

INFORME DE PRUEBAS A UN WIS

GRUPO C1.02.03



Autores:

- Alejandro Campano Galán – alecamgal1@alum.us.es
- Juan Jesús Campos Garrido – juacamgar2@alum.us.es
- David Cortabarra Romero – davcorrom@alum.us.es
- Alejandro García Sánchez-Hermosilla – alegarsan11@alum.us.es
- Pablo Mera Gómez – pabmergom@alum.us.es

Repositorio: <https://github.com/Alejandrocg024/Acme-L3-D01.git>

Fecha del documento: 15/02/2023

Tabla de contenidos.

Resumen general..... 3

Tabla de revisión. 4

Introducción. 5

Contenido..... 6

Conclusiones. 8

Bibliografía. 9

Resumen general.

Tras cursar distintas asignaturas sobre el desarrollo software, aprendimos como desarrollar correctamente un sistema de información web. Para lograr un correcto funcionamiento, aprendimos que es necesario aplicar distintos tipos de pruebas en los proyectos y así, asegurar que el sistema funciona de la forma en la que deseamos y que no contiene errores ni elementos que hagan que el producto no sea el deseado.

Tabla de revisión.

Fecha	Versión	Descripción
11/02/2023	0.1	Creación de una primera versión
12/02/2023	0.2	Creación de todo el documento y redacción del informe
15/02/2023	1.0	Corrección de errores. Versión final

Introducción.

En este documento, hablaremos sobre los conocimientos previos que tenemos sobre hacer pruebas a un sistema de información que sigue la arquitectura WIS. Trataremos diversos temas desde el tipo de prueba que podemos realizar sobre nuestro sistema, hasta la granularidad que las pruebas podrían tener y distintos tipos de escenario para cada prueba. También hablaremos sobre las características fundamentales que debe seguir cada prueba y una breve configuración de las mismas.

Contenido.

Basándonos en los conocimientos aprendidos de las asignaturas de los cursos anteriores, hemos aprendido diversidad de cosas sobre los sistemas de información. Pero a la hora de hablar sobre realizar pruebas a los sistemas de información con una arquitectura WIS, nos centraremos en la asignatura de Diseño y pruebas 1. En esta asignatura, aprendimos que las pruebas tienen distintos tipos de granularidad. Esto quiere decir que dependiendo del tipo de granularidad, las pruebas comprobarán una sola parte del código como pueden ser las pruebas unitarias, por ejemplo. Otro tipo de pruebas podrán comprobar el sistema entero como son las pruebas de aceptación en las cuales se verifica que el sistema desarrollado es lo que el cliente quiere. Aparte de estos dos tipos de granularidad, también podremos encontrar pruebas de Integración, pruebas de exploración, y, sobre todo, un tipo de pruebas que serán las que veremos en esta asignatura que son las pruebas End-to-End que consisten en simular que el usuario está interactuando con el sistema y sus acciones.

Además de los tipos de pruebas también encontramos vocabulario explícito sobre el entorno de pruebas, como por ejemplo SUT, las siglas de System Under Test que se refiere al sistema que está siendo probado. Por otro lado, también encontramos los casos de pruebas que nos comprueban cómo funciona el sistema y tenemos dos tipos. Por un lado, tendremos los casos de pruebas positivos que son el resultado de realizar una prueba que un usuario final ejecutaría para su uso normal con datos correctos y válidos. Por otro lado, tenemos los casos de pruebas negativos, en los que el sistema se prueba ejecutándolo con datos incorrectos o no válidos para probar que el sistema funciona correctamente en estas situaciones. También, nos encontraremos con otras palabras como bugs o fallos.

También, aprendimos que las pruebas deben de seguir una serie de características comunes que son:

1. Una prueba debe de ser rápida de ejecutar
2. Una prueba debe de ejecutarse de forma independiente
3. Una prueba debe de repetirse y no cambiar su valor, es decir no debe depender de factores externos
4. Una prueba debe de comprobarse de forma automática sin ningún tipo de supervisión
5. Las pruebas deben de actualizarse al mismo tiempo que el código se está probando

Además de obtener información sobre como son las pruebas y sus tipos, también obtuvimos conocimientos sobre la configuración de estas pruebas. Por ejemplo, a la hora de realizar pruebas a procedimientos que realizan operaciones con la base de datos, puede que no nos interese utilizar esta conexión por la base de datos ya que puede ralentizar la prueba, entonces, hacemos uso de unos “dobles” que imitan la funcionalidad del sistema. También, utilizamos un tipo de dobles que llamamos “mocks” que realmente están vacíos y lo debemos de configurar previamente y nos sirven para testear la interacción entre objetos.

Por último, uno de los conceptos más importante que hemos aprendido sobre las pruebas, es que como tienen un coste y consumen tiempo, no podemos hacer infinitas pruebas, por lo que no podremos ser tan exhaustivos como nos gustaría y siempre se podría escapar algún caso.

Conclusiones.

Después de cursar las anteriores asignaturas descubrimos que realizar pruebas a un sistema es una parte trivial a la hora de desarrollar un producto software, ya que es la manera para comprobar si el código del sistema funciona como realmente deseamos y si no sufre ningún tipo de fallo inesperado, bug en el código o brechas de seguridad. En conclusión, es una tarea que en un principio, no parece tan importante, pero es una tarea fundamental y que debe tomar gran importancia en el trabajo de un desarrollador, para obtener un producto perfecto al finalizar su trabajo.

Bibliografía.

Intencionalmente en blanco.