

# UI TESTING

Esta práctica se ha centrado en el testing de la página web [opencart.abstracta.us](https://opencart.abstracta.us) utilizando las diversas técnicas de test explicadas en la asignatura. Para automatizar las diferentes acciones realizadas en la web, hemos utilizado *Selenium Web Driver*, y para definir los test cases, hemos utilizado el framework *Cucumber* para Java.

GitHub: <https://github.com/1631153/UI-Testing>

*(El enunciado pone que la práctica debe ser privada. En el momento de acabar este documento, te hemos enviado una invitación que iremos renovando hasta que la aceptes. Si en el momento de revisar la práctica no tienes acceso, por favor, envía un correo a [1631153@uab.cat](mailto:1631153@uab.cat), y lo pongo en público o vuelvo a enviar invitación. Suelo contestar casi de inmediato. Y si eso pasa, disculpa los problemas)*

## INDICE

<b>FEATURE 1: Registro de usuario .....</b>	<b>2</b>
<b>FEATURE 2: Inicio de sesión .....</b>	<b>5</b>
<b>FEATURE 3: Carrito de compras .....</b>	<b>7</b>
<b>FEATURE 4: Cambio de moneda .....</b>	<b>9</b>
<b>FEATURE 5: Contraseña olvidada .....</b>	<b>10</b>
<b>FEATURE 6: Comparar productos .....</b>	<b>12</b>
<b>FEATURE 7: Gestión de devoluciones.....</b>	<b>13</b>
<b>FEATURE 8: Comprar como invitado .....</b>	<b>14</b>
<b>FEATURE 9: Buscar productos en OpenCart .....</b>	<b>15</b>
<b>FEATURE 10: Ordenar productos en la página de resultados .....</b>	<b>16</b>
<b>FEATURE 11: Gestión de la contraseña de la cuenta .....</b>	<b>18</b>
<b>FEATURE 12: Gestión de la información personal de la cuenta .....</b>	<b>21</b>
<b>FEATURE 13: Gestionar direcciones de envío .....</b>	<b>23</b>
<b>FEATURE 14: Valoración de productos .....</b>	<b>26</b>
<b>FEATURE 15: Guardar productos en la lista de deseos .....</b>	<b>29</b>
<b>FEATURES EXTRA .....</b>	<b>32</b>
<b>STEPS.....</b>	<b>33</b>

# FEATURE 1: Registro de usuario

Este feature permite comprobar que le es posible a los usuarios crear una cuenta proporcionando información personal, como nombre, correo electrónico y contraseña, para acceder a funciones exclusivas y personalizadas en la plataforma.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 5 escenarios:

## - **Scenario: Registro exitoso de un nuevo usuario**

Se empieza en la página principal. El programa actúa como usuario y lo primero que hace es ir a la página de registro y poner sus datos, además de aceptar la política de privacidad. Una vez se da al botón de continuar, se confirma que haya salido un mensaje que confirma que se ha realizado el registro con éxito.

```
Scenario: Registro exitoso de un nuevo usuario
Given the user is in the homepage
When the user clicks to go to the registration page
And the user fills in the registration form with the following details:


|            |                     |
|------------|---------------------|
| First Name | Javi                |
| Last Name  | David               |
| Email      | javi.david@test.com |
| Telephone  | 123456789           |
| Password   | Password123         |
| Confirm    | Password123         |


And the user agrees to the privacy policy
And the user clicks on the continue button
Then the user should see a confirmation message
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheRegistrationPage()
- void theUserFillsInTheRegistrationFormWithTheFollowingDetails (Map<String, String> userDetails)
- void TheUserAgreesToThePrivacyPolicy()
- void TheUserClicksOnTheContinueButton()
- void theUserShouldSeeAConfirmationMessage()

## - **Scenario: Registro fallido de un nuevo usuario (formulario vacío)**

Dado que una persona normal pondría los datos directamente, se ha realizado un escenario sobre qué pasaría si se intentara registrar sin ningún dato. Lo que se busca es activar todos los mensajes de aviso posibles.

```
Scenario: Registro fallido de un nuevo usuario (formulario vacío)
Given the user is in the homepage
When the user clicks to go to the registration page
And the user clicks on the continue button
Then the user should see a bunch of warning messages
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheRegistrationPage()
- void TheUserClicksOnTheContinueButton()
- void TheUserShouldSeeABunchOfWarningMessages()

- **Scenario: Registro fallido de un nuevo usuario (email)**

Se empieza en la página principal. El programa actúa como un usuario normal y corriente, pero se equivoca a la hora de escribir el correo electrónico. Lo que se busca es activar únicamente el mensaje de aviso del correo.

```
Scenario: Registro fallido de un nuevo usuario (email)
Given the user is in the homepage
When the user clicks to go to the registration page
And the user fills in the registration form with the following details:
    | First Name | 1631153 |
    | Last Name  | David  |
    | Email      | sukabliat.com |
    | Telephone  | 12345d6789 |
    | Password   | Password123 |
    | Confirm    | Password123 |
And the user clicks on the continue button
Then the user should see a email warning message
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheRegistrationPage()
- void theUserFillsInTheRegistrationFormWithTheFollowingDetails (Map<String, String> userDetails)
- void TheUserClicksOnTheContinueButton()
- void theUserShouldSeeAEmailWarningMessage()

- **Scenario: Registro fallido de un nuevo usuario (email repetido)**

Teniendo en cuenta que todo se guarda en una base de datos, el programa lo pone a prueba tratando de registrarse con el mismo correo que se ha registrado previamente. Se comprueba que el mensaje de aviso debería ser diferente a cuando se equivoca al escribirlo o directamente no hay uno.

```
Scenario: Registro fallido de un nuevo usuario (email repetido)
Given the user is in the homepage
When the user clicks to go to the registration page
And the user fills in the registration form with the following details:
    | First Name | 1631153 |
    | Last Name  | David  |
    | Email      | f@f.com |
    | Telephone  | 12345d6789 |
    | Password   | Password123 |
    | Confirm    | Password123 |
And the user agrees to the privacy policy
And the user clicks on the continue button
Then the user should see a used email warning message
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheRegistrationPage()
- void theUserFillsInTheRegistrationFormWithTheFollowingDetails (Map<String, String> userDetails)
- void TheUserAgreesToThePrivacyPolicy()
- void TheUserClicksOnTheContinueButton()
- void theUserShouldSeeAUsedEmailWarningMessage()

- **Scenario: Registro fallido de un nuevo usuario (nombre y teléfono)**

Teniendo en cuenta que el email puede introducirse incorrectamente lo intentamos también con otros parámetros. Por desgracia, Name y Telephone no contiene un diferenciador de caracteres y numero por lo que, al introducir caracteres incorrectos, falla.

```
Scenario: Registro fallido de un nuevo usuario (nombre y telefono)
Given the user is in the homepage
When the user clicks to go to the registration page
And the user fills in the registration form with the following details:
    | First Name | 1631153 |
    | Last Name  | David   |
    | Email      | f@f.com |
    | Telephone  | 12345d6789 |
    | Password   | Password123 |
    | Confirm    | Password123 |
And the user agrees to the privacy policy
And the user clicks on the continue button
Then the page should indicate that the provided parameters are invalid
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheRegistrationPage()
- void theUserFillsInTheRegistrationFormWithTheFollowingDetails (Map<String, String> userDetails)
- void TheUserAgreesToThePrivacyPolicy()
- void TheUserClicksOnTheContinueButton()
- void ThePageShouldIndicateThatTheProvidedParametersAreInvalid()

## FEATURE 2: Inicio de sesión

Este feature permite comprobar que es posible autenticarse ingresando las credenciales correctas, accediendo a su cuenta para gestionar pedidos, preferencias y otros servicios.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 3 escenarios:

### - **Scenario: Inicio de sesión en la pagina**

Se empieza en la página principal. El programa actúa como usuario y lo primero que hace es ir a la página de login y poner sus datos correctamente. Una vez se da al botón de Login, se confirma que la página Account esté activada.

```
Scenario: Inicio de sesión en la pagina
Given the user is in the homepage
When the user clicks to go to the login page
And the user fills email and password:
  | Email      | javi.david@test.com |
  | Password   | Password123         |
And the user clicks on the Login button
Then the user should see the account page
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheLoginPage()
- void theUserFillsEmailAndPassword(Map<String, String> userDetails)
- void TheUserClicksOnTheLoginButton()
- void TheUserShouldSeeTheAccountPage()

### - **Scenario: Inicio de sesión fallido en la página (valores incorrectos)**

Se empieza en la página principal. El programa actúa como usuario y lo primero que hace es ir a la página de login y poner sus datos incorrectamente. Se busca activar un mensaje de error.

```
Scenario: Inicio de sesión fallido en la pagina (valores incorrectos)
Given the user is in the homepage
When the user clicks to go to the login page
And the user fills email and password:
  | Email      | javi.david21312321312test.com |
  | Password   | sdfasfeawad                   |
And the user clicks on the Login button
Then the user should see a no matching message
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheLoginPage()
- void theUserFillsEmailAndPassword(Map<String, String> userDetails)
- void TheUserClicksOnTheLoginButton()
- void theUserShouldSeeANoMatchingMessage()

- **Scenario: Inicio de sesión fallido en la página (valores incorrectos)**

Se empieza en la página principal. El programa actúa como usuario y lo primero que hace es ir a la página de login e intentar iniciar sesión sin ningún dato. Se busca activar un mensaje de error.

