

**Testing report**

C3.034

<https://github.com/juanlo017/c1.034>

| **Apellidos, Nombre** | **Email** |
| --- | --- |
| Castro Vázquez, Jose Ignacio | joscasvaz@alum.us.es |

**Tabla de contenidos**

[**Resumen ejecutivo 3**](#_43v012my8e3e)

[**Tabla de revisiones 4**](#_ouk80crf36ih)

[**Introducción 5**](#_u0bfbulml1a6)

[**Contenidos 6**](#_8fv8g9nag1vc)

[Testing Funcional 6](#_16uf1f4hu30m)

[Contract (Client) 6](#_afzejltayomc)

[Progress Logs (Client) 15](#_s5w0xf15j4g2)

[Testing de rendimiento 22](#_n58wbmsbsymj)

[**Conclusiones 27**](#_obajb8esrr12)

[**Bibliografía 28**](#_588w00qf3urg)

# 

# **Resumen ejecutivo**

En este documento se detalla todo el proceso de testing formal de los requisitos #6 y #7 del Student #2.

# **Tabla de revisiones**

| **Versión** | **Descripción** | **Fecha** |
| --- | --- | --- |
| v1.0 | Creación del documento | 14/05/2024 |
| v1.1 | Versión final julio | 06/07/2024 |
| v2.0 | Versión final Third Call (octubre) | 20/12/2024 |

# **Introducción**

Se ha procedido a testear formalmente los requisitos #6 y #7 del Student #2.

# 

# **Contenidos**

## **Testing Funcional**

En esta sección se listan los casos de prueba de las funcionalidades con una descripción de cómo se han llevado a cabo y si han sido efectivos a la hora de encontrar bugs.

## **Contract (Client)**

| Funcionalidad | Descripción | Bugs |
| --- | --- | --- |
| LIST-SAFE | Navegar como usuario con rol client a la url, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. | No se ha encontrado ningún bug. |
| LIST-HACK | Navegar como usuario sin rol o rol distinto a client a la url, comprobar que efectivamente no muestra la vista y que lanza error el sistema. | No se ha encontrado ningún bug. |
| SHOW-SAFE | Navegar como usuario con rol client y dueño de la entidad, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. | No se ha encontrado ningún bug. |
| SHOW-HACK | Navegar como usuario sin rol client o con un client ajeno a la entidad a la url, comprobar que efectivamente no se muestra la vista y que lanza error el sistema. | No se ha encontrado ningún bug. |
| CREATE-SAFE | * Navegar como usuario con rol client y dueño de la entidad, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. * Probar todos los inputs del formulario de acuerdo con la metodología dada en clase:   CODE:  Probar sin código  Probar sólo con el guión  Probar sin número  Probar sin letras  Probar sin guión  Probar con letras de más  Probar con menos números  Probar con números de más  Probar con letras de otro charset  Probar código duplicado en la bbdd  Probar con código válido  PROVIDER NAME:  Probar vacío  Probar código html  Probar código de lógica  Probar con otros charset  Probar con 1 carácter  Probar con 2 caracteres (min+1)  Probar con el máx+1  Probar con el máx  CUSTOMER NAME:  Probar vacío  Probar código html  Probar código de lógica  Probar con otros charset  Probar con 1 carácter  Probar con 2 caracteres (min+1)  Probar con el máx+1  Probar con el máx  GOALS:  Probar vacío  Probar código html  Probar código de lógica  Probar con otros charset  Probar con 1 carácter  Probar con 2 caracteres (min+1)  Probar con el máx+1  Probar con el máx  BUDGET:  Probar sin budget  Probar budget negativo  Probar con 0  Probar con 1.01  Probar con el máximo +0.01  Probar con el máximo, que es el budget que falta para alcanzar el coste del proyecto  Probar con una divisa distinta a la del coste del proyecto  Probar con una divisa inválida  PROJECT:  Probar vacío  Probar con un proyecto del desplegable | No se ha encontrado ningún bug. |
| UPDATE-SAFE | * Navegar como usuario con rol client y dueño de la entidad, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. * Probar todos los inputs del formulario de acuerdo con la metodología dada en clase:   CODE:  Probar sin código  Probar sólo con el guión  Probar sin número  Probar sin letras  Probar sin guión  Probar con letras de más  Probar con menos números  Probar con números de más  Probar con letras de otro charset  Probar código duplicado en la bbdd  Probar con el antiguo código  Probar con nuevo código válido  PROVIDER NAME:  Probar vacío  Probar código html  Probar código de lógica  Probar con otros charset  Probar con 1 carácter  Probar con 2 caracteres (min+1)  Probar con el máx+1  Probar con el máx  CUSTOMER NAME:  Probar vacío  Probar código html  Probar código de lógica  Probar con otros charset  Probar con 1 carácter  Probar con 2 caracteres (min+1)  Probar con el máx+1  Probar con el máx  GOALS:  Probar vacío  Probar código html  Probar código de lógica  Probar con otros charset  Probar con 1 carácter  Probar con 2 caracteres (min+1)  Probar con el máx+1  Probar con el máx  BUDGET:  Probar sin budget  Probar budget negativo  Probar con 0  Probar con 1.01  Probar con el máximo +0.01  Probar con el máximo, que es el budget que falta para alcanzar el coste del proyecto  Probar con una divisa distinta a la del coste del proyecto  Probar con una divisa inválida  PROJECT:  Probar vacío  Probar con un proyecto del desplegable | No se ha encontrado ningún bug. |
| PUBLISH-SAFE | * Navegar como usuario con rol client y dueño de la entidad, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. * Probar todos los inputs del formulario de acuerdo con la metodología dada en clase:   CODE:  Probar sin código  Probar sólo con el guión  Probar sin número  Probar sin letras  Probar sin guión  Probar con letras de más  Probar con menos números  Probar con números de más  Probar con letras de otro charset  Probar código duplicado en la bbdd  Probar con el antiguo código  Probar con nuevo código válido  PROVIDER NAME:  Probar vacío  Probar código html  Probar código de lógica  Probar con otros charset  Probar con 1 carácter  Probar con 2 caracteres (min+1)  Probar con el máx+1  Probar con el máx  CUSTOMER NAME:  Probar vacío  Probar código html  Probar código de lógica  Probar con otros charset  Probar con 1 carácter  Probar con 2 caracteres (min+1)  Probar con el máx+1  Probar con el máx  GOALS:  Probar vacío  Probar código html  Probar código de lógica  Probar con otros charset  Probar con 1 carácter  Probar con 2 caracteres (min+1)  Probar con el máx+1  Probar con el máx  BUDGET:  Probar sin budget  Probar budget negativo  Probar con 0  Probar con 1.01  Probar con el máximo +0.01  Probar con el máximo, que es el budget que falta para alcanzar el coste del proyecto  Probar con una divisa distinta a la del coste del proyecto  Probar con una divisa inválida  PROJECT:  Probar vacío  Probar con un proyecto del desplegable | No se ha encontrado ningún bug. |
| DELETE-SAFE | * Navegar como usuario con rol client y dueño de la entidad, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. * Eliminar el contract de bbdd | No se ha encontrado ningún bug. |

## 

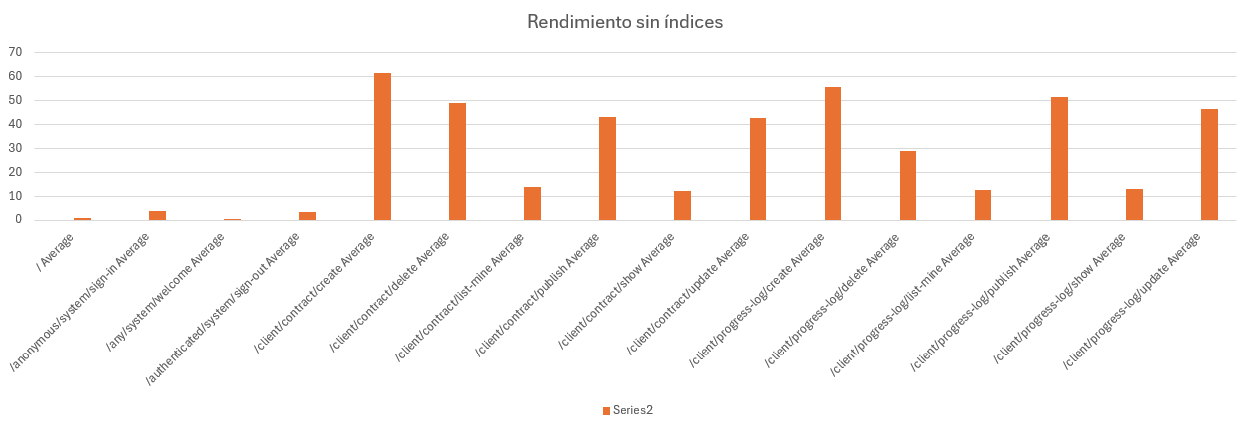
## **Progress Logs (Client)**

| Funcionalidad | Descripción | Bugs |
| --- | --- | --- |
| LIST-SAFE | Navegar como usuario con rol client a la url, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. | No se ha encontrado ningún bug. |
| LIST-HACK | Navegar como usuario sin rol, rol distinto a client o con un client ajeno a la url y comprobar que efectivamente no muestra la vista y que lanza error el sistema. | No se ha encontrado ningún bug. |
| SHOW-SAFE | Navegar como usuario con rol client y dueño de la entidad, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. | No se ha encontrado ningún bug. |
| SHOW-HACK | Navegar como usuario sin rol client o con un client ajeno a la entidad a la url, comprobar que efectivamente no se muestra la vista y que lanza error el sistema. | No se ha encontrado ningún bug. |
| CREATE-SAFE | * Navegar como usuario con rol client y dueño de la entidad, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. * Probar todos los inputs del formulario de acuerdo con la metodología dada en clase:   RECORD ID:  Probar sin record id  Probar con un guión  Probar con los dos guines  Probar sin número  Probar sin letras  Probar con dos letras al comienzo distintas de “PG”  Probar sin el primer guión  Probar sin el segundo guión  Probar con letras de más al principio  Probar con letras de más en medio  Probar con menos números  Probar con números de más  Probar con caracteres de otro charset  Probar record id duplicado en la bbdd  COMPLETENESS:  Probar sin completeness  Probar completeness negativo  Probar con 0  Probar con 0.99  Probar con 0.01  Probar con 1.01  COMMENT:  Probar vacío  Probar código html  Probar código de lógica  Probar con otro charset  Probar con el máx+1  Probar con el máx  Probar con 1 carácter  RESPONSIBLE PERSON:  Probar vacío  Probar código html  Probar código de lógica  Probar con otros charset  Probar con 1 carácter  Probar con 2 caracteres (min+1)  Probar con el máx+1  Probar con el máx  CONTRACT:  Probar vacío  Probar con un contract del desplegable | No se ha encontrado ningún bug. |
| UPDATE-SAFE | * Navegar como usuario con rol client y dueño de la entidad, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. * Probar todos los inputs del formulario de acuerdo con la metodología dada en clase:   RECORD ID:  Probar sin record id  Probar con un guión  Probar con los dos guines  Probar sin número  Probar sin letras  Probar con dos letras al comienzo distintas de “PG”  Probar sin el primer guión  Probar sin el segundo guión  Probar con letras de más al principio  Probar con letras de más en medio  Probar con menos números  Probar con números de más  Probar con caracteres de otro charset  Probar record id duplicado en la bbdd  Probar con el antiguo record id  Probar con un nuevo record id válido  COMPLETENESS:  Probar sin completeness  Probar completeness negativo  Probar con 0  Probar con 0.99  Probar con 0.01  Probar con 1.01  COMMENT:  Probar vacío  Probar código html  Probar código de lógica  Probar con otro charset  Probar con el máx+1  Probar con el máx  Probar con 1 carácter  RESPONSIBLE PERSON:  Probar vacío  Probar código html  Probar código de lógica  Probar con otro charset  Probar con el máx+1  Probar con el máx  Probar con 1 carácter  CONTRACT:  Probar vacío  Probar con un contract del desplegable | No se ha encontrado ningún bug. |
| PUBLISH-SAFE | * Navegar como usuario con rol client y dueño de la entidad, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. * Probar todos los inputs del formulario de acuerdo con la metodología dada en clase:   RECORD ID:  Probar sin record id  Probar con un guión  Probar con los dos guines  Probar sin número  Probar sin letras  Probar con dos letras al comienzo distintas de “PG”  Probar sin el primer guión  Probar sin el segundo guión  Probar con letras de más al principio  Probar con letras de más en medio  Probar con menos números  Probar con números de más  Probar con caracteres de otro charset  Probar record id duplicado en la bbdd  Probar con el antiguo record id  Probar con un nuevo record id válido  COMPLETENESS:  Probar sin completeness  Probar completeness negativo  Probar con 0  Probar con 0.99  Probar con un decimal pequeño  Probar con 1.01  COMMENT:  Probar vacío  Probar código html  Probar código de lógica  Probar con otro charset  Probar con el máx+1  Probar con el máx  Probar con 1 carácter  RESPONSIBLE PERSON:  Probar vacío  Probar código html  Probar código de lógica  Probar con otro charset  Probar con el máx+1  Probar con el máx  Probar con 1 carácter  CONTRACT:  Probar vacío  Probar con un contract del desplegable |  |
| DELETE-SAFE | * Navegar como usuario con rol client y dueño de la entidad, comprobar que efectivamente se muestra la vista y no lanza ningún error el sistema. * Eliminar el progress log de bbdd | No se ha encontrado ningún bug. |

