Estrutura Abstrata de Dados

FILA

Implementação com ligações simples

(usando ponteiros e memória dinâmica)

Conceitos gerais

Elementos

- Os elementos de uma fila são processados por ordem de chegada
  - o **primeiro** elemento a **entrar** é o **primeiro** a **sair (FIFO** - "First In First Out")

Operações

- Algumas operações realizam-se na **frente** e outras na **cauda** da Fila
  - a **inserção** de um novo elemento
    - consiste em acrescentar este elemento na cauda da fila (torna-se a cauda)
  - a **remoção** de um elemento
    - retira o elemento que se encontra na frente da fila
    - torna o elemento que está a seguir, caso exista, na frente da fila
    - só é possível de realizar se a Fila não estiver vazia
  - a **fila vazia**
    - acontece depois de ser retirado o último elemento da fila
    - só é possível realizar a operação de inserção na fila

Operações básicas (únicas) sobre uma EAD FILA

Criar uma Fila (vazia)

Q = **criarFila**()

Verificar se uma Fila está vazia

filaVazia(Q)

Inserir um novo elemento numa Fila (na cauda)

Q = **juntar**(X, Q)

Remover um elemento de uma Fila (o que está na frente)

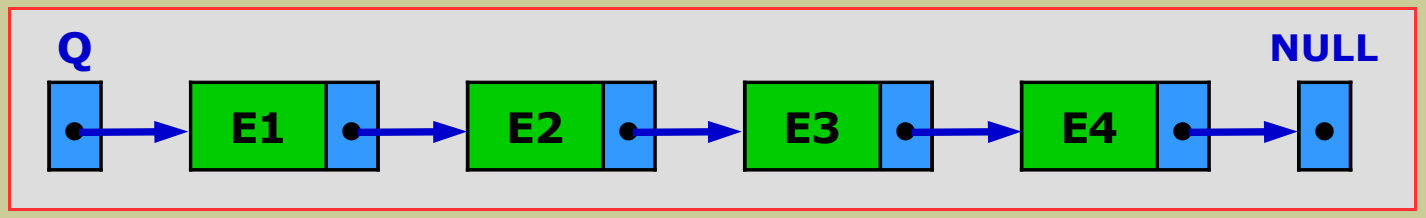
Q = **remover**(Q)

Consultar uma Fila (devolver o elemento da frente)

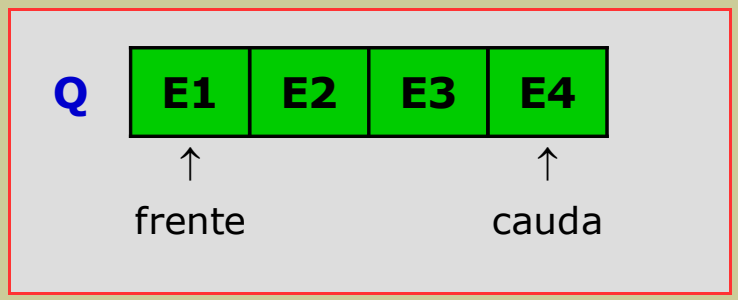
X = **frente**(Q)

Representação gráfica

EAD Fila

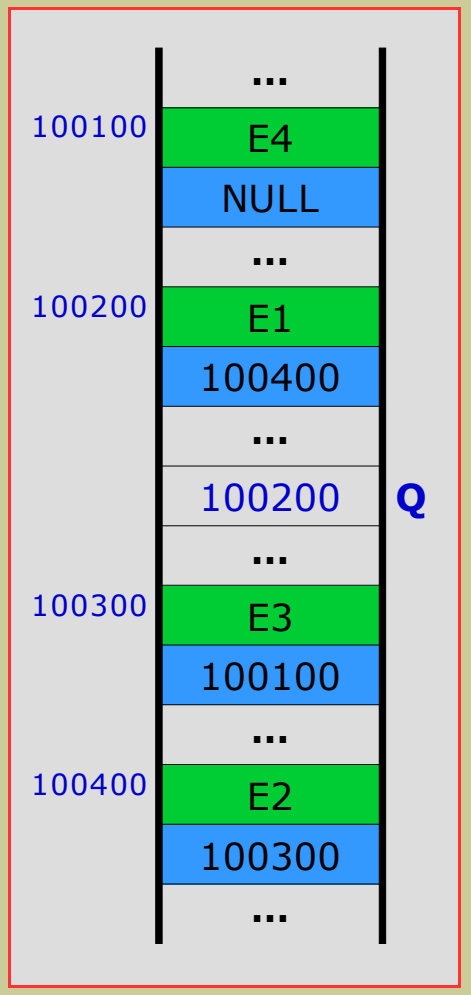


Analogia



Identificador Q

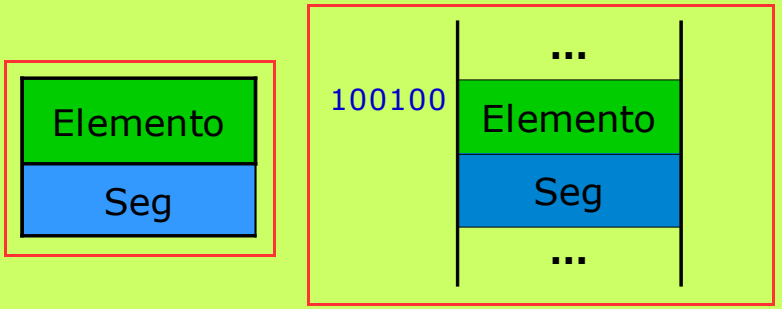
- É um **ponteiro** para a **frente** da **fila**



