	Diseño y Pruebas 2 Documentación de la entrega D02
	Planificación Individual

# Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

## Documentación de la entrega D04

### Testing



Grado en Ingeniería Informática – Ingeniería del Software


Diseño y Pruebas 2

Curso 2023 – 2024

Fecha	Versión
27/05/2024	v1r1


Grupo de prácticas: G6-64		
Autores por orden alfabético	Rol	Correo electrónico
Aguayo Orozco, Sergio - 25604244T	Desarrollador	ahydul1@gmail.com
García Lama, Gonzalo - 47267072W	Desarrollador, Tester	gongarlam@alum.us.es
Huecas Calderón, Tomás - 17476993Y	Desarrollador	tomhuecal@alum.us.es
Fernández Pérez, Pablo - 54370557Y	Desarrollador, Analista	pablofp.33@gmail.com
Youssafi Benichikh, Karim - 28823709V	Desarrollador, operador, manager	karyouben@alum.us.es

Repositorio: <https://github.com/karyouben/Acme-SF-D04>

	<p>Diseño y Pruebas 2 Documentación de la entrega D02</p>
	<p><b>Planificación Individual</b></p>


## Índice de contenido

1. Resumen ejecutivo3
2. Control de versiones4
3. Introducción5
4. Contenido6
5. Conclusiones6
6. Bibliografía20

	<p>Diseño y Pruebas 2 Documentación de la entrega D02</p>
	<p><b>Planificación Individual</b></p>


## 1. Resumen ejecutivo

El documento de pruebas está dividido en dos apartados. El primero está centrado en pruebas funcionales y el segundo está centrado en pruebas de rendimiento.

	Diseño y Pruebas 2 Documentación de la entrega D02
	<b>Planificación Individual</b>

## 2. Control de versiones


Fecha	Versión	Descripción
20/05/2024	v1r0	Creado el documento
27/05/2024	v2r0	Terminado el documento

	Diseño y Pruebas 2 Documentación de la entrega D02
	<b>Planificación Individual</b>

### 3. Introducción

A continuación, mostramos los resultados de las pruebas. El primer apartado incluye un listado con los casos de prueba implementados, agrupados por implementación, con descripción sobre su efectividad a la hora de encontrar errores.

El segundo apartado incluye información sobre el rendimiento de la aplicación. Gráficas e intervalos de confianza 95% tomados en dos ordenadores distintos, además de un contraste de hipótesis de confianza respecto a qué ordenador es más potente.

	Diseño y Pruebas 2 Documentación de la entrega D02
	<b>Planificación Individual</b>

## 4. Contenido

No se ha implementado ningún índice porque el framework crea los índices para id y los atributos de navegación automáticamente, no hay otro tipo de índice que haya que implementar según los queries de los repositorios relevantes al proyecto, historias de usuarios o asociaciones.

Además, en el create al haber probado mas de 50 datos en solo esa feature al ejecutar todos los test a veces eclipse se ralentiza o se queda pillado y no corre bien el create, por lo que habría que ejecutarlo de nuevo

Si hay algún tipo de error puede ser porque algún compañero haya tocado sus csv. Mi rama es karyouben y funciona perfectamente todo ahí:

<https://github.com/karyouben/Acme-SF-D04.git>


Se añaden también los csv usados para la metodología de datos probados y hacking en todos los tests:

(hoja de cálculo de la metodología usada en testing)

[https://docs.google.com/spreadsheets/d/1AjAxVLVrZ2YukrSqX\\_RwuzNsiU\\_QYZ8VvQnXHhI2X7Y/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1AjAxVLVrZ2YukrSqX_RwuzNsiU_QYZ8VvQnXHhI2X7Y/edit?usp=sharing)

(hoja de cálculo de la metodología usada en hacking)

<https://docs.google.com/spreadsheets/d/1dCT-zC0ePqIXSqy1jAUUZAcXpZhy07yZKieTTbzIRjw/edit#gid=0>

	<p style="text-align: center;">Diseño y Pruebas 2 Documentación de la entrega D02</p>
	<p style="text-align: center;"><b>Planificación Individual</b></p>

