



/DEFENSA FINAL HITO 3

ESTRUCTURA DE DATOS II

EDSON IVER CONDORI CONDORI
SIS10929449



EL ALTO, MAYO DE 2023

UNIFRANZ
Internacionalízate

INNOVACIÓN
EN EDUCACIÓN



/CONTENIDO (EDD).



/01

/PARTE TEORICA



Se explicara
conceptos
elementales de
introduccion a
DBAII.



/02

/PARTE PRACTICA



Se presentara la
aplicacion de la
parte teorica para
la resolucion de los
requerimientos.





/COMENCAMOS!

> /EDD



/01 /PARTE TEORICA



/CONCEPTOS



1. ¿A QUE SE REFIERE CUANDO SE HABLA DE ESTRUCTURA DE DATOS?

Nos referimos a cómo se organizan y almacenan los datos en un sistema informático para que puedan ser procesados y utilizados de manera efectiva en un programa.



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

Existen varios pero los mas comunes son los siguientes:

1. Arreglos: almacena elementos en una secuencia y se accede a ellos mediante un índice.
2. Listas Enlazadas: almacena elementos en nodos que tienen referencias a los siguientes elementos en la lista.
3. Pilas: El último elemento agregado es el primero en ser retirado.
4. Colas: El primer elemento agregado es el primero en ser retirado.
5. Árboles: Organiza elementos jerárquicamente en una estructura de árbol con un nodo raíz y varios nodos secundarios.

/CONCEPTOS



3. ¿POR QUÉ SON ÚTILES LAS ESTRUCTURAS DE DATOS?

1. Nos permiten tener una variedad de herramientas para solucionar ciertos tipos de problemas.
2. Nos permiten hacer un software más eficiente optimizando recursos.



4. ¿QUÉ ES UNA PILA?

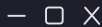
Una pila es una estructura de datos en la que el último elemento agregado es el primero en ser retirado.

Se puede agregar y eliminar elementos solo desde la cima de la pila.



5. ¿QUÉ ES STACK EN JAVA, UNA STACK SERÁ LO MISMO QUE UNA PILA?

En si tanto Stack y Pila son lo mismo en estructura de datos, ambos tienen el mismo principio donde el primero en ingresar es el ultimo en salir.



/CONCEPTOS



6. ¿QUÉ ES TOPE EN UNA PILA?

Podríamos decir que el tope en una pila es la cantidad de elementos actuales que se encuentran dentro de esta.



7. ¿QUÉ ES MAX EN UNA PILA?

Se diría que el MAX en una pila es la cantidad de elementos máximos que puede recibir o soportar dicha pila.



8. ¿A QUE SE REFIERE LOS MÉTODOS `EsVACIA()` Y `EsLLENA()` EN UNA PILA?

Metodo `EsVacia`: Se utiliza para verificar si una pila está vacía o si tiene elementos.

Metodo `EsLlena`: Se utiliza para verificar si una pila está llena.



/CONCEPTOS



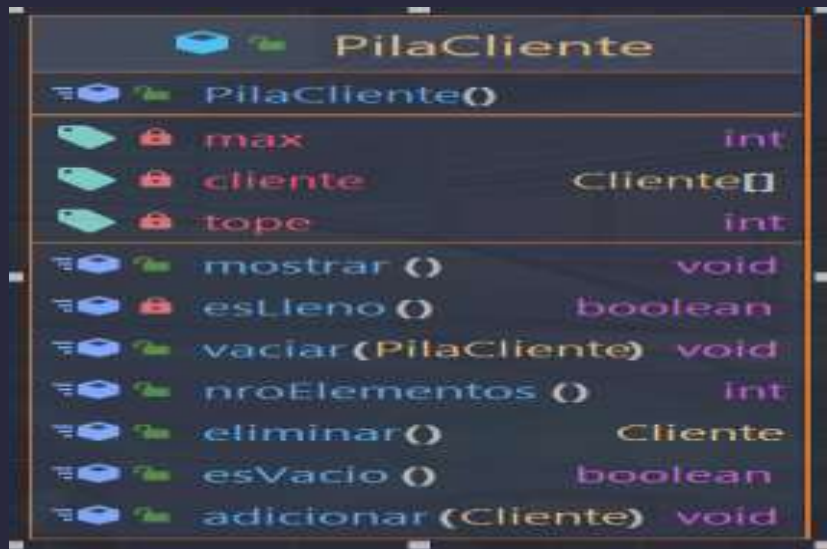
9. ¿QUÉ SON LOS MÉTODOS ESTÁTICOS EN JAVA?

