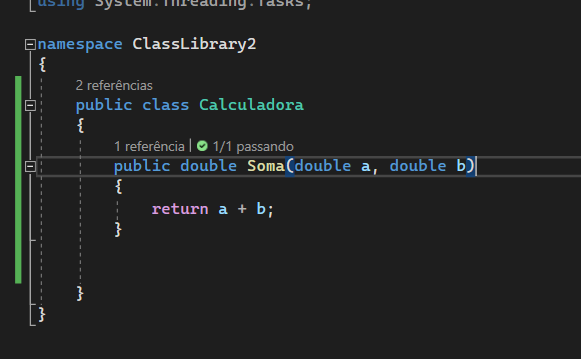
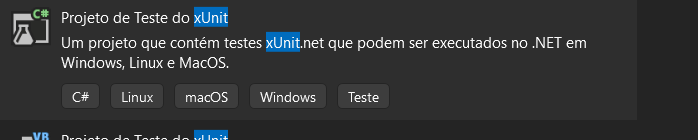
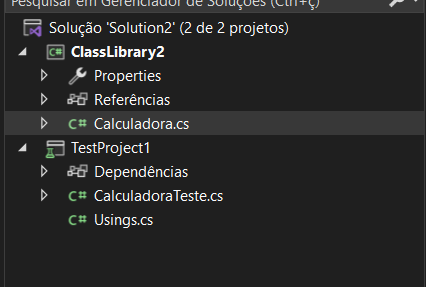
**XUNIT**

**Adicionando XUnit ao projeto:**

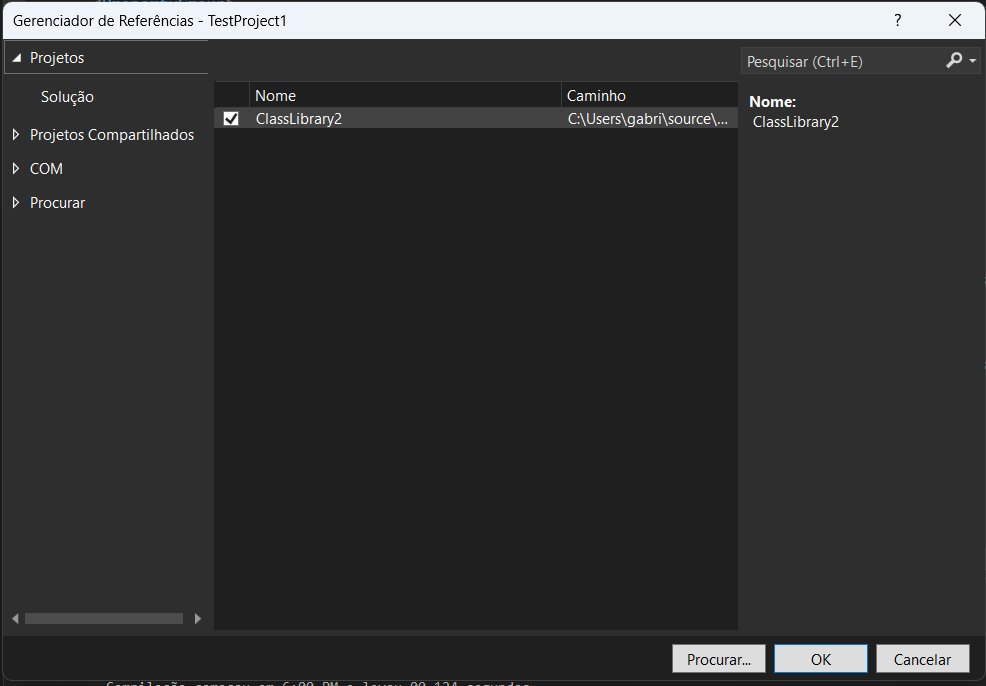
Basicamente existem duas formas de adicionarmos os testes a um projeto, podemos adicionar a dependência do XUnit ao projeto e criar uma estrutura de pastas especificas, ou podemos adicionar um projeto de testes na solução e adicionarmos os projetos que serão testados como dependências nesse projeto de teste:



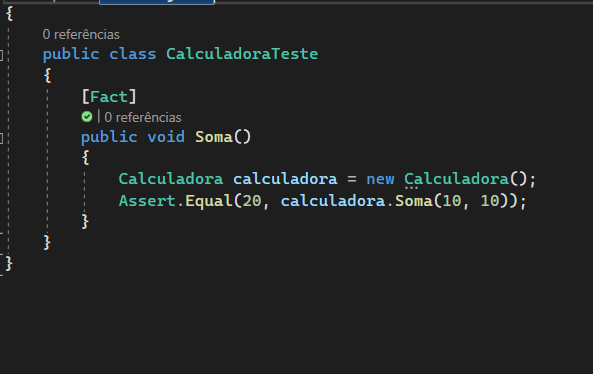








Após isso podemos criar as nossas classes de teste:

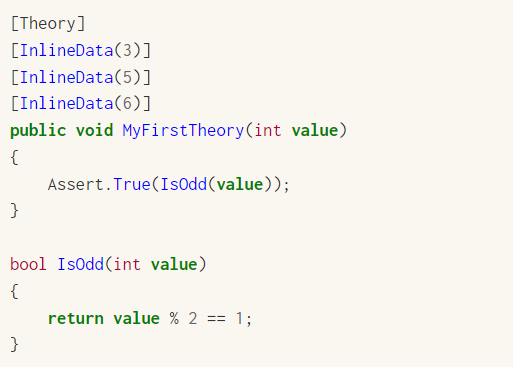


**Entendendo fatos e teorias:**

**Fatos** são testes que são sempre verdadeiros. Eles testam condições invariantes.

As **teorias** são testes que só são verdadeiros para um determinado conjunto de dados.

Em suma se usarmos apenas um teste que use Fatos, mas teste diferentes valores se um desses valores não corresponder com a análise todo o teste terá um erro, porém se usarmos uma teoria nesse método com valores diferentes cada valor será tratado como um teste separado:

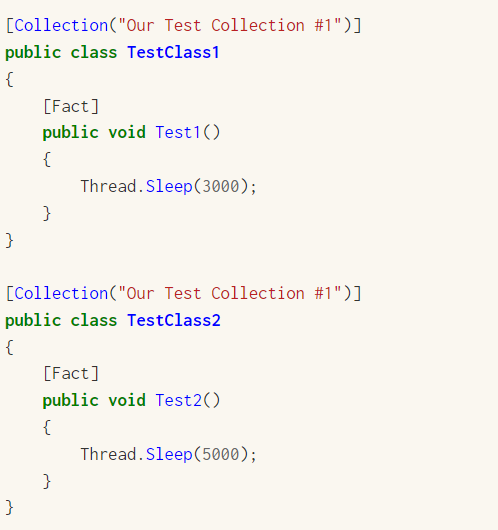


**Paralelismo de testes:**

Apartir da versão 2.0 do xUnit foi introduzido o conceito de coleções de testes, uma coleção tem os seus testes executados sem paralelismo e os testes que não estão na mesma coleção são executados em paralelo.

Por padrão todas as classes têm a sua própria coleção e todos os métodos dentro dela são executados um após o outro, porém se esses métodos estivessem em classes separadas eles seriam executados simultaneamente.

Podemos compartilhar coleções entre as classes através da anotação “[Collection]”:



**Contexto compartilhado entre classes:**

Contexto compartilhado na mesma classe:

Para isso usamos o construtor como um método “before” e a interface “dispose” para eliminar da memória:



Contexto compartilhado entre outras classes:

Basicamente para termos isso temos que criar uma IClassFigure que basicamente é uma figura de uma classe que sera passada entre as demais classes:

