

# **EVALUACIÓN PROCESUAL HITO 3**

**ESTRUCTURA DE DATOS  
LUIS ALVAREZ MEDINA**



# **MANEJO DE CONCEPTOS**

01

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

Es la forma en que se organiza y almacenan los datos en una computadora para que puedan ser utilizados y procesados de forma eficiente. La estructura de datos bien diseñadas y eficientes puede mejorar significativamente el rendimiento y la escalabilidad de los programas informáticos.

02

## ¿CUÁLES SON LOS TIPOS DE ESTRUCTURA QUE EXISTE?

### ESTRUCTURAS DE DATOS LINEALES

1. Arreglos (Arrays)
2. Listas (Linked lists)
3. Pilas (Stacks)
4. Colas (Queues)

### ESTRUCTURAS DE DATOS NO LINEALES

1. Arboles(Trees)
2. Grafos (Graphs)
3. Tablas hash (Hash tables)
4. Heaps

03

## ¿APOYANDOSE EN EL LINK ADJUNTO, EXPLIQUE, POR QUE SON UTILES LAS ESTRUCTURAS DE DATOS?

Las estructuras de datos son útiles porque nos permiten tener una batería de herramientas para poder solucionar ciertos problema, donde también lo utilizamos para hacer un software mas eficiente optimizando sus recursos.

04

## ¿QUE ES UNA PILA?

Es una estructura de datos lineal, que sigue un política LIFO (Last In, First Out), lo que quiere decir que el primer ítem en ingresar será el ultimo ítem en salir. Esta estructura se lo utiliza comunmente para la planificación y la evaluación de expresiones matemáticas.

05

## ¿QUE ES STACK en JAVA, una STACK será lo mismo que una pila?

STACK en java tiene algunas limitaciones a diferencia de una PILA, esta clase STACK es una subclase de la clase VECTOR, que se considera una estructura de datos obsoleta en Java.

Ahora si son iguales se diría que si ya que son términos iguales los cuales se pueden utilizar de manera intercambiable cuando se habla de la implementación de la estructura de datos.



06

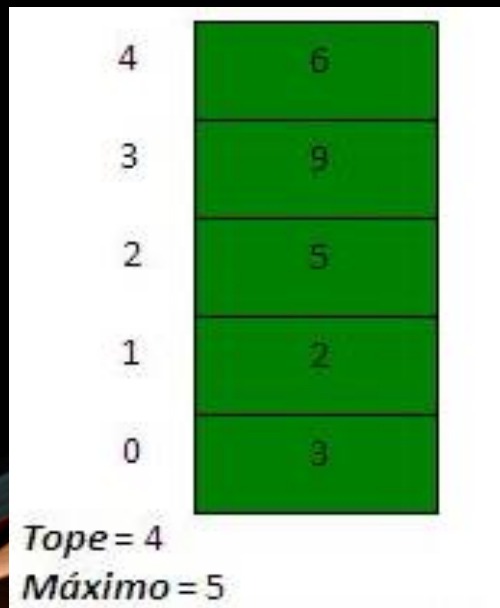
## ¿QUE ES TOPE EN UNA PILA?

Tope es al cantidad de ELEMENTOS o ITEMS que se encuentran dentro del rango de max de la pila ya sea igual o menor a este.

07

## ¿QUE ES MAX EN UNA PILA?

MAX es la cantidad de ELEMENTOS o ITEMS que puede recibir una pila por ejemplo si en un pila el max es 5 no se podrá ingresar mas 5 elementos el tope tendrá que ser menor a 5 o igual a 5 pero nunca mayor a 5.



## ¿A QUE SE REFIERE LOS METODOS **esVACIO()** Y **esLLENA()** EN UNA PILA?

**El método esVacio():** lo utilizamos para verificar si la pila esta vacía o tiene algún ítem dentro.

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

**El método esLlena():** lo utilizamos para verificar si la pila esta llena o si la pila a un puede recibir ítems.

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

09

## ¿QUE SON LOS METODOS ESTATICOS EN JAVA?

Los métodos estáticos, son métodos que pertenecen a una clase en lugar de a una instancia de la clase. Esto significa que se puede llamar a un método estático en cualquier parte del código sin crear un objeto de la clase que lo contiene.

10

## ¿A TRAVÉS DE UN GRAFICO, MUESTRE LOS MÉTODOS MÍNIMOS QUE DEBERÍA TENER UNA PILA?

PilaLibros		
m	PilaLibros ()	
m	esVacio ()	boolean
m	nroElem ()	int
m	mostrar ()	void
m	vaciar (PilaLibros )	void
m	esllena ()	boolean
m	adicionar (Libro)	void
m	eliminar ()	Libro



# **PARTE PRACTICA**



# CREAR LAS CLASES NECESARIAS PARA LA PILA CLIENTES



1. Crear la clase clientes.
2. Crear la clase PilaCliente.
3. Crear la clase main
4. Crear un paquete de nombre PilaDeClientes

```
package PilaDeClientes;
```

```
public class MianClientes {
```

```
    public static void main(String[] args) {
```

```
        Cliente clie1 = new Cliente("Luis", "Alvarez", 19, "direccion1", "Masculino");
        Cliente clie2 = new Cliente("Yamiley", "Lipe", 18, "direccion2", "Femenino");
        Cliente clie3 = new Cliente("Juan", "Mamani", 20, "direccion3", "Masculino");
        Cliente clie4 = new Cliente("Carolina", "Paucara", 22, "direccion4", "Femenino");
        Cliente clie5 = new Cliente("Eddy", "Apaza", 25, "direccion5", "Masculino");
```

```
        PilaCliente clientes = new PilaCliente();
        clientes.adicionarClientes(clie1);
        clientes.adicionarClientes(clie2);
        clientes.adicionarClientes(clie3);
        clientes.adicionarClientes(clie4);
        clientes.adicionarClientes(clie5);
```

```
    }
}
```

Mostrar a los Clientes

DATOS DEL CLIENTE

Nombre: Eddy  
Apellido: Apaza  
Edad: 25  
Direccion: direccion5  
Genero: Masculino

DATOS DEL CLIENTE

Nombre: Carolina  
Apellido: Paucara  
Edad: 22  
Direccion: direccion4  
Genero: Femenino

DATOS DEL CLIENTE

Nombre: Juan  
Apellido: Mamani  
Edad: 20  
Direccion: direccion3  
Genero: Masculino

DATOS DEL CLIENTE

Nombre: Yamiley  
Apellido: Lipe  
Edad: 18  
Direccion: direccion2  
Genero: Femenino

DATOS DEL CLIENTE

Nombre: Luis  
Apellido: Alvarez  
Edad: 19  
Direccion: direccion1  
Genero: Masculino

## DETERMINAR CUANTOS CLIENTES SON MAYORES DE 20 AÑOS

1. El método deberá llamarse `mayoresCiertaEdad(Pila, edadMayor)`

2. El método debe ser creado en la clase MAIN como un método estático.

3. El método recibe 2 parámetros

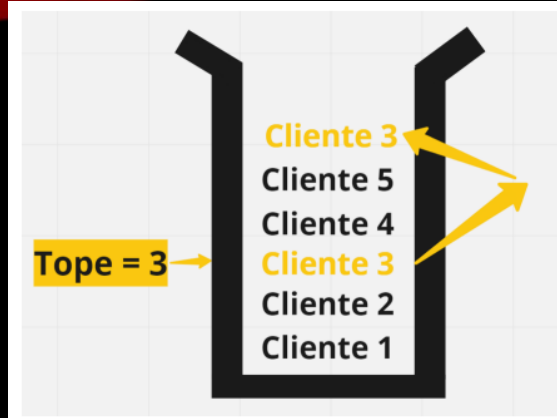
- La Pila de Clientes
- El valor de la edad

```
public static void mayoresCiertaEdad(PilaCliente pila, int edadMayor){  
  
    PilaCliente aux = new PilaCliente();  
    Cliente clienteeliminado = null;  
    int cont = 0;  
    while(!pila.esVacio()){  
  
        clienteeliminado = pila.eliminarCliente();  
  
        if(clienteeliminado.getEdad() > edadMayor){  
            cont = cont+1;  
        }  
  
        aux.adicionarClientes(clienteeliminado);  
    }  
    pila.vaciar(aux);  
    System.out.println("hay "+ cont + " clientes con una edad mayor a "+ edadMayor);  
}
```

```
hay 2 clientes con una edad mayor a 20
```

13

# MOVER EL KESIMO ELEMENTO DE LA PILA



1. El método deberá llamarse **KEsimoPosicion(Pila,valorTope)**
2. El método debe ser creado en la clase MAIN como un método estático.
3. El método recibe 2 parámetros
  - La Pila de Clientes
  - El valor(int) de la posición que moverá al final de la pila

```
public static void kEsimoposicion(PilaCliente pila, int valorTope){

    PilaCliente aux = new PilaCliente();
    Cliente clienteeliminado = null;
    Cliente almacli = null;
    while(!pila.esVacio()){

        clienteeliminado = pila.eliminarCliente();

        if(pila.nroClientes() +1 == valorTope){
            almacli = clienteeliminado;
        }
        else{
            aux.adicionarClientes(clienteeliminado);
        }
    }
    pila.vaciar(aux);
    pila.adicionarClientes(almacli);

}
```

ANTES

Mostrar a los Clientes

DATOS DEL CLIENTE  
 Nombre: Eddy  
 Apellido: Apaza  
 Edad: 25  
 Direccion: direccion5  
 Genero: Masculino

DATOS DEL CLIENTE  
 Nombre: Carolina  
 Apellido: Paucara  
 Edad: 22  
 Direccion: direccion4  
 Genero: Femenino

DATOS DEL CLIENTE  
 Nombre: Juan  
 Apellido: Mamani  
 Edad: 20  
 Direccion: direccion3  
 Genero: Masculino

DATOS DEL CLIENTE  
 Nombre: Yamiley  
 Apellido: Lipe  
 Edad: 18  
 Direccion: direccion2  
 Genero: Femenino

DATOS DEL CLIENTE  
 Nombre: Luis  
 Apellido: Alvarez  
 Edad: 19  
 Direccion: direccion1  
 Genero: Masculino

DESPUES

Mostrar a los Clientes

DATOS DEL CLIENTE  
 Nombre: Juan  
 Apellido: Mamani  
 Edad: 20  
 Direccion: direccion3  
 Genero: Masculino

DATOS DEL CLIENTE  
 Nombre: Eddy  
 Apellido: Apaza  
 Edad: 25  
 Direccion: direccion5  
 Genero: Masculino

DATOS DEL CLIENTE  
 Nombre: Carolina  
 Apellido: Paucara  
 Edad: 22  
 Direccion: direccion4  
 Genero: Femenino

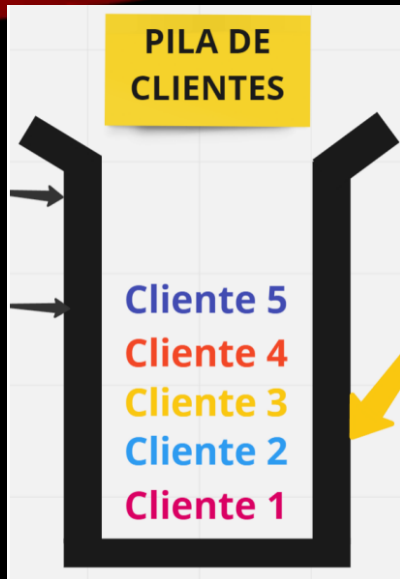
DATOS DEL CLIENTE  
 Nombre: Yamiley  
 Apellido: Lipe  
 Edad: 18  
 Direccion: direccion2  
 Genero: Femenino

DATOS DEL CLIENTE  
 Nombre: Luis  
 Apellido: Alvarez  
 Edad: 19  
 Direccion: direccion1  
 Genero: Masculino



14

# CAMBIAR LA DIRECCION DE ALGUNOS CLIENTES DE LA PILA.



```
public static void asignaDireccion(PilaCliente pila, String nuevaDireccion){

    PilaCliente aux = new PilaCliente();
    Cliente clienteeliminado = null;
    while(!pila.esVacio()){

        clienteeliminado = pila.eliminarCliente();

        if(clienteeliminado.getGenero().equals("Femenino")){
            clienteeliminado.setDireccion(nuevaDireccion);
        }
        aux.adicionarClientes(clienteeliminado);
    }
    pila.vaciar(aux);
}
```

ANTES

Mostrar a los Clientes

DATOS DEL CLIENTE  
Nombre: Eddy  
Apellido: Apaza  
Edad: 25  
Direccion: direccion5  
Genero: Masculino

DATOS DEL CLIENTE  
Nombre: Carolina  
Apellido: Paucara  
Edad: 22  
Direccion: direccion4  
Genero: Femenino

DATOS DEL CLIENTE  
Nombre: Juan  
Apellido: Mamani  
Edad: 20  
Direccion: direccion3  
Genero: Masculino

DATOS DEL CLIENTE  
Nombre: Yamiley  
Apellido: Lipe  
Edad: 18  
Direccion: direccion2  
Genero: Femenino

DATOS DEL CLIENTE  
Nombre: Luis  
Apellido: Alvarez  
Edad: 19  
Direccion: direccion1  
Genero: Masculino

DESPUES

Mostrar a los Clientes

DATOS DEL CLIENTE  
Nombre: Eddy  
Apellido: Apaza  
Edad: 25  
Direccion: direccion5  
Genero: Masculino

DATOS DEL CLIENTE  
Nombre: Carolina  
Apellido: Paucara  
Edad: 22  
Direccion: extranca  
Genero: Femenino

DATOS DEL CLIENTE  
Nombre: Juan  
Apellido: Mamani  
Edad: 20  
Direccion: direccion3  
Genero: Masculino

DATOS DEL CLIENTE  
Nombre: Yamiley  
Apellido: Lipe  
Edad: 18  
Direccion: extranca  
Genero: Femenino

DATOS DEL CLIENTE  
Nombre: Luis  
Apellido: Alvarez  
Edad: 19  
Direccion: direccion1  
Genero: Masculino

1. El método deberá llamarse **asignaDireccion(Pila,nuevaDireccion)**

2. El método debe ser creado en la clase MAIN como un método estático.

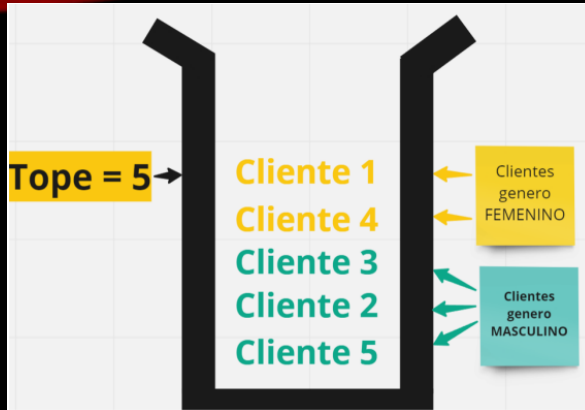
3. El método recibe 2 parámetros

- La Pila de Clientes
- El valor(String) de la nueva dirección.

5. Cambiar la dirección del cliente siempre y cuando el genero sea **FEMENINO**



# MOVER ITEMS DE LA PILA



1. El método deberá llamarse **reordenaPila(Pila)**
2. El método debe ser creado en la clase **MAIN** como un método estático.
3. El método recibe 1 parámetros
  - La Pila de Clientes
4. Mover a la base todos los clientes del genero masculino y los genero femenino moverlos al final.

```
public static void reordenarPila (PilaCliente pila){

    PilaCliente aux = new PilaCliente();
    Cliente clienteeliminado = null;
    PilaCliente almacenador = new PilaCliente();
    while(!pila.esVacio()){

        clienteeliminado = pila.eliminarCliente();

        if(clienteeliminado.getGenero().equals("Femenino")){

            almacenador.adicionarClientes(clienteeliminado);
        }
        else {
            aux.adicionarClientes(clienteeliminado);
        }
    }
    pila.vaciar(aux);
    pila.vaciar(almacenador);
}
```

## ANTES

Mostrar a los Clientes

DATOS DEL CLIENTE  
Nombre: Eddy  
Apellido: Apaza  
Edad: 25  
Direccion: direccion5  
Genero: Masculino

DATOS DEL CLIENTE  
Nombre: Carolina  
Apellido: Paucara  
Edad: 22  
Direccion: direccion4  
Genero: Femenino

DATOS DEL CLIENTE  
Nombre: Juan  
Apellido: Mamani  
Edad: 20  
Direccion: direccion3  
Genero: Masculino

DATOS DEL CLIENTE  
Nombre: Yamiley  
Apellido: Lipe  
Edad: 18  
Direccion: direccion2  
Genero: Femenino

DATOS DEL CLIENTE  
Nombre: Luis  
Apellido: Alvarez  
Edad: 19  
Direccion: direccion1  
Genero: Masculino

## DESPUES

Mostrar a los Clientes

DATOS DEL CLIENTE  
Nombre: Carolina  
Apellido: Paucara  
Edad: 22  
Direccion: direccion4  
Genero: Femenino

DATOS DEL CLIENTE  
Nombre: Yamiley  
Apellido: Lipe  
Edad: 18  
Direccion: direccion2  
Genero: Femenino

DATOS DEL CLIENTE  
Nombre: Eddy  
Apellido: Apaza  
Edad: 25  
Direccion: direccion5  
Genero: Masculino

DATOS DEL CLIENTE  
Nombre: Juan  
Apellido: Mamani  
Edad: 20  
Direccion: direccion3  
Genero: Masculino

DATOS DEL CLIENTE  
Nombre: Luis  
Apellido: Alvarez  
Edad: 19  
Direccion: direccion1  
Genero: Masculino