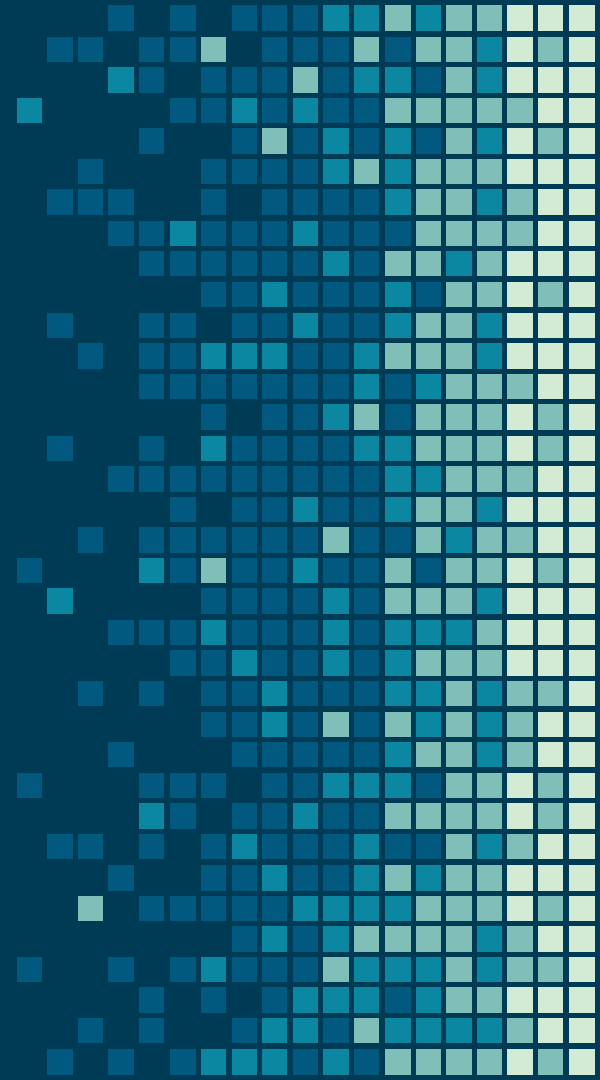


# EVALUACION PROCESUAL HITO 3

ESTRUCTURA DE DATOS

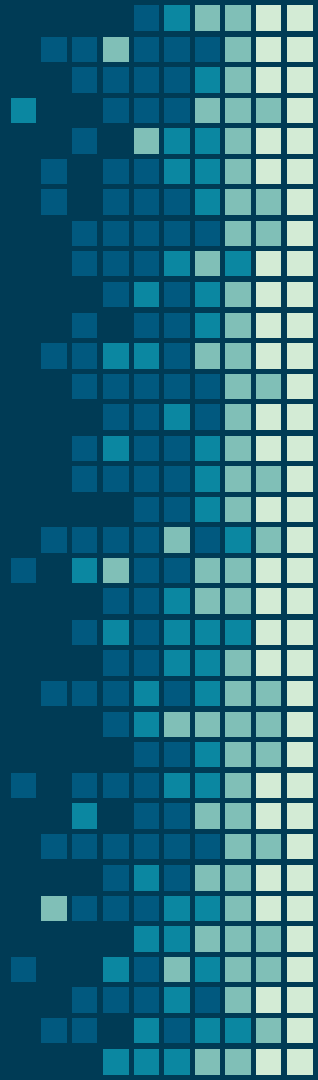
ESTUDIANTE: NILBER MAYTA CUNO





# PARTE TEORICA

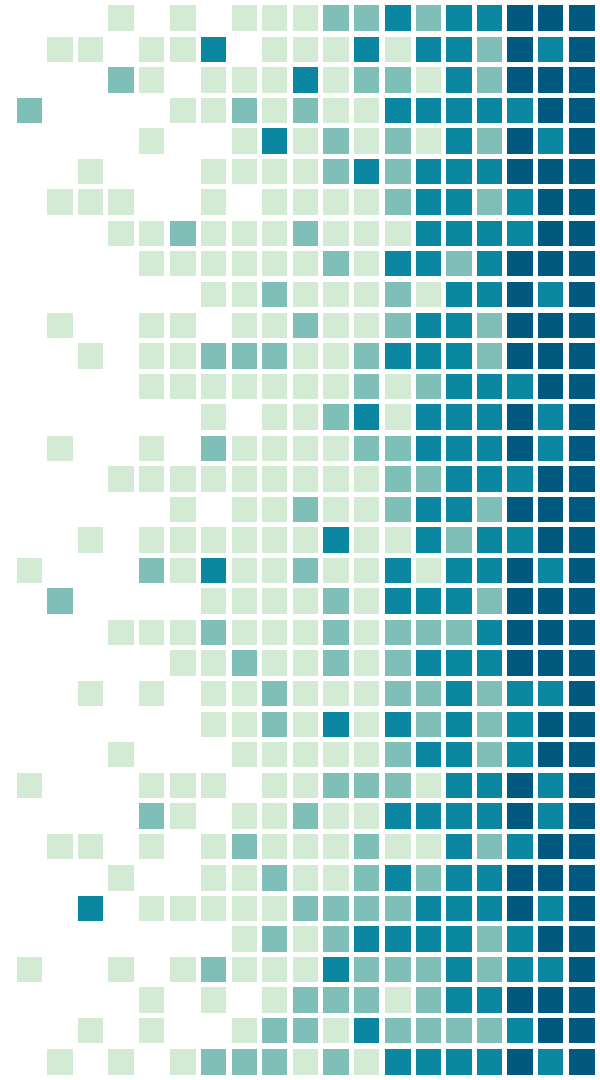
Manejo de conceptos



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

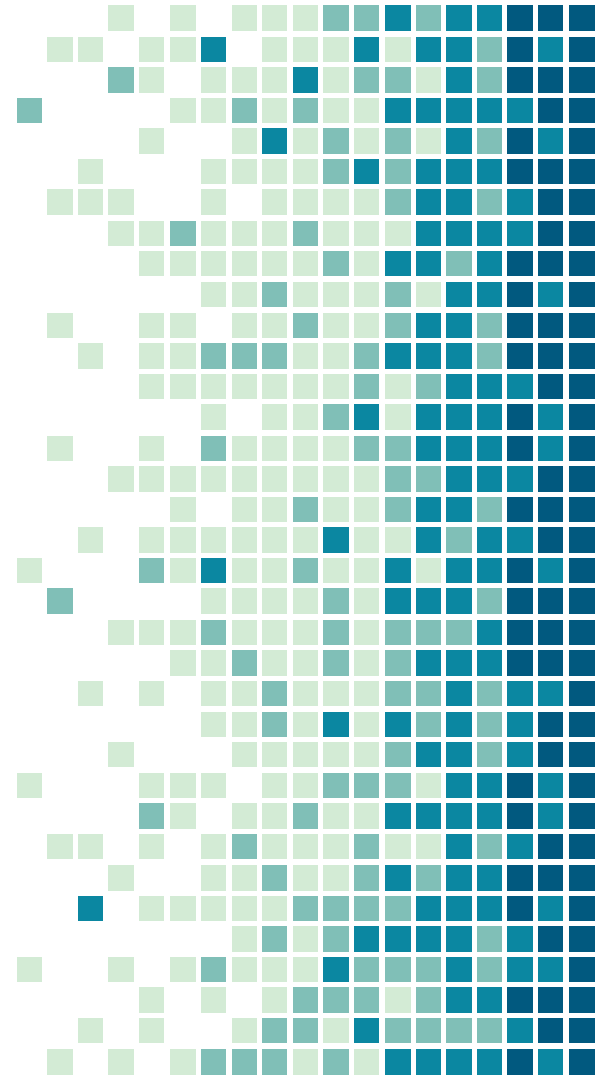
