

# Reporte de Testing



**Grupo:** C3.039

**Repositorio:** <https://github.com/pabalcber/C3.039-Acme-SF>

**Integrantes:**

Nombre	Apellidos	Correo Corporativo
Sheng	Chen	sheche1@alum.us.es
Jun	Yao	junyao@alum.us.es

Tabla de versiones:

Fecha	Versión	Descripción de los cambios	Sprint
25/05/2024	1.0	Creación del documento y redacción de sus contenidos	4

# 1.Índice

<b>1.Índice</b>	3
<b>2. Resumen Ejecutivo</b>	4
<b>3.Tabla de revisiones</b>	5
<b>4. Introducción</b>	6
<b>5.Pruebas Funcionales</b>	7
5.1. Create TraningModule	7
5.2. Update TraningModule	8
5.3 Publish TraningModule	8
5.4 Delete TrainingModule	9
5.5 List TrainingModule	10
5.6 Show trainingModule	10
5.7. Create trainingSession	11
5.8. Update trainingSession	12
5.9 Delete TrainingSession	13
5.10 List ProgressLog	14
5.11 Show trainingSession	15
<b>6.Pruebas de Rendimiento</b>	17
6.1. Gráficos e Intervalos de Confianza	17
6.2. Contraste de Hipótesis	18
En conclusión, tras realizar el proceso de testing durante este Sprint, se ha podido comprobar que lo desarrollado en los sprints anteriores cumple con los requisitos en términos generales. Sin embargo, el análisis estadístico muestra un p-valor de 0, lo cual indica diferencias significativas en algunos aspectos del rendimiento del sistema. Esto significa que algunos cambios introducidos sí han tenido un impacto relevante. Este resultado nos ha permitido identificar áreas específicas de mejora y realizar ajustes críticos necesarios para optimizar el rendimiento del sistema.	20
<b>7.Conclusión</b>	20
<b>8.Bibliografía</b>	21

## **2. Resumen Ejecutivo**

Este informe de pruebas resume los resultados obtenidos en el proyecto del estudiante 3. En la sección de pruebas funcionales, se describen los casos de prueba realizados, organizados por funcionalidad, con una evaluación de su eficacia en la detección de errores. Por otro lado, la sección 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 empleada fue rigurosa, respaldada por herramientas como Eclipse y una planificación meticulosa. Este informe tiene como objetivo ofrecer una visión clara de la calidad y el rendimiento de nuestro software.

### **3.Tabla de revisiones**

Número de revisión	Fecha	Descripción
1	16/10/2024	Revisión final antes de la entrega

## **4. Introducción**

En el ámbito del desarrollo de software, las pruebas son esenciales para asegurar la calidad y el rendimiento del producto final. Este informe presenta los resultados de las pruebas funcionales y de rendimiento realizadas en el contexto del estudiante 3 de nuestro proyecto. En la sección de pruebas funcionales, se describen los casos de prueba implementados y su eficacia en la detección de errores en diversas funcionalidades del software. Por otro lado, en la sección de pruebas de rendimiento, se examina el tiempo de ejecución en diferentes entornos, proporcionando información sobre la eficiencia y la capacidad del sistema. Este informe tiene como objetivo ofrecer una evaluación objetiva y completa de la calidad del software desarrollado.

## 5.Pruebas Funcionales

El testing del estudiante 3 se ha llevado a cabo en la rama **master** 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 **master**. Los tests implementados son los siguientes:

### 5.1. Create TrainingModule

#### Caso de Prueba 1: Creación Exitosa

- **Descripción:** Este caso de prueba examina el escenario donde el usuario completa todos los campos obligatorios correctamente y procede a crear un TrainingModule..
- **Efectividad en la Detección de Errores:** Este caso de prueba demuestra su efectividad al detectar posibles fallas en la validación de campos obligatorios y en el proceso de creación del TrainingModule.

