**26/05/2024**

**Reporte de Testing**

Logotipo

Descripción generada automáticamente

**Grupo:** C1.39

**Repositorio:** <https://github.com/pabalcber/C1.039-Acme-SF>

**Integrantes:**

|  |  |  |
| --- | --- | --- |
| Nombre | Apellidos | Correo Corporativo |
| María del Mar | Ávila Maqueda | maravimaq@alum.us.es |
|  |  |  |

Tabla de versiones:

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción de los cambios | Sprint |
| 26/05/2024 | 1.0 | Creación del documento y redacción de sus contenidos | 4 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **1.Índice**

[**1.Índice** 3](#_Toc167629311)

[**2. Resumen Ejecutivo** 5](#_Toc167629312)

[**3.Tabla de revisiones** 6](#_Toc167629313)

[**4. Introducción** 7](#_Toc167629314)

[**5.Pruebas Funcionales** 8](#_Toc167629315)

[5.1. Create Contract 8](#_Toc167629316)

[5.2. Update Contract 9](#_Toc167629317)

[5.3 Publish Contract 9](#_Toc167629318)

[5.4 Delete Contract 10](#_Toc167629319)

[5.5 List Contract 11](#_Toc167629320)

[5.6 Show Contract 12](#_Toc167629321)

[5.7. Create ProgressLog 12](#_Toc167629322)

[5.8. Update ProgressLog 13](#_Toc167629323)

[5.9 Delete ProgressLog 15](#_Toc167629324)

[5.10 List ProgressLog 15](#_Toc167629325)

[5.11 Show ProgressLog 16](#_Toc167629326)

[**6.Pruebas de Rendimiento** 18](#_Toc167629327)

[6.1. Gráficos e Intervalos de Confianza 18](#_Toc167629328)

[6.2. Contraste de Hipótesis 21](#_Toc167629329)

[6.3. Procedimiento **¡Error! Marcador no definido.**](#_Toc167629330)

[**7.Conclusión** 24](#_Toc167629331)

[**8.Bibliografía** 25](#_Toc167629332)

# **2. Resumen Ejecutivo**

Este informe de pruebas resume los resultados de las pruebas realizadas en el proyecto del estudiante 2. En el apartado de pruebas funcionales, se presentan los casos de prueba implementados, agrupados por funcionalidad, con una evaluación de su efectividad en la detección de errores. Por otro lado, el apartado de pruebas de rendimiento proporciona gráficos y un intervalo de confianza del 95% para el tiempo de ejecución en diferentes computadoras, junto con un contraste de hipótesis sobre su potencia relativa. La metodología utilizada fue rigurosa, apoyada en herramientas como Eclipse y una planificación sistemática. Este informe busca ofrecer una visión clara de la calidad y el rendimiento de nuestro software.

# **3.Tabla de revisiones**

|  |  |  |
| --- | --- | --- |
| Número de revisión | Fecha | Descripción |
| 1 | 27/05/2024 | Revisión final antes de la entrega |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# **4. Introducción**

En el ámbito del desarrollo de software, las pruebas son cruciales para garantizar la calidad y el rendimiento del producto final. Este informe presenta los resultados de las pruebas funcionales y de rendimiento, llevadas a cabo en el contexto del estudiante 2 de nuestro proyecto. En el apartado de pruebas funcionales, se detallan los casos de prueba implementados y su capacidad para detectar errores en diversas funcionalidades del software. Por otro lado, en el apartado de pruebas de rendimiento, se analiza el tiempo de ejecución en diferentes entornos, proporcionando insights sobre la eficiencia y la capacidad del sistema. Este informe busca proporcionar una evaluación objetiva y completa de la calidad del software desarrollado.

