

Testing report – Student 2



Grupo: C3.Y10

Repositorio: [margargar81/Acme-SF-D04-24.5.0 \(github.com\)](https://github.com/margargar81/Acme-SF-D04-24.5.0)

Integrantes:

Nombre	Apellidos	Correo Corporativo
María del Carmen	García García	Margargar81@alum.us.es

Tabla de versiones:

Fecha	Versión	Descripción de los cambios	Sprint
20/10/2024	1.0	Creación del documento y redacción inicial	4

Índice

1. Resumen Ejecutivo
2. Introducción
3. Contenido del Testing
 - 3.1. Casos de prueba sobre Contract
 - 3.2. Casos de prueba sobre ProgressLog
4. Pruebas de Rendimiento
5. Análisis de Rendimiento
6. Optimización mediante índices
7. Intervalo de Confianza
8. Contraste de Hipótesis
9. Conclusión del análisis de rendimiento
10. Conclusiones

Resumen Ejecutivo

Este informe resume las pruebas realizadas en los casos de prueba correspondientes a las funcionalidades de la entidad ProgressLog y Contract descritas en los requisitos del proyecto. Se incluye tanto testing funcional como hacking y análisis de la cobertura de código para los servicios asociados. El objetivo de estas pruebas es garantizar que las funcionalidades principales operen de manera correcta, detectando errores y vulnerabilidades en la implementación de las clases ProgressLog y Contract.

Introducción

En el contexto de este proyecto, se realizaron pruebas detalladas sobre los servicios de los contratos (Contract) y los registros de progreso (ProgressLog). En esta sección se presenta un análisis de los resultados obtenidos a partir de las pruebas para las tareas 6 y 7 indicadas en los requisitos del estudiante 2, incluyendo la validación de las operaciones de creación, actualización, publicación y borrado de los elementos correspondientes.

Contenido del Testing

Casos de prueba sobre Contract

Creación y actualización de un contrato (Create/Update Contract)

- **Caso de prueba 1: Creación Exitosa**

Descripción: Se verifica que el cliente puede crear un contrato existente con datos válidos.

Resultado esperado: Los cambios se reflejan correctamente en la interfaz y en la base de datos.

Efectividad: Alta, se asegura la integridad de los datos ingresados y la persistencia correcta en la base de datos.

- **Caso de prueba 2: Actualización Exitosa**

Descripción: Se verifica que el cliente puede modificar un contrato existente con datos válidos.

Resultado esperado: Los cambios se reflejan correctamente en la interfaz y en la base de datos.

Efectividad: Alta, se asegura que la operación de actualización mantiene la coherencia del sistema.

- **Caso de prueba 3: Campos Inválidos**
Descripción: Se intenta actualizar el contrato con información no válida (por ejemplo, presupuesto negativo, divisa no correspondiente a la del proyecto relacionado con el contrato...).
Resultado esperado: Se emite un error que impide la actualización.
Efectividad: Alta, ya que evita la corrupción de datos.
- **Caso de prueba 4: Campos Vacíos**
Descripción: Se intenta crear un registro dejando campos obligatorios vacíos.
Resultado esperado: Error de validación que impide la creación del registro.
Efectividad: Se confirma que el sistema maneja adecuadamente la validación de campos obligatorios.
- **Caso de prueba 5: Hacking**
Descripción: Se prueba si el sistema permite actualizar contratos ya publicados o contratos de otros usuarios.
Resultado esperado: Acceso denegado, con mensaje de error.
Efectividad: El sistema muestra la seguridad esperada al evitar actualizaciones no autorizadas.

Listar Contratos (List Contract)

- **Caso de prueba 1: Listar contratos de un cliente**
Descripción: Se prueba la capacidad del sistema para listar los contratos asociados a un cliente específico.
Resultado esperado: Los contratos asociados al cliente se listan correctamente en la interfaz de usuario.
Efectividad: Alta, se asegura que solo los contratos del cliente se muestren correctamente.
- **Caso de prueba 2: Hacking**
Descripción: Se intenta acceder a la lista de contratos de un cliente sin la debida autorización.
Resultado esperado: Acceso denegado con mensaje de error.
Efectividad: El sistema garantiza la seguridad al evitar accesos no autorizados.