```
Scenario: Inicio de sesión fallido en la pagina (formulario vacio)
Given the user is in the homepage
When the user clicks to go to the login page
And the user clicks on the Login button
Then the user should see a no matching message
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheLoginPage()
- void theUserFillsEmailAndPassword(Map<String, String> userDetails)
- void TheUserClicksOnTheLoginButton()
- void theUserShouldSeeANoMatchingMessage()

## FEATURE 3: Carrito de compras

Este feature permite comprobar la facilidad que tendrán los usuarios al agregar, eliminar y modificar productos que quieren comprar.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 5 escenarios:

- **Scenario: Añadir un producto al carrito**

Se empieza en la página principal. El programa actúa como usuario, inicia sesión y lo primero que hace es añadir al carrito uno de los productos disponibles en la tienda online.

```
Scenario: Añadir un producto al carrito
Given the user is in the homepage
And the user logs
When the user clicks on MacBook from Featured
And the user clicks Add to Cart button
Then the user should see the cart summary
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserIsOnTheLoginPage()
- void theUserClicksOnMacBookFromFeatured()
- void theUserClicksAddToCartButton()
- void theUserShouldSeeTheCartSummary()

- **Scenario: Comprar un producto**

Se empieza en la página principal y se vuelve a iniciar sesión. El programa continua la automatización y sigue, yendo al carrito de compras y aceptando los términos necesarios para avanzar y comprar. Al final, se debería de ver un mensaje de confirmación.

```
Scenario: Comprar un producto
Given the user is in the homepage
And the user logs
When the user is in the Shopping Cart page
And the user clicks on Checkout button
And the user clicks on the continue button
And the user agrees to the privacy policy
And the user clicks on another continue button
And the user clicks on the Confirm Order button
Then the user should see an order confirmation message
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserIsOnTheLoginPage()
- void TheUserIsInTheShoppingCartPage()
- void TheUserClicksOnCheckoutButton()
- void TheUserClicksOnTheContinueButton()
- void TheUserAgreesToThePrivacyPolicy()
- void TheUserClicksOnAnotherContinueButton()
- void TheUserClicksOnTheConfirmOrderButton()
- void theUserShouldSeeAnOrderConfirmationMessage()

- **Scenario: Confirmar compra**

Se empieza en la página principal y se vuelve a iniciar sesión. El programa finaliza confirmando que la compra se muestra en el historial de compra, observando la fecha de compra.

```
Scenario: Confirmar compra
Given the user is in the homepage
And the user logs
When the user is on My Account page
And the user clicks on Order History
Then the user should see the date of the last product to confirm
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserIsOnTheLoginPage()
- void TheUserIsOnMyAccountPage()
- void TheUserClicksOnOrderHistory()
- void TheUserShouldSeeTheDateOfTheLastProductToConfirm()

- **Scenario: Eliminar productos del carrito (uno solo)**

Se empieza en la página principal. El programa actúa como usuario y lo primero que hace es añadir al carrito uno de los productos disponibles en la tienda online. Luego, lo borra.

```
Scenario: Eliminar productos del carrito (uno solo)
Given the user is in the homepage
When the user have a product in the Shopping Cart
And the user removes one product
Then the product should disappear from the cart
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserHaveAProductInTheShoppingCart()
- void TheUserRemovesOneProduct()
- void TheProductShouldDisappearFromTheCart()

- **Scenario: Eliminar productos del carrito (todos)**

Se empieza en la página principal. El programa actúa como usuario y lo primero que hace es añadir al carrito varios productos disponibles en la tienda online. Luego, los borra todos.

```
Scenario: Eliminar productos del carrito (todos)
Given the user is in the homepage
When the user have multiple products in the Shopping Cart
And the user removes one by one
Then the product should disappear from the cart
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserHaveMultipleProductsInTheShoppingCart()
- void TheUserRemovesOneByOne()
- void TheProductShouldDisappearFromTheCart()



## FEATURE 4: Cambio de moneda

Este feature permite comprobar el funcionamiento de la opción de ver los precios en diferentes monedas según su ubicación o preferencia, mejorando la experiencia de compra internacional.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado un escenario para cada moneda disponible:

- **Scenario: Cambiar la moneda a euros**

Se empieza en la página principal, donde está la opción para cambiar la moneda de compra. En este caso a EUROS y se comprueba que todos los precios hayan cambiado.

```
Scenario: Cambiar la moneda a euros
Given the user is in the homepage
When the user clicks on the currency dropdown to select "EUR"
Then the prices on the homepage should be displayed all in "€"
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void theUserClicksOnTheCurrencyDropdownToSelectCurrency(String currency)
- void thePricesOnTheHomepageShouldBeDisplayedAllInCurrency(String currencySymbol)

- **Scenario: Cambiar la moneda a libras**

Se empieza en la página principal, donde está la opción para cambiar la moneda de compra. En este caso a POUNDING STERLING y se comprueba que todos los precios hayan cambiado.

```
Scenario: Cambiar la moneda a libras
Given the user is in the homepage
When the user clicks on the currency dropdown to select "GBP"
Then the prices on the homepage should be displayed all in "£"
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void theUserClicksOnTheCurrencyDropdownToSelectCurrency(String currency)
- void thePricesOnTheHomepageShouldBeDisplayedAllInCurrency(String currencySymbol)

- **Scenario: Cambiar la moneda a dolares**

Se empieza en la página principal, donde está la opción para cambiar la moneda de compra. En este caso a DOLLARS y se comprueba que todos los precios hayan cambiado.

```
Scenario: Cambiar la moneda a dolares
Given the user is in the homepage
When the user clicks on the currency dropdown to select "USD"
Then the prices on the homepage should be displayed all in "$"
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void theUserClicksOnTheCurrencyDropdownToSelectCurrency(String currency)
- void thePricesOnTheHomepageShouldBeDisplayedAllInCurrency(String currencySymbol)

## FEATURE 5: Contraseña olvidada

Este feature permite comprobar que los usuarios tengan un proceso seguro para recuperar o restablecer su contraseña en caso de que la hayan olvidado, asegurando el acceso a su cuenta.

Usando un correo temporal se ha comprobado que la página no cuenta realmente con un funcionamiento completo para este apartado, por lo que se centrará únicamente en los mensajes de confirmación o error.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 3 escenarios:

- **Scenario: Contraseña olvidada en correo valido**

Se empieza en la página principal. El programa actúa como un usuario despistado que ha olvidado su contraseña y lo primero que hace es ir a la página de login y darle a la opción de 'Forgotten Password' y poner su correo. Al darle a continuar debería salir un mensaje de que se ha enviado un correo para recuperar la contraseña.

```
Scenario: Contraseña olvidada en correo valido
Given the user is in the homepage
When the user clicks to go to the login page
And the user clicks on forgotten password
And the user fills email:
  | Email          | javi.david@test.com |
And the user clicks on the continue button
Then the user should see a confirmation message on another page
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheLoginPage()
- void TheUserClicksOnForgottenPassword()
- void theUserFillsEmail(Map<String, String> userDetails)
- void TheUserClicksOnTheContinueButton()
- void TheUserShouldSeeAConfirmationMessageOnAnotherPage()

- **Scenario: Contraseña olvidada sin correo**

Se empieza en la página principal. El programa actúa como un usuario despistado que ha olvidado su contraseña y lo primero que hace es ir a la página de login y darle a la opción de 'Forgotten Password', pero no pone su correo. Al darle a continuar debería salir un mensaje de error.

```
Scenario: Contraseña olvidada sin correo
Given the user is in the homepage
When the user clicks to go to the login page
And the user clicks on forgotten password
And the user clicks on the continue button
Then the user should see a warning message on the same page
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheLoginPage()
- void TheUserClicksOnForgottenPassword()
- void TheUserClicksOnTheContinueButton()
- void TheUserShouldSeeAWarningMessageOnTheSamePage()

- **Scenario: Contraseña olvidada en correo invalido**

Se empieza en la página principal. El programa actúa como un usuario despistado que ha olvidado su contraseña y lo primero que hace es ir a la página de login y darle a la opción de 'Forgotten Password', pero pone su correo mal. Al darle a continuar debería salir un mensaje de error.

```
Scenario: Contraseña olvidada en correo invalido
Given the user is in the homepage
When the user clicks to go to the login page
And the user clicks on forgotten password
And the user fills email:
    | Email          | sdfghjfds |
And the user clicks on the continue button
Then the user should see a warning message on the same page
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserClicksToGoToTheLoginPage()
- void TheUserClicksOnForgottenPassword()
- void theUserFillsEmail(Map<String, String> userDetails)
- void TheUserClicksOnTheContinueButton()
- void TheUserShouldSeeAWarningMessageOnTheSamePage()

## FEATURE 6: Comparar productos

Este feature permite comprobar que los usuarios puedan comparar las características de varios productos para tomar decisiones de compra más informadas. En el código se toman en cuenta las siguientes características: Precio, Memoria y Procesador.

Se ha realizado un escenario que analizar:

- **Scenario: Comparar dos productos**

Se empieza en la página principal. El programa busca dos productos similares para comparar y se escoge el mejor de estos dos para poner en el carrito.

```
Scenario: Comparar dos productos
Given the user is in the homepage
When the user searches for Laptop & Notebooks
And the user clicks compare for HP
And the user clicks compare for MacBook
And the user navigates to the product comparison page
And Add To Cart the better one
Then the best laptop is in the shopping cart
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserSearchesForLaptopNotebooks()
- void TheUserClicksCompareForHP()
- void TheUserClicksCompareForMacBook()
- void TheUserNavigatesToTheProductComparisonPage()
- void AddToCartthebetterone()
- void TheBestLaptopIsInTheShoppingCart()

*Algo que me gustaría comentar es que se intentó poner un escenario más donde se comparaban tres productos, pero la página no contiene características definidas en cada uno de sus productos, por lo que, no serviría.*

## FEATURE 7: Gestión de devoluciones

Este feature permite comprobar el sistema para que los usuarios soliciten devoluciones de los productos, gestionen el estado de sus solicitudes y obtengan soporte relacionado.