## 

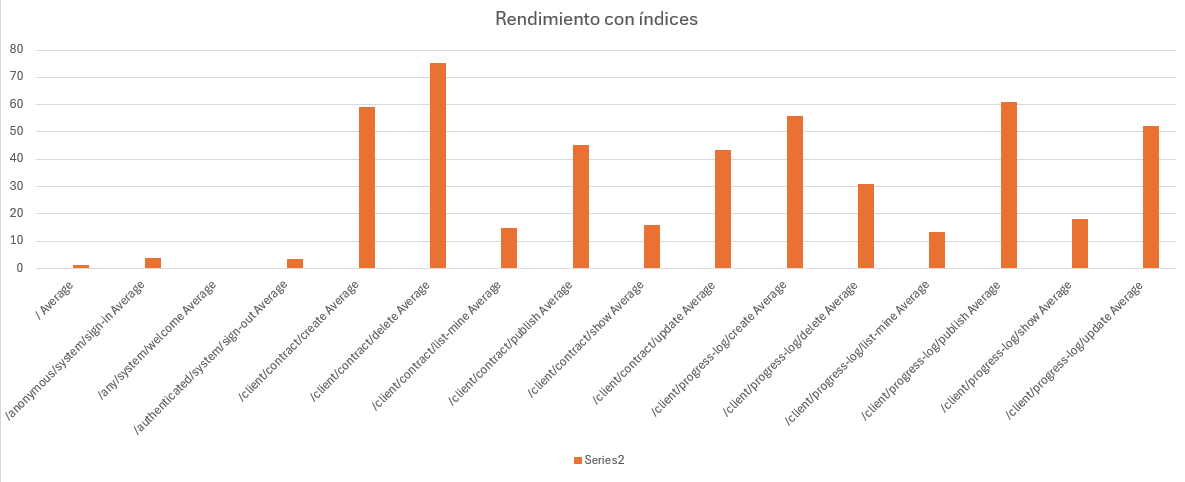
## **Testing de rendimiento**

A continuación la siguiente tabla muestra los tiempos de respuesta de cada petición del sistema por funcionalidad antes de la optimización del sistema:

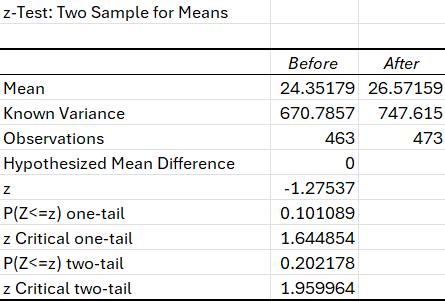


Vemos que los tiempos de peticiones entran en el intervalo de aceptación del cliente.

A continuación la misma tabla tras haber incluido índices en las tablas de la base de datos:



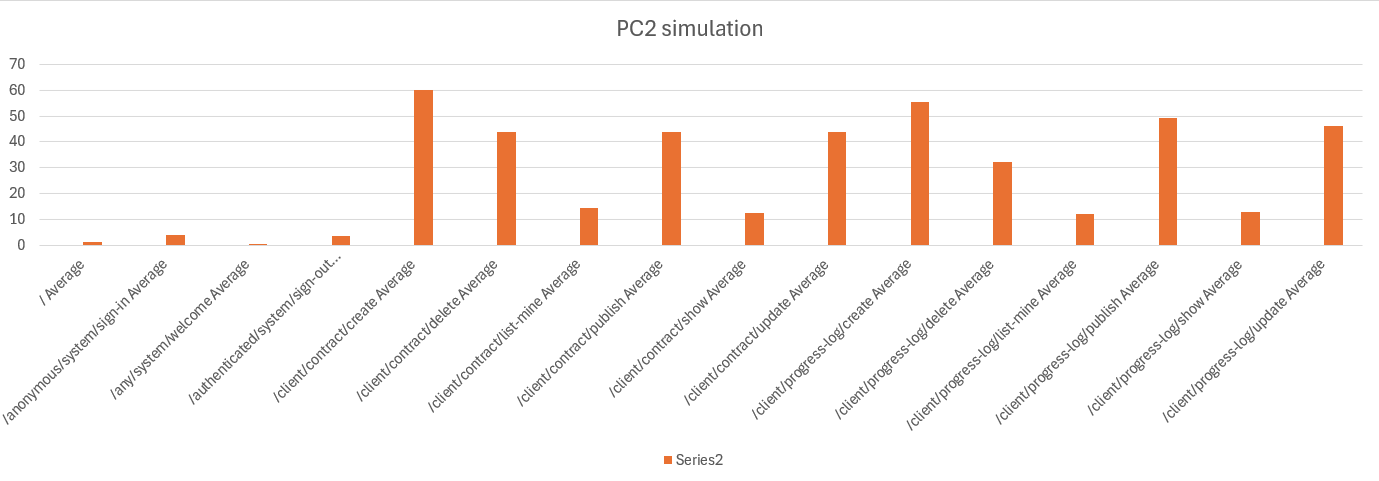
Vemos que los tiempos de peticiones entran en el intervalo de aceptación del cliente y se puede apreciar que los tiempos apenas varían, salvo algunos que han aumentado de manera poco significativa. Veremos el contraste con confianza 95%:



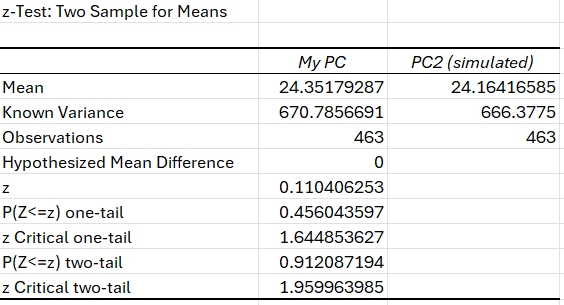
Como se puede observar de los datos sacados del documento el z-analysis muestra un valor 0.202178, superior al delta de 0.05, por lo que podemos concluir que el sistema no ha mejorado tras la implementación de índices en base de datos pese a haber intentado aumentar el rendimiento con distintas combinaciones de dichos índices.

Finalmente, veremos el z-analysis del mismo sistema. Como en esta convocatoria (octubre) soy el único integrante de mi grupo que se presenta, siguiendo con las indicaciones del anexo del project statement, le sumaré y/o restaré aleatoriamente el 10% a los tiempos promedio de request en mi equipo para simular el comportamiento de otro ordenador.

La siguiente tabla muestra los tiempos de respuesta de cada petición del sistema usando la versión optimizada (sin anotaciones) simulando el comportamiento de otro equipo como mencionamos en el párrafo anterior y siguiendo las indicaciones del anexo. Para ello hemos empleado la función RAND() de Excel, que genera un número aleatorio entre 0 y 1. Con la fórmula =IF(RAND()<0.5,0.9,1.1) lo que haremos es evaluar si ese número aleatorio está o no por encima de 0.5. En caso de no estarlo, restamos un 10% multiplicando por 0.9 el antiguo valor, y análogamente en el caso contrario se lo sumaremos multiplicando por 1.1. El resultado nos arroja la siguiente gráfica:



A continuación veremos el contraste con confianza 95% entre las pruebas ejecutadas en nuestro equipo y la simulación:



Como era de esperar, observando el valor de alfa podemos concluir que aleatoreizar los tiempos sumando o restando el 10% de su valor no ha supuesto un cambio significativo en los tiempos (está cerca de 1).

# 

# **Conclusiones**

En conclusión, en esta entrega hemos testeado a fondo las funcionalidades implementadas por el Student #2, donde hemos encontrado y subsanado algunos bugs. También pudimos analizar el rendimiento de nuestra aplicación y mejorarla gracias a los índices de la base de datos, aunque dicha mejora no sea relevante como nos muestra el z-analysis. Hemos comparado el rendimiento del sistema con una simulación siguiendo las instrucciones del anexo y hemos comprobado gracias al z-analysis que no hay diferencias significativas con los tiempos originales.

# 

# **Bibliografía**

Intencionalmente en blanco.