Mostrar Detalles de Contrato (Show Contract)

- **Caso de prueba 1: Mostrar detalles del contrato**
Descripción: Se prueba la capacidad del sistema para mostrar los detalles de un contrato específico.
Resultado esperado: Los detalles del contrato se muestran correctamente.
Efectividad: Alta, se asegura que los detalles del contrato se muestren correctamente.
- **Caso de prueba 2: Hacking**
Descripción: Se intenta acceder a los detalles de un contrato sin la debida autorización.
Resultado esperado: Acceso denegado con mensaje de error.
Efectividad: El sistema asegura la privacidad de los datos al evitar accesos no autorizados.

Eliminar Contrato (Delete Contract)

- **Caso de prueba 1: Eliminar un contrato existente**
Descripción: Se prueba la capacidad del cliente para eliminar un contrato en modo borrador.
Resultado esperado: El contrato se elimina correctamente de la base de datos.
Efectividad: Alta, se asegura que solo los contratos en modo borrador puedan ser eliminados.
- **Caso de prueba 2: Hacking**
Descripción: Se intenta eliminar un contrato ya publicado o perteneciente a otro usuario.
Resultado esperado: Acceso denegado con mensaje de error.
Efectividad: El sistema evita eliminaciones no autorizadas de contratos.

Publicar Contrato (Publish Contract)

- **Caso de prueba 1: Publicación exitosa**
Descripción: Se prueba que un cliente puede publicar un contrato cumpliendo todas las restricciones.
Resultado esperado: El contrato se publica correctamente.
Efectividad: Alta, se asegura que los contratos se publiquen bajo las reglas correctas.
- **Caso de prueba 2: Hack**
Descripción: Se prueba si el sistema permite publicar contratos de otros usuarios.
Resultado esperado: Acceso denegado, con mensaje de error.
Efectividad: El sistema muestra la seguridad esperada al evitar actualizaciones no autorizadas.

Casos de prueba sobre ProgressLog

Crear y Actualizar Registro de Progreso (Create/Update ProgressLog)

- **Caso de prueba 1: Creación Exitosa**

Descripción: Se verifica que el cliente puede crear un registro de progreso válido con todos los datos correctos.

Resultado esperado: Registro de progreso creado correctamente y almacenado en la base de datos.

Efectividad: Alta, se asegura la integridad de los datos ingresados y la persistencia correcta.

- **Caso de prueba 1: Actualización Exitosa**

Descripción: Se verifica que el cliente puede actualizar un registro de progreso existente con todos los datos correctos.

Resultado esperado: Registro de progreso creado correctamente y almacenado en la base de datos.

Efectividad: Alta, se asegura la integridad de los datos ingresados y la persistencia correcta.

- **Caso de prueba 3: Campos Vacíos**

Descripción: Se intenta crear un registro dejando campos obligatorios vacíos.

Resultado esperado: Error de validación que impide la creación del registro.

Efectividad: Se confirma que el sistema maneja adecuadamente la validación de campos obligatorios.

- **Caso de prueba 4: Datos Inválidos**

Descripción: Se ingresan valores incorrectos en los campos, como fechas futuras o porcentajes negativos.

Resultado esperado: Error de validación que bloquea el progreso.

Efectividad: Se identifican correctamente los errores de validación.

- **Caso de prueba 5: Hacking**

Descripción: Se intenta acceder al formulario de creación de registro de progreso sin estar autenticado o con un rol incorrecto.

Resultado esperado: Acceso denegado y mensaje de error.

Efectividad: Se garantiza la seguridad del sistema.

Listar Registros de Progreso (List ProgressLog)

- **Caso de prueba 1: Listar registros de progreso asociados a un contrato**
Descripción: Se prueba la capacidad del sistema para listar los registros de progreso asociados a un contrato específico.
Resultado esperado: Los registros de progreso asociados al contrato se listan correctamente en la interfaz de usuario.
Efectividad: Alta, se asegura que los registros del contrato se muestren correctamente.
- **Caso de prueba 2: Hacking**
Descripción: Se intenta acceder a la lista de registros de progreso de un contrato sin la debida autorización.
Resultado esperado: Acceso denegado con mensaje de error.
Efectividad: El sistema garantiza la seguridad al evitar accesos no autorizados.
-

Mostrar Detalles de Registro de Progreso (Show ProgressLog)

- **Caso de prueba 1: Mostrar detalles de un registro de progreso**
Descripción: Se prueba la capacidad del sistema para mostrar los detalles de un registro de progreso específico.
Resultado esperado: Los detalles del registro de progreso se muestran correctamente.
Efectividad: Alta, se asegura que los detalles del registro se muestren correctamente.
- **Caso de prueba 2: Hacking**
Descripción: Se intenta acceder a los detalles de un registro de progreso sin la debida autorización.
Resultado esperado: Acceso denegado con mensaje de error.
Efectividad: El sistema asegura la privacidad de los datos al evitar accesos no autorizados.

