

# DP2 2023-2024

## Testing report

### Acme-SF-D04

Repositorio: <https://github.com/andpizcer/DP2-C1.032>

#### Miembro:

- Avilés Carrera, Francisco, [fraavicar1@alum.us.es](mailto:fraavicar1@alum.us.es)

**Tutor:** Patricia Jiménez Aguirre

GRUPO C1.032 (G3 teoría)

27/05/2024

# Historial de versiones

| Fecha      | Versión | Descripción de los cambios   | Sprint |
|------------|---------|--|--------|
| 19/05/2024 | V1      | <ul style="list-style-type: none"><li>Creación del documento.</li></ul>                              | 4      |
| 16/05/2024 | V2      | <ul style="list-style-type: none"><li>Adición de las pruebas funcionales.</li></ul>                  | 4      |
| 25/05/2024 | V3      | <ul style="list-style-type: none"><li>Adición de las pruebas de rendimiento y conclusiones</li></ul> | 4      |

## Índice

|                                |    |
|--------------------------------|----|
| 1. Resumen ejecutivo .....     | 4  |
| 2. Introducción .....          | 4  |
| 3. Pruebas funcionales .....   | 4  |
| 4. Pruebas de rendimiento..... | 11 |
| 5. Conclusiones.....           | 16 |
| 6. Bibliografía .....          | 16 |

# 1. Resumen ejecutivo

En este documento se detallará el requisito individual del reporte de las pruebas. Está organizado en dos capítulos: pruebas funcionales y pruebas de rendimiento.

## 2. Introducción

El informe de pruebas contendrá en el primer capítulo un listado con los casos de prueba realizados. También se resaltarán su eficacia para encontrar errores. En el segundo y último capítulo se proporcionarán los gráficos adecuados y un intervalo de confianza del 95% del tiempo que tarda el proyecto en servir las pruebas funcionales en dos ordenadores diferentes. Además, se incluirá un contraste de hipótesis del 95% de confianza sobre qué ordenador es más potente.

## 3. Pruebas funcionales

En primer lugar, detallaré las pruebas realizadas para la feature `/client/contract`. Primeramente, explicaré los archivos `.safe`:

- **List:**
  - 1º) Listar los contratos.
- **Show:**
  - 1º) Mostrar un contrato en draftMode.
  - 2º) Mostrar un contrato público.
- **Create:**
  - 1º) Intentar crear un contrato sin datos.
  - 2º) Intentar crear varios contratos, con valores que no son válidos para cada atributo. Para el atributo código pruebo el valor nulo, uno que esté cogido y dos con formato incorrecto. Para los atributos String pruebo los valores nulo y el máximo posible más uno. Para el atributo presupuesto pruebo los valores nulo, con un presupuesto negativo, con únicamente la moneda, con únicamente la cantidad del presupuesto, con un presupuesto mayor al máximo posible, con una moneda de un dígito y con una moneda de dos dígitos. Para el atributo proyecto pruebo el valor nulo.
  - 3º) Crear un contrato con datos válidos.
  - 4º) Crear 5 contratos más con datos válidos. En cada uno de ellos se solaparán los valores mínimo, mínimo más uno, uno intermedio, el máximo menos uno y el máximo de cada uno de los atributos String (nombre de cliente, nombre de proveedor y objetivos) y numéricos (presupuesto). Pondré valores variados en el atributo código y en el proyecto asociado.
  - 5º) Crear un contrato con intento de hacking en los Strings.

- **Update:**

1º) Intentar actualizar un contrato sin datos.

2º) Intentar actualizar varias veces un contrato, con valores que no son válidos para cada atributo. Para el atributo código pruebo el valor nulo, uno que esté cogido y dos con formato incorrecto. Para los atributos String pruebo los valores nulo y el máximo posible más uno. Para el atributo presupuesto pruebo los valores nulo, con un presupuesto negativo, con únicamente la moneda, con únicamente la cantidad del presupuesto, con un presupuesto mayor al máximo posible, con una moneda de un dígito y con una moneda de dos dígitos. Para el atributo proyecto pruebo el valor nulo.

