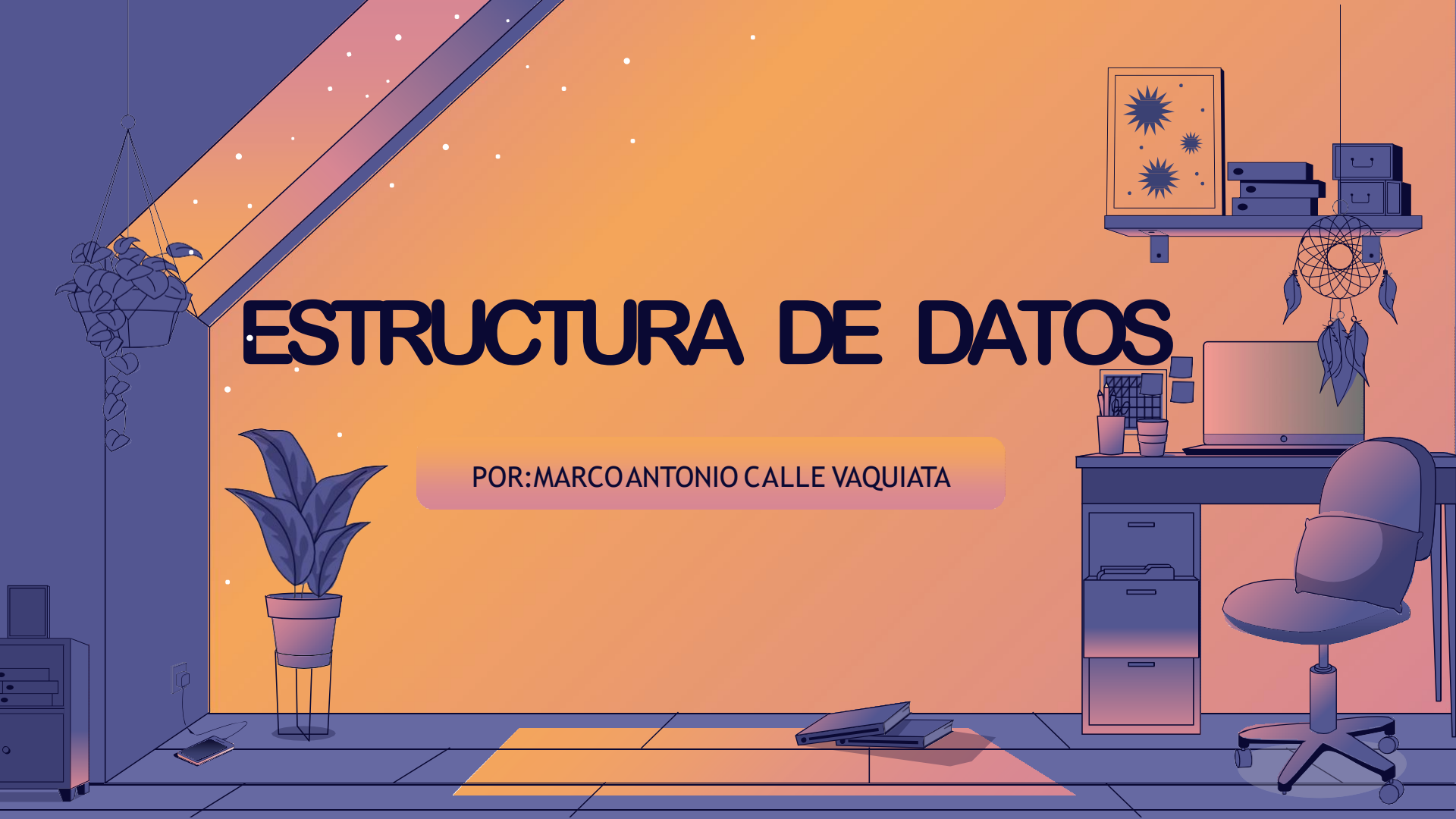

ESTRUCTURA DE DATOS
POR:MARCO ANTONIO CALLE VAQUIATA

OBJETIVOS

MANEJO DE CONCEPTOS

Se explicara conceptos elementales sobre El lenguaje procesual en Estructura de datos

PARTE PRACTICA

Se presentara la aplicación de la parte teórica para la resolución de requerimientos.