Nodos e ligações

Definição de um nodo de uma AED FILA (em linguagem C)

```
struct NodoFila{
    INFOFila  Elemento;
    struct NodoFila *Seg;
};
typedef struct NodoFila *PNodoFila;
```



- Cada nodo aponta para o nodo seguinte da fila
- O nodo da cauda da fila aponta para NULL
- A memória para os elementos (nodos) é
  - atribuída quando um elemento é inserido na fila
  - libertada quando um elemento é removido da fila

Implementação

Utilização de um ponteiro associado à frente da Fila

- Ponteiro que aponta para o elemento mais antigo da fila
- Será o primeiro a ser removido
- Permite o acesso à cauda da fila
    - é a seguir à cauda que um novo elemento será inserido
- Quando nulo (NULL) significa fila vazia

Fila vazia

- Quando a frente da fila é um ponteiro nulo (NULL)

Fila cheia

- Quando não há memória para alocar um novo elemento

Operações sobre a estrutura INFOFila

Mostrar os dados de um elemento do tipo INFOFila

void **mostrarElementoFila** (**INFOFila** X)

- operação que mostra/guarda os dados/informação do elemento X do tipo INFOFila

Criar um elemento do tipo INFOFila

INFOFila criarElementoFila ()

- operação que cria um elemento do tipo INFOFila, com dados vindos de uma fonte adequada

Comparar dois elementos do tipo INFOFila

int **compararElementosF** (**INFOFila** X, **INFOFila** Y)

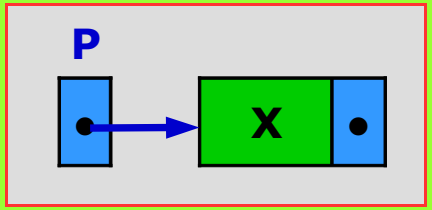
- operação que compara dois elementos do tipo INFOFila (segundo o campo que é a chave)
- devolve: **-1** (X < Y), **0** (X = Y) ou **1** (X > Y)

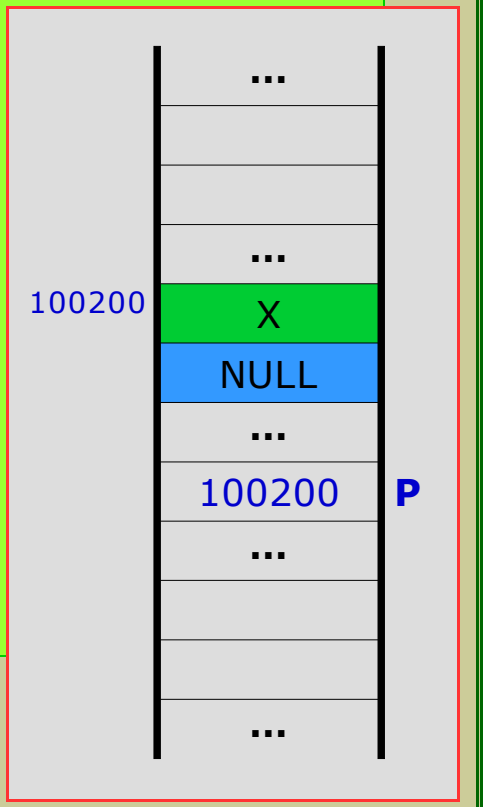
Operações básicas sobre um nodo

Criar um nodo de uma Fila

Entrada: a informação propriamente dita (elemento X), que fará parte de um nodo
Saída: um ponteiro para um **nodo** com informação (elemento X) e ligação a NULL

```
PNodoFila criarNodoFila (INFOFila X)
{
  PNodoFila P;
  P = (PNodoFila) malloc (sizeof(struct NodoFila));
  if (P == NULL)
     return  NULL;
  P→Elemento = X;
  P→Seg = NULL;
  return P;
}
```

