

# ESTRUCTURA DE DATOS

H4

# DATOS PERSONALES

NOMBRE COMPLETO: MIJAIL OLIVER CHOQUE AMARO

MATERIA: EDD(ESTRUCTURA DE DATOS)

DOCENTE: WILLIAM BARRA PAREDES

FECHA DE ENTREGA: 14/06/2023 00:00:00

UNIVERSIDAD: UNIFRANZ



eclipse

Manejo de conceptos

H4



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

- Cuando se habla de Estructura de Datos, se refiere a la organización y almacenamiento de datos de manera eficiente, de modo que se pueda acceder y manipular de manera efectiva. Las estructuras de datos proporcionan métodos y operaciones para almacenar, organizar y gestionar datos en diferentes formas, dependiendo de los requisitos específicos.

## 2. ¿Que significa FIFO?

- FIFO significa "First-In, First-Out" (Primero en entrar, primero en salir). Es una política de manejo de datos en la que el primer elemento que ingresa es el primero en ser procesado o eliminado.

### 3. ¿Muestra la diferencia entre LIFO y FIFO?

- La diferencia entre LIFO (Last-In, First-Out) y FIFO (First-In, First-Out) radica en el orden en el que se procesan o eliminan los elementos de una estructura de datos. En LIFO, el último elemento que ingresa es el primero en ser procesado o eliminado, mientras que en FIFO, el primer elemento que ingresa es el primero en ser procesado o eliminado.



## 4. ¿Qué es una COLA?

- Una cola es una estructura de datos en la que los elementos se insertan al final y se eliminan desde el principio. Sigue el principio FIFO, donde el primer elemento que ingresa es el primero en ser procesado o eliminado.

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

- En Java, "Queue" es una interfaz que representa una cola y define los métodos básicos para operar con ella. Sí, una "Queue" en Java se refiere a una cola en términos generales, siguiendo el principio FIFO.



## 6. ¿Qué es INI o REAR en una COLA

- En una cola, "INI" o "REAR" se refiere al extremo posterior de la cola, donde se insertan los elementos nuevos.

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

- En una cola, "FIN" o "FRONT" se refiere al extremo frontal de la cola, desde donde se eliminan los elementos.

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

- Adjunte los métodos

- Los métodos "esVacia()" y "esLlena()" en una cola se utilizan para verificar si la cola está vacía o llena, respectivamente.

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

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



## 9. ¿Qué son los métodos estáticos en JAVA?

- Los métodos estáticos en Java son aquellos que pertenecen a la clase en sí, no a una instancia específica de la clase. Pueden ser llamados sin tener que crear un objeto de la clase. Los métodos estáticos se utilizan para operaciones que no requieren acceder o modificar variables de instancia, y son útiles para funciones utilitarias, operaciones matemáticas, etc.

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

- Generar el diagrama con el editor INTELLIJ IDEA



Parte practica

H4



# EJERCICIO 11



Cliente	
Cliente (String, String, int, String, String, String)	
nombres	String
apellidos	String
edad	int
tipo	String
genero	String
pais	String
getPais ()	String
getNombres ()	String
setTipo (String)	void
setGenero (String)	void
getTipo ()	String
getGenero ()	String
setNombres (String)	void
setApellidos (String)	void
getEdad ()	int
setEdad (int)	void
getApellidos ()	String
setPais (String)	void
mostrar ()	void

ColaDeClientes	
ColaDeClientes ()	
max	int
ini	int
cliente	Cliente []
fin	int
eliminar ()	Cliente
getIni ()	int
esLlena ()	boolean
mostrar ()	void
getFin ()	int
nroElem ()	int
adicionar (Cliente)	void
vaciar (ColaDeClientes)	void
esVacia ()	boolean

Mostrando la COLA DE CLIENTES

Cliente  
 Nombre: John  
 Apellidos: Doe  
 Edad: 30  
 Pais: Estados Unidos  
 Genero: Masculino  
 Tipo: GOLD

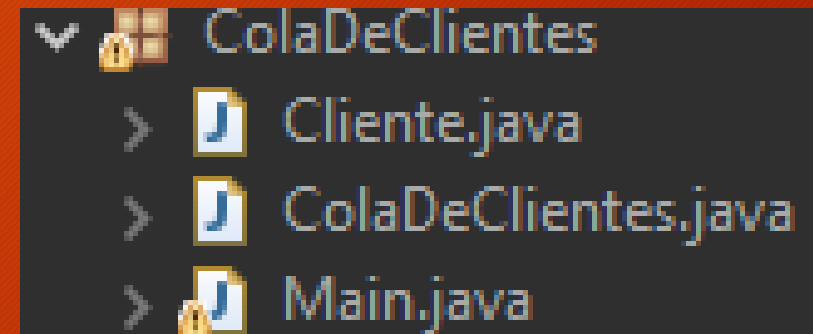
Cliente  
 Nombre: Karen  
 Apellidos: Smith  
 Edad: 25  
 Pais: Canadá  
 Genero: Femenino  
 Tipo: VIP

