



UNIVERSIDAD  
DE SEVILLA

# E2 Testing report D04

**Group:** C1.031

**Student #2**

**Name:** Nuñez Sanchez Juan  
**Email:** juanunsan2@alum.us.es

**Repository:** <https://github.com/alvarous/Acme-One-C1.031>

## Contenido

Contenido.....	2
Resumen ejecutivo.....	3
Registro de cambios.....	4
Introducción .....	5
Testing Funcional .....	6
Test de Rendimiento .....	8
Conclusión.....	9
Bibliografía .....	10

## Resumen ejecutivo

En este documento veremos todos los casos de prueba y nos daremos cuenta de que hemos encontrado algunos bugs en view de Contract, un bug en la implementación de la API y hemos detectado problemas en la base de datos debido a la insuficiencia de pruebas en la población de los datos que dificultaban la creación de estas pruebas.

# Registro de cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Testing Report	Juan Nuñez Sanchez	08/07/2024

## Introducción

En este documento, se incluirán tanto los casos de prueba que hemos llevado a cabo como un análisis detallado de estos y una evaluación del rendimiento de nuestro proyecto en un ordenador de gama media.

## Testing Funcional

Primero comentaremos las diferentes pruebas realizadas para realizar testing funcional para las entidades `contract` y `progressLogs` creadas por el estudiante:

### CONTRACT.

**Listar:** Para verificar que la funcionalidad de listado funciona correctamente, iniciamos sesión como cliente y accedimos a la lista de contratos. Tras confirmar que los campos se muestran correctamente, dimos por concluida la prueba. Esta prueba no detectó ningún bug.









**Mostrar:** Para asegurarnos de que la función de visualización funciona adecuadamente, iniciamos sesión como cliente y accedimos a la lista de contratos para luego ingresar en uno de ellos. Tras verificar que los campos se muestran de manera correcta, dimos por finalizada la prueba. Esta prueba no detectó ningún bug.

**Crear:** Para comprobar que la creación de un contrato funciona correctamente, probamos diferentes posibilidades: intentamos crearlo vacío, crearlo con todos los campos válidos excepto uno vacío, y crear un contrato ya existente. También probamos cada campo de entrada uno por uno con diferentes valores no válidos. Detectamos un bug que impedía la creación de un contrato vacío y provocaba un error 500, el cual ha sido solucionado.

**Eliminar:** Para verificar que la eliminación funciona correctamente, accedimos a un contrato sin publicar y lo eliminamos. No se pudo probar la función de "unbind" ya que no es posible eliminar un contrato de un contrato ya publicado.

**Actualizar:** Para asegurar el correcto funcionamiento de la actualización, repetimos exactamente las mismas pruebas realizadas para la creación de un contrato. Estas pruebas no detectaron ningún bug.

**Publicar:** Para verificar la funcionalidad de publicación de contratos, repetimos las mismas pruebas que en la creación y actualización. Además, probamos cada campo de entrada uno por uno con diferentes valores no válidos.

acme.features.client.contract		90,0 %	1.316	146	1.462
> ClientContractUpdateService.java		88,0 %	324	44	368
> ClientContractCreateService.java		87,0 %	287	43	330
> ClientContractPublishService.java		89,9 %	320	36	356
> ClientContractDeleteService.java		91,4 %	128	12	140
> ClientContractShowService.java		95,7 %	156	7	163
> ClientContractListAllService.java		94,3 %	66	4	70
> ClientContractController.java		100,0 %	35	0	35

## Progress Logs.

**Listar:** Para verificar que la funcionalidad de listado funciona correctamente, iniciamos sesión como cliente y accedimos a la lista de contratos, entramos en uno de ellos y accedemos a sus progress logs. Tras confirmar que los campos se muestran correctamente, dimos por concluida la prueba. Esta prueba no detectó ningún bug.

**Mostrar:** Para asegurarnos de que la función de visualización funciona adecuadamente, iniciamos sesión como cliente y accedimos a la lista de contratos, entramos en uno de ellos y accedemos a sus progress logs y accedemos a uno de ellos. Tras verificar que los campos se muestran de manera correcta, dimos por finalizada la prueba. Esta prueba no detectó ningún bug.

**Crear:** Para comprobar que la creación de un progress log funciona correctamente, probamos diferentes posibilidades: intentamos crearlo vacío, crearlo con todos los campos válidos excepto uno vacío, y crear un contrato ya existente. También probamos cada campo de entrada uno por uno con diferentes valores no válidos. No se detectó ningún bug.

**Eliminar:** Para verificar que la eliminación de un progress log funciona correctamente, accedimos a un progress log no publicado y lo eliminamos. No se pudo probar la función de "unbind" ya que no es posible eliminar un progress log ya publicado.

**Actualizar:** Para asegurar el correcto funcionamiento de la actualización, repetimos exactamente las mismas pruebas realizadas para la creación de un progress log. Estas pruebas no detectaron ningún bug.

