

Trabajo Práctico 9 - Pruebas Unitarias

5.1 Preparamos el entorno:

Instalamos VS.Code: <https://code.visualstudio.com/download>

Desde linea de comandos clonamos el proyecto MiSimpleApp, entramos a la carpeta y abrimos VS.Code

```
git clone https://github.com/ingsoft3ucc/MiSimpleApp.git
```

```
cd MiSimpleApp
```

```
code .
```

Revisamos el código

Cerramos VS.Code

Nos movemos una carpeta hacia arriba y creamos un nuevo proyecto de pruebas unitarias con NUnit:

```
cd ..
```

```
dotnet new nunit -n MiSimpleAppTests
```

Entramos a la carpeta del nuevo proyecto y **agregamos los paquetes NUnit y**

NUnit.ConsoleRunner. Luego le agregamos al proyecto de pruebas una referencia al proyecto que vamos a probar. Nos movemos una carpeta hacia arriba y lo vemos en VS.Code

```
cd MiSimpleAppTests
```

```
dotnet add package NUnit
```

```
dotnet add package NUnit.ConsoleRunner
```

```
dotnet add reference ../MiSimpleApp/MiSimpleApp.csproj
```

```
cd ..
```

```
code .
```

```
● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % cd MiSimpleApp
● agostinamorellato@Agostinas-MacBook-Pro MiSimpleApp % code .
zsh: command not found: code
● agostinamorellato@Agostinas-MacBook-Pro MiSimpleApp % cd ..
● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % dotnet new nunit -n MiSimpleAppTests
The template "NUnit 3 Test Project" was created successfully.

Processing post-creation actions...
Restoring /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj...
  Determining projects to restore...
    Restored /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj (in 35.
Restore succeeded.
```

```
● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % cd MiSimpleAppTests
● agostinamorellato@Agostinas-MacBook-Pro MiSimpleAppTests % dotnet add package NUnit
  Determining projects to restore...
  Writing /var/folders/41/kx949byd2hb4x2s0hrqv565m000gn/T/tmpRyNlly.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7.0.400/trustedroots/codesignctl.pem'.
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7.0.400/trustedroots/timestampctl.pem'.
info : Adding PackageReference for package 'NUnit' into project '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj'.
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/nunit/index.json
info :     OK https://api.nuget.org/v3/registration5-gz-semver2/nunit/index.json 1073ms
info : Restoring packages for /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj...
info : Package 'NUnit' is compatible with all the specified frameworks in project '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj'.
info : PackageReference for package 'NUnit' version '3.13.3' updated in file '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj'.
info : Assets file has not changed. Skipping assets file writing. Path: /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/obj/project.assets.json
log : Restored /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj (in 72 ms).
```

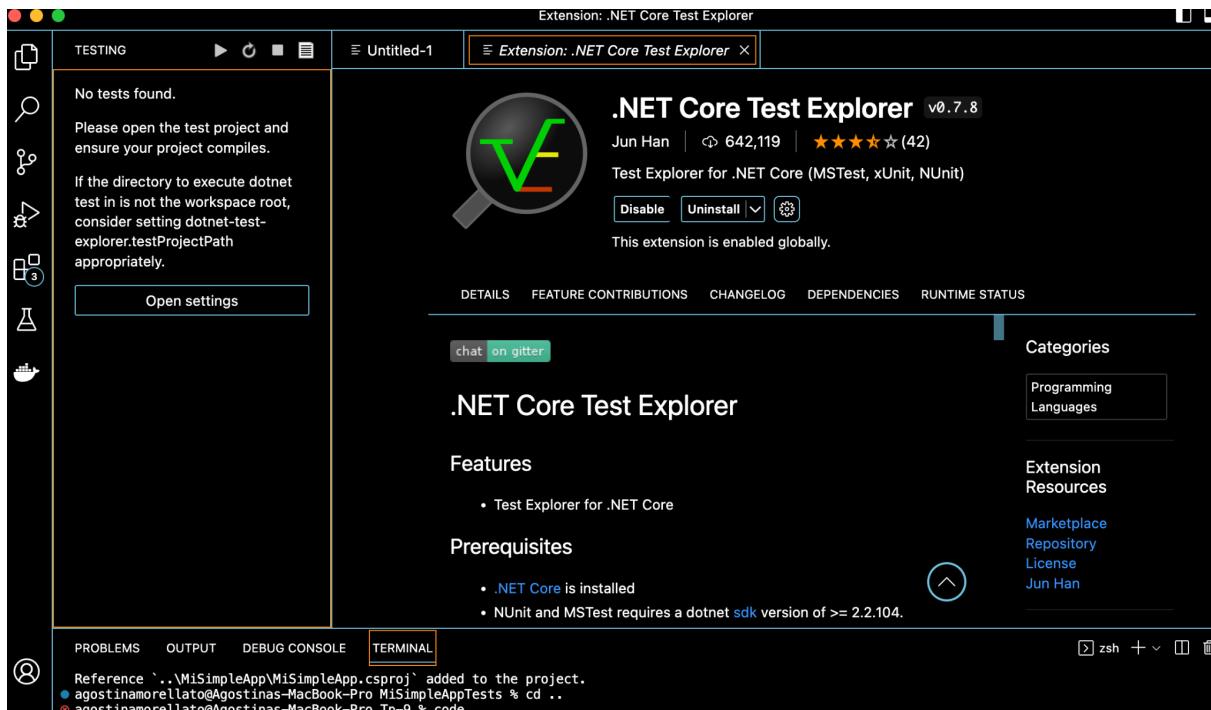
```

● agostinamorellato@Agostinas-MacBook-Pro MiSimpleAppTests % dotnet add package NUnit.ConsoleRunner
  Determining projects to restore...
  Writing /var/folders/41/kx949bydzhbx4z50hrqv565m000gn/T/tmpcrm68Y.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7.0.400/trustedroots/codesignctl.pem'.
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7.0.400/trustedroots/timestampctl.pem'.
info : Adding PackageReference for package 'NUnit.ConsoleRunner' into project '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj'.
info : GET https://api.nuget.org/v3/registration5-gz-semver2/nunit.consolerunner/index.json
info : OK https://api.nuget.org/v3/registration5-gz-semver2/nunit.consolerunner/index.json 326ms
info : Restoring packages for /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj...
info : GET https://api.nuget.org/v3-flatcontainer/nunit.consolerunner/index.json
info : OK https://api.nuget.org/v3-flatcontainer/nunit.consolerunner/index.json 284ms
info : GET https://api.nuget.org/v3-flatcontainer/nunit.consolerunner/3.16.3/nunit.consolerunner.3.16.3.nupkg
info : OK https://api.nuget.org/v3-flatcontainer/nunit.consolerunner/3.16.3/nunit.consolerunner.3.16.3.nupkg 26ms
info : Installed NUnit.ConsoleRunner 3.16.3 from https://api.nuget.org/v3/index.json with content hash bhc4f00Vd570xaulPu2dMuUWK6hLj5fWLwmUnvbxEA9w41sZNe6Xb3VPkcp+50riQqgmtcbbhX-oShoz62Jg==.
info : Package 'NUnit.ConsoleRunner' is compatible with all the specified frameworks in project '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj'.
info : PackageReference for package 'NUnit.ConsoleRunner' version '3.16.3' added to file '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj'.
info : Generating MSBuild file '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/obj/MiSimpleAppTests.csproj.nuget.g.props'.
info : Writing assets file to disk. Path: /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/obj/project.assets.json
log : Restored /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/MiSimpleAppTests.csproj (in 1.83 sec).

● agostinamorellato@Agostinas-MacBook-Pro MiSimpleAppTests % dotnet add reference ..\MiSimpleApp\MiSimpleApp.csproj
Reference `..\MiSimpleApp\MiSimpleApp.csproj` added to the project.
● agostinamorellato@Agostinas-MacBook-Pro MiSimpleAppTests % cd ..
● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % code .
zsh: command not found: code

