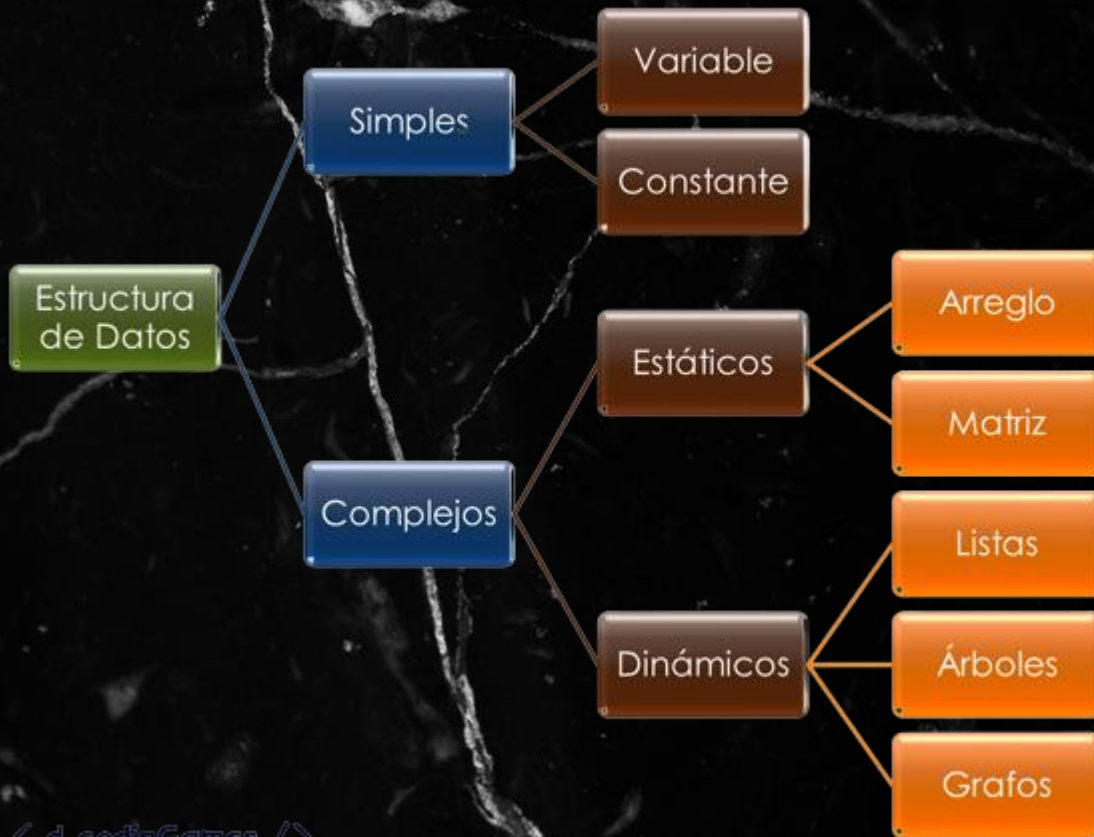


Evaluación Procesual Hito 3

Estudiante: Heber Mollericona Miranda

Cl: 13409189

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



La estructura de datos se refiere a la forma en que se organizan y almacenan los datos en un sistema informático. Es decir, se trata de un conjunto de herramientas, técnicas y metodologías para organizar y manipular datos de manera eficiente y efectiva.

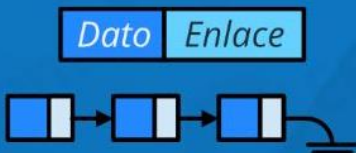
¿Cuáles son los TIPOS DE ESTRUCTURA QUE EXISTE?

ARREGLOS



Almacenan múltiples datos en una sola variable

LISTAS ENLAZADAS



Se componen de nodos que tienen dos atributos(dato y enlace).

PILAS



Solamente pueden eliminar o insertar en la cima de la lista.

COLAS



Permite emular el comportamiento de una fila o cola de la vida real.