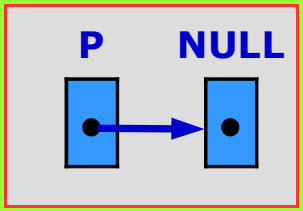


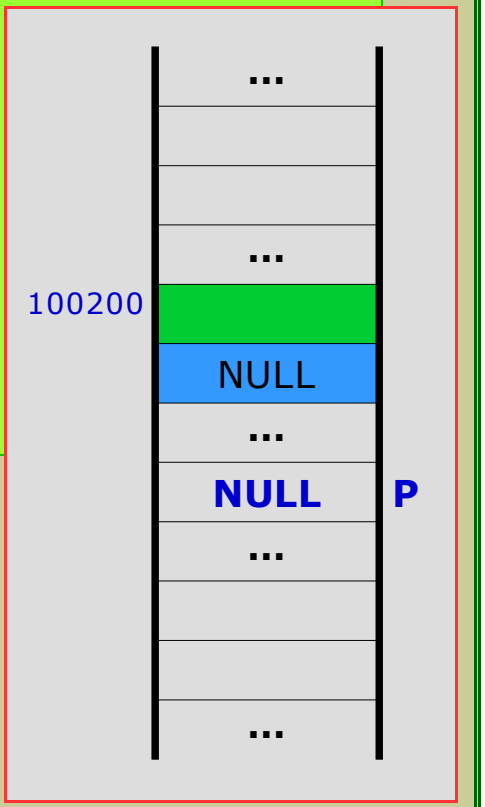


Libertar/destruir um nodo de uma Fila

Operação: libertar toidos os nodos de uma Fila
Entrada: um ponteiro para o nodo que se pretende destruir
Saída: o ponteiro a apontar para NULL

```
PNodoFila libertarNodoFila (PNodoFila P)
{
  P→Seg = NULL;
  free(P);
  P = NULL;
  return P;
}
```





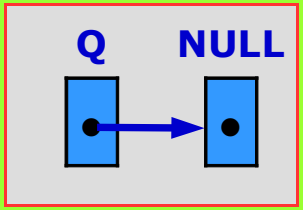
Operações básicas (únicas) sobre uma Fila

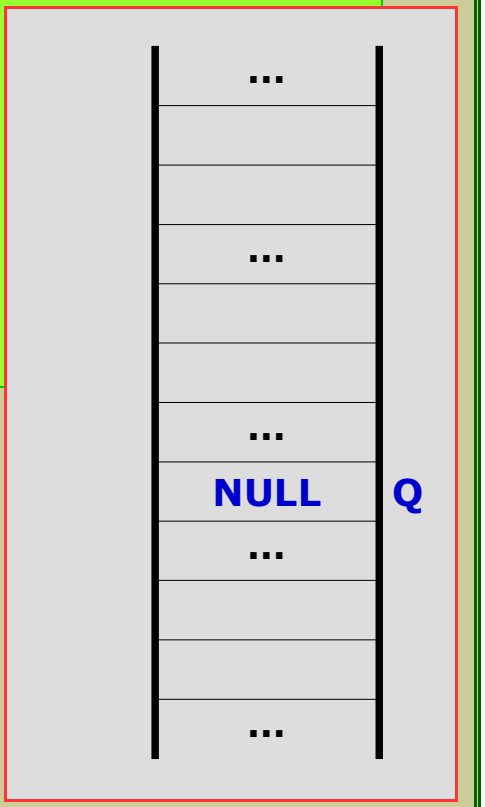
Criar uma Fila

Entrada: nada

Saída: devolve uma Fila Q vazia (sem elementos)

```
PNodoFila criarFila ()
{
  PNodoFila  Q;
  Q = NULL;
  return  Q;
}
```





Verificar se uma Fila está vazia

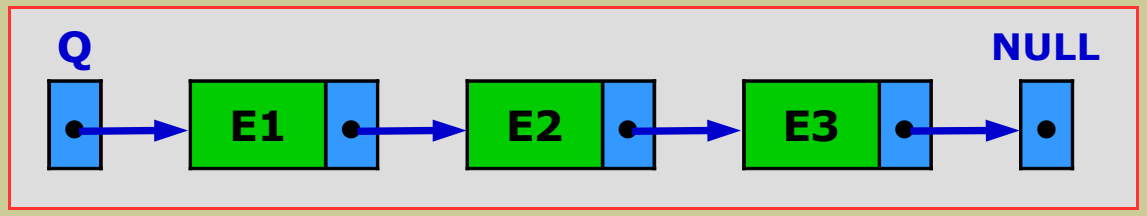
Entrada: uma Fila Q

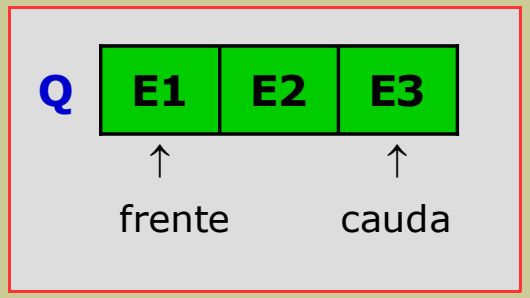
Saída: 1 (se a Fila Q está vazia) ou 0 (se a Fila Q não está vazia)

```
int filaVazia (PNodoFila Q)
{
  if (Q == NULL)
      return 1;
  else
      return 0;
}
```

