

**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

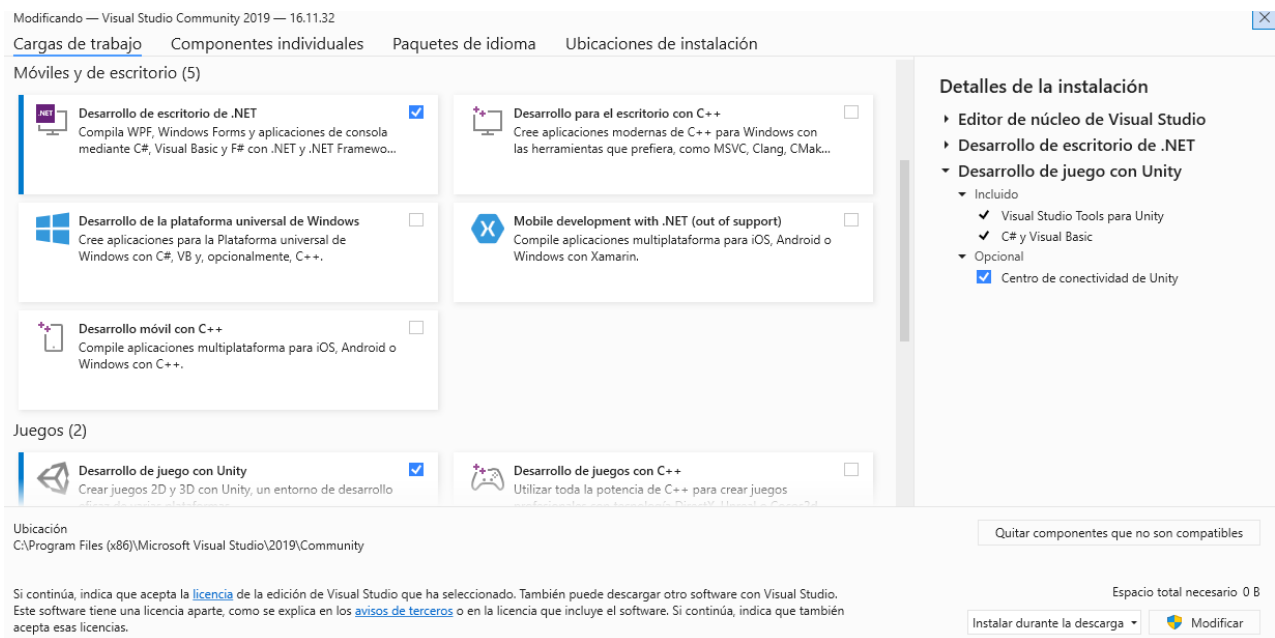
Siguiendo las indicaciones del apartado ‘Depurar el código’ del Pdf ‘Visual Studio\_2019\_Depuración y pruebas’, añadir las capturas de pantalla indicadas a la actividad de la plataforma Moodle

### Ref. Web

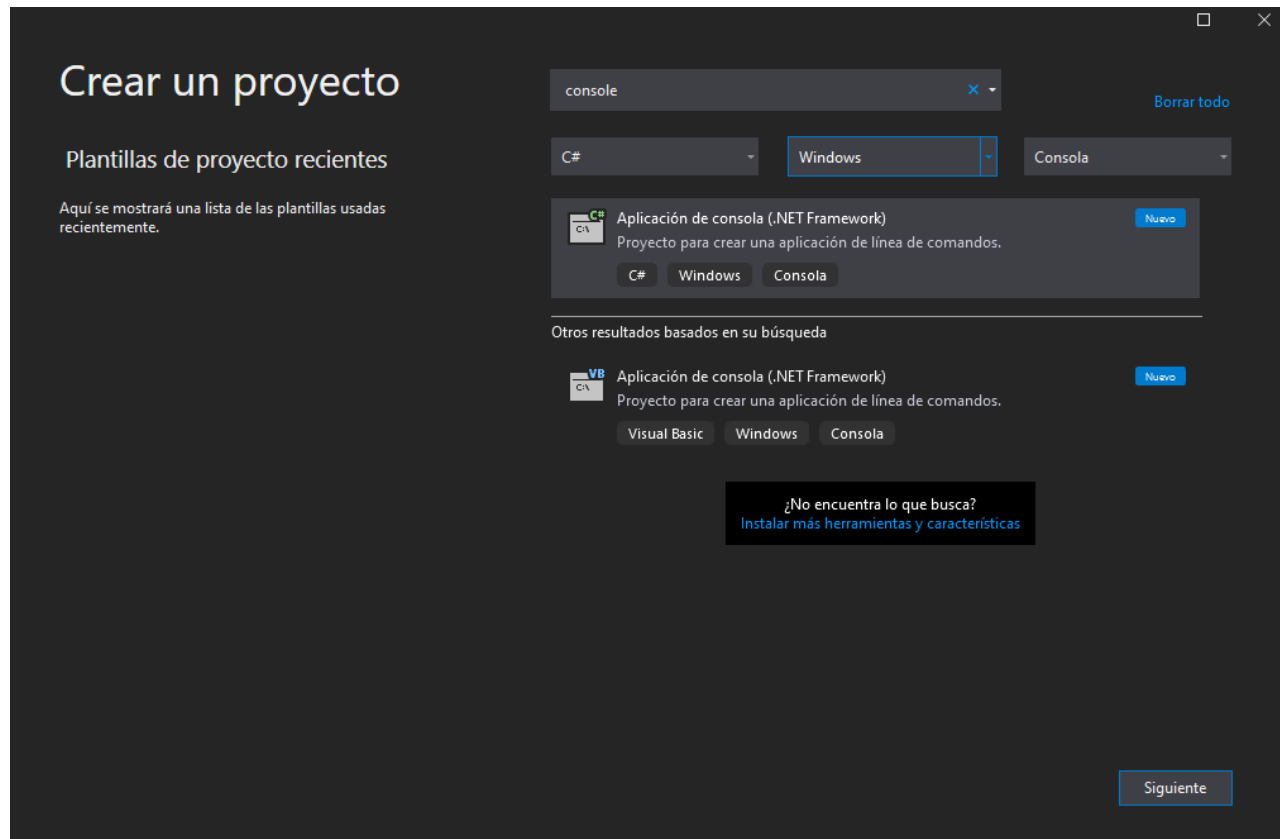
Capturas a añadir:

- Proyecto creado

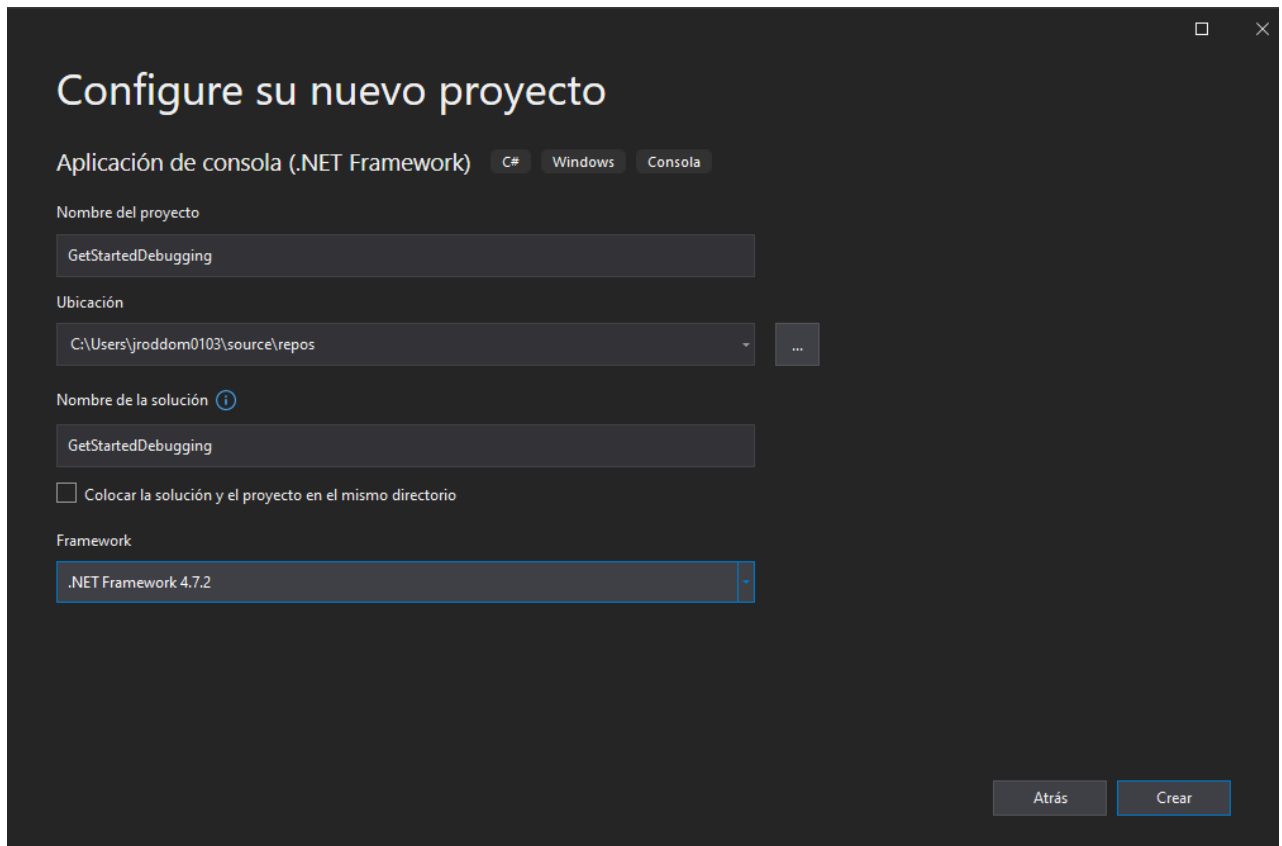
Utilizamos Visual Studio 2019, es decir, una versión anterior. Esto lo hacemos para tener una versión más estable con más errores corregidos (downgrade). Mientras tanto, se trabaja en una versión más reciente, con el fin de tener el software actualizado.