## Pruebas Funcionales

### TrainingModule


```

79     }
80   }
81
82   Multiple markers at this line
83   - +1
84   - +1
85   - 1 of 4 branches missed.
86   if (startBeforeCreation) {
87     final boolean createOneWeekBeforeStartMinimum = MomentHelper.isLongEnough(object.getTrainingModule().getCreationMoment(), object.getStartPeriod(), 7, ChronoUnit.DAYS);
88     super.state(createOneWeekBeforeStartMinimum, PERIOD_START, "developer.trainingSession.form.error.start-before-creation");
89   }
90 }
91
92

```


En las restricciones de fechas del creation/update del training sección sale el coverage amarillo porque una de las ramas no se puede probar, pues el creation moment es una fecha que se crea por defecto por el sistema y dichas restricciones están hecha para un doble check del mismo.

Create.safe	<p>Se inicia sesión como developer1 y se prueba a crear “Módulos de entrenamiento “de todas las formas que permite el sistema, comprobando los límites y los mensajes de error. No se detecta ningún error.</p> <p>Se sigue la metodología de clase de ir probando todos los campos en blanco, los errores de cada campo y después probar un gran número de datos positivos, en mi caso he probado mas de 30 datos aproximadamente asegurando los límites.</p>
Delete.safe	<p>Se inicia sesión como developer1 y se elimina un proyecto que tiene hijos publicados para que salte la restricción y otro que tiene hijos no publicados el cual se borra satisfactoriamente.</p>


	<p>Diseño y Pruebas 2 Documentación de la entrega D02</p>
	<p><b>Planificación Individual</b></p>

List.safe	<p>Se inicia sesión como developer1 y se listan sus “Módulos de entrenamiento” en inglés y en español. No se detecta ningún error.</p>
Update.safe	<p>Se inicia sesión como developer1 y se prueba a actualizar “Módulos de entrenamiento “en especial el de id 650 de todas las formas que permite el sistema, comprobando los límites y los mensajes de error. No se detecta ningún error.</p> <p>Se sigue la metodología de clase de ir probando todos los campos en blanco, los errores de cada campo y después probar un gran número de datos positivos, en mi caso he probado más de 30 datos aproximadamente asegurando los límites.</p>
Publish.safe	<p>Se inicia sesión como developer1 y se publica un “Modulo de entrenamiento”. Se prueba a publicar el “Modulo de entrenamiento” cuando no tiene “trainning Sesións” asociadas para que salte la restricción, y cuando hay algún training sección sin publicar para que también, salte finalmente se prueba a publicar un “training Module” con todos sus hijos publicados y se publica satisfactoriamente. No se detecta ningún error.</p> <p>Como el publish también actualiza los datos, se ha seguido la misma metodología que en el créate/update, de crear todos los valores posibles tanto positivos como negativos siguiendo la metodología de clase.</p>



	Diseño y Pruebas 2 Documentación de la entrega D02
	<b>Planificación Individual</b>


Show.safe	Se inicia sesión como developer1, se listan su training modules y se muestra cada training Module uno a uno. No se detecta ningún error.
Delete.hack	Primero se hace get de /developer/trainingModule/delete con id 656 (¿id=656), con id 1503 con usuario anónimo. Luego se inicia sesión como developer2 y se hace lo mismo. Finalmente se inicia sesión como developer1 y se hace hacking para id 656 (publicado) y el id 1583. No se detecta ningún problema.
List.hack	Se hace /developer/trainingModule/list con usuario anónimo. No se detecta ningún problema.
Update.hack	Se hace get de /developer/trainingModule/update con id 650 (no está publicado) y con id 6503 (no existe) con usuario anónimo. Se hace lo mismo como developer2 y se crea un trainingModule para hackear con la consola para el id 656 (del developer1) y 6503. Finalmente se inicia sesión como developer1 y se intenta actualizar el proyecto con id 656 (ya publicado) y 6503 (no existe). No se detecta ningún problema.
Publish.hack	Se hace get de /developer/trainingModule/publish con id 656 (no está publicado) y con id 6503 (no existe) con usuario anónimo. Se hace lo mismo como developer2 y se crea un proyecto para hackear con la consola para el id 656 (del developer1) y 6503 Finalmente se inicia sesión como developer1 y se intenta