Son métodos que pertenecen a la clase en sí, en lugar de pertenecer a una instancia particular de la clase.

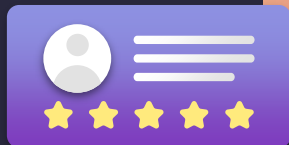
Pueden ser utilizados sin crear un objeto de la clase y se definen utilizando la palabra clave "static".



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



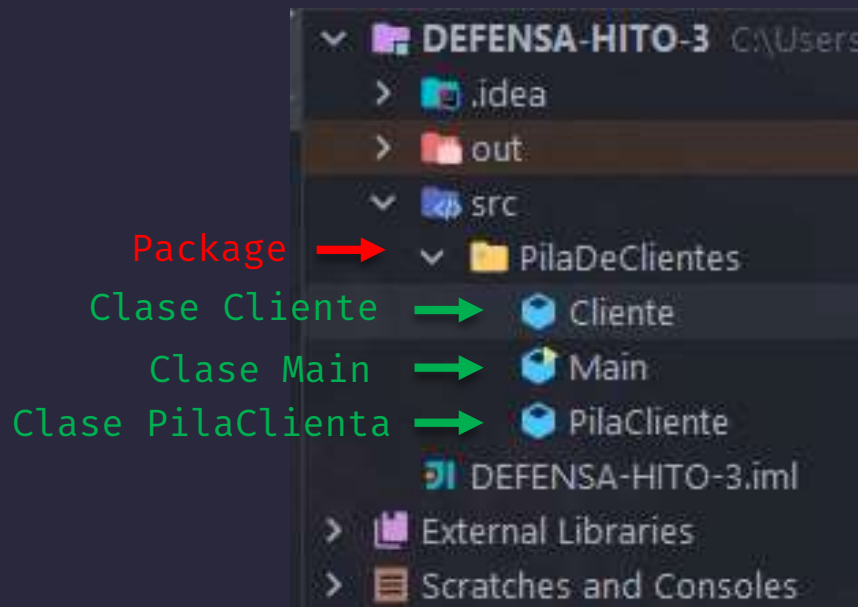
PilaCliente	
PilaCliente()	
max	int
cliente	Cliente[]
tope	int
mostrar ()	void
esLleno ()	boolean
vaciar(PilaCliente)	void
nroElementos ()	int
eliminar()	Cliente
esVacio ()	boolean
adicionar(Cliente)	void



/02 /PARTE PRACTICA



/ 11. Crear las clases necesarias para la PILA DE CLIENTES. ●●●



/SUGERENCIAS

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)

/La clase MAIN con la creación de 5 clientes y agregados a la PILA

Creación de Clientes

Agregado a la Pila

```

DEFENSA-HITO-3 - src - PilaDeClientes - Main - asignaDireccion
Cliente.java x PilaCliente.java x Main.java x

public class Main {
    no usages

    public static void main(String[] args) {

        // EJERCICIO 1

        Cliente client1 = new Cliente(nombre: "Carla", apellido: "Mendoza", edad: 18, direccion: "Villa Fatima", genero: "Femenino");
        Cliente client2 = new Cliente(nombre: "Juan", apellido: "Gonzalez", edad: 27, direccion: "Mercedario", genero: "Masculino");
        Cliente client3 = new Cliente(nombre: "Oscar", apellido: "Choque", edad: 25, direccion: "Rio Seco", genero: "Masculino");
        Cliente client4 = new Cliente(nombre: "Ximena", apellido: "Ramos", edad: 19, direccion: "Alto Tejer", genero: "Femenino");
        Cliente client5 = new Cliente(nombre: "Pablo", apellido: "Gutierrez", edad: 28, direccion: "Colacoto", genero: "Masculino");

        PilaCliente pila = new PilaCliente();

        pila.adicionar(client1);
        pila.adicionar(client2);
        pila.adicionar(client3);
        pila.adicionar(client4);
        pila.adicionar(client5);

        // EJERCICIO 1
        System.out.println("=====");
        System.out.println("===== EJERCICIO 1 =====");
        System.out.println("=====");

        pila.mostrar();
    }
}