- Arreglos: Los elementos se almacenan en una secuencia ordenada y se acceden mediante un índice numérico.
- Listas: Las listas se caracterizan por su flexibilidad y su capacidad para agregar, eliminar y modificar elementos en cualquier posición.
- Pilas: Las pilas funcionan según el principio "último en entrar, primero en salir" (LIFO).
- Colas: Las colas funcionan según el principio "primero en entrar, primero en salir" (FIFO).
- Árboles: Cada nodo en un árbol tiene un padre y cero o más hijos, y los nodos se organizan en una estructura jerárquica.
- Grafos: Los nodos en un grafo se denominan vértices y las relaciones entre ellos se denominan aristas.

¿Qué es una PILA?

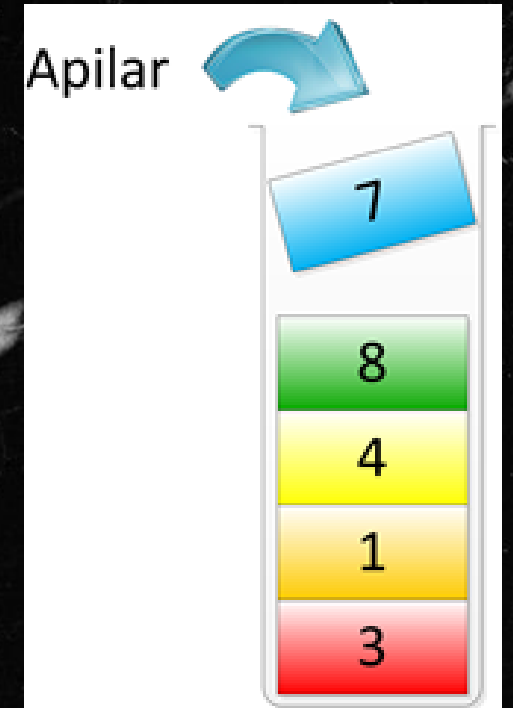
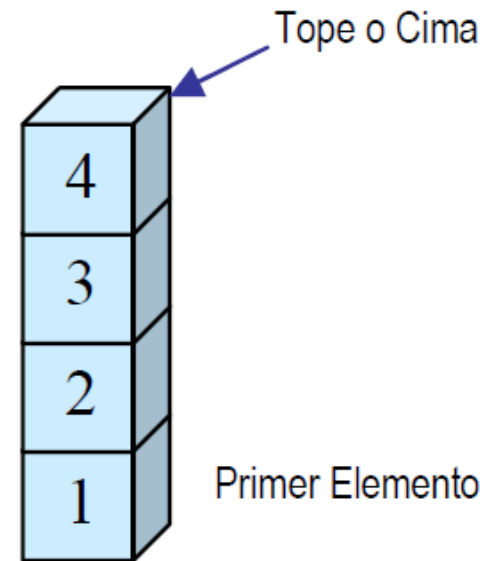
Las pilas son estructuras de datos que se utilizan para almacenar elementos en una secuencia específica. Las pilas funcionan según el principio "último en entrar, primero en salir" (LIFO).

