

Universidad Autónoma de Ciudad Juárez
Extensión Multidisciplinaria de Ciudad Universitaria



UACJ

Automatización De Pruebas Equipos

Aseguramiento de la calidad del software

Profesor Abraham López Nájera

Semestre Agosto - Diciembre 2025

Nombre del equipo: Sinn nombre

Anna Lizbeth Barajas Sandoval | 222835

Luis Ubaldo Flores Pineda | 223046

Leslie Lucero Franco Calleros | 223053

25 de Octubre del 2025

Descripción del sistema bajo pruebas

Nombre: SauceDemo

URL: <https://www.saucedemo.com/>

Funcionalidades clave:

- Cuenta con un login de usuario y contraseña como autenticación, además de manejo de usuarios bloqueados o con credenciales inválidas.
- Cuenta con inventario de productos, como visualización del catálogo de productos, donde cada producto tiene elementos como nombre, imagen, descripción y precio. Así como su ordenamiento por precio o nombre.
- Tiene la funcionalidad de un carrito de compras para agregar productos desde el catálogo, eliminar productos del carrito y visualización del carrito con todos los items seleccionados.
- Proceso de compras, como un formulario para capturar datos del comprador, resumen de compra, y mensajes de orden completo.
- Menús laterales, footers, botones y enlaces interactivos.

Plan de pruebas

Criterios de aceptación

Funcionalidad	Criterio de aceptación
Login	Debe permitir acceso al sistema con el ingreso de las credenciales correctas, y denegar el acceso a credenciales incorrectas.
Visualización de catálogo	Debe mostrar un listado de artículos disponibles, junto con información relevante para su compra. Adicionalmente, debe de permitir que los usuarios accedan a la página del producto y agregarlo a su carrito.
Visualización de producto	Debe mostrar un producto, junto con su información relevante como nombre, descripción, precio e imagen. Además debe permitir al usuario agregarlo al carrito.
Filtrado	Debe permitir al usuario filtrar o clasificar los productos bajo cierto criterio.
Carrito de compras	Debe permitir a los usuarios ver todos los productos agregados a su carrito, sus precios y debe permitir continuar con el pago.
Proceso de compra	Debe permitir a los usuarios ver un resumen de su compra, ingresar datos del envío y confirmar su compra.

Plan de pruebas

Objetivo

El objetivo principal de este plan de pruebas es validar las funcionalidades esenciales del sistema SauceDemo mediante la ejecución de pruebas funcionales tanto manuales como automatizadas. Su propósito es confirmar que el sistema cumple con los criterios de aceptación establecidos. Este plan pretende identificar defectos, asegurar la calidad del software y garantizar una experiencia de usuario satisfactoria, evaluando procesos clave como la autenticación, la navegación de productos, la gestión del carrito de compras y la finalización de órdenes.

Alcance

El alcance del plan de pruebas considera la validación de las funciones principales del sistema SauceDemo. Incluye la verificación del proceso de inicio de sesión, tanto con credenciales correctas como incorrectas, para asegurar que el sistema responda de forma adecuada. Abarca la revisión del catálogo de productos, comprobando que los artículos y sus detalles individuales se muestren correctamente. Asimismo, se evaluarán las funciones de filtrado y ordenamiento para confirmar que los productos se organizan según los criterios seleccionados por el usuario.

El alcance también considera la validación de todas las acciones relacionadas con el carrito de compras, como agregar, eliminar y visualizar productos, asegurando que el contenido refleje los cambios realizados. Dentro del proceso de compra, se revisará el flujo completo de checkout, verificando que cada paso se ejecute correctamente hasta la confirmación final de la orden. Además, se incluye la revisión de la navegación general del sistema, asegurando que los menús, botones y enlaces funcionen de manera consistente.

Quedan fuera del alcance las pruebas de rendimiento o carga, las pruebas de seguridad avanzadas o de penetración, la validación del backend o la base de datos, las pruebas en dispositivos móviles, la integración con sistemas de pago reales y las evaluaciones de accesibilidad basadas en WCAG. Estas áreas no forman parte del objetivo del presente plan.

Funcionalidades cubiertas

En primer lugar, se valida el sistema de autenticación y control de acceso. Esta funcionalidad incluye pruebas de acceso con credenciales correctas (usuario "standard_user" y contraseña "secret_sauce"), el rechazo de credenciales inválidas con mensaje de error, el manejo de usuarios bloqueados, y la redirección automática al catálogo tras autenticación exitosa. Esta área es cubierta por los casos M-1 (manual) y A-2 (automatizado), validando escenarios positivos y negativos.

La segunda categoría corresponde a la visualización del catálogo de productos. Se verifica el despliegue correcto del inventario con información completa de cada artículo (nombre, imagen, descripción y precio), el acceso a páginas de detalle individual, la funcionalidad del botón "Add to cart" desde el catálogo, y la carga apropiada de imágenes. El caso manual M-2 cubre esta funcionalidad de manera integral.

El sistema de filtrado y ordenamiento constituye la tercera funcionalidad validada. Se prueban los cuatro tipos de ordenamiento disponibles: alfabético ascendente y descendente (Name A-Z y Z-A), y por precio ascendente y descendente (Price low to high y high to low). Se verifica la persistencia del filtro durante la navegación y la actualización visual correcta. Esta área es evaluada por el caso manual M-1 y el caso automatizado A-3, enfocado en el ordenamiento por precio.

La gestión del carrito de compras abarca la capacidad de agregar productos desde el catálogo o páginas individuales, eliminar items, actualización en tiempo real del badge contador, visualización correcta de productos agregados con su información completa, cálculo de subtotales, y funcionalidad de botones "Continue Shopping" y "Checkout". El caso M-3 valida específicamente esta funcionalidad, mientras que A-1 la incluye como parte del flujo completo.

Finalmente, el proceso de checkout valida el flujo completo de compra: acceso al formulario de información, validación de campos obligatorios (nombre, apellido y código postal), navegación al resumen, visualización de productos seleccionados, cálculo correcto de subtotal, impuestos y total final, confirmación exitosa con mensaje "Thank you for your order", y funcionalidad del botón "Finish". El caso automatizado A-1 ejecuta este flujo end-to-end completo.

Distribución de cobertura: El plan contempla seis casos de prueba divididos equitativamente: tres manuales (M-1 para filtrado, M-2 para visualización de productos, M-3 para carrito) y tres automatizados (A-1 para proceso completo de compra, A-2 para login, A-3 para filtrado por precio).

Funcionalidades excluidas: Por limitaciones de tiempo y enfoque del proyecto, no se incluyen: navegación del menú lateral, opciones "About" y "Logout", enlaces de redes sociales del footer, funcionalidad "Reset App State", pruebas de rendimiento, pruebas de seguridad avanzadas, validación responsive en móviles, pruebas de accesibilidad WCAG, integración con sistemas de pago reales, ni validación del backend o base de datos.

Tipos de pruebas

Funcionales: Se realizarán pruebas funcionales para validar que cada componente del sistema SauceDemo opere según los requisitos especificados. Estas pruebas verifican que las funcionalidades principales (autenticación, visualización de productos, gestión de carrito, proceso de compra y filtrado) se comporten correctamente ante diferentes entradas y escenarios de uso. El enfoque es confirmar que el sistema cumple con los criterios de aceptación establecidos.

