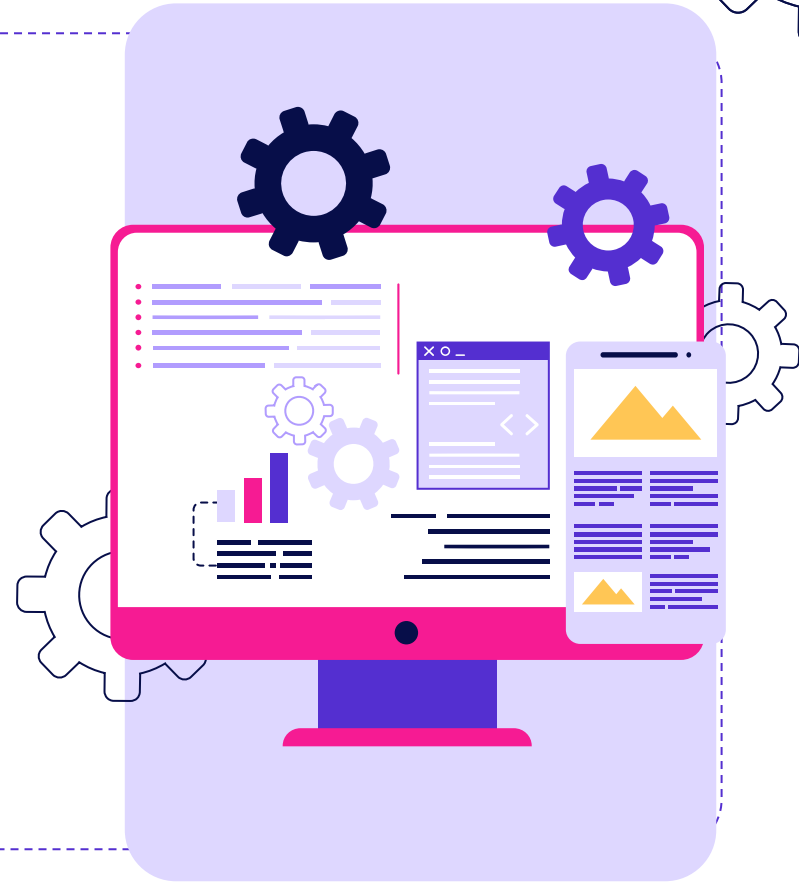
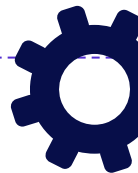


# PRUEBAS UNITARIAS

Claudia Pereira Cuba



# ÍNDICE



**01**

**¿Qué es una prueba unitaria?**

**02**

**¿Qué es NUnit?**

**03**

**Características de la librería “NUnit”**

**04**

**Instalación de “Nunit”**

**05**

**Uso y ejemplo**

**06**

**Conclusión**



# ¿Qué es una prueba unitaria?

Es un método de verificación que permite comprobar el correcto funcionamiento de las unidades individuales del código, generalmente funciones o métodos, en un programa. Su objetivo es garantizar que cada unidad funcione de acuerdo con las especificaciones requeridas, facilitando así la detección temprana de errores en el desarrollo, son conocidas también como “unit testing”, en lugar de probar todo el software, este método lo divide en secciones más pequeñas lo que permite verificar el funcionamiento adecuado de cada unidad individualmente.

