

# Testing report

## Información general

**Autor:** Rafael Pulido Cifuentes

**Fecha:** 26/05/2024

**Grupo:** C1.006

**Repositorio:** <https://github.com/Danielruizlopezcc/Acme-SF-D04.git>

**Miembros:**

- Alberto Carmona Sicre ([albcarsic@alum.us.es](mailto:albcarsic@alum.us.es))
- Ramón Gavira Sánchez ([ramgavsan@alum.us.es](mailto:ramgavsan@alum.us.es))
- Marco Padilla Gómez ([marpadgom1@alum.us.es](mailto:marpadgom1@alum.us.es))
- Rafael Pulido Cifuentes ([rafpulcif@alum.us.es](mailto:rafpulcif@alum.us.es))
- Daniel Ruiz López ([danruilop1@alum.us.es](mailto:danruilop1@alum.us.es))

## Tabla de contenidos

- [Testing report](#)
- [Información general](#)
- [Tabla de contenidos](#)
- [Resumen ejecutivo](#)
- [Tabla de revisión](#)
- [Introducción](#)
- [Contenidos](#)
  - [Testing funcional](#)
    - AdministratorBannerListService
    - AdministratorBannerShowService
    - AdministratorBannerCreateService
    - AdministratorBannerUpdateService
    - AdministratorBannerDeleteService

- Cobertura
- Testing de rendimiento
  - Análisis del rendimiento
  - Intervalo de confianza
  - Hypothesis contrast

# Resumen ejecutivo

El proyecto Acme-SF-D01 de la asignatura Diseño y Pruebas II, es un proyecto con fines meramente educativos, con el que se busca mejorar las habilidades y trabajar como desarrolladores web. El objetivo es aprender a producir un sistema de información web típico de tamaño pequeño a mediano basándose en una especificación de requisitos informal y métodos y herramientas de potencia industrial.

# Tabla de revisión

Nº de revisión	Fecha	Descripción
001	25/05/24	Creación del documento, estructura e introducción
002	26/05/24	Testing funcional y testing de rendimiento

# Introducción

Este documento tiene como objetivo presentar el reporte de pruebas realizadas a los servicios de la clase Banner. Se mostrarán los resultados obtenidos en cada uno de los servicios probados, incluyendo los casos de prueba tanto positivos como negativos probados en cada uno de los servicios, así como también los resultados obtenidos en la cobertura de código de cada uno de los servicios.

Además, se tratarán los bugs y errores encontrados durante la ejecución de las pruebas y las soluciones que se les ha dado a los mismos.

También se verán reflejados en este documento los resultados obtenidos tras el análisis del rendimiento de los servicios, acompañados de gráficas y tablas que permitirán visualizar de manera más clara todos los resultados.

Por último, se presentarán las conclusiones obtenidas tras la realización de las pruebas y se propondrán posibles mejoras a los servicios probados.

# Contenidos

## Testing funcional

### AdministratorBannerListService

- **Safe testing:** Para el testing legal de este servicio únicamente fué necesario mostrar los banners del administrador
- **Hacking:**
  - Listado de banners sin estar logueado
  - Listado de banners estando logueado con otro rol.
- **Bugs:** No se encontraron bugs

### AdministratorBannerShowService

- **Safe testing:** Se mostraron los datos de banners creados por el administrador
- **Hacking:**
  - Mostrar datos de un banner sin haber iniciado sesión
  - Mostrar datos de un banner habiendo iniciado sesión con otro rol
- **Bugs:** No se encontraron bugs

### AdministratorBannerCreateService

- **Safe testing:**
  - Creación de banners en los que los datos eran incorrectos: Nulos en aquellos obligatorios, formato incorrecto, fechas que no cumplían los requisitos de duración y creación, tamaño excesivo y código duplicado
  - Creación de módulos en los que los datos si eran correctos: Formato correcto, tiempos de duración permitidos, tamaños aceptados y dejando en blanco solo aquellos que eran opcionales.
- **Hacking:**
  - Acceder al formulario de creación sin haber iniciado sesión
  - Acceder al formulario de creación habiendo iniciado sesión con otro rol

- **Bugs:** Se encontró un bug cuando se intentó crear un báner en el que alguna de las fechas era nula
- **Soluciones:** Para solucionar el bug, en el método validate, se añadió la comprobación a ambas fechas de que estas no podían ser nulas si se quería comprobar el número de días entre dos fechas. Este solución se implementó también en el servicio de actualización

