

Informe de Pruebas

Capítulo 1: Pruebas Funcionales

Element	Coverage	covered Instructions	missed Instructions	Total Instructions
> acme.entities.student2	59,3 %	70	48	118
> acme.entities.student3	67,1 %	116	57	173
> acme.entities.student4	57,6 %	102	75	177
> acme.entities.student5	63,2 %	168	98	266
▼ acme.features.administrator.banner	86,3 %	593	94	687
> AdministratorBannerController.java	100,0 %	30	0	30
> AdministratorBannerCreateService.java	92,0 %	183	16	199
> AdministratorBannerListService.java	93,3 %	56	4	60
> AdministratorBannerShowService.java	94,2 %	65	4	69
> AdministratorBannerUpdateService.java	92,1 %	187	16	203
> AdministratosBannerDeleteService.java	57,1 %	72	54	126
> acme.features.any.claim	10,1 %	32	284	316
> acme.features.auditor.auditRecord	5,7 %	89	1.460	1.549
> acme.features.auditor.codeAudit	7,2 %	90	1.152	1.242

Introducción

Este informe presenta los resultados de las pruebas funcionales y de rendimiento realizadas en el proyecto "Acme-SF" respecto a los banners. 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. 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 "analyser" observamos que hay solo una línea en amarillo:
 - Líneas amarillas:
 - **Línea 47:** `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, el inicio de pantalla sea posterior al 2022/07/30 00:00 y anterior al 2200/01/01 00:00, lo mismo para el acabado de 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 menos el destino del enlace.
- **Efectividad:** En cuanto a la cobertura, observamos que tenemos un valor de 92,0%. Al ejecutar el "analyser" observamos que tenemos todas las líneas verdes menos todos los asserts, los cuales están en amarillo y siempre están en amarillo debido a que siempre son distintos de null y 2 líneas:
 - **Línea 52:** `if (!super.getBuffer().getErrors().hasErrors("displayFinish") && !super.getBuffer().getErrors().hasErrors("displayStart"))`
 - **Línea 56:** ``super.state(MomentHelper.isAfterOrEqual(object.getDisplayFinish(), maximumPeriod) && object.getDisplayFinish().after(object.getDisplayStart()), "displayFinish", "administrator.banner.form.error.displayFinish");``
 - Estas líneas están en amarillo debido a que siempre se ha comprobado que pasen las dos cosas o ninguna, no se han hecho test de solo una true y la otra false.

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

- **Descripción:** Prueba la funcionalidad que permite a los administradores actualizar los detalles de los banners almacenados en el sistema. Esta característica debe asegurarse de que solo los administradores puedan ser actualizarlos, que cuando se actualice el banner el inicio de pantalla sea posterior al 2022/07/30 00:00 y anterior al 2200/01/01 00:00, lo mismo para el acabado de 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 menos el destino del enlace.

- **Efectividad:** En cuanto a la cobertura, observamos que tenemos un valor de 92,1%. Al ejecutar el "replayer" observamos que hay algunas líneas en amarillo.
 - **Línea 33:** `super.state(MomentHelper.isAfterOrEqual(object.getDisplayFinish(), maximumPeriod) && object.getDisplayFinish().after(object.getDisplayStart()), "displayFinish", "administrator.banner.form.error.endDisplayPeriod");` Esta línea está en amarillo debido a que siempre se ha comprobado que pasen las dos cosas o ninguna, no se han hecho test de solo una true y la otra false.
 - **Líneas 39, 45, 63 y 70:** `assert object != null;` La condición que siempre se cumple es que "object" siempre va a ser distinto de null.

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

- **Descripción:** Prueba la funcionalidad que permite a los administradores borrar banners 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