# **5.Pruebas Funcionales**

El testing del estudiante 2 se ha llevado a cabo en la rama **feature/S2-Man09** donde **funcionan perfectamente**. Se deja indicado porque justo antes de entregarlo algunos compañeros tocaron su csv, y es probable que repercuta a la hora de ejecutar los mismos al haberlos grabado con el recorder, por lo que puede ser conveniente revisarlos y ejecutarlos en la rama **feature/S2-Man09**. Los tests implementados son los siguientes:

## 5.1. Create Contract

**Caso de Prueba 1: Creación Exitosa**

* **Descripción:** Se prueba el escenario en el que el usuario completa todos los campos obligatorios correctamente y presiona el botón de crear contrato.
* **Efectividad en la Detección de Errores:** Este caso de prueba detecta de manera efectiva posibles fallos en la lógica de validación de campos obligatorios y en la funcionalidad de creación de contratos.

**Caso de Prueba 2: Campos Obligatorios Vacíos**

* **Descripción:** Se verifica cómo el sistema maneja el intento de crear un contrato con campos obligatorios vacíos.
* **Efectividad en la Detección de Errores:** Este caso de prueba identifica de manera efectiva la validación de campos obligatorios y garantiza que el sistema responda correctamente a esta situación.

**Caso de Prueba 3: Datos Inválidos**

* **Descripción:** Se ingresa datos inválidos en uno o más campos del formulario de creación de contrato y se evalúa la respuesta del sistema.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo para identificar posibles fallos en la validación de datos y la gestión de errores al intentar crear un contrato con información incorrecta.

**Caso de Prueba 4: Límites de Entrada**

* **Descripción:** Se evalúa cómo el sistema maneja los límites de entrada para campos como la longitud máxima del cliente del contrato.
* **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para garantizar que el sistema pueda manejar correctamente los límites de entrada y evitar posibles desbordamientos o errores de formato.

**Caso de Prueba 5: Hacking**

* **Descripción:** Se evalúa cómo el sistema maneja los posibles Get Hacking y Post Hacking(proyecto no publicado).
* **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para garantizar que el sistema no es vulnerable a usuarios con intenciones maliciosas.

## 5.2. Update Contract

**Caso de Prueba 1: Actualización Exitosa**

* **Descripción:** Se prueba el escenario en el que el usuario selecciona un contrato existente, realiza cambios válidos en los campos y confirma la actualización.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo para detectar posibles errores en la lógica de actualización de contratos y asegurar que los cambios se reflejen correctamente en el sistema.

**Caso de Prueba 2: Datos Inválidos en la Actualización**

* **Descripción:** Se ingresa información inválida en uno o más campos al intentar actualizar un contrato y se verifica la respuesta del sistema.
* **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para detectar posibles fallos en la validación de datos durante la actualización de contratos, garantizando la integridad de la información.

**Caso de Prueba 3: Actualización de Campos Obligatorios Vacíos**

* **Descripción:** Se intenta actualizar un contrato con campos obligatorios vacíos y se observa cómo el sistema responde.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo para asegurar que el sistema maneje correctamente la actualización de contratos con campos obligatorios vacíos y valide adecuadamente la información ingresada.

**Caso de Prueba 4: Hacking**

* **Descripción:** Se prueba si el sistema permite actualizar contratos ya publicados y contratos de otros usuarios.
* **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para garantizar que el sistema no es vulnerable a usuarios con intenciones maliciosas.

## 5.3 Publish Contract

**Caso de Prueba 1: Publicación Exitosa**

* **Descripción:** Se prueba el escenario en el que el usuario selecciona un contrato existente y lo publica con éxito.
* **Efectividad en la Detección de Errores:** Este caso de prueba verifica de manera efectiva que el contrato seleccionado se publique correctamente, sin errores en el proceso de publicación.

**Caso de Prueba 2: Strings Inválidos**

* **Descripción:** Se ingresa información inválida en uno o más campos al intentar publicar un contrato y se verifica la respuesta del sistema.
* **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para detectar posibles fallos en la validación de datos durante la actualización de contratos, garantizando la integridad de la información.

**Caso de Prueba 3: Validación del Presupuesto**

* **Descripción:** Se ingresan datos de presupuesto inválidos al intentar publicar un contrato y se verifica la respuesta del sistema.
* **Efectividad en la Detección de Errores:** Este caso de prueba asegura que el sistema valide correctamente el presupuesto ingresado para el contrato, evitando publicaciones con presupuestos incorrectos.

