

Informe de Pruebas

Grupo C2.008



<https://github.com/Luis-Giraldo-Santiago3/Acme-SF-D04>

Alejandro Vargas Muñiz → alevarmun1@alum.us.es

David Vargas Muñiz → davvarmun@alum.us.es

Luis Giraldo Santiago → luisgirsan@gmail.com

Rafael Molina García → rafmolgar2@alum.us.es

04/07/2024

TABLA DE CONTENIDOS

1. RESUMEN EJECUTIVO
2. TABLA DE REVISIÓN
3. INTRODUCCIÓN
4. CONTENIDO
5. CONCLUSIONES
6. BIBLIOGRAFÍA

1. RESUMEN EJECUTIVO

El presente informe tiene como objetivo proporcionar un análisis general de sobre los casos de pruebas sometidas al código realizado por el grupo.

2. TABLA DE REVISIÓN

Fecha	Versión	Descripción
04/07/2024	1.0	TestingReport Goup

3. INTRODUCCIÓN

El presente informe presenta un análisis sobre los casos de pruebas realizado a la entidad banner. Este informe se centra en analizar la cobertura de los casos de pruebas y la optimización del código.

4. Contenido

Capítulo 1: Pruebas Funcionales

Introducción

Este informe presenta los resultados de las pruebas funcionales y de rendimiento realizadas en el proyecto "Acme-SF" respecto a banner. Las pruebas funcionales verifican la efectividad en la detección de errores, mientras que las pruebas de rendimiento evalúan el tiempo de respuesta en diferentes computadoras y entre tener o no tener índices. Los resultados ofrecen una visión sobre la estabilidad y eficiencia del proyecto.

Casos de Prueba Implementados para "Banners":

Característica 1: Listado de banners ("AdministratorBannerListService.java")

- **Descripción:** Prueba la funcionalidad que permite a los administradores listar los banners almacenados en el sistema. Esta característica debe asegurarse de que solo se muestren los banners a los administradores.
- **Efectividad:** En cuanto a la cobertura, observamos que tenemos un valor de 93,3%. Al ejecutar el "replayer" observamos que todas las líneas están en verde, por lo que se ha entrado y se ha ejecutado todo correctamente, de ahí el alto porcentaje y observamos que hay una líneas en amarillo:
 - Líneas amarillas:
 - Línea 40: `assert object != null;` La condición que siempre se cumple es que "object" siempre va a ser distinto de null.

Característica 2: Detalles de banners("AdministratorBannerShowService.java")

- **Descripción:** Prueba la funcionalidad que permite a los administradores ver los detalles de los banners almacenados en el sistema. Esta característica debe asegurarse de que solo se muestren los detalles de los banners a los administradores.
- **Efectividad:** En cuanto a la cobertura, observamos que tenemos un valor de 94,2%. Al ejecutar el "replayer" observamos que todas las líneas están en verde, por lo que se ha entrado y se ha ejecutado todo correctamente, de ahí el alto porcentaje y observamos que hay una líneas en amarillo:
 - Líneas amarillas:
 - Línea 40: `assert object != null;` La condición que siempre se cumple es que "object" siempre va a ser distinto de null.

Característica 3: Crear banners ("AdministratorBannerCreateService.java")

- **Descripción:** Prueba la funcionalidad que permite a los administradores crear nuevos banners en el sistema. Esta característica debe asegurarse de que solo los administradores puedan crear banners, que cuando se crea el banner, se guarda el instante de registro, el inicio de visualización en pantalla sea posterior al instante de registro y anterior al 2200/01/01 00:00, lo mismo para el acabado de visualización pantalla pero además tiene que ser 7 días posterior al inicio de pantalla,

que el enlace de imagen esté entre 0 y 255 caracteres, que el Eslogan esté entre 0 y 75 caracteres y que el destino del enlace esté entre 0 y 255 caracteres. Todos los atributos son obligatorios.

- **Efectividad:** En cuanto a la cobertura, observamos que tenemos un valor de 92,5%. Al ejecutar el "replayer" observamos que hay algunas líneas en amarillo:

- o Líneas amarillas:

- ☐ **Línea 70:**

```
super.state(MomentHelper.isAfterOrEqual(object.getDisplayFinish(), maximumPeriod) &&
object.getDisplayFinish().after(object.getDisplayStart()),
"displayFinish",
"administrator.banner.form.error.displayFinish");
```

Esta línea está en amarillo, porque no se ha dado el caso en que estuviera fuera de los siete días y antes de la fecha de final y esto es normal ya que no tendría sentido y esto pasara.

