

2da Evaluación de Laboratorio de Computación II

thatwrongboy@gmail.com [Cambiar de cuenta](#)



* Indica que la pregunta es obligatoria

Comienzo de la Evaluación

Proceso de conversión de un JSON / flujo de bytes (Stream) en un objeto. * 1 punto

Seleccione cual es la definición correcta:

- ☐ Hilos
- ☐ Serialización
- ☐ Colecciones
- ☐ Deserialización
- ☐ Propiedades



```
using System;
```

* 1 punto

```
namespace Estructura
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            Trabajador Trabajador1 = new Trabajador(120000, 25000);
```

```
            Trabajador1.modificaSalario(Trabajador1, 10000);
```

```
            Console.WriteLine(Trabajador1);
```

```
        }
```

```
    }
```

```
    public struct Trabajador
```

```
    {
```

```
        public double salarioBase, comision;
```

```
        public Trabajador (int salarioBase, int comision)
```

```
        {
```

```
            this. salarioBase = salarioBase;
```

```
            this.comision = comision;
```

```
        }
```

```
        public override string ToString()
```

```
        {
```

```
            return string.Format("Salario y comisión del Trabajador {0}, {1}, ",  
this.salarioBase, this.comision);
```

```
        }
```

```
        public void modificaSalario(Trabajador traba, double aumentar)
```

```
        {
```

```
            traba.salarioBase += aumentar;
```

```
            traba.comision += aumentar*10;
```

```
        }
```

```
    }
```

```
}
```

Indique cual es la respuesta, si se cambia struct por class en el programa:



Salario y comisión del Trabajador 130000, 35000



- ☐ Ninguna de las anteriores
- ☐ Salario y comisión del Trabajador 130000, 125000
- ☐ Salario y comisión del Trabajador 220000, 125000
- ☐ Salario y comisión del Trabajador 120000, 25000

Representa una colección de pares clave-valor organizados según la clave.

* 1 punto

Seleccione cual es la instrucción correcta:

- ☐ LinkedListNode
- ☐ SortedDictionary
- ☐ KeyValuePair
- ☐ Dictionary
- ☐ LinkedList

Para los tipos Enum que instrucción o palabra reservada por C# se utiliza * 1 punto para que se convierta a cadena de caracteres:

Escriba la sentencia con todos sus detalles:

Tu respuesta



```
using System;  
using System.IO;
```

*

1 punto

```
namespace archivos  
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
  
            TextReader Leer_archivo;  
            Leer_archivo = new StreamReader("archivo.doc");  
            Console.WriteLine(Leer_archivo.ReadToEnd());  
            Leer_archivo.Close();  
  
        }  
    }  
}
```

Para obtener el punto, debe seleccionar todo lo que hace el código fuente correctamente:

- ☐ Lee lo escrito en el archivo
- ☐ Escribe un mensaje dentro del archivo
- ☐ Crea un archivo
- ☐ Añade el mensaje dentro del archivo sin borrarlo
- ☐ Cierra el archivo



```
using System;  
using System.Threading;  
using System.Threading.Tasks; // Necesario para utilizar la clase Task.
```

* 1 punto

```
namespace MyApp  
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
  
            tarea.Start();  
  
            Task tarea2 = new Task(() =>  
            {  
                for (int j = 0; j < 100; j++)  
                {  
  
                    Thread.Sleep(1000);  
                    Console.WriteLine("Tarea Correspondiente al hilo: " +  
miThread + " Ejecutandose desde el Main");  
                }  
            });  
  
            Console.ReadLine();  
        }  
  
        static void EjecutarTarea()  
        {  
            for (int i = 0; i < 100; i++)  
            {  
                var miThread = Thread.CurrentThread.ManagedThreadId;  
                Thread.Sleep(1000);  
                Console.WriteLine("Esta vuelta de bucle corresponde al Thread: "  
+ miThread);  
            }  
        }  
    }  
}
```

En el siguiente código fuente, indique cuales son las líneas que faltan:



- ☐ Task tarea = new Task(EjecutarTarea);
- ☐ tarea2.Start();
- ☐ var miThread = Thread.CurrentThread.ManagedThreadId;
- ☐ Thread tarea = new Thread(EjecutarTarea);
- ☐ var miThread = Task.CurrentTask.ManagedTaskId;



```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;
public class Test
{
    public static void Main(string[] args)
    {
        Stack<int> numeros = new Stack<int>();

        Console.WriteLine("¿Numeros: ");

        foreach (int numero in new int[] { 10, 8, 6, 7, 8, 9 })
        {

            numeros.Push(numero);

        }

        Console.WriteLine("Recorriendo la Pila: ");

        foreach (int elemento in numeros)
        {

            Console.WriteLine(elemento);

        }
        Console.WriteLine("Eliminando la Pila: ");
        numeros.Pop();
        foreach (int elemento in numeros)
        {

            Console.WriteLine(elemento);

        }

    }
}
```

*

1 punto

Para obtener el punto debe indicar correctamente, todas las sentencias que se realizan en el programa:



- ☐ Elimina un elemento al final
- ☐ Introduce un elemento al inicio
- ☐ Se crea una Pila
- ☐ Agrega un elemento al final
- ☐ Elimina un elemento al inicio
- ☐ Se crea una Cola

Representa un elemento clave-valor *

1 punto

Seleccione cual es la instrucción correcta:

- ☐ Dictionary
- ☐ KeyValuePair
- ☐ SortedDictionary
- ☐ Ninguno
- ☐ Stack

Colección de elementos de tipo clave-valor organizados en función del código hash de la clave. Sólo se puede tener acceso a los elementos de la colección a través de la clave del elemento.

* 1 punto

Seleccione cual es la instrucción correcta:

- ☐ SortedList
- ☐ Hashtable
- ☐ Queue
- ☐ Dictionary
- ☐ ArrayList



Representa un nodo en una colección LinkedList. Esta clase no se puede heredar. * 1 punto

Seleccione cual es la instrucción correcta:

- ☐ LinkedList
- ☐ ArrayList
- ☐ SortedList
- ☐ LinkedListNode
- ☐ List

Es una forma de agrupar objetos, sus elementos no tienen por qué compartir el mismo tipo de datos (por ejemplo, los objetos de cualquier tipo se pueden agrupar en una sola colección del tipo Object), y su número de elementos puede aumentar y reducirse dinámicamente a medida que cambian las necesidades del programa. * 1 punto

Seleccione cual es el concepto correcto:

- ☐ Serialización
- ☐ Propiedades
- ☐ Concurrencias
- ☐ Colecciones
- ☐ Hilos



Es una lista del tipo FIFO: First In First Out - primero en entrar, primero en salir. * 1 punto

Seleccione cual es la instrucción correcta:

- ☐ Hashtable
- ☐ Queue
- ☐ ArrayList
- ☐ Stack
- ☐ SortedList



* 1 punto

```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;
public class Test
{
    public static void Main(string[] args)
    {
        LinkedList<int> numeros = new LinkedList<int>();

        Console.WriteLine("¿Numeros ");

        foreach (int numero in new int[] {10,8,6,7,8,9})
        {

            numeros.AddFirst(numero);

        }

        numeros.Remove(8);

        LinkedListNode<int> nodoImportante = new
LinkedListNode<int>(15);
        numeros.AddLast(nodoImportante);
        Console.WriteLine("Los elementos que se introducen son los
siguientes: ");

        foreach (int elemento in numeros)
        {

            Console.WriteLine(elemento);

        }

        for (LinkedListNode<int> nodo=numeros.First; nodo!=null;
nodo=nodo.Next)
        {
            int numero = nodo.Value;
            Console.WriteLine(numero);

        }
    }
}
```



```
}  
}
```

Para obtener el punto debe indicar correctamente, todas las sentencias que se realizan en el programa:

- ☐ Agrega un nodo al inicio de la lista doblemente enlazada
- ☐ Trabaja con un nodo en una colección de la lista doblemente enlazada
- ☐ Remueve la ultima ocurrencia del valor de la lista doblemente enlazada
- ☐ Crea un lista doblemente enlazada
- ☐ Remueve la primera ocurrencia del valor de la lista doblemente enlazada
- ☐ Agrega un nodo al final de la lista doblemente enlazada

Indique que se puede hacer en los siguientes segmentos de memoria: * 2 puntos

Tome en cuenta que para obtener los 2 puntos, no se debe fallar en ninguna de las diferencias:

	Objetos	Los llamados a las funciones	Variables superglobales	Los parámetros de las funciones llamadas	Las variables locales	Otra información de alto uso de memoria
Heap	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stack	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;
public class Test
{
    public static void Main(string[] args)
    {
        List<int> numeros = new List<int>();
        Console.WriteLine("¿Para Salir precionar 0: ");

        int elem = 1;

        while (elem != 0 )
        {

            elem = int.Parse(Console.ReadLine());
            numeros.Add(elem);

        }

        numeros.RemoveAt(numeros.Count - 1);
        Console.WriteLine("Los elementos que se introducen son los
siguientes: ");

        foreach (int elemento in numeros)
        {

            Console.WriteLine(elemento);

        }
    }
}
```

* 1 punto

Para obtener el punto debe indicar correctamente, todas las sentencias que se realizan en el programa:

- ☐ Declara una lista de colección
- ☐ Agrega un nodo al inicio de la lista
- ☐ Agrega un elemento u objeto a final de la lista
- ☐ Remueve un nodo al final de la lista



☐ Remueve el elemento a la especificación del índice de la lista

Es una versión ordenada de Hashtable. En este tipo de colección, se puede acceder a los elementos por la clave o por el índice.

* 1 punto

Seleccione cual es la instrucción correcta:

- ☐ ArrayList
- ☐ SortedList
- ☐ Stack
- ☐ Hashtable
- ☐ Dictionary

Es una lista del tipo LIFO: Last In First Out - último en entrar, primero en salir.

* 1 punto

Seleccione cual es la instrucción correcta:

- ☐ Stack
- ☐ ArrayList
- ☐ SortedList
- ☐ Hashtable
- ☐ Queue



★ 1 punto

```
using System;
using System.Runtime.CompilerServices;
using static System.Runtime.InteropServices.JavaScript.JSType;

namespace MyApp // Note: actual namespace depends on the project
name.
{
    internal class Program
    {
        private static int acumulador = 0;
        static void Main(string[] args)
        {
            Parallel.For(0, 100, dato =>
            {
                Console.WriteLine($"Acumulador vale {acumulador}. Tarea " +
                    $"realizada por el hilo
{Thread.CurrentThread.ManagedThreadId}");
                if (acumulador == 0)
                {
                    acumulador += dato;
                    Thread.Sleep(100);
                }
                else
                {
                    acumulador += dato;
                    Thread.Sleep(100);
                }
            });
        }
    }
}
```

En el siguiente código fuente, indique cual es la respuesta correcta:

```
Microsoft Visual Studio Debug Console
Acumulador vale 0. Tarea realizada por el hilo 9
Acumulador vale 0. Tarea realizada por el hilo 6
Acumulador vale 0. Tarea realizada por el hilo 4
Acumulador vale 0. Tarea realizada por el hilo 8
Acumulador vale 0. Tarea realizada por el hilo 1
Acumulador vale 1. Tarea realizada por el hilo 4
Acumulador vale 1. Tarea realizada por el hilo 9
Acumulador vale -1. Tarea realizada por el hilo 1
Acumulador vale -1. Tarea realizada por el hilo 6
Acumulador vale -1. Tarea realizada por el hilo 9
Acumulador vale -81. Tarea realizada por el hilo 4
Acumulador vale -158. Tarea realizada por el hilo 1
Acumulador vale -158. Tarea realizada por el hilo 9
Acumulador vale -83. Tarea realizada por el hilo 4
Acumulador vale -83. Tarea realizada por el hilo 6
Acumulador vale -83. Tarea realizada por el hilo 9
Acumulador vale -83. Tarea realizada por el hilo 1
Acumulador vale -83. Tarea realizada por el hilo 4
Acumulador vale 36. Tarea realizada por el hilo 6
Acumulador vale 36. Tarea realizada por el hilo 9
Acumulador vale 143. Tarea realizada por el hilo 4
Acumulador vale 143. Tarea realizada por el hilo 6
Acumulador vale 143. Tarea realizada por el hilo 1
Acumulador vale 132. Tarea realizada por el hilo 8
Acumulador vale 143. Tarea realizada por el hilo 9
Acumulador vale 161. Tarea realizada por el hilo 4
Acumulador vale 154. Tarea realizada por el hilo 18
Acumulador vale 154. Tarea realizada por el hilo 6
Acumulador vale 53. Tarea realizada por el hilo 1
```

Acumulador vale 45. Tarea realizada por el hilo 6

☐ Opción 3

☐ Ninguna de las anteriores

