

PRUEBA : MARIA DEL MAR POLO URRIAGO

Valores Corporativos: “lo básico que esperamos de todos nuestros compañeros

1. . Prueba Técnica QA Junior

- para la siguiente prueba técnica no es indispensable que tenga un periodo de entrega lo más importante es que la puedan resolver, aprender y cuestionarse, la prueba se realiza con el fin de conocer su experiencia y conocimientos no solo teóricos sino también prácticos en el que hacer día a día como tester funcionales, es importante que no se limiten resuelvan esta prueba si quieren utilizando herramientas que conozcan o recomendando herramientas

1. para iniciar con la prueba deben ingresar a la siguiente Url
<https://www.saucedemo.com/> y realizar una prueba exploratoria con cada uno de los usuarios que se encuentran allí para que conozcan el flujo completo del ecommerce
2. teniendo ya claro el flujo con cada uno de los usuarios, deben crear una estrategia de pruebas que van a seguir, esta debe incluir también cómo van a realizar el seguimiento de los bugs encontrados
3. deberán crear los casos de prueba correspondientes a cada usuario en los que deberán abarcar los flujos completos del ecommerce, los casos de pruebas se deben crear y subir a un repositorio compartido público(github), para esto pueden crearlos en una plantilla de su preferencia o con algún lenguaje de texto plano con estructura
4. en ese mismo repositorio van a subir los bugs encontrados y documentados, al terminar la prueba el repositorio debe ser compartido al siguiente correo javier.morales@grupor5.com tener en cuenta que el repositorio sea público para poder acceder y revisar la prueba
5. este punto no es obligatorio realizarlo, pero sera un plus para la prueba, consta de automatizar al menos 3 casos de prueba de los que tienen creados para las pruebas manuales, en la empresa actualmente estamos utilizando el framework Selenium-python-cucumber (opcional), subir al mismo repositorio de la prueba

