

# Trabajo Práctico N°9

## Ejercicio 5

```
Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9
$ git clone https://github.com/ingsoft3ucc/MiSimpleApp.git
Cloning into 'MiSimpleApp'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9
$ cd MiSimpleApp

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9/MiSimpleApp (main)
$ code .

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9/MiSimpleApp (main)
$ cd ..

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9
$ dotnet new nunit -n MiSimpleAppTests
La plantilla "NUnit 3 Test Project" se creó correctamente.

Procesando acciones posteriores a la creación...
Restaurando C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleAppTests\MiSimpleAppTests.csproj:
  Determinando los proyectos que se van a restaurar...
  Se ha restaurado C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleAppTests\MiSimpleAppTests.csproj (en
  26.18 sec).
```

```
Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9/MiSimpleAppTests
$ dotnet add package NUnit.ConsoleRunner
  Determinando los proyectos que se van a restaurar...
  Writing C:\Users\Usuario\AppData\Local\Temp\tmpACA3.tmp
info : X.509 certificate chain validation will use the default trust store selected by .NET for code signing.
info : X.509 certificate chain validation will use the default trust store selected by .NET for timestamping.
info : Agregando PackageReference para el paquete "NUnit.ConsoleRunner" al proyecto "C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\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 1059 ms
info : Restaurando paquetes para C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\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 866 ms
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 792 ms
info : Se instaló NUnit.ConsoleRunner 3.16.3 de https://api.nuget.org/v3/index.json con el hash de contenido bhc4f80VdS70xLau1Pu2dMuUMK6hLj5fWlWmUnvbxvEA9w41sZNe6Xb3vPkcp+5
OrIggmtcRbNx+oShoz62Jg==.
info : El paquete "NUnit.ConsoleRunner" es compatible con todos los marcos de trabajo especificados del proyecto "C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleAppTests\MiSimpleAppTests.csproj".
info : Se agregó PackageReference para la versión "3.16.3" del paquete "NUnit.ConsoleRunner" al archivo "C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleAppTests\MiSimpleAppTests.csproj".
info : Generación de archivo MSBuild C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleAppTests\obj\MiSimpleAppTests.csproj.nuget.g.props.
info : Escribiendo el archivo de recursos en el disco. Ruta de acceso: C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleAppTests\obj\project.assets.json
log : Se ha restaurado C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleAppTests\MiSimpleAppTests.csproj (en 3.82 sec).

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9/MiSimpleAppTests
$ dotnet add reference ../MiSimpleApp/MiSimpleApp.csproj
Se ha agregado la referencia "../MiSimpleApp/MiSimpleApp.csproj" al proyecto.
```

