

## Manejo de Alerts en <https://demoqa.com/alerts>

### 1) Preparar proyecto

- Paquetes NuGet recomendados:
  - Selenium.WebDriver
  - Selenium.WebDriver.ChromeDriver (o el driver que uses)
  - Selenium.Support
  - DotNetSeleniumExtras.WaitHelpers (para ExpectedConditions)
  - NUnit (u otro framework de test que uses)
- using necesarios en C#:

```
using System;  
using NUnit.Framework;  
using OpenQA.Selenium;  
using OpenQA.Selenium.Chrome;  
using OpenQA.Selenium.Support.UI;  
using SeleniumExtras.WaitHelpers;
```

### 2) Concepto clave

Las alertas son modales y *bloquean* la interacción con la página. Antes de interactuar con elementos de la página, debes manejar la alerta (aceptar/dismiss/leer/escribir). Para alertas que aparecen con retraso, usar WebDriverWait y la condición AlertIsPresent().

(Referencia de la página de demo con ejemplos de alertas). [demoqa.com](https://demoqa.com)

### 3) Código completo de ejemplo (NUnit) — archivo AlertsDemoTests.cs

Este código contiene 4 tests: Simple alert, Timer alert, Confirm alert (Aceptar/Dismiss) y Prompt (SendKeys).

```
using System;  
using NUnit.Framework;  
using OpenQA.Selenium;  
using OpenQA.Selenium.Chrome;  
using OpenQA.Selenium.Support.UI;  
using SeleniumExtras.WaitHelpers;
```

namespace DemoQA.Alerts

```

{

[TestFixture]

public class AlertsDemoTests

{

    private IWebDriver driver;

    private WebDriverWait wait;




[SetUp]

public void SetUp()

{

    var options = new ChromeOptions();

    // options.AddArgument("--headless"); // descomenta si quieres ejecutar sin UI

    driver = new ChromeDriver(options);

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

    wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

    driver.Navigate().GoToUrl("https://demoqa.com/alerts");

}






[TearDown]

public void TearDown()

{

    driver.Quit();

}






// 1) Simple Alert - click -> alert inmediato -> leer y aceptar

[Test]

public void SimpleAlert_Accept()

{

    // Localizador robusto: demoqa suele usar id="alertButton"

    var simpleBtn =.FindElementSafe(By.Id("alertButton"),

By.XPath("//button[text()='Click me']"));


```

```

simpleBtn.Click();

// Esperar a que aparezca la alerta
IAlert alert = wait.Until(ExpectedConditions.AlertIsPresent());
Console.WriteLine("Texto alerta simple: " + alert.Text);
alert.Accept();
// Aquí puedes añadir asserts si la página muestra resultado tras aceptar
}

// 2) Timer Alert - alerta aparece tras unos segundos (esperar explícitamente)
[Test]
public void TimerAlert_WaitAndAccept()
{
    // DemoQA tiene un botón de "On button click, alert will appear after 5 seconds"
    var timerBtn = FindElementSafe(By.Id("timerAlertButton"),
        By.XPath("//button[contains(text(),'Timer')] | //button[contains(text(),'after 5
seconds')]"));
    timerBtn.Click();

    // Esperar la alerta (puede tardar ~5 segundos)
    IAlert alert = wait.Until(ExpectedConditions.AlertIsPresent());
    Console.WriteLine("Texto timer alert: " + alert.Text);
    alert.Accept();
}

// 3) Confirm Alert - dismiss y accept
[Test]
public void ConfirmAlert_DismissAndAccept()
{
    var confirmBtn = FindElementSafe(By.Id("confirmButton"),
        By.XPath("//button[contains(text(),'Confirm')]"));

```

```
confirmBtn.Click();

IAlert confirmation = wait.Until(ExpectedConditions.AlertIsPresent());
Console.WriteLine("Confirm text: " + confirmation.Text);
// Dismiss (equivalente a cancelar)
confirmation.Dismiss();

// Opcional: volver a abrir y aceptar
confirmBtn.Click();
confirmation = wait.Until(ExpectedConditions.AlertIsPresent());
confirmation.Accept();
}

// 4) Prompt Alert - enviar texto y aceptar
[Test]
public void PromptAlert_SendKeysAndAccept()
{
    var promptBtn = FindElementSafe(By.Id("promptButton"),
        By.XPath("//button[contains(text(),'Prompt')] | //button[contains(text(),'prompt')]"));
    promptBtn.Click();

    IAlert prompt = wait.Until(ExpectedConditions.AlertIsPresent());
    Console.WriteLine("Prompt text: " + prompt.Text);

    // Enviar texto al prompt
    string texto = "Hola desde Selenium";
    prompt.SendKeys(texto);
    prompt.Accept();

    // Luego puedes leer en la página el resultado (demoqa muestra el texto en un
    // elemento)
}
```

```

// Ej: By.Id("promptResult") (si existe)
// var result = driver.FindElement(By.Id("promptResult")).Text;
// Assert.IsTrue(result.Contains("Hola desde Selenium"));

}

// ----- Helpers -----

// Intentar localizar por Id primero, si no existe usar el fallback XPath (robusto ante cambios).
private IWebElement FindElementSafe(By primary, By fallback)
{
    try
    {
        return wait.Until(d => d.FindElement(primary));
    }
    catch (WebDriverTimeoutException)
    {
        // intentar fallback
        return wait.Until(d => d.FindElement(fallback));
    }
    catch (NoSuchElementException)
    {
        return driver.FindElement(fallback);
    }
}
}

```

#### 4) Notas y buenas prácticas

1. **Usa AlertIsPresent()**: evita NoAlertPresentException y UnhandledAlertException para alertas que aparecen con retardo.

2. **No interactúes con la página mientras la alerta esté presente:** siempre switchTo().Alert() y Accept()/Dismiss() antes de seguir.
3. **Localizadores robustos:** demoqa suele exponer id para cada botón (alertButton, timerAlertButton, confirmButton, promptButton) — si el id falla usa XPath por texto del botón. (Referencia: la página Alerts de demoqa). [demoqa.com](http://demoqa.com)
4. **Prompt (SendKeys):** sólo funciona si el navegador/alerta soportan entrada — algunos navegadores en modo headless podrían comportarse distinto.
5. **Tiempo de espera para la alerta con temporizador:** demoqa usualmente hace aparecer la alerta a los ~5 segundos; ajusta WebDriverWait si tu entorno es más lento.
6. **Manejo de excepciones:** captura NoAlertPresentException, UnhandledAlertException y WebDriverTimeoutException para tests más robustos y mensajes claros.

## 5) Ejemplo de verificación (assert) — Prompt

Después de aceptar el prompt, demoqa suele mostrar el texto en un elemento id="promptResult" (o similar). Añade una comprobación para confirmar que el texto enviado aparece en la página:

```
// tras prompt.Accept();  
  
var result = wait.Until(d => d.FindElement(By.Id("promptResult"))).Text;  
  
Assert.IsTrue(result.Contains("Hola desde Selenium"), "El texto enviado al prompt no aparece en la página.");
```

## 6) Problemas comunes y soluciones rápidas

- **UnhandledAlertException al intentar operar en la página:** significa que aún hay una alerta abierta. Haz driver.SwitchTo().Alert().Accept() o Dismiss() antes.
- **Alert no aparece (temporizador):** aumenta el tiempo del WebDriverWait.
- **SendKeys no escribe en prompt:** asegúrate de que no hay extensiones o configuraciones del navegador que bloqueen modales; prueba con navegador no headless.
- **Los IDs cambian:** usa XPath por texto del botón (más robusto si la UI muestra siempre el mismo texto).