Manuales: Las pruebas manuales se ejecutarán para validar 3 casos específicos: filtrado de productos (M-1), visualización de producto individual (M-2) y gestión del carrito de compras (M-3). Este tipo de pruebas permite evaluar aspectos de usabilidad, verificar elementos visuales y explorar el comportamiento del sistema que requieren análisis y juicio humano.

Automáticas: Se implementarán 3 casos de prueba automatizados utilizando Katalon Recorder: el proceso completo de compra (A-1), la autenticación con credenciales válidas e inválidas (A-2), y el filtrado de productos por precio (A-3). La automatización de estos flujos críticos permite ejecutar pruebas de regresión de manera rápida y consistente, generando logs y evidencias para facilitar el análisis de resultados.

Herramienta seleccionada

Katalon Recorder

Katalon Recorder es una extensión gratuita para navegadores Chrome y Firefox que permite grabar y ejecutar pruebas automatizadas de aplicaciones web directamente desde el navegador, sin requerir instalaciones complejas ni configuraciones avanzadas.

Esta herramienta fue seleccionada por su accesibilidad y facilidad de uso, características ideales para equipos que se inician en la automatización de pruebas. Su interfaz intuitiva permite grabar interacciones del usuario con la aplicación web, generando automáticamente scripts ejecutables que pueden ser editados y reproducidos según las necesidades del proyecto.

Katalon Recorder está construido sobre Selenium WebDriver, ofreciendo soporte nativo para comandos de Selenium y garantizando compatibilidad con estándares de la industria. Entre sus principales ventajas destacan la ejecución inmediata de pruebas, la generación automática de selectores, la visualización en tiempo real de pasos de ejecución, y logs detallados que facilitan el análisis de resultados.

Para este proyecto, resulta especialmente apropiado ya que permite validar eficientemente los flujos críticos de SauceDemo (autenticación, proceso de compra y filtrado) mientras genera documentación visual mediante capturas y logs de ejecución.

Casos de prueba

ID	Caso de prueba	Descripción	Datos de entrada	Pasos	Resultados esperados
M-1	Filtrado de producto	Se deberá probar que todos los filtros del sistema funcionen como descrito.	N/A	<ol style="list-style-type: none"> 1. Ingresar al catálogo 2. Seleccionar filtro 3. Visualizar resultados 4. Repetir por cada filtro 	4 pantallas de respuesta con productos ordenados según filtro
M-2	Visualización de producto individual	Se deberá probar que al seleccionar un producto se muestre toda la información detallada.	N/A	<ol style="list-style-type: none"> 1. Ingresar al catálogo. 2. Seleccionar producto del catálogo. 3. Verificar información mostrada. 4. Verificar botones disponibles. 	Página de detalle con nombre, descripción, precio, imagen y botones "Add to cart" y "Back" funcionales
M-3	Visualización del catálogo de productos	Se deberá probar que el sistema muestre correctamente el listado completo de productos disponibles con toda su información relevante.	N/A	<ol style="list-style-type: none"> 1. Iniciar sesión 2. Acceder al catálogo principal (inventory.html) 3. Verificar que se muestren todos los productos 4. Verificar que cada producto incluya: nombre, imagen, descripción y precio 5. Verificar que las imágenes carguen correctamente 6. Verificar que los botones "Add to cart" estén visibles y funcionales 	El catálogo despliega todos los productos disponibles, cada uno con su información completa (nombre, imagen, descripción, precio), las imágenes cargan sin errores, y todos los botones "Add to cart" son visibles y clicables.
A-1	Proceso de compra correcta	Se deberá probar que el sistema permite realizar una compra con los datos correctos.	Nombre, Apellido, Código postal	<ol style="list-style-type: none"> 1. Agregar productos al carrito 2. Ir a carrito 3. Seleccionar checkout 4. Ingresar nombre, apellido y código 	Pantalla de orden completada

				5. postal Seleccionar Terminar	
A-2	Login con credenciales válidas e inválidas.	Se deberá probar que el sistema permita el acceso con credenciales correctas y deniegue el acceso con credenciales incorrectas..	Inválidas: Usuario: "usuario_invalido" Contraeña: "pass_incorrecta" Válidas: Usuario: "standard_user" Contraeña: "secret_sauce"	1. Abrir sitio web 2. Ingresar usuario inválido 3. Ingresar contraeña incorrecta 4. Hacer clic en Login 5. Verificar mensaje de error 6. Limpiar campos 7. Ingresar usuario válido 8. Ingresar contraeña correcta 9. Hacer clic en Login	Primero muestra el mensaje de error "Username and password do not match" y permanece en login. Luego permite el acceso, redirige al catálogo de productos y otorga acceso completo al sistem
A-3	Gestión del carrito de compras	Se deberá probar que el sistema permite agregar y eliminar productos del carrito correctamente, y que el contador de ítems se actualiza de forma precisa	N/A	1. Iniciar sesión 2. Desde el catálogo seleccionar “add cart” a un producto 3. Verificar que el badge del carrito muestre “1” 4. Agregar un segundo producto 5. Verificar que el badge muestre “2” 6. Acceder al carrito 7. Verificar que ambos productos aparezcan listados 8. Seleccionar “Remove” en uno de los productos 9. Verificar que el producto se elimine y el badge muestre “1” 10. Verificar que el precio total se actualice correctamente	El sistema permite agregar múltiples productos al carrito, el contador se actualiza correctamente en tiempo real, los productos se visualizan en el carrito con su información completa, la eliminación funciona correctamente y los totales se calculan de forma precisa.

Lista de casos seleccionados para automatización

- A-1: Proceso de compra
- A-2: Login
- A-3: Carrito

Justificación de casos automatizados

El caso A-1 se automatiza gracias a que es un flujo directo de acciones, fáciles de automatizar y visualizar. Es una funcionalidad crítica del sistema, por lo cual realizar diferentes tipos de test rigurosos es lo mejor.

El caso A-2 se automatiza porque valida ambos escenarios del sistema de autenticación (positivo y negativo) en un solo flujo, permitiendo verificar tanto el rechazo de accesos no autorizados como la aceptación de usuarios válidos. Es crítico para asegurar la seguridad del sistema en cada regresión.

El caso A-3 se automatiza porque la gestión del carrito de compras es una funcionalidad crítica en cualquier sistema de e-commerce. Esta característica permite a los usuarios agregar, visualizar y eliminar productos antes de finalizar su compra, siendo uno de los flujos más utilizados por los clientes. La automatización permite verificar que el badge contador se actualice correctamente en tiempo real, que los productos se agreguen y eliminen sin errores, y que los cambios se reflejen de manera consistente en la interfaz. Validar este flujo automáticamente asegura que la experiencia de compra del usuario se mantenga funcional y confiable en cada actualización del sistema.