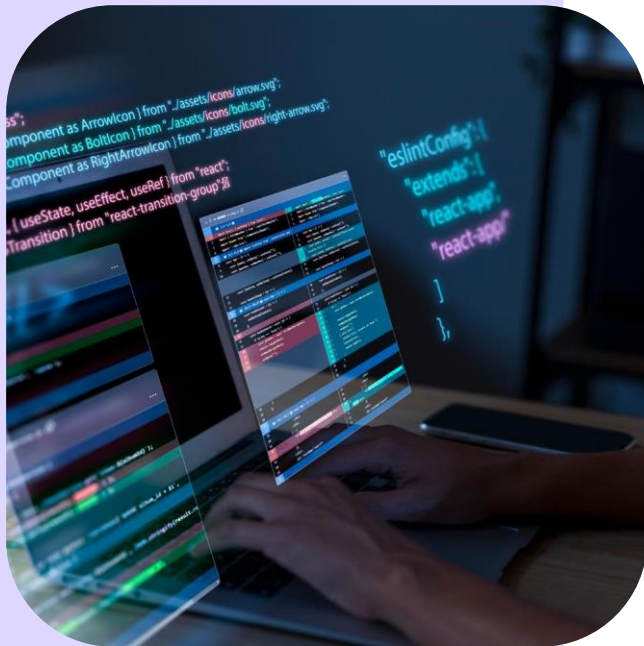


# ¿Qué es NUnit?



NUnit es un marco de pruebas unitarias para todos los lenguajes .Net. Inicialmente adaptado de JUnit , la versión de producción actual, la versión 3, ha sido completamente reescrita con muchas características nuevas y compatibilidad con una amplia gama de plataformas .NET.





# 03

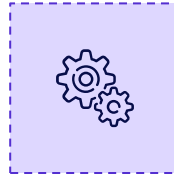
## Características de la librería



# Características de la librería “NUnit”



**Soporta los  
lenguajes C#,  
J#, Visual Basic  
y C++**