3º) Actualizar un contrato con datos válidos.

4º) Actualizar 5 contratos más con datos válidos. En cada uno de ellos se solaparán los valores mínimo, mínimo más uno, uno intermedio, el máximo menos uno y el máximo de cada uno de los atributos String (nombre de cliente, nombre de proveedor y objetivos) y numéricos (presupuesto). Pondré valores variados en el atributo código y en el proyecto asociado.

5º) Actualizar un contrato con intento de hacking en los Strings.

- **Publish:**

1º) Intentar publicar un contrato sin datos.

2º) Intentar publicar varias veces un contrato, con valores que no son válidos para cada atributo. Para el atributo código pruebo el valor nulo, uno que esté cogido y dos con formato incorrecto. Para los atributos String pruebo los valores nulo y el máximo posible más uno. Para el atributo presupuesto pruebo los valores nulo, con un presupuesto negativo, con únicamente la moneda, con únicamente la cantidad del presupuesto, con una cantidad que supere el presupuesto del proyecto y con una moneda que no sea con la que está medida el presupuesto. Para el atributo proyecto pruebo el valor nulo.

3º) Publicar un contrato con datos válidos.

4º) Publicar 5 contratos más con datos válidos. En cada uno de ellos se solaparán los valores mínimo, mínimo más uno, uno intermedio, el máximo menos uno y el máximo de cada uno de los atributos String (nombre de cliente, nombre de proveedor y objetivos). El presupuesto y el proyecto no lo tocaré para no pasarme del máximo coste que se puede asociar a ese proyecto. Pondré valores variados en el atributo código.

5º) Publicar un contrato con intento de hacking en los Strings.

- **Delete:**

1º) Borrar un contrato.

A continuación, detallaré los archivos **.hack**:

- **List:**
  - 1º) Intentar listar los contratos sin haber iniciado sesión.
- **Show:**
  - 1º) Intentar mostrar un contrato que no existe sin haber iniciado sesión.
  - 2º) Intentar mostrar un contrato que existe sin haber iniciado sesión.
  - 3º) Intentar mostrar un contrato que no existe siendo un cliente.
  - 4º) Intentar mostrar un contrato que no está publicado siendo un cliente, pero el contrato es de otro cliente.
- **Create:**
  - 1º) Intentar crear un contrato sin haber iniciado sesión.
- **Update:**
  - 1º) Intentar actualizar un contrato sin haber iniciado sesión.
  - 2º) Intentar actualizar un contrato que no existe siendo un cliente.
  - 3º) Intentar actualizar un contrato que ya está publicado siendo un cliente y siendo mi contrato.
  - 4º) Intentar actualizar un contrato siendo un cliente, pero el contrato no es mío.
- **Publish:**
  - 1º) Intentar publicar un contrato sin haber iniciado sesión.
  - 2º) Intentar publicar un contrato que no existe siendo un cliente.
  - 3º) Intentar publicar un contrato que ya está publicado siendo un cliente y siendo mi contrato.
  - 4º) Intentar publicar un contrato siendo un cliente, pero el contrato no es mío.
- **Delete:**
  - 1º) Intentar eliminar un contrato sin haber iniciado sesión.
  - 2º) Intentar eliminar un contrato que no existe siendo un cliente.
  - 3º) Intentar eliminar un contrato que ya está publicado siendo un cliente y siendo mi contrato.
  - 4º) Intentar eliminar un contrato siendo un cliente, pero el contrato no es mío.

Con la realización de las pruebas no se han encontrado **errores** graves, pero sí he retocado algunas cosas. Gracias a la realización de la prueba **publicar** de **.safe**, me di cuenta de que se me olvidó declarar el mensaje (la label) que debía aparecer cuando un contrato se pasaba del coste permitido. También, esta misma prueba, me ayudó a darme cuenta de que la función que comprobaba si con el coste de mi contrato el coste del proyecto era igual o menor, no era muy eficaz. Esto es debido a que yo recorría los proyectos en el servicio, algo que cambié e introduje una consulta en el repositorio.

