

**Pasos a seguir:**

- Guardar este documento en nuestro equipo.
- Renombrar según el siguiente ejemplo: ApellidoNombre\_ActividadesUD4
- En el transcurso de las clases, el alumnado irá completando el documento según se indique en el enunciado de las actividades.

<b>Actividades 4.1.....</b>	<b>2</b>
<b>Actividades 4.2.....</b>	<b>3</b>
<b>Actividades 4.3.....</b>	<b>4</b>
<b>Actividades 4.4.....</b>	<b>5</b>

## Actividades 4.1

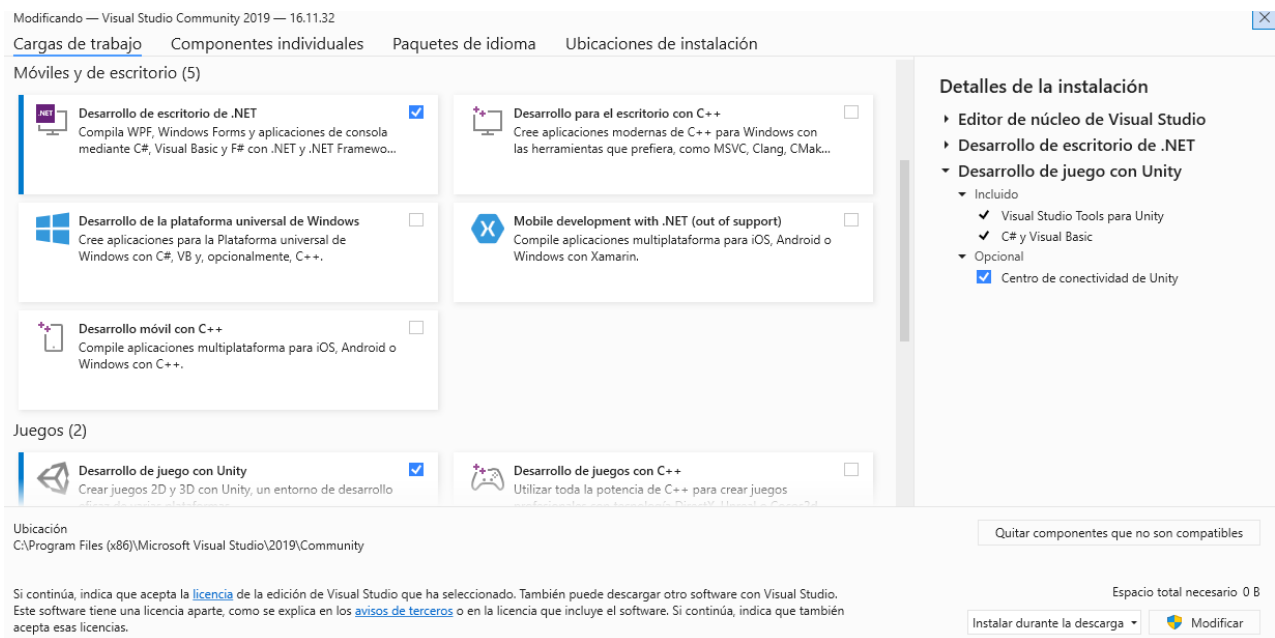
Siguiendo las indicaciones del apartado ‘Depurar el código’ del Pdf ‘Visual Studio\_2019\_Depuración y pruebas’, añadir las capturas de pantalla indicadas a la actividad de la plataforma Moodle