Eliminar Registro de Progreso (Delete ProgressLog)

- **Caso de prueba 1: Eliminar un registro de progreso existente**
Descripción: Se prueba la capacidad del cliente para eliminar un registro de progreso asociado a un contrato en modo borrador.
Resultado esperado: El registro se elimina correctamente de la base de datos.
Efectividad: Alta, se asegura que solo los registros asociados a contratos en modo borrador puedan ser eliminados.

- **Caso de prueba 2: Hacking**

Descripción: Se intenta eliminar un registro de progreso ya publicado o asociado a otro cliente.

Resultado esperado: Acceso denegado con mensaje de error.

Efectividad: El sistema evita eliminaciones no autorizadas de registros de progreso.

Publicar Registro de Progreso (Publish ProgressLog)

- **Caso de prueba 1: Publicación exitosa**

Descripción: Se prueba que un cliente puede publicar un registro de progreso cumpliendo todas las restricciones.

Resultado esperado: El registro de progreso se publica correctamente.

Efectividad: Alta, se asegura que los registros de progreso se publiquen bajo las reglas correctas.

- **Caso de prueba 2: Validación de completitud**

Descripción: Se prueba que el sistema valida correctamente que la completitud del progreso es válida, es decir, mayor que el registro anterior y no superior al 100%.

Resultado esperado: El sistema bloquea registros de progreso con valores inválidos de completitud.

Efectividad: Alta, se asegura la integridad de los datos de progreso publicados.

Conclusiones

Las pruebas realizadas para los estos han demostrado ser efectivas para validar la funcionalidad y seguridad del sistema. No se han detectado bugs en los escenarios positivos, mientras que los casos de prueba negativos han sido gestionados correctamente por el sistema. Esto confirma que las operaciones en ProgressLog y Contract funcionan según lo esperado y que las medidas de seguridad y validación están correctamente implementadas.

Pruebas de Rendimiento

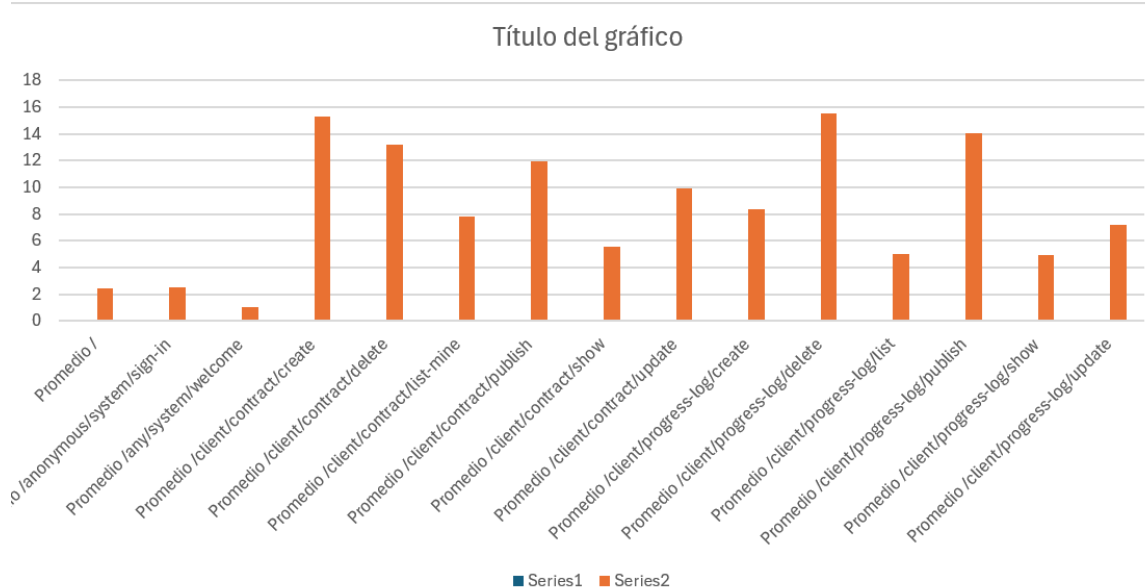
El análisis de rendimiento en el sistema **Acme-SF** se centró en medir el tiempo de respuesta de las operaciones clave de las entidades **Contract** y **ProgressLog**, con el objetivo de identificar cuellos de botella y mejorar la eficiencia del sistema a través de optimizaciones específicas, como la inclusión de índices en las consultas de base de datos.