Por último, detallaré la pruebas realizadas para la feature `/client/progressLog`. Primeramente, explicaré los archivos `.safe`:

- **List:**
  - 1º) Listar los progress logs.
- **Show:**
  - 1º) Mostrar un progress log en draftMode.
  - 2º) Mostrar un progress log público.
- **Create:**
  - 1º) Intentar crear un progress log sin datos.
  - 2º) Intentar crear varios progress log, con valores que no son válidos para cada atributo. Para el atributo recordId pruebo el valor nulo, uno que esté cogido y tres con formato incorrecto. Para los atributos String pruebo los valores nulo y el máximo posible más uno. Para el atributo completeness pruebo los valores nulo, con un porcentaje negativo, con porcentaje igual a 0, con un porcentaje mayor a 100 y con porcentaje igual o menor al mayor porcentaje publicado para los progress logs de ese contrato.
  - 3º) Crear un progress log con datos válidos.
  - 4º) Crear 5 progress logs más con datos válidos. En cada uno de ellos se solaparán los valores mínimo, mínimo más uno, uno intermedio, el máximo menos uno y el máximo de cada uno de los atributos String (comentario y persona responsable) y numéricos (completeness). Pondré valores variados en el atributo recordId.
  - 5º) Crear un progress log con intento de hacking en los Strings.
- **Update:**
  - 1º) Intentar actualizar un progress log sin datos.
  - 2º) Intentar actualizar varios progress log, con valores que no son válidos para cada atributo. Para el atributo recordId pruebo el valor nulo, uno que esté cogido y tres con formato incorrecto. Para los atributos String pruebo los valores nulo y el máximo posible más uno. Para el atributo completeness pruebo los valores nulo, con un porcentaje negativo, con porcentaje igual a 0, con un porcentaje mayor a 100 y con porcentaje igual o menor al mayor porcentaje publicado para los progress logs de ese contrato.
  - 3º) Actualizar un progress log con datos válidos.
  - 4º) Actualizar 5 progress logs más con datos válidos. En cada uno de ellos se solaparán los valores mínimo, mínimo más uno, uno intermedio, el máximo menos uno y el máximo de cada uno de los atributos String (comentario y persona responsable) y numéricos (completeness). Pondré valores variados en el atributo recordId.
  - 5º) Actualizar un progress log con intento de hacking en los Strings.

- **Publish:**

- 1º) Intentar publicar un progress log sin datos.
- 2º) Intentar publicar varios progress log, con valores que no son válidos para cada atributo. Para el atributo recordId pruebo el valor nulo, uno que esté cogido y tres con formato incorrecto. Para los atributos String pruebo los valores nulo y el máximo posible más uno. Para el atributo completeness pruebo los valores nulo, con un porcentaje negativo, con porcentaje igual a 0, con un porcentaje mayor a 100 y con porcentaje igual o menor al mayor porcentaje publicado para los progress logs de ese contrato.
- 3º) Publicar un progress log con datos válidos.
- 4º) Publicar 5 progress logs más con datos válidos. En cada uno de ellos se solaparán los valores mínimo, mínimo más uno, uno intermedio, el máximo menos uno y el máximo de cada uno de los atributos String (comentario y persona responsable) y numéricos (completeness). Pondré valores variados en el atributo recordId.
- 5º) Publicar un progress log con intento de hacking en los Strings.

- **Delete:**

- 1º) Borrar un progress log.

A continuación, detallaré los archivos **.hack:**

- **List**

- 1º) Intentar listar los progress logs de un contrato que no exista sin haber iniciado sesión.
- 2º) Intentar listar los progress logs de un contrato que no esté publicado sin haber iniciado sesión.
- 3º) Intentar listar los progress logs de un contrato publicado sin haber iniciado sesión.
- 4º) Intentar listar los progress logs de un contrato que no exista siendo un cliente.
- 5º) Intentar listar los progress logs de un contrato que no esté publicado siendo un cliente.
- 6º) Intentar listar los progress logs de un contrato publicado siendo un cliente, pero el contrato no es mío.