```
Microsoft Visual Studio Debug Console
Acumulador vale 0. Tarea realizada por el hilo 10
Acumulador vale 0. Tarea realizada por el hilo 8
Acumulador vale 0. Tarea realizada por el hilo 6
Acumulador vale 0. Tarea realizada por el hilo 1
Acumulador vale 0. Tarea realizada por el hilo 4
Acumulador vale -51. Tarea realizada por el hilo 6
Acumulador vale -51. Tarea realizada por el hilo 4
Acumulador vale -51. Tarea realizada por el hilo 8
Acumulador vale -51. Tarea realizada por el hilo 10
Acumulador vale -51. Tarea realizada por el hilo 1
Acumulador vale -210. Tarea realizada por el hilo 4
Acumulador vale -210. Tarea realizada por el hilo 6
Acumulador vale -242. Tarea realizada por el hilo 1
Acumulador vale -242. Tarea realizada por el hilo 10
Acumulador vale -242. Tarea realizada por el hilo 8
Acumulador vale -374. Tarea realizada por el hilo 6
Acumulador vale -374. Tarea realizada por el hilo 4
Acumulador vale -408. Tarea realizada por el hilo 10
Acumulador vale -408. Tarea realizada por el hilo 8
Acumulador vale -408. Tarea realizada por el hilo 1
Acumulador vale -549. Tarea realizada por el hilo 4
Acumulador vale -549. Tarea realizada por el hilo 6
Acumulador vale -585. Tarea realizada por el hilo 1
Acumulador vale -585. Tarea realizada por el hilo 8
Acumulador vale -585. Tarea realizada por el hilo 10
Acumulador vale -729. Tarea realizada por el hilo 6
Acumulador vale -729. Tarea realizada por el hilo 4
Acumulador vale -767. Tarea realizada por el hilo 8
Acumulador vale -767. Tarea realizada por el hilo 10
Acumulador vale -767. Tarea realizada por el hilo 1
```

☐ Opción 2

```
Microsoft Visual Studio Debug Console
Acumulador vale 0. Tarea realizada por el hilo 1
Acumulador vale 0. Tarea realizada por el hilo 6
Acumulador vale 0. Tarea realizada por el hilo 8
Acumulador vale 0. Tarea realizada por el hilo 9
Acumulador vale 150. Tarea realizada por el hilo 4
Acumulador vale 151. Tarea realizada por el hilo 9
Acumulador vale 151. Tarea realizada por el hilo 4
Acumulador vale 151. Tarea realizada por el hilo 6
Acumulador vale 151. Tarea realizada por el hilo 1
Acumulador vale 310. Tarea realizada por el hilo 1
Acumulador vale 310. Tarea realizada por el hilo 6
Acumulador vale 310. Tarea realizada por el hilo 8
Acumulador vale 310. Tarea realizada por el hilo 4
Acumulador vale 310. Tarea realizada por el hilo 9
Acumulador vale 474. Tarea realizada por el hilo 9
Acumulador vale 474. Tarea realizada por el hilo 8
Acumulador vale 474. Tarea realizada por el hilo 6
Acumulador vale 474. Tarea realizada por el hilo 1
Acumulador vale 649. Tarea realizada por el hilo 1
Acumulador vale 649. Tarea realizada por el hilo 8
Acumulador vale 649. Tarea realizada por el hilo 4
Acumulador vale 829. Tarea realizada por el hilo 1
Acumulador vale 829. Tarea realizada por el hilo 6
Acumulador vale 829. Tarea realizada por el hilo 9
Acumulador vale 829. Tarea realizada por el hilo 4
```

☐ Opción 1




```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;

public class Test
{
    public static void Main(string[] args)
    {
        Dictionary<string, int> edades = new Dictionary<string, int>();

        Console.WriteLine("Edades: ");

        edades.Add("Fernando", 45);
        edades.Add("Natalia", 15);
        edades["Pedro"] = 34;
        edades["Maria"] = 23;

        foreach (KeyValuePair<string, int> persona in edades)
        {

            Console.WriteLine("Nombre: {0} Edad {1}", persona.Key,
persona.Value);
        }

    }
}
```

* 1 punto

Para obtener el punto debe indicar correctamente, que colecciones se aplican en el programa:

- ☐ Diccionario ordenado
- ☐ Lista ordenada
- ☐ Par clave-valor
- ☐ Lista de arreglo
- ☐ Diccionario



