

# **INFORME DE PRUEBAS PARA LOS CASOS DE USO**

Alfonso Arroyo

David Pilaguano

Stalin Uvidia

**Departamento de Ciencias de la Computación, Universidad de  
las Fuerzas Armadas ESPE**

**14538 – Fundamentos de la Ingeniería de Software**

**Ing. Jenny Alexandra Ruiz Robalino**

**02 de septiembre de 2024**

## **1. Introducción.**

Este informe detalla el proceso mediante el cual se llevaron a cabo las pruebas de aceptación en los casos de uso del proyecto. Se utilizaron diversas herramientas y metodologías, con un enfoque particular en la herramienta Cucumber, que permite la automatización de pruebas mediante la creación de escenarios en lenguaje natural. El propósito de estas pruebas es verificar que cada funcionalidad del sistema opere de manera correcta y eficiente, asegurando que los usuarios finales puedan interactuar con el sistema de manera intuitiva y sin errores.

## **2. Objetivos**

### **2.1. Objetivo General**

Desarrollar pruebas de aceptación en el prototipo de los casos de uso del proyecto, todo esto mediante la herramienta de Cucumber para validar el correcto funcionamiento en reglas aprobadas o fallidas.

### **2.2. Objetivos Específicos**

- Establecer un entorno de desarrollo necesario para el desarrollo de pruebas automatizadas mediante la configuración del lenguaje de programación Java en ellos prototipos de casos de uso de todas las pantallas
- Realizar capturas de la ejecución para simular la interacción con el usuario en cada pantalla verificando todos los casos de uso y requisitos funcionales
- Recopilar los resultados de las pruebas automatizadas en un archivo Excel para analizar los resultados y obtener posibles mejoras

## **3. Entradas a utilizar**

Se emplearán todos los campos de texto de cada pantalla utilizada como entradas a utilizar. En las tablas 1 2, y 3 se visualizan las entradas, con sus respectivos estados esperados al realizar las pruebas de aceptación del programa.

Cada campo de texto será sometido a diferentes pruebas, incluyendo la introducción de datos válidos, inválidos, datos límite, y pruebas de campo vacío, para asegurar que la aplicación maneje correctamente todas las posibles entradas de los usuarios.

Tabla 1 Ingreso al sistema | Login

VARIABLE	CLASE DE EQUIVALENCIA		ESTADO	REPRESENTANTE
userTxt passTxt	Ec1: userTxt:("txtUser") passTxt:("txtPass")		Valido	Admin admin resultado Exito!
	Ec2: userTxt:("txtUser") passTxt:("txtUser")		Invalido	Emilio Padilla Resultado Usuario o contraseña incorrecta por favor inténtelo de nuevo
	Ec3: userTxt:("UpperCase") passTxt:("TxtPass")		Invalido	JUAN admin Resultado Usuario o contraseña incorrecta por favor inténtelo de nuevo
	Ec4: userTxt:("TxtUser") passTxt:("UpperCase")		Invalido	pedro ESCOBAR Resultado Usuario o contraseña incorrecta por favor inténtelo de nuevo

Tabla 2 Añadir productos | AddProducts verify the ID

VARIABLE	CLASE DE EQUIVALENCIA		ESTADO	REPRESENTANTE
idFldText id	Ec1:  if (!id.matches("\\d*")) {		Invalido	Entrada: "a"  Resultado:  El campo ID solo puede contener números, entrada inválida
	Ec2:  if (!id.matches("\\d*")) {		Invalido	Entrada: "s"  Resultado:  El campo ID solo puede contener números, entrada inválida
	Ec3:  if (!id.matches("\\d*")) {		Invalido	Entrada: "1"  Resultado:  El ID debe comenzar con un 0
	Ec4:  if (!id.matches("\\d*")) {		Valido	Entrada: "0123"  Resultado:  Producto añadido exitosamente

Tabla 3 Editar Productor | EditProducts verify the ID

VARIABLE	CLASE DE EQUIVALENCIA		ESTADO	REPRESENTANTE
Product product	Ec1: <code>if {cloudDB.productExists(id)}</code>		Valido	Entrada: "03"  Resultado:  Id ingresado existente, permite la modificación en la base de datos
	Ec2: <code>if {cloudDB.productExists(id)}</code>		Invalido	Entrada: "a"  Resultado:  Id ingresado inexistente, no se permite la modificación en la base de datos
	Ec3: <code>if {cloudDB.productExists(id)}</code>		Invalido	Entrada: "     "  Resultado:  Id ingresado inexistente, no se permite la modificación en la base de datos
	Ec4: <code>if {cloudDB.productExists(id)}</code>		Invalido	Entrada: "admin"  Resultado:  Id ingresado inexistente, no se permite la modificación en la base de datos

En las tablas mencionadas, se detallan cada uno de estos campos, los valores de entrada probados, y el comportamiento esperado del sistema en cada caso. Esto proporciona una base clara y completa para evaluar si el programa cumple con los criterios de aceptación definidos.

