2. Escribir tests

El siguiente ejemplo es muy básico, pues solo posee lo necesario. Los siguientes puntos ofrecerán más información al respecto.

[TestFixture]

public class MainPageModelTests

{

[Test]

public void ButtonClickTest()

{

var mainPageModel = new MainPageModel();

mainPageModel.ChangeTextCommand.Execute(null);

Assert.AreEqual("You clicked the button", mainPageModel.LabelText);

}

}

2.1. Anotaciones

N-Unit soporta las siguientes anotaciones para configurar pruebas.

|  |  |
| --- | --- |
| Anotación | Uso |
| ***Category*** | Especifica una o más cateforías para el test |
| ***Culture*** | Especifica “culturas” para cada uno de los tests o accesorios que deberían funcionar |
| ***Indicates*** | Indica el test que debería saltarse a menos que funcione explícitamente |
| ***Ignore*** | Indica el test que no debería funcionar por alguna razon |
| **MaxTime** | Especifica el tiempo máximo en milisegundos para que un test de casos de uso funcione |
| ***OneTimeSetUp*** | Identifica los métodos que son llamades una vez la prioridad afecte a cualquiera de los sucesores de los tests |
| ***OneTimeTearDown*** | Identifica los métodos que son llamados después de todos los sucesores de los tests |
| ***Platform*** | Especifica la plataforma para cada test o accesorio que debería funcionar |
| ***Random*** | Especifica la generación de valores aleatorios como argumentos a una prueba parametrizada |
| ***Repeat*** | Especifica el método decorado que debería ser ejecutado varias veces |
| ***TearDown*** | Indica al método del accesorio llamado inediatamente después de cada método de las pruebas |
| ***Test*** | Marca un método del accesorio que representa una prueba |
| ***TestCase*** | Marca el método con parámetros como un test y proviene argumentos en esa línea de código |
| ***TimeOut*** | Proviene un valor de tiempo muerto en milisegundos para pruebas de casos de uso |
| ***SetUp*** | Indica un método del accesorio llamado inmediatamente antes de cada método de cada prueba |
| ***Retry*** | Provoca que el test vuelva a funcionar si este falla, hasta un número máximo de veces |

2.2. Ejemplos

En los siguientes puntos se adjuntarán líneas de código que muestran ejemplos de las anotaciones mostradas anteriormente.

2.1.1 SetUp

Este código se usa generalmente antes de ejecutar una prueba, con el propósito de no repetir cada una.

[SetUp]

public void Initialize() {

*//driver = new FirefoxDriver();*

browser = ConfigurationManager.AppSettings["browser"];

switch (browser)

{

case "Chrome":

driver = new ChromeDriver();

break;

case "Firefox":

driver = new FirefoxDriver();

break;

}

driver.Manage().Window.Maximize();

}

2.1.2 TearDown

Es una anotación que actúa al contrario del SetUp, lo que significa que el código escrito con el atributo es ejecutado al último momento (después de pasar por las otras líneas de código).

[TearDown]

public void EndTest()

{

if (TestContext.CurrentContext.Result.Outcome.Status == NUnit.Framework.Interfaces.TestStatus.Failed) {

test.Log(LogStatus.Fail, TestContext.CurrentContext.Result.Message);

}

if (TestContext.CurrentContext.Result.Outcome.Status == NUnit.Framework.Interfaces.TestStatus.Passed)

{

test.Log(LogStatus.Pass, "Test Succesfull. ");

}

driver.Close();

report.EndTest(test);

report.Flush();

report.Close();

}

2.1.3 Test

Es la declaración de los pasos que van a ser ejecutados en la prueba.

[Test]

public void testLogin(string user, string message)

{

HomePage objHomePage = new HomePage(driver);

objHomePage.goToLogin();

LoginPage objLoginPage = new LoginPage(driver);

objLoginPage.loginWithUser(user);

MyAccountPage objAccountPage = new MyAccountPage(driver);

objAccountPage.waitForLoad();

Assert.AreEqual(objAccountPage.UserMessage(),message);

}

2.1.4 TestCase

Esta anotación reemplaza la prueba que vimos anteriormente, y permite crear varias pruebas que ejecutarán el mismo código con diferentes introducciones de datos.

[TestCase(“user1”, “Welcome user1”)]

[TestCase(“user2”, “Invalid User”)]

[TestCase(“user3”,”Not Registered User”)]

public void testLogin(string user, string pw){}

2.1.5 TestCaseSource

Esta anotación es similar a la anterior, y es utilizada cuando la lista de registros es muy grande. Esta toma las propiedades estáticas públicas que se aportan en cada entrada del registro. Nos permite cargar registros de un archivo.csv o de nuestra base de datos.

[TestCaseSource("TestCases")]

public void testLogin(string user, string pw){}