



Escuela Técnica Superior de  
**Ingeniería Informática**

## Testing report Do4

Group C1.020 | Diseño y Pruebas II | 27/05/2024

Fecha	Versión	Autor
27-05-2024	1.0	Vacaro Goytia, Gabriel Carrera Bernal, Álvaro Martín Vergara, Jesús González Quintanilla, Diego

### Miembros:

- Vacaro Goytia, Gabriel ([gabvacgoy@alum.us.es](mailto:gabvacgoy@alum.us.es))
- Carrera Bernal, Álvaro ([alvcarber1@alum.us.es](mailto:alvcarber1@alum.us.es))
- Martín Vergara, Jesús ([jesmarver@alum.us.es](mailto:jesmarver@alum.us.es))
- González Quintanilla, Diego ([diegonqui@alum.us.es](mailto:diegonqui@alum.us.es))

Repositorio de Github: <https://github.com/DP2-C1-020/Acme-SF-Do4>

Contenido

Resumen ejecutivo ..... 3

Introducción ..... 4

Contenido ..... 5

    Pruebas funcionales: ..... 5

    Análisis de desempeño: ..... 7

Bibliografía..... 11

## Resumen ejecutivo

El propósito de este informe es ofrecer una descripción detallada de los diferentes procedimientos seguidos que deben considerarse en el ámbito del testing formal del proyecto Acme-SF, desde la generación de suficientes datos de ejemplo, hasta la meticulosa comprobación de la correcta implementación de los requisitos funcionales y las distintas herramientas estadísticas para comparar los tiempos de ejecución entre peticiones, por último la comparación de un mismo ordenador (antes y después de la mejora de la base de datos mediante índices).

En resumen, se ha utilizado un enfoque diligente para abordar y solucionar los errores encontrados durante el proceso de testeo con la finalidad de asegurar un producto de alto nivel con el que se satisfagan las expectativas del cliente y se han generado documentos que lo avalan.

## Introducción

En este tipo de informe, se pretende realizar una explicación de como se ha realizado el testing funcional y el análisis de desempeño, con el objetivo de entregar un proyecto de alto nivel con garantías. Para lograrlo, se lleva a cabo una serie de pruebas en relación con el requisito funcional grupal 11 “Operaciones sobre Banners”.

Este escrito se ha organizado según el documento de anexo proporcionado en enseñanza virtual, en primer lugar, con una portada con las credenciales de los autores del reporte, una tabla de versiones en la que se especifican las modificaciones realizadas en este documento clasificadas por número y con sus fechas respectivas; seguidamente un resumen ejecutivo el cual pretende poner en contexto al lector sobre el contenido del documento, una introducción al documento donde se describe el contenido de forma sucinta y se trata la estructura del documento en este último párrafo introductorio.

El trabajo presenta dos apartados clave: uno sobre pruebas funcionales y otro sobre pruebas de rendimiento. En el primero, se detallan los casos de prueba implementados, organizados por característica, con evaluaciones sobre su efectividad para detectar errores. En el segundo, se proporcionan gráficos pertinentes y un intervalo de confianza del 95% para el tiempo de respuesta del proyecto ante las solicitudes funcionales y la comparación pertinente. Estos capítulos constituyen una evaluación exhaustiva de la calidad y eficiencia del proyecto en desarrollo, por último, una conclusión del analista y la bibliografía utilizada.

## Contenido

### Pruebas funcionales:

Para cada característica se han realizado tanto pruebas positivas y negativas (X.safe) como intentos de hacking (Y.hack), para la realización de estas pruebas, se ha seguido las recomendaciones dictadas en el documento “L04-S01 – Formal testing”:

Para no caer en redundancias, listaré las pruebas realizadas según el tipo de operación, pues esto será lo que determinará el modo de actuar de la prueba:

### OPERACIONES "LIST-MINE"

#### *Pruebas Seguras:*

- Listar todos los banners desde todos los administradores.

#### *Pruebas de Hacking:*

- Intentar acceder a la URL con un rol incorrecto (manager, auditor, sin rol, etc...).

### OPERACIONES "SHOW"

#### *Pruebas Seguras:*

- Seleccionar varios banners de una lista para ver sus detalles.

#### *Pruebas de Hacking:*

- Intentar acceder a los detalles de un banner con un rol incorrecto.
- Intentar acceder a un banner inexistente con un identificador inválido.

