



# **TAREA DE DEFENSA**

## **HITO 4**

**RODRIGO CRISTHIAN TORREZ DE LA CRUZ**





## MANEJO DE CONCEPTOS

### 1. ¿A que se refiere cuando se habla de **ESTRUCTURA DE DATOS**?

**RESPUESTA:** Estructura de datos en el contexto de la programación y la informática, se refiere a la forma en que se organizan y almacenan los datos en una computadora para que puedan ser utilizados y manipulados de manera eficiente.

### 2. ¿Que significa FIFO?

**RESPUESTA:** FIFO es el acrónimo de "First-In, First-Out", que se traduce al español como "primero en entrar, primero en salir". Es un principio o regla que se aplica en varias áreas, incluyendo la gestión de colas, el almacenamiento de datos y la planificación de procesos.

### 3. ¿Qué diferencia hay entre LIFO y FIFO?

**RESPUESTA:** La diferencia entre LIFO y FIFO se encuentra en el orden de eliminación o procesamiento de los elementos. LIFO sigue el orden de "último en entrar, primero en salir", mientras que FIFO sigue el orden de "primero en entrar, primero en salir". Estos principios son aplicados en diversas áreas, como la gestión de colas, el almacenamiento de datos y la planificación de procesos.



## MANEJO DE CONCEPTOS

### 4. ¿Qué es una cola?

**RESPUESTA:** Una cola es una estructura de datos que sigue el principio FIFO, donde los elementos se agregan al final y se eliminan desde el principio. Es útil para gestionar datos en el orden en que fueron agregados, como una fila de espera.

### 5. ¿Qué es QUEUE en JAVA, una QUEUE será lo mismo que una COLA?

**RESPUESTA:** La interfaz Queue en Java, ubicada en el paquete java.util, define un conjunto de métodos que permiten manipular y operar una estructura de datos basada en el principio FIFO (First-In, First-Out) de una cola. Esto significa que los elementos se agregan al final de la cola y se eliminan desde el principio.

### 6. ¿Qué es INI o REAR en una COLA?

**RESPUESTA:** “

- "Ini" (también conocido como "front" o "head") se refiere al inicio de la cola, es decir, al primer elemento que se agregó y será el próximo en ser eliminado.
- "Rear" se refiere al final de la cola, es decir, al último elemento que se agregó.

Cuando se agrega un elemento a la cola, se actualiza la posición del "rear", y cuando se elimina un elemento de la cola, se actualiza la posición del "ini" (o "front" o "head"). Estos términos se utilizan para describir las operaciones y la lógica de una cola y pueden variar ligeramente según la implementación específica utilizada.



## MANEJO DE CONCEPTOS

### 7. ¿Qué es FIN o FRONT en una COLA?

**RESPUESTA:** "Front" (frente): Se refiere al inicio de la cola, es decir, a la posición o referencia al primer elemento que se agregó y será el próximo en ser eliminado cuando se realice una operación de desencolado (dequeue()). También se puede utilizar para referirse a la operación de acceder al elemento en el inicio de la cola (peek()), sin eliminarlo.

"Rear" (trasero): Se refiere al final de la cola, es decir, a la posición o referencia al último elemento que se agregó. Cuando se realiza una operación de encolado (enqueue()), el nuevo elemento se agrega al final de la cola, y la posición del "rear" se actualiza para apuntar al nuevo elemento agregado.

