



Proyecto: Website zonafit.com.co

Plan de pruebas de automatización

Historia de revisiones

Versión	Autor(es)	Descripción	Fecha
1.0	Nilton Rodríguez	Creación del documento	2 enero de 2022

Tabla de contenido

1. Introducción	4
2. Alcance	5
2.1. Casos de prueba.....	5
3. Roles y Responsabilidades.....	6
4. Riesgos y Planes de contingencia	7
5. Ambiente y Herramientas de prueba.	8
5.1. Herramientas de prueba	8
5.2. Arquitectura del framework de automatización	8
5.3. Ambiente de pruebas	9
6. Criterios de entrada y salida.....	10
6.1. Criterios de entrada	10
6.2. Criterios de salida	10
7. Planificación de la ejecución de las pruebas.....	11
8. Reporte de pruebas	12

1. Introducción

Zonafit.co es el website de una tienda en línea especializada en productos de nutrición deportiva. El alcance para el actual plan de pruebas es el carrito de compras y la realización de pedidos, el alcance del detalle de los productos y la forma de pago será descrito en el la sección Alcance. Dado el actual acceso al proyecto, las pruebas se desarrollarán sobre el ambiente de producción utilizando el navegador Google Chrome en sistemas operativos Windows y MacOS. Para la automatización de las pruebas se utilizará como patrón de diseño Page Object Model (POM), se desarrollará con Java, usando el framework Selenium, la librería Page Factory y Cucumber. Entre los riesgos se encuentra la variabilidad en la existencia de los productos, la diferencia entre productos con selección de sabores y los posibles cambios por el fin de la temporada navideña. Como plan de contingencia ante los riesgos, se utilizará productos con selección de un sabor, para mantener estabilidad y uniformidad en las pruebas.

2. Alcance

Se realizarán pruebas de caja negra (automatizadas) las funcionalidades del carrito de compras, selección de productos y generación de orden de pedidos en el navegador Google Chrome. Dada la variedad en opciones de sabores, solo se han incluido en la aleatorización de productos aquellos que requieren la selección de una única opción de sabor. Se utilizará como opción de corresponsal bancario para los pagos.

2.1. Casos de prueba

Feature: Order Products

As

a client

I WANT TO

make an order

SO THAT

I can buy the products of the store.

Background:

Given a client that want to buy in zonafit

Scenario: Clear the cart

When the client select a product

But the client go to the cart and confirm to clear it

Then the client see a message with the cart empty

Scenario: Make an order

When the client select products and confirm the cart

And the client fill the billing form and confirm the purchase

Then the client see the billing and payment information

3. Roles y Responsabilidades

ROLES	RESPONSABILIDADES
Manger de QA	Planificación y monitoreo de las pruebas automatizadas. Reporte de defectos
Analista QA	Diseño e implementación de las pruebas. Ejecución de las pruebas automatizadas. Reporte de resultados de las pruebas.
Product Owner / Stakeholders	Toma de Desiciones.

4. Riesgos y Planes de contingencia

No	Riesgos	Probabilidad de ocurrencia (1-4)	Impacto (1-4)	Severidad	Plan de contingencia
1	El website implement cambios por el fin de la temporada navideña.	2	4	8	
2	Los productos no cuentan con existencia.	2	3	6	La selección es aleatoria en la opción de productos, donde se muestran productos con existencias disponibles.
3.	Los productos cuentan con diversas opciones respecto a los sabores a elegir.	2	3	6	Se seleccionarán productos que solo requieran elegir 1 opción de sabor.

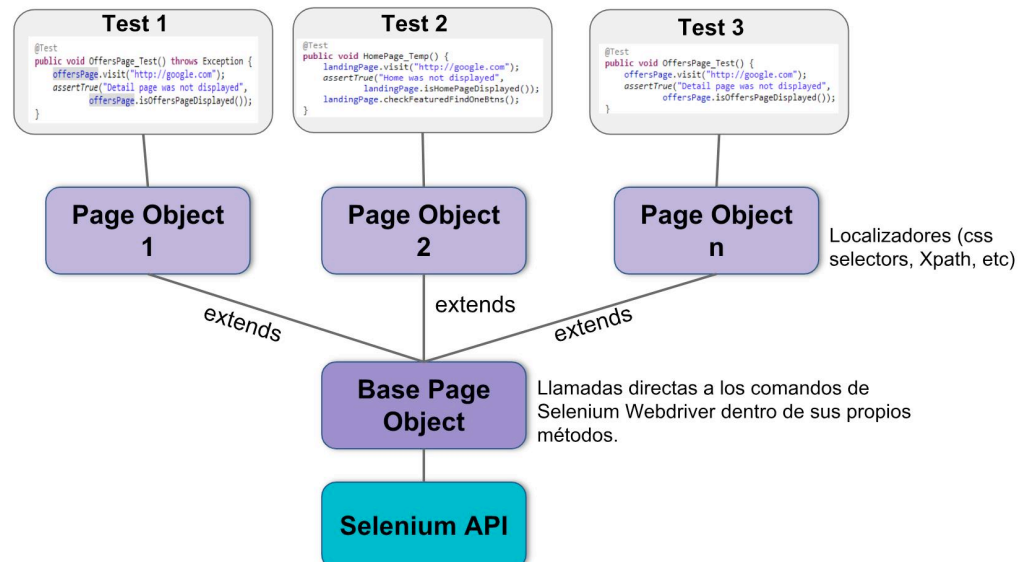
5. Ambiente y Herramientas de prueba.

