



## Ingeniería de software

### Laboratorio 8: Automatización de UI testing con Selenium

#### Resumen

En este laboratorio usted aprenderá a utilizar las herramientas de Selenium WebDriver y Selenium IDE, las cuales le serán útiles para realizar pruebas funcionales automáticas en el nivel de la interfaz de usuario. El laboratorio es una guía donde se presentan ejemplos básicos para que usted comprenda el funcionamiento de la herramienta.

#### Pre-requisitos

Haber completado el laboratorio 5 del curso

#### Primera parte - Selenium IDE

Selenium es una herramienta ampliamente utilizada en la industria de desarrollo para realizar pruebas, en este caso Selenium IDE es una extensión que se instala en un navegador. La extensión permite “grabar” una serie de acciones y hacer verificación de resultados, lo que hace el componente es crear el código asociado a la grabación de una prueba y posteriormente ejecutar las acciones correspondientes. Su objetivo es probar funcionalidades básicas.

La extensión se puede instalar en diversos navegadores, entre ellos Firefox y Chrome. Sin embargo, en este laboratorio todos los ejemplos se harán utilizando Google Chrome, el proceso de uso e instalación de dependencias en otros navegadores no cambia demasiado respecto a lo presentado en este laboratorio.

##### 1. Instalación de Selenium IDE

Para instalar la extensión, es necesario que ingrese en el siguiente enlace:

<https://chrome.google.com/webstore/detail/selenium-ide/mooikfahbdckldjindioackbalphokd/rrelated?hl=en> y de click en “Añadir a Chrome” o “Add to Chrome” cómo lo muestra la imagen a continuación



Home > Extensions > Selenium IDE



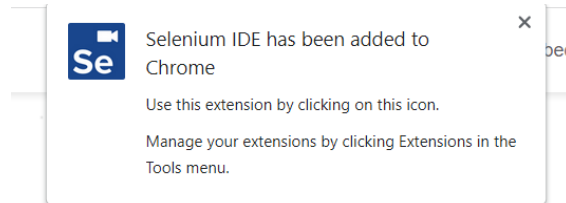
Selenium IDE

[seleniumhq.org](https://seleniumhq.org) Featured

★★★★★ 235 Developer Tools | 600,000+ users

Add to Chrome

Deben obtener un mensaje donde se instaló correctamente.



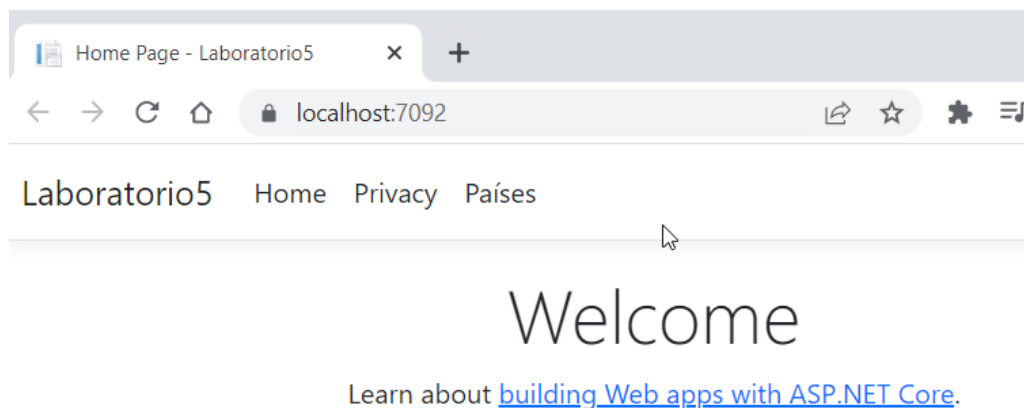
Usted puede fijar la extensión de Selenium IDE para que tenga un fácil acceso cuando necesite utilizarla. Esto se consigue, dando click al elemento marcado en la imagen 2, en las extensiones que se muestran, ubique la de Selenium IDE y seleccione la opción de fijar.

