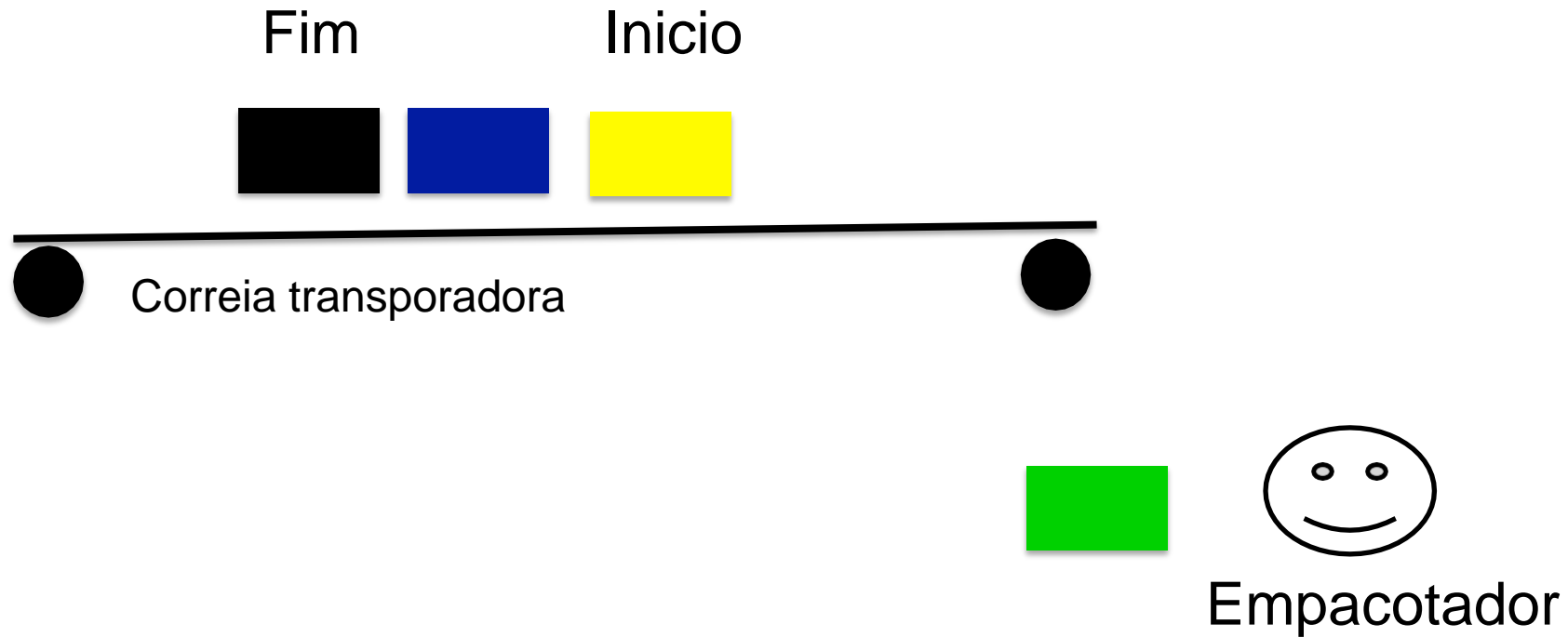


Empacotador

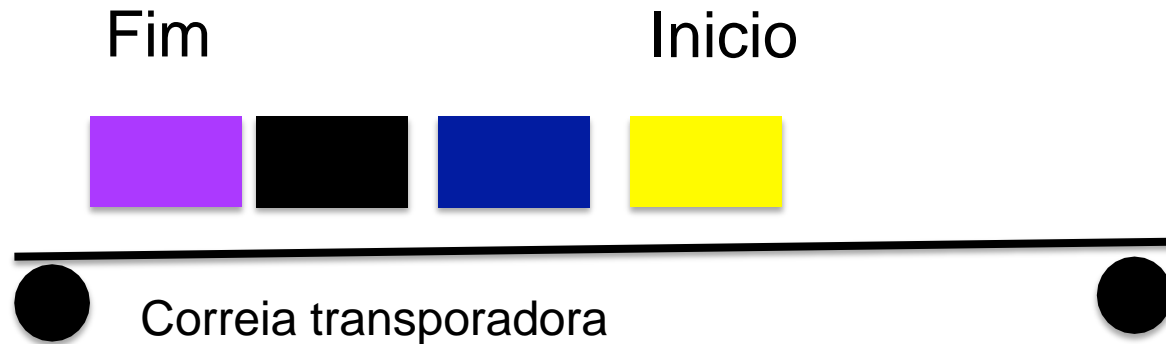
Quando um produto chega ao fim do processo, a peça anterior passa a ser o início da fila.