Se ha realizado un escenario que analizar:

- **Scenario: Devolver un MacBook que acaba de comprar**

Se empieza en la página principal. Para corroborar el funcionamiento de las devoluciones se compra un producto, llamando a las funciones previas del Feature 3 en una función. Aquí, se buscará el ultimo pedido realizado y lo devolverá.

```
Scenario: Devolver un MacBook que acaba de comprar
Given the user is in the homepage
When the user buys a product
And the user goes to Order History in My Account
And the user opens the details of the last order
And the user clicks on the return button
And the user fill the obligatory options to return the product
Then the user should see in the Return page the product
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserBuysAProduct()
- void TheUserGoesToOrderHistoryInMyAccount()
- void TheUserOpensTheDetailsOfTheLastOrder()
- void TheUserClicksOnTheReturnButton()
- void TheUserFillTheObligatoryOptionsToReturnTheProduct()
- void TheUserShouldSeeInTheReturnPageTheProduct()

## FEATURE 8: Comprar como invitado

Este feature permite comprobar la posibilidad de hacer compras sin necesidad de crear una cuenta, ofreciendo un proceso de compra rápido y sencillo para los usuarios.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 2 escenarios:

- **Scenario: Realizar una compra como invitado**

Se empieza en la página principal. El programa actúa como usuario y lo primero que hace es añadir al carrito uno de los productos disponibles en la tienda online. Después compra el producto desde el carrito de compra sin registrarse, aceptando los términos necesarios para avanzar. Al final, se debería de ver un mensaje de confirmación.

```
Scenario: Realizar una compra como invitado
Given the user is in the homepage
When the user add products to the Shopping Cart
And the user buys the product with the Guess option
Then the user should be able to enter the shipping details without an account
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserAddProductsToTheShoppingCart()
- void TheUserBuysTheProductWithTheGuessOption()
- void TheUserShouldBeAbleToEnterTheShippingDetailsWithoutAnAccount()

- **Scenario: Intentar realizar una compra sin proporcionar datos de envío**

Se empieza en la página principal. El programa actúa como usuario y lo primero que hace es añadir al carrito uno de los productos disponibles en la tienda online. Después intenta comprar el producto desde el carrito de compra sin registrarse, olvidando ingresar una dirección de envío. Se debería de ver un mensaje de error.

```
Scenario: Intentar realizar una compra sin proporcionar datos de envío
Given the user is in the homepage
When the user add products to the Shopping Cart
And the user tries to proceed without a shipping details
Then warning messages about the shipping details should appear
```

Steps usados:

- void TheUserIsInTheIndexPage()
- void TheUserAddProductsToTheShoppingCart()
- void TheUserTriesToProceedWithoutAShippingDetails()
- void WarningMessageAboutTheShippingDetailsShouldAppear()

# FEATURE 9: Buscar productos en OpenCart

Este feature permite comprobar la posibilidad de que el usuario pueda encontrar fácilmente productos en la tienda. Así, como usuario de OpenCart, quiero poder buscar productos por su nombre en la barra de búsqueda.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 2 escenarios:

## - **Scenario: Realizar una compra como invitado**

Se empieza en la página principal. El programa actúa como usuario y lo primero que hace es buscar 'iPhone' en el buscador para que aparezcan los resultados relacionados con el producto de Apple.

```
Scenario: Buscar un producto existente
Given El usuario está en la página de inicio para buscar
When Introduce "iPhone" en la barra de búsqueda
And Hace clic en el botón de buscar
Then Aparecen resultados relevantes relacionados con "iPhone"
```

Steps usados:

- void usuarioEnPaginaInicio()
- void introduceEnBarraBusqueda(String producto)
- void clicEnBotonBuscar()
- void aparecenResultadosRelevantes(String producto) + boolean  
verificarElementoConTexto(String selector, String textoEsperado)

## - **Scenario: Realizar una compra como invitado**

Se empieza en la página principal. El programa actúa como usuario, pero no tiene idea de lo que buscar por lo que introduzca en el buscador es erróneo y no debería aparecer ningún producto.

```
Scenario: Buscar un producto inexistente
Given El usuario está en la página de inicio para buscar
When Introduce "XYZ123" en la barra de búsqueda
And Hace clic en el botón de buscar
Then No aparecen resultados y se muestra un mensaje de "Your shopping cart is empty!"
```

Steps usados:

- void usuarioEnPaginaInicio()
- void introduceEnBarraBusqueda(String producto)
- void clicEnBotonBuscar()
- void noAparecenResultados(String mensaje) + boolean  
verificarElementoConTexto(String selector, String textoEsperado)

## FEATURE 10: Ordenar productos en la página de resultados

Este feature permite comprobar la posibilidad de que el usuario pueda encontrar productos fácilmente. Así, como usuario de la tienda online, quiero ordenar los productos según diferentes criterios.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 6 escenarios:

- **Scenario: Ordenar productos por precio de menor a mayor**

Se empieza en la página de Escritorios (Desktops). El programa actúa como usuario y selecciona la opción para ordenar los productos de menor a mayor precio.

```
Scenario: Ordenar productos por precio de menor a mayor
Given El usuario está en la categoría de Desktops
When El usuario selecciona "Price (Low > High)" en el menú de ordenación
Then Los productos deberían mostrarse ordenados por precio en orden ascendente
```

Steps usados:

- void elUsuarioEstaEnLaCategoríaDeDesktops()
- void elUsuarioSeleccionaEnElMenuDeOrdenacion(String criterio)
- void losProductosDeberianMostrarseOrdenadosPorPrecioAscendente() + void verificarOrdenDePrecios(boolean ascendente)

- **Scenario: Ordenar productos por precio de mayor a menor**

Se empieza en la página de Escritorios (Desktops). El programa actúa como usuario y selecciona la opción para ordenar los productos de mayor a menor precio.

```
Scenario: Ordenar productos por precio de mayor a menor
Given El usuario está en la categoría de Desktops
When El usuario selecciona "Price (High > Low)" en el menú de ordenación
Then Los productos deberían mostrarse ordenados por precio en orden descendente
```

Steps usados:

- void elUsuarioEstaEnLaCategoríaDeDesktops()
- void elUsuarioSeleccionaEnElMenuDeOrdenacion(String criterio)
- void losProductosDeberianMostrarseOrdenadosPorPrecioDescendente() + void verificarOrdenDePrecios(boolean ascendente)

- **Scenario: Ordenar productos por nombre de la A a la Z**

Se empieza en la página de Escritorios (Desktops). El programa actúa como usuario y selecciona la opción para ordenar los productos en orden alfabético.

```
Scenario: Ordenar productos por nombre de la A a la Z
Given El usuario está en la categoría de Desktops
When El usuario selecciona "Name (A - Z)" en el menú de ordenación
Then Los productos deberían mostrarse ordenados alfabéticamente de la A a la Z
```

Steps usados:

- void elUsuarioEstaEnLaCategoríaDeDesktops()
- void elUsuarioSeleccionaEnElMenuDeOrdenacion(String criterio)
- void losProductosDeberianMostrarseOrdenadosAlfabeticamenteAZ() + void verificarOrdenAlfabetico(boolean ascendente)



- **Scenario: Ordenar productos por nombre de la Z a la A**

Se empieza en la página de Escritorios (Desktops). El programa actúa como usuario y selecciona la opción para ordenar los productos en orden alfabético inverso.

```
Scenario: Ordenar productos por nombre de la Z a la A
Given El usuario está en la categoría de Desktops
When El usuario selecciona "Name (Z - A)" en el menú de ordenación
Then Los productos deberían mostrarse ordenados alfabéticamente de la Z a la A
```

Steps usados:

- void elUsuarioEstaEnLaCategoríaDeDesktops()
- void elUsuarioSeleccionaEnElMenuDeOrdenacion(String criterio)
- void losProductosDeberianMostrarseOrdenadosAlfabeticamenteZA() + void verificarOrdenAlfabetico(boolean ascendente)

- **Scenario: Ordenar productos por calificación decedentes**

Se empieza en la página de Escritorios (Desktops). El programa actúa como usuario y selecciona la opción para ordenar los productos en base a las calificaciones de otros usuarios del más valorado al menos.

```
Scenario: Ordenar productos por calificación decedentes
Given El usuario está en la categoría de Desktops
When El usuario selecciona "Rating (Highest)" en el menú de ordenación
Then Los productos deberían mostrarse con las calificaciones más altas primero
```

Steps usados:

- void elUsuarioEstaEnLaCategoríaDeDesktops()
- void elUsuarioSeleccionaEnElMenuDeOrdenacion(String criterio)
- void losProductosDeberianMostrarseConLasCalificacionesMasAltasPrimero() + void verificarOrdenDeCalificaciones(boolean descendente)

- **Scenario: Ordenar productos por calificación acedentes**

Se empieza en la página de Escritorios (Desktops). El programa actúa como usuario y selecciona la opción para ordenar los productos en base a las calificaciones de otros usuarios del menos valorado al más.

```
Scenario: Ordenar productos por calificación acedentes
Given El usuario está en la categoría de Desktops
When El usuario selecciona "Rating (Lowest)" en el menú de ordenación
Then Los productos deberían mostrarse con las calificaciones más bajas primero
```

Steps usados:

- void elUsuarioEstaEnLaCategoríaDeDesktops()
- void elUsuarioSeleccionaEnElMenuDeOrdenacion(String criterio)
- void losProductosDeberianMostrarseConLasCalificacionesMasBajasPrimero() + void verificarOrdenDeCalificaciones(boolean descendente)

