


ESTRUCTURA DE DATOS

FREDDY MACHACA MAMANI

The background is black with several thin, white, hand-drawn style lines. A small cyan square is located in the upper right quadrant. On the left side, there are several horizontal white lines of varying lengths. The text '01' is rendered in a large, bold, cyan font, with the '0' having a pixelated or blocky appearance.

01

PARTE TEORICA

The background is black with several thin, white, hand-drawn style lines. A small cyan square is located in the upper right quadrant. On the left side, there are several horizontal white lines of varying lengths. The text '01' is rendered in a large, bold, cyan font, with the '0' having a pixelated or blocky appearance.

QUE ES 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.



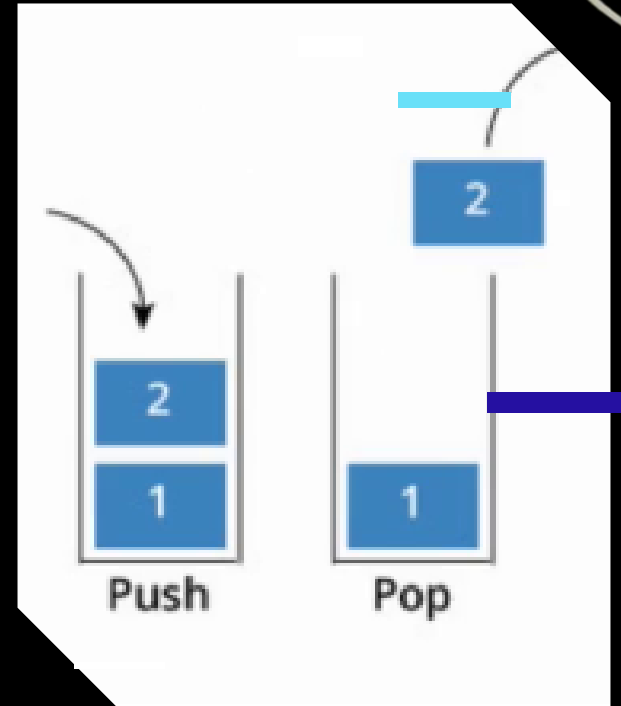
TIPOS DE ESTRUCTURA

En programación estructurada se utilizan tres tipos de estructuras:

- Secuenciales, aquellas que se ejecutan una después de otra siguiendo el orden en que se han escrito.
- Decisión, que permiten omitir parte del código o seleccionar el flujo de ejecución de entre dos o más alternativas.
- Iterativas, que se utilizan para repetir la ejecución de cierta parte del programa.

QUE ES ESTRUCTURA 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.



LIFO

PILA

Una pila se define formalmente como una colección de datos a los cuales se puede acceder mediante un extremo

TOPE

Numero de elementos que tiene la pila

MAX

Numero máximo de elementos que soporta la pila

STACK

Un objeto de la clase Stack es una pila. Permite almacenar objetos y luego recuperarlos en el orden inverso en el cual se insertaron.

esVacia() y esLlena() en una PILA

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


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

Métodos Estáticos

Un método static en Java es un método que pertenece a la clase y no al objeto. Un método static solo puede acceder a variables o tipos de datos declarados como static. Un método static sólo puede acceder a datos static.

METODOS MINIMOS DE UNA PILA

PilaCliente		
f	tope	int
f	pila1	Clientes []
f	max	int
m	eliminar()	Clientes
m	vaciar(PilaCliente)	void
m	esvacio()	boolean
m	llenar()	void
m	eslleno()	boolean
m	nroElementos()	int
m	insertar(Clientes)	void
m	mostrar()	void

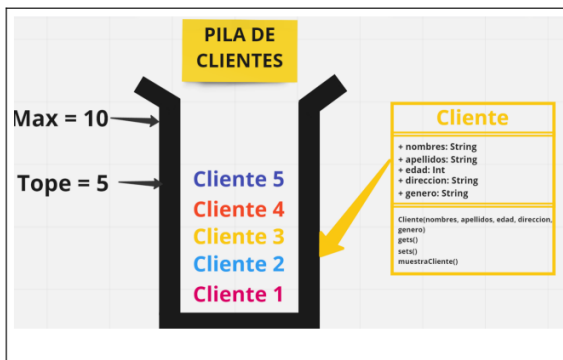
The background is black with several thin white lines of varying lengths and orientations. A small cyan square is located in the upper right quadrant. The text '02' is rendered in a large, cyan, pixelated font on the left side.

