UI tests con Selenium

Guillermo Carrera Trasobares

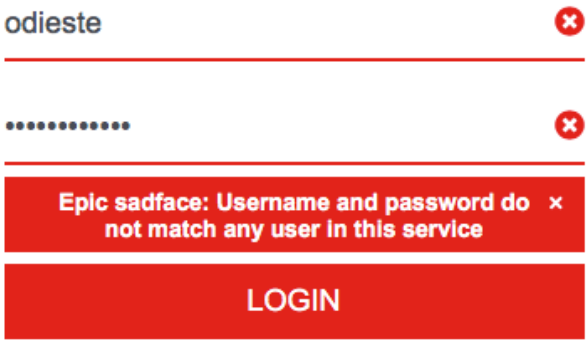
Repositorio: https://github.com/gcarrerat/pruebas_selenium

El ejercicio se realizará usando https://www.saucedemo.com. Las páginas simulan la funcionalidad habitual de un servidor web usual. Otra cosa es que no funciona muy bien.

A la hora de comprobar que el ejercicio está bien hecho, no implementéis varios tests dentro del mismo caso de prueba, ya que el servidor puede dejar de funcionar correctamente. Es preferible que cada test sea independiente de los demás, y que antes de lanzar un test recarguéis la página (cosa bastante razonable, ya que se supone que cada test debería tener su fixture).

Parte A - Los tests a realizar son los siguientes:

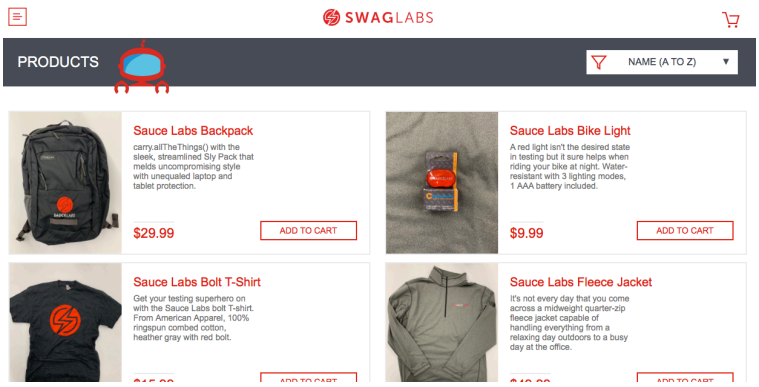
1. Intente acceder (**Sign in**) con un usuario incorrecto. En este caso, deberá recibir el siguiente mensaje de error:



Nota: El caso de prueba no tiene como objetivo mostrar la pantalla anterior, sino verificar que es la pantalla que se obtiene mediante el chequeo de alguno de sus elementos. **Por ejemplo, chequear que el texto "Username and password do not match" está presente en la página.**

Test: LoginIncorrectUser

2. Intente acceder con un usuario/password correcto. En este caso, deberá comprobar que se accede a la página principal del servidor.



Test: LoginValidUser

3. Acceda a la opción “About” del menú lateral izquierdo. Deberá ser redirigido a la página https://saucelabs.com.

   Test: AboutMenuRedirection

4. Seleccione el *item* “Sauce Labs Backpack”. En la página de detalle del producto, añádalo al *shopping cart*. Compruebe que el *item* se ha añadido correctamente.

   Test: BackpackShoppingCart

5. Repita el caso (4). Vuelva atrás (“Back to products”) y añada el *item* “Sauce Labs Bolt T-Shirt” al *shopping cart*. Compruebe que el *shopping cart* contiene dos *items*.

   Test: ShoppingCartTwoItems

6. Repita el caso (4) y elimine el *item* “Sauce Labs Backpack” del *shopping cart*. Compruebe que el *item* ha sido eliminado.

   Test: ShoppingCartRemoveOneItem

7. Repita el caso (5) y seleccione “Checkout”. Compruebe que accede a la página de petición de información.

   Test: CheckoutTwoElements

8. Repita el caso (7) y seleccione “Continue”. Compruebe que el total (antes de impuestos) se ha calculado correctamente.

   Test: ContinueAndVerifyCheckout

9. Repita el caso (8) y seleccione “Finish”. Compruebe que en la siguiente interacción con el servidor el *shopping cart* está vacío.

   Test: EmptyShoppingCart

10. Escoja la opción de ordenación “Price (low to high)”. Compruebe que efectivamente los items están ordenados por precio.