✓ Test1 ▶	MiSimpleAppTests > C# UnitTest1.cs > Tests > Test1
	<pre> 1 namespace MiSimpleAppTests; 2 3 public class Tests 4 { 5     [SetUp] 6     public void Setup() 7     { 8     } 9 10    [Test] 11    public void Test1() 12    { 13        Assert.Pass(); 14    } 15 }</pre>

```

2 references
public class Reservation
{
    1 reference
    public User MadeBy { get; set; }

    1 reference
    public bool CanBeCancelledBy(User user)
    {
        if (user.IsAdmin)
        {
            return true;
        }
        if (MadeBy==user)
        {
            return true;
        }
        return false;

        //return (user.IsAdmin || MadeBy == user);
    }
}

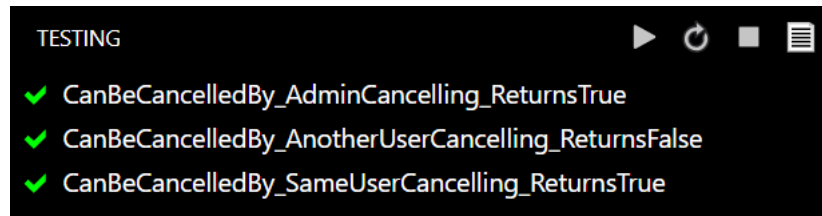
4 references
public class User
{
    2 references
    public bool IsAdmin { get; set; }
}
```

```

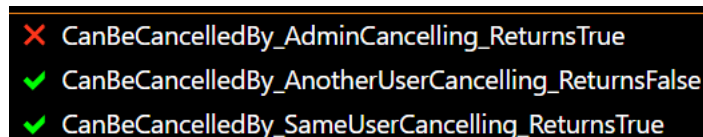
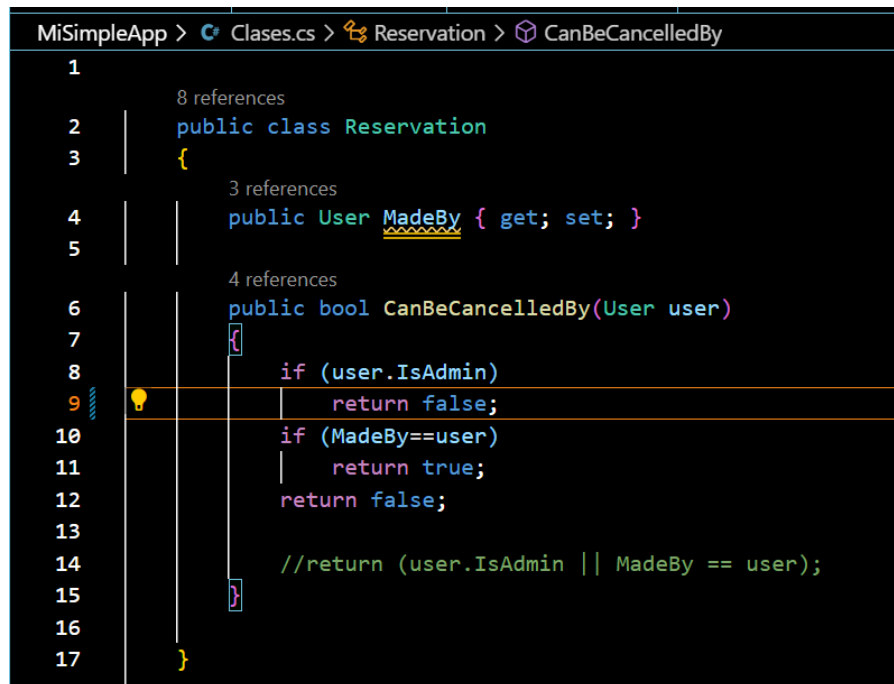
Starting test discovery
Registrando diagnóstico de TestHost en el archivo:
c:\Users\Usuario\AppData\Roaming\Code\logs\20231003T143001\window1\exthost\ms-dotnettools.csharp\testLogs\vsTestLogs.host.
23-10-03_14-42-24_63514_8.txt
NUnit Adapter 4.4.0.0: Test discovery starting
NUnit Adapter 4.4.0.0: Test discovery complete
Found 3 tests in 3s

Starting test run
[Passed] CanBeCancelledBy_AdminCancelling_ReturnsTrue
[Passed] CanBeCancelledBy_AnotherUserCancelling_ReturnsFalse
[Passed] CanBeCancelledBy_SameUserCancelling_ReturnsTrue
==== Summary ====
Passed! - Failed:    0, Passed:    3, Skipped:    0, Total:    3, Duration: 512ms

```



Nota: se mantiene la convención de nombres.



```

[Passed] CanBeCancelledBy_AnotherUserCancelling_ReturnsFalse
[Passed] CanBeCancelledBy_SameUserCancelling_ReturnsTrue
==== Summary ====
Failed! - Failed:    1, Passed:    2, Skipped:    0, Total:    3, Duration: 644ms

```

```

8 references
public class Reservation
{
    3 references
    public User MadeBy { get; set; }

    4 references
    public bool CanBeCancelledBy(User user)
    {
        /*
        if (user.IsAdmin)
            return true;
        if (MadeBy==user)
            return true;
        return false;
        */

        return (user.IsAdmin || MadeBy == user);
    }
}

