

PROCESUAL **HITO 4**

Josias Jonathan Leon Luis

Estructura De Datos

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

En el ámbito de la informática, 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 FIN o FRONT 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");  
}
```


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

```
public class ColaClientes {  
  
    3 usages  
    private int max;  
    9 usages  
    private int fin;  
    7 usages  
    private int ini;  
    3 usages  
    private Cliente[] clientes;  
  
    8 usages  
    public ColaClientes(int max) {  
        this.max = max;  
        this.clientes = new Cliente[this.max + 1];  
        this.ini = 0;  
        this.fin = 0;  
    }  
  
    6 usages  
    public boolean esVacia() {  
        if (this.ini == 0 & this.fin == 0) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
  
    1 usage  
    public boolean esLlena() {  
        if (this.fin == this.max) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
  
    public int nroElem() { return fin - ini; }  
}
```

```
}  
}  
public Cliente eliminar() {  
    Cliente elementoEliminado = null;  
    if (!esVacia()) {  
        this.ini++;  
        elementoEliminado = this.clientes[ini];  
        if (ini == fin) {  
            ini = 0;  
            fin = 0;  
        }  
        return elementoEliminado;  
    } else {  
        System.out.println("Cola de numeros vacia");  
    }  
    return elementoEliminado;  
}  
  
public void mostrar () {  
    Cliente elem = null;  
    if (esVacia())  
        System.out.println("Cola Vacia");  
    else {  
        System.out.println("\nDatos de la Cola de clientes");  
        ColaClientes aux = new ColaClientes(fin);  
        while (!esVacia()) {  
            elem = this.eliminar();  
            aux.adicionar (elem);  
            elem.mostrarCliente();  
        }  
        vaciar(aux);  
    }  
}  
  
public void vaciar (ColaClientes cola) {  
    while (!cola.esVacia())  
        adicionar(cola.eliminar());  
}
```

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

```
public class Main {  
    public static void main(String[] args) {  
        Cliente cli1 = new Cliente( Nombres: "Josias", Apellidos: "Leon", Edad: 33, Pais: "Bolivia", Genero: "Masculino", tipo: "SILVER");  
        Cliente cli2 = new Cliente( Nombres: "Adolf", Apellidos: "Hitler", Edad: 56, Pais: "Bolivia", Genero: "Masculino", tipo: "GOLD");  
        Cliente cli3 = new Cliente( Nombres: "Alejandra", Apellidos: "Maine", Edad: 19, Pais: "Chile", Genero: "Femenino", tipo: "VIP");  
        Cliente cli4 = new Cliente( Nombres: "Josef", Apellidos: "Stalin", Edad: 74, Pais: "Argentina", Genero: "Masculino", tipo: "GOLD");  
        Cliente cli5 = new Cliente( Nombres: "Saul", Apellidos: "Saulero", Edad: 42, Pais: "Rusia", Genero: "Masculino", tipo: "VIP");  
  
        ColaClientes Cola = new ColaClientes( max: 100);  
        Cola.adicionar(cli1);  
        Cola.adicionar(cli2);  
        Cola.adicionar(cli3);  
        Cola.adicionar(cli4);  
        Cola.adicionar(cli5);  
    }  
}
```



12. Inicializar la cola de clientes.

```
ucturaDeDatos DAEDA 4 ▶ public static void main(String[] args) {
idea 5
HITO3 6
out 7
src 8
  Campeonato.Clases 9
  Ejercicio_1 10
  H2DEFENSA 11
  Hito3 12
  > Lab1 13
  > Lab2 14
  > Lab3 15
  > Lab4 16
  > PilaDeCadenas 17
  > PilaDeClientes
Practica_H4.Main x

