

ESTRUCTURA DE DATOS

ESTUDIANTE:

Andrés Vladimir Quiroga Huariste SIS9208422

DOCENTE:

Ing. William Roddy Barra Paredes

MATERIA:

Estructura de Datos

GESTIÓN:

2022

MANEJO DE CONCEPTOS

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

R. Las estructuras de datos es una rama de las ciencias de la computación que estudia y aplica diferentes formas de organizar información dentro de una aplicación, para manipular, buscar e insertar estos datos de manera eficiente.

• ¿Cuáles son los TIPOS DE ESTRUCTURA QUE EXISTE?

R. Estructura Secuencial.

- Estructuras de Repetición.
- Estructuras Condicionales.

¿Apoyándose en el link adjunto, explique, por qué son útiles las estructuras de datos?.

R. Las estructuras de datos son útiles porque nos permiten tener una batería de herramientas para solucionar ciertos tipos de problemas.

► ¿Qué es una PILA?

R. Una pila es una estructura lineal en la que los elementos pueden ser añadidos o eliminados solo por el final

¿Qué es STACK en JAVA, una STACK será lo mismo que una PILA?

R. Una pila (stack) es un objeto similar a una pila de platos, donde se puede agregar y sacar datos sólo por el extremo superior.

¿Qué es TOPE en una PILA?

R. Números de elementos que tiene una pila

¿Qué es MAX en una PILA?

R. Es el numero máximo de elementos que soporta la pila

¿A que se refiere los métodos esVacia() y esLlena() en una PILA?

R.

► ¿Qué son los métodos estáticos en JAVA?

R. El modificador static en Java es una de las palabras reservadas en este lenguaje que le da a quien programa, la posibilidad de establecer cómo será el acceso a determinados componentes del código que se está desarrollando.

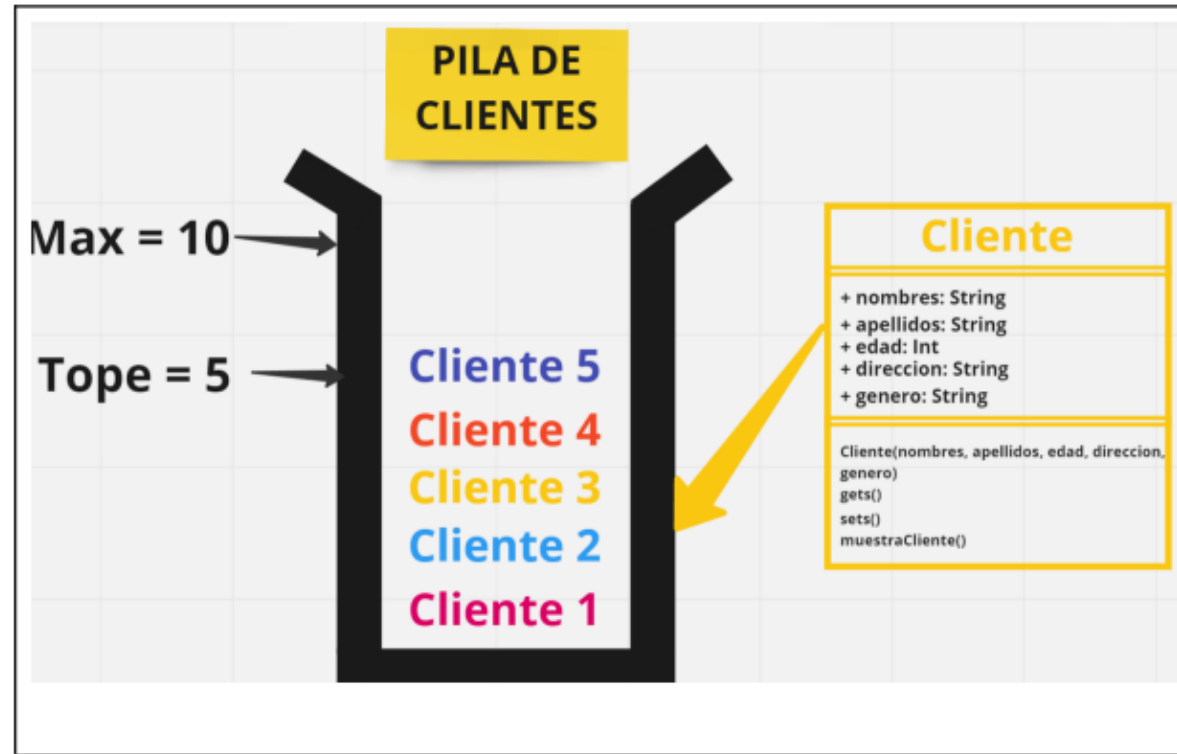
¿A través de un gráfico, muestre los métodos mínimos que debería de tener una PILA?