Pilas



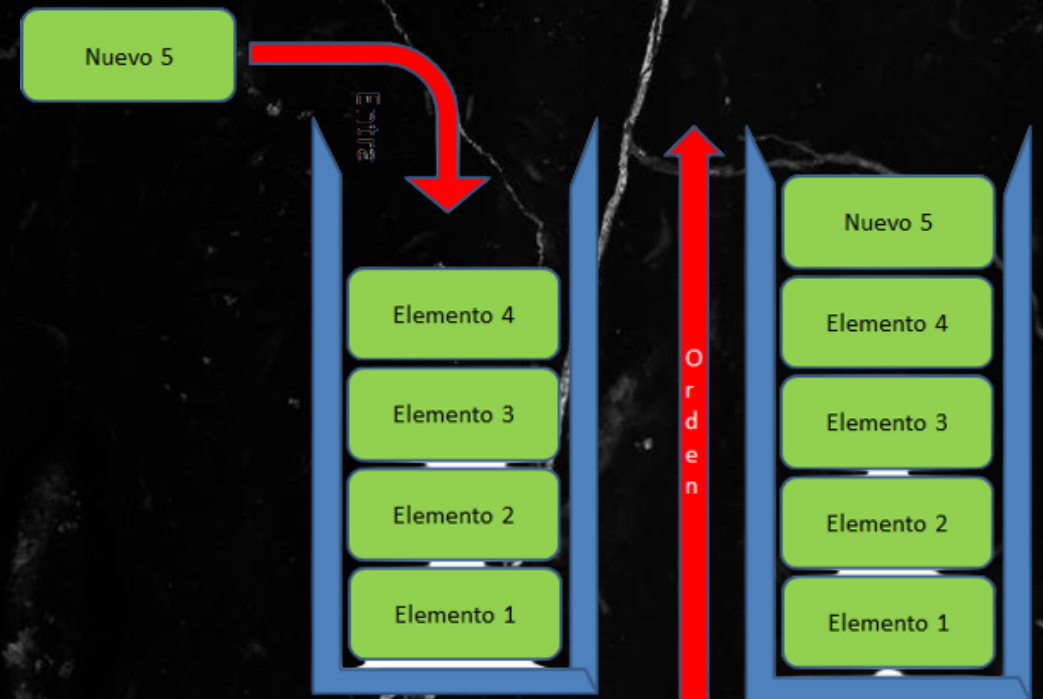
```
public class Nodo {
    private int valor;
    private Nodo siguiente;
    public void setValor(int valor) {
        this.valor = valor;
    }
    public void setSiguiente(Nodo siguiente) {
        this.siguiente = siguiente;
    }
    public int getValor() {
        return this.valor;
    }
    public Nodo getSiguiente() {
        return this.siguiente;
    }
}

public class ListaCircular {
    private Nodo inicio;
    private Nodo fin;
    private int tamaño;
    public void agregar(Nodo nuevo) {
        if (inicio == null) {
            inicio = nuevo;
            fin = nuevo;
        } else {
            fin.siguiente = nuevo;
            fin = nuevo;
        }
        tamaño++;
    }
    public void listar() {
        if (inicio == null) {
            return;
        }
        Nodo aux = inicio;
        while (aux != null) {
            System.out.print(aux.getValor() + " ");
            aux = aux.getSiguiente();
        }
        System.out.println();
    }
    public void borrar() {
        if (inicio == null) {
            return;
        }
        if (inicio == fin) {
            inicio = null;
            fin = null;
        } else {
            Nodo aux = inicio;
            while (aux.siguiente != fin) {
                aux = aux.siguiente;
            }
            aux.siguiente = inicio;
        }
        tamaño--;
    }
    public void resetear() {
        inicio = null;
        fin = null;
        tamaño = 0;
    }
    public void buscar(int valor) {
        if (inicio == null) {
            return;
        }
        Nodo aux = inicio;
        while (aux != null) {
            if (aux.getValor() == valor) {
                System.out.println("Encontrado el valor " + valor);
                return;
            }
            aux = aux.getSiguiente();
        }
        System.out.println("No se encontró el valor " + valor);
    }
    public void mostrar() {
        System.out.println("Inicio: " + inicio.getValor());
        System.out.println("Fin: " + fin.getValor());
        System.out.println("Tamaño: " + tamaño);
    }
}
```



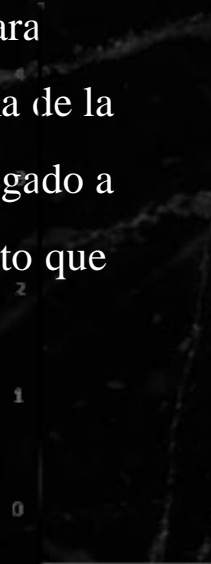
¿Qué es STACK en JAVA, una STACK será lo mismo que una PILA?

Una Stack en Java proporciona los métodos `push()` y `pop()` para agregar y eliminar elementos de la pila, respectivamente, así como otros métodos útiles como `peek()` para obtener el elemento en la parte superior de la pila sin eliminarlo, `empty()` para verificar si la pila está vacía y `search()` para buscar la posición de un elemento en la pila.



