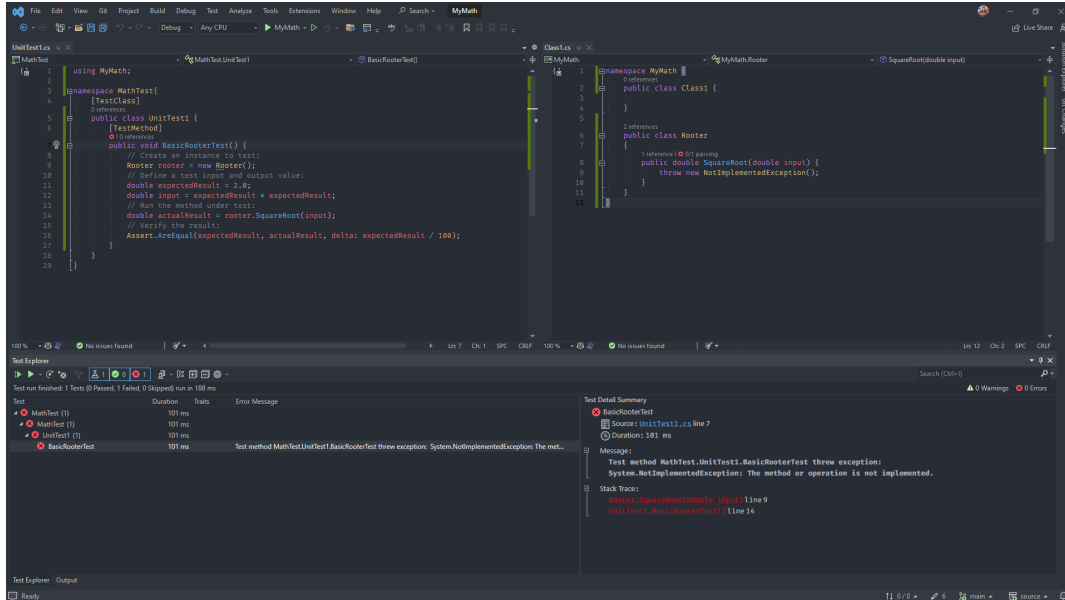


## A. Badilla Olivas - B80874 - lab 3 - Ingeniería de Software - ciclo II 2023 g1

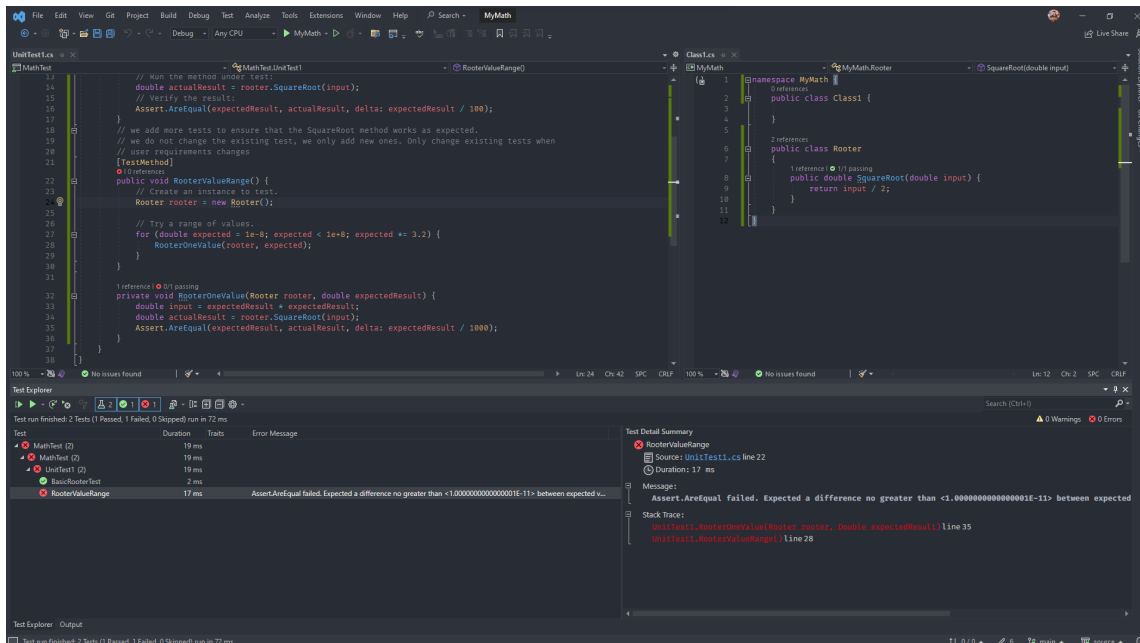
A) De la sección, Create a test and generate code, explique muy brevemente ¿Por qué la prueba que ejecutó en el paso #6 falló?