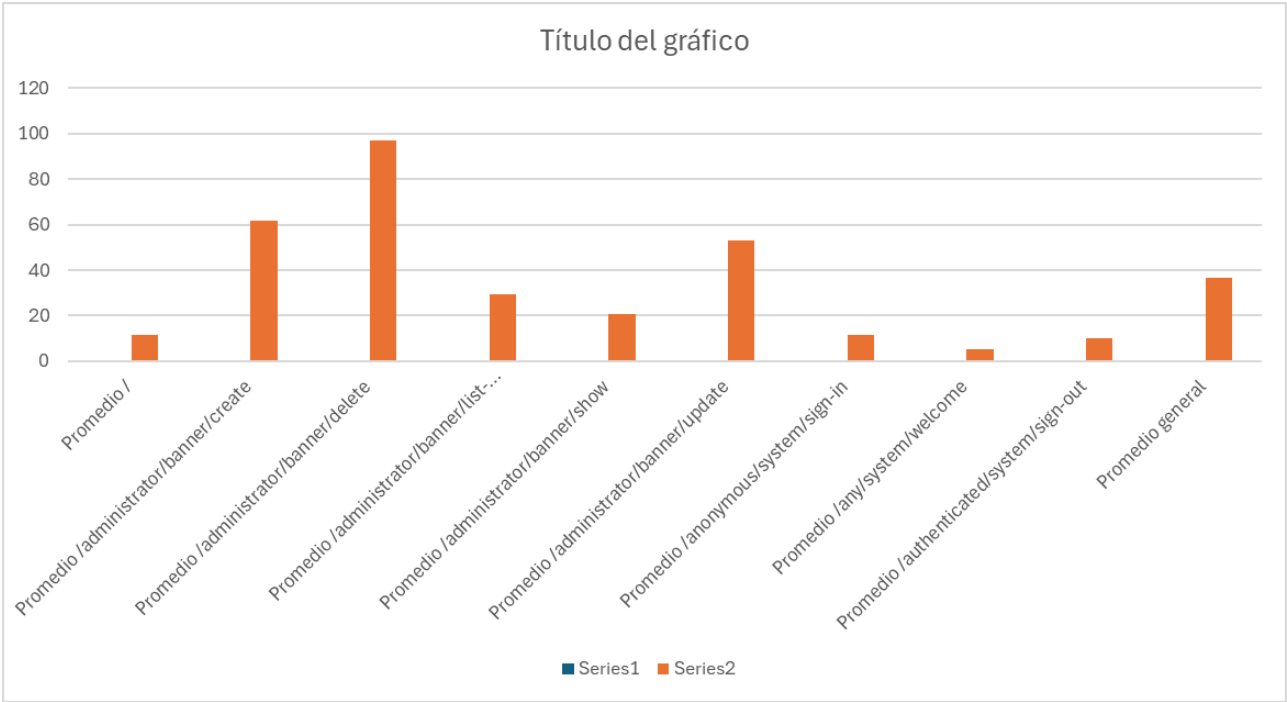
Este capítulo analiza las pruebas de rendimiento realizadas en el proyecto "Acme-SF" respecto a las características de los banners. El objetivo es evaluar el tiempo de respuesta y la capacidad del sistema bajo diferentes cargas. Utilizamos una metodología que incluye la simulación de escenarios reales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