- **Show:**

- 1º) Intentar mostrar un progress log que no existe sin haber iniciado sesión.
- 2º) Intentar mostrar un progress log que existe sin haber iniciado sesión.
- 3º) Intentar mostrar un progress log que no existe siendo un cliente.
- 4º) Intentar mostrar un progress log que no está publicado siendo un cliente, pero el progress log es de otro cliente.



- **Create:**
  - 1º) Intentar crear un progress log de un contrato que no exista sin haber iniciado sesión.
  - 2º) Intentar crear un progress log de un contrato que no esté publicado sin haber iniciado sesión.
  - 3º) Intentar crear un progress log de un contrato publicado sin haber iniciado sesión.
  - 4º) Intentar crear un progress log de un contrato que no exista siendo un cliente.
  - 5º) Intentar crear un progress log de un contrato que no esté publicado siendo un cliente.
  - 6º) Intentar crear un progress log de un contrato publicado siendo un cliente, pero el contrato no es mío.
- **Update:**
  - 1º) Intentar actualizar un progress log sin haber iniciado sesión.
  - 2º) Intentar actualizar un progress log que no existe siendo un cliente.
  - 3º) Intentar actualizar un progress log que ya está publicado siendo un cliente y siendo mi contrato.
  - 4º) Intentar actualizar un progress log siendo un cliente, pero el progress log no es mío y está publicado.
  - 5º) Intentar actualizar un progress log siendo un cliente, pero el progress log no es mío y no está publicado.
- **Publish:**
  - 1º) Intentar publicar un progress log sin haber iniciado sesión.
  - 2º) Intentar publicar un progress log que no existe siendo un cliente.
  - 3º) Intentar publicar un progress log que ya está publicado siendo un cliente y siendo mi contrato.
  - 4º) Intentar publicar un progress log siendo un cliente, pero el progress log no es mío y está publicado.
  - 5º) Intentar publicar un progress log siendo un cliente, pero el progress log no es mío y no está publicado.
- **Delete:**
  - 1º) Intentar eliminar un progress log sin haber iniciado sesión.
  - 2º) Intentar eliminar un progress log que no existe siendo un cliente.
  - 3º) Intentar eliminar un progress log que ya está publicado siendo un cliente y siendo mi contrato.
  - 4º) Intentar eliminar un progress log siendo un cliente, pero el progress log no es mío y está publicado.
  - 5º) Intentar eliminar un progress log siendo un cliente, pero el progress log no es mío y no está publicado.

Con la realización de estas pruebas no se han encontrado **errores** graves, pero si he eliminado o añadido algunas cosas. Primeramente, he añadido progress logs a contratos del cliente 2, porque si no era muy difícil realizar algunas pruebas. Gracias a la prueba de **listar** del **.safe** me di cuenta de que no era necesario guardar un booleano para mostrar el botón de crear o no otro progress log. Esto se debe a que esa pantalla únicamente es accesible para los clientes que hayan creado el contrato de esos progress logs, algo que ya se comprueba en el **authorise**. También quiero destacar la eliminación, gracias a las pruebas de **crear**, **actualizar** y **editar** de **.safe**, de la validación que compruebe que el progress log se está asociando a un contrato publicado. No es necesaria ya que coge la id de este de la petición hecha anteriormente.

## 4. Pruebas de rendimiento

Primeramente, tengo que decir que con la introducción de los Índices fue necesario añadir una ordenación determinada en una consulta del repositorio de los progress logs, para que cuando se comprobaran en sus tests, la ordenación de estos en el listado tuviera el mismo orden que antes de los Índices y no me saltaran errores FAILED. La ordenación introducida es por el id de los progress logs, ya que me di cuenta de que antes se ordenaban así automáticamente.

Los tests han sido probados en mi ordenador (Student2) y en los del Student3.