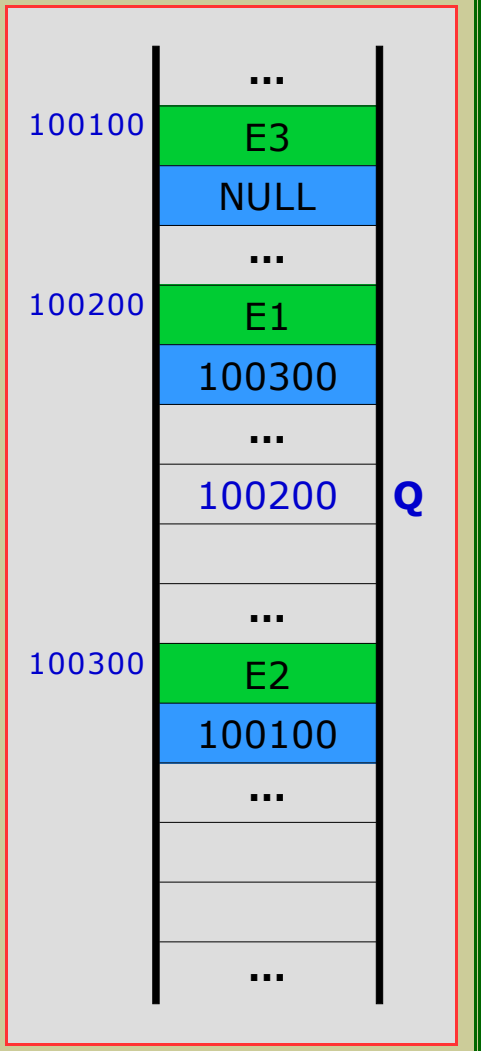
Inserir elemento numa Fila

- Operação **juntar**: Q = juntar(X, Q)
  - Fila Q antes da operação



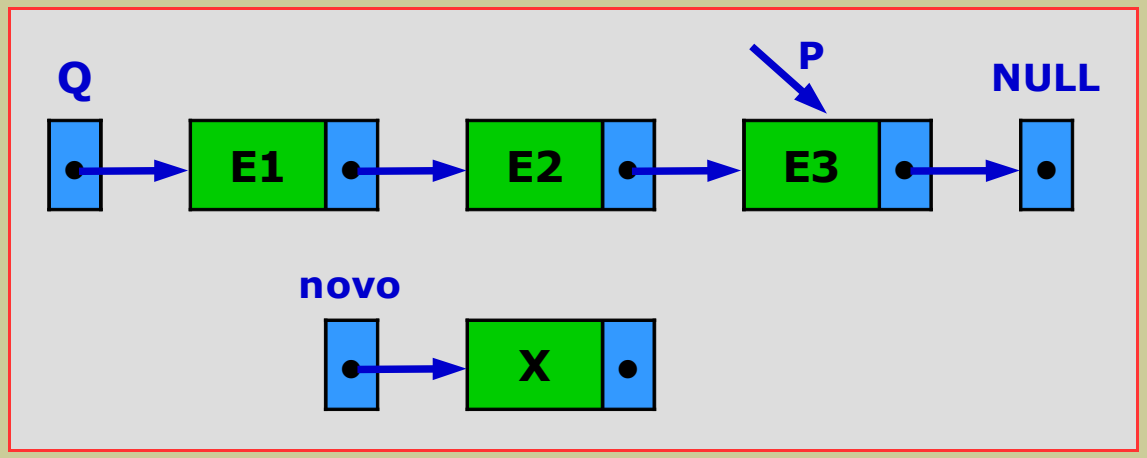


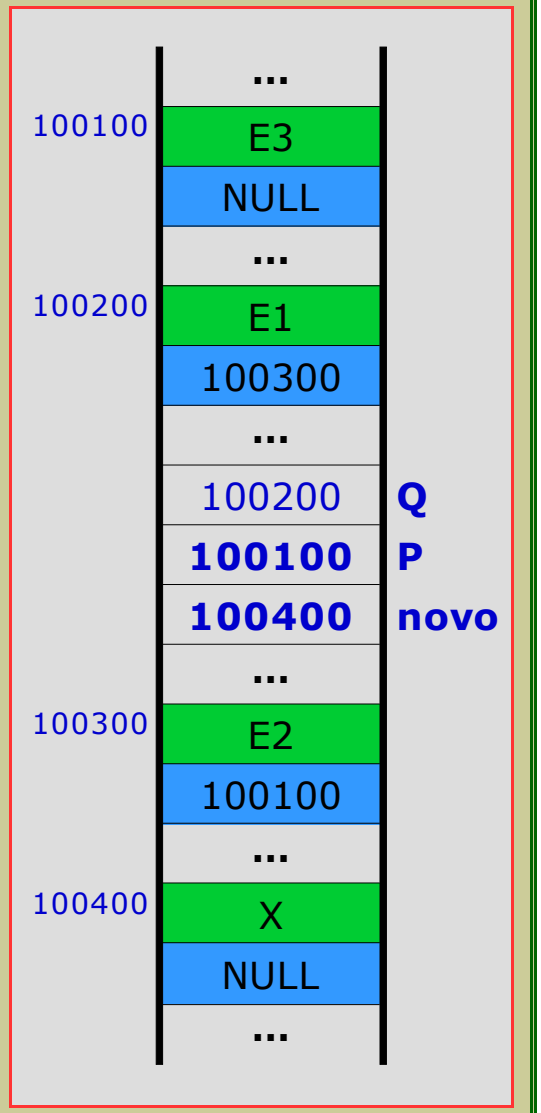
Q = juntar(X, Q)

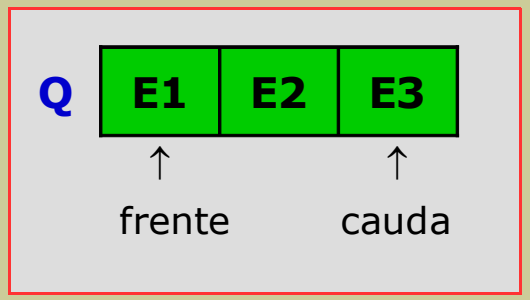


Inserir elemento numa Fila

- Operação **juntar**: Q = juntar(X, Q)
  - Passo 1:
    - procurar a cauda da Fila (ponteiro **P**)
    - criar um nodo com o novo elemento X (apontado por **novo**)