**Caso de Prueba 4: Moneda Inválida**

* **Descripción:** Se intenta publicar un contrato con una moneda no válida y se evalúa la respuesta del sistema.
* **Efectividad en la Detección de Errores:** Este caso de prueba es importante para garantizar que el sistema detecte y maneje adecuadamente monedas no válidas al intentar publicar un contrato.

**Caso de Prueba 5: Actualización de Campos Obligatorios Vacíos**

* **Descripción:** Se intenta publicar un contrato con campos obligatorios vacíos y se observa cómo el sistema responde.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo para asegurar que el sistema maneje correctamente la publicación de contratos con campos obligatorios vacíos y valide adecuadamente la información ingresada.

**Caso de Prueba 6: Hacking**

* **Descripción:** Se prueba si el sistema permite publicar contratos de otros usuarios.
* **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para garantizar que el sistema no es vulnerable a usuarios con intenciones maliciosas.

## 5.4 Delete Contract

**Caso de Prueba 1: Eliminar contrato existente**

* **Descripción:** Se prueba la capacidad del sistema para eliminar un contrato existente.
* **Efectividad en la Detección de Errores:** El caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede eliminar correctamente un contrato existente sin generar errores o inconsistencias en la base de datos.

**Caso de Prueba 2: Verificar autorización para eliminar contrato**

* **Descripción:** Se verifica si el sistema valida correctamente la autorización del cliente para eliminar un contrato. La eliminación solo debería estar permitida si el contrato está en modo borrador y el cliente que realiza la solicitud es el propietario del contrato.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que garantiza que solo los clientes que sean propietarios de contratos en modo borrador puedan eliminarlos, evitando así eliminaciones no autorizadas.

**Caso de Prueba 3: Hacking**

* **Descripción:** Se prueba si el sistema permite eliminar contratos ya publicados.
* **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para garantizar que el sistema no es vulnerable a usuarios con intenciones maliciosas.

## 5.5 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.
* **Efectividad en la Detección de Errores:** El caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede recuperar correctamente los contratos asociados a un cliente y mostrarlos en la interfaz de usuario.

**Caso de Prueba 2: Verificar autorización para listar contratos**

* **Descripción:** Se verifica si el sistema valida correctamente la autorización para listar los contratos de un cliente. Todos los clientes deberían poder ver solo sus propios contratos.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que garantiza que solo los contratos asociados al cliente que realiza la solicitud sean listados, evitando así la revelación de información confidencial de otros clientes.

**Caso de Prueba 3: Verificar formato de los datos mostrados**

* **Descripción:** Se prueba si el sistema muestra correctamente los datos de los contratos listados en el formato esperado.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que verifica que los datos mostrados para cada contrato estén en el formato adecuado y sean fácilmente comprensibles para el usuario final.

## 5.6 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.
* **Efectividad en la Detección de Errores:** El caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede recuperar correctamente los detalles del contrato solicitado y mostrarlos en la interfaz de usuario.

**Caso de Prueba 2: Verificar autorización para mostrar detalles del contrato**

* **Descripción:** Se verifica si el sistema valida correctamente la autorización para mostrar los detalles de un contrato. Solo se deben mostrar los detalles de contratos que pertenezcan al cliente que realiza la solicitud.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que garantiza que solo se muestren los detalles de los contratos no en modo borrador y que pertenezcan al cliente que realiza la solicitud, evitando así la exposición de información confidencial de otros clientes.

**Caso de Prueba 3: Verificar formato de los datos mostrados**

* **Descripción:** Se prueba si el sistema muestra correctamente los detalles del contrato en el formato esperado.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que verifica que los detalles del contrato se muestren en el formato adecuado y sean fácilmente comprensibles para el usuario final.

