



PILAS

ESTRUCTURA DE DATOS

Estudiante Ilia Araceli Sarzo Laura

Codigo: SIS14125434

Docente: William Barra




Manejo de Conceptos

1.

¿A que se refiere cuando se habla de estructura de datos?

R.- En ciencias de la computación, una estructura de datos 1 es una forma particular de organizar información en un computador para que pueda ser utilizada de manera eficiente. 2

3 4 Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas.



Manejo de Conceptos

2. ¿CUÁLES SON LOS TIPOS DE ESTRUCTURA QUE EXISTE?

● ARRAYS

● LISTAS ENLAZADAS

● PILAS

● COLAS

● ARBOLES BINARIOS



3.

¿Apoyándose en el link adjunto, explique, porque son útiles las estructura de datos?

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

Manejo de Conceptos

4. ¿Qué es una Pila?

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

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

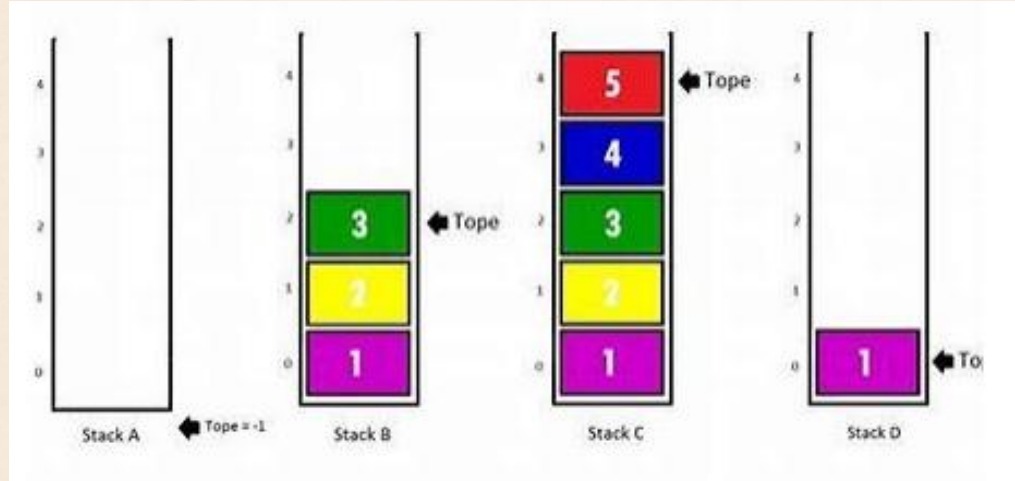
Una pila (stack) es un objeto similar a una pila de platos, donde se puede agregar y sacar datos sólo por el extremo superior.

Manejo de Conceptos

6.

¿Qué es TOPE en una Pila?

Es el numero de elementos que tiene una pila.



Manejo de Conceptos

7. ¿Qué es MAX en una Pila?

Es el numero máximo de elementos que soporta la pila.

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

Si (Tope = 0) Verifica si no hay elementos almacenados en la pila

Entonces

Hacer BAND \leftarrow VERDADERO (la pila es vacía)

Si no

Hacer BAND \leftarrow FALSO (La pila no esta llena)

2.Fin del condicional del paso 1

Manejo de Prácticos

```
$studio_list = array();  
$result = mysql::query("SELECT DISTINCT(studio) as studio, COUNT(*) as count FROM image WHERE day_id = '$day_id' AND enabled='y' GROUP BY studio");  
while($studio_list = mysql::fetch($shots_result)) {  
    $day_info = metadata::day_info($day->shot_date, $studio_list->studio, "quick");  
    $tmp_studio_list[] = array("studio" => $studio_list->studio, "count" => $studio_list->count, "title" => $day_info->title);  
}  
$day->studio_list = $tmp_studio_list;  
$return[$day->shot_date] = $day;
```

```
return $return;
```

```
function day_images_list($date, $studio) {  
    global $global_studio_list;  
    if(!in_array($studio, $global_studio_list)) die("error studio");  
    $date = mysql::escape($date);  
    if(mysql::count("image_date", "shot_date = '$date'") < 1) die('date not found');  
    $studio = intval($studio);
```

```
$return = array();
```

```
$result = mysql::query("SELECT image.id as image_id, image, image_date WHERE image_date.id=image.day_id AND image_date.shot_date='$date' AND image.enabled='y' AND i");  
while($image = mysql::fetch($result)) {  
    $image->copyright = metadata::get_copyright($image->image_id);  
    $image->models = metadata::get_models($image->image_id);  
    $return[$image->image_id] = $image;
```

