
PLAN DE TEST DEL SISTEMA

Superhéroes-topsecret

PREFACIO

Este documento describe el plan de test del sistema Angular superhéroes-topsecret, cuyo objetivo principal es realizar pruebas manuales y automatizadas al aplicativo.

Alcance Este documento de plan de test es la base de la fase de test del proyecto. Describe los siguientes tópicos: Plan de pruebas basado en pruebas manuales y automatizadas.

HISTORIA DEL DOCUMENTO

Fecha	Versión	Comentarios	Autor
20 octubre 2023	1.0	Versión inicial	Felipe Cruz
	1.1	Revisada por el equipo	

TABLA DE CONTENIDOS

1	INTRODUCCION	1
1.1	Propósito	1
1.2	Objetivos del Plan.....	1
1.3	Alcance del Testing	1
1.4	Criterios de Entrada.....	2
1.5	Criterios de Salida	2
2	Test del Sistema	3
2.1	Estrategia de Test del Sistema.....	3
2.2	Pruebas en Operación Normal	4
2.2.1	Método POST.....	
2.2.2	Método GET	
2.2.2	Método DELETE.....	
2.3	Pruebas en Condiciones de Excepción	4
2.3.1	Condición de Excepción 1	
2.4	Criterios de Éxito/Fracaso de Pruebas	4
2.5	Entregables	5
3	Configuración del Test.....	6
3.1	Hardware	6
3.2	Software	6
3.3	Otros.....	6
3.4	Ambiente	6
4	Tareas	7
4.1	Actividades	7
4.2	Responsabilidades	7
4.2.1	Responsabilidades del Grupo de Desarrollo	7
4.2.2	Responsabilidades del Grupo deTesting.....	7
4.2.3	Responsabilidades de la Gerencia	7
4.3	Planificación	8
	Glosario (Definiciones y Siglas)	9

1 INTRODUCCION

1.1 Propósito

El sistema desarrollado en Java es una API REST creada para la tercera semana del programa Eureka de 3IT. El proyecto es uno de índole ultrasecreto, el cual recopila información clasificada acerca de las organizaciones de superhéroes conocidas de la franquicia de Marvel. Presenta una base de datos que muestra la información sobre algunas organizaciones. ¡Esta información es de tipo clasificada!

1.2 Objetivos del Plan

El Plan de Test del Sistema especifica los procesos de test y de verificación que se realizarán con el objeto de:

- Identificar defectos y fallas.
- Medir rendimiento.
- Evaluar la calidad.
- Determinar el cumplimiento de los requerimientos.

Los objetivos de este plan son:

- Definir y detallar todas las tareas que se desarrollarán para probar el sistema.
- Definir el plan y la persona o grupo responsable de cada tarea.
- Definir las herramientas de prueba y el ambiente necesario a la conducción de las actividades de test.
- Definir los ítems y funcionalidades que serán probados.

1.3 Alcance del Testing

El Plan de Testing del Sistema es una especificación de alto nivel de los requerimientos funcionales y de calidad que serán probados, del ambiente de testing, de la estrategia de testing, de las responsabilidades y de los criterios de éxito.

El comportamiento de un producto bajo testing será comparado con las especificaciones de los requerimientos que fueron usados para implementar el sistema, incluyendo todos los cambios que han sido aprobados e implementados.

Los casos de prueba y los criterios de éxito serán derivados de este plan general y serán especificados en el documento de Especificaciones de Testing del Sistema.

El alcance del test del sistema es probar la funcionalidad completa y el rendimiento del aplicativo superhéroes-topsecret.

1.4 Criterios de Entrada

Para poder comenzar la fase de pruebas del sistema, se deben cumplir los siguientes criterios:

- Sistema completamente integrado de acuerdo con lo solicitado para la entrega 3 del programa Eureka, siendo un aplicativo de microservicios.

1.5 Criterios de Salida

Para considerar que todos los pasos de prueba fueron realizados se requerirá de:

- **Todos los casos de prueba ejecutados:** Casos de pruebas ejecutados con evidencia de la ejecución de cada uno.
- **Requisitos funcionales cumplidos:** Las pruebas funcionales han sido ejecutadas y es comprobable que el resultado obtenido es equivalente al resultado esperado descrito en las especificaciones de los requerimientos que fueron usados para implementar el sistema.
- **Requisitos no funcionales cumplidos:** Las pruebas no funcionales han sido ejecutadas, de acuerdo con criterios como el rendimiento, la escalabilidad y la seguridad.
- **Rendimiento aceptable:** El sistema ha sido sometido a pruebas de rendimiento y carga, con resultados en rangos acordados.
- **Plan de pruebas actualizado:** En función de cualquier modificación o cambio que haya surgido durante el testeó, el plan de pruebas es actualizado.

2 Test del Sistema

El sistema por probar se define como un aplicativo capaz de gestionar dos entidades en una base de datos, estos son las organizaciones y los superhéroes que pertenecen a estas. Con gestionar se hace referencia a ser capaz de crear, revisar y eliminar las entidades basado en los métodos HTTP. Por el lado de la interfaz gráfica, el usuario puede visualizar las entidades creadas y cambiarse de ventanas.

