[TestClass]

public class PersonaUnitTest

{

[TestMethod]

public void DNI\_Invalido\_Texto()

{

// CASO 1 DNI con COMA

string dniComa = "30.999,999";

try

{

// DNI Invalido?

Persona personaPrimero = new Persona("Juan", "Lopez", dniComa, Persona.ENacionalidad.Argentino);

Assert.Fail("Sin excepción para DNI inválido: {0}.", dniComa);

}

catch (Exception e)

{

Assert.IsInstanceOfType(e, typeof(DniInvalidoException));

}

// CASO 2 DNI con texto

string dniTexto = "30a00123";

try

{

// DNI Invalido?

Persona personaUltimo = new Persona("Juan", "Lopez", dniTexto, Persona.ENacionalidad.Argentino);

Assert.Fail("Sin excepción para DNI inválido: {0}.", dniTexto);

}

catch (Exception e)

{

Assert.IsInstanceOfType(e, typeof(DniInvalidoException));

}

}

/// <summary>

/// Comprobar que el DNI no pueda ser menor a 1

/// Arroja NacionalidadInvalidaException

/// </summary>

[TestMethod]

public void DNI\_Invalido\_Bajo()

{

// CASO 1 DNI menor a 1

int dniPrimero = 0;

try

{

// DNI Invalido?

Persona personaPrimero = new Persona("Juan", "Lopez", dniPrimero, Persona.ENacionalidad.Argentino);

}

catch (Exception e)

{

Assert.IsInstanceOfType(e, typeof(NacionalidadInvalidaException));

return;

}

Assert.Fail("Sin excepción para DNI inválido: {0}.", dniPrimero);

}

/// <summary>

/// Comprobar que el DNI no pueda ser mayor a 99.999.999

/// </summary>

[TestMethod]

public void DNI\_Invalido\_Alto()

{

// CASO 2 DNI mayor a 99.999.999

int dniUltimo = 100000000;

try

{

// DNI Invalido?

Persona personaUltimo = new Persona("Juan", "Lopez", dniUltimo, Persona.ENacionalidad.Extranjero);

}

catch (Exception e)

{

Assert.IsInstanceOfType(e, typeof(NacionalidadInvalidaException));

return;

}

Assert.Fail("Sin excepción para DNI inválido: {0}.", dniUltimo);

}

/// <summary>

/// Comprobar los rangos de DNI para Argentinos

/// </summary>

[TestMethod]

public void DNI\_Validos\_Argentino()

{

// CASO 1 DNI al azar

int dniMedio = new Random().Next(1, 89999999);

Persona personaMedio = new Persona("Juan", "Lopez", dniMedio, Persona.ENacionalidad.Argentino);

// Cargó OK?

Assert.AreEqual(personaMedio.DNI, dniMedio);

// CASO 2 primer DNI válido

int dniPrimero = 1;

Persona personaPrimero = new Persona("Juan", "Lopez", dniPrimero, Persona.ENacionalidad.Argentino);

// Cargó OK?

Assert.AreEqual(personaPrimero.DNI, dniPrimero);

// CASO 3 último DNI válido

int dniUltimo = 89999999;

Persona personaUltimo = new Persona("Juan", "Lopez", dniUltimo, Persona.ENacionalidad.Argentino);

// Cargó OK?

Assert.AreEqual(personaUltimo.DNI, dniUltimo);

// CASO 4 DNI string

string dniString = "89.999.999";

Persona personaString = new Persona("Juan", "Lopez", dniString, Persona.ENacionalidad.Argentino);

// Cargó OK?

Assert.AreEqual(personaString.DNI, 89999999);

}

/// <summary>

/// Comprobar los rangos de DNI para Extranjeros

/// </summary>

[TestMethod]

public void DNI\_Validos\_Extranjeros()

{

// CASO 1 DNI al azar

int dniMedio = new Random().Next(90000000, 99999999);

Persona personaMedio = new Persona("Juan", "Lopez", dniMedio, Persona.ENacionalidad.Extranjero);

// Cargó OK?

Assert.AreEqual(personaMedio.DNI, dniMedio);

// CASO 2 primer DNI válido

int dniPrimero = 90000000;

Persona personaPrimero = new Persona("Juan", "Lopez", dniPrimero, Persona.ENacionalidad.Extranjero);

// Cargó OK?

Assert.AreEqual(personaPrimero.DNI, dniPrimero);

// CASO 3 último DNI válido

int dniUltimo = 99999999;

Persona personaUltimo = new Persona("Juan", "Lopez", dniUltimo, Persona.ENacionalidad.Extranjero);

// Cargó OK?

Assert.AreEqual(personaUltimo.DNI, dniUltimo);

// CASO 4 DNI string

string dniString = "99.999.999";

Persona personaString = new Persona("Juan", "Lopez", dniString, Persona.ENacionalidad.Extranjero);

// Cargó OK?

Assert.AreEqual(personaString.DNI, 99999999);

}

}