1273

1274

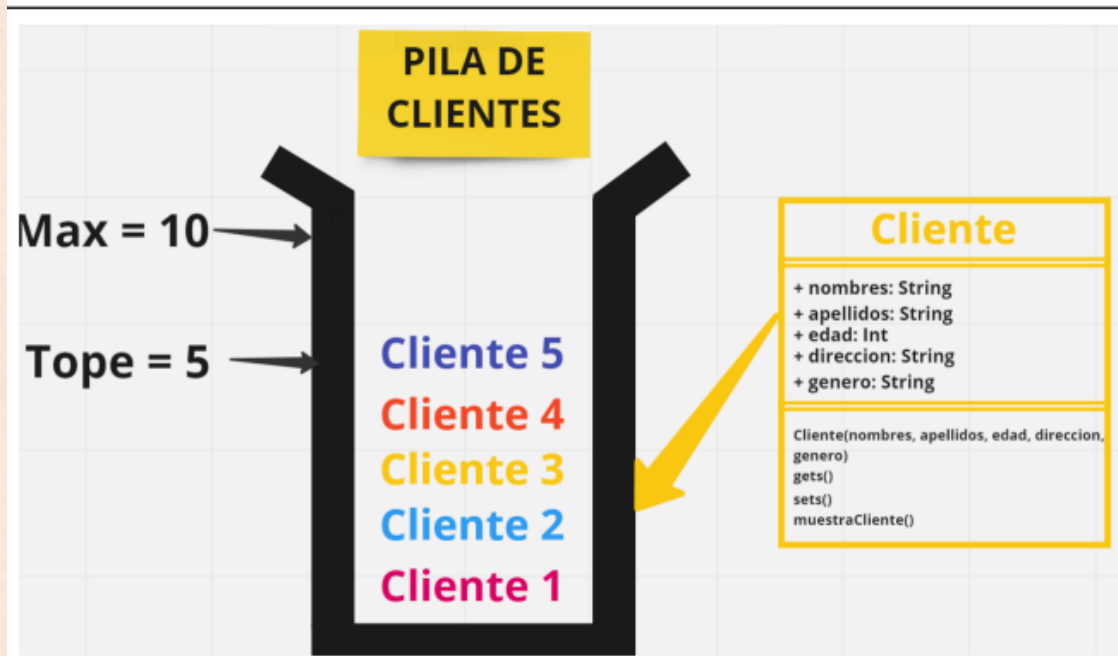
1275

```
function day_list() {
```


11.

Manejo de Conceptos

Crear las clases necesarias para la PILA DE CLIENTES



11.

Manejo de Conceptos

```
package PilasDeClientes;

public class Cliente {
    4 usages
    private String nombre;
    4 usages
    private String apellidos;
    4 usages
    private int edad;
    4 usages
    private String direccion;
    4 usages
    private String genero;

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

    public Cliente(){
        this.nombre = "";
        this.apellidos = "";
        this.edad = 0;
        this.direccion = "";
        this.genero = "";
    }

    public String getNombre() {
        return nombre;
    }

    1 usage
    public String getApellidos() {
        return apellidos;
    }
}
```

```
1 usage
public int getEdad() {
    return edad;
}

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

1 usage
public String getGenero() {
    return genero;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

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;
}
}
```

11.

Manejo de Conceptos

```
public void mostrarCliente(){  
    System.out.println("\nMostrando datos del jugador");  
    System.out.println("Nombre: " + this.getNombre());  
    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");  
}
```

11.

Manejo de Conceptos

```
package PilasDeClientes;

public class PilaCliente {
    4 usages
    private int max;
    10 usages
    private int tope;
    3 usages
    private Cliente[] Clientes;

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

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

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

```
}

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

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

2 usages
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 vacia");
    }

    return elementoEliminado;
}

public void llenar () {
}
```

11.

Manejo de Conceptos

```
1 }
2
3 1 usage
4 public void mostrar () {
5     Cliente elem = null;
6     if (esVacio())
7         System.out.println("Pila Vacía");
8     else {
9         System.out.println(" Datos de la Pila de Clientes");
10        PilaCliente aux = new PilaCliente(this.max);
11        while (!esVacio()) {
12            elem = this.eliminar();
13            aux.adicionar (elem);
14            elem.mostrarCliente();
15        }
16        vaciar(aux);
17    }
18 }
19
20 1 usage
21 public void vaciar (PilaCliente pila) {
22     while (!pila.esVacio()) {
23         adicionar(pila.eliminar());
24     }
25 }
26 }
```

11.

Manejo de Conceptos