```

```

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9
$ cd MiSimpleAppTests

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9/MiSimpleAppTests
$ dotnet test
Determinando los proyectos que se van a restaurar...
Se ha restaurado C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleApp\MiSimpleApp.csproj (en 254 ms).
Se ha restaurado C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleAppTests\MiSimpleAppTests.csproj (en 508 ms).
C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleApp\Clases.cs(4,21): warning CS8618: El elemento propiedad "MadeBy" que no acepta valores NULL debe contener
un valor distinto de NULL al salir del constructor. Considere la posibilidad de declarar el elemento propiedad como que admite un valor NULL. [C:\Users\Usuario\OneDrive\E
scritorio\2023\Ing SW 3\tp9\MiSimpleApp\MiSimpleApp.csproj]
MiSimpleApp -> C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleApp\bin\Debug\net7.0\MiSimpleApp.dll
MiSimpleAppTests -> C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleAppTests\bin\Debug\net7.0\MiSimpleAppTests.dll
Serie de pruebas para C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\MiSimpleAppTests\bin\Debug\net7.0\MiSimpleAppTests.dll (.NETCoreApp,Version=v7.0)
Herramienta de línea de comandos de ejecución de pruebas de Microsoft(R), versión 17.7.0-preview-23364-03+bc17bb9693cfc4778ded51aa8ab7f1065433f989 (x64)
Copyright (c) Microsoft Corporation. Todos los derechos reservados.

Iniciando la ejecución de pruebas, espere...
1 archivos de prueba en total coincidieron con el patrón especificado.

Conectas! - Con error: 0, Superado: 3, Omitido: 0, Total: 3, Duración: 40 ms - MiSimpleAppTests.dll (net7.0)

```

## Ejercicio 6

```

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9
$ git clone https://github.com/ingsoft3ucc/SimpleWebAPI.git
Cloning into 'SimpleWebAPI'...
remote: Enumerating objects: 150, done.
remote: Counting objects: 100% (150/150), done.
Receiving objects: 100% (150/150), 28.41 K
Resolving deltas: 100% (60/60), done./115) esolving deltas: 0% (0/60)
remote: Compressing objects: 100% (115/115), done.
remote: Total 150 (delta 60), reused 33 (delta 9), pack-reused 0

```

```

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9
$ dotnet new nunit -n SimpleWebAPITests
La plantilla "NUnit 3 Test Project" se creó correctamente.

Procesando acciones posteriores a la creación...
Restaurando C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\SimpleWebAPITests\SimpleWebAPITests.csproj:
Determinando los proyectos que se van a restaurar...
Se ha restaurado C:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9\SimpleWebAPITests\SimpleWebAPITests.csproj (e
n 825 ms).
Restauración realizada correctamente.

```

```
Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/tp9/SimpleWebAPITests
$ dotnet add reference ../SimpleWebAPI/SimpleWebAPI/SimpleWebAPI.csproj
Se ha agregado la referencia "..\SimpleWebAPI\SimpleWebAPI\SimpleWebAPI.csproj" al proyecto.
```

```
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());
        }
    }
}
```

Arrange: Se prepara el entorno, creando una instancia del controlador WeatherForecastController y un logger para el mismo.

Act: Se realiza la acción a probar, en este caso, llama al método Get() del controlador.

Assert: Se verifica que el resultado (result) no sea nulo y que el número de elementos devueltos sea 5.

```
Iniciando la ejecución de pruebas, espere...
1 archivos de prueba en total coincidieron con el patrón especificado.

Correctas! - Con error:    0, Superado:    1, Omitido:    0, Total:    1, Duración: 114 ms - SimpleWebAPITests.dll (net7.0)
```