- ☐ **Línea 39, 46, 65 y 71:** `assert object != null;` La condición que siempre se cumple es que "object" siempre va a ser distinto de null.

Característica 4: Actualizar banner("AdministratorBannerUpdateService.java")

- **Descripción:** Prueba la funcionalidad que permite a los administradores actualizar los banners en el sistema. Esta característica debe asegurarse de que solo los administradores puedan actualizar banners, que cuando se actualice, el inicio de visualización en pantalla sea posterior al instante de registro y anterior al 2200/01/01 00:00, lo mismo para el acabado de visualización pantalla pero además tiene que ser 7 días posterior al inicio de pantalla, que el enlace de imagen esté entre 0 y 255 caracteres, que el Eslogan esté entre 0 y 75 caracteres y que el destino del enlace esté entre 0 y 255 caracteres. Todos los atributos son obligatorios.
- **Efectividad:** En cuanto a la cobertura, observamos que tenemos un valor de 92,7%. Al ejecutar el "replayer" observamos que hay algunas líneas en amarillo:

- o Líneas amarillas:

- ☐ **Línea 70:**

```
super.state(MomentHelper.isAfterOrEqual(object.getDisplayFinish(), maximumPeriod) &&
object.getDisplayFinish().after(object.getDisplayStart()),
"displayFinish",
"administrator.banner.form.error.displayFinish");
```

Esta línea está en amarillo, porque no se ha dado el caso en que estuviera fuera de los siete días y antes de la fecha de final y esto es normal ya que no tendría sentido y esto pasara.

- ☐ **Línea 39, 45, 64 y 71:** `assert object != null;` La condición que siempre se cumple es que "object" siempre va a ser distinto de null.

Característica 6: Borrar banners ("AdministratorBannerDeleteService.java")

- **Descripción:** Prueba la funcionalidad que permite a los administradores borrar contratos del sistema. Esta característica debe asegurarse de que solo los administradores autenticados puedan eliminar banners.
- **Efectividad:** En cuanto a la cobertura, observamos que tenemos un valor de 57,1%. Al ejecutar el "analyser" observamos que hay algunas líneas en amarillo y en rojo.
 - Líneas amarillas:
 - **Líneas 43, 50, 56:** `assert object != null;` La condición que siempre se cumple es que "object" siempre va a ser distinto de null.
 - Líneas rojas: se corresponden al método "unbind", este método no se ejecuta debido a que no se tiene que ejecutar, ya que sirve para mostrar los datos cuando devuelve un error y en el delete no hay errores que devolver por tanto no se cargan los datos.