```
Cliente.java x PilaCliente.java x Main.java x
1 package PilasDeClientes;
2
3 public class Main {
4     public static void main( String[] arg){
5         Cliente cli1 = new Cliente( nombre: "Micaela", apellidos: "Romero", edad: 20, direccion: "LaPaz", genero: "Femenino");
6         Cliente cli2 = new Cliente( nombre: "Iris", apellidos: "Velasco", edad: 19, direccion: "Cochabamba", genero: "Femenino");
7         Cliente cli3 = new Cliente( nombre: "Miguel", apellidos: "Estrada", edad: 23, direccion: "Oruro", genero: "Masculino");
8         Cliente cli4 = new Cliente( nombre: "Andres", apellidos: "Quiroga", edad: 22, direccion: "LaPaz", genero: "Masculino");
9         Cliente cli5 = new Cliente( nombre: "Josias", apellidos: "Leon", edad: 21, direccion: "Tarija", genero: "Masculino");
10
11
12         PilaCliente pila = new PilaCliente( max: 10);
13         pila.adicionar(cli1);
14         pila.adicionar(cli2);
15         pila.adicionar(cli3);
16         pila.adicionar(cli4);
17         pila.adicionar(cli5);
18
19         pila.mostrar();
20
21     }
22 }
23
```


11.

Manejo de Conceptos

Mostrando datos del jugador

Nombre: Josias

Apellidos: Leon

Edad 21

Direcccion: Tarija

Genero: Masculino

Mostrando datos del jugador

Nombre: Andres

Apellidos: Quiroga

Edad 22

Direcccion: LaPaz

Genero: Masculino

Mostrando datos del jugador

Nombre: Miguel

Apellidos: Estrada

Edad 23

Direcccion: Oruro

Genero: Masculino

Mostrando datos del jugador

Nombre: Miguel

Apellidos: Estrada

Edad 23

Direcccion: Oruro

Genero: Masculino

Mostrando datos del jugador

Nombre: Iris

Apellidos: Velasco

Edad 19

Direcccion: Cochabamba

Genero: Femenino

Mostrando datos del jugador

Nombre: Micaela

Apellidos: Romero

Edad 20

Direcccion: LaPaz

Genero: Femenino

Process finished with exit code 0

12.

Manejo de Conceptos

Determinar cuántos CLIENTES son mayores de 20 años.

```
// determinar cuantos clientes son mayores de 20 años
public static void MayorCiertaEdad(PilaCliente pila, int edadMayor){
    PilaCliente aux = new PilaCliente( max: 10);
    int MayorEdad = 0;
    Cliente Valorestraido = null;
    while (pila.esVacio() == false){
        Valorestraido = pila.eliminar();
        if (Valorestraido.getEdad() > edadMayor){
            MayorEdad = MayorEdad +1;
        }
        aux.adicionar(Valorestraido);
    }
    System.out.println("Clientes de mas de " + edadMayor + "son: " + MayorEdad);
}

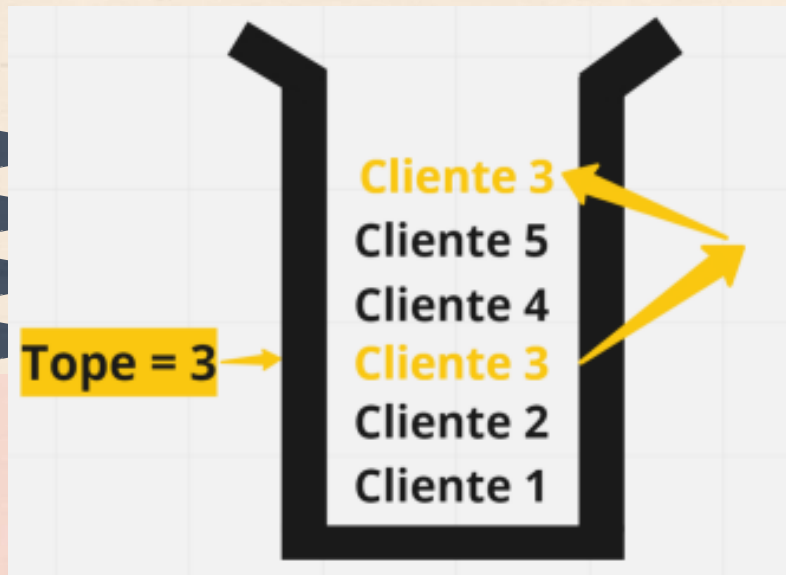
public static void kEsimoPosicion(PilaCliente pila, int valorTope){
    PilaCliente aux = new PilaCliente( max: 10);
    Cliente valor = null;

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

Manejo de Conceptos

13.