### OPERACIONES "CREATE"

#### *Pruebas Seguras:*

- Intentar crear un nuevo Banner.
- Verificar que el sistema rechace datos inválidos para cada atributo (incoherencias con fechas, campos demasiado largos, formatos incorrectos entre otros)..
- Verificar que el sistema acepte datos válidos para cada atributo.

#### *Pruebas de Hacking:*

- No se encontraron formas de hacking.

## OPERACIONES "UPDATE"

### *Pruebas Seguras:*

- Actualizar un Banner existente.
- Verificar que el sistema rechace datos inválidos para cada atributo (incoherencias con fechas, campos demasiado largos, formatos incorrectos entre otros).
- Verificar que el sistema acepte datos válidos para cada atributo.

### *Pruebas de Hacking:*

- No se encontraron formas de hacking.

## OPERACIONES "DELETE"

### *Pruebas Seguras:*

- Eliminar correctamente un Banner cuando es posible.

### *Pruebas de Hacking:*

- Intentar acceder al borrado de un Banner sin tener los permisos del mismo mediante URLs

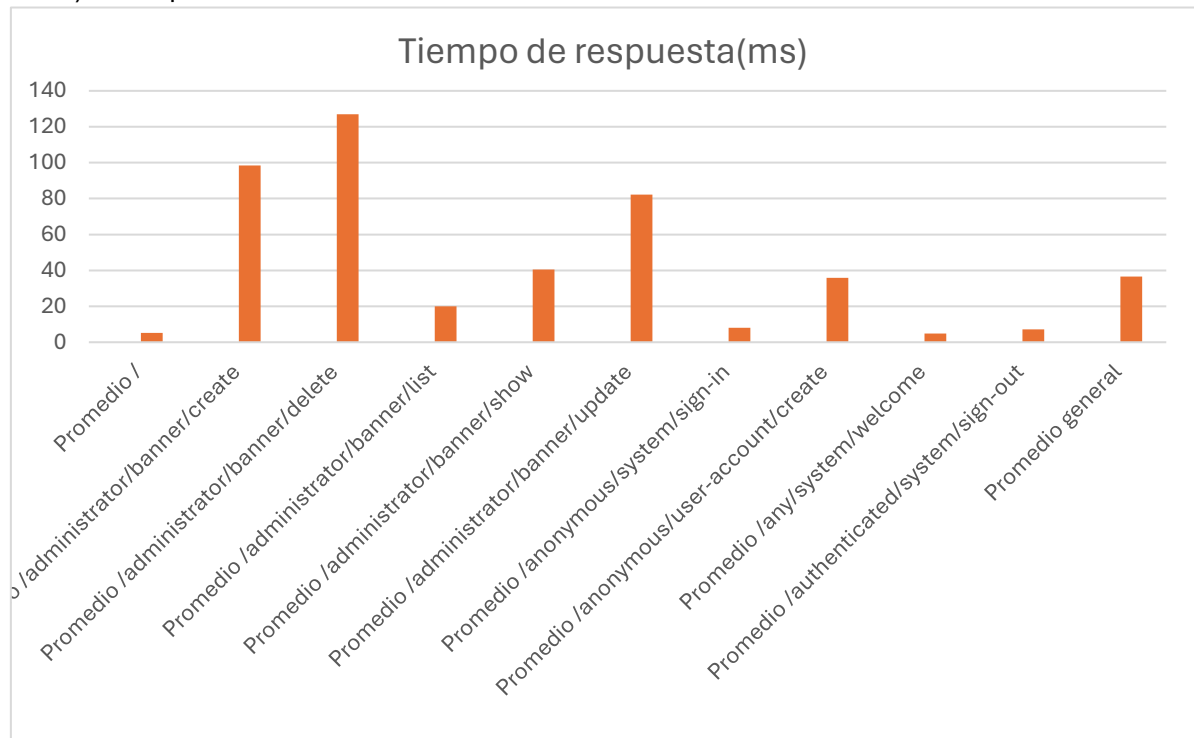
Por último, véase aquí las pruebas implementadas y una categorización sobre su efectividad en esta última fase del proyecto y el coverage de cada feature.