```

/ Imagen de la salida de la consola en donde se muestran todos los ítems de la pila

```
Cliente.java x PilaCliente.java x Main.java x
3 ▶ public class Main {
Main x
C:\Users\MAQ\.jdk\openjdk-19.0.2\bin\java.exe
===== EJRCICIO 1 =====
Mostrando la pila de Clientes
Mostrando Cliente

Nombre: Pablo
Apellido: Gutierrez
Edad: 28
Direccion: Calacoto
Genero: Masculino
} Cliente 1

Mostrando Cliente

Nombre: Ximena
Apellido: Ramos
Edad: 19
Direccion: Alto Tejar
Genero: Femenino
} Cliente 2
```

```
Cliente.java x PilaCliente.java x
3 ▶ public class Main {
Main x
Mostrando Cliente

Nombre: Oscar
Apellido: Choque
Edad: 25
Direccion: Rio Seco
Genero: Masculino
} Cliente 3

Mostrando Cliente

Nombre: Juan
Apellido: Gonzales
Edad: 27
Direccion: Mercedario
Genero: Masculino
} Cliente 4
```

```
Cliente 5
Mostrando Cliente

Nombre: Carla
Apellido: Mendoza
Edad: 18
Direccion: Villa Fatima
Genero: Femenino
```

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

```
// EJERCICIO 2
// DETERMINAR CUANTOS CLIENTES TIENEN UNA EDAD MAYOR DE 20
1 usage
public static void EdadMayor_20(PilaCliente pila, int edad) {
```

```
    PilaCliente aux = new PilaCliente();
    Cliente item = null;
    int cont = 0;
```

```
    while (!pila.esVacio()) {
        item = pila.eliminar();
        if (item.getEdad() > edad) {
            cont = cont + 1;
        }
    }
```

```
    aux.adicionar(item);
```

```
}
```

```
pila.vaciar(aux);
```

```
System.out.printf("\nClientes con una edad mayor a " + edad + " años: " + cont);
```

```
}
```

Código que
Resuelve el
Problema

/SUGERENCIAS

- o El método deberá llamarse mayoresCiertaEdad(Pila, edadMayor)
- o El método debe ser creado en la clase MAIN como un método estático.
- o El método recibe 2 parámetros
 - La Pila de Clientes
 - El valor de la edad.

/ Imagen de la salida de la consola.

LLAMANDO AL METODO

```
// EJERCICIO 2
// Determinar cuántos CLIENTES son mayores de 20 años.
// O El método deberá llamarse mayoresCiertaEdad(Pila, edadMayor)
// O El método debe ser creado en la clase MAIN como un método estático.
// O El método/ recibe 2 parámetros
//■ La Pila de Clientes
//■ El valor de la edad.

System.out.println("-----");
System.out.println("===== EJERCICIO 2 =====");
System.out.println("-----");
EdadMayor_20(pila, edad: 20);
```

SALIDA DE CONSOLA

-HITO-3 > src > PilaDeClientes > Main > asignaDireccion

Client.java x Pi

Main x

```
-----
===== EJERCICIO 2 =====
-----

Clientes con una edad mayor a 20 años: 3
```

/ 13.Mover el k-ésimo elemento al final de la pila.

asignaDireccion

PilaCliente.java × Main.java ×

Código que Resuelve el Problema

```
// EJERCICIO 3
// Mover el kEsimoPosicion elemento al final de la pila.
1 usage
public static void kEsimoPosicion(PilaCliente pila, int valorTope) {
    PilaCliente aux = new PilaCliente();
    Cliente nombKesimo = null; // almacena el cliente del kEsimo elemento
    Cliente clienteBorado; // almacena temporalmente el cliente de cada elemento

    while (!pila.esVacio()) {
        clienteBorado = pila.eliminar();
        if (pila.nroElementos() + 1 == valorTope) {
            nombKesimo = clienteBorado;
        } else {
            aux.adicionar(clienteBorado);
        }
    }

    pila.vaciar(aux);
    pila.adicionar(nombKesimo);
}
```

