

ESTRUCTURA DE DATOS

UNIDAD 1

SEMANA 5

UPN.EDU.PE

PRESENTACIÓN DE LA SESIÓN

Logro de la Sesión y Temario



Al finalizar el curso, el estudiante desarrolla un proyecto aplicando estructuras de datos dinámicas en el lenguaje C# con entorno gráfico; demostrando capacidad de análisis, pensamiento lógico y buenas prácticas de programación e implementa funciones definidas de usuarios y sus diferentes tipos

Temario

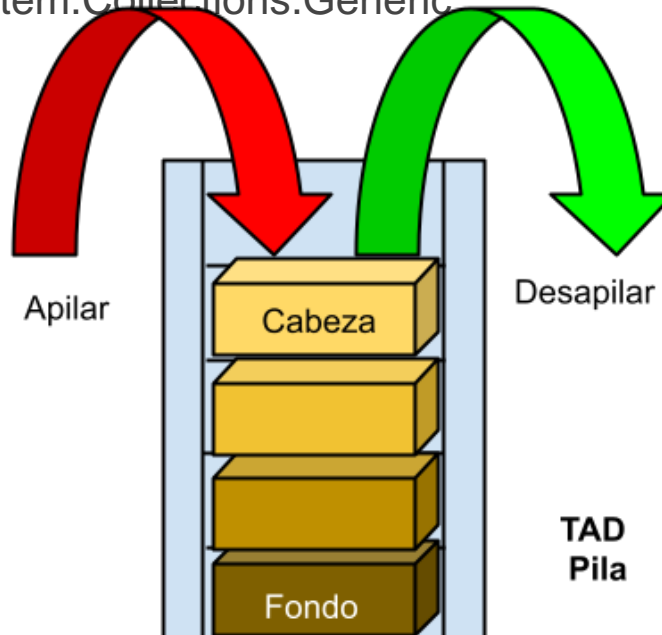
- PILAS
- Operaciones. Notación prefija, infija, postfija o polaca. Análisis y evaluación de expresiones.

TAD pila (stack) . Tipo Abstracto de Datos – PILA.



En una pila lo último que entra es lo primero que

```
using System;  
// TADS fuertemente tipados using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
// TADS pila, colas, etc.  
using System.Collections;
```



```
namespace EjemploPila  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            string palabra;  
  
            Stack miPila = new Stack();  
            miPila.Push("hola");  
            miPila.Push("buenos");  
            miPila.Push("días");  
  
            for (byte i = 0; i < 3; i++)  
            {  
                palabra =  
                    (string)miPila.Pop();  
  
                Console.WriteLine(palabra);  
            }  
  
            Console.ReadKey();  
        }  
    }  
}
```



CONCEPTO

- Los elementos de una pila se eliminan en orden inverso al que se insertaron; es decir, el último elemento que se mete en la pila es el primero que se saca.
- Por ejemplo: Una pila de libros que se exhiben en una librería o una pila de platos. Es de suponer que si el cocinero necesita un plato limpio, tomará el que está encima de todos, que es el último que se colocó en



CODIFICACIÓN



Pilas



Las pilas son estructuras de datos lineales, como los arreglos, ya que los componentes ocupan lugares sucesivos en la estructura y cada uno de ellos tiene un único sucesor y un único predecesor, con excepción del último y del primero, respectivamente



Una pila se define formalmente como una colección de datos a los cuales se puede acceder mediante un extremo, que se conoce generalmente como tope.

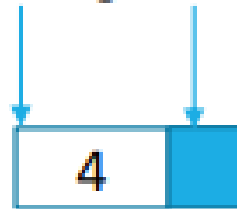


Las pilas no son estructuras fundamentales de datos Para su representación requieren el uso de otras estructuras de datos, como arreglos o listas.