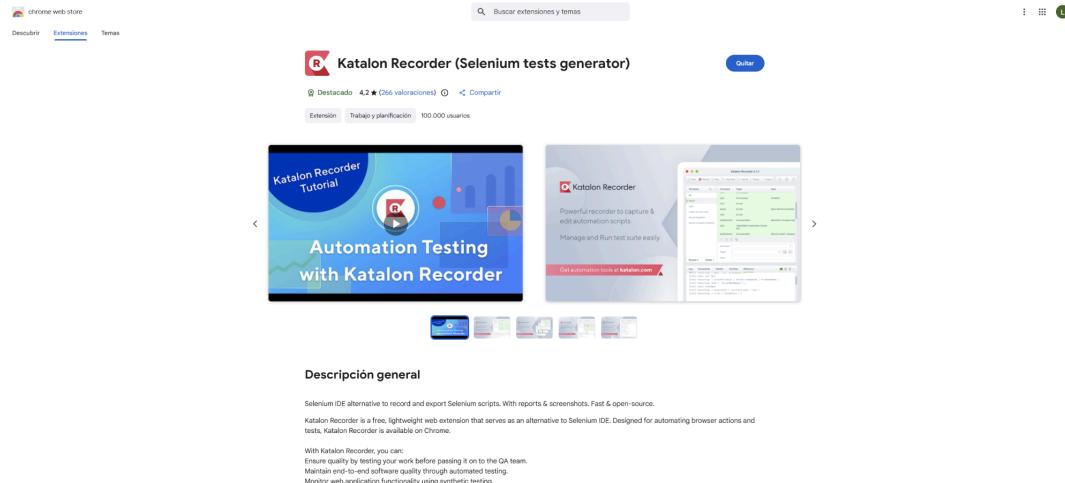
Herramienta configurada

Instalación

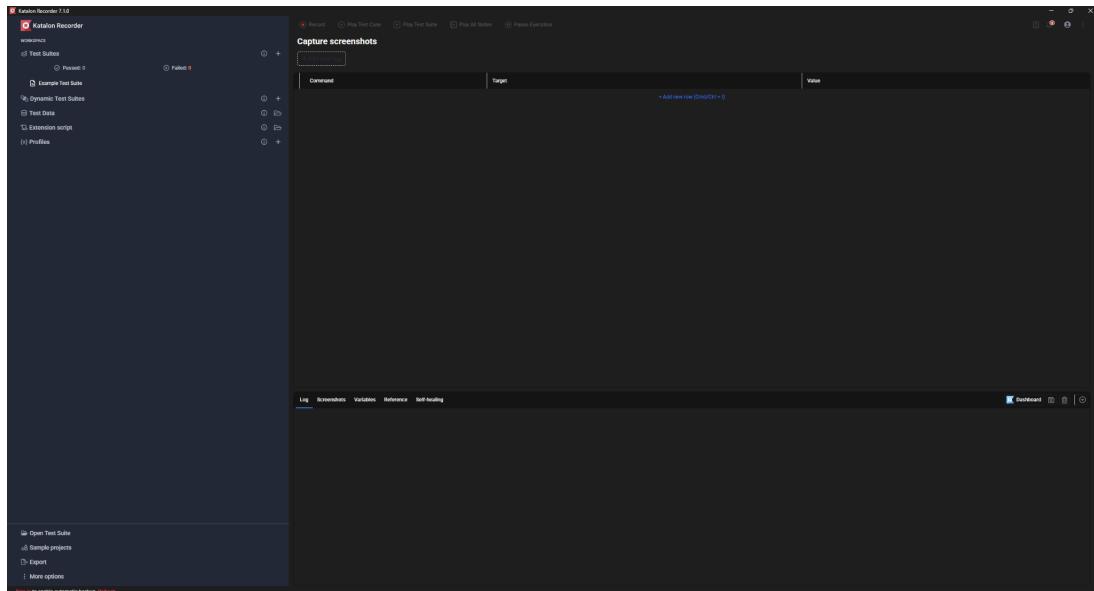
1. Abrir el navegador, de preferencia en Chrome, pero también es posible con el navegador de Microsoft Edge
2. Buscar en el navegador “Extensiones” y el primer enlace debería ser el siguiente: [“https://chromewebstore.google.com/category/extensions?hl=es”](https://chromewebstore.google.com/category/extensions?hl=es)
3. En el buscador se ingresara el siguiente nombre “Katalon Recorder”
4. Seleccionar la primera opción que aparezca, (en caso de que no sea la primera opción tiene forma de un cuadrado con un recorte triangular del lado derecho y una R en el centro)
5. Una vez dentro, seleccionar el botón de instalar la extensión
6. Una vez instalado, entrar en la sección de extensiones del lado derecha superior del navegador y seleccionar el ícono en forma de rompecabezas (a veces no se encuentra como ajuste favorito, entonces también se puede encontrar en la sección de los 3 puntos en las opciones iniciales).
7. Selecciona “Katalon Recorder”
8. Aparecerá una ventana
9. De lado izquierdo se encontrará la sección de “Test Suites”
10. Una vez seleccionado se encontrará una serie de opciones del lado superior
11. Selecciona la opción “Record”
12. De ahí es realizar la serie de pasos que correspondan para realizar la automatización
13. Una vez terminado solo sería concluir la grabación, ya sea seleccionando “stop” de la barra flotante inferior de la ventana o desde la ventana de la extensión
14. Una vez terminada la grabación se puede verificar que se grabó correctamente ejecutando la opción “Play Test Case” para que se ejecute automáticamente
15. Al final se puede observar en la sección de terminal algunos datos como los logs.