#### 4. Ambiente de Desarrollo de pruebas

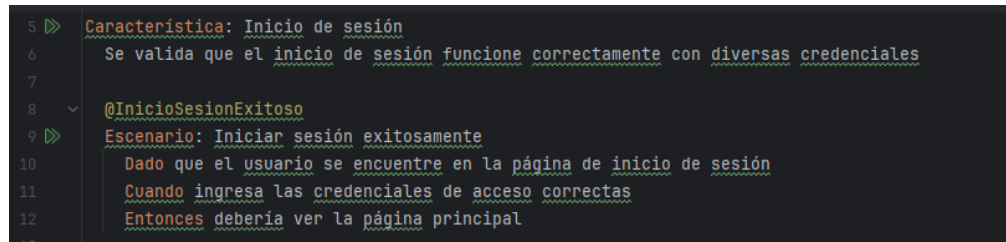
Para el desarrollo de pruebas, se utilizó la herramienta Java con el IDE IntelliJ en el cual se creo un ambiente, en el mismo se creo diferentes archivos con el propósito de desarrollar las pruebas, esto se lo realiza mediante el uso de algunas dependencias como se recomendaba en la página web Cucumber <https://cucumber.io/docs/installation/java/>

A continuación, se detallará mas el contenido de los diferentes archivos dentro del proyecto de pruebas:

- Archivo `iniciarSesion.feature`: en este se encuentran los casos a evaluar dentro del caso de uso Iniciar Sesión, en el cual se puede encontrar los escenarios y las preguntas que se realiza en cada caso.

Figura 1

*Archivo `iniciarSesion.feature`:*



```
5 >> Característica: Inicio de sesión
6     Se valida que el inicio de sesión funcione correctamente con diversas credenciales
7
8     @InicioSesionExitoso
9 >> Escenario: Iniciar sesión exitosamente
10    Dado que el usuario se encuentre en la página de inicio de sesión
11    Cuando ingresa las credenciales de acceso correctas
12    Entonces debería ver la página principal
13
```

En la figura 1 se muestra el feature "Inicio de sesión", en el cual existe un primer escenario llamado "Iniciar sesión exitosamente". Este escenario es considerado exitoso ya que se verifica que, al ingresar las credenciales de acceso correctas en la página de inicio de sesión, el usuario debería ser redirigido a la página principal, lo que confirma que el proceso de autenticación ha sido completado correctamente.

- Archivo `loginPasos.java`: en este archivo se encuentran los pasos evaluados en lenguaje java con entradas de datos para probar el escenario de inicio de sesión del proyecto de pruebas automatizadas

Figura 2

*Archivo `loginPasos.java`*

```

7 public class loginPasos { Alfonso Alejandro Arroyo Paredes +1
9     FrmLogin frmLogin = new FrmLogin(); 18 usages
10     FrmUniversoDelFomix frmUniversoDelFomix = new FrmUniversoDelFomix(); no usages
11
12     @Dado("que el usuario se encuentre en la página de inicio de sesión") Alfonso Alejandro Arroyo Paredes
13     public void que_el_usuario_se_encuentre_en_la_página_de_inicio_de_sesión() { frmLogin.setVisible(true); }
14
15
16
17     @Cuando("ingresa las credenciales de acceso correctas") Alfonso Alejandro Arroyo Paredes
18     public void ingresa_las_credenciales_de_acceso_correctas() {
19         frmLogin.userTxt.setText("admin");
20         frmLogin.passTxt.setText("admin");
21         frmLogin.loginBtnTxtMouseClicked( evt: null);
22     }
23
24
25     @Entonces("debería ver la página principal") Alfonso Alejandro Arroyo Paredes
26     public void debería_ver_la_página_principal() {
27         if (!frmLogin.isVisible()) {
28             System.out.println("La página principal está abierta.");
29         } else {
30             System.out.println("La página principal no se ha abierto.");
31         }
32     }
33
34     @Cuando("ingresa las credenciales de acceso incorrectas") Alfonso Alejandro Arroyo Paredes
35     public void ingresa_las_credenciales_de_acceso_incorrectas() {
36         frmLogin.userTxt.setText("usuario_incorrecto");
37         frmLogin.passTxt.setText("contraseña_incorrecta");
38         frmLogin.loginBtnTxtMouseClicked( evt: null);
39     }
40
41     @Cuando("ingresa un nombre de usuario correcto y una contraseña incorrecta") Alfonso Alejandro Arroyo Paredes
42     public void ingresa_un_nombre_de_usuario_correcto_y_una_contraseña_incorrecta() {
43         frmLogin.userTxt.setText("admin");

```

Dentro de la figura anterior se muestra como se crearon las funciones en base al .feature anterior determinado en el escenario. Para acceder a todos los atributos y entradas del programa se usa las importaciones del proyecto.

- Archivo añadirProductos.feature: Este archivo define los escenarios de prueba para la funcionalidad de añadir productos. Incluye casos de uso tanto exitosos como fallidos, especificando los pasos y condiciones a evaluar. Los escenarios detallan las acciones del usuario y las respuestas esperadas del sistema, asegurando una prueba exhaustiva de la capacidad de añadir productos y manejar entradas incorrectas o situaciones especiales.