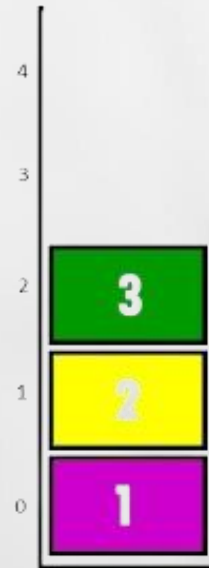
¿Qué es TOPE en una PILA?

"TOPE" (o "top") es el término que se utiliza para referirse al elemento que se encuentra en la cima de la pila, es decir, el último elemento que se ha agregado a la misma. El tope de la pila es el primer elemento que se elimina cuando se realiza una operación de eliminación (pop) en la pila.



Stack A

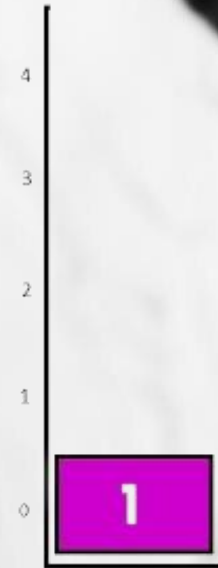
← Tope = -1



Stack B



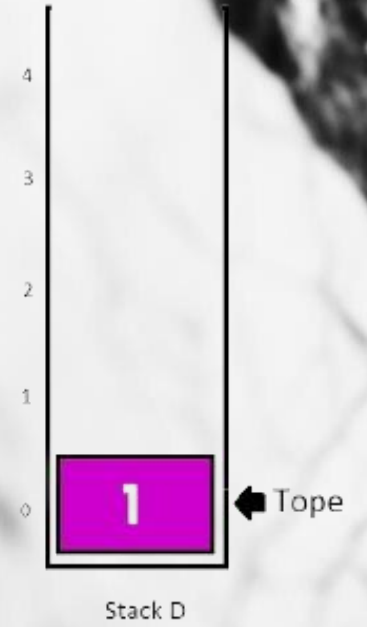
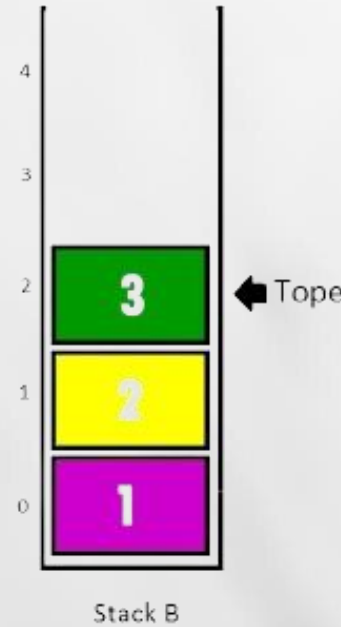
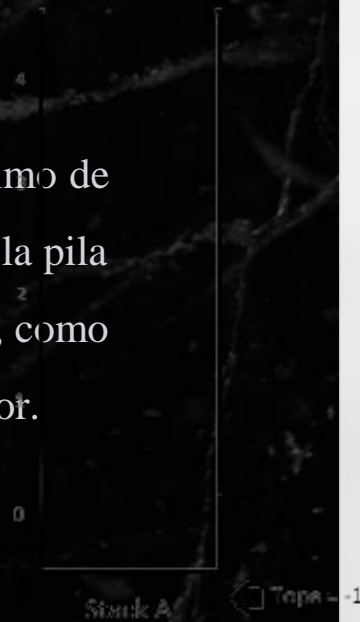
Stack C



Stack D

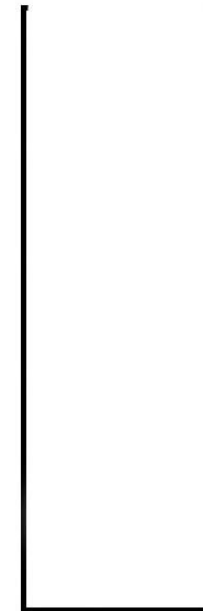
¿Qué es MAX en una PILA?

"MAX" puede referirse al número máximo de elementos que se pueden almacenar en la pila o al elemento de mayor valor en la pila, como se mencionó en mi respuesta anterior.