Análisis de Rendimiento

Inicialmente, se realizaron pruebas de rendimiento para medir el tiempo de respuesta de las principales funcionalidades, antes de implementar optimizaciones. Las pruebas incluyeron operaciones como:

- Listar contratos y registros de progreso (List).
- Mostrar detalles de un contrato o registro de progreso (Show).
- Crear y actualizar contratos y registros de progreso (Create/Update).
- Eliminar contratos y registros de progreso (Delete).
- Publicar contratos y registros de progreso (Publish).

request-method	request-path	response-status	time
	Promedio /		2,4339641
	Promedio /anonymous/system/sign-in		2,54271333
	Promedio /any/system/welcome		1,04081795
	Promedio /client/contract/create		15,31213
	Promedio /client/contract/delete		13,1595
	Promedio /client/contract/list-mine		7,84145294
	Promedio /client/contract/publish		11,92065
	Promedio /client/contract/show		5,58773636
	Promedio /client/contract/update		9,90112857
	Promedio /client/progress-log/create		8,36621538
	Promedio /client/progress-log/delete		15,5117
	Promedio /client/progress-log/list		5,04957273
	Promedio /client/progress-log/publish		14,019
	Promedio /client/progress-log/show		4,92515
	Promedio /client/progress-log/update		7,1684
	Promedio general		4,94967513



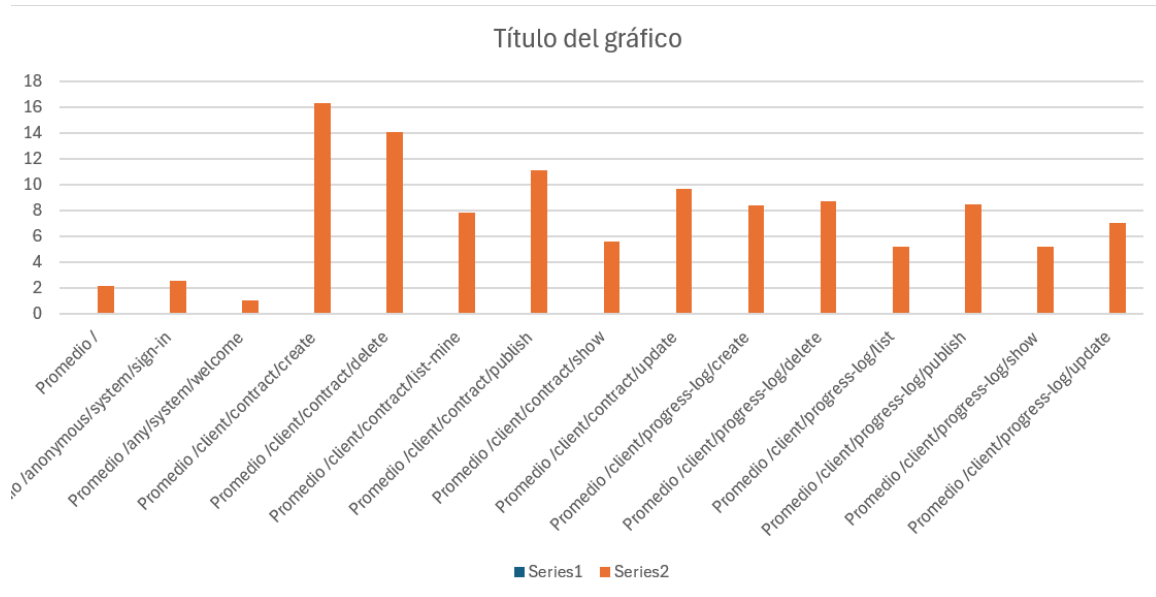
En particular, las operaciones de **creación, actualización, borrado y publicación** resultaron ser las más lentas. Estas acciones involucran una mayor cantidad de datos a procesar y validaciones más complejas, lo que contribuye a su mayor latencia.

Optimización mediante índices

Para mejorar el rendimiento de las consultas, se añadieron **índices a las tablas de las entidades Contract y ProgressLog**. Los índices permiten acelerar la búsqueda y filtrado de registros en las consultas más costosas, como las de publicación y actualización, reduciendo el número de registros que la base de datos necesita escanear.