---

## Ejercicio 8

```

Id: 70, Title: voluptatem laborum magni
Id: 71, Title: et iusto veniam et illum aut fuga
Id: 72, Title: sint hic doloribus consequatur eos non id
Id: 73, Title: consequuntur deleniti eos quia temporibus ab aliquid at
Id: 74, Title: enim unde ratione doloribus quas enim ut sit sapiente
Id: 75, Title: dignissimos eum dolor ut enim et delectus in
Id: 76, Title: doloremque officiis ad et non perferendis
Id: 77, Title: necessitatibus quasi exercitationem odio
Id: 78, Title: quam voluptatibus rerum veritatis
Id: 79, Title: pariatur consequatur quia magnam autem omnis non amet
Id: 80, Title: labore in ex et explicabo corporis aut quas
Id: 81, Title: tempora rem veritatis voluptas quo dolores vero
Id: 82, Title: laudantium voluptate suscipit sunt enim enim
Id: 83, Title: odit et voluptates doloribus alias odio et
Id: 84, Title: optio ipsam molestias necessitatibus occaecati facilis veritatis dolores aut
Id: 85, Title: dolore veritatis porro provident adipisci blanditiis et sunt
Id: 86, Title: placeat quia et porro iste
Id: 87, Title: nostrum quis quasi placeat
Id: 88, Title: sapiente omnis fugit eos
Id: 89, Title: sint soluta et vel magnam aut ut sed qui
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

```

Esta aplicación hace uso de una interfaz `IApiService` y una clase `ApiService` para realizar una solicitud HTTP a una API externa y recuperar una lista de objetos `Post`.

Define una clase llamada `Post` que tiene propiedades `UserId`, `Id`, `Title`, y `Body`.

- Se configura un servicio utilizando `ServiceCollection`.
- Se agrega un cliente HTTP.
- Se llama al método `GetMyModelsAsync` para obtener una lista de modelos de una API externa (<https://jsonplaceholder.typicode.com/posts/>).
- Se itera sobre los modelos y se imprime el `Id` y el `Título` de cada uno.

Al utilizar `Moq`, se hace una implementación simulada de `IApiService` que se utiliza en lugar de la implementación real durante las pruebas. Esto permite simular el comportamiento del servicio y controlar las respuestas que debe devolver sin depender de una API externa real.

```

Starting test discovery
Registrando diagnóstico de TestHost en el archivo:
c:\Users\Usuario\AppData\Roaming\Code\logs\20231016T170629\window1\exthost\ms-dotnettools.csharp\testLogs\vsTestLogs.
host.23-10-16_17-08-02_85535_7.txt
NUnit Adapter 4.5.0.0: Test discovery starting
NUnit Adapter 4.5.0.0: Test discovery complete
Found 1 tests in 2s

Starting test run
[Passed] GetMyModelsAsync_ReturnsDataFromHttpClient
==== Summary ====
Passed! - Failed: 0, Passed: 1, Skipped: 0, Total: 1, Duration: 1s

```

```

using System;
using System.Collections.Generic;
using System.Net;
using System.Net.Http;

```

```

using System.Net.Http.Json;
using System.Threading.Tasks;
using Microsoft.Extensions.DependencyInjection;
using Moq;
using NUnit.Framework;
using Moq.Protected;

namespace MiNotSoSimpleAppTests
{
    public class ApiServiceTests
    {
        [Test]
        public async Task GetMyModelsAsync_ReturnsDataFromHttpClient()
        {
            // Arrange
            var serviceCollection = new ServiceCollection();
            var mockResponse = new HttpResponseMessage(HttpStatusCode.OK)
            {
                Content = new StringContent("[{ \"UserId\": 1, \"Id\": 1, \"Title\": \"Test Title\", \"Body\": \"Test Body\" }]")
            };

            var mockHttpMessageHandler = new Mock<HttpMessageHandler>();
            mockHttpMessageHandler.Protected()
                .Setup<Task<HttpResponseMessage>>("SendAsync", ItExpr.IsAny<HttpRequestMessage>(), ItExpr.IsAny<CancellationToken>())
                .ReturnsAsync(mockResponse);

            serviceCollection.AddTransient<IApiService>(_ => new ApiService(new HttpClient(mockHttpMessageHandler.Object)));
            var serviceProvider = serviceCollection.BuildServiceProvider();

            var apiService = serviceProvider.GetRequiredService<IApiService>();

            // Act
            var result = await apiService.GetMyModelsAsync();

            // Assert
            Assert.IsNotNull(result);
            Assert.AreEqual(1, result.Count());
            Assert.AreEqual("Test Title", result.FirstOrDefault().Title);
        }
    }
}

```

Este test verifica si el método `GetMyModelsAsync` de la clase `ApiService` funciona correctamente. Se obtiene una respuesta que contiene un objeto `Post` con el título "Test Title". El mock del manejador de mensajes HTTP simula la respuesta de la API externa.