Falló porque el método de clase Router Square Root no está implementado.

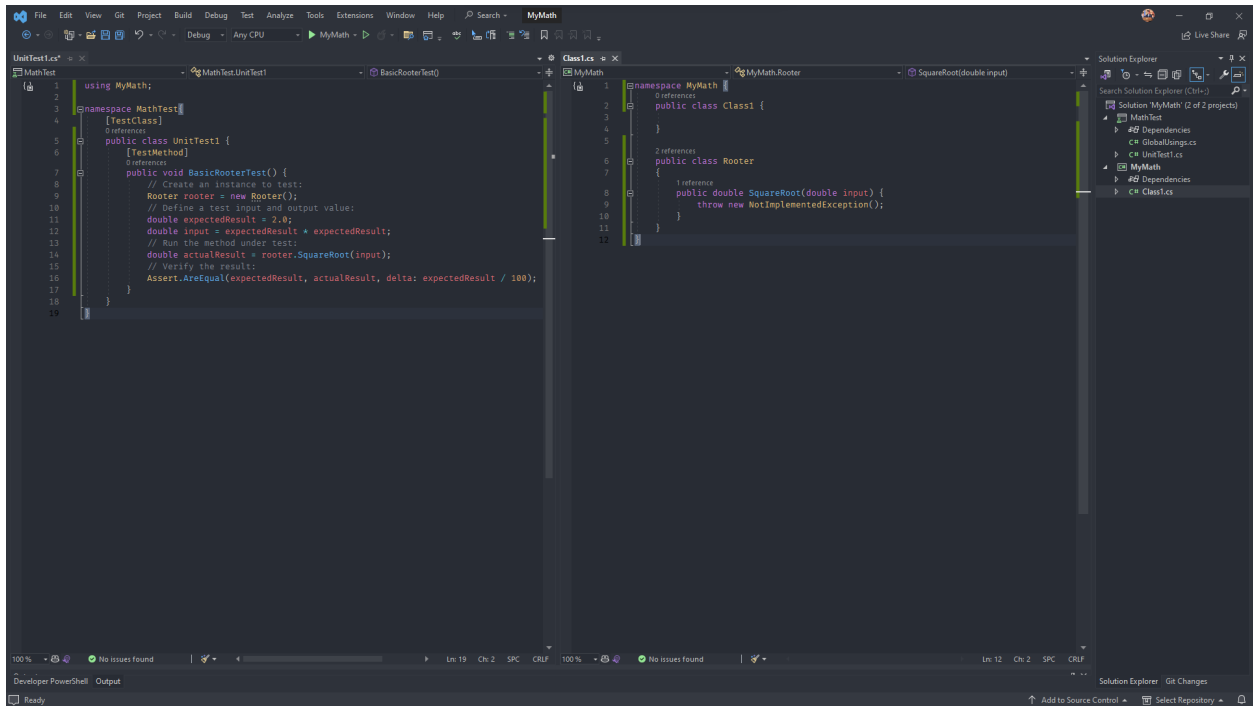
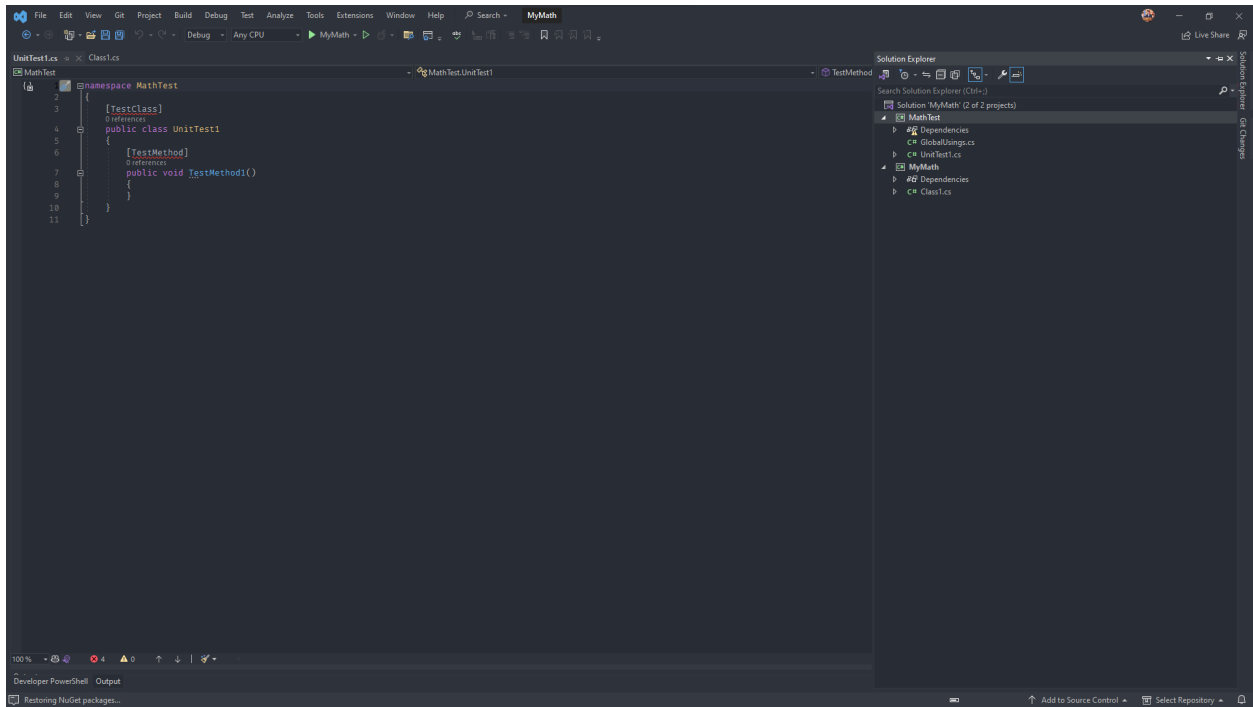