Cliente  
 Nombre: Saul  
 Apellidos: Pérez  
 Edad: 60  
 Pais: Bolivia  
 Genero: Masculino  
 Tipo: GOLD

Cliente

LINK: <https://github.com/choqueoliver35/ESTRUCTURA-DE-DATOS/blob/main/HITO%204/EVALUACION%20PROCESUAL/ColaDeClientes/Main.java>

```
Cliente cliente1 = new Cliente("John", "Doe",  
30, "Estados Unidos", "Masculino", "GOLD");  
Cliente cliente2 = new Cliente("Karen",  
"Smith", 25, "Canadá", "Femenino", "VIP");  
Cliente cliente3 = new Cliente("Saul", "Pérez",  
61, "Bolivia", "Masculino", "GOLD");  
Cliente cliente4 = new Cliente("Robert",  
"Gomez", 28, "Argentina", "Femenino", "VIP");  
Cliente cliente5 = new Cliente("Bruss", "Lee",  
71, "Bolivia", "Masculino", "GOLD");  
ColaDeClientes cola = new ColaDeClientes();  
cola.adicionar(cliente1);  
cola.adicionar(cliente2);  
cola.adicionar(cliente3);  
cola.adicionar(cliente4);  
cola.adicionar(cliente5);  
cola.mostrar();
```



LINK: <https://github.com/choqueoliver35/ESTRUCTURA-DE-DATOS/blob/main/HITO%204/EVALUACION%20PROCESUAL/ColaDeClientes/Main.java>



# EJERCICIO 12

```
Mostrando la COLA DE CLIENTES
```

```
Cliente
```

```
Nombre: John
```

```
Apellidos: Doe
```

```
Edad: 30
```

```
Pais: Estados Unidos
```

```
Genero: Masculino
```

```
Tipo: GOLD
```

```
Cliente
```

```
Nombre: Karen
```

```
Apellidos: Smith
```

```
Edad: 25
```

```
Pais: Canadá
```

```
Genero: Femenino
```

```
Tipo: VIP
```

```
Cliente
```

```
Nombre: Saul
```

```
Apellidos: Pérez
```

```
Edad: 60
```

```
Pais: Bolivia
```

```
Genero: Masculino
```

```
Tipo: GOLD
```

```
Cliente
```

LINK: <https://github.com/choqueoliver35/ESTRUCTURA-DE-DATOS/tree/main/HITO%204/EVELUACION%20PROCESUAL/ColaDeClientes>

```
Cliente cliente1 = new Cliente("John", "Doe", 30,
"Estados Unidos", "Masculino", "GOLD");
Cliente cliente2 = new Cliente("Karen", "Smith",
25, "Canadá", "Femenino", "VIP");
Cliente cliente3 = new Cliente("Saul", "Pérez", 61,
"Bolivia", "Masculino", "GOLD");
Cliente cliente4 = new Cliente("Robert", "Gomez",
28, "Argentina", "Femenino", "VIP");
Cliente cliente5 = new Cliente("Bruss", "Lee", 71,
"Bolivia", "Masculino", "GOLD");
ColaDeClientes cola = new ColaDeClientes();
cola.adicionar(cliente1);
cola.adicionar(cliente2);
cola.adicionar(cliente3);
cola.adicionar(cliente4);
cola.adicionar(cliente5);
cola.mostrar();
```

# EJERCICIO 13

```
("John", "Doe", 30, "Estados Unidos", "Masculino", "GOLD");  
("Karen", "Smith", 25, "Canadá", "Femenino", "VIP");  
("Saul", "Pérez", 61, "Bolivia", "Masculino", "GOLD");  
("Robert", "Gomez", 28, "Argentina", "Femenino", "VIP");  
("Bruss", "Lee", 71, "Bolivia", "Masculino", "GOLD");
```

```
DeClientes();
```

```
P","Bolivia");
```

```
Cliente  
Nombre: Saul  
Apellidos: Pérez  
Edad: 61  
Pais: Bolivia  
Genero: Masculino  
Tipo: VIP
```

```
Cliente  
Nombre: Robert  
Apellidos: Gomez  
Edad: 28  
Pais: Argentina  
Genero: Femenino  
Tipo: VIP
```

```
Cliente  
Nombre: Bruss  
Apellidos: Lee  
Edad: 71  
Pais: Bolivia  
Genero: Masculino  
Tipo: VIP
```

LINK: <https://github.com/choqueoliver35/ESTRUCTURA-DE-DATOS/tree/main/HITO%204/EVELUACION%20PROCESUAL/ColaDeClientes>



```
private static void
promocionNacionalidad(ColaDeClientes cola,
String_TipoCliente, String_Nacionalidad) {
// TODO Auto-generated method stub
Cliente item = null;
int nroElem = cola.nroElem();
for(int i = 1; i <= nroElem; i++) {
item = cola.eliminar();
if(item.getTipo() != TipoCliente &&
item.getPais().equals(Nacionalidad)) {
item.setTipo(TipoCliente);
}
else {
}
cola.adicionar(item);
}
}
```

# EJERCICIO 14

```
Mostrando la COLA DE CLIENTES
Cliente
Nombre: Saul
Apellidos: Pérez
Edad: 61
Pais: Bolivia
Genero: Masculino
Tipo: GOLD

Cliente
Nombre: Bruss
Apellidos: Lee
Edad: 71
Pais: Bolivia
Genero: Masculino
Tipo: GOLD

Cliente
Nombre: John
Apellidos: Doe
Edad: 30
Pais: Estados Unidos
Genero: Masculino
Tipo: GOLD
```

LINK: <https://github.com/choqueoliver35/ESTRUCTURA-DE-DATOS/tree/main/HITO%204/EVELUACION%20PROCESUAL/ColaDeClientes>

```
private static void subirarriba(ColaDeClientes  
cola, int edad) {  
    // TODO Auto-generated method stub  
    Cliente item = null;  
    ColaDeClientes aux1 = new ColaDeClientes();  
    ColaDeClientes aux2 = new ColaDeClientes();  
    while(!cola.esVacia()) {  
        item = cola.eliminar();  
        if(item.getEdad() > 60) {  
            aux1.adicionar(item);  
        }  
        else {  
            aux2.adicionar(item);  
        }  
    }  
    cola.vaciar(aux1);  
    cola.vaciar(aux2);  
}
```



# EJERCICIO 15

```
//subirarriba(cola,60);

//cola.mostrar();

//Ejercicio 15

Cliente cliente6 = new Cliente("John", "Doe", 30, "Estados Unidos", "Masc
Cliente cliente7 = new Cliente("Saul", "Smith", 25, "Canadá", "Femenino",
Cliente cliente8 = new Cliente("Juan", "Pérez", 60, "México", "Masculino"
Cliente cliente9 = new Cliente("Maria", "Gomez", 28, "Argentina", "Femeni
Cliente cliente10 = new Cliente("Robert", "Lee", 60, "Bolivia", "Masculin

ColaDeClientes cola2 = new ColaDeClientes();
cola2.adicionar(cliente6);
cola2.adicionar(cliente7);
cola2.adicionar(cliente8);
cola2.adicionar(cliente9);
cola2.adicionar(cliente10);

cambiarVuelo(cola,cola2,"Saul");

cola2.mostrar();

}
```

<terminated> main [Java Application] C:\Users\usuario\.p2\pool\p  
Genero: Masculino  
Tipo: GOLD  
  
Mostrando la COLA DE CLIENTES  
Cliente  
Nombre: Saul  
Apellidos: Pérez  
Edad: 61  
Pais: Bolivia  
Genero: Masculino  
Tipo: GOLD  
  
Cliente  
Nombre: John  
Apellidos: Doe  
Edad: 30  
Pais: Estados Unidos  
Genero: Masculino  
Tipo: GOLD  
  
Cliente  
Nombre: Saul  
Apellidos: Smith  
Edad: 25  
Pais: Canadá  
Genero: Femenino

LINK: <https://github.com/choqueoliver35/ESTRUCTURA-DE-DATOS/tree/main/HITO%204/EVELUACION%20PROCESUAL/ColaDeClientes>

```
private static void cambiarVuelo(ColaDeClientes
cola, ColaDeClientes cola2, String nombre) {
// TODO Auto-generated method stub
Cliente item = null;
int nroElem = cola.nroElem();
ColaDeClientes aux1 = new ColaDeClientes();
ColaDeClientes aux2 = new ColaDeClientes();
for(int i = 1; i <= nroElem; i++) {
item = cola.eliminar();
if(item.getNombres().equals(nombre)) {
aux1.adicionar(item);
}
cola.adicionar(item);
}
aux2.vaciar(cola2);
cola2.vaciar(aux1);
cola2.vaciar(aux2);
}
```

MUCHAS GRACIAS

H4