02

PARTE PRACTICA

Clase Cliente

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 estar dentro de este paquete)

PilaCliente		
m	PilaCliente (int)	
f	max	int
f	tope	int
f	pila1	Clientes []
m	eliminar ()	Clientes
m	eslleno ()	boolean
m	vaciar (PilaCliente)	void
m	nroElementos ()	int
m	insertar (Clientes)	void
m	llenar ()	void
m	mostrar ()	void
m	esvacio ()	boolean

Clientes		
m	Clientes (String, String, int, String, String)	
f	apellido	String
f	direccion	String
f	edad	int
f	nombre	String
f	genero	String
m	setApellido (String)	void
m	setNombre (String)	void
m	getApellido ()	String
m	setDireccion (String)	void
m	getGenero ()	String
m	getNombre ()	String
m	setGenero (String)	void
m	mostrarDatos ()	void
m	setEdad (int)	void
m	getEdad ()	int
m	getDireccion ()	String

02

Clase Cliente

—

```
public class Main {  
    public static void main(String[] args) {  
        Clientes cliente1 = new Clientes( nombre: "Juan", apellido: "Perez", edad: 20, direccion: "Acre", genero: "Masculino");  
        Clientes cliente2 = new Clientes( nombre: "Maria", apellido: "Gomez", edad: 25, direccion: "Mirapiranga", genero: "Femenino");  
        Clientes cliente3 = new Clientes( nombre: "Pedro", apellido: "Gonzales", edad: 30, direccion: "Distrito federal", genero: "Masculino");  
        Clientes cliente4 = new Clientes( nombre: "Luis", apellido: "Perez", edad: 35, direccion: "Rio de janeiro", genero: "Masculino");  
        Clientes cliente5 = new Clientes( nombre: "Ana", apellido: "Gomez", edad: 40, direccion: "Avenida tiradentes", genero: "Femenino");  
  
        PilaCliente pila = new PilaCliente( max: 5);  
        pila.insertar(cliente1);  
        pila.insertar(cliente2);  
        pila.insertar(cliente3);  
        pila.insertar(cliente4);  
        pila.insertar(cliente5);  
    }  
}
```

02

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){
    int contador = 0;
    Clientes elem = null;
    if(pila.esvacio()){
        System.out.println("La pila esta vacia");
    } else {
        while(!pila.esvacio()){
            elem = pila.eliminar();
            if(elem.getEdad() > edadMayor){
                contador++;
            }
        }
    }
    System.out.println("Hay " + contador + " clientes mayores de " + edadMayor + " años");
}
```

Hay 4 clientes mayores de 20 años

02

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

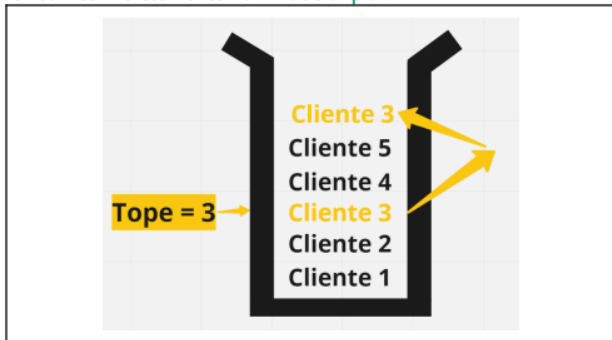
- El método deberá llamarse **mayoresCiertaEdad(Pila, edadMayor)**
- El método debe ser creado en la clase **MAIN** como un método estático.
- El método recibe 2 parámetros
 - La Pila de Clientes
 - El valor de la edad.
- Adjuntar los siguientes
 - El **código** del método que resuelve el problema.
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

```
public static void mayoresCiertaEdad(PilaCliente pila, int edadMayor){
    int contador = 0;
    Clientes elem = null;
    if(pila.esvacio()){
        System.out.println("La pila esta vacia");
    } else {
        while(!pila.esvacio()){
            elem = pila.eliminar();
            if(elem.getEdad() > edadMayor){
                contador++;
            }
        }
    }
    System.out.println("Hay " + contador + " clientes mayores de " + edadMayor + " años");
}
```

Hay 4 clientes mayores de 20 años

02

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.