### 8. ¿A que se refiere los métodos esVacia() y esLLena() en una COLA?

**RESPUESTA:** esVacia() es un método que verifica si la cola no contiene elementos, mientras que esLLena() se utiliza para determinar si la cola ha alcanzado su capacidad máxima (aunque en la mayoría de los casos no es relevante para colas dinámicas). Estos métodos permiten realizar comprobaciones sobre el estado de la cola antes de realizar ciertas operaciones para garantizar que la cola esté en el estado esperado.

```
public boolean esVacio(){  
    if(ini==0&&fin==0){  
        return true;}else{  
        return false;  
    }  
}
```

```
public boolean esLLeno(){  
    if(fin==max){  
        return true;}  
    else{  
        return false;  
    }  
}
```





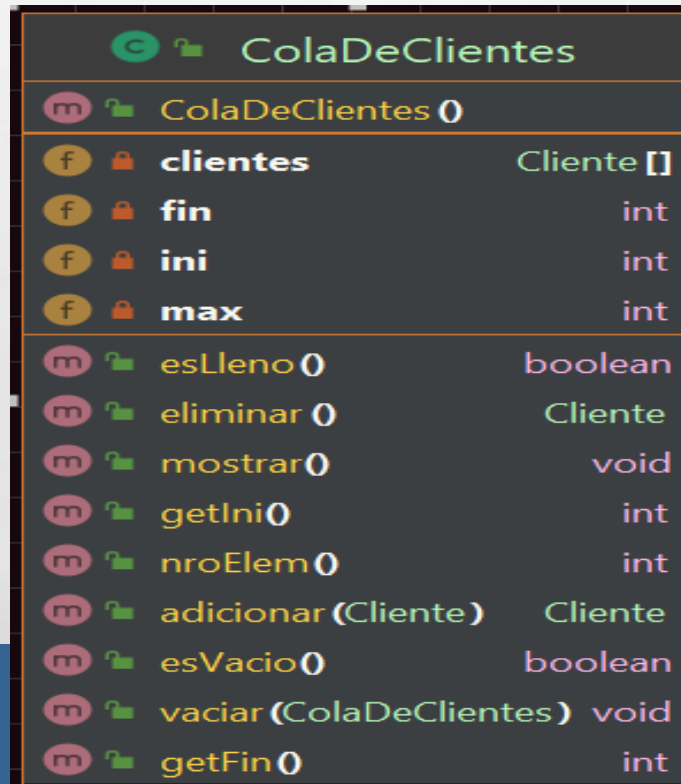
## MANEJO DE CONCEPTOS

9. ¿Qué son los métodos estáticos en JAVA?

**RESPUESTA:** Los métodos estáticos en Java pertenecen a una clase en lugar de a una instancia específica y se pueden acceder directamente a través del nombre de la clase. Son útiles para proporcionar funcionalidades compartidas y no pueden acceder a variables de instancia.

10. ¿A través de un gráfico, muestre los métodos mínimos que debería de tener una COLA?

**RESPUESTA:** El gráfico representa visualmente la estructura mínima de una cola y los métodos asociados que permiten su uso adecuado.



ColaDeClientes		
ColaDeClientes ()		
f	clientes	Cliente []
f	fin	int
f	ini	int
f	max	int
m	esLleno ()	boolean
m	eliminar ()	Cliente
m	mostrar ()	void
m	getIni ()	int
m	nroElem ()	int
m	adicionar (Cliente)	Cliente
m	esVacio ()	boolean
m	vaciar (ColaDeClientes)	void
m	getFin ()	int



PARTE PRACTICA



## PARTE PRACTICA

### 11. Crear las clases necesarias para la PILA DE CLIENTES

```
no usages
public static void main(String[] args) {

    ColaDeClientes colaClientes=new ColaDeClientes();
    Cliente cliente1 = new Cliente( nombres: "Rodrigo", apellidos: "Torrez", edad: 60, pais: "Bolivia", genero: "Masculino", tipo: "Gold");
    Cliente cliente2 = new Cliente( nombres: "Emeth", apellidos: "Brount", edad: 38, pais: "Mexico", genero: "Masculino", tipo: "Silver");
    Cliente cliente3 = new Cliente( nombres: "Louren", apellidos: "McFly", edad: 68, pais: "Bolivia", genero: "Femenino", tipo: "Gold");
    Cliente cliente4 = new Cliente( nombres: "George", apellidos: "McFly", edad: 90, pais: "Bolivia", genero: "Masculino", tipo: "Silver");
    Cliente cliente5 = new Cliente( nombres: "Saul", apellidos: "McFly", edad: 70, pais: "Peru", genero: "Masculino", tipo: "Gold");
    //-----

    colaClientes.adicionar(cliente1);
    colaClientes.adicionar(cliente2);
    colaClientes.adicionar(cliente3);
    colaClientes.adicionar(cliente4);
    colaClientes.adicionar(cliente5);
    colaClientes.mostrar();

}
```

