

Deliverable-05

Testing lessons learned report

Workgroup: E7.04

Date: 02/06/2022

Miembros:

Baños González, Alejandro (alebangon@alum.us.es)

Flores Rodríguez, Román (romflorod@alum.us.es)

Grosso Gómez de Terreros, Javier (javgrogom@alum.us.es)

Gutiérrez Ceballos, Pablo (pabgutceb@alum.us.es)

Ibáñez Montero, Julia (julibamon@alum.us.es)

Roldán Cadena, Jesús (jesrolcad@alum.us.es)

Repositorio de Github: <https://github.com/jesrolcad/Acme-Toolkits>

Índice

1. Resumen ejecutivo	1
2. Tabla de control de cambios	2
3. Introducción	3
4. Fundamentos del testing aprendidos	4
5. Conclusiones	6
6. Referencias bibliográficas	7

1. Resumen ejecutivo

En este documento se procede a recoger las lecciones aprendidas sobre testing a lo largo de nuestro proyecto.

Estas lecciones han sido divididas según el tipo de testing, el cual lo hemos distinguido en manual, automático o de rendimiento (performance report).

Sobre el testing manual y el automático, a parte de lo aprendido, expondremos las ventajas y desventajas de estos. Sobre el performance report, detallamos cómo nos ha ayudado a comprender la diferencia entre hardwares y cómo esto ayuda a mejorar el código.

Finalmente, el equipo concluye con que gracias a la elaboración de este documento hemos podido aclarar lo aprendido en la asignatura, junto a las ventajas y desventajas de las diferentes formas de testing.

2. Tabla de control de cambios

Número de revisión	Descripción	Fecha
1	Creación del documento con todas sus secciones. Se ha completado de forma completa la sección 4. <i>Fundamentos del testing aprendidos</i> , y se han detallado las conclusiones obtenidas tras la redacción del documento.	26/05/2022
2	Revisión general del documento. Corrección de faltas de ortografía.	31/05/2022
3		

3. Introducción

En este informe se recoge los diferentes puntos aprendidos en las tareas de testing a lo largo de la asignatura. Para ello, a continuación se describen las distintas secciones en las que se divide este documento.

En sección 4. *Fundamentos del testing aprendidos* se detallan los contenidos sobre testing aprendidos a lo largo del curso. Esta sección se ha dividido en las subsecciones 4.1 *Fundamentos del testing manual aprendidos*, 4.2 *Fundamentos del testing automáticos aprendidos* y 4.3 *Conocimientos aprendidos sobre el performance report*.

Las conclusiones obtenidas tras la realización de este informe sobre lo aprendido en la tarea de testing se encuentran en la sección 5. *Conclusiones*. En dicha sección agrupamos lo más importante acerca de lo aprendido sobre fundamentos de testing.

La sección 6. *Referencias bibliográficas* recoge la bibliografía usada para la elaboración del informe. En este documento esta sección se encuentra vacía porque no se ha usado ningún tipo de documentación bibliográfica para redactarlo.

Finalmente, este informe cuenta con la siguiente estructura: portada, índice, resumen ejecutivo, tabla de control de cambios, introducción, contenidos (que se haya en la sección *Fundamentos del testing aprendidos*), conclusiones y referencias bibliográficas.

4. Fundamentos del testing aprendidos

4.1 Fundamentos del testing manual aprendidos

A lo largo del proyecto hemos aprendido bastantes fundamentos sobre el testing de la aplicación, tanto de forma manual como automatizada.

De forma manual hemos aprendido a probar el uso de formularios, insertando de forma manual los datos. También hemos aprendido a realizar pruebas de hacking manuales que consisten en que un rol determinado acceda por url a partes de la aplicación que se encuentran restringidas. Algunos casos que hemos probado de manera manual son, por ejemplo, que Inventor no pueda acceder a la Toolkit de otro inventor o rellenar formularios con valores con spam.

También hemos aprendido que de esa forma se pueden encontrar errores específicos que normalmente no aparecen en métodos automatizados, ya que dependen de factores muy específicos y no de los límites en los parámetros de la entidad. Por ejemplo, hemos corregido ciertos errores de valores nulos, como por ejemplo cuando se crea un Patronage, que requiere una fecha de inicio y una fecha de fin. Como la validación de la fecha de fin depende de la fecha de inicio, debíamos comprobar que dejando la fecha de inicio en blanco y rellenando la fecha de fin no se producía un `NullPointerException`.

La comprobación manual, aún siendo útil, ha demostrado ser un trabajo bastante pesado, por lo que era necesario hacer también realizar tests automatizados.

4.2 Fundamentos del testing automatico aprendidos

En los tests automatizados, hemos aprendido lo importante que es realizar pruebas empleando los valores límites de los atributos de las entidades en los formularios. Además, hemos aprendido a realizar pruebas E2E, las cuales no habíamos desarrollado en ninguna otra asignatura a lo largo de la carrera.

La metodología de testing que hemos aprendido se basa en que tanto para los casos positivos como para los negativos es imprescindible diseñar diversos casos de prueba que contemplen todos los posibles casos de datos (incluyendo los casos límite) para todas las funcionalidades implementadas. Por ejemplo, en el caso del precio de venta de los Items, probamos con valores de -0.01 o 0.01.

Gracias al soporte ofrecido para el desarrollo de pruebas automatizadas, hemos podido comprobar de forma más sencilla si los formularios y las funciones están implementados de forma correcta, aceptando la información cuando es correcta, mostrando mensajes de error cuando es incorrecta o denegando el acceso si no estás autorizado a esa vista.

También es muy importante asegurarse de que los datos abarcan todas las posibles variables y restricciones en los casos de prueba usados.

4.3 Conocimiento aprendido en el performance report

La redacción y confección del informe de rendimiento nos acerca directamente a comprender y entender los datos extraídos del testing de manera analítica y estadística. Durante la redacción del documento de pruebas de rendimiento hemos aprendido conceptos clave como MIR (Most Inefficient Request), benchmarking o refactoring.

Las pruebas de rendimiento nos han ayudado a comprender cómo las diferentes especificaciones de un ordenador afectan significativamente al rendimiento del mismo, observándose esto al comparar la velocidad de ejecución de los tests en los 2 equipos que se utilizaron para realizar dicho caso de pruebas.

Gracias a las pruebas de rendimiento, los integrantes del grupo hemos comprendido de manera matemática (concretamente, mediante intervalos de confianza y contrastes de hipótesis) cómo de eficiente es nuestro código.

5. Conclusiones

En definitiva, hemos podido comprobar personalmente el valor que una buena metodología de testing proporciona a cualquier clase de proyecto informático. Además de que hemos podido comprobar que existe cierta variedad de métodos, cada uno con sus ventajas y desventajas. Por ello, el equipo saca estos aspectos en clave respecto a lo aprendido en testing:

- De forma manual es más fácil ver el porqué de los errores y se puede buscar fallos más específicos y enfocados a un mal uso de la aplicación.
Pero aún siendo útil e intuitivo, es la forma más pesada de testing y requiere mucho tiempo.
- De forma automática se puede simplificar mucho la tarea de testing, pero requiere que se desarrollen casos con variedad de datos, los cuales necesitan abarcar el número máximo de variables posibles tanto positivas como negativas al rellenar los diferentes cuestionarios.
- Por último, el performance report ha permitido extraer de forma analítica y estadística los datos extraídos durante el testing automático, permitiéndonos analizar la eficiencia de nuestro código y comprender cómo dependiendo del hardware usado, se puede mejorar o no el rendimiento del software.

6. Referencias bibliográficas