	Diseño y Pruebas 2 Documentación de la entrega D02
	<b>Planificación Individual</b>


	actualizar el proyecto con id 656 (ya publicado) y con un id inexistente 6503. No se detecta ningún problema.
Show.hack	Se hace get /developer/trainingModule/show con id 656 y 1654 con usuario anónimo. Se inicia sesión como developer2 y se hace lo mismo. No se detecta ningún problema

### Training Sesión


Create.safe	<p>Se inicia sesión como developer1 y se prueba a crear training sessions de todas las formas que permite el sistema, comprobando los límites y los mensajes de error. No se detecta ningún error.</p> <p>Se sigue la metodología de clase de ir probando todos los campos en blanco, los errores de cada campo y después probar</p>
-------------	--

	Diseño y Pruebas 2 Documentación de la entrega D02
	<b>Planificación Individual</b>

	<p>un gran número de datos positivos, en mi caso he probado más de 30 datos aproximadamente asegurando los límites.</p>
Delete.safe	<p>Se inicia sesión como developer1 y se elimina un training sesión en draft mode</p>
List.safe	<p>Se inicia sesión como developer1 y se listan los trainings sessions. No se detecta ningún error.</p>
List-by-trainingModule.safe	<p>Se inicia sesión como developer1 y se listan los trainings sessions asociadas a cada trainingModule. Se listan todos en ingles y algunos en español.</p>
Publish.safe	<p>Se inicia sesión como developer1 y se publican y se crean varios trainings sessions, probando todos los límites de pruebas.</p> <p>Se sigue la metodología de clase de ir probando todos los campos en blanco, los errores de cada campo y después probar un gran número de datos positivos, en mi caso he probado más de 30 datos aproximadamente asegurando los límites.</p>
Show.safe	<p>Se inicia sesión como developer1, se listan su training sessions y se muestra cada una. No se detecta ningún error.</p>
Update.safe	<p>Se inicia sesión como developer1 y se prueba a actualizar los datos de varios trainings modules de todas las formas que</p>

	Diseño y Pruebas 2 Documentación de la entrega D02
	<b>Planificación Individual</b>

	<p>permite el sistema, comprobando los límites y los mensajes de error. No se detecta ningún error.</p> <p>Se sigue la metodología de clase de ir probando todos los campos en blanco, los errores de cada campo y después probar un gran número de datos positivos, en mi caso he probado más de 30 datos aproximadamente asegurando los límites.</p>
List.hack	<p>Se hace /developer/training-session/list con usuario anónimo. No se detecta ningún problema.</p>
List-by-trainingModule.hack	<p>developer/training-session/list-by-training-module?trainingModuleId=656 y 6743 con usuario anónimo. Luego se hace lo mismo iniciado sesión como developer2. No se detecta ningún problema.</p>
Publish.hack	<p>/developer/training-module/publish con ids 751 y 7224 tanto con usuario anónimo como siendo developer2. Además, se crea un training session siendo developer2 para hackear usando la consola pretendiendo publicar los ids 751 (del developer1), 7224 (no existe) y 126 (no publicado del developer2). No se detecta ningún problema.</p>
Delete.hack	<p>/developerr/training-session/delete con ids 724 y 16342 tanto con usuario anónimo como siendo developer2. Además, se crea un training sesión siendo developer2 para hackear usando la consola pretendiendo eliminar los ids 724 (del developer1), 16342</p>

	Diseño y Pruebas 2 Documentación de la entrega D02
	<b>Planificación Individual</b>

	(no existe) y 722 (no publicado del developer1). No se detecta ningún problema.
Show.hack	Se hace get /developer/training-session/show con id 721 y 1673 siendo usuario anónimo. Se inicia sesión como developer2 y se hace lo mismo. No se detecta ningún problema
Update.hack	Se hace get /developer/training-session/update con ids 656 y 6503 tanto con usuario anónimo como siendo developer2. Se crea un training session para hackear usando la consola e intentar actualizar los ids 650, 656 y 6503. No se detectó ningún problema.



