



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**



FACULTAD DE INGENIERIA

PROYECTO FINAL

"IMPLEMENTACIÓN BÁSICA DE UNA PILA: OPERACIONES"

ALUMNA: ELISA DANIELA RIOS HERRERA

SEMESTRE 2021-II

PROFESOR: M.I. MARCO ANTONIO MARTÍNEZ QUINTANA

ESTRUCTURA DE DATOS Y ALGORITMOS I

ELABORADO EN AGOSTO DE 2021

Objetivo

Mostrar cómo funcionan las operaciones básicas de una pila en lenguaje C y el código que se debe emplear para manejarlas.

Este documento pertenece al contenido del proyecto final de la asignatura Estructura de Datos y Algoritmos I, en donde se podrá encontrar el objetivo del proyecto, la introducción al proyecto, el enfoque del proyecto y el algoritmo que se implementó antes de codificar dicho proyecto.

Este proyecto también incluye los resultados finales para verificar que funciona correctamente, una tabla de los costos propuestos para llevar a cabo el proyecto fuera de lo académico (si se diera el caso), el link del repositorio donde se encuentran todos los recursos empleados en este proyecto y un manual o guía rápida de usuario.

Por último, encontraremos la conclusión de la realización del proyecto, se anexarán también las distintas referencias que nos facilitaron su desarrollo, así como un glosario para comprender los términos empleados en éste.

Este proyecto trata principalmente del uso de las pilas en la programación. La pila es una lista de elementos que se caracteriza por operar la inserción y eliminación de elementos únicamente por el extremo de su estructura, este extremo por lo regular se denomina tope o límite.

Un ejemplo característico de la pila en la computación se presenta durante la ejecución de un programa mientras el dispositivo procesa las funciones a las que se van llamando. Otros ejemplos suceden en los navegadores web cuando se almacenan los sitios que previamente visitamos, o cuando el usuario quiere regresar a la página anterior.

El nombre de pila se deriva de la metáfora de una pila de platos en una cocina, ya que el último elemento en entrar es el único elemento accesible en todo momento.

La pila emplea dos operaciones básicas, push que sirve para insertar un elemento, este elemento estará en el tope de la pila, y pop que sirve para eliminar este mismo último elemento agregado.

Algoritmos parciales de algunas operaciones aplicadas

- **Para la operación PUSH** (insertar un elemento en la pila del programa)

Si $\text{limite} < \text{máximo}$

entonces

$\text{limite} = \text{limite} + 1$

$\text{pila}[\text{limite}] = \text{dato}$

si no

escribir "La pila se encuentra en su límite"

Fin

FIN

- **Para la operación POP** (eliminar el último elemento agregado en la pila del programa)

Si $\text{limite} > 0$

entonces

$\text{dato} = \text{pila}[\text{limite}]$

$\text{limite} = \text{limite} - 1$

si no

escribir "Ya no existen elementos en la pila"

Fin

FIN

Código fuente

/*

PROYECTO FINAL: IMPLEMENTACIÓN BÁSICA DE UNA PILA

ESTRUCTURA DE DATOS Y ALGORITMOS I

-- Por: Elisa Daniela Rios Herrera --

FI, UNAM.

*/

#include <stdio.h>

#include <stdlib.h> // Esta librería nos permite el uso de free y malloc

#include <stdbool.h> // Esta librería nos permite el uso de booleanos

struct nodo // con struct estamos definiendo a una estructura, en este caso el nodo

{

int n;

struct nodo *sig;

};

// Llamamos a las funciones con void

void agregar(int n); // se aplica la función push para agregar

void eliminar(void); // se aplica la función pop para eliminar

void imprimir(void);