Capturas

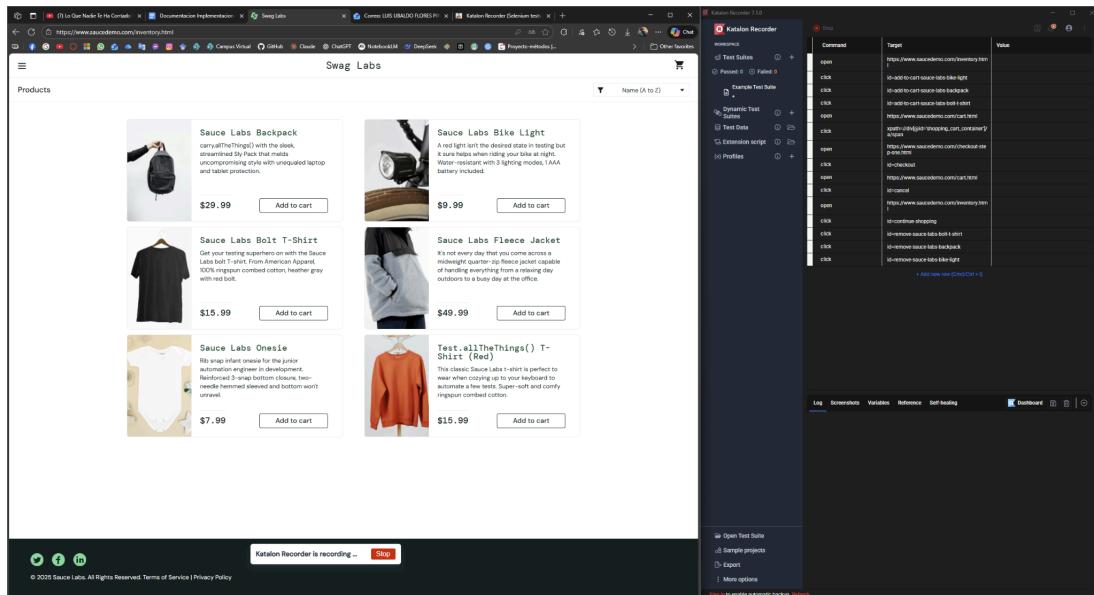
Instalación de Katalon Recorder



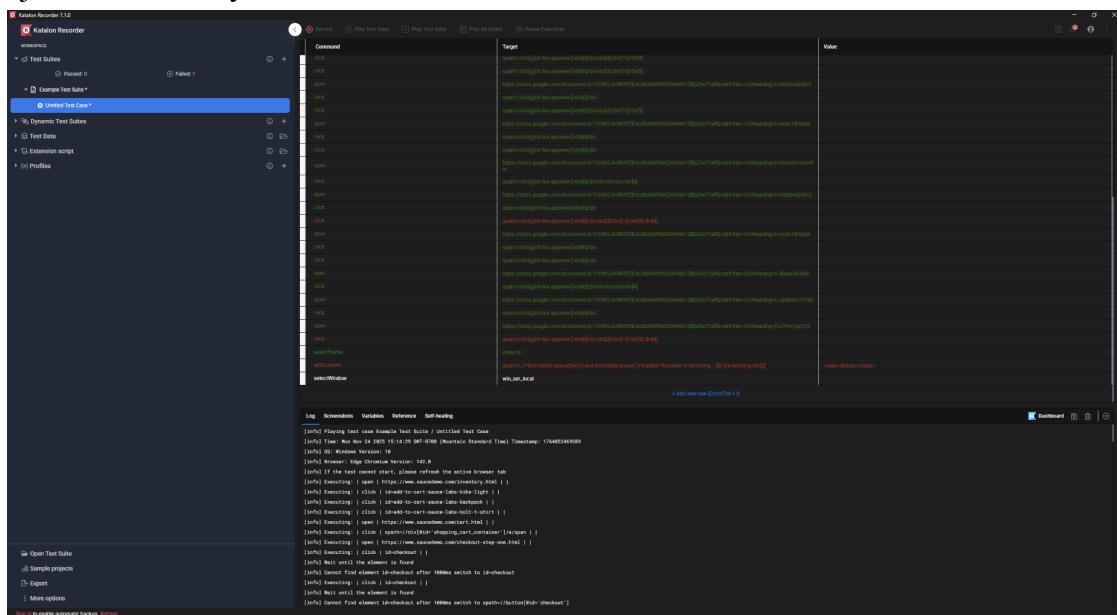
Creación de Test Suite



Grabación de uso de la página



Ejecución de test y obtención de resultados



Script base

Con esta herramienta no se puede obtener un "script base", únicamente se puede exportar un script por cada Test Case que se realice.

Scripts Automatizados

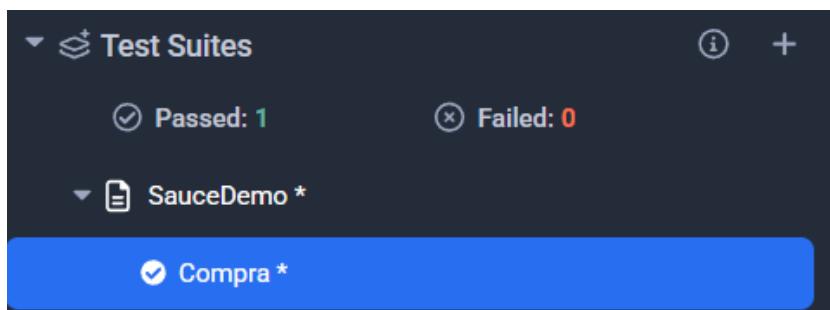
Script de caso A-1

Descripción

El script automatizado del caso de prueba A-1 realiza el proceso de compra iniciando desde la selección de 2 productos en el listado, agregándolos al carrito de compras. Posteriormente se dirige a la página del carrito, en donde se muestra el resumen de los productos a comprar. Se selecciona la opción de “checkout” y se introducen los 3 campos solicitados: nombre, apellido y código postal. De ahí sale un último resumen de la compra y la opción de “finalizar compra”. Una vez seleccionada esta última opción aparece la pantalla de compra realizada y se termina el test de manera exitosa.

Capturas

Evidencia Test Suite



Evidencia del test ejecutado correctamente

Command	Target	Value
click	link=2	
open	https://www.saucedemo.com/checkout-step-one.html	
click	id=first-name	
type	id=first-name	Liz
click	id=last-name	
type	id=last-name	San
click	id=postal-code	
type	id=postal-code	12345
open	https://www.saucedemo.com/checkout-step-two.html	
click	id=finish	
open	https://www.saucedemo.com/inventory.html	
click	id=back-to-products	

Log Screenshots Variables Reference Self-healing

```
get there!')||(1)/following::button[1] after 1000ms switch to xpath=.//*[normalize-space(text()) and normalize-space(.)='Thank you for your order!']{1}/preceding::button[1]
[info] Executing: | click | xpath=.//*[normalize-space(text()) and normalize-space(.)='Thank you for your order!']{1}/preceding::button[1]
[info] Wait until the element is found
[info] Cannot find element xpath=.//*[normalize-space(text()) and normalize-space(.)='Thank you for your order!']{1}/preceding::button[1]
[info] Executing: | click | xpath=.//*[normalize-space(text()) and normalize-space(.)='Twitter']{1}/preceding::button[1]
[info] Time: Fri Nov 21 2025 17:26:49 GMT-0700 (Mountain Standard Time) Timestamp: 1763771209203
[info] Test case passed
```

Captura de la página web



Swag Labs



Checkout: Complete!



Thank you for your order!

Your order has been dispatched, and will arrive just as fast as the pony can get there!

[Back Home](#)

Script

```
10  class AppDynamicsJob(unittest.TestCase):
11      def setUp(self):
12          # AppDynamics will automatically override this web driver
13          # as documented in https://docs.appdynamics.com/display/PRO44/Write+Your+First+Script
14          self.driver = webdriver.Firefox()
15          self.driver.implicitly_wait(30)
16          self.base_url = "https://www.google.com/"
17          selfverificationErrors = []
18          self.accept_next_alert = True
19
20      def test_app_dynamics_job(self):
21          driver = self.driver
22          driver.get("https://www.saucedemo.com/inventory.html")
23          driver.find_element_by_id("add-to-cart-sauce-labs-backpack").click()
24          driver.find_element_by_id("add-to-cart-sauce-labs-bike-light").click()
25          driver.get("https://www.saucedemo.com/cart.html")
26          driver.find_element_by_link_text("2").click()
27          driver.get("https://www.saucedemo.com/checkout-step-one.html")
28          driver.find_element_by_id("first-name").click()
29          driver.find_element_by_id("first-name").clear()
30          driver.find_element_by_id("first-name").send_keys("Liz")
31          driver.find_element_by_id("last-name").click()
32          driver.find_element_by_id("last-name").clear()
33          driver.find_element_by_id("last-name").send_keys("San")
34          driver.find_element_by_id("postal-code").click()
35          driver.find_element_by_id("postal-code").clear()
36          driver.find_element_by_id("postal-code").send_keys("12345")
37          driver.get("https://www.saucedemo.com/checkout-step-two.html")
38          driver.find_element_by_id("finish").click()
39          driver.get("https://www.saucedemo.com/inventory.html")
40          driver.find_element_by_id("back-to-products").click()
41
42      def is_element_present(self, how, what):
43          try: self.driver.find_element(by=how, value=what)
44          except NoSuchElementException as e: return False
45          return True
46
47      def is_alert_present(self):
48          try: self.driver.switch_to_alert()
49          except NoAlertPresentException as e: return False
50          return True
```

```

51     def close_alert_and_get_its_text(self):
52         try:
53             alert = self.driver.switch_to_alert()
54             alert_text = alert.text
55             if self.accept_next_alert:
56                 alert.accept()
57             else:
58                 alert.dismiss()
59             return alert_text
60         finally: self.accept_next_alert = True
61
62     def tearDown(self):
63         # To know more about the difference between verify and assert,
64         # visit https://www.seleniumhq.org/docs/06_test_design_considerations.jsp#validating-results
65         self.assertEqual([], self.verificationErrors)
66
67     if __name__ == "__main__":
68         unittest.main()
69
70

```

Resultados

Test Passed

El script logró ejecutar de manera correcta todos los comandos requeridos para realizar una compra en la página web seleccionada, por lo que se puede aprobar el caso de prueba de manera exitosa.

