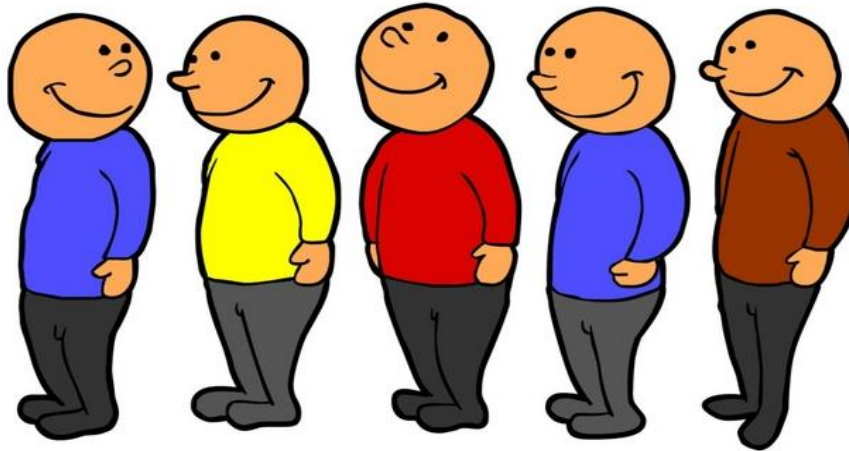
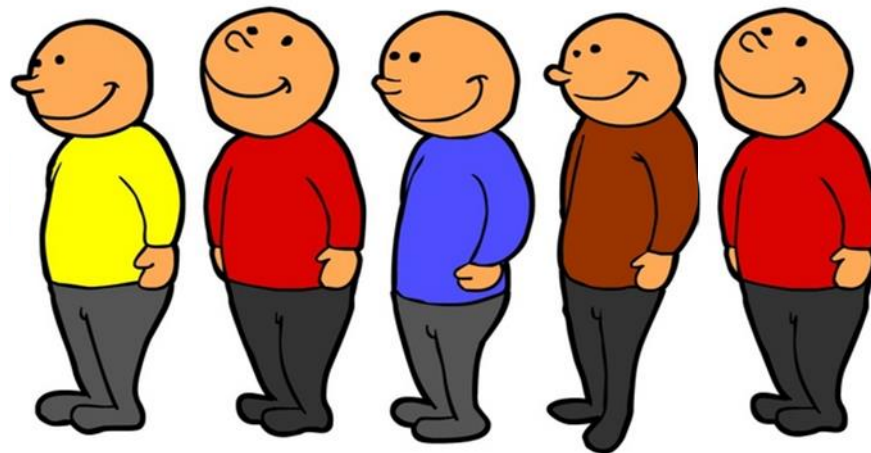


Tema 3 . Estructuras lineales estáticas y dinámicas

3.2 Filas o Colas



Estructura de datos



□ Filas.

- Representación en memoria estática y dinámica.
- Operaciones básicas con filas.
- Tipos de filas: fila simple, fila circular y filas dobles.
- Aplicaciones: Colas de prioridad

Introducción

- La fila es una estructura de datos donde la inserción de elementos se hace en un extremo (el fin de la cola) y la recuperación/borrado de elementos se hace en el otro extremo (el inicio de la fila).
- fila de las tortillas
- fila en la ventanilla del banco
- fila de impresión en un sistema multiprocesador.

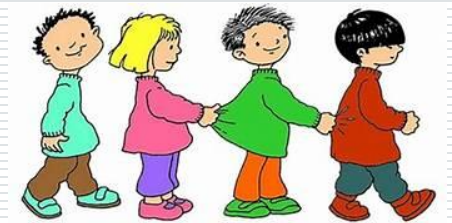


Aplicaciones del tda colas

- ☐ Sistemas donde el orden de atención coincide con el orden de llegada
 - ☐ Cola de impresión(recurso compartido de uso exclusivo).
 - ☐ Cola de procesos (en el planificador de tareas).
 - ☐ Atención de solicitudes (web o bases de datos).
 - ☐ Productores y consumidores de datos(buffer de teclado).
-

Aplicaciones de colas

- ❑ procesos de las fábricas,
- ❑ cola del banco,
- ❑ transporte público,
- ❑ cola del supermercado
- ❑ diseños de Call center.
- ❑ Servidores de internet
- ❑ Bandeja de entrada de correo.



Como el primer elemento insertado es el primero en ser recuperado, se hace referencia a las filas como estructuras FIFO (*first-in, first-out*).

- Normalmente se trabaja con dos tipos de filas: lineal y circular.

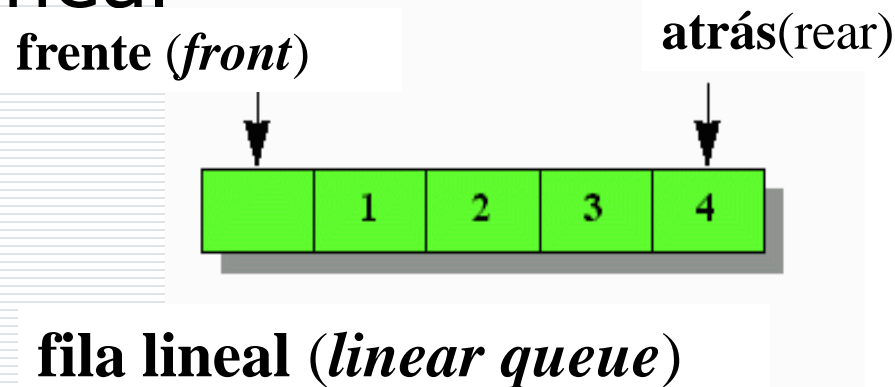


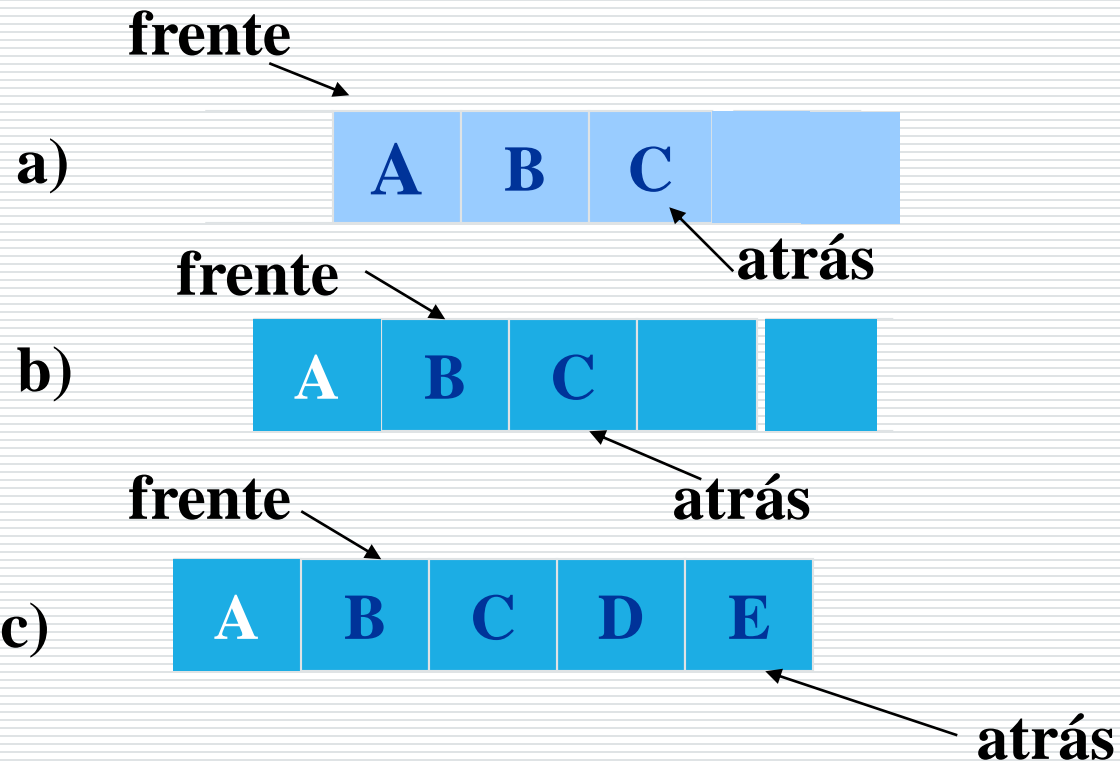
Tipos de colas

- **Cola simple:** Estructura lineal donde los elementos salen en el mismo orden en que llegan.
- **Cola circular:** Representación lógica de una cola simple en un arreglo.