**Es multiplataforma,  
pudiendo  
ejecutarse en  
Windows y Linux.**



**Se puede  
integrar con el  
Visual Studio en  
cualquiera de  
sus versiones.**



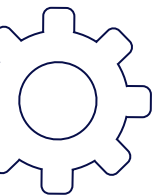
# Características de la librería “Nunit”



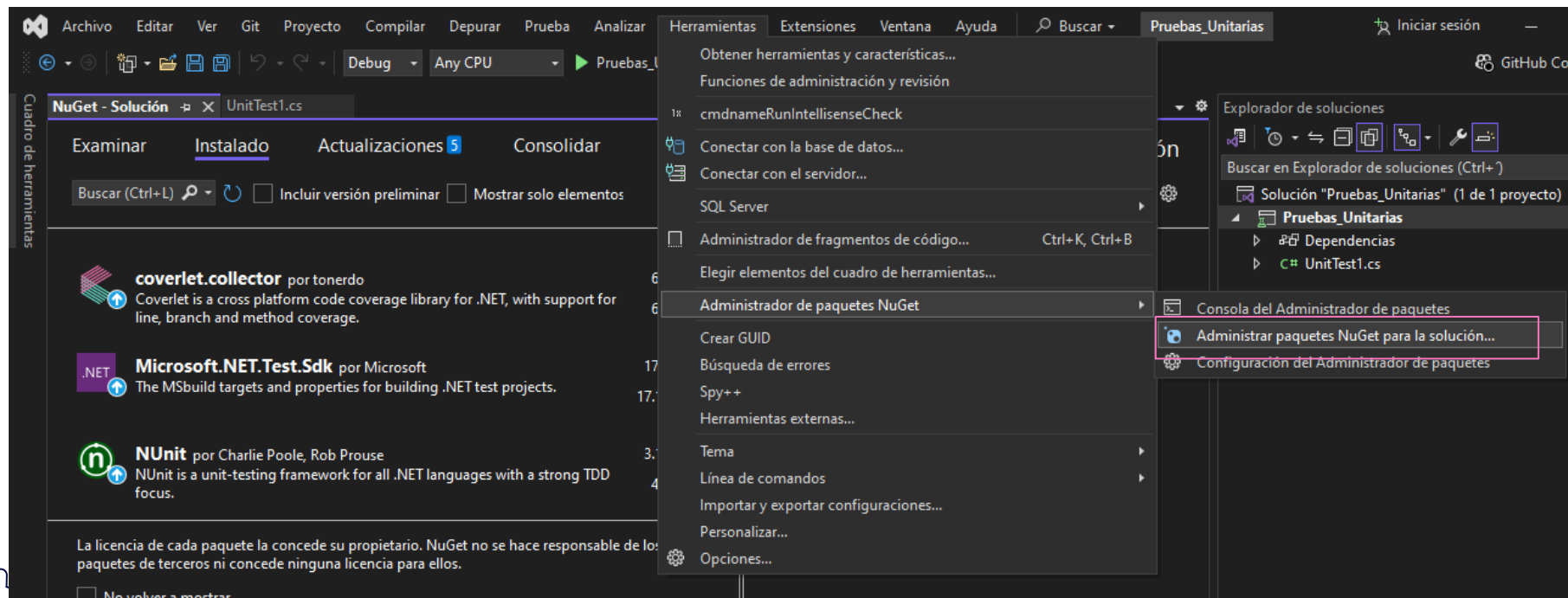
**Atributos personalizados:** Son anotaciones que se colocan sobre métodos o clases. Ej. [Test], [SetUp], [TearDown], etc. **Aserciones:** Métodos proporcionados por NUnit para comprobar y comparar valores en las pruebas. Ej. **Assert.AreEqual(), Assert.IsTrue(), etc.**



