**3.1 – Pilas**

**Qué es una Pila.**

Una pila (stack en [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s)) es una lista ordenada o [estructura de datos](https://es.wikipedia.org/wiki/Estructura_de_datos) que permite almacenar y recuperar datos, siendo el modo de acceso a sus elementos de tipo [LIFO](https://es.wikipedia.org/wiki/LIFO) (del inglés Last In, First Out, «último en entrar, primero en salir»). Esta estructura se aplica en multitud de supuestos en el área de [informática](https://es.wikipedia.org/wiki/Inform%C3%A1tica) debido a su simplicidad y capacidad de dar respuesta a numerosos procesos.

**Cuáles son las operaciones básicas de una pila.**

Habitualmente, junto a las dos operaciones básicas de apilar y desapilar (push, pop), las pilas puede implementar otra serie de funciones:

* Crear (constructor): crea la pila vacía.
* Tamaño (size): regresa el número de elementos de la pila.
* Apilar (push): añade un elemento a la pila.
* Desapilar (pop): lee y retira el elemento superior de la pila.
* Leer último (top o peek): lee el elemento superior de la pila sin retirarlo.
* Vacía (empty): devuelve cierto si la pila está sin elementos o falso en caso de que contenga alguno.

Una pila puede implementarse fácilmente ya sea mediante una matriz o una lista enlazada. Lo que identifica a una estructura de datos como una pila en cualquier caso no es su estructura sino su interfaz: al usuario solamente se le permite colocar y extraer datos en el modo que se espera de una pila y algunas otras operaciones auxiliares.

**Aplicaciones de una Pila.**

Las pilas suelen emplearse en los siguientes contextos:

* Evaluación de expresiones en [notación postfija](https://es.wikipedia.org/wiki/Notaci%C3%B3n_postfija) ([notación polaca inversa](https://es.wikipedia.org/wiki/Notaci%C3%B3n_polaca_inversa)).
* Reconocedores sintácticos de [lenguajes independientes del contexto](https://es.wikipedia.org/w/index.php?title=Lenguaje_libre_de_contexto&action=edit&redlink=1).
* Implementación de [recursividad](https://es.wikipedia.org/wiki/Recursividad).

**Programa ejemplo en C# de un TDA Pila.**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ejemplo

{

class Program

{

static void Main(string[] args)

{

var tiempo = new Stopwatch();

tiempo.Start();

long Memoria, Memoriaa, Memoriaaa;

Memoria = GC.GetTotalMemory(true);

Stack miPila = new Stack();

int opcion;//opcion del menu

do

{

Console.Clear();//se limpia consola

opcion = menu();//muestra menu y espera opción

switch (opcion)

{

case 1:

agregar(ref miPila);

break;

case 2:

eliminar(ref miPila);

break;

case 3:

limpiar(ref miPila);

break;

case 4:

imprimir(miPila);

break;

case 5:

break; //salir

default:

mensaje("ERROR: la opción no es valida. Intente de nuevo.");

Console.WriteLine("\nPresione una tecla para continuar. . . ");

Console.ReadKey();

Console.Clear();

break;

}

}

while (opcion != 5);

mensaje("El programa a finalizado.");

Console.Clear();

tiempo.Stop();

Memoriaa = GC.GetTotalMemory(true);

Memoriaaa = Memoriaa - Memoria;

Console.WriteLine("Memoria Inicial: " + Memoria + " bytes");

Console.WriteLine("Memoria Final: " + Memoriaa + " bytes");

Console.WriteLine("Memoria Total: " + Memoriaaa + " bytes");

Console.WriteLine("\nEl tiempo fue de " + tiempo.Elapsed.TotalSeconds + " segundos");

Console.WriteLine("\nPresione una tecla para continuar . . . ");

Console.ReadKey();

}

// añade un nuevo elemento a la pila

static void agregar(ref Stack pila)

{

Console.Write("\n>Ingrese valor: ");

try

{

int valor = Convert.ToInt32(Console.ReadLine());

if (valor > 99 || valor <= 0)

{

mensaje("Solo números del 1 al 99");

}

else

{

pila.Push(valor);

imprimir(pila);

}

}

catch

{

mensaje("Error: Solo números del 1 al 99");

}

}

/\*\* Elimina todo los elementos de la pila \*/

static void limpiar(ref Stack pila)

{

pila.Clear();

imprimir(pila);

}

/\*\* Elimina elemento de la pila \*/

static void eliminar(ref Stack pila)

{

if (pila.Count > 0)

{

int valor = (int)pila.Pop();

mensaje("Elemento " + valor + " eliminado");

}

else

{

imprimir(pila);

}

}

/\*\* muestra menu y retorna opción \*/

static int menu()

{

//Console.Clear();

Console.WriteLine("Stack Menu");

Console.WriteLine("1.- Agregar elemento");

Console.WriteLine("2.- Eliminar elemento");

Console.WriteLine("3.- Vaciar Pila");

Console.WriteLine("4.- Ver pila");

Console.WriteLine("5.- Termina programa");

Console.Write("Ingresa tu opción: ");

try

{

int valor = Convert.ToInt32(Console.ReadLine());

return valor;

}

catch

{

return 0;

}

}

/\*\* Muestra mensaje del programa al usuario \*/

static void mensaje(String texto)

{

if (texto.Length > 0)

{

Console.WriteLine("{0}", texto);

Console.WriteLine("\nPresione cualquier tecla para continuar...");

Console.ReadKey();

}

}

/\*\* Imprime pila \*/

static void imprimir(Stack pila)

{

if (pila.Count > 0)

{

Console.WriteLine("");

foreach (int dato in pila)

{

Console.WriteLine(" | |");

if (dato < 10)

Console.WriteLine(" | 0{0} |", dato);

else

Console.WriteLine(" | {0} |", dato);

Console.WriteLine(" |\_\_\_\_\_\_|");

}

Console.WriteLine("\nPresione cualquier tecla para continuar...");

Console.ReadKey();

}

else

{

mensaje("La Pila esta vacia");

}

}

}

}

<http://www.nachocabanes.com/csharp/curso2015/csharp11b.php#:~:text=Una%20pila%20nos%20permitir%C3%A1%20introducir,es%20lo%20primero%20en%20salir>

<https://es.wikipedia.org/wiki/Pila_(inform%C3%A1tica)>

<https://www.jc-mouse.net/net/c-sharp/pilas-con-c-sharp-ejercicio-resuelto>