- Crea una nueva colección de servicios.
- `var mockResponse = new HttpResponseMessage(HttpStatusCode.OK)` crea una nueva respuesta HTTP simulada con un código de estado 200 (OK).
- `Content = new StringContent("[{ \"UserId\": 1, \"Id\": 1, \"Title\": \"Test Title\", \"Body\": \"Test Body\" }])` establece el contenido de la respuesta, JSON que representa una colección de objetos `Post`.
- Se establece que cuando se llame al método `SendAsync` (que es el método que envía la solicitud HTTP), debe devolver la `mockResponse` que se creó anteriormente.
- `serviceCollection.AddTransient<IApiService>(_ => new ApiService(new HttpClient(mockHttpMessageHandler.Object)))` agrega un servicio de tipo `IApiService` a la colección de servicios. Este servicio utiliza un objeto `ApiService`, pero en lugar de usar un `HttpClient` real, se le pasa el mock del manejador de mensajes HTTP.
- Llama al método `GetMyModelsAsync` del `IApiService`. En este caso, la llamada se redirige al mock del manejador de mensajes HTTP que devuelve la respuesta simulada.
- `Assert.IsNotNull(result);`
  - Asegura que el resultado no sea nulo. Esto verifica que se haya recibido una respuesta de la API (incluso si la respuesta está vacía).
- `Assert.AreEqual(1, result.Count());`
  - Verifica que la cantidad de elementos en la lista de resultados sea 1. Esto confirma que se recibió un solo objeto `Post`.
- `Assert.AreEqual("Test Title", result.FirstOrDefault().Title);`
  - Verifica que el título del primer objeto `Post` en la lista sea "Test Title".

### Hacerlo fallar, arreglarlo y volverlo a correr

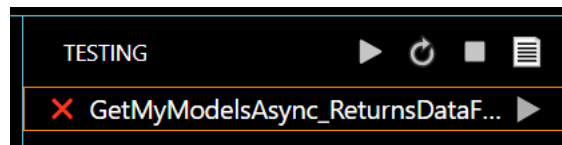
```
var mockResponse = new HttpResponseMessage(HttpStatusCode.OK)
{
    Content = new StringContent("[{ \"UserId\": 1, \"Id\": 1, \"Title\": \"Test Title 1\", \"Body\": \"Test Body 1\" }, { \"UserId\": 2,
```

Para que devuelva mas de un objeto.

```
Starting test run
[Failed] GetMyModelsAsync_ReturnsDataFromHttpClient
    Message:
        Expected: 1
        But was: 2
    Stack Trace:
        at MiNotSoSimpleAppTests.ApiServiceTests.GetMyModelsAsync_ReturnsDataFromHttpClient() in
        c:\Users\Usuario\OneDrive\Escritorio\2023\Ing SW 3\tp9-ej8\MiNotSoSimpleAppTests\UnitTest1.cs:line 42
        at NUnit.Framework.Internal.TaskAwaitAdapter.GenericAdapter`1.GetResult()
        at NUnit.Framework.Internal.AsyncToSyncAdapter.Await(Func`1 invoke)
        at NUnit.Framework.Internal.Commands.TestMethodCommand.RunTestMethod(TestExecutionContext context)
        at NUnit.Framework.Internal.Commands.TestMethodCommand.Execute(TestExecutionContext context)
        at NUnit.Framework.Internal.Execution.SimpleWorkItem.<>c__DisplayClass4_0.<PerformWork>b__0()
        at NUnit.Framework.Internal.ContextUtils.<>c__DisplayClass1_0`1.<DoIsolated>b__0(Object _)
```