La estructura de datos es un modo de representar informacion, que cuenta con un comportamiento interno que se rigue por determinadas reglas.

Estas permiten almacenar datos de manera que puedan ser utilizados de manera eficiente.



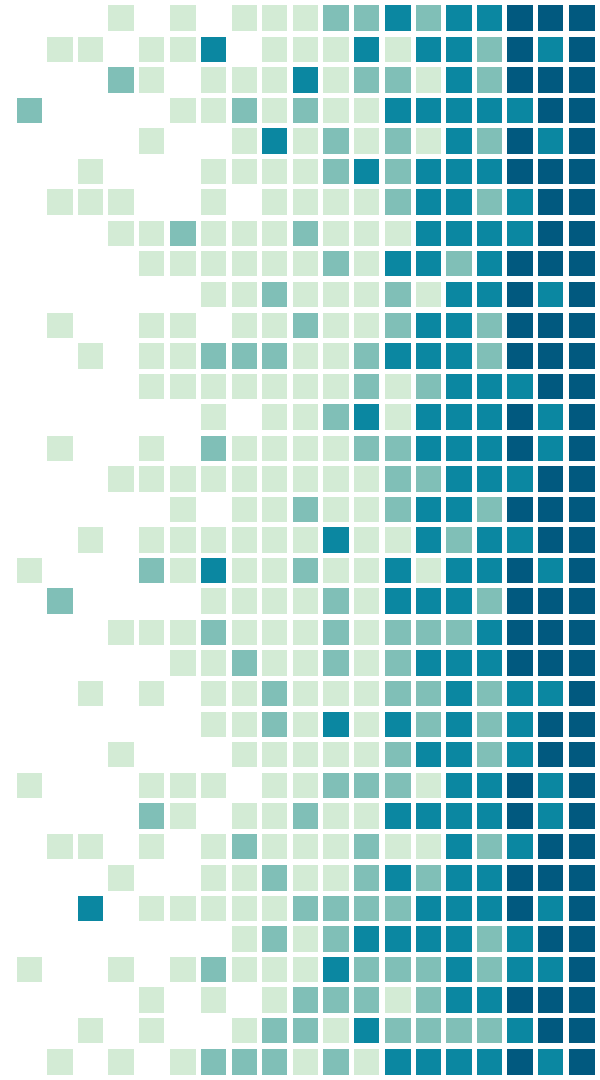
## 2. ¿Cuáles son los TIPOS DE ESTRUCTURA QUE EXISTE?

Existen diversas estructuras de datos como ser los arrays, las listas, las pilas, arboles, grafos entre otros.