PILAS



- La clase Nodo
- Crea un Nodo formado por los campos:
- **Dato (de tipo entero)** que corresponde al valor que contiene el nodo,
- **siguientePtr (de tipo apuntador)** que corresponde a la dirección del siguiente nodo en la pila.



```
// definicion del nodo de
class Nodo
{

public:
    int dato;    // INFO,
    Nodo *siguientePtr;

};
```

PILAS



- La clase Pila
- Crea la estructura Pila, formada por el nodo topePtr. Que contiene la dirección del nodo en la cima o tope de la pila.
- Se definen los métodos de la clase Pila.

```
// definicion de la pila de nodos
class Pila
{

private:
    Nodo *topePtr; // P, apuntador al
    bool estaVacia();

public:
    Pila();
    ~Pila();
    void mete(int valor);
    int saca();
    int nodoInicio();
    void mueve(Pila &p1);
    void muestra();
    int cantidadElementos();
    int nodoMayor();
    void eliminaNodo(int referencia);
    void concatena(Pila &p1);
};
```

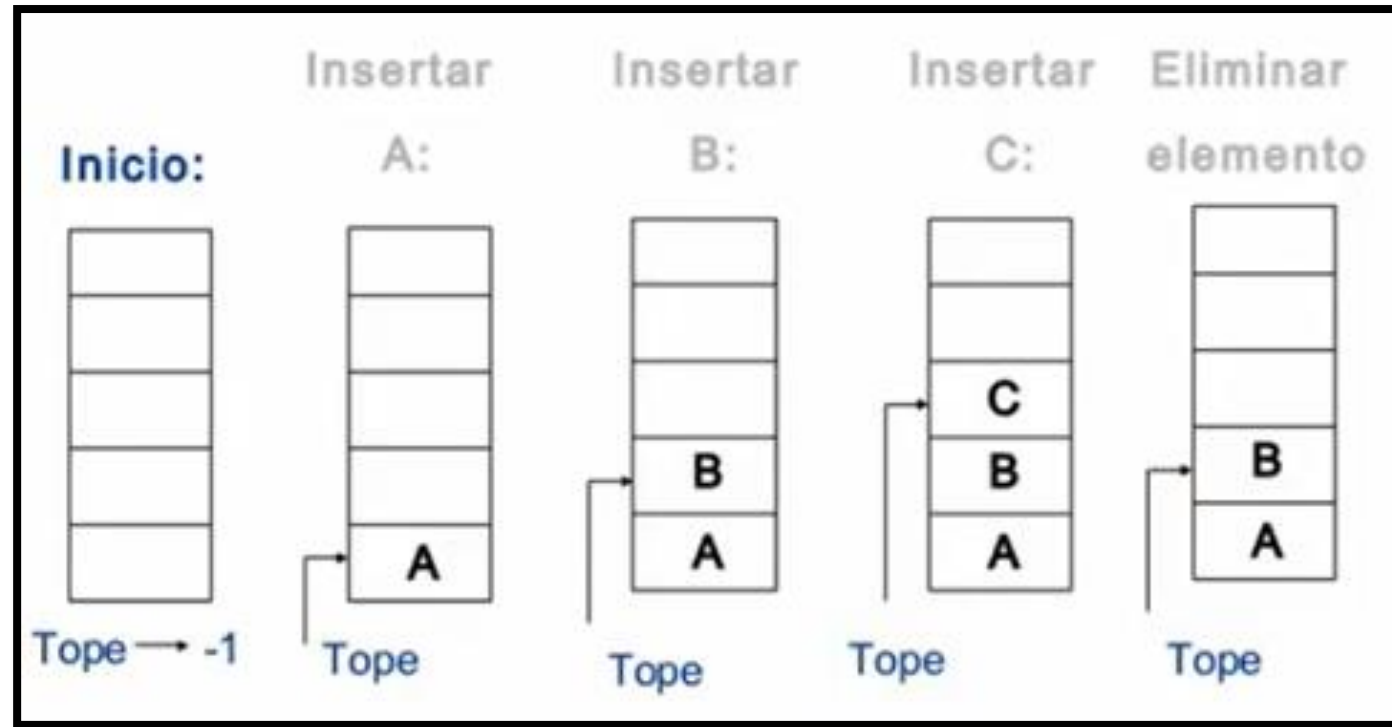
PILAS



Form1

Nombre:

Nodo c
Nodo B
Nodo A



Ejemplo de pila - consola



Desarrolle un programa en consola con un menú para el usuario que permita agregar elementos a una pila, eliminar elementos de la pila, vaciar la pila completa y mostrar la pila en pantalla, el programa solo debe aceptar números enteros del 1 al 99. Si el usuario introduce una opción que no existe o un valor erróneo el programa de avisar al usuario con un mensaje en pantalla y continuar con el programa.

```
static void Main(string[] args)
{
    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.");
                break;
        }
    }
    while(opcion!=5);
    mensaje("El programa a finalizado.");
}
```

EJEMPLO DE PILA - CONSOLA



```
/** 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);
    }
}
```

EJEMPLO DE PILA - CONSOLA



```
/** muestra menu y retorna opción */
static int menu()
{
    //Console.Clear();
    Console.WriteLine("\n          Stack Menu\n");
    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(" JJ:> 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("\n
=====");
        Console.WriteLine(" JJ:> {0}",texto);
        Console.WriteLine("
=====");
        Console.WriteLine("\n    JJ:>Presione cualquier tecla para
continuar...");
        Console.ReadKey();
    }
}
```

EJEMPLO DE PILA - CONSOLA



```
/** 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");
    }
}
```



DECLARACIÓN DE UN NODO



1. Crear un aplicativo para el prototipo del ingreso de los usuarios. Mostrar lo siguiente:
 - ☐ Insertar los nombres en la lista doble
 - ☐ Mostrar los nombre en la lista doble



CLASE – NODO



```
class Nodo
{
    public Nodo siguiente;
    public Nodo anterior;

    public string nombre;

    public Nodo (string nom)
    {
        siguiente = null;
        anterior = null;
        nombre = nom;
    }

    //get set para siguiente
    public Nodo getSiguiente()
    {
        return siguiente;
    }
}
```

```
public void setSiguiente(Nodo set)
{
    siguiente = set;
}
//get set para anterior
public Nodo getAnterior()
{
    return anterior;
}

public void setAnterior(Nodo ant)
{
    anterior = ant;
}
}
```



CLASE - LISTA



```
class Lista
{
    //declaramos variables de tipo nodo
    public Nodo primero;
    public Nodo ultimo;
    public Nodo raiz;

    //creamos el constructor
    public Lista()
    {
        primero = null;
        ultimo = null;
        raiz = null;
    }
}
```

```
//creamos el metodo para crear los nodos
public void insertar(string nom)
{
    if(primerο == null)
    {
        Nodo nuevo = new Nodo(nom);
        primero = nuevo; //se crea el primer nodo
        ultimo = primero; //el nuevo nodo para a ser el ultimo
    }
    else
    {
        Nodo nuevo = new Nodo(nom);
        ultimo.siguiente = nuevo;
        nuevo.anterior = ultimo;
        ultimo = nuevo;
    }
}
```



