



Aseguramiento De La Calidad Del Software

Actividad 09: Laboratorio

Trabajo en clase

Profesora:

Alicia Marcela Salazar Hernandez

Integrantes:

Erika Cerdas Mejias - 2022138199

Leonardo Céspedes Tenorio - 2022080602

Kevin Chang Chang - 2022039050

Frankmin Feng Zhong - 202208924

Fecha de Entrega: 16 de octubre

II Semestre 2024

## Tabla de contenidos

Plan de Pruebas.....	3
Resultados de las pruebas.....	7
Selenium.....	7
Cypress.....	7
Playwright.....	7
Análisis comparativo.....	8
Selenium.....	8
Cypress.....	8
Playwright.....	9
Anexos.....	9
Instalación de Selenium.....	9
Instalación de Cypress.....	9
Instalación de Playwright.....	10

## Plan de Pruebas

**Objetivo:** Verificar que el sistema de gestión de productos funcione correctamente y cumpla con todos los requerimientos especificados, validando que los usuarios puedan registrarse, iniciar sesión, crear, editar, visualizar y eliminar productos de manera efectiva.

### Métricas:

- Tasa de Éxito en Registro de Usuarios:
  - Porcentaje de registros exitosos frente al número total de intentos de registro.
- Tiempo Promedio de Respuesta:
  - Tiempo promedio que tarda el sistema en procesar un registro, inicio de sesión, creación y edición de productos.
- Porcentaje de Errores en Entrada de Datos:
  - Porcentaje de errores de formato (números de teléfono, correos electrónicos, etc.) en comparación con el total de entradas realizadas durante el registro y la creación de productos.
- Tasa de Corrección de Errores:
  - Porcentaje de errores corregidos durante el ingreso de datos tras la validación automática (por ejemplo, campos obligatorios no completados).
- Satisfacción del Usuario Administrador:
  - Medición de la satisfacción mediante encuestas sobre la usabilidad y eficiencia del sistema.
- Tiempo de Procesamiento por Acción:
  - Promedio de tiempo requerido para completar acciones (registro, inicio de sesión, creación, edición y eliminación de productos).
- Número de Productos Gestionados:
  - Cantidad total de productos creados, editados y eliminados en un período específico.
- Porcentaje de Errores en Creación de Productos:
  - Proporción de intentos de creación de productos que resultan en errores frente a los que se completan con éxito.

### Escenarios de Prueba

ID	CP001
Objetivo	Verificar que un usuario pueda registrarse exitosamente.
Descripción	Este caso de prueba valida el flujo de registro de un nuevo usuario en el sistema.
Precondiciones	El usuario no debe estar registrado en el sistema.
Pasos	<ul style="list-style-type: none"><li>● Navegar a la página de inicio.</li><li>● Hacer clic en el botón de inicio de sesión.</li><li>● Hacer clic en el enlace de registro.</li><li>● Completar todos los campos del formulario de registro con datos válidos.</li></ul>

	<ul style="list-style-type: none"> <li>● Hacer clic en el botón de registro.</li> </ul>
Condiciones	El usuario debe estar registrado y ver la página de confirmación.
Métricas	Tiempo de respuesta del registro, tasa de éxito de registro.
Estado	

ID	CP002
Objetivo	Verificar que un usuario registrado puede iniciar sesión correctamente.
Descripción	Este caso de prueba asegura que un usuario pueda acceder a su cuenta con credenciales válidas.
Precondiciones	El usuario debe estar registrado en el sistema.
Pasos	<ul style="list-style-type: none"> <li>● Navegar a la página de inicio.</li> <li>● Hacer clic en el botón de inicio de sesión.</li> <li>● Ingresar el correo electrónico y la contraseña.</li> <li>● Hacer clic en el botón de inicio de sesión.</li> </ul>
Condiciones	El usuario debe estar en el panel de administración.
Métricas	Tiempo de respuesta del inicio de sesión, tasa de éxito de inicio de sesión.
Estado	

ID	CP003
Objetivo	Verificar que un usuario pueda crear un nuevo producto.
Descripción	Este caso de prueba valida el proceso de creación de un nuevo producto en el sistema.
Precondiciones	<ul style="list-style-type: none"> <li>● El usuario debe haber iniciado sesión.</li> <li>● El usuario debe ser administrador.</li> </ul>
Pasos	<ul style="list-style-type: none"> <li>● Hacer clic en "Nuevo Producto".</li> <li>● Completar todos los campos requeridos.</li> <li>● Hacer clic en el botón de crear.</li> </ul>
Condiciones	El producto debe estar creado y visible en la lista de productos.
Métricas	Tiempo de respuesta de creación, tasa de éxito de creación.
Estado	