# FEATURE 11: Gestión de la contraseña de la cuenta

Este feature permite comprobar la posibilidad de que el usuario pueda mantener la seguridad de la cuenta. Así, como usuario registrado en la tienda online, quiero cambiar mi contraseña para proteger mi cuenta.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 6 escenarios:

## - **Escenario: Cambiar la contraseña**

Se empieza con el programa iniciando sesión como usuario y cambiando su contraseña por una completamente diferente.

```
# Escenario para cambiar la contraseña
Scenario: Cambiar la contraseña
  Given El usuario está logeado en su cuenta
    | input-email | javi.david@test.com |
    | input-password | Password123 |
  When El usuario hace clic en el enlace "Password"
  And El usuario introduce una nueva contraseña:
    | input-password | NuevaPass45 |
  And El usuario introduce la confirmación de la nueva contraseña:
    | input-confirm | NuevaPass45 |
  And El usuario hace clic en el botón de continuar para guardar la nueva contraseña
  Then La contraseña del usuario debería cambiarse con éxito
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_introduce\_una\_nueva\_contraseña(Map<String, String> newPasswordDetails) + void fillForm(Map<String, String> fields)
- void el\_usuario\_introduce\_la\_confirmacion\_de\_la\_nueva\_contraseña(Map<String, String> confirmPasswordDetails) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el boton\_de\_continuar\_para\_guardar\_la\_nueva\_contraseña() + void clickButton(By locator)
- void la\_contraseña\_del\_usuario\_debería\_cambiarse\_con\_éxito()

## - **Escenario: Restaurar la contraseña a su valor original**

Se comprueba que se pueda iniciar sesión con la nueva contraseña. Y una vez es así, se restaura la contraseña anterior, para futuros features y no alterar el código.

```
# Escenario para restaurar la contraseña a su valor original
Scenario: Restaurar la contraseña a su valor original
  Given El usuario está logeado en su cuenta
    | input-email | javi.david@test.com |
    | input-password | NuevaPass45 |
  When El usuario hace clic en el enlace "Password"
  And El usuario introduce una nueva contraseña:
    | input-password | Password123 |
  And El usuario introduce la confirmación de la nueva contraseña:
    | input-confirm | Password123 |
  And El usuario hace clic en el botón de continuar para guardar la nueva contraseña
  Then La contraseña del usuario debería cambiarse con éxito
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_introduce\_una\_nueva\_contraseña(Map<String, String> newPasswordDetails) + void fillForm(Map<String, String> fields)
- void el\_usuario\_introduce\_la\_confirmacion\_de\_la\_nueva\_contraseña(Map<String, String> confirmPasswordDetails) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el boton\_de\_continuar\_para\_guardar\_la\_nueva\_contraseña() + void clickButton(By locator)
- void la\_contraseña\_del\_usuario\_debería\_cambiarse\_con\_éxito()

- **Scenario: Intentar cambiar la contraseña sin confirmar la nueva contraseña**

Se empieza con el programa iniciando sesión como usuario e intentando cambiar su contraseña, sin escribir ninguna nueva.

```
# Escenario para intentar cambiar la contraseña sin escribir ninguna contraseña
Scenario: Intentar cambiar la contraseña sin confirmar la nueva contraseña
  Given El usuario está logeado en su cuenta
    | input-email | javi.david@test.com |
    | input-password | Password123 |
  When El usuario hace clic en el enlace "Password"
  And El usuario hace clic en el botón de continuar para guardar la nueva contraseña
  Then El sistema debería mostrar un mensaje de error en la contraseña
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_hace\_clic\_en\_el\_boton\_de\_continuar\_para\_guardar\_la\_nueva\_contraseña() + void clickButton(By locator)
- void el\_sistema\_debería\_mostrar\_un\_mensaje\_de\_error\_en\_la\_contraseña()

- **Scenario: Intentar cambiar la contraseña sin confirmar la nueva contraseña**

Se empieza con el programa iniciando sesión como usuario y cambiando su contraseña por una completamente diferente, pero se olvida de escribir la contraseña una segunda vez.

```
# Escenario para intentar cambiar la contraseña sin confirmar la nueva contraseña
Scenario: Intentar cambiar la contraseña sin confirmar la nueva contraseña
  Given El usuario está logeado en su cuenta
    | input-email | javi.david@test.com |
    | input-password | Password123 |
  When El usuario hace clic en el enlace "Password"
  And El usuario introduce una nueva contraseña:
    | input-password | NuevaPass45 |
  And El usuario hace clic en el botón de continuar para guardar la nueva contraseña
  Then El sistema debería mostrar un mensaje de error indicando que las contraseñas no coinciden
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_introduce\_una\_nueva\_contraseña(Map<String, String> newPasswordDetails) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_boton\_de\_continuar\_para\_guardar\_la\_nueva\_contraseña() + void clickButton(By locator)
- void el\_sistema\_debería\_mostrar\_un\_mensaje\_de\_error\_indicando\_que\_las\_contraseñas\_no\_coinciden()

- **Scenario: Intentar cambiar la contraseña con un carácter sin confirmar la nueva contraseña**

Se empieza con el programa iniciando sesión como usuario y cambiando su contraseña por una completamente diferente y muy pobre, pero se olvida de escribir la contraseña una segunda vez.

```
# Escenario para intentar cambiar la contraseña escribiendo solo un caracter sin confirmar la nueva contraseña
Scenario: Intentar cambiar la contraseña con un caracter sin confirmar la nueva contraseña
    Given El usuario está logeado en su cuenta
        | input-email | javi.david@test.com |
        | input-password | Password123 |
    When El usuario hace clic en el enlace "Password"
    And El usuario introduce una nueva contraseña:
        | input-password | a |
    And El usuario hace clic en el botón de continuar para guardar la nueva contraseña
    Then El sistema debería mostrar un mensaje de error en la contraseña
    And El sistema debería mostrar un mensaje de error indicando que las contraseñas no coinciden
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_introduce\_una\_nueva\_contraseña(Map<String, String> newPasswordDetails) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_boton\_de\_continuar\_para\_guardar\_la\_nueva\_contraseña() + void clickButton(By locator)
- void el\_sistema\_debería\_mostrar\_un\_mensaje\_de\_error\_en\_la\_contraseña()
- void el\_sistema\_debería\_mostrar\_un\_mensaje\_de\_error\_indicando\_que\_las\_contraseñas\_no\_coinciden()

- **Scenario: Intentar cambiar la contraseña con contraseñas que no coinciden**

Se empieza con el programa iniciando sesión como usuario y cambiando su contraseña por una completamente diferente, pero se equivoca al escribir la confirmación de la nueva contraseña.

```
# Escenario para intentar cambiar la contraseña con contraseñas que no coinciden
Scenario: Intentar cambiar la contraseña con contraseñas que no coinciden
    Given El usuario está logeado en su cuenta
        | input-email | javi.david@test.com |
        | input-password | Password123 |
    When El usuario hace clic en el enlace "Password"
    And El usuario introduce una nueva contraseña:
        | input-password | NuevaPass45 |
    And El usuario introduce la confirmación de la nueva contraseña:
        | input-confirm | NuevaPass46 |
    And El usuario hace clic en el botón de continuar para guardar la nueva contraseña
    Then El sistema debería mostrar un mensaje de error indicando que las contraseñas no coinciden
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_introduce\_una\_nueva\_contraseña(Map<String, String> newPasswordDetails) + void fillForm(Map<String, String> fields)
- void el\_usuario\_introduce\_la\_confirmacion\_de\_la\_nueva\_contraseña(Map<String, String> confirmPasswordDetails) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_boton\_de\_continuar\_para\_guardar\_la\_nueva\_contraseña() + void clickButton(By locator)
- void el\_sistema\_debería\_mostrar\_un\_mensaje\_de\_error\_indicando\_que\_las\_contraseñas\_no\_coinciden()

## FEATURE 12: Gestión de la información personal de la cuenta

Este feature permite comprobar la seguridad y persistencia de los datos. Así, como usuario registrado en la tienda online, quiero gestionar los detalles de mi cuenta, pero no puedo eliminarla.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 4 escenarios:

### - **Escenario: Actualizar la información personal**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de edición de la cuenta para cambiar todos los campos por unos diferentes.

```
# Escenario para actualizar la información personal
Scenario: Actualizar la información personal
  Given El usuario está logeado en su cuenta
    | input-email      | javi.david@test.com |
    | input-password   | Password123         |
  When El usuario hace clic en el enlace "Edit Account"
  And El usuario actualiza su información personal con
    | input-firstname | Javier      |
    | input-lastname  | González   |
    | input-email     | david.javi@test.com |
    | input-telephone | 987654321  |
  Then El mensaje de éxito debería aparecer
  And El usuario hace clic en el enlace "Edit Account"
  And La información personal del usuario debería actualizarse correctamente
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_actualiza\_su\_informacion\_personal(Map<String, String> userDetails)
- void el\_mensaje\_de\_exito\_deberia\_aparecer() + void verifySuccessMessage(String message)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void la\_informacion\_personal\_del\_usuario\_deberia\_actualizarse\_correctamente() + void verifyUpdatedPersonalInfo(Map<String, String> expectedValues)

### - **Escenario: Restaurar la información personal a los valores originales**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de edición para restaurar los valores por defecto de la cuenta.