```

Instalamos extensión ".NET Core Test Explorer"



Se abre el archivo settings.json y escribimos el nombre del proyecto de pruebas y el nombre de la dll resultante de la compilación del proyecto de pruebas:

```
{
  "dotnet-test-explorer.testProjectPath": "MiSimpleAppTests",
  "dotnet-test-explorer.testFileSuffix": ".Tests.dll"
}
```

Guardamos el archivo, volvemos hacer click en el ícono de pruebas y ahora sí tenemos la posibilidad de correr pruebas haciendo click en el botón de Play

```

settings.json — Tp-9
{
    "dotnet-test-explorer.testProjectPath": "MiSimpleAppTests",
    "dotnet-test-explorer.testFileSuffix": ".Tests.dll"
}

```



```

UnitTest1.cs — Tp-9
namespace MiSimpleAppTests;
public class Tests
{
    [SetUp]
    public void Setup()
    {
    }

    [Test]
    public void Test1()
    {
        Assert.Pass();
    }
}

```

Revisamos el código del proyecto MiSimpleApp:

```

Clases.cs — Tp-9
public class Reservation
{
    public User MadeBy { get; set; }
    public bool CanBeCancelledBy(User user)
    {
        if (user.IsAdmin)
            return true;
        if (MadeBy==user)
            return true;
        return false;
    }
}

public class User
{
    public bool IsAdmin { get; set; }
}

```

The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: Shows the project structure with files like `settings.json`, `.vscode`, `Program.cs`, `Clases.cs`, `MiSimpleApp.csproj`, and `UnitTest1.cs`.
- OPEN EDITORS**:
 - `settings.json`
 - `Program.cs` (highlighted)
- Program.cs Content**:

```
1 using System;
2
3 class Program
4 {
5     static void Main()
6     {
7         Reservation reservation = new Reservation();
8         User user=new User();
9         user.IsAdmin=true;
10        bool result=reservation.CanBeCancelledBy(user);
11        Console.WriteLine(result);
12    }
13 }
```

The screenshot shows the VS Code interface with the following details:

- TESTING** sidebar: Shows test cases:
 - CanBeCancelledBy_AdminCancelling_ReturnsTrue
 - CanBeCancelledBy_AnotherUserCancelling_ReturnsFalse
 - CanBeCancelledBy_SameUserCancelling_ReturnsTrue
- OPEN EDITORS**:
 - `settings.json`
 - `UnitTest1.cs` (highlighted)
- UnitTest1.cs Content**:

```
34 //Arrange
35 User user=new User();
36 Reservation reservation=new Reservation();
37 reservation.MadeBy=user;
38
39 //Act
40 bool result=reservation.CanBeCancelledBy(user);
41
42 //Assert
43 Assert.That(result,Is.True);
44
45
46
47
48 [Test]
49 0 references | Run Test | Debug Test
50 public void CanBeCancelledBy_AnotherUserCancelling_ReturnsFalse()
51 {
52     //Arrange
53     User user=new User();
54     Reservation reservation=new Reservation();
55     reservation.MadeBy=user;
56
57     //Act
58     User newUser=new User();
59     bool result=reservation.CanBeCancelledBy(newUser);
60
61     //Assert
62     Assert.That(result,Is.False);
63 }
```

5.2: Creamos nuestros Tests:

Dado que el método `CanBeCancelledBy` de la clase `Reservation` tiene una lógica con 3 caminos posibles, debemos probar esos 3 caminos:

Modificamos nuestro archivo `UnitTest1.cs` del proyecto `MiSimpleAppTests`

5.4 Ejecutamos los tests:

```

1  namespace MiSimpleAppTests;
2
3  [TestFixture]
4  public class Tests
5  {
6      [SetUp]
7      public void Setup()
8      {
9      }
10
11     [Test]
12     public void CanBeCancelledBy_AdminCancelling_ReturnsTrue()
13     {
14         //Arrange
15
16         User user=new User();
17         user....;
18         user.IsAdmin=true;
19         Reservation reservation=new Reservation();
20
21         //Act
22
23         bool result=reservation.CanBeCancelledBy(user);
24
25         //Assert
26
27         //Assert.IsTrue(result);
28         Assert.That(result, Is.True);
    }
}

```

Explicar qué hace cada uno de los tests:

Los tests unitarios en el código están probando el comportamiento de un método llamado `CanBeCancelledBy` en la clase `Reservation`.