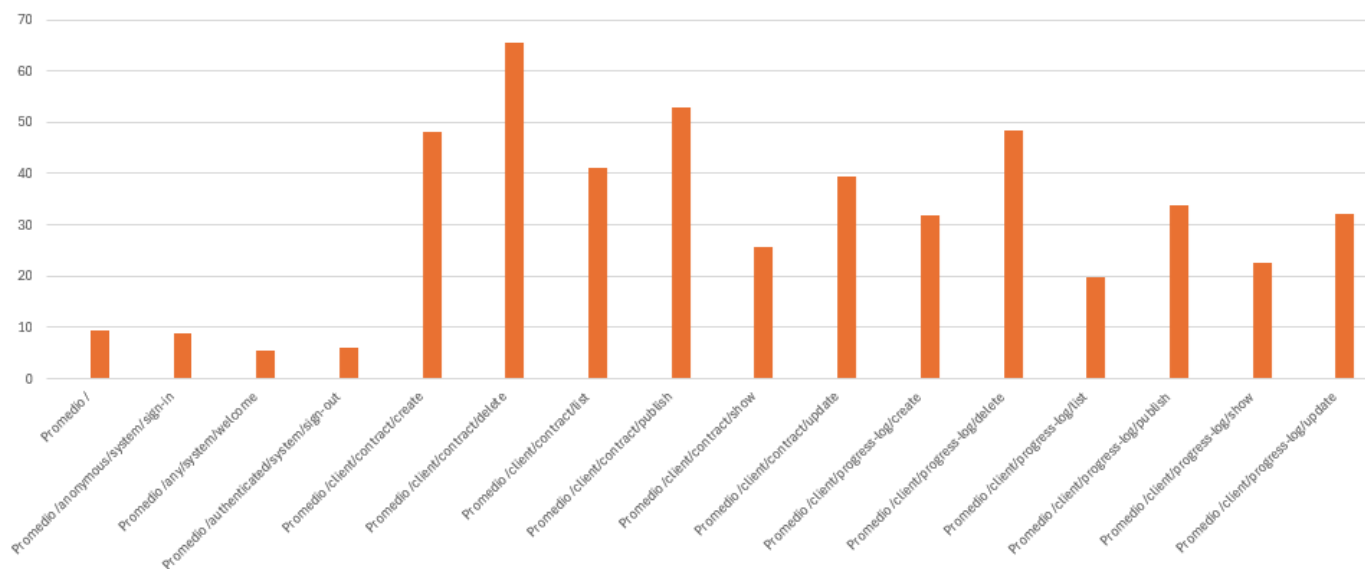
En primer lugar, detallaré los resultados obtenidos en **mi ordenador**, cuyas especificaciones son las siguientes:

- **Procesador:** Cpu Intel Core I5 4ª Gen 4210u / 1.7Ghz.
- **Número de núcleos:** Dual-core.
- **Almacenamiento principal:** 1Tb Hdd / 5400 Rpm.
- **Almacenamiento añadido:** 240Gb SSD.
- **Memoria RAM principal:** Ram 4Gb ( 1 X 4Gb ), 1600Mhz.
- **Memoria RAM añadida:** Ram DDR3 4Gb, 1600Mhz.

A continuación se expondrán los resultados obtenidos.

1º) Promedio de las peticiones, con su correspondiente gráfica:

| request-path                            | time       |
|---|------------|
| Promedio /                              | 9,45340851 |
| Promedio /anonymous/system/sign-in      | 8,89541702 |
| Promedio /any/system/welcome            | 5,31498776 |
| Promedio /authenticated/system/sign-out | 5,8688     |
| Promedio /client/contract/create        | 48,1381588 |
| Promedio /client/contract/delete        | 65,48965   |
| Promedio /client/contract/list          | 40,9673528 |
| Promedio /client/contract/publish       | 52,8973933 |
| Promedio /client/contract/show          | 25,6097586 |
| Promedio /client/contract/update        | 39,2796533 |
| Promedio /client/progress-log/create    | 31,9354057 |
| Promedio /client/progress-log/delete    | 48,518     |
| Promedio /client/progress-log/list      | 19,7004217 |
| Promedio /client/progress-log/publish   | 33,7668577 |
| Promedio /client/progress-log/show      | 22,4386125 |
| Promedio /client/progress-log/update    | 31,9914192 |
| Promedio general                        | 27,230224  |



## 2º) Intervalo de confianza del 95%:

|                           |             |              |            |            |
|---------------------------|-------------|--------------|------------|------------|
| Media                     | 27,23022395 | Interval(ms) | 25,1623365 | 29,2981114 |
| Error típico              | 1,052631714 | Interval(s)  | 0,02516234 | 0,02929811 |
| Mediana                   | 23,2119     |              |            |            |
| Moda                      | 28,3034     |              |            |            |
| Desviación estándar       | 24,14178192 |              |            |            |
| Varianza de la muestra    | 582,8256342 |              |            |            |
| Curtosis                  | 22,82531261 |              |            |            |
| Coefficiente de asimetría | 3,294989921 |              |            |            |
| Rango                     | 253,9073    |              |            |            |
| Mínimo                    | 2,2563      |              |            |            |
| Máximo                    | 256,1636    |              |            |            |
| Suma                      | 14323,0978  |              |            |            |
| Cuenta                    | 526         |              |            |            |
| Nivel de confianza(95,0%) | 2,06788748  |              |            |            |

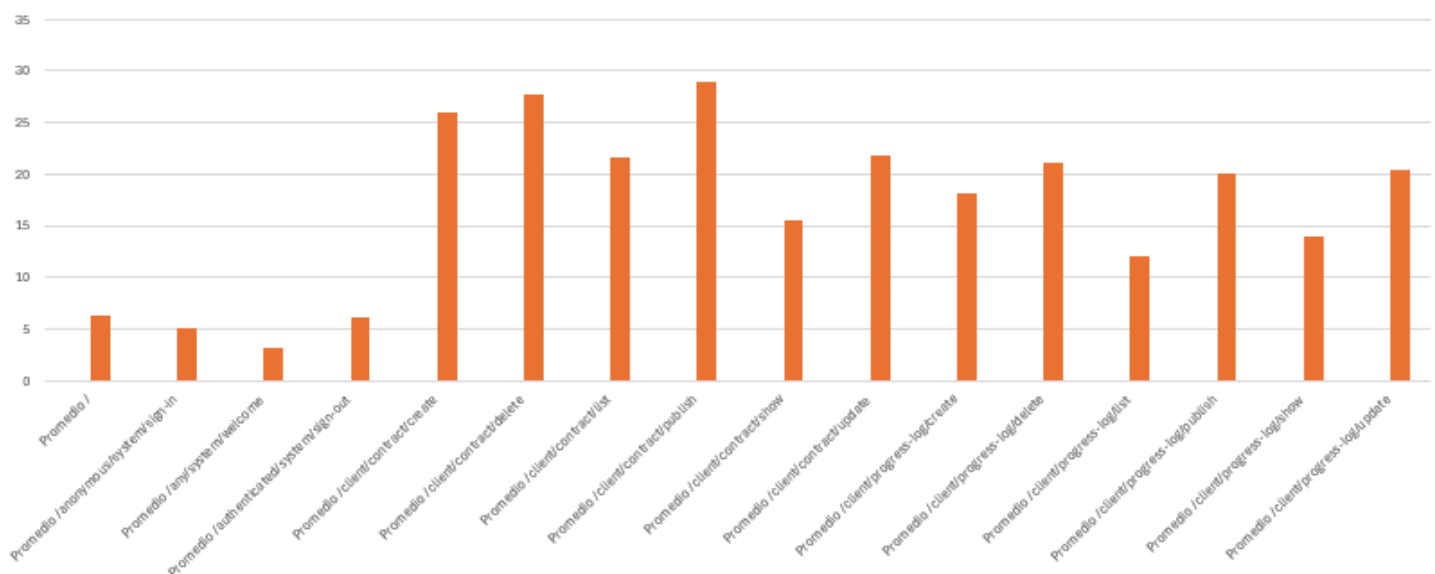
En segundo lugar, detallaré los resultados obtenidos en el **ordenador del Student3**, cuyas especificaciones son las siguientes:

- **Procesador:** AMD Ryzen™ 5 7520U / 2.8Ghz.
- **Número de núcleos:** 4.
- **Almacenamiento principal:** 512GB SSD.
- **Memoria RAM principal:** LPDDR5 16GB.