- **Cola de Prioridades:** Estructura lineal en la cual los elementos se insertan en cualquier posición de la cola y se remueven solamente por el frente.
- **Cola Doble (Bicola):** Estructura lineal en la que los elementos se pueden añadir o quitar por cualquier extremo de la cola (cola bidireccional).

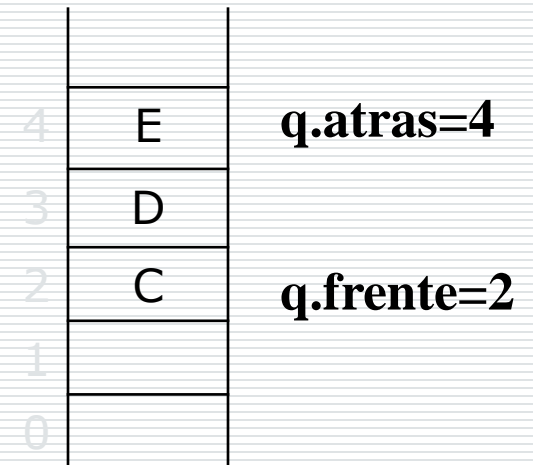
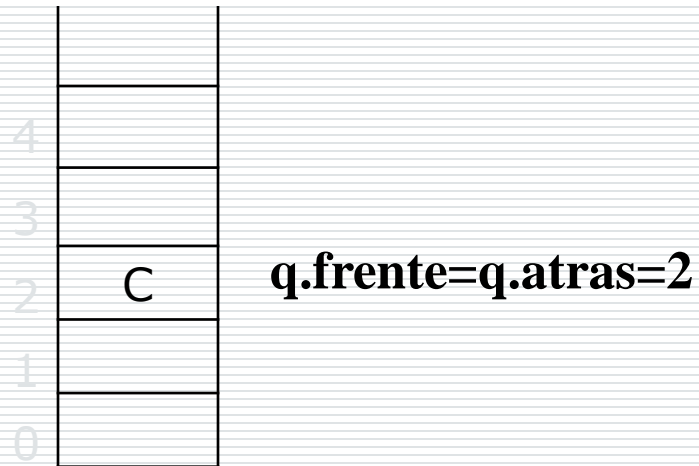
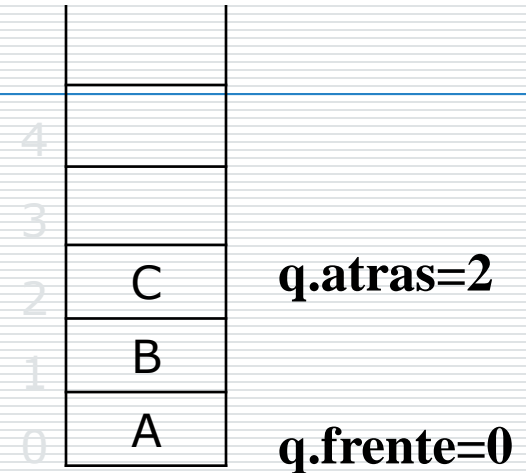
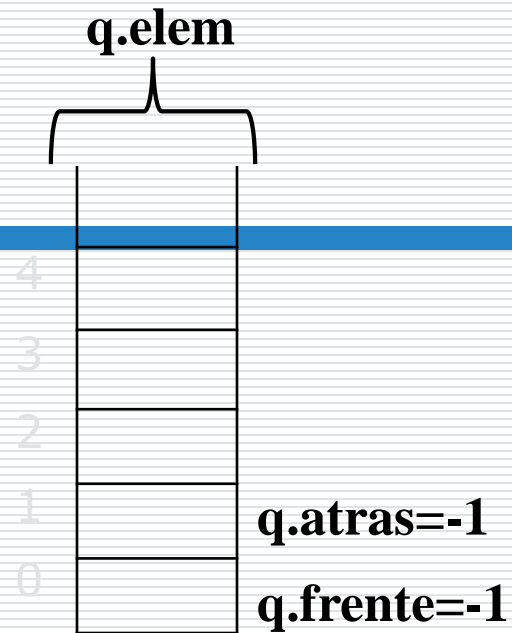
introducción

