

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

D01 – WIS Testing Report



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas II

Curso 2023 – 2024

Fecha	Versión
16/02/2024	v1r1

Grupo de prácticas: C1.002	
Autores	Correo Corporativo
José Ramón Baños Botón	josbanbot@alum.us.es
Manuel Palacios Pineda	manpalpin@alum.us.es
Rubén Pérez Garrido	rubpergar@alum.us.es
Javier Ramírez Núñez	javramnun@alum.us.es
Sonia María Rus Morales	sonrusmor@alum.us.es

Link repositorio: <https://github.com/Manpalpin/Acme-SF-D01-24.1.0.git>



Índice

1. Tabla de versiones.....	3
2. Introducción.....	4
3. Contenido.....	5
4. Conclusiones.....	6
5. Bibliografía.....	6



1. *Tabla de versiones*

Fecha	Versión	Descripción
16/02/2024	v1r0	Creación del documento
16/02/2024	v1r1	Entrega 1



2. Introducción

En este documento, compartiremos nuestros conocimientos previos sobre el testing en un Sistema de Información Web (WIS), en el contexto de la asignatura Diseño y Pruebas II. Discutiremos los diferentes tipos de test que conocemos, centrándonos especialmente en los tests unitarios. Abordaremos las clases que suelen ser objeto de testeo en este contexto. Es fundamental señalar que esta información es familiar para el grupo antes de comenzar esta asignatura, lo que nos proporciona una base sólida para comprender y aplicar los conceptos discutidos en este curso. Nuestro objetivo es ofrecer una visión general de los conceptos clave y las mejores prácticas que hemos aprendido en proyectos anteriores, contribuyendo así a nuestro proceso de aprendizaje y promoviendo la discusión dentro del grupo y con el profesorado.

3. Contenido

Algunos de los tipos de tests conocidos por el grupo previos a la asignatura son:

- Test unitarios:

Son aquellas pruebas que se enfocan en validar exclusivamente una funcionalidad del sistema. Siguen una estructura 'Arrange/Fixture', 'Act', 'Assert'. Las pruebas pueden ser realizadas de forma manual o automatizada. Pueden ser tanto afirmativas, que verifican que la funcionalidad se ejecuta correctamente cuando corresponde, como negativas, que confirman que se generan las excepciones cuando corresponde.

Dentro de una arquitectura Modelo-Vista-Controlador (MVC), entendemos que estas pruebas se suelen llevar a cabo para validar el funcionamiento de los métodos en las clases "Servicio" y "Controlador". En las pruebas de "Servicio", se verifica el funcionamiento de la lógica de negocio, mientras que en las pruebas de "Controlador", se evalúa el comportamiento de las solicitudes HTTP del cliente al servidor. Para las pruebas de "Controlador", con el fin de evitar depender del sistema, se pueden emplear dobles de prueba utilizando 'Stub', 'Mock' o 'Fake'. 'Stub' devuelve un valor por defecto, 'Mock' verifica que se llame al método correspondiente sin considerar el valor, y 'Fake' crea una réplica simplificada del método para evitar recurrir al real.

- Test de integración:

Son aquellos test que verifican que los diferentes módulos y/o servicios usados por el sistema funcionan en armonía cuando trabajan en conjunto.

- Test de aceptación:

Estas pruebas son diseñadas para validar si un sistema cumple con los requisitos del negocio. Se ejecutan con el software en funcionamiento y se enfocan en emular las acciones de los usuarios para asegurar el logro de los objetivos.

- Text exploratorio:

El testing exploratorio es una metodología en la que se aprende sobre la aplicación, se diseñan casos de prueba y se ejecutan simultáneamente. Cada sesión de testing tiene un objetivo definido y un límite de tiempo para su realización.

- Text End-To-End:

Son aquellos test que replican el comportamiento de los usuarios con el sistema. Estas pruebas verifican que los flujos que sigue un usuario de los resultados que se esperen.



4. Conclusiones

Las pruebas son fundamentales porque nos posibilitan verificar el correcto funcionamiento del sistema que hemos desarrollado antes de su finalización total. Esto evita tener que corregir errores que podrían resultar más costosos a medida que el sistema crece. Además, nos ayudan a detectar de manera más sencilla los problemas dentro del código. También es esencial saber seleccionar el tipo de prueba adecuado según lo que se desee validar.

5. Bibliografía

Diapositivas de teoría de la asignatura Diseño y Pruebas I.