R.

- Añadir
- Eliminar
- Vaciar

Parte practica

11. Crear las clases necesarias para la **PILA DE CLIENTES**.



- Crear la clase **Cliente**
- Crear la clase **PilaCliente**
- Crear la clase **Main**.
- Crear un **paquete** de nombre **PilaDeClientes** (todas las clases deberán de estar dentro de este paquete)
- **Adjuntar los siguientes.**
 - La clase **MAIN** con la creación de **5 clientes y agregados a la PILA**.
 - Una imagen de la salida de la consola en donde se muestran **todos los items de la pila**.

```
package PilaClientes;

public class Cliente {
    private String nombres;
    private String apellidos;
    private int edad;
    private String direccion;
    private String genero;

    public Cliente(String nombres , String apellidos , int edad , String direccion , String genero){
        this.nombres = nombres;
        this.apellidos = apellidos;
        this.edad = edad;
        this.direccion = direccion;
        this.genero = genero;
    }

    public String getNombres() {
        return nombres;
    }

    public String getApellidos() {
        return apellidos;
    }

    public int getEdad() {
        return edad;
    }

    public String getDireccion() {
        return direccion;
    }

    public String getGenero() {
        return genero;
    }
}
```

```
public void setNombres(String nombres) {
    this.nombres = nombres;
}

public void setApellidos(String apellidos) {
    this.apellidos = apellidos;
}

public void setEdad(int edad) {
    this.edad = edad;
}

public void setDireccion(String direccion) {
    this.direccion = direccion;
}

public void setGenero(String genero) {
    this.genero = genero;
}

public void mostrarCliente(){
    System.out.println("\n Mostrando Clientes"); // ALT + 92
    System.out.println(" Nombres: " + this.getNombres());
    System.out.println(" Apellidos: " + this.getApellidos());
    System.out.println(" Edad: " + this.getEdad());
    System.out.println(" Direccion: " + this.getDireccion());
    System.out.println(" Genero: " + this.getGenero());
}
}
```

```

package PilaClientes;

import PiladeLibros.Libro;
import PiladeLibros.PiladeLibros;

public class PilaCliente {
    private int max;
    private int tope;
    private Cliente[] clientes;

    public PilaCliente(int max){
        this.tope = 0;
        this.max = max;
        clientes = new Cliente[this.max + 1];
    }

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

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

    public int nroElem(){
        return this.tope;
    }
}

```

```

public void adicionar(Cliente nuevocliente){
    if (this.esLlenar() == false){
        this.tope = this.tope + 1;
        this.clientes [this.tope] = nuevocliente;
    } else {
        System.out.println("La Pila de Clientes esta llena.");
    }
}

public Cliente eliminar(){
    Cliente elementoEliminado = null;

    if (this.esVacio() == false){
        elementoEliminado = this.clientes[this.tope];
        this.tope = this.tope - 1;
    } else {
        System.out.println("La Pila de Clientes esta vacia.");
    }
    return elementoEliminado;
}

public void llenar(){

}

public void mostrarPilaCliente(){
    Cliente elem = null;
    if (esVacio()){
        System.out.println("Pila Vacía, no hay Item que mostrar");
    }
    else
    {
        System.out.println("\n\n Mostrando la Pila ");
        PilaCliente aux = new PilaCliente(this.max);
        while (!esVacio())
        {
            elem = this.eliminar();
            aux.adicionar(elem);
            elem.mostrarCliente();
        }
        vaciar(aux);
    }
}

public void vaciar(PilaCliente aux){
    while (!aux.esVacio())
        adicionar(aux.eliminar());
}
}

```

```
package PilaClientes;
```

```
public class Main {
```

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

```
        Cliente client1 = new Cliente("Andres", "Quiroga", 23, "Villa Tunari", "Masculino");
```

```
        Cliente client2 = new Cliente("Vladimir", "Putin", 53, "Moscu", "Masculino");
```

```
        Cliente client3 = new Cliente("Tony", "Stark", 60, "New York", "Masculino");
```

```
        Cliente client4 = new Cliente("Sindy", "Mendoza", 29, "Villa Fatima", "Femenino");
```

```
        Cliente client5 = new Cliente("Sasha", "Greg", 28, "Villa Carmen", "Femenino");
```

```
        PilaCliente pila = new PilaCliente(10);
```

```
        pila.adicionar(client1);
```

```
        pila.adicionar(client2);
```

```
        pila.adicionar(client3);
```