1. **CanBeCancelledBy_AdminCancelling_ReturnsTrue:**

- Se configuran las condiciones iniciales para el test en **Arrange**. Se crea una instancia de la clase `User` y otra de `Reservation`. Y se establece que el usuario es un administrador (IsAdmin=true).
- Se llama al método `CanBeCancelledBy` en la instancia de `Reservation` en **Act** y se pasa el usuario como argumento.
- Se verifica si el resultado del método `CanBeCancelledBy` es verdadero cuando un administrador intenta cancelar la reserva en **Assert**. Se utiliza el método `Assert.That(result, Is.True)` para afirmar que el resultado debe ser verdadero.

2. **CanBeCancelledBy_SameUserCancelling_ReturnsTrue:**

- Se configuran condiciones iniciales similares a las del primer test, pero en este caso, el usuario que hace la reserva (MadeBy) es el mismo que está tratando de cancelarla.(**Arrange**)
- Se llama al método `CanBeCancelledBy` con el mismo usuario.(**Act**)
- Se verifica que el resultado del método sea verdadero, lo que significa que un usuario puede cancelar su propia reserva.(**Assert**)

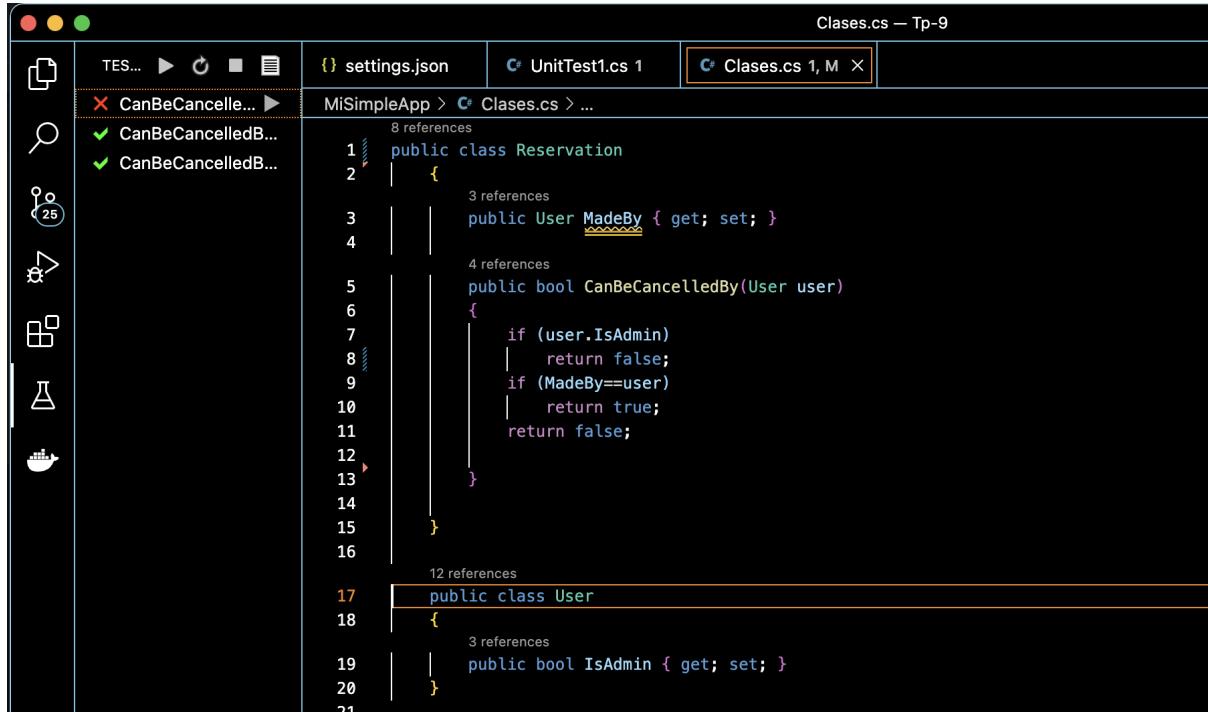
3. **CanBeCancelledBy_AnotherUserCancelling_ReturnsFalse:**

- Se configuran condiciones iniciales de nuevo con un usuario que hace la reserva y otro usuario que intenta cancelarla.
- Se llama al método `CanBeCancelledBy` con el nuevo usuario como argumento.
- Se verifica que el resultado sea falso, por lo que, otro usuario no puede cancelar la reserva hecha por un usuario diferente.

Estos tests están siguiendo el patrón AAA (Arrange-Act-Assert), donde se preparan las condiciones iniciales (Arrange), se realiza una acción (Act), y luego se verifica el resultado (Assert) para asegurarse de que el método `CanBeCancelledBy` se comporte como se espera en diferentes situaciones.

Modificamos la lógica de nuestro código bajo prueba haciendo que devuelva false la linea 9 (recordar guardar el archivo):