/SUGERENCIAS

- o El método deberá llamarse kEsimoPosicion(Pila, valorTope)
- o El método debe ser creado en la clase MAIN como un método estático.
- o El método recibe 2 parámetros
 - La Pila de Clientes
 - El valor(int) de la posición que moverá al final de la pila.



/ Llamando al Método



LLAMANDO AL METODO

```
// EJERCICIO 3  
System.out.println("\n\n+++++");  
System.out.println("===== EJERCICIO 3 =====");  
System.out.println("+++++");  
  
kEsimoPosicion(pila, valorTope: 3);  
pila.mostrar();
```



/ Imagen de la salida de la consola.

```

Main x
+++++
===== EJRCICIO 3 =====
+++++

Mostrando la pila de Clientes
Mostrando Cliente

Nombre: Oscar
Apellido: Choque
Edad: 25
Direccion: Rio Seco
Genero: Masculino

Mostrando Cliente

Nombre: Pablo
Apellido: Gutierrez
Edad: 28
Direccion: Calacoto
Genero: Masculino

```

Cliente
kEsimo
Movido

```

Mostrando Cliente

Nombre: Ximena
Apellido: Ramos
Edad: 19
Direccion: Alto Tejar
Genero: Femenino

Mostrando Cliente

Nombre: Juan
Apellido: Gonzales
Edad: 27
Direccion: Mercedario
Genero: Masculino

Mostrando Cliente

Nombre: Carla
Apellido: Mendoza
Edad: 18
Direccion: Villa Fatima
Genero: Femenino

```

/ 14.Cambiar la dirección de algunos CLIENTES de la PILA.

Código que Resuelve el Problema

```
03 // EJERCICIO 4
04 // Cambiar la dirección de algunos CLIENTES de la PILA.
05 // SIEMPRE Y CUANDO ESTOS CLIENTES SEAN DEL GENERO FEMENINO
06 // usage
07 @ public static void asignaDireccion(PilaCliente pila, String NuevaDireccion){
08
09     PilaCliente aux = new PilaCliente();
10     Cliente item = null;
11     int cont = 0;
12
13     while (!pila.esVacio())
14     {
15         item = pila.eliminar();
16         if(item.getGenero()=="Femenino"){
17             cont = cont +1;
18         }
19         if (item.getGenero()=="Femenino"){
20             item.setDireccion(NuevaDireccion);
21         }
22         aux.adicionar(item);
23     }
24     pila.vaciar(aux);
25 }
```

/SUGERENCIAS

- o El método deberá llamarse `asignaDireccion(Pila, nuevaDireccion)`

- o El método debe ser creado en la clase MAIN como un método estático.

- o El método recibe 2 parámetros

- La Pila de Clientes
- El valor(String) de la nueva dirección.

- o Cambiar la dirección del cliente siempre y cuando el género sea FEMENINO.



/ Llamando al Método



LLAMANDO AL METODO

```
// EJERCICIO 4
~~~~~
System.out.println("=====");
System.out.println("===== EJERCICIO 4 =====");
System.out.println("=====");

asignaDireccion(pila, NuevaDireccion: "BELLA VISTA");
pila.mostrar();
```



/ Imagen de la salida de la consola.

