Informe del testing WIS del grupo E3.08

# Portada

Número de grupo: E3.08

URL del repositorio: <https://github.com/antsermen/Acme-Toolkits-E3.08.git>

Miembros (+ emails):

- José Ramón Arias Expósito ([jrae1112jrae@gmail.com](mailto:rae1112jrae@gmail.co)),

- Manuel Carnero Vergel ([mancarver1@alum.us.es](mailto:mancarver1@alum.us.es)),

- Juan Carlos Moreno Pérez ([morenoperezjuancarlos@gmail.com](mailto:morenoperezjuancarlos@gmail.com)),

- Pablo Santos Pérez ([pablo.santos.232001@gmail.com](mailto:pablo.santos.232001@gmail.com)),

- David Sabugueiro Troya ([dsabugueiro14@gmail.com](mailto:dsabugueiro14@gmail.com)),

- Antonio Roberto Serrano Mena([robertoserranomena@gmail.com](mailto:robertoserranomena@gmail.com))

Fecha: 28/02/2022

# Tabla de contenidos

[**1.** Portada 1](#_Toc97149272)

[**2.** Tabla de contenidos 2](#_Toc97149273)

[**3.** Resumen ejecutivo 3](#_Toc97149274)

[**4.** Tabla de revisión 3](#_Toc97149275)

[**5.** Introducción 4](#_Toc97149276)

[**6.** Contenidos 4](#_Toc97149277)

[**7.** Conclusiones 5](#_Toc97149278)

[**8.** Bibliografía 5](#_Toc97149279)

# Resumen ejecutivo

En este documento podemos encontrarnos con 8 apartados: una portada, que contiene el número de grupo, la URL al repositorio en GitHub, los miembros (junto a sus emails) y la fecha de creación del documento; una tabla de contenidos, que sería un índice típicamente conocido para un documento; un resumen ejecutivo, en el que se resumen los contenidos de cada apartado del documento; una tabla de revisión, formada por los datos de las revisiones realizadas sobre el documento (número de revisión, fecha y descripción); una introducción, que resume aquellos conocimientos que tenemos respecto al testing de un WIS; los contenidos del documento, como su propio nombre indica, recoge los conceptos que hemos considerado relevantes a la hora de hablar del testing de un WIS; unas conclusiones, extraídas a partir de la finalización del documento y una bibliografía, que contiene las fuentes que nos sirvieron de apoyo para el documento (en caso de no encontrarse ninguna, aparecerá “Intencionalmente en blanco.”).

# Tabla de revisión

|  |  |  |
| --- | --- | --- |
| **N. º** | **Fecha** | **Descripción** |
| 1 | 28/02/2022 | Revisión de los contenidos iniciales del documento. |
| 2 | 02/03/2022 | Revisión del documento completo para su puesta a punto para ser entregado. |

# Introducción

Nuestros conocimientos sobre el testing de un WIS son más limitados aún que sobre sus arquitecturas, por lo que seremos más breves si cabe a la hora de hablar de ello.

# Contenidos

Haciendo un repaso rápido a todo lo relacionado con pruebas de WIS previamente no hemos encontrado realmente mucha información al respecto, por lo que hablaremos brevemente sobre lo que sabemos sobre el “testing” tras haber cursado la asignatura Diseño y Pruebas I.

Con respecto al “testing”, cada prueba verifica que un aparte del software, el sujeto bajo prueba (SUT), trabaja apropiadamente. Un conjunto de pruebas implementadas para una aplicación es un banco de pruebas.

Las pruebas no pueden ser exhaustivas. Las pruebas del programa se pueden utilizar para mostrar la presencia de errores, pero nunca para mostrar su ausencia.

**Pruebas unitarias**

El sujeto bajo prueba es una unidad. La mayoría de las pruebas en una suite de pruebas deben ser pruebas unitarias. Tienen la granularidad más baja (alcance más estrecho) de todas las pruebas. Deben correr muy rápido. Por lo general, los colaboradores externos son reemplazados por dobles de prueba, especialmente si se trata de bases de datos, archivos o internet.

Una buena práctica es implementar una clase de pruebas por cada clase de producción. Estructura de un test:

· **Arrange**: Configurar los datos de prueba.

· **Act**: Ejecutar sujeto bajo prueba.

· **Assert**: Afirmar que se devuelven los resultados esperados.

Los datos de prueba utilizados para organizar un test se denominan normalmente *fixture*.

Cuando da un resultado positivo significa que todo funciona como era de esperar.

Cuando da un resultado negativo hay que comprobar que maneja los errores adecuadamente.

# Conclusiones

Al prácticamente no tener idea del testing de un WIS, este apartado realmente no tiene mucho más fundamento. Pero, en resumen, conocemos globalmente cómo se llevan a cabo pruebas unitarias gracias a la asignatura de “Diseño y Pruebas I”, y si el testing de un WIS es similar a ello, no nos supondrá mucha dificultad el seguir esta parte de la asignatura.

# Bibliografía

Intencionalmente en blanco.