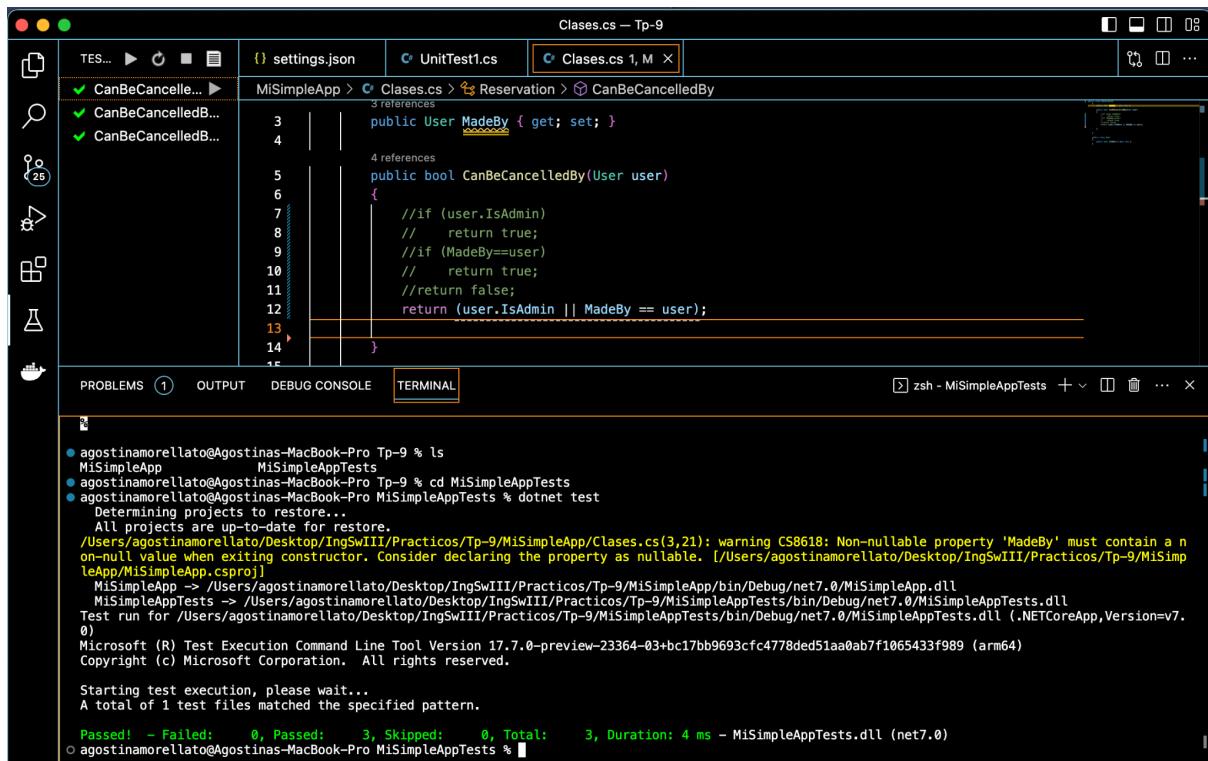
Volvemos a ejecutar los tests:



```
Clases.cs - Tp-9
settings.json UnitTest1.cs Clases.cs 1, M X
MiSimpleApp > Clases.cs > ...
public class Reservation
{
    public User MadeBy { get; set; }
}
public bool CanBeCancelledBy(User user)
{
    if (user.IsAdmin)
        return false;
    if (MadeBy==user)
        return true;
    return false;
}
public class User
{
    public bool IsAdmin { get; set; }
}
```

Dejamos la línea 9 de nuestro código bajo prueba como estaba originalmente:

Por último ejecutamos nuestros tests desde la linea de comandos. Nos posicionamos en el directorio de nuestro proyecto de pruebas y ejecutamos el comando dotnet test



```
Clases.cs - Tp-9
settings.json UnitTest1.cs Clases.cs 1, M X
MiSimpleApp > Clases.cs > Reservation > CanBeCancelledBy
public User MadeBy { get; set; }
public bool CanBeCancelledBy(User user)
{
    //if (user.IsAdmin)
    //    return true;
    //if (MadeBy==user)
    //    return true;
    //return false;
    return (user.IsAdmin || MadeBy == user);
}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
zsh - MiSimpleAppTests + × ⓘ ...
```

```
astinamorellato@Agostinas-MacBook-Pro Tp-9 % ls
MiSimpleApp
MiSimpleAppTests
astinamorellato@Agostinas-MacBook-Pro Tp-9 % cd MiSimpleAppTests
astinamorellato@Agostinas-MacBook-Pro MiSimpleAppTests % dotnet test
Determining projects to restore...
All projects are up-to-date for restore.
MiSimpleAppPracticos.Tp-9/MiSimpleApp/Clases.cs(3,21): warning CS8618: Non-nullable property 'MadeBy' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleApp/MiSimpleApp.csproj]
MiSimpleApp -> /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleApp/bin/Debug/net7.0/MiSimpleApp.dll
MiSimpleAppTests -> /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/bin/Debug/net7.0/MiSimpleAppTests.dll
Test run for /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiSimpleAppTests/bin/Debug/net7.0/MiSimpleAppTests.dll (.NETCoreApp, Version=v7.0)
Microsoft (R) Test Execution Command Line Tool Version 17.7.0-preview-23364-03+bc17bb9693fc4778ded51aa0ab7f106543f989 (arm64)
Copyright (c) Microsoft Corporation. All rights reserved.

Starting test execution, please wait...
A total of 1 test files matched the specified pattern.

Passed! - Failed: 0, Passed: 3, Skipped: 0, Total: 3, Duration: 4 ms - MiSimpleAppTests.dll (net7.0)
astinamorellato@Agostinas-MacBook-Pro MiSimpleAppTests %
```

6- Desarrollo de Pruebas Unitarias sobre una WebAPI:

Clonamos nuestro repo SimpleWebAPI

```
git clone https://github.com/ingsoft3ucc/SimpleWebAPI.git
```