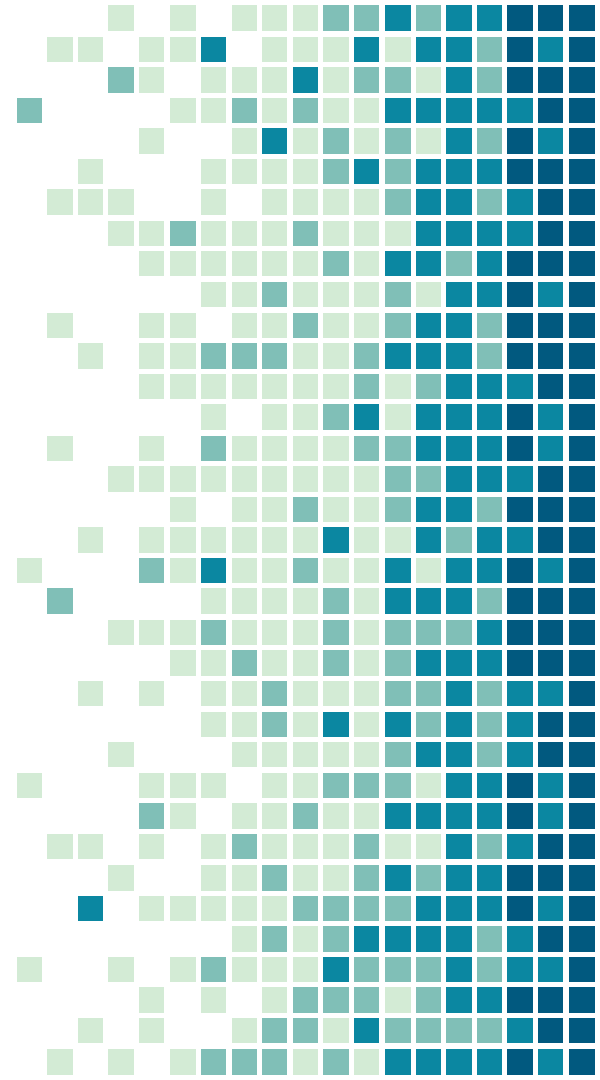
### 3. ¿Apoyándose en el link adjunto, explique, por qué son útiles las estructuras de datos?.

Las estructuras de datos son buenas para procesos de optimización haciendo que los datos se procesen de una manera más eficiente, además de tener un respaldo de una acción anterior.



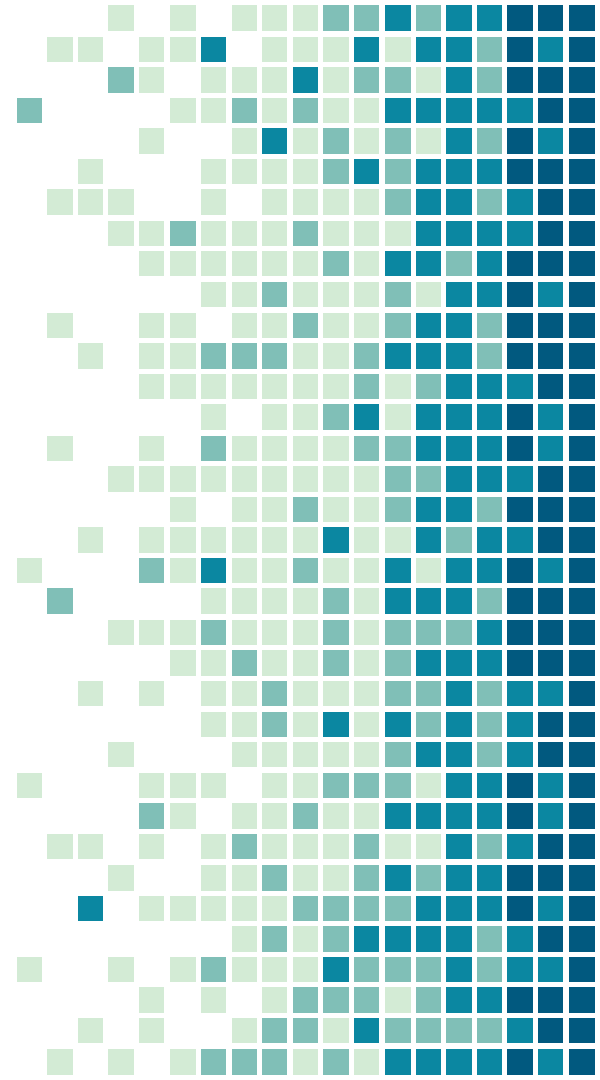
## 4. ¿Qué es una PILA?

Una pila es una estructura de datos lineal, que permite el acceso solo de los elementos recién agregados, es decir el ultimo item que agregamos es el primero que eliminamos.



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

Un stack es simplemente una pila, es simplemente como se dice le dice a una pila en ingles, esta nos permite almacenar item y luego recuperarlos en un orden inverso.



## 6. ¿Qué es TOPE en una PILA?

El TOPE en una pila es nos representa el numero de items que tengamos en ese momento en la pila.

## 7. ¿Qué es MAX en una PILA?

Mientras que el MAX representa el numero maximo de items que puede tener una pila.

```
public Pila(){  
    this.max = 10;  
    this.tope = 0;  
    this.items = new String[max+1];  
}
```



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

El metodo esVacia() nos manda un valor de tipo booleano que nos indica si existe algun item almacenado en la pila. esLLena() por otro lado nos dice si ya alcanzamos el limite de nuestra pila.