ID	CP004
Objetivo	Verificar que un usuario pueda editar un producto existente.
Descripción	Este caso de prueba valida el proceso de edición de un producto en el sistema.
Precondiciones	Al menos un producto existente en la base de datos.
Pasos	<ul style="list-style-type: none"> <li>• Navegar a la lista de productos.</li> <li>• Hacer clic en el icono de edición de un producto.</li> <li>• Modificar los campos necesarios.</li> <li>• Hacer clic en el botón de actualizar.</li> </ul>
Condiciones	El producto debe ser actualizado y reflejar los cambios en la lista de productos.
Métricas	Tiempo de respuesta de edición, tasa de éxito de edición.
Estado	

ID	CP005
Objetivo	Verificar que un usuario pueda visualizar la lista de productos.
Descripción	Este caso de prueba asegura que un usuario pueda acceder a la lista de productos.
Precondiciones	Ninguna
Pasos	Navegar a la sección de productos.
Condiciones	El usuario debe ver la lista de productos.
Métricas	Tiempo de respuesta de carga, tasa de éxito de visualización.
Estado	

ID	CP006
Objetivo	Verificar que un usuario pueda eliminar un producto existente.
Descripción	Este caso de prueba valida el proceso de eliminación de un producto en el sistema.
Precondiciones	Al menos un producto existente en la base de datos.
Pasos	<ul style="list-style-type: none"> <li>• Navegar a la lista de productos.</li> <li>• Hacer clic en el icono de eliminar de un producto.</li> <li>• Confirmar la eliminación en el diálogo de alerta.</li> </ul>

Condiciones	El producto debe ser eliminado y ya no debe aparecer en la lista de productos.
Métricas	Tiempo de respuesta de eliminación, tasa de éxito de eliminación.
Estado	

ID	CP007
Objetivo	Verificar el manejo de errores al intentar crear un producto sin ingresar todos los campos requeridos.
Descripción	Este caso de prueba valida que el sistema maneje correctamente la falta de datos en la creación de productos.
Precondiciones	El usuario debe haber iniciado sesión.
Pasos	<ul style="list-style-type: none"> <li>• Navegar a la sección de productos.</li> <li>• Hacer clic en "Nuevo Producto".</li> <li>• No completar ningún campo.</li> <li>• Hacer clic en el botón de crear.</li> </ul>
Condiciones	El usuario debe permanecer en la página de creación de producto con un mensaje de error visible.
Métricas	Tiempo de respuesta de error, tasa de manejo de errores.
Estado	

## Resultados de las pruebas

### *Selenium*

```
> python .\prueba.py
• DevTools listening on ws://127.0.0.1:64332/devtools/browser/db3fe42e-52e1-45e5-af9f-4b3e2e5f21cc
•
DevTools listening on ws://127.0.0.1:64362/devtools/browser/e257ab49-f82b-49ce-aa3e-fb513b481aaf
•
DevTools listening on ws://127.0.0.1:64390/devtools/browser/4f01ba25-7d02-4f54-ab6b-2a7fcc785e78
•
DevTools listening on ws://127.0.0.1:64418/devtools/browser/215112e7-7ecb-440e-9d5d-710d73ac2081
•
DevTools listening on ws://127.0.0.1:64449/devtools/browser/e101ed55-782c-4e1f-adb4-c8df1874fe70
•
DevTools listening on ws://127.0.0.1:64477/devtools/browser/c48a08d1-e8e7-4bd5-a09d-92551d866a23
•
DevTools listening on ws://127.0.0.1:64505/devtools/browser/2b2ea80e-8b1e-4d78-986f-a093235935d5
•
-----
Ran 7 tests in 70.813s

OK
```

### *Cypress*

```
▼ Tests Para Farmacia Jodori
  ✓ 1. Crear una cuenta
  ✓ 2. Iniciar sesión
  ✓ 3. Crear un producto
  ✓ 4. Editar un producto
  ✓ 5. Visualizar productos
  ✓ 6. Eliminar un producto
  ✓ 7. Fallar al intentar crear un producto incompleto
```

### *Playwright*

```
→ Lab-QA git:(main) x npx playwright test --headed
Running 7 tests using 1 worker

  ✓ 1 playwright.spec.js:8:1 > CP001 (4.7s)
  ✓ 2 playwright.spec.js:33:1 > CP002 (2.2s)
  ✓ 3 playwright.spec.js:52:1 > CP003 (2.5s)
  ✓ 4 playwright.spec.js:88:1 > CP004 (2.8s)
  ✓ 5 playwright.spec.js:124:1 > CP005 (2.1s)
  ✓ 6 playwright.spec.js:137:1 > CP006 (4.6s)
  ✓ 7 playwright.spec.js:162:1 > CP007 (2.6s)
```

## Análisis comparativo

### *Selenium*

**Mantenibilidad:** Selenium permite la fácil actualización y modificación de scripts de prueba gracias a su estructura basada en código y su amplia comunidad de soporte.