Datos de la Cola de clientes

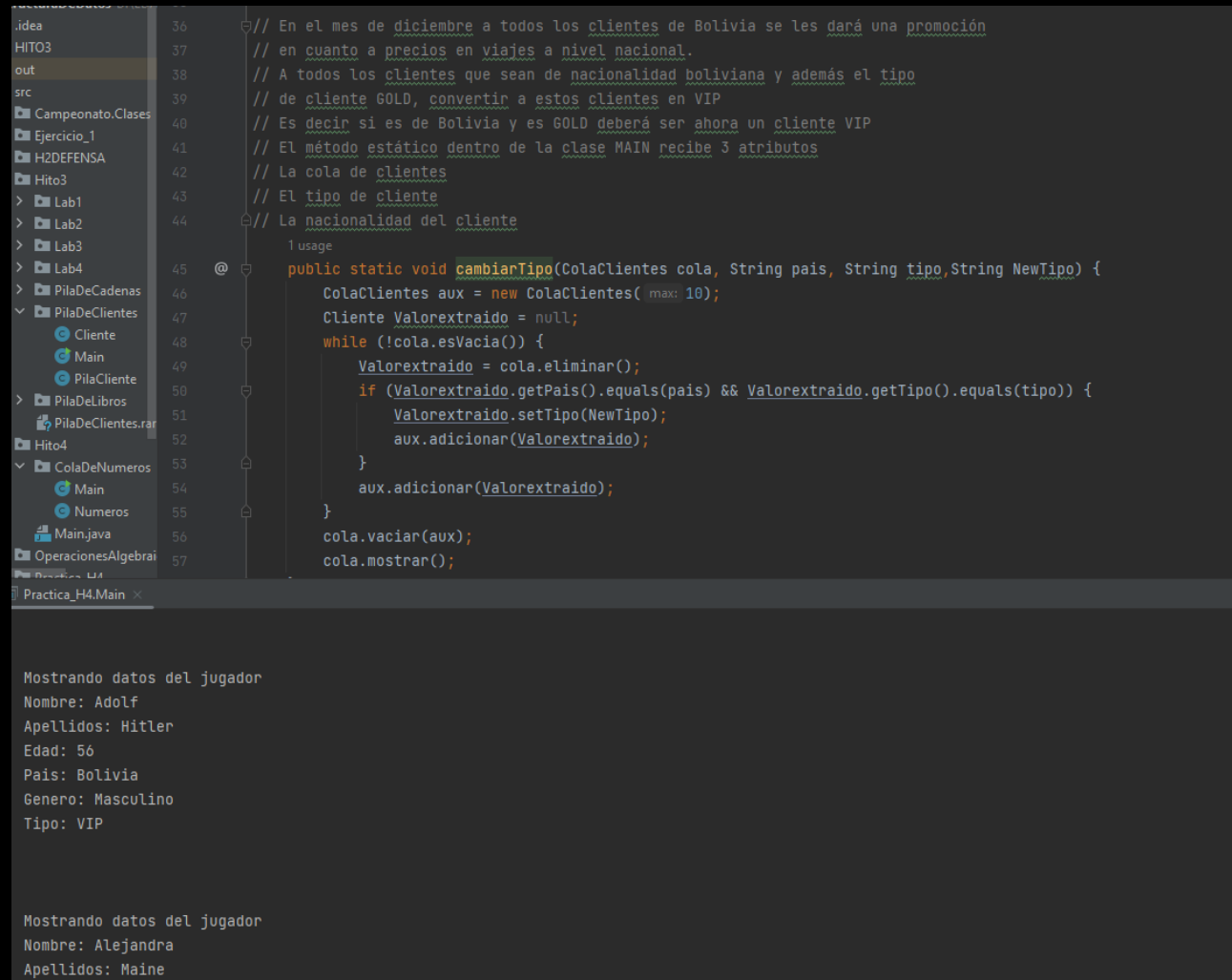
Mostrando datos del jugador
Nombre: Josias
Apellidos: Leon
Edad: 33
Pais: Bolivia
Genero: Masculino
Tipo: SILVER

Mostrando datos del jugador
Nombre: Adolf
Apellidos: Hitler
Edad: 56
Pais: Bolivia
Genero: Masculino
Tipo: GOLD

Mostrando datos del jugador
Nombre: Alejandra
Apellidos: Maine
Edad: 19
```



13.Promoción para usuarios de Bolivia.



```
36  // En el mes de diciembre a todos los clientes de Bolivia se les dará una promoción
37  // en cuanto a precios en viajes a nivel nacional.
38  // A todos los clientes que sean de nacionalidad boliviana y además el tipo
39  // de cliente GOLD, convertir a estos clientes en VIP
40  // Es decir si es de Bolivia y es GOLD deberá ser ahora un cliente VIP
41  // El método estático dentro de la clase MAIN recibe 3 atributos
42  // La cola de clientes
43  // El tipo de cliente
44  // La nacionalidad del cliente
45  @ 1 usage
46  public static void cambiarTipo(ColaClientes cola, String pais, String tipo,String NewTipo) {
47      ColaClientes aux = new ColaClientes( maxi 10);
48      Cliente Valorextraido = null;
49      while (!cola.esVacia()) {
50          Valorextraido = cola.eliminar();
51          if (Valorextraido.getPais().equals(pais) && Valorextraido.getTipo().equals(tipo)) {
52              Valorextraido.setTipo(NewTipo);
53              aux.adicionar(Valorextraido);
54          }
55          aux.adicionar(Valorextraido);
56      }
57      cola.vaciar(aux);
58      cola.mostrar();
59  }
```

Mostrando datos del jugador
Nombre: Adolf
Apellidos: Hitler
Edad: 56
Pais: Bolivia
Genero: Masculino
Tipo: VIP

Mostrando datos del jugador
Nombre: Alejandra
Apellidos: Mainé

Adolf es el único que era de Bolivia y de tipo GOLD, ahora es VIP

14.Moviendo clientes en la cola.

```
src
├── Campeonato.Clases
├── Ejercicio_1
├── H2DEFENSA
├── Hito3
├── Lab1
├── Lab2
├── Lab3
├── Lab4
├── PilaDeCadenas
├── PilaDeClientes
│   ├── Cliente
│   ├── Main
│   └── PilaCliente
├── PilaDeLibros
├── PilaDeClientes.rar
├── Hito4
├── ColaDeNumeros
│   ├── Main
│   └── Numeros
├── Main.java
├── OperacionesAlgebrai
├── ...
└── Practica_H4.Main

60 //Determinar cuantos clientes son mayores a 60 años
61 1 usage
62 @
63 public static void MoverClientesMayores(ColaClientes cola, int edadMayor) {
64     ColaClientes aux = new ColaClientes( max: 100);
65     ColaClientes aux2 = new ColaClientes( max: 100);
66     Cliente Valorextraido = null;
67     while (!cola.esVacia()) {
68         Valorextraido = cola.eliminar();
69         if (Valorextraido.getEdad() >= edadMayor) {
70             aux2.adicionar(Valorextraido);
71         }
72     }
73     cola.vaciar(aux2);
74     cola.vaciar(aux);
75     cola.mostrar();
76 }
77
78 @
79 public static void Cambiarcolaclientes(ColaClientes colaA, ColaClientes colaB,String Nombre
```

Practica_H4.Main x

"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.2.2\lib\idea.

Datos de la Cola de clientes

Mostrando datos del jugador

Nombre: Josef

Apellidos: Stalin

Edad: 74

Pais: Argentina

Genero: Masculino

Tipo: GOLD

Josef es el único mayor a 60 años, y fue movido al inicio de la cola.



15. Moviendo clientes entre 2 colas.

Saul de cola A fue movido a la cola B

```
public static void CambiarColaClientes(ColaClientes colaA, ColaClientes colaB, String Nombre) {
    int nroElemColaA = colaA.nroElem();
    int nroElemColaB = colaB.nroElem();
    ColaClientes aux = new ColaClientes( max: 100);
    ColaClientes aux2 = new ColaClientes( max: 100);

    Cliente valorEliminado = null;
    for (int i = 1; i <= nroElemColaA; i++) {
        valorEliminado = colaA.eliminar();
        if (valorEliminado.getNombres().equals(Nombre)) {
            colaB.adicionar(valorEliminado);
        } else {
            colaA.adicionar(valorEliminado);
        }
    }

    for (int i = 1; i <= nroElemColaB; i++) {
        valorEliminado = colaB.eliminar();
        if (valorEliminado.getNombres().equals(Nombre)) {
            colaB.adicionar(valorEliminado);
        } else {
            aux2.adicionar(valorEliminado);
        }
    }

    colaA.vaciar(aux);
    colaB.vaciar(aux2);
}
```

Practica_H4.Main

Genero: Masculino
Tipo: GOLD

Datos de la Cola de clientes

Mostrando datos del jugador
Nombre: Saul
Apellidos: Saulero
Edad: 42
País: Rusia
Genero: Masculino
Tipo: VIP