5.1. Herramientas de prueba

Herramienta	Función
Selenium WebDriver	API para automatizar sistemas Web.
Page Factory	Librería que optimiza la implementación de POM
Cucumber	Herramienta de BDD para desarrollar casos de prueba y el reporte de las pruebas.
Gradle	Creación de la estructura de proyectos y uso e importación de librerías.
Chromedriver	Crea una instancia del navegador Google Chrome.

5.2. Arquitectura del framework de automatización

Arquitectura de Testing Automatizado (Selenium WebDriver)



Se utilizará el patrón Page Object Model (POM) para mapear las páginas del sistema a clases page que permitan aislar las acciones de las diferentes páginas y agrupar en una misma clase los webElements de una página y las acciones que se pueden llevar a cabo.

El POM permite concentrar los localizadores en las clases “Page”, de manera que cuando el sistema cambia y es necesario actualizar el código de dichas estrategias de localización de los webElements, solo se tendrá que hacer una vez en la clase “Page”, haciendo que los test no necesiten ningún cambio, a menos que la lógica del funcionamiento haya cambiado o se agregaran o eliminaran funcionalidades al sistema.

Page Factory es una librería que provee una sintaxis y semántica particular a partir del uso de anotaciones (@) para crear los objetos de la página web (WebElements) estáticos. A pesar del cambio en la sintaxis, Page Factory aplica el uso de localizadores y puede usarse en conjunto con el modelo POM, principalmente para elementos dinámicos.

5.3. Ambiente de pruebas

Navegadores	Google Chrome
Sistemas Operativos	Windows, MacOS.

6. Criterios de entrada y salida

6.1. Criterios de entrada

Las funcionalidades e y las funcionalidades han sido probadas manualmente.

El framework se encuentra instalado y listo para la operación.

El website se encuentra disponible.

Los defectos críticos encontrados durante pruebas manuales han sido resueltos y cerrados.

6.2. Criterios de salida

Ejecución de todos los casos de prueba automatizados.

Se logra suficiente cobertura de los requerimientos y funcionalidades bajo las pruebas.

Ningún defecto de severidad alta se encuentra abierto.

7. Planificación de la ejecución de las pruebas

Lista de funcionalidades a ser automatizadas.

No.	Funcionalidad	Comentarios
1	Vaciar el carrito de compras.	
2	Realización de una orden de compra completa.	Es necesario que haya productos disponibles.




Es necesario que las funcionalidades a automatizar se desarrollen, implementen y prueben manualmente para que tengan un nivel determinado de estabilidad cuando comienzan las tareas de automatización.

Dado el carácter cambiante propio de la dinámica del websie, se especifica que debe haber productos disponibles, de los cuales se elegirá únicamente aquellos que únicamente soliciten elegir un sabor, puesto que es la estabilidad mínima para garantizar la correcta ejecución de las tareas de automatización.

8. Reporte de pruebas

El reporte automático de las pruebas se obtendrá a través de Cucumber y será publicado en un link generado directamente por esta herramienta. El reporte informará sobre los resultados de la ejecución de cada caso de prueba, incluirá las pruebas que pasaron y las pruebas que fallaron, la tasa de éxito y el tiempo transcurrido.

2 PASSED

100% passed 2 executed	9 minutes ago Last Run	1 minute, 20.59 seconds Duration
 Mac OS X	 Java HotSpot(TM) 64-Bit Server VM 25.111-b14	 cucumber-jvm 6.11.0

Delete Report

✓ file:///Users/niltonrodriguez/Documents/Programming/SofkaU/Introduccio%CC%81n_a_la_automatizacio%CC%81n/RetoPageFactory/src/test/resources/features/zonafit_form.feature

[Report](#) [Edit](#)

Feature: Order

As a client I WANT TO make an order SO THAT I can buy the products of the store.

Background:

✓ **Given** a client that want to buy in zonafit

Scenario: Clear the cart

✓ **When** the client select a product

✓ **But** the client go to the cart and confirm to clear it

✓ **Then** the client see a message with the cart empty

Scenario: Make an order

✓ **When** the client select products and confirm the cart

✓ **And** the client fill the billing form and confirm the purchase

✓ **Then** the client see the billing and payment information