Creamos proyecto de pruebas

```
dotnet new nunit -n SimpleWebAPITests
```

Entramos a la carpeta del nuevo proyecto y le agregamos al proyecto de pruebas una referencia al proyecto que vamos a probar. Nos movemos una carpeta hacia arriba y lo vemos en VS.Code

```
cd SimpleWebAPITests
```

```
dotnet add reference ../SimpleWebAPI/SimpleWebAPI/SimpleWebAPI.csproj
```

```
cd ..
```

```
code .
```

```
● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % git clone https://github.com/ingsoft3ucc/SimpleWebAPI.git
  Cloning into 'SimpleWebAPI'...
  remote: Enumerating objects: 150, done.
  remote: Counting objects: 100% (150/150), done.
  remote: Compressing objects: 100% (115/115), done.
  remote: Total 150 (delta 60), reused 33 (delta 9), pack-reused 0
  Receiving objects: 100% (150/150), 28.41 KiB | 538.00 KiB/s, done.
  Resolving deltas: 100% (60/60), done.
● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % dotnet new nunit -n SimpleWebAPITests
  The template "NUnit 3 Test Project" was created successfully.

  Processing post-creation actions...
  Restoring /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/SimpleWebAPITests/SimpleWebAPITests.csproj:
    Determining projects to restore...
    Restored /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/SimpleWebAPITests/SimpleWebAPITests.csproj (in 125 ms).
  Restore succeeded.

● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % cd SimpleWebAPITests
● agostinamorellato@Agostinas-MacBook-Pro SimpleWebAPITests % dotnet add reference ../SimpleWebAPI/SimpleWebAPI/SimpleWebAPI.csproj
  Reference '../SimpleWebAPI/SimpleWebAPI/SimpleWebAPI.csproj' added to the project.
● agostinamorellato@Agostinas-MacBook-Pro SimpleWebAPITests % cd ..
```

Se abre el archivo settings.json y escribimos el nombre del proyecto de pruebas y el nombre de la dll resultante de la compilación del proyecto de pruebas:

```
{
  "dotnet-test-explorer.testProjectPath": "SimpleWebAPITests",
  "dotnet-test-explorer.testFileSuffix": ".Tests.dll"
}
```

Creamos Tests

Reemplazamos el código de UnitTest1.cs por:

```
using Microsoft.Extensions.Logging;
using NUnit.Framework;
using SimpleWebAPI.Controllers;
using System;
using System.Linq;

namespace SimpleWebAPI.Tests
{
  [TestFixture]
  public class WeatherForecastControllerTests
  {
    [Test]
    public void Get>ReturnsWeatherForecasts()
    {
      // Arrange
```

```

    ILogger<WeatherForecastController> logger = new
LoggerFactory().CreateLogger<WeatherForecastController>();
    var controller = new WeatherForecastController(logger);

    // Act
    var result = controller.Get();

    // Assert
    Assert.NotNull(result);
    Assert.AreEqual(5, result.Count());
}
}
}

```

The screenshot shows the Visual Studio Code interface. The left sidebar contains the Explorer, showing the project structure with files like `settings.json`, `UnitTest1.cs` (the current file), and `Clases.cs`. The right side shows the code editor with the following content:

```

1  using Microsoft.Extensions.Logging;
2  using NUnit.Framework;
3  using SimpleWebAPI.Controllers;
4  using System;
5  using System.Linq;
6
7  namespace SimpleWebAPI.Tests
8  {
9      [TestFixture]
10     public class WeatherForecastControllerTests
11     {
12         [Test]
13         public void Get_ReturnsWeatherForecasts()
14         {
15             // Arrange
16             ILogger<WeatherForecastController> logger = new LoggerFactory().CreateLogger();
17             var controller = new WeatherForecastController(logger);
18
19             // Act
20             var result = controller.Get();
21
22             // Assert
23             Assert.NotNull(result);
24             Assert.AreEqual(5, result.Count());
25         }
26     }
27 }

```

The terminal at the bottom shows the following command-line output:

```

agostinamorellato@Agostinas-MacBook-Pro SimpleWebAPITests % dotnet add reference ../SimpleWebAPI/SimpleWebAPI/SimpleWebAPI.csproj
Reference `../SimpleWebAPI/SimpleWebAPI/SimpleWebAPI.csproj` added to the project.
agostinamorellato@Agostinas-MacBook-Pro SimpleWebAPITests % cd ..

```

Explicar qué hace este test:

Este test unitario está probando el método `Get` de la clase `WeatherForecastController`.

1. Arrange: Se crea una instancia de un logger

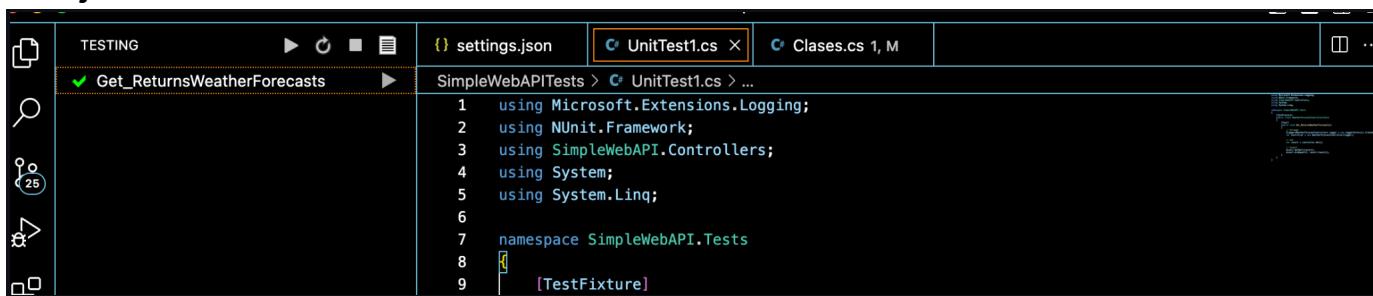
(`ILogger<WeatherForecastController>`) utilizando `Microsoft.Extensions.Logging` para registrar eventos y seguimiento. Se crea una instancia del controlador `WeatherForecastController` pasando el logger como argumento.

2. Act: Se llama al método GET en la instancia del controlador para obtener los pronósticos del tiempo.

3. Assert:

- `Assert.NotNull(result)`: Se verifica que el “result” no sea nulo, lo que asegura que el método GET haya retorna algún valor.
- `Assert.AreEqual(5, result.Count())`: Se verifica que el número de pronósticos del tiempo en el resultado sea igual a 5. Esto está probando si el método GET está generando una lista de pronósticos con una longitud de 5 elementos.
Esto es para asegurarse de que el controlador funcione como se espera en una aplicación web API.