int size(void); // para indicar el tamaño de la pila

// bool es una variable que solo puede resultar verdadera o falsa

bool vacia(void); // para indicar si la pila se encuentra vacia

int ultimo(void); // para mostrar el último elemento agregado; si no hay elementos devuelve 0

```
struct nodo *sup = NULL; // aplicamos punteros
```

```
// NULL hace referencia a un dato válido
```

```
int main()
```

```
{
```

```
    int eleccion;
```

```
    int n;
```

```
    while (eleccion != 9)
```

```
    {
```

```
        printf("\n***SISTEMA DE ELEMENTOS EN UNA PILA***\n\nOPCIONES\n\n1.-  
Agregar elemento\n2.- Eliminar elemento\n3.- Imprimir "
```

```
"la pila\n4.- Imprimir cuanto mide la pila\n5.- Consultar si la pila esta vacia\n6.- "
```

```
        "Mostrar el ultimo elemento agregado\n\nPresione 9 si desea salir\n\n\tPor  
favor elija una opcion: ");
```

```
        scanf("%d", &eleccion);
```

```
        switch (eleccion)
```

```
        {
```

```
        case 1:
```

```
            printf("Ingrese el numero a agregar: \n"); // el caso 1 llama a la funcion agregar
```

```
            scanf("%d", &n);
```

```
            agregar(n);
```

```
            break;
```

```
        case 2: // el caso 2 llama a la función eliminar
```

```
            eliminar();
```

```
            break;
```

```
        case 3: // el caso 3 llama a la función imprimir
```

```
            imprimir();
```

```

    break;

case 4: // el caso 4 llama a la función de tamaño
    printf("La pila mide %d\n elementos", size());
    break;

case 5: // el caso 5 llama a la función para comprobar si la pila está vacía o
tiene elementos
    if (vacía())
    {
        printf("La pila se encuentra vacía\n");
    }
    else
    {
        printf("La pila NO se encuentra vacía\n");
    }
    break;

case 6: // el caso 6 llama a la función ultimo
    printf("El ultimo elemento es: %d\n", ultimo());
    break;
}
}
}

// FUNCIÓN: Para consultar el tamaño
int size(void)
{
    int contador = 0;
    if (sup == NULL)
        return contador;

```

```

struct nodo *temp = sup;
while (temp != NULL)
{
    contador++;
    temp = temp->sig;
}
return contador;
}

```

// FUNCIÓN: Para consultar si la pila tiene o no elementos

```

bool vacia(void)
{
return sup == NULL;
}

```

```

int ultimo()
{
    if (sup == NULL)
        return -1;
    return sup->n;
}

```

// FUNCIÓN: Para agregar un elemento a la pila (función push)

```

void agregar(int n)
{
    printf("Se ha agregado correctamente el elemento: %d\n", n);
    struct nodo *nuevoNodo = malloc(sizeof(struct nodo));

```

```
nuevoNodo->n = n;
nuevoNodo->sig = sup;
sup = nuevoNodo;
}
```

// FUNCIÓN: Para imprimir los elementos de nuestra pila

```
void imprimir(void)
{
    printf("La pila se conforma por los siguientes elementos: \n");
    struct nodo *temp = sup;
    while (temp != NULL)
    {
        printf("%d\n", temp->n);
        temp = temp->sig;
    }
}
```

// FUNCIÓN: Para eliminar el ultimo elemento agregado (función pop)

```
void eliminar(void)
{
    printf("Se ha eliminado el ultimo elemento agregado\n");
    if (sup != NULL)
    {
        struct nodo *temp = sup;
        sup = sup->sig;
        free(temp); // se usa free para liberar memoria que se había asignado antes
    }
}
```