```
Clientes cliente1 = new Clientes( nombre: "Juan", apellido: "Perez", edad: 20, direccion: "Acre", genero: "Masculino");
Clientes cliente2 = new Clientes( nombre: "Maria", apellido: "Gomez", edad: 25, direccion: "Mirapiranga", genero: "Femenino");
Clientes cliente3 = new Clientes( nombre: "Pedro", apellido: "Gonzales", edad: 30, direccion: "Distrito federal", genero: "Masculino");
Clientes cliente4 = new Clientes( nombre: "Luis", apellido: "Perez", edad: 35, direccion: "Rio de janeiro", genero: "Masculino");
Clientes cliente5 = new Clientes( nombre: "Ana", apellido: "Gomez", edad: 40, direccion: "Avenida tiradentes", genero: "Femenino");
```

```
public static void moverKesimo(PilaCliente pila, Clientes k){
    PilaCliente aux = new PilaCliente( max: 10);
    Clientes valorExtraidoPila=null;
    while(!pila.esvacio()){
        valorExtraidoPila = pila.eliminar();
        if(valorExtraidoPila != k){
            aux.insertar(valorExtraidoPila);
        }
    }
    pila.vaciar(aux);
    pila.insertar(k);
    pila.mostrar();
}
```

Elementos de la pila

-----DATOS CLIENTES-----

Nombre: Pedro
Apellido: Gonzales
Edad: 30
Direccion: Distrito federal
Genero: Masculino

-----DATOS CLIENTES-----

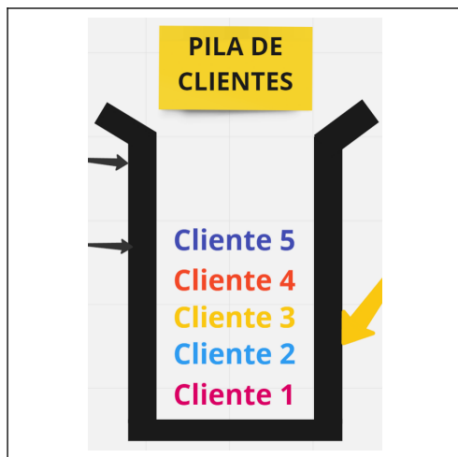
Nombre: Ana
Apellido: Gomez
Edad: 40
Direccion: Avenida tiradentes
Genero: Femenino

-----DATOS CLIENTES-----

Nombre: Luis
Apellido: Perez

02

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.**

```
public static void asignaDireccion(PilaCliente pila, String nuevaDireccion){
    PilaCliente aux= new PilaCliente( max: 10);
    Clientes valorExtraidoPila=null;
    while(!pila.esvacio()){
        valorExtraidoPila = pila.eliminar();

        if(valorExtraidoPila.getGenero().equals("Femenino")){
            valorExtraidoPila.setDireccion(nuevaDireccion);
            valorExtraidoPila.mostrarDatos();
        }
        aux.insertar(valorExtraidoPila);
    }
    pila.vaciar(aux);
}
```

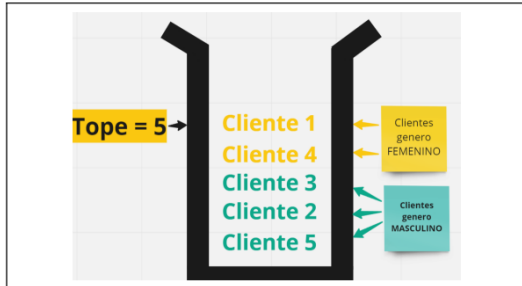
-----DATOS CLIENTES-----

Nombre: Ana
Apellido: Gomez
Edad: 40
Direccion: Av. Siempre Viva 123
Genero: Femenino

-----DATOS CLIENTES-----

Nombre: Maria
Apellido: Gomez
Edad: 25
Direccion: Av. Siempre Viva 123
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.

```
public static void reordenaPila(PilaCliente pila){
    PilaCliente aux= new PilaCliente( max: 10);
    Clientes valorExtraidoPila=null;
    while(!pila.esvacio()){
        valorExtraidoPila = pila.eliminar();
        if(valorExtraidoPila.getGenero().equals("Femenino")){
            valorExtraidoPila.mostrarDatos();
        }
        else{
            aux.insertar(valorExtraidoPila);
        }
    }
    pila.vaciar(aux);
    pila.mostrar();
}
```

Nombre: Ana
 Apellido: Gomez
 Edad: 40
 Direccion: Avenida tiradentes
 Genero: Femenino

-----DATOS CLIENTES-----

Nombre: Maria
 Apellido: Gomez
 Edad: 25
 Direccion: Mirapiranga
 Genero: Femenino

Elementos de la pila

-----DATOS CLIENTES-----

Nombre: Luis
 Apellido: Perez
 Edad: 35
 Direccion: Rio de janeiro
 Genero: Masculino

-----DATOS CLIENTES-----

Nombre: Pedro
 Apellido: Gonzales
 Edad: 30
 Direccion: Distrito federal
 Genero: Masculino

-----DATOS CLIENTES-----

Nombre: Juan
 Apellido: Perez
 Edad: 20
 Direccion: Acre
 Genero: Masculino