```
# Escenario para restaurar la información personal a los valores originales
Scenario: Restaurar la información personal a los valores originales
  Given El usuario está logeado en su cuenta
    | input-email      | david.javi@test.com |
    | input-password   | Password123         |
  When El usuario hace clic en el enlace "Edit Account"
  And El usuario actualiza su información personal con
    | input-firstname | Javi      |
    | input-lastname  | David     |
    | input-email     | javi.david@test.com |
    | input-telephone | 123456789 |
  Then El mensaje de éxito debería aparecer
  And El usuario hace clic en el enlace "Edit Account"
  And La información personal del usuario debería restaurarse correctamente
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_actualiza\_su\_informacion\_personal(Map<String, String> userDetails)
- void el\_mensaje\_de\_exito\_deberia\_aparecer() + void verifySuccessMessage(String message)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void la\_informacion\_personal\_del\_usuario\_deberia\_restaurarse\_correctamente() + void verifyUpdatedPersonalInfo(Map<String, String> expectedValues)

- **Scenario: Actualizar la información personal con datos inválidos**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de edición, pero tratando de cambiar los campos estando vacíos.

```
# Escenario para actualizar la información personal con datos inválidos
Scenario: Actualizar la información personal con datos inválidos
  Given El usuario está logeado en su cuenta
    | input-email      | javi.david@test.com |
    | input-password   | Password123         |
  When El usuario hace clic en el enlace "Edit Account"
  And El usuario deja los campos vacíos y hace clic en "Continue"
  Then El sistema debería mostrar los mensajes de error correspondientes
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_deja\_los\_campos\_vacios\_y\_hace\_clic\_en\_continue()
- void el\_sistema\_deberia\_mostrar\_mensajes\_de\_error() + void verifyErrorMessages(Map<String, String> errorMessages)

- **Scenario: Actualizar la información personal con datos inválidos**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de edición, pero tratando de cambiar los campos por unos completamente erróneos.

```
# Escenario para actualizar la información personal excediendo los límites de caracteres
Scenario: Actualizar la información personal con datos inválidos
  Given El usuario está logeado en su cuenta
    | input-email      | javi.david@test.com |
    | input-password   | Password123         |
  When El usuario hace clic en el enlace "Edit Account"
  And El usuario actualiza su información personal con
    | input-firstname | JaviJaviJaviJaviJaviJaviJaviJaviJaviJavi |
    | input-lastname  | DavidDavidDavidDavidDavidDavidDavidDavid |
    | input-email     | d@t                                           |
    | input-telephone | 123456789123456789123456789123456         |
  Then El sistema debería mostrar los mensajes de error correspondientes
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_actualiza\_su\_informacion\_personal(Map<String, String> userDetails)
- void el\_sistema\_deberia\_mostrar\_mensajes\_de\_error() + void verifyErrorMessages(Map<String, String> errorMessages)

## FEATURE 13: Gestionar direcciones de envío

Este feature permite comprobar que se pueda gestionar la dirección correcta de envío para recibir pedidos. Así, como usuario de la tienda, quiero poder agregar, editar y eliminar direcciones de envío de mi cuenta.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 7 escenarios:

### - **Scenario: Intentar crear una nueva dirección sin rellenar**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de dirección de envío, pero trata de continuar sin haber rellenado ningún campo, por lo que los errores deben saltar.

```
# Escenario para intentar crear una nueva dirección sin rellenar ningún campo
Scenario: Intentar crear una nueva dirección sin rellenar ningún campo
Given El usuario está logeado en su cuenta
| input-email | javi.david@test.com |
| input-password | Password123 |
When El usuario hace clic en el enlace "Address Book"
And El usuario intenta agregar una nueva dirección sin completar ningún campo
Then El sistema debería mostrar un error indicando que todos los campos son obligatorios
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void tryAddingEmptyAddress() + void clickButton(By locator)
- void verifyAllFieldsError() + void verifyErrorMessages(Map<String, String> errorMessages)

### - **Scenario: Añadir una nueva dirección de envío**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de dirección de envío, rellenando los campos necesarios para añadir una dirección.

```
# Escenario para añadir una nueva dirección de envío
Scenario: Añadir una nueva dirección de envío
Given El usuario está logeado en su cuenta
| input-email | javi.david@test.com |
| input-password | Password123 |
When El usuario hace clic en el enlace "Address Book"
And El usuario agrega una nueva dirección con:
| input-firstname | Javier |
| input-lastname | David |
| input-company | MiEmpresa |
| input-address-1 | Calle Falsa 123 |
| input-address-2 | Piso 3, Puerta A |
| input-city | Madrid |
| input-postcode | 28000 |
| input-country | Spain |
| input-zone | Madrid |
Then La dirección debería ser guardada y aparecer en la lista de direcciones del usuario
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void addNewAddress(Map<String, String> addressDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void verifyAddressSaved() + verifyAddressInList(String[] expectedAddress)

## - **Scenario: Añadir una nueva dirección de envío**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de dirección de envío para borrar los campos de información, por lo que los errores deben saltar.

```
# Escenario para intentar editar una dirección borrando la información de todos los campos
Scenario: Intentar editar una dirección borrando la información de todos los campos
Given El usuario está logeado en su cuenta
| input-email | javi.david@test.com |
| input-password | Password123 |
When El usuario hace clic en el enlace "Address Book"
And El usuario intenta editar una dirección borrando la información de todos los campos
Then El sistema debería mostrar un error indicando que todos los campos son obligatorios
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void el\_usuario\_intenta\_editar\_borrando\_campos() + void performActionOnAddress(String[] expectedAddress, By actionButtonLocator)
- void verifyAllFieldsError() + void verifyErrorMessages(Map<String, String> errorMessages)

## - **Scenario: Editar una dirección existente excediendo el límite de caracteres de los campos**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de dirección de envío para rellenar los campos necesarios para añadir una dirección, pero poniendo muchos caracteres para llegar a su límite.

```
# Escenario para editar una dirección existente excediendo el límite de caracteres de los campos
Scenario: Editar una dirección existente excediendo el límite de caracteres de los campos
Given El usuario está logeado en su cuenta
| input-email | javi.david@test.com |
| input-password | Password123 |
When El usuario hace clic en el enlace "Address Book"
And El usuario edita la dirección a:
| input-firstname | aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa |
| input-lastname | aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa |
| input-company | aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa |
| input-address-1 | aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa |
| input-address-2 | aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa |
| input-city | aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa |
| input-postcode | 01234567890 |
| input-country | --- Please Select --- |
| input-zone | --- Please Select --- |
Then El sistema debería mostrar un error indicando que todos los campos son obligatorios
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void editAddress(Map<String, String> updatedAddress) + void performActionOnAddress(String[] expectedAddress, By actionButtonLocator) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void verifyAllFieldsError() + void verifyErrorMessages(Map<String, String> errorMessages)



#### - **Scenario: Editar una dirección existente**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de dirección de envío, editando los campos necesarios para añadir una dirección.

```
# Escenario para editar una dirección existente
Scenario: Editar una dirección existente
  Given El usuario está logeado en su cuenta
    | input-email | javi.david@test.com |
    | input-password | Password123 |
  When El usuario hace clic en el enlace "Address Book"
  And El usuario edita la dirección a:
    | input-firstname | David |
    | input-lastname | Javier |
    | input-company | MiNuevaEmpresa |
    | input-address-1 | Calle Verdadera 456 |
    | input-address-2 | Piso 4, Puerta B |
    | input-city | Barcelona |
    | input-postcode | 28001 |
    | input-country | Spain |
    | input-zone | Barcelona |
  Then La dirección debería ser actualizada correctamente en la cuenta del usuario
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void editAddress(Map<String, String> updatedAddress) + void performActionOnAddress(String[] expectedAddress, By actionButtonLocator) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void verifyAddressUpdated() + verifyAddressInList(String[] expectedAddress)

#### - **Scenario: Eliminar una dirección de envío**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de dirección de envío, eliminando la dirección.

```
# Escenario para eliminar una dirección de envío
Scenario: Eliminar una dirección de envío
  Given El usuario está logeado en su cuenta
    | input-email | javi.david@test.com |
    | input-password | Password123 |
  When El usuario hace clic en el enlace "Address Book"
  And El usuario elimina la dirección
  Then La dirección debería ser eliminada y ya no aparecer en la lista de direcciones
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void deleteAddress() + void performActionOnAddress(String[] expectedAddress, By actionButtonLocator)
- void verifyAddressDeleted() + verifyAddressInList(String[] expectedAddress)

#### - **Scenario: Eliminar una dirección de envío**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de dirección de envío, tratando de eliminar la dirección sin éxito.

```
# Escenario para intentar borrar una dirección cuando solo existe una
Scenario: Intentar borrar una dirección cuando solo existe una
  Given El usuario está logeado en su cuenta
    | input-email | javi.david@test.com |
    | input-password | Password123 |
  When El usuario hace clic en el enlace "Address Book"
  And El usuario intenta eliminar la única dirección que tiene
  Then El sistema debería mostrar un error indicando que no se puede eliminar la última dirección
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) + void clickButton(By locator)
- void tryDeletingDefaultAddress() + void clickButton(By locator)
- void verifyCannotDeleteLastAddress() + verifyAddressInList(String[] expectedAddress)

## FEATURE 14: Valoración de productos

Este feature permite comprobar si se puede compartir la experiencia con otros usuarios. Es decir, que, como cliente de la tienda, quiero valorar los productos en su página correspondiente.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 6 escenarios:

### - **Scenario: Valorar un producto con estrellas**

Se empieza en la página del producto a valorar. Se pone un nombre, comentario y estrellas para dar tu opinión. Un mensaje de éxito debería aparecer al completarlo.