### Ref. Web

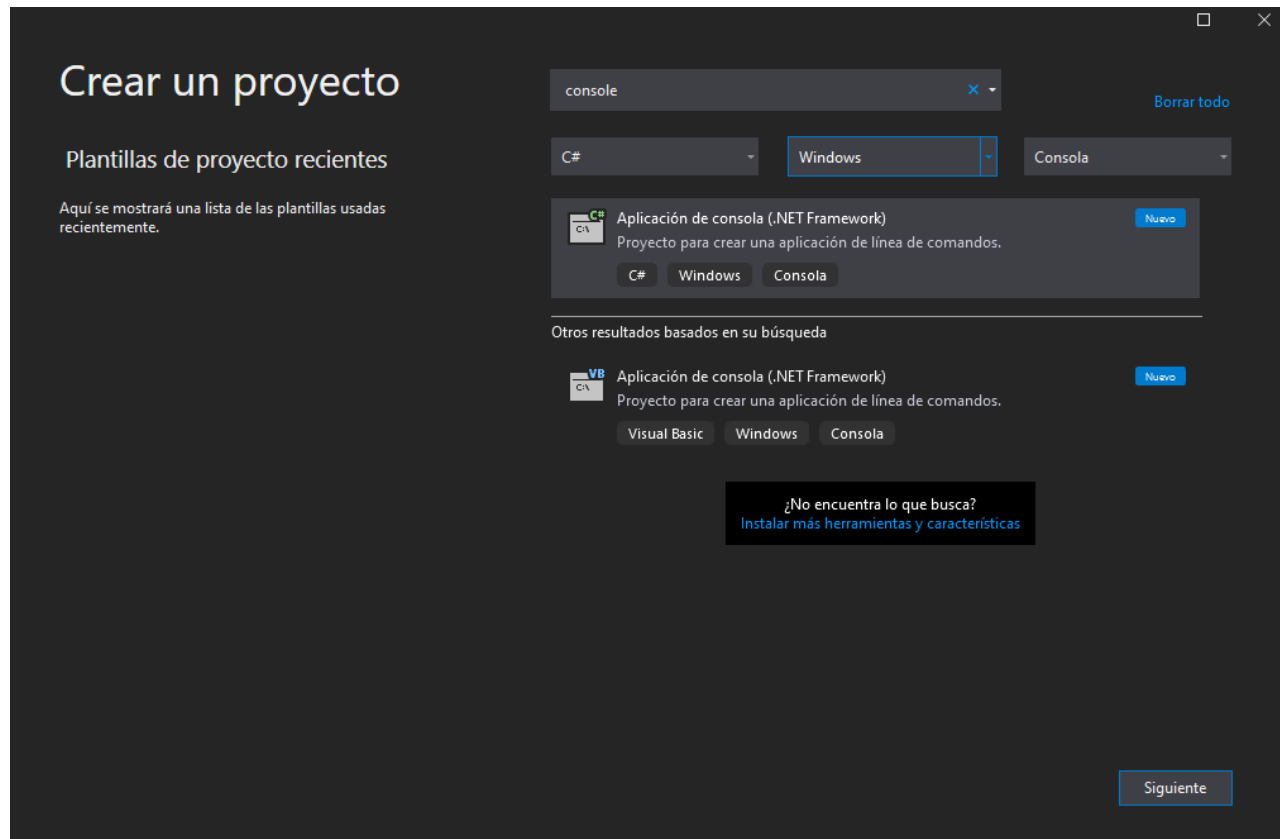
Capturas a añadir:

- Proyecto creado

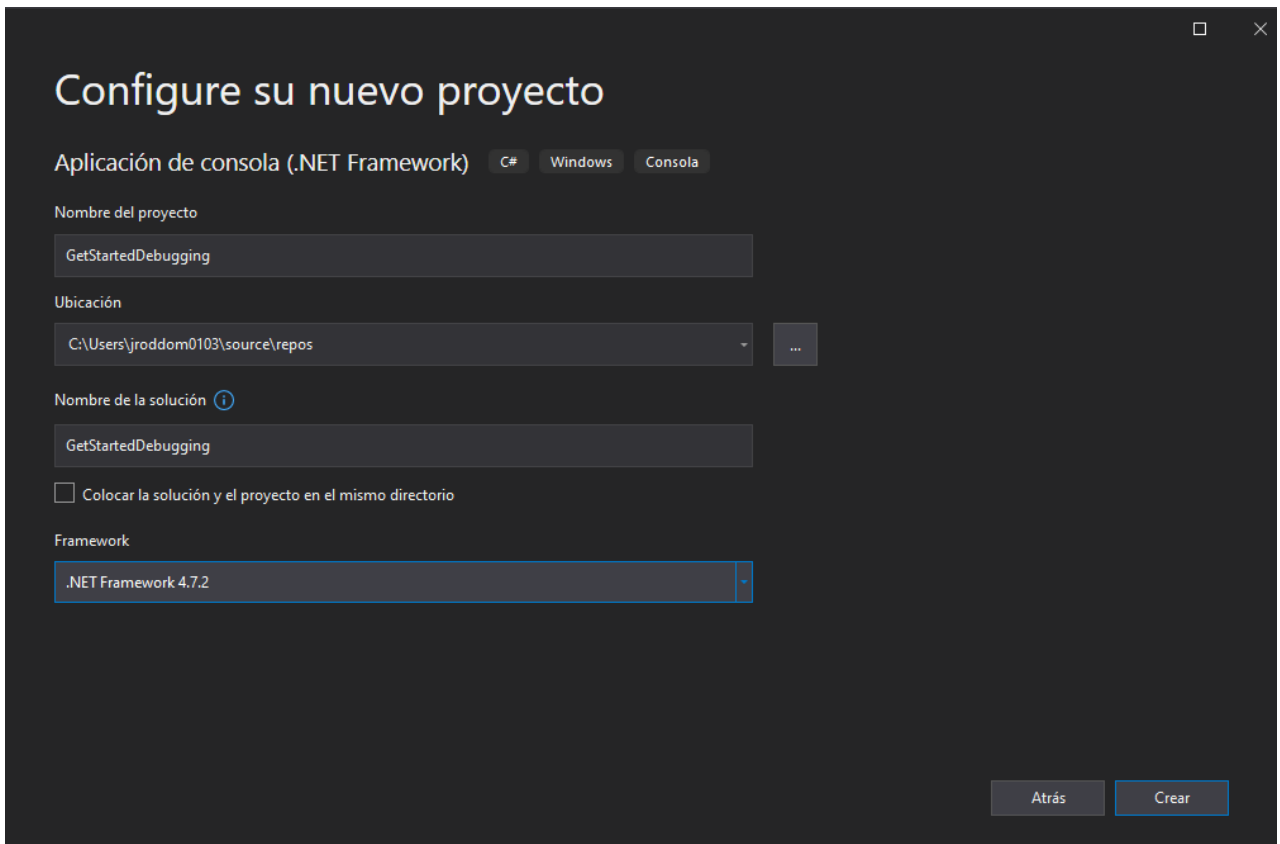
Utilizamos Visual Studio 2019, es decir, una versión anterior. Esto lo hacemos para tener una versión más estable con más errores corregidos (downgrade). Mientras tanto, se trabaja en una versión más reciente, con el fin de tener el software actualizado.



Instalamos el framework de .NET.



Seleccionamos el framework de .NET para poder hacer un proyecto con las características que nos ofrece, específicamente una aplicación de consola mediante C#.