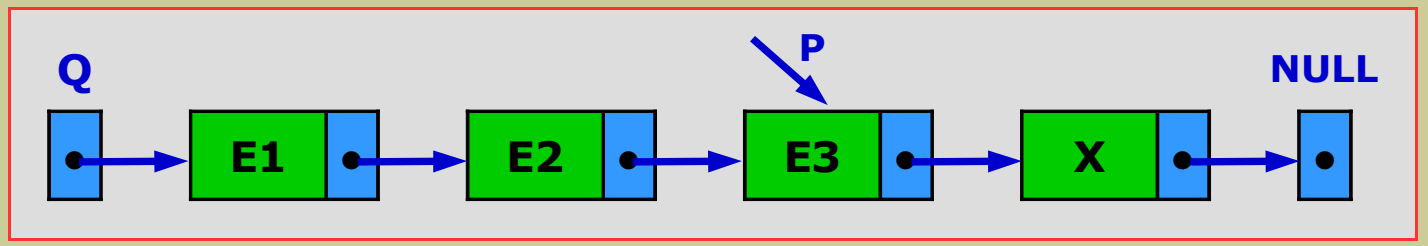


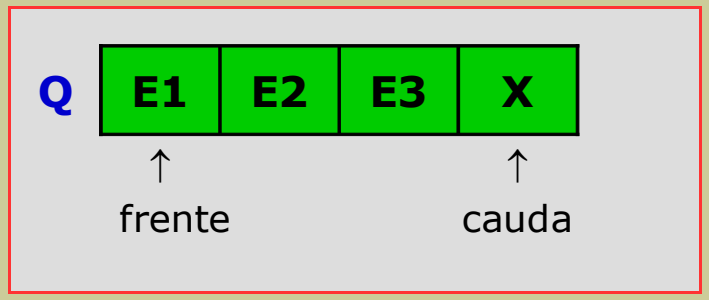


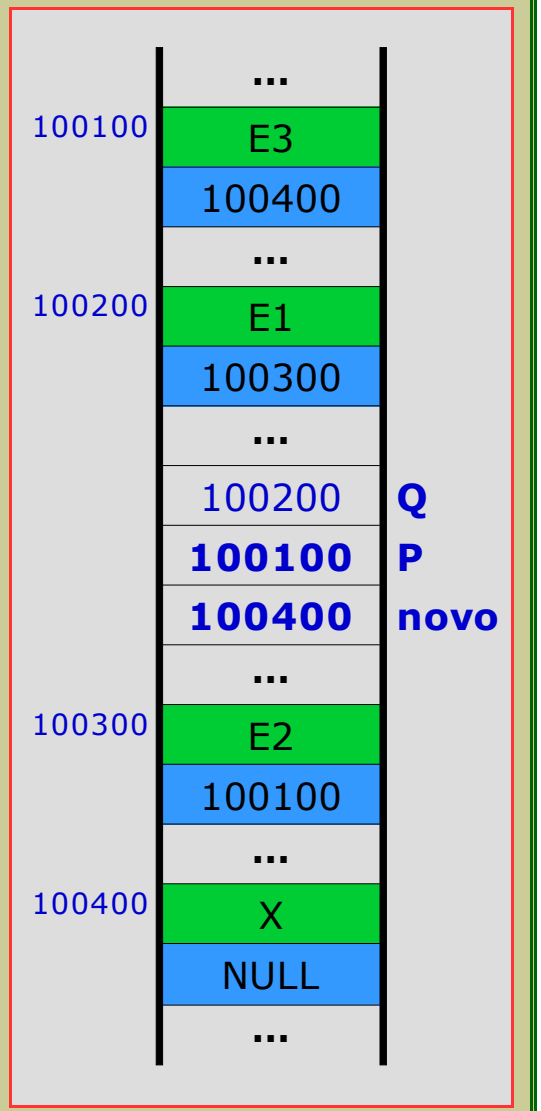

Q
E1
E2
E3
frente
cauda

Inserir elemento numa Fila

- Operação **juntar**: Q = juntar(X, Q)
  - Passo 2:
    - inserir na cauda o nodo com o novo elemento (Novo)







