

# WIS Testing

**Group:** L3-D01

**Repository:** <https://github.com/adrrf/acme-life-long-learning>

**Student #1:**

**UVUS:** manvazmar3

**Name:** Vázquez Martín, Manuel

**Roles:** Developer

**Email:** manvazmar3@alum.us.es

**Student #2:**

**UVUS:** angmunpri

**Name:** Muñoz Prieto, Ángel

**Roles:** Developer, Analyst

**Email:** angmunpri@alum.us.es

**Student #3:**

**UVUS:** adrromflo

**Name:** Romero Flores, Adrián

**Roles:** Developer, Manager

**Email:** adrromflo@alum.us.es

**Student #4:**

**UVUS:** jorromlim

**Name:** Romero Limón, Jorge

**Roles:** Developer

**Email:** jorromlim@alum.us.es

**Student #5:**

**UVUS:** alematcap

**Name:** Mateo Capilla, Alejandro

**Roles:** Developer

**Email:** alematcap@alum.us.es

**Date:** Sevilla a 16 de febrero de 2023

<b>INTRODUCCIÓN</b>	<b>3</b>
<b>CONTENIDO</b>	<b>3</b>
<b>TESTS AUTOMATIZADOS</b>	<b>3</b>
TESTS UNITARIOS	3
TEST DE INTEGRACIÓN	3
TEST END TO END O TEST DE SISTEMA	3
<b>TESTS NO AUTOMATIZADOS</b>	<b>3</b>
TEST DE ACEPTACIÓN	3
<b>OTROS TESTS</b>	<b>3</b>
<b>CONCLUSIONES</b>	<b>4</b>

#### Executive Summary:

En este documento se detalla nuestro conocimiento sobre el testeo de aplicaciones web a lo largo del desarrollo de un proyecto software. Distinguimos entre tests automatizados, tales como pruebas unitarias, de integración y End to End, y no automatizados, como las pruebas de aceptación. Además, contemplamos la existencia de otro grupo el cual no comprueba si nuestro sistema correctamente sino que comprueba si tiene ciertos niveles de calidad, lo cual también es importante a la hora de cumplir los requisitos del cliente.

## Introducción

Como ingenieros de software nuestro principal objetivo es desarrollar software que sea funcional. Para comprobar si nuestro software funciona correctamente tenemos varios enfoques a los que tratar. Uno de estos enfoques es aprovechar herramientas para que podamos desarrollar pruebas que se ejecuten automáticamente y estas nos digan si nuestro software pasa o no la prueba. Otro enfoque completamente distinto es que un humano, ajeno al desarrollador del software, compruebe si el software cumple o no los requisitos especificados. En este documento recogeremos todo el conocimiento que tenemos sobre que pruebas podemos aplicar a una aplicación web.

## Contenido

### Tests automatizados

Estas pruebas son desarrolladas por el mismo equipo de desarrollo y prueban si los resultados que da nuestro código son correctos. Idealmente, son desarrolladas por una persona que no ha desarrollado el código a probar. Hay varios tipos de pruebas automatizadas dependiendo sobre todo de que cantidad de código estemos probando.

#### Tests unitarios

Es la unidad de prueba más atómica que nuestro software puede pasar. El objetivo principal de estas pruebas automáticas son comprobar que una pequeña parte del código, ya sea una función o un objeto (si estamos hablando de programación orientada a objetos), funciona correctamente. Normalmente, son los primeros tests en diseñar y en ejecutar en toda nuestra *test suite* (conjunto de todos los tests).

#### Test de integración

Es el siguiente nivel respecto a tamaño de código probado. Una vez las pruebas unitarias pasan, estos test comprueban como todos los elementos funcionan correctamente probándolos en conjunto, no aislados.

#### Test End to End o Test de sistema

Es el último nivel respecto a tamaño de código probado. Una vez pasadas las pruebas unitarias y pruebas de integración, estas pruebas prueban el sistema en su totalidad, intentando “simular” como actuaría un humano frente al sistema.

### Tests no automatizados

#### Test de aceptación

Estas pruebas no pueden llegar a ser desarrolladas y ejecutadas automáticamente, sino que necesitamos a una tercera persona, fuera del equipo de desarrollo, que compruebe que el software desarrollado cumple con los requisitos mencionados. Normalmente, esta tercera persona es el cliente al cual le estamos desarrollando el propio software.

#### Otros tests

Además de estas pruebas, existen tests que comprueban el sistema en apartados muy específicos, no solo en funcionalidad, sino en calidad. Hablamos de tests de seguridad, tests de instalación, tests destructivos, tests de rendimiento, tests de accesibilidad, etc.

## Conclusiones

Como resultado final, este documento refleja nuestro conocimiento acerca del testeo de una aplicación web. Vemos como tenemos varios enfoques a aplicar en cuanto a comprobar si nuestro sistema funciona como es debido. Como punto final, también sabemos que gran parte del valor no lo da solo la funcionalidad de un sistema, sino su calidad, por lo que también es importante comprobar si la calidad de nuestro software llena las expectativas del cliente.