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

- **Descripción:** Se evalúa cómo el sistema maneja el intento de crear un TrainingModule con campos obligatorios sin completar.
- **Efectividad en la Detección de Errores:** Este caso de prueba es eficaz para identificar la validación de campos obligatorios y asegurar una respuesta adecuada del sistema ante esta situación

#### Caso de Prueba 3: Datos Inválidos

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

#### Caso de Prueba 4: Límites de Entrada

- **Descripción:** Se evalúa cómo el sistema gestiona los límites de entrada.
- **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 verifica cómo el sistema responde ante posibles intentos de hacking, tanto de tipo Get Hacking como Post Hacking.
- **Efectividad en la Detección de Errores:** Este caso de prueba es esencial para asegurar que el sistema no sea vulnerable a usuarios con intenciones maliciosas y demuestra su eficacia en este aspecto.

## 5.2. Update TrainingModule

### Caso de Prueba 1: Actualización Exitosa

- **Descripción:** Se prueba el escenario en el que el usuario selecciona un TrainingModule 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 TrainingModules 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 TrainingModule 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 TrainingModules, garantizando la integridad de la información.

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

- **Descripción:** Se intenta actualizar un TrainingModule 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 TrainingModules con campos obligatorios vacíos y valide adecuadamente la información ingresada.

### Caso de Prueba 4: Hacking

- **Descripción:** Se prueba si el sistema permite actualizar TrainingModules que no corresponden al usuario actual.
- **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 y demuestra su eficacia en este aspecto.

## 5.3 Publish TrainingModule

### Caso de Prueba 1: Publicación Exitosa

- **Descripción:** Se prueba el escenario en el que el usuario selecciona un TrainingModule existente y lo publica con éxito.
- **Efectividad en la Detección de Errores:** Este caso de prueba verifica de manera efectiva que el TrainingModule 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: Actualización de Campos Obligatorios Vacíos**

- **Descripción:** Se intenta publicar un TrainingModule 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 TrainingModules con campos obligatorios vacíos y valide adecuadamente la información ingresada.

#### **Caso de Prueba 4: Hacking**

- **Descripción:** Se prueba si el sistema permite publicar TrainingModules de otros usuarios.
- **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para garantizar que el sistema no sea vulnerable a usuarios con intenciones maliciosas y demuestra su eficacia en este aspecto.

### **5.4 Delete TrainingModule**

#### **Caso de Prueba 1: Eliminar TrainingModule Existente**

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

#### **Caso de Prueba 2: Verificar autorización para eliminar TrainingModule**

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

### **Caso de Prueba 3: Hacking**

- **Descripción:** Se prueba si el sistema permite eliminar TrainingModules ya publicados.
- **Efectividad en la Detección de Errores:** Este caso de prueba es crucial para garantizar que el sistema no sea vulnerable a usuarios con intenciones maliciosas y demuestra su eficacia en este aspecto.

## **5.5 List TrainingModule**

### **Caso de Prueba 1: Listar trainingModule de un developer**

- **Descripción:** Se prueba la capacidad del sistema para listar los TrainingModules asociados a un usuario específico.
- **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede recuperar correctamente los TrainingModules asociados a un usuario y mostrarlos en la interfaz de usuario.

### **Caso de Prueba 2: Verificar autorización para listar trainingModule**

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

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

- **Descripción:** Se prueba si el sistema muestra correctamente los datos de los TrainingModules 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 TrainingModule estén en el formato adecuado y sean fácilmente comprensibles para el usuario final.

## **5.6 Show trainingModule**

### **Caso de Prueba 1: Mostrar detalles del trainingModule**

- **Descripción:** Se prueba la capacidad del sistema para mostrar los detalles de un TrainingModule específico.

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

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

- **Descripción:** Se verifica si el sistema valida correctamente la autorización para mostrar los detalles de un TrainingModule. Solo se deben mostrar los detalles de TrainingModules que pertenezcan al 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 los TrainingModules que pertenezcan al usuario que realiza la solicitud, evitando así la exposición de información confidencial de otros usuarios.

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

- **Descripción:** Se prueba si el sistema muestra correctamente los detalles del TrainingModule 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 TrainingModule se muestren en el formato adecuado y sean fácilmente comprensibles para el usuario final.