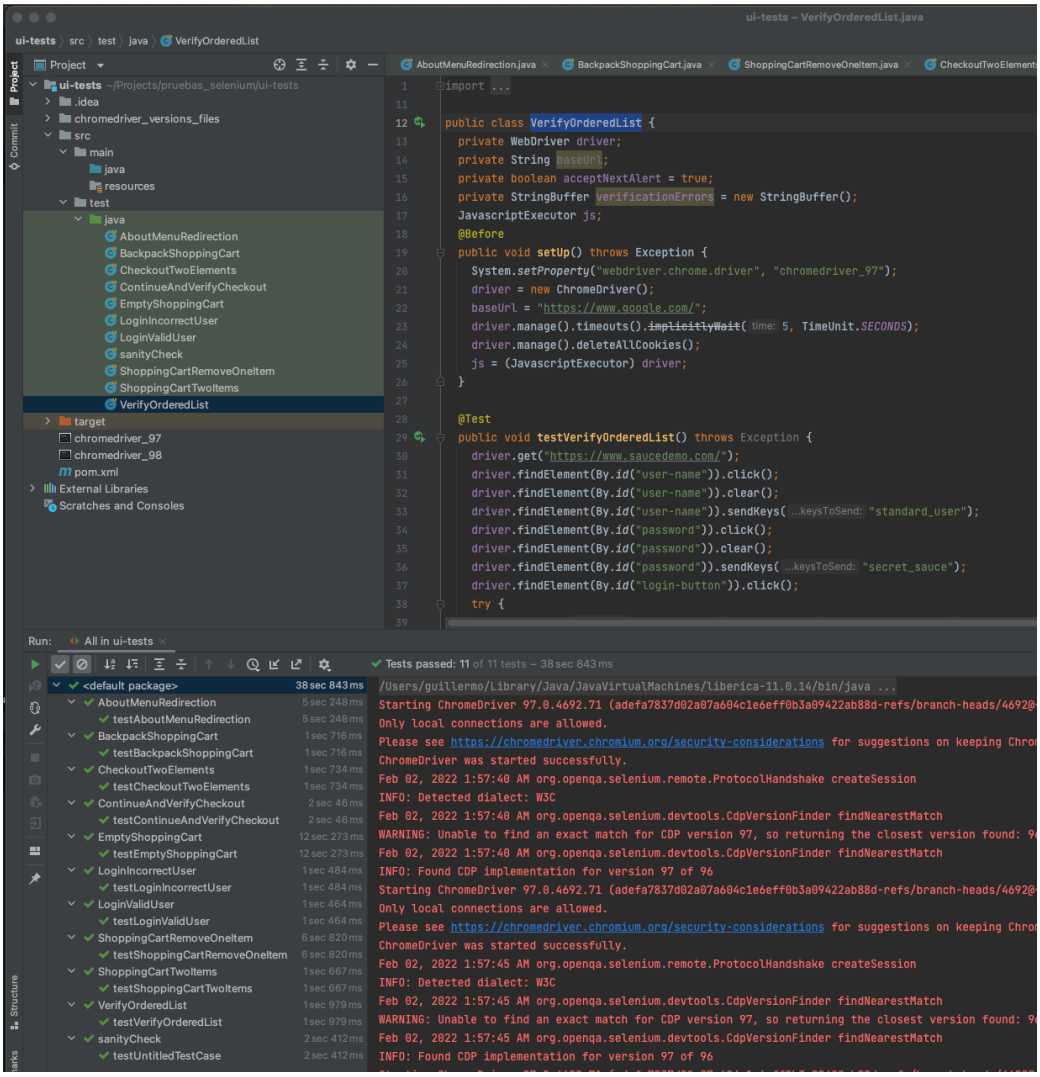
    Test: VerifyOrderedList

Parte B - Implemente los casos de prueba *WebDriver*. Describa brevemente los cambios principales que ha tenido que hacer al código exportado por Selemium IDE/Kantalon (si es que ha habido alguno). Si no quieren repetir código, pueden crear *helper methods*.

Cambios realizados al código:

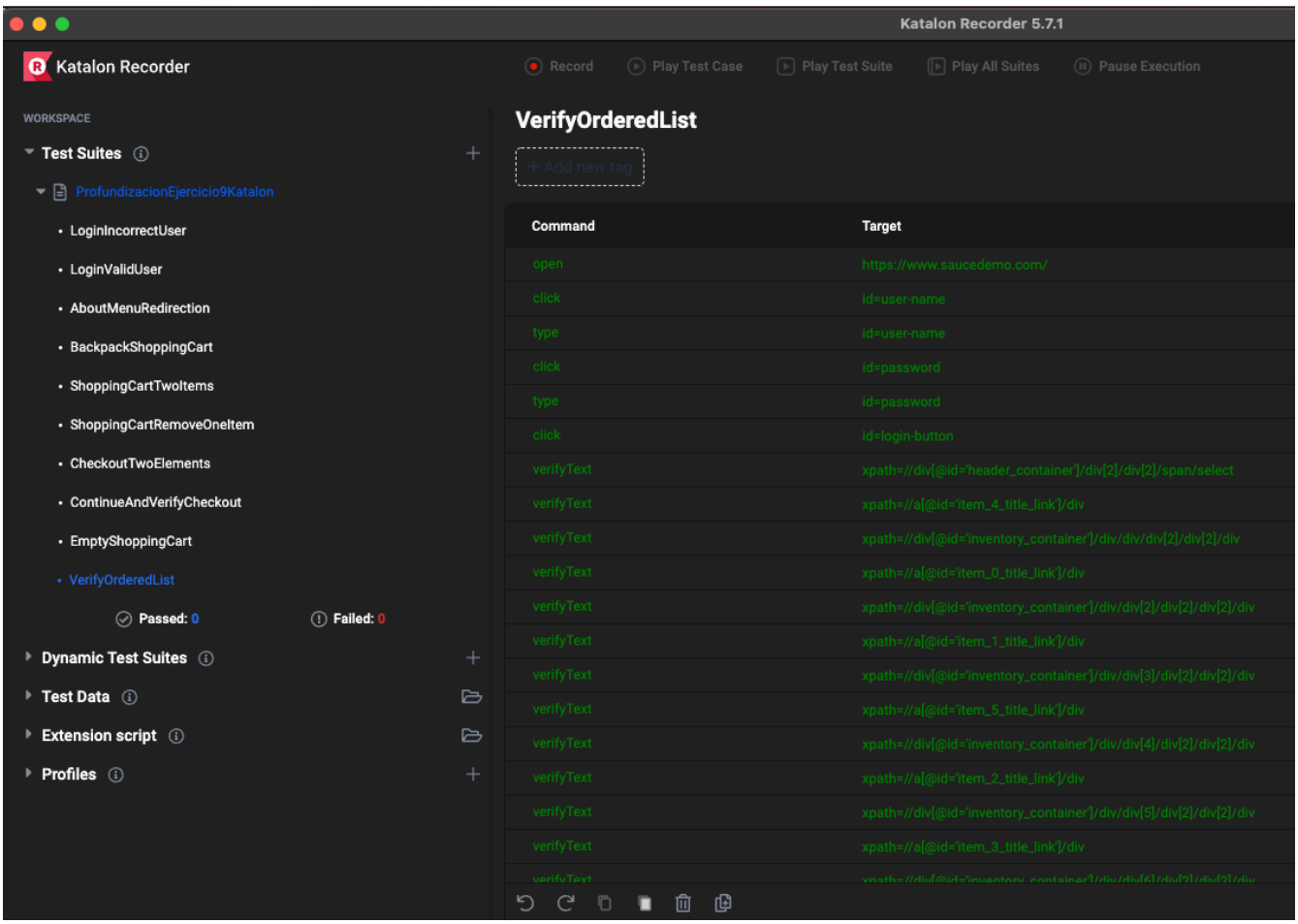
- Configurar los test para que utilicen el webdriver chromedriver_97 (versión Mac m1)
- Cambiar el timeout de 60 segundos a 5 segundos (importante porque en algunas comprobaciones para saber si un elemento existe, si no hay acciones posteriores el test espera el timeout completo antes de validarlo)
- Añadida la opción deleteAllCookies() al principio de cada test (Los elementos del carrito no se borran entre tests sucesivos)
- Corregidos errores de mayúsculas/minúsculas que hacen fallar los test al evaluar texto en la página
- Corregido error de assertEquals en elementos de un drop-down en el que los elementos no están separados por saltos de línea
- Modificado el select del dropdown para que utilice texto visible en lugar de label (la página no cumple las normas de accesibilidad en el drop-down y por lo tanto no se puede utilizar label o aria-label para seleccionar los elementos, se utiliza directamente el texto visible o el campo value)

Proyecto

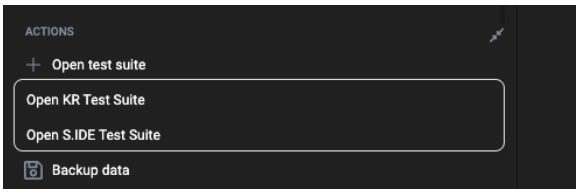


El proyecto (**ui-tests**) se puede importar directamente como un proyecto Maven. Los tests se pueden ejecutar en batch dado que son independientes y las cookies se eliminan al principio de cada ejecución.

Kantalon Recorder Tests



Se incluye el fichero **ProfundizacionEjercicio9Katalon.krecorder** que contiene los tests en formato KR. La suite con los 10 tests se puede importar desde el menú de acciones de la herramienta Katalon Recorder. Se recomienda borrar las cookies entre ejecuciones dado que el carrito conserva los elementos.



Una vez abierto el fichero, los tests se pueden exportar a múltiples formatos: