**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 Ingenieria 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 el prototipo de 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.

1. **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 todo 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

1. **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

Descripción generada automáticamenteTabla 1 Ingreso al sistema | Login

Tabla 2 Añadir productos | AddProducts verify the ID

Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamenteTabla 3 Editar Productor | EditProducts verify the ID

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.

1. **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:*

*Texto

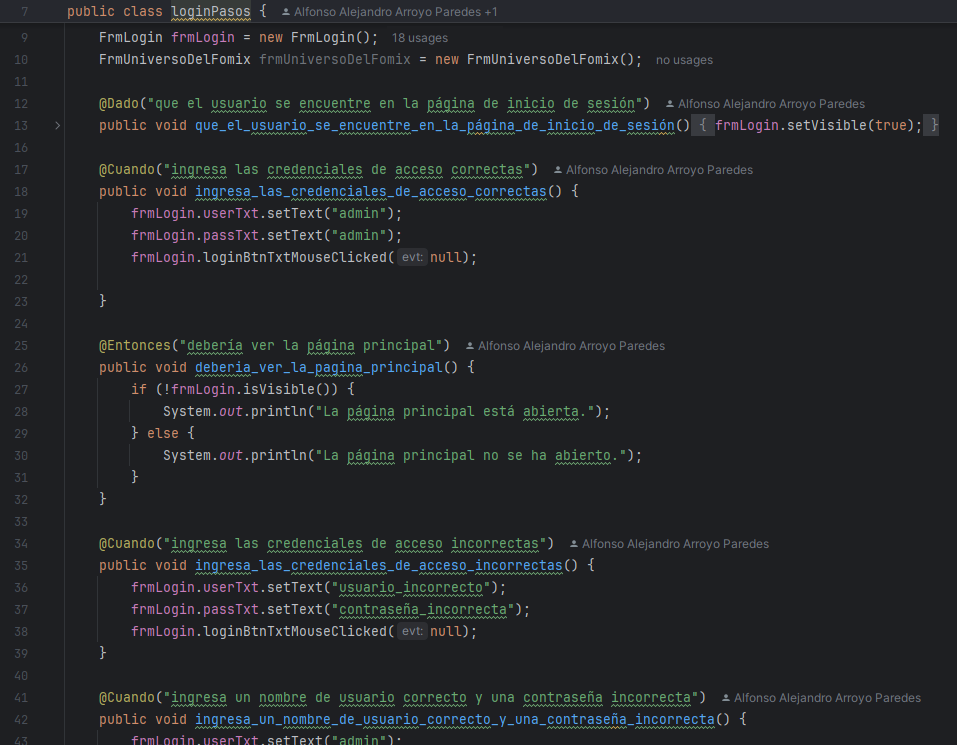
Descripción generada automáticamente*

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*



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

Texto

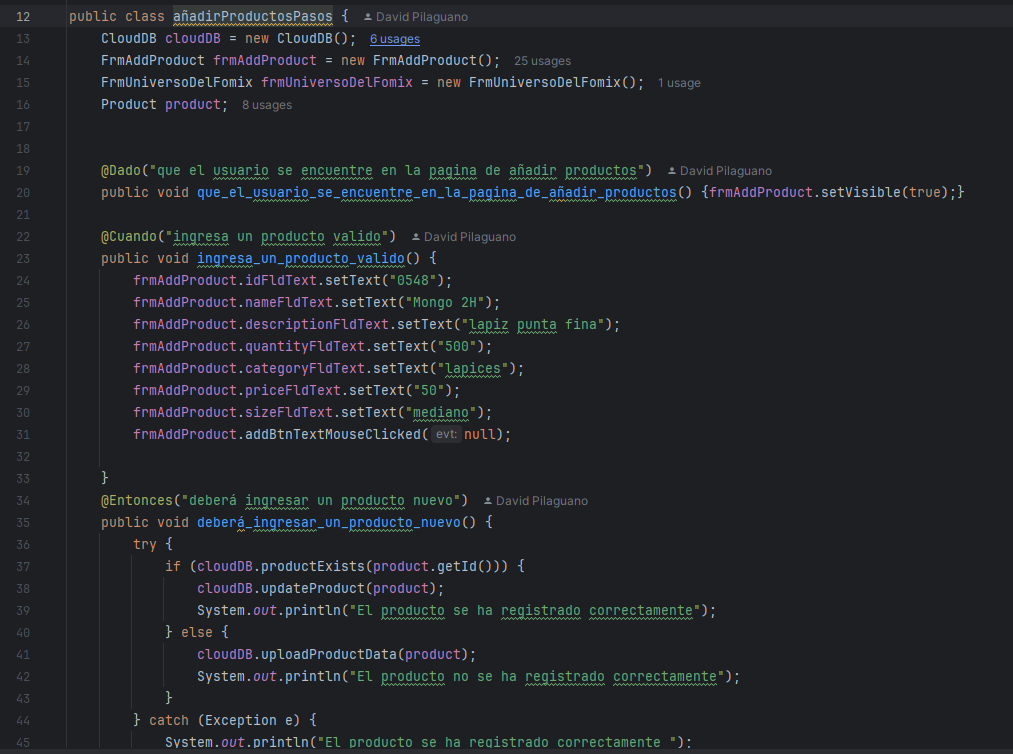
Descripción generada automáticamente*Archivo añadirProductos.feature:*

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*

**

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.

eliminar

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

editar

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Universo del fomixTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

1. **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*

*Captura de pantalla de un celular

Descripción generada automáticamente*

Figura 12

*Terminal de ejecución del archivo añadirProductos.feature*

*Texto

Descripción generada automáticamente*

Figura 13

*Terminal de ejecución del archivo editarProductos.feature*

Texto

Descripción generada automáticamente

Figura 14

*Terminal de ejecución del archivo eliminarProductos.feature*

Texto

Descripción generada automáticamente

Universo del fomix, botón inventario  
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. **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



1. **Conclusiones**

-a

1. **Recomendaciones**

-a