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

- El método "esVacia()" se utiliza para comprobar si la pila no contiene elementos. Este método devuelve un valor booleano que indica si la pila está vacía o no. Si la pila está vacía, el método devuelve "true", y si la pila contiene al menos un elemento, devuelve "false".
- El método "esLLena()" se utiliza para comprobar si la pila ha alcanzado su capacidad máxima. Este método también devuelve un valor booleano que indica si la pila está llena o no. Si la pila está llena, el método devuelve "true", y si todavía hay espacio disponible para agregar elementos, devuelve "false".



Pila vacía



Pila Llena

¿Qué son los métodos estáticos en JAVA?

En Java, un método estático es un método que se asocia con la clase y no con una instancia particular de la misma. Esto significa que se puede invocar el método estático sin tener que crear un objeto de la clase.

```
public static void mayoresCiertaEdad(PilaClientes pila, int edad){
    PilaClientes aux = new PilaClientes();

    Clientes item = null;
    int cont = 0;

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

        if(item.getEdad() > edad){
            cont++;
        }

        aux.adicionar(item);
    }
    pila.vaciar(aux);
    System.out.println("Hay " + cont + " clientes con la edad mayor a " + edad);
}
```

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

PilaClientes		
m	PilaClientes ()	
f	max	int
f	tope	int
f	cliente	Clientes []
m	eliminar ()	Clientes
m	esVacio ()	boolean
m	esLleno ()	boolean
m	adicionar (Clientes)	void
m	nroElem ()	int
m	vaciar (PilaClientes)	void
m	mostrar ()	void
m	llenar (int)	void

. Crear las clases necesarias para la PILA DE CLIENTES

```
public static void main(String[] args) {  
    Clientes cliente1 = new Clientes( nombres: "Adrian", apellidos: "Condori", edad: 18, direccion: "Plaza Murillo", genero: "Masculino");  
    Clientes cliente2 = new Clientes( nombres: "Matias", apellidos: "Chambi", edad: 32, direccion: "Calle 2", genero: "Masculino");  
    Clientes cliente3 = new Clientes( nombres: "Helen", apellidos: "Rodriguez", edad: 46, direccion: "Calle Gregorio Lanza", genero: "Femenino");  
    Clientes cliente4 = new Clientes( nombres: "Daniela", apellidos: "Leuca", edad: 20, direccion: "Calle Linares", genero: "Femenino");  
    Clientes cliente5 = new Clientes( nombres: "Kevin", apellidos: "Gonzales", edad: 29, direccion: "Av. Juan Pablo II", genero: "Masculino");  
  
    PilaClientes pila = new PilaClientes();  
    pila.adicionar(cliente1);  
    pila.adicionar(cliente2);  
    pila.adicionar(cliente3);  
    pila.adicionar(cliente4);  
    pila.adicionar(cliente5);  
}
```

Mostrando la Pila De Clientes:

Datos de Cliente:

Nombre: Daniela

Apellido: Leuca

Edad: 20

Direccion: Calle Linares

Genero: Femenino

Datos de Cliente:

Nombre: Helen

Apellido: Rodriguez

Edad: 46

Direccion: Calle Gregorio Lanza

Genero: Femenino

Datos de Cliente:

Nombre: Kevin

Apellido: Gonzales

Edad: 29

Direccion: Av. Juan Pablo II

Genero: Masculino

Datos de Cliente:

Nombre: Matias

Apellido: Chambi

Edad: 32

Direccion: Calle 2

Genero: Masculino

Datos de Cliente:

Nombre: Adrian

Apellido: Condori

Edad: 18

Direccion: Plaza Murillo

Genero: Masculino

Determinar cuántos CLIENTES son mayores de 20 años.

```
public static void mayoresCiertaEdad(PilaClientes pila, int edad){
    PilaClientes aux = new PilaClientes();

    Clientes itemCliente = null;
    int cont = 0;

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

        if(itemCliente.getEdad() > edad){
            cont++;
        }

        aux.adicionar(itemCliente);
    }
    pila.vaciar(aux);
    System.out.println("Existen "+cont+" clientes con edad mayor a "+edad);
}
```

```
C:\Users\ASUS\.jdk\openjdk-19.0.2\bin\java.exe
Existen 3 clientes con edad mayor a 25
```