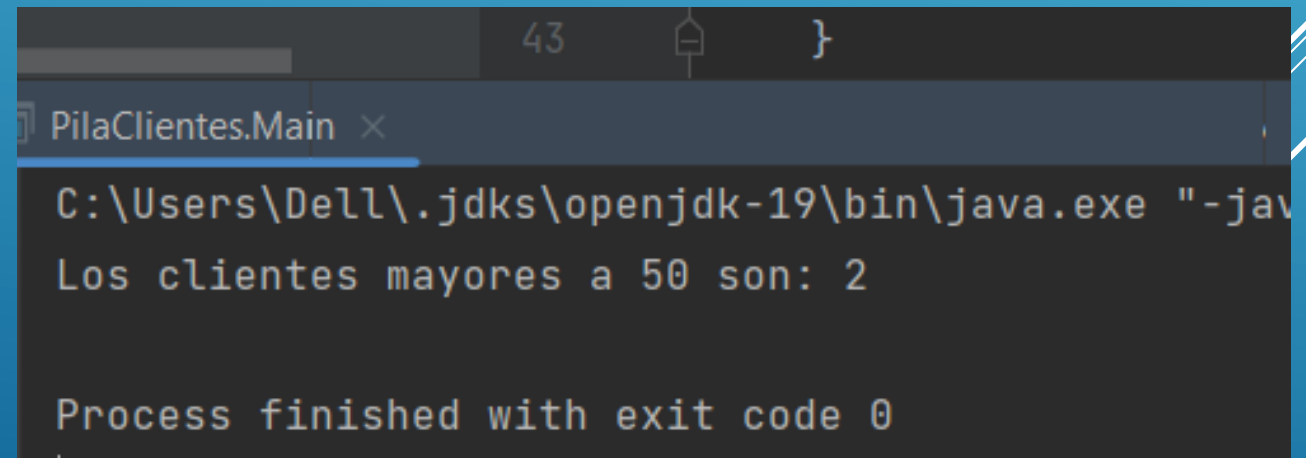
```
        pila.adicionar(client4);
```

```
        pila.adicionar(client5);
```


12. Determinar cuántos **CLIENTES** son mayores de 20 años.

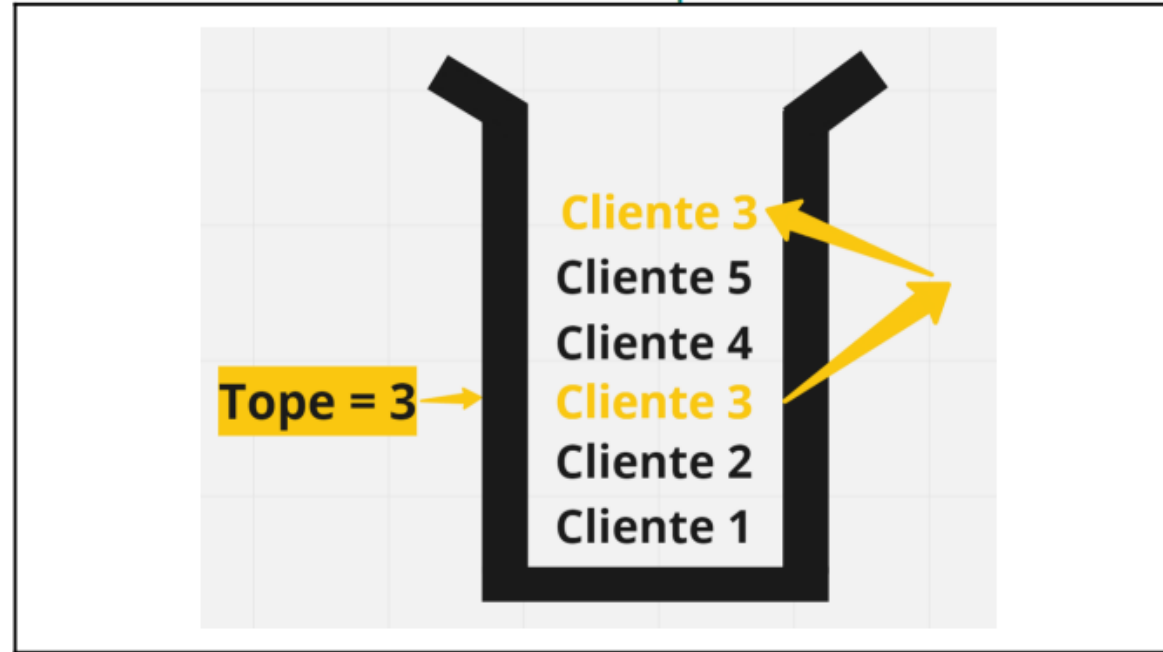
- El método deberá llamarse **mayoresCiertaEdad(Pila, edadMayor)**
- El método debe ser creado en la clase **MAIN** como un método estático.
- El método recibe 2 parámetros
 - La Pila de Clientes
 - El valor de la edad.
- Adjuntar los siguientes
 - El **código** del método que resuelve el problema.
 - Una **imagen** de la salida de la consola.

```
public static void MayoresCiertaEdad(PilaCliente pila, int edadMayor) {  
    PilaCliente aux = new PilaCliente(10);  
    Cliente valorextraido = null;  
    int mayoresedad = 0;  
  
    while (pila.esVacio() == false) {  
        valorextraido = pila.eliminar();  
  
        if (valorextraido.getEdad() > edadMayor) {  
            mayoresedad = mayoresedad + 1;  
        }  
        aux.adicionar(valorextraido);  
    }  
    System.out.println("Los clientes mayores a " + edadMayor + " son: " + mayoresedad);  
}
```