- En ambas filas, la inserción de datos se realiza en el fin de la fila se mueven hacia adelante y se recuperan(borran) del inicio de la fila. La siguiente figura ilustra una fila lineal:





tda fila



Métodos

insertar(x) --> Agrega un elemento al final de la fila

eliminar() --> Elimina el primer elemento de la fila.

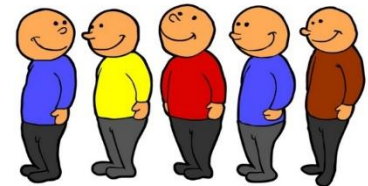
info() --> Retorna el primer elemento en la fila.

filavacia() --> Retorna true si no existen elementos,
false en caso contrario.

filallena() --> Retorna true cuando la fila esta llena
false en caso contrario.

vaciarfila() --> Elimina todos los elementos de la fila.

busca(x) --> Regresa true si encuentra al
elemento x en la fila.



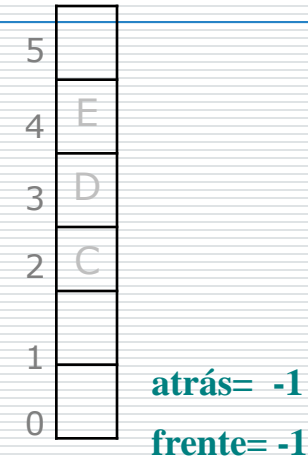
imprimefila() --> Muestra el contenido de la fila.

contar() --> Retorna la cantidad de elementos
en la fila.

Algoritmo filavacia()

{Retorna true si no existen elementos, false en caso contrario. }

1. Inicio
2. Si $frente == -1$ y $atras == -1$
 - 2.1 Regresa true;
 - 2.2 Si no
 - 2.3 Regresa false
3. Fin del condicional 2
4. Fin



Algoritmo filallena()

{Retorna true cuando la fila está llena (si no existen espacios libres), false en caso contrario. }

1. Inicio

2. Si ($\text{atras} == \text{max} - 1$)

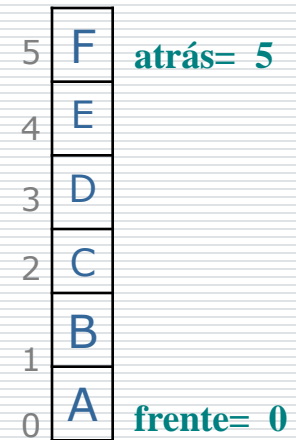
2.1 regresa true

2.2 si no

2.3 regresa false

3 fin cond 2

4. fin



Algoritmo inserta(dato)

{Agrega un elemento al final de la fila. }

1. Inicio

2. Si No(filallena()) //verifica si hay espacio, $\text{final} < \text{max}-1$

entonces $\text{atras} = \text{atras} + 1,$
 $\text{elem}[\text{atras}] = \text{dato}$

1.1 Si $\text{atras} == 0$ entonces //se inserto el primero

$\text{frente} = 0$

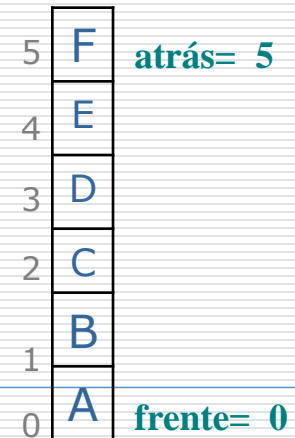
1.2 fin_condicional_1.1

si no

Escribir "error Desbordamiento"

3. Fin el condicional 1.

4. Fin.



Algoritmo elimina()