## 2. Configuración de pruebas en Selenium IDE

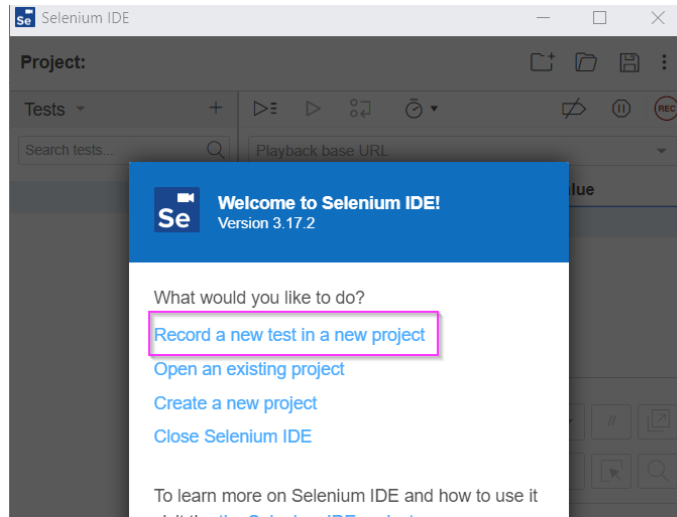
Tanto para Selenium IDE como para Selenium Web Driver, es necesario contar con la dirección del servidor donde está alojada la aplicación sobre la cual se desea hacer las pruebas. En este caso, al tratarse de un ejemplo-guía, se va realizar con la aplicación creada en el laboratorio 5.

Existe la posibilidad de configurar el ISS Express para que el sistema, a pesar de estar en el servidor local, pueda tener una especie de URL. No obstante, en este caso no se va a profundizar en los detalles para conseguir esto por simplicidad, pero usted podría investigar cómo lograr esto si es de su interés.

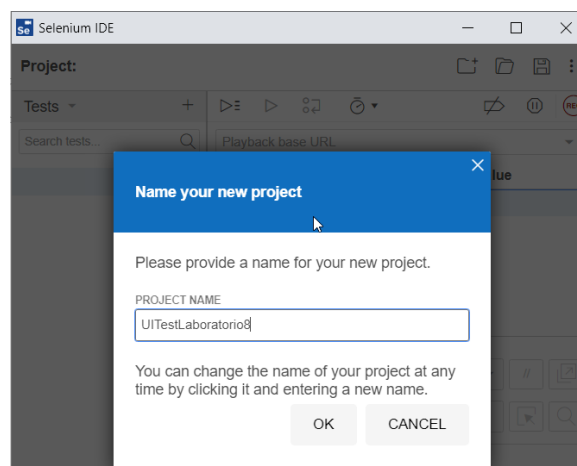
Dado que Selenium necesita una URL donde se encuentra el recurso alojado, es necesario conseguir esta información. La manera más sencilla es ejecutar el laboratorio 5 y obtener la dirección, cómo en la imagen a continuación donde se usa localhost:7092



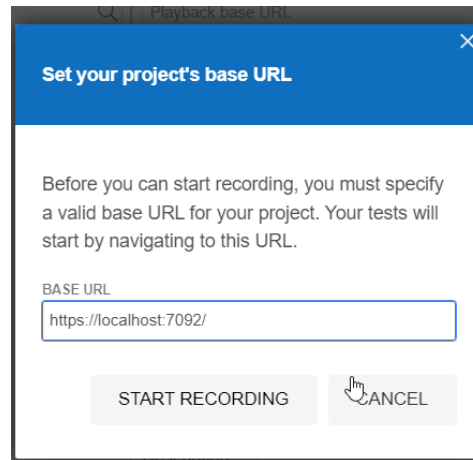
Abran la extensión de selenium y seleccione la opción Record a new test in a new project.



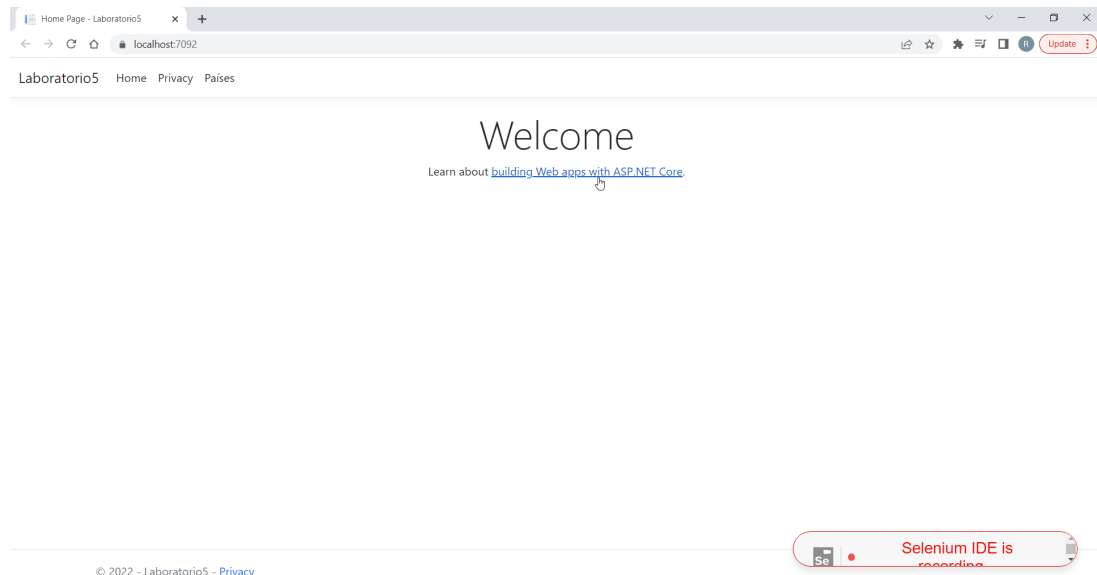
Seguidamente ponga el nombre cómo UITestLaboratorio8