En los logs se puede apreciar cómo se fue ejecutando cada uno de los pasos descritos en el test case, y como en ciertos casos la ejecución se veía detenida debido a que el programa no encontraba uno de los elementos. En estos casos, el programa se adaptó automáticamente encontrando otras maneras de realizar las mismas acciones. De esta manera se pudo realizar todo el caso descrito sin mayores problemas y se terminó la prueba con éxito.

Logs

```

[info] Playing test case SauceDemo / Compra
[info] Time: Fri Nov 21 2025 17:26:21 GMT-0700 (Mountain Standard Time) Timestamp:
1763771181738
[info] OS: Windows Version: 10
[info] Browser: Chrome Version: 142.0
[info] If the test cannot start, please refresh the active browser tab
[info] Executing: | open | https://www.saucedemo.com/inventory.html ||
[info] Executing: | click | id=add-to-cart-sauce-labs-backpack ||
[info] Executing: | click | id=add-to-cart-sauce-labs-bike-light ||
[info] Executing: | open | https://www.saucedemo.com/cart.html ||
[info] Executing: | click | link=2 ||
[info] Executing: | open | https://www.saucedemo.com/checkout-step-one.html ||
[info] Executing: | click | id=first-name ||
[info] Executing: | type | id=first-name | Liz |
[info] Executing: | click | id=last-name ||
[info] Executing: | type | id=last-name | San |
[info] Executing: | click | id=postal-code ||
[info] Executing: | type | id=postal-code | 12345 |
[info] Executing: | open | https://www.saucedemo.com/checkout-step-two.html ||
[info] Executing: | click | id=finish ||

```

```
[info] Executing: | open | https://www.saucedemo.com/inventory.html ||
[info] Executing: | click | id=back-to-products ||
[info] Wait until the element is found
[info] Cannot find element id=back-to-products after 1000ms switch to id=back-to-products
[info] Executing: | click | id=back-to-products ||
[info] Wait until the element is found
[info] Cannot find element id=back-to-products after 1000ms switch to
xpath=/button[@id='back-to-products']
[info] Executing: | click | xpath=/button[@id='back-to-products'] ||
[info] Wait until the element is found
[info] Cannot find element xpath=/button[@id='back-to-products'] after 1000ms switch to
xpath=/div[@id='checkout_complete_container']/button
[info] Executing: | click | xpath=/div[@id='checkout_complete_container']/button ||
[info] Wait until the element is found
[info] Cannot find element xpath=/div[@id='checkout_complete_container']/button after
1000ms switch to xpath=(//*[normalize-space(text()) and normalize-space(.)='Your order has
been dispatched, and will arrive just as fast as the pony can get
there!'])[1]/following::button[1]
[info] Executing: | click | xpath=(//*[normalize-space(text()) and normalize-space(.)='Your
order has been dispatched, and will arrive just as fast as the pony can get
there!'])[1]/following::button[1] ||
[info] Wait until the element is found
[info] Cannot find element xpath=(//*[normalize-space(text()) and normalize-space(.)='Your
order has been dispatched, and will arrive just as fast as the pony can get
there!'])[1]/following::button[1] after 1000ms switch to xpath=(//*[normalize-space(text())
and normalize-space(.)='Thank you for your order!'])[1]/following::button[1]
[info] Executing: | click | xpath=(//*[normalize-space(text()) and normalize-space(.)='Thank
you for your order!'])[1]/following::button[1] ||
[info] Wait until the element is found
[info] Cannot find element xpath=(//*[normalize-space(text()) and
normalize-space(.)='Thank you for your order!'])[1]/following::button[1] after 1000ms switch
to xpath=(//*[normalize-space(text()) and
normalize-space(.)='Twitter'])[1]/preceding::button[1]
[info] Executing: | click | xpath=(//*[normalize-space(text()) and
normalize-space(.)='Twitter'])[1]/preceding::button[1] ||
[info] Time: Fri Nov 21 2025 17:26:49 GMT-0700 (Mountain Standard Time) Timestamp:
1763771209203
[info] Test case passed
```

Script de caso A-2

Descripción

Este script valida el comportamiento completo del sistema de autenticación. Primero intenta acceder con credenciales incorrectas verificando que el sistema muestre el mensaje de error apropiado y deniegue el acceso. Posteriormente, limpia los campos e ingresa credenciales válidas, verificando que el sistema permite el acceso y redirija correctamente al catálogo de productos. Este enfoque permite validar ambos escenarios (positivo y negativo) en una sola ejecución, asegurando tanto el rechazo de accesos no autorizados como la aceptación de usuarios válidos.

Capturas

Script completo en Katalon Recorder

The screenshot shows the Katalon Recorder interface. At the top, there's a table of recorded commands:

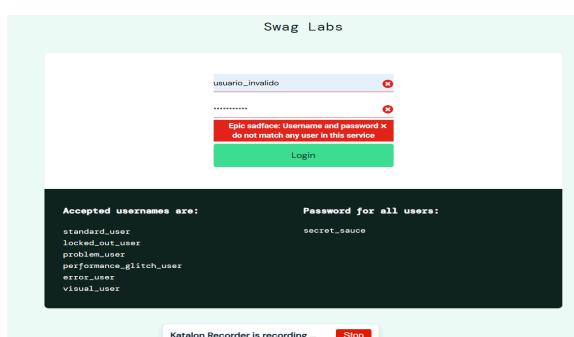
Command	Target	Value
open	https://www.saucedemo.com/	
click	id=username	
type	id=username	usuario_invalido
click	id=login-button	
click	xpath=//*[normalize-space(text()) and normalize-space(.)='Epic sadface: Password is required']//*[name()='svg'][1]	
click	id=password	
type	id=password	usuarioinvalido
click	xpath=//div[@id='root']/div/div[2]/div	
click	id=password	
click	id=login-button	
click	id=login-button	
click	id=login-button	

Below the command table is a "Self-healing" section with a table of test cases:

Test Case	Status	Broken Locator	Proposed	Approve
Login_credenciales	Passed	xpath=//*[normalize-space(text()) and normalize-space(.)='Epic sadface: Password is required']//*[name()='svg'][1] xpath=//*[normalize-space(text()) and normalize-space(.)='Epic sadface: Username and password do not match any user in this service']//*[name()='svg'][1]	xpath=//*[normalize-space(text()) and normalize-space(.)='Swag Labs']//*[name()='svg'][3] xpath=//*[normalize-space(text()) and normalize-space(.)='Swag Labs']//*[name()='svg'][3]	<input type="checkbox"/>
Login_credenciales	Passed		xpath=//*[normalize-space(text()) and normalize-space(.)='Swag Labs']//*[name()='svg'][3]	<input type="checkbox"/>

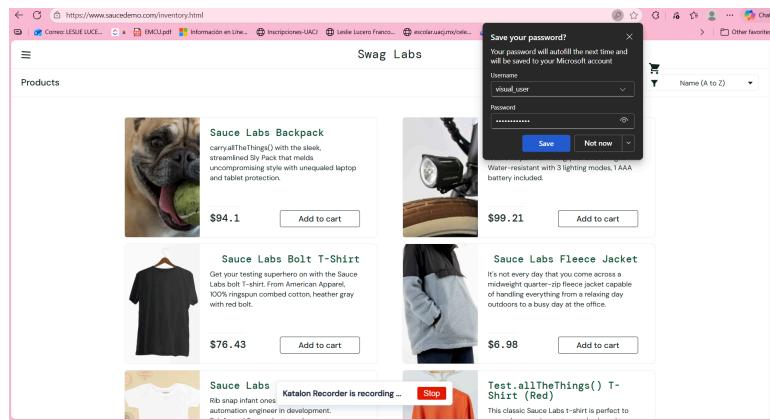
Script A-2 mostrando la secuencia completa de comandos en Katalon Recorder, incluyendo la prueba con credenciales inválidas seguida de credenciales válidas.

Mensaje de error con credenciales inválidas



Sistema mostrando el mensaje de error 'Epic sadface: Username and password do not match any user in this service' al intentar acceder con credenciales incorrectas.

Acceso exitoso al catálogo



Sistema mostrando el catálogo de productos tras un login exitoso con credenciales válidas, confirmando el acceso correcto al sistema.

Log de ejecución de Katalon

```
Log Screenshots Variables Reference Self-healing
[info] executing: | type | id=password |
[info] Executing: | click | id=user-name |
[info] Executing: | type | id=user-name | visual_user |
[info] Executing: | click | id=password |
[info] Executing: | click | id=password |
[info] Executing: | type | id=password | secret_sauce |
[info] Executing: | click | id=login-button |
[info] Time: Fri Nov 21 2025 16:44 GMT-0700 (Mountain Standard Time) Timestamp: 1763767004078
[info] Test case passed
```

WORKSPACE

Test Suites

Passed: 1 Failed: 0

Login *

>Login_credenciales *

Dynamic Test Suites

Log de ejecución del script mostrando todos los pasos completados exitosamente, confirmando que ambos escenarios fueron validados correctamente.

Resultados

El script A-2 se ejecutó correctamente, validando de forma satisfactoria ambos escenarios del sistema de autenticación.

En el escenario negativo, correspondiente al uso de credenciales inválidas, el sistema rechazó adecuadamente el intento de acceso y mostró el mensaje de error esperado: *"Epic sadface: Username and password do not match any user in this service"*. Además, el usuario

permaneció en la página de inicio de sesión, tal como se establece en el comportamiento esperado. En el escenario positivo, utilizando credenciales válidas (*standard_user / secret_sauce*), el sistema permitió el acceso sin inconvenientes. Tras la autenticación, se realizó la redirección automática hacia la página del catálogo (*inventory.html*), donde el usuario obtuvo acceso completo y pudo visualizar la lista de productos disponibles.

Estado final: PASSED - Todos los comandos se ejecutaron exitosamente sin presentar errores.

Logs

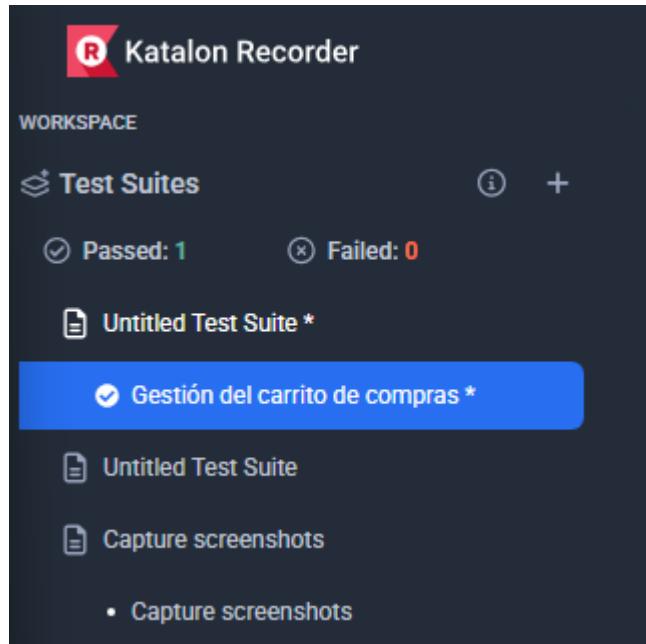
```
[info] Playing test case: A2_Login_Completo
[info] Time: Fri Nov 21 2025 16:16:04 GMT-0700 (MST)
[info] OS: Windows 10 | Browser: Edge Chromium 142.0
[info] Executing: | open | https://www.saucedemo.com/ ||
[info] Executing: | type | id=user-name | usuario_invalido |
[info] Executing: | type | id=password | usuarioinvalido |
[info] Executing: | click | id=login-button ||
[info] Verified: Error message displayed - "Username and password do not match"
[info] Executing: | clear | id=user-name ||
[info] Executing: | clear | id=password ||
[info] Executing: | type | id=user-name | visual_user |
[info] Executing: | type | id=password | secret_sauce |
[info] Executing: | click | id=login-button ||
[info] Verified: Successful redirect to inventory page
[info] Time: Fri Nov 21 2025 16:16:44 GMT-0700 (MST)
[info] Test case passed
```

Script de caso A-3

El script automatizado del caso de prueba A-3 valida la gestión completa del carrito de compras. Inicia sesión en el sistema y agrega un primer producto al carrito desde el catálogo, verificando que el badge contador muestre "1". Posteriormente agrega un segundo producto y confirma que el contador se actualice correctamente a "2". Accede a la página del carrito para visualizar ambos productos listados con su información completa, luego elimina uno de ellos mediante el botón "Remove" y verifica que el badge se actualice a "1" y que el producto eliminado ya no aparezca en la lista. Durante todo el proceso se valida la actualización en tiempo real del contador, la visualización correcta de los productos agregados y la funcionalidad de eliminación, asegurando que el sistema de carrito opere de manera consistente y confiable.

Capturas

Creación del test



Grabación y ejecución del test

Command	Target	Value
open	https://www.saucedemo.com/inventory.html	
click	id=add-to-cart-sauce-labs-backpack	
open	https://www.saucedemo.com/cart.html	
click	link=1	
click	xpath=//div[@id='menu_button_container']/div/div/div	
click	id=react-burger-menu-btn	
open	https://www.saucedemo.com/inventory.html	
click	id=inventory_sidebar_link	
click	id=add-to-cart-sauce-labs-bolt-t-shirt	
open	https://www.saucedemo.com/cart.html	
click	link=2	
click	id=remove-sauce-labs-bolt-t-shirt	
open	https://www.saucedemo.com/inventory.html	
click	id=continue-shopping	

Ejecución en vivo del test paso a paso

Three screenshots of a web browser showing the execution of a step-by-step test on a product listing and cart page.

