

# DP2 2021-2022

## Tests Report

### Proyecto Acme Toolkits

Enlace proyecto: <https://github.com/TomasCB24/Acme-Toolkits.git>

Miembros:

- Tomás Camero Borrego | [tomcambor@alum.us.es](mailto:tomcambor@alum.us.es)
- Ezequiel González Macho | [ezegonmac@alum.us.es](mailto:ezegonmac@alum.us.es)
- Ismael Pérez Ortiz | [ismperort@alum.us.es](mailto:ismperort@alum.us.es)
- Pablo Rivera Jiménez | [pabrivjim@alum.us.es](mailto:pabrivjim@alum.us.es)
- Miguel Romero Arjona | [migromarj@alum.us.es](mailto:migromarj@alum.us.es)
- Juan Salado Jurado | [juasaljur@alum.us.es](mailto:juasaljur@alum.us.es)

Tutor: Rafael Corchuelo Gil

GRUPO E7.03

Versión 1.0

02/06/2022

## Tabla de contenido

Resumen ejecutivo.....	3
Historial de versiones.....	3
Introducción.....	3
Contenido.....	3
Conclusión.....	4

## Resumen ejecutivo

Desde que comenzó el desarrollo del proyecto Acme Toolkits, nuestro equipo ha ido adquiriendo nuevas habilidades por medio de la formación recibida. En concreto, en los últimos *sprints* hemos aprendido a diseñar a implementar los *tests* necesarios para asegurar la funcionalidad requerida por el cliente y una calidad adecuada de la aplicación, principalmente desde el punto de vista de la seguridad.

En concreto en los dos últimos *sprints* hemos profundizado bastante en los distintos tipos de *testing* existentes en el mundo del desarrollo software y hemos aplicado este conocimiento por medio del framework Acme proporcionado en la asignatura.

## Historial de versiones

Fecha	Versión	Descripción de los cambios	Entregable
25/05/2022	0.1	Creación del documento y realización del borrador inicial	5
26/05/2022	1.0	Añadidas secciones de “Contenido” y “Conclusión”	5

## Introducción

Desde que comenzó el desarrollo del proyecto Acme Toolkits, nuestro equipo ha ido adquiriendo nuevas habilidades por medio de la formación recibida. En concreto, en los últimos *sprints* hemos aprendido a diseñar a implementar los *tests* necesarios para asegurar la funcionalidad requerida por el cliente y una calidad adecuada de la aplicación, principalmente desde el punto de vista de la seguridad y el rendimiento.

En concreto en los dos últimos *sprints* hemos profundizado bastante en los distintos tipos de *testing* existentes en el mundo del desarrollo software y hemos aplicado este conocimiento por medio del framework Acme proporcionado en la asignatura.

## Contenido

## Testing funcional

Al inicio estudiamos el vocabulario básico relacionado con el testing, que aunque ya conocíamos previamente, nos ha servido para recordar sus significados. Hablamos de términos como requisitos, características de un producto software, “bugs” o fallos. Además, hemos repasados conceptos como SUT (*System Under Test*) que en nuestro caso se trata de la aplicación en desarrollo, el “fixture”, que en nuestro caso solo comprende las bases de datos, aunque en otros casos puede comprender “mocks” o datos en la nube por ejemplo, o test suite, que comprende el conjunto de tests que se van a ejecutar.

Por otra parte, en cuanto al diseño de los tests hemos aprendido tres tipos de casos de prueba, los casos positivos, donde se realizan acciones legales que no producen error, casos negativos, que también son legales pero estas deben producir un error, y de “hackeo” que se basan en comprobar que no se pueden realizar acciones ilegales, por ejemplo acceder a ciertas zonas de la aplicación que no están permitidas para algún rol específico de la aplicación.

En la práctica, usamos dos plugins de eclipse. Uno de ellos enfocado en medir la cobertura conseguida con los test implementados de una forma sencilla, *Coverage Runner*, y otro que nos permite inspeccionar el código en busca de errores o “malos olores”, sin necesidad de ejecutar la aplicación, *Sonar Lint*.

## Rendimiento

Otro de los aspectos del testing vistos durante el curso, ha sido el rendimiento de la aplicación y cómo probar que ésta cumple con los requisitos de rendimiento requeridos. En la práctica, aprendimos como medir este rendimiento y cómo generar un reporte donde analizar el funcionamiento de la aplicación bajo distintos puntos de vista.

## Conclusión

En definitiva, tras el desarrollo de una aplicación desde sus inicios hasta la entrega de esta con su funcionalidad completa, hemos adquirido un amplio conocimiento en el ámbito del *testing*.

## Bibliografía

Intencionadamente en blanco.