

Ejercicios básicos de colecciones y delegados en C#

Iris Pérez Aparicio

2º DAM

Campus FP Emprende Humanes

Ejercicio 1

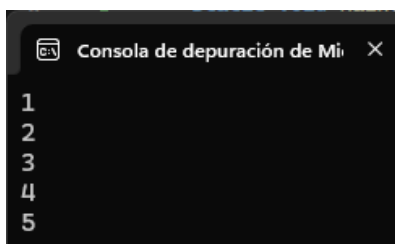
Crea una lista de enteros, añade los números del 1 al 5 y muestra todos los elementos por consola.

```
using System;

class Program
{
    static void Main()
    {
        // Creo una lista de enteros
        List<int> numeros = new List<int>();

        // Agrego los números del 1 al 5
        for (int i = 1; i <= 5; i++)
        {
            numeros.Add(i);
        }

        // Mostrar todos los elementos de la lista por consola
        foreach (int numero in numeros)
        {
            Console.WriteLine(numero);
        }
    }
}
```



Consola de depuración de Mi

```
1
2
3
4
5
```

Ejercicio 2

Crea una cola de cadenas. Añade tres nombres y luego elimina (dequeue) y muestra cada uno hasta que la cola esté vacía.

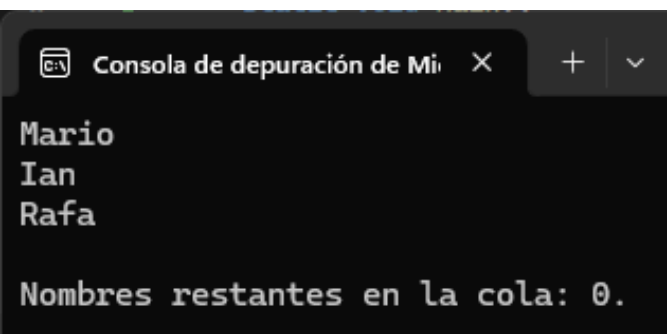
```
using System;

class Program
{
    static void Main()
    {
        // Creo una cola de cadenas
        Queue<string> nombres = new Queue<string>();

        // Agrego tres nombres
        nombres.Enqueue("Mario");
        nombres.Enqueue("Ian");
        nombres.Enqueue("Rafa");

        // Elimino y muestro cada nombre hasta que la cola esté vacía
        while (nombres.Count > 0)
        {
            string nombre = nombres.Dequeue();
            Console.WriteLine(nombre);
        }

        // Muestro que la cola está vacía
        Console.WriteLine($"Nombres restantes en la cola: {nombres.Count}");
    }
}
```



```
Consola de depuración de Mi X + v
Mario
Ian
Rafa
Nombres restantes en la cola: 0.
```

Ejercicio 3

Crea una pila de caracteres. Añade las letras 'A', 'B' y 'C'. Extrae (pop) y muestra cada letra hasta que la pila esté vacía.

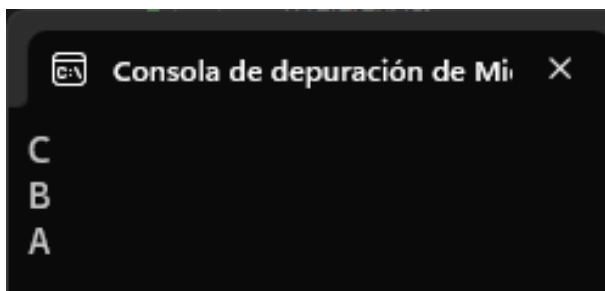
```
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        // Creo una pila de caracteres
        Stack<char> letras = new Stack<char>();

        // Agrego las letras 'A', 'B' y 'C'
        letras.Push('A');
        letras.Push('B');
        letras.Push('C');

        // Extraigo (pop) y muestro cada letra hasta que la pila esté
vacía
        while (letras.Count > 0)
        {
            char letra = letras.Pop();
            Console.WriteLine(letra);
        }

        // Muestro que la pila está vacía
        Console.WriteLine("\nNúmero de letras restantes en la pila: " +
letras.Count);
    }
}
```



Ejercicio 4

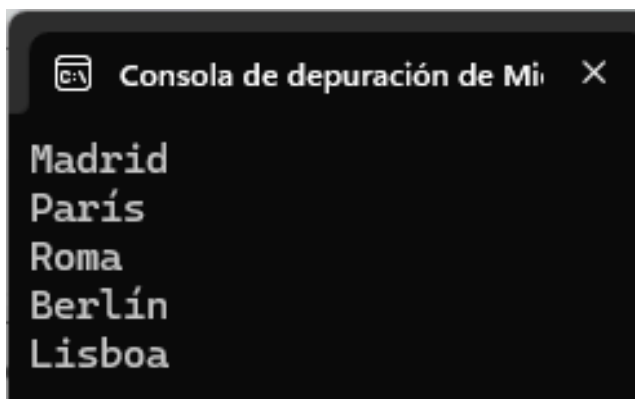
Crea un diccionario que relacione nombres de países (clave) con sus capitales (valor).
Añade al menos tres pares y muestra todas las capitales.

```
using System;

class Program
{
    static void Main()
    {
        // Creo un diccionario de países (clave) y sus capitales (valor)
        Dictionary<string, string> paisesCapitales = new
        Dictionary<string, string>()
        {
            {"España", "Madrid"},
            {"Francia", "París"}
        };

        // Añado más pares clave valor
        paisesCapitales.Add("Italia", "Roma");
        paisesCapitales.Add("Alemania", "Berlín");
        paisesCapitales.Add("Portugal", "Lisboa");

        // Muestro todas las capitales
        foreach (var capital in paisesCapitales.Values)
        {
            Console.WriteLine(capital);
        }
    }
}
```



Consola de depuración de Mi

Madrid
París
Roma
Berlín
Lisboa

Ejercicio 5

Crea un conjunto de números enteros. Añade los números 1, 2, 2, 3, 4, 4, 5. Muestra todos los elementos del conjunto y observa qué ocurre con los duplicados.

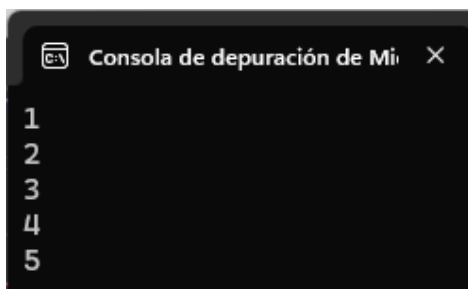
```
using System;

class Program
{
    static void Main()
    {
        // Creo un conjunto de números enteros y agrego números (algunos duplicados)
        HashSet<int> numeros = new HashSet<int> { 1, 2, 2, 3 };

        // Agrego más números
        numeros.Add(4);
        numeros.Add(4);
        numeros.Add(5);

        // Muestro todos los elementos del conjunto
        foreach (int numero in numeros)
        {
            Console.WriteLine(numero);
        }

        /* RESULTADO: En un conjunto (HashSet)
         * no se agregan números duplicados,
         * cada elemento es único */
    }
}
```



```
1
2
3
4
5
```

En un conjunto (HashSet) no se pueden agregar números duplicados, cada elemento en el conjunto es único.