

# INFORME SOBRE LAS PRUEBAS

---

**GRUPO: E3.07**

**URL:** <https://github.com/juacasalb/Acme-One>

Juan Castro Albertos ([juacasalb@alum.us.es](mailto:juacasalb@alum.us.es))

Francisco Javier de la Prada Prados ([fraprapra1@alum.us.es](mailto:fraprapra1@alum.us.es))

Miguel Gaviro Martínez ([miggavmar@alum.us.es](mailto:miggavmar@alum.us.es))

Álvaro Gómez Nieto ([alvgomnie@alum.us.es](mailto:alvgomnie@alum.us.es))

## Historial de versiones

Fecha	Versión	Descripción de los cambios	Sprint
2/06/2022	1.0	Creación de documentos para el “Deriverable 5” y creación de la extensión <i>jar</i> que alberga el proyecto completo.	5

## Índice

1. [Introducción](#)
2. [Resumen ejecutivo](#)
3. [Contenido](#)
4. [Conclusión](#)
5. [Bibliografía](#)

## 1. Introducción

En primer lugar, el testing de software es un proceso para verificar y validar la funcionalidad de un programa o una aplicación de software con el objetivo de garantizar que el producto de software esté libre de defectos. En nuestro caso, hemos planteado escenarios de pruebas para las distintas funcionalidades de nuestro sistema, con casos positivos y negativos, que veremos con mayor profundidad a lo largo del documento.

## 2. Resumen ejecutivo

En este documento se resumen los conocimientos adquiridos gracias al conjunto de pruebas realizadas en esta asignatura.

## 3. Contenido

Para realizar las pruebas de nuestro proyecto, además de la correspondiente clase de cada test, se han creado también unos ficheros con extensión csv, los cuáles proporcionaban a las pruebas los datos necesarios para llevar a cabo un testing completo de los diferentes casos de uso que existen.

Los test realizados son del tipo E2E, de extremo a extremo, que quiere decir que se interactúa directamente con la aplicación. En ellos, se abría el navegador y de forma autónoma comprobaba que todas las pruebas eran correctas. Además, los tests extienden a TestHarness, cosa que nos ha facilitado bastante a la hora de hacerlos.

Hemos realizado tres tipos de pruebas:

### Pruebas positivas

En ellas se comprueba que la prueba se ejecuta correctamente. En el caso de crear una entidad, se inicia sesión, se accede al formulario de creación, y se rellenan los campos con parámetros válidos, sacados del csv que hemos comentado al principio del apartado. Posteriormente, vemos que no existen errores, y accedemos al elemento creado, para comprobar que sus atributos coinciden con los introducidos.

### Pruebas negativas

Este tipo de pruebas, al contrario que las positivas, consisten en introducir datos incorrectos con el fin de que el sistema devuelva errores. Estos datos pueden ser acceder a un elemento del menú que no está creado, introducir datos inválidos en una entidad, o dejar datos vacíos al crearla o actualizarla.

### Pruebas de hacking

Las pruebas de hacking, por su parte, sirven para confirmar la seguridad de la aplicación. Al hacerlas, intentamos acceder a urls prohibidas para el usuario autenticado, realizar acciones propias de otro usuario, hacer funcionalidades que pertenecen a otro rol, etc.

## 4. Conclusión

En conclusión, como su propio nombre indica, en la asignatura de Diseño y Pruebas II hemos aprendido dos de los aspectos fundamentales de un proyecto software, como son el diseño e implementación de código, y las pruebas para comprobar que todo lo demás se ha realizado correctamente. Gracias a ellas, que cubren una muy gran parte del código, se pueden detectar errores tempranos, los cuales se corregirán con mayor facilidad que uno que no se haya testeado y se detecte en una fase más avanzada del desarrollo. Por último, la realización de estas pruebas funcionales nos ha ayudado a adquirir conocimientos sobre este campo, ya que es la primera vez que nos las encontramos, y a sentir la importancia que tienen en un proyecto software.

## 5. Bibliografía

<https://profile.es/blog/que-es-el-testing-de-software/>