INTRODUCCION

La representación de la información es fundamental en ciencias de la computación y en informática.
El propósito principal de la mayoría de los programas de computadoras es almacenar y recuperar información, además de realizar cálculos.
De modo práctico, los requisitos de almacenamiento y tiempo de ejecución exigen que tales programas deban organizar su información de un modo que soporte procesamiento eficiente. Por estas razones, el estudio de estructuras de datos y de los algoritmos que las manipulan constituye el núcleo central de la informática y de la computación.

Manejo de Conceptos

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

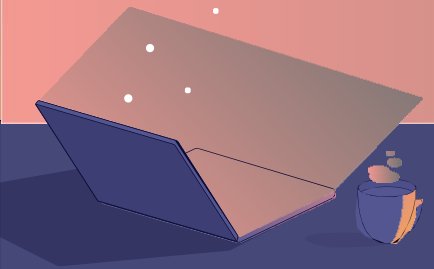
En el ámbito de la informática, las estructuras de datos son aquellas que nos permiten, como desarrolladores, organizar la información de manera eficiente, y en definitiva diseñar la solución correcta para un determinado problema

¿Cuáles son los TIPOS DE ESTRUCTURA QUE EXISTE?

Hay tres tipos de estructuras de datos lineales: Listas enlazadas. Pilas. Co las

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

Las estructuras de datos son una forma de organizar los datos en la computadora, de tal manera que nos permita realizar unas operaciones con ellas de forma muy eficiente.



¿Qué es una PILA?

Una pila (stack) es una colecciona ordenada de elementos en la cual los datos se insertan o se retiran por el mismo extremo llamado "parte superior" de la pila.

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

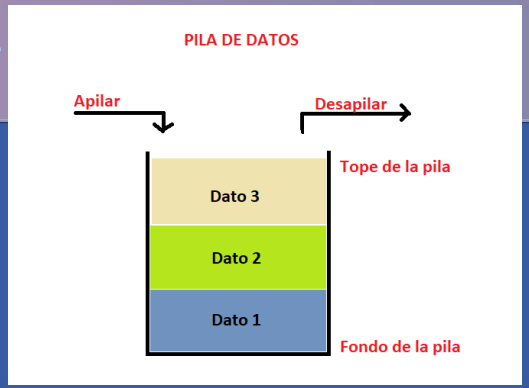
Una pila (stack) es un objeto similar a una pila de platos, donde se puede agregar y sacar datos sólo por el extremo superior. En computación esta estructura es utilizada ampliamente, aunque muchas veces los usuarios ni siquiera se percaten.

¿Qué es TOPE en una PILA?

Una pila se define formalmente como una **colección de datos a los cuales se puede acceder mediante un extremo**, que se conoce generalmente como tope.

¿Qué es MAX en una PILA?

variable auxiliar que se denomina TOPE. Está variable se utiliza para indicar el último elemento que se insertó en la pila.



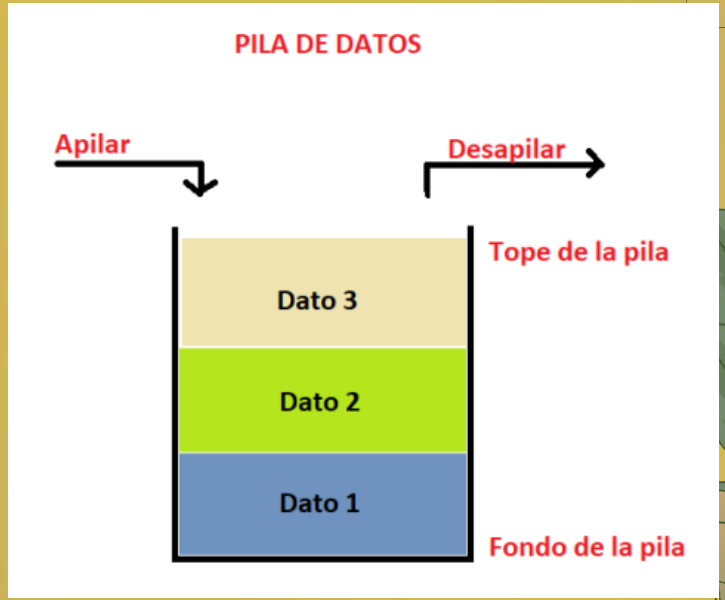
¿A que se refiere los métodos esVacia() y esLLena() en una PILA?.