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

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

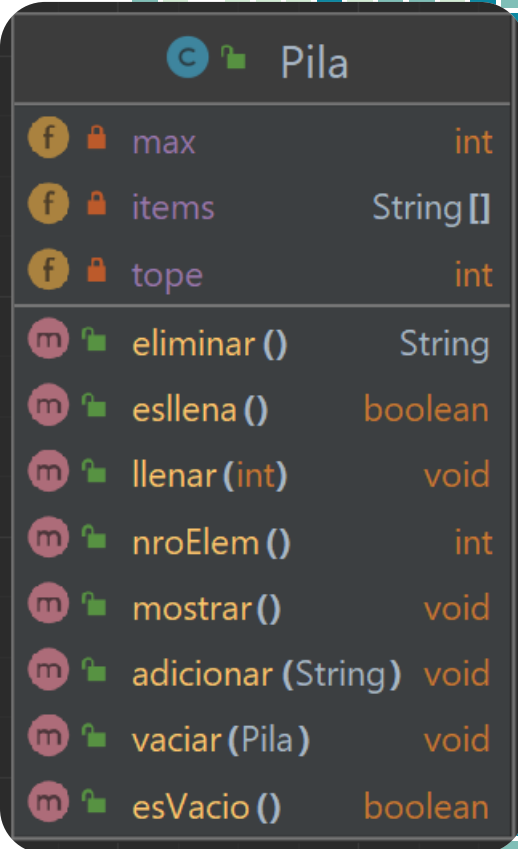
## 9. ¿Qué son los métodos estáticos en JAVA?

Los metodos estaticos (static) son aquellos que desarrollamos en nuestra clase **main** y nos permite invocarla sin crear previamente ningun objeto.

```
static void funcion(){  
    }  
}
```

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

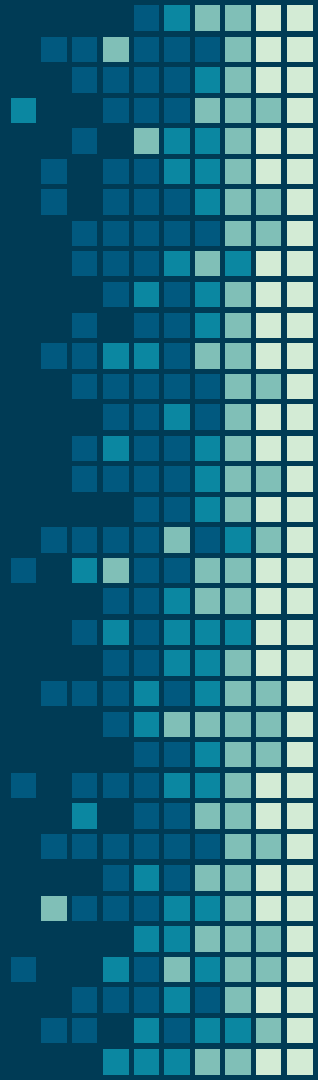
Los metodos necesarios para crear un pila son:



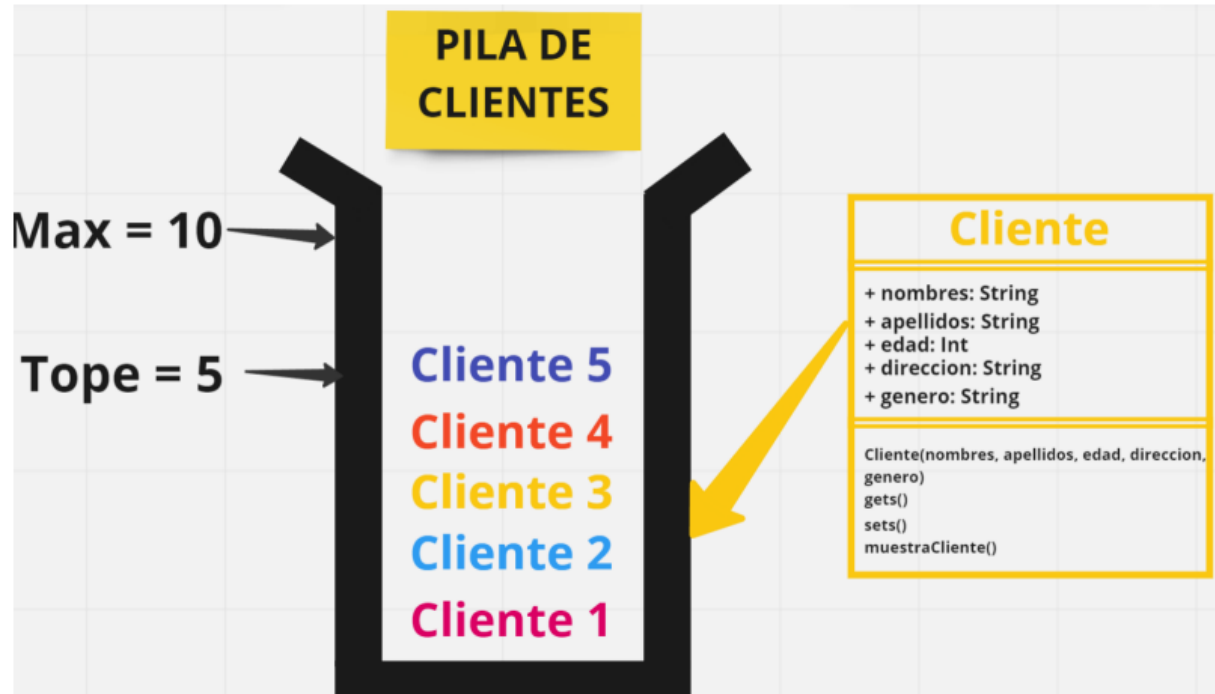
| Pila |                    |           |
|------|--------------------|-----------|
| f    | max                | int       |
| f    | items              | String [] |
| f    | tope               | int       |
| m    | eliminar ()        | String    |
| m    | esllena ()         | boolean   |
| m    | llenar (int)       | void      |
| m    | nroElem ()         | int       |
| m    | mostrar ()         | void      |
| m    | adicionar (String) | void      |
| m    | vaciar (Pila)      | void      |
| m    | esVacio ()         | boolean   |