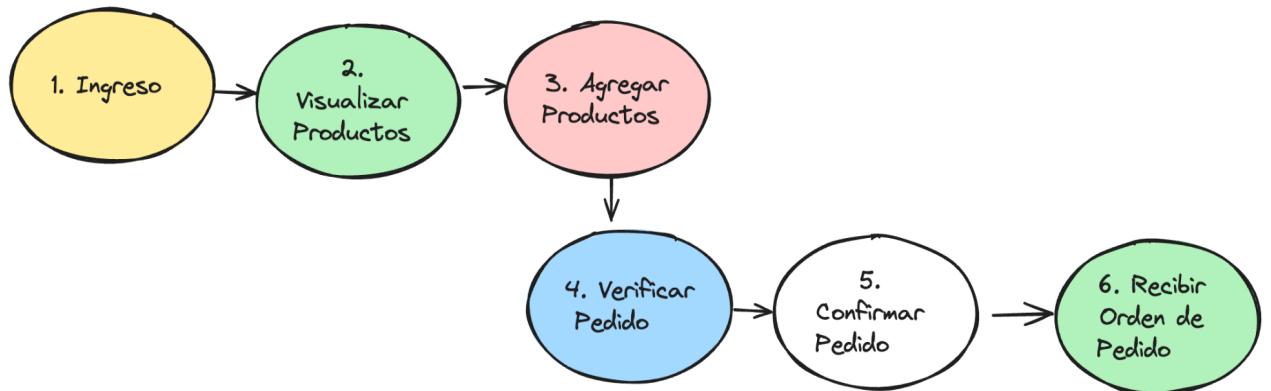
SOLUCIÓN DE LA PRUEBA

1. Se realizan pruebas exploratorias con cada usuario para evidenciar el flujo correspondiente de un Ecommerce.

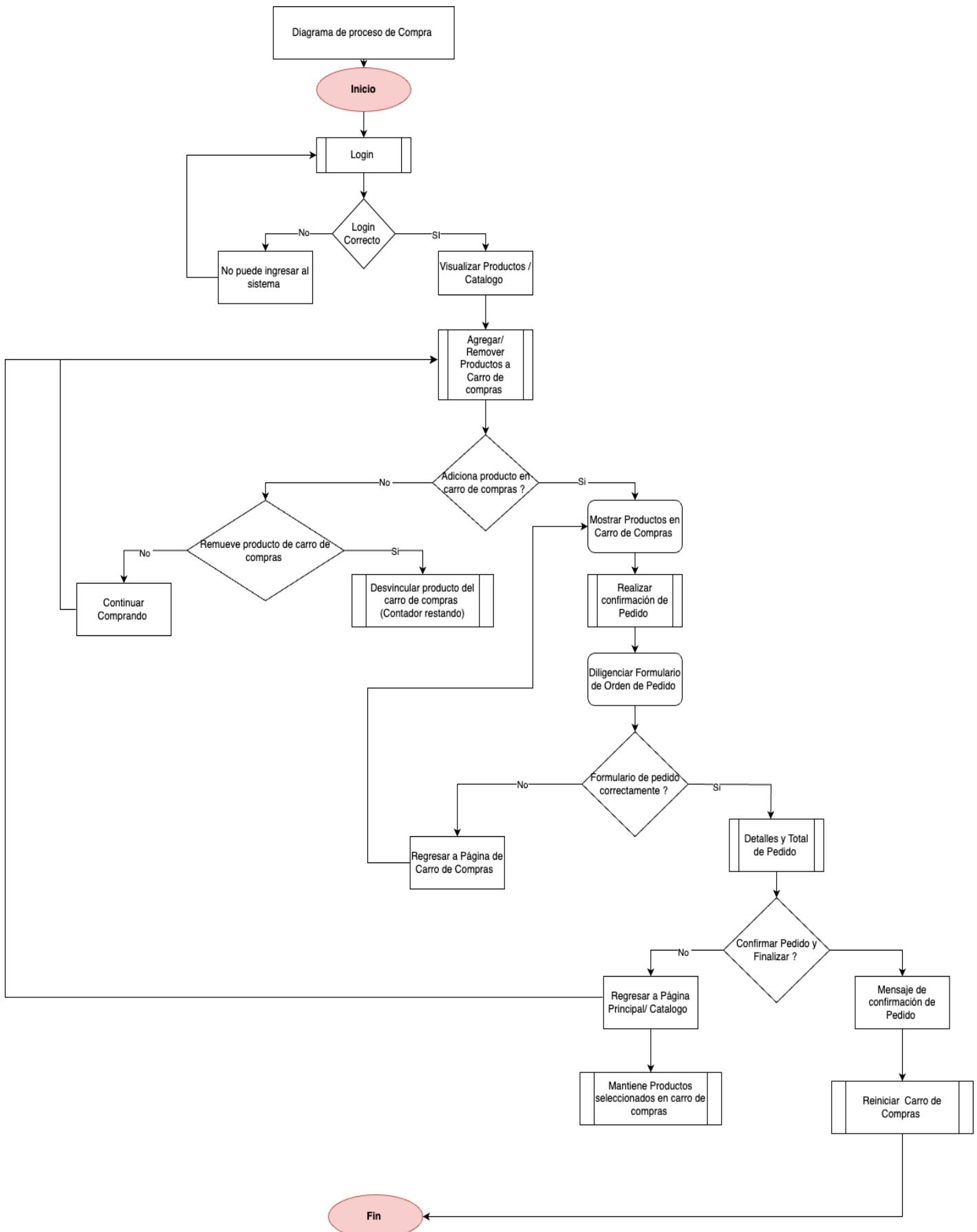
Usuario	Exploración - Evidencia
standard user	Usuario que puede acceder satisfactoriamente al catálogo de productos, puede evidenciar todos los productos disponibles con su información correspondiente, puede ingresar a cada producto, realizar los cuatro filtros de ordenamiento de productos, añadir y remover productos desde la card (ids: 4,0,1,5,2,3), acceder al carro de compras, remover producto desde el carro de compras, regresar a la página principal con el botón “continue Shopping”, realizar checkout de los productos existentes en el carro de compras, cancelar checkout y regresar al carro de compras, diligenciar formulario de orden de pedido,cancelar orden de pedido, finalizar pedido, regresar a la página principal una vez realizada la orden de pedido, interactuar con el menú lateral de la página principal y hacer logout.
locked_out_user	Usuario que al ingresar “Username” y “password” en validación de cuenta se evidencia que dicho usuario ha sido bloqueado.
problem_user	Usuario que puede acceder satisfactoriamente al catálogo de productos, puede evidenciar todos los productos disponibles con su información correspondiente pero con misma imagen de producto para todos, al ingresar a cada producto se presentan defectos en el botón “remove” o “add to cart”,no puede realizar los cuatro filtros de ordenamiento de productos, sólo puede añadir y remover 3 productos del catálogo de productos (ids: 5, 1, 3) desde la card, acceder al carro de compras, remover producto desde el carro de compras, regresar a la página principal con el botón “continue Shopping”, realizar checkout de los productos existentes en el carro de compras, cancelar checkout y regresar al carro de compras, no puede diligenciar formulario de orden de pedido por defectos encontrados,cancelar orden de pedido,no puede finalizar flujo de pedido, interactuar con el menú lateral de la página principal y hacer logout.
performance_glitche_user	Se evidencia latencia en las acciones de ingresar a la página, confirmar pedido y regresar al home luego de realizar un pedido.
error_user	Usuario que puede acceder satisfactoriamente al catálogo de productos, puede evidenciar todos los productos disponibles con su información correspondiente, al ingresar a cada producto se presentan defectos en el botón “remove” o “add to cart”, no puede realizar los cuatro filtros de ordenamiento de productos por defectos encontrados, sólo puede añadir y remover 3 productos del catálogo de productos (ids: 4, 0, 2) desde la card, acceder al carro de compras, remover producto desde el carro de compras, regresar a la página principal con el botón “continue Shopping”, realizar checkout de los productos existentes en el carro de compras, cancelar checkout y regresar al carro de compras, puede diligenciar formulario de orden de pedido pero con defectos encontrados,cancelar orden de pedido, no puede finalizar