**Actualmente soporta los frameworks 1.1/2.0.**

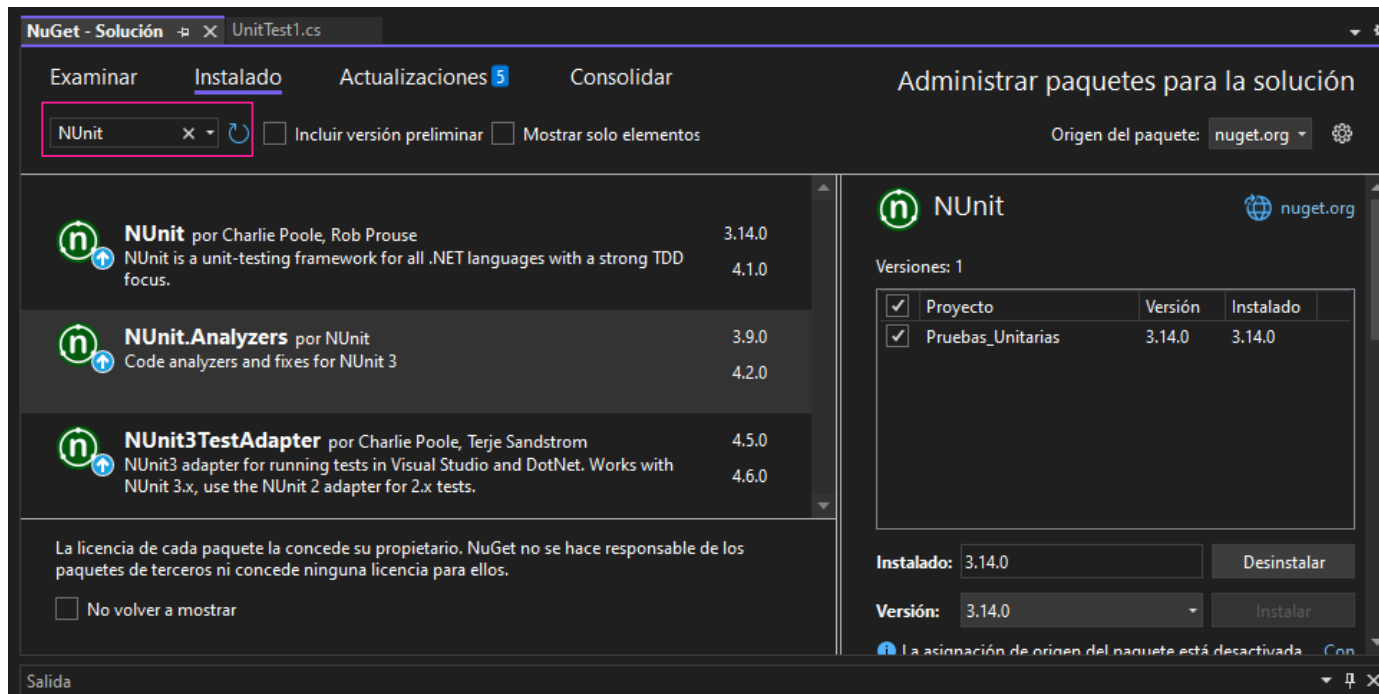


# Instalación de “Nunit y NUnit3TestAdapter”





# Instalación de “Nunit y NUnit3TestAdapter”



The screenshot shows the NuGet Package Manager window in Visual Studio. The 'Instalado' tab is selected, and the search filter is set to 'NUnit'. The list of installed packages includes NUnit, NUnit.Analyzers, and NUnit3TestAdapter. The right-hand pane shows the details for the selected package, NUnit, including its version (3.14.0) and a table of installed packages.

**NuGet - Solución** -p x UnitTest1.cs

Examinar Instalado Actualizaciones 5 Consolidar

NUnit x ↻ ☐ Incluir versión preliminar ☐ Mostrar solo elementos

Administrar paquetes para la solución

Origen del paquete: nuget.org

**NUnit** por Charlie Poole, Rob Prouse 3.14.0  
NUnit is a unit-testing framework for all .NET languages with a strong TDD focus. 4.1.0

**NUnit.Analyzers** por NUnit 3.9.0  
Code analyzers and fixes for NUnit 3 4.2.0

**NUnit3TestAdapter** por Charlie Poole, Terje Sandstrom 4.5.0  
NUnit3 adapter for running tests in Visual Studio and DotNet. Works with NUnit 3.x, use the NUnit 2 adapter for 2.x tests. 4.6.0

La licencia de cada paquete la concede su propietario. NuGet no se hace responsable de los paquetes de terceros ni concede ninguna licencia para ellos.

☐ No volver a mostrar

**NUnit** nuget.org

Versión: 1

	Proyecto	Versión	Instalado
<input checked="" type="checkbox"/>	Proyecto		
<input checked="" type="checkbox"/>	Pruebas_Unitarias	3.14.0	3.14.0