## AdministratorBannerUpdateService















- **Safe testing:**
  - Actualización de báners en los que los datos eran incorrectos: Nulos en aquellos obligatorios, formato incorrecto, fechas que no cumplían los requisitos de duración y creación, tamaño excesivo y código duplicado
  - Actualización de báners en los que los datos si eran correctos: Formato correcto, tiempos de duración permitidos, tamaños aceptados y dejando en blanco solo aquellos que eran opcionales.
- **Hacking:** Desde las herramientas del desarrollador, se modificó el id del báner que iba a actualizar, de esta forma se probó a actualizar un báner que no había sido creado
- **Bugs:** No se encontraron bugs

## AdministratorBannerDeleteService

- **Safe testing:** Eliminación de báners creados por el administrador
- **Hacking:** Desde las herramientas del desarrollador, se modificó el id del módulo que iba a eliminar, de esta forma se probó a eliminar un módulo que no había sido creado
- **Bugs:** No se encontraron bugs

## Cobertura

A continuación se muestra el porcentaje de cobertura de sentencias de todos los servicios asociados a Banner

▼  acme.features.administrator.banner		88,7 %	643	82	725
>  AdministratorBannerDeleteService.java		60,9 %	78	50	128
>  AdministratorBannerCreateService.java		92,6 %	199	16	215
>  AdministratorBannerUpdateService.java		94,6 %	212	12	224
>  AdministratorBannerListService.java		92,6 %	50	4	54
>  AdministratorBannerController.java		100,0 %	29	0	29
>  AdministratorBannerShowService.java		100,0 %	75	0	75

Observamos que el coverage en el servicio de eliminación es considerablemente menor que el de el resto de los servicios. Al realizar el testing de este servicio observamos que no existen casos negativos a la hora de eliminar los báners ya que realizando acciones legales, siempre es posible

eliminar un báner creado. Este caso particular es el que impide que se ejecuten las siguientes lineas en ningun caso.

```
@Override
public void unbind(final Banner object) {
    assert object != null;

    Dataset dataset;

    dataset = super.unbind(object, "instantiationMoment", "startDisplayPeriod", "endDisplayPeriod", "slogan", "picture", "link");

    super.getResponse().addData(dataset);
}
```

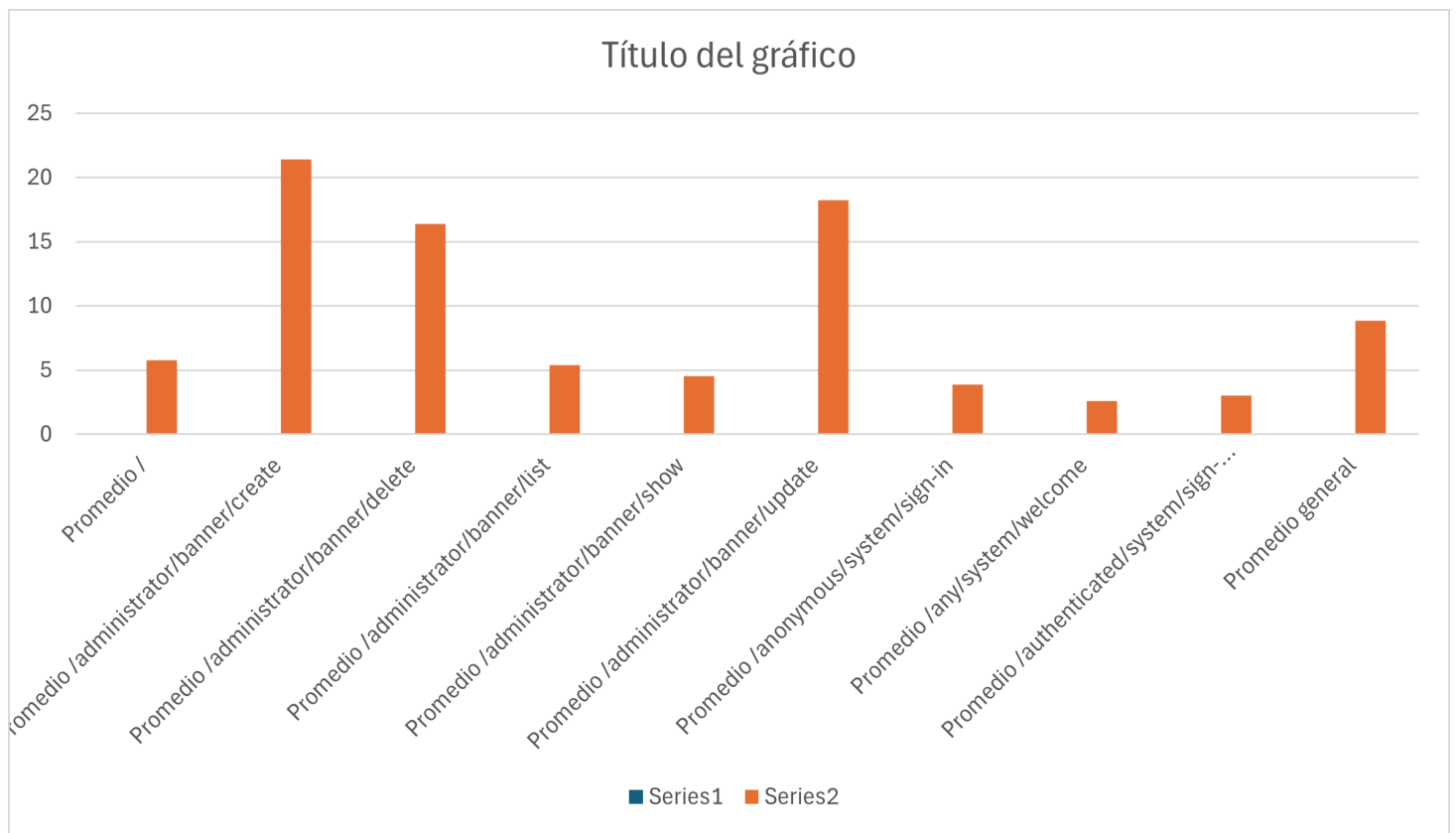
Una de las soluciones que por tanto se consideran es la eliminación del método por completo ya que con las pruebas de testing queda comprobado que no se está utilizando para nada.