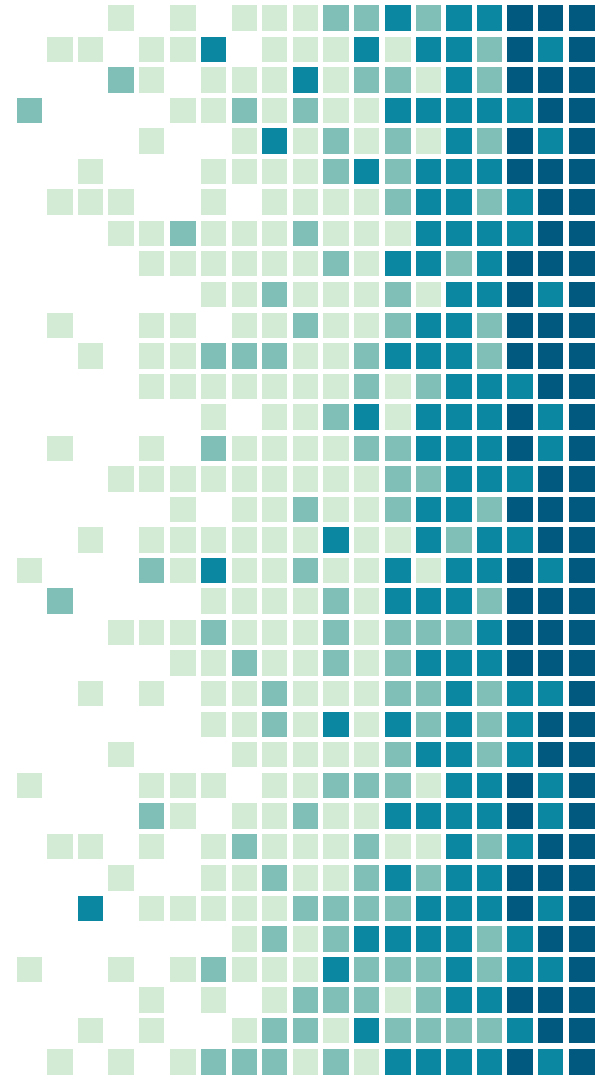
# PARTE PRACTICA




## 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 ítems de la pila.



A decorative graphic in the top right corner consisting of a grid of blue and green squares of varying shades, arranged in a pattern that tapers to the right.

```
public class Main {
```

```
no usages
```

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

```
    Cliente cli1 = new Cliente( nombre: "Juan", apellido: "Perez", edad: 35, direccion: "Calle 123", genero: "Masculino");
```

```
    Cliente cli2 = new Cliente( nombre: "Ana", apellido: "Perez", edad: 28, direccion: "Avenida 456", genero: "Femenino");
```

```
    Cliente cli3 = new Cliente( nombre: "Carlos", apellido: "Rodriguez", edad: 42, direccion: "Calle 789", genero: "Masculino");
```

```
    Cliente cli4 = new Cliente( nombre: "Maria", apellido: "Hernandez", edad: 23, direccion: "Calle 1011", genero: "Femenino");
```

```
    Cliente cli5 = new Cliente( nombre: "Luis", apellido: "Martines", edad: 55, direccion: "Avenida 1213", genero: "Masculino");
```

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

```
    pila.adicionar(cli1);
```

```
    pila.adicionar(cli2);
```

```
    pila.adicionar(cli3);
```

```
    pila.adicionar(cli4);
```

```
    pila.adicionar(cli5);
```

```
    pila.mostrar();
```

"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.3.3\lib\idea\_rt.jar=61094:C:\Program Files\JetBrains\IntelliJ IDEA 2022.3.3\bin" -jar C:\Program Files\JetBrains\IntelliJ IDEA 2022.3.3\bin\idea\_rt.jar 61094

Mostrando la Pila de libros

Mostrando Cliente

Nombre: Luis

Apellido: Martines

Edad: 55

Direccion: Avenida 1213

Genero: Masculino

Mostrando Cliente

Nombre: Maria

Apellido: Hernandez

Edad: 23

Direccion: Calle 1011

Genero: Femenino

Mostrando Cliente

Nombre: Carlos

Apellido: Rodriguez

Edad: 42

Direccion: Calle 789

Genero: Masculino

Mostrando Cliente

Nombre: Ana

Apellido: Perez

Edad: 28

Direccion: Avenida 456

Genero: Femenino

Mostrando Cliente

Nombre: Juan

Apellido: Perez

Edad: 35

Direccion: Calle 123

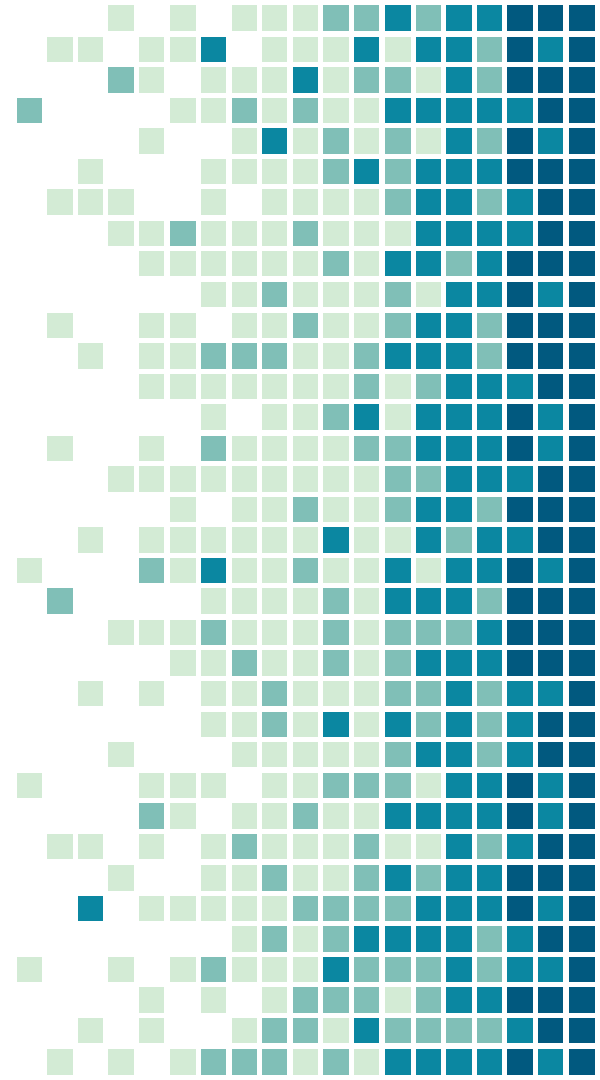
Genero: Masculino

Process finished with exit code 0



## 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.



usage

```
static void mayoresCiertaEdad(PilaCliente pila, int edadMayor){  
    PilaCliente aux = new PilaCliente();  
    Cliente clienteEliminado = null;  
    int cont = 0;  
    while (!pila.esVacio()){  
        clienteEliminado = pila.eliminar();  
        if(clienteEliminado.getEdad()>edadMayor){  
            cont = cont+1;  
        }  
        aux.adicionar(clienteEliminado);  
    }  
    pila.vaciar(aux);  
    System.out.println("Los clientes con una edad mayor a "+edadMayor+" son: " + cont);  
}
```

```
mayoresCiertaEdad(pila, edadMayor: 25);
```



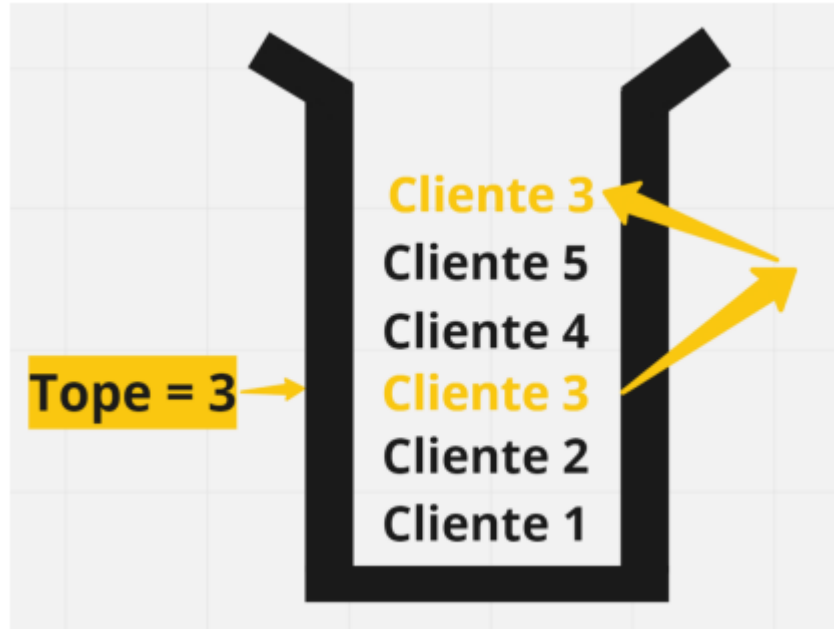
```
Genero: Masculino
```

```
Los clientes con una edad mayor a 25 son: 4
```