```
Scenario: Valorar un producto con estrellas
Given El usuario está en la página del producto que desea valorar
When El usuario accede a la sección de valoraciones
And El usuario escribe su nombre como "Juan Pérez"
And El usuario escribe un comentario "Buen producto, cumple con lo prometido."
And El usuario selecciona una calificación de 4 estrellas
And El usuario hace clic en el botón "Continue"
Then Debería aparecer un mensaje de éxito que dice "Thank you for your review. It has been submitted to the webmaster for approval."
```

Steps usados:

- void el\_usuario\_esta\_en\_la\_pagina\_del\_producto\_que\_desea\_valorar() + void clickButton(By locator)
- void el\_usuario\_accede\_a\_la\_seccion\_de\_valoraciones()
- void el\_usuario\_escribe\_su\_nombre\_como(String nombre)
- void el\_usuario\_escribe\_un\_comentario(String comentario)
- void el\_usuario\_selecciona\_una\_calificacion\_de\_estrellas(int estrellas) + void clickButton(By locator)
- void el\_usuario\_hace\_clic\_en\_el\_boton(String boton) + void clickButton(By locator)
- void deberia\_aparecer\_un\_mensaje\_de\_exito\_que\_dice(String mensajeExito) + void verifySuccessMessage(String message)

### - **Scenario: Intentar enviar una valoración sin escribir un comentario**

Se empieza en la página del producto a valorar. Se pone un nombre y estrellas para dar tu opinión, pero falta un comentario. Un mensaje de aviso debería aparecer.

```
Scenario: Intentar enviar una valoración sin escribir un comentario
Given El usuario está en la página del producto que desea valorar
When El usuario accede a la sección de valoraciones
And El usuario escribe su nombre como "Juan Pérez"
And El usuario selecciona una calificación de 4 estrellas
And El usuario hace clic en el botón "Continue"
Then Debería aparecer un mensaje de error "Warning: Review Text must be between 25 and 1000 characters!"
```

Steps usados:

- void el\_usuario\_esta\_en\_la\_pagina\_del\_producto\_que\_desea\_valorar() + void clickButton(By locator)
- void el\_usuario\_accede\_a\_la\_seccion\_de\_valoraciones()
- void el\_usuario\_escribe\_su\_nombre\_como(String nombre)
- void el\_usuario\_selecciona\_una\_calificacion\_de\_estrellas(int estrellas) + void clickButton(By locator)
- void el\_usuario\_hace\_clic\_en\_el\_boton(String boton) + void clickButton(By locator)
- void deberia\_aparecer\_un\_mensaje\_de\_error(String mensajeError)

- **Scenario: Intentar enviar una valoración excediendo el límite de caracteres de un comentario**

Se empieza en la página del producto a valorar. Se pone un nombre, comentario y estrellas para dar tu opinión. Un mensaje de error debería aparecer por superar los límites de carácter.

```
Scenario: Intentar enviar una valoración excediendo el límite de caracteres de un comentario
Given El usuario está en la página del producto que desea valorar
When El usuario accede a la sección de valoraciones
And El usuario escribe su nombre como "Juan Pérez"
And El usuario escribe un comentario "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
And El usuario selecciona una calificación de 4 estrellas
And El usuario hace clic en el botón "Continue"
Then Debería aparecer un mensaje de error "Warning: Review Text must be between 25 and 1000 characters!"
```

Steps usados:

- ```
- void el_usuario_esta_en_la_pagina_del_producto_que_desea_valorar() + void
  clickButton(By locator)
- void el_usuario_accede_a_la_seccion_de_valoraciones()
- void el_usuario_escribe_su_nombre_como(String nombre)
- void el_usuario_escribe_un_comentario(String comentario)
- void el_usuario_selecciona_una_calificacion_de_estrellas(int estrellas) + void
  clickButton(By locator)
- void el_usuario_hace_clic_en_el boton(String boton) + void clickButton(By locator)
- void deberia_aparecer_un_mensaje_de_error(String mensajeError)
```

- **Scenario: Intentar enviar una valoración sin seleccionar una calificación**

Se empieza en la página del producto a valorar. Se pone un nombre y comentario, pero falta una calificación. Un mensaje de aviso debería aparecer.

```
Scenario: Intentar enviar una valoración sin seleccionar una calificación
  Given El usuario está en la página del producto que desea valorar
  When El usuario accede a la sección de valoraciones
  And El usuario escribe su nombre como "Juan Pérez"
  And El usuario escribe un comentario "Buen producto, pero podría mejorar."
  And El usuario hace clic en el botón "Continue"
  Then Debería aparecer un mensaje de error "Warning: Please select a review rating!"
```

Steps usados:

- ```
- void el_usuario_esta_en_la_pagina_del_producto_que_desea_valorar() + void
  clickButton(By locator)
- void el_usuario_accede_a_la_seccion_de_valoraciones()
- void el_usuario_escribe_su_nombre_como(String nombre)
- void el_usuario_escribe_un_comentario(String comentario)
- void el_usuario_hace_clic_en_el_botón(String botón) + void clickButton(By locator)
- void debería_aparecer_un_mensaje_de_error(String mensajeError)
```

- **Scenario: Intentar enviar una valoración sin escribir un nombre**

Se empieza en la página del producto a valorar. Se pone un comentario y estrellas para dar tu opinión, pero falta un nombre. Un mensaje de aviso debería aparecer.

```
Scenario: Intentar enviar una valoración sin escribir un nombre
  Given El usuario está en la página del producto que desea valorar
  When El usuario accede a la sección de valoraciones
  And El usuario escribe un comentario "Buen producto, pero podría mejorar."
  And El usuario selecciona una calificación de 4 estrellas
  And El usuario hace clic en el botón "Continue"
  Then Debería aparecer un mensaje de error "Warning: Review Name must be between 3 and 25 characters!"
```

Steps usados:

- ```
- void el_usuario_esta_en_la_pagina_del_producto_que_desea_valorar() + void
  clickButton(By locator)
- void el_usuario_accede_a_la_seccion_de_valoraciones()
- void el_usuario_escribe_un_comentario(String comentario)
- void el_usuario_selecciona_una_calificacion_de_estrellas(int estrellas) + void
  clickButton(By locator)
- void el_usuario_hace_clic_en_el_boton(String boton) + void clickButton(By locator)
- void deberia_aparecer_un_mensaje_de_error(String mensajeError)
```

- **Scenario: Intentar enviar una valoración excediendo el límite de caracteres de un nombre**

Se empieza en la página del producto a valorar. Se pone un nombre, comentario y estrellas para dar tu opinión. Un mensaje de error debería aparecer por superar los límites de carácter.

```
Scenario: Intentar enviar una valoración excediendo el limite de caracteres de un nombre
Given El usuario está en la página del producto que desea valorar
When El usuario accede a la sección de valoraciones
And El usuario escribe su nombre como "JuanJuanJuan PérezPérezPérez"
And El usuario escribe un comentario "Buen producto, pero podría mejorar."
And El usuario selecciona una calificación de 4 estrellas
And El usuario hace clic en el botón "Continue"
Then Debería aparecer un mensaje de error "Warning: Review Name must be between 3 and 25 characters!"
```

Steps usados:

- void el\_usuario\_esta\_en\_la\_pagina\_del\_producto\_que\_desea\_valorar() + void clickButton(By locator)
- void el\_usuario\_accede\_a\_la\_seccion\_de\_valoraciones()
- void el\_usuario\_escribe\_su\_nombre\_como(String nombre)
- void el\_usuario\_escribe\_un\_comentario(String comentario)
- void el\_usuario\_selecciona\_una\_calificacion\_de\_estrellas(int estrellas) + void clickButton(By locator)
- void el\_usuario\_hace\_clic\_en\_el\_boton(String boton) + void clickButton(By locator)
- void deberia\_aparecer\_un\_mensaje\_de\_error(String mensajeError)

## FEATURE 15: Guardar productos en la lista de deseos

Este feature permite comprobar si se puede comprar productos más tarde o realizar algún seguimiento. Así, como usuario, quiero poder agregar productos a mi lista de deseos, ver los productos guardados y eliminarlos desde la sección de lista de deseos.

Se han realizado varios escenarios que abarcan diferentes casos que analizar. En este feature se han realizado 8 escenarios:

### - **Scenario: Agregar un producto a la lista de deseos**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de detalles de un producto cualquiera para añadirlo a la lista de deseos.

```
Scenario: Agregar un producto a la lista de deseos
Given El usuario está logeado en su cuenta
  | input-email | javi.david@test.com |
  | input-password | Password123 |
And El usuario está en la página de detalles de un producto
When El usuario hace clic en el botón "Add to Wish List" para anadir el producto a lista de deseos
Then El sistema debería mostrar una notificación confirmando que el producto "HP LP3065" ha sido agregado
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void usuarioEnPaginaDeDetallesDelProducto() + void clickButton(By locator)
- void usuarioHaceClicEnBoton(String boton)
- void sistemaMuestraNotificacionProductoAgregado(String producto)

### - **Scenario: Verificar el producto en la lista de deseos**

Se empieza en una nueva pestaña, con el programa iniciando sesión como usuario y verifica que el producto anterior se encuentra en la lista de deseos.

```
Scenario: Verificar el producto en la lista de deseos
Given El usuario está logeado en su cuenta
  | input-email | javi.david@test.com |
  | input-password | Password123 |
When El usuario hace clic en el botón "wishlist-total" para ir a la lista de deseos
Then La lista debería mostrar el producto "HP LP3065" previamente agregado
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void usuarioHaceClicEnBotonIrAListaDeseos(String boton) + void clickButton(By locator)
- void listaDeDeseosMuestraProducto(String nombreProducto)

### - **Scenario: Agregar nuevamente un producto ya en la lista de deseos**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de detalles de un producto cualquiera para añadirlo a la lista de deseos.