	flujo de pedido, interactuar con el menú lateral de la página principal y hacer logout.
visual_user	<p>Usuario que puede acceder satisfactoriamente al catálogo de productos, puede evidenciar todos los productos disponibles con su información correspondiente sin embargo un producto (id: 4) no se evidencia con la imagen correcta, puede ingresar a cada producto y se evidencia correcto funcionamiento de los botones “remove” y “add to cart”, realizar los cuatro filtros de ordenamiento de productos, añadir y remover productos desde la card (ids: 4,0,1,5,2,3), acceder al carro de compras, remover producto desde el carro de compras, regresar a la página principal con el botón “continue Shopping”, realizar checkout de los productos existentes en el carro de compras, cancelar checkout y regresar al carro de compras, diligenciar formulario de orden de pedido, cancelar orden de pedido, finalizar pedido, regresar a la página principal una vez realizada la orden de pedido, interactuar con el menú lateral de la página principal y hacer logout.</p> <p>Se evidencian errores de visualización (front)</p>

1.1. Steps principales del flujo evidenciado



1.2. DIAGRAMA DE FLUJO



2. ESTRATEGIA DE PRUEBAS

La estrategia de pruebas diseñada para la página que he denominado 'Swag Labs' se enfoca en asegurar la calidad y funcionalidad de la plataforma de comercio electrónico en cada etapa del flujo del usuario. La planificación de este plan de pruebas inicia identificando el debido proceso y explorando cada flujo con los usuarios dados para la creación de casos de prueba manuales y automatizados.

El flujo del usuario, desde el ingreso a la página hasta la recepción de la orden de compra, se estructurarán los casos de prueba necesarios utilizando metodología Gherkin, para es La ejecución de casos de prueba se abordará el flujo completo contemplando casos válidos y casos inválidos dónde su resultado de dicha ejecución y los resultados se documentará en la herramienta Testlodge, y dado que se tiene la herramienta Trello para la gestión de tareas del proyecto se documentaran los Bugs en el tablero ágil "Backlog Bugs" para que los miembros del equipo de desarrollo puedan dar solución.

2.1. Fase de Planificación:

- Revisión del Flujo:**

Analizar detenidamente el flujo del usuario en cada paso: ingreso, visualización de productos, agregar productos, verificar pedido, confirmar pedido y recibir orden de pedido.

Identificar posibles puntos críticos y áreas de riesgo en cada paso del flujo.

2.2. Creación de Casos de Prueba:

- Ingreso a la Página:**

- ❖ Validar la carga exitosa de la página de inicio.
- ❖ Verificar la correcta presentación de la interfaz de usuario y la disponibilidad de elementos clave (botones de inicio de sesión, menús de navegación, etc.).

- Visualizar Productos:**

- ❖ Crear casos de prueba para garantizar que los productos se carguen

correctamente.