==== Summary ====

Failed! - Failed: 1, Passed: 0, Skipped: 0, Total: 1, Duration: 912ms



Luego:

```
var mockResponse = new HttpResponseMessage(HttpStatusCode.OK)
{
    Content = new StringContent("[{ \"UserId\": 1, \"Id\": 1, \"Title\": \"Test Title\", \"Body\": \"Test Body\" } ]")
};
```

```
Starting test run
[Passed] GetMyModelsAsync_ReturnsDataFromHttpClient
==== Summary ====
Passed! - Failed: 0, Passed: 1, Skipped: 0, Total: 1, Duration: 783ms
```

### Modificar el mock para que devuelva una colección de Posts y en un nuevo test verificar que la cantidad devuelta por el mock coincida con la esperada en el nuevo test.

```
/*var mockResponse = new HttpResponseMessage(HttpStatusCode.OK)
{
    Content = new StringContent("[{ \"UserId\": 1, \"Id\": 1, \"Title\": \"Test Title\", \"Body\": \"Test Body 1\" },
{ \"UserId\": 2, \"Id\": 2, \"Title\": \"Test Title\", \"Body\": \"Test Body 2\" },
{ \"UserId\": 3, \"Id\": 3, \"Title\": \"Test Title\", \"Body\": \"Test Body 3\" },
{ \"UserId\": 4, \"Id\": 4, \"Title\": \"Test Title\", \"Body\": \"Test Body 4\" } ]")
};*/
```



```
// Assert
Assert.IsNotNull(result);
Assert.AreEqual(4, result.Count());
Assert.AreEqual("Test Title", result.FirstOrDefault().Title);
```

```
Starting test run
[Passed] GetMyModelsAsync_ReturnsDataFromHttpClient
==== Summary ====
Passed! - Failed:    0, Passed:    1, Skipped:    0, Total:    1, Duration: 752ms
```

## Ejercicio 9

Archivo del .yaml del workflow:

```
name: CI/CD Pipeline

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout Repo
        uses: actions/checkout@v2

      - name: Verificar versión de .NET Core
        run: dotnet --version

      - name: Setup .NET Core
        uses: actions/setup-dotnet@v1
        with:
          dotnet-version: '7.0'

      - name: Restore Dependencies
        run: dotnet restore MiNotSoSimpleApp/MiNotSoSimpleApp.csproj

      - name: Build
        run: dotnet build MiNotSoSimpleApp/MiNotSoSimpleApp.csproj --configuration Release

      - name: Run Tests
        run: dotnet test MiNotSoSimpleAppTests/MiNotSoSimpleAppTests.csproj --configuration Debug --no-build --verbosity normal

  deploy:
    runs-on: ubuntu-latest
    needs: build

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Build the Docker image
        run: docker build . --file Dockerfile --tag ${ secrets.DOCKERHUB_USERNAME }}/simple-web-api-gh:latest

      - name: Log in to Docker Hub
        run: docker login -u ${ secrets.DOCKERHUB_USERNAME } -p ${ secrets.DOCKERHUB_PASSWORD }

      - name: Push Docker image to Docker Hub
        run: |
          docker push ${ secrets.DOCKERHUB_USERNAME }}/simple-web-api-gh:latest

      - name: Clean up
        run: docker logout
        if: always()
```

- **on: push** : Indica que este flujo de trabajo se ejecutará cuando se haga un "push" al repositorio.
  - **branches: - main** : Especifica que el flujo de trabajo se ejecutará solo cuando se haga un "push" a la rama "main".