## Testing de rendimiento

### Análisis del rendimiento

Tras la ejecución de los tests, y con los datos recopilados sobre las peticiones realizadas en los mismos, se ha llevado a cabo un análisis de los resultados obtenidos en cuanto al rendimiento la aplicación tomando como referencia el tiempo de respuesta de los servicios, en concreto, de los servicios asociados a la entidad Banner. A continuación, se proporcionan los datos del promedio de cada petición, aunque para una mayor claridad, también se proporcionan una gráfica de barras.

Promedio /		5,76256
Promedio /administrator/banner/create		21,4011
Promedio /administrator/banner/delete		16,3697
Promedio /administrator/banner/list		5,37687
Promedio /administrator/banner/show		4,52494
Promedio /administrator/banner/update		18,259
Promedio /anonymous/system/sign-in		3,8635
Promedio /any/system/welcome		2,60881
Promedio /authenticated/system/sign-out		3,00227
Promedio general		8,86466



Lo primero que resalta al observar la gráfica es que hay 3 peticiones que destacan por tener un tiempo de respuesta muy superior al resto, estas peticiones, como podrían haber sido de esperar, son todas relacionadas con los banners y son la creación, la actualización y la eliminación de los mismos, teniendo ambas 3, aproximadamente el doble o más tiempo de respuesta que el promedio de todas las peticiones realizadas.

A pesar de esto, el resto de peticiones rondan los 0.003 segundos lo que hace que el tiempo promedio sea muy bajo, de aproximadamente 0.008 segundos.

## Intervalo de confianza

Una vez analizados los tiempos de respuesta, se ha calculado el intervalo de confianza de la media los mismos, con un nivel de confianza del 95%, con el objetivo de comprobar si la aplicación, para las *features* de los banners cumple con el requisito de rendimiento establecido: *que el tiempo de respuesta de las peticiones no supere el segundo de media*.

La herramienta utilizada para el análisis del rendimiento ha sido excel, que proporciona un complemento llamado *Herramientas para el análisis* que permite realizar cálculos de estadística descriptiva dando la opción de establecer un nivel de confianza del 95%, con este nivel de confianza podremos esperar que los tiempos promedios de ejecución, estén en un 95% de los casos dentro del intervalo de confianza.

Para obtener el intervalo de confianza, hemos debido de establecer el límite superior y el límite inferior (tanto en milisegundos como en segundos).

- **Límite superior:** Media + Nivel de confianza
- **Límite inferior:**  $\max(\text{Media} - \text{Nivel de confianza}, 0)$

A continuación se muestran los datos del análisis estadístico así como del intervalo de confianza de la media de los tiempos de respuesta de las peticiones.

