



PILAS

ESTRUCTURA DE DATOS

ESTUDIANTE: IRIS MICHELLE VELASCO ARUQUIPA

CODIGO: SIS14125434

DOCENTE: WILLIAM BARRA



¿A QUE SE REFIERE CUANDO SE HABLA DE ESTRUCTURA 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?

- ARRAYS
- LISTAS ENLAZADAS
- COLAS
- PILAS
- ARBOLES BINARIOS





¿APOYÁNDOSE EN EL LINK ADJUNTO,
EXPLIQUE, POR QUÉ SON ÚTILES LAS
ESTRUCTURAS DE DATOS?.

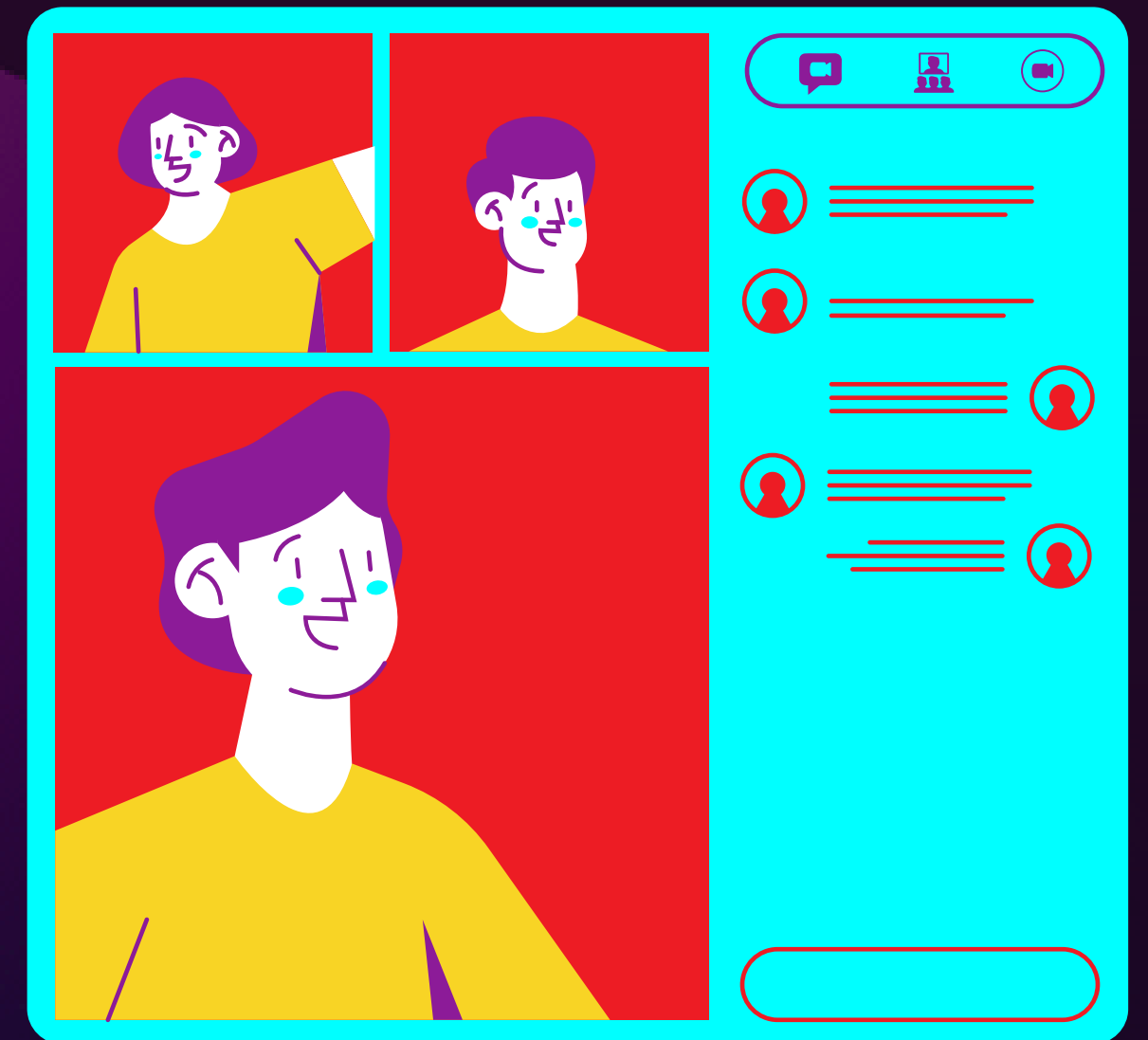
Las estructuras de datos son útiles
porque nos permiten tener una batería de
herramientas para solucionar ciertos
tipos de problemas.



