

PRUEBA UNITARIA

Entornos de desarrollo-1ºDAM



Florida universitaria
Gabriel, Javi, Adrián, Martín, Víctor

Contenido

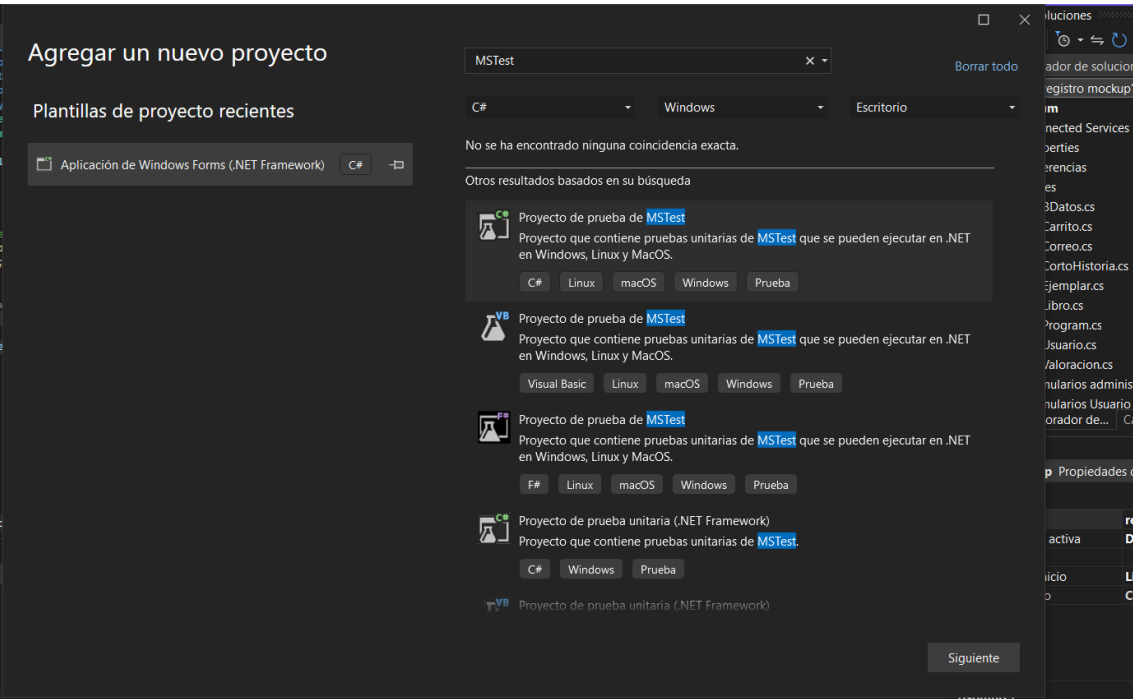
Paso 1:	2
El método del que vamos a hacer la prueba unitaria es el método “importeTotal” de la clase Libro.....	2
Crear un nuevo proyecto (Proyecto de prueba (.NET Framework))	2
.....	2
Agregar una prueba unitaria en el nuevo proyecto	2
PASO 2:	3
Le introducidos un valor a la variable “precio” otro a la variable “cantidad” , introducimos la operación que se realiza en el método en “resultado” y que valor esperamos en “resultadoEsperado” y lo comprobamos con “Assert.AreEqual(resultado,resultadoEsperado)” (obviamente aquí sabemos que dará error)	3
.....	3
Ejecutamos la prueba :.....	3
Paso 3:	4
Ahora cambiamos el valor de “resultadoEsperado” para que sea correcto y hacer la prueba unitaria en condiciones	4
Y ejecutamos la prueba unitaria por segunda vez	4

Paso 1:

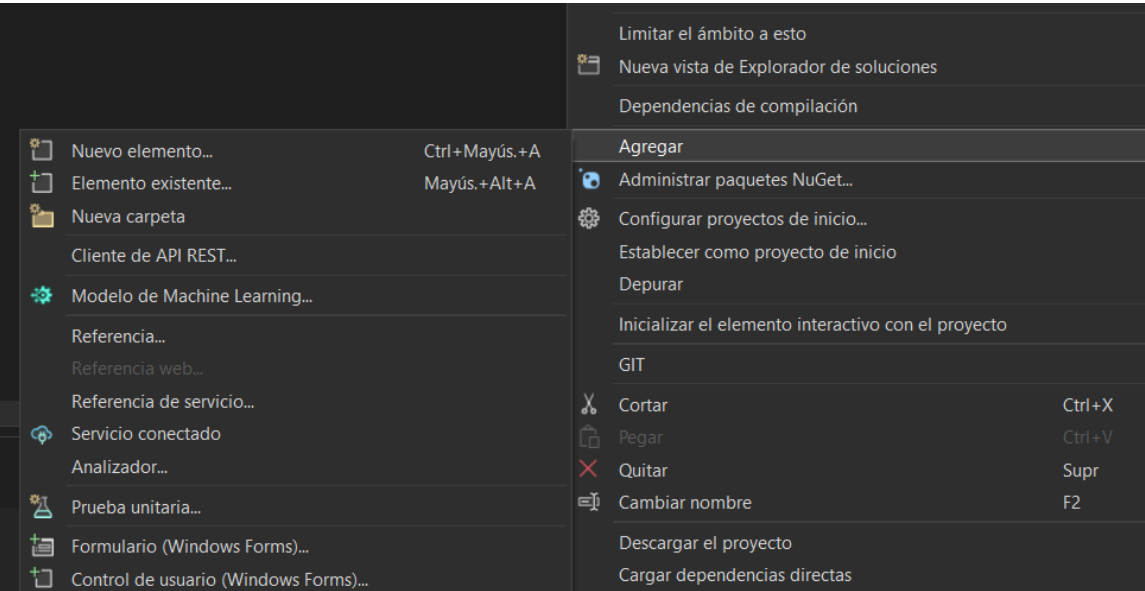
El método del que vamos a hacer la prueba unitaria es el método “importeTotal” de la clase Libro

```
3 referencias | javier armero lopez, Hace 1 día | 1 autor, 1 cambio
public double importeTotal(double precio, int cantidad)
{
    return precio * cantidad;
}
```

