Paso 0: Preparar entorno

Asegúrate de tener:

- .NET 8 SDK instalado
- Java 16+ instalado
- Navegadores instalados: Chrome y Edge
- PowerShell abierto como administrador (opcional para permisos)

Paso 1: Crear la carpeta del proyecto

Abre PowerShell y ejecuta:

Crea la carpeta principal

mkdir C:\Users\Jorge\SeleniumGridParallel

cd C:\Users\Jorge\SeleniumGridParallel

Crear proyecto de prueba

dotnet new classlib -n SeleniumGridParallel

cd SeleniumGridParallel

Paso 2: Agregar paquetes NuGet

Ejecuta los siguientes comandos para instalar todos los paquetes necesarios:

dotnet add package NUnit --version 4.4.0

dotnet add package NUnit3TestAdapter --version 5.2.0

dotnet add package Microsoft.NET.Test.Sdk --version 18.0.1

dotnet add package Selenium.WebDriver --version 4.38.0

dotnet add package Selenium.WebDriver.ChromeDriver --version 142.0.7444.17500

dotnet add package Selenium.WebDriver.MicrosoftDriver --version 17.17134.0

dotnet add package coverlet.collector --version 6.0.4

Paso 3: Configurar .csproj

Edita el archivo SeleniumGridParallel.csproj y reemplaza todo con:

<Project Sdk="Microsoft.NET.Sdk">

```
  <PropertyGroup>
    <TargetFramework>net8.0</TargetFramework>
    <LangVersion>latest</LangVersion>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
    <IsPackable>false</IsPackable>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="coverlet.collector" Version="6.0.4" />
    <PackageReference Include="Microsoft.NET.Test.Sdk" Version="18.0.1" />
    <PackageReference Include="NUnit" Version="4.4.0" />
    <PackageReference Include="NUnit3TestAdapter" Version="5.2.0" />
    <PackageReference Include="Selenium.WebDriver" Version="4.38.0" />
    <PackageReference Include="Selenium.WebDriver.ChromeDriver"
Version="142.0.7444.17500" />
    <PackageReference Include="Selenium.WebDriver.MicrosoftDriver"
Version="17.17134.0" />
  </ItemGroup>

  <ItemGroup>
    <Using Include="NUnit.Framework" />
  </ItemGroup>

</Project>
```

Paso 4: Crear archivo de pruebas

1. Borra Class1.cs
2. Crea UnitTest1.cs con este contenido:

```
using NUnit.Framework;
using OpenQA.Selenium;
using OpenQA.Selenium.Remote;
```

```
using OpenQA.Selenium.Chrome;
using System;

namespace SeleniumGridParallel
{
  [TestFixture("chrome")]
  [TestFixture("edge")]
  [Parallelizable(ParallelScope.Fixtures)]
  public class SeleniumGridTest
  {
    private RemoteWebDriver driver = null!;
    private readonly string browser;
    private readonly string gridUrl = "http://127.0.0.1:4444/wd/hub";

    public SeleniumGridTest(string browser) => this.browser = browser;

    [SetUp]
    public void Setup()
    {
      if (browser == "chrome")
      {
        var options = new ChromeOptions();
        options.AddArgument("--no-sandbox");
        options.AddArgument("--disable-dev-shm-usage");
        driver = new RemoteWebDriver(new Uri(gridUrl), options.ToCapabilities(),
TimeSpan.FromSeconds(60));
      }
      else if (browser == "edge")
      {
        var options = new OpenQA.Selenium.Edge.EdgeOptions();
        driver = new RemoteWebDriver(new Uri(gridUrl), options.ToCapabilities(),
TimeSpan.FromSeconds(60));
```

```
        }

        driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(5);
    }

    [Test]
    public void OpenGoogleDotCom()
    {
        driver.Navigate().GoToUrl("https://www.google.com");
        Assert.That(driver.Title.Contains("Google"), Is.True);
    }

    [Test]
    public void OpenWikipediaDotOrg()
    {
        driver.Navigate().GoToUrl("https://www.wikipedia.org");
        Assert.That(driver.Title.Contains("Wikipedia"), Is.True);
    }

    [TearDown]
    public void Teardown()
    {
        try { driver?.Quit(); } catch { }
        finally { driver?.Dispose(); driver = null!; }
    }
  }
}
```

Paso 5: Configurar Selenium Grid en Windows

1. Descarga Selenium Server 4.31.0: https://github.com/SeleniumHQ/selenium/releases

2. Descarga manualmente los drivers:

- ChromeDriver → coloca en C:\SeleniumDrivers\chromedriver.exe
- EdgeDriver → coloca en C:\SeleniumDrivers\msedgedriver.exe

3. Abrir terminal para el **Hub**:

java -jar "C:\ruta\a\selenium-server-4.31.0.jar" hub

4. Abrir otra terminal para cada **Node**:

java -Dwebdriver.chrome.driver="C:\SeleniumDrivers\chromedriver.exe" -Dwebdriver.edge.driver="C:\SeleniumDrivers\msedgedriver.exe" -jar "C:\ruta\a\selenium-server-4.31.0.jar" node --hub http://127.0.0.1:4444

Paso 6: Ejecutar los tests

Desde la carpeta del proyecto:

dotnet restore

dotnet build

dotnet test --logger "console;verbosity=detailed"

Ahora tus tests correrán en paralelo en Chrome y Edge usando Selenium Grid, con Assert.That compatible con NUnit 4 y drivers manuales.