Figura 3

*Archivo añadirProductos.feature:*

```

5 >> Característica: Añadir Productos
6
7     @ProductoAñadidoExitosamente
8 >> Escenario: Añadir un producto nuevo exitosamente
9
10     Dado que el usuario se encuentre en la pagina de añadir productos
11     Cuando ingresa un producto valido
12     Entonces deberá ingresar un producto nuevo

```

En la figura 3 se muestra el feature "Añadir Producto", en el cual existe un primer escenario llamado "Añadir un producto nuevo exitosamente". Este escenario es considerado exitoso ya que se verifica que, el usuario ingrese un producto valido para que este sea subido a la base de datos exitosamente y se mantenga en la misma pantalla.

- Archivo añadirProductosPasos.java: en este archivo se encuentran los pasos evaluados en lenguaje java con entradas de datos válidas para probar el escenario de ingreso de un producto a la base de datos.

Figura 4

*Archivo añadirProductosPasos.java*

```
12 public class añadirProductosPasos { // David Pilaguano
13     CloudDB cloudDB = new CloudDB(); // 6 usages
14     FrmAddProduct frmAddProduct = new FrmAddProduct(); // 25 usages
15     FrmUniversoDelFomix frmUniversoDelFomix = new FrmUniversoDelFomix(); // 1 usage
16     Product product; // 8 usages
17
18
19     @Dado("que el usuario se encuentre en la pagina de añadir productos") // David Pilaguano
20     public void que_el_usuario_se_encuentre_en_la_pagina_de_añadir_productos() {frmAddProduct.setVisible(true);}
21
22     @Cuando("ingresa un producto valido") // David Pilaguano
23     public void ingresa_un_producto_valido() {
24         frmAddProduct.idFldText.setText("0548");
25         frmAddProduct.nameFldText.setText("Mongo 2H");
26         frmAddProduct.descriptionFldText.setText("lapiz punta fina");
27         frmAddProduct.quantityFldText.setText("500");
28         frmAddProduct.categoryFldText.setText("lapices");
29         frmAddProduct.priceFldText.setText("50");
30         frmAddProduct.sizeFldText.setText("mediano");
31         frmAddProduct.addBtnTextMouseClicked(evt: null);
32
33     }
34
35     @Entonces("deberá ingresar un producto nuevo") // David Pilaguano
36     public void deberá_ingresar_un_producto_nuevo() {
37         try {
38             if (cloudDB.productExists(product.getId())) {
39                 cloudDB.updateProduct(product);
40                 System.out.println("El producto se ha registrado correctamente");
41             } else {
42                 cloudDB.uploadProductData(product);
43                 System.out.println("El producto no se ha registrado correctamente");
44             }
45         } catch (Exception e) {
46             System.out.println("El producto se ha registrado correctamente ");
47         }
48     }
49 }
```

Dentro de la figura 4 se muestra cómo se crearon las funciones en base al .feature anterior determinado en el escenario, esto mediante los atributos y entradas del programa se pueden usar por las importaciones del proyecto.

- Archivo eliminarProductos.feature: Este archivo define los escenarios de prueba para la funcionalidad de eliminar productos. Incluye casos de uso exitosos, donde se verifica que el usuario pueda eliminar productos existentes, así como situaciones fallidas, como cuando se intenta eliminar un producto que no existe en la base de datos.

Figura 5

*Archivo eliminarProductos.feature:*

```
Característica: Eliminar Productos
@ProductoEliminadoExito
Escenario: Producto Eliminado de manera correcta
    Dado que el usuario se encuentre en la página de eliminar productos
    Cuando el usuario borre un producto
    Entonces el producto se deberá eliminar
```

En la figura 5 se muestra el feature "Eliminar Producto" en el cual existe un primer escenario llamado "Eliminar un producto exitosamente". Este escenario es considerado exitoso ya que se verifica que el usuario selecciona un producto existente y confirma su eliminación. El sistema debe eliminar el producto de la base de datos y notificar al usuario que la operación se completó correctamente.

- Archivo eliminarProductosPasos.java: En este archivo se encuentran los pasos evaluados en lenguaje Java para probar el escenario de eliminación de productos del proyecto de pruebas automatizadas. Aquí se implementan las funciones necesarias para acceder a la base de datos, eliminar el producto seleccionado, y verificar que la eliminación se realizó correctamente, asegurando que el sistema responde adecuadamente a esta operación.



