

|  |
| --- |
| GUÍA N° 5 – Pilas |

# LOGRO GENERAL DE LA SESIÓN DE APRENDIZAJE

Al finalizar la sesión, el estudiante desarrolla algoritmos de mediana dificultad utilizando Pilas.

# OBJETIVOS ESPECÍFICOS DE LA PRÁCTICA DE LABORATORIO

Desarrolla algoritmos con Pilas, implementando operaciones de inserción y eliminación usando arreglos.

# MATERIALES Y EQUIPOS

* Netbeans
* Computadoras

# PROCEDIMIENTO

Implementación de Pilas utilizando arreglos.

***Creación de la clase PilaArray:***

Class PilaArray{

private final int MAX;

private final int Pila[ ];

private int cima;

public PilaArray(int tamanio ) {

this.MAX =tamanio-1;

this.Pila = new int[tamanio];

this.cima = -1;

***Método para verificar si la Pila está vacía***

public boolean isPilaVacia( ){

if (cima == -1) {

return true;

}else{

return false;

}

}

***Método para verificar si la Pila está llena***

public boolean isPilaLlena( ){

if (cima == MAX) {

return true;

}else{

return false;

}

}

***Método para insertar un elemento en la Pila (Push)***

public void insertarPila(int elem) {

if (cima == MAX) { // if (isPilaLlena( )) {

System.out.println(“Pila llena. No se puede insertar");

}else{

cima++;

Pila[cima]=elem;

}

}

***Método para eliminar un elemento en la Pila (Pop)***

public int eliminarPila( ) {

if (cima==-1) { // if (isPilaVacia( )) {

System.out.println(“Cola Vacía. No hay elementos");

return 0;

} else {

int fuera = Pila[cima];

cima--;

return fuera;

}

}

***Método para vaciar la Pila***

public void vaciarPila( ) {

cima = -1;

}

***Método para mostrar elementos de la Pila***

public void mostrarPila( ) {

if (isPilaVacia()) {

System.out.println(“Pila Vacía. No hay elementos");

} else {

String texto="";

for (int i = 0; i <= cima; i++) {

texto = texto + "\n"+Pila[i];

}

System.out.println(“Datos de la Pila “+texto); }

}

***Método para consultar el último elemento de la Pila***

public int ultimoPila( ) {

if (isPilaVacia()) {

System.out.println(“Pila Vacía");

return 0;

}else{

System.out.println(“Ultimo elemento:” +Pila[cima] );

return Pila[cima];

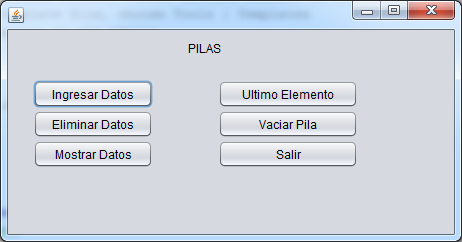
}

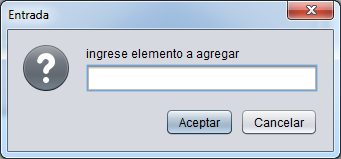
}

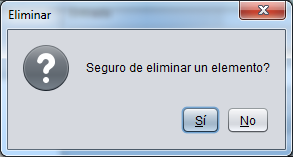
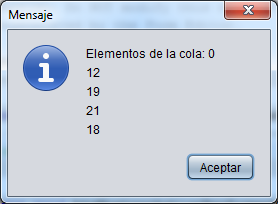
Elabore un menú para realizar las operaciones con Pilas, utilizando los métodos indicados.

**EJERCICIOS**

1.- Realice un programa para ingresar, eliminar y mostrar elementos de una Pila.







Pila

* El botón **Ingresar Elemento** permitirá el ingreso de un elemento de la Pila (Push)
* El botón **Eliminar Elemento** permitirá eliminar un elemento de la Pila (Pop)
* El botón **Mostrar Elementos** presentará un JOptionPane con los elementos de la Pila.
* El botón **Vaciar cola** eliminará todos los elementos de la Pila, dejándola vacía.

Sugerencias:

* Crear una clase (ejemplo: PilaArray) con los atributos de tipo int: MAX, Pila[], cima
* Constructor 1 à **PilaArray()** con las siguientes inicializaciones:

MAX=20;

Pila = new int[MAX+1];

Cima =0;

* Constructor 2 à **ColaArray(MAX)** con las siguientes inicializaciones:

MAX=MAX;

Pila = new int[MAX+1];

cima = 0;

* Método IsPilaLlena( ) à retorno boolean. Verifica que la Pila está llena.
* Método IsPilaVacia( ) à retorno boolean. Verifica que la Pila está vacía.
* Método InsertarPila( ) à void. Inserta un elemento en la Pila. Previamente verifica si la Pila está llena; en cuyo caso debe mostrar un mensaje con JOptionPane.
* Método EliminarPila( ) à void. Elimina un elemento en la Pila. Previamente verifica si la Pila está vacía; en cuyo caso debe mostrar un mensaje con JOptionPane.
* Método MostrarPila( )à Muestra los elementos de la Pila (con JOptionPane). Previamente verifica si la Pila está vacía; en cuyo caso debe mostrar un mensaje con JOptionPane.
* Método VaciarPila( ) à Elimina todos los elementos de la Pila.
* Método primeroPila( ) à muestra el primer elemento de la Pila.

En la clase que corresponde a la interfaz:

* Crear un objeto de la clase PilaArray a nivel de clase. (ejemplo. objPila)
* En el Constructor, inicializar el objeto:

objPila = new PilaArray(10);