Resultados del proyecto

Capturas del código probado en Dev C++

```
Proyecto Final (Pila).c
1  /*
2     PROYECTO FINAL: IMPLEMENTACIÓN BÁSICA DE UNA PILA
3     ESTRUCTURA DE DATOS Y ALGORITMOS I
4     -- Por: Elisa Daniela Rios Herrera --
5
6     FI, UNAM.
7
8  */
9
10
11
12 #include <stdio.h>
13 #include <stdlib.h> // Esta librería nos permite el uso de free y malloc
14 #include <stdbool.h> // Esta librería nos permite el uso de booleanos
15
16
17 struct nodo // con struct estamos definiendo a una estructura, en este caso el nodo
18 {
19     int n;
20     struct nodo *sig;
21 };
22
23 // Llamamos a las funciones con void
24 void agregar(int n); // se aplica la función push para agregar
25 void eliminar(void); // se aplica la función pop para eliminar
26 void imprimir(void);
27 int size(void); // para indicar el tamaño de la pila
28 // bool es una variable que solo puede resultar verdadera o falsa
29 bool vacia(void); // para indicar si la pila se encuentra vacía
30 int ultimo(void); // para mostrar el último elemento agregado; si no hay elementos devuelve 0
31
32 struct nodo *sup = NULL; // aplicamos punteros
33 // NULL hace referencia a un dato válido
34
35 int main()
36 {
37     int eleccion;
```

```
Proyecto Final (Pila).c
38     int n;
39     while (eleccion != 9)
40     {
41         printf("\n***SISTEMA DE ELEMENTOS EN UNA PILA***\n\nOPCIONES\n\n1.- Agregar elemento\n2.- Eliminar elemento\n3.- Imprimir "
42             "la pila\n4.- Imprimir cuanto mide la pila\n5.- Consultar si la pila esta vacia\n6.- "
43             "Mostrar el ultimo elemento agregado\n\nPresione 9 si desea salir\n\n\tPor favor elija una opcion: ");
44         scanf("%d", &eleccion);
45         switch (eleccion)
46         {
47             case 1:
48                 printf("Ingrese el numero a agregar: \n"); // el caso 1 llama a la funcion agregar
49                 scanf("%d", &n);
50                 agregar(n);
51                 break;
52             case 2: // el caso 2 llama a la funcion eliminar
53                 eliminar();
54                 break;
55             case 3: // el caso 3 llama a la funcion imprimir
56                 imprimir();
57                 break;
58             case 4: // el caso 4 llama a la funcion de tamaño
59                 printf("La pila mide %d\n elementos", size());
60                 break;
61             case 5: // el caso 5 llama a la funcion para comprobar si la pila está vacía o tiene elementos
62                 if (vacía())
63                 {
64                     printf("La pila se encuentra vacia\n");
65                 }
66                 else
67                 {
68                     printf("La pila NO se encuentra vacia\n");
69                 }
70                 break;
71             case 6: // el caso 6 llama a la funcion ultimo
72                 printf("El ultimo elemento es: %d\n", ultimo());
73                 break;
74         }
75     }
```

```

Proyecto Final (Pila).c
75 | }
76 | }
77 |
78 | // FUNCIÓN: Para consultar el tamaño
79 | int size(void)
80 | {
81 |     int contador = 0;
82 |     if (sup == NULL)
83 |         return contador;
84 |     struct nodo *temp = sup;
85 |     while (temp != NULL)
86 |     {
87 |         contador++;
88 |         temp = temp->sig;
89 |     }
90 |     return contador;
91 | }
92 |
93 | // FUNCIÓN: Para consultar si la pila tiene o no elementos
94 | bool vacia(void)
95 | {
96 |     return sup == NULL;
97 | }
98 |
99 | int ultimo()
100 | {
101 |     if (sup == NULL)
102 |         return -1;
103 |     return sup->n;
104 | }
105 |
106 | // FUNCIÓN: Para agregar un elemento a la pila (función push)
107 | void agregar(int n)
108 | {
109 |     printf("Se ha agregado correctamente el elemento: %d\n", n);
110 |     struct nodo *nuevoNodo = malloc(sizeof(struct nodo));
111 |     nuevoNodo->n = n;
112 |     nuevoNodo->sig = sup;
113 |     sup = nuevoNodo;
114 | }
115 |
116 | // FUNCIÓN: Para imprimir los elementos de nuestra pila
117 | void imprimir(void)
118 | {
119 |     printf("La pila se conforma por los siguiente elementos: \n");
120 |     struct nodo *temp = sup;
121 |     while (temp != NULL)
122 |     {
123 |         printf("%d\n", temp->n);
124 |         temp = temp->sig;
125 |     }
126 | }
127 |
128 | // FUNCIÓN: Para eliminar el ultimo elemento agregado (función pop)
129 | void eliminar(void)
130 | {
131 |     printf("Se ha eliminado el ultimo elemento agregado\n");
132 |     if (sup != NULL)
133 |     {
134 |         struct nodo *temp = sup;
135 |         sup = sup->sig;
136 |         free(temp); // se usa free para liberar memoria que se había asignado antes
137 |     }
138 | }

```

Funcionamiento del código

1. Para agregar un elemento a la pila

```
C:\Users\elisa\Desktop\Proyecto Final (Pila).exe

***SISTEMA DE ELEMENTOS EN UNA PILA***

OPCIONES

1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado

Presione 9 si desea salir

        Por favor elija una opcion: 1
Ingrese el numero a agregar:
5
Se ha agregado correctamente el elemento: 5
```

2. Para eliminar el último elemento de la pila

```
C:\Users\elisa\Desktop\Proyecto Final (Pila).exe

***SISTEMA DE ELEMENTOS EN UNA PILA***

OPCIONES

1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado

Presione 9 si desea salir

        Por favor elija una opcion: 1
Ingrese el numero a agregar:
4
Se ha agregado correctamente el elemento: 4

***SISTEMA DE ELEMENTOS EN UNA PILA***

OPCIONES

1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado

Presione 9 si desea salir

        Por favor elija una opcion: 1
Ingrese el numero a agregar:
6
Se ha agregado correctamente el elemento: 6

***SISTEMA DE ELEMENTOS EN UNA PILA***

OPCIONES

1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado


Presione 9 si desea salir

        Por favor elija una opcion: 2
Se ha eliminado el ultimo elemento agregado
```

Primero ingresamos el número 4 a la pila y después el número 6, finalizamos eliminando el último elemento, quiere decir que eliminamos el número 6 de la pila.

Esto lo comprobaremos en el siguiente paso.