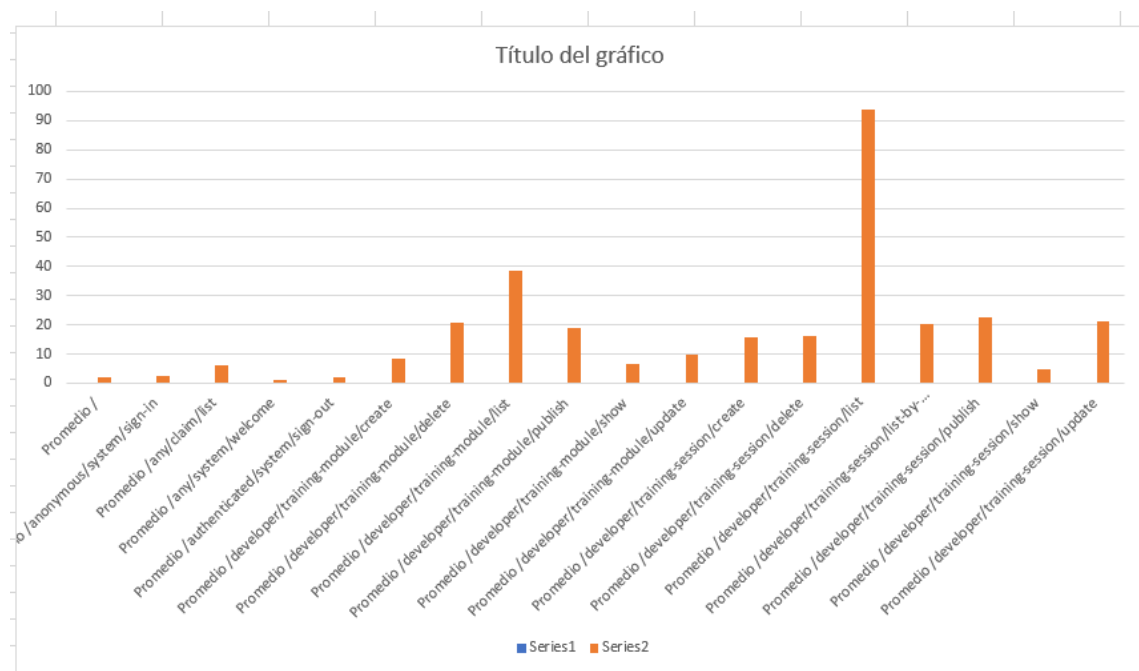
## Diseño y Pruebas 2 Documentación de la entrega D02


### Planificación Individual

## Pruebas de Rendimiento

### PC A

request-path	response-status	time
Promedio /		2,206714355
Promedio /anonymous/system/sign-in		2,550179134
Promedio /any/claim/list		6,1671
Promedio /any/system/welcome		1,310933278
Promedio /authenticated/system/sign-out		1,9625401
Promedio /developer/training-module/creat		8,580781188
Promedio /developer/training-module/delet		20,76382682
Promedio /developer/training-module/list		38,36321884
Promedio /developer/training-module/publi		19,09861429
Promedio /developer/training-module/show		6,408042
Promedio /developer/training-module/upda		9,770155808
Promedio /developer/training-session/creat		15,63511535
Promedio /developer/training-session/delet		16,048843
Promedio /developer/training-session/list		93,72016667
Promedio /developer/training-session/list-b		20,2927228
Promedio /developer/training-session/publi		22,70182182
Promedio /developer/training-session/show		4,576604418
Promedio /developer/training-session/upda		21,27351234
Promedio general		14,72902632



	Diseño y Pruebas 2 Documentación de la entrega D02
	<b>Planificación Individual</b>