Figura 6

*Archivo eliminarProductosPasos.java*

```
7 public class eliminarProductosPasos { ± David Pilaguano
8     FrmDeleteProduct frmDeleteProduct=new FrmDeleteProduct(); 3 usages
9     FrmUniversoDelFomix frmUniversoDelFomix = new FrmUniversoDelFomix(); 1 usage
10
11     @Dado("que el usuario se encuentre en la pagina de eliminar productos") ± David Pilaguano
12     public void que_el_usuario_se_encuentre_en_la_pagina_de_eliminar_productos() {frmDeleteProduct.setVisible(true);}
13
14     @Cuando("el usuario borre un producto") ± David Pilaguano
15     public void el_usuario_borre_un_producto() {
16         int selectedRow;
17         frmDeleteProduct.selectedRow=1;
18         selectedRow=frmDeleteProduct.selectedRow;
19         if(selectedRow==1)
20         {
21             System.out.println("Porfavor Seleccione un producto a eliminar");
22         }
23     }else{
24         System.out.println("Producto Eliminado Correctamente");
25     }
26
27 }
28
29 @Entonces("el producto se debera eliminar") ± David Pilaguano
30 public void el_producto_se_debera_eliminar() {
31
32 }
33
34 @Entonces("el producto no se eliminara") ± David Pilaguano
35 public void el_producto_no_se_eliminar() {
36
37 }
38
39 @Cuando("el usuario presione el boton regresar") ± David Pilaguano
40 public void elUsuarioPresioneElBotonRegresar() {
```

Dentro de la figura 6 se muestra cómo se crearon las funciones en base al .feature anterior determinado en el escenario. Se utilizan los métodos necesarios para buscar el producto en la base de datos, ejecutar la operación de eliminación, y luego verificar que el producto ya no está disponible. Esto asegura que el proceso de eliminación funcione sin errores y que el sistema responda de acuerdo a lo esperado.

- Archivo editarProductos.feature: Este archivo define los escenarios de prueba para la funcionalidad de edición de productos. Los escenarios incluyen la modificación de detalles como el nombre o el precio de un producto y verifican que el sistema actualice correctamente la base de datos con los nuevos valores introducidos.

Figura 7

*Archivo editarProductos.feature:*

```
Característica: Editar Productos
@ProductoEditadoExito
Escenario: Producto Editado de manera correcta
  Dado que el usuario se encuentre en la pagina de EditarProductos
  Cuando presione el boton editar
  Entonces el producto se debera editar
```

En la figura 7 se presenta el feature "Editar Producto", que contiene un escenario llamado "Editar un producto existente exitosamente". En este escenario, se valida que un usuario pueda modificar los detalles de un producto existente, como el nombre o el precio. El sistema debe actualizar la información en la base de datos y reflejar los cambios de manera inmediata.

- Archivo `editarProductosPasos.java`: Dentro de este archivo se encuentran los pasos evaluados en lenguaje Java para probar el escenario de edición de productos. Se implementan las funciones que permiten modificar los atributos de un producto, como el nombre o el precio, y verificar que los cambios se guardan correctamente en la base de datos, manteniendo la integridad de la información.

Figura 8

*Archivo `editarProductosPasos.java`:*

```

11 public class editarProductosPasos {  ⚡ David Pilaguano
12     FrmEditProduct frmEditProduct=new FrmEditProduct(); 2 usages
13     Product product; no usages
14     FrmUniversoDelFomix frmUniversoDelFomix= new FrmUniversoDelFomix(); 1 usage
15
16     @Dado("que el usuario se encuentre en la pagina de EditarProductos")  ⚡ David Pilaguano
17     public void que_el_usuario_se_encuentre_en_la_pagina_de_editar_productos() {
18         frmEditProduct.setVisible(true);}
19
20     @Cuando("presione el boton editar")  ⚡ David Pilaguano
21     public void presione_el_boton_editar() {
22         frmEditProduct.saveProduct();
23         System.out.println("Producto editado correctamente");
24     }
25
26
27     @Entonces("el producto se debiera editar")  ⚡ David Pilaguano
28     public void el_producto_se_debera_editar() {
29     }
30
31     @Entonces("el producto no se editara")  ⚡ David Pilaguano
32     public void elProductoNoSeEditara() {
33     }
34
35
36     @Entonces("regresara a la pagina principal")  ⚡ David Pilaguano
37     > public void regresaraALaPaginaPrincipal() { frmUniversoDelFomix.setVisible(true); }
38
39
40
41     @Cuando("presione el boton de regresar")  ⚡ David Pilaguano
42     > public void presioneElBotonDeRegresar() { System.out.println("boton ejecutado correctamente"); }
43
44 }
45
46

```

La figura 8 muestra cómo se implementaron los pasos en Java para el escenario de edición de productos. Aquí se manejan los cambios en los datos del producto, asegurando que los nuevos valores se guarden correctamente y que el sistema maneje adecuadamente cualquier error, como la introducción de valores no válidos.

- Archivo universoDelFomix.feature: Este archivo abarca un conjunto de escenarios para probar la funcionalidad integral del sistema de inventario del "Universo del Fomix".

Los escenarios incluyen la interacción entre diferentes módulos del sistema, como añadir, editar y eliminar productos, para garantizar que todas las funcionalidades trabajen en conjunto de manera coherente.

Figura 9

*Archivo universoDelFomix.feature:*

```

Característica: universoDelFomix

@BotonVerInventario
Escenario: funcionalidad boton ver Inventario
  Dado que se encuentre en la ventana principal
  Cuando presione el boton Ver Inventario
  Entonces se diriga a esa ventan

```

En la figura 9 se muestra el feature "Universo del Fomix", que incluye un conjunto de escenarios para probar la funcionalidad integral del sistema. Este archivo abarca la interacción entre diferentes módulos, como añadir, editar y eliminar productos, para garantizar que el sistema funcione de manera coherente cuando se utilizan varias funciones en secuencia.

- Archivo universoDelFomixPasos.java: Este archivo contiene los pasos en lenguaje Java que corresponden a los escenarios globales del proyecto, abarcando la funcionalidad integral del sistema "Universo del Fomix". Aquí se coordinan las operaciones de los distintos módulos, asegurando que funcionen en conjunto sin errores y validando que todas las acciones del usuario son manejadas correctamente por el sistema.

Figura 10

*Archivo universoDelFomixPasos.java:*

```

6 public class universoDelFomixPasos {  ± David Pilaguano
7     FrmEditProduct frmEditProduct =new FrmEditProduct(); 1 usage
8     FrmAddProduct frmAddProduct = new FrmAddProduct(); 1 usage
9     FrmViewInventory frmViewInventory=new FrmViewInventory(); 1 usage
10    FrmDeleteProduct frmDeleteProduct=new FrmDeleteProduct(); 1 usage
11
12
13    @Dado("que se encuentre en la ventana principal")  ± David Pilaguano
14    > public void que_se_encuentre_en_la_ventana_principal() { frmUniversoDelFomix.setVisible(true); }
15
16
17    @Cuando("presione el boton Ver Inventario")  ± David Pilaguano
18    public void presione_el_boton_ver_inventario() {
19
20    }
21
22
23    @Entonces("se diriga a esa ventan")  ± David Pilaguano
24    > public void se_diriga_a_esa_ventan() { frmViewInventory.setVisible(true); }
25
26
27
28    @Cuando("presione el boton aadirProducto")  ± David Pilaguano
29    public void presioneElBotonAadirProducto() {
30
31    }
32
33    @Entonces("se dirige a la venta aadirProductos")  ± David Pilaguano
34    > public void seDirigeALaVentaAadirProductos() { frmAddProduct.setVisible(true); }
35
36
37    @Cuando("presione el boton eliminarProducto")  ± David Pilaguano
38    public void presioneElBotonEliminarProducto() {
39        frmDeleteProduct.setVisible(true);
40    }
41
42
43    @Entonces("se diriga a esa ventana eliminarProducto")  ± David Pilaguano
44    public void seDirigaAEsaVentanaEliminarProducto() {
45
46    }

```

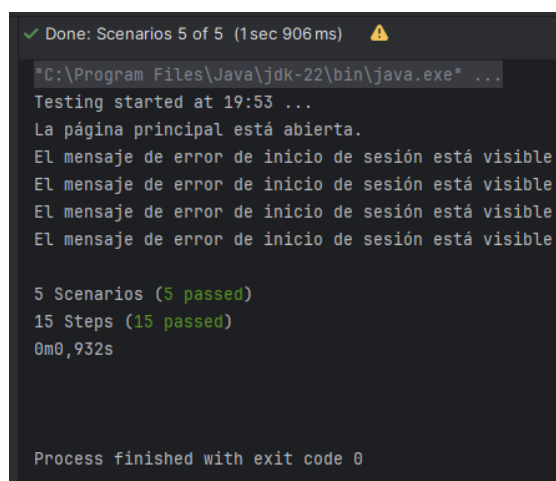
La figura 10 presenta el archivo "universoDelFomixPasos.java", que implementa los pasos para probar el sistema de manera global. Aquí se coordinan las acciones de los diferentes módulos para asegurarse de que todas las funciones trabajen juntas sin problemas. Los métodos en este archivo validan que la interacción entre las diversas partes del sistema no cause conflictos y que los resultados sean consistentes con las expectativas del usuario.

## 5. Ejecución de pruebas

Los archivos .feature ejecutarán el código de las clases de pasos para cada caso de prueba. Estas clases sirven como un puente entre los escenarios de prueba y la implementación técnica que interactúa con la aplicación. Los métodos en estas clases inicializan el entorno de pruebas, aseguran que comience en un estado controlado, y validan si el sistema se comporta según lo esperado, generando reportes de fallas si es necesario. Cada paso en un archivo .feature es implementado por un método en la clase de pasos, que realiza acciones como navegar, llenar formularios, o interactuar con la interfaz. Al ejecutar los archivos .feature en Cucumber, se muestran los resultados de las pruebas en la terminal, indicando los escenarios y pasos exitosos, junto con el tiempo de ejecución. Además, las clases de pasos manejan casos especiales, garantizando la estabilidad de la aplicación frente a entradas inválidas o situaciones excepcionales.