3. Para imprimir los elementos de la pila

 C:\Users\elisa\Desktop\Proyecto Final (Pila).exe

```
***SISTEMA DE ELEMENTOS EN UNA PILA***  
  
OPCIONES  
  
1.- Agregar elemento  
2.- Eliminar elemento  
3.- Imprimir la pila  
4.- Imprimir cuanto mide la pila  
5.- Consultar si la pila esta vacia  
6.- Mostrar el ultimo elemento agregado  
  
Presione 9 si desea salir  
  
      Por favor elija una opcion: 3  
La pila se conforma por los siguiente elementos:  
4
```

Podemos observar que el único elemento existente en la pila es el número 4, ya que anteriormente se eliminó el número 6 (quien fue el último elemento agregado)

4. Para imprimir el tamaño de la pila

Para verificar el tamaño de la pila agregaremos antes algunos elementos

```
      Por favor elija una opcion: 1  
Ingrese el numero a agregar:  
7  
Se ha agregado correctamente el elemento: 7
```

```
      Por favor elija una opcion: 1  
Ingrese el numero a agregar:  
8  
Se ha agregado correctamente el elemento: 8
```

```
      Por favor elija una opcion: 1  
Ingrese el numero a agregar:  
9  
Se ha agregado correctamente el elemento: 9
```


Agregamos como elementos al número 7, 8 y 9 respectivamente

Ahora sí consultaremos el tamaño

```
      Por favor elija una opcion: 4  
La pila mide 4  
elementos
```

Podemos observar que el tamaño de la pila es correcto, ya que anteriormente en nuestra pila teníamos por elemento al número 4, más los otros elementos que agregamos, nos da un total de 4

5. Para consultar si la pila tiene elementos o no

 C:\Users\elisa\Desktop\Proyecto Final (Pila).exe

```
***SISTEMA DE ELEMENTOS EN UNA PILA***
```


```
OPCIONES
```

- 1.- Agregar elemento
- 2.- Eliminar elemento
- 3.- Imprimir la pila
- 4.- Imprimir cuanto mide la pila
- 5.- Consultar si la pila esta vacia
- 6.- Mostrar el ultimo elemento agregado

```
Presione 9 si desea salir
```

```
        Por favor elija una opcion: 5  
La pila NO se encuentra vacia
```

6. Para mostrar el último elemento agregado a la pila

 C:\Users\elisa\Desktop\Proyecto Final (Pila).exe

```
***SISTEMA DE ELEMENTOS EN UNA PILA***
```

```
OPCIONES
```

- 1.- Agregar elemento
- 2.- Eliminar elemento
- 3.- Imprimir la pila
- 4.- Imprimir cuanto mide la pila
- 5.- Consultar si la pila esta vacia
- 6.- Mostrar el ultimo elemento agregado

```
Presione 9 si desea salir
```

```
        Por favor elija una opcion: 6  
El ultimo elemento es: 9
```

Recursos informativos para llevar a cabo correctamente el proyecto

Software	Microsoft Windows 10 Home Dev C++ Microsoft Word
Hardware	HUAWEI Matebook 13
Tipo de sistema	x64-based PC

Tabla de costos propuestos

FACTOR	VALOR
Algoritmos	100 MXN
Código fuente	500 MXN
Manual de usuario	250 MXN
Costo total aproximado	850 MXN – ochocientos cincuenta pesos

Repositorio de GitHub

El proyecto y sus recursos empleados se puede encontrar en:

<https://github.com/elisrh11/EDA-I>

Guía rápida de usuario

Estimado usuario:

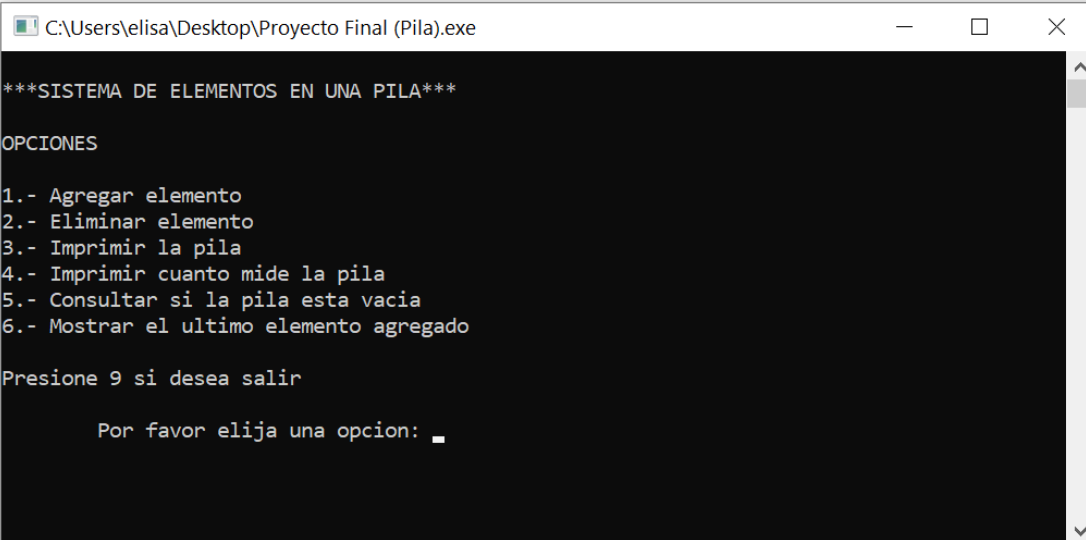
Para utilizar fácilmente este programa lo ejecutará desde el archivo .exe