2.1 Estrategia de Test del Sistema

El sistema será probado con pruebas funcionales y no funcionales. Para las primeras se realizarán pruebas con las herramientas Postman y Selenium, a modo de probar requests y elementos del aplicativo; Por otro lado, para las pruebas no funcionales se utilizará la herramienta SonarCloud para escanear el código fuente.

Entradas:

- Solicitudes HTTP: Peticiones/requests enviadas a diferentes endpoints de la API utilizando métodos HTTP (POST, GET y DELETE).
- Datos de prueba: Se entregarán datos de prueba en formato JSON para gestionar los recursos.
- Criterios de prueba: Se definirán criterios específicos para las pruebas para establecer si se cumple o no con el resultado esperado en ellas.
- Acciones automatizadas para la página web.

Salidas:

- Respuestas HTTP: Respuestas a las peticiones/requests enviadas a la API.
- Actualización en la Base de Datos: Gestión de los recursos en la base de datos, esto es, creación y eliminación de estos en sus respectivas tablas y relaciones entre ellos.
- Rendimiento: Métricas de rendimiento, como la velocidad de respuesta de las peticiones/requests.
- Errores y Excepciones: Gestión de excepciones mediante mensajes de errores comprensibles.
- Ejecución de las acciones programadas.

2.2 Pruebas en Operación Normal

Se entiende por modo de operación normal cuando el sistema provee la funcionalidad y se puede reconocer que el sistema la está entregando.

2.2.1 *Método POST*

- Pruebas de creación de organización.
- Prueba de creación de superhéroe.

2.2.2 *Método GET*

- Pruebas de revisión de las organizaciones.
- Prueba de revisión de una organización específica.
- Prueba de revisión de los superhéroes.
- Prueba de revisión de un superhéroe específico.

2.2.3 *Método DELETE*

- Pruebas de eliminación de una organización.
- Prueba de eliminación de un superhéroe.

2.3 Pruebas en Condiciones de Excepción

2.3.1 *Método POST*

- Se enviarán solicitudes POST con parámetros no válidos.

2.3.2 *Método GET*

- Se enviarán solicitudes GET a recursos no existentes.

2.3.3 *Método DELETE*

- Se enviarán solicitudes DELETE a recursos no existentes.

2.4 Criterios de Éxito/Fracaso de Pruebas

Una prueba se considera exitosa si entrega el resultado esperado a partir de los datos de entrada proporcionados. Por tanto, un grupo de pruebas será exitoso cuando todas las pruebas que lo componen lo son. En caso opuesto, las pruebas se considerarán fallidas, afectando al grupo al que pertenecen.

Los valores que puede tomar un resultado serán:

Exitoso – Parcialmente - Fallido.

2.5 Entregables

Los resultados de las pruebas serán documentados en la herramienta TestLink.

3 Configuración del Test

Esta sección establece los componentes del ambiente de testing

3.1 Hardware

- Computadora, ya sea ordenador fijo, Notebook, entre otros.

3.2 Software

- Windows 11.
- Git.
- IDE para Java, por ejemplo, IntelliJ o Eclipse con el plugin de Lombok.
- JDK 11.
- Spring Tool Suite.
- Postman.
- Selenium.
- Navegador web, por ejemplo, Google Chrome.

3.3 Otros

- Datos de prueba, por ejemplo, en formato JSON.

3.4 Ambiente

En un Notebook con Windows 11, a partir del código fuente disponible en GitHub, se realiza la clonación del repositorio y se ejecuta en el IDE Eclipse. Para ello se debe tener instalado el JDK 11 con el plugin de Lombok y Spring Tool Suite. Una vez ejecutado, se realizan las requests mediante la herramienta Postman y se puede acceder a la base de datos H2 en el localhost en el puerto 8090.

4 Tareas

4.1 Actividades

La secuencia de actividades para probar el sistema es:

- Clonar el repositorio del proyecto.
- Carga el código fuente en SonarCloud.
- Ejecutar la aplicación en el IDE.
- Desarrollar las pruebas con Selenium.
- Desarrollar las pruebas con Postman.
- Documentar los resultados.

4.2 Responsabilidades

Esta sección establece las responsabilidades de cada grupo que participa en la fase de pruebas.

4.2.1 *Responsabilidades del Grupo de Desarrollo*

- Ejecutar las pruebas unitarias
- Ejecutar y probar la integración de bajo nivel
- Corregir los problemas reportados
- Poner a disposición de los Testers (QA) el código

4.2.2 *Responsabilidades del Grupo de Testing*

- Planificar las pruebas del sistema
- Configurar el ambiente de prueba
- Ejecutar las pruebas del sistema
- Escribir el reporte de test

4.2.3 *Responsabilidades de la Gerencia*

- Proveer recursos
- Aceptación final y aprobación de la liberación del producto
- [otras responsabilidades]

4.3 Planificación

La planificación estimada de las actividades de testing será realizada en la semana del 23 de octubre de 2023, con fecha de entrega de los reportes el viernes 27 del mismo mes.

Glosario (Definiciones y Siglas)

API: Application Programming Interface.

Endpoint: Dirección de una API.

HTTP: Hypertext Transfer Protocol.

IDE: Entorno de Desarrollo Integrado.

QA: Quality Assurance.

REST: Representational State Transfer.