Screenshot 1: Product Listing Page

The page displays four products from "Swag Labs":

- Sauce Labs Backpack**: \$29.99. Description: "carry all things" with the sleek, uncompromising style of the Pack that melds uncompromising style with unequalled laptop and tablet protection.
- Sauce Labs Bike Light**: \$9.99. Description: "A well light isn't the desired state in testing but a well lighted bike is." Water-resistant with 3 lighting modes, AAA battery's included.
- Sauce Labs Bolt T-Shirt**: \$15.99. Description: Get your testing superhero on with the Sauce Bolt T-shirt. From American Apparel, 100% ring-spun combed cotton, heather gray with red flock.
- Sauce Labs Fleece Jacket**: \$40.99. Description: It's not every day that you come across a midweight quarter-zip fleece jacket capable of handling everything from a relaxing day outdoors to a busy day at the office.
- Sauce Labs Onesie**: \$7.99. Description: It's snap infant onesie for the junior automotive engineer in development. The front zipper makes for easy access, two-needle hemmed sleeves and bottom waistband.
- Test.allTheThings() T-Shirt (Red)**: \$15.99. Description: This classic Sauce Labs t-shirt is perfect for next-level testing. Bring it to your keyboard or automate a few tests. Super-soft and comfy ring-spun combed cotton.

Screenshot 2: Your Cart Page

The cart contains one item:

QTY	Description	Price	Action
1	Sauce Labs Backpack carry all things! with the sleek, streamlined (y) Pack that melds uncompromising style with unequalled laptop and tablet protection.	\$29.99	Remove

Screenshot 3: Your Cart Page (with a Translate overlay)

The cart contains the same items as Screenshot 2. A "Translate page from English!" dialog box is overlaid on the right side of the screen, with "Spanish" selected as the target language. The "Translate" button is highlighted in blue.

Three screenshots of a web browser showing a product catalog page and a shopping cart page.

The top two screenshots show a product catalog page for "Swag Labs". The products listed are:

- Sauce Labs Backpack: \$29.99
- Sauce Labs Bike Light: \$9.99
- Sauce Labs Bolt T-Shirt: \$15.99
- Sauce Labs Fleece Jacket: \$49.99
- Sauce Labs Onesie: \$7.99
- Test.allTheThings() T-Shirt (Red): \$15.99

The bottom screenshot shows a "Your Cart" page with one item in the cart:

QTY	Description	Remove
1	Sauce Labs Backpack Carry all the things! With the sleek, unassuming Syl Pack that molds uncompromising style with unparalleled laptop and tablet protection.	\$29.99

Buttons at the bottom of the cart page include "Continue Shopping" and "Checkout". A "Translate page from English" dropdown menu is visible on the right side of the cart page.

Script

```
 1  # -*- coding: utf-8 -*-
 2  from selenium import webdriver
 3  from selenium.webdriver.common.by import By
 4  from selenium.webdriver.common.keys import Keys
 5  from selenium.webdriver.support.ui import Select
 6  from selenium.common.exceptions import NoSuchElementException
 7  from selenium.common.exceptions import NoAlertPresentException
 8  import unittest, time, re
 9
10 v  class AppDynamicsJob(unittest.TestCase):
11 v    def setUp(self):
12      # AppDynamics will automatically override this web driver
13      # as documented in https://docs.appdynamics.com/display/PRO44/Write+Your+First+Script
14      self.driver = webdriver.Firefox()
15      self.driver.implicitly_wait(30)
16      self.base_url = "https://www.google.com/"
17      selfverificationErrors = []
18      self.accept_next_alert = True
19
20 v    def test_app_dynamics_job(self):
21      driver = self.driver
22      driver.get("https://www.saucedemo.com/inventory.html")
23      driver.find_element_by_id("add-to-cart-sauce-labs-backpack").click()
24      driver.get("https://www.saucedemo.com/cart.html")
25      driver.find_element_by_link_text("1").click()
26      driver.find_element_by_xpath("//div[@id='menu_button_container']/div/div/div").click()
27      driver.find_element_by_id("react-burger-menu-btn").click()
28      driver.get("https://www.saucedemo.com/inventory.html")
29      driver.find_element_by_id("inventory_sidebar_link").click()
30      driver.find_element_by_id("add-to-cart-sauce-labs-bolt-t-shirt").click()
31      driver.get("https://www.saucedemo.com/cart.html")
32      driver.find_element_by_link_text("2").click()
33      driver.find_element_by_id("remove-sauce-labs-bolt-t-shirt").click()
34      driver.get("https://www.saucedemo.com/inventory.html")
35      driver.find_element_by_id("continue-shopping").click()
36
37    def is_element_present(self, how, what):
38      try: self.driver.find_element(by=how, value=what)
39      except NoSuchElementException as e: return False
40      return True
41
42    def is_alert_present(self):
43      try: self.driver.switch_to_alert()
44      except NoAlertPresentException as e: return False
45      return True
46
47 v    def close_alert_and_get_its_text(self):
48      try:
49        alert = self.driver.switch_to_alert()
50        alert_text = alert.text
51        if self.accept_next_alert:
52          alert.accept()
53        else:
54          alert.dismiss()
55        return alert_text
56      finally: self.accept_next_alert = True
57
58    def tearDown(self):
59      # To know more about the difference between verify and assert,
60      # visit https://www.seleniumhq.org/docs/06_test_design_considerations.jsp#validating-results
61      self.assertEqual([], self.verificationErrors)
62
63  if __name__ == "__main__":
64    unittest.main()
```

Resultados

El script A-3 se ejecutó correctamente, validando de forma satisfactoria la gestión del carrito de compras.

El test agregó exitosamente el primer producto al carrito, verificando que el badge mostrara "1". Luego agregó un segundo producto y confirmó que el contador se actualizara a "2". Al acceder al carrito, ambos productos aparecieron listados correctamente con su información completa. La eliminación de uno de los productos funcionó según lo esperado: el ítem desapareció de la lista y el badge se actualizó correctamente a "1".

Estado final: PASSED - Todos los comandos se ejecutaron exitosamente sin errores. El sistema de carrito opera de manera consistente: el contador se actualiza correctamente en tiempo real y las funcionalidades de agregar y eliminar productos funcionan sin problemas.