- ❖ Verificar el ordenamiento de los productos por orden alfabético y precio menor y mayor
- ❖ Confirmar que la información del producto (nombre, precio, imagen) se muestre correctamente.

- **Agregar Productos:**

- ❖ Desarrollar casos de prueba para agregar productos al carrito.
- ❖ Validar que el contador del carrito se actualice correctamente.
- ❖ Verificar que los productos seleccionados aparezcan correctamente en el carrito.

- **Verificar Pedido:**

- ❖ Crear casos de prueba para revisar el resumen del pedido antes de la confirmación.
- ❖ Validar que los productos seleccionados y la cantidad sean correctos.
- ❖ Verificar que el total del pedido sea preciso.

- **Confirmar Pedido:**

- ❖ Desarrollar casos de prueba para la autorización y confirmación del pedido.
- ❖ Validar las opciones de pago y la integración con pasarelas de pago.
- ❖ Confirmar que se genere una confirmación de pedido y se muestre al usuario.

- **Recibir Orden de Pedido:**

- ❖ Crear casos de prueba para confirmar que el usuario recibe una confirmación de orden de pedido
- ❖ Crear casos de prueba que el usuario pueda volver al menú

2.3. Ejecución de Casos de Prueba:

En la ejecución de casos de prueba válidos y no válidos de cada step del punto 2.2. utilizaremos los siguientes tipos de pruebas:

- **Pruebas Funcionales:** Verificar que cada función de la plataforma funcione

según lo esperado.

- **Pruebas de Usabilidad:** Evaluar la facilidad de uso del sistema en cada paso del flujo del usuario.
- **Pruebas de Interfaz de Usuario (UI):** Confirmar que la interfaz de usuario sea coherente y atractiva.

2.4. Reporte de Resultados:

- **Ingreso a la Página:** Documentar los resultados de la prueba de inicio de sesión, incluyendo casos válidos e inválidos.
- **Visualizar Productos:** Registrar resultados de la prueba de visualización de productos, incluyendo casos de éxito y fallos.
- **Agregar Productos:** Documentar la ejecución de casos de prueba para agregar productos al carrito
- **Verificar Pedido:** Informar sobre la revisión del resumen del pedido.
- **Confirmar Pedido:** Documentar la ejecución de la autorización y confirmación del pedido.
- **Recibir Orden de Pedido:** Registrar los resultados de la prueba de recepción de orden o pedido realizado.

2.5. Reporte de Bugs en Trello:

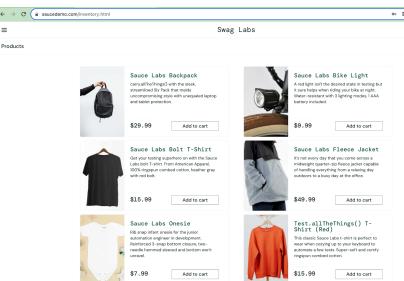
URL Tablero Trello : <https://trello.com/b/K85Yaw4n/proyecto-ecommerce-r5>

- Ingreso a la Página: Reportar cualquier problema relacionado con el inicio de sesión en una tarjeta de Trello.
- Visualizar Productos: Crear tarjetas para problemas de visualización de productos.
- Agregar Productos: Reportar cualquier error relacionado con la adición de productos al carrito.
- Verificar Pedido: Crear tarjetas para problemas en la revisión del resumen del pedido.
- Confirmar Pedido: Reportar bugs relacionados con la autorización y confirmación del pedido.
- Recibir Orden de Pedido: Crear tarjetas para problemas en la recepción de orden o pedido realizado.

3. CASOS DE PRUEBA

Los casos de prueba se podrán evidenciar en un archivo de la carpeta Documents del proyecto ‘PythonProject’ alojado en mi github <https://github.com/mademap/test-R5>.

Haré mención de los casos en la tabla que se muestra a continuación para denotar la ejecución y el resultado obtenido.

Feature	Case	Expected Result	Result	Status
<u>usuario</u> <u>puede</u> <u>loguearse en</u> <u>la página</u>	<u>Scenario: Login correcto</u> <u>Given un usuario que introduce</u> <u><user_name> e introduce una contraseña en el</u> <u>campo <password></u> <u>When se hace click en el <boton></u> <u>Then debería cargarse la <página></u> <u>And los siguientes id de: <productos></u>	Ingreso correctamente con el usuario ‘standard_user’ y cargar la página principal (Inventory) listando correctamente las cards de productos de la página.		Satisfactorio

Scenario: Login incorrecto
Given un usuario que introduce su <user_name> e introduce una contraseña en el campo <password>
When se hace click en el <boton>
Then debería de mostrarse el mensaje <mensaje>
And no permitir el acceso a la pagina.