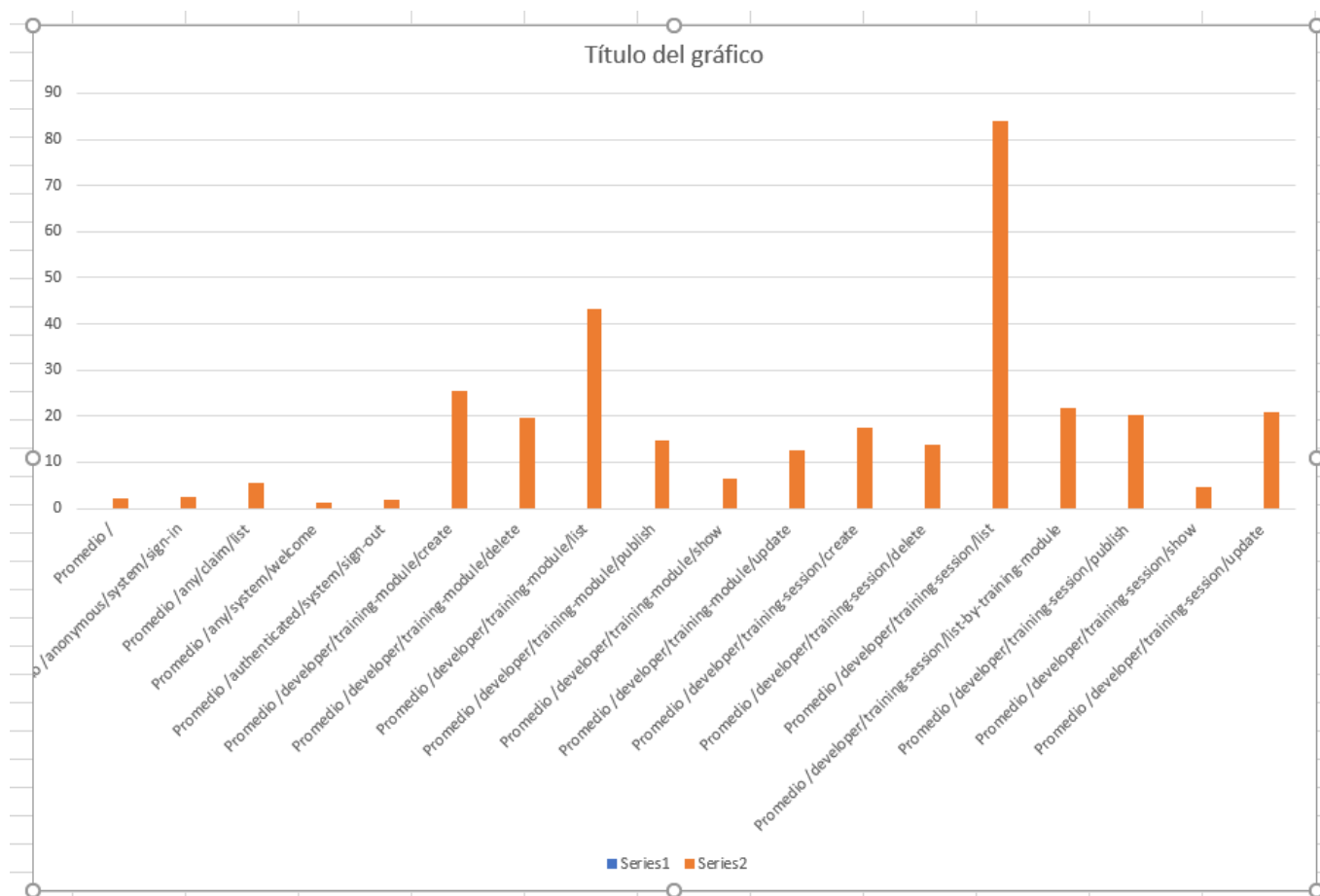
## PC B

net request-path	response-status	time
Promedio /		2,20607541
Promedio /anonymous/system/sign-in		2,42741692
Promedio /any/claim/list		5,5981
Promedio /any/system/welcome		1,22067571
Promedio /authenticated/system/sign-out		1,84986667
Promedio /developer/training-module/create		25,38845
Promedio /developer/training-module/delete		19,6298077
Promedio /developer/training-module/list		43,223799
Promedio /developer/training-module/publish		14,7564294
Promedio /developer/training-module/show		6,48276957
Promedio /developer/training-module/update		12,7187481
Promedio /developer/training-session/create		17,4113254
Promedio /developer/training-session/delete		13,8189143
Promedio /developer/training-session/list		84,1259333
Promedio /developer/training-session/list-by-training-module		21,6817626
Promedio /developer/training-session/publish		20,2226959
Promedio /developer/training-session/show		4,78164032
Promedio /developer/training-session/update		20,7426377
Promedio general		15,4097073




## Diseño y Pruebas 2 Documentación de la entrega D02

### Planificación Individual



El PC B por lo general tiene tiempos más altos como se puede observar en la comparativa de las dos gráficas.




	<p style="text-align: center;">Diseño y Pruebas 2 Documentación de la entrega D02</p>
<b>Planificación Individual</b>	

	A	B	C	D	E	F
1	<b>PCA</b>					
2				interval(ms)	15,6636276	13,794425
3	Media	14,72902632			0,01566363	0,01379443
4	Error típico	0,476300655				
5	Mediana	8,296399				
6	Moda	4,4934				
7	Desviación estándar	15,49990936				
8	Varianza de la muestra	240,2471901				
9	Curtosis	11,85779655				
10	Coefficiente de asimetría	2,540482046				
11	Rango	137,1575				
12	Mínimo	0,6916				
13	Máximo	137,8491				
14	Suma	15598,03888				
15	Cuenta	1059				
16	Nivel de confianza(95,0%)	0,934601303				
17						

	<b>PCB</b>					
				interval(ms)	16,4587459	14,3606688
	Media	15,40970735			0,01645875	0,01436067