Ruta de la feature	X.safe	Y.hack	Efectividad	Coverage
/administrator/Banner/...	List Show Create Update Delete	List Show Delete	Alta, Se encontraron fallos en el ámbito de validación de las fechas, además se descubrió que existía un máximo número de caracteres para una URL, esto mejoró la robustez de la aplicación	87.8 %

## Análisis de desempeño:

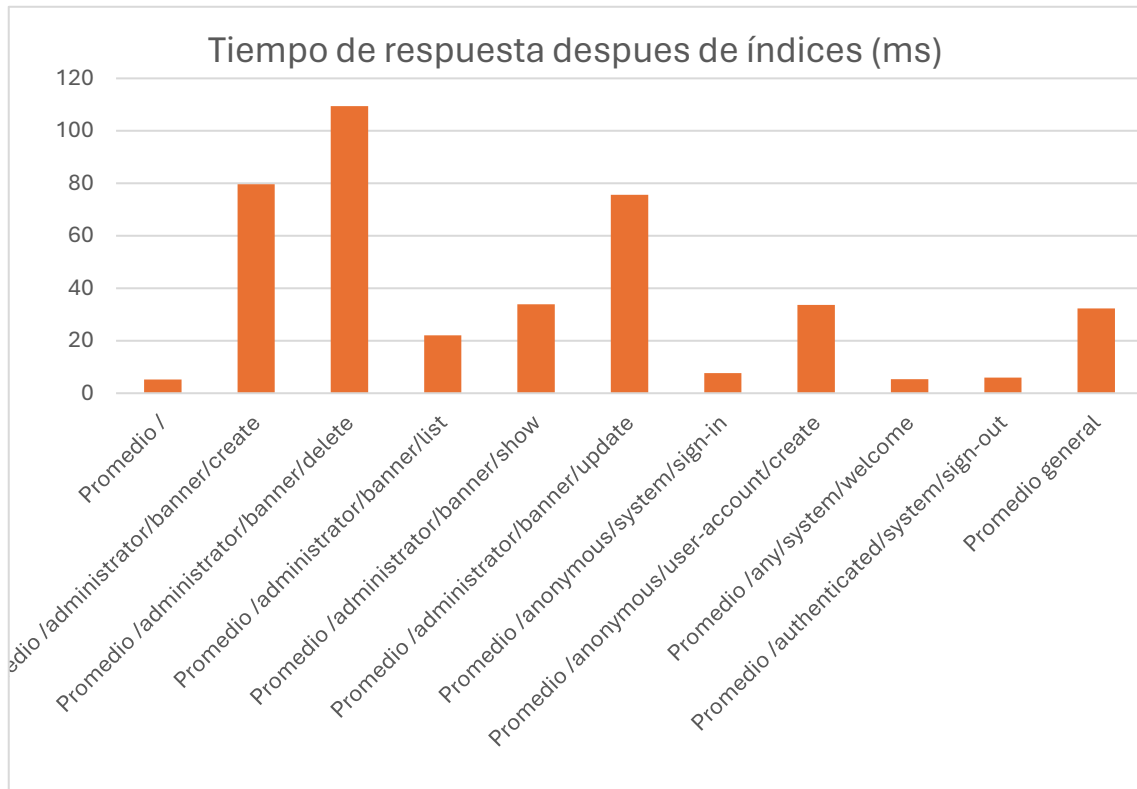
En este apartado veremos las estadísticas propias de la aplicación:

### 1) Sin optimizar índices:



En este caso podemos observar que el MIR es la característica de borrado de un banner, seguido de cerca de la creación, ambas características, aun siendo las más lentas, siguen estando en un valor aceptable de 100-130 ms aproximadamente, esta diferencia con respecto al resto de características (exceptuando update) se debe a la necesidad de acceder a la base de datos para poder realizar la operación, lo que genera un cuello de botella.

- 2) Tras realizar la modificación en base de datos para agilizar búsquedas mediante los índices, obtenemos este nuevo resultado:



Podemos apreciar una pequeña mejora en los métodos que más tiempo consumían debido a su necesidad de acceder a la base de datos, lo cual gracias a los índices se ha atenuado en gran medida.

