

**INSTITUTO POLITÉCNICO NACIONAL**  
**UPIICSA**  
**INGENIERÍA DE PRUEBAS**

# **Plan de pruebas**

**Automatización Selenium**

**PROF.: Ramon Cruz Martínez**

**ALUMNOS:**

- **Martínez Lozano Armando**
- **Medina Gómez Jonathan Ramón**
- **Mendoza Olvera Diego Alejandro**
- **Paz Abedoy Christian**

**GRUPO: 6NV61**

## **Plan de pruebas: Tienda web**

### **Objetivo**

El objetivo de este plan de pruebas es diseñar, desarrollar y poner en marcha un conjunto de pruebas automáticas utilizando la herramienta Selenium. El propósito principal es revisar de forma eficiente y confiable las funciones más importantes y los flujos clave que siguen los usuarios en la página web.

Esta automatización nos permitirá probar rápidamente que las nuevas actualizaciones no hayan descompuesto funciones que ya existían (lo que se conoce como "pruebas de regresión"). Esto se hará después de cada cambio en el código, ayudando a mantener la página estable, reduciendo el tiempo de pruebas manuales y encontrando errores mucho más rápido.

### **Alcance**

El alcance de la automatización se centrará en los flujos funcionales críticos de la aplicación, que garantizan la operación principal de la tienda. Los procesos por automatizar incluyen:

- Flujo de cuentas de usuario:
  - Prueba de registro de un nuevo usuario con datos válidos.
  - Prueba de inicio de sesión (Login) con credenciales correctas.
  - Validación de cierre de sesión (Logout).
- Flujo de descubrimiento de producto:
  - Verificar la navegación del menú principal a una página de categoría.
  - Validar la selección de un producto desde la categoría para ver su página de detalle.
- Flujo de gestión del carrito:
  - Automatizar el proceso de "Añadir al Carrito" desde la página de un producto.
  - Validar la actualización de cantidades (ej. cambiar 1 producto por 3) desde el carrito.
  - Validar la eliminación de un producto del carrito.
- Flujo de compra (Checkout):
  - Prueba completa del proceso de pago (checkout) con un método de prueba, desde el carrito hasta la página de "Compra Exitosa".

- Flujo post-compra:
  - Verificar que una compra completada se refleje correctamente en el "Historial de Pedidos" del usuario.
  - Validar la cancelación de un pedido desde el historial.

#### **Criterios de aceptación:**

- El script de prueba se ejecuta de principio a fin sin fallos técnicos (timeouts, errores de Selenium, o excepciones no controladas).
- El script completa el formulario de registro y verifica que es redirigido a la página de "Bienvenida" o "Mi Cuenta".
- El script inicia sesión con credenciales válidas y verifica la presencia de un elemento que confirme el inicio de sesión.
- El script intenta iniciar sesión con credenciales incorrectas y verifica que el mensaje de error "Correo o contraseña incorrectos" está visible.
- El script añade un producto y verifica que el contador del carrito se actualiza.
- El script elimina un producto del carrito y verifica que el carrito muestra un estado de "vacío".
- El script completa exitosamente el formulario de pago (con datos de prueba) y es redirigido a la página de "Compra Exitosa".
- El script navega al "Historial de Pedidos" después de la compra y verifica que el nuevo número de pedido aparece en la lista.
- El script ejecuta la acción de "Cancelar Pedido" y verifica que el estado del pedido en la web cambia a "Cancelado".
- El script verifica que el módulo de recomendaciones está presente y visible en la página del producto (no se valida la lógica de las recomendaciones, solo que el componente cargó).

#### **Recursos:**

- **Personal:** Tester Diego Alejandro Mendoza Olvera.
- **Entorno de pruebas:**
  - Plataforma: Navegador web brave browser.
  - Dispositivo: Computadora.
  - Sistema operativo: Linux (Debian).
  - Entorno: Servidor en línea ejecutado por render.

- **Estrategia de pruebas:**

- Pruebas funcionales: Validar que cada módulo cumpla con su propósito (registro, inicio de sesión, carrito, pagos, historial, cancelaciones).
- Pruebas de integración: Verificar la interacción entre los módulos de registro, inicio de sesión, categorías, carrito, pagos, historial y cancelaciones.

## Casos de prueba

### Registro e inicio de sesión

Caso de Prueba	Descripción
CP-001	Verificar registro exitoso con correo válido.
CP-002	Validar error al registrar correo duplicado.
CP-003	Probar inicio de sesión con credenciales correctas.
CP-004	Verificar mensaje de error con credenciales incorrectas.
CP-005	Validar cierre de sesión.

### Carrito de compras

Caso de Prueba	Descripción
CP-006	Agregar productos al carrito.
CP-007	Eliminar productos del carrito.
CP-008	Actualizar cantidad de productos.

### Compras y métodos de pago

Caso de Prueba	Descripción
CP-009	Simular pago exitoso con método válido.
CP-010	Validar error con método de pago rechazado.

## Historial de pedidos

Caso de Prueba	Descripción
CP-011	Confirmar que las compras se registren correctamente.
CP-012	Verificar visualización de pedidos anteriores.
CP-013	Validar detalle de pedido (productos, fecha, total, estado).

## Cancelación de pedidos

Caso de Prueba	Descripción
CP-014	Probar cancelación de un pedido recién hecho.
CP-015	Validar notificación de cancelación.

## Recomendaciones

Caso de Prueba	Descripción
CP-016	Validar que se muestre el apartado de recomendaciones después de una compra.

## Cronograma:

ACTIVIDAD	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4
Creación de scripts				✓
Ejecución de scripts				✓

## Informes:

Se generará un informe final de pruebas que incluirá:

- Resultados obtenidos (éxito/fallo).
- Número de defectos detectados.
- Número de defectos corregidos.
- Recomendaciones finales para la siguiente versión.