```
using System;
using System.Threading;
using System.Threading.Tasks;

namespace Hilo
{
    internal class Program
    {
        static void Main(string[] args)
        {

            var hilo1 = new Thread(() =>
            {
                for (int i = 0; i < 5; i++)
                {
                    Console.WriteLine("Hilo 1");
                    Thread.Sleep(1000);
                }

            });
            hilo1.Start();

            Thread hilo2 = new Thread(segundo);
            hilo2.Start();

            Thread hilo3 = new Thread(tercero);
            hilo3.Start();
        }
        static void segundo()
        {
            for (int i = 0; i < 5; i++)
            {
                Console.WriteLine("Hilo 2");
                Thread.Sleep(1000);
            }
        }
        static void tercero()
        {
            for (int i = 0; i < 5; i++)
            {
```

*

1 punto



```
        Console.WriteLine("Hilo 3");  
        Thread.Sleep(1000);  
    }  
}  
}
```

En el siguiente código fuente, indique cuales son las líneas que faltan:

- ☐ tareaTerminada.TrySetResult(true);
- ☐ var tareaTerminada = new ThreadCompletionSource<bool>();
- ☐ var tareaTerminada = new TaskCompletionSource<bool>();
- ☐ tareaTerminada.TrySetFinally(true);
- ☐ var resultado = tareaTerminada.Task.Result;
- ☐ var resultado = tareaTerminada.Thread.Result;

Seleccione las líneas que son necesarias para crear una propiedad. *

1 punto

Tome en cuenta que para obtener el punto, se debe tener las tres líneas del formato escritas de forma correcta:

- ☐ { set { return myVar; } get { myVar = value; }}
- ☐ { get { return myVar; } set { myVar = value; }}
- ☐ private int MyProperty
- ☐ private int myVar;
- ☐ public int myVar;
- ☐ public int MyProperty



Implementa una matriz cuyo tamaño aumenta o disminuye dinámicamente.

* 1 punto

Seleccione cual es la instrucción correcta:

- ☐ LinkedListNode
- ☐ LinkedList
- ☐ List
- ☐ ArrayList
- ☐ SortedList

Para los tipos Enum que **símbolo** se utiliza para trabajar con valores nulos (null).

* 1 punto

Tu respuesta

La clase base abstracta de todos los tipos de enumeración. Proporciona una serie de métodos para obtener información sobre un tipo de enumeración y sus valores.

* 1 punto

Seleccione cual es la clase correcta:

- ☐ Hilos
- ☐ Delegados
- ☐ Class
- ☐ Struct
- ☐ Enum



Representa una lista doblemente enlazada. *

1 punto

Seleccione cual es la instrucción correcta:

- ☐ List
- ☐ ArrayList
- ☐ LinkedList
- ☐ SortedList
- ☐ LinkedListNode

Representa una colección de pares clave-valor clasificados
ascendentemente por la clave.

* 1 punto

Seleccione cual es la instrucción correcta:

- ☐ LinkedList
- ☐ ArrayList
- ☐ LinkedListNode
- ☐ List
- ☐ SortedList



```
using System;  
using System.IO;
```

*

1 punto

```
namespace archivos  
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
  
            StreamWriter archivo = File.AppendText("archivo.doc");  
            string mensaje;  
            mensaje= Console.ReadLine();  
            archivo.WriteLine(mensaje);  
            archivo.Close();  
            Console.ReadKey();  
  
        }  
    }  
}
```

Para obtener el punto, debe seleccionar todo lo que hace el código fuente correctamente:

- ☐ Crea un archivo
- ☐ Escribe un mensaje dentro del archivo
- ☐ Cierra el archivo
- ☐ Añade el mensaje dentro del archivo sin borrarlo
- ☐ Lee lo escrito en el archivo



```
using System;  
using System.IO;
```

*

1 punto

```
namespace archivos  
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
            TextWriter archivo;  
            archivo = new StreamWriter("archivo.doc");  
            string mensaje;  
            mensaje= Console.ReadLine();  
            archivo.WriteLine(mensaje);  
            archivo.Close();  
            Console.Clear();  
            Console.WriteLine("Se ha guardado el archivo ...");  
            Console.ReadKey();  
        }  
    }  
}
```

Para obtener el punto, debe seleccionar todo lo que hace el código fuente correctamente:

- ☐ Crea un archivo
- ☐ Añade el mensaje dentro del archivo sin borrarlo
- ☐ Lee lo escrito en el archivo
- ☐ Cierra el archivo
- ☐ Escribe un mensaje dentro del archivo