Validación de 'campos requeridos' en el formulario de logueo cuando no se digita ninguno de éstos y no permitir ingresar a la página.

Swag Labs

The screenshot shows a login form with two fields: 'standard_user' and 'Password'. Both fields have red error messages: 'Epic sadface: standard_user is required' and 'Epic sadface: Password is required'. A green 'Login' button is at the bottom.

Accepted usernames are:
standard_user
locked_out_user
problem_user
user1, user2, glitch_user
error_user
visual_user

Password for all users:
secret_sauce

Swag Labs

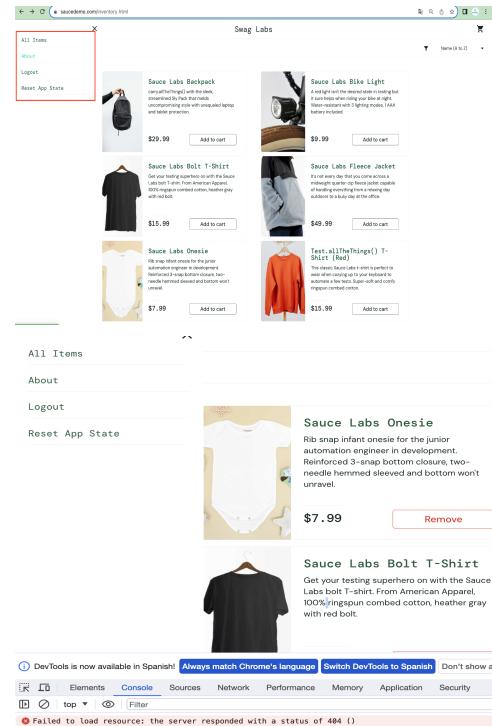
The screenshot shows a login form with two fields: 'Username' and 'Password'. Both fields have red error messages: 'Epic sadface: Username is required' and 'Epic sadface: Password is required'. A green 'Login' button is at the bottom.

Satisfactorio

Feature:
usuario
puede
observar
menú
lateral

Scenario: navegacion menu lateral
Given como un usuario logueado
satisfactoriamente
And se encuentra en el menu lateral izquierdo de la pagina
When se hace click a la siguiente <opcion_menu>
Then debería mostrar la siguiente URL <URL>

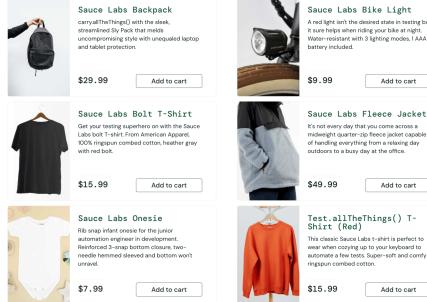
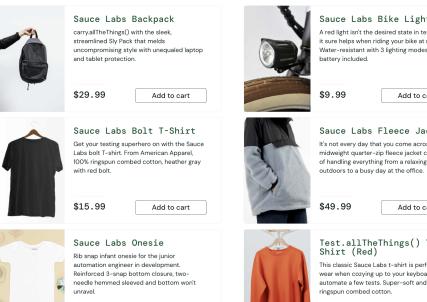
Direccionar al usuario 'standard_user' correctamente a la URL correspondiente en cada opción del menú lateral de la página

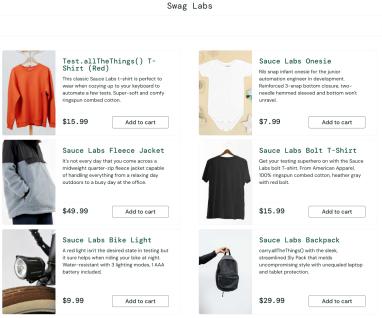
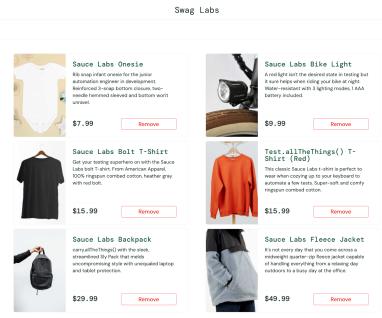


satisfactorio

En las opciones : 'All items', 'About', 'Logout' del menú se realiza el direccionamiento correcto.