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

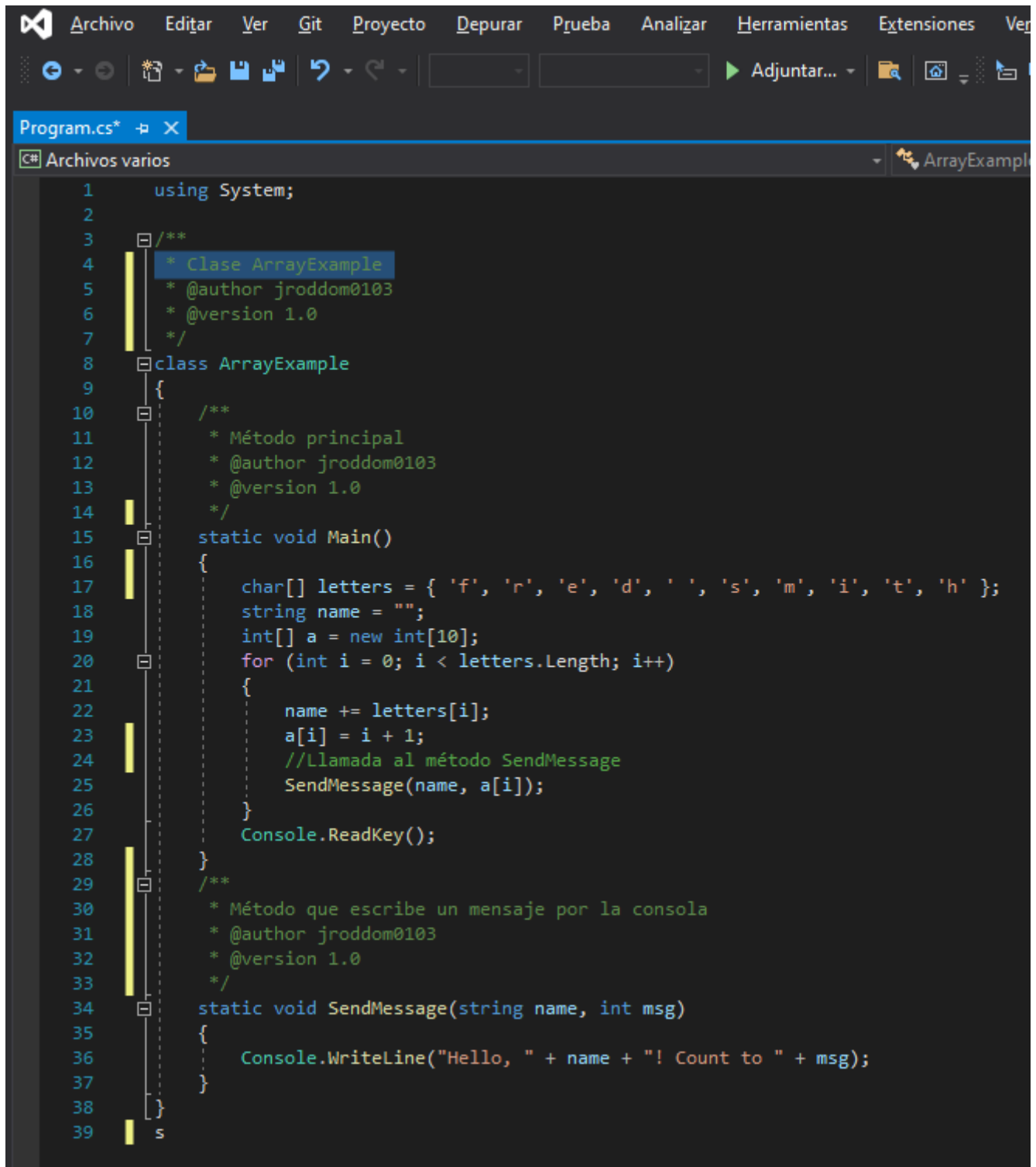
☐ Colocar la solución y el proyecto en el mismo directorio

Framework

.NET Framework 4.7.2

Atrás Crear

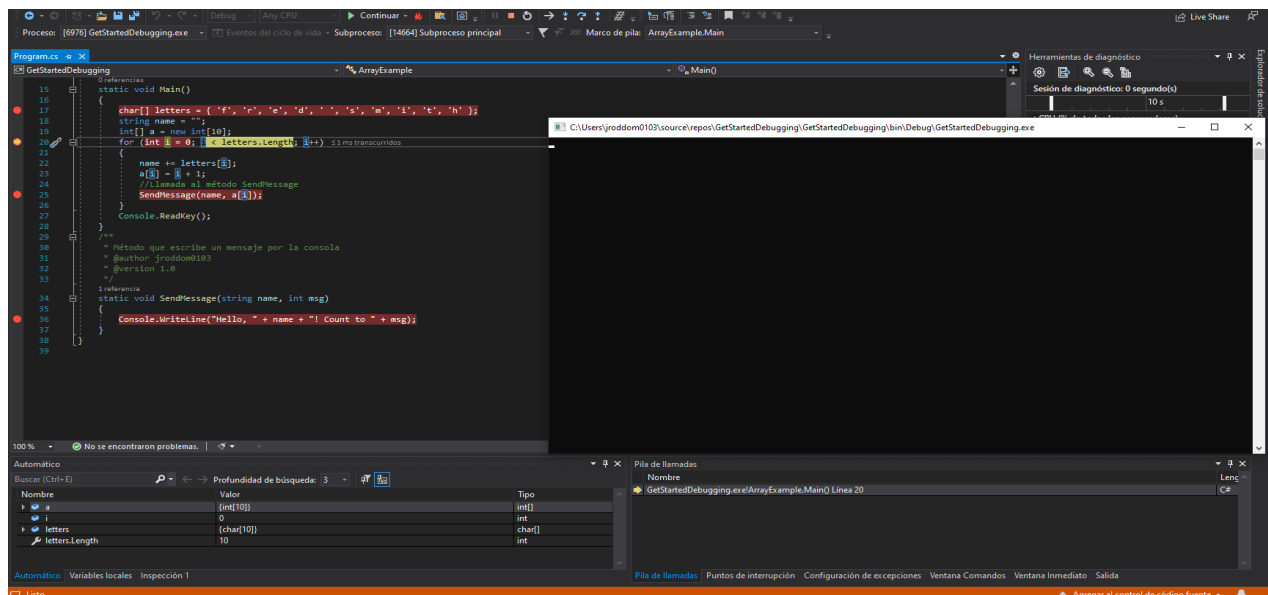
Elegimos nombre de proyecto, además de revisar la ruta y la versión del framework utilizada.



```
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
```