¿QUÉ ES UNA PILA?

Una pila es una estructura de datos de entradas ordenadas tales que solo se introduce y elimina por un extremo, llamado cima o tope.

Una estructura de datos tipo pila permite agregar nodos a la pila y eliminarlos de esta sólo desde su parte superior.





¿QUÉ ES STACK EN JAVA?

Stack in java es una sección de memoria que contiene métodos, variables locales y variables de referencia.

¿UNA STACK SERÁ LO MISMO QUE 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 TOPE EN UNA PILA?

Esta función es la que se va incrementado o disminuyendo a medida que se valla ingresando o sacando dato.

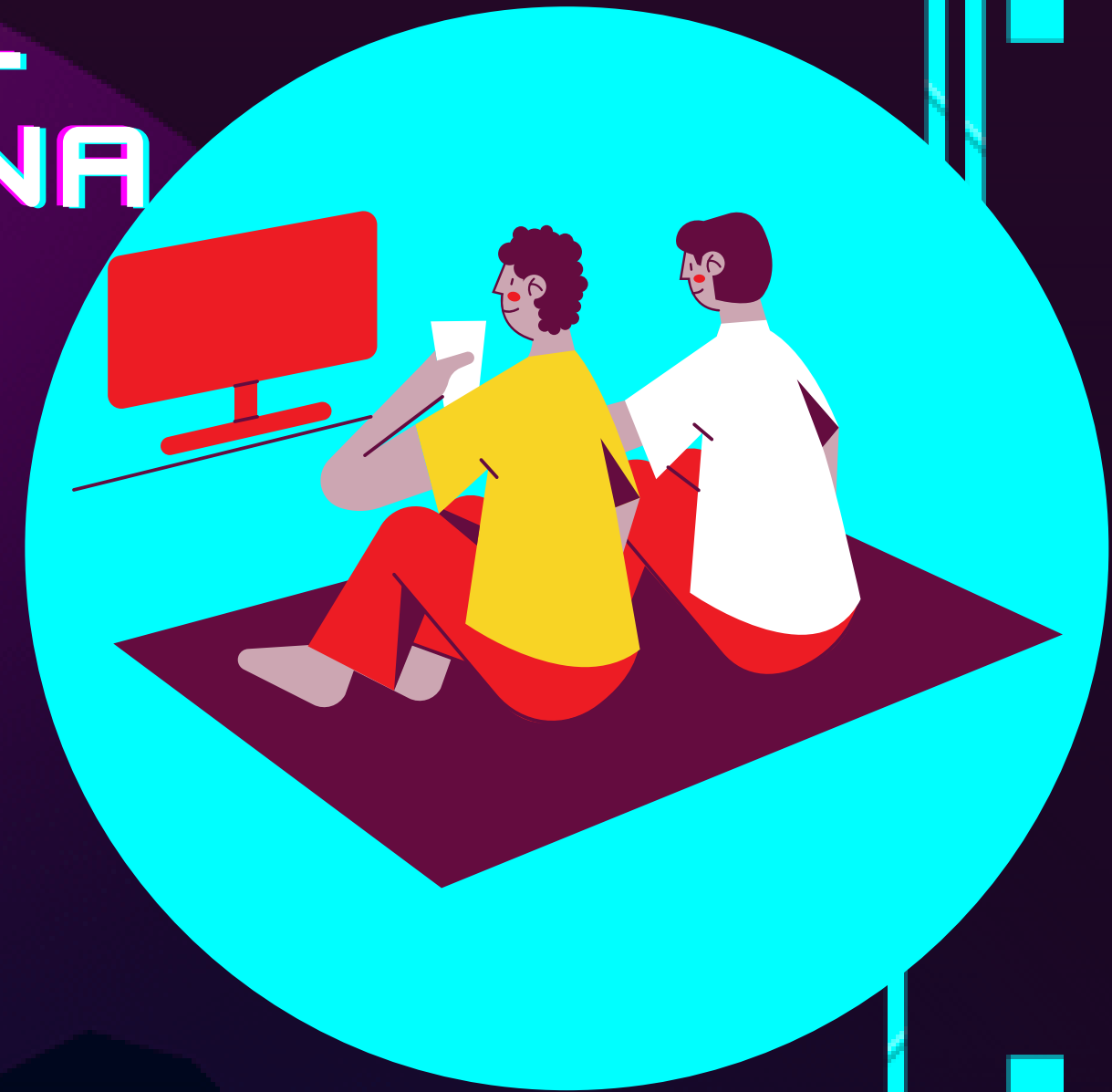
¿QUÉ ES MAX EN UNA PILA?