Configure su nuevo proyecto

Aplicación de consola (.NET Framework) C# Windows Consola

Nombre del proyecto

GetStartedDebugging

Ubicación

C:\Users\jroddom0103\source\repos

Nombre de la solución ⓘ

GetStartedDebugging

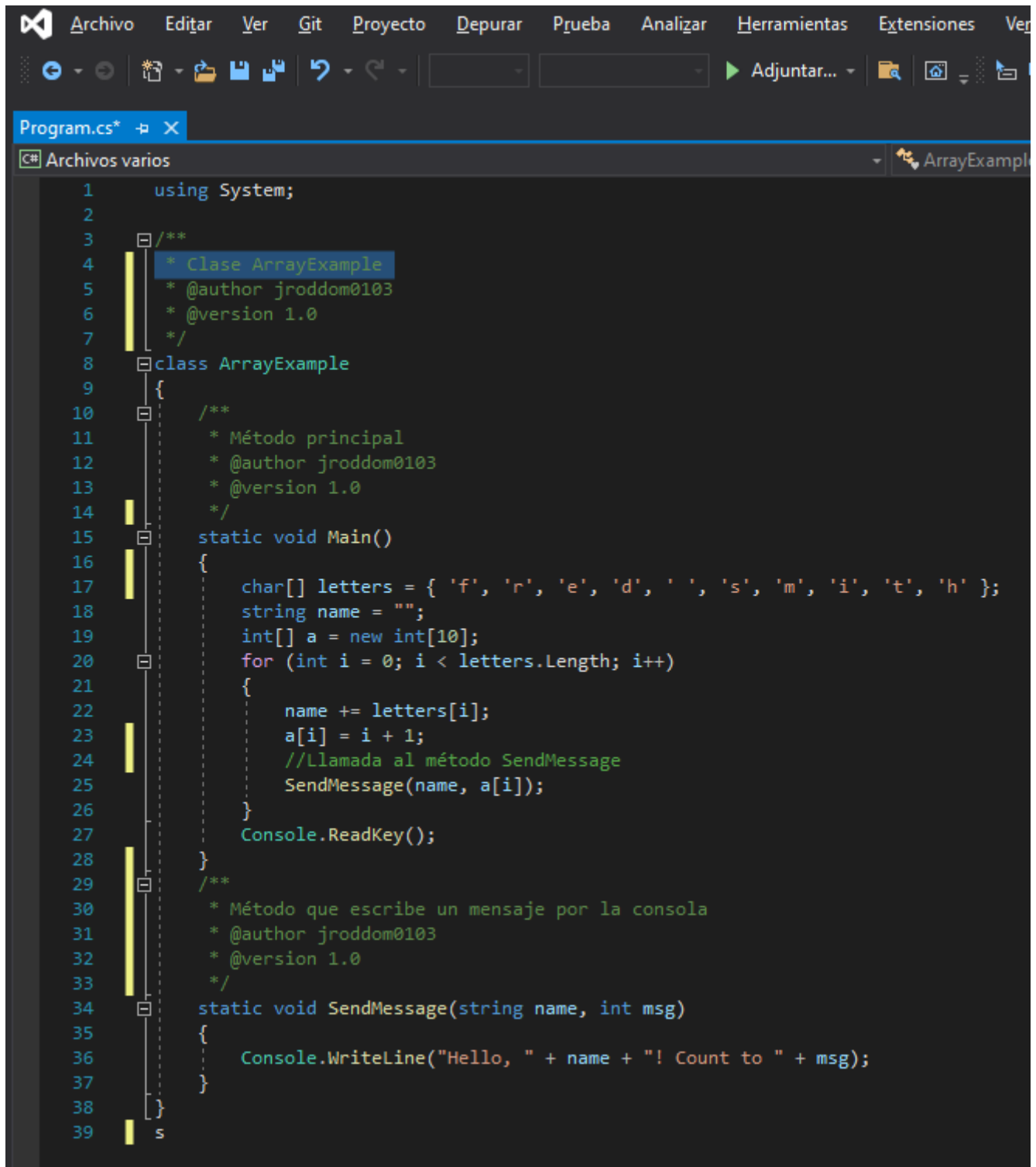
☐ Colocar la solución y el proyecto en el mismo directorio