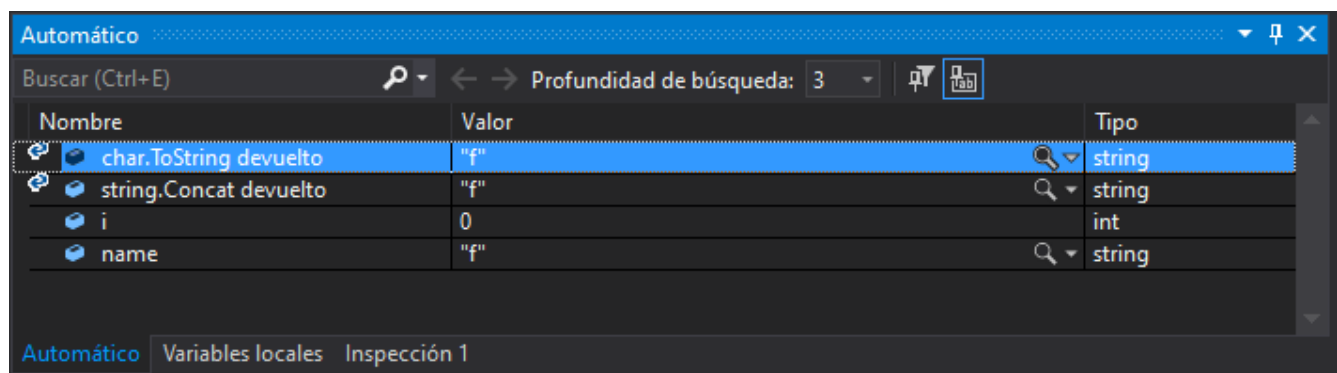
Se sustituye el código generado y se comenta, indicando una pequeña descripción del método o clase.

- Iniciación del depurador



Se colocan los puntos de depuración a la izquierda del código (puntos rojos), que permitirán controlar el orden de ejecución manualmente, de punto en punto, o de línea en línea. La flecha amarilla indica el código que se encuentra en ejecución. En la estructura de bucle for, se indica el valor de la variable *i*, que irá cambiando conforme se vaya ejecutando, hasta que termine el bucle. El uso de un punto de ruptura en la llamada a un método y un método en sí, es reiterativo, por lo que con sólo un punto es suficiente.

- Ventana automático



La ventana automático permite visualizar las variables que se están usando y se han usado en el momento en el que se ha parado el depurador. Esta ventana puede ser de utilidad cuando se quiere corregir un error relacionado con una variable que se usó en cierto punto.

- 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.

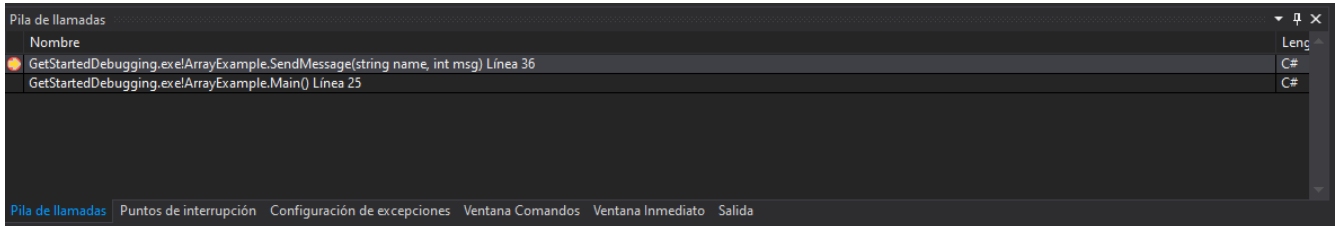
- Ventana Inspección (variable name)

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.