{Borra el primer elemento de la fila. }

1.Inicio

2. dato=-1

3. Si No(filavacia())//verifica que no este vacía frente<> -1

3.1 entonces dato= elem[frente]

3.1.1 Si frente==atrás //hay un solo elemento

dato= C

entonces

frente=-1, atrás=-1 //hay mas de un elemento

si no

frente= frente +1

3.1.2 fin del condicional 1.1

3.2 si no Escribir“Subdesbordamiento”

4. Fin el condicional 3.

5. retorna dato

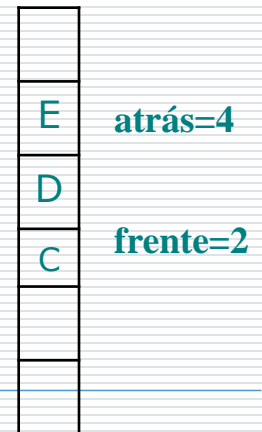
6. Fin



Algoritmo info()

{retorna el valor del primer elemento, sin eliminarlo de la fila. }

1. Inicio
2. dato= -1
3. Si (filavacia()) //verifica que no este vacía entonces
 - 3.1 Escribir "No hay información que mostrar"
 - si no
 - 3.2 dato= elem[frente]
4. regresa dato
5. Fin el condicional del paso_3
6. Fin



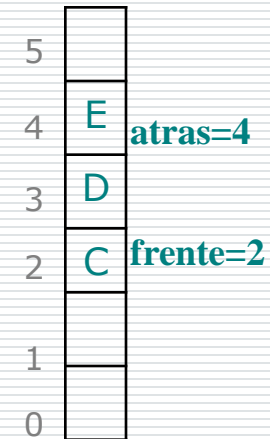
Algoritmo vaciarfila()

{elimina todos los elementos de la fila. }

1. Inicio

2. Si (filavacia())
entonces

fin



Algoritmo vaciarfila()

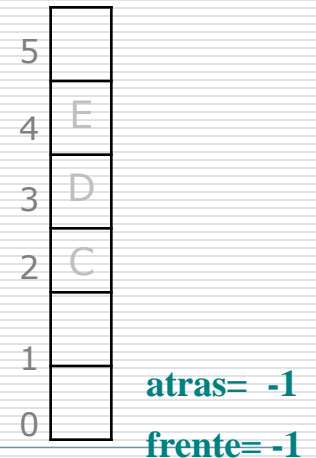
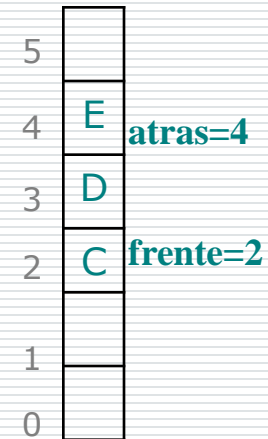
1. Inicio

2. Si no (filavacia())

entonces frente= -1;

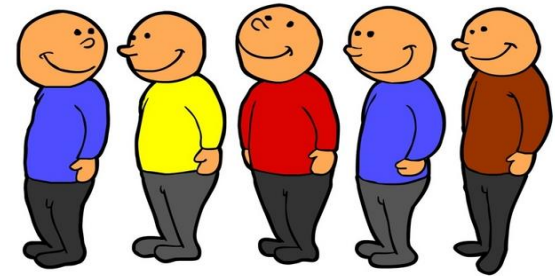
atras= -1;

