UNIVERSIDAD PRIVADA DE TACNA

INGENIERIA DE SISTEMAS

TITULO:

Informe de Laboratorio 01:Elaboración de Pruebas UnitariasArchivo

CURSO:

Calidad y Pruebas de Software

DOCENTE:

Ing. Patrick Cuadros Quiroga

INTEGRANTES:

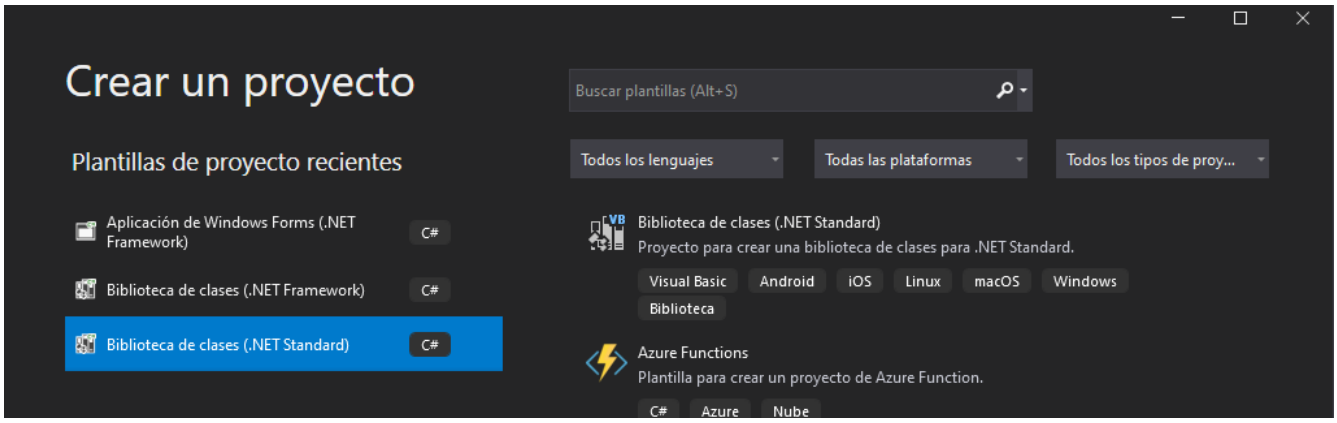
Katerin Almendra Merino Quispe (2018060918)

Tacna

Índice

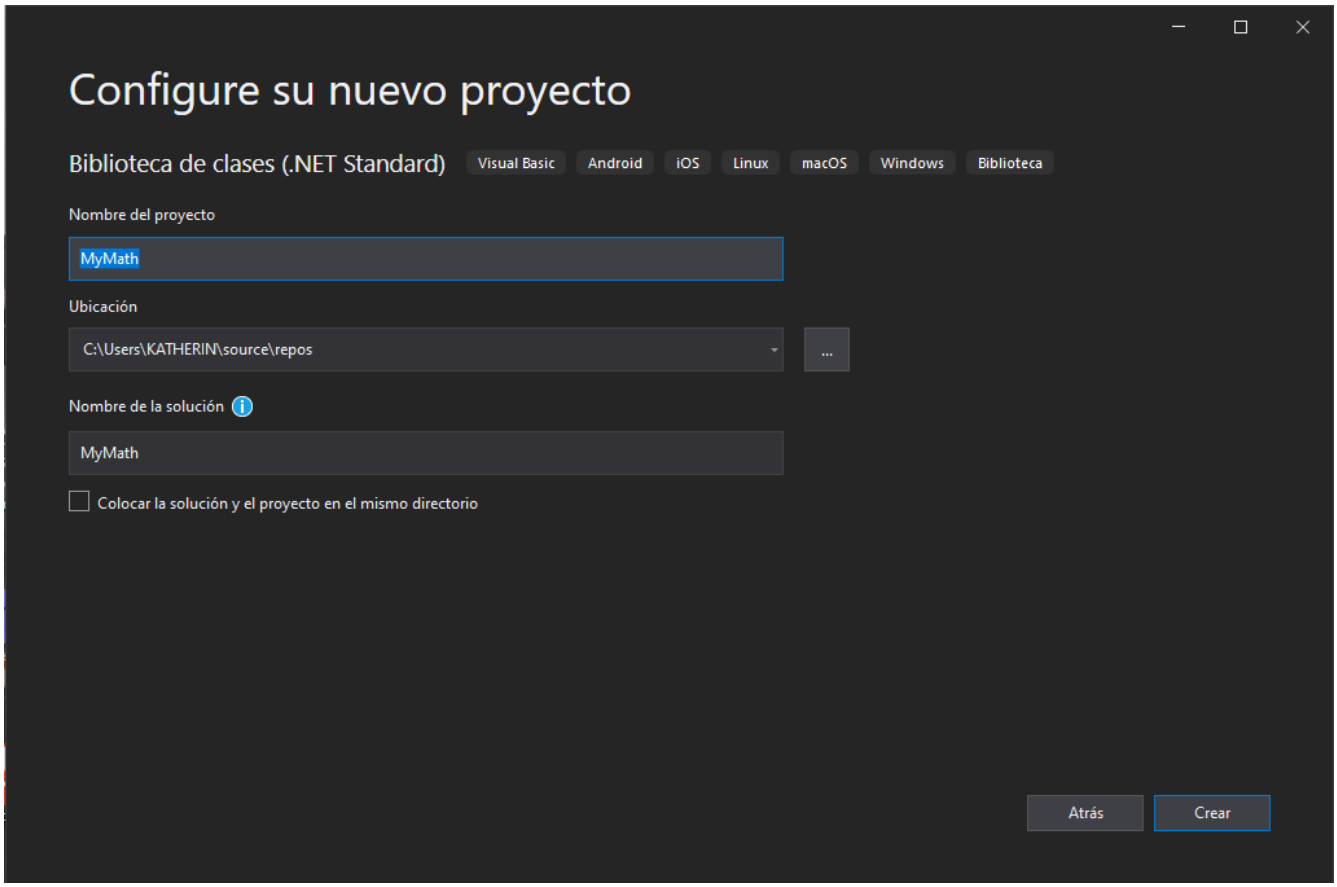
1. Parte 1: Creación de una prueba y generación del código