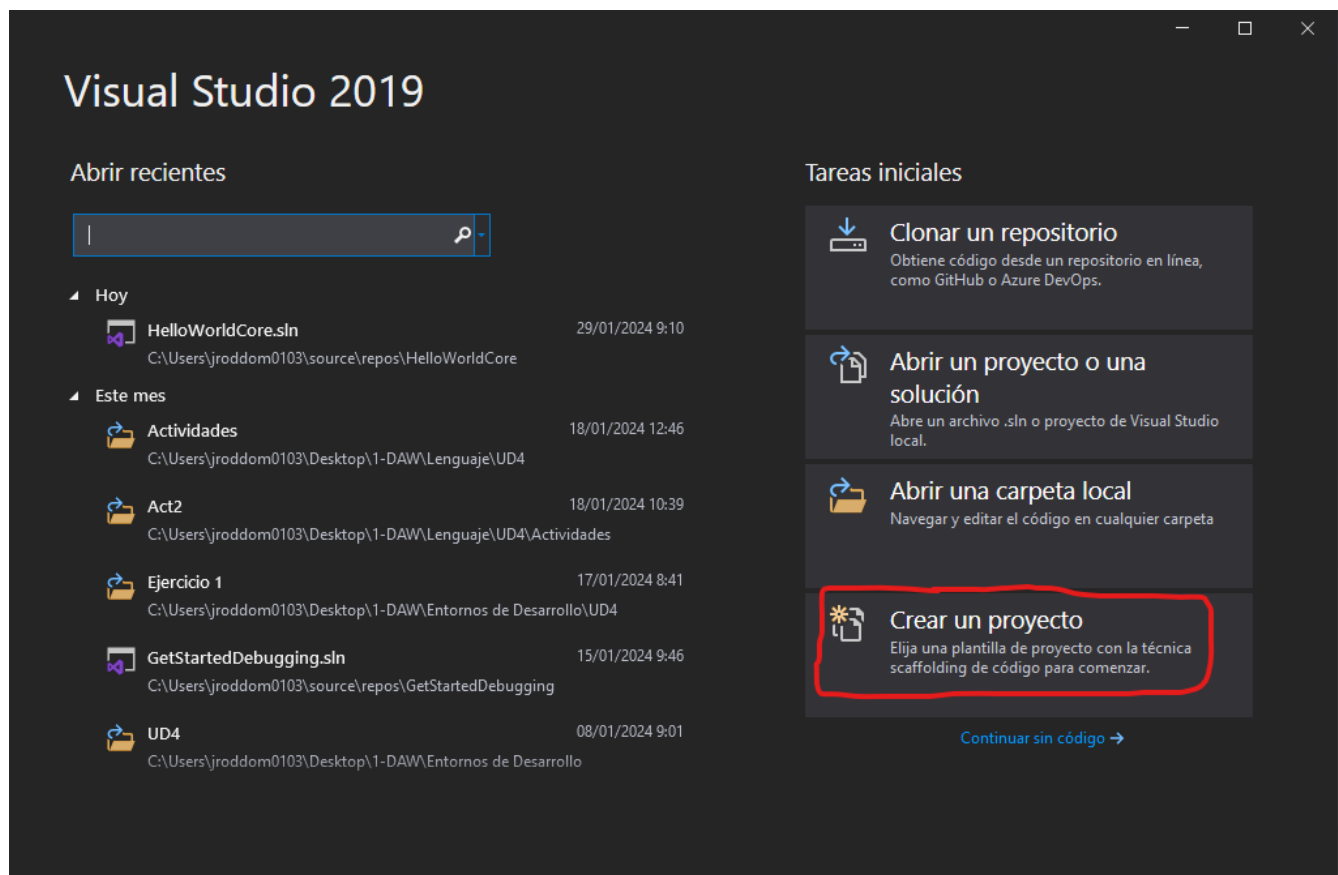
## Actividades 4.2

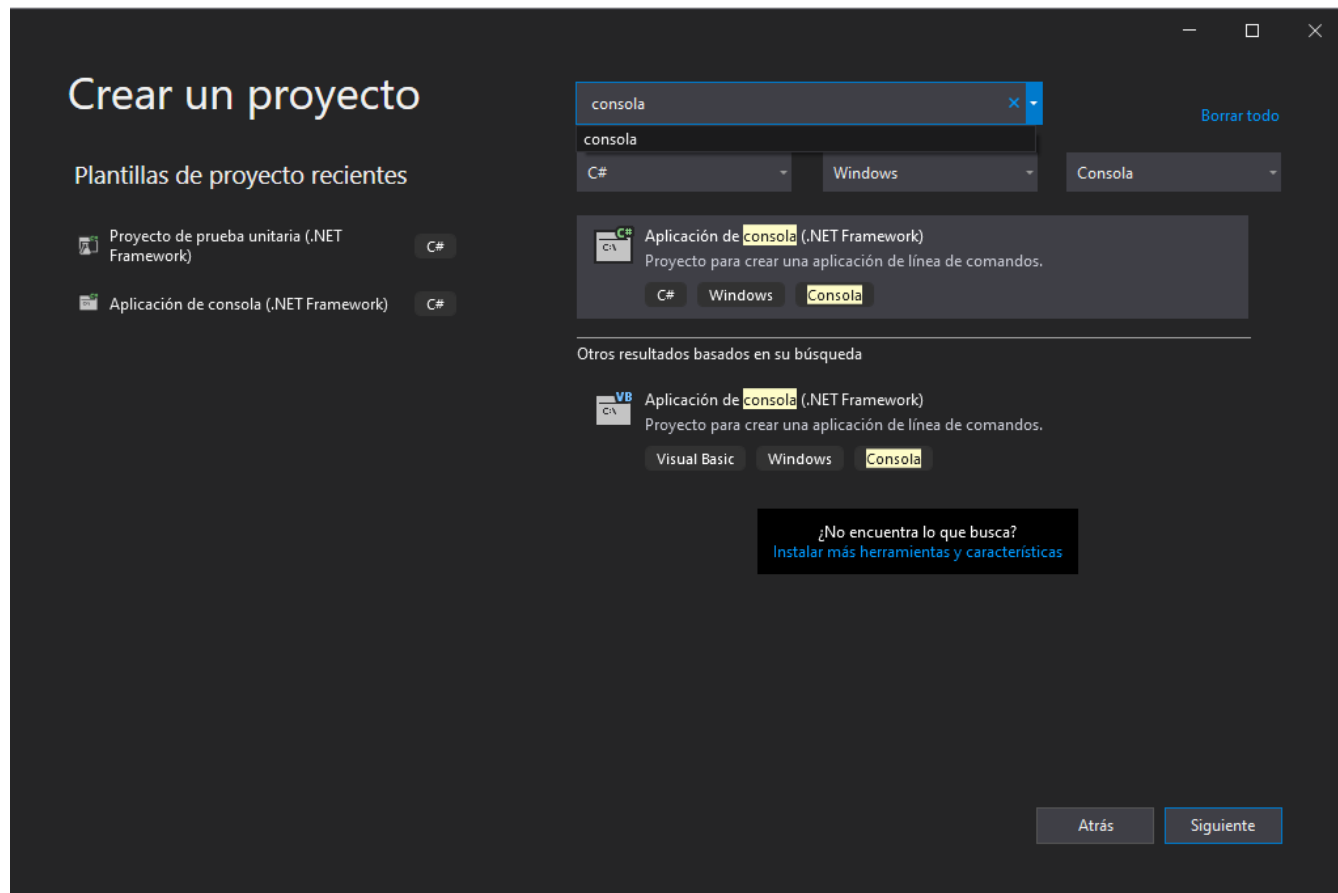
Siguiendo las indicaciones del apartado ‘Pruebas unitarias’ del Pdf ‘Visual Studio\_2019\_Depuración y pruebas’, añadir las capturas de pantalla indicadas a la actividad de la plataforma Moodle

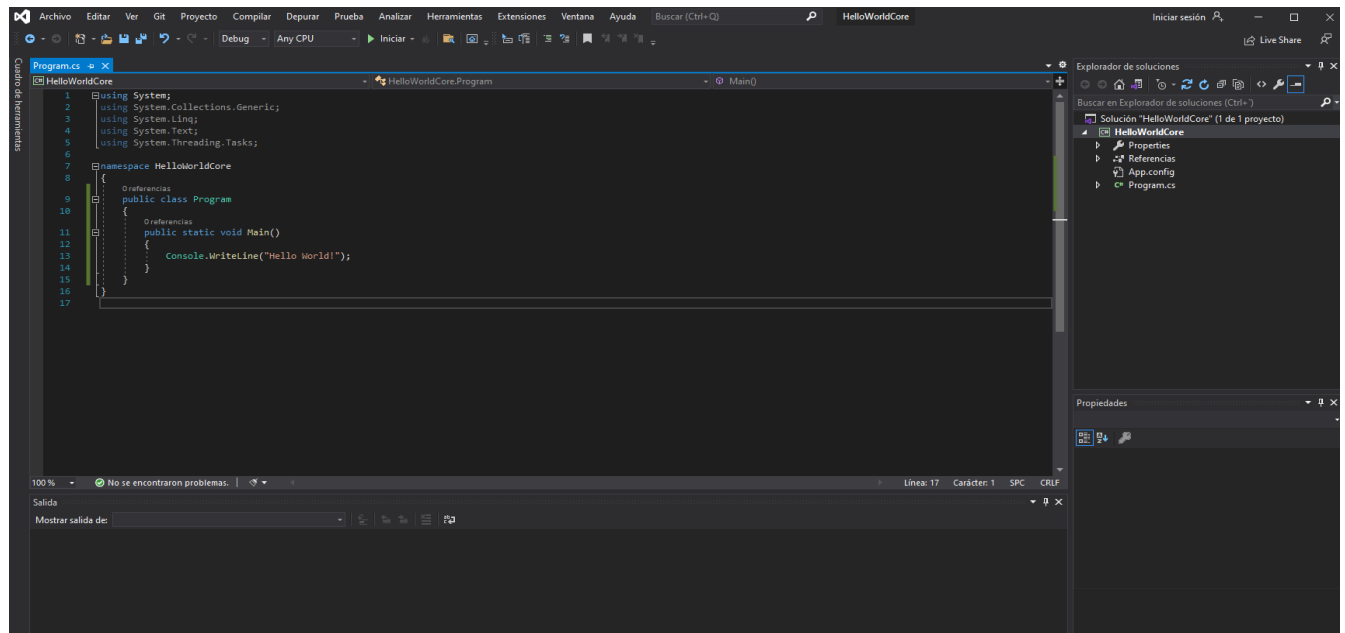
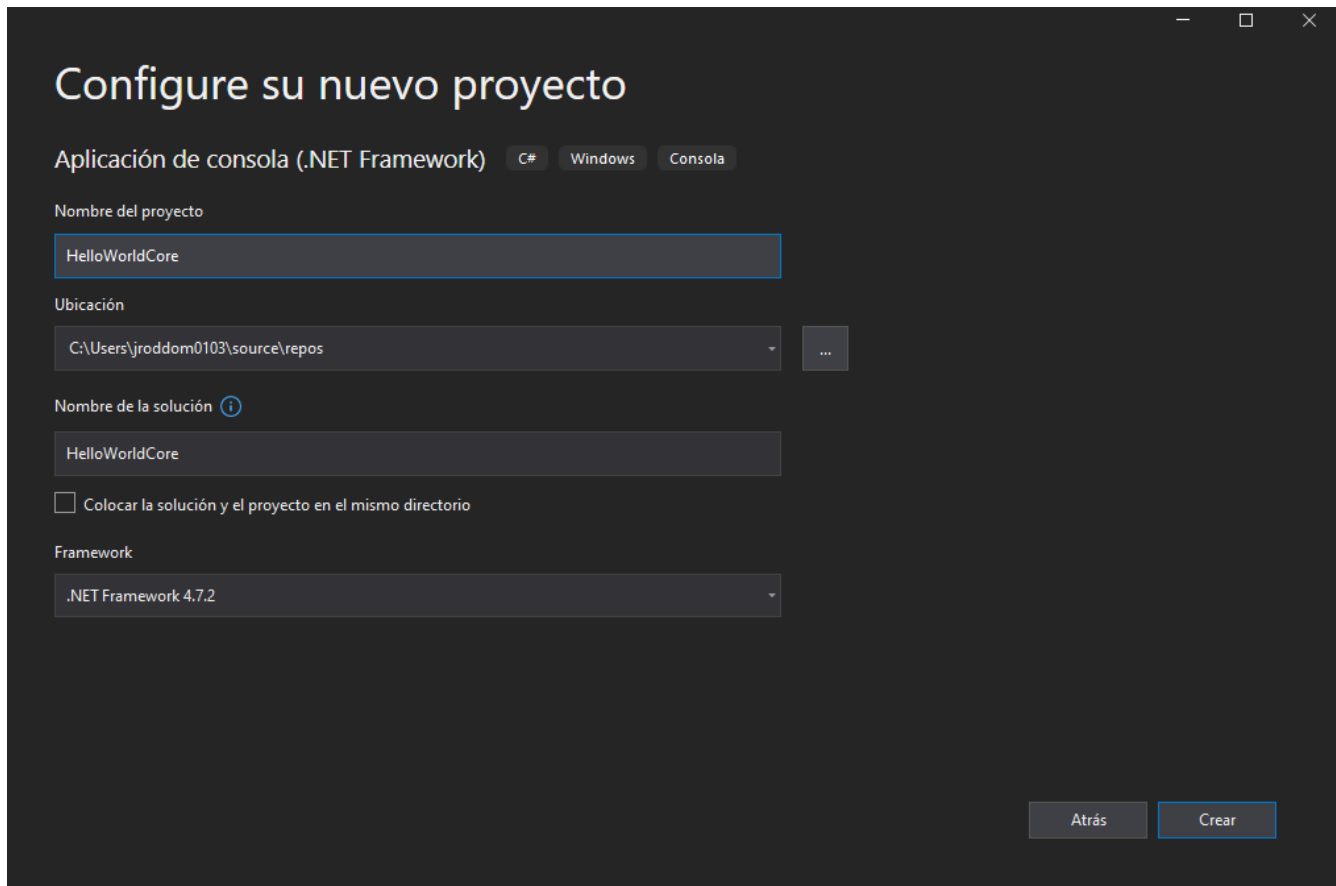
Ref. Web