A continuación, se solicita la URL o dirección del servidor donde se encuentra alojada la aplicación. Como se mencionó anteriormente el sistema está en localhost con el puerto asignado por Visual Studio. Esta dirección se obtiene desde la barra de búsqueda del navegador, copie y pegue la ruta, del mismo modo que se muestra en la imagen.



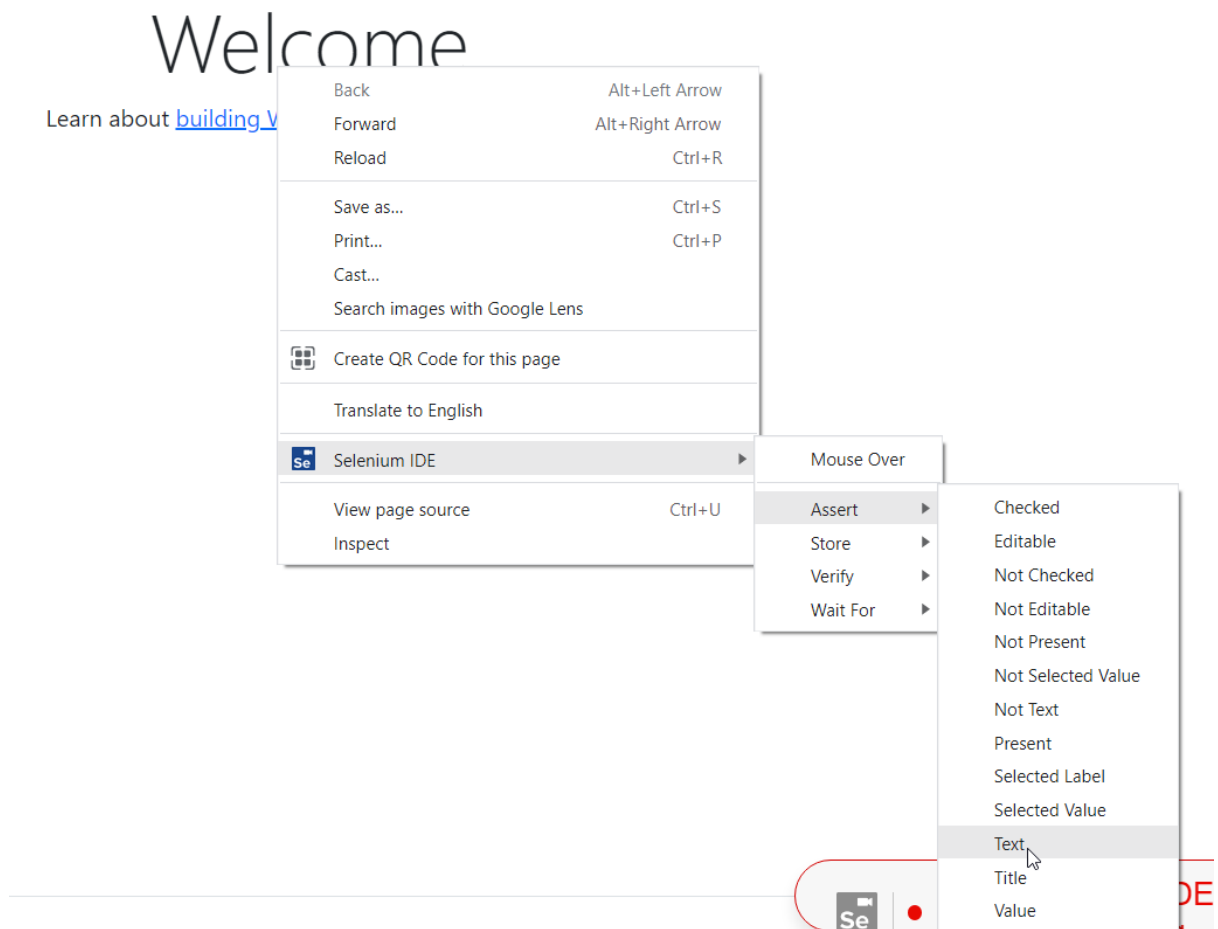
Una vez escrita la URL, dé click en *Start Recording*. Una vez hecho esto y de manera inmediata, se abrirá una nueva ventana de Google Chrome, maximice el tamaño de la ventana y verifique el resultado obtenido es similar a los que se muestra en la siguiente imagen. En este momento todo está preparado para empezar a grabar una secuencia de acciones que usted desee probar, las acciones quedan registradas según usted lo indique e incluso cuando usted dé click sobre algún lugar de la pantalla, esto se registrará como una acción que será posteriormente probada.