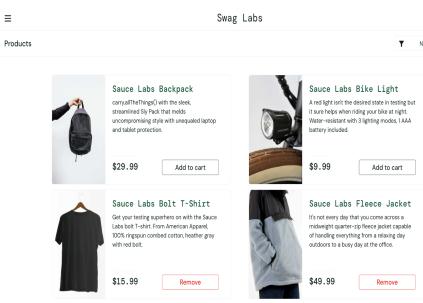
En la opción del menú 'Reset App State' no se evidencia funcionalidad y sale un error 404 en inspeccionador de consola

<p>Feature: usuario puede listar productos</p>	<p>Scenario: Validar carga exitosa de la página de Inventory Given El usuario accede a la página de Swag Labs When La página se carga completamente Then El usuario visualiza la página</p>	<p>Visualizar correctamente el catálogo de productos cuando se loguea el usuario 'standard_user'</p>		<p>satisfactorio Una vez se ingresa a la página se listan todos los productos en la página principal</p>
	<p>Scenario: ver cuadricula de productos Given como un usuario When se loguea satisfactoriamente Then debería observar la lista de productos con los siguientes parametros: (foto, nombre, descripción, precio) And un boton "add to cart".</p>	<p>Visualizar los datos de foto,nombre,descripción,precio de los productos de la página principal y un botón de 'Add to Cart' en cada card cuando se loguea el usuario 'standard_user'</p>		<p>satisfactorio se visualiza los datos y el botón en el listado principal o catálogo</p>
	<p>Scenario: Confirmar que la información del producto se muestra correctamente Given El usuario está en la página de productos de Swag Labs When se selecciona un producto Then Debería poder ver información precisa sobre el nombre, descripción, precio e imagen de cada producto</p>	<p>Visualizar los datos de nombre, descripción, precio e imagen del producto cuando el usuario 'standard_user' selecciona un producto desde la página principal</p>		<p>satisfactorio Se visualiza la información en cada producto del catalogo</p>

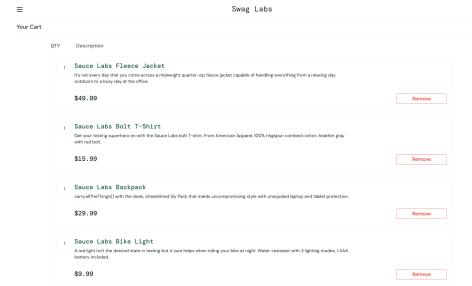
	<p><u>Scenario: ordenamiento de productos</u> <u>Given como un usuario logueado satisfactoriamente</u> <u>When se hace click en la lista desplegable "Name": <lista></u> <u>Then debería ordenarse los productos así: <id_productos></u></p>	<p>Ordenar los productos del catálogo de producto en orden de la A a Z, de Z a A, de Menor a Mayor precio, de Mayor a Menor precio cuando el usuario 'standard_user' liste las opciones de ordenamiento</p>		<p>A to Z: Satisfactorio</p> <p>Z to A: Satisfactorio</p> <p>Price low -High : Satisfactorio</p> <p>Price High-low: Satisfactorio</p>
<p><u>Feature:</u> <u>usuario</u> <u>puede</u> <u>agregar/remover</u> <u>productos</u></p>	<p><u>Scenario: agregar productos al carro de compras</u> <u>Given como un usuario logueado satisfactoriamente</u> <u>When se hace click en el boton "Add to cart" de un producto</u> <u>Then debe agregarse el producto al carro de compras.</u></p>	<p>Agregar los productos seleccionados por el usuario 'standard_user' al carro de compras</p>		<p>Satisfactorio: Se agregan correctamente todos uno o todos los productos al carro de compras</p>

