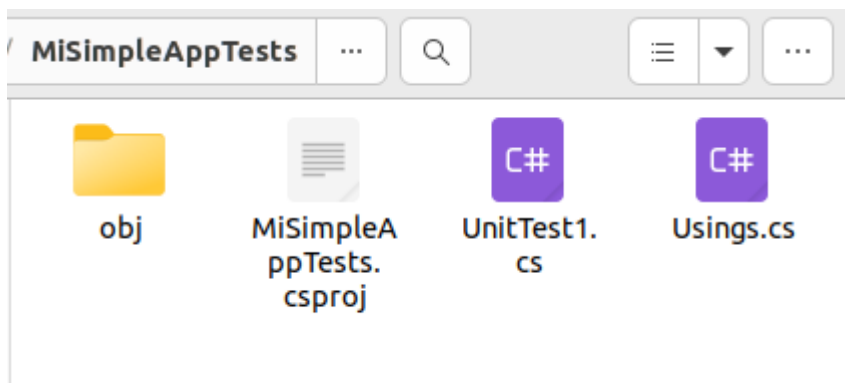
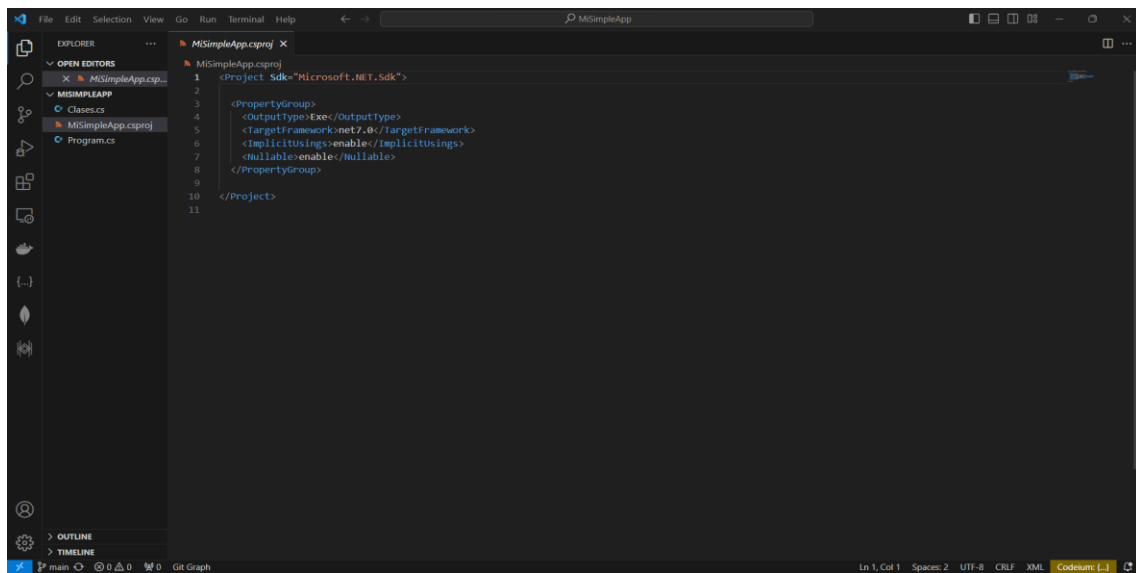


5)

```
C:\Users\gusta>code --version
1.83.1
f1b07bd25dfad64b0167beb15359ae573aecd2cc
x64
```

```
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3> 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.
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3> cd MiSimpleApp
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\MiSimpleApp> code .
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\MiSimpleApp>
```



TESTING...

✓ Test1

UnitTest1.cs X

MiSimpleAppTests > UnitTest1.cs > Tests > Test1

```
1 namespace MiSimpleAppTests;
2
3 0 references
4 public class Tests
5 {
6     [SetUp]
7     0 references
8     public void Setup()
9     {
10     }
11
12     [Test]
13     ✓ | 0 references
14     public void Test1()
15     {
16         Assert.Pass();
17     }
18 }
```

TESTING...

✓ Test1

UnitTest1.cs X

MiSimpleAppTests > UnitTest1.cs > Tests > Test1

```
1 namespace MiSimpleAppTests;
2
3 0 references
4 public class Tests
5 {
6     [SetUp]
7     0 references
8     public void Setup()
9     {
10     }
11
12     [Test]
13     ✓ | 0 references
14     public void Test1()
15     {
16         Assert.Pass();
17     }
18 }
```

```
Classes.cs 1
MiSimpleApp > Classes.cs > ...
1
2      2 references
3      public class Reservation
4      {
5          1 reference
6          public User MadeBy { get; set; }
7
8          1 reference
9          public bool CanBeCancelledBy(User user)
10         {
11             if (user.IsAdmin)
12                 return true;
13             if (MadeBy==user)
14                 return true;
15             return false;
16         }
17     }
18
19     4 references
20     public class User
21     {
22         2 references
23         public bool IsAdmin { get; set; }
24     }
```