### 3. Grabar acciones en Selenium IDE

Para grabar una prueba, como se mencionó anteriormente, se puede dar click sobre algún botón o lugar de la pantalla, pero otra manera es registrar las acciones mediante un Assert.

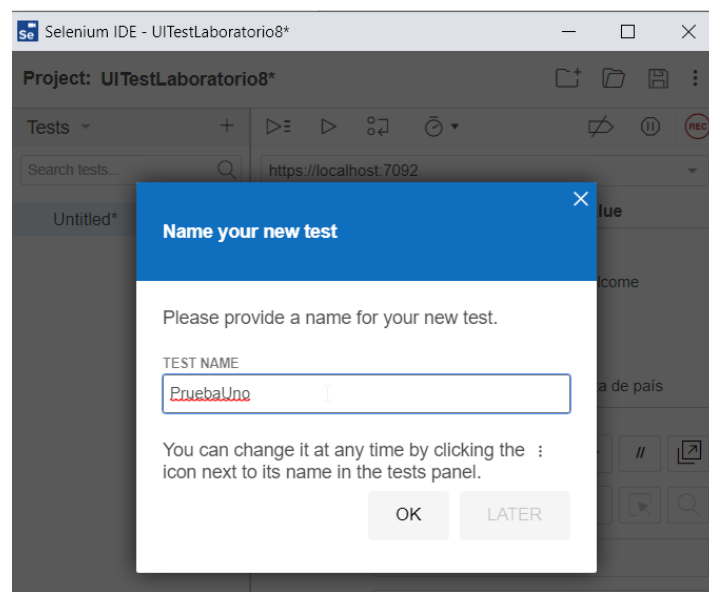
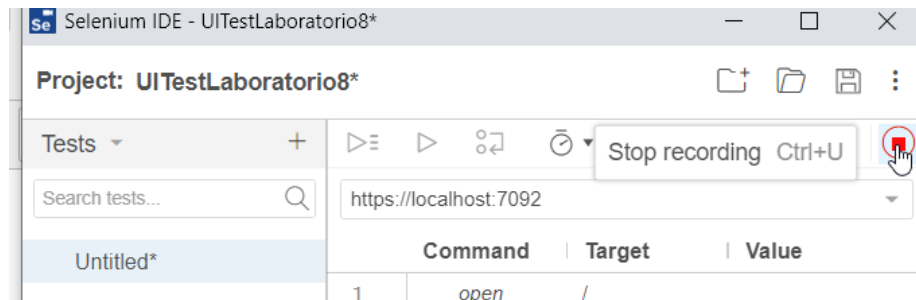
Dé click derecho exactamente encima de del texto que enuncia “Welcome” y seleccione la opción de Selenium IDE > Assert Text, tal y como se muestra en la imagen.