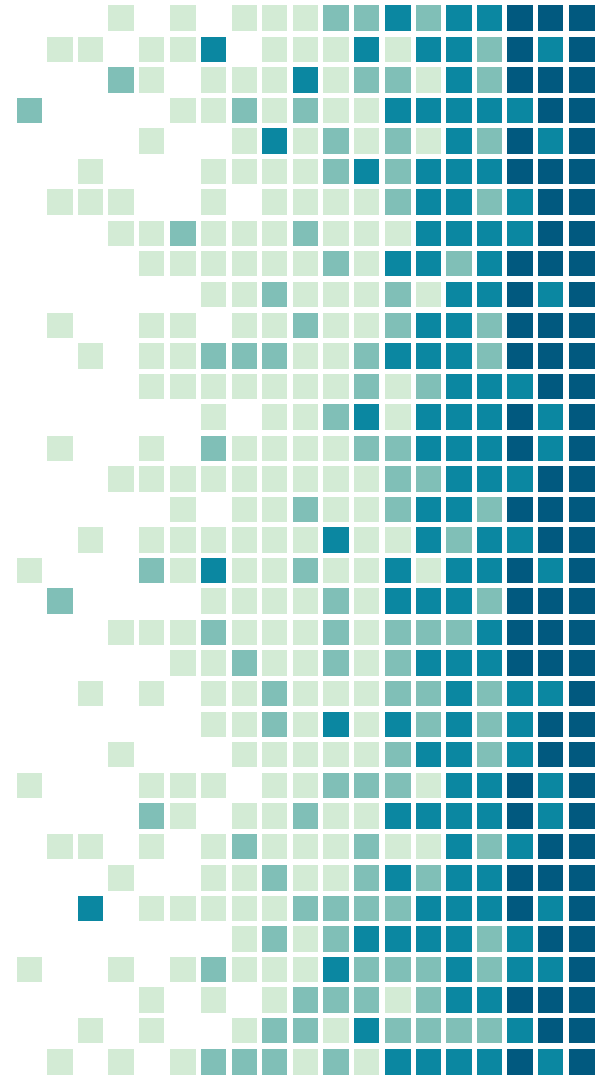
```
Process finished with exit code 0
```

```
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.



```
static void kEsimoPosicion(PilaCliente pila, int valorTope){
    PilaCliente aux = new PilaCliente();
    Cliente nombreKesimo = null;
    Cliente clienteEliminado = null;
    int valor = valorTope - 1;

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

        if(pila.nroElem() == valor){
            nombreKesimo = clienteEliminado;
        }
        else {
            aux.adicionar(clienteEliminado);
        }
    }
    pila.adicionar(nombreKesimo);
    pila.vaciar(aux);
}
```

```
kEsimoPosicion(pila, valorTope: 3);

pila.mostrar();
```

"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.3.3\lib\idea\_rt.jar=61237:C:\Program Files\JetBrains\IntelliJ IDEA 2022.3.3\bin" -Dfile.encoding=UTF-8

Mostrando la Pila de Clientes

Mostrando Cliente

Nombre: Luis

Apellido: Martinez

Edad: 55

Direccion: Avenida 1213

Genero: Masculino

Mostrando Cliente

Nombre: Maria

Apellido: Hernandez

Edad: 23

Direccion: Calle 1011

Genero: Femenino

Mostrando Cliente

Nombre: Carlos

Apellido: Rodriguez

Edad: 42

Direccion: Calle 789

Genero: Masculino

Mostrando Cliente

Nombre: Ana

Apellido: Perez

Edad: 28

Direccion: Avenida 456

Genero: Femenino

Mostrando Cliente

Nombre: Juan

Apellido: Perez

Edad: 35

Direccion: Calle 123

Genero: Masculino

Process finished with exit code 0

## 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.





```
static void asignaDIRECCION(PilaCliente pila, String nuevaDIRECCION){
    PilaCliente aux = new PilaCliente();
    Cliente clienteEliminado = null;
    while (!pila.esVacio()){
        clienteEliminado = pila.eliminar();
        if(Objects.equals(clienteEliminado.getGenero(), b: "Femenino")){
            clienteEliminado.setDIRECCION(nuevaDIRECCION);
        }
        aux.adicionar(clienteEliminado);
    }
    pila.vaciar(aux);
}
```

```
asignaDIRECCION(pila, nuevaDIRECCION: "Avenida La Paz 652");

pila.mostrar();
```

```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" -agentlib:jdwp=transport=dt_socket,address=127.0.0.1:61261,suspend=y,server=n -javaagent:C:\Users\nilbe\AppData\Local\JetBrains\
Connected to the target VM, address: '127.0.0.1:61261', transport: 'socket'
Mostrando la Pila de Clientes

Mostrando Cliente
Nombre: Luis
Apellido: Martines
Edad: 55
Direccion: Avenida 1213
Genero: Masculino

Mostrando Cliente
Nombre: Maria
Apellido: Hernandez
Edad: 23
Direccion: Avenida La Paz 652
Genero: Femenino

Mostrando Cliente
Nombre: Carlos
Apellido: Rodriguez
Edad: 42
Direccion: Calle 789
Genero: Masculino

