

Autores - Versión del plan de pruebas

Versión	Fecha	Descripción del cambio	Autor	Aprobó
1.0	30/06/2025	Versión inicial	Bárbara R. Urbano Moreno	-

Introducción del Plan de Prueba

El presente documento tiene como objetivo presentar la planificación de pruebas para “Prueba técnica de QA” detallando la estrategia de pruebas, el análisis de potenciales riesgos de seguridad, estimación de tiempos y recursos necesarios.

Objetivo del Plan de Prueba

El objetivo de este plan de pruebas es verificar el correcto funcionamiento del sistema de registro y autenticación de usuarios con especial énfasis en la seguridad y validación de los datos. Buscando asegurar que todas las funcionalidades trabajen correctamente e identificar posibles fallos en el sistema. Se incluyen pruebas automatizadas para validar el flujo completo de registro, login y borrado de usuario.

Descripción del producto de prueba técnica

El sistema a probar es una plataforma de gestión y registro de usuarios que incluye las siguientes funcionalidades:

- Registro de nuevos usuarios
- Login
- Borrado de usuario

ÍNDICE	Página
1 - ALCANCE DE LAS PRUEBAS	2
2 - ESTRATEGÍA DE PRUEBAS.....	2
3 - ANÁLISIS DE RIESGOS DE SEGURIDAD.....	3
4 - RECURSOS NECESARIOS.....	3
5 - CRONOGRAMA ESTIMADO.....	4

ALCANCE DE LAS PRUEBAS

- In scope
 - Registro de nuevos usuarios
 - Inicio de sesión
 - Validaciones de campos
 - Manejo de errores
 - Seguridad básica
 - Automatización del flujo completo de registro, login y borrado de usuario utilizando Playwright, en un entorno con Node js y TypeScript
- Out of scope
 - Autenticación con redes sociales (Google, Facebook, Microsoft, Apple, Teléfono)
 - Autenticación de doble factor (2FA)
 - Recuperación de contraseña
 - Almacenamiento y encriptación de contraseñas

ESTRATEGIA DE PRUEBAS

Tipos de pruebas:

- **Pruebas funcionales:** Verificar que registro y login funcionen correctamente
- **Pruebas de validación de formularios:** Campos vacíos, botones habilitados cuando los campos están vacíos.
- **Pruebas de seguridad básica:**
 - Comprobación de mensaje de 3 intentos de sesión fallidas, contraseñas seguras.

Método:

- **Pruebas manuales** con ejecución paso a paso
- **Casos de prueba** en Excel con resultados esperados
- Documento con la **ejecución de los casos de prueba** con los resultados obtenidos y **reporte de defectos**.
- **Automatización:**

Debido a que actualmente no poseo experiencia previa realizando test automatizados, en el cronograma que he realizado se ha dedicado un tiempo de estudio básico e introductorio de herramientas como (Playwright, Node js y TypeScript). Para una comprensión del flujo de automatización y buenas prácticas en estas.

- Se utilizará Playwright con Node js y TypeScript
- Los flujos automatizados cubrirán:
 - Registro de usuario con datos válidos
 - Inicio de sesión con credenciales válidas
 - Eliminación de cuenta desde el perfil del usuario
- Los scripts de prueba estarán almacenados en GitHub

ANÁLISIS DE RIESGOS DE SEGURIDAD

Riesgo identificado	Impacto	Caso de prueba asociado	Severidad	Mitigación
Visualización de contraseña en texto plano	Una persona podría visualizar la contraseña que se está ingresando	n7: Validación de caracteres ocultos en el campo de contraseña.	Alta	Se sugiere que los caracteres ingresados estén representados con símbolos como (● o *)
Posibles ataques de fuerza bruta por múltiples intentos de login fallidos	Una persona podría intentar ingresar muchas veces con distintas contraseñas hasta lograr acceso.	n8: Bloqueo temporal de cuenta tras múltiples intentos fallidos de login.	Media	Se sugiere implementar bloqueo temporal tras 3 intentos fallidos.
Registro de usuario con contraseñas débiles	Los usuarios podrían registrarse con contraseñas fáciles de adivinar.	n9: Visualización y validación de requisitos de contraseña segura en el formulario de registro.	Media	Se sugiere que se genere una política de contraseña segura por ejemplo:(Al menos 10 caracteres, combinación de mayúsculas, minúsculas, números y caracteres especiales)

RECURSOS NECESARIOS

- **Recursos humanos:**
 - QA tester trainee (Yo misma)

- **Tecnologías/herramientas:**

- Navegador web (Chrome)
- Google docs
- Hojas de cálculos de google
- Tiempo disponible: 3 días desde la recepción de la consigna de la prueba técnica.
- Node js, TypeScript
- Playwright para pruebas automatizadas
- IDE VS code
- GitHub para control de versiones y entrega de la prueba técnica
- Proyecto clonado desde el repositorio público de GitHub
- Comandos disponibles:
 - Instalación **npm install**
 - Para ejecutar los tests **npx playwright test**
 - Para ver el reporte de los resultados **npx playwright show-report**

CRONOGRAMA ESTIMADO

Día	Actividad
Día 1	<ul style="list-style-type: none"> ● Estudio del enunciado y definición del alcance ● Elaboración del Plan de Prueba ● Análisis de las funcionalidades del sistema (Registro, Login, Borrado) ● Identificación de validaciones importantes y riesgos de seguridad ● Estudio de automatización de pruebas <ul style="list-style-type: none"> ○ ¿Qué es Playwright? ¿Cómo funciona? ○ ¿Qué debo tener instalado en vs code? Instalación del entorno (Node js, TypeScript, Playwright)
Día 2	<ul style="list-style-type: none"> ● Diseño de casos de prueba manuales (registro, login, validaciones, manejo de errores, seguridad básica) ● Ejecución de pruebas manuales y documentación de resultados ● Implementación del test automatizado de registro de usuario
Día 3	<ul style="list-style-type: none"> ● Automatización del flujo completo: <ul style="list-style-type: none"> ○ Login válido ○ Borrado de usuario ● Validación del correcto funcionamiento de los scripts

- | | |
|--|---|
| | <ul style="list-style-type: none">• Registro de la ejecución de estos• Finalizar redacción del informe de casos de prueba, ejecución y resultados tanto manuales como automatizados. |
|--|---|