Mover el k-ésimo elemento al final de la pila



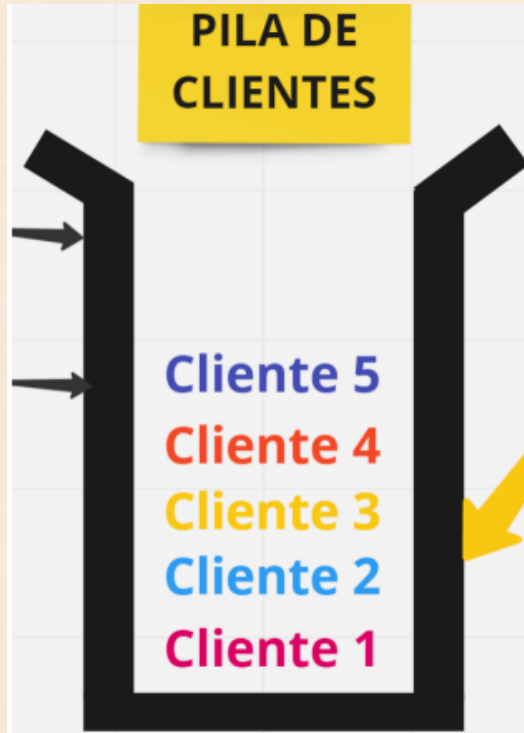
```
// cambiar la direccion de algunos clientes de la pila

1 usage
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();
}
```

14.

Manejo de Conceptos

Cambiar la dirección de algunos CLIENTES de la PILA.



```
// cambiar la direccion de algunos clientes de la pila

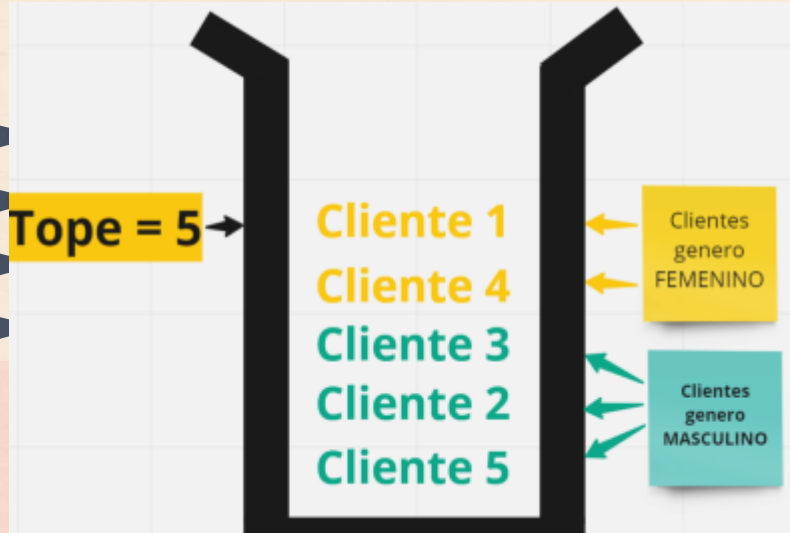
1 usage
public static void asignaDireccion (PilaCliente pila, String nuevaDireccion) {
    PilaCliente aux = new PilaCliente( max: 10);
    Cliente valor = null;
    while (!pila.esVacio()) {
        valor = pila.eliminar();
        if(valor.getGenero() == "Femenino"){
            valor.setDireccion(nuevaDireccion);
            aux.adicionar(valor);
        }
        else{
            aux.adicionar(valor);
        }
    }
    pila.vaciar(aux);
    pila.adicionar(valor);
    pila.mostrar();
}

// Nuevo ITEMS de la Pila
```

15.

Manejo de Conceptos

Mover ÍTEMS de la PILA.



Manejo de Conceptos

```
// Mover ITEMS 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();
    }
}
```

GRACIAS POR SU ATENCION



eate.iliaaraceli.sarzo.la@unifranz.edu.bo

[Yilialaura.tiktok](https://www.tiktok.com/Yilialaura.tiktok)