Figura 11

*Terminal de ejecución del archivo inicioSesion.feature*



```
✓ Done: Scenarios 5 of 5 (1sec 906ms) ⚠
"C:\Program Files\Java\jdk-22\bin\java.exe" ...
Testing started at 19:53 ...
La página principal está abierta.
El mensaje de error de inicio de sesión está visible.
El mensaje de error de inicio de sesión está visible.
El mensaje de error de inicio de sesión está visible.
El mensaje de error de inicio de sesión está visible.

5 Scenarios (5 passed)
15 Steps (15 passed)
0m0,932s

Process finished with exit code 0
```

Figura 12

*Terminal de ejecución del archivo añadirProductos.feature*

```
Product data updated successfully!
El producto se ha registrado correctamente
Ingresa un producto inválido
Product data updated successfully!
Product data updated successfully!
El producto se ha registrado correctamente
no regreso a la ventana anterior

3 Scenarios (3 passed)
9 Steps (9 passed)
0m5,906s

Process finished with exit code 0
```

Figura 13

*Terminal de ejecución del archivo editarProductos.feature*

```
Producto editado correctamente
Producto editado correctamente
boton ejecutado correctamente

3 Scenarios (3 passed)
9 Steps (9 passed)
0m8,600s

Process finished with exit code 0
```

Figura 14

*Terminal de ejecución del archivo eliminarProductos.feature*

```
Producto Eliminado Correctamente
Producto Eliminado Correctamente

3 Scenarios (3 passed)
9 Steps (9 passed)
0m5,359s

Process finished with exit code 0
```

Figura 15

*Terminal de ejecución del archivo universoDelFomix.feature*

```
4 Scenarios (4 passed)
12 Steps (12 passed)
0m13,920s

Process finished with exit code 0
```

## 6. Resultados obtenidos

Los resultados obtenidos de las pruebas automatizadas se documentaron para cada archivo .feature ejecutado en el proyecto. A través de la ejecución de estos escenarios en Cucumber, se verificó el comportamiento del sistema en diversas situaciones, asegurando que las funcionalidades respondan conforme a las especificaciones. A continuación, se presenta un resumen de los resultados obtenidos, con capturas de pantalla que ilustran el desempeño y las áreas que requieren mejoras en cada caso.

A continuación, se visualizará imágenes donde consta los escenarios tanto el Passed como el Failed de los requisitos hechos con la herramienta “Cucumber”:

Figura 15

<b>Feature: inicioSesion: 15 total, 15 passed</b>		787 ms
		<a href="#">Collapse</a>   <a href="#">Expand</a>
<b>Cucumber</b>		787 ms
<b>Inicio de sesión</b>		787 ms
<b>Iniciar sesión exitosamente</b>		506 ms
que el usuario se encuentre en la página de inicio de sesión	passed	441 ms
ingresa las credenciales de acceso correctas	passed	65 ms
debería ver la página principal	passed	0 ms
<b>Intentar iniciar sesión con credenciales incorrectas</b>		94 ms
que el usuario se encuentre en la página de inicio de sesión	passed	62 ms
ingresa las credenciales de acceso incorrectas	passed	31 ms
debería ver un mensaje de error de inicio de sesión	passed	1 ms
<b>Intentar iniciar sesión con un usuario correcto pero contraseña incorrecta</b>		52 ms
que el usuario se encuentre en la página de inicio de sesión	passed	33 ms
ingresa un nombre de usuario correcto y una contraseña incorrecta	passed	19 ms
debería ver un mensaje de error de inicio de sesión	passed	0 ms
<b>Intentar iniciar sesión con una contraseña correcta pero un usuario incorrecto</b>		65 ms
que el usuario se encuentre en la página de inicio de sesión	passed	40 ms
ingresa un nombre de usuario incorrecto y una contraseña correcta	passed	25 ms
debería ver un mensaje de error de inicio de sesión	passed	0 ms
<b>Intentar iniciar sesión dejando los campos en blanco</b>		70 ms
que el usuario se encuentre en la página de inicio de sesión	passed	48 ms



Figura 16

<b>Feature: inicioSesion:</b>		780 ms
<b>15 total, 4 failed, 11 passed</b>		
		<a href="#">Collapse</a>   <a href="#">Expand</a>
■	<b>Inicio de sesión</b>	780 ms
■	<b>Iniciar sesión exitosamente</b>	498 ms
	que el usuario se encuentre en la página de inicio de sesión	passed 428 ms
	ingresa las credenciales de acceso correctas	passed 70 ms
	debería ver la página principal	passed 0 ms
■	<b>Intentar iniciar sesión con credenciales incorrectas</b>	66 ms
	que el usuario se encuentre en la página de inicio de sesión	passed 46 ms
	ingresa las credenciales de acceso incorrectas	passed 20 ms
	debería ver un mensaje de error de inicio de sesión	failed 0 ms
■	<b>Intentar iniciar sesión con un usuario correcto pero contraseña incorrecta</b>	64 ms
	que el usuario se encuentre en la página de inicio de sesión	passed 34 ms
	ingresa un nombre de usuario correcto y una contraseña incorrecta	passed 30 ms
	debería ver un mensaje de error de inicio de sesión	failed 0 ms
■	<b>Intentar iniciar sesión con una contraseña correcta pero un usuario incorrecto</b>	82 ms
	que el usuario se encuentre en la página de inicio de sesión	passed 58 ms
	ingresa un nombre de usuario incorrecto y una contraseña correcta	passed 24 ms
	debería ver un mensaje de error de inicio de sesión	failed 0 ms
■	<b>Intentar iniciar sesión dejando los campos en blanco</b>	70 ms
	que el usuario se encuentre en la página de inicio de sesión	passed 43 ms