6.4 Ejecutamos el test:



```
TESTING      ► ○ ■ ┌ ┘ | settings.json | UnitTest1.cs | Clases.cs 1, M | ┌ ┘ ..  
Get_ReturnsWeatherForecasts ► SimpleWebAPITests > UnitTest1.cs > ...  
1 using Microsoft.Extensions.Logging;  
2 using NUnit.Framework;  
3 using SimpleWebAPI.Controllers;  
4 using System;  
5 using System.Linq;  
6  
7 namespace SimpleWebAPI.Tests  
8 [TestFixture]  
9
```

8.1 Preparamos el entorno.

Clonamos una app de consola en .NET Core que hace uso de un servicio externo (una llamada a una API Rest) y la abrimos en VS.Code

```
git clone https://github.com/ingsoft3ucc/MiNotSoSimpleApp.git
```

```
cd MiNotSoSimpleApp
```

```
code .
```

```
zsh: command not found: code  
● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % git clone https://github.com/ingsoft3ucc/MiNotSoSimpleApp.git  
Cloning into 'MiNotSoSimpleApp'...  
remote: Enumerating objects: 22, done.  
remote: Counting objects: 100% (22/22), done.  
remote: Compressing objects: 100% (21/21), done.  
remote: Total 22 (delta 6), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (22/22), 5.19 KiB | 5.19 MiB/s, done.  
Resolving deltas: 100% (6/6), done.
```

Ejecutamos la app y vemos como nos devuelve 100 items desde la API:

```
8  {
9      0 referencias
10     class Program
11     {
12         0 referencias
13         static async Task Main()
14         {
15             var serviceProvider = new ServiceCollection()
16                 .AddHttpClient()
17                 .AddTransient<IApiService, ApiService>()
18                 .BuildServiceProvider();
19
20             var apiService = serviceProvider.GetRequiredService<IApiService>();
21
22             var myModels = await apiService.GetMyModelsAsync();
23             foreach (var model in myModels)
24             {
25                 Console.WriteLine($"Id: {model.Id}, Title: {model.Title}");
26             }
27         }
28     }
29 
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Id: 90, Title: ad iusto omnis odit dolor voluptatibus
Id: 91, Title: aut amet sed
Id: 92, Title: ratione ex tenetur perferendis
Id: 93, Title: beatae soluta recusandae
Id: 94, Title: qui qui voluptates illo iste minima
Id: 95, Title: id minus libero illum nam ad officiis
Id: 96, Title: quaerat velit veniam amet cupiditate aut numquam ut sequi
Id: 97, Title: quas fugiat ut perspiciatis vero provident
Id: 98, Title: laboriosam dolor voluptates
Id: 99, Title: temporibus sit alias delectus eligendi possimus magni
Id: 100, Title: at nam consequatur ea labore ea harum
El programa "[88618] MiNotSoSimpleApp.dll" terminó con el código 0 (0x0).
```

Analizamos el código

Identificamos el servicio externo y su interfaz a mockear:

En el código, se puede identificar como servicio externo Clase ApiService, implementa la interfaz IApiService y se encarga de realizar llamadas HTTP al servicio externo para obtener los modelos MyModel. Utiliza una instancia de HttpClient para hacer las solicitudes.

La interfaz que se puede "mockear" para probar la lógica de la aplicación de manera aislada, es Interfaz IApiService, define un contrato para la comunicación con un servicio externo que se encarga de obtener una colección de modelos llamados MyModel.

- Cerramos VS.Code y creamos el proyecto de NUnit:

cd ..

dotnet new nunit -n MiNotSoSimpleAppTests

cd MiNotSoSimpleAppTests

dotnet add reference ..\MiNotSoSimpleApp\MiNotSoSimpleApp.csproj

dotnet add package NUnit

dotnet add package Moq

dotnet add package NUnit3TestAdapter

dotnet add package Microsoft.Extensions.Http

dotnet add package Microsoft.Extensions.DependencyInjection

Abrimos VS.Code y configuramos settings:

cd ..

```
● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % dotnet new nunit -n MiNotSoSimpleAppTests
The template "NUnit 3 Test Project" was created successfully.

Processing post-creation actions...
Restoring /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj:
  Determining projects to restore...
    Restored /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj (in 131 ms).
Restore succeeded.

● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % cd MiNotSoSimpleAppTests
● agostinamorellato@Agostinas-MacBook-Pro MiNotSoSimpleAppTests % dotnet add reference ../MiNotSoSimpleApp/MiNotSoSimpleApp.csproj
Reference `..\MiNotSoSimpleApp\MiNotSoSimpleApp.csproj` added to the project.

● agostinamorellato@Agostinas-MacBook-Pro MiNotSoSimpleAppTests % dotnet add package NUnit
  Determining projects to restore...
    Writing /var/folders/41/kx949byd2hb4x2s0hrqv565m000gn/T/tmpQEPfsA.tmp
    info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7.0.400/trustedroots/codesignctl.pem'.
    info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7.0.400/trustedroots/timestampctl.pem'.
    info : Adding PackageReference for package 'NUnit' into project '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
    info : CACHE https://api.nuget.org/v3/registration5-gz-semver2/nunit/index.json
    info : Restoring packages for /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj...
    info :   GET https://api.nuget.org/v3-flatcontainer/system.net.http.json/index.json
    info :     OK https://api.nuget.org/v3-flatcontainer/system.net.http.json/index.json 440ms
    info :   GET https://api.nuget.org/v3-flatcontainer/system.net.http.json/7.0.0/system.net.http.json.7.0.0.nupkg
    info :     OK https://api.nuget.org/v3-flatcontainer/system.net.http.json/7.0.0/system.net.http.json.7.0.0.nupkg 21ms
    info :   GET https://api.nuget.org/v3-flatcontainer/system.text.json/index.json
    info :     OK https://api.nuget.org/v3-flatcontainer/system.text.json/index.json 726ms
    info :   GET https://api.nuget.org/v3-flatcontainer/system.text.json/7.0.0/system.text.json.7.0.0.nupkg
    info :     OK https://api.nuget.org/v3-flatcontainer/system.text.json/7.0.0/system.text.json.7.0.0.nupkg 22ms
    info :   GET https://api.nuget.org/v3-flatcontainer/system.text.encodings.web/index.json
    info :     OK https://api.nuget.org/v3-flatcontainer/system.text.encodings.web/index.json 788ms
    info :   GET https://api.nuget.org/v3-flatcontainer/system.text.encodings.web/7.0.0/system.text.encodings.web.7.0.0.nupkg
    info :     OK https://api.nuget.org/v3-flatcontainer/system.text.encodings.web/7.0.0/system.text.encodings.web.7.0.0.nupkg 23ms
    info : Installed System.Text.Encodings.Web 7.0.0 from https://api.nuget.org/v3/index.json with content hash 0P6umVGxc0Z0MvZQBVigj4/U31Pw72ITihDWP9WiWDm+q5aoe0GaJivsfYgq53o6dxH7DcXWiCTL7+o02C6dmq=.
    info : Installed System.Net.Http.Json 7.0.0 from https://api.nuget.org/v3/index.json with content hash K74te5euBJLzsprZji9juSop5mLWRzlpSNqBqZBE7/2AiPbFtM7BSeDB/tmB1jjil3bNNeJYBNfIxobs6uXh0==.
    info : Installed System.Text.Json 7.0.0 from https://api.nuget.org/v3/index.json with content hash DaGSsVqKsn/ia6RG8frjwmJonfos0srquhw09TlT8KRw5i43E+4gs+/bZj4K0vShj5H9imCuXupb4RmS+dBy3w==.
    info : Package 'NUnit' is compatible with all the specified frameworks in project '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
    info : PackageReference for package 'NUnit' version '3.13.3' updated in file '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
    info : Generating MSBuild file /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/obj/MiNotSoSimpleAppTests.csproj.nuget.g.targets.
    info : Writing assets file to disk. Path: /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/obj/project.assets.json
    log : Restored /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj (in 4.13 sec).
```

```

    Writing /var/folders/41/kx949byd2hb4x2s0hrqv565m000gn/T/tmpF1F8Pf.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7
.0.400/trustedroots/codesignctl.pem'.
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7
.0.400/trustedroots/timestampctl.pem'.
info : Adding PackageReference for package 'Moq' into project '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/T
p-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : CACHE https://api.nuget.org/v3/registration5-gz-semver2/moq/index.json
info : Restoring packages for /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSo
SimpleAppTests.csproj...
info : Package 'Moq' is compatible with all the specified frameworks in project '/Users/agostinamorellato/Desktop/In
gSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : PackageReference for package 'Moq' version '4.20.69' added to file '/Users/agostinamorellato/Desktop/IngSwIII
/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : Writing assets file to disk. Path: /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppT
ests/obj/project.assets.json
log : Restored /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests
.csproj (in 77 ms).
● agostinamorellato@Agostinas-MacBook-Pro MiNotSoSimpleAppTests % dotnet add package NUnit3TestAdapter
  Determining projects to restore...
  Writing /var/folders/41/kx949byd2hb4x2s0hrqv565m000gn/T/tmp7xdVft.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7
.0.400/trustedroots/codesignctl.pem'.
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7
.0.400/trustedroots/timestampctl.pem'.
info : Adding PackageReference for package 'NUnit3TestAdapter' into project '/Users/agostinamorellato/Desktop/IngSwI
II/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : CACHE https://api.nuget.org/v3/registration5-gz-semver2/nunit3testadapter/index.json
info : Restoring packages for /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSo
SimpleAppTests.csproj...
info : Package 'NUnit3TestAdapter' is compatible with all the specified frameworks in project '/Users/agostinamorell
ato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : PackageReference for package 'NUnit3TestAdapter' version '4.5.0' updated in file '/Users/agostinamorellato/De
sktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : Generating MSBuild file /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/obj/Mi
NotSoSimpleAppTests.csproj.nuget.g.props.
info : Writing assets file to disk. Path: /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppT
ests/obj/project.assets.json
log : Restored /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests
.csproj (in 77 ms).
● agostinamorellato@Agostinas-MacBook-Pro MiNotSoSimpleAppTests % dotnet add package Microsoft.Extensions.Http
  Determining projects to restore...

```

```

● agostinamorellato@Agostinas-MacBook-Pro MiNotSoSimpleAppTests % dotnet add package Microsoft.Extensions.Http
  Determining projects to restore...
  Writing /var/folders/41/kx949byd2hb4x2s0hrqv565m000gn/T/tmpQ1BLdw.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7
.0.400/trustedroots/codesignctl.pem'.
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7
.0.400/trustedroots/timestampctl.pem'.
info : Adding PackageReference for package 'Microsoft.Extensions.Http' into project '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : CACHE https://api.nuget.org/v3/registration5-gz-semver2/microsoft.extensions.http/index.json
info : Restoring packages for /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSo
SimpleAppTests.csproj...
info : Package 'Microsoft.Extensions.Http' is compatible with all the specified frameworks in project '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : PackageReference for package 'Microsoft.Extensions.Http' version '7.0.0' added to file '/Users/agostinamorell
ato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : Writing assets file to disk. Path: /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppT
ests/obj/project.assets.json
log : Restored /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests
.csproj (in 79 ms).
● agostinamorellato@Agostinas-MacBook-Pro MiNotSoSimpleAppTests % dotnet add package Microsoft.Extensions.DependencyInjection
  Determining projects to restore...
  Writing /var/folders/41/kx949byd2hb4x2s0hrqv565m000gn/T/tmpRoseX4.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7
.0.400/trustedroots/codesignctl.pem'.
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/local/share/dotnet/sdk/7
.0.400/trustedroots/timestampctl.pem'.
info : Adding PackageReference for package 'Microsoft.Extensions.DependencyInjection' into project '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.extensions.dependencyinjection/index.json
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.extensions.dependencyinjection/index.json 91
1ms
info : Restoring packages for /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSo
SimpleAppTests.csproj...
info : Package 'Microsoft.Extensions.DependencyInjection' is compatible with all the specified frameworks in project
 '/Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : PackageReference for package 'Microsoft.Extensions.DependencyInjection' version '7.0.0' added to file '/Users
/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj'.
info : Writing assets file to disk. Path: /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppT
ests/obj/project.assets.json
log : Restored /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/MiNotSoSimpleAppTests
.csproj (in 77 ms).