- apilar (valor): también conocido como push agrega el valor al tope de la pila.
- retirar (): también conocido como pop retira el último elemento apilado.
- cima (): devuelve el valor del elemento que está en la cima de la pila.
- esVacia (): retorna true si la pila no ha sido inicializada.
- buscar (valor): retorna la true si el elemento a buscar existe en la pila.
- eliminar(): elimina la pila
- listar (): imprime en pantalla los elementos de la pila.

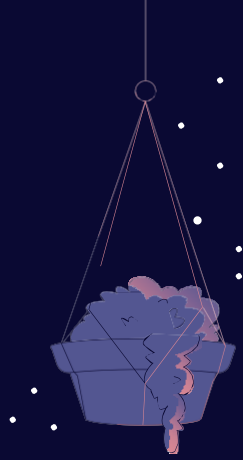
¿Qué son los métodos estáticos en JAVA?

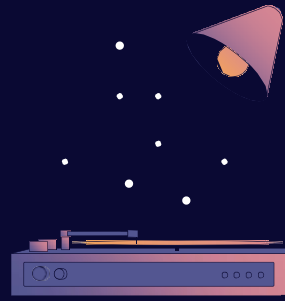
Un método estático es un **método que tiene sentido invocarla sin crear previamente ningun objeto**

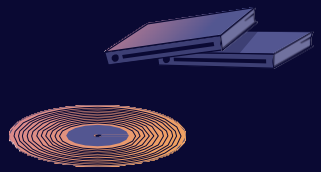
¿A través de un gráfico, muestre los métodos mínimos que debería de tener una PILA?



PARTE PRACTICA

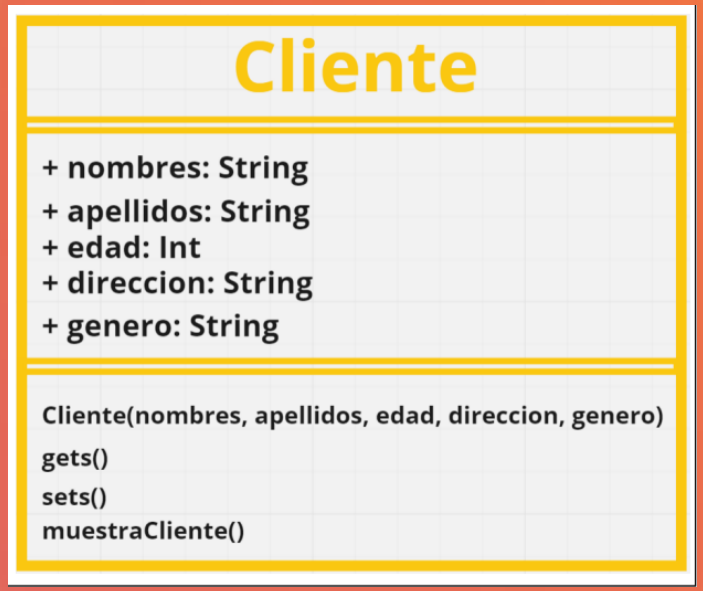






Consigna:

Crear un sistema para gestionar una pila de CLIENTES. Es decir el sistema debe poder gestionar objetos de tipo cliente en la PILA (PILA DE OBJETOS).



SOLUCION:

```java
package HITO3_EVALUACION_2022;

public class Cliente {
    3 usages
    private String Nombres;
    3 usages
    private String Apellidos;
    3 usages
    private int Edad;
    3 usages
    private String Direccion;
    3 usages
    private String Genero;

    5 usages
    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;

    }
```

```java
1 usage
public String getNombres() {
    return Nombres;
}

public void setNombres(String nombres) {
    Nombres = nombres;
}

1 usage
public String getApellidos() {
    return Apellidos;
}

public void setApellidos(String apellidos) {
    Apellidos = apellidos;
}

2 usages
public int getEdad() {
    return Edad;
}
```

