

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Testing Report



ACME SOFTWARE FACTORY

OUR FIRST PROJECT IN D&T

Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2023 – 2024

Grupo de prácticas: C1-009

Autores por orden alfabético:

Martínez Cano, Juan

| | |
|---|--|
|  | Diseño y Pruebas II Acme-Software-Factory |
| | Testing Report |

Índice de contenido

| | |
|-----------------------------|----|
| 1. Resumen ejecutivo | 3 |
| 2. Tabla de revisiones..... | 4 |
| 3. Introducción | 5 |
| 4. Contenido..... | 6 |
| 5. Conclusiones | 9 |
| 6. Bibliografía | 11 |

| | |
|---|--|
|  | Diseño y Pruebas II Acme-Software-Factory |
| | Testing Report |

1. Resumen ejecutivo

En este informe se presenta un análisis detallado de las pruebas y los resultados obtenidos para las funcionalidades correspondientes al módulo Student 5.

| | |
|---|--|
|  | Diseño y Pruebas II Acme-Software-Factory |
| | Testing Report |

2. Tabla de revisiones

| Fecha | Versión | Descripción |
|------------|---------|--------------------------------|
| 25/05/2024 | 1.0 | Primera versión del documento. |

| | |
|---|--|
|  | Diseño y Pruebas II Acme-Software-Factory |
| | Testing Report |

3. Introducción

El documento se divide en dos partes principales: pruebas funcionales y pruebas de rendimiento. El objetivo de este informe es proporcionar un análisis detallado de los casos de prueba y del rendimiento de los mismo.

La primera parte del informe abarca las pruebas funcionales, las cuales se centran en evaluar las funcionalidades de dos entidades específicas: `code_audits` y `audits_records`. Se presentan los casos de prueba realizados, organizados por entidad, con descripciones claras y precisas para cada uno. Esto permite una evaluación concisa y efectiva de las funcionalidades asociadas a `code_audits` y `audits_records`.

En la segunda parte, el documento se centrará en proporcionar gráficos detallados que muestran los tiempos de respuesta del sistema durante las pruebas funcionales mencionadas anteriormente. Se comparará el desempeño del proyecto en dos entornos distintos, proporcionando un análisis más exhaustivo del rendimiento. Esta comparación permitirá identificar las diferencias en eficiencia y capacidad de manejo de carga entre los dos sistemas evaluados.

| | |
|---|--|
|  | Diseño y Pruebas II Acme-Software-Factory |
| | Testing Report |

4. Contenido

















Pruebas funcionales.

Requirement #6

Para este requisito, se han implementado las siguientes pruebas:

- List: En este test se ha comprobado que el listado de los code audits se realiza de manera correcta.
- Show: Se ha comprobado que al entrar en un code audit se muestran los datos específicos de ese code audit correctamente.
- Update: Se comprueba que se realizan las actualizaciones de manera correcta, por otro lado, también se ha probado que al actualizar algún campo con valores incorrectos se muestre el error pertinente.
- Delete: Se ha comprobado que se puede eliminar un code audit sin ningún problema.
- Publish: Se comprueba que se puede publicar un code audit sin ningún impedimento, así como las restricciones al publicarlos sin cumplir todas las necesidades.
- Create: Se han comprobado todas las restricciones posibles en los campos de creación de un code audit.
- Create2: Se han creado una batería de code audits que comprueben los campos a rellenar con valores máximos y mínimos así como algunos casos particulares como intentos de script injection
- Wrong-role: Este test comprueba que un user anónimo no puede realizar las acciones de un auditor.
- Wrong-user: Se ha comprobado iniciando sesión como el auditor2 no pueda acceder a los code audits del auditor1 para realizar acciones sobre estos.

El cuanto al coverage, nos queda un total del 92,5% de la funcionalidad cubierta por los test.

| | |
|---|---|
| ▼  acme.features.auditor.codeAudit |  92,5 % |
| >  AuditorCodeAuditController.java |  100,0 % |
| >  AuditorCodeAuditShowService.java |  96,9 % |
| >  AuditorCodeAuditCreateService.java |  94,3 % |
| >  AuditorCodeAuditListService.java |  93,7 % |
| >  AuditorCodeAuditUpdateService.java |  93,5 % |
| >  AuditorCodeAuditPublishService.java |  91,6 % |
| >  AuditorCodeAuditDeleteService.java |  81,8 % |

| | |
|---|--|
|  | Diseño y Pruebas II Acme-Software-Factory |
| | Testing Report |

















Como podemos observar la funcionalidad número 6 queda en prácticamente su totalidad, probando prácticamente la mayor parte de las líneas de código posible con el objetivo de asegurar una buena ejecución del código.

Requirement #7

Para este requisito, se han implementado las siguientes pruebas:

- List: En este test se ha comprobado que el listado de los audit record se realiza de manera correcta.
- Show: Se ha comprobado que al entrar en un audit record se muestran los datos específicos de ese audit record correctamente.
- Update: Se comprueba que se realizan las actualizaciones de manera correcta, por otro lado, también se ha probado que al actualizar algún campo con valores incorrectos se muestre el error pertinente.
- Delete: Se ha comprobado que se puede eliminar un audit record sin ningún problema.
- Publish: Se comprueba que se puede publicar un audit record sin ningún impedimento.
- Create: Se han comprobado todas las restricciones posibles en los campos de creación de un audit record.
- Create2: Se han creado una batería de audit record que comprueben los campos a rellenar con valores máximos y mínimos, así como algunos casos particulares ya mencionados anteriormente.
- Wrong-role: Este test comprueba que un user anónimo no puede realizar las acciones de un auditor.
- Wrong-user: Se ha comprobado iniciando sesión como el auditor2 no pueda acceder a los audit record del auditor1 para realizar acciones sobre estos.

En cuanto al coverage, se ha realizado una cobertura del 93,9% de la funcionalidad gracias a los test.

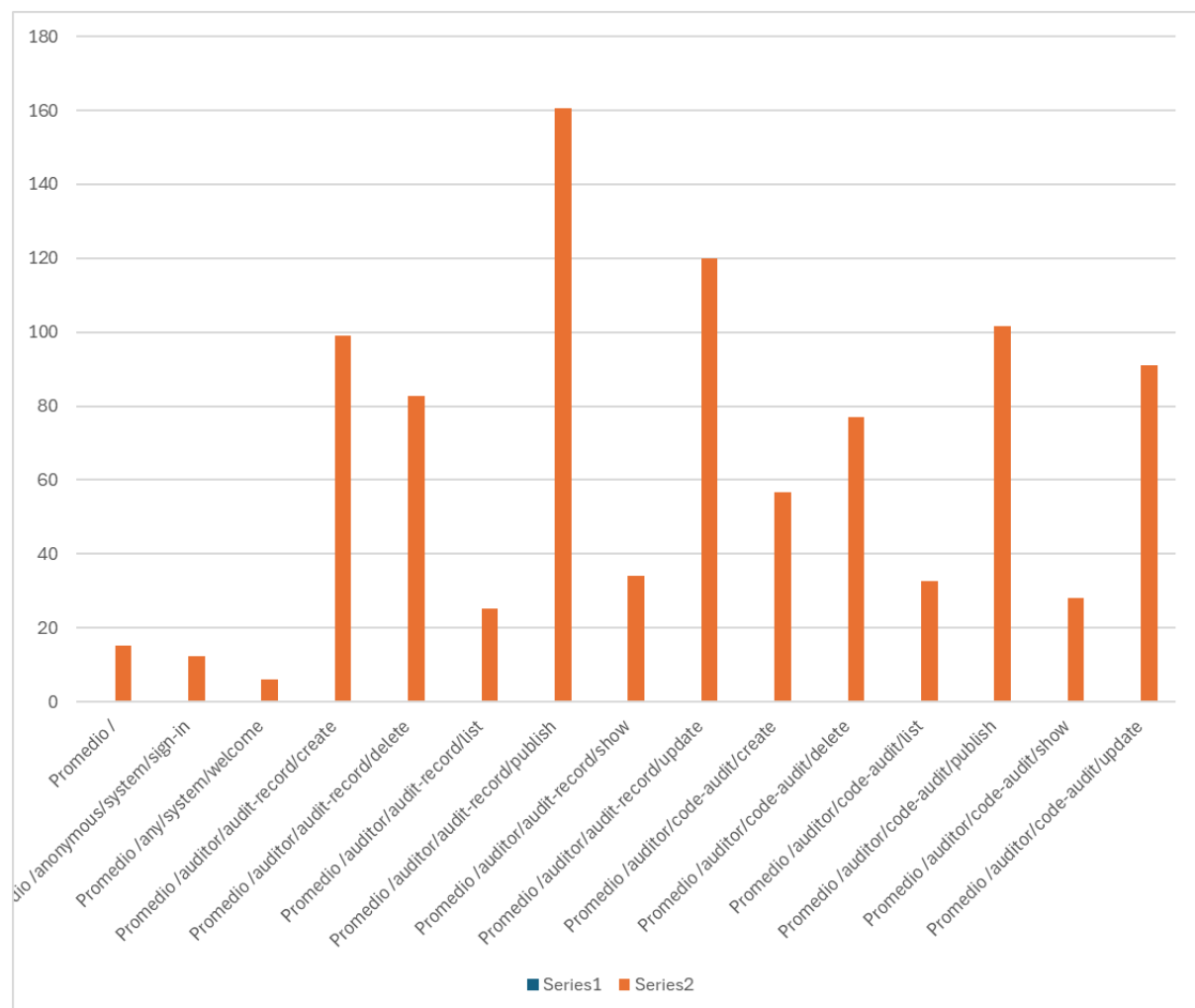
| | | |
|---|---|---------|
| ▼  acme.features.auditor.auditRecord |  | 93,9 % |
| >  AuditorAuditRecordController.java |  | 100,0 % |
| >  AuditorAuditRecordShowService.java |  | 96,0 % |
| >  AuditorAuditRecordListService.java |  | 95,8 % |
| >  AuditorAuditRecordCreateService.java |  | 95,1 % |
| >  AuditorAuditRecordPublishService.java |  | 95,0 % |
| >  AuditorAuditRecordUpdateService.java |  | 95,0 % |
| >  AuditorAuditRecordDeleteService.java |  | 80,0 % |

Como podemos observar la funcionalidad número 7 queda prácticamente cubierta, probando prácticamente todas las líneas de código posible con el objetivo de asegurar una buena ejecución del código.

Pruebas de rendimiento.

Rendimiento con pc personal


Se ha realizado un estudio del rendimiento de las pruebas anteriores, se ha generado esta gráfica para observar claramente cuáles son las pruebas más ineficientes.



Como podemos observar, las peticiones más ineficientes es claramente el publish de audit-record.

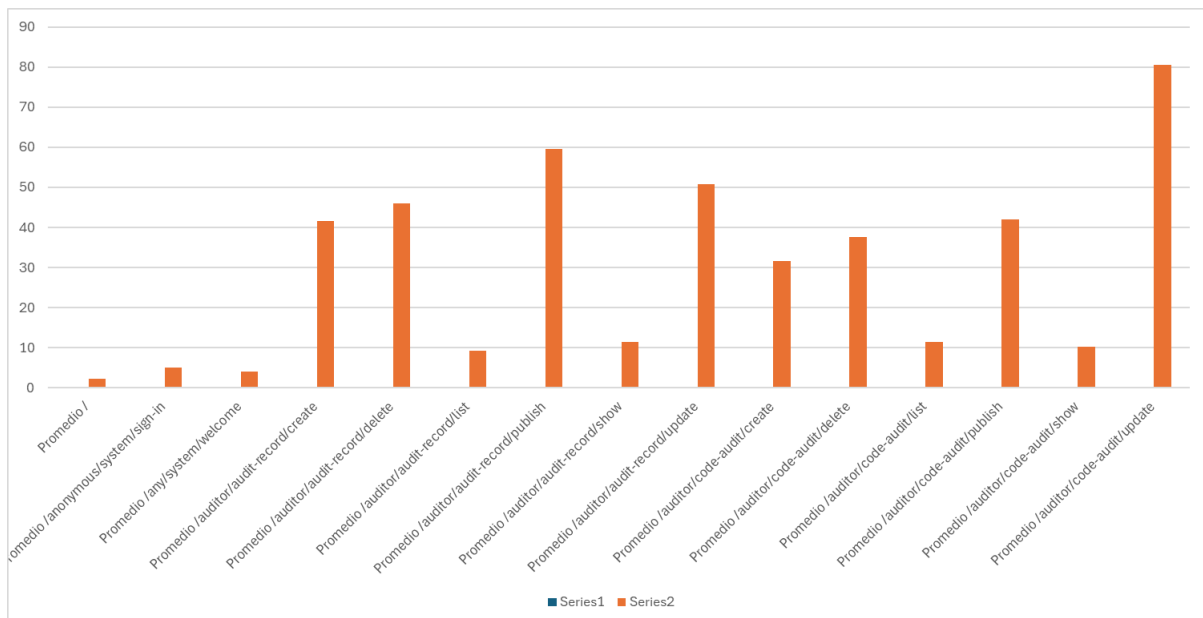
También se ha calculado el intervalo de confianza del 95 % para el tiempo que tarda en atender las solicitudes anteriores mi portátil. Nos quedaría:

| | | |
|--------------|-------------|------------|
| Interval(ms) | 32.94273749 | 44.4549275 |
| Interval(s) | 0.032942737 | 0.04445493 |

| | |
|---|--|
|  | Diseño y Pruebas II Acme-Software-Factory |
| | Testing Report |

Rendimiento con pc de Eduardo Bustamante Lucena:

Se ha vuelto a realizar un estudio del rendimiento de las pruebas anteriores, pero esta vez con el ordenador de mi compañero, se ha generado esta gráfica para observar claramente cuáles son las pruebas más ineficientes.



También se ha calculado el intervalo de confianza del 95 % para el tiempo que tarda en atender las solicitudes anteriores. Nos quedaría:

| | | |
|--------------|------------|------------|
| Promedio(ms) | 13.7257311 | 19.0068863 |
| Promedio(s) | 0.01372573 | 0.01900689 |

5. Conclusiones

En conclusión, se ha cubierto casi en su totalidad las funcionalidades requeridas mediante las pruebas mencionadas. Los casos de prueba realizados han demostrado que tanto las entidades **code_audits** como **audits_records** funcionan correctamente bajo las condiciones evaluadas. Además, el análisis de rendimiento ha revelado una clara

| | |
|---|--|
|  | Diseño y Pruebas II Acme-Software-Factory |
| | Testing Report |

mejora al ejecutar las pruebas en el ordenador de mi compañero, con un notable incremento en la eficiencia y tiempos de respuesta del sistema. Esta comparación entre diferentes entornos ha permitido identificar oportunidades para optimizar aún más el rendimiento del proyecto. En general, los resultados obtenidos son positivos y proporcionan una base sólida para futuras mejoras y desarrollos.

| | |
|---|--|
|  | Diseño y Pruebas II Acme-Software-Factory |
| | Testing Report |

6. Bibliografía

<https://ev.us.es/>