```

8.2 Escribimos Test:

Reemplazamos el código de UnitTest1.cs por:

The screenshot shows the Visual Studio Code interface with the following details:

- TEST EXPLORER**: Configure a test framework to see your tests here. Options include "Configure Python Tests" and "Install Additional Test Extensions...".
- .NET TEST EXPLORER**: Shows a test named "GetMyModelsAsync_ReturnsDataFr...".
- Unit test code (UnitTest1.cs):**

```
14 public class ApiServiceTests
15 {
16     [Test]
17     public async Task GetMyModelsAsync_ReturnsDataFrom...
18     {
19         // Arrange
20         var serviceCollection = new ServiceCollection();
21         var mockResponse = new HttpResponseMessage(HttpStatusCode.OK);
22         {
23             Content = new StringContent("{\"UserId\": \"123\"}");
24         };
25
26         var mockHttpMessageHandler = new Mock<HttpMessageHandler>()
27             .Setup<Task<HttpResponseMessage>>("SendAsync")
28             .Returns<...
29             (local variable) ServiceCollection serviceCollection;
30             serviceCollection.AddTransient<IApiService>(_...
31             serviceProvider = serviceCollection.BuildServiceProvider();
32
33             var apiService = serviceProvider.GetRequiredService<...
34
35             // Act
36             var result = await apiService.GetMyModelsAsync();
37
38         }
39     }
40 }
```
- TERMINAL**: Shows the command "dotnet test" being run.

Ejecutamos Test en VSCode y en linea de consola.

```
● agostinamorellato@Agostinas-MacBook-Pro Tp-9 % cd MiNotSoSimpleAppTests
● agostinamorellato@Agostinas-MacBook-Pro MiNotSoSimpleAppTests % dotnet test
  Determining projects to restore...
  All projects are up-to-date for restore.
  MiNotSoSimpleApp -> /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleApp/bin/Debug/net7.0/MiNotSoSimpleApp.dll
  MiNotSoSimpleAppTests -> /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/bin/Debug/net7.0/MiNotSoSimpleAppTests.dll
  Test run for /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/bin/Debug/net7.0/MiNotSoSimpleAppTests.dll (.NETCoreApp, Version=v7.0)
  Microsoft (R) Test Execution Command Line Tool Version 17.7.0-preview-23364-03+bc17bb9693fcf4778ded51aa0ab7f1065433
  989 (arm64)
  Copyright (c) Microsoft Corporation. All rights reserved.

  Starting test execution, please wait...
  A total of 1 test files matched the specified pattern.

  Passed! - Failed:    0, Passed:    1, Skipped:    0, Total:    1, Duration: 95 ms - MiNotSoSimpleAppTests.dll
  net7.0
● agostinamorellato@Agostinas-MacBook-Pro MiNotSoSimpleAppTests %
```

Hacerlo fallar, arreglarlo y volverlo a correr

The screenshot shows a Visual Studio Code interface with the following components:

- .NET TEST EXPLORER**: Shows a single test case named "GetMyModelsAsync_ReturnsDataFr..." with a green checkmark.
- CODE EDITOR**: Displays C# code for a unit test. The code uses Moq to mock an HttpResponseMessage and an IApiService. It then adds the service to a service collection, builds a ServiceProvider, and gets the required service. Finally, it acts by calling GetMyModelsAsync, asserts that the result is not null, has a count of 1, and contains the expected title.
- TERMINAL**: Shows the command-line output of the test run. It includes the path to the test assembly, the test run configuration, the Microsoft Test Execution Command Line Tool version, and the copyright information. It then starts the test execution, indicating 1 test file matched the pattern. The output ends with a summary: "Passed! - Failed: 0, Passed: 1, Skipped: 0, Total: 1, Duration: 88 ms - MiNotSoSimpleAppTests.dll (net7.0)".

```
25     var mockHttpMessageHandler = new Mock<HttpMessageHandler>()
26     mockHttpMessageHandler.Protected()
27         .Setup<Task<HttpResponseMessage>>("SendAsync")
28             .ReturnsAsync(mockResponse);
29
30     serviceCollection.AddTransient<IApiService>(_ : string, serviceProvider : ServiceCollection) =>
31         serviceProvider.AddSingleton(mockHttpMessageHandler.Object);
32
33     var apiService = serviceProvider.GetRequiredService<IApiService>();
34
35     // Act
36     var result = await apiService.GetMyModelsAsync();
37
38     // Assert
39     Assert.IsNotNull(result);
40     Assert.AreEqual(1, result.Count());
41     Assert.AreEqual("Test Title", result.FirstOrDefault().Title);
42
43 }
44
45 }
```

```
MiNotSoSimpleAppTests -> /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/bin/Debug/net7.0/MiNotSoSimpleAppTests.dll
Test run for /Users/agostinamorellato/Desktop/IngSwIII/Practicos/Tp-9/MiNotSoSimpleAppTests/bin/Debug/net7.0/MiNotSoSimpleAppTests.dll (.NETCoreApp, Version=v7.0)
Microsoft (R) Test Execution Command Line Tool Version 17.7.0-preview-23364-03+bc17bb9693fc4778ded51aa0ab7f1065433f
989 (arm64)
Copyright (c) Microsoft Corporation. All rights reserved.

Starting test execution, please wait...
A total of 1 test files matched the specified pattern.

Passed! - Failed: 0, Passed: 1, Skipped: 0, Total: 1, Duration: 88 ms - MiNotSoSimpleAppTests.dll (net7.0)
o agostinamorellato@Agostinas-MacBook-Pro MiNotSoSimpleAppTests %
```