A continuación se expondrán los resultados obtenidos.

1º) **Promedio de las peticiones, con su correspondiente gráfica:**

| request-path                            | time        |
|---|-------------|
| Promedio /                              | 6,290440426 |
| Promedio /anonymous/system/sign-in      | 5,088612766 |
| Promedio /any/system/welcome            | 3,167377551 |
| Promedio /authenticated/system/sign-out | 6,2034      |
| Promedio /client/contract/create        | 25,96562059 |
| Promedio /client/contract/delete        | 27,77988333 |
| Promedio /client/contract/list          | 21,5526566  |
| Promedio /client/contract/publish       | 28,99429    |
| Promedio /client/contract/show          | 15,48293793 |
| Promedio /client/contract/update        | 21,88569333 |
| Promedio /client/progress-log/create    | 18,19361143 |
| Promedio /client/progress-log/delete    | 21,08591667 |
| Promedio /client/progress-log/list      | 12,00747826 |
| Promedio /client/progress-log/publish   | 19,99888462 |
| Promedio /client/progress-log/show      | 13,99426875 |
| Promedio /client/progress-log/update    | 20,36982308 |
| Promedio general                        | 15,44766236 |



2º) Intervalo de confianza del 95%:

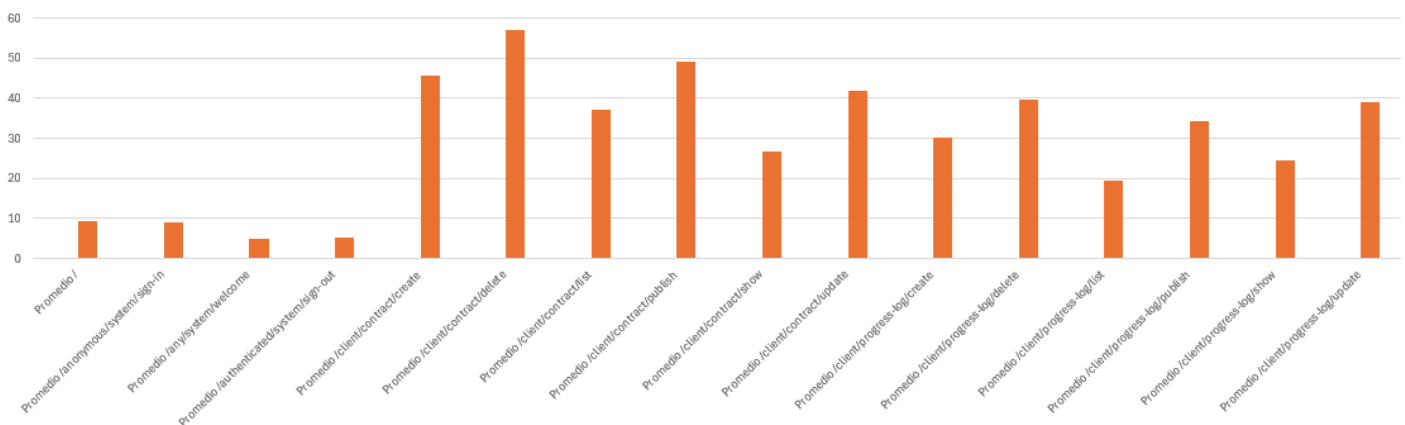
|                           |             |  |              |            |            |
|---------------------------|-------------|--|--------------|------------|------------|
|                           |             |  | Interval(ms) | 14,4285662 | 16,4667585 |
| Media                     | 15,44766236 |  | Interval(s)  | 0,01442857 | 0,01646676 |
| Error típico              | 0,51875786  |  |              |            |            |
| Mediana                   | 15,18415    |  |              |            |            |
| Moda                      | #N/D        |  |              |            |            |
| Desviación estándar       | 11,89755064 |  |              |            |            |
| Varianza de la muestra    | 141,5517112 |  |              |            |            |
| Curtosis                  | 35,89302461 |  |              |            |            |
| Coficiente de asimetría   | 3,885189552 |  |              |            |            |
| Rango                     | 145,2351    |  |              |            |            |
| Mínimo                    | 1,7536      |  |              |            |            |
| Máximo                    | 146,9887    |  |              |            |            |
| Suma                      | 8125,4704   |  |              |            |            |
| Cuenta                    | 526         |  |              |            |            |
| Nivel de confianza(95,0%) | 1,019096109 |  |              |            |            |