Resultados de las Pruebas de Rendimiento

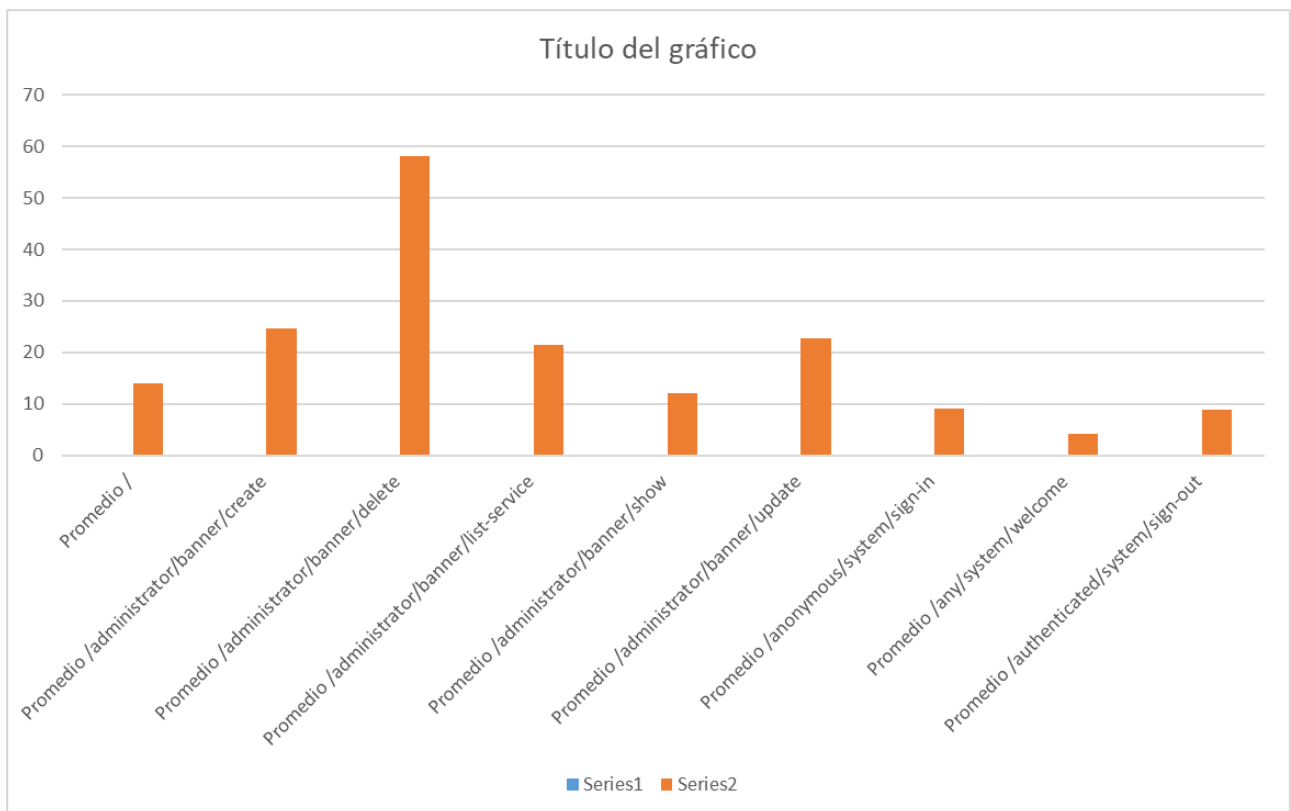
Gráficos

Gráfico del tiempo para la ejecución de las características del gerente sin la optimización en PC1.

- **Gráfico 1:** Gráfico del tiempo para la ejecución de las características del adminsitrador(banners) en PC1.



- **Gráfico 2:** Gráfico del tiempo para la ejecución de las características del administrador (banners) 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:**

PC1:		
<i>Columna1</i>		
Media	36,5774686	
Error típico	2,20357531	
Mediana	36,8863	
Moda	#N/D	
Desviación es	34,2795869	
Varianza de la	1175,09008	
Curtosis	6,01219177	
Coeficiente de	1,77085478	
Rango	238,4094	
Mínimo	2,8798	
Máximo	241,2892	
Suma	8851,7474	
Cuenta	242	
Nivel de confia	4,34072651	
Interval (ms)	32,2367421	40,9181951
Interval (s)	0,03223674	0,0409182

- **PC2:**

- **Nivel de Confianza sin la optimización:**

PC2:		
<i>Columna1</i>		
Media	18,4022778	
Error típico	0,91124768	
Mediana	18,8747	
Moda	#N/D	
Desviación es	13,4235166	
Varianza de la	180,190798	
Curtosis	4,14626535	
Coeficiente de	1,57573174	
Rango	77,0255	
Mínimo	1,5024	
Máximo	78,5279	
Suma	3993,29429	
Cuenta	217	
Nivel de confia	1,796076	
Interval(ms)	16,6062018	20,1983538
Interval(s)	0,0166062	0,02019835

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 para medias de dos muestras		
	<i>After</i>	<i>Before</i>
Media	35,9961137	16,9987895
Varianza (conocida)	1175,09008	180,190798
Observaciones	241	240
Diferencia hipotética de las medias	0	
z	8,00877399	
P(Z<=z) una cola	5,5511E-16	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	1,1102E-15	
Valor crítico de z (dos colas)	1,95996398	

- Si observamos el "p-value" para "Before" y "After", su valor es de $1,1192 * 10^{-15}$, como no está contenido en el intervalo $(0.05, 1.00]$ entonces los cambios dieron como resultado una desmejora significativa; los tiempos de muestras son globalmente diferentes . De hecho, el tiempo de ejecución medio es mayor después de optimizar, esto es debido a que la optimización que hemos hecho tiene complejidad logarítmica respecto la complejidad lineal que tenemos antes de optimizar, al tener tan pocos datos nos empeora el rendimiento, por lo que esta optimización es ideal para tener una mayor cantidad de datos.