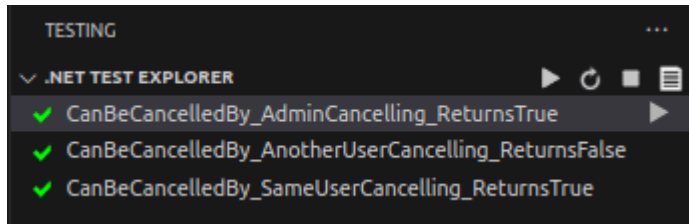
```
Program.cs X
MiSimpleApp > Program.cs > ...
1 using System;
2
3 0 references
4 class Program
5 {
6     0 references
7     static void Main()
8     {
9         Reservation reservation = new Reservation();
10        User user=new User();
11        user.IsAdmin=true;
12        bool result=reservation.CanBeCancelledBy(user);
13        Console.WriteLine(result);
14    }
15 }
```

```

UnitTest1.cs X
MiSimpleAppTests > UnitTest1.cs > Tests
1 namespace MiSimpleAppTests;
2
3 [TestFixture]
4 0 references
5 public class Tests
6 {
7     [SetUp]
8     0 references
9     public void Setup()
10    {
11    }
12
13    [Test]
14    0 references
15    public void CanBeCancelledBy_AdminCancelling_ReturnsTrue()
16    {
17        //Arrange
18        User user=new User();
19        user.IsAdmin=true;
20        Reservation reservation=new Reservation();
21
22        //Act
23        bool result=reservation.CanBeCancelledBy(user);
24
25        //Assert
26        //Assert.IsTrue(result);
27        Assert.That(result,Is.True);
28        //Assert.That(result==true);
29    }
30
31
32    [Test]
33    0 references
34    public void CanBeCancelledBy_SameUserCancelling_ReturnsTrue()
35    {
36        //Arrange
37        User user=new User();
38        Reservation reservation=new Reservation();
39        reservation.MadeBy=user;
40
41        //Act
42        bool result=reservation.CanBeCancelledBy(user);
43
44        //Assert
45        Assert.That(result,Is.True);
46    }
47
48
49    [Test]
50    0 references
51    public void CanBeCancelledBy_AnotherUserCancelling_ReturnsFalse()
52    {
53        //Arrange
54        User user=new User();
55        Reservation reservation=new Reservation();
56        reservation.MadeBy=user;
57
58        //Act
59        User newUser=new User();
60        bool result=reservation.CanBeCancelledBy(newUser);
61
62        //Assert
63        Assert.That(result,Is.False);
64    }
65
66 }

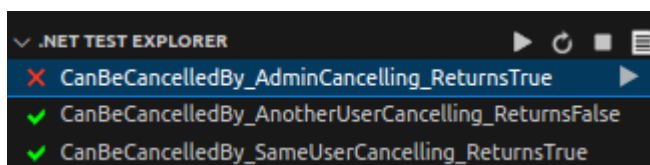
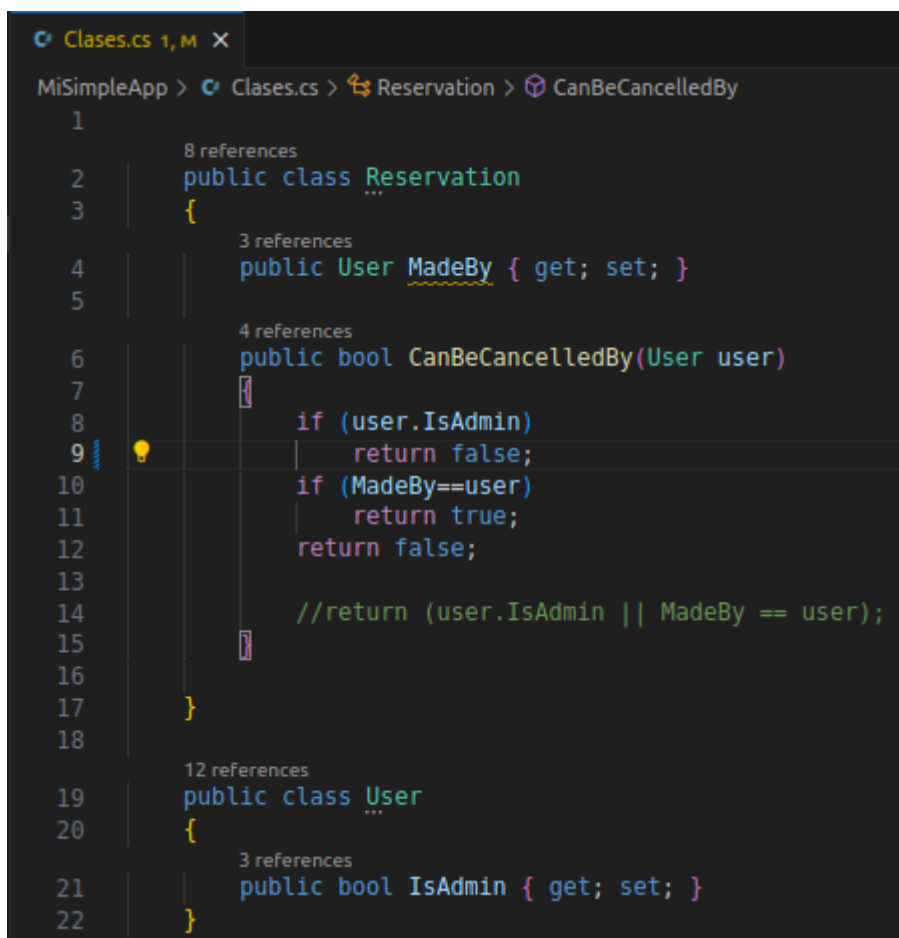
```