Capítulo 2: Pruebas de Rendimiento

Introducción

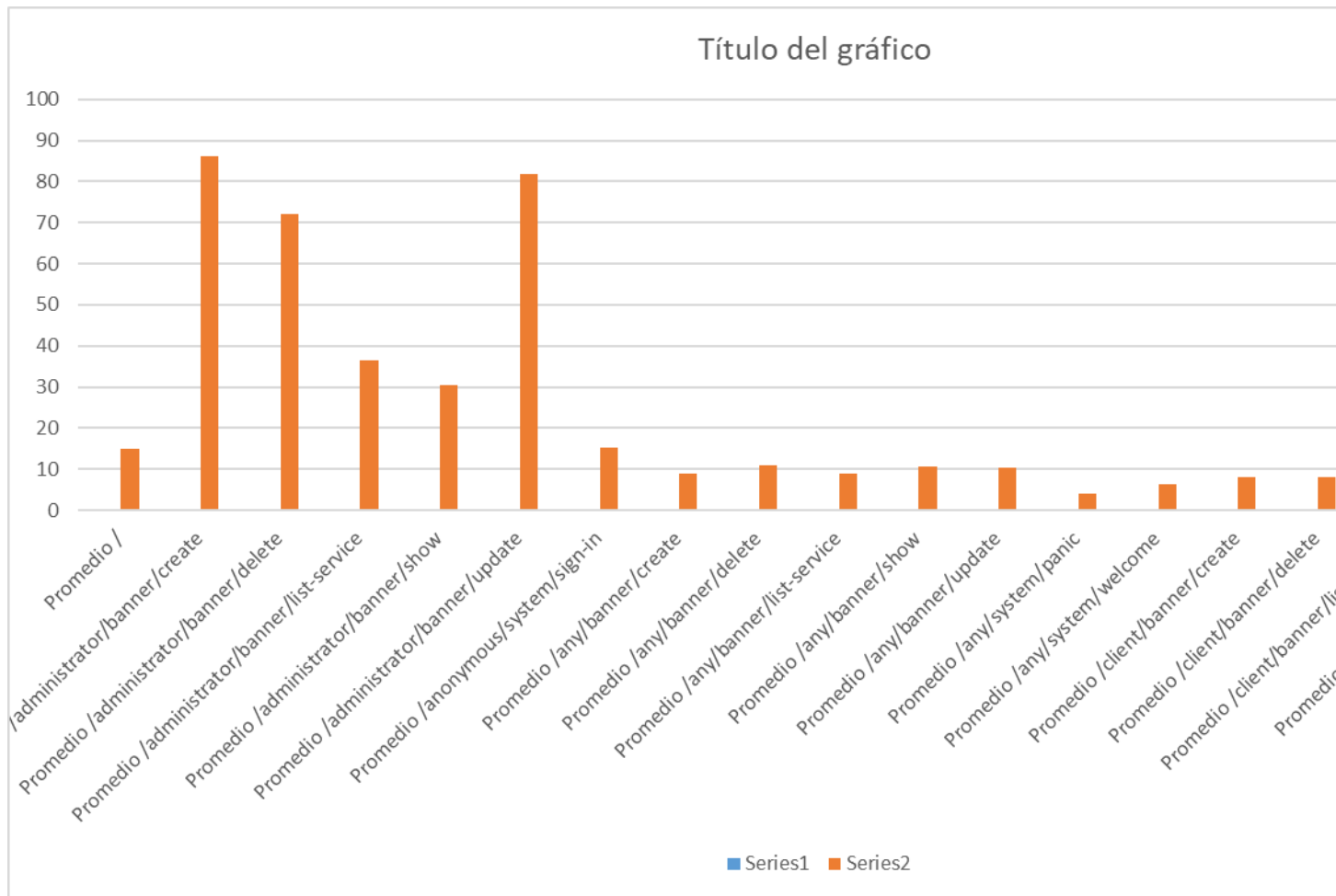
Este capítulo analiza las pruebas de rendimiento realizadas en el proyecto "Acme-SF" respecto a las características del administrador. El objetivo es evaluar el tiempo de respuesta y la capacidad del sistema bajo diferentes cargas. Utilizamos una metodología que incluye la simulación de escenarios reales y el análisis estadístico de los resultados para garantizar la eficiencia y estabilidad del sistema.