## 5.7. Create ProgressLog

**Caso de Prueba 1: Crear registro de progreso válido**

* **Descripción:** Se prueba la capacidad del sistema para crear un registro de progreso válido para un contrato específico.
* **Efectividad en la Detección de Errores:** El caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede crear correctamente un registro de progreso con datos válidos proporcionados por el usuario.

**Caso de Prueba 2: Campos Obligatorios Vacíos**

* **Descripción:** Se verifica cómo el sistema maneja el intento de crear un progressLog con campos obligatorios vacíos.
* **Efectividad en la Detección de Errores:** Este caso de prueba identifica de manera efectiva la validación de campos obligatorios y garantiza que el sistema responda correctamente a esta situación.

**Caso de Prueba 3: Datos Inválidos**

* **Descripción:** Se ingresa datos inválidos en uno o más campos del formulario de creación y se evalúa la respuesta del sistema.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo para identificar posibles fallos en la validación de datos y la gestión de errores al intentar crear un contrato con información incorrecta.

**Caso de Prueba 4: Límites de Entrada**

* **Descripción:** Se evalúa cómo el sistema maneja los límites de entrada para campos como la longitud máxima del comentario del contrato o el rango de completeness permitido.
* **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para garantizar que el sistema pueda manejar correctamente los límites de entrada y evitar posibles desbordamientos o errores de formato.

**Caso de Prueba 5: Verificar autorización para crear registro de progreso**

* **Descripción:** Se verifica si el sistema valida correctamente la autorización para crear un registro de progreso. El registro de progreso solo debería poder ser creado si el contrato asociado está en modo borrador y el cliente que realiza la solicitud es el propietario del contrato.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que garantiza que solo se puedan crear registros de progreso para contratos no en modo borrador y que pertenezcan al cliente que realiza la solicitud, evitando así la creación de registros no autorizados.

**Caso de Prueba 6: Validar unicidad del identificador de registro**

* **Descripción:** Se prueba si el sistema valida correctamente la unicidad del identificador del registro de progreso, evitando la creación de registros con identificadores duplicados.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que asegura que no se puedan crear registros de progreso con identificadores que ya existen en el sistema, previniendo duplicaciones indeseadas.

## 5.8. Update ProgressLog

**Caso de Prueba 1: Actualizar registro de progreso existente**

* **Descripción:** Se prueba la capacidad del sistema para actualizar un registro de progreso existente.
* **Efectividad en la Detección de Errores:** El caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede actualizar correctamente un registro de progreso con los datos proporcionados por el usuario.

**Caso de Prueba 2: Campos Obligatorios Vacíos**

* **Descripción:** Se verifica cómo el sistema maneja el intento de crear un progressLog con campos obligatorios vacíos.
* **Efectividad en la Detección de Errores:** Este caso de prueba identifica de manera efectiva la validación de campos obligatorios y garantiza que el sistema responda correctamente a esta situación.

**Caso de Prueba 3: Datos Inválidos**

* **Descripción:** Se ingresa datos inválidos en uno o más campos del formulario de creación y se evalúa la respuesta del sistema.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo para identificar posibles fallos en la validación de datos y la gestión de errores al intentar crear un contrato con información incorrecta.

**Caso de Prueba 4: Límites de Entrada**

* **Descripción:** Se evalúa cómo el sistema maneja los límites de entrada para campos como la longitud máxima del comentario del contrato o el rango de completeness permitido.
* **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para garantizar que el sistema pueda manejar correctamente los límites de entrada y evitar posibles desbordamientos o errores de formato.

**Caso de Prueba 5: Verificar autorización para actualizar registro de progreso**

* **Descripción:** Se verifica si el sistema valida correctamente la autorización para actualizar un registro de progreso. La actualización solo debería estar permitida si el contrato asociado está en modo borrador y el cliente que realiza la solicitud es el propietario del contrato.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que garantiza que solo se puedan actualizar registros de progreso asociados a contratos en modo borrador y que pertenezcan al cliente que realiza la solicitud, evitando así actualizaciones no autorizadas.

