

Facultad:	Ingeniería
Escuela:	Computación
Asignatura:	Programación con Estructuras de Datos

Tema: Estructura Pila.

Competencia

- Desarrolla sistemas de información informáticos mediante la integración de principios matemáticos, ciencia computacional y prácticas de ingeniería, considerando estándares de calidad y mejores prácticas validadas por la industria del software.

Materiales y Equipo

- Guía Número 5
- Computadora con programa Microsoft Visual C#.

Introducción Teórica

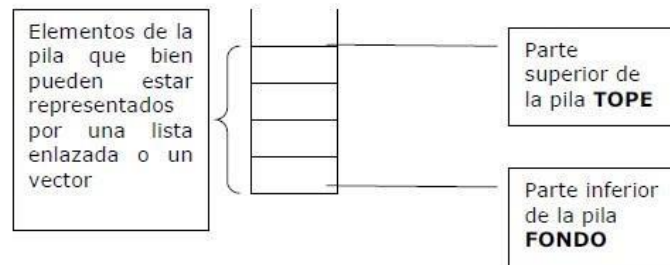
ESTRUCTURA DINÁMICA PILA

Una pila (stack) es una lista ordenada de elementos en la que todas las inserciones y supresiones se realizan por un mismo extremo de la lista. A una pila se le pueden añadir y retirar nuevos nodos únicamente de su parte superior, la cima de la pila. Por esta razón, se conoce una pila como una estructura de datos **LIFO por last-in, first out**, es decir último en entrar primero en salir.

Cuando se dice que la pila está ordenada, se quiere decir que hay un elemento al que se puede acceder primero (el que está encima de la pila), otro elemento al que se puede acceder en segundo lugar (justo el elemento que está debajo de la cima), un tercero, etc. No se requiere que las entradas se puedan comparar utilizando el operador “menor que” y pueden ser de cualquier tipo.

Se referencia una pila mediante un apuntador al elemento superior de la misma. El miembro de enlace en el último nodo de la pila se define a NULL, para indicar que se trata de la parte inferior de la pila misma.

Gráficamente una Pila puede representarse de la siguiente manera:



Las operaciones típicas de la pila incluyen: Crear pila, verificar si está vacía, sacar un elemento de la pila, meter un elemento de la pila.

Procedimiento

Ejemplo 1

1. Cree un proyecto en Visual C# en modo consola (nómbrelo como usted quiera) y cree dos clases adicionales al proyecto: la clase **nodo** y la clase **Pila**
2. En la clase **nodo** codifique lo siguiente:

```
class nodo
{
    public char info;
    //sgte es un objeto que simula el puntero
    public nodo sgte;
}
```

3. En la clase **Pila** tendremos el siguiente código (se presentan bloques por cada método):

```
class Pila
{
    public nodo tope;

    public Pila()
    {
        tope = null;
    }
}
```

Atributos y constructor

```

public void Push(char valor)
{
    nodo aux = new nodo();
    aux.info = valor;
    //si no hay nada en la pila
    if (tope == null)
    {
        //se asigna el aux como el único nodo
        tope = aux;
        //como solo hay un dato en la pila el puntero se dirige a null
        aux.sgte = null;
    }
    else
    {
        //si la pila no está vacía el dato se inserta arriba de la pila
        //es decir en el tope
        aux.sgte = tope;
        tope = aux;
    }
}

```

Método Push

```

public void mostrar()
{
    nodo puntero; puntero = tope;
    Console.WriteLine("{0}", puntero.info);
    //se recorre el puntero del tope a el fondo
    while (puntero.sgte != null)
    {
        puntero = puntero.sgte;
        Console.WriteLine("{0}", puntero.info);
    }
}

```

Método Mostrar

```

public char Pop()
{
    char valor = ' ';
    if (tope == null)
        Console.WriteLine("Pila vacia");
    else
    {
        //se elimina el último en entrar y se retorna el valor que se elimina
        valor = tope.info;
        tope = tope.sgte;
    }
    return valor;
}

