

# Programación

---

## Ejercicio 1: Stack (Pilas)

Implementar un programa en C# que utilice una pila (Stack) para invertir el orden de una lista de elementos. El programa debe tomar una lista de números como entrada, empujar cada elemento a una pila y luego desplegar los elementos en orden inverso sacándolos de la pila.

**Hint:** Mostrar el proceso en la consola del sistema

## Ejercicio 2: Notación RPN (Pilas)

Se solicita implementar un programa en C# que evalúe expresiones aritméticas en notación postfija (también conocida como notación polaca inversa o RPN). La notación postfija es una forma de escribir expresiones matemáticas donde los operadores siguen a los operandos. Por ejemplo, la expresión "2 + 3" en notación postfija se escribiría como "2 3 +".

### Ejemplo de uso:

```
evaluateRpn (5 3 4 * + 7 -) -> 10
```

**Hint:** Mostrar el proceso en la consola del sistema

## Ejercicio 3: Filtrando valores (Colas)

Se solicita implementar un programa en C# que solicite por consola al usuario la cantidad de valores que desea alojar en una cola, posteriormente deberá solicitar cada uno de los valores vía consola hasta llegar al número establecido en el primer paso. Luego se debe aplicar un método que filtre los valores de la cola dejando sólo los números pares pero manteniendo el orden relativo de los elementos antes de las eliminaciones.

**Hint:** El usuario solo podrá cargar valores enteros en caso contrario se debe avisar del error y reiterar la solicitud del valor en cuestión.

## Ejercicio 4: Palíndromos (Colas)

Se solicita implementar un programa en C# que solicite por consola al usuario el ingreso de una cadena posteriormente se debe implementar un método que determine si la cadena ingresada es palindroma o no; en caso afirmativo retornar true y caso contrario false.

### Ejemplo de uso:

```
IsPalindrome ("Orejero ") -> True
```

```
IsPalindrome ("Automóvil ") -> False
```

## Ejercicio 5: Ordenación (Colas)

Reutilizando la carga de valores de una cola vía consola (ej3) en esta ocasión se solicita implementar un método que reciba dos parámetros, una cola de números enteros y un valor n que puede tomar dos valores 1 o -1.

En caso de recibir 1 el método deberá ordenar los valores de la cola de manera ascendente, de recibir -1 de forma descendente y en caso contrario el método no realizará la ordenación y retornará -1. Al momento de ordenar los valores no se podrá utilizar ninguna estructura de datos adicional, el procedimiento se deberá realizar con operaciones Enqueue y Dequeue. Finalmente el método deberá retornar la Cola ordenada.

## Ejercicio 6: Búsqueda I (ArrayList)

Se solicita implementar el algoritmo Binary Search en C# utilizando ArrayLists.

## Ejercicio 7: Búsqueda II (ArrayList)

Se solicita implementar el algoritmo Lineal Search en C# utilizando ArrayLists.

## Ejercicio 8: Ordenación (ArrayList)

Se solicita implementar el algoritmo SelectionSort en C# utilizando ArrayLists.

## Ejercicio 9: Ordenación (ArrayList)

Se solicita implementar el algoritmo BubbleSort en C# utilizando ArrayLists.

## Ejercicio 10: Multiplicación de matrices (List)

Se solicita implementar un método en C# que reciba por parámetro dos matrices de números enteros y calcule la multiplicación entre ambas estructuras devolviendo la matriz resultante. Posteriormente se solicita imprimir en la consola del sistema dicha matriz representando la misma de manera fidedigna en la consola del sistema. (i.e que se vea como matriz a nivel visual)