

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Informe sobre Test de SIW



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2022 – 2023

Fecha	Versión
15/02/2023	2

Grupo: C1.03.05	
Repositorio: https://github.com/JuanCarlosRL/Acme-L3-D01	
Miembros	Correo
Fernández Mota, Francisco	frafermot@alum.us.es
García García, Jaime	jaigargar1@alum.us.es
Peláez Moreno, Antonio	antpelmor@alum.us.es
Ramírez López, Juan Carlos	juaramlop2@alum.us.es
Sánchez Mayorga, Alejandro	alesanmay@alum.us.es

Tutor: Jimenez Aguirre, Patricia

Tabla de Contenidos

Tabla de Contenidos	2
1. Tabla de revisión	3
2. Introducción	4
3. Contenido	5
4. Conclusión	6
5. Bibliografía	7

1. Tabla de revisión

Nº de revisión	Fecha	Descripción
1	14/02/2023	Creación del documento
2	15/02/2023	Añadidos los puntos 2, 3 y 4
3	16/02/2023	Corrección tras la revisión

2. Introducción

Este informe recoge todo el contenido que conocemos relacionado con las pruebas de un SIW (Sistema de Información Web).

Este documento tiene 5 puntos. El primero, después del índice, es una breve introducción. Después encontramos una tabla de revisión en la que se encuentran las versiones por las que ha pasado el documento. A continuación, encontramos todo lo que los miembros del grupo conocemos sobre testing de un sistema de información web, y una conclusión. Por último, se encuentra la bibliografía.

3. Contenido

En asignaturas anteriores hemos tratado el tema del testing. En concreto, en DP1 estuvimos trabajando con tests unitarios sobre repositorios, servicios y controladores. Vimos que según la finalidad del test, puede ser sociable o aislado, es decir, se comunica con el resto de componentes del sistema (como tomar datos de la base de datos), o no. Por ejemplo, para hacer tests para los repositorios era necesario que fueran sociables, para comprobar que los métodos seleccionan correctamente los datos esperados. Sin embargo, para los servicios usábamos test aislados. Para ello, usábamos la herramienta Mockito, que nos permitía generar un mock, el cual podíamos configurar para predeterminar una respuesta. Por último, a la hora de testear controladores, simulamos una llamada al controlador y comprobamos que la respuesta del servidor fuese la correcta, al igual que la vista y los modelos fuesen los esperados.

Para realizar todo esto usábamos diferentes notaciones como @Test, @Mock o @Mockito.

Por otro lado, también hemos tratado las pruebas de carga en la asignatura de EGC. La finalidad de estos tests es probar el rendimiento del sistema bajo una gran cantidad de peticiones. De esta forma podemos medir la estabilidad y el desempeño del sistema.

4. Conclusión

Esta es la información que conocemos actualmente sobre testing de sistemas de información web. Esperamos que gracias a esta asignatura podamos ampliar nuestro conocimiento sobre el tema.

5. Bibliografía

Intencionadamente en blanco.