

INFORME DE PRUEBAS PARA EL CASO DE USO: INICIAR SESIÓN

Barrionuevo Lindsay, Rivera Joel & Yaranga Leonardo

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

14568: Ingeniería de Requisitos

Ing. Jenny Alexandra Ruíz Robalino

26 de mayo de 2024

DESARROLLO DEL PROCESO

1. Introducción

En el proceso de Ingeniería de Requisitos, se busca cubrir las necesidades del usuario final en el proyecto desarrollado, en nuestro contexto específico, se busca automatizar el proceso de ingreso de pagos de la Escuela Politécnica Nacional para generar un reporte mensual de pagos, que será manejado por la Licenciada Patricia López como Tesorera de Urbanización en la gestión de pagos, además de residentes para su revisión. Con ello, se pretende dar una separación de perfiles de usuario para brindar transparencia en que cualquier operación será realizada por la tesorera.

En este contexto, se desarrolló un prototipo de fidelidad media de nuestro primer requisito funcional, el Inicio de Sesión, donde se busca validar los perfiles según las credenciales de la tesorería o de un residente de la urbanización, además de credenciales erróneas. En este informe se mostrará el método por el que se realizaron las pruebas, la tecnología utilizada y los resultados obtenidos.

2. Objetivos

2.1.Objetivo General

Desarrollar pruebas de aceptación en el prototipo del Caso de Uso de Iniciar Sesión mediante las herramientas de Selenium y Cucumber para validar su correcto funcionamiento en Reglas aprobadas o fallidas.

2.2.Objetivos Específicos

- Establecer el entorno de desarrollo necesario para el desarrollo de pruebas automatizadas mediante la configuración del lenguaje de programación Python en el prototipo de Página de Inicio de Sesión.
- Escribir scripts en Python utilizando la herramienta de Selenium para simular la interacción con el usuario en la página de inicio de sesión, verificando la aceptación de credenciales correctas y erróneas.
- Implementar criterios de éxito y de fracaso para las pruebas en el prototipo, verificando la respectiva redirección al perfil de tesorera o residente, y su presentación de mensajes de error apropiados en caso de obtener un failed.
- Recopilar los resultados de las pruebas automatizadas en un archivo de formato pdf, para analizar los resultados y obtener posibles mejoras.

3. Entradas a Utilizar

Al ser nuestro Caso de Uso una Página de Inicio de Sesión, existen dos campos de ingreso de datos, el Username o Usuario, y la Password o Contraseña. En las Tablas 1 y 2 se visualizan las entradas a utilizar, con sus respectivos estados esperados al realizar las pruebas.

Tabla 1

Entradas a Utilizar para las pruebas del perfil de Tesorera/Admin

Variable	Equivalencia	Entrada	Estado
Usuario Tesorera	usuarioU=user	Patricia123	Válido
	usuarioU!=user	OOOOOOOOO	Failed
Contraseña Tesorera	passwordU=password	Patricia123	Válido
	passwordU!=password	Patricia123	Failed

Tabla 2

Entradas a Utilizar para las pruebas del perfil de Residente

Variable	Equivalencia	Estado	Entrada
Usuario Residente	usuarioU=user	Usuario123	Válido
	usuarioU!=user	User123	Failed
	usuarioU!=user	aaaaa	Failed
Contraseña Residente	passwordU=password	Usuario123	Válido
	passwordU!=password	User12w4	Failed
	passwordU!=password	Usuario12356	Failed

4. Ambiente de Desarrollo de Pruebas

Para el desarrollo de pruebas, se utilizó la herramienta Python con el IDE PyCharm, en el mismo se creó un ambiente virtual en el cual se creó diferentes archivos con el propósito de desarrollar las pruebas, esto se lo realiza mediante las librerías “behave” y “selenium”, sin embargo, la implementación de los Features, se lo realizó gracias a Cucumber.

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

- Archivo pruebaLogin.feature: Dentro del mismo se encuentran los casos a evaluar dentro del caso de uso, en el mismo podemos encontrar, los escenarios, y las preguntas frecuentes que se realiza en cada caso.

Figura 1

Archivo pruebaLogin.feature

```
Feature: login funcionalidad para la aplicacion UrbTreasurySystem
  Scenario: Validar nombre de usuario y contraseña de administrador
    Given abrir navegador
    When proporcionando un nombre de usuario y contraseña válidos
    Then Verificando pagina
```

En la anterior figura se puede observar un ejemplo del escenario de ingreso de sesión del administrador, para ello, se determinó el escenario de validar el usuario y contraseña, junto con los objetivos de Obtener, Cuando?, y por último Después, para así determinar las acciones que se realizan dentro de la prueba respectiva. El objetivo de las pruebas es realizarlas dentro de un mismo requisito funcional, por eso se tienen otras 4 pruebas más, de las cuales 1 es del residente y las otras el objetivo de determinar si las validaciones están en estado correcto.