	Error típico	0,534553097				
	Mediana	10,9262				
	Moda	3,854				
	Desviación estándar	16,50202726				
	Varianza de la muestra	272,3169038				
	Curtosis	11,01387085				
	Coefficiente de asimetría	2,505571398				
	Rango	141,9693				
	Mínimo	0,6247				
	Máximo	142,594				
	Suma	14685,4511				
	Cuenta	953				
	Nivel de confianza(95,0%)	1,049038527				


El PCA tiene intervalo de confianza 95% (15,66, 13,79) y el PCB tiene intervalo (16,45, 14,36). Son intervalos más que aceptables.

A continuación, calculamos la hipótesis de contraste con 95% confianza para intentar averiguar que ordenador es más potente:

	Diseño y Pruebas 2 Documentación de la entrega D02
Planificación Individual	


A	B	C
Prueba z para medias de dos muestras		
	PCA	PCB
	62,224101	65,2397
Media	14,6841349	15,357365
Varianza (conocida)	240,24719	272,3169
Observaciones	1058	952
Diferencia hipotética de las medias	0	
z	-0,9398365	
P(Z<=z) una cola	0,17365071	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,34730142	
Valor crítico de z (dos colas)	1,95996398	

Como podemos observar, el valor P es menor que 0.2 Por esta razón sabemos que comparar las medias de los tiempos es una buena manera de averiguar que ordenador es más potente. En este caso PCA es mejor al tener una media de tiempos menor. Ese ordenador tiene un RYZEN 9000 y una 4090, el cual es bastante moderno, mientras que PCA tiene una CPU algo más antigua y bastante menos potente un RYZEN 7000, además de tener en cuenta que el PCA es un pc de sobremesa y PCB es un portátil.

	<p>Diseño y Pruebas 2 Documentación de la entrega D02</p>
	<p><b>Planificación Individual</b></p>

## 5. Conclusiones

Este documento recoge las pruebas realizadas los cuales han servido para encontrar algún error en el código y calcular el rendimiento de nuestro código compara con dos ordenadores.

	<p>Diseño y Pruebas 2 Documentación de la entrega D02</p>
	<p><b>Planificación Individual</b></p>

## 6. Bibliografía

Intencionadamente en blanco