**Caso de Prueba 6: Verificar consistencia de datos actualizados**

* **Descripción:** Se prueba si el sistema mantiene la consistencia de los datos al actualizar un registro de progreso. Los datos actualizados deben reflejarse correctamente en el registro sin introducir inconsistencias.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que asegura que los datos actualizados se reflejen correctamente en el registro de progreso sin introducir inconsistencias en el sistema.

## 5.9 Delete ProgressLog

**Caso de Prueba 1: Eliminar registro de progreso existente**

* **Descripción:** Se prueba la capacidad del sistema para eliminar un registro de progreso existente.
* **Efectividad en la Detección de Errores:** El caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede eliminar correctamente un registro de progreso existente sin generar errores o inconsistencias en la base de datos.

**Caso de Prueba 2: Verificar autorización para eliminar registro de progreso**

* **Descripción:** Se verifica si el sistema valida correctamente la autorización para eliminar un registro de progreso. La eliminación solo debería estar permitida si el contrato asociado está en modo borrador y el cliente que realiza la solicitud es el propietario del contrato.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que garantiza que solo se puedan eliminar registros de progreso asociados a contratos en modo borrador y que pertenezcan al cliente que realiza la solicitud, evitando así eliminaciones no autorizadas.

**Caso de Prueba 3: Verificar consistencia de datos eliminados**

* **Descripción:** Se prueba si el sistema mantiene la consistencia de los datos al eliminar un registro de progreso. Los datos eliminados no deben dejar residuos ni introducir inconsistencias en el sistema.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que asegura que al eliminar un registro de progreso, se eliminen correctamente todos los datos asociados sin dejar residuos ni introducir inconsistencias en el sistema.

## 5.10 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.
* **Efectividad en la Detección de Errores:** El caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede recuperar correctamente los registros de progreso asociados a un contrato específico y mostrarlos en la interfaz de usuario.

**Caso de Prueba 2: Verificar autorización para listar registros de progreso**

* **Descripción:** Se verifica si el sistema valida correctamente la autorización para listar los registros de progreso. Solo se deben mostrar los registros asociados a contratos cuyo cliente es el usuario que realiza la solicitud.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que garantiza que solo se muestren los registros de progreso asociados a contratos cuyo cliente es el usuario que realiza la solicitud, evitando así la exposición de información confidencial de otros clientes.

**Caso de Prueba 3: Verificar formato de los datos mostrados**

* **Descripción:** Se prueba si el sistema muestra correctamente los datos de los registros de progreso en el formato esperado.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que verifica que los datos mostrados para cada registro de progreso estén en el formato adecuado y sean fácilmente comprensibles para el usuario final.

## 5.11 Show ProgressLog

**Caso de Prueba 1: Mostrar detalles del registro de progreso**

* **Descripción:** Se prueba la capacidad del sistema para mostrar los detalles de un registro de progreso específico.
* **Efectividad en la Detección de Errores:** El caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede recuperar correctamente los detalles del registro de progreso solicitado y mostrarlos en la interfaz de usuario.

**Caso de Prueba 2: Verificar autorización para mostrar detalles del registro de progreso**

* **Descripción:** Se verifica si el sistema valida correctamente la autorización para mostrar los detalles de un registro de progreso. Solo se deben mostrar los detalles de registros de progreso asociados a contratos cuyo cliente es el usuario que realiza la solicitud.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que garantiza que solo se muestren los detalles de registros de progreso asociados a contratos cuyo cliente es el usuario que realiza la solicitud, evitando así la exposición de información confidencial de otros clientes.

**Caso de Prueba 3: Verificar formato de los datos mostrados**

* **Descripción:** Se prueba si el sistema muestra correctamente los detalles del registro de progreso en el formato esperado.
* **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que verifica que los detalles del registro de progreso se muestren en el formato adecuado y sean fácilmente comprensibles para el usuario final.