### Cliente

- + nombres: **String**
- + apellidos: **String**
- + edad: **Int**
- + pais: **String**
- + genero: **String**
- + tipo: **String**

Cliente(nombres, apellidos, edad, pais, genero, tipo)  
gets()  
sets()  
muestraCliente()

Mostrando datos del Cliente

Nombre: Rodrigo  
Apellidos: Torrez  
Edad: 60  
Pais: Bolivia  
Genero: Masculino  
Tipo: Gold

Mostrando datos del Cliente

Nombre: Emeth  
Apellidos: Brount  
Edad: 38  
Pais: Mexico  
Genero: Masculino  
Tipo: Silver

Mostrando datos del Cliente

Nombre: Louren  
Apellidos: McFly  
Edad: 68  
Pais: Bolivia  
Genero: Femenino  
Tipo: Gold

Mostrando datos del Cliente

Nombre: George  
Apellidos: McFly  
Edad: 90  
Pais: Bolivia  
Genero: Masculino  
Tipo: Silver

Mostrando datos del Cliente

Nombre: Saul  
Apellidos: McFly  
Edad: 70  
Pais: Peru  
Genero: Masculino  
Tipo: Gold



## PARTE PRACTICA

### 12. Inicializar la cola de clientes.

- Crear una cola con 5 clientes.
  - En la clase MAIN deberán estar los 5 clientes.
  - Mostrar todos los datos de la cola de clientes
- Adjuntar los siguientes
  - El **código** del método que resuelve el problema.
  - Una **imagen** de la salida de la consola.
  - Link que me lleve a la clase main (GitHub)

```
no usages
public static void main(String[] args) {

    ColaDeClientes colaClientes=new ColaDeClientes();
    Cliente cliente1 = new Cliente( nombres: "Rodrigo", apellidos: "Torrez", edad: 60, pais: "Bolivia", genero: "Masculino", tipo: "Gold");
    Cliente cliente2 = new Cliente( nombres: "Emeth", apellidos: "Brount", edad: 38, pais: "Mexico", genero: "Masculino", tipo: "Silver");
    Cliente cliente3 = new Cliente( nombres: "Louren", apellidos: "McFly", edad: 68, pais: "Bolivia", genero: "Femenino", tipo: "Gold");
    Cliente cliente4 = new Cliente( nombres: "George", apellidos: "McFly", edad: 90, pais: "Bolivia", genero: "Masculino", tipo: "Silver");
    Cliente cliente5 = new Cliente( nombres: "Saul", apellidos: "McFly", edad: 70, pais: "Peru", genero: "Masculino", tipo: "Gold");
    //-----

    colaClientes.adicionar(cliente1);
    colaClientes.adicionar(cliente2);
    colaClientes.adicionar(cliente3);
    colaClientes.adicionar(cliente4);
    colaClientes.adicionar(cliente5);
    colaClientes.mostrar();

}
```

```
Mostrando datos del Cliente
Nombre: Rodrigo
Apellidos: Torrez
Edad: 60
Pais: Bolivia
Genero: Masculino
Tipo: Gold
```

```
Mostrando datos del Cliente
Nombre: Emeth
Apellidos: Brount
Edad: 38
Pais: Mexico
Genero: Masculino
Tipo: Silver
```

```
Mostrando datos del Cliente
Nombre: Louren
Apellidos: McFly
Edad: 68
Pais: Bolivia
Genero: Femenino
Tipo: Gold
```

```
Mostrando datos del Cliente
Nombre: George
Apellidos: McFly
Edad: 90
Pais: Bolivia
Genero: Masculino
Tipo: Silver
```

```
Mostrando datos del Cliente
Nombre: Saul
Apellidos: McFly
Edad: 70
Pais: Peru
Genero: Masculino
Tipo: Gold
```





