

RETO 1

Casos de prueba

Casos de prueba funcionales:

- Verificar el registro exitoso con correo y contraseña válidos.
- Verificar el login exitoso con correo y contraseña válidos.
- Validar que se muestren mensajes de error para correo y contraseña inválidos (por ejemplo, sin "@" en el correo).
- Validar que se muestren mensajes de error si falta alguno de los campos obligatorios.

Casos de prueba de validación de entradas:

- Ingresar una contraseña menor a 5 caracteres y verificar la aparición de un mensaje de error.
- Ingresar un correo sin "@" y validar que se rechaza.

Casos de prueba de regresión:

- Asegurar que las funcionalidades actuales no se vean afectadas por la implementación del nuevo proceso de autenticación.

Tipo de prueba: Funcional.

Descripción: Validar que el sistema permita el registro y login con un correo y contraseña válidos.

Entrada: Correo con "@" y contraseña con más de 5 caracteres.

Resultado esperado: Registro/login exitoso.

Resultado real: [Aquí se documenta después de la ejecución].

Agregar más casos de prueba según el tipo funcional, validación, y regresión.

Técnica de diseño de casos de prueba

Para generar los casos de prueba, usaría las siguientes técnicas:

- **Técnica de partición de equivalencia:** Esta técnica ayuda a agrupar entradas que deberían ser tratadas de manera similar (por ejemplo, emails con "@" vs. sin "@", contraseñas mayores a 5 caracteres vs. menores).
- **Técnica de análisis de valor límite:** Útil para validar las contraseñas justo en el límite (por ejemplo, una contraseña de 4 caracteres vs. una de 5).
- **Técnica de casos de uso:** Diseñar casos de prueba basados en las historias de usuario presentadas.

Explicación de por qué se utilizó partición de equivalencia y análisis de valor límite.

Bugs encontrados

Durante la ejecución de las pruebas, se podrían identificar bugs como:

- **Error en la validación del correo:** Si el sistema acepta correos sin "@".
- **Problema de usabilidad:** Si los mensajes de error no son claros para el usuario.

- **Validación inconsistente:** Si la validación de la contraseña es inconsistente (por ejemplo, si a veces acepta menos de 5 caracteres).

Bug #1: El correo sin "@" es aceptado (si se encuentra este bug).

Descripción: El sistema debería rechazar correos no válidos.

Prioridad: Alta.

Sugerencia de mejora: Añadir validaciones adicionales en el backend.

Pruebas automatizadas

Podrías implementar diferentes tipos de pruebas:

- **Pruebas unitarias:** Para probar la lógica de validación (validación de correo y contraseña).
- **Pruebas de servicios (API testing):** Si el registro y login se manejan a través de APIs, puedes usar herramientas como Postman o frameworks como RestAssured para asegurarte de que las respuestas sean las correctas.
- **Pruebas de UI (Interfaz de Usuario):** Automatizadas usando Selenium, donde puedas simular el registro y login en la interfaz web y verificar los comportamientos esperados.
- **Pruebas de rendimiento (performance testing):** Para asegurarte de que el sistema responde de manera adecuada bajo carga. Herramientas como JMeter podrían ser útiles aquí.

Tipos de pruebas realizadas: Unitarias, UI, Servicios.

Framework utilizado: Selenium con Python para UI, JUnit/TestNG para unitarias, Postman para API testing.

Resultados: Describir los resultados obtenidos después de la ejecución automática.

Estructura del proyecto de automatización

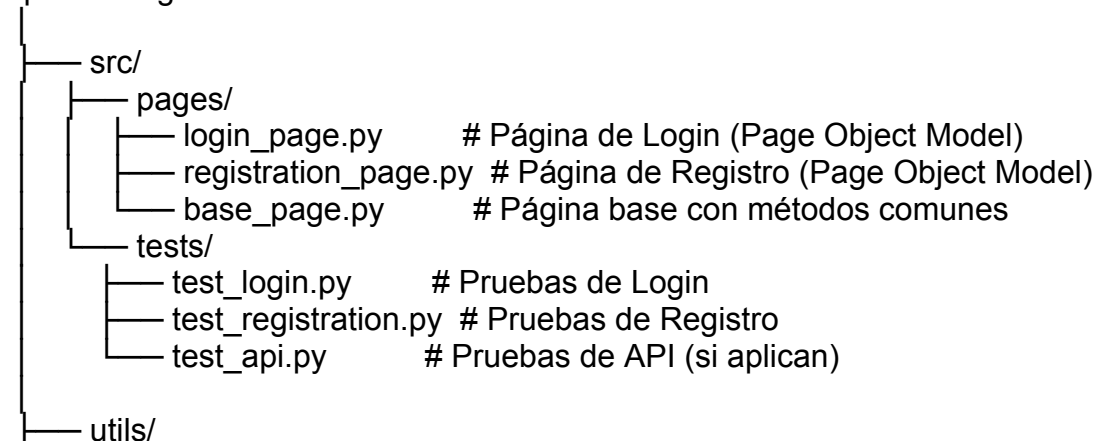
Proyecto: qaChallenge

Lenguaje recomendado: Python

Framework: Selenium para UI, JUnit/TestNG para pruebas unitarias, RestAssured para API testing.

Posible Estructura de carpetas:

qaChallenge/



```
|
|   └─ helper.py           # Utilidades o funciones comunes
|   └─ config.py          # Configuraciones (URLs, credenciales, etc.)
|
└─ requirements.txt        # Lista de dependencias del proyecto
(Selenium, PyTest, etc.)
└─ pytest.ini             # Configuración de pytest
└─ README.md              # Descripción del proyecto y cómo ejecutar
las pruebas
```

Conclusión

Resumen de los hallazgos y la efectividad de las pruebas automatizadas.