B) De la sección, Extend the range of inputs, explique muy brevemente ¿Por qué la prueba que ejecutó en el paso #2 falló?

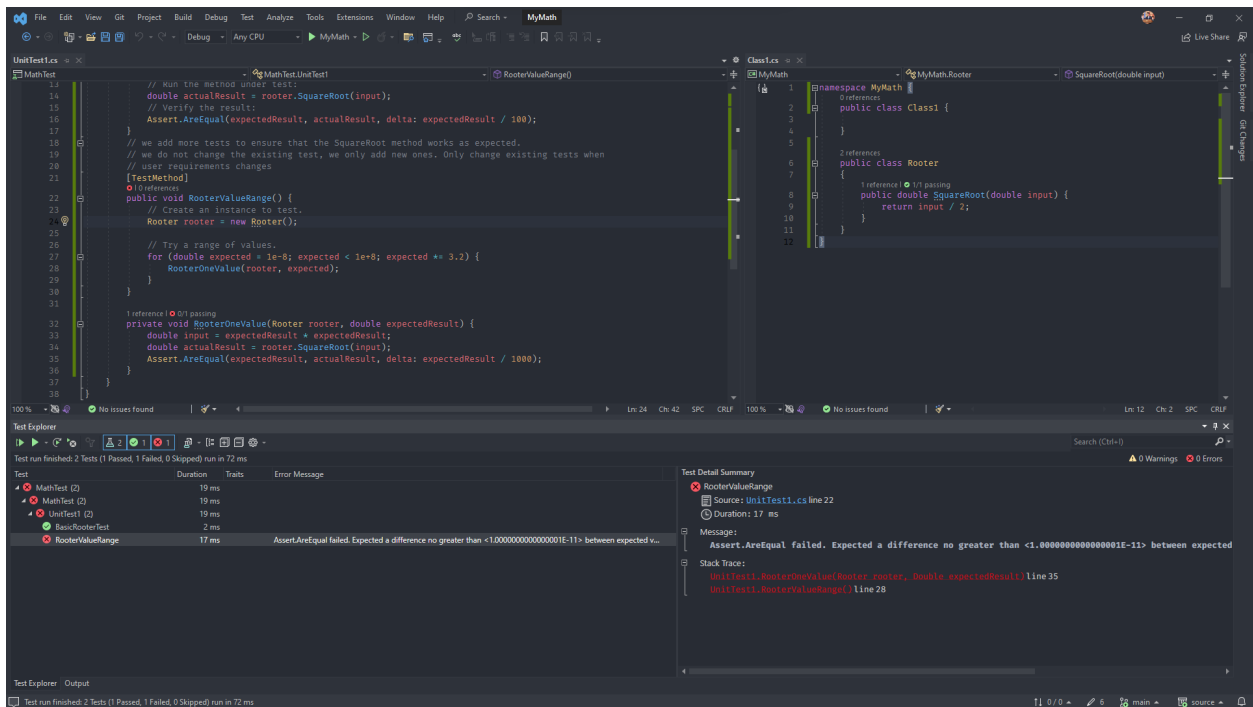
