



Escuela Técnica Superior de
Ingeniería Informática

Testing Report

Deliverable-05

Miembros del equipo

Apellidos, Nombre	Correo electrónico	Roles
Gallardo Roco, Raúl	raugalroc@alum.us.es	Manager Desarrollador
Gómez Arias, Nuria	nurgomari@alum.us.es	Desarrollador
Ortiz Guerra, Juan Antonio	juaortgue@alum.us.es	Desarrollador
Páez Páez, Jesús	jespaepae@alum.us.es	Desarrollador Operador
Piñero Calera, Borja	borpincal@alum.us.es	Desarrollador
Ramos Berciano, Pablo	pabramber@alum.us.es	Desarrollador

Repositorio: <https://github.com/raugalroc/Acme-Toolkits>

Grupo E2.06

Fecha: 02 de Junio de 2022

Índice

Resumen Ejecutivo	3
Tabla de Revisión	3
Introducción	3
Contenidos	3
Conclusiones	4
Bibliografía	4

Resumen Ejecutivo

En nuestro grupo E2.06 tenemos conocimientos sobre este tipo de proyectos debido a asignaturas como Diseño y Pruebas I e Introducción a la ingeniería del Software y Sistemas de la Información 1 y 2. Gracias a ello podemos desarrollar sistemas con características parecidas a las que se requieren para este proyecto.

Nuestra intención es completar todas las características de cada Sprint para intentar obtener la máxima calificación posible en la asignatura.

Tabla de Revisión

Versión	Fecha	Descripción
1.0	2/06/2022	Primera versión

Introducción

En este documento se tratarán los aspectos que el equipo ha aprendido de cómo funcionan y cómo están realizadas las pruebas de una arquitectura WIS tras el paso por la asignatura de Diseño y Pruebas II y tras la realización del proyecto Acme Toolkits.

Contenidos

Hemos vuelto a ver los diferentes tipos de pruebas que ya se vieron en DP1, como pueden ser las pruebas estáticas, dinámicas, de caja negra, E2E, etc. Concretamente, de estas últimas con diferencia a la asignatura anteriormente dicha, se ha llevado a cabo más en práctica pruebas E2E, es decir, pruebas que comprueban el funcionamiento de la aplicación entera, en lugar de las pruebas unitarias que se usaron junto con mock en DP1. En este sentido, hemos aprendido a cómo a realizar las pruebas metodológicamente, de forma que primero se realiza un test de forma general parametrizado, cuyos parámetros serán los distintos datos que se quieran probar desde un archivo de almacenamiento, en nuestro caso, csv.

Por otro lado, a la hora de generalizar las funciones java de test, hemos repasado:

- Positivos: happy path con resultado esperado
- Negativos: no happy path con resultado esperado
- Hacking: no happy path con posible resultado inesperado

De esta forma, es recomendable, siempre que se pueda, realizar uno de cada tipo para probar una determinada funcionalidad / característica.

Por último, también hemos aprendido a cómo generar esos archivos csv con los datos de prueba de forma metodológica. Para ello, dependerá del tipo de funcionalidad:

- **Mostrar:** en este caso se deben recuperar los datos que haya en la base de datos que sean necesarios para comprobar que son los esperados.
- **Edición:** en este caso se debe probar, más profundamente, los casos negativos, pues, a la hora de poblar la base de datos, si se ha hecho metodológicamente, se habrán probado todos los tipos válidos (valores extremos, como por ejemplo).

Conclusiones

Como hemos podido comprobar, hemos aprendido las principales características del testing una arquitectura WIS, su funcionamiento general, sus elementos más importantes y las tecnologías que pueden ser usadas en ellos y las interacciones que pueden darse durante el proceso.

Bibliografía

Intencionadamente en blanco.