CLASE - FORM1



```
public partial class Form1 : Form
{
    Lista lista = new Lista();
    public Form1()
    {
        InitializeComponent();
    }

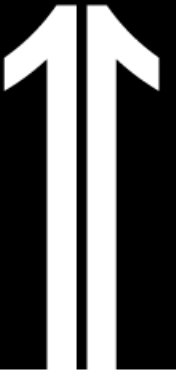
    private void BtnCrear_Click(object sender,
EventArgs e)
    {
        lista.insertar(Convert.ToString(TxtNombre.Text));
        TxtNombre.Text = "";
    }
}
```

```
private void BtnMostrar_Click(object sender, EventArgs e)
{
    lista.raiz = lista.primerono;
    while(lista.raiz != null)
    {
        //MessageBox.Show(lista.raiz.nombre);
        lbtLista.Items.Add(Convert.ToString(lista.raiz.nombre));
        lista.raiz = lista.raiz.siguiente; //apunta el nodo siguiente
    }

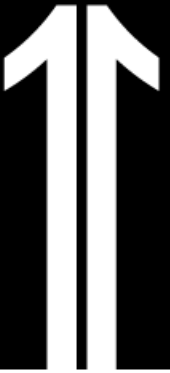
    lista.raiz = lista.ultimo;
    while (lista.raiz != null)
    {
        //MessageBox.Show(lista.raiz.nombre);
        lbtLista.Items.Add(Convert.ToString(lista.raiz.nombre));
        lista.raiz = lista.raiz.anterior; //apunta el nodo anterior
    }
}
```



PROBLEMAS



- Resolver la actividad planteada en la plataforma virtual de aprendizajes



1. Se requiere el proceso que genere un aplicativo de listas (Pilas) teniendo los siguientes requerimientos:
 - a) Ingresar Alimentos de la pila.
 - b) Generar búsquedas de elementos de la pila por cada entrada.
 - c) Generar una modificación por dicha entrada.
 - d) Eliminar un nodo de la lista de entrada.
 - e) Mostrar una pila de entrada.
 - f) Eliminar la lista de las filas por completo.
 - g) Generar una salida final cuenta.

```
Menú Pilas con listas enlazadas
1.- Insertar elemento
2.- Buscar elemento
3.- Modificar elemento
4.- Eliminar Nodo de pila
5.- Desplegar Pila
6.- Eliminar Pila
7.- Salir
Ingrese su opción : 5
IMPRIMIR PILA
20
5
12
```

PROBLEMAS

Problema 1- PILAS

1. En una feria de libros se establece el (Código, Nombre , Precio) para la visualización de una lista determinada al ingresar por 8 libros. Utilizando la estructura de una pila lo último que entra es lo primero que sale. Por lo cual se requiere una aplicación en el lenguaje C#.

- a) Apilar la lista de la pila
- b) Desapilar la pila
- c) Mostar en los resultado en el DataGridView.

3 Label

3 TextBox

2 Butoon

1 DataGridView

The screenshot shows a Windows application window titled "Form3". Inside the window, there are three text boxes for input, labeled "Codigo", "Nombre", and "Precio". To the right of the "Codigo" text box is a button labeled "apilar". To the right of the "Precio" text box is a button labeled "desapilar". Below these input fields is a DataGridView control. The DataGridView has three columns with headers "Codigo", "Nombre", and "Precio". The body of the DataGridView is currently empty.

PROBLEMAS

Problema 2- PILAS

2. Se requiere la presentación de Ventas del salón de peluquería con 10 clientes (Código, Nombre , Apellidos y Precio) con el promedio en general. Utilizando la estructura de una pila lo último que entra es lo primero que sale.

Por lo cual se requiere una aplicación en el lenguaje C#.

- a) Generar el promedio de las ventas.
 - b) Generar la búsqueda por código
 - c) Eliminar la pila por código.
- 5 Label
 - 4 TextBox
 - 3 Butoon
 - 1 DataGridView

The screenshot shows a Windows application window titled "Form2". It contains a data entry form with four labels and text boxes: "Codigo" (456), "Nombre" (MARIA), "Apellidos" (MESA), and "Precio" (30). To the right of the form are five buttons: "apilar" (highlighted with a blue border), "desapilar", "PROMEDIO", "BUSCAR", and "ELIMINAR". Below the form is a table with three columns: "Codigo", "Nombre", and "Precio". The table contains two rows of data: (456, MARIA, 30) and (123, JUAN, 15). The first row is highlighted in blue. Above the table, the text "22.5" is displayed, likely representing the average price.