Columna1				
		Interval (ms)	7,50168833	10,2276336
Media	8,86466098	Interval (s)	0,00750169	0,01022763
Error típico	0,6919718			
Mediana	4,02265			
Moda	#N/D			
Desviación estándar	10,8531536			
Varianza de la muestra	117,790944			
Curtosis	10,2927806			
Coefficiente de asimetría	2,69426947			
Rango	80,2815			
Mínimo	1,5615			
Máximo	81,843			
Suma	2180,7066			
Cuenta	246			
Nivel de confianza(95,0%)	1,36297265			

Como podemos observar, gracias al cálculo del intervalo de confianza, podemos afirmar que el tiempo de respuesta de las peticiones de la aplicación se encuentra dentro del rango de  $[0,1]$ , en concreto  $[0.007, 0.010]$  segundos, cumpliendo así con el requisito de rendimiento establecido.

## Hypothesis contrast

A pesar de cumplir con el requisito de rendimiento establecido, se quiso llevar a cabo una refactorización mediante el uso de índices para aumentar el rendimiento de las *queries* del repositorio asociado a la entidad Banner, sin embargo, el número limitado de *queries* requeridas y su simpleza intrínseca, no permitió que se pudiera llevar a cabo una refactorización mediante índices.

En su lugar, se decidió realizar el contraste de rendimiento en dos máquinas diferentes con el objetivo de comprobar si, independientemente de la refactorización, el rendimiento de la aplicación se veía afectado por la máquina en la que se ejecutaba. Para ello, se ejecutaron los tests en dos máquinas diferentes. La primera vez se ejecutó en una máquina con un procesador AMD Ryzen 7 y una memoria de 32GB, mientras que la segunda vez se ejecutó en una máquina con un procesador Intel Core i7 y una memoria de 16GB.

Para poder realizar una comparación objetiva de los rendimientos, se ha llevado a cabo un Z-test en excel con el complemento *Herramientas para el análisis*.

A continuación se muestran los datos del análisis de los tiempos en ambas máquinas, así como los resultados del Z-test.

Before				After				Prueba z para medias de dos muestras			
									Before	After	
Media	8,86466098			Media	14,2695243			Media	8,5667902	13,9473171	
Error típico	0,6919718			Error típico	0,8079212			Varianza (conocida)	117,790944	199,084682	
Mediana	4,02265			Mediana	8,8852			Observaciones	245	304	
Moda	#N/D			Moda	#N/D			Diferencia hipotética de las medias	0		
Desviación estándar	10,8531536			Desviación estándar	14,1097371			z	-5,04893795		
Varianza de la muestra	117,790944			Varianza de la muestra	199,084682			P(Z<=z) una cola	2,2214E-07		
Curtosis	10,2927806			Curtosis	9,2862432			Valor crítico de z (una cola)	1,64485363		
Coefficiente de asimetría	2,69426947			Coefficiente de asimetría	2,47448482			Valor crítico de z (dos colas)	4,4427E-07		
Rango	80,2815			Rango	110,2475			Valor crítico de z (dos colas)	1,95996398		
Mínimo	1,5615			Mínimo	1,973						
Máximo	81,843			Máximo	112,2205						
Suma	2180,7066			Suma	4352,2049						
Cuenta	246			Cuenta	305						
Nivel de confianza(95,0%)	1,36297265			Nivel de confianza(95,0%)	1,58982583						
Interval (ms)	7,50168833	10,2276336		Interval (ms)	12,6796984	15,8593501					
Interval (s)	0,00750169	0,01022763		Interval (s)	0,0126797	0,01585935					

Como podemos observar, el valor crítico de z (dos colas) es  $4,4427 \cdot 10^{-7}$  el cual se encuentra considerablemente por debajo de  $\alpha$  (que para este test tenía un valor de 0.05). Este resultado nos indica que la diferencia de tiempos ha sido significativa, sin embargo, debemos todavía comprobar si la diferencia es positiva o negativa. Esto es simple de comprobar ya que si echamos un vistazo a los datos estadísticos, en concreto a la media, podremos observar que la máquina con el procesador AMD Ryzen 7 y 32GB de memoria ha obtenido un tiempo promedio de respuestas de 0.008 segundos frente a los 0.014 segundos de la máquina con el procesador Intel Core i7 y 16GB de memoria. Por tanto, podemos afirmar que el rendimiento de la aplicación se ha visto afectado considerablemente de forma negativa cuando se han ejecutado las peticiones en una máquina de peores características.