Figura 17

<b>Feature: añadirProductos: 9 total, 9 passed</b>			6.45 s
			<a href="#">Collapse</a>   <a href="#">Expand</a>
	Cucumber		6.45 s
	Añadir Productos		6.45 s
	Añadir un producto nuevo exitosamente		3.23 s
	que el usuario se encuentre en la pagina de añadir productos	passed	837 ms
	ingresa un producto valido	passed	2.39 s
	deberá ingresar un producto nuevo	passed	1 ms
	Añadir un producto nuevo de manera incorrecta		3.08 s
	que el usuario se encuentre en la pagina de añadir productos	passed	113 ms
	ingresa un producto invalido	passed	1.98 s
	debera mostrar un mensaje de error	passed	988 ms
	Desea regresar a una ventana anterior		137 ms
	que el usuario se encuentre en la pagina de añadir productos	passed	88 ms
	Presiona el boton de regreso	passed	49 ms
	debera regresar a una ventana anterior	passed	0 ms

Figura 18

<b>Feature: añadirProductos:</b>		9.44 s
<b>9 total, 1 failed, 1 ignored, 7 passed</b>		
		<a href="#">Collapse</a>   <a href="#">Expand</a>
■	<b>Añadir Productos</b>	9.44 s
■	<b>Añadir un producto nuevo exitosamente</b>	9.28 s
	que el usuario se encuentre en la pagina de añadir productos	passed 675 ms
	ingresa un producto valido	passed 8.61 s
	deberá ingresar un producto nuevo	passed 1 ms
■	<b>Añadir un producto nuevo de manera incorrecta</b>	51 ms
	que el usuario se encuentre en la pagina de añadir productos	passed 46 ms
	ingresa un producto invalido	failed 5 ms
	debera mostrar un mensaje de error	ignored 0 ms
■	<b>Desea regresar a una ventana anterior</b>	109 ms
	que el usuario se encuentre en la pagina de añadir productos	passed 67 ms
	Presiona el boton de regreso	passed 42 ms
	debera regresar a una ventana anterior	passed 0 ms

Figura 19

Feature: editarProductos: 9 total, 9 passed			7.58 s
			<a href="#">Collapse</a>   <a href="#">Expand</a>
Cucumber			7.58 s
Editar Productos			7.58 s
Producto Editado de manera correcta			4.46 s
que el usuario se encuentre en la pagina de EditarProductos	passed		2.55 s
presione el boton editar	passed		1.92 s
el producto se debera editar	passed		0 ms
Fallo al Editar un producto			1.94 s
que el usuario se encuentre en la pagina de EditarProductos	passed		1.06 s
presione el boton editar	passed		876 ms
el producto no se editara	passed		1 ms
Regresar			1.18 s
que el usuario se encuentre en la pagina de EditarProductos	passed		1.16 s
presione el boton de regresar	passed		0 ms
regresara a la pagina principal	passed		22 ms

Figura 20

**Feature: editarProductos:** 8.25 s  
**9 total, 1 failed, 1 ignored, 7 passed**

[Collapse](#) | [Expand](#)

■	<b>Editar Productos</b>	8.25 s
■	<b>Producto Editado de manera correcta</b>	4.40 s
	que el usuario se encuentre en la pagina de EditarProductos	passed 2.58 s
	presione el boton editar	passed 1.82 s
	el producto se debera editar	passed 0 ms
■	<b>Fallo al Editar un producto</b>	2.51 s
	que el usuario se encuentre en la pagina de EditarProductos	passed 1.24 s
	presione el boton editar	passed 1.27 s
	el producto no se editara	passed 0 ms
■	<b>Regresar</b>	1.34 s
	que el usuario se encuentre en la pagina de EditarProductos	passed 1.34 s
	presione el boton de regresar	failed 0 ms
	regresara a la pagina principal	ignored 1 ms

Figura 21

Feature: eliminarProductos: 9 total, 9 passed			5.47 s
			<a href="#">Collapse</a>   <a href="#">Expand</a>
Cucumber			5.47 s
■ Eliminar Productos			5.47 s
■ Producto Eliminado de manera correcta			2.45 s
	que el usuario se encuentre en la pagina de eliminar productos	passed	2.45 s
	el usuario borre un producto	passed	0 ms
	el producto se debera eliminar	passed	0 ms
■ Producto Eliminado de manera incorrecta			1.55 s
	que el usuario se encuentre en la pagina de eliminar productos	passed	1.55 s
	el usuario borre un producto	passed	0 ms
	el producto no se eliminara	passed	0 ms
■ El boton funciona de manera correcta			1.48 s
	que el usuario se encuentre en la pagina de eliminar productos	passed	1.46 s
	el usuario presione el boton regresar	passed	1 ms
	la ventana aparecera	passed	20 ms

