

UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA DE SISTEMAS



Guía Práctica de Laboratorio

“Pruebas Unitaria con Mstest”

**Que se presenta para el curso:
“CALIDAD Y PRUEBAS DE
SOFTWARE”**

Docente:

Mag. Ing. Patrick Cuadros Quiroga

Estudiante:

Rodrigo Lira Alvarez (2019063331)

**TACNA – PERÚ
2025**

Laboratorio N°1 “Pruebas Unitarias con MSTEST”

I. OBJETIVOS

- Comprender el funcionamiento de las pruebas unitarias dentro de una aplicación utilizando el Framework de pruebas de Microsoft (MSTest).

II. REQUERIMIENTOS

Conocimientos:

- Conocimientos básicos de gestión de base de datos de series de tiempo
- Conocimientos básicos de shell (línea de comandos).

Hardware:

- Virtualization activada en el BIOS..
- CPU SLAT-capable feature.
- Al menos 4GB de RAM.

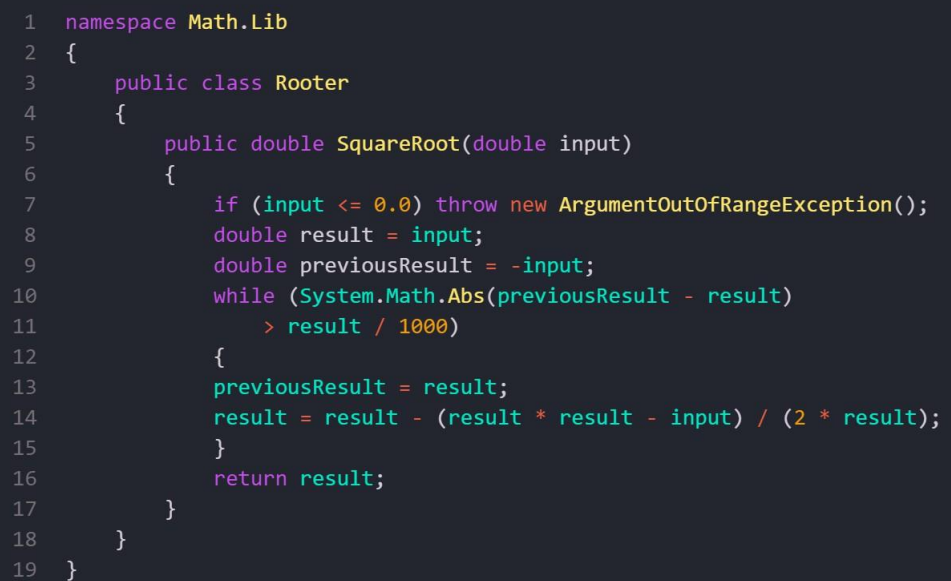
Software:

- Windows 10 64bit: Pro, Enterprise o Education (1607 Anniversary Update, Build 14393 o Superior)
- Docker Desktop
- Powershell versión 7.x
- Net 6 o superior
- Visual Studio Code

III. CONSIDERACIONES INICIALES

- Clonar el repositorio mediante git para tener los recursos necesarios

IV. DESARROLLO



```
1 namespace Math.Lib
2 {
3     public class Rooter
4     {
5         public double SquareRoot(double input)
6         {
7             if (input <= 0.0) throw new ArgumentOutOfRangeException();
8             double result = input;
9             double previousResult = -input;
10            while (System.Math.Abs(previousResult - result)
11                > result / 1000)
12            {
13                previousResult = result;
14                result = result - (result * result - input) / (2 * result);
15            }
16            return result;
17        }
18    }
19 }
```

```

1  using Microsoft.VisualStudio.TestTools.UnitTesting;
2  using Math.Lib;
3
4  namespace Math.Tests
5  {
6      [TestClass]
7      public class RooterTests
8      {
9          [TestMethod]
10         public void BasicRooterTest()
11         {
12             Rooter rooter = new Rooter();
13             double expectedResult = 2.0;
14             double input = expectedResult * expectedResult;
15             double actualResult = rooter.SquareRoot(input);
16             Assert.AreEqual(expectedResult, actualResult, delta: expectedResult / 100);
17         }
18
19         [TestMethod]
20         public void RooterValueRange()
21         {
22             Rooter rooter = new Rooter();
23             for (double expected = 1e-8; expected < 1e+8; expected *= 3.2)
24             {
25                 RooterOneValue(rooter, expected);
26             }
27         }
28
29         private void RooterOneValue(Rooter rooter, double expectedResult)
30         {
31             double input = expectedResult * expectedResult;
32             double actualResult = rooter.SquareRoot(input);
33             Assert.AreEqual(expectedResult, actualResult, delta: expectedResult / 1000);
34         }
35
36         [TestMethod]
37         public void RooterTestNegativeInputx()
38         {
39             Rooter rooter = new Rooter();
40             try
41             {
42                 rooter.SquareRoot(-10);
43             }
44             catch (System.ArgumentOutOfRangeException)
45             {
46                 return;
47             }
48             Assert.Fail();
49         }
50
51         [TestMethod]
52         public void RooterTestZeroInput()
53         {
54             Rooter rooter = new Rooter();
55             try
56             {
57                 rooter.SquareRoot(0);
58             }
59             catch (System.ArgumentOutOfRangeException)
60             {
61                 return;
62             }
63             Assert.Fail("Expected ArgumentOutOfRangeException for input zero.");
64         }
65     }
66 }
67
68 }