Por último, se expondrá un **contraste de hipótesis del 95%** sobre qué ordenador es más potente:

|                                      |                  |                  |
|--------------------------------------|------------------|------------------|
| Prueba z para medias de dos muestras |                  |                  |
|                                      |                  |                  |
|                                      | <i>Student 2</i> | <i>Student 3</i> |
| Media                                | 27,23022395      | 15,44766236      |
| Varianza (conocida)                  | 582,8256342      | 141,5517112      |
| Observaciones                        | 526              | 526              |
| Diferencia hipotética de las medias  | 0                |                  |
| z                                    | 10,04038026      |                  |
| P(Z<=z) una cola                     | 0                |                  |
| Valor crítico de z (una cola)        | 1,644853627      |                  |
| Valor crítico de z (dos colas)       | 0                |                  |
| Valor crítico de z (dos colas)       | 1,959963985      |                  |

Comenzando por analizar el contraste de hipótesis, tenemos que comprobar si el valor crítico de  $z$  (dos colas) está en el intervalo  $[0, 0.05)$ , siendo  $0.05$  igual a  $1$  menos el nivel de confianza (95%). Como es  $0$ , podemos comparar las medias de las peticiones en un ordenador y otro. Como se puede comprobar, la media del segundo ordenador (15) es prácticamente la mitad que la del primero (27), siendo este infinitamente más rápido. Esto se debe a las especificaciones de uno y otro, siendo mi ordenador de hace 10 años y el de mi compañero recientemente adquirido.

Analizando las gráficas, en ambas se puede comprobar que las peticiones que más tardan son las de crear y publicar un contrato, algo lógico teniendo en cuenta que son los servicios más largos. Resaltan la tardanza de las peticiones de eliminar en ambas entidades. Echando un vistazo a ambos servicios, he podido comprobar que en el servicio de eliminar un contrato, también eliminaba los progress logs asociados a él previamente. Esto es algo innecesario debido a que si un contrato se puede eliminar significa que no está publicado, no pudiendo tener tampoco progress logs asociados. Corrigiendo esto, las gráficas en mi ordenador al correr otra vez los tests quedan de la siguiente manera, disminuyendo el promedio que se tarda en eliminar un contrato. Aunque sigue siendo de las peticiones que mas tardan, ha disminuido casi 10 ms, tardando antes 65 ms y ahora 56 ms.



## 5. Conclusiones

Concluyendo, las pruebas funcionales realizadas para las features `/client/contract` y `/client/progressLog` han demostrado ser eficaces en la detección y corrección de errores. A través de diversas pruebas se ha asegurado el funcionamiento y seguridad de las funcionalidades implementadas. La implementación de correcciones, como la optimización con las consultas, ha resultado en un sistema más eficiente. En general, el sistema ha demostrado ser capaz de manejar los casos de uso esperados y resistir intentos de hacking, con ajustes menores necesarios para mejorar el rendimiento.

Por otro lado, las pruebas de rendimiento han revelado diferencias significativas en la capacidad de procesamiento entre dos ordenadores con diferentes especificaciones. En concreto, los resultados muestran que el ordenador más moderno es mucho más rápido, con un tiempo promedio de respuesta casi la mitad del ordenador más antiguo. Además, el análisis de las peticiones ha destacado que las operaciones de creación y publicación son las más lentas, lo cual es lógico debido a la complejidad de estos procesos.

## 6. Bibliografía

Intencionalmente en blanco.