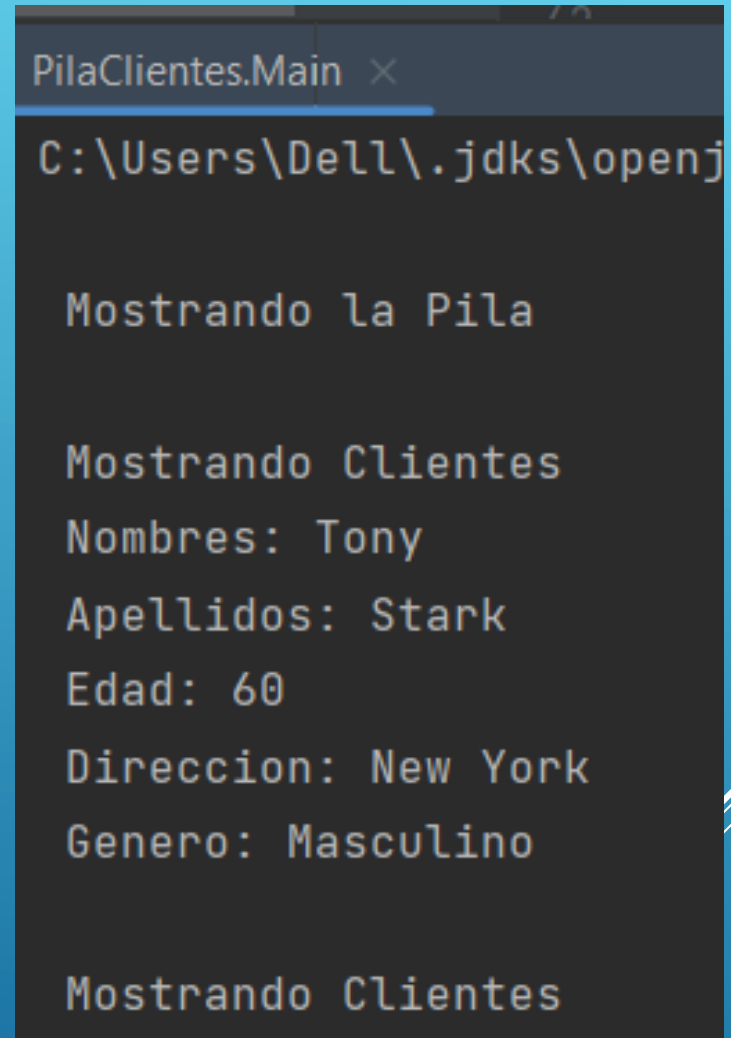
```
43      }  
PilaClientes.Main x  
C:\Users\Dell\.jdk\openjdk-19\bin\java.exe "-jav  
Los clientes mayores a 50 son: 2  
  
Process finished with exit code 0
```