En las estadísticas, también se ve reflejado en una disminución tanto de la cota superior como de la inferior del intervalo de confianza:

Time Before	Time After	Before							After							
6,6219	5,774															
10,0431	7,3602	Media	36,993163	interval(m	28,752	45,2343			Media	32,687925	Interval(n	25,7939	39,582			
6,7164	5,831	Error típico	4,1698748	interval(s)	0,02875	0,04523			Error típico	3,4882693	Interval(s)	0,02579	0,03958			
5,3095	4,4832	Mediana	9,7778						Mediana	9,918						
4,206	4,4758	Moda	35,8721						Moda	33,7006						
4,438	8,4252	Desviación estándar	50,557045						Desviación estándar	42,293017						
4,6851	4,5356	Varianza de la muestra	2556,0148						Varianza de la muestra	1788,6993						
6,229	6,2724	Curtosis	9,7507773						Curtosis	8,8116333						
4,2366	4,2082	Coefficiente de asimetría	2,5782103						Coefficiente de asimetría	2,4257568						
4,9073	5,6751	Rango	337,3203						Rango	281,9225						
3,9313	4,0669	Mínimo	3,4397						Mínimo	3,5448						
4,8278	5,1154	Máximo	340,76						Máximo	285,4673						
3,9237	3,7711	Suma	5437,995						Suma	4805,125						
5,0207	5,1438	Cuenta	147						Cuenta	147						
3,8179	3,5448	Nivel de confianza(95,0%)	8,2411139						Nivel de confianza(95,0%)	6,8940258						
4,8936	6,1475															
4,1448	3,6332															
5,17368824	5,203729412															
11,3473	8,0603															
160,6323	164,3059															
103,7243	102,0779															
242,9946	147,9996															
54,8442	53,5683															
102,4075	99,0664															



Por último, en el contraste de hipótesis, nos damos cuenta de que el p-value (0.42593574), se sitúa entre 0.05 y 1, por lo tanto, concluimos que no han sido cambios que se hayan traducido en una mejora de gran significancia, aunque podamos ver que la media es débilmente menor.

Prueba z para medias de dos muestras		
	<i>6,6219</i>	<i>5,774</i>
Media	37,1968205	32,8684425
Varianza (conocida)	2556,01	1788,7
Observaciones	147	147
Diferencia hipotética de las medias	0	
z	0,79616568	
P(Z<=z) una cola	0,21296787	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,42593574	
Valor crítico de z (dos colas)	1,95996398	

## Conclusiones

Sin esta fase de testing formal, no habiéramos sido capaces de aseverar con una rigurosidad de nivel profesional que la calidad del producto Acme-SF era alta. Esta etapa ha sido esencial no solo para identificar y corregir errores, sino también para asegurarnos doblemente de que cada aspecto de la aplicación (tanto individual como grupal) funciona según lo esperado bajo diversas condiciones. Este proceso meticuloso nos ha permitido detectar fallos críticos en las características relacionadas a Banner, y sospechamos que, en un entorno laboral real, el testing formal debería extenderse a todas y cada una de las propiedades de la aplicación, pues la probabilidad de una implementación perfecta en el primer intento es bastante baja. Errores que podrían haber pasado desapercibidos en una revisión más superficial, que, de no haber sido corregidos, podrían haberse traducido en problemas graves y potenciales fallos durante el uso por parte del cliente, afectando su experiencia y la reputación del producto.

Finalmente, el uso de comprobaciones estadísticas sobre los datos del producto, como el Z-Test y el intervalo de confianza, no solo han sido fundamentales para validar el rendimiento y la estabilidad de la aplicación, sino que también para aprender las capacidades promedio y límite de la tecnología con la que estamos trabajando, además de una vaga idea sobre su escalabilidad. Estas herramientas nos han permitido evaluar de manera objetiva si los cambios realizados han tenido un impacto significativo. El Z-Test, en particular, ha proporcionado una forma precisa de comparar los tiempos de ejecución antes y después de las modificaciones, mientras que el intervalo de confianza nos ha ofrecido una medida de la precisión de nuestras estimaciones. Estas metodologías han asegurado que los resultados obtenidos no son producto del azar, sino reflejan mejoras o consistencias reales en el rendimiento del producto.

En resumen, el testing formal no es solo una fase más del desarrollo, sino un sello de calidad que asegura un buen producto y nos proporciona información para mejorar tanto las implementaciones ya realizadas como las características futuras.

## Bibliografía

Web de la universidad de Sevilla - <https://ev.us.es>