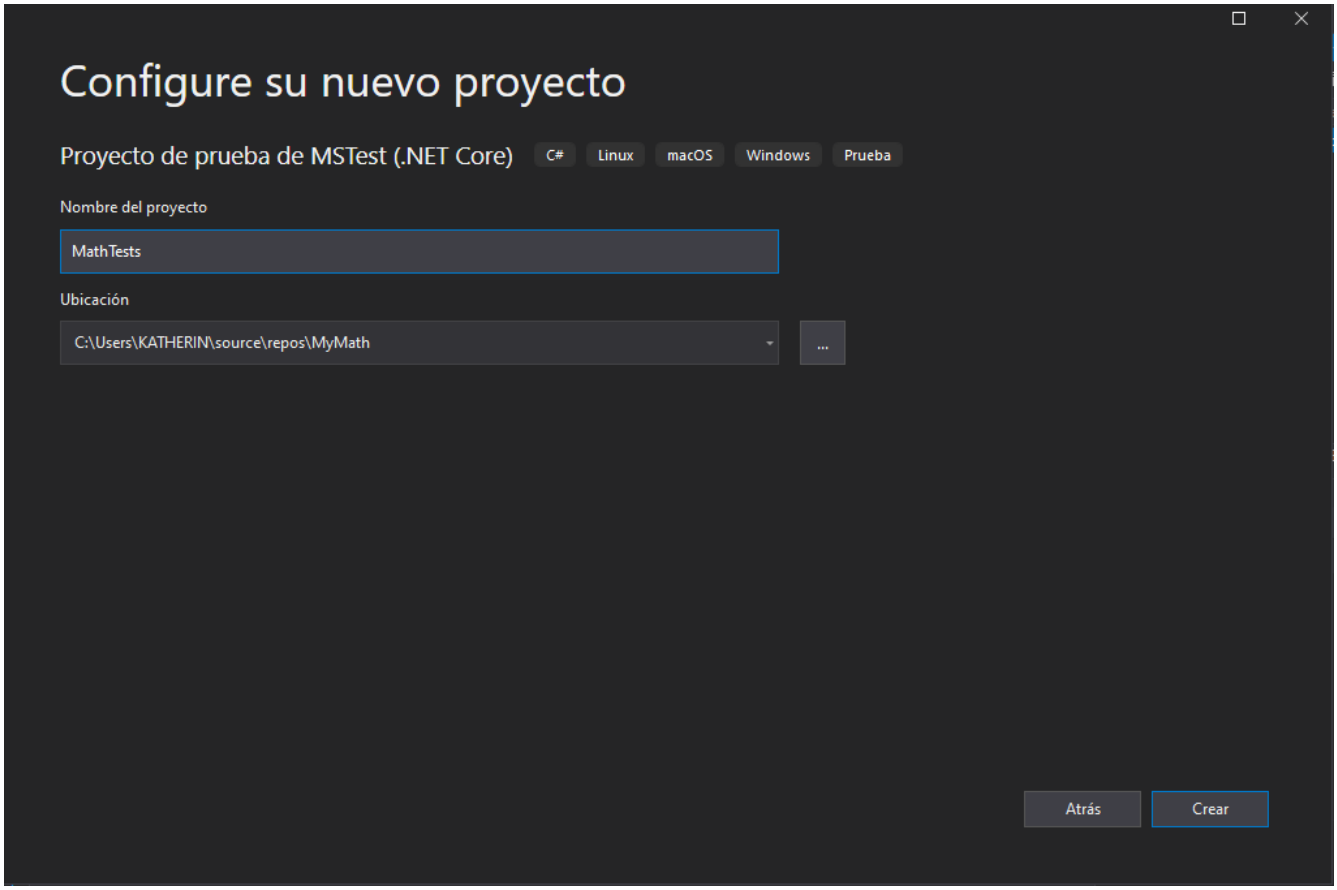
– Cree un proyecto de biblioteca de clases .NET Standard en C char. Este proyecto contendrá el código que quiere probar. Asigne al proyecto el nombre MyMath.



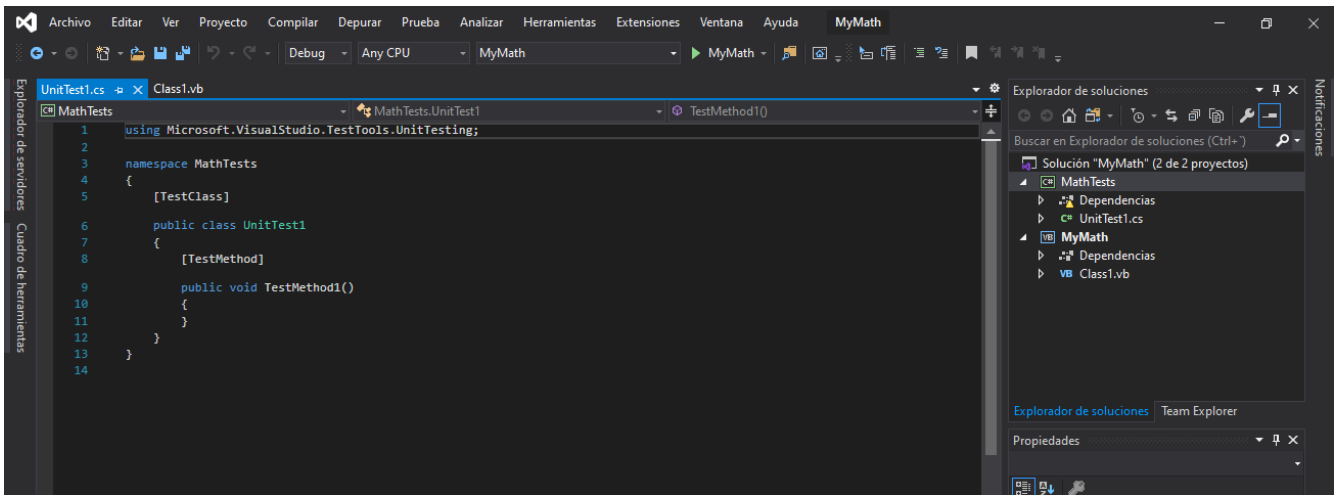
– En la misma solución, agregue un nuevo proyecto de prueba de MSTest (.NET Core) . Asigne al proyecto de prueba el nombre MathTests.