Crear un nuevo proyecto (Proyecto de prueba (.NET Framework))



Agregar una prueba unitaria en el nuevo proyecto



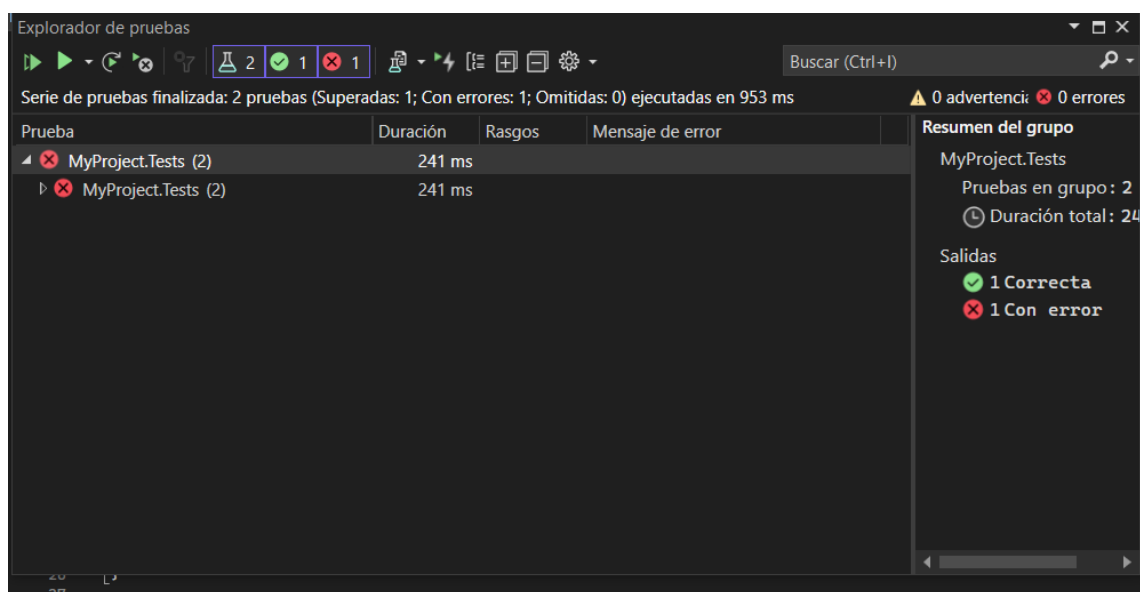
PASO 2:

Le introducimos un valor a la variable “precio” otro a la variable “cantidad” , introducimos la operación que se realiza en el método en “resultado” y que valor esperamos en “resultadoEsperado” y lo comprobamos con “Assert.AreEqual(resultado,resultadoEsperado)” (obviamente aquí sabemos que dará error)

```
[TestClass]
0 referencias | 0 cambios | 0 autores, 0 cambios
public class UnitTest1
{
    [TestMethod]
    0 referencias | 0 cambios | 0 autores, 0 cambios
    public void TestImporteTotal()
    {
        var precio = 10.0;
        var cantidad = 5;
        var resultado = precio*cantidad;
        var resultadoEsperado = 40.0;

        Assert.AreEqual(resultado, resultadoEsperado);
    }
}
```

Ejecutamos la prueba :



Explorador de pruebas

Serie de pruebas finalizada: 2 pruebas (Superadas: 1; Con errores: 1; Omitidas: 0) ejecutadas en 953 ms

Prueba	Duración	Rasgos	Mensaje de error
MyProject.Tests (2)	241 ms		
MyProject.Tests (2)	241 ms		

Resumen del grupo

MyProject.Tests

Pruebas en grupo: 2

Duración total: 241 ms

Salidas

- 1 Correcta
- 1 Con error

Paso 3:

Ahora cambiamos el valor de “resultadoEsperado” para que sea correcto y hacer la prueba unitaria en condiciones

```
namespace MyProject.Tests
{
    [TestClass]
    0 referencias | 0 cambios | 0 autores, 0 cambios
    public class UnitTest1
    {
        [TestMethod]
        ✓ | 0 referencias | 0 cambios | 0 autores, 0 cambios
        public void TestImporteTotal()
        {
            var precio = 10.0;
            var cantidad = 5;
            var resultado = precio*cantidad;
            var resultadoEsperado = 50.0;

            Assert.AreEqual(resultado, resultadoEsperado);
        }
    }
}
```

Y ejecutamos la prueba unitaria por segunda vez

Explorador de pruebas

2 2 0

Buscar (Ctrl+I)

Serie de pruebas finalizada: 2 pruebas (Superadas: 2; Con errores: 0; Omitidas: 0) ejecutadas en 723 ms

0 advertenci: 0 errores

Prueba	Duración	Rasgos	Mensaje de error
✓ MyProject.Tests (2)	13 ms		
▶ MyProject.Tests (2)	13 ms		

Resumen del grupo

MyProject.Tests

Pruebas en grupo: 2

⌚ Duración total: 13

Salidas

✓ 2 Correcta