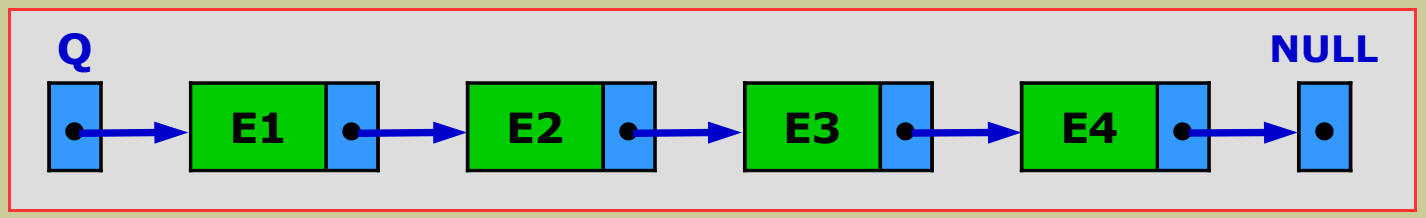
Inserir elemento numa Fila

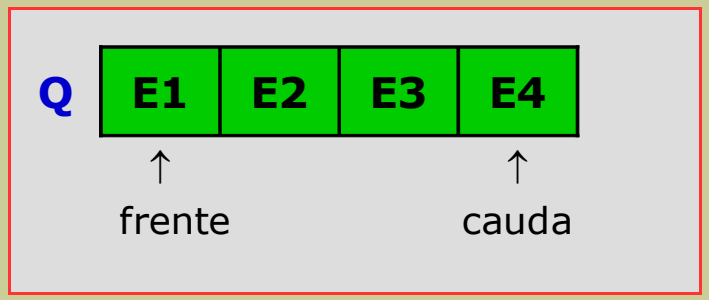
Operação: inserir um elemento na cauda de uma Fila (juntar)
Entrada: um elemento X a inserir e uma Fila Q
Saída: a Fila Q atualizada (com mais um elemento)

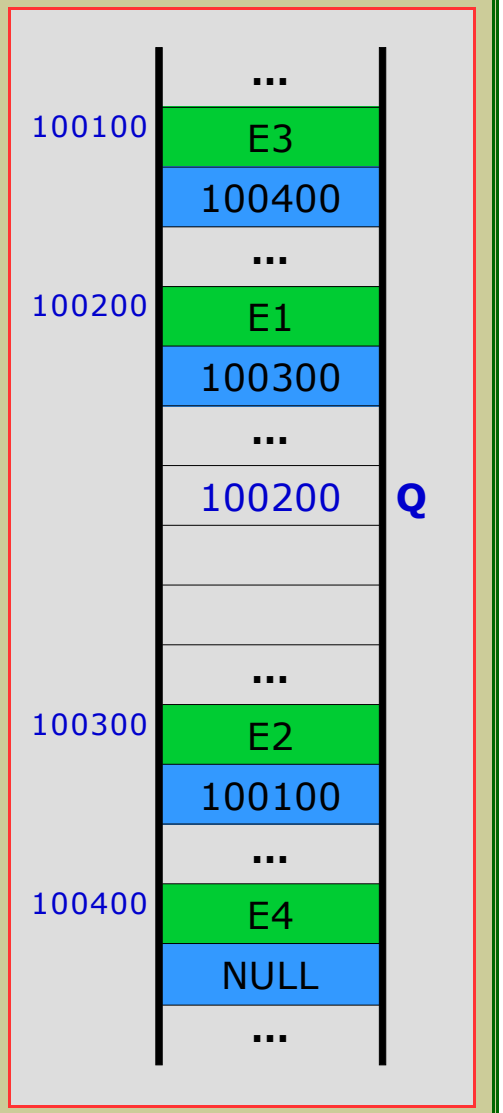
```
PNodoFila juntar (INFOFila X, PNodoFila Q) {
  PNodoFila  P, novo;
  novo = criarNodoFila(X);
  if (novo == NULL)
     return Q;
  if (filaVazia(Q))
     return novo;
  P = Q;
  while (P→Seg != NULL)      // Passo 1
     P = P→Seg;
  P→Seg = novo;         // Passo 2
  return Q;
}
```