Esta variable se utiliza para indicar el
último
elemento que se insertó en la pila.

¿A QUE SE REFIERE EL MÉTODO ESVACIA() EN UNA PILA?

Una pila vacía no contiene elemento alguno dentro de la estructura y el tope de la misma apunta a nulo.



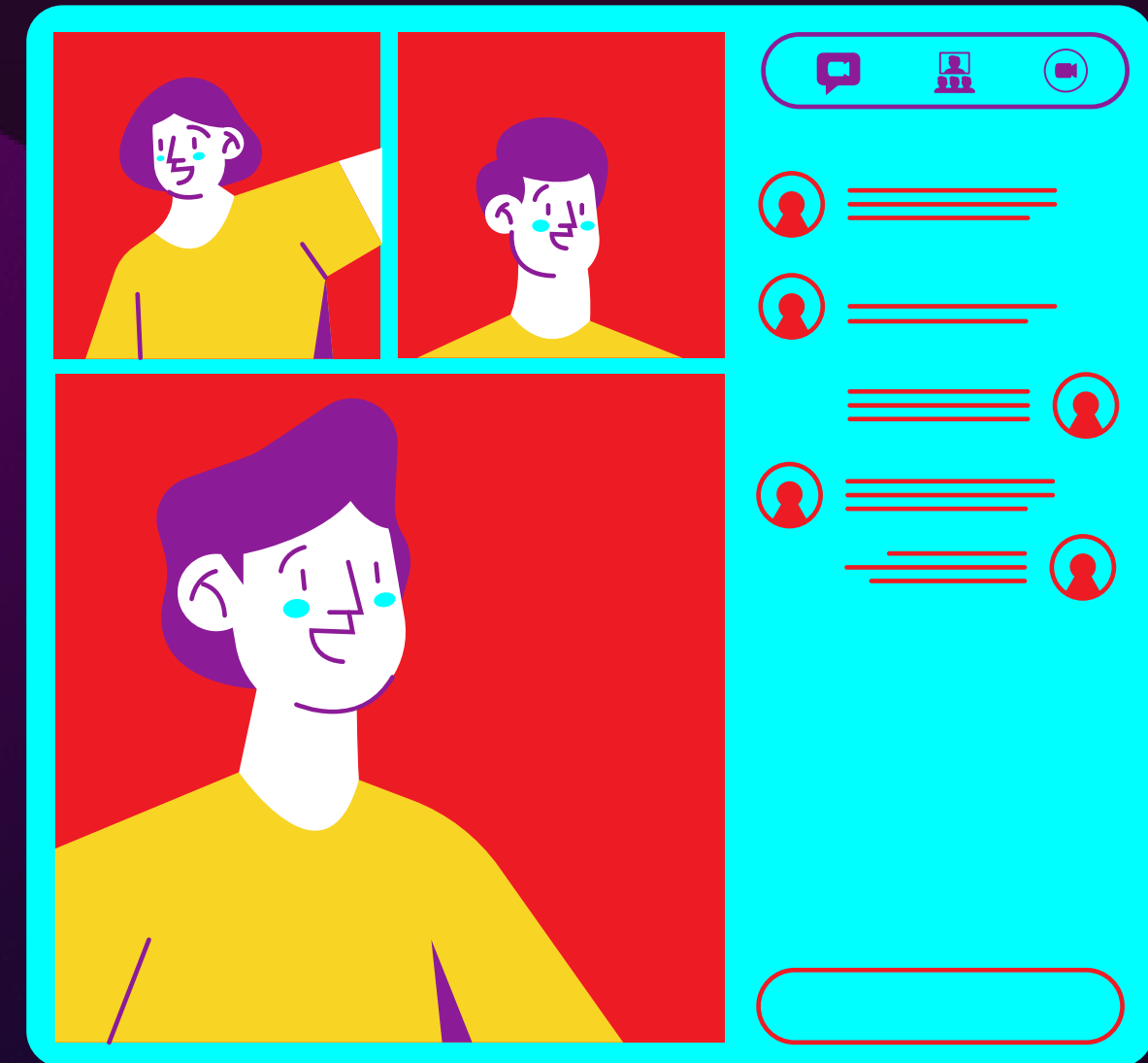
¿A QUE SE REFIERE EL MÉTODO ESLLENA() EN UNA PILA?