```

```
2024-11-04T12:05:34 Arguments
2024-11-04T12:05:34 -records: /.../coverage/cobertura.xml
2024-11-04T12:05:34 -targetdir: Cobertura
2024-11-04T12:05:34 -recordsdir: HTML
2024-11-04T12:05:35 Writing record file: Cobertura.xml
2024-11-04T12:05:35 Record generation took 0.02 seconds
```

Information

Parser: MultiReport (3x Cobertura)
Assemblies: 1
Classes: 1
Files: 1

Coverage date: 4/11/2024 - 11:59:40 -
4/11/2024 - 12:05:37

Line coverage

Covered lines: 12
Uncovered lines: 0
Coverable lines: 12
Total lines: 19
Line coverage: 100%

100⁰0

Branch coverage

100⁰0

Covered branches: 6
Total branches: 6
Branch coverage: 100%

Method coverage

Feature is only available for sponsors
Upgrade to PRO version

Risk Hotspots

No risk hotspots found.

Coverage

V. Actividades Encargadas

1. Adicionar un nuevo escenario de prueba donde se maneje una excepción con un mensaje que diga "El valor ingresado es invalido, solo se puede ingresar números positivos".

```
1 namespace Math.Lib
2 {
3     public class Rooter
4     {
5         public double SquareRoot(double input)
6         {
7             if (input <= 0.0)
8                 throw new ArgumentOutOfRangeException(nameof(input), "El valor ingresado es invalido, solo se puede ingresar números positivos");
9
10            double result = input;
11            double previousResult = -input;
12            while (System.Math.Abs(previousResult - result) > result / 1000) // Usar System.Math.Abs aqui
13            {
14                previousResult = result;
15                result = result - (result * result - input) / (2 * result);
16            }
17            return result;
18        }
19    }
20 }
21
22
```

```
1 [TestMethod]
2 public void RooterTestZeroInput()
3 {
4     Rooter rooter = new Rooter();
5     try
6     {
7         rooter.SquareRoot(0);
8     }
9     catch (ArgumentOutOfRangeException ex)
10    {
11        StringAssert.Contains(ex.Message, "El valor ingresado es invalido, solo se puede ingresar números positivos");
12        return;
13    }
14    Assert.Fail("Expected ArgumentOutOfRangeException for input zero with a specific message.");
15 }
```

```
PS C:\Users\Stryker\Desktop\Final-2024-II\CALIDAD_SOFTWARE\UNIT-II\lab-2024-ii-cps-u2-01-Luis-Caxi-Calani\MyMath> dotnet test --collect:"XPlat Code Coverage"
Determinando los proyectos que se van a restaurar...
Todos los proyectos están actualizados para la restauración.
Math.Lib -> C:\Users\Stryker\Desktop\Final-2024-II\CALIDAD_SOFTWARE\UNIT-II\lab-2024-ii-cps-u2-01-Luis-Caxi-Calani\MyMath\Math.Lib\bin\Debug\net8.0\Math.Lib.dll
Math.Tests -> C:\Users\Stryker\Desktop\Final-2024-II\CALIDAD_SOFTWARE\UNIT-II\lab-2024-ii-cps-u2-01-Luis-Caxi-Calani\MyMath\Math.Tests\bin\Debug\net8.0\Math.Tests.dll
Serie de pruebas para C:\Users\Stryker\Desktop\Final-2024-II\CALIDAD_SOFTWARE\UNIT-II\lab-2024-ii-cps-u2-01-Luis-Caxi-Calani\MyMath\Math.Tests\bin\Debug\net8.0\Math.Tests.dll (.NETCoreApp,Version=v8.0)
Versión 17.11.0 (x64) de VSTest

Correctas! - Con error:    0, Superado:    4, Omitido:    0, Total:    4, Duración: 75 ms - Math.Tests.dll (net8.0)

Hay actualizaciones de carga de trabajo disponibles. Ejecute "dotnet workload list" para obtener más información.
```