Framework

.NET Framework 4.7.2

Atrás Crear

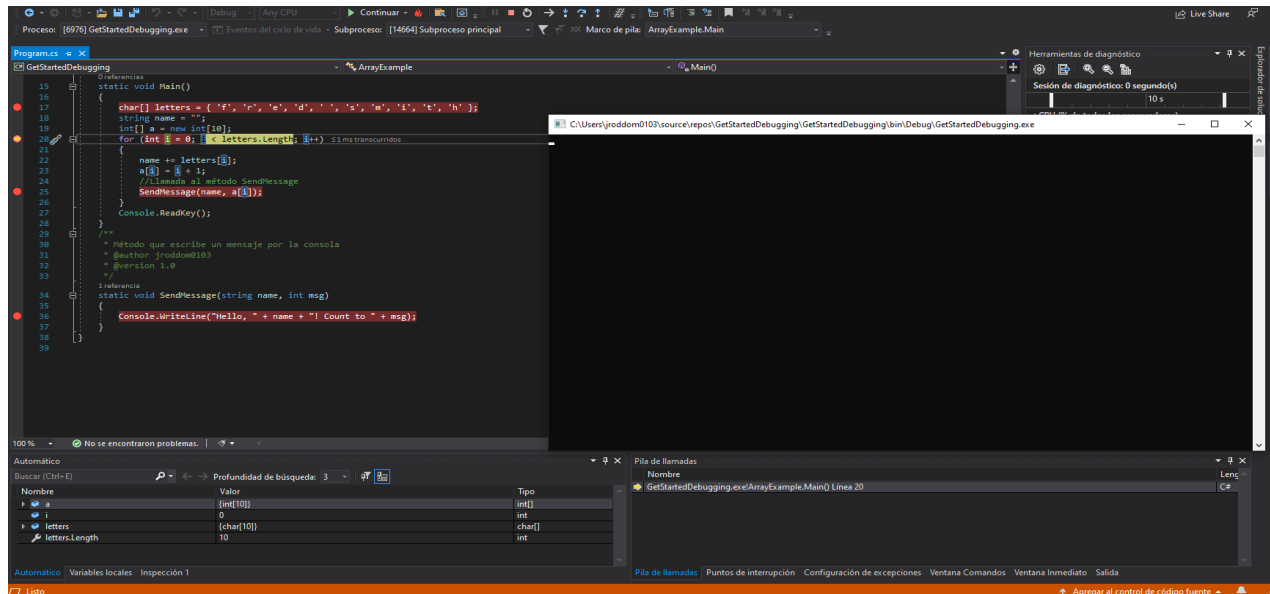
Elegimos nombre de proyecto, además de revisar la ruta y la versión del framework utilizada.



```
1  using System;
2
3  /**
4   * Clase ArrayExample
5   * @author jroddom0103
6   * @version 1.0
7   */
8  class ArrayExample
9  {
10     /**
11      * Método principal
12      * @author jroddom0103
13      * @version 1.0
14      */
15     static void Main()
16     {
17         char[] letters = { 'f', 'r', 'e', 'd', ' ', 's', 'm', 'i', 't', 'h' };
18         string name = "";
19         int[] a = new int[10];
20         for (int i = 0; i < letters.Length; i++)
21         {
22             name += letters[i];
23             a[i] = i + 1;
24             //Llamada al método SendMessage
25             SendMessage(name, a[i]);
26         }
27         Console.ReadKey();
28     }
29     /**
30      * Método que escribe un mensaje por la consola
31      * @author jroddom0103
32      * @version 1.0
33      */
34     static void SendMessage(string name, int msg)
35     {
36         Console.WriteLine("Hello, " + name + "! Count to " + msg);
37     }
38 }
39 s
```

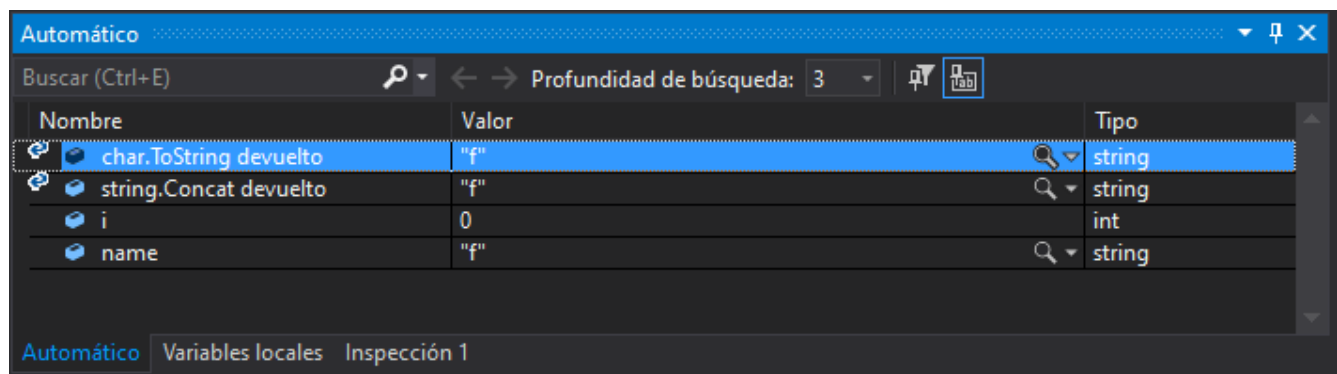
Se sustituye el código generado y se comenta.

