

# Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

## Entregable 5

### Conocimiento adquirido sobre testing



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2021 – 2022

**Grupo:** E6.02

**Repositorio:** <https://github.com/INoelia/Acme-Toolkits>

**Nombre:** Cruz Duárez, Sara

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

**Nombre:** Delgado Sánchez, José María

**Email:** josedelsan9@alum.us.es

**Nombre:** López Durán, Noelia

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

**Nombre:** Molina Arregui, Rosa María

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


**Nombre:** Nadal García, Ricardo

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

**Nombre:** Varela Soult, Carlos


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

**Fecha:** Sevilla, Mayo 29, 2022.

	<p>Diseño y Pruebas 2</p> <p>Entregable 5</p>
---	---

## Índice

Resumen ejecutivo	3
<b>Tabla de revisiones</b>	<b>3</b>
Introducción	3
Contenido	4
Testing end-to-end	4
Cobertura	4
Cobertura de caminos	4
Cobertura de datos	4
Testing de rendimiento	5
Conclusión	5
Bibliografía	6

	<p>Diseño y Pruebas 2 Entregable 5</p>
---	--

## 1. Resumen ejecutivo

Tras el desarrollo de la asignatura y finalizar el proyecto, hemos adquirido un mayor conocimiento sobre el testing en una arquitectura WIS.

Ha sido de suma importancia seguir aprendiendo sobre el testing en una arquitectura WIS para poder desarrollar el proyecto con mayor agilidad y aumentar la optimización del proyecto. Esto se resume en una reducción de los costes, del tiempo empleado y del riesgo de aparición de errores.

## 2. Tabla de revisiones

Fecha	Nº de revisión	Descripción
29/05/2022	1	Durante una reunión, todos los miembros del grupo leímos el documento, cambiamos aquello en lo que no estábamos de acuerdo y dimos por finalizado el documento.


## 3. Introducción

Entre la taxonomía de testing que se ha presentado en la asignatura se ha enfocado en el testing end-to-end, donde el objetivo principal es que el software sea probado como una simulación del usuario interactuando con la aplicación mientras que la máquina prueba que todas las acciones se realicen de acuerdo a lo esperado.

Para implementar dicha tipología de testing, se han implementado casos de prueba positivos, negativos y de hacking para cada una de las funcionalidades que contiene la aplicación. Estos casos de prueba consisten en ficheros que utilizan Gecko Driver para simular una experiencia real del usuario interactuando con el navegador Firefox y donde se comprueban resultados devueltos por la aplicación para asegurarse que se devuelven los resultados esperados.

Por otra parte, se ha estudiado el testing de rendimiento. La aproximación de esta topología es hacer un análisis estadístico de las peticiones de los tests para asegurar que el sistema responde en un tiempo adecuado respecto al número de peticiones realizadas y las características técnicas del usuario que las realiza.

Para desarrollar dicha tipología de testing, se estudian los ficheros de log de “request logs” y “test-case logs”, los cuales se encuentran en la carpeta de logs del proyecto y se inspeccionan en Excel para obtener sendos gráficos que devuelven la media en segundos de las peticiones y los casos de prueba respectivamente.

	<p>Diseño y Pruebas 2 Entregable 5</p>
---	--

## 4. Contenido

### ❖ Testing end-to-end

El objetivo de este tipo de test es el de comprobar el funcionamiento de la aplicación de una forma “natural” tal y como el usuario interactúa con la aplicación y con la interfaz web a la misma vez que se comprueba que el sistema actúa de forma esperada.

Los casos de test consisten en scripts que usan Gecko Driver para simular cómo interactúa un usuario con la interfaz usando Firefox, haciendo clic en formularios , rellenando datos y enviando datos además de obteniendo los datos resultados y comprobando que los obtenidos son los datos que se esperaban

### ❖ Cobertura

#### ➤ Cobertura de caminos

Es el porcentaje de caminos recorridos por los tests ejecutados. El objetivo es recorrer el máximo de caminos posibles. Los caminos de ejecución se clasifican como secuencias, condicionales y bucles.


- Secuencias, asegurar que cada secuencia de instrucciones del código es ejecutada al menos una vez.
- Condicionales, asegurar que cada condición se ejecuta al menos una vez.
- Bucles, asegurar que cada bucle se ejecuta cero, una y varias veces.

#### ➤ Cobertura de datos

Se puede abarcar desde puntos de vista diferentes: datos de muestra, datos de listas y datos de formulario.

- Datos de muestra, son los datos que usamos para poblar la base de datos. Estos datos deben ser consistentes con el modelo de dominio que se extrae de los requisitos.
- Datos de lista, son datos de los datos de muestra presentados en formato lista. Pueden contener atributos derivados o caracterizaciones del formato i18n que no están incluidos en los datos de muestra.
- Corresponde con los datos de muestra presentados en formato formulario. Pueden contener atributos derivados o caracterizaciones del formato i18n que no están incluidos en los datos de muestra.

Se debe comprobar que los atributos con valor nulo se muestran correctamente y que se permite la navegación a las páginas de detalles si fuera necesario.

	<p>Diseño y Pruebas 2 Entregable 5</p>
---	--

## ❖ Testing de rendimiento

La prueba de rendimiento es una técnica de prueba de software no funcional que determina cómo la estabilidad, la velocidad, la escalabilidad y la capacidad de respuesta de una aplicación se mantiene bajo una determinada carga de trabajo. Es un paso clave para asegurar la calidad del software, pero desafortunadamente, a menudo se ve como una reflexión posterior, en aislamiento, y para comenzar una vez que se completan las pruebas funcionales, y en la mayoría de los casos, después de que el código está listo para ser distribuido.

Los requisitos de rendimiento se expresan comúnmente como requisitos no funcionales. Normalmente el cliente establece un tiempo máximo para atender las solicitudes al sistema.

Un fracaso en el test es una desviación del resultado esperado. Verificar si los datos de rendimiento se desvían de las expectativas requiere realizar un análisis estadístico en un nivel de confianza determinado y es lo que conseguimos con los test de rendimiento. Un análisis estadístico en un nivel de confianza común consiste en inferir un intervalo para la media poblacional con un nivel de confianza dado. Este análisis se utiliza para verificar si un proyecto cumple con un requisito de rendimiento determinado. Otro análisis común que hemos realizado consiste en comparar dos muestras para determinar si las medias de sus poblaciones correspondientes pueden considerarse iguales o no en un nivel de confianza dado. Este análisis se utiliza para comprobar si una refactorización da como resultado un mejor rendimiento o no pero nosotros lo hemos usado para comparar el rendimiento de dos de nuestros ordenadores.


## 5. Conclusión

Aunque es cierto que hay algunas nociones sobre el testing que traíamos ya sabiendas de otras asignaturas, como la cobertura de caminos, había otras muchas que era la primera vez que las veíamos, por ejemplo las pruebas con Gecko driver, que nos permiten simular que somos un usuario pulsando en la web, es la primera vez que las vemos y nos ha parecido interesante ya que ver exactamente con que se está interactuando es útil a la hora de encontrar errores o para debuggear.

También ha sido una novedad el análisis de rendimiento y el uso de excel para hacer todos los cálculos, dándonos una mayor vista sobre los distintos ámbitos a tener en cuenta en un proyecto de software..

## 6. Bibliografía

L03-S03 Functional Testing

	<div>Diseño y Pruebas 2</div> <div>Entregable 5</div>
---	---

L03-S05 Performance Testing