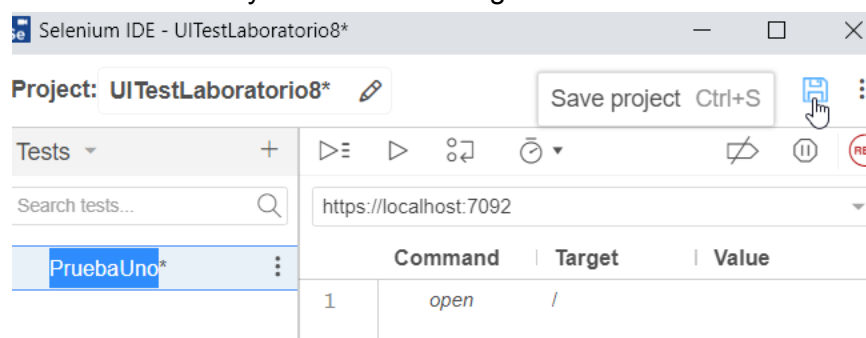
Agregue un par de asserts más en la página de inicio, por ejemplo el link de “Learn about [building Web apps with ASP.NET Core](#).” exista y otro más. Luego diríjase al tab de Países y haga otro assert en la pantalla, cómo que el titulo exista.

Después de registrar estas tres acciones, detenga la grabación de la prueba.

Esto se hace volviendo a la ventana de Selenium IDE y presionando el botón indicado en la siguiente imagen. En esta misma ventana se muestran el listado de acciones registradas, a veces por error se agregan acciones que no se desean o Selenium agrega algunas automáticamente, en este panel usted puede borrar esas acciones en caso que lo considere pertinente. Al detener la grabación, se solicita que dé un nombre a la prueba, nómbrela como PrimeraUno.



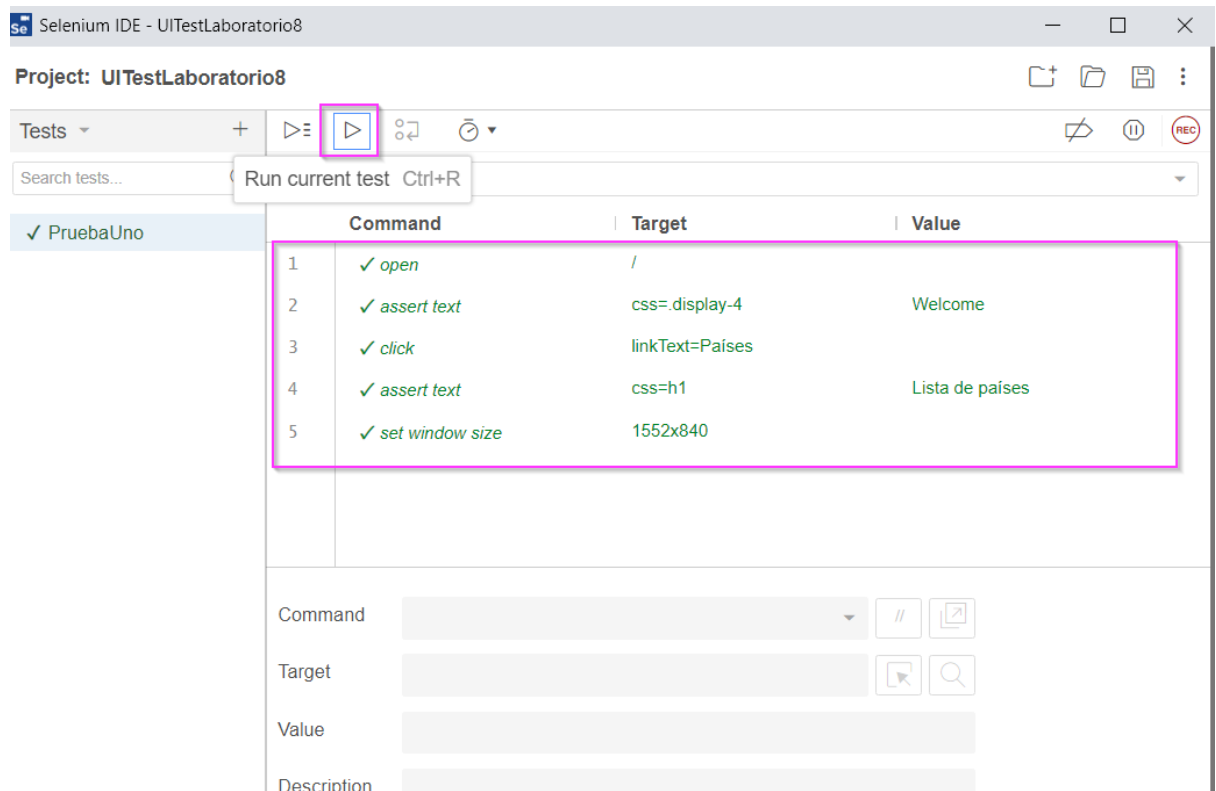
Es recomendable guardar la prueba antes de hacer la ejecución de la misma, una buena idea es guardar estas pruebas (y sus resultados) en una carpeta dentro del mismo repositorio donde está alojada la aplicación (en el remoto y local), usted puede crear este directorio y nombrarlo como guste.



Para ejecutar la prueba, basta con presionar el botón mostrado en la siguiente imagen. Observará que al realizar esto, de manera automática se va a abrir una nueva ventana de Chrome y (aunque en este caso no se observa) van a ocurrir una a una todas las acciones que fueron registradas en la prueba. Al final se espera



obtener éxito en todas las acciones y obtener un resultado similar al mostrado en la imagen.



En este caso la prueba que se hizo es bastante sencilla, dado que el objetivo es que aprendiera cual es funcionamiento básico de Selenium IDE.

## Entregable parte 1

Para este laboratorio será necesario crear una prueba más elaborada que cuente por lo menos con 5 tipos distintos de assert. Además debe crear un video de la ejecución del mismo y del resultado obtenido, además de agregar el archivo generado por selenium IDE.

Se recomienda que observe los siguientes materiales para poder comprender mejor cómo funciona Selenium IDE:

1. Selenium IDE for Beginners: <https://www.youtube.com/watch?v=m4KpTvEz3vg>
2. Grabar pruebas en Selenium: <https://www.youtube.com/watch?v=aWyKKIlepxQ>
3. Selenium IDE Commands: <https://ui.vision/rpa/docs/selenium-ide>

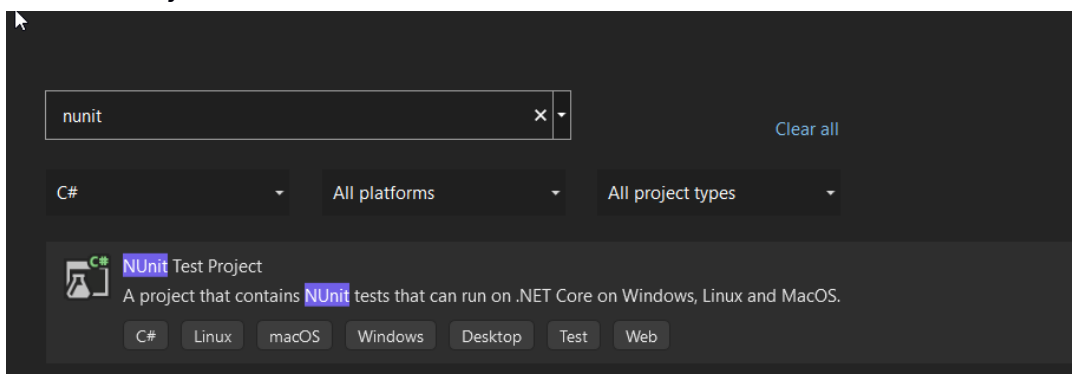


## Segunda parte - Selenium web driver

El componente de Selenium denominado web driver tiene por objetivo realizar pruebas similares a las mostradas anteriormente con el IDE, pero más robustas, el componente tiene soporte para diversos lenguajes de programación web y en este caso se hará uso dentro de una aplicación desarrollada en el laboratorio 5.

### 1. Instalación de paquetes

Para este laboratorio 8 vamos a crear una nueva solución con un proyecto de testing. Cree un proyecto de testing cómo lo aprendió en la clase de unit testing. Llámelo TestProjectLaboratorio8

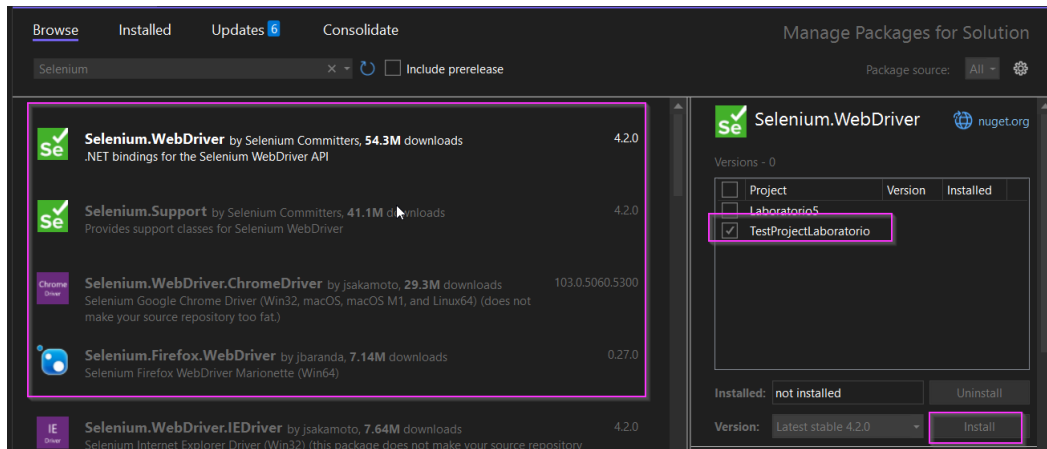


Seguidamente ingrese al administrador de paquetes NuGet. En la barra de búsqueda de paquetes para instalar, escriba Selenium, aquí obtendrá múltiples resultados. La idea es instalar los siguientes paquetes:

1. Selenium.WebDriver
2. Selenium.Support
3. Selenium.Firefox.WebDriver (si usa firefox)
4. Selenium.WebDriver.ChromeDriver

En el caso de este laboratorio, todos los ejemplos se realizan utilizando Chrome. No obstante, la instalación y escritura del código para Firefox es completamente análoga, usted podría aplicar los cambios necesarios en caso que lo considere pertinente. En las imágenes se marcan los paquetes específicos que usted debe instalar. Instale estos componentes únicamente en el proyecto de testing.





## 2. Codificación de pruebas con Selenium web driver

En el proyecto de testing creé una nueva clase y ponle de nombre Selenium. Ahora, en el interior de la clase creada, copie el siguiente código:



```
TestProject.Selenium

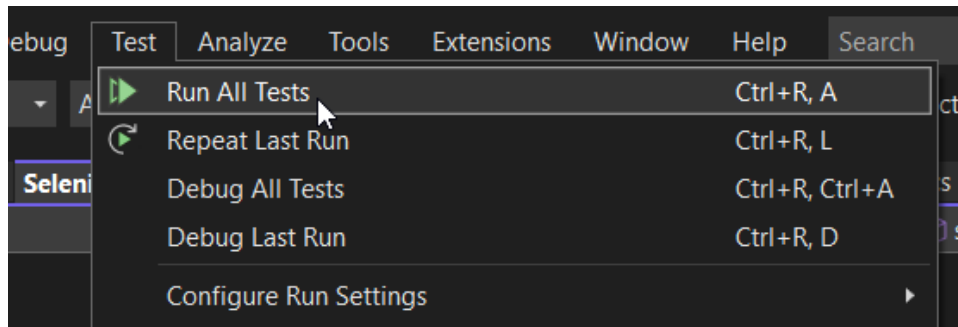
using NUnit.Framework;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;

namespace TestProject
{
    0 references
    public class Selenium
    {
        IWebDriver driver;
        [SetUp]
        0 references
        public void startBrowser()
        {
            driver = new ChromeDriver();
        }

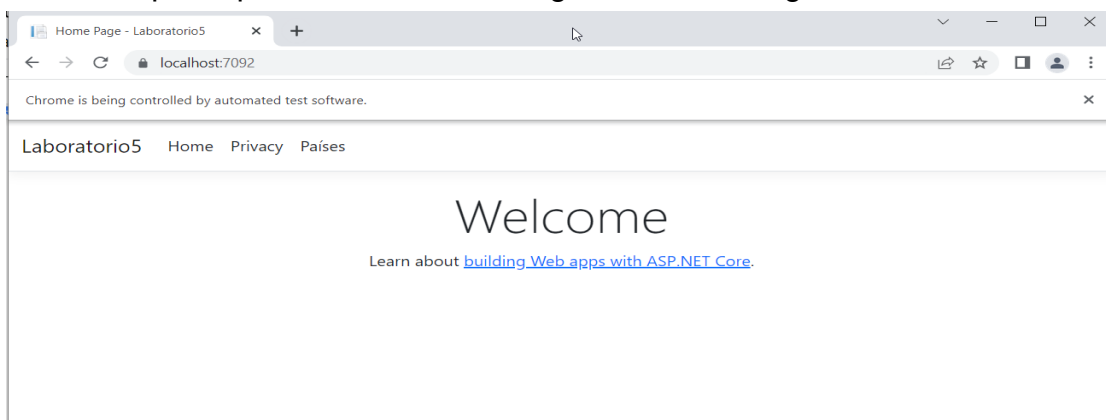
        [Test]
        ✓ | 0 references
        public void PruebaIngresoCrearPaíses()
        {
            ///Arrange
            /// Crea el driver de Chrome
            string URL = "https://localhost:7092/";
            /// Pone la pantalla en full screen
            driver.Manage().Window.Maximize();
            ///Act
            /// Se va a la URL indicada
            driver.Url = URL;
            ///Assert
            ///Debe usar otro tipo de assert
            Assert.IsNotNull(driver);
        }
    }
}
```

Antes de ejecutar la prueba, es necesario que observe que debe poner la URL de su proyecto. Tampoco se ha asignado un Assert que no tiene nada que ver con la prueba. En el caso de Selenium webdriver, es en estos métodos que se registran las acciones y existe un gran número de posibles acciones y combinaciones que se pueden realizar. En este caso, note que se registran dos acciones; abrir una ventana de Chrome en pantalla completa y además ingresar a la URL indicada.

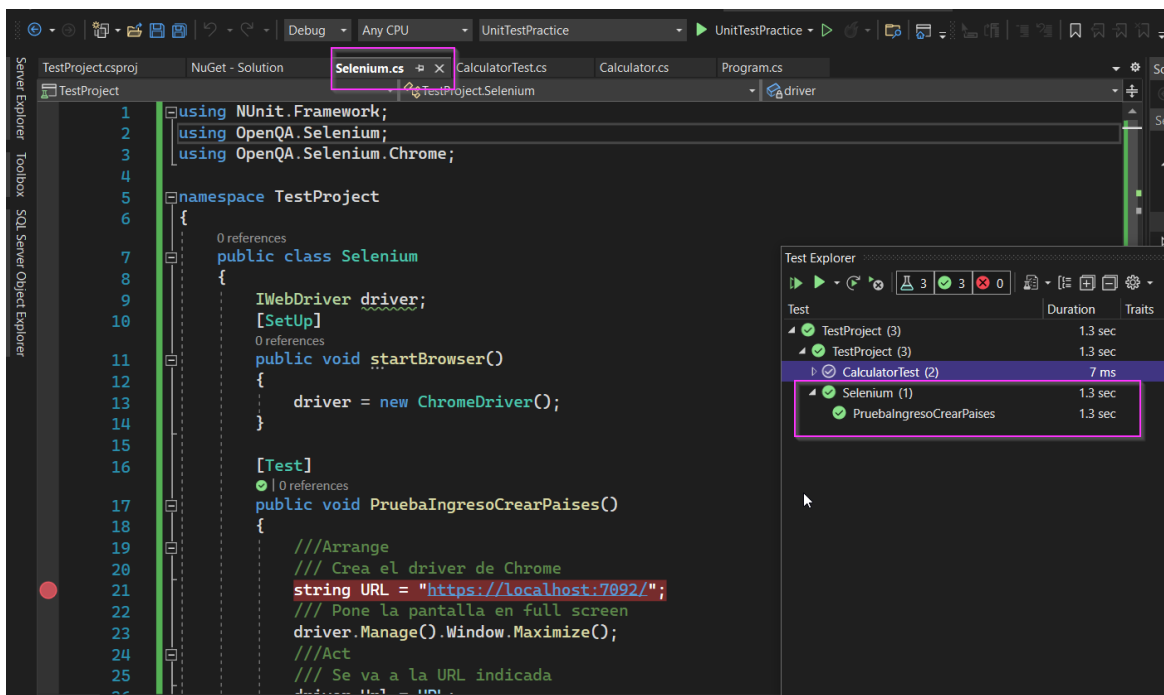
Para poder ejecutar esta prueba el proyecto del laboratorio 5 debe estar en modo ejecución. Además para ejecutar la prueba diríjase a la opción de test



Notara la que la prueba abrirá un navegador cómo el siguiente:



Además los resultados de la prueba deben ser cómo el siguiente:





## Entregable parte 2

Para esta parte del laboratorio debe crear una mejor prueba donde se pueda navegar de llegar al formulario de crear un país y crear un país. Además haga los asserts necesarios para ver si los mensajes de confirmación aparecen. Debe entregar el proyecto del laboratorio de test además de un video donde enseñe la ejecución de la prueba.

Materiales :

1. <https://www.udemy.com/course/selenium-ga/learn/lecture/4640676#overview>
2. <https://www.guru99.com/selenium-csharp-tutorial.html>

## Referencias

- Materiales de cada parte
- Material basado en laboratorio basado por Edwin Brenes