

UNIVERSIDAD PRIVADA "FRANZ TAMAYO"
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS

DEFENSA HITO 4 TAREA FINAL

Estudiante: Víctor Emanuel Quispe Pari

Asignatura: ESTRUCTURA DE DATOS

Carrera: INGENIERÍA DE SISTEMAS

Paralelo: EDD (1)

Docente: Lic. William Barra Paredes



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

Se refiere a una forma de organizar datos, de tal manera que nos permite realizar unas operaciones con ellas de manera eficiente y en definitiva diseñar la solución correcta para un determinado problema.

How many shortest-length paths are there to get from your house to the doughnut shop?

$\binom{n}{k} = \frac{n!}{(n-k)!k!}$

4 up's
4 right's

$\binom{11}{7} = \binom{11}{4} = 330$ paths

$e^{i\pi} + 1 = 0$

| P | Q | R | P V Q | P V R | (P V Q) ^ (P V R) |
|---|---|---|-------|-------|-------------------|
| T | T | T | T | T | T |
| T | T | F | T | T | T |
| T | F | T | T | T | T |
| T | F | F | T | T | T |
| F | T | T | T | T | T |
| F | T | F | T | F | F |
| F | F | T | F | T | F |
| F | F | F | F | F | F |

$\neg, 11, 15, 19, 23, \dots$

$a_1 - a_0 = 4$
 $a_2 - a_1 = 4$
 $a_3 - a_2 = 4$
 \vdots
 $a_n - a_{n-1} = 4$
 $a_n - a_0 = 4n$
 $a_n = a_0 + 4n$

Find $\neg + 12 + 17 + 22 + \dots + 342$

$S_n = \neg + 12 + 17 + 22 + \dots + 342$
 $+ S_n = 342 + 337 + 332 + 327 + \dots + \neg$
 $2S_n = 349 + 349 + 349 + 349 + \dots + 349$
 $2S_n = 349 \cdot 68$
 $S_n = \frac{349 \cdot 68}{2}$
 $S_n = 11866$

Original:
 $\exists x \forall y (x \geq 2y \rightarrow x > y + 1)$
Converse:
 $\exists x \forall y (x > y + 1 \rightarrow x \geq 2y)$
Negation:
 $\neg [\exists x \forall y (\neg (x \geq 2y) \vee x > y + 1)]$
 $\forall x \exists y (x \geq 2y \wedge x \leq y + 1)$
Contrapositive:
 $\exists x \forall y (x \leq y + 1 \rightarrow x < 2y)$

$v - e + f = 2$

P.I.E. Example:

$6! - \left[\binom{6}{1}5! - \binom{6}{2}4! + \binom{6}{3}3! - \binom{6}{4}2! + \binom{6}{5}1! \right]$

There are six dogs to give 13 tacos.

Use a 'stars and bars' diagram to illustrate the first and sixth dog get 3 tacos, the second dog gets none, the third dog gets 5 and the fourth dog gets one.

