

DP2 2021-2022

TESTING WIS

Proyecto Acme Toolkits

<https://github.com/josperrod9/Acme-Toolkits.git>

Miembros:

- Pablo Benítez Oliva(pabbenoli@alum.us.es)
- Pablo Giráldez Álvarez(pabgiralv@alum.us.es)
- Alberto Monedero Martín(albmonmar3@alum.us.es)
- Andreu Montagut Osuna(andmonosu@alum.us.es)
- José Pérez Rodríguez(josperrod9@alum.us.es)
- Mario Rey Carmona(marreycar1@alum.us.es)

Tutor: José González Enríquez

GRUPO E7-01

Versión 1.0

Índice

https://github.com/josperrod9/Acme-Toolkits.git	1
Miembros:	1
GRUPO E7-01	1
Versión 1.0	1
Índice	2
TABLA DE REVISIONES	3
Fecha	3
Versión	3
Descripción de los cambios	3
INTRODUCCIÓN	3
CONTENIDO	4
BIBLIOGRAFÍA	4

1. TABLA DE REVISIONES

Fecha	Versión	Descripción de los cambios
23/02/2022	V1	<ul style="list-style-type: none">• Creación y realización del documento

2. INTRODUCCIÓN

En este documento vamos a hablar de nuestros conocimientos previos adquiridos en la asignatura de Diseño y Pruebas 1 acerca del testing de los sistemas de información web (WIS). En este documento vamos a explicar la estrategia de Testing que seguimos en el proyecto de DP1 y de los beneficios que este aporta a un sistema de información web. Además, vamos a hablar sobre los tests que se realizan de los Servicios y los Controladores, y por último hablaremos de la importancia de la cobertura del código mediante pruebas.

En cuanto a la estructura que va a tener el apartado Contenido el cual se encuentra en la página 4 de este documento, se divide en cuatro párrafos y que son: importancia de recorrer toda las ramas de un método, tests de Servicios, tests de Controladores y el objetivo del testing. En los apartados que explicamos los tests de Servicios y Controladores hemos dado una serie de ejemplos sobre los métodos que se pueden probar en cada tipo de test.

3. CONTENIDO

Para testear al completo un sistema de información web (WIS), en la asignatura de Diseño y Pruebas 1 aprendimos que a la hora de realizar los test de una sistema web, debíamos de probar todas las posibilidades de un método, es decir, debíamos de intentar pasar por el mayor número de ramas del método que se generan al utilizar estructuras condicionales en ese método.

Por un lado, vamos hacer test sobre los Servicios, en los que comprobamos el correcto funcionamiento de los métodos contenidos en esta clase, sea este un método save con el que guardamos los datos de una clase en la base de datos o métodos delete con los que eliminamos una clase de la base de datos, etc. Además, de probar los métodos de los servicios, se prueban los métodos de cada modelo de datos que tengamos en el sistema Web, que suelen ser métodos con los que modificamos datos de atributos que pertenecen a esa clase.

Por otro lado, hicimos tests de los Controladores, en los que usábamos Mockito para mockear las vistas que están definidas en cada uno de los controladores y con ello probar que el funcionamiento de cada una de las vistas es correcto. Un ejemplo de este tipo de test, sería la prueba de una vista de tipo Post Mapping, que es usada para insertar un usuario nuevo en la base de datos de la aplicación web. Para el testeo de esta clase debemos mockear la vista e introducir unos parámetros que se hubieran introducido en el formulario de registro del usuario, una vez se introducen dichos parámetros comprobaremos que el sistema no ha mostrado ningún error a la hora de insertar los datos en la base de datos mockeada.

El objetivo del testing que nosotros perseguimos era tener cubierto la mayoría del código con las pruebas, ya que esta es una manera de asegurarse de que el software que hemos desarrollado funciona correctamente y permite a los desarrolladores darse cuenta de pequeños fallos en el sistema que deberían de ser solucionados. El testing también nos permite construir un sistema completo, robusto y de calidad que cumple con las expectativas del cliente.

4. BIBLIOGRAFÍA

Intencionalmente en blanco