Proyecto: OrangeHRM

Por: Juan Daniel Gonzalez Puerta

Estrategia de Prueba Automatizadas
Training sofkau

Historia de revisiones

Versión	Autor(es)	Descripción	Fecha
1.0	Juan Daniel Gonzalez	Estrategias de pruebas POM	Octubre 2021

Tabla de Contenidos

1. Introducción

En esta Estrategia para la realización de pruebas automatizadas se describe el alcance de las pruebas, el ambiente de pruebas, los recursos necesarios, las herramientas a utilizar para la ejecución de las pruebas del sitio web OrangeHRM.

2. Alcance

Se realizarán pruebas de caja negra (automatizadas) a las funcionalidades asignadas para el reto del curso “automatización de pruebas”.
En este caso el alcance son los módulos:

- **inicio de sesión**
- **Módulo administrativo Para el menú con ruta: Admin - User Management – Users.**

3. Análisis de riesgos

Criterios de aceptación:

- **CA001:** Inicio de sesión exitoso.
- **CA002:** Inicio de sesión donde el password es inválido.
- **CA003:** Inicio de sesión donde el usuario es inválido.
- **CA004:** Inicio de sesión donde no se introducen credenciales.
- **CA005:** Verificar la funcionalidad de reseteo en las búsquedas mediante una combinación de búsqueda cualquiera.

Antes de empezar con la ejecución de pruebas se empezara con un testeo de software (smoke test) para asegurarnos de que el ambiente, funcionalidades básicas y críticas del programa funcionan correctamente, este será un testeo rápido y no exhaustivo, por lo tanto debemos enfocarnos en reconocer cuales son las funciones básicas y críticas, para este caso se verificara que se abra de manera correcta el sitio web y que la información desplegada sea legible, entendible, ordenada y coherente con lo que se esperaría encontrar en un sitio web.

Vamos a realizar las pruebas en cierto orden en base a un análisis de riegos, donde se calculara el riesgo como: **riesgo = probabilidad*impacto**, y el orden será del mayor riesgo al menor.

También se tendrá en cuenta la cobertura que tendremos al realizar estas pruebas en cuanto a los navegadores y dispositivos móviles, sin embargo no se utilizaran todas las opciones del mercado ya que son muchas, nos enfocaremos en Google Chrome en Windows 10.

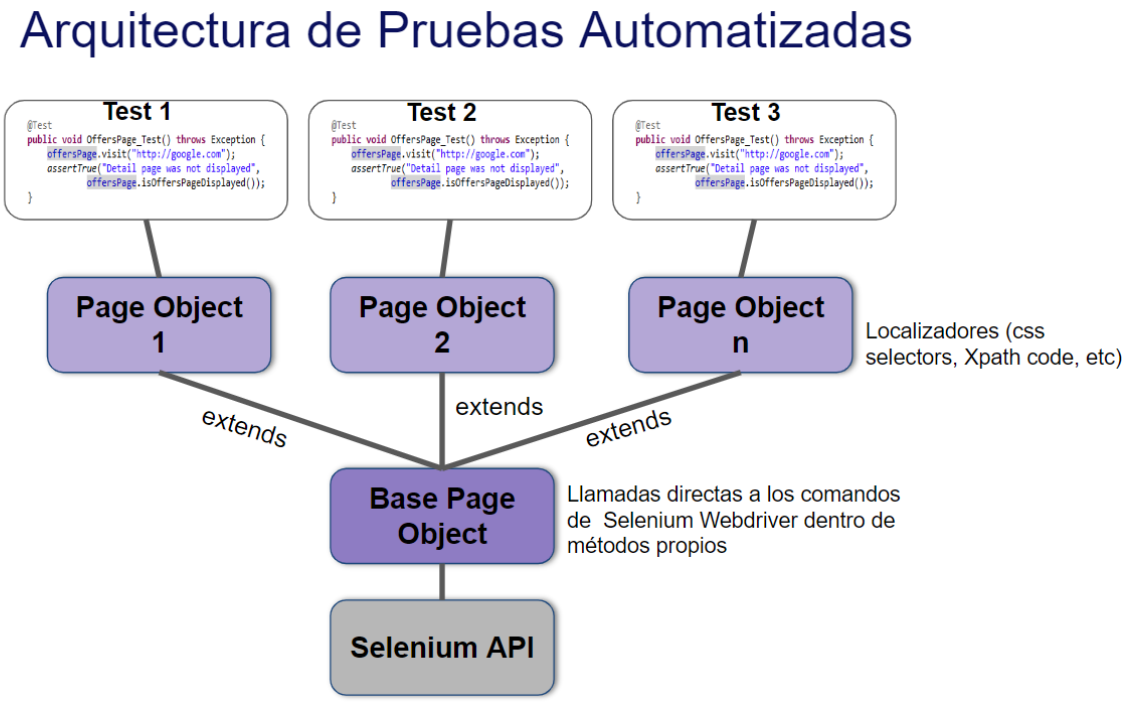
No	Criterios de aceptación.	Probabilidad (1-4)	Impacto (1-4)	Riesgo
1	**CA001**	1	4	4
2	**CA002**	1	4	4
3	**CA003**	1	4	4
4	**CA004**	1	4	4
5	**CA005**	1	2	2

4. Ambiente y Herramientas de Pruebas

4.1 Herramientas de Pruebas

Herramienta	Función
Selenium WebDriver	API para automatizar sistemas Web
JUnit testing framework	Ejecución y Reporte de las pruebas
Gradle	Creación de la estructura de proyectos y uso e importación de librerías
Chromedriver	Crea una instancia del navegador Chrome

4.2 Arquitectura del framework de automatización



Utilizaremos 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 a la vez agrupar todos los webElements de una página y las acciones que se pueden llevar a cabo, en una misma clase.

La clase “Base” permite aislar todo el framework de la versión del API de Selenium WD que estemos utilizando. De esta forma si hay algún cambio en los comandos del API no tenemos que cambiar todas las clases sino solo la clase “Base”.

El Page Object Model también nos ayuda a concentrar los localizadores en estas clases “Page”, de forma que cuando el sistema cambia y es necesario actualizar el código de los css selectors, xpath o lo que hayamos utilizado para localizar los webElements, solo tenemos que cambiarlo una sola vez en la clase “Page” y los “Tests”, que son el último nivel, no necesitan ningún cambio (a menos que haya cambiado la lógica de funcionamiento y dentro de los cambios se hayan eliminado o agregado funcionalidades al sistema).