Quando um produto chega ao fim do processo, a peça anterior passa a ser o início da fila.



Quando chega um novo produto, ele é colocado no fim da fila.



Empacotador

CODIFICANDO ...

TAD: Fila

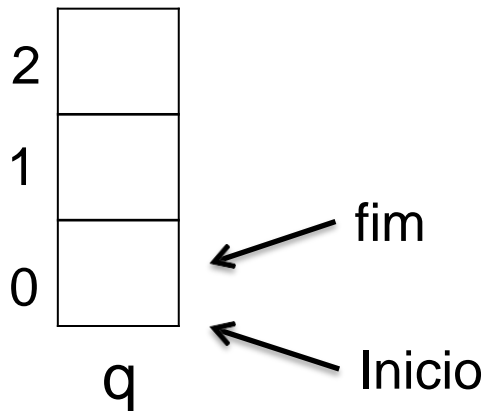
- Definindo o tipo de dados Fila. Observe, que tivemos que incluir os dois atributos, que marca o inicio e o fim da fila.

```
typedef struct {  
    int v[MAX];  
    int inicio, fim;  
} Fila;
```



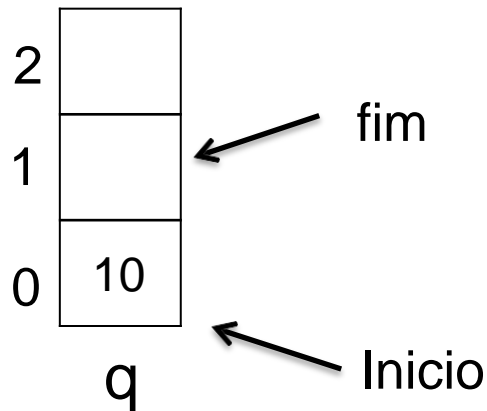
Antes de codificar vou pensar um pouco.
Eu sei que quando eu enfileiro estarei
incrementando o fim e quando eu
desenfileiro eu incremento o inicio. Mas
vou simular antes de codificar.

Considerando uma fila de no máximo 3 elementos:



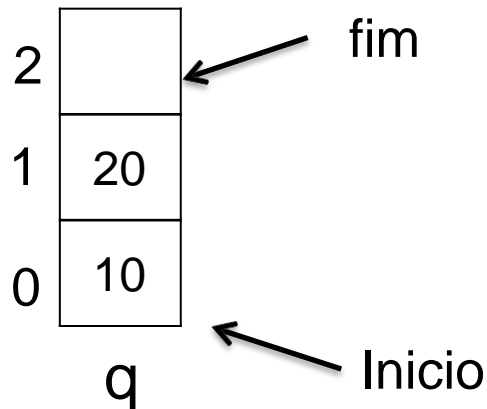
```
Fila *q = criaFila ();  
enqueue (q, 10);  
enqueue (q, 20);  
enqueue (q, 30);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
enqueue (q, 50);  
enqueue (q, 60);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));
```

Considerando uma fila de no máximo 3 elementos:



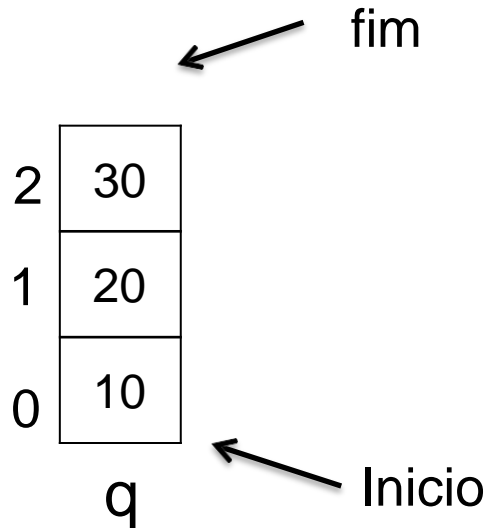
```
Fila *q = criaFila ();  
enqueue (q, 10);  
enqueue (q, 20);  
enqueue (q, 30);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
enqueue (q, 50);  
enqueue (q, 60);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));
```


Considerando uma fila de no máximo 3 elementos:



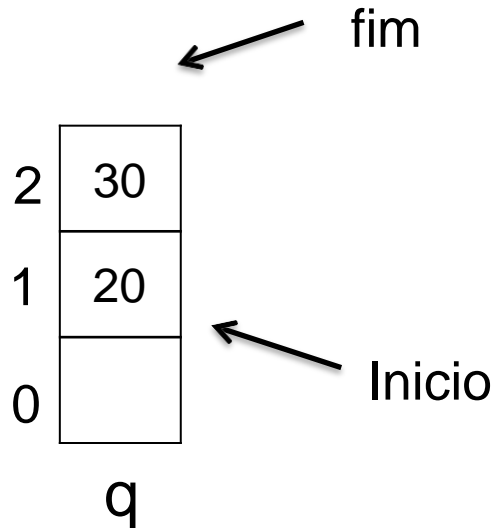
```
Fila *q = criaFila ();  
enfileira (q, 10);  
enfileira (q, 20);  
enfileira (q, 30);  
printf ("%d\n", desinfileira(q));  
printf ("%d\n", desinfileira(q));  
enfileira (q, 50);  
enfileira (q, 60);  
printf ("%d\n", desinfileira(q));  
printf ("%d\n", desinfileira(q));  
printf ("%d\n", desinfileira(q));
```

Considerando uma fila de no máximo 3 elementos:



```
Fila *q = criaFila ();
enqueue (q, 10);
enqueue (q, 20);
enqueue (q, 30);
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
enqueue (q, 50);
enqueue (q, 60);
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
```

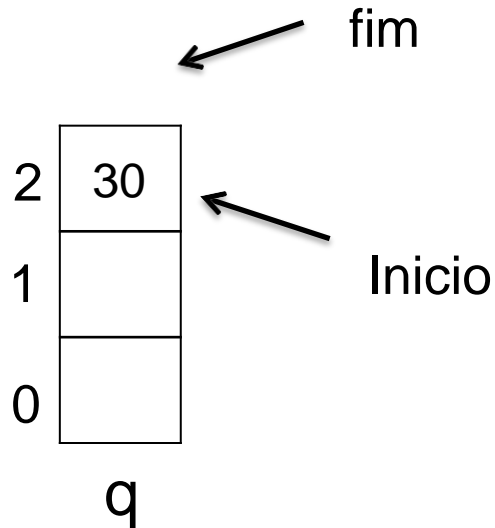
Considerando uma fila de no máximo 3 elementos:



```
Fila *q = criaFila ();
enqueue (q, 10);
enqueue (q, 20);
enqueue (q, 30);
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
enqueue (q, 50);
enqueue (q, 60);
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
```

10

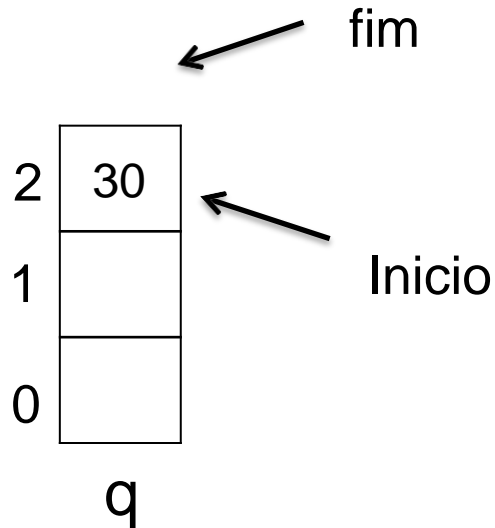
Considerando uma fila de no máximo 3 elementos:



10
20

```
Fila *q = criaFila ();  
enqueue (q, 10);  
enqueue (q, 20);  
enqueue (q, 30);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
enqueue (q, 50);  
enqueue (q, 60);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));
```

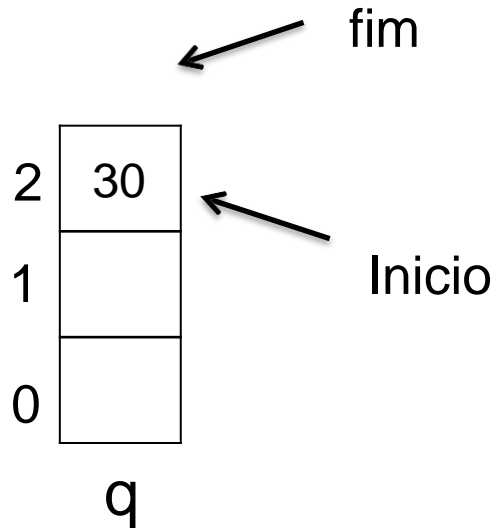
Considerando uma fila de no máximo 3 elementos:



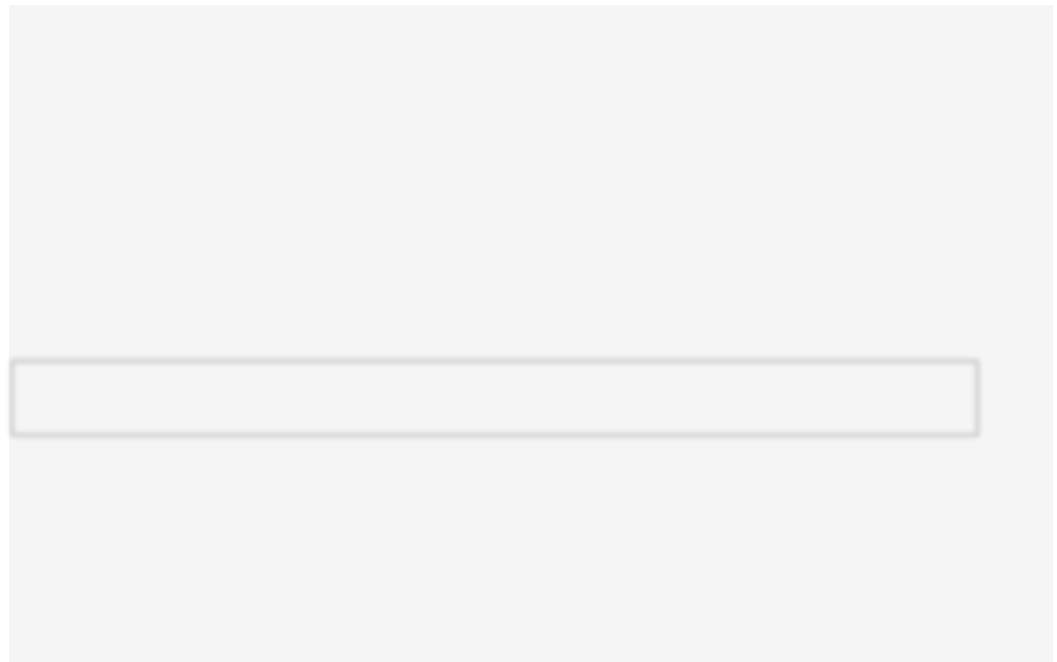
10
20

```
Fila *q = criaFila ();  
enqueue (q, 10);  
enqueue (q, 20);  
enqueue (q, 30);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
enqueue (q, 50);  
enqueue (q, 60);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));
```

Considerando uma fila de no máximo 3 elementos:

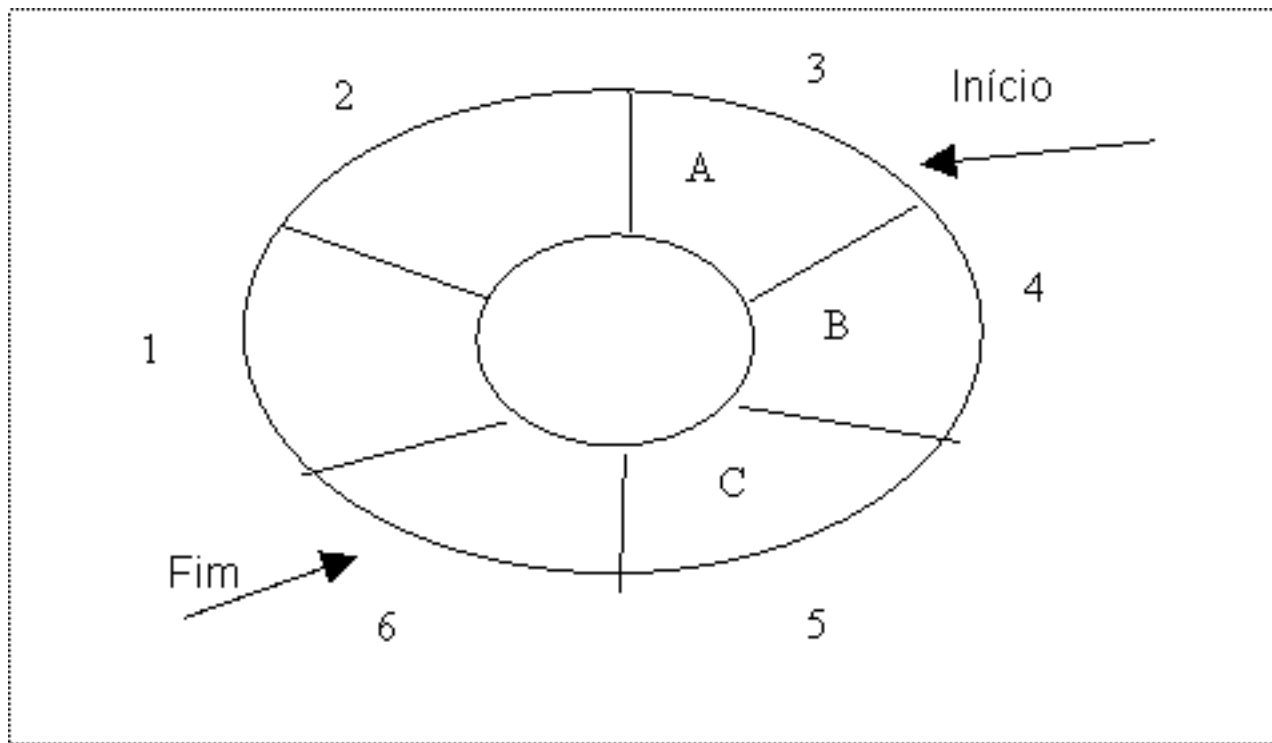


Erro: Mesmo tendo áreas vagas, eu não posso mais adicionar elemento.



TAD: Fila

- Solução, vetor circular:



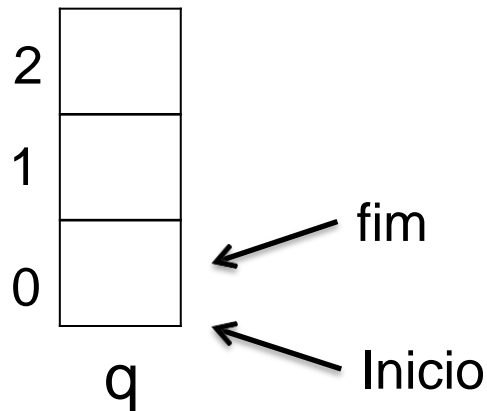
Depois do último elemento, temos o primeiro e antes do primeiro temos o último.

Fonte: <http://www2.dc.ufscar.br/~bsi/materiais/ed/u5.html>



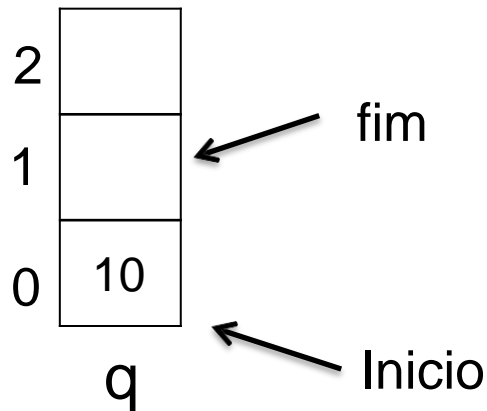
Agora vou simular novamente, mas considerando meu vetor circular.

Considerando uma fila de no máximo 3 elementos:



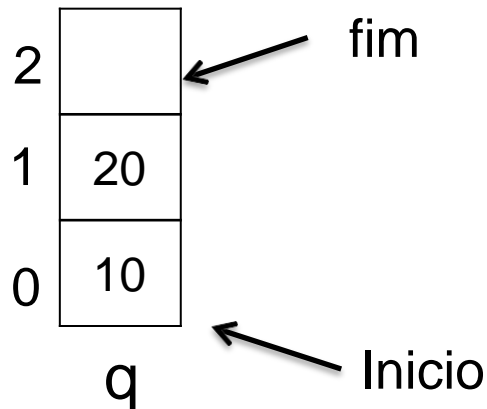
```
Fila *q = criaFila ();  
enqueue (q, 10);  
enqueue (q, 20);  
enqueue (q, 30);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
enqueue (q, 50);  
enqueue (q, 60);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));
```

Considerando uma fila de no máximo 3 elementos:



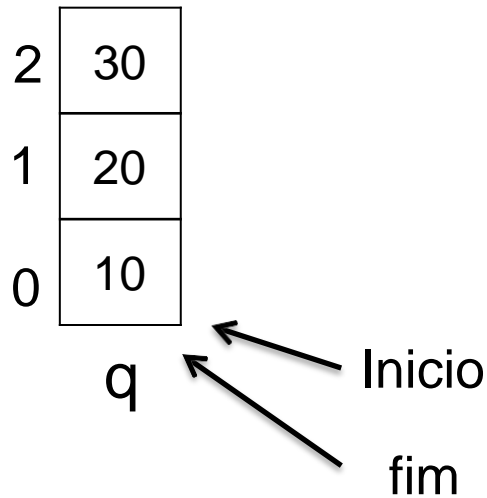
```
Fila *q = criaFila ();  
enqueue (q, 10);  
enqueue (q, 20);  
enqueue (q, 30);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
enqueue (q, 50);  
enqueue (q, 60);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));
```

Considerando uma fila de no máximo 3 elementos:



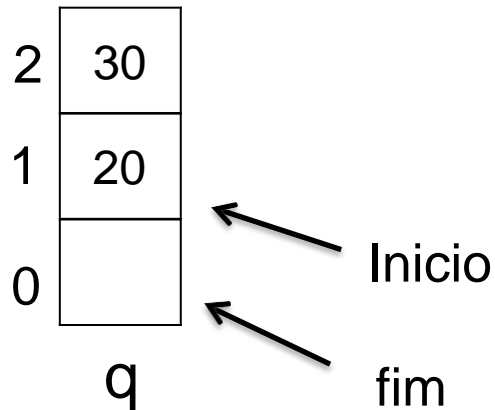
```
Fila *q = criaFila ();
enqueue (q, 10);
enqueue (q, 20);
enqueue (q, 30);
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
enqueue (q, 50);
enqueue (q, 60);
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
```

Considerando uma fila de no máximo 3 elementos:



```
Fila *q = criaFila ();
enqueue (q, 10);
enqueue (q, 20);
enqueue (q, 30);
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
enqueue (q, 50);
enqueue (q, 60);
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
```

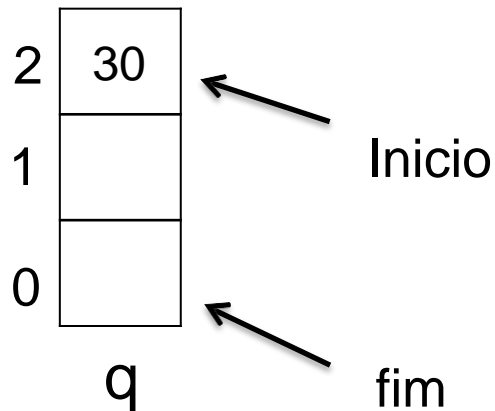
Considerando uma fila de no máximo 3 elementos:



```
Fila *q = criaFila ();
enqueue (q, 10);
enqueue (q, 20);
enqueue (q, 30);
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
enqueue (q, 50);
enqueue (q, 60);
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
printf ("%d\n", dequeue(q));
```

10

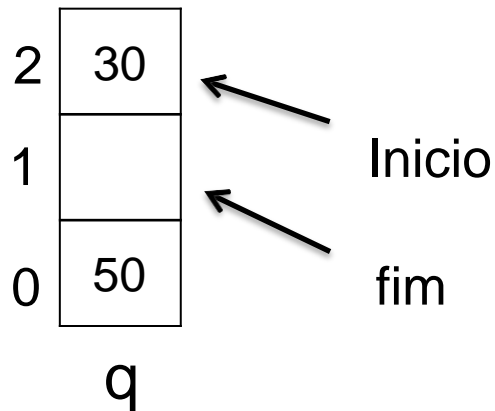
Considerando uma fila de no máximo 3 elementos:



10
20

```
Fila *q = criaFila ();  
enqueue (q, 10);  
enqueue (q, 20);  
enqueue (q, 30);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
enqueue (q, 50);  
enqueue (q, 60);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));
```

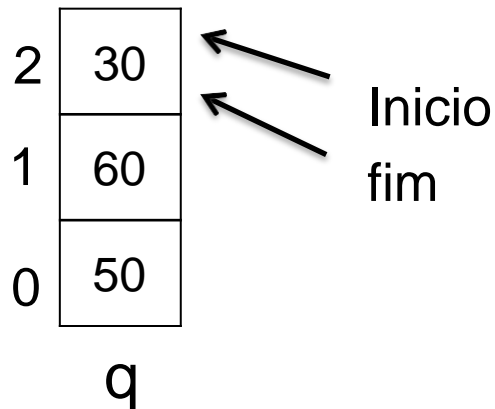
Considerando uma fila de no máximo 3 elementos:



10
20

```
Fila *q = criaFila ();  
enqueue (q, 10);  
enqueue (q, 20);  
enqueue (q, 30);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
enqueue (q, 50);  
enqueue (q, 60);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));
```

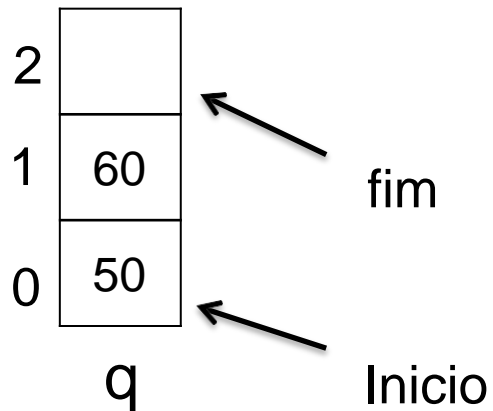
Considerando uma fila de no máximo 3 elementos:



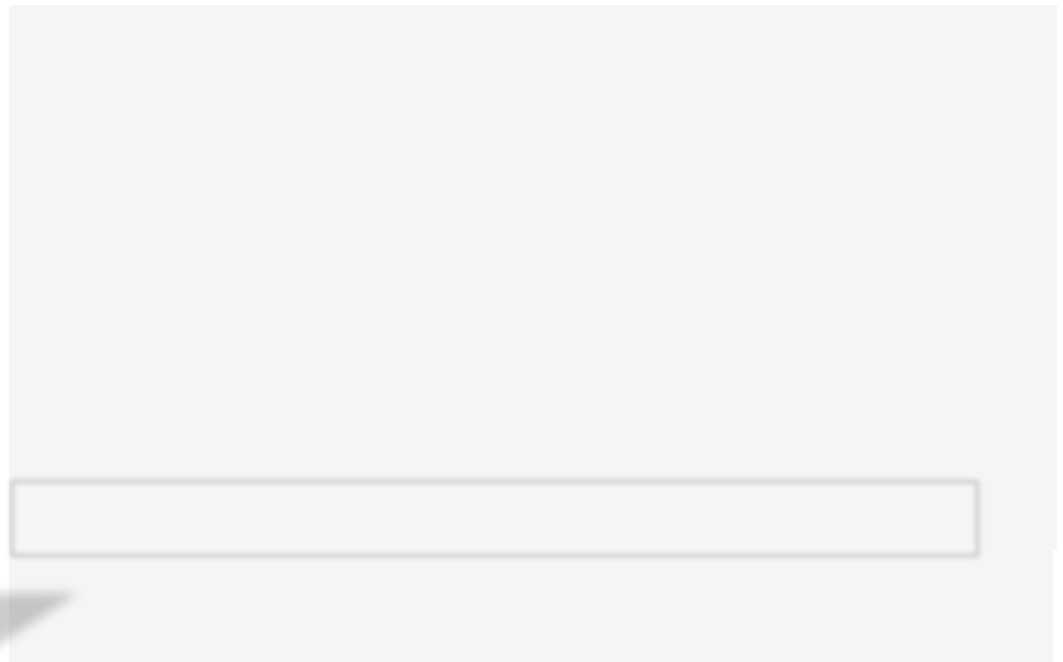
10
20

```
Fila *q = criaFila ();  
enqueue (q, 10);  
enqueue (q, 20);  
enqueue (q, 30);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
enqueue (q, 50);  
enqueue (q, 60);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));
```

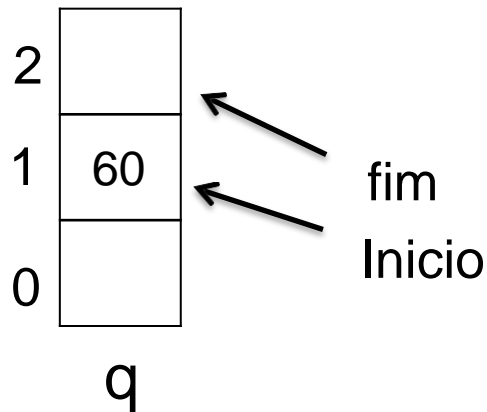

Considerando uma fila de no máximo 3 elementos:



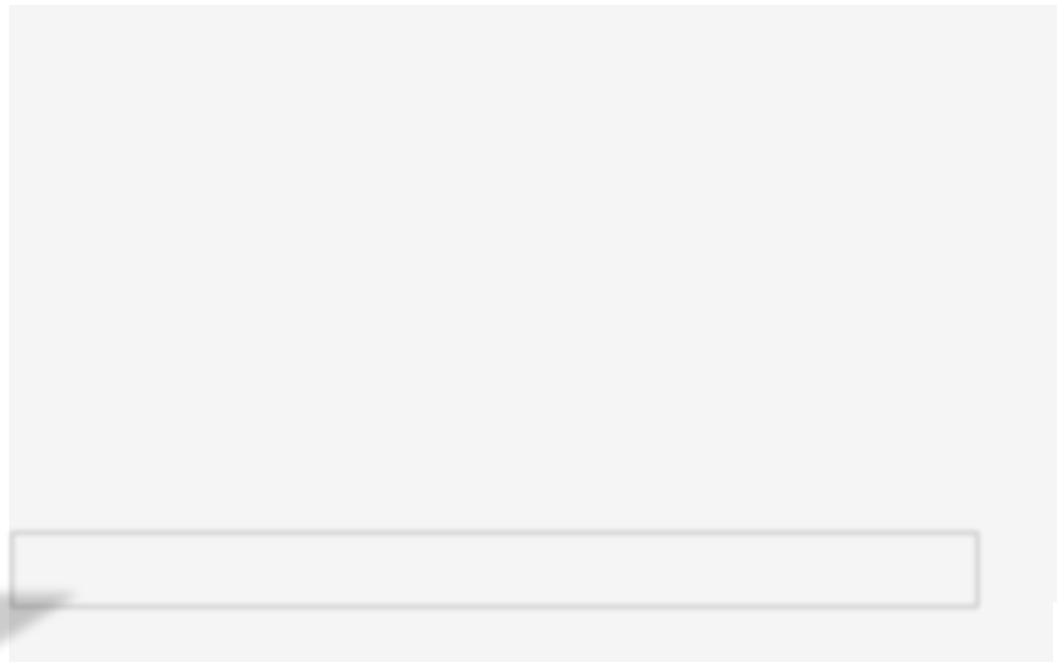
10
20
30



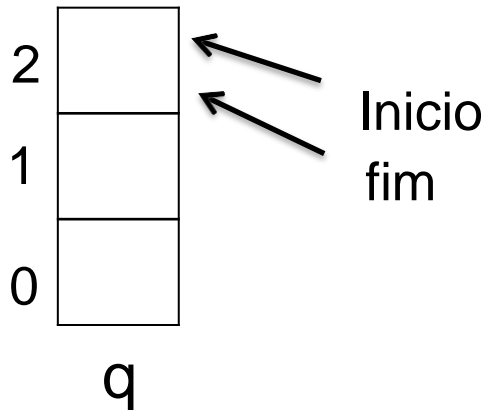
Considerando uma fila de no máximo 3 elementos:



```
10
20
30
50
```



Considerando uma fila de no máximo 3 elementos:




10
20
30
50
60

```
Fila *q = criaFila ();  
enqueue (q, 10);  
enqueue (q, 20);  
enqueue (q, 30);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
enqueue (q, 50);  
enqueue (q, 60);  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));  
printf ("%d\n", dequeue(q));
```

TAD: Fila

- Testando nossa codificação:

```
int main () {  
    Fila *q = criaFila ();  
    enfileira (q, 10);  
    enfileira (q, 20);  
    enfileira (q, 30);  
    printf ("%d\n", desinfileira(q));  
    printf ("%d\n", desinfileira(q));  
    enfileira (q, 40);  
    printf ("%d\n", desinfileira(q));  
    printf ("%d\n", desinfileira(q));  
    return 0;  
}
```



O que será
impresso ?

Exemplos de Aplicação

Fila: Aplicações

- Em controle de processos no sistema operacional, fila de processos.
- Em algoritmo de busca em grafo (busca em largura).
- Algoritmo de caminho mínimo (Dijkstra).
- Algoritmo de ordenação.

Referências Bibliográficas

- AARON, Tanenbaun. **Estruturas de Dados usando C.** São Paulo: Makron Books, 1995.
- PEREIRA, Silvio. **Estruturas de Dados Fundamentais.** São Paulo: Editora Erica, 1996.
- VELOSO, Paulo et al. **Estruturas de Dados.** Rio de Janeiro: Editora Campus, 1983.