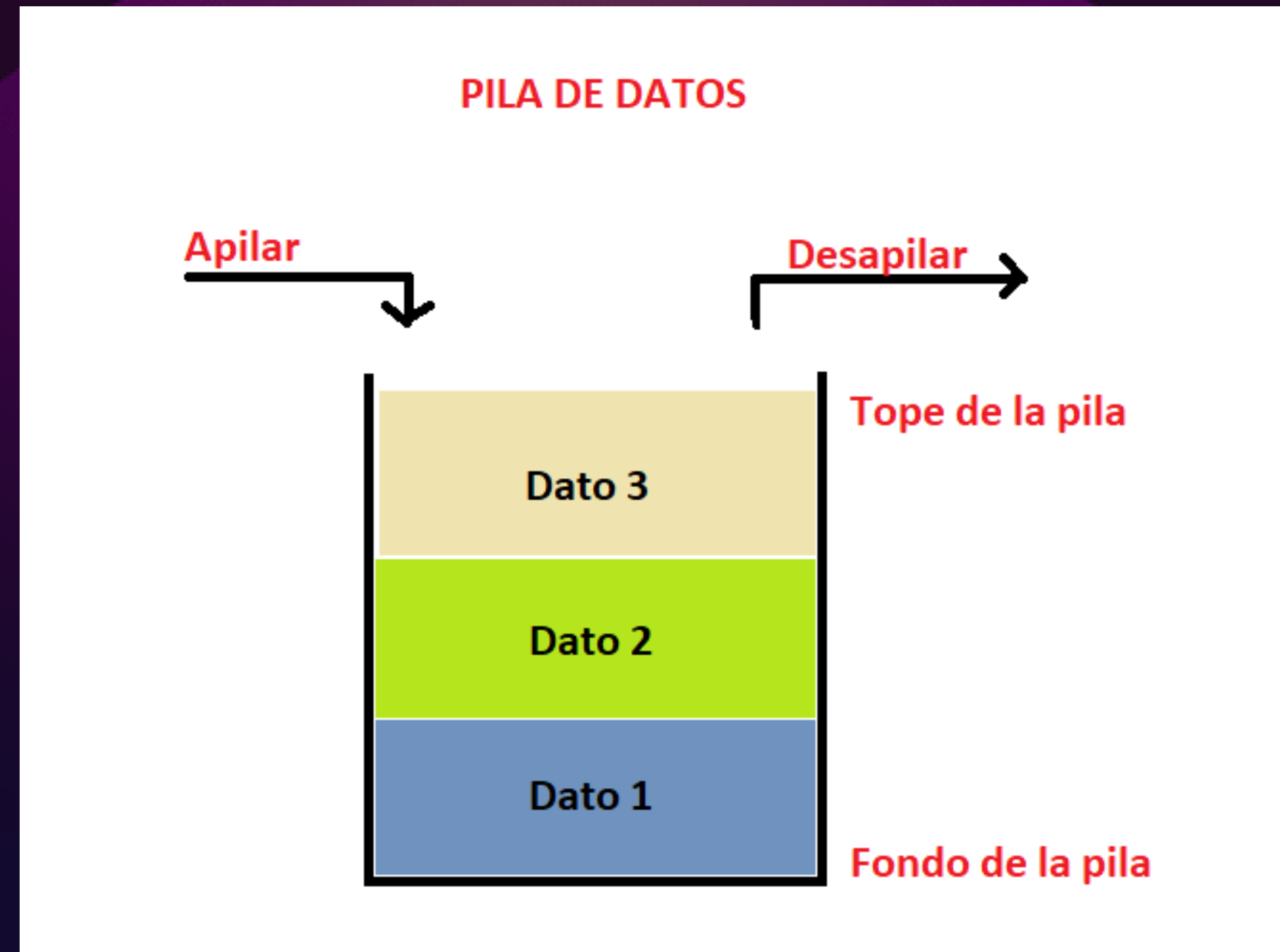
Por definición, una estructura de datos tipo pila tiene un tamaño fijo. Cuando la pila ha almacenado el número máximo de nodos definido, se dice que la pila está llena.