Codigo	Nombre	Precio
456	MARIA	30
123	JUAN	15



CONCLUSIONES

Una pila nos permitirá introducir un nuevo elemento en la cima ("**apilar**", en inglés "**push**") y quitar el elemento que hay en la cima ("**desapilar**", en inglés "pop").

Este tipo de estructuras se suele denotar también usando las siglas "**LIFO**" (**Last In First Out**: lo último en entrar es lo primero en salir).

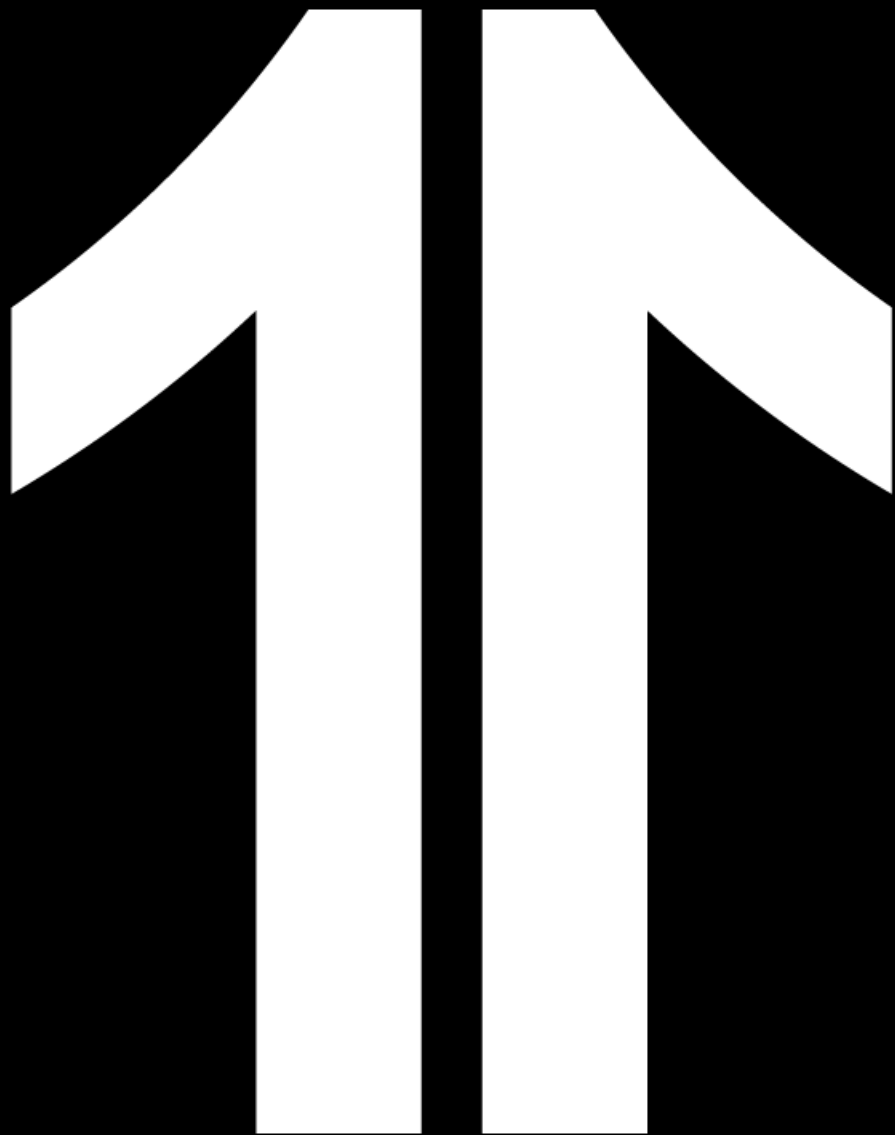
¿Preguntas o Comentarios?





REFERENCIAS BIBLIOGRÁFICAS

- HerbertSchildtFundamentosdeC#3.0
- <https://pdfcoffee.com/fundamentos-de-c-30-4-pdf-free.html>



GRACIAS

