

Proyecto Transversal: Calidad de Software Aplicada a Sistemas de Dominio Específico

Modalidad de trabajo

- Equipos de 4 o 5 estudiantes.

Automatización y Evaluación de Atributos de Calidad

Sitio Web Objetivo: <https://opencart.abstracta.us/>

Tecnologías Requeridas:

- Java
- Selenium WebDriver
- Maven
- Apache POI (manejo de Excel)
- Patrón de diseño Page Object Model (POM)

Objetivo del Proyecto:

Desarrollar un conjunto de pruebas automatizadas sobre la tienda de demostración de OpenCart utilizando buenas prácticas de automatización. El objetivo es evaluar la capacidad técnica del estudiante en el diseño de pruebas, estructura de framework, selección de selectores adecuados, sincronización, validaciones, y manejo de datos externos.

Casos de Prueba a Implementar:

1. Registro de Usuario:

- Leer datos de usuarios desde un archivo Excel (inputData.xlsx, hoja UsuariosRegistro).
- Completar el formulario de registro en OpenCart.
- Verificar que el mensaje de éxito se muestre.

2. Inicio de Sesión:

- Leer combinaciones de email y contraseña desde el Excel (LoginData).
- Validar el inicio de sesión exitoso y el manejo de errores cuando las credenciales son inválidas.

3. Búsqueda y Agregado al Carrito (con ciclo):

- Leer una lista de productos desde Excel (ProductosBusqueda).
- Iterar por cada producto:
 - Buscar el producto en la tienda.
 - Verificar que aparece en los resultados.
 - Agregarlo al carrito.

4. Verificación de Productos en el Carrito:

- Verificar que los productos agregados desde el Excel se encuentren efectivamente en el carrito.

5. Escritura de Resultados en Excel:

- Registrar en un nuevo archivo Excel (outputData.xlsx) los productos que fueron agregados exitosamente al carrito.

NOTA: Todos los elementos resaltados en AMARILLO son opcionales y se consideran un plus de desarrollo y serán considerados como puntos adicionales al momento de evaluar el proyecto.

Requisitos Técnicos Específicos:

- Implementar todo el proyecto utilizando el patrón Page Object Model (POM).
- Usar selectores CSS y/o XPath claros y estables.
- Aplicar todos los tipos de esperas disponibles en Selenium:
 - Implícitas
 - Explícitas
 - Fluent Wait
- Utilizar Apache POI para:
 - Leer datos de entrada desde archivos Excel
 - Escribir resultados de prueba en Excel
- Usar aserciones para validar resultados esperados.

Estructura Esperada del Proyecto:

- pages/: Clases POM de cada página.
- tests/: Clases de prueba por caso funcional.
- utils/: Lectura/escritura de Excel, manejo de WebDriverWaits.
- resources/: Archivos de datos Excel (inputData.xlsx, outputData.xlsx).

Entrega Final:

- Proyecto en repositorio GitHub (o carpeta ZIP).
- Archivos Excel usados.
- Evidencias de ejecución (capturas).
- Documento explicando la estrategia de automatización.

Recomendaciones:

- Prioriza la legibilidad y mantenibilidad del código.
- No uses rutas absolutas para localizadores.
- Asegúrate de validar todos los pasos con aserciones claras.

- Utiliza logs o comentarios para indicar el flujo general.

RÚBRICA DE EVALUACIÓN

Criterio	Puntaje Fase 1
Proyecto en repositorio GitHub	15 pts
Archivos Excel usados.	Puntos adicionales
Evidencias de ejecución (capturas, reportes TestNG).	20 pts
Documento explicando la estrategia de automatización.	15 pts
Entregables completos y bien estructurados	10 pts
Presentación final y defensa del proyecto	30 pts
Trabajo colaborativo y evidencias de trabajo ó rol individual	10 pts
Total	100 pts