☆☆☆||☆☆☆☆☆|☆||☆☆☆|

$A = \{2, 4, \text{Ⓜ}, \text{⚙}\}$

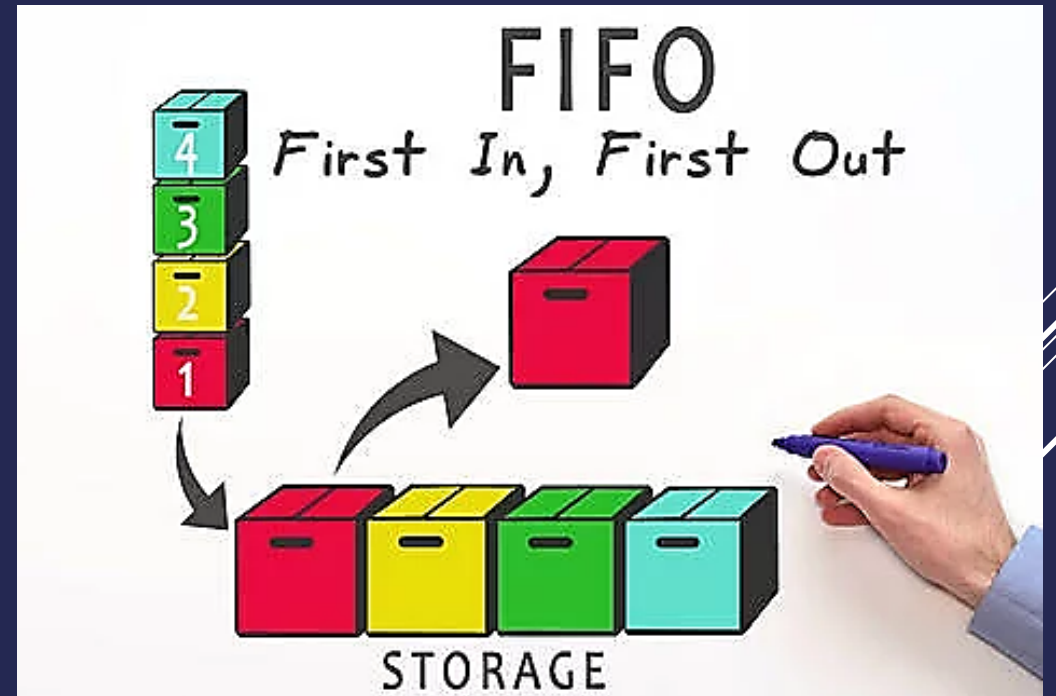
Onto

One-to-One

$(A \cup B \cup C) \cup (A \cap B \cap C)$

2. ¿Que significa FIFO?

Significa primero en entrar, primero en salir.
Es un concepto utilizado en estructuras de datos, contabilidad de costes y teoría de colas. Guarda analogía con las personas que esperan en una cola y van siendo atendidas en el orden en que llegaron, es decir, que "la primera persona que entra es la primera persona que sale".



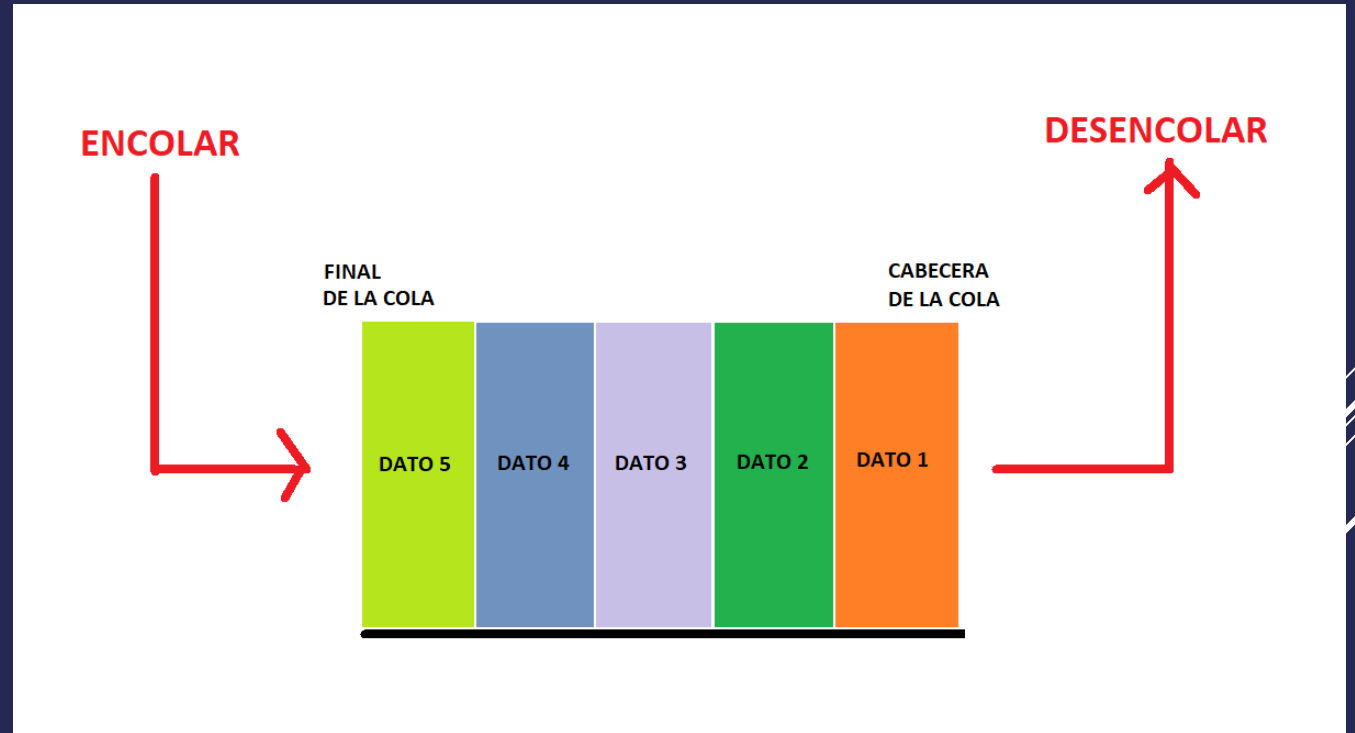
3. ¿Muestra la diferencia entre LIFO y FIFO?

| LIFO | FIFO |
|--|--|
| El nuevo elemento se inserta encima del elemento existente, de modo que el elemento más nuevo puede estar en la parte superior y eliminarse primero. | El nuevo elemento se inserta debajo del elemento existente, de modo que el elemento más antiguo puede estar en la parte superior y eliminarse primero. |
| Por lo tanto, el primer elemento que se ingresará en este enfoque, sale Último. | Por lo tanto, el primer elemento que se ingresará en este enfoque, sale Primero. |
| La estructura de datos que implementa LIFO es Pila. | La estructura de datos que implementa FIFO es Cola. |

4. ¿Qué es una COLA?

Una cola es una estructura de datos que almacena elementos en una lista y permite acceder a los datos por uno de los dos extremos de la lista.

Un elemento se inserta en la cola (parte final) de la lista y se suprime o elimina por la frente (parte inicial, cabeza) de la lista.



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

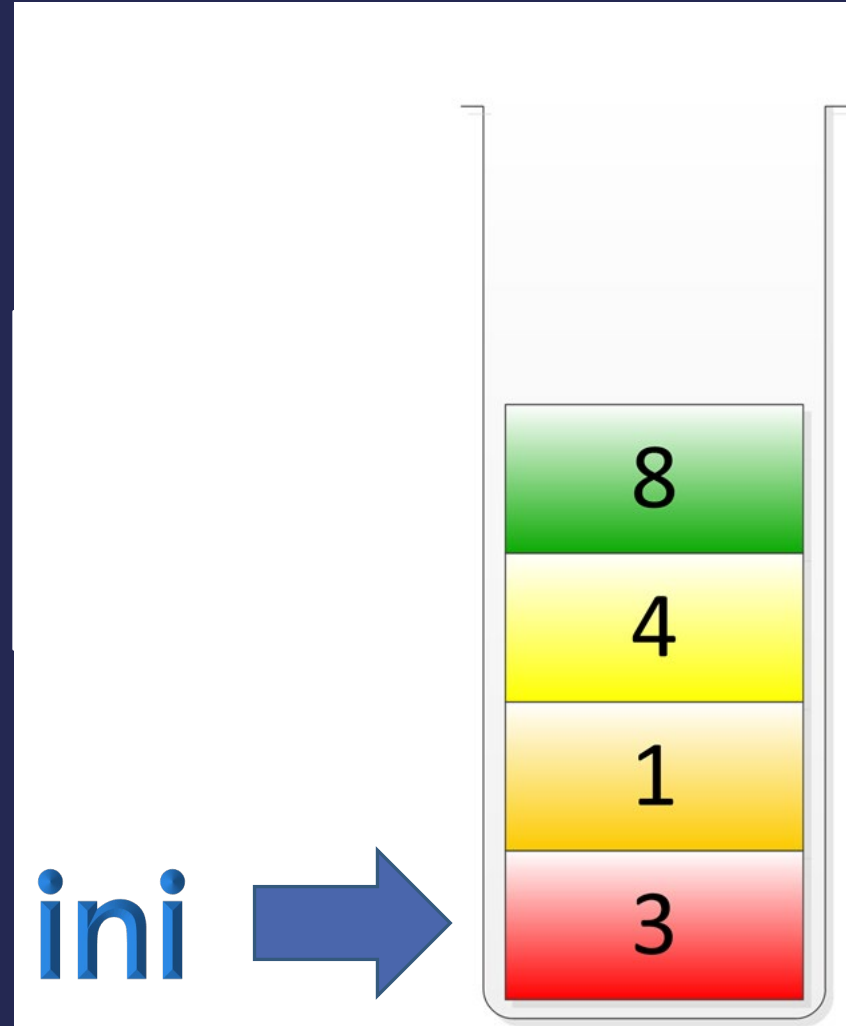
Queue en español es Cola, por lo tanto es lo mismo que una Cola. Pero en Java también es una interfaz que permite implementar de forma rápida y sencilla; la construcción de una cola, agregar elementos y luego irlos retirando.

**COLA = QUEUE
(Cola en Inglés)**

```
1. package test;
2.
3. import java.util.LinkedList;
4. import java.util.Queue;
5.
6. public class Colas {
7.
8.     public static void main(String[] args) {
9.         Queue<Integer> cola = new LinkedList<>();
10.
11.         System.out.println("Agregando valores");
12.         for (int i = 0; i < 5; i++) {
13.             cola.add(i);
14.             System.out.println("Valor: " + i);
15.         }
16.
17.         System.out.println("\nRetirando valores");
18.         while (cola.peek() != null) {
19.             System.out.println("Valor: " + cola.poll());
20.         }
21.     }
22. }
```


6. ¿Qué es INI o REAR en una COLA?

Es una variable de la clase Cola, que cumple la función de almacenar la posición del primer elemento para indicar donde se realiza la eliminación de un elemento.



7. ¿Qué es FIN o FRONT en una COLA?

Es una variable de la clase Cola, que cumple la función de almacenar la posición siguiente de la Cola, para indicar donde se realizará la inserción de un nuevo elemento.

fin →



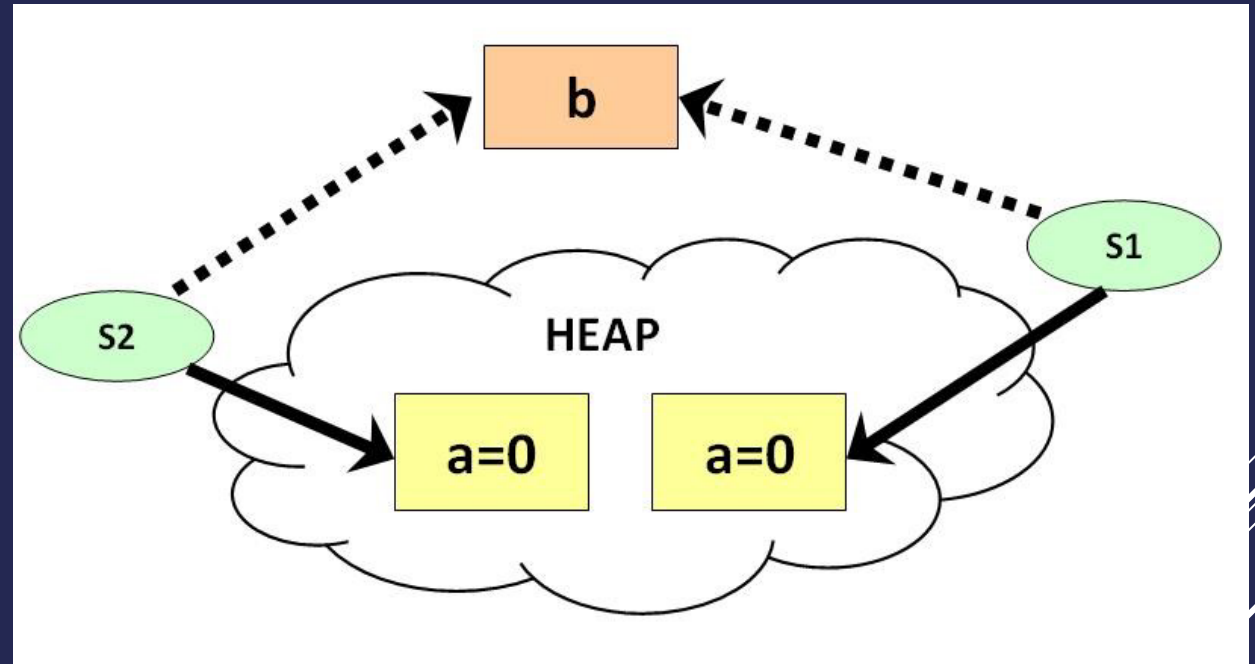
8. ¿A que se refiere los métodos `esVacia()` y `esLlena()` en una COLA?