Estos tests comprueban el comportamiento de la función CanBeCancelledBy en la clase Reservation. El primero verifica si un administrador puede cancelar una reserva, el segundo verifica si el usuario que creó la reserva puede cancelarla y el tercero verifica que otro usuario no puede cancelar una reserva que no creó. Cada uno de estos tests sigue un patrón similar con las secciones Arrange para la preparación, Act para la acción y Assert para la verificación.



```
Starting test execution, please wait...
A total of 1 test files matched the specified pattern.
Results File: /tmp/test-explorer-kB750R/0.trx

Passed! - Failed:    0, Passed:    3, Skipped:    0, Total:    3, Duration: 9 ms -
MiSimpleAppTests.dll (net7.0)
```



```
Clases.cs 1 X
MiSimpleApp > Clases.cs > ...
1
2      8 references
3      public class Reservation
4      {
5          3 references
6          public User MadeBy { get; set; }
7
8          4 references
9          public bool CanBeCancelledBy(User user)
10         {
11             if (user.IsAdmin)
12                 return true;
13             if (MadeBy==user)
14                 return true;
15             return false;
16
17             //return (user.IsAdmin || MadeBy == user);
18         }
19
20     12 references
21     public class User
22     {
23         3 references
24         public bool IsAdmin { get; set; }
25     }
```

```
Clases.cs 1, M
MiSimpleApp > Clases.cs > ...
1
2   8 references
3   public class Reservation
4   {
5       3 references
6       public User MadeBy { get; set; }
7
8       4 references
9       public bool CanBeCancelledBy(User user)
10      {
11          /*
12          if (user.IsAdmin)
13              return true;
14          if (MadeBy==user)
15              return true;
16          return false;
17          */
18          return (user.IsAdmin || MadeBy == user);
19      }
20
21      12 references
22      public class User
23      {
24          3 references
25          public bool IsAdmin { get; set; }
26      }
```

```
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: 9 ms - MiSimpleAppT
ests.dll (net7.0)
```

6)

```

UnitTest1.cs x
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  {
9      [TestFixture]
10     // 0 references
11     public class WeatherForecastControllerTests
12     {
13         [Test]
14         // 0 references
15         public void Get_ReturnsWeatherForecasts()
16         {
17             // Arrange
18             ILogger<WeatherForecastController> logger = new LoggerFactory();
19             var controller = new WeatherForecastController(logger);
20
21             // Act
22             var result = controller.Get();
23
24             // Assert
25             Assert.NotNull(result);
26             Assert.AreEqual(5, result.Count());
27         }
28     }
29 }

```

Este test comprueba si el método Get del controlador WeatherForecastController es capaz de devolver una colección de pronósticos del clima que no sea nula y que contenga exactamente cinco elementos. Si ambas aserciones se cumplen, el test se considera exitoso y confirma que el método está funcionando como se espera.

```

settings.json x
.vscode > settings.json > ...
1  {
2      "dotnet-test-explorer.testProjectPath": "SimpleWebAPITests",
3      "dotnet-test-explorer.testFileSuffix": ".Tests.dll"
4  }

```

```

TESTING
.NET TEST EXPLORER
✓ Test1

```

```

Starting test execution, please wait...
A total of 1 test files matched the specified pattern.
Results File: /tmp/test-explorer-ci0JzJ/0.trx

Passed! - Failed:    0, Passed:    1, Skipped:    0, Total:    1, Duration: 33 ms -
SimpleWebAPITests.dll (net7.0)

```