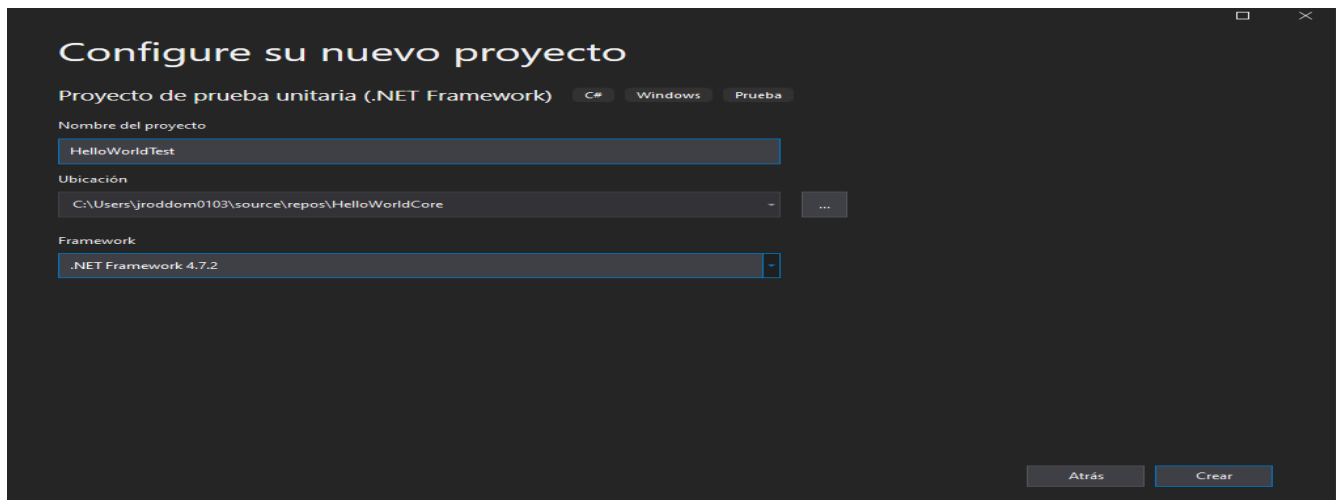
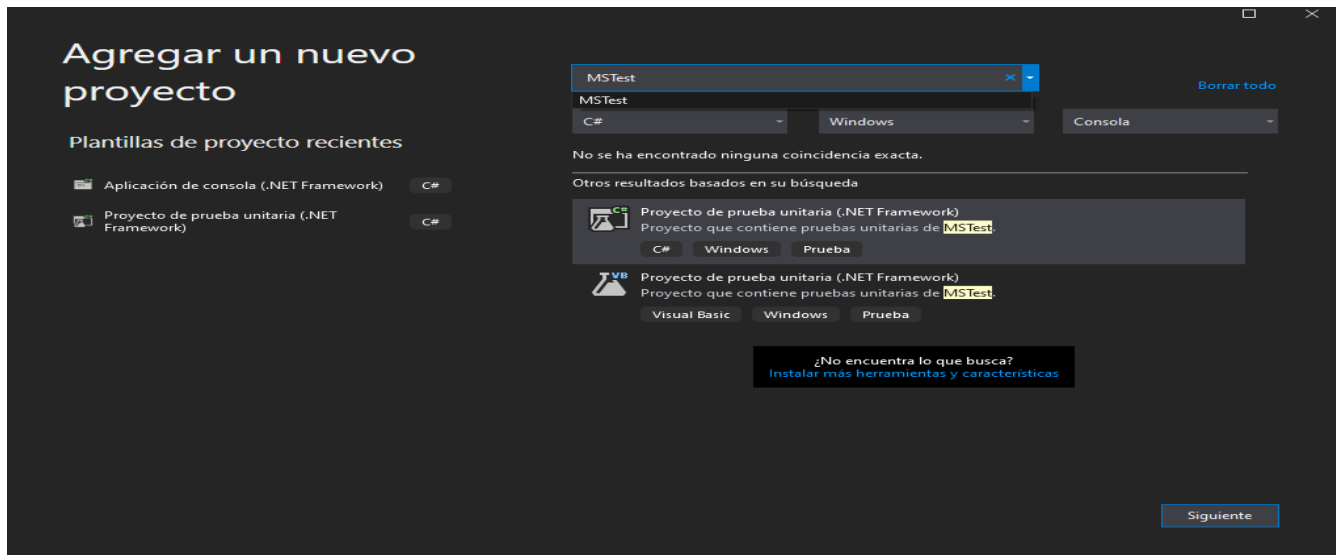
Capturas a añadir:

- Proyecto creado

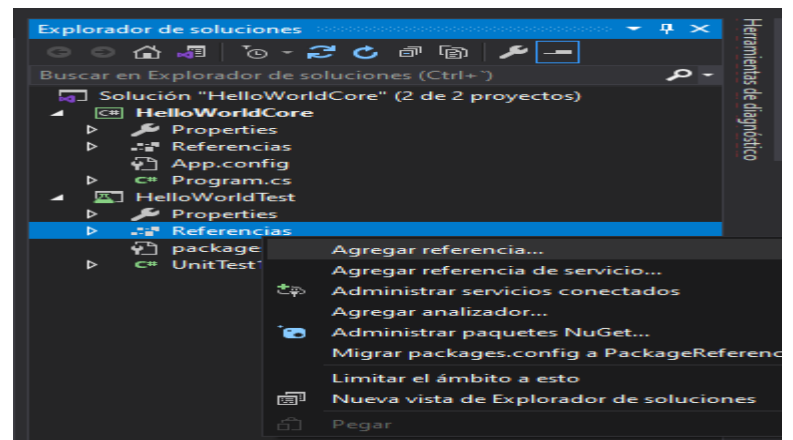
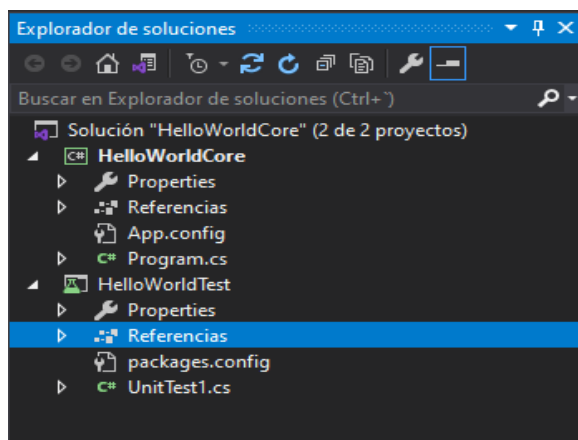


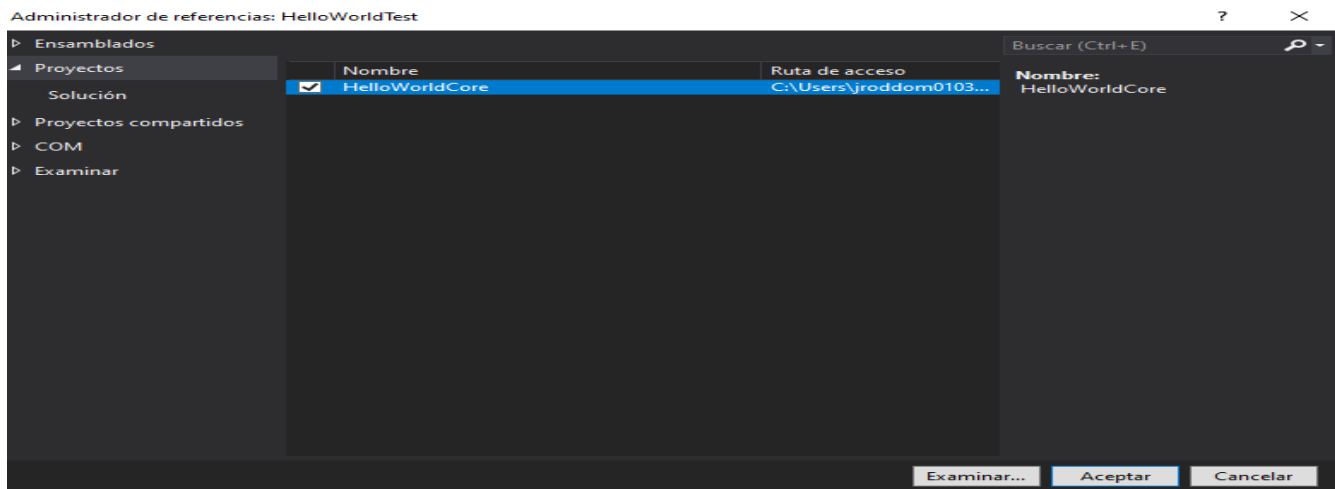






- Agrega referencia de proyecto.

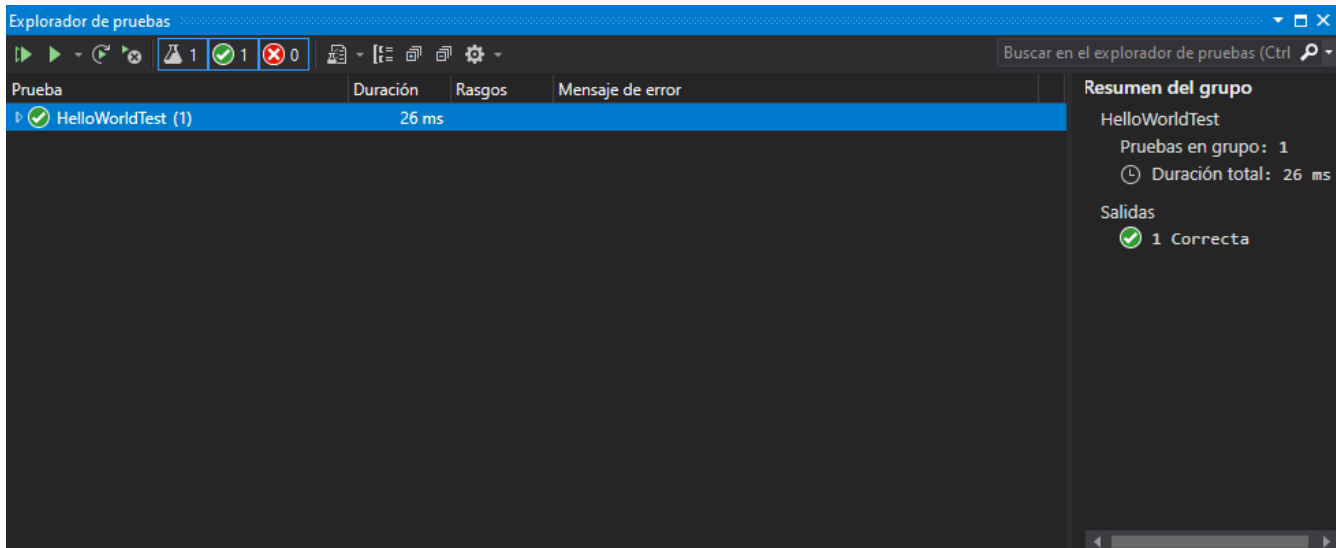
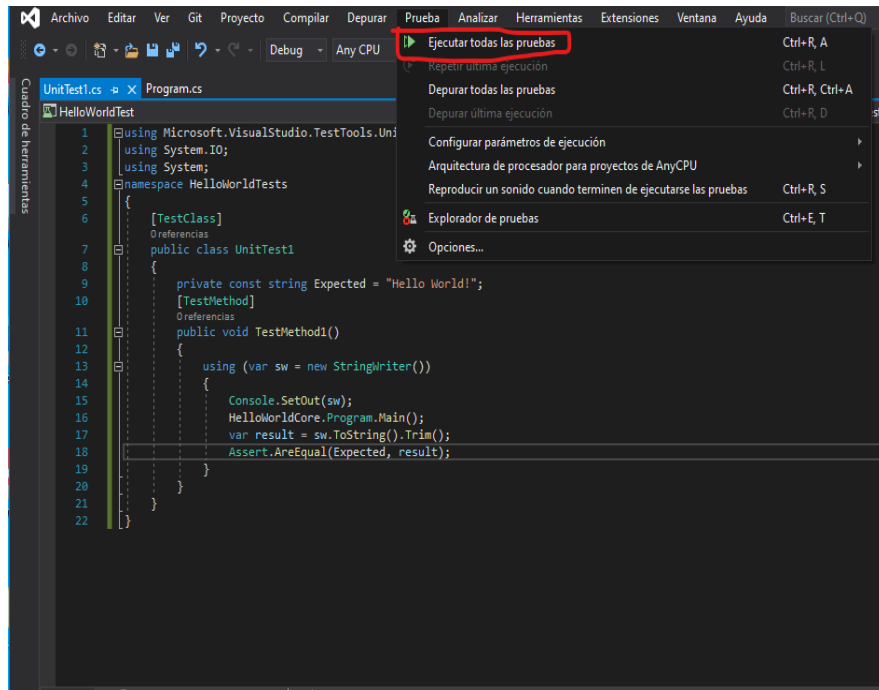


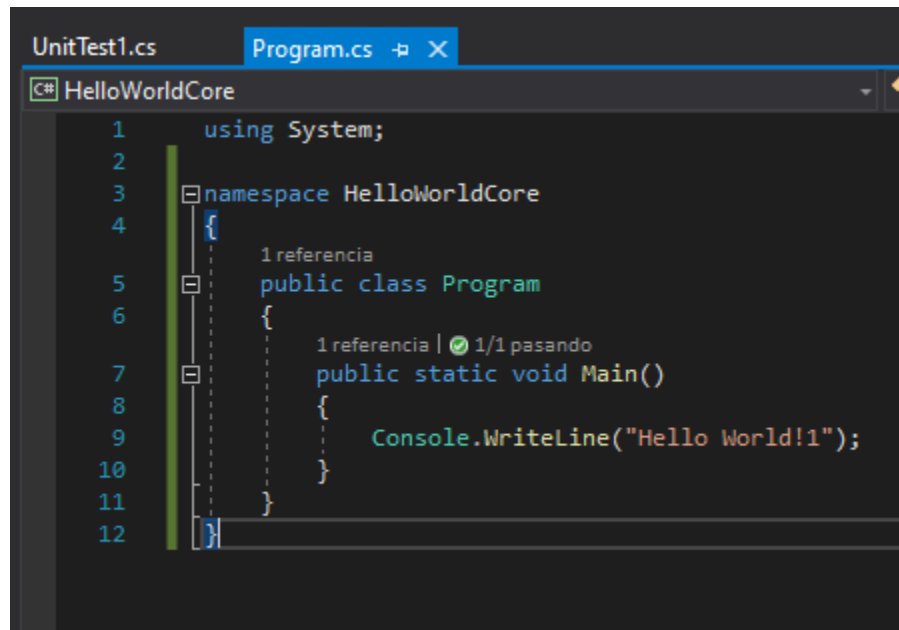


- Agregue código al método de prueba unitaria.

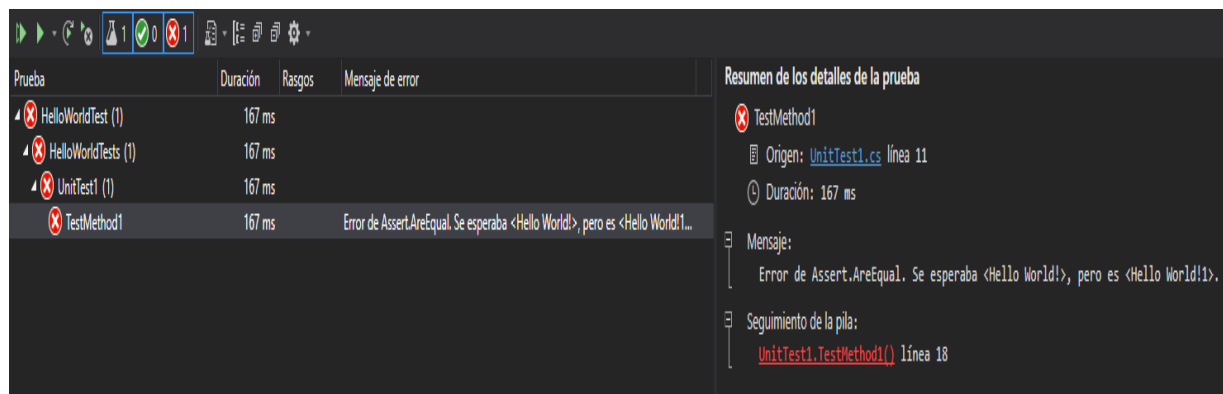
```
1  using Microsoft.VisualStudio.TestTools.UnitTesting;
2  using System.IO;
3  using System;
4  namespace HelloWorldTests
5  {
6      //Atributo de prueba unitaria de la clase
7      [TestClass]
8      public class UnitTest1
9      {
10         //Variable String con el texto esperado
11         private const string Expected = "Hello World!";
12         //Atributo de prueba unitaria del método
13         [TestMethod]
14         //Método que comprueba si el texto que sale como resultado coincide con el String esperado
15         public void TestMethod1()
16         {
17             using (var sw = new StringWriter())
18             {
19                 Console.SetOut(sw);
20                 //Llamada al método Main de la clase Program cuyo namespace es HelloWorldCore
21                 HelloWorldCore.Program.Main();
22                 var result = sw.ToString().Trim();
23                 //Código usado en .NET que compara dos cadenas de caracteres
24                 Assert.AreEqual(Expected, result);
25             }
26         }
27     }
28 }
```

- Ventana Explorador de pruebas.





```
1 using System;
2
3 namespace HelloWorldCore
4 {
5     1 referencia
6     public class Program
7     {
8         1 referencia | 1/1 pasando
9         public static void Main()
10        {
11            Console.WriteLine("Hello World!1");
12        }
13    }
14 }
```



Prueba	Duración	Rasgos	Mensaje de error
HelloWorldTest (1)	167 ms		
HelloWorldTests (1)	167 ms		
UnitTest1 (1)	167 ms		
TestMethod1	167 ms		Error de Assert.AreEqual. Se esperaba <Hello World!>, pero es <Hello World!1...

Resumen de los detalles de la prueba

❌ TestMethod1

Origen: [UnitTest1.cs](#) línea 11

⌚ Duración: 167 ms

Mensaje:

Error de Assert.AreEqual. Se esperaba <Hello World!>, pero es <Hello World!1>.

Seguimiento de la pila:

[UnitTest1.TestMethod1\(\)](#) línea 18

- Ventana Test Explorer

- Calculadora MStest

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Calculadora
8  {
9      4 referencias
10     public class Program
11     {
12         0 referencias
13         public static void Main(string[] args)
14         {
15             char opcion;
16             double a, b;
17             do
18             {
19                 Console.WriteLine("\n" + "Seleccione una opción: " + "\n" + "\n");
20                 Console.WriteLine("\n" + "  1º) Suma" + "\n" + "  2º) Resta" + "\n" + "  3º) Multiplicación" + "\n" + "  4º) División" + "\n" + "  0º) Salir del Programa" + "\n" + "\n");
21                 opcion = Console.ReadKey().KeyChar;
22                 if (opcion != '0')
23                 {
24                     Console.WriteLine("\n" + "Introduce el 1er número: ");
25                     a = double.Parse(Console.ReadLine());
26                     Console.WriteLine("\n" + "Introduce el 2do número: ");
27                     b = double.Parse(Console.ReadLine());
28
29                     switch (opcion)
30                     {
31                         case '1':
32                             Console.WriteLine("\n" + "Suma = " + suma(a, b));
33                             break;
34
35                         case '2':
36                             Console.WriteLine("\n" + "Resta = " + resta(a, b));
37                             break;
38
39                         case '3':
40                             Console.WriteLine("\n" + "Multiplicación = " + multiplicacion(a, b));
41                             break;

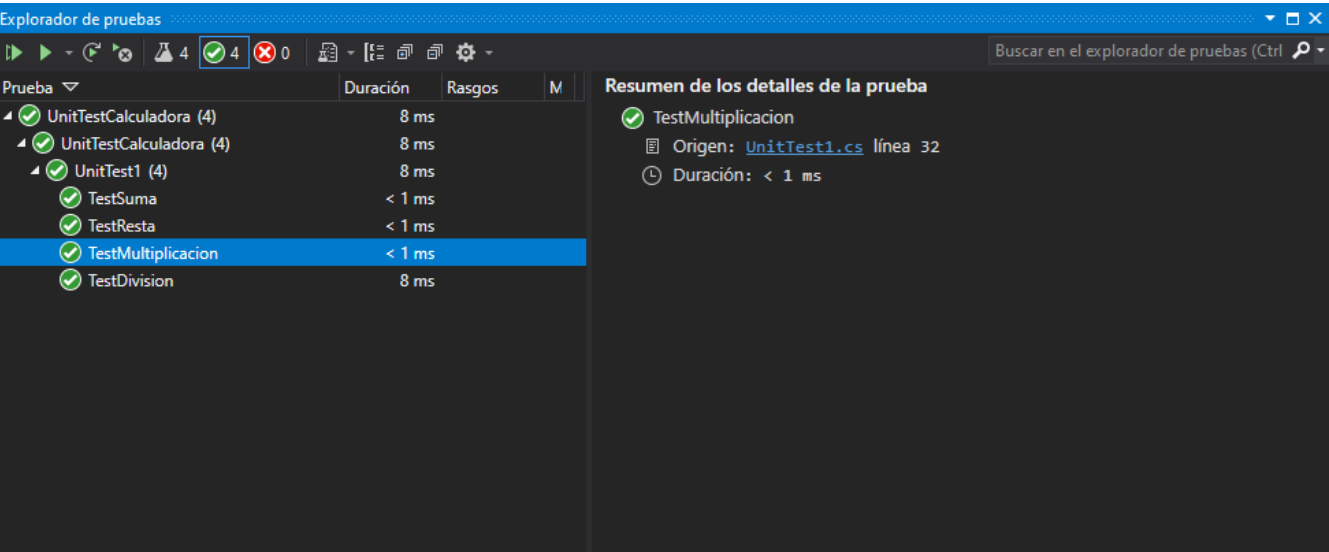
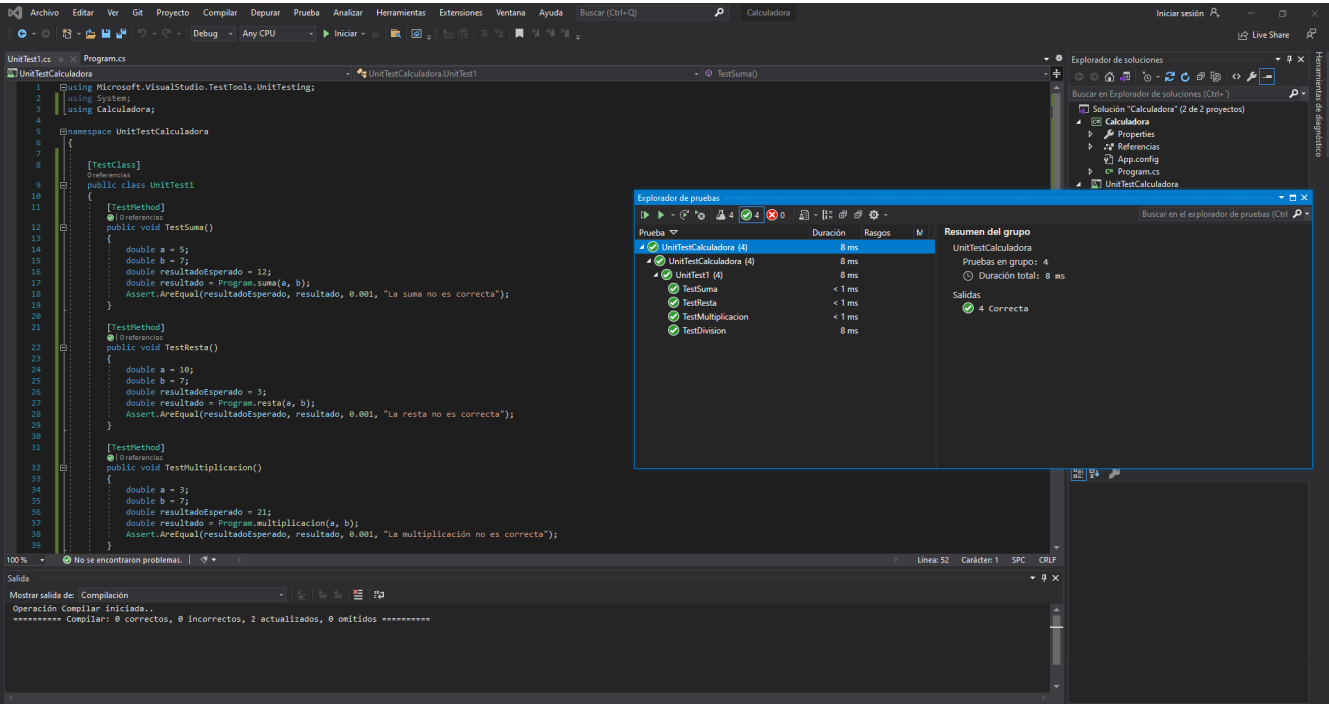
```

```

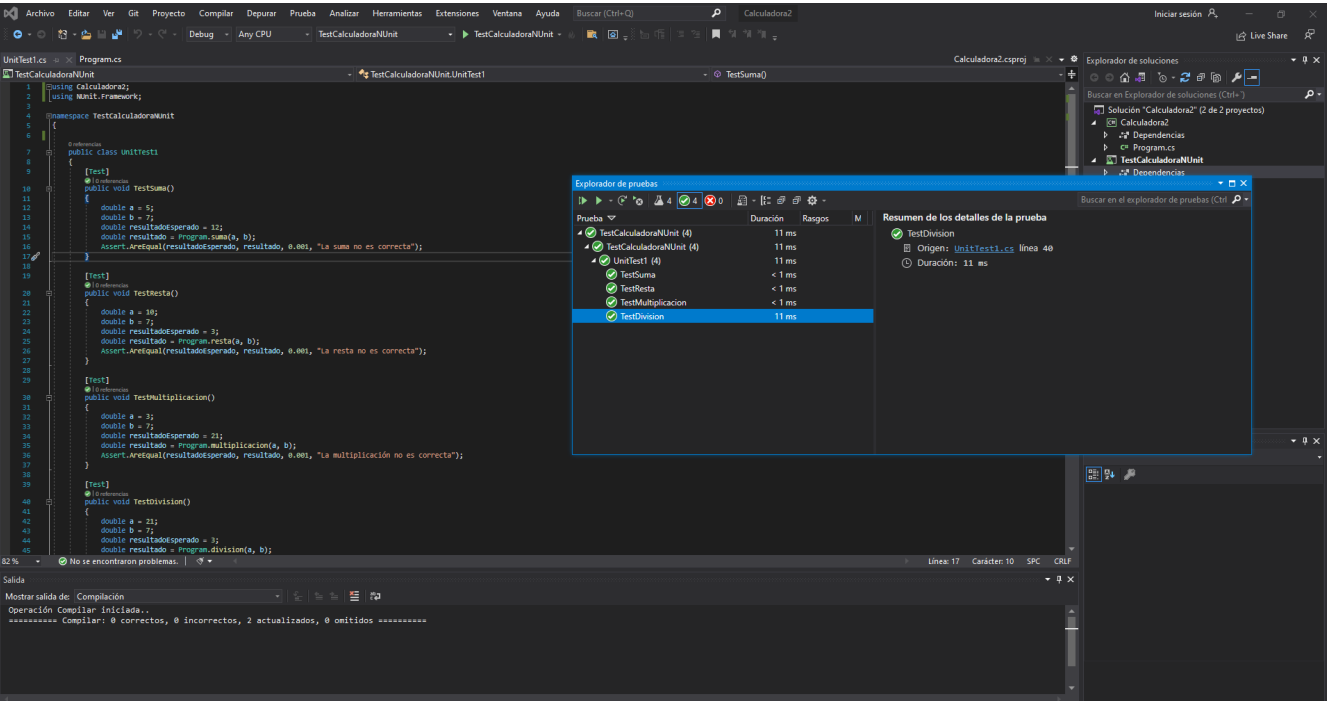
UnitTest1.cs  Program.cs
Calculadora - Calculadora.Program - Main(string[] args)
39     case '3':
40         Console.WriteLine("\n" + "Multiplicación = " + multiplicacion(a, b));
41         break;
42
43     case '4':
44         Console.WriteLine("\n" + "División = " + division(a, b));
45         break;
46     }
47
48     else
49     {
50         Console.WriteLine("Saliendo...");
51         break;
52     }
53     Console.ReadKey();
54     } while (opcion != '0');
55 }
56
57 2 referencias 1/1 parámetro
58 public static double suma(double a, double b)
59 {
60     double suma = a + b;
61     return suma;
62 }
63
64 2 referencias 1/1 parámetro
65 public static double resta(double a, double b)
66 {
67     double resta = a - b;
68     return resta;
69 }
70
71 2 referencias 1/1 parámetro
72 public static double multiplicacion(double a, double b)
73 {
74     double multi = a * b;
75     return multi;
76 }
77
78 2 referencias 1/1 parámetro
79 public static double division(double a, double b)
80 {
81     double divi = a / b;
82     return divi;
83 }

```

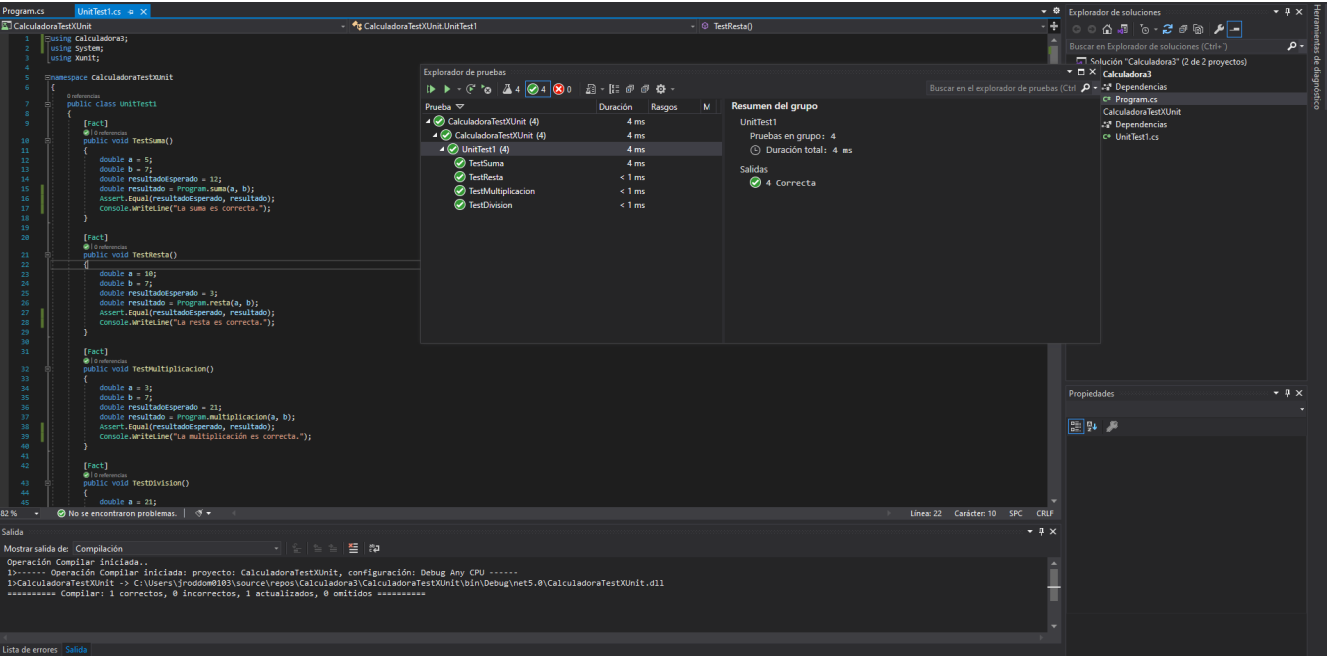




- Calculadora Nunit



- Calculadora Xunit



### 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>