```

Método Pop

4. En el Main o programa principal escribimos el siguiente código

```
static void Main(string[] args)
{
    string cadena;
    char caracter;
    Pila mipila = new Pila();
    Pila pilafrase = new Pila();

    //PARTE 1
    //SE INGRESAN CARACTERES DE MANERA INDIVIDUAL A LA PILA Y LUEGO SE MUESTRAN
    Console.WriteLine("\t\t---PARTE I---");
    //ingresamos elementos a la pila
    mipila.Push('p');
    mipila.Push('b');
    mipila.Push('z');
    mipila.Push('s');
    //mostramos el contenido de la pila
    mipila.mostrar();
    Console.ReadLine();
    //sacamos un elemento de la pila
    mipila.Pop();
    mipila.mostrar();
    Console.ReadLine();

    //PARTE 2
    //SE INGRESA UNA CADENA Y SE INGRESAN TODOS LOS
    //CARACTERES DE DICHA CADENA
    Console.WriteLine("\t\t---PARTE II---");
    Console.Write("Ingrese la palabra: "); cadena = Console.ReadLine();

    for (int i = 0; i < cadena.Length; i++)
    {
        //extrae un caracter de la cadena y lo convierte en char
        caracter = char.Parse(cadena.Substring(i, 1));
        pilafrase.Push(caracter);
    } //imprimir el contenido de la pila
    pilafrase.mostrar();
    Console.ReadLine();
}
```

5. Ejecute el programa y observe su funcionamiento.

Ejemplo 2

En este ejercicio se utilizaran funciones de pilas ya creadas en Visual C# pero se utilizan objetos por lo que habrá que hacer las conversiones necesarias.

- Cree un nuevo proyecto en Visual C# en modo consola.
- Agregue al inicio el nombre de espacio Collections de la siguiente manera: `using System.Collections;`
- Dentro del método `static void Main(string[] args)` ingrese el siguiente código:

```
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);
```

```

        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 ha finalizado.");
}

/** añade un nuevo elemento a la pila */
static void agregar(ref Stack pila)
{
    Console.WriteLine("\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("\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(" > 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(" > {0}", texto);
        Console.WriteLine(" =====");
        Console.WriteLine("\n >Presione 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");
    }
}
}

```

Desarrollo de habilidades

Ejercicio 1

Modifique los ejemplos 2 y 3 de la guía anterior (Colas) de forma que funcionen simulando pila (ejemplo 2) y atendiendo a los empleados como una pila (ejemplo 3).

Ejercicio 2

En entorno gráfico crear un programa que solicite una expresión matemática y que revise si los paréntesis están balanceados

Ejemplo: $(a+b-2)$ Paréntesis balanceado

$(a+b-2))$ Paréntesis desbalanceado

En el mismo proyecto crear un programa que lea una palabra e imprima un mensaje indicando si es palíndromo o no. Una palabra es palíndromo cuando se lee igual hacia adelante que hacia atrás. Ejemplo: oso, radar, reconocer, rotor, seres, somos, etc.

Investigación complementaria

Investigación 1.

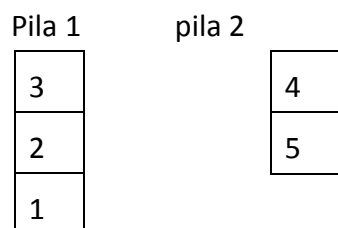
Un almacén tiene capacidad para apilar n contenedores. Cada contenedor tiene un número de identificación. Cuando se desea retirar un contenedor específico, deben retirarse primero los contenedores que están encima de él y colocarlos en otra pila, efectuar el retiro y regresarlos.

Almacén

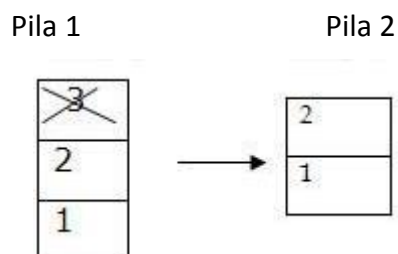
5
4
3
2
1

Se desea retirar el contenedor 3

Paso 1. Se retiran los contenedores que están encima de él y se colocan en otra pila



Paso 2. Se efectúa el retiro



Paso 3. Regresarlos