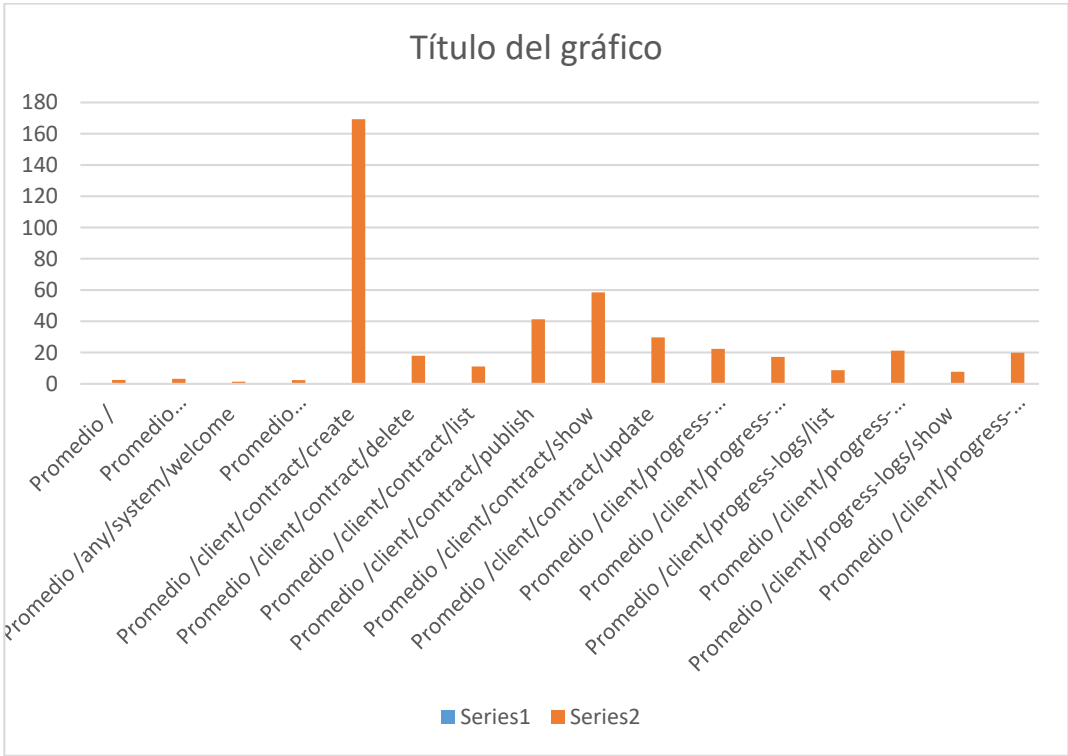
**Publicar:** Para verificar la funcionalidad de publicación de progress logs, repetimos las mismas pruebas que en la creación y actualización.

acme.features.client.progresslog	92,0 %	1.056	92	1.148
> ClientProgressLogsCreateServ	90,6 %	202	21	223
> ClientProgressLogsUpdateSer	91,3 %	211	20	231
> ClientProgressLogsDeleteSer	86,8 %	112	17	129
> ClientProgressLogsPublishSer	93,5 %	246	17	263
> ClientProgressLogsListService	92,6 %	151	12	163
> ClientProgressLogsShowServi	95,2 %	99	5	104
> ClientProgressLogsController	100,0 %	35	0	35

## Test de Rendimiento.

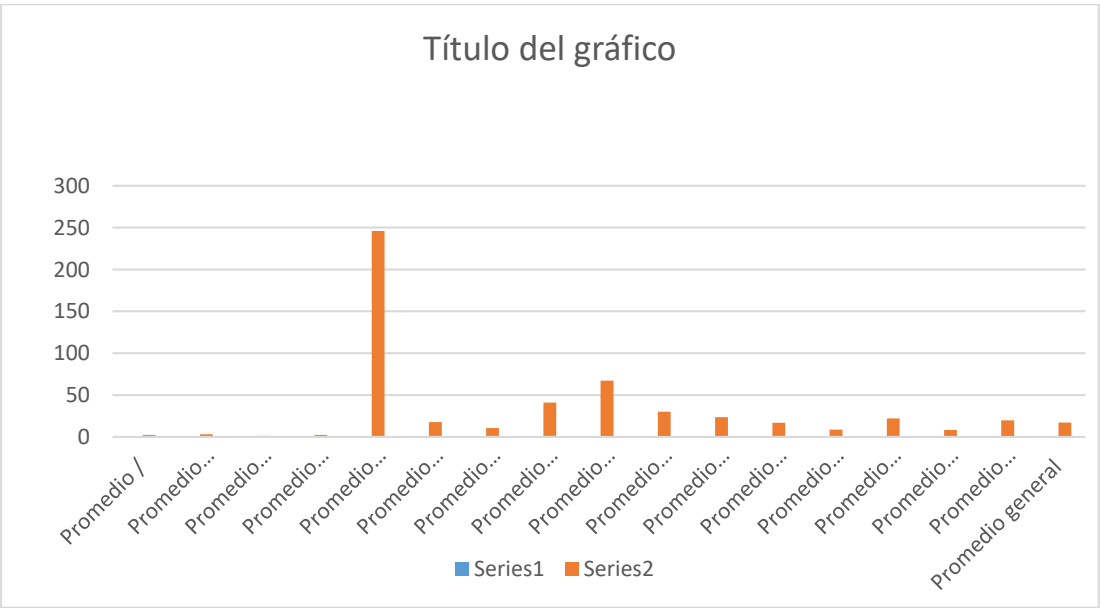
Para el test de rendimiento se han registrado los tiempos promedio que ha tomado la aplicación en responder a cada solicitud. Este gráfico presenta esos resultados.

Esta primera sin Índices:



Como podemos observar, la petición más ineficiente con diferencia es la de crear contratos. Se ha deducido que la causa de esto puede ser porque tiene que crear el objeto desde 0 y tiene que acceder a la lista de proyectos para asignárselo al contrato.

Estos son los resultados para el promedio con Índices:







## Conclusión

En conclusión, aunque realizar pruebas puede ser una tarea tediosa y repetitiva, hacerlo de manera sistemática es fundamental para detectar errores y fallos en el proyecto. Además, el análisis de rendimiento nos permite ver claramente cómo la introducción de índices en la base de datos mejora el funcionamiento, aunque para haberlo notado más en este caso hubiera sido necesario que los índices tuvieran más peso en las llamadas. En general, estas actividades son muy beneficiosas y esenciales durante el desarrollo de un producto de software.

## Bibliografía

Intencionalmente en blanco.