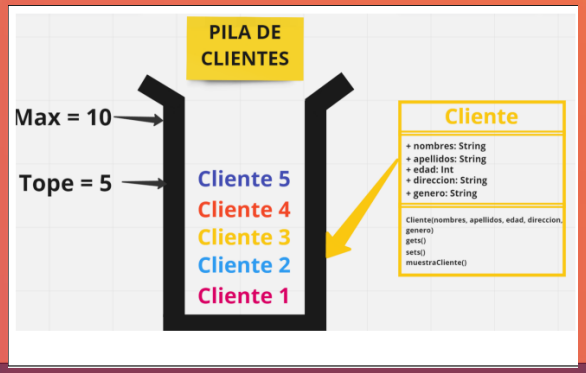
SOLUCION:

```java
public void setEdad(int edad) {
    Edad = edad;
}

1 usage
public String getDireccion() {
    return Direccion;
}

1 usage
public void setDireccion(String direccion) {
    Direccion = direccion;
}

3 usages
public String getGenero() {
    return Genero;
}

public void setGenero(String genero) {
    Genero = genero;
}

1 usage
public void mostrarCliente() {
```

```java
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("Direccion: " + this.getDireccion());
    System.out.println("Genero: " + this.getGenero());

    System.out.println("\n");
}
}
```

1.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 ítems de la pila**.



SOLUCION:

```java
package HIT03_EVALUACION_2022;

public class PilaCliente {

    4 usages
    private int max;
    10 usages
    private int tope;
    3 usages
    private Cliente[] Clientes;


    7 usages
    public PilaCliente(int max) {
        this.tope = 0;
        this.max = max;
        this.Clientes = new Cliente[this.max + 1];
    }
    8 usages
    public boolean esVacio () {
        if (tope == 0) {
            return true;
        } else {
            return false;
        }
```

```java
}
1 usage
public boolean esLleno () {
    if (tope == max) {
        return true;
    } else {
        return false;
    }


}
1 usage
public int nroElem () { return this.tope; }
15 usages
public void adicionar (Cliente nuevoCliente) {
    if (this.esLleno() == false) {
        this.tope = this.tope + 1;
        this.Clientes[this.tope] = nuevoCliente;
    } else {
        System.out.println("La pila de numeros está llena");
    }
}
```

SOLUCION:

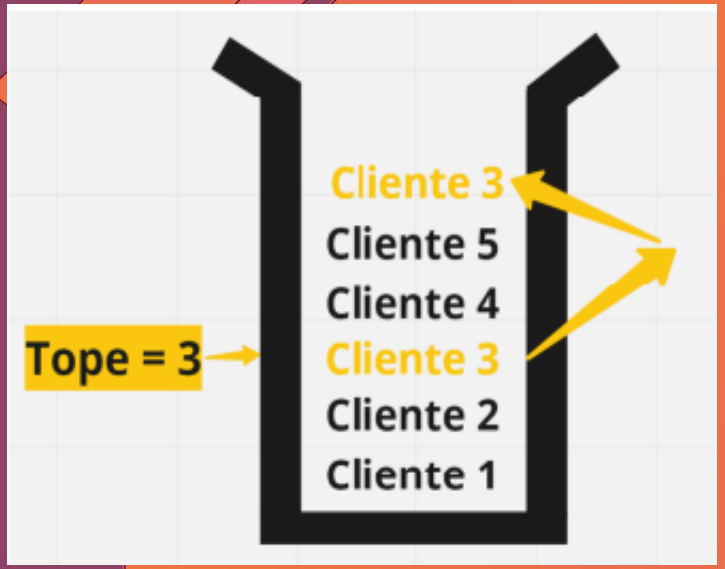
```
public Cliente eliminar () {
    Cliente elementoEliminado = null;

    if (!this.esVacio()) {
        elementoEliminado = (this.Clientes[this.tope]);
        this.tope = this.tope - 1;
    } else {
        System.out.println("La pila de libros está vacia");
    }
    return elementoEliminado;
}
public void llenar () {


}
3 usages
public void mostrar () {
    Cliente elem = null;
    if (esVacio())
        System.out.println("Pila Vacia");
    else {
        System.out.println("\nDatos de la Pila de clientes");
        PilaCliente aux = new PilaCliente(this.max);
        while (!esVacio()) {
```

```
            System.out.println("Pila Vacia");
        else {
            System.out.println("\nDatos de la Pila de clientes");
            PilaCliente aux = new PilaCliente(this.max);
            while (!esVacio()) {
                elem = this.eliminar();
                aux.adicionar (elem);
                elem.mostrarCliente();
            }
            vaciar(aux);
        }

    }
    5 usages
    public void vaciar (PilaCliente pila) {
        while (!pila.esVacio())
            adicionar(pila.eliminar());
    }
}
```

Determinar cuántos CLIENTES son mayores de 20 años

```java
public static void mayoresCiertasEdad(PilaCliente pila, int edadMayor) {
    PilaCliente aux = new PilaCliente( max: 10);
    int MayoresXEdad = 0;
    Cliente Valorextraido = null;
    while (pila.esVacio() == false) {
        Valorextraido = pila.eliminar();
        if (Valorextraido.getEdad() > edadMayor) {
            MayoresXEdad = MayoresXEdad + 1;
        }
        aux.adicionar(Valorextraido);
    }
    System.out.println("\nLa cantidad de clientes con mas de " + edadMayor +" son: " + MayoresXEdad);

}
```

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.

Mover el k-ésimo elemento al final de la pila

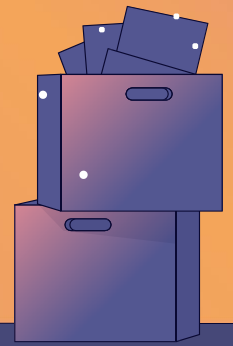
```java
public static void kEsimoPosicion(PilaCliente pila, int valorTope)  {
    PilaCliente aux = new PilaCliente( max: 10);
    Cliente valor = null;

    while (pila.esVacio() == false) {
        if (pila.nroElem() !=  valorTope) {
            aux.adicionar(pila.eliminar());
        } else {
            valor = pila.eliminar();
        }
    }
    pila.vaciar(aux);
    pila.adicionar(valor);
    pila.mostrar();
}
```

Cambiar la dirección de algunos CLIENTES de la PILA.

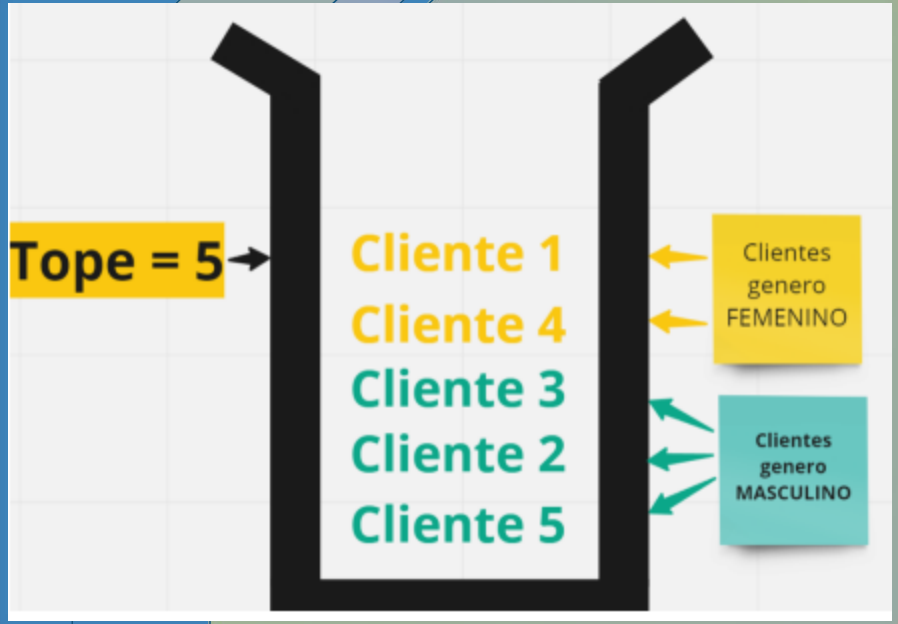
- El método deberá llamarse **asignaDireccion(Pila, nuevaDireccion)**
- El método debe ser creado en la clase **MAIN** como un método estático.
- El método recibe 2 parámetros
  - La Pila de Clientes
  - El valor(String) de la nueva dirección.
- **Cambiar la dirección del cliente** siempre y cuando el género sea **FEMENINO**.

Cambiar la dirección de algunos CLIENTES de la PILA.

```java
public static void asignaDireccion(PilaCliente pila, String nuevaDireccion) {
    PilaCliente aux = new PilaCliente( max: 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.mostrar();
}
```



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**.

Mover ÍTEMS de la PILA.

```java
public static void reordenaPila(PilaCliente pila) {
    PilaCliente aux1 = new PilaCliente( max: 10);
    PilaCliente aux2 = new PilaCliente( max: 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.mostrar();
}
```


¡GRACIAS!