fin

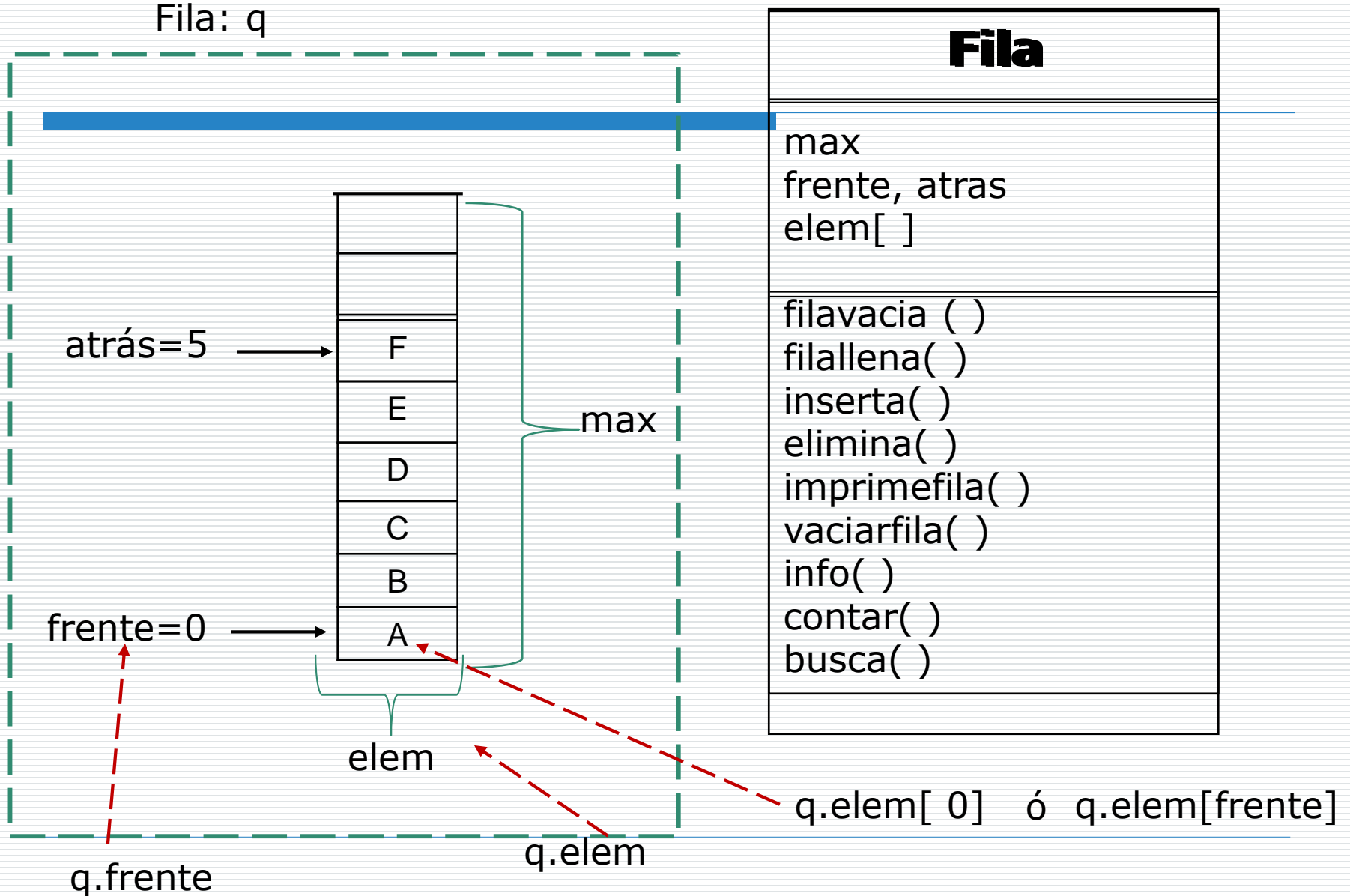


IMPLEMENTACION EN JAVA

- Elabora el diagrama UML del TDA cola lineal estática.
- Escribe en Java la definición de la clase cola estática, y una aplicación para probar el comportamiento de un objeto de esta clase.



□ Diagrama UML del TDA Fila estática



// Implementación del TDA Fila Estática en Java

```
import java.io.*;  
import java.lang.*;
```

```
class Fila
```

```
{ int frente, atras, max;  
  private int elem[];
```

```
  public Fila(int n ) {  
    max= n;  
    elem=new int[max];  
    frente=-1;  
    atras=-1;  
  }
```

```
public vaciarFila() {  
    if !(filavacia())  
    { frente=-1;  
      atrás= -1;  
    }  
}
```

```
}
```

Fuentes de información

- Estructuras de Datos, Cairo – Guardati, Mc Graw Hill
- Estructuras de Datos, Román Martínez, Elda Quiroga, Thomson Learning