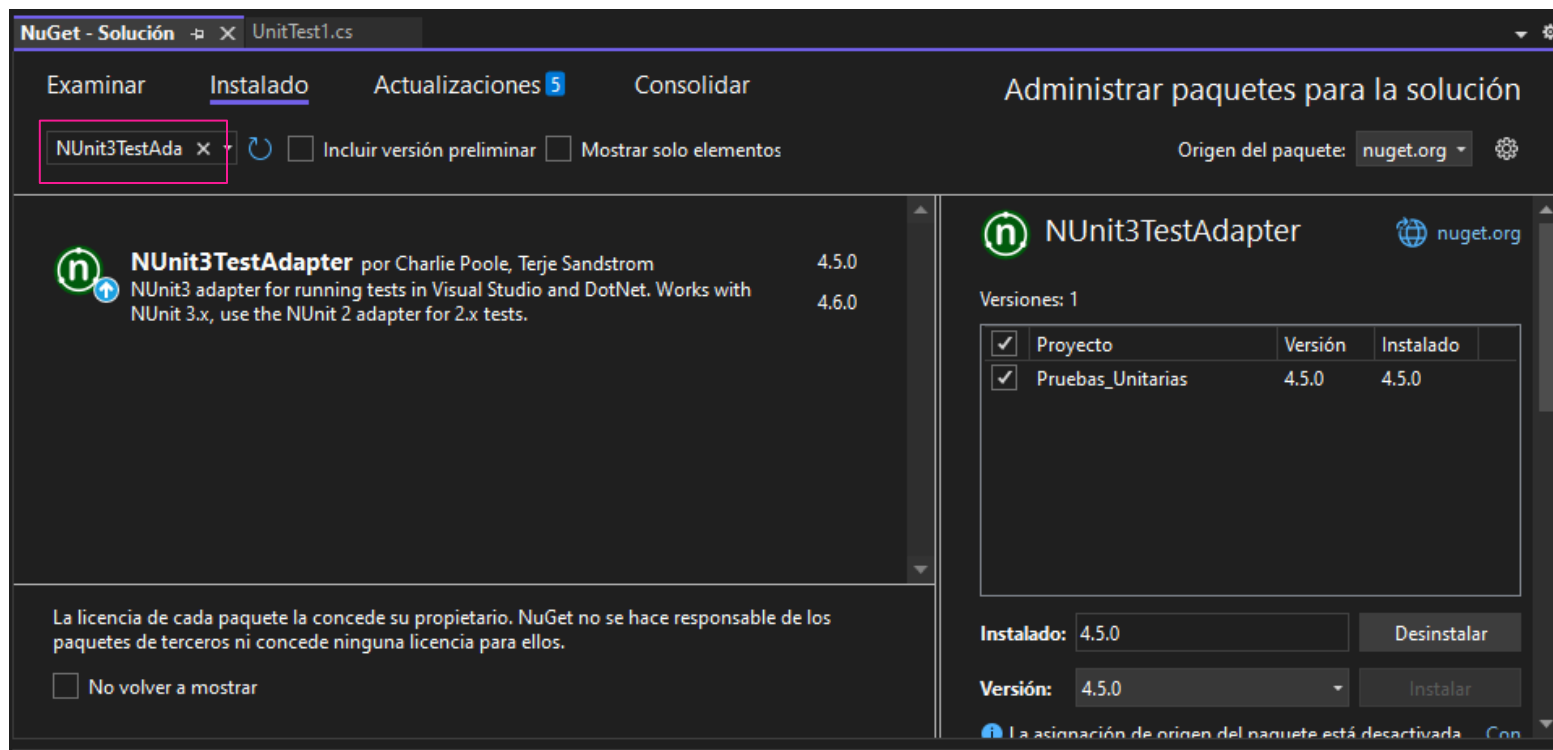
Instalado: 3.14.0 Desinstalar

Versión: 3.14.0 Instalar

La asignación de origen del paquete está desactivada. Con

Salida

# Instalación de “Nunit y NUnit3TestAdapter”



The screenshot shows the NuGet Package Manager interface in Visual Studio. The 'Instalado' (Installed) tab is selected. The package 'NUnit3TestAdapter' is highlighted in the left pane. The right pane shows the package details, including the version 4.5.0 and the 'Pruebas\_Unitarias' project.

**NUnit3TestAdapter** por Charlie Poole, Terje Sandstrom

NUnit3 adapter for running tests in Visual Studio and DotNet. Works with NUnit 3.x, use the NUnit 2 adapter for 2.x tests.

4.5.0 4.6.0

La licencia de cada paquete la concede su propietario. NuGet no se hace responsable de los paquetes de terceros ni concede ninguna licencia para ellos.

☐ No volver a mostrar

**NUnit3TestAdapter** nuget.org

Versiones: 1

<input checked="" type="checkbox"/>	Proyecto	Versión	Instalado
<input checked="" type="checkbox"/>	Pruebas_Unitarias	4.5.0	4.5.0