```
Scenario: Agregar nuevamente un producto ya en la lista de deseos
Given El usuario está logeado en su cuenta
  | input-email | javi.david@test.com |
  | input-password | Password123 |
And El usuario está en la página de detalles de un producto
When El usuario hace clic en el botón "Add to Wish List" para anadir el producto a lista de deseos
Then El sistema debería mostrar una notificación confirmando que el producto "HP LP3065" ha sido agregado
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void usuarioEnPaginaDeDetallesDelProducto() + void clickButton(By locator)
- void usuarioHaceClicEnBoton(String boton)
- void sistemaMuestraNotificacionProductoAgregado(String producto)

- **Scenario: Verificar que el producto no se duplica**

Se empieza con el programa iniciando sesión como usuario y yendo a la lista de deseos para comprobar que el producto añadido anteriormente no se ha duplicado.

```
Scenario: Verificar que el producto no se duplica
Given El usuario está logeado en su cuenta
  | input-email | javi.david@test.com |
  | input-password | Password123 |
When El usuario hace clic en el botón "wishlist-total" para ir a la lista de deseos
Then La lista de deseos no debería contener duplicados de "HP LP3065"
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void usuarioHaceClicEnBotonIrAListaDeseos(String boton) + void clickButton(By locator)
- void listaDeDeseosSinDuplicados(String nombreProducto)

- **Scenario: Agregar un producto diferente a la lista de deseos**

Se empieza con el programa iniciando sesión como usuario y yendo a la página de detalles de un producto cualquiera para añadirlo a la lista de deseos.

```
Scenario: Agregar un producto diferente a la lista de deseos
Given El usuario está logeado en su cuenta
  | input-email | javi.david@test.com |
  | input-password | Password123 |
And El usuario está en la página de detalles de un producto diferente
When El usuario hace clic en el botón "Add to Wish List" para añadir el producto a lista de deseos
Then El sistema debería mostrar una notificación confirmando que el producto "Samsung Galaxy Tab 10.1" ha sido agregado
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void usuarioEnPaginaDeDetallesDelProducto() + void clickButton(By locator)
- void usuarioHaceClicEnBoton(String boton)
- void sistemaMuestraNotificacionProductoAgregado(String producto)

- **Scenario: Verificar que ambos productos están en la lista de deseos**

Se empieza con el programa iniciando sesión como usuario y yendo a la lista de deseos para comprobar que todos los productos están añadidos correctamente.

```
Scenario: Verificar que ambos productos están en la lista de deseos
Given El usuario está logeado en su cuenta
  | input-email | javi.david@test.com |
  | input-password | Password123 |
When El usuario hace clic en el botón "wishlist-total" para ir a la lista de deseos
Then La lista debería mostrar el producto "HP LP3065" previamente agregado
And La lista debería mostrar el producto "Samsung Galaxy Tab 10.1" previamente agregado
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void usuarioHaceClicEnBotonIrAListaDeseos(String boton) + void clickButton(By locator)
- void listaDeDeseosMuestraProducto(String nombreProducto)
- void listaDeDeseosMuestraProducto(String nombreProducto)

- **Scenario: Eliminar un producto de la lista de deseos**

Se empieza con el programa iniciando sesión como usuario y yendo a la lista de deseos para eliminar un producto.

```
Scenario: Eliminar un producto de la lista de deseos
Given El usuario está logeado en su cuenta
  | input-email | javi.david@test.com |
  | input-password | Password123 |
When El usuario hace clic en el botón "wishlist-total" para ir a la lista de deseos
And El usuario hace clic en el botón "Remove" junto al producto "HP LP3065"
Then El producto "HP LP3065" debería ser eliminado de la lista de deseos
And El sistema debería mostrar una notificación confirmando que el producto ha sido eliminado
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void usuarioHaceClicEnBotonIrAListaDeseos(String boton) + void clickButton(By locator)
- void usuarioHaceClicEnEliminar(String boton, String producto)
- void productoEliminadoDeListaDeseos(String producto)
- void sistemaMuestraNotificacionProductoEliminado()

- **Scenario: Eliminar todos los productos de la lista de deseos y verificar la lista vacía**

Se empieza con el programa iniciando sesión como usuario y yendo a la lista de deseos para eliminar todos los productos.

```
Scenario: Eliminar todos los productos de la lista de deseos y verificar la lista vacía
Given El usuario está logeado en su cuenta
  | input-email | javi.david@test.com |
  | input-password | Password123 |
When El usuario hace clic en el botón "wishlist-total" para ir a la lista de deseos
And El usuario hace clic en el botón "Remove" junto al producto "Samsung Galaxy Tab 10.1"
Then El producto "Samsung Galaxy Tab 10.1" debería ser eliminado de la lista de deseos
And El sistema debería mostrar una notificación confirmando que el producto ha sido eliminado
And El sistema debería mostrar un mensaje indicando "Your shopping cart is empty!" cuando no haya productos en la lista
```

Steps usados:

- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) + void clickButton(By locator) + void fillForm(Map<String, String> fields)
- void usuarioHaceClicEnBotonIrAListaDeseos(String boton) + void clickButton(By locator)
- void usuarioHaceClicEnEliminar(String boton, String producto)
- void productoEliminadoDeListaDeseos(String producto)
- void sistemaMuestraNotificacionProductoEliminado()
- void sistemaMuestraMensajeListaVacía(String mensaje)

# FEATURES EXTRA

Uno de los miembros del grupo ha decidido hacer más features, concretamente dos, por si alguno de los features anteriores estaba mal implementado o no era lo bastante bueno. En esta pagina se comentarán brevemente, ya que no pertenecen realmente a los features principales, sino que son más un seguro para nosotros.

- **FEATURE 16: Suscripción a boletín de noticias**

Este feature permite comprobar si se puede recibir las ultimas ofertas y novedades. Así, como usuario, quiero poder suscribirme y desuscribirme al boletín de noticias. En este feature se han realizado 2 escenarios.

- **FEATURE 17: Contactar con soporte al cliente**

Este feature permite comprobar si se puede resolver problemas con mi pedido o cuenta. Así, como usuario, quiero poder contactar con el soporte al cliente. En este feature se han realizado 3 escenarios.



# STEPS

| Funciones:                                                                                           | Veces usadas: |
|------------------------------------------------------------------------------------------------------|---------------|
| void TheUserIsInTheIndexPage()                                                                       | 23            |
| void TheUserClicksToGoToTheRegistrationPage()                                                        | 5             |
| void<br>theUserFillsInTheRegistrationFormWithTheFollowingDetails(Map<String,<br>String> userDetails) | 4             |
| void TheUserAgreesToThePrivacyPolicy()                                                               | 4             |
| void TheUserClicksOnTheContinueButton()                                                              | 9             |
| void theUserShouldSeeAConfirmationMessage()                                                          | 1             |
| void theUserShouldSeeAEmailWarningMessage()                                                          | 1             |
| void theUserShouldSeeAUsedEmailWarningMessage()                                                      | 1             |
| void TheUserShouldBeInTheSamePage()                                                                  |               |
| void TheUserShouldSeeABunchOfWarningMessages()                                                       | 1             |
| void ThePageShouldIndicateThatTheProvidedParametersAreInvalid()                                      | 1             |
|                                                                                                      |               |
| void TheUserClicksToGoToTheLoginPage()                                                               | 6             |
| void theUserFillsEmailAndPassword(Map<String, String> userDetails)                                   | 2             |
| void TheUserClicksOnTheLoginButton()                                                                 | 3             |
| void TheUserShouldSeeTheAccountPage()                                                                | 1             |
| void theUserShouldSeeANoMatchingMessage()                                                            | 2             |
|                                                                                                      |               |
| void TheUserIsOnTheLoginPage()                                                                       | 3             |
| void TheUserIsInTheShoppingCartPage()                                                                | 1             |
| void TheUserIsOnMyAccountPage()                                                                      | 1             |
| void theUserClicksOnMacBookFromFeatured()                                                            | 1             |
| void theUserClicksAddToCartButton()                                                                  | 1             |
| void TheUserClicksOnCheckoutButton()                                                                 | 1             |
| void TheUserClicksOnAnotherContinueButton()                                                          | 1             |
| void TheUserClicksOnTheConfirmOrderButton()                                                          | 1             |
| void TheUserClicksOnOrderHistory()                                                                   | 1             |
| void TheUserHaveAProductInTheShoppingCart()                                                          | 1             |
| void TheUserRemovesOneProduct()                                                                      | 1             |
| void TheUserHaveMultipleProductsInTheShoppingCart()                                                  | 1             |
| void TheUserRemovesOneByOne()                                                                        | 1             |
| void TheProductShouldDisappearFromTheCart()                                                          | 2             |
| void theUserShouldSeeTheCartSummary()                                                                | 1             |
| void theUserShouldSeeAnOrderConfirmationMessage()                                                    | 1             |
| void TheUserShouldSeeTheDateOfTheLastProductToConfirm()                                              | 1             |
|                                                                                                      |               |
| theUserClicksOnTheCurrencyDropdownToSelectCurrency(String<br>currency)                               | 3             |
| void thePricesOnTheHomepageShouldBeDisplayedAllInCurrency(String<br>currencySymbol)                  | 3             |
|                                                                                                      |               |
| void TheUserClicksOnForgottenPassword()                                                              | 3             |
| void theUserFillsEmail(Map<String, String> userDetails)                                              | 2             |
| void TheUserShouldSeeAConfirmationMessageOnAnotherPage()                                             | 1             |