¿QUÉ SON LOS MÉTODOS ESTÁTICOS EN JAVA?

El método estático en Java es un método que pertenece a la clase y no al objeto. Un método estático solo puede acceder a datos estáticos.



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



PRACTICA

CREAR LAS CLASES NECESARIAS PARA LA PILA DE CLIENTES.



creamos una clase
cliente con sus
parámetros

```
package PilaDeClientes;

public class Cliente {
    3 usages
    private String Nombre;
    3 usages
    private String Apellido;
    3 usages
    private int Edad;
    3 usages
    private String Direccion;
    3 usages
    private String genero;
```

PARAMETRO DE LA CLASE CLIENTE

```
public Cliente(String Nombre,String Apellido, int Edad, String Direccion, String genero)  
  
    this.Nombre=Nombre;  
    this.Apellido=Apellido;  
    this.Edad=Edad;  
    this.Direccion=Direccion;  
    this.genero=genero;
```


GET

```
public String getNombre() {  
    return Nombre;  
}  
  
1 usage  
public String getApellido() {  
    return Apellido;  
}  
  
2 usages  
public int getEdad() {  
    return Edad;  
}  
  
1 usage  
public String getDireccion() {  
    return Direccion;  
}  
  
3 usages  
public String getGenero() {  
    return genero;  
}
```

SET

```
public void setNombre(String nombre) {  
    Nombre = nombre;  
}  
  
public void setApellido(String apellido) {  
    Apellido = apellido;  
}  
  
public void setEdad(int edad) {  
    Edad = edad;  
}  
  
1 usage  
public void setDireccion(String direccion) {  
    Direccion = direccion;  
}  
  
public void setGenero(String genero) {  
    this.genero = genero;  
}
```

METODO MOSTRAR

```
public void MostrarCliente()
{
    System.out.println("\n Mostrando Libro");
    System.out.println(" Nombre: " + this.getNombre());
    System.out.println(" Apellido: " + this.getApellido());
    System.out.println(" Edad: " + this.getEdad());
    System.out.println(" Direccion: " + this.getDireccion());
    System.out.println(" Genero: " + this.getGenero());
}
```



CREAMOS UNA CLASE DE NOMBRE PILA_CLIENTE

```
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];  
    }  
}
```


DECLARAMOS NUESTRAS VARIABLES
Y CREAMOS UN CONSTRUCTOR



ESTE METODO ES PARA VER SI LA PILA ESTA VACIA

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

ES UN METOD BOOLEAN
USAREMOS UN IF PARA
PREGUNTAR SI LA PILA
ESTA LLENA O VACIA
Y ASI PODEMOS SABER SI
LA PILA ESTA VACIA



METODO ES LLENO

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

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

METODO NUMERO DE ELEMENTOS

En este método pondremos un return para consultar cuantos elementos tiene la pila



METODO PARA AGREGAR UN NUEVO ELEMENTO A LA PILA

```
public void adicionar (Cliente nuevoLCiente) {  
    if (this.esLleno() == false) {  
        this.tope = this.tope + 1;  
        this.Clientes[this.tope] = nuevoLCiente;  
    } else {  
        System.out.println("La pila de clientes está llena");  
    }  
}
```


Metodo eliminar

```
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 Clientes está vacía");  
    }  
    return elementoEliminado;  
}
```

METODO PARA VER SI
LA PILA ESTA VACIA

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



DETERMINAR CUÁNTOS CLIENTES SON MAYORES DE 20 AÑOS.

```
public static void MayorCiertaEdad(PilaCliente pila, int edadMayor){
    PilaCliente aux = new PilaCliente( max: 10);
    int MayorEdad = 0;
    Cliente Valorestraído = null;
    while (pila.esVacio() == false){
        Valorestraído = pila.eliminar();
        if (Valorestraído.getEdad() > edadMayor){
            MayorEdad = MayorEdad +1;
        }
        aux.adicionar(Valorestraído);
    }
    System.out.println("Clientes de mas de " + edadMayor + "son: " + MayorEdad);
}
```



MOVER EL K-ÉSIMO ELEMENTO AL FINAL DE LA PILA

```
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.

```
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.

```
public static void reordenaPila(PilaCliente pila, String reacomodar){
    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();
    }
}
```