**Confiabilidad:** Selenium puede generar algunos falsos negativos debido a la dependencia del estado del navegador y la velocidad de la red, aunque se pueden mitigar con técnicas de espera.

**Cobertura de pruebas:** Ofrece una buena cobertura de pruebas, permitiendo la automatización de una amplia variedad de interacciones con la interfaz de usuario en diferentes navegadores.

**Eficiencia:** Aunque las pruebas pueden ser más lentas debido a la interacción con el navegador, su rendimiento se puede optimizar utilizando estrategias adecuadas de diseño de pruebas.

**Facilidad de uso:** Selenium tiene una curva de aprendizaje moderada, pero sus diversas bibliotecas y documentación lo hacen relativamente sencillo de usar y mantener.

### *Cypress*

**Mantenibilidad:** Cypress facilita la actualización y modificación de pruebas con una estructura intuitiva y un sistema de pruebas basado en JavaScript que es fácil de leer y entender.

**Confiabilidad:** Cypress tiene una alta confiabilidad y reduce los falsos negativos gracias a su capacidad para manejar el tiempo de espera y su enfoque en la ejecución de pruebas en un entorno real del navegador.

**Cobertura de pruebas:** Ofrece una cobertura de pruebas exhaustiva, permitiendo la automatización de pruebas de extremo a extremo, unitarias y de integración de manera efectiva.

**Eficiencia:** Las pruebas en Cypress se ejecutan rápidamente, ya que se ejecutan directamente en el navegador y permiten el monitoreo en tiempo real de las acciones.

**Facilidad de uso:** Cypress es muy fácil de usar y configurar, con una interfaz gráfica intuitiva y una documentación extensa que simplifica la escritura y el mantenimiento de scripts de prueba.



## ***Playwright***

**Mantenibilidad:** Playwright facilita la mantenibilidad gracias a su arquitectura moderna y bien estructurada, compatible con patrones de diseño como el Page Object Model (POM).

**Confiabilidad:** Playwright es altamente confiable. Su capacidad para manejar tiempos de espera automáticos, sincronización con el DOM y captura precisa de interacciones reduce significativamente los falsos negativos. Además, soporta entornos headless y pruebas en paralelo, mejorando la estabilidad de las pruebas.

**Cobertura de pruebas:** Playwright ofrece una cobertura de pruebas muy completa, soportando la automatización de múltiples navegadores (Chromium, Firefox, WebKit) y sistemas operativos, incluyendo Windows, macOS y Linux.

**Eficiencia:** Las pruebas en Playwright son rápidas y eficientes, ya que optimiza la sincronización y la ejecución paralela. Puede manejar múltiples navegadores de manera simultánea y su ejecución en modo "headless" mejora la velocidad en comparación con herramientas tradicionales.

**Facilidad de uso:** Aunque tiene una curva de aprendizaje ligeramente más pronunciada que Cypress, sigue siendo fácil de usar. Su API es clara y moderna, y la documentación es extensa y bien organizada.

## **Anexos**

### Instalación de Selenium

1. Se ocupa descargar Python
2. Ejecutar **pip install selenium**

### Instalación de Cypress

1. Se tener descargar Node.js
2. Ejecutar el **npm init**
3. Luego ejecutar **npm install cypress --save-dev**
4. Agregar la línea de **"cypress:open": "cypress open"** al .json en "scripts"

```

package.json > ...
1  {
2    "name": "cypress-framework",
3    "version": "1.0.0",
4    "description": "cypressproject",
5    "main": "index.js",
6    "scripts": {
7      "test": "test",
8      "cypress:open": "cypress open"
9    },
10   "keywords": [
11     "e2e",
12     "test",
13     "cypress",
14     "autentification",
15     "framework"
16   ],

```

5. Ejecutar **npm run cypress:open**

#### Instalación de Playwright

1. Se necesita tener **Node.js** y **npm**
2. Ejecutar el comando **npm install @playwright/test --save-dev**
3. Para correr las pruebas utilizar el comando **npx playwright test --headed**