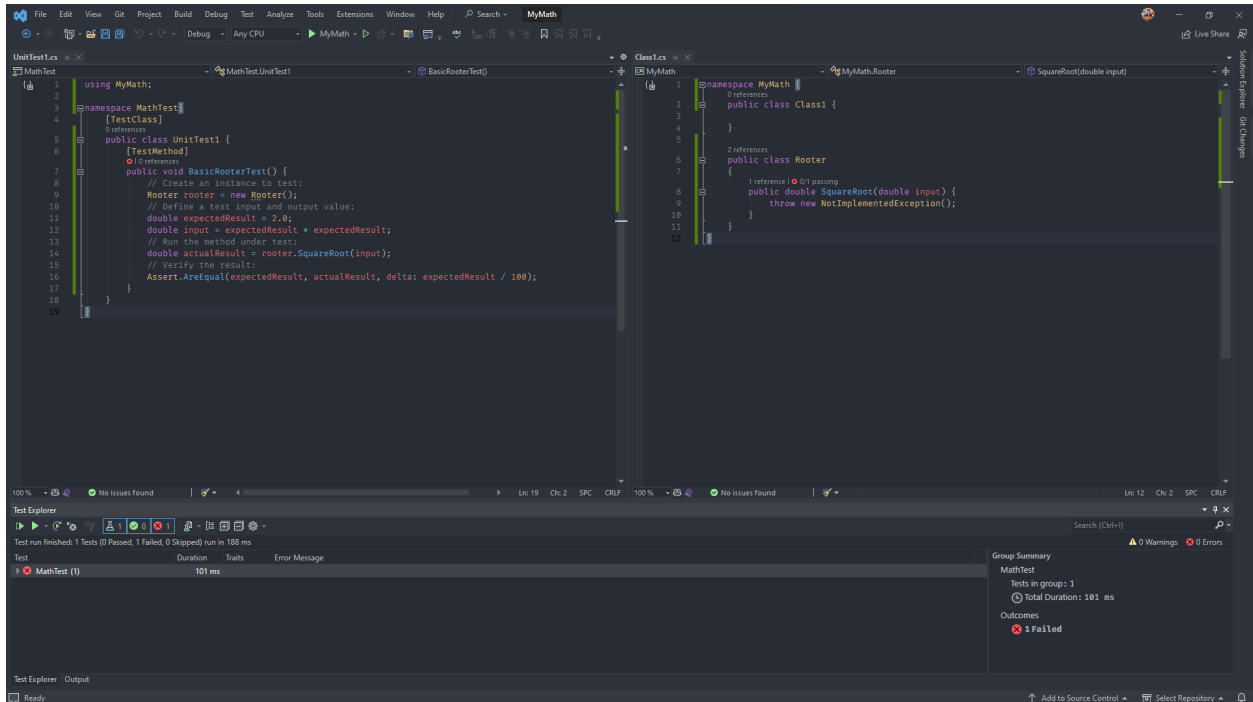
Falló porque el resultado fue diferente al esperado y la diferencia superó el margen de error que se estableció en el unit test.