- **jobs:**
  - **name: Checkout Repo:** Utiliza el comando **checkout** para clonar el repositorio.
  - **name: Setup .NET Core:** Configura el entorno de .NET Core.
  - **name: Restore Dependencies:** Restaura las dependencias del proyecto.
  - **name: Build:** Compila la aplicación.
  - **name: Run Tests:** Ejecuta las pruebas.
- **deploy:**
  - **name: Build the Docker image:** Construye una imagen Docker a partir del Dockerfile.
  - **name: Log in to Docker Hub:** Inicia sesión en Docker Hub.
  - **name: Push Docker image to Docker Hub:** Sube la imagen Docker a Docker Hub.
  - **name: Clean up:** Cierra la sesión en Docker Hub, independientemente de si el paso anterior tuvo éxito o no.

### Dockerfile:

```
FROM mcr.microsoft.com/dotnet/aspnet:7.0 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443
EXPOSE 5254

FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build
WORKDIR /src
COPY ["MiNotSoSimpleApp/MiNotSoSimpleApp.csproj", "MiNotSoSimpleApp/"]
RUN dotnet restore "MiNotSoSimpleApp/MiNotSoSimpleApp.csproj"
COPY . .
WORKDIR "/src/MiNotSoSimpleApp"
RUN dotnet build "MiNotSoSimpleApp.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "MiNotSoSimpleApp.csproj" -c Release -o /app/publish /p:UseAppHost=false

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
#ENTRYPOINT ["dotnet", "MiNotSoSimpleApp.dll"]
CMD ["/bin/bash"]
```

23 workflow runs			Event ▾	Status ▾	Branch ▾	Actor ▾
✓	Update main.yml	CI/CD Pipeline #23: Commit f16cb66 pushed by aaraya0	main		2 minutes ago 1m 20s	...
✗	dockerfileeeeeee	CI/CD Pipeline #22: Commit 087ee71 pushed by aaraya0	main		2 minutes ago 1m 3s	...
✗	Update main.yml	CI/CD Pipeline #21: Commit b22e915 pushed by aaraya0	main		8 minutes ago 49s	...
✗	AAA	CI/CD Pipeline #20: Commit 24f0b27 pushed by aaraya0	main		10 minutes ago 49s	...
✗	Update Dockerfile	CI/CD Pipeline #19: Commit c616415 pushed by aaraya0	main		16 minutes ago 1m 0s	...
✗	Update main.yml	CI/CD Pipeline #18: Commit ff91b52 pushed by aaraya0	main		35 minutes ago 55s	...
⚠	Update main.yml	CI/CD Pipeline #17: Commit dc12ecb pushed by aaraya0	main		36 minutes ago 48s	...

Los errores encontrados durante los primeros workflow runs estaban relacionados con problemas de versionado y de definición de rutas. A continuación, se muestra en más detalle la ejecución correcta del flujo de trabajo:

build			Search logs	↺	⚙
succeeded 5 minutes ago in 30s					
>	✓	Set up job			1s
>	✓	Checkout Repo			1s
>	✓	Verificar versión de .NET Core			3s
>	✓	Setup .NET Core			11s
>	✓	Restore Dependencies			3s
>	✓	Build			4s
>	✓	Run Tests			1s
>	✓	Post Checkout Repo			0s
>	✓	Complete job			0s

```
18
19 Test Run Successful.
20 Total tests: 1
21     Passed: 1
22 Total time: 1.0386 Seconds
23
24 Build succeeded.
25     0 Warning(s)
26     0 Error(s)
27
28 Time Elapsed 00:00:02.09
```

### deploy


succeeded 4 minutes ago in 36s


Search logs

- |                                     |     |
|-------------------------------------|-----|
| > ✓ Set up job                      | 1s  |
| > ✓ Checkout code                   | 1s  |
| > ✓ Build the Docker image          | 23s |
| > ✓ Log in to Docker Hub            | 0s  |
| > ✓ Push Docker image to Docker Hub | 8s  |
| > ✓ Clean up                        | 0s  |
| > ✓ Post Checkout code              | 0s  |
| > ✓ Complete job                    | 0s  |

## aaraya0 / simple-web-api-gh

### Description

This repository does not have a description 

 Last pushed: 6 minutes ago

**Link al repositorio: <https://github.com/aaraya0/tp9-ej8>**