# Ejercicios Colecciones BCL Fáciles

Descargar estos ejercicios

## Índice


- Ejercicio 1
- Ejercicio 2
- Ejercicio 2

# Ejercicio 1

Crea un **tipo valor corectamente** para representar un componente electrónico. De forma que tendrá dos campos:

- uno para el **Nombre** del componente.
- otro para el **Peso** (float) del componente.

Además deberás añadir el constructor y el método ToString.

En el programa principal implementa los siguientes **métodos como cuerpo de expresión**, usando el TAD cola **Queue** de Componentes:

- CreaCola, que inicializa una cola y la devuelve.
- Primero: devuelve, sin desencolar, el elemento que ocupe la primera posición.
- Encola, introduce un nuevo elemento en la cola (ambos valores pasados por parámetros).
- SacaCola, elimina el elemento en la cabeza de la cola y lo devuelve.
- Vacia, averigua si una cola no tiene nada en su interior.

En la Main crea el código necesario para probar todos los métodos.

**Solucion:**

```
public struct Componente
{
    public readonly string Elemento;
    public readonly float Peso;

    public Componente(string elemento, float peso)
    {
        Elemento = elemento;
        Peso = peso;
    }
    override public string ToString()
    {
        return $"Elemento: {Elemento} - Peso: {Peso}";
    }
}

class Program
{

    public static Queue<Componente> CreaCola() => new Queue<Componente>();
    public static Componente Primero(Queue<Componente> cola) => cola.Peek();
    public static void Encola(Queue<Componente> cola, Componente componente) => cola.Enq
    public static Componente SacaCola(Queue<Componente> cola) => cola.Dequeue();
    public static bool Vacia(Queue<Componente> cola) => cola.Count == 0;

    static void Main(string[] args)
    {
        Queue<Componente> cola = CreaCola();
        Encola(cola, new Componente("Resistencia", 0.1f));
        Encola(cola, new Componente("Condensador", 0.2f));
        Encola(cola, new Componente("Transistor", 0.3f));
        Console.WriteLine(Primero(cola));
        Console.WriteLine($"La cola tiene {cola.Count} elementos");
        Console.WriteLine($"La cola está vacía: {Vacia(cola)}");
        Console.WriteLine(SacaCola(cola));
        Console.WriteLine($"La cola tiene {cola.Count} elementos");
        Console.WriteLine(Primero(cola));

        Console.ReadLine();
    }
```

```
}
```

## Ejercicio 2

Usando el tipo valor del ejercicio anterior, crea una Pila de la BCl con los siguientes elementos:

- "Cobre", 10.5f
- "Plata", 20.5f
- "Oro", 30.5f
- "Platino", 40.5f
- "Hierro", 50.5f

Deberás invertir la pila, mostrarla y volver a dejarla igual y mostrarla. Para ello tendrás que usar, al menos, una cola auxiliar.

**Solucion:**

```
 static void Main(string[] args)
{
        Stack<Componente> pila = new Stack<Componente>();
        Queue<Componente> cola = new Queue<Componente>();
        Stack<Componente> pila2 = new Stack<Componente>();

        pila.Push(new Componente("Cobre", 10.5f));
        pila.Push(new Componente("Plata", 20.5f));
        pila.Push(new Componente("Oro", 30.5f));
        pila.Push(new Componente("Platino", 40.5f));
        pila.Push(new Componente("Hierro", 50.5f));

        foreach (var c in pila) Console.Write($"{c} - ");
        while(pila.Count > 0) cola.Enqueue(pila.Pop());
        while(cola.Count > 0) pila.Push(cola.Dequeue());
        Console.WriteLine("\n\n");
        foreach (var c in pila)  Console.Write($"{c} - ");
        while(pila.Count > 0)   pila2.Push(pila.Pop());
         Console.WriteLine("\n\n");
        foreach (var c in pila2)  Console.Write($"{c} - ");

        Console.ReadLine();
}
```

## Ejercicio 2

Idem del anterior pero usando solamente la colección pila como auxiliar para hacer la inversión.

```
static void Main(string[] args)
{
        Stack<Componente> pila = new Stack<Componente>();
        Queue<Componente> cola = new Queue<Componente>();
        Stack<Componente> pila2 = new Stack<Componente>();

        pila.Push(new Componente("Cobre", 10.5f));
        pila.Push(new Componente("Plata", 20.5f));
        pila.Push(new Componente("Oro", 30.5f));
        pila.Push(new Componente("Platino", 40.5f));
        pila.Push(new Componente("Hierro", 50.5f));

        foreach (var c in pila) Console.Write($"{c} - ");
        while (pila.Count > 0) pila2.Push(pila.Pop());
        Console.WriteLine("\n\n");
        foreach (var c in pila2) Console.Write($"{c} - ");
        while (pila2.Count > 0) pila.Push(pila2.Pop());
        Console.WriteLine("\n\n");
        foreach (var c in pila) Console.Write($"{c} - ");

        Console.ReadLine();
}
```