## Otras evidencias



## A. Badilla Olivas - B80874 - lab 3 - Ingeniería de Software - ciclo II 2023 g1



## A. Badilla Olivas - B80874 - lab 3 - Ingeniería de Software - ciclo II 2023 g1

The screenshot displays the Visual Studio IDE with two main code files open: `MathTest.cs` and `MyMath.cs`.

**MathTest.cs** (Left Panel):

```
13 // Run the method under test:
14 double actualResult = rooter.SquareRoot(input);
15 // Verify the result:
16 Assert.AreEqual(expectedResult, actualResult, delta: expectedResult / 100);
17
18 // we add more tests to ensure that the SquareRoot method works as expected.
19 // we do not change the existing test, we only add new ones. Only change existing tests when
20 // user requirements changes
21 [TestMethod]
22 public void RooterValueRange() {
23     // Create an instance to test.
24     Rooter rooter = new Rooter();
25
26     // Try a range of values.
27     for (double expected = 1e-8; expected < 1e+8; expected *= 3.2) {
28         RooterOneValue(rooter, expected);
29     }
30
31     // reference 1/1 passing
32     private void RooterOneValue(Rooter rooter, double expectedResult) {
33         double input = expectedResult * expectedResult;
34         double actualResult = rooter.SquareRoot(input);
35         Assert.AreEqual(expectedResult, actualResult, delta: expectedResult / 1000);
36     }
37 }
38 }
```

**MyMath.cs** (Right Panel):

```
1 namespace MyMath {
2     public class Class1 {
3     }
4
5     public class Rooter {
6     }
7     public double SquareRoot(double input) {
8         double result = input; // Initialize the result with the input value
9         double previousResult = -input; // Initialize previousResult with an unlikely value
10
11         // Iterate while the difference between previousResult and result is larger than result
12         while (Math.Abs(previousResult - result) > result / 1000) {
13             previousResult = result; // Store the current result in previousResult
14
15             // Update result using the Newton-Raphson method formula
16             result = result - (result * result - input) / (2 * result);
17         }
18
19         return result; // Return the final approximated square root
20     }
21 }
22 }
```

**Test Explorer** (Bottom Left):

Test	Duration	Traits	Error Message
MathTest (2)	26 ms		
MathTest (2)	26 ms		
UnitTest1 (2)	26 ms		
BasicRooterTest	26 ms		
RooterValueRange	< 1 ms		

Test run finished: 2 Tests (2 Passed, 0 Failed, 0 Skipped) run in 79 ms

0 failed tests (Shift+click to toggle)

Test Detail Summary

- RooterValueRange
- Source: UnitTest1.cs line 22
- Duration: < 1 ms

Test Explorer Output

Test run finished: 2 Tests (2 Passed, 0 Failed, 0 Skipped) run in 79 ms

## A. Badilla Olivas - B80874 - lab 3 - Ingeniería de Software - ciclo II 2023 g1

The screenshot shows the Visual Studio IDE with the following details:

- UnitTests** window: Shows the `RooterTestNegativeInput` test failing. The error message is `Assert.ThrowsException failed. No exception thrown. ArgumentOutOfRangeException exception was expected.`
- Test Explorer** window: Shows the test results table below.
- Test Detail Summary** window: Shows the stack trace for the failed test, pointing to `UnitTests1.RooterTestNegativeInput() line 40`.

Test	Duration	Traits	Error Message
MathTest (3)	16 ms		
RooterTestNegativeInput	14 ms		Assert.ThrowsException failed. No exception thrown. ArgumentOutOfRangeException exception was expected.
RooterValueRange	< 1 ms		

The screenshot shows the Visual Studio IDE after the test has been fixed. The `RooterTestNegativeInput` test now passes.

Test	Duration	Traits	Error Message
MathTest (3)	19 ms		
RooterTestNegativeInput	< 1 ms		
RooterValueRange	< 1 ms		