- `esVacia()`: Se refiere al método que devuelve un valor booleano que indica si la Cola esta vacía o no.
- `esLlena()`: Se refiere al método que devuelve un valor booleano que indica si la Cola esta llena o no.

```
public boolean esVacia() {  
    if (this.ini == 0 && this.fin == 0) {  
        return true;  
    } else {  
        return false;  
    }  
}  
  
public boolean esLlena() {  
    if (this.fin == this.max) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

9. ¿Qué son los métodos estáticos en JAVA?

Un método estático es un método que tiene sentido invocarla sin crear previamente ningún objeto, esto quiere decir que pertenece a la clase y solo tiene acceso a datos estáticos.

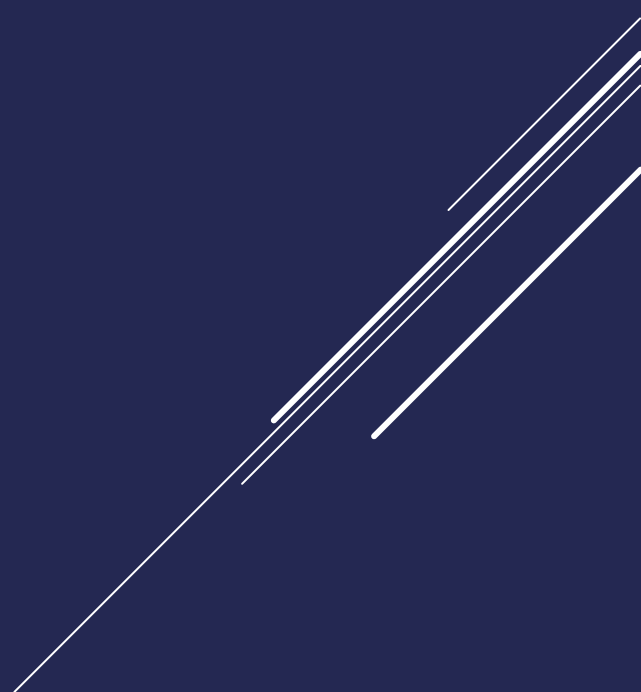


10.¿A través de un gráfico, muestre los métodos mínimos que debería de tener una COLA?

| C  Numeros | | |
|--|------------------|---------|
|   | vaciar (Numeros) | void |
|   | esLlena () | boolean |
|   | nroElem () | int |
|   | adicionar (int) | void |
|   | esVacia () | boolean |
|   | eliminar () | int |
|   | mostrar () | void |

11. Crear las clases necesarias para la COLA DE CLIENTES.

- Crear la clase Cliente.
- Crear la clase ColaDeClientes.
- Crear la clase Main.
- Crear un paquete de nombre ColaDeClientes (todas las clases deberán de estar dentro de este paquete)



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.**
- 
- A series of four parallel white lines of varying lengths, slanted upwards from left to right, located in the bottom right corner of the slide.

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.

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.

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.

GRACIAS

<https://github.com/VQP5/EDA311/blob/main/Hito4/Procesual/Cliente.java>

<https://github.com/VQP5/EDA311/blob/main/Hito4/Procesual/ColaDeClientes.java>

<https://github.com/VQP5/EDA311/blob/main/Hito4/Procesual/Main.java>