Después de esta optimización, se observaron mejoras en los tiempos de respuesta:

request-method	request-path	response-status time
	Promedio /	2,19698462
	Promedio /anonymous/system/sign-in	2,56354667
	Promedio /any/system/welcome	1,06310769
	Promedio /client/contract/create	16,33629
	Promedio /client/contract/delete	14,11265
	Promedio /client/contract/list-mine	7,8774
	Promedio /client/contract/publish	11,1316
	Promedio /client/contract/show	5,60356364
	Promedio /client/contract/update	9,71687143
	Promedio /client/progress-log/create	8,38836923
	Promedio /client/progress-log/delete	8,7675
	Promedio /client/progress-log/list	5,25314545
	Promedio /client/progress-log/publish	8,4891
	Promedio /client/progress-log/show	5,25651667
	Promedio /client/progress-log/update	7,07353333
	Promedio general	4,8568599



En promedio, la optimización global del sistema generó una mejora de todas las solicitudes, alcanzando un menor promedio que originalmente, aunque se observa que algunas de las solicitudes no cambian apenas como la creación o borrado de un contrato, es más visible el cambio en la entidad de progressLog.

Intervalo de Confianza

Para asegurar que los tiempos de respuesta se mantuvieran dentro de los requisitos de rendimiento, se calculó el **intervalo de confianza del 95%** para las solicitudes evaluadas. Este análisis estadístico confirma que el **tiempo de respuesta promedio** de las operaciones, después de las optimizaciones, se encuentra dentro de los límites esperados para un sistema de estas características, con una confianza del 95%.

Before			After		
Media	5,18804779		Media	5,05014896	
Error típico	0,41441949		Error típico	0,39502398	
Mediana	3,55305		Mediana	3,3424	
Moda	1,0275		Moda	#N/D	
Desviación estándar	6,03403887		Desviación estándar	5,75163597	
Varianza de la muestra	36,4096251		Varianza de la muestra	33,0813164	
Curtosis	19,0117732		Curtosis	20,3029533	
Coefficiente de asimetría	3,49372899		Coefficiente de asimetría	3,57575585	
Rango	47,7911		Rango	48,0923	
Mínimo	0,6243		Mínimo	0,6446	
Máximo	48,4154		Máximo	48,7369	
Suma	1099,86613		Suma	1070,63158	
Cuenta	212		Cuenta	212	
Nivel de confianza(95,0%)	0,81693298		Nivel de confianza(95,0%)	0,77869918	
Interval(ms)	4,37111481	6,00498077	Interval(ms)	4,27144978	5,82884813
Interval(s)	0,00437111	0,00600498	Interval(s)	0,00427145	0,00582885

El intervalo de confianza proporciona una medida de la variabilidad en los tiempos de respuesta, indicando que es poco probable que los tiempos de respuesta excedan los **60 ms**, cumpliendo así con los requisitos del sistema, que especifican que las solicitudes no deben exceder el segundo de respuesta.

Contraste de Hipótesis

Para verificar si las mejoras en el rendimiento son estadísticamente significativas, se aplicó un **contraste de hipótesis**. Este método permite evaluar si la diferencia en los tiempos de respuesta antes y después de la optimización es lo suficientemente grande como para ser atribuida a los cambios implementados (en este caso, la inclusión de índices en las consultas).

Prueba z para medias de dos muestras		
	<i>before</i>	<i>after</i>
Media	5,18692867	5,0492415
Varianza (conocida)	36,4096251	33,0813164
Observaciones	213	213
Diferencia hipotética de las medias	0	
z	0,24105669	
P(Z<=z) una cola	0,40475559	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,80951118	
Valor crítico de z (dos colas)	1,95996398	

El resultado del contraste de hipótesis mostró que las diferencias en los tiempos de respuesta entre los dos conjuntos de datos no son estadísticamente significativas. Esto confirma que las optimizaciones aplicadas no tuvieron un impacto real y medible en la mejora del rendimiento del sistema.

La falta de mejora en el rendimiento observada puede atribuirse a varios factores. Uno de ellos podría ser el tamaño de los datos de prueba. Es posible que los datos utilizados para evaluar el rendimiento no sean lo suficientemente grandes como para detectar mejoras significativas debido a los cambios implementados. Además, otros factores, como la complejidad de las consultas, la estructura de la base de datos y el entorno de ejecución, también podrían influir en los resultados.

Conclusión del análisis de rendimiento

En resumen, las pruebas de rendimiento han demostrado ser efectivas para identificar áreas de mejora en el sistema, específicamente en las operaciones que involucraban grandes cantidades de datos o validaciones complejas. La inclusión de índices en las tablas de las entidades más utilizadas resultó en una ligera mejora en el rendimiento general del sistema. Los análisis estadísticos, como el intervalo de confianza y el contraste de hipótesis, confirmaron que el sistema cumple con los requisitos de rendimiento bajo diferentes cargas.