### **5.7. Create trainingSession**

#### **Caso de Prueba 1: Crear trainingSession válido**

- **Descripción:** Se prueba la capacidad del sistema para crear un trainingSession válido para una trainingModule específica.
- **Efectividad en la Detección de Errores:** Este 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 una trainingSession 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 ingresan 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 una trainingSession 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 location de la trainingSession 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 trainingSession**

- **Descripción:** Se verifica si el sistema valida correctamente la autorización para crear una trainingSession. La trainingSession solo debería poder ser creada si el TrainingModule asociado está en modo borrador y el usuario que realiza la solicitud es el propietario del TrainingModule.
- **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 trainingSessions para TrainingModules en modo borrador y que pertenezcan al usuario que realiza la solicitud, evitando así la creación de sesiones no autorizadas.

#### **Caso de Prueba 6: Validar unicidad del identificador de trainingSession**

- **Descripción:** Se prueba si el sistema valida correctamente la unicidad del identificador de la trainingSession, evitando la creación de sesiones 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 trainingSessions con identificadores que ya existen en el sistema, previniendo duplicaciones indeseadas.

## **5.8. Update trainingSession**

#### **Caso de Prueba 1: Actualizar registro de trainingSession existente**

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

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

- **Descripción:** Se verifica cómo el sistema maneja el intento de actualizar una trainingSession 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 ingresan datos inválidos en uno o más campos del formulario de actualización 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 actualizar una trainingSession 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 location de la trainingSession 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 trainingSession**

- **Descripción:** Se verifica si el sistema valida correctamente la autorización para actualizar un trainingSession. La actualización solo debería estar permitida si el TrainingModule asociado está en modo borrador y el usuario que realiza la solicitud es el propietario del TrainingModule.
- **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 trainingSession asociados a TrainingModules en modo borrador y que pertenezcan al usuario 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 TrainingSession**

#### **Caso de Prueba 1: Eliminar TrainingSession existente**

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

#### **Caso de Prueba 2: Verificar Autorización para Eliminar TrainingSession**

- **Descripción:** Se verifica si el sistema valida correctamente la autorización para eliminar una TrainingSession. La eliminación solo debería estar permitida si el TrainingModule asociado está en modo borrador y el usuario que realiza la solicitud es el propietario del TrainingModule.
- **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 TrainingSessions asociadas a TrainingModules en modo borrador y que pertenezcan al usuario 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 una TrainingSession. 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 una TrainingSession, 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 TrainingSessions Asociadas a un TrainingModule**

- **Descripción:** Se prueba la capacidad del sistema para listar las trainingSessions asociadas a un TrainingModule específico.
- **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo para identificar errores, ya que verifica si el sistema puede recuperar correctamente las trainingSessions vinculadas a un TrainingModule específico y mostrarlas en la interfaz de usuario.

#### **Caso de Prueba 2: Verificar Autorización para Listar TrainingSessions**

- **Descripción:** Se verifica si el sistema valida correctamente la autorización para listar las trainingSessions. Solo se deben mostrar las sesiones asociadas a TrainingModules cuyo propietario 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 las trainingSessions asociadas a TrainingModules cuyo propietario es el usuario que realiza la solicitud, evitando así la exposición de información confidencial de otros usuarios.

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

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

## **5.11 Show trainingSession**

### **Caso de Prueba 1 Mostrar Detalles de la TrainingSession**

- **Descripción:** Se prueba la capacidad del sistema para mostrar los detalles de una TrainingSession específica.
- **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede recuperar correctamente los detalles de la TrainingSession solicitada y mostrarlos en la interfaz de usuario.

### **Caso de Prueba 2: Verificar Autorización para Mostrar Detalles de la TrainingSession**

- **Descripción:** Se verifica si el sistema valida correctamente la autorización para mostrar los detalles de una TrainingSession. Solo se deben mostrar los detalles de las TrainingSessions asociadas a TrainingModules cuyo propietario 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 las TrainingSessions asociadas a TrainingModules cuyo propietario es el usuario que realiza la solicitud, evitando así la exposición de información confidencial de otros usuarios.

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