## PARTE PRACTICA

### 13. Promoción para usuarios de Bolivia.

- En el mes de diciembre a todos los clientes de Bolivia se les dará una promoción en cuanto a precios en viajes a nivel nacional.
  - A todos los clientes que sean de nacionalidad boliviana y además el tipo de cliente GOLD, convertir a estos clientes en VIP
  - Es decir si es de Bolivia y es GOLD deberá ser ahora un cliente VIP
- El método estático dentro de la clase MAIN recibe 3 atributos
  - La cola de clientes
  - El tipo de cliente
  - La nacionalidad del cliente.
- Adjuntar los siguientes
  - El **código** del método que resuelve el problema.
  - Una **imagen** de la salida de la consola.
  - Link que me lleve a la clase main (GitHub)

```
public static void Gold_a_Vip(ColaDeClientes cola,String tipo, String nacionalidad){
    ColaDeClientes aux = new ColaDeClientes();
    Cliente cont;
    while(!cola.esVacio()){
        cont=cola.eliminar();
        if(Objects.equals(cont.getPais(),nacionalidad) && Objects.equals(cont.getTipo(),tipo)){
            cont.setTipo("Vip");
        }
        aux.adicionar(cont);
    }
    cola.vaciar(aux);
    cola.mostrar();
}
```

`Gold_a_Vip(colaClientes, tipo: "Gold", nacionalidad: "Bolivia");`

ANTES

```
Mostrando datos del Cliente
Nombre: Rodrigo
Apellidos: Torrez
Edad: 60
Pais: Bolivia
Genero: Masculino
Tipo: Gold
```

```
Mostrando datos del Cliente
Nombre: Emeth
Apellidos: Brount
Edad: 38
Pais: Mexico
Genero: Masculino
Tipo: Silver
```

```
Mostrando datos del Cliente
Nombre: Louren
Apellidos: McFly
Edad: 68
Pais: Bolivia
Genero: Femenino
Tipo: Gold
```

```
Mostrando datos del Cliente
Nombre: George
Apellidos: McFly
Edad: 90
Pais: Bolivia
Genero: Masculino
Tipo: Silver
```

```
Mostrando datos del Cliente
Nombre: Saul
Apellidos: McFly
Edad: 70
Pais: Peru
Genero: Masculino
Tipo: Gold
```

DESPUES

```
Mostrando datos del Cliente
Nombre: Rodrigo
Apellidos: Torrez
Edad: 60
Pais: Bolivia
Genero: Masculino
Tipo: Vip
```

```
Mostrando datos del Cliente
Nombre: Emeth
Apellidos: Brount
Edad: 38
Pais: Mexico
Genero: Masculino
Tipo: Silver
```

```
Mostrando datos del Cliente
Nombre: Louren
Apellidos: McFly
Edad: 68
Pais: Bolivia
Genero: Femenino
Tipo: Vip
```

```
Mostrando datos del Cliente
Nombre: George
Apellidos: McFly
Edad: 90
Pais: Bolivia
Genero: Masculino
Tipo: Silver
```

```
Mostrando datos del Cliente
Nombre: Saul
Apellidos: McFly
Edad: 70
Pais: Peru
Genero: Masculino
Tipo: Gold
```