El programa te ofrece 6 opciones diferentes que puedes ejecutar, las cuales son:

- Agregar un elemento a la pila (1)
- Eliminar elemento de la pila (2)
- Imprimir la pila (3)
- Imprimir cuanto mide la pila (4)
- Consultar si la pila está vacía o tiene elementos (5)
- Mostrar el último elemento agregado (6)

A continuación, te mostraremos detalladamente como emplear estas opciones con el programa.

Al inicializar el programa te mostrará esta bienvenida y el menú con las consultas disponibles que podrás realizar:



```
C:\Users\elisa\Desktop\Proyecto Final (Pila).exe

***SISTEMA DE ELEMENTOS EN UNA PILA***

OPCIONES

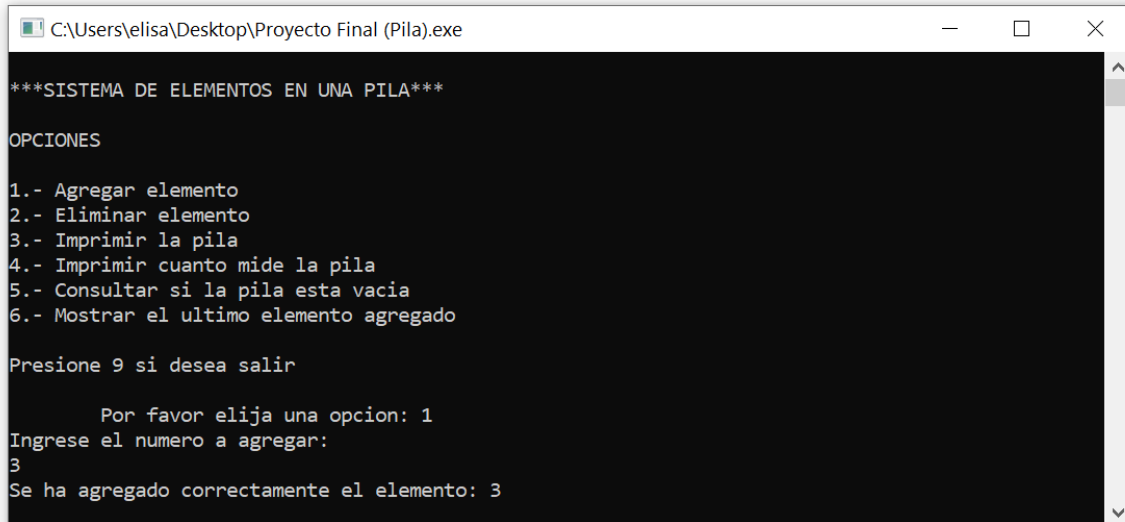
1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado

Presione 9 si desea salir

Por favor elija una opcion: _
```

1. Si deseas **agregar** un elemento a la pila, tendrás que elegir la opción 1, dar ENTER y el programa te pedirá ingresar que elemento deseas agregar:

El programa te dirá que agregó exitosamente tu elemento.



```
C:\Users\elisa\Desktop\Proyecto Final (Pila).exe

***SISTEMA DE ELEMENTOS EN UNA PILA***

OPCIONES

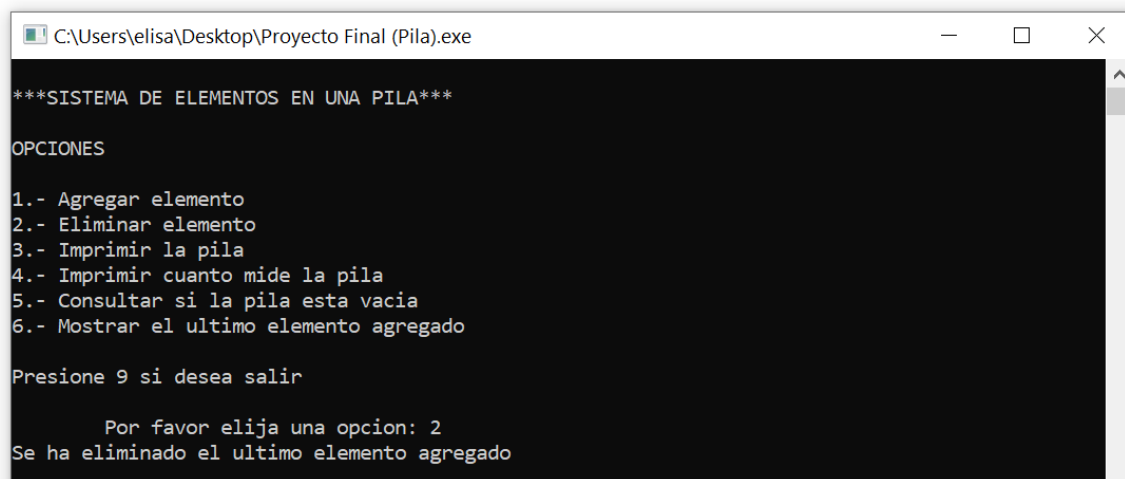
1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado

Presione 9 si desea salir

      Por favor elija una opcion: 1
Ingrese el numero a agregar:
3
Se ha agregado correctamente el elemento: 3
```