```
using System;
```

* 1 punto

```
namespace MyApp // Note: actual namespace depends on the project  
name.
```

```
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
            for (int i = 0; i < 500; i++)  
            {  
  
            }  
            Console.ReadLine();  
        }  
  
        {  
            int nTarea = (int)o;  
            Console.WriteLine($"El correo N°:  
{Thread.CurrentThread.ManagedThreadId} ha comenzado su tarea " +  
nTarea);  
            Thread.Sleep(1000);  
            Console.WriteLine($"El correo N°:  
{Thread.CurrentThread.ManagedThreadId} ha terminado su tarea " +  
nTarea);  
        }  
    }  
}
```

En el siguiente código fuente, indique cuales son las líneas que faltan:

- ☐ static void EjecutarTarea(Object o)
- ☐ ThreadPool.QueueUserWorkItem(EjecutarTarea, i);
- ☐ static void EjecutarTarea()
- ☐ int nTarea = o;
- ☐ ThreadPool.StarkUserWorkItem(EjecutarTarea);
- ☐ int nTarea = (int)o;




```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;
public class Test
{
    public static void Main(string[] args)
    {
        Queue<int> numeros = new Queue<int>();

        Console.WriteLine("¿Numeros: ");

        foreach (int numero in new int[] { 10, 8, 6, 7, 8, 9 })
        {

            numeros.Enqueue(numero);

        }

        Console.WriteLine("Recorriendo la Cola: ");

        foreach (int elemento in numeros)
        {

            Console.WriteLine(elemento);

        }
        Console.WriteLine("Eliminando la cola: ");
        numeros.Dequeue();
        foreach (int elemento in numeros)
        {

            Console.WriteLine(elemento);

        }

    }
}
```

*

1 punto

Para obtener el punto debe indicar correctamente, todas las sentencias que se realizan en el programa:



- ☐ Elimina un elemento al inicio
- ☐ Agrega un elemento al final
- ☐ Se crea una Pila
- ☐ Se crea una Cola
- ☐ Elimina un elemento al final
- ☐ Introduce un elemento al inicio

Es un diccionario ordenado con operaciones de inserción y eliminación, * 1 punto
lo que hace de él una alternativa útil a SortedList.

Seleccione cual es la instrucción correcta:

- ☐ KeyValuePair
- ☐ SortedDictionary
- ☐ Stack.
- ☐ Dictionary
- ☐ Hashtable



* 1 punto

```
using System;
using System.Runtime.CompilerServices;

namespace MyApp // Note: actual namespace depends on the project
name.
{
    internal class Program
    {
        static void Main(string[] args)
        {

            Console.ReadLine();
        }
        static void EjecutarTarea()
        {
            for (int i = 0; i < 10; i++)
            {
                var miThread = Thread.CurrentThread.ManagedThreadId;
                Thread.Sleep(1000);
                Console.WriteLine("Esta vuelta de bucle corresponde al Thread: "
+ miThread);
            }
        }

        {
            for (int i = 0; i < 10; i++)
            {
                var miThread = Thread.CurrentThread.ManagedThreadId;
                Thread.Sleep(1000);
                Console.WriteLine("Esto es otra tarea Tarea Correspondiente al
hilo: " + miThread);
            }
        }
    }
}
```

En el siguiente código fuente, indique cuales son las respuestas correctas:

☐ Task tareas = Task.Run(() => EjecutarTarea());



- ☐ Task tareas2 = tareas.ContinueWith(EjecutarOtraTarea);
- ☐ static void EjecutarOtraTarea(Task obj)
- ☐ Thread tareas = Thread.Run() => EjecutarTarea();
- ☐ Task tareas = tareas.ContinueWith(EjecutarTarea);
- ☐ static void EjecutarOtraTarea()

Representa una matriz unidimensional dinámica. *

1 punto

Seleccione cual es la instrucción correcta:

- ☐ List
- ☐ SortedList
- ☐ LinkedList
- ☐ ArrayList
- ☐ LinkedListNode

Atrás

Siguiente

Página 2 de 3

Borrar

formulario

Nunca envíes contraseñas a través de Formularios de Google.

Este contenido no ha sido creado ni aprobado por Google. [Denunciar abuso](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

Google Formularios

