Clases

```
namespace Universidad
{
    public interface IProfesor
    {
        public string GetNombreCompleto();
    }

    public class NameValidator
    {
        public Boolean validateName(string nombre) { return true; }
    }

    public class Profesor: IProfesor
    {
        private string _nombre;
        private string _apellido;
        private NameValidator validator = new NameValidator();

        public Profesor(string nombre, string apellido)
        {
            if (validator.validateName(nombre))
            {
                throw new InvalidDataException();
            }
            else
            {
                _nombre = nombre;
                _apellido = apellido;
            }
        }

        public string GetNombreCompleto() { return $""; }
    }

    public class Materia
    {
        private string _nombre;
```

```
        private IProfesor? _profesor;

        public Materia(string nombre)
        {
            _nombre = nombre;
        }

        public string Nombre { get { return _nombre; } }
        public IProfesor? Profesor
        {
            get { return _profesor; } set { if (value is not null) {
_profesor = value; } }
        }

        public string? NombreProfesor()
        {
            if (_profesor is not null) { return
_profesor.GetNombreCompleto(); }
            else { return null; }
        }

    }
}
```

```
using Moq;
using Universidad;

namespace TestingExample
{
    [TestClass]
    public class UnitTestProfesor
    {
        Materia matematicas;
        Materia ingles;

        [TestMethod]
        public void Profesor_NuevoProfesor_RetornaNombre()
        {
            Profesor profesor = new Profesor("Esteban", "Quito");
            Assert.AreEqual("Esteban Quito",
profesor.GetNombreCompleto());
        }

        [TestMethod]
        public void Profesor_NuevoProfesor_NoPuedeTenerNombreVacio()
        {
            Assert.ThrowsException<InvalidDataException>(() => new
Profesor("", "Quito"));
        }


        [TestInitialize]
        public void Init()
        {
            Materia matematicas = new Materia("Matematicas");
            Materia ingles = new Materia("Ingles");
        }

        [TestMethod]
        public void
Materia_TieneProfesorAsignado_RetornaNombreProfesor()
```

```
        {
            // arrange
            Mock<IProfesor> profesor = new Mock<IProfesor>();
            profesor.Setup(prof =>
prof.GetNombreCompleto()).Returns("Esteban Quito");
            matematicas.Profesor = profesor.Object;

            //act
            string? nombreProfesor = matematicas.NombreProfesor();

            // assert
            profesor.Verify(prof => prof.GetNombreCompleto(),
Times.Once());
            Assert.AreEqual("Esteban Quito", nombreProfesor);
        }

        [TestMethod]
        public void Validator_NameValidator_RetornoTrueSiCumple()
        {
            NameValidator validator = new NameValidator();
            Assert.IsTrue(validator.validateName("pepe"));
        }

    }
}
```