2. Si deseas **eliminar** el último elemento de la pila, tendrás que elegir la opción 2 y dar ENTER, el último elemento se eliminará exitosamente.

Recuerda, esta función solo aplica para el último elemento agregado previamente (función pop de una pila)



```
C:\Users\elisa\Desktop\Proyecto Final (Pila).exe

***SISTEMA DE ELEMENTOS EN UNA PILA***

OPCIONES

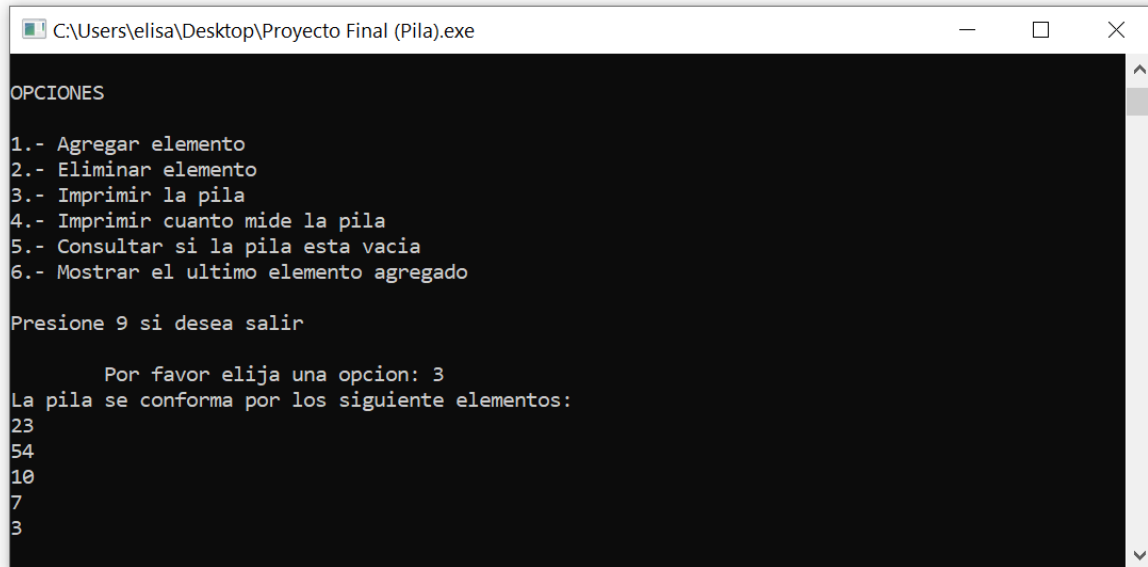
1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado

Presione 9 si desea salir

      Por favor elija una opcion: 2
Se ha eliminado el ultimo elemento agregado
```


3. Si deseas **imprimir** la pila después de que ya hayas agregado o eliminado elementos, tendrás que elegir la opción 3 y dar ENTER.