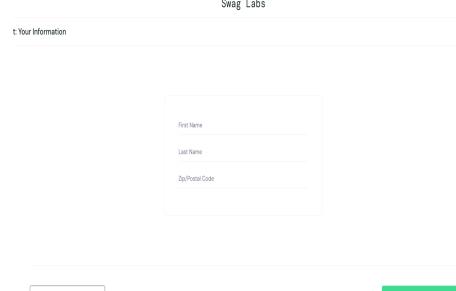
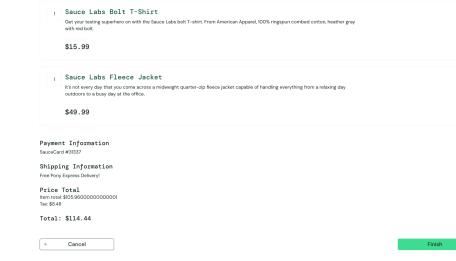
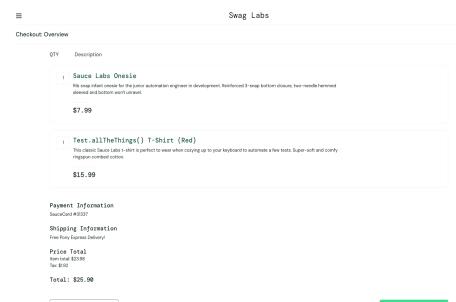
Scenario: remover productos desde card
Given como un usuario loqueado satisfactoriamente
And tiene un producto agregado
When se hace click en el boton "remove" de un producto que no desea
Then debe eliminarse el producto del carrito de compras
And el boton pasara a llamarse "Add to cart"

Eliminar un producto adicionado en el carro de compras cuando el usuario 'standard_user' da click en el boton 'Remove' de la card y habilitar el boton 'add to cart'



Satisfactorio:
Se evidencia funcionamiento correcto de remover productos adicionados al carro previamente y se habilita el botón 'add to cart'

	<p><u>Scenario: Remover producto del carro de compras</u></p> <p><u>Given el usuario logueado satisfactoriamente</u></p> <p><u>And tiene un producto agregado al carro de compras</u></p> <p><u>When da clic en el 'Remover' de un producto desde el carro de compras</u></p> <p><u>Then debe descontarse las unidades</u></p> <p><u>And podrá volver a la página principal</u></p>	<p>Contabilizar las unidades en el carro de compras y eliminar el registro de un producto del carro de compras cuando el usuario 'standard_user' decida eliminar un producto desde la página del carro de compras.</p>		<p>Satisfactorio:</p> <p><u>Se agregan varios productos al carro de compras, posteriormente se remueven y se evidencia el contador de productos disminuyendo las cantidades correctamente.</u></p>
	<p><u>Scenario: Verificar que los productos seleccionados aparezcan correctamente en el carro</u></p> <p><u>Given El usuario ha agregado productos al carro</u></p> <p><u>When Accede al carro de compras</u></p> <p><u>Then Deberían aparecer los productos seleccionados correctamente en el carro</u></p>	<p>Evidenciar todos los productos agregados desde la card del producto al carro de compras cuando el usuario 'standard_user' dé click en el icono de carro de compras</p>		<p>Satisfactorio:</p> <p><u>Se agregan uno y varios productos al carro de compras y se visualiza en cada adición que se adicionan de forma correcta en el carro de compras</u></p>

<p>Feature: usuario puede hacer checkout a su carrito de compras</p>	<p>Scenario: checkout valido Given como un usuario logueado satisfactoriamente And tiene productos agregados en el carrito de compras When se hace click en el boton "Checkout" Then deberá aparecer un formulario para digitar su informacion And el resumen de la compra, para finalizarlo se da click en el boton "Finish" And observa el siguiente mensaje "Thank you for your order!"</p>	<p>Evidenciar y diligenciar el formulario de datos de envío luego de que el usuario 'standard_user' decida confirmar el pedido que tiene en su carro de compras para que luego pueda evidenciar el resumen del pedido (productos, precios, #Id de pedido).</p>	 	<p>Satisfactorio: Se realiza confirmación de pedido a comprar, se evidencia y diligencia formulario de datos de envío, se visualizan los datos completos de los productos seleccionados, se totaliza el valor correctamente.</p>
	<p>Scenario: resumen del pedido antes de la confirmación Given El usuario ha agregado productos al carrito When Revisa el resumen del pedido antes de la confirmación Then Debería ver la lista de productos seleccionados y su cantidad</p>	<p>Listar los productos seleccionados para comprar con su información correspondiente cuando el usuario 'standar_user' decida confirmar una compra y diligenciar los datos de envío requeridos</p>		<p>Satisfactorio: Se evidencia la información correspondiente de los productos confirmados a comprar</p>