Remover elemento de uma Fila

- Operação **remover**: Q = remover(Q)
  - Fila Q antes da operação

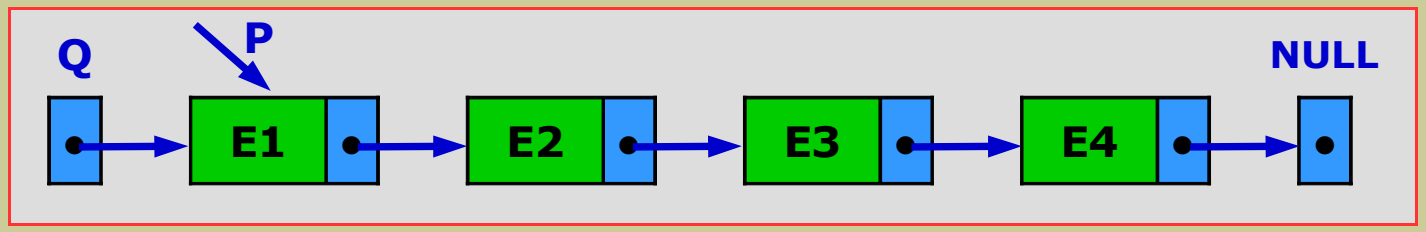


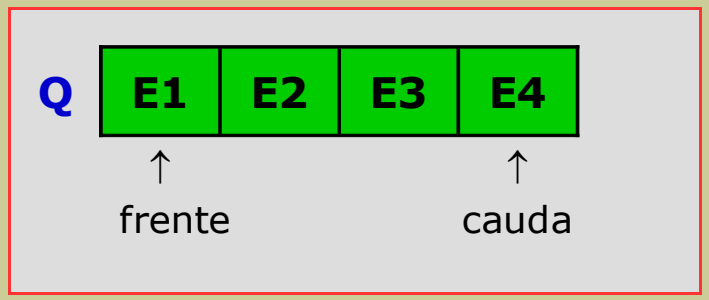


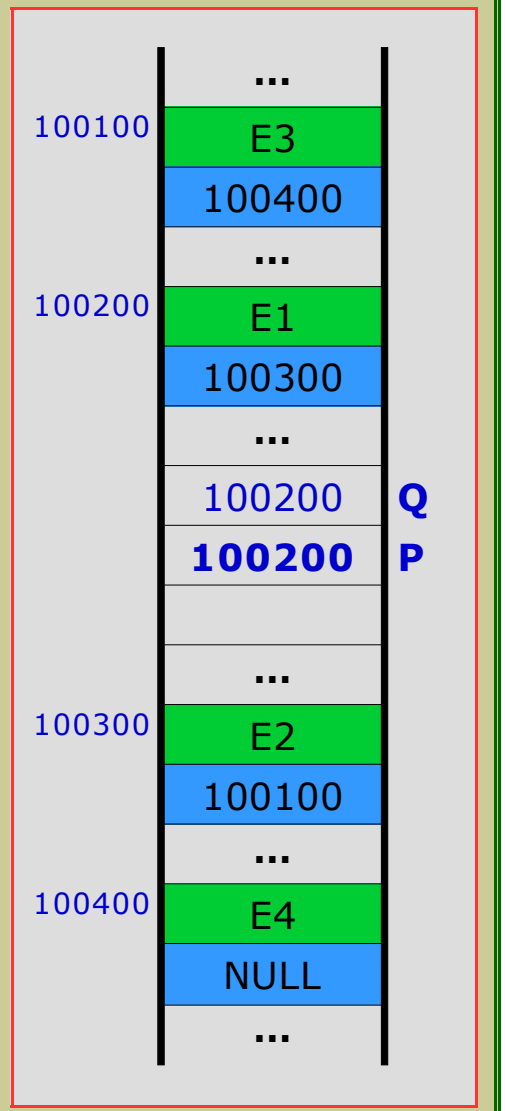


Remover elemento de uma Fila

- Operação **remover**: Q = remover(Q)
  - Passo 1:
    - marcar a frente da Fila Q com um ponteiro P