- Iniciación del depurador



Se colocan los puntos de depuración a la izquierda del código, que harán que el programa pase a esos puntos y te da información de las variables, permitiendo localizar errores.

- Ventana automático



La ventana automático permite visualizar las variables que se están usando en el punto donde se ha parado el depurador.

- Ventana variables locales

Nombre	Valor	Tipo
char.ToString devuelto	"f"	string
string.Concat devuelto	"f"	string
letters	{char[10]}	char[]
name	"f"	string
a	{int[10]}	int[]
i	0	int

Automático Variables locales Inspección 1

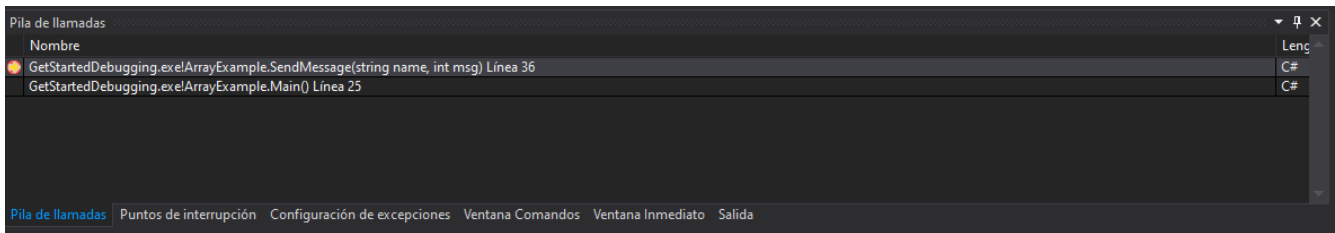
La ventana de variables locales muestra todas las variables disponibles, usadas y no usadas.

- Ventana Inspección (variable name)

Inspección 1		
Buscar (Ctrl+E) Profundidad de búsqueda: 3		
Nombre	Valor	Tipo
name	"f"	string
msg	1	int
a	{int[10]}	int[]
i	0	int
letters	{char[10]}	char[]
Agregar elemento a inspección		
Automático Inspección 1 Variables locales		

La ventana inspección permite la selección de una o más variables específicas, a las que podrás ver su cambio de valor durante la ejecución de código.

- Ventana Pila llamada (método SendMessage)



```
/**
 * Método que escribe una letra y un contador por la consola
 * @author jroddom0103
 * @version 1.0
 */
1 referencia
static void SendMessage(string name, int msg)
{
    Console.WriteLine("Hello, " + name + "! Count to " + msg);
}
```

La ventana Pila de llamadas permite ver la ejecución de funciones, para entender el flujo del programa y corregir errores relacionados a métodos.



### Actividades 4.2

Siguiendo las indicaciones del apartado ‘Pruebas unitarias’ del Pdf ‘Visual Studio\_2019\_Depuración y pruebas’, añadir las capturas de pantalla indicadas a la actividad de la plataforma Moodle

Ref. Web

Capturas a añadir:

- Proyecto creado
- Agrega referencia de proyecto.
- Agregue código al método de prueba unitaria.
- Ventana Explorador de pruebas.
- Ventana Test Explorer

### Actividades 4.3

Siguiendo las indicaciones del siguiente enlace añadir las capturas de pantalla indicadas a la actividad de la plataforma Moodle

[Ref. Web](#)

Capturas a añadir:

- Utils.java y Vectors.java añadidas al proyecto.
- Añade clases de pruebas: VectorsJUnit3Test y UtilsJUnit3Test
- Ejecuta pruebas modificando los test para obtener diferentes resultados
- Repite el proceso para JUnit4

#### **Actividades 4.4**

Proyecto Unity con Vuforia para implementar proyecto de Realidad Aumentada para sistema operativo Android

Para iniciar la actividad, acceder al siguiente enlace para la realización de un formulario sobre diferentes proveedores de AR: <https://forms.gle/HkBgnkMconXa4irVA>