# **6.Pruebas de Rendimiento**

## 6.1. Gráficos e Intervalos de Confianza

**Ordenador 1**

**Primera ejecución**

Tabla, Excel

Descripción generada automáticamente

Gráfico, Gráfico de barras

Descripción generada automáticamente

**Segunda Ejecución:**

Tabla, Excel

Descripción generada automáticamente

Gráfico, Gráfico de barras

Descripción generada automáticamente con confianza media

**Ordenador 2**

**Primera Ejecución:**

Gráfico

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente**Segunda Ejecución:**

## 

## 6.2. Contraste de Hipótesis

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteOrdenador 1:**

Tabla

Descripción generada automáticamente

**Conclusiones:**

Las métricas de rendimiento de ambas iteraciones muestran una consistencia notable. Tanto la media como la mediana se sitúan alrededor de 12.43 ms y 10.04 ms respectivamente, indicando un rendimiento estable en ambas ejecuciones. Además, el error típico y la desviación estándar, aproximadamente de 0.37 ms en ambas iteraciones, sugieren una dispersión similar de los datos alrededor de la media. La varianza de la muestra, muestra una pequeña variación entre las iteraciones, con valores de alrededor de 130.83 y 128.66 respectivamente, lo que señala una consistencia en la variabilidad de los datos. Los coeficientes de asimetría y curtosis, alrededor de 3.36 y 16.98 para el coeficiente de asimetría, y aproximadamente 3.97 y 28.16 para la curtosis, indican una distribución sesgada y con colas pesadas en los datos, persistente en ambas iteraciones. Además, el rango, el mínimo y el máximo, muestran una coherencia entre ambas ejecuciones, con valores que oscilan entre 1.35 y 114.13 ms. Finalmente, la prueba Z para medias de dos muestras revela un valor de z muy cercano a 0, lo que sugiere que no existe una diferencia significativa entre las medias de las dos muestras.

Tabla

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente**Ordenador 2**

**Conclusiones:**

Las métricas de rendimiento para ambas iteraciones presentan una similitud significativa. La media es aproximadamente 11.21 ms en la primera iteración y 11.05 ms en la segunda, mientras que la mediana es alrededor de 9.17 ms y 8.91 ms respectivamente, lo que indica un rendimiento consistente. El error típico es ligeramente mayor en la primera iteración (0.33 ms) en comparación con la segunda (0.31 ms). La desviación estándar y la varianza también son similares, con la primera iteración mostrando valores de 10.29 y 105.97, y la segunda iteración valores de 9.51 y 90.36, respectivamente. Estos valores sugieren que la dispersión de los datos alrededor de la media es comparable en ambas iteraciones.

El coeficiente de asimetría y la curtosis son altos en ambas iteraciones, con la primera iteración mostrando valores de 3.36 y 16.98, y la segunda iteración 3.08 y 15.64, respectivamente, lo que indica una distribución sesgada con colas pesadas en los datos. El rango, el mínimo y el máximo son consistentes entre ambas iteraciones, con el rango siendo de aproximadamente 101.50 ms en la primera iteración y 91.27 ms en la segunda.

La prueba Z para medias de dos muestras muestra un valor de z extremadamente pequeño, cercano a 0, lo que indica que no hay una diferencia significativa entre las medias de las dos muestras. En resumen, las métricas sugieren un rendimiento estable y consistente entre ambas iteraciones, sin diferencias significativas en la media o en la dispersión de los datos.

# **7.Conclusión**

Las pruebas funcionales realizadas en la rama feature/S2-Man09 han demostrado que las funcionalidades clave, como la creación, actualización, publicación, eliminación y visualización de contratos y registros de progreso, operan correctamente. Los casos de prueba abordan varios escenarios, incluyendo datos inválidos, campos obligatorios vacíos, límites de entrada y posibles intentos de hacking. La efectividad en la detección de errores ha sido alta, asegurando la robustez del sistema ante diferentes tipos de entradas y validaciones. Dado que algunas modificaciones recientes en el archivo CSV pueden afectar la ejecución de estos tests, se recomienda realizar una revisión y ejecución en la rama feature/S2-Man09 para garantizar su precisión y confiabilidad.

# **8.Bibliografía**

En blanco intencionalmente.