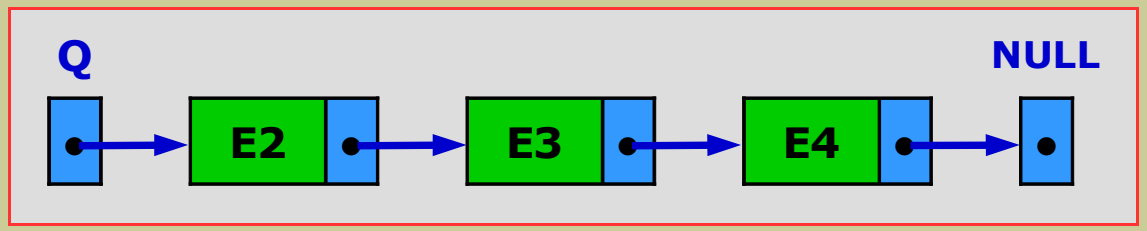


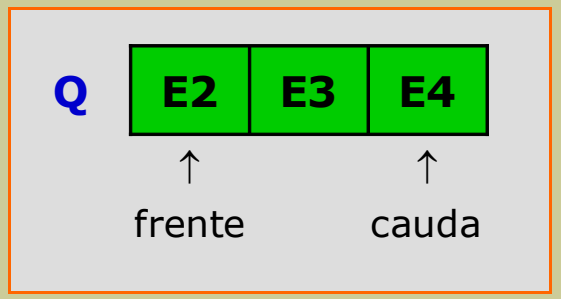


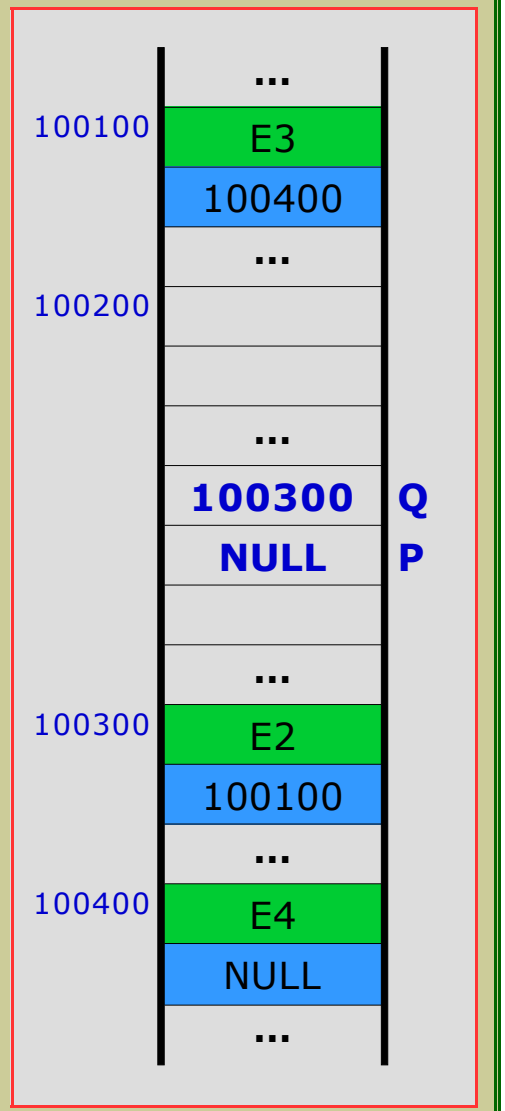


Remover elemento de uma Fila

- Operação **remover**: Q = remover(Q)
  - Passo 2:
    - remover o nodo da frente da Fila Q
      - a frente da Fila passa a ser o nodo seguinte (com 4)
    - libertar o nodo com a frente de Q anterior (P)







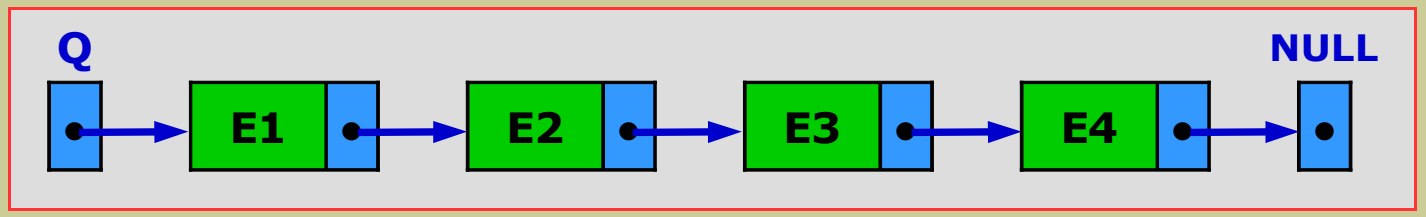
Remover elemento de uma Fila

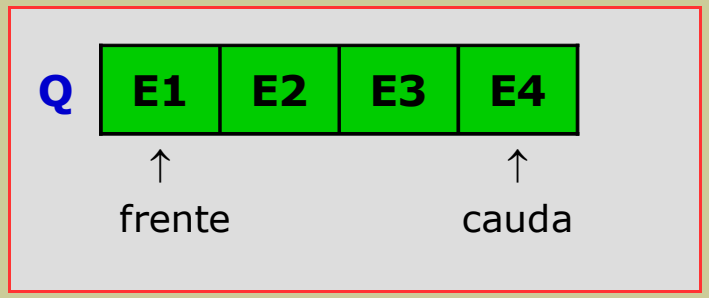
Operação: remove o elemento da frente de uma Fila
Entrada: uma Fila Q
Saída: a Fila Q atualizada (sem o elemento da frente)

```
PNodoFila remover (PNodoFila Q)
{
  PNodoFila P;
  P = Q;
  Q = Q→Seg;
  P = libertarNodoFila(P);
  return Q;
}
```

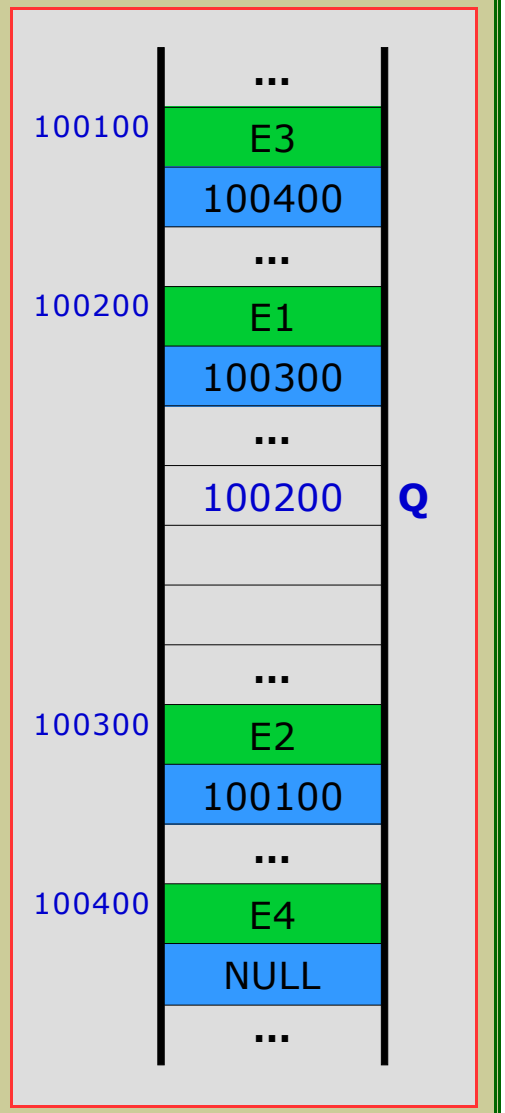
Consultar uma Fila

- Representação gráfica





frente(Q) = **E1** (Q→Elemento)



Consultar uma Fila

Operação: consulta o elemento da frente de uma Fila (frente)
Entrada: uma Fila Q
Saída: o elemento que está na frente da Fila Q (do tipo INFOFila)

```
INFOFila frente (PNodoFila Q)
{
  return Q→Elemento;
}
```