Figura 22

## Feature: eliminarProductos: 9 total, 1 failed, 8 passed

8.78 s

[Collapse](#) | [Expand](#)

■	■	Eliminar Productos	8.78 s
	■	Producto Eliminado de manera correcta	3.49 s
		que el usuario se encuentre en la pagina de elimnar productos	passed 3.49 s
		el usuario borre un producto	passed 0 ms
		el producto se debera eliminar	passed 0 ms
	■	Producto Eliminado de manera incorrecta	2.58 s
		que el usuario se encuentre en la pagina de elimnar productos	passed 2.58 s
		el usuario borre un producto	passed 1 ms
		el producto no se eliminara	failed 0 ms
	■	El boton funciona de manera correcta	2.71 s
		que el usuario se encuentre en la pagina de elimnar productos	passed 2.69 s
		el usuario presione el boton regresar	passed 0 ms
		la ventana aparecera	passed 14 ms

Figura 23

## Feature: universoDelFomix: 12 total, 12 passed 15.90 s

[Collapse](#) | [Expand](#)

Cucumber	15.90 s
<ul style="list-style-type: none"> <li>universoDelFomix           <ul style="list-style-type: none"> <li>funcionalidad boton ver Inventario               <ul style="list-style-type: none"> <li>que se encuentre en la ventana principal passed 4.93 s</li> <li>presione el boton Ver Inventario passed 1 ms</li> <li>se diriga a esa ventan passed 14 ms</li> </ul> </li> <li>funcionalidad boton aadirProducto               <ul style="list-style-type: none"> <li>que se encuentre en la ventana principal passed 3.62 s</li> <li>presione el boton aadirProducto passed 0 ms</li> <li>se dirige a la venta aadirProductos passed 9 ms</li> </ul> </li> <li>funcionalidad boton eliminarProducto               <ul style="list-style-type: none"> <li>que se encuentre en la ventana principal passed 3.68 s</li> <li>presione el boton eliminarProducto passed 10 ms</li> <li>se diriga a esa ventana eliminarProducto passed 1 ms</li> </ul> </li> <li>funcionalidad boton editarProducto               <ul style="list-style-type: none"> <li>que se encuentre en la ventana principal passed 3.62 s</li> <li>presione el boton editarProducto passed 1 ms</li> <li>se diriga a esa ventana editarProducto passed 8 ms</li> </ul> </li> </ul> </li> </ul>	15.90 s



Figura 24

## Feature: universoDelFomix: 12 total, 1 failed, 11 passed

14.22 s

[Collapse](#) | [Expand](#)

■	universoDelFomix	14.22 s
■	funcionalidad boton ver Inventario	4.64 s
■	que se encuentre en la ventana principal	passed 4.64 s
■	presione el boton Ver Inventario	passed 0 ms
■	se diriga a esa ventan	failed 0 ms
■	funcionalidad boton aadirProducto	3.29 s
■	que se encuentre en la ventana principal	passed 3.28 s
■	presione el boton aadirProducto	passed 0 ms
■	se dirige a la venta aadirProductos	passed 10 ms
■	funcionalidad boton eliminarProducto	3.02 s
■	que se encuentre en la ventana principal	passed 3.01 s
■	presione el boton eliminarProducto	passed 6 ms
■	se diriga a esa ventana eliminarProducto	passed 2 ms
■	funcionalidad boton editarProducto	3.27 s
■	que se encuentre en la ventana principal	passed 3.26 s
■	presione el boton editarProducto	passed 0 ms
■	se diriga a esa ventana editarProducto	passed 10 ms

## **7. Conclusiones**

1. Este proceso de pruebas nos demostró que, cuando seguimos una buena metodología y usamos las herramientas adecuadas, podemos asegurarnos de que el sistema funcione como se espera ya que todo fue como un rompecabezas que armamos poco a poco hasta conseguir un resultado eficiente.
2. Las pruebas automatizadas no solo ayudaron a validar las funciones del sistema, sino que también nos permitieron identificar y corregir errores que de otra manera podrían haber pasado desapercibidos, fue como tener un control de calidad constante durante el desarrollo de este software.
3. La herramienta llamada “Cucumber” fue clave para realizar las pruebas de manera efectiva porque nos permitió simular diferentes escenarios y asegurarnos de que el sistema responde bien en cada uno de ellos, lo que hace que el software sea más robusto y confiable.

## **8. Recomendaciones**

1. Aunque el sistema pasó las pruebas, es importante seguir buscando formas de optimizarlo y no conformarnos con que funcione bien sino también buscar cómo mejorarlo más en futuras versiones o actualizaciones.
2. En el desarrollo de este software surgieron muchas ideas de agregar más y más funciones al sistema, pero lo más importante es asegurarse de que cada nueva función sea realmente útil y que el sistema siga siendo fácil de usar para los q van a utilizarlo a diario.
3. Es importante planificar el mantenimiento continuo y las actualizaciones para que el sistema siga siendo confiable a largo plazo y también hay que estar atentos a posibles problemas y estar listos para solucionarlos rápidamente cuando surjan.