Instalado: 4.5.0 Desinstalar

Versión: 4.5.0 Instalar

La asignación de origen del paquete está desactivada. Con

# Instalación de “Nunit y NUnit3TestAdapter”

```
Consola del Administrador de paquetes
Origen del paquete: Todo [icon] Proyecto predeterminado: Pruebas_Unitarias [icon] [icon] [icon]
PM>
PM> Install-Package NUnit
Restaurando paquetes para C:\Users\PC\Documents\Pruebas_Unitarias\Pruebas_Unitarias\Pruebas_Unitarias.csproj...
GET https://api.nuget.org/v3-flatcontainer/nunit/index.json
OK https://api.nuget.org/v3-flatcontainer/nunit/index.json 585 ms
GET https://api.nuget.org/v3-flatcontainer/nunit/4.1.0/nunit.4.1.0.nupkg
OK https://api.nuget.org/v3-flatcontainer/nunit/4.1.0/nunit.4.1.0.nupkg 34 ms
Se ha instalado NUnit 4.1.0 de https://api.nuget.org/v3/index.json a C:\Users\PC\.nuget\packages\nunit\4.1.0 con el hash de
contenido MT/DpAhjtitytzhTgTqIhBuWx4y26PKfDepYUHUM+5uv4TsryHC2jwFo5e6NhWkApCm/G6kZ80dRjdJFuAxq3rg==.
GET https://api.nuget.org/v3-vulnerabilities/index.json
OK https://api.nuget.org/v3-vulnerabilities/index.json 34 ms
GET https://api.nuget.org/v3-vulnerabilities/2024.08.06.11.56.07/vulnerability.base.json
GET https://api.nuget.org/v3-vulnerabilities/2024.08.06.11.56.07/2024.08.06.23.56.09/vulnerability.update.json

Consola del Administrador de paquetes
Origen del paquete: Todo [icon] Proyecto predeterminado: Pruebas_Unitarias [icon] [icon] [icon]
PM> Install-Package NUnit3TestAdapter
Restaurando paquetes para C:\Users\PC\Documents\Pruebas_Unitarias\Pruebas_Unitarias\Pruebas_Unitarias.csproj...
GET https://api.nuget.org/v3-flatcontainer/nunit3testadapter/index.json
OK https://api.nuget.org/v3-flatcontainer/nunit3testadapter/index.json 1055 ms
GET https://api.nuget.org/v3-flatcontainer/nunit3testadapter/4.6.0/nunit3testadapter.4.6.0.nupkg
OK https://api.nuget.org/v3-flatcontainer/nunit3testadapter/4.6.0/nunit3testadapter.4.6.0.nupkg 41 ms
Se ha instalado NUnit3TestAdapter 4.6.0 de https://api.nuget.org/v3/index.json a C:\Users\PC\.nuget\packages\nunit3testadapter
\4.6.0 con el hash de contenido R7e1+a4vuV\YS+ItfL7f\rrG+JBvVeVLX4mHzFEZo4WlqEK18Zz27AqvQSAqo+BtIzUCo4aAJMYa56VXS4hudw==.
CACHE https://api.nuget.org/v3-vulnerabilities/index.json
CACHE https://api.nuget.org/v3-vulnerabilities/2024.08.06.11.56.07/vulnerability.base.json
CACHE https://api.nuget.org/v3-vulnerabilities/2024.08.06.11.56.07/2024.08.06.23.56.09/vulnerability.update.json
Instalando el paquete NuGet NUnit3TestAdapter 4.6.0.
Generación de archivo MSBuild C:\Users\PC\Documents\Pruebas_Unitarias\Pruebas_Unitarias\obj
100%
```



# Uso y ejemplo de “Nunit” en C#

```
GestionarLibros
└─ GestionarLibros.Libro
  using System;
  using System.Collections.Generic;
  using System.Linq;
  using System.Text;
  using System.Threading.Tasks;

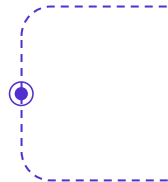
  namespace GestionarLibros
  {
    3 referencias
    public partial class Libro
    {
      2 referencias | 1/1 pasando
      public string Titulo { get; set; }
      2 referencias | 1/1 pasando
      public string Autor { get; set; }
      4 referencias | 1/1 pasando
      public double Precio { get; set; }
      4 referencias | 2/2 pasando
      public bool Disponible { get; set; }

      1 referencia
      public Libro(string titulo, string autor, double precio, bool disponible)
      {
        Titulo = titulo;
        Autor = autor;
        Precio = precio;
        Disponible = disponible;
      }

      2 referencias | 2/2 pasando
      public double CalcularPrecioConDescuento(double porcentajeDescuento)
      {
        if (porcentajeDescuento < 0 || porcentajeDescuento > 100)
          throw new ArgumentException("El porcentaje de descuento debe estar entre 0 y 100.");

        return Precio - (Precio * (porcentajeDescuento / 100));
      }

      2 referencias | 2/2 pasando
      public bool EstaDisponible()
      {
        return Disponible;
      }
    }
  }
```



