

Análisis de caso

Utilización de hooks

Ashley Rodriguez.

Situación inicial

El equipo de pruebas de la empresa EZTest quiere automatizar el login de su aplicación web. En cada escenario se abre y cierra el navegador de forma repetitiva, lo que genera mucho código duplicado. Para solucionar esto, han decidido usar Hooks de Cucumber y así concentrar esa lógica en un solo lugar.

1. Crea un archivo. feature sencillo:

```
> test > resources > features > Login.feature
Feature: Login

  Scenario: Abrir Google
    Given que abro la página de Google
    Then verifico que el título contiene "Google"
```

2. Definir Hooks:

```
2
3 public class Hooks {
4
5     @Before
6     public void openBrowser() {
7         // Configura EdgeDriver automáticamente
8         WebDriverManager.edgedriver().setup();
9
10        // Opciones del navegador
11        EdgeOptions options = new EdgeOptions();
12        options.addArguments("--start-maximized");
13
14        // Crea el driver y lo guarda en DriverFactory
15        WebDriver driver = new EdgeDriver(options);
16        DriverFactory.setDriver(driver);
17
18        // Abre la URL base
19        driver.get(url:"https://www.saucedemo.com/");
20    }
```

```

    }

    @After
    public void closeBrowser() {
        // Cierra el navegador y limpia el driver
        DriverFactory.cleanup();
    }
}

```

3. Steps limpios:

```

src > test > java > com > ejemplo > J StepsLogin.java > ...
1  package com.ejemplo;
2
3  import static org.junit.jupiter.api.Assertions.assertTrue;
4  import org.openqa.selenium.WebDriver;
5
6  import com.ejemplo.support.DriverFactory;
7
8  import io.cucumber.java.en.Given;
9  import io.cucumber.java.en.Then;
10
11 public class StepsLogin {
12
13     private WebDriver driver;
14
15     public StepsLogin() {
16         this.driver = DriverFactory.getDriver();
17     }
18
19     @Given("que abro la página de Google")
20     public void que_abro_la_pagina_de_google() {
21         driver.get(url:"https://www.google.com");
22     }
23
24     @Then("verifico que el título contiene {string}")
25     public void verifico_que_el_titulo_contiene(String tituloEsperado) {
26         String titulo = driver.getTitle();
27         assertTrue(titulo.contains(tituloEsperado),
28             "El título esperado \"" + tituloEsperado + "\" no coincide con \"" + titulo + "\"");
29     }
30 }
31

```

4. Ejecutar y comprobar:

```
Administrador: Símbolo del sistema
[INFO] --- resources:3.3.1:testResources (default-testResources) @ hook ---
[INFO] Copying 1 resource from src\test\resources to target\test-classes
[INFO] --- compiler:3.10.1:testCompile (default-testCompile) @ hook ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 4 source files to C:\Users\ashle\OneDrive\Desktop\hook\target\test-classes
[INFO] --- surefire:3.2.5:test (default-test) @ hook ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Results:
[INFO] Tests run: 0, Failures: 0, Errors: 0, Skipped: 0
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.364 s
[INFO] Finished at: 2025-08-26T00:05:56-04:00
[INFO] -----
C:\Users\ashle\OneDrive\Desktop\hook>
```

no logre que arrojara los resultados de los steps ni abría una pagina web