–

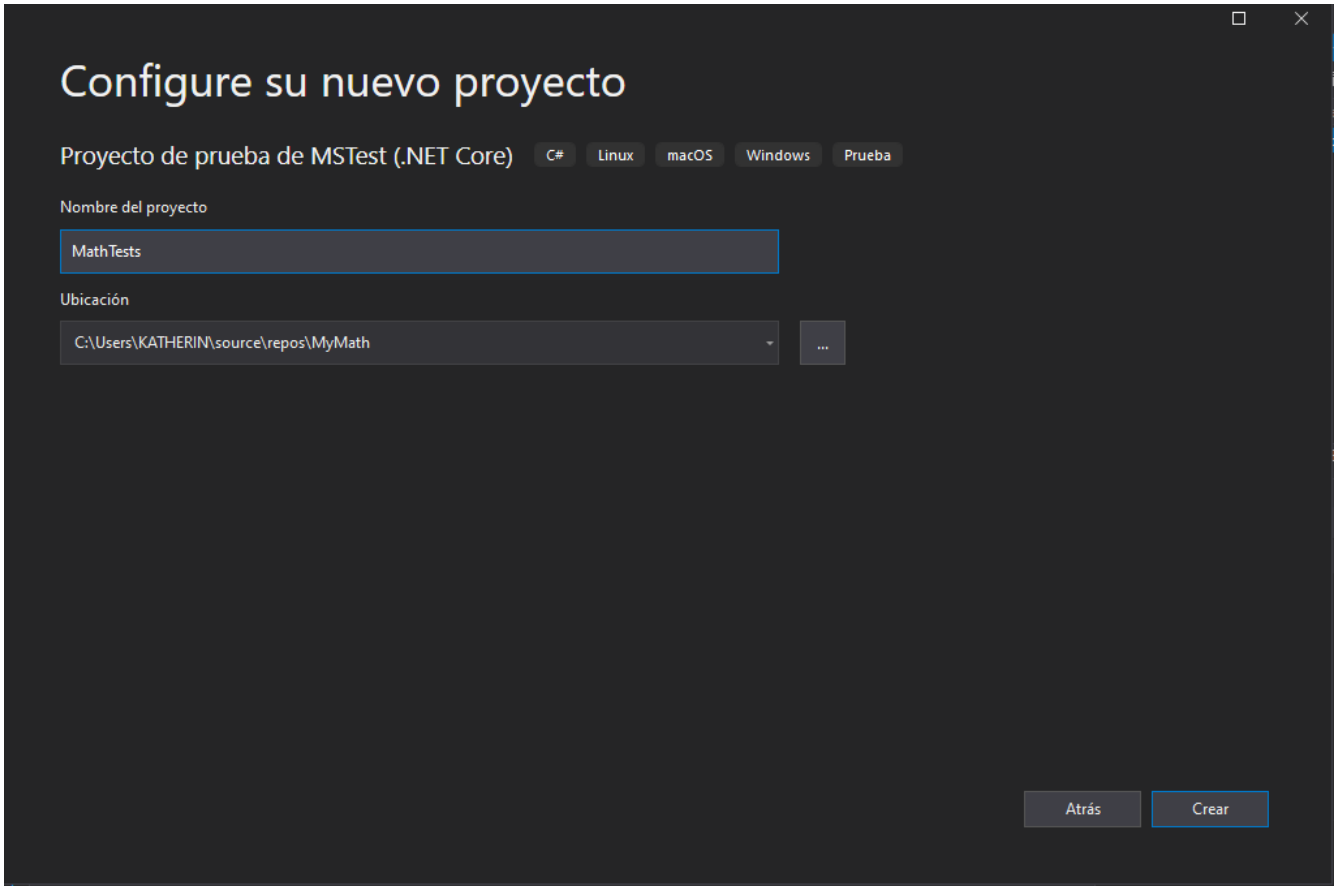




– Escriba un método de prueba simple que compruebe el resultado obtenido para una entrada específica. Agregue el código siguiente a la clase UnitTest1:



– En el cuadro de diálogo Generar tipo, establezca Proyecto en MyMath, el proyecto de biblioteca de clases y, luego, elija Aceptar.



– Genere un método a partir del código de prueba. Coloque el cursor en SquareRoot y, luego, en el menú de bombilla, elija Generar método Rooter.SquareRoot"

– Ejecute la prueba unitaria

2. Comprobación de un cambio en el código

– En el archivo Class1.cs, mejore el código de SquareRoot:

– En el Explorador de pruebas, elija Ejecutar todo.

3. Extensión del intervalo de entradas

- En la clase de prueba, agregue la siguiente prueba para un intervalo de valores de entrada:

- En el Explorador de pruebas, elija Ejecutar todo.

- Inspeccione el método que se está probando para ver qué puede ser incorrecto. Modifique el código SquareRoot de la forma siguiente:

- En el Explorador de pruebas, elija Ejecutar todo.

4. Agregue pruebas para casos excepcionales

- Agregue una nueva prueba para entradas negativas:

- En el Explorador de pruebas, elija Ejecutar todo.

- Corrija el código SquareRoot; para ello, agregue la siguiente instrucción if al principio del método:

- En el Explorador de pruebas, elija Ejecutar todo.

5. Refactorizar el codigo en pruebas

- Cambie la línea que calcula result en el método SquareRoot de la manera siguiente:

- Elija Ejecutar todo y compruebe que todas las pruebas se siguen superando.

6. Parte 2: Creación de una prueba unitaria utilizando un framework de pruebas (XUnit)

- Ejecutamos los siguientes comandos:

7. Crea una prueba

- Actualice el proyecto PrimeService.Tests :

- Ejecuta el corredor de prueba.La prueba falla porque IsPrimeno se ha implementado. Usando el enfoque TDD, escriba solo el código suficiente para que esta prueba pase. Actualice IsPrimecon el siguiente código:

- Ejecutar dotnet test. La prueba pasa.

- Reemplace el siguiente código:

- En el código anterior, [Theory]y [InlineData]habilite probar varios valores menores que dos. Dos es el número primo más pequeño.Ejecutar dotnet test, dos de las pruebas fallan:

- Para pasar todas las pruebas, actualice el IsPrimemétodo con el siguiente código:

- actualice el código de destino.El IsPrime método completado no es un algoritmo eficiente para probar la primalidad.