P08- Pruebas de aceptación

Indicaciones sobre las soluciones

Dado que hay ciertos errores que cometéis que hemos visto son bastante recurrentes, os comentamos algunos de ellos para que los tengáis en cuenta.

POM.XML

- Todos los tests se ejecutan con el plugin ***surefire*** (es el único plugin que necesitáis, además del plugin ***maven-compiler-plugin***)
- En el pom únicamente debéis tener las dependencias ***junit-jupiter-engine***, y ***selenium-java*** (está en las transparencias)

TESTS JUNIT

- Tenéis que usar en todas las clases que contienen los tests un método anotado con @BeforeEach y otro con @AfterEach (este último para cerrar el navegador)
- TODOS los assert se implementan en los tests, no en las PO

EJERCICIO 1:

- Los tests contienen código webdriver. Se implementan en src/test/java y no es necesario implementar nada en src/main/java
- El método submit() sólo lo usaremos para los formularios. Dicho método envía los datos al servidor y espera a que se cargue la nueva página. NO uséis el método click() para pulsar sobre el botón de un formulario (este método no espera a la carga de la nueva página).

EJERCICIO 2:

- Los tests NO contienen código webdriver que dependa del código html.
- Los atributos de las clases que representan nuestras Page Object (PO) son únicamente de tipos de la librería Webdriver.
- Para cada elemento de tipo WebElement podéis usar el localizador que creáis conveniente. Los objetos de tipo WebElement se instancian en el constructor de la Page Object (siempre que sea posible).
- Las Page Object se crean con new(), y la creación de nuevas páginas debeb estar en los métodos de las Page Object y no en los tests. Si la ejecución de un método de la PO provoca que se acceda a una nueva página, dicho método devolverá la nueva página.
- En la clase que contiene los tests, NO debéis usar más atributos de los necesarios, que básicamente son el objeto de tipo WebDriver, y los objetos que representan las Page Objects.

EJERCICIO 3:

- Los tests NO contienen código webdriver que dependa del código html, eso significa que tampoco pueden contener objetos de tipo Alert.
- Los atributos de las clases que representan nuestras Page Object son únicamente de tipos de la librería Webdriver.
- Las Page Object se crean siempre con la clase PageFactory, y sus atributos están anotados con @FindBy.

PARA LOS EJERCICIOS 2 y 3:

- Las page object NO se instancian en el código del driver, excepto la primera page object. El resto de páginas se crean en el código de las page object, como resultado de la ejecución de sus métodos, tal y como se indica en las transparencias de teoría (ver tr. 23 y 24)
- Todas la PO se implementan en src/main/java