

# DP2 2021-2022

## Knowledge of WIS testing



### Repositorio:

<https://github.com/mpadillatabuenca/Acme-Toolkit.git>

### Miembros:

- José Manuel Bejarano Pozo (josbezpoz@alum.us.es)
- Mario Espinosa Rodríguez (maresprod5@alum.us.es)
- Andrea Meca Sánchez (andmecsan@alum.us.es)
- Manuel Padilla Tabuenca (maresprod5@alum.us.es)
- Ezequiel Pérez Sosa (ezeopersos@alum.us.es)
- Javier Terroba Orozco (javteroro@alum.us.es)

GRUPO G1-E2-05

Versión 1.0.1

25/02/2022

# Tabla de contenidos

<b>Tabla de contenidos</b>	<b>1</b>
<b>Historial de versiones</b>	<b>1</b>
<b>Resumen ejecutivo</b>	<b>1</b>
<b>Introducción</b>	<b>1</b>
<b>Contenido</b>	<b>2</b>
Conceptos relevantes	2
Experiencia previa con las pruebas	3
<b>Conclusión</b>	<b>3</b>
<b>Bibliografía</b>	<b>3</b>

## Historial de versiones

Fecha	Versión	Descripción de los cambios
26/02/2022	1.0	Creación del documento y primera versión del mismo

## Resumen ejecutivo

¿Qué conozco acerca de los tests?

- Razones para hacer tests.
- Tipos de pruebas y diferencia en tiempo de ejecución en función de la granularidad.
- Características necesarias de los tests.
- Conceptos relacionados con los tests.

¿Qué experiencia previa tengo con los tests?

- Experiencia en pruebas unitarias y pruebas de la funcionalidad de los controladores.

# Introducción

En este reporte, realizado de forma individual por mí, hablaré acerca de mis conocimientos previos a diseño y pruebas 2 acerca de las pruebas de un sistema de información web, centrándome especialmente en los conocimientos adquiridos en otras asignaturas durante la carrera, centrándome en conceptos relevantes en el contexto de las pruebas y en mi experiencia previa a esta asignatura con las pruebas de un WIS .

Seguiré la siguiente estructura:

- Conceptos relevantes.
- Experiencia previa con las pruebas

## Contenido

### Conceptos relevantes

- La razón por la que se automatizan las pruebas es que principalmente porque en la mayoría de ocasiones realizar las pruebas es una tarea tan repetitiva que se tiende a cometer errores durante las pruebas, además de que una vez programadas son mucho más rápidas de ejecutar de lo que sería comprobarlo todo manualmente.
- Hay muchos tipos de tests, aunque los que he estudiado previamente son los unitarios.
- Hay pruebas tanto positivas como negativas, que comprueban casos en los que la aplicación tiene que ejecutarse sin problemas o tiene que manejar ciertos tipos de errores.
- Deben ser rápidos, individuales, repetibles, capaces de detectar por sí mismo si se cumple lo que se esté comprobando y además se deben crear o actualizar cuando el código se está desarrollando.
- Podemos encontrar diferentes tipos de test en función de la granularidad que presenten, donde a menor granularidad más rápido se ejecutan. Entre otros existen los test unitarios, tests de integración y test de aceptación.
- Los previamente mencionados tests unitarios son los encargados de probar funcionalidades concretas de la aplicación. Son los de menor alcance pero mayor velocidad de ejecución.
- Los test se encuentran divididos en 3 partes: Arrange (Preparación de los datos), Act (ejecución del test) y Assert (comprobación de que el resultado sea el esperado).
- Bug: Fallo en la ejecución del código de forma inesperada.

- Debug: Proceso de identificar y eliminar errores de un código.
- Fallo: Error en el código. Suelen ser la fuente de los bugs.
- Caso de prueba (test case): Conjunto de valores con los que se ejecuta un test para probar un caso concreto.
- Banco de pruebas (test suite): Conjunto de tests de la aplicación.

## Experiencia previa con las pruebas

En cuanto a mi experiencia previa a esta asignatura con las pruebas, toda proviene de Diseño y Pruebas 1. Principalmente aprendí a realizar las pruebas unitarias para la aplicación que desarrollamos, lo cual me sirvió para darme cuenta de lo útiles que son ya que pudimos encontrar diversos fallos en la aplicación que sin estas pruebas es muy probable que no hubiésemos podido encontrar, y con ello tampoco los podríamos haber arreglado. Además de esto aprendí a comprobar si estaban bien hechas las pruebas y a comprobar qué partes del código estaban siendo probadas y cuáles no gracias a la funcionalidad de ejecutar los tests con cobertura. También nos enseñaron cómo realizar pruebas parametrizadas aunque no tengo demasiada experiencia con este tipo. Además de esto aprendimos a utilizar Stubs y Mocks, especialmente útiles al hacer los test de los controladores de la aplicación.

## Conclusión

En la parte de contenido he incluido los conceptos que considero de mayor relevancia respecto a las pruebas de un sistema de información web, centrándome un poco más en las pruebas unitarias que como he comentado anteriormente son en las que más se han centrado las asignaturas que he dado y por ello en las que tengo mayor experiencia y de las que tengo mayores conocimientos. En cuanto al resto de tipos de pruebas conozco los conceptos generales de las mismas pero desconozco los detalles concretos. Además de los conceptos ya comentados he incluido la experiencia que tengo con las pruebas, que como he comentado provienen de DP1 donde aprendí a crear pruebas unitarias y a emplearlas para depurar mi código. Viendo la utilidad de las mismas espero que el resto de pruebas sean similares ya que ayudan a la hora de desarrollar la aplicación. Como digo parto de una base de conocimiento de pruebas unitarias pero con ninguna experiencia en el resto de tipos de pruebas.

## Bibliografía

Para desarrollar este informe me he basado en los conocimientos que he adquirido en asignaturas previas y en los apuntes de DP1.