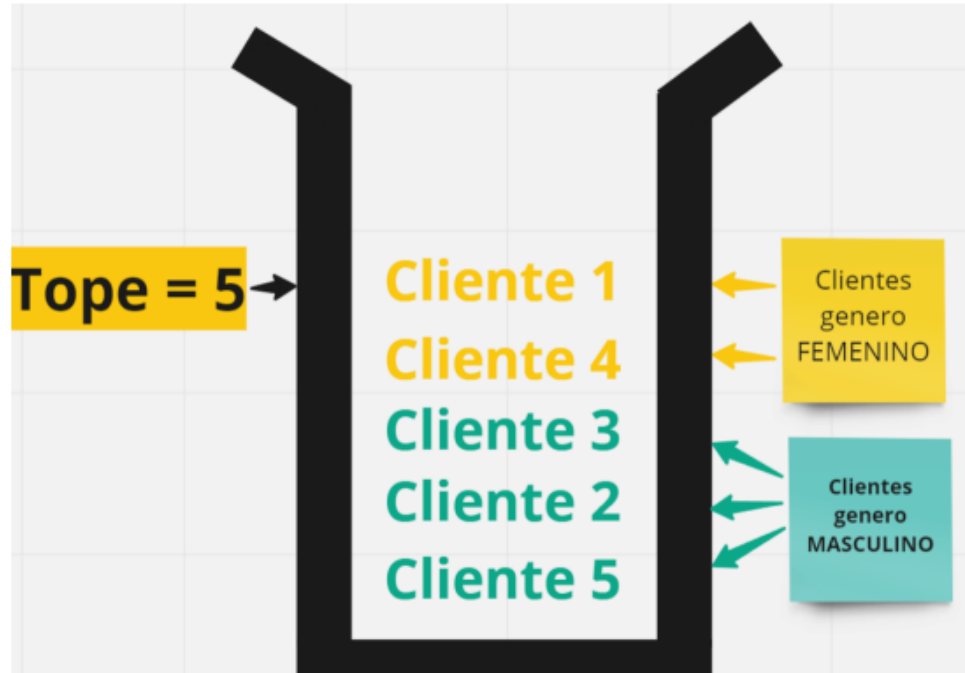
Mostrando Cliente
Nombre: Ana
Apellido: Perez
Edad: 28
Direccion: Avenida La Paz 652
Genero: Femenino

Mostrando Cliente
Nombre: Juan
Apellido: Perez
Edad: 35
Direccion: Calle 123
Genero: Masculino

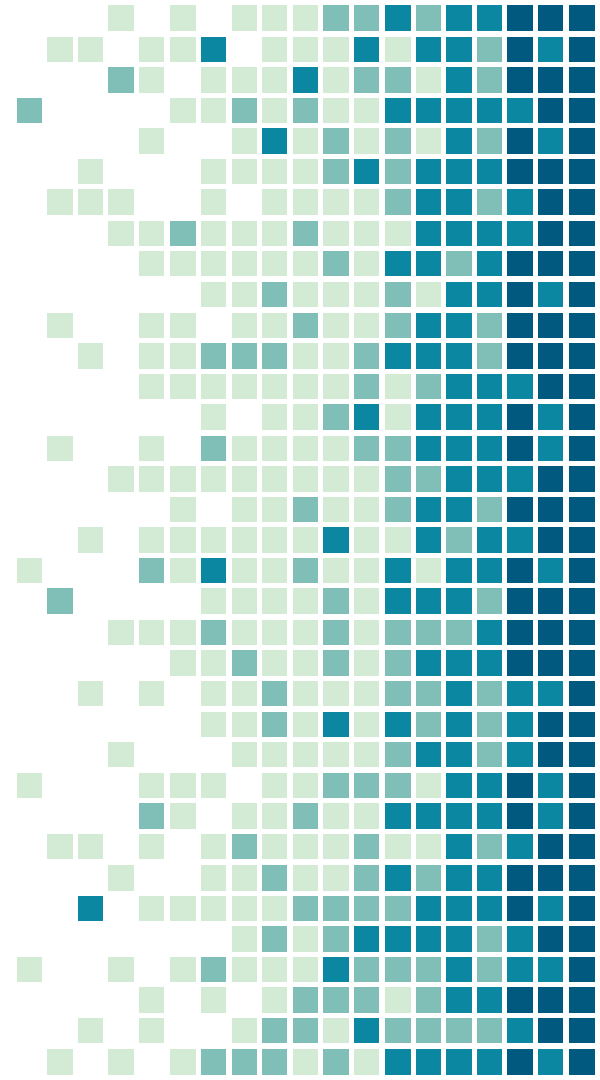
Disconnected from the target VM, address: '127.0.0.1:61261', transport: 'socket'

Process finished with exit code 0
```

## 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.



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

    while (!pila.esVacio()){
        clienteEliminado = pila.eliminar();
        if(clienteEliminado.getGenero().equals("Femenino")){
            aux1.adicionar(clienteEliminado);
        }
        if(clienteEliminado.getGenero().equals("Masculino")){
            aux2.adicionar(clienteEliminado);
        }
    }
    pila.vaciar(aux1);
    pila.vaciar(aux2);
}
```

```
reordenaPila(pila);
```

```
pila.mostrar();
```

```
}
```

```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.3.3\lib\idea_rt.jar=61297:C:\Program Files\JetBrains\IntelliJ IDEA 2022.3.3\bin" -Dfile.encoding=UTF-8
Mostrando la Pila de Clientes

Mostrando Cliente
Nombre: Luis
Apellido: Martines
Edad: 55
Direccion: Avenida 1213
Genero: Masculino

Mostrando Cliente
Nombre: Carlos
Apellido: Rodriguez
Edad: 42
Direccion: Calle 789
Genero: Masculino

Mostrando Cliente
Nombre: Juan
Apellido: Perez
Edad: 35
Direccion: Calle 123
Genero: Masculino

Mostrando Cliente
Nombre: Maria
Apellido: Hernandez
Edad: 23
Direccion: Calle 1011
Genero: Femenino

Mostrando Cliente
Nombre: Ana
Apellido: Perez
Edad: 28
Direccion: Avenida 456
Genero: Femenino

Process finished with exit code 0
```