## PARTE PRACTICA

### 14. Moviendo clientes en la cola.

- Mover al inicio todos los clientes mayores a 60 años.
  - Es decir si el cliente es mayor a 60 deberá de moverlo al inicio de la cola.
- El método recibe 2 parámetros
  - La Cola de Clientes
  - El valor(int) de la edad.
- Adjuntar los siguientes
  - El código del método que resuelve el problema.
  - Una imagen de la salida de la consola.
  - Link que me lleve a la clase main (GitHub)

```
public static void Mayores_a_60(ColaDeClientes cola, int edad){
    ColaDeClientes aux = new ColaDeClientes();
    ColaDeClientes aux1 = new ColaDeClientes();
    Cliente item;

    while (!cola.esVacio()){
        item=cola.eliminar();

        if(item.getEdad() > edad){
            aux.adicionar(item);
        }else {
            aux1.adicionar(item);
        }
    }
    cola.vaciar(aux);
    cola.vaciar(aux1);
    cola.mostrar();
}
```

ANTES

```
Mostrando datos del Cliente
Nombre: Rodrigo
Apellidos: Torrez
Edad: 60
Pais: Bolivia
Genero: Masculino
Tipo: Gold
```

```
Mostrando datos del Cliente
Nombre: Emeth
Apellidos: Brount
Edad: 38
Pais: Mexico
Genero: Masculino
Tipo: Silver
```

```
Mostrando datos del Cliente
Nombre: Louren
Apellidos: McFly
Edad: 68
Pais: Bolivia
Genero: Femenino
Tipo: Gold
```

```
Mostrando datos del Cliente
Nombre: George
Apellidos: McFly
Edad: 90
Pais: Bolivia
Genero: Masculino
Tipo: Silver
```

```
Mostrando datos del Cliente
Nombre: Saul
Apellidos: McFly
Edad: 70
Pais: Peru
Genero: Masculino
Tipo: Gold
```

DESPUES

```
Mostrando datos del Cliente
Nombre: Saul
Apellidos: McFly
Edad: 70
Pais: Peru
Genero: Masculino
Tipo: Gold
```

```
Mostrando la COLA DE CLIENTES:
Mostrando datos del Cliente
Nombre: Louren
Apellidos: McFly
Edad: 68
Pais: Bolivia
Genero: Femenino
Tipo: Gold
```

```
Mostrando datos del Cliente
Nombre: George
Apellidos: McFly
Edad: 65
Pais: Bolivia
Genero: Masculino
Tipo: Silver
```

```
Mostrando datos del Cliente
Nombre: Rodrigo
Apellidos: Torrez
Edad: 60
Pais: Bolivia
Genero: Masculino
Tipo: Gold
```

```
Mostrando datos del Cliente
Nombre: Emeth
Apellidos: Brount
Edad: 38
Pais: Mexico
Genero: Masculino
Tipo: Silver
```



