

# PROCESUAL HITO 4

## ESTRUCTURA DE DATOS

JOEL REYNALDO CONDORI TUMIRI

# 1. ¿A QUE SE REFIERE CUANDO SE HABLA DE ESTRUCTURA DE DATOS?

- las estructuras de datos son aquellas que nos permiten, como desarrolladores, organizar la información de manera eficiente, y en definitiva diseñar la solución correcta para un determinado problema

## 2. ¿QUÉ SIGNIFICA FIFO?

**FIFO: Primero En Entrar, Primero en Salir**

**Del inglés: "First In, First Out" La definición del FIFO es simple: el lote de stock que primera entra, es el que primero sale. Es el método más utilizado para los productos perecederos como los alimentos, que tienen fecha de caducidad.**

### 3. ¿MUESTRA LA DIFERENCIA ENTRE LIFO Y FIFO?

- El el método FIFO las primeras mercancías en entrar serán las primeras en salir, mientras que en el método LIFO, las ultimas mercancías en entrar serán las primeras en salir.



## 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?

Una Cola o Queue es una **estructura de datos que sigue la Filosofía FIFO del ingles First In – First Out** que en español sería “Primero en entrar primero en salir”.

Queue es lo mismo que Cola.

## 6. ¿QUÉ ES INI O REAR EN UNA COLA?

Un INI es una variable utilizada para saber la posición real del primer elemento.

## 7. ¿Qué es INI o REAR en una COLA?

Un fin es la variable que marca la siguiente posición en entrar a la cola

## 8. ¿A QUE SE REFIERE LOS MÉTODOS ESVACIA() Y ESLLENA() EN UNA COLA?

El método esVacia() es cuando la cola no contiene elementos almacenados, de forma que para realizar cambios, primero habrá que almacenar información en la cola.

El método esLlena() es cuando FIN es igual a INI, impidiendo poder seguir llenando la cola.



## 9. ¿QUÉ SON LOS MÉTODOS ESTÁTICOS EN JAVA?

- Un método estático es un **método** que tiene sentido invocarla sin crear previamente ningún objeto.

# 10. A TRAVÉS DE UN GRÁFICO, MUESTRE LOS MÉTODOS MÍNIMOS QUE DEBERÍA DE TENER UNA COLA.

```
public class Cliente {  
    3 usages  
    private String Nombres;  
    3 usages  
    private String Apellidos;  
    3 usages  
    private int Edad;  
    3 usages  
    private String Pais;  
    3 usages  
    private String Genero;  
    3 usages  
    private String tipo;  
  
    10 usages  
    public Cliente(String Nombres, String Apellidos, int Edad, String Pais, String Genero, String tipo) {  
        this.Nombres = Nombres;  
        this.Apellidos = Apellidos;  
        this.Edad = Edad;  
        this.Pais = Pais;  
        this.Genero = Genero;  
        this.tipo = tipo;  
    }  
  
    3 usages  
    public String getNombres() { return Nombres; }  
  
    1 usage  
    public String getApellidos() { return Apellidos; }  
  
    2 usages  
    public int getEdad() { return Edad; }  
  
    2 usages  
    public String getPais() { return Pais; }  
  
    1 usage  
    public String getGenero() { return Genero; }
```

```
    2 usages  
    public int getEdad() { return Edad; }  
  
    2 usages  
    public String getPais() { return Pais; }  
  
    1 usage  
    public String getGenero() { return Genero; }  
  
    2 usages  
    public String getTipo() { return tipo; }  
  
    public void setNombres(String nombres) { Nombres = nombres; }  
  
    public void setApellidos(String apellidos) { Apellidos = apellidos; }  
  
    public void setEdad(int edad) { Edad = edad; }  
  
    public void setPais(String pais) { Pais = pais; }  
  
    public void setGenero(String genero) { Genero = genero; }  
  
    1 usage  
    public void setTipo(String tipo) { this.tipo = tipo; }  
  
    1 usage  
    public void mostrarCliente() {  
        System.out.println("\nMostrando datos del jugador");  
        System.out.println("Nombre: " + this.getNombres());  
        System.out.println("Apellidos: " + this.getApellidos());  
        System.out.println("Edad: " + this.getEdad());  
        System.out.println("Pais: " + this.getPais());  
        System.out.println("Genero: " + this.getGenero());  
        System.out.println("Tipo: " + this.getTipo());  
  
        System.out.println("\n");  
    }  
}
```



Project: practitando C:\Users\Joel\Idea

- > .idea
- > out
- > src
  - ColaDeClientes
    - Cliente
    - ColaClientes
    - Main
- practicando.iml
- External Libraries
- Scratches and Consoles

Database

Notifications

### Cliente

f	tipo	String
f	Apellidos	String
f	Genero	String
f	Pais	String
f	Nombres	String
f	Edad	int
m	setTipo(String)	void
m	getTipo()	String
m	getEdad()	int
m	getApellidos()	String
m	setPais(String)	void
m	setApellidos(String)	void
m	setGenero(String)	void
m	setEdad(int)	void
m	getPais()	String
m	getNombres()	String
m	getGenero()	String
m	mostrarCliente()	void
m	setNombres(String)	void

Powered by yFiles

Version Control Run TODO Problems Terminal Services Profiler Build

Build completed successfully in 968 ms (15 minutes ago)

Project: practitando C:\Users\Joel\Idea

- > .idea
- > out
- > src
  - ColaDeClientes
    - Cliente
    - ColaClientes
    - Main
- practicando.iml
- External Libraries
- Scratches and Consoles

Database

Notifications

### ColaClientes

m	ColaClientes(int)	
f	max	int
f	fin	int
f	ini	int
f	Clientes	Cliente[]
m	adicionar(Cliente)	void
m	eliminar()	Cliente
m	mostrar()	void
m	esLlena()	boolean
m	nroElem()	int
m	esVacia()	boolean
m	vaciar(ColaClientes)	void

Powered by yFiles

Structure Bookmarks



# 11. CREAR LAS CLASES NECESARIAS PARA LA PILA DE CLIENTES

```
Project
Main.java x ColaClientes.java x Cliente.java x
1 package ColaDeClientes;
2
3 public class Main {
4     public static void main(String[] args) {
5         Cliente c1 = new Cliente( Nombres: "Cristian", Apellidos: "Achu", Edad: 20, Pais: "Brasil", Genero: "Masculino", tipo: "Silver");
6         Cliente c2 = new Cliente( Nombres: "Dania", Apellidos: "Mamani", Edad: 30, Pais: "Chile", Genero: "Femenino", tipo: "Gold");
7         Cliente c3 = new Cliente( Nombres: "Marcos", Apellidos: "Castaneda", Edad: 80, Pais: "Mexico", Genero: "Masculino", tipo: "Vip");
8         Cliente c4 = new Cliente( Nombres: "Daniel", Apellidos: "Rivaldo", Edad: 60, Pais: "Paraguay", Genero: "Masculino", tipo: "Silver");
9         Cliente c5 = new Cliente( Nombres: "Ana", Apellidos: "Guzman", Edad: 65, Pais: "Bolivia", Genero: "Femenino", tipo: "Gold");
10
11
12         ColaClientes colaCli = new ColaClientes( max: 100);
13         colaCli.adicionar(c1);
14         colaCli.adicionar(c2);
15         colaCli.adicionar(c3);
16         colaCli.adicionar(c4);
17         colaCli.adicionar(c5);
18
19         //colaCli.mostrar();
```

# 12.INICIALIZAR LA COLA DE CLIENTES

```
C:\Users\Joel\.jdk\openjdk-18.0.2\bin\java.exe "-javaagent:E:\
Datos de la Cola de clientes

Mostrando datos del jugador
Nombre: Cristian
Apellidos: Achu
Edad: 20
Pais: Brasil
Genero: Masculino
Tipo: Silver

Mostrando datos del jugador
Nombre: Dania
Apellidos: Mamani
Edad: 30
Pais: Chile
Genero: Femenino
Tipo: Gold
```

```
Run: Main x
Mostrando datos del jugador
Nombre: Marcos
Apellidos: Castaneda
Edad: 80
Pais: Mexico
Genero: Masculino
Tipo: Vip

Mostrando datos del jugador
Nombre: Daniel
Apellidos: Rivaldo
Edad: 60
Pais: Paraguay
Genero: Masculino
Tipo: Silver

Mostrando datos del jugador
Nombre: Ana
Apellidos: Guzman
Edad: 65
Pais: Bolivia
Genero: Femenino
Tipo: Gold

Process finished with exit code 0
Version Control Run TODO Problems Terminal Service
Build completed successfully in 1 sec 476 ms (moments ago)
```

# 13.PROMOCION PARA USUARIOS DE BOLIVIA

```
src
├── ColaDeClientes
│   ├── Cliente
│   └── ColaClientes
└── Main
    ├── practicante.iml
    └── External Libraries
        └── Scratches and Consoles

23
24
25 // En el mes de diciembre a todos los clientes de Bolivia se les dará una promoción
26 // en cuanto a precios en viajes a nivel nacional.
27 // A todos los clientes que sean de nacionalidad boliviana y además el tipo
28 // de cliente GOLD, convertir a estos clientes en VIP
29 // Es decir si es de Bolivia y es GOLD deberá ser ahora un cliente VIP
30 // El método estático dentro de la clase MAIN recibe 3 atributos
31 // La cola de clientes
32 // El tipo de cliente
33 // La nacionalidad del cliente
34 @ public static void cambiarTipo(ColaClientes Cola, String Pais, String tipo,String NewTipo) {
35     ColaClientes aux = new ColaClientes( max: 10);
36     Cliente Valorextraido = null;
37     while (!Cola.esVacia()) {
38         Valorextraido = Cola.eliminar();
39         if (Valorextraido.getPais().equals(Pais) && Valorextraido.getTipo().equals(tipo)) {
40             Valorextraido.setTipo(NewTipo);
41             aux.adicionar(Valorextraido);
42         }
43         aux.adicionar(Valorextraido);
44     }
45     Cola.vaciar(aux);
46     Cola.mostrar();
47 }
48
```

Mostrando datos del jugador

Nombre: Ana

Apellidos: Guzman

Edad: 65

Pais: Bolivia

Genero: Femenino

Tipo: Vip

Process finished with exit code 0

```
Cliente c5 = new Cliente( Nombres: "Ana", Apellidos: "Guzman", Edad: 65, Pais: "Bolivia", Genero: "Femenino", tipo: "Gold");
```

# 14.MOVIENDO CLIENTES EN LA COLA

```
48
49 //Determinar cuantos clientes son mayores a 60 años
50 //Mover al inicio todos los clientes mayores a 60 años.
51 //Es decir si el cliente es mayor a 60 deberá de moverlo al inicio de la cola.
52 //El método recibe 2 parámetros
53 //La Cola de Clientes
54 //El valor(int) de la edad
55 @ public static void MoverClientesMayores(ColaClientes cola, int edadMayor) {
56     ColaClientes aux = new ColaClientes( max: 100);
57     ColaClientes aux2 = new ColaClientes( max: 100);
58     Cliente Valorextraido = null;
59     while (!cola.esVacia()) {
60         Valorextraido = cola.eliminar();
61         if (Valorextraido.getEdad() >= edadMayor) {
62             aux2.adicionar(Valorextraido);
63         }
64         aux.adicionar(Valorextraido);
65     }
66     cola.vaciar(aux2);
67     cola.vaciar(aux);
68     cola.mostrar();
69 }
```

```
Run: Main x |
C:\Users\Joel\.jdk\openjdk-18.0.2\bin\java.exe "-javaagent:E:\In

Datos de la Cola de clientes

Mostrando datos del jugador
Nombre: Marcos
Apellidos: Castaneda
Edad: 80
Pais: Mexico
Genero: Masculino
Tipo: Vip
```