Resultados de las Pruebas de Rendimiento

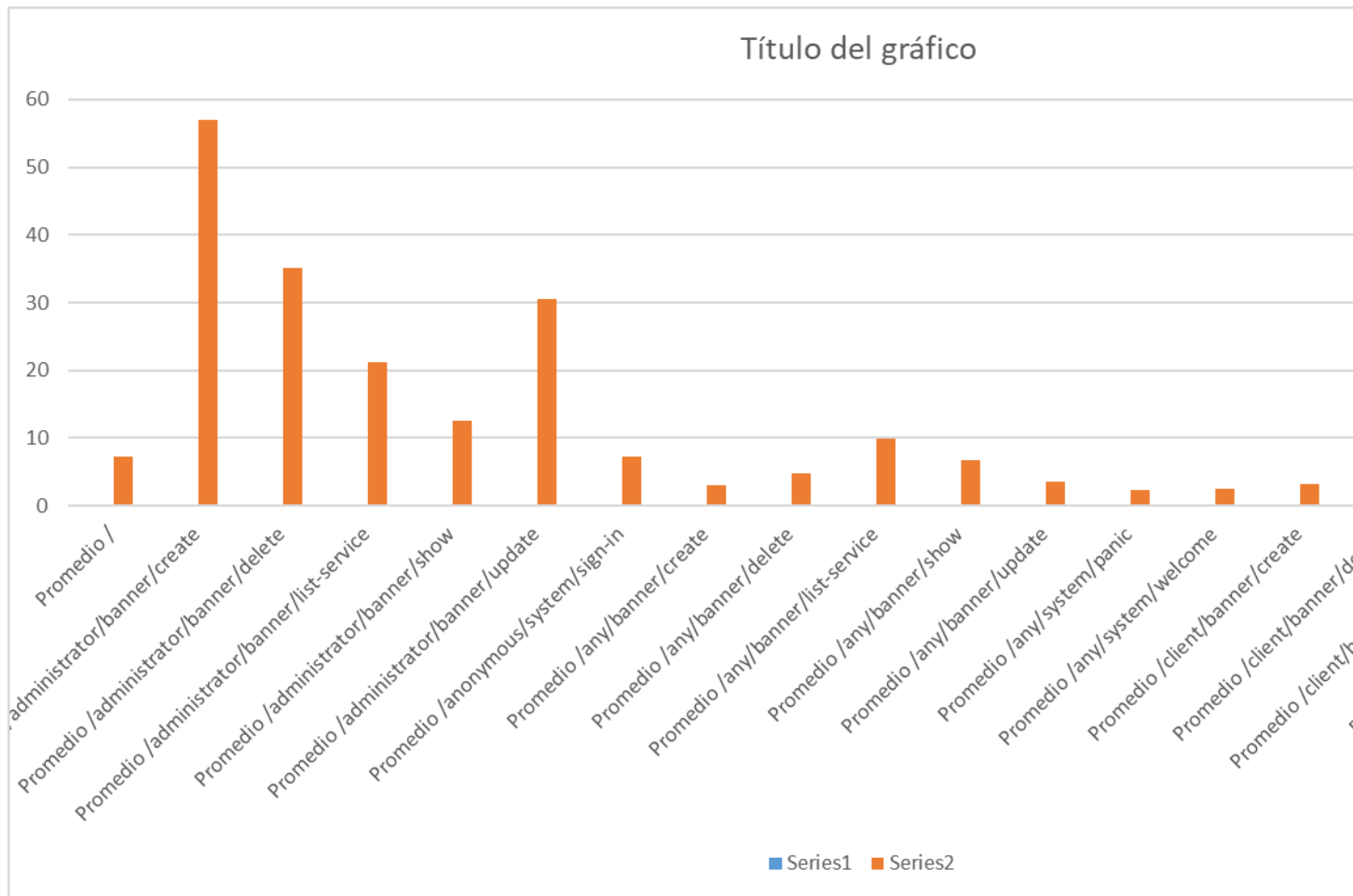
Gráficos

Gráficos del tiempo para la ejecución de las características del administrador en PC1.

- **Gráfico 1:** Gráfico del tiempo para la ejecución de las características del administrador en PC1.