El programa te mostrará los elementos que conforman la pila.



```
C:\Users\elisa\Desktop\Proyecto Final (Pila).exe

OPCIONES

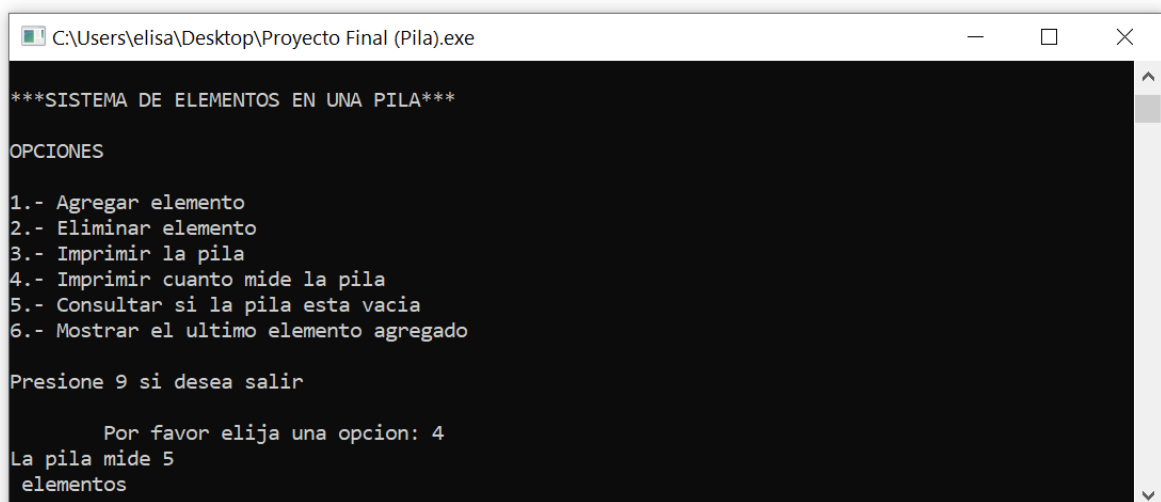
1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado

Presione 9 si desea salir

      Por favor elija una opcion: 3
La pila se conforma por los siguiente elementos:
23
54
10
7
3
```

4. Si deseas **imprimir el tamaño de la pila**, tendrás que elegir la opción 4 y dar ENTER.

El programa te mostrará cuantos elementos conforman la pila.



```
C:\Users\elisa\Desktop\Proyecto Final (Pila).exe

***SISTEMA DE ELEMENTOS EN UNA PILA***

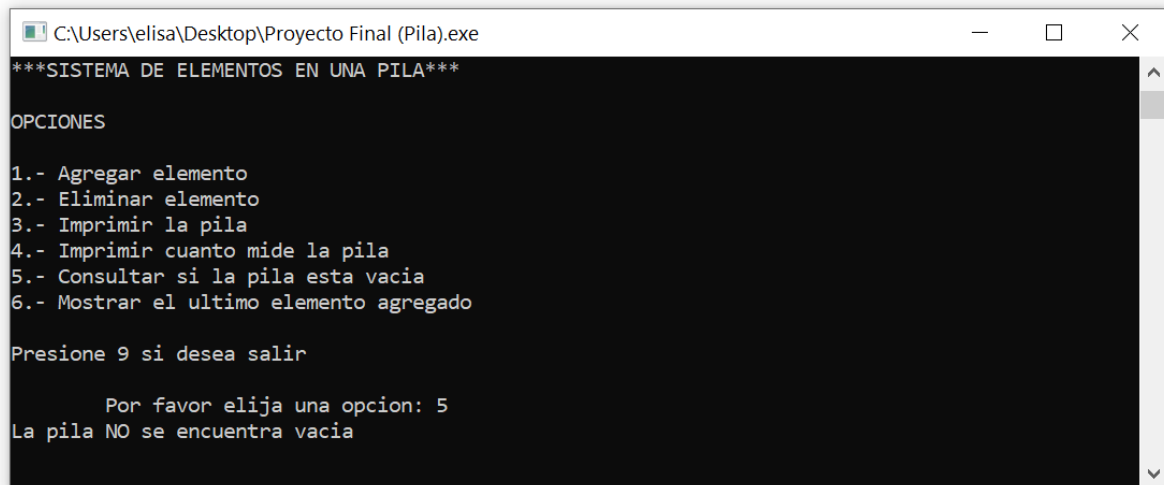
OPCIONES

1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado

Presione 9 si desea salir

      Por favor elija una opcion: 4
La pila mide 5
elementos
```

