Documentación: automatización

Selenium: desafío avanzado

En primer lugar generé un método para inicializar un webdriver para el browser de Firefox y forzar al driver a permanecer en espera activa durante 1 segundo durante la ejecución de cada función. Siguiendo la misma lógica generé una función para finalizar la ejecución del browser.

```
    @BeforeClass
    public static void inicializarDriver() {
        driver = new FirefoxDriver();
        driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    }

    @AfterClass
    public static void cerrarDriver() {
        driver.quit();
    }
```

Desafío 1 - grabar y ejecutar en código

Primero realice todas las acciones manualmente a través del browser, utilizando como herramienta complementaria el selenium building tool. Esto me permitió llevar un registro de las acciones realizadas y registrar las rutas de acceso a los distintos elementos.

```
@Test
    public void parte1() {
        Accedo al sitio en el browser por medio del web driver,
        driver.get("http://demo.opencart.com");
       Dentro del sitio busco el product a través de la function find element a partir de su xpath.
        WebElement iphone = driver.findElement(By.xpath("//div[@id='content']/div[2]/div[2]/div/div[1]/a/img"));
        Accedo al elemento.
        iphone.click();
        Este elemento del html contiene el nombre del producto, así que simplemente fue necesario acceder al
        nombre y almacenarlo en una variable.
        String iPhoneTitle = driver.findElement(By.xpath("//div[2]/div/div/div[1]/div[2]/h1")).getText();
        Valido que el nombre sea efectivamente el correcto.
         assertEquals(iPhoneTitle,"iPhone");
}
```

Desafío 2 - parametrizar la prueba

```
    @Test
    public void parte2(){
```

```
        Accedo al sitio.
        driver.get("http://demo.opencart.com");
         Cargo un listado de los 5 productos a buscar desde un archivo externo.
         String[] products = RWfile.readFile(sourceProducts);
                for (int i=0; i<products.length; i++){
                        Selecciono el elemento correspondiente a la barra de búsqueda.
                        WebElement actualProduct = driver.findElement(By.xpath("//div[@id='search']/input"));
                        Esta es una función para simular la interacción con el teclado.
                        actualProduct.sendKeys(products[i]);
                        Simulo clickear aceptar la búsqueda.
                        driver.findElement(By.xpath("//header/div/div/div[2]/div/span/button")).click();
                        Como en el primer desafío, busque el elemento con el nombre del producto.
                        String pageText =
                        driver.findElement(By.xpath("//div[2]/div/div/div[4]/div/div/div[2]/h4/a")).getText();
                        Valido que el nombre sea el correcto.
                        assertEquals(products[i],pageText);
                }
    }
```

Desafío 3 - obtener toda la lista de productos

```
    @Test
    public void parte3(){
        Accedo al sitio
        driver.get("http://demo.opencart.com");
        Realizo una búsqueda de todos los productos ingresando el caracter espacio.
        WebElement search = driver.findElement(By.xpath("//div[@id='search']/input"));
         search.sendKeys(" ");
         driver.findElement(By.xpath("//header/div/div/div[2]/div/span/button")).click();
        Inicializo una lista para almacenar todos los productos encontrados
         List<String> allProducts = new LinkedList<String>();
        Verifico que exista una siguiente pagina. La idea es recorrer todas las páginas y en cada página recorrer todos los elementos
        para insertarlos en la lista y posteriormente persistir los elementos de la lista en un archivo.
         while(driver.findElements(By.xpath("//ul[@class='pagination']//a[.='>']")).size() != 0){
                Inspeccionando el html encontre que todos los elementos pertenencen a la clase "product-thumb", por lo que
                realice una búsqueda de todos los elementos y los guarde en una lista.
                 List<WebElement> bar = driver.findElements(By.className("product-thumb"));
                        for(WebElement current : bar){
                                El elemento “img” contiene un atributo con el nombre del producto.
                                String actualProduct =
                                current.findElement(By.tagName("img")).getAttribute("title");
                                Lo agrego a la lista.
                                allProducts.add(actualProduct);
                        }
                        Esta función podría no ser necesaria con la verificación previa, pero lo que hace es verificar que antes de
                        acceder a la siguiente página el elemento exista o se encuentre habilitado.
                        WebDriverWait waiter = new WebDriverWait(driver, 5000);
                        waiter.until(
                        ExpectedConditions.presenceOfElementLocated(By.xpath("//ul[@class='pagination']//a[.='>']
                        ")) );
                        driver.findElement(By.xpath("//ul[@class='pagination']//a[.='>']")).click();

        }
                Genero un nuevo archivo con los productos encontrados
                RWfile.writeFile(destProductsFound, allProducts);
                File fAllProducts = new File(sourceAllProducts);
                File fProductsFound = new File(destProductsFound);
```

```
            try {
                    Valido que el archivo contenga todos los productos del sitio
                    assertTrue(Files.equal(fAllProducts, fProductsFound));
            } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            }
    }
```

Desafío 4 - verificar que se puede acceder a TODOS los productos

```
@Test
    public void parte4() {
        Accedo al sitio
        driver.get("http://demo.opencart.com");
        Leo un archivo con el listado de productos.
        String[] allProducts = RWfile.readFile(destProductsFound);
        Busco el producto por medio de la barra de búsqueda.
        WebElement search = driver.findElement(By.xpath("//div[@id='search']/input"));
         search.sendKeys(productToSearch);
         driver.findElement(By.xpath("//header/div/div/div[2]/div/span/button")).click();
        Accedo al producto encontrado.
         List<WebElement> bar = driver.findElements(By.className("product-thumb"));
         WebElement pageElement = bar.get(0);
        Guardo el nombre en una variable.
         String elementName = pageElement.findElement(By.tagName("img")).getAttribute("title");
        Valido que el producto forme parte del listado
         assertTrue(Arrays.asList(allProducts).contains(elementName));
    }
```