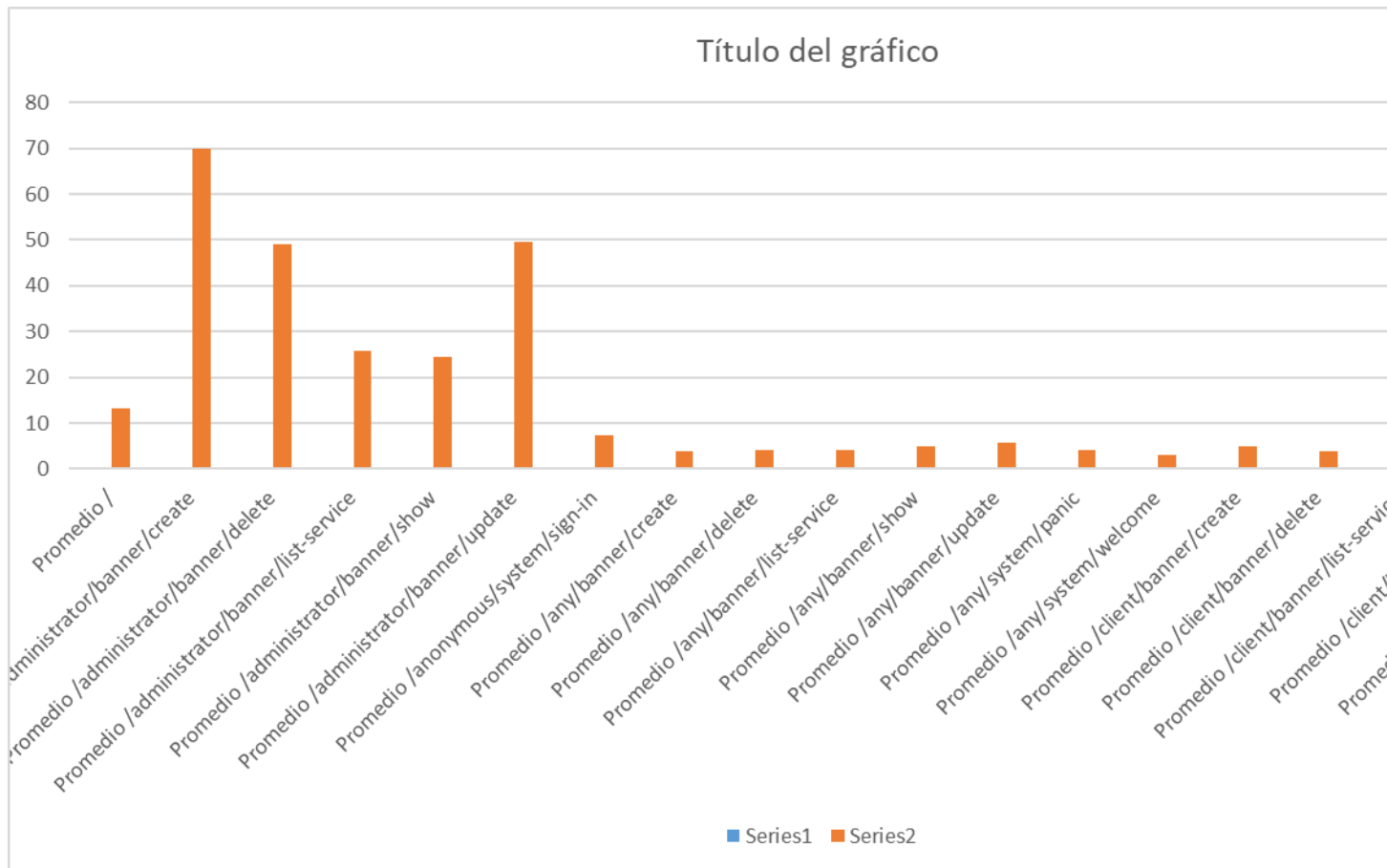


- **Gráfico 2:** Gráfico del tiempo para la ejecución de las características del administrador con optimización en PC1.



Gráficos del tiempo para la ejecución de las características del administrador en PC1.

- **Gráfico 3:** Gráfico del tiempo para la ejecución de las características del administrador con optimización en PC2.



Intervalo de Confianza

Presentar el intervalo de confianza del 95% para el tiempo de respuesta en las dos computadoras.

- PC1:
 - Nivel de confianza sin la optimización: 6,036016122
 - Nivel de confianza con la optimización: 3,801919374
- PC2:
 - Nivel de confianza con la optimización: 4,401855747

Contraste de Hipótesis

Presentar el contraste de hipótesis con un 95% de confianza sobre cuál computadora es más potente.

- **Resultado del Contraste de Hipótesis:**

○ **Prueba z entre los datos del mismo PC**

Prueba z para medias de dos muestras		
	<i>Tiempos de PC1 sin índices</i>	<i>Tiempos de PC1 con índices</i>
Media	56,27540872	29,34383119
Varianza (conocida)	2224,781	882,6575
Observaciones	218	218
Diferencia hipotética de las medias	0	
z	7,133271742	
P(Z<=z) una cola	4,90052E-13	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	9,80105E-13	
Valor crítico de z (dos colas)	1,959963985	

○ **Prueba z entre los datos de distintos PC**

Prueba z para medias de dos muestras		
	<i>Tiempos de PC1 con índices</i>	<i>Tiempo de PC2 con índices</i>
Media	29,34383119	39,96678165
Varianza (conocida)	882,6575	1183,2
Observaciones	218	218
Diferencia hipotética de las medias	0	
z	-3,450827277	
P(Z<=z) una cola	0,000279436	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,000558871	
Valor crítico de z (dos colas)	1,959963985	

Como podemos observar en la primera imagen, al fijarnos en el valor crítico de z(dos cola), vemos que nos da un $9,80105 \cdot 10^{-13}$, este número está entre [0.00,0.05), todo esto nos indica que ha habido una mejora de tiempo al añadir índices al código.

Cuando hacemos la misma prueba con los tiempos de los dos PCs optimizados, nos da un valor de 0,000558871, esto quiere decir que al cambiar de PC a uno mejor, también obtenemos una mejora del tiempo que ya habíamos mejorado.

5. CONCLUSIONES

Al realizar este reporte, podemos observar que los casos de pruebas realizadas a banner cubren más del 80% del código realizado para esta entidad.

Y para el tiempo, vemos que al añadir los índices se obtiene una mejora del tiempo y que además, se puede seguir mejorando si se tiene un PC de alta gama.

6. BIBLIOGRAFÍA

Intencionalmente en blanco.