5. Si deseas **consultar si la pila se encuentra vacía** o tiene elementos, tendrá que elegir la opción 5 y dar ENTER.



```
C:\Users\elisa\Desktop\Proyecto Final (Pila).exe
***SISTEMA DE ELEMENTOS EN UNA PILA***

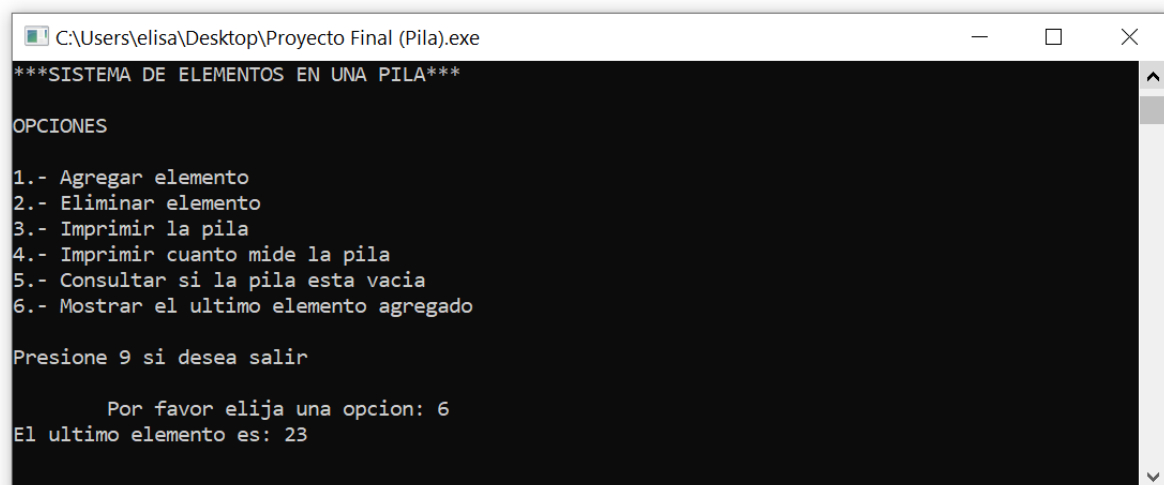
OPCIONES

1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado

Presione 9 si desea salir

      Por favor elija una opcion: 5
La pila NO se encuentra vacia
```

6. Si deseas que el programa te muestre el último elemento agregado, tendrás que elegir la opción 6 y dar ENTER.



```
C:\Users\elisa\Desktop\Proyecto Final (Pila).exe
***SISTEMA DE ELEMENTOS EN UNA PILA***

OPCIONES

1.- Agregar elemento
2.- Eliminar elemento
3.- Imprimir la pila
4.- Imprimir cuanto mide la pila
5.- Consultar si la pila esta vacia
6.- Mostrar el ultimo elemento agregado

Presione 9 si desea salir

      Por favor elija una opcion: 6
El ultimo elemento es: 23
```

Para salir del programa bastará con oprimir 9 y dar ENTER.

Conclusión

Un algoritmo es aplicar una serie de determinados pasos para buscar la correcta solución a un problema. Esto es de suma importancia dentro de la programación y su desarrollo, pues en este método se basa el programar. Se necesita, como primer paso, comprender el problema, después se realiza el plan, se ejecuta o prueba el plan y como último paso se analiza una solución. Los algoritmos tienen relación con las estructuras de datos las cuales comprendemos mejor con el desarrollo del curso de Estructuras de Datos y Algoritmos I y el desarrollo de este proyecto, estas estructuras son una compilación de datos organizados y las operaciones que derivan estos mismos. El algoritmo está compuesto por detalladas instrucciones que se organizan en forma de estructuras, algunas son las secuenciales, iterativas (de repetición) y las condicionales (de selección) e incorpora estructuras básicas en la computación para resolver un problema, y se utilizan en diferentes situaciones.

Se logró satisfactoriamente aprender a usar herramientas para resolver problemas un tanto más complejos con ayuda del desarrollo de un código sencillo y rápido, pero que nos facilita comprender a fondo el comportamiento de las estructuras de datos dentro de él y su relación con otros operadores o instrucciones.

Al finalizar la realización de este proyecto quedé muy contenta y satisfecha, pues comprendí mucho mejor el funcionamiento de las estructuras de datos y más en específico el funcionamiento de las pilas. Esto fue de gran ayuda para complementar mi conocimiento junto con las actividades que estuvimos realizando a lo largo del curso con el M.I. Marco.

Referencias

Joyanes Aguilar, L., & Zahonero Martínez, I. (2004). Algoritmos y estructuras de datos: una perspectiva en C.

Aldea, M., & González, M. Estructuras de Datos.

Lipschutz, S., de Apodaca, M. O. O., Yañez, L. H., & Sánchez, A. V. (1987). *Estructura de datos*. McGraw-Hill.

Recurso Dev C++: <https://sourceforge.net/projects/orwelldevcpp/>

Glosario

Boolean. Dato lógico que solo puede tener los valores *true* (verdadero) o *false* (falso)

Compilador. Software que traduce un programa escrito en un lenguaje de programación de alto nivel en lenguaje máquina

Hardware. Conjunto de componentes físicos de los que está hecho el equipo.

Pila. Representa una estructura lineal de datos en que se puede agregar o quitar elementos únicamente por uno de los extremos

Programa. (*en programación*) Conjunto de pasos lógicos escritos en un lenguaje de programación que nos permite realizar una tarea específica

Sistema operativo o Software. Conjunto de programas o aplicaciones, instrucciones y reglas informáticas que hacen posible el funcionamiento del equipo.