|                                                                        |   |
|------------------------------------------------------------------------|---|
| void TheUserShouldSeeAWarningMessageOnTheSamePage()                    | 2 |
|                                                                        |   |
| void TheUserSearchesForLaptopNotebooks()                               | 1 |
| void TheUserClicksCompareForHP()                                       | 1 |
| void TheUserClicksCompareForMacBook()                                  | 1 |
| void TheUserNavigatesToTheProductComparisonPage()                      | 1 |
| void AddToCartthebetterone()                                           | 1 |
| void TheBestLaptopIsInTheShoppingCart()                                | 1 |
|                                                                        |   |
| void TheUserBuysAProduct()                                             | 1 |
| void TheUserGoesToOrderHistoryInMyAccount()                            | 1 |
| void TheUserOpensTheDetailsOfTheLastOrder()                            | 1 |
| void TheUserClicksOnTheReturnButton()                                  | 1 |
| void TheUserFillTheObligatoryOptionsToReturnTheProduct()               | 1 |
| void TheUserShouldSeeInTheReturnPageTheProduct()                       | 1 |
|                                                                        |   |
| void TheUserAddProductsToTheShoppingCart()                             | 2 |
| void TheUserBuysTheProductWithTheGuessOption()                         | 1 |
| void TheUserTriesToProceedWithoutAShippingDetails()                    | 1 |
| void<br>TheUserShouldBeAbleToEnterTheShippingDetailsWithoutAnAccount() | 1 |
| void WarningMessageAboutTheShippingDetailsShouldAppear()               | 1 |
|                                                                        |   |

| Funciones:                                                                             | Veces usadas: |
|----------------------------------------------------------------------------------------|---------------|
| void usuarioEnPaginaInicio()                                                           | 2             |
| void introduceEnBarraBusqueda(String producto)                                         | 2             |
| void clicEnBotonBuscar()                                                               | 2             |
| void aparecenResultadosRelevantes(String producto)                                     | 2             |
| void noAparecenResultados(String mensaje)                                              | 2             |
| (private) boolean verificarElementoConTexto(String selector, String textoEsperado)     | 2             |
|                                                                                        |               |
| void elUsuarioEstaEnLaCategoriaDeDesktops()                                            | 6             |
| void elUsuarioSeleccionaEnElMenuDeOrdenacion(String criterio)                          | 6             |
| void losProductosDeberianMostrarseOrdenadosPorPrecioAscendente()                       | 1             |
| void losProductosDeberianMostrarseOrdenadosPorPrecioDescendente()                      | 1             |
| void losProductosDeberianMostrarseOrdenadosAlfabeticamenteAZ()                         | 1             |
| void losProductosDeberianMostrarseOrdenadosAlfabeticamenteZA()                         | 1             |
| void losProductosDeberianMostrarseConLasCalificacionesMasAltasPrimero()                | 1             |
| void losProductosDeberianMostrarseConLasCalificacionesMasBajasPrimero()                | 1             |
| (private) void verificarOrdenDePrecios(boolean ascendente)                             | 2             |
| (private) void verificarOrdenAlfabetico(boolean ascendente)                            | 2             |
| (private) void verificarOrdenDeCalificaciones(boolean descendente)                     | 2             |
|                                                                                        |               |
| void el_usuario_esta_logeado_en_su_cuenta(Map<String, String> userDetails)             | 30            |
| void el_usuario_hace_clic_en_el_enlace_password(String enlace)                         | 21            |
| void el_usuario_introduce_una_nueva_contraseña(Map<String, String> newPasswordDetails) | 5             |
| void el_usuario_introduce_la_confirmacion_de_la_nueva_contraseña(Map<String, String>   | 3             |

|                                                                                                  |    |
|--------------------------------------------------------------------------------------------------|----|
| confirmPasswordDetails)                                                                          |    |
| void el_usuario_hace_clic_en_el_boton_de_continuar_para_guardar_la_nueva_contraseña()            | 6  |
| void el_sistema_debería_mostrar_un_mensaje_de_error_en_la_contraseña()                           | 2  |
| void el_sistema_debería_mostrar_un_mensaje_de_error_indicando_que_las_contraseñas_no_coinciden() | 3  |
| void la_contraseña_del_usuario_debería_cambiarse_con_éxito()                                     | 2  |
| (private) void clickButton(By locator)                                                           | 24 |
| (private) void fillForm(Map<String, String> fields)                                              | 7  |
|                                                                                                  |    |
| void el_usuario_actualiza_su_informacion_personal(Map<String, String> userDetails)               | 3  |
| void el_mensaje_de_exito_deberia_aparecer()                                                      | 2  |
| void la_informacion_personal_del_usuario_deberia_actualizarse_correctamente()                    | 1  |
| void la_informacion_personal_del_usuario_deberia_restaurarse_correctamente()                     | 1  |
| void el_usuario_deja_los_campos_vacios_y_hace_clic_en_continue()                                 | 1  |
| void el_sistema_deberia_mostrar_mensajes_de_error()                                              | 2  |
| (private) void verifySuccessMessage(String message)                                              | 2  |
| (private) void verifyUpdatedPersonalInfo(Map<String, String> expectedValues)                     | 2  |
| (private) void verifyErrorMessages(Map<String, String> errorMessages)                            | 3  |
|                                                                                                  |    |
| void tryAddingEmptyAddress()                                                                     | 1  |
| void verifyAllFieldsError()                                                                      | 3  |
| void addNewAddress(Map<String, String> addressDetails)                                           | 1  |
| void verifyAddressSaved()                                                                        | 1  |
| void el_usuario_intenta_editar_borrando_campos()                                                 | 1  |
| void editAddress(Map<String, String> updatedAddress)                                             | 2  |
| void verifyAddressUpdated()                                                                      | 1  |
| void deleteAddress()                                                                             | 1  |
| void verifyAddressDeleted()                                                                      | 1  |
| void tryDeletingDefaultAddress()                                                                 | 1  |
| void verifyCannotDeleteLastAddress()                                                             | 1  |
| (private) boolean verifyAddressInList(String[] expectedAddress)                                  | 4  |
| (private) boolean isAddressMatching(String cellText, String[] expectedAddress)                   | 2  |
| (private) void performActionOnAddress(String[] expectedAddress, By actionButtonLocator)          | 3  |
|                                                                                                  |    |
| void el_usuario_esta_en_la_pagina_del_producto_que_desea_valorar()                               | 6  |
| void el_usuario_accede_a_la_seccion_de_valoraciones()                                            | 6  |
| void el_usuario_escribe_su_nombre_como(String nombre)                                            | 5  |
| void el_usuario_escribe_un_comentario(String comentario)                                         | 5  |
| void el_usuario_selecciona_una_calificacion_de_estrellas(int estrellas)                          | 5  |
| void el_usuario_hace_clic_en_el_boton(String boton)                                              | 6  |
| void deberia_aparecer_un_mensaje_de_exito_que_dice(String mensajeExito)                          | 3  |
| void deberia_aparecer_un_mensaje_de_error(String mensajeError)                                   | 5  |
|                                                                                                  |    |
| void usuarioEnPaginaDeDetallesDelProducto()                                                      | 2  |
| void usuarioHaceClicEnBoton(String boton)                                                        | 3  |
| void sistemaMuestraNotificacionProductoAgregado(String producto)                                 | 3  |
| void usuarioHaceClicEnBotonIrAListaDeseos(String boton)                                          | 5  |
| void listaDeDeseosMuestraProducto(String nombreProducto)                                         | 3  |
| void listaDeDeseosSinDuplicados(String nombreProducto)                                           | 1  |
| void usuarioEnPaginaDeDetallesDeUnProductoDiferente()                                            | 1  |

|                                                                             |   |
|-----------------------------------------------------------------------------|---|
| void usuarioHaceClicEnEliminar(String boton, String producto)               | 2 |
| void productoEliminadoDeListaDeseos(String producto)                        | 2 |
| void sistemaMuestraNotificacionProductoEliminado()                          | 5 |
| void sistemaMuestraMensajeListaVacía(String mensaje)                        | 1 |
|                                                                             |   |
| void elUsuarioHaceClicEn(String opcion)                                     | 2 |
|                                                                             |   |
| void elUsuarioEstaEnLaPaginaDeContacto()                                    | 3 |
| void elUsuarioIngresaDatosDeContacto(Map<String, String> dataTable)         | 2 |
| void elUsuarioIngresaDatosDeContacto()                                      | 1 |
| void elSistemaDeberiaRedirigirALaPaginaDeExito()                            | 1 |
| void elSistemaDeberiaMostrarMensajesDeErrorIndicandoLosCamposObligatorios() | 2 |

Steps más usados:

- void TheUserIsInTheIndexPage() : Hace que el programa comience en la pagina principal, osea, en <http://opencart.abstracta.us/index.php?route=common/home>
- void TheUserClicksOnTheContinueButton() : La página está llena de botones que ponen continue, esta función hace click en esos botones.
- void TheUserClicksToGoToTheLoginPage() : Muchas funcionalidades son accesibles si se inicia sesión, por lo que se está loggeando constantemente. (Versión Javi)
- void el\_usuario\_esta\_logeado\_en\_su\_cuenta(Map<String, String> userDetails) : Muchas funcionalidades son accesibles si se inicia sesión, por lo que se está loggeando constantemente. (Versión David)
- void el\_usuario\_hace\_clic\_en\_el\_enlace\_password(String enlace) : La página está llena de textos que llevan a otras paginas de la web, esta función hace click en dichos enlaces.
- (private) void clickButton(By locator) : Una función privada para hacer click a cualquier indicador.
- (private) void fillForm(Map<String, String> fields) : Una función privada para rellenar un formulario. Es complementario de iniciar sesión.