## PARTE PRACTICA

### 15. Moviendo clientes entre 2 colas..

- Por razones de promociones de vuelo, es necesario cambiar de vuelo a ciertos clientes.
  - Crear 2 colas con 5 clientes.
  - Todos los clientes cuyo nombre sea **Saul** deberán ser agregados a la cola **B** al inicio.
- Adjuntar los siguientes
  - El código del método que resuelve el problema.
  - Una imagen de la salida de la consola.
  - Link que me lleve a la clase main (GitHub)

```
public static void main(String[] args) {  
  
    ColaDeClientes colaClientes=new ColaDeClientes();  
    Cliente cliente1 = new Cliente( nombres: "Rodrigo", apellidos: "Torrez", edad: 60, pais: "Bolivia", genero: "Masculino", tipo: "Gold");  
    Cliente cliente2 = new Cliente( nombres: "Emeth", apellidos: "Brount", edad: 38, pais: "Mexico", genero: "Masculino", tipo: "Silver");  
    Cliente cliente3 = new Cliente( nombres: "Louren", apellidos: "McFly", edad: 68, pais: "Bolivia", genero: "Femenino", tipo: "Gold");  
    Cliente cliente4 = new Cliente( nombres: "George", apellidos: "McFly", edad: 65, pais: "Bolivia", genero: "Masculino", tipo: "Silver");  
    Cliente cliente5 = new Cliente( nombres: "Saul", apellidos: "McFly", edad: 70, pais: "Peru", genero: "Masculino", tipo: "Gold");  
    //-----  
  
    ColaDeClientes colaClientes2=new ColaDeClientes();  
    Cliente cliente6 = new Cliente( nombres: "Biff", apellidos: "Tannen", edad: 50, pais: "Peru", genero: "Masculino", tipo: "Gold");  
    Cliente cliente7 = new Cliente( nombres: "Jose", apellidos: "Fernandez", edad: 34, pais: "Chile", genero: "Masculino", tipo: "Silver");  
    Cliente cliente8 = new Cliente( nombres: "Marty", apellidos: "Mercury", edad: 28, pais: "Bolivia", genero: "Masculino", tipo: "Silver");  
    Cliente cliente9 = new Cliente( nombres: "Mario", apellidos: "Choque", edad: 20, pais: "Mexico", genero: "Masculino", tipo: "Gold");  
    Cliente cliente10 = new Cliente( nombres: "Luigi", apellidos: "Conde", edad: 64, pais: "Argentina", genero: "Masculino", tipo: "Gold");  
  
    colaClientes.adicionar(cliente1);  
    colaClientes.adicionar(cliente2);  
    colaClientes.adicionar(cliente3);  
    colaClientes.adicionar(cliente4);  
    colaClientes.adicionar(cliente5);  
    colaClientes.mostrar();  
  
    colaClientes2.adicionar(cliente6);  
    colaClientes2.adicionar(cliente7);  
    colaClientes2.adicionar(cliente8);  
    colaClientes2.adicionar(cliente9);  
    colaClientes2.adicionar(cliente10);  
    colaClientes2.mostrar();  
}
```

ANTES

```
Mostrando datos del Cliente  
Nombre: Rodrigo  
Apellidos: Torrez  
Edad: 60  
Pais: Bolivia  
Genero: Masculino  
Tipo: Gold
```

```
Mostrando datos del Cliente  
Nombre: Emeth  
Apellidos: Brount  
Edad: 38  
Pais: Mexico  
Genero: Masculino  
Tipo: Silver
```

```
Mostrando datos del Cliente  
Nombre: Louren  
Apellidos: McFly  
Edad: 68  
Pais: Bolivia  
Genero: Femenino  
Tipo: Gold
```

```
Mostrando datos del Cliente  
Nombre: George  
Apellidos: McFly  
Edad: 90  
Pais: Bolivia  
Genero: Masculino  
Tipo: Silver
```

```
Mostrando datos del Cliente  
Nombre: Saul  
Apellidos: McFly  
Edad: 70  
Pais: Peru  
Genero: Masculino  
Tipo: Gold
```

DESPUES

```
Mostrando COLA de clientes B  
Mostrando la COLA DE CLIENTES:  
Mostrando datos del Cliente  
Nombre: Biff  
Apellidos: Tannen  
Edad: 50  
Pais: Peru  
Genero: Masculino  
Tipo: Gold
```

```
Mostrando la COLA DE CLIENTES:  
Mostrando datos del Cliente  
Nombre: Rodrigo  
Apellidos: Torrez  
Edad: 60  
Pais: Bolivia  
Genero: Masculino  
Tipo: Gold
```

```
Mostrando datos del Cliente  
Nombre: George  
Apellidos: McFly  
Edad: 65  
Pais: Bolivia  
Genero: Masculino  
Tipo: Silver
```

```
Mostrando datos del Cliente  
Nombre: Louren  
Apellidos: McFly  
Edad: 68  
Pais: Bolivia  
Genero: Femenino  
Tipo: Gold
```

```
Mostrando datos del Cliente  
Nombre: Emeth  
Apellidos: Brount  
Edad: 38  
Pais: Mexico  
Genero: Masculino  
Tipo: Silver
```