- Archivo environment.py: Dentro de este archivo se encuentra toda la configuración del test, en el mismo encontramos secciones donde se da el formato al pdf, y a su vez se le analiza y determina el texto que contendrá el mismo. Por último, también, en este archivo se crea como tal el pdf y se almacena en los reportes.

- Archivo login_method.py: Este archivo permite como tal, crear las funciones de cada escenario que se determinó anteriormente, para comprender de mejor manera, se utilizara el mismo escenario:

Figura 2

Archivo login_method.py

```
@given('abrir navegador')
def open_browser(context):
    ensure_pdf_initialized(context)
    setup_browser(context)
    context.driver.get('http://localhost/Codigo/index.html')
    screenshot_path = take_screenshot(context, 'open_browser')
    add_screenshot_to_pdf(context.pdf, screenshot_path, "Abrir navegador")

@when('proporcionando un nombre de usuario y contraseña válidos')
def validUserNameAndPassword(context):
    ensure_pdf_initialized(context)
    context.driver.find_element(By.NAME, 'usuario').send_keys('Patricia123')
    context.driver.find_element(By.NAME, 'password').send_keys('Patricia123')
    screenshot_path = take_screenshot(context, 'validUserNameAndPassword')
    context.driver.find_element(By.NAME, 'submit').click()
    add_screenshot_to_pdf(context.pdf, screenshot_path, "Proporcionar credenciales válidas")

@then('Verificando pagina')
def verificacion_AdminHome(context):
    ensure_pdf_initialized(context)
    WebDriverWait(context.driver, 10).until(EC.title_is("Admin"))
    assert context.driver.title == "Admin"
    screenshot_path = take_screenshot(context, 'verificacion_AdminHome')
    add_screenshot_to_pdf(context.pdf, screenshot_path, "Verificación de la página de administrador")
```

Dentro de la figura anterior se muestra cómo se crearon las funciones en base a lo anteriormente determinado en el escenario, para acceder a dichas acciones de este, se usa las librerías anteriormente mencionadas, para así, determinar que función se ejecuta dependiendo el caso.

- Función open_Browser: Como su nombre indica, es aquella función que se encarga de abrir el navegador y colocar el URL determinado para el requisito funcional.

- Función `validUserNameAndPassword`: Esta función se encarga directamente de hacer la prueba como tal, es decir, se encarga tanto de ingresar los datos en el apartado, como también de enviar los mismos.
- Función `verificacion_AdminHome`: Su funcionamiento principal, es el de validar, si el resultado que se espera es el correcto, teniendo así un resultado correcto o fallido.

Estas funciones se repiten dependiendo del escenario, o sea, el mismo varia ligeramente, pero la lógica es la misma. Cabe recalcar que, mientras se realizan la ejecución de las funciones, se van tomando las capturas necesarias para los reportes en formato pdf.

5. Ejecución de Pruebas

Una vez creado el ambiente de ejecución con Selenium y su configuración para automatizar las pruebas mediante el Lenguaje de Programación Python, se realizan la ejecución de las pruebas automatizadas con los Scripts previamente mencionados. Para ello, gracias a la Ayuda de Selenium, dichas pruebas son capturadas a maneras de imagen, para posteriormente, añadirlas a un archivo en formato pdf.

Es importante mencionar que, para la ejecución de este código, nos ubicamos en el IDE PyCharm que es ambientado para la configuración del lenguaje de programación Python, y con el comando Behave podemos ejecutar las pruebas automatizadas.

6. Resultados Obtenidos

Una vez que se han realizado las pruebas, hemos obtenido cinco reportes en formato pdf de las Pruebas de Aceptación realizadas, a continuación, se adjuntarán dichos pdfs para pasar a su explicación específica.

Prueba 1 – Usuario y Contraseña Correcta del Administrador – Passed

Para esta Prueba, se han utilizado las credenciales correctas de la Tesorera/Administrador, para saber ello, nos podemos ubicar en la [Tabla 1](#), las credenciales de Estado Válido, aquí podemos apreciar cómo al ser validado correctamente, se redirecciona a una página que tiene un título de “Admin”, obteniendo como conclusión que: “El escenario ha pasado todas las pruebas exitosamente”.

Prueba 2 – Usuario y Contraseña Correcta del Residente

Para esta Prueba, se han utilizado las credenciales correctas del Residente, para saber ello, nos podemos ubicar en la [Tabla 2](#), las credenciales de Estado Válido, aquí podemos apreciar cómo al ser validado correctamente, se redirecciona a una página que tiene un título de “Residente”, obteniendo como conclusión que: “El escenario ha pasado todas las pruebas exitosamente”.

Prueba 3 – Usuario y Contraseña incorrectas Administrador/Residente

Para esta Prueba, se han utilizado las credenciales incorrectas tanto para el residente como para el perfil de administrador, teniendo usuario y contraseña incorrectos para saber ello, nos podemos ubicar en la [Tabla 2](#), las credenciales de Estado Failed, aquí podemos apreciar cómo las credenciales son borradas de los campos y nos envía un mensaje de error, obteniendo como conclusión que: “El testeo se realizó correctamente. Sin embargo, el ingreso a la página de Administrador o Residente ha fallado debido a credenciales incorrectas, lo cual es el resultado esperado”.

Prueba 4 - Contraseña incorrecta para el usuario Residente

Para esta Prueba, se han utilizado la credencial correcta en el usuario, pero incorrecta en la contraseña, teniendo usuario y contraseña incorrectos para saber ello, nos podemos ubicar en la [Tabla 2](#), las credenciales de Estado Failed en contraseña, pero válido en usuario, aquí podemos apreciar cómo las credenciales son borradas de los campos y nos envía un mensaje de error, obteniendo como conclusión que: “El testeó se realizó correctamente. Sin embargo, el ingreso a la página de Administrador o Residente ha fallado debido a credenciales incorrectas, lo cual es el resultado esperado”.

Prueba 5 – Usuario incorrecto para el usuario Administrador/Tesorerera

Para esta Prueba, se han utilizado la credencial correcta en la contraseña, pero incorrecta en el usuario, teniendo usuario y contraseña incorrectos para saber ello, nos podemos ubicar en la [Tabla 1](#), las credenciales de Estado Failed en usuario, pero válido en contraseña, aquí podemos apreciar cómo las credenciales son borradas de los campos y nos envía un mensaje de error, obteniendo como conclusión que: “El testeó se realizó correctamente. Sin embargo, el ingreso a la página de Administrador o Residente ha fallado debido a credenciales incorrectas, lo cual es el resultado esperado”.

7. Conclusiones

- Estas pruebas realizadas con Selenium y Cucumber nos confirmaron que el prototipo de inicio de sesión valida correctamente tanto las credenciales válidas como las erróneas para los perfiles de Administrador y Residente. Las pruebas confirmaron que el sistema responde adecuadamente a cada tipo de entrada.
- El entorno de desarrollo configurado en Python con PyCharm, junto con las librerías "behave" y "selenium", nos resultó una herramienta de mucha utilidad a la hora de realizar pruebas. La integración con Cucumber nos permitió visualizar una estructura

clara de los escenarios de prueba, facilitando la comprensión y mantenimiento del código de prueba.

- La funcionalidad de generación de reportes en formato PDF que fue implementada, funcionó correctamente, proporcionándonos un documento claro que incluye los resultados y las capturas de pantalla relevantes. Esto facilita la revisión y el análisis de los resultados de las pruebas.

8. Recomendaciones

- Recomendamos ampliar los casos de prueba para incluir escenarios adicionales que puedan surgir en situaciones reales. Por ejemplo, deberíamos considerar pruebas de inactividad del usuario, sesiones expiradas, y la validación de formatos especiales para usuarios y contraseñas. Estos casos adicionales nos permitirán cubrir un rango más amplio de posibles problemas y asegurar que el sistema maneje correctamente todas las situaciones posibles.
- Además, dado que las aplicaciones y sistemas pueden evolucionar con el tiempo, es fundamental que mantengamos nuestro entorno de pruebas actualizado. Debemos monitorear continuamente los cambios en el sistema que podrían afectar el inicio de sesión y actualizar nuestros scripts de prueba en consecuencia para garantizar que sigan siendo relevantes y efectivos.
- Finalmente, sería útil considerar la inclusión de más aspectos funcionales en nuestro proceso de pruebas automatizadas. Por ejemplo, podríamos realizar pruebas de integración con otros módulos del sistema que puedan influir en el inicio de sesión o el manejo de usuarios. Esto nos permitirá tener una visión más completa del comportamiento del sistema.