Mover el k-ésimo elemento al final de la pila.

```
public static void kEsimoPosicion(PilaClientes pila, int posicion){
    PilaClientes aux = new PilaClientes();
    Clientes itemCliente = null;
    Clientes saveCl = null;

    while (!pila.esVacio()){
        itemCliente = pila.eliminar();
        if(pila.nroElem()+1 == posicion){
            saveCl = itemCliente;
        }else {
            aux.adicionar(itemCliente);
        }
    }
    pila.vaciar(aux);
    pila.adicionar(saveCl);
}
```

Mostrando la Pila De Clientes:

Datos de Cliente:

Nombre: Matias

Apellido: Chambi

Edad: 32

Direccion: Calle 2

Genero: Masculino

Datos de Cliente:

Nombre: Kevin

Apellido: Gonzales

Edad: 29

Direccion: Av. Juan Pablo II

Genero: Masculino

Datos de Cliente:

Nombre: Daniela

Apellido: Levca

Edad: 20

Direccion: Calle Linares

Genero: Femenino

Datos de Cliente:

Nombre: Helen

Apellido: Rodriguez

Edad: 46

Direccion: Calle Gregorio Lanza

Genero: Femenino

Datos de Cliente:

Nombre: Adrian

Apellido: Condori

Edad: 18

Direccion: Plaza Murillo

Genero: Masculino

Cambiar la dirección de algunos CLIENTES de la PILA

```
public static void asignaDireccion(PilaClientes pila, String nuevaDireccion){
    PilaClientes aux = new PilaClientes();

    Clientes itemCliente = null;
    while (!pila.esVacio()){
        itemCliente = pila.eliminar();

        if(itemCliente.getGenero().equals("Femenino")){
            itemCliente.setDireccion(nuevaDireccion);
            aux.adicionar(itemCliente);
        }
        else {
            aux.adicionar(itemCliente);
        }
    }
    pila.vaciar(aux);
}
```

Mostrando la Pila De Clientes:

Datos de Cliente:

Nombre: Kevin

Apellido: Gonzales

Edad: 29

Direccion: Av. Juan Pablo II

Genero: Masculino

Datos de Cliente:

Nombre: Daniela

Apellido: Leuca

Edad: 20

Direccion: Calle 7

Genero: Femenino

Datos de Cliente:

Nombre: Helen

Apellido: Rodriguez

Edad: 46

Direccion: Calle 7

Genero: Femenino

Datos de Cliente:

Nombre: Matias

Apellido: Chambi

Edad: 32

Direccion: Calle 2

Genero: Masculino

Datos de Cliente:

Nombre: Adrian

Apellido: Condori

Edad: 18

Direccion: Plaza Murillo

Genero: Masculino

Mover ÍTEMS de la PILA

```
public static void reordenaPila(PilaClientes pila){
    PilaClientes auxGenMas = new PilaClientes();
    PilaClientes auxGenFem = new PilaClientes();

    Clientes item = null;

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

        if(item.getGenero().equals("Masculino")){
            auxGenMas.adicionar(item);
        }else{
            auxGenFem.adicionar(item);
        }
    }
    pila.vaciar(auxGenMas);
    pila.vaciar(auxGenFem);
}
```

Mostrando la Pila De Clientes:

Datos de Cliente:

Nombre: Daniela

Apellido: Leuca

Edad: 20

Direccion: Calle Linares

Genero: Femenino

Datos de Cliente:

Nombre: Helen

Apellido: Rodriguez

Edad: 46

Direccion: Calle Gregorio Lanza

Genero: Femenino

Datos de Cliente:

Nombre: Kevin

Apellido: Gonzales

Edad: 29

Direccion: Av. Juan Pablo II

Genero: Masculino

Datos de Cliente:

Nombre: Matias

Apellido: Chambi

Edad: 32

Direccion: Calle 2

Genero: Masculino

Datos de Cliente:

Nombre: Adrian

Apellido: Condori

Edad: 18

Direccion: Plaza Murillo

Genero: Masculino