

EJERCICIO 1:

Realizar una prueba funcional automatizada (Prueba E2E) de un flujo de compra en la página

<https://www.saucedemo.com/> que incluya:

- Autenticarse con el usuario: `standard_user` y password: `secret_sauce`
 - Agregar dos productos al carrito
 - Visualizar el carrito
 - Completar el formulario de compra
 - Finalizar la compra hasta la confirmación: **"THANK YOU FOR YOUR ORDER"**
-
- Como se muestra en la imagen se ha agregado lo solicitado agregando productos al carrito y la autenticación necesaria para loguearse.

```
@ejemplo
Scenario Outline: Agregar articulo al carrito de compras
    Given que un nuevo cliente accede hasta la web de compras
    When el ingresa su <user> y <password>
    And el da click en Login
    And el agrega la cantidad <cantidad> de productos al carrito
    And el ve los productos listados en el carro de compras
    And el da click en Checkout
    And el completa el formulario con los datos <nombre>, <apellido> y <postal>
    And el da click en Continue
    And el da click en Finish
    Then el visualiza el mensaje de confirmacion <mensaje>
Examples:
    | user          | password      | cantidad | nombre | apellido | postal | mensaje                               |
    | standard_user | secret_sauce | 2        | Kat    | Sanchez  | 11111  | Thank you for your order             |
```

- Configuración a nivel de cabecera

```
package stepdefinitions;

import ...

public class RetoSteps {

    @Managed(driver = "chrome")
    private WebDriver navegador;
    private Actor actor = Actor.named("Kath");
    private HomeLogin homeLogin = new HomeLogin();
```

- Registro de cliente nuevo

```
@Given("^que un nuevo cliente accede hasta la web de compras$")
public void queUnNuevoClienteAccedeHastaLaWebDeCompras() {
    actor.can(BrowseTheWeb.with(navegador));
    actor.wasAbleTo(Open.browserOn(homeLogin));
}
```

- Autenticarse con el usuario: standard_user y password: secret_sauce

```
@When("^el ingresa su (.*?) y (.*?)$")
public void elIngresaSuUserYPassworD(String user, String password) {
    actor.wasAbleTo(
        Login.realizarLogin(user, password)
    );
}
```

- Agregar dos productos al carrito

```
@And("^el da click en Login$")
public void elDaClickEnLogin() { actor.attemptsTo(Click.on(BUTTON_LOGIN)); }

@And("^el agrega la cantidad (.*?) de productos al carrito$")
public void elAgregaLaCantidadCantidadDeProductosAlCarrito(String cantidad) {
    actor.wasAbleTo(
        SelectProducts.elegirCantidadProductos(LISTADO_PRODUCTOS, cantidad)
    );
}
```

- Visualizar el carrito
- Completar el formulario de compra

```
@Then("^el ve los productos listados en el carro de compras$")
public void elVeLosProductosListadosEnElCarroDeCompras() { actor.attemptsTo(Click.on(ICONO_CARRITO)); }

@And("^el da click en Checkout$")
public void elDaClickEnCheckout() { actor.attemptsTo(Click.on(BOTON_CHECKOUT)); }

@And("^el completa el formulario con los datos (.*) (.*) y (.*)$")
public void elCompletaElFormularioConLosDatosNombreApellidoYPostal(String nombre, String apellido, String codigoPostal) {
    actor.wasAbleTo(
        LlenarFormulario.cantidad(nombre, apellido, codigoPostal)
    );
}
```

- Finalizar la compra hasta la confirmación: “THANK YOU FOR YOUR ORDER”

```
@And("^el da click en Continue$")
public void elDaClickEnContinue() { actor.attemptsTo(Click.on(BOTON_CONTINUE)); }

@And("^el da click en Finish$")
public void elDaClickEnFinish() { actor.attemptsTo(Click.on(BOTON_FINISH)); }

@Then("^el visualiza el mensaje de confirmacion (.*)$")
public void elVisualizaElMensajeDeConfirmacionMensaje(String mensaje) {
    actor.should(seeThat(MostrarMensaje.mensajeMostrado(MENSAJE_FINAL), containsString(mensaje.toUpperCase())));
}
```

user	password	cantidad	nombre	apellido	postal	mensaje
standard_user	secret_sauce	2	Kat	Sanchez	11111	Thank you for your order