# Uso y ejemplo de “Nunit” en C#

```
GestonarLibros
GestorLibros.Tests.LibroTests

using GestonarLibros;
using NUnit.Framework;
using GestorLibros;
using System;

namespace GestorLibros.Tests
{
    [TestFixture]
    0 referencias
    public class LibroTests
    {
        private Libro _libro;

        [SetUp]
        0 referencias
        public void Setup()
        {
            _libro = new Libro("El principito", "Antoine de Saint", 29.99, true);
        }

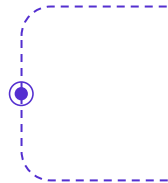
        [Test]
        0 referencias
        public void CalcularPrecioConDescuento_CorrectoDescuento()
        {
            double resultado = _libro.CalcularPrecioConDescuento(10);
            Assert.AreEqual(26.99, resultado, 0.01); //margen de error para comparar doubles
        }
    }
}
```

Precio con descuento =  $29.99 - (29.99 \times 0.10)$

Precio con descuento =  $29.99 - 2.999$

Precio con descuento = 26.991

# Uso y ejemplo de “Nunit” en C#

A screenshot of a Visual Studio code editor window. The title bar shows 'GestionarLibros' and 'GestorLibros.Tests.Libro Tests'. The code is in C# and shows two Nunit test methods. The first method, 'CalcularPrecioConDescuento\_DescuentoInvalido\_Excepcion()', is marked with a green circle icon and '0 referencias'. It uses 'Assert.Throws<ArgumentException>' to verify that an exception is thrown when 'CalcularPrecioConDescuento(-1)' is called. The second method, 'EstaDisponible\_LibroDisponible\_ReturnsTrue()', is also marked with a green circle icon and '0 referencias'. It uses 'Assert.IsTrue()' to verify that 'EstaDisponible()' returns true. The code is syntax-highlighted with colors: green for attributes, blue for keywords, and black for identifiers and literals.

```
[Test]
0 referencias
public void CalcularPrecioConDescuento_DescuentoInvalido_Excepcion()
{
    Assert.Throws<ArgumentException>(() => _libro.CalcularPrecioConDescuento(-1));
}

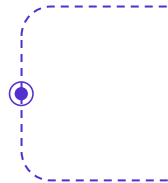
[Test]
0 referencias
public void EstaDisponible_LibroDisponible_ReturnsTrue()
{
    Assert.IsTrue(_libro.EstaDisponible());
}
```



# Uso y ejemplo de “Nunit” en C#

```
[Test]
0 | 0 referencias
public void EstaDisponible_LibroNoDisponible_ReturnsFalse()
{
    _libro.Disponible = false;
    Assert.IsFalse(_libro.EstaDisponible());
}

[Test]
0 | 0 referencias
public void Constructor_CreaLibroCorrectamente()
{
    Assert.AreEqual("El principito", _libro.Titulo);
    Assert.AreEqual("Antoine de Saint", _libro.Autor);
    Assert.AreEqual(29.99, _libro.Precio, 0.01);
    Assert.IsTrue(_libro.Disponible);
}
}
```



# Conclusión

Las pruebas unitarias son una herramienta fundamental en el desarrollo de software moderno, permitiendo verificar el funcionamiento de unidades individuales de código. NUnit se destaca como un marco para implementar pruebas en el ecosistema .NET, ofreciendo versatilidad, compatibilidad multiplataforma y una integración fluida con entornos de desarrollo como Visual Studio. Sus características, como atributos y aserciones, facilitan la creación y ejecución de pruebas efectivas, contribuyendo a mejorar la calidad del software y la detección temprana de errores.