```
Main x
=====
===== EJRCICIO 4 =====
=====

Mostrando la pila de Clientes
Mostrando Cliente

Nombre: Oscar
Apellido: Choque
Edad: 25
Direccion: Rio Seco
Genero: Masculino

Mostrando Cliente

Nombre: Pablo
Apellido: Gutierrez
Edad: 28
Direccion: Calacoto
Genero: Masculino
```

Se Mantiene
Dirección

Se Mantiene
Dirección

```
Main x

Mostrando Cliente

Nombre: Ximena
Apellido: Ramos
Edad: 19
Direccion: BELLA VISTA
Genero: Femenino

Mostrando Cliente

Nombre: Juan
Apellido: Gonzales
Edad: 27
Direccion: Mercedario
Genero: Masculino

Mostrando Cliente

Nombre: Carla
Apellido: Mendoza
Edad: 18
Direccion: BELLA VISTA
Genero: Femenino
```

Dirección
Cambiada

Se Mantiene
Dirección

Dirección
Cambiada

/ 15.Mover ÍTEMS de la PILA.

Código que
Resuelve el
Problema

```
te.java x PilaCliente.java x Main.java x

// EJERCICIO 5
// REORDENAR LA PILA LOS DE GENERO MASCULINO MANDARLOS A LA BASE
// LOS DE GENERO FEMENINO MANDARLOS ENCIMA DE LOS DE GENERO MASCULINO
// Usage
public static void reordenaPila(PilaCliente pila) {
    PilaCliente aux = new PilaCliente(); // ALMACENAMOS LOS ELEMENTOS DE LA PILA
    PilaCliente auxMasculos = new PilaCliente(); // ALMACENAMOS LOS ELEMENTOS DE LA PILA del GENERO MASCULINO
    PilaCliente auxFemeninos = new PilaCliente(); // ALMACENAMOS LOS ELEMENTOS DE LA PILA el GENERO FEMENINO
    Cliente item;

    // Separando los elementos en las pilas auxiliares correspondientes según su género
    while (!pila.esVacio()) {
        item = pila.eliminar();
        if (item.getGenero().equals("Masculino")) { // para los de genero Masculino
            auxMasculos.adicionar(item);
        } else {
            auxFemeninos.adicionar(item); // para los de genero Femenino
        }
    }

    pila.vaciar(aux); // VACIAMOS LA PILA ORIGINAL

    // Agregando los elementos masculinos a la base de la pila
    while (!auxMasculos.esVacio()) {
        item = auxMasculos.eliminar();
        pila.adicionar(item);
    }

    // Agregando los elementos femeninos encima de los masculinos en la pila
    while (!auxFemeninos.esVacio()) {
        item = auxFemeninos.eliminar();
        pila.adicionar(item);
    }
}
```

/SUGERENCIAS

- o El método deberá llamarse reordenaPila(Pila)
- o El método debe ser creado en la clase MAIN como un método estático.
- o El método recibe 1 parámetro
 - La Pila de Clientes
- o Mover a la base todos los clientes del género masculino y los del género femenino moverlos al final.



/ Llamando al Método



LLAMANDO AL METODO

```
//EJERCICIO 5  
  
System.out.println("=====");  
System.out.println("===== EJERCICIO 5 =====");  
System.out.println("=====");  
reordenaPila(pila);  
pila.mostrar();  
}
```



/ Imagen de la salida de la consola.

```
=====
===== EJRCICIO 5 =====
=====

Mostrando la pila de Clientes
Mostrando Cliente

Nombre: Ximena
Apellido: Ramos
Edad: 19
Direccion: BELLA VISTA
Genero: Femenino

Mostrando Cliente

Nombre: Carla
Apellido: Mendoza
Edad: 18
Direccion: BELLA VISTA
Genero: Femenino
```

Cliente de
genero
Femenino
en Filan
De Pila

```
Mostrando Cliente

Nombre: Oscar
Apellido: Choque
Edad: 25
Direccion: Rio Seco
Genero: Masculino

Mostrando Cliente

Nombre: Pablo
Apellido: Gutierrez
Edad: 28
Direccion: Calacoto
Genero: Masculino

Mostrando Cliente

Nombre: Juan
Apellido: Gonzales
Edad: 27
Direccion: Mercedario
Genero: Masculino
```

Clientes de
genero
Masculino en
Base de Pila



¡GRACIAS POR SU ATENCION!



Email: condoedsoniver@gmail.com

Telefono: +591 72096981



ESTRUCTURA DE DATOS



EL ALTO, MAYO DE 2023



INNOVACIÓN
EN EDUCACIÓN