Scenario: Verificar que el total del pedido sea preciso
Given El usuario ha revisado el resumen del pedido
When Confirma la compra
Then La página de confirmación debería mostrar el total del pedido de forma precisa

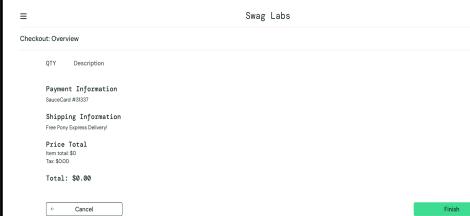
Totalizar correctamente el precio de la compra de acuerdo a la suma de los valores de cada producto a comprar + el impuesto cuando el usuario 'standar_user' esté en el paso de confirmar compra.

Payment Information
SavedCard #4337
Shipping Information
Free Next Day Delivery!
Price Total
Tax Total
Total: \$114.44
+ Cancel

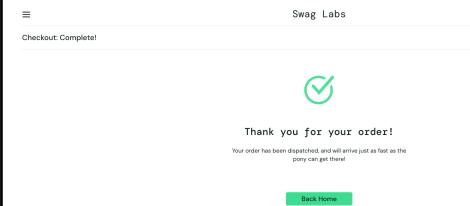
Falló:
Se realizan varias compras una con algunos productos, otra con todos los productos y se realiza la suma del precio de cada producto a comprar + el impuesto y el precio de compra total se computa incorrectamente.

Scenario: checkout invalido
Given como un usuario logueado satisfactoriamente
And no tiene productos en el carrito de compras
When se hace click en el boton "Checkout"
Then debera aparecer un formulario para digitar su informacion
And el resumen de la compra en cero sin finalizar.

Permitir al usuario 'standard_user' hacer 'Checkout' a un pedido sin productos agregados en el carro de compras



The screenshot shows the 'Checkout Overview' page from the Swag Labs website. It displays payment and shipping information sections, both of which are currently empty. The total price is listed as \$0.00. There are 'Cancel' and 'Finish' buttons at the bottom.



The screenshot shows the 'Checkout Complete' page from the Swag Labs website. It features a large green checkmark icon and a message saying 'Thank you for your order! Your order has been dispatched, and will arrive just as fast as the pony can get there!'. A 'Back Home' button is visible at the bottom.

Satisfactorio:
Se evidencia que al no tener productos agregados en el carro de compras se puede hacer checkout y completar el flujo de compra normalmente.

Se reporta Bug ya que es considerado un checkout invalido.

4. BUGS:

Los bugs se encontrarán documentados en el tablero de Trello establecido para el proyecto.

Estarán clasificados por tipo de Bug y Prioridad

- Bug Bloqueante : Aquel defecto encontrado que impide cualquier acción del flujo completo de la página
- Bug Incoherente : Aquel defecto encontrado que realiza alguna acción- operación pero de manera distorsionada o incorrecta pero no se impide el flujo completo de la página
- Bug Trivial : Aquel defecto visual, de diseño o ortografía.

Los bugs en el tablero estarán clasificados de la siguiente manera :

- Etiqueta Roja : Bug Bloqueante - Prioridad Alta
- Etiqueta Naranja : Bug Incoherente - Prioridad Media
- Etiqueta Amarilla : Bug Trivial - Prioridad Baja

URL : <https://trello.com/b/K85Yaw4n/proyecto-eccomerce-r5>

5. AUTOMATIZACIÓN

Realicé la automatización de tres escenarios utilizando Python y Selenium y un enfoque de BDD (Behavior-Driven Development) con el lenguaje Gherkin.

El proyecto creado se denomina ‘PythonProject’ se encuentra en mi GitHub <https://github.com/mademap/test-R5> y contiene la configuración y código fuente necesario para automatizar los siguientes escenarios :

```
Scenario: Successful Login
    Given a standard_user is a customer
    When they enter their username and password
    Then they click the login button and log in

Scenario: Buy Product
    Given standard_user logs in successfully
    And selects a product
    When confirm the product
    And fill in the order information
    Then they should be able to complete the purchase
```

Scenario: Remove Product from Shopping Cart

Given standard_user logs in correctly
And has a product added to the shopping cart
When they click the remove button for a product they dont want to buy
Then they will return to the main product page