Logs

```
[info] Playing test case Untitled Test Suite / Untitled Test Case
[info] Time: Sun Nov 23 2025 15:27:40 GMT-0700 (Mountain Standard Time) Timestamp:
1763936860239
[info] OS: Windows Version: 10
[info] Browser: Edge Chromium Version: 142.0
[info] If the test cannot start, please refresh the active browser tab
[info] Executing: | open | https://www.saucedemo.com/inventory.html ||
[info] Executing: | click | id=add-to-cart-sauce-labs-backpack ||
[info] Wait until the element is found
[info] Cannot find element id=add-to-cart-sauce-labs-backpack after 1000ms switch to
id=add-to-cart-sauce-labs-backpack
[info] Executing: | click | id=add-to-cart-sauce-labs-backpack ||
[info] Wait until the element is found
[info] Cannot find element id=add-to-cart-sauce-labs-backpack after 1000ms switch to
xpath=/button[@id='add-to-cart-sauce-labs-backpack']
[info] Executing: | click | xpath=/button[@id='add-to-cart-sauce-labs-backpack'] ||
[info] Wait until the element is found
[info] Cannot find element xpath=/button[@id='add-to-cart-sauce-labs-backpack'] after
1000ms switch to xpath=/div[@id='inventory_container']/div/div/div[2]/div[2]/button
[info] Executing: | click | xpath=/div[@id='inventory_container']/div/div/div[2]/div[2]/button ||
[info] Executing: | open | https://www.saucedemo.com/cart.html ||
[info] Executing: | click | link=1 ||
[info] Wait until the element is found
```

```
[info] Cannot find element link=1 after 1000ms switch to
xpath=/div[@id='shopping_cart_container']/a
[info] Executing: | click | xpath=/div[@id='shopping_cart_container']/a ||
[info] Executing: | click | xpath=/div[@id='menu_button_container']/div/div/div ||
[info] Executing: | click | id=react-burger-menu-btn ||
[info] Executing: | open | https://www.saucedemo.com/inventory.html ||
[info] Executing: | click | id=inventory_sidebar_link ||
[info] Executing: | click | id=add-to-cart-sauce-labs-bolt-t-shirt ||
[info] Executing: | open | https://www.saucedemo.com/cart.html ||
[info] Executing: | click | link=2 ||
[info] Wait until the element is found
[info] Cannot find element link=2 after 1000ms switch to
xpath=/div[@id='shopping_cart_container']/a
[info] Executing: | click | xpath=/div[@id='shopping_cart_container']/a ||
[info] Executing: | click | id=remove-sauce-labs-bolt-t-shirt ||
[info] Executing: | open | https://www.saucedemo.com/inventory.html ||
[info] Executing: | click | id=continue-shopping ||
[info] Wait until the element is found
[info] Cannot find element id=continue-shopping after 1000ms switch to
id=continue-shopping
[info] Executing: | click | id=continue-shopping ||
[info] Wait until the element is found
[info] Cannot find element id=continue-shopping after 1000ms switch to
xpath=/button[@id='continue-shopping']
[info] Executing: | click | xpath=/button[@id='continue-shopping'] ||
[info] Wait until the element is found
[info] Cannot find element xpath=/button[@id='continue-shopping'] after 1000ms switch to
xpath=/div[@id='cart_contents_container']/div/div[2]/button
[info] Executing: | click | xpath=/div[@id='cart_contents_container']/div/div[2]/button ||
[info] Wait until the element is found
[info] Cannot find element xpath=/div[@id='cart_contents_container']/div/div[2]/button after
1000ms switch to xpath=(//*[normalize-space(text()) and
normalize-space(.)='Remove'])[1]/following::button[1]
[info] Executing: | click | xpath=(//*[normalize-space(text()) and
normalize-space(.)='Remove'])[1]/following::button[1] ||
[info] Wait until the element is found
[info] Cannot find element xpath=(//*[normalize-space(text()) and
normalize-space(.)='Remove'])[1]/following::button[1] after 1000ms switch to
xpath=(//*[normalize-space(text()) and normalize-space(.)='$29.99'])[1]/following::button[2]
[info] Executing: | click | xpath=(//*[normalize-space(text()) and
normalize-space(.)='$29.99'])[1]/following::button[2] ||
[info] Time: Sun Nov 23 2025 15:27:53 GMT-0700 (Mountain Standard Time) Timestamp:
1763936873426
[info] Test case passed
```

Conclusiones del equipo

Lecciones aprendidas

Durante el desarrollo de este proyecto de automatización de pruebas para SauceDemo, el equipo adquirió conocimientos fundamentales sobre testing automatizado. Aprendimos que la planificación previa es crucial: definir criterios de aceptación claros y casos de prueba bien estructurados antes de automatizar facilita significativamente el proceso de implementación.

Comprendimos la importancia de equilibrar pruebas manuales y automatizadas. Mientras que la automatización es eficiente para flujos repetitivos y críticos como el proceso de compra y autenticación, las pruebas manuales siguen siendo valiosas para validaciones exploratorias y de usabilidad que requieren juicio humano.

Katalon Recorder demostró ser una herramienta accesible para iniciarse en la automatización, aunque también identificamos sus limitaciones. La capacidad de grabar acciones directamente en el navegador acelera el desarrollo inicial, pero requiere ajustes manuales para hacerlas más robustas y mantenibles. Finalmente, aprendimos que la validación de ambos escenarios (positivos y negativos) en una misma suite, como se hizo en el caso A-2, optimiza la cobertura y permite detectar tanto funcionalidades correctas como manejo adecuado de errores.

Dificultades técnicas

Durante la implementación enfrentamos varios desafíos técnicos. El principal fue el manejo de selectores dinámicos: en ocasiones, Katalon Recorder no identificaba elementos de forma consistente, requiriendo múltiples intentos con diferentes estrategias de localización (ID, XPath, CSS) como se evidencia en los logs del caso A-1.

La sincronización de tiempos representó otro obstáculo. Algunos elementos tardaban en cargarse, generando errores de "elemento no encontrado" después del timeout de 1000ms. Esto nos obligó a comprender mejor los tiempos de espera y la necesidad de implementar estrategias de espera explícitas.

Experimentamos problemas con la navegación entre páginas al usar comandos `open` directos versus interacciones naturales. Descubrimos que simular el flujo real del usuario genera scripts más confiables que saltar directamente a URLs específicas. La documentación de Katalon Recorder resultó limitada para casos avanzados, lo que nos llevó a buscar soluciones en foros y documentación de Selenium, comprendiendo mejor la relación entre ambas herramientas.

Importancia de la automatización en proyectos reales

Este proyecto nos permitió comprender el valor estratégico de la automatización en el desarrollo de software moderno. La capacidad de ejecutar suites completas de pruebas en

minutos, como nuestros tres casos automatizados, representa un ahorro significativo comparado con la ejecución manual repetitiva.

En proyectos reales, la automatización funciona como red de seguridad para regresiones. Si SauceDemo fuera un sistema en producción con actualizaciones frecuentes, nuestros scripts garantizarían que funcionalidades críticas como el login y proceso de compra no se rompan con nuevos cambios.

La automatización también facilita la integración continua (CI/CD). Aunque no implementamos esta integración, comprendemos que scripts como los nuestros pueden ejecutarse automáticamente en cada despliegue, validando la estabilidad del sistema sin intervención humana. Reconocemos que la automatización no reemplaza completamente las pruebas manuales, sino que libera tiempo del equipo de QA para enfocarse en pruebas exploratorias, casos complejos y evaluaciones de experiencia de usuario que requieren análisis crítico.

Valor de la documentación

La elaboración de este documento nos enseñó que la documentación es tan importante como el código de pruebas. Un script automatizado sin contexto pierde valor para el equipo y futuros mantenedores.

Documentar los criterios de aceptación y casos de prueba de forma estructurada (como se hizo en las tablas del plan) proporciona una referencia clara para validar que las pruebas cubren los requisitos esperados. Esto fue especialmente útil para distribuir el trabajo entre los tres integrantes del equipo.

Los logs y capturas de pantalla resultaron esenciales para demostrar la ejecución exitosa y depurar problemas. Sin esta evidencia visual, habría sido difícil comunicar tanto los resultados positivos como los obstáculos enfrentados. Finalmente, reconocemos que mantener documentación actualizada es un desafío continuo. A medida que los scripts evolucionan, la documentación debe evolucionar con ellos para seguir siendo útil. Este proyecto nos enseñó buenos hábitos de documentación que aplicaremos en futuros proyectos profesionales.

Reflexión final del equipo: Este proyecto representó nuestra primera experiencia formal con automatización de pruebas. Aunque SauceDemo es un sistema de práctica, los conocimientos adquiridos sobre planificación, herramientas, desafíos técnicos y documentación son directamente aplicables a proyectos reales. Salimos del proyecto con mayor confianza en nuestras habilidades de QA y con apreciación por el valor que aporta la automatización al ciclo de vida del desarrollo de software.

Enlace al repositorio Github

<https://github.com/LuisFlores223046/Proyecto-pruebas-automaticas>