13. Mover el **k-ésimo** elemento al final de la pila.



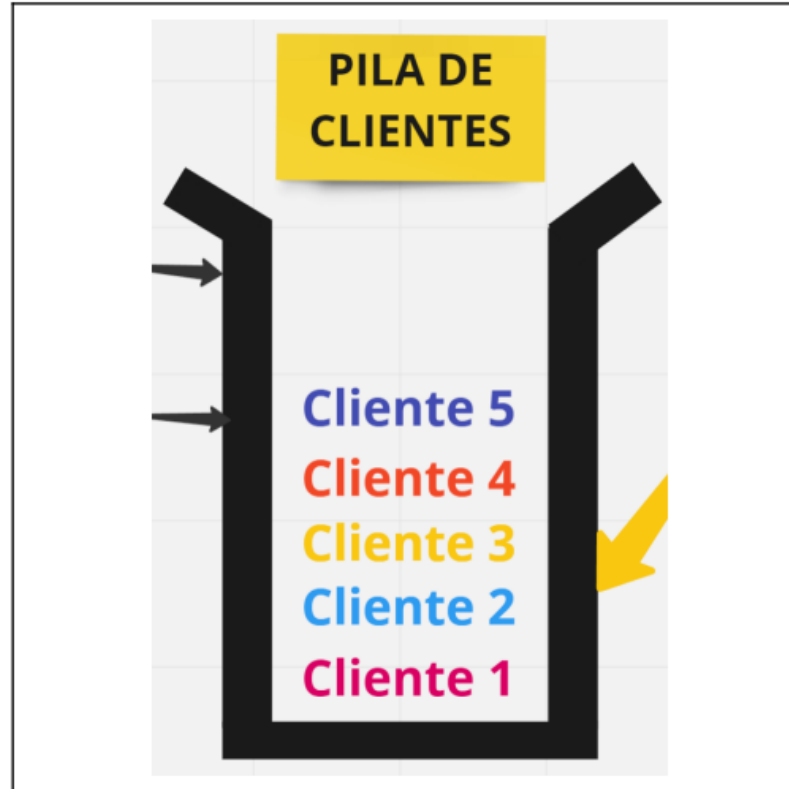
- El método deberá llamarse **kEsimoPosicion(Pila, valorTope)**
- El método debe ser creado en la clase **MAIN** como un método estático.
- El método recibe 2 parámetros
 - La Pila de Clientes
 - El valor(int) de la posición que moverá al final de la pila.
- Adjuntar los siguientes
 - El **código** del método que resuelve el problema.
 - Una **imagen** de la salida de la consola.

```
public static void kesimoPosicion(PilaCliente pila, int valorTope) {  
    PilaCliente aux = new PilaCliente(10);  
    Cliente valor = null;  
  
    while (pila.esVacio() == false) {  
        if (pila.nroElem() != valorTope) {  
            aux.adicionar(pila.eliminar());  
        } else {  
  
            valor = pila.eliminar();  
        }  
    }  
    pila.vaciar(aux);  
    pila.adicionar(valor);  
    pila.mostrarPilaCliente();  
}
```



```
PilaClientes.Main x  
C:\Users\Dell\.jdk\openj  
  
Mostrando la Pila  
  
Mostrando Clientes  
Nombres: Tony  
Apellidos: Stark  
Edad: 60  
Direccion: New York  
Genero: Masculino  
  
Mostrando Clientes
```

14. Cambiar la dirección de algunos **CLIENTES** de la **PILA**.



- El método deberá llamarse **asignaDireccion(Pila, nuevaDireccion)**
- El método debe ser creado en la clase **MAIN** como un método estático.
- El método recibe 2 parámetros
 - La Pila de Clientes
 - El valor(String) de la nueva dirección.
- **Cambiar la dirección del cliente** siempre y cuando el género sea **FEMENINO**.
- Adjuntar los siguientes
 - El **código** del método que resuelve el problema.
 - Una **imagen** de la salida de la consola.

```
public static void asignaDIRECCION(PilaCliente pila , String nuevaDIRECCION){
    PilaCliente aux = new PilaCliente(10);
    Cliente valor = null;

    while (!pila.esVacio()){
        valor = pila.eliminar();

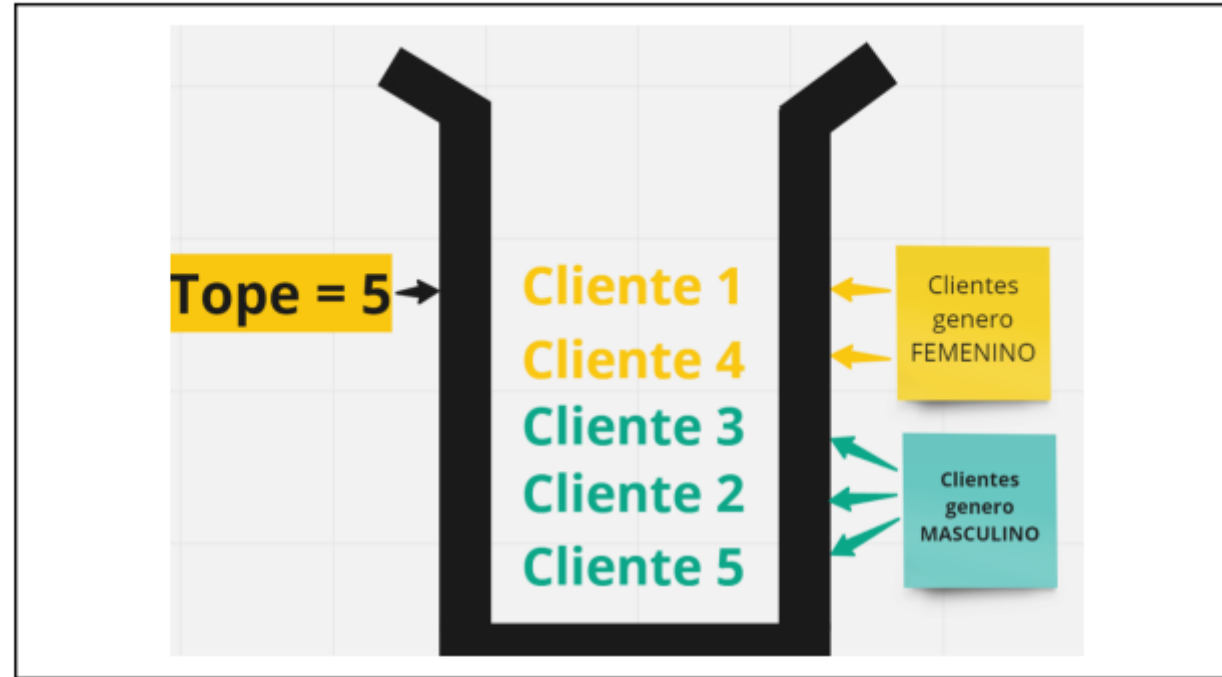
        if (valor.getGenero() == "Femenino"){
            valor.setDIRECCION(nuevaDIRECCION);
            aux.adicionar(valor);
        } else {

            aux.adicionar(valor);
        }
    }
    pila.vaciar(aux);
    pila.adicionar(valor);
    pila.mostrarPilaCliente();
}
```

```
Mostrando Clientes
Nombres: Sasha
Apellidos: Greg
Edad: 28
DIRECCION: Rusia
Genero: Femenino
```

```
Mostrando Clientes
Nombres: Sindy
Apellidos: Mendoza
Edad: 29
DIRECCION: Rusia
Genero: Femenino
```

15.Mover ÍTEMS de la PILA.



- El método deberá llamarse **reordenaPila(Pila)**
- El método debe ser creado en la clase **MAIN** como un método estático.
- El método recibe 1 parámetro
 - La Pila de Clientes
- Mover a la base todos los **clientes del género masculino** y los del género **femenino moverlos al final**.
- Adjuntar los siguientes
 - El **código** del método que resuelve el problema.
 - Una **imagen** de la salida de la consola.

```
public static void reordenaPila(PilaCliente pila){
    PilaCliente aux1 = new PilaCliente(10);
    PilaCliente aux2 = new PilaCliente(10);
    Cliente valor = null;

    while (!pila.esVacio()){
        valor = pila.eliminar();
        if (valor.getGenero() != "Femenino"){
            aux1.adicionar(valor);
        } else {

            aux2.adicionar(valor);
        }
    }
    pila.vaciar(aux1);
    pila.vaciar(aux2);
    pila.mostrarPilaCliente();
}
```

```
Mostrando Clientes
Nombres: Sasha
Apellidos: Greg
Edad: 28
Direccion: Villa Carmen
Genero: Femenino
```

```
Mostrando Clientes
Nombres: Sindy
Apellidos: Mendoza
Edad: 29
Direccion: Villa Fatima
Genero: Femenino
```

```
Mostrando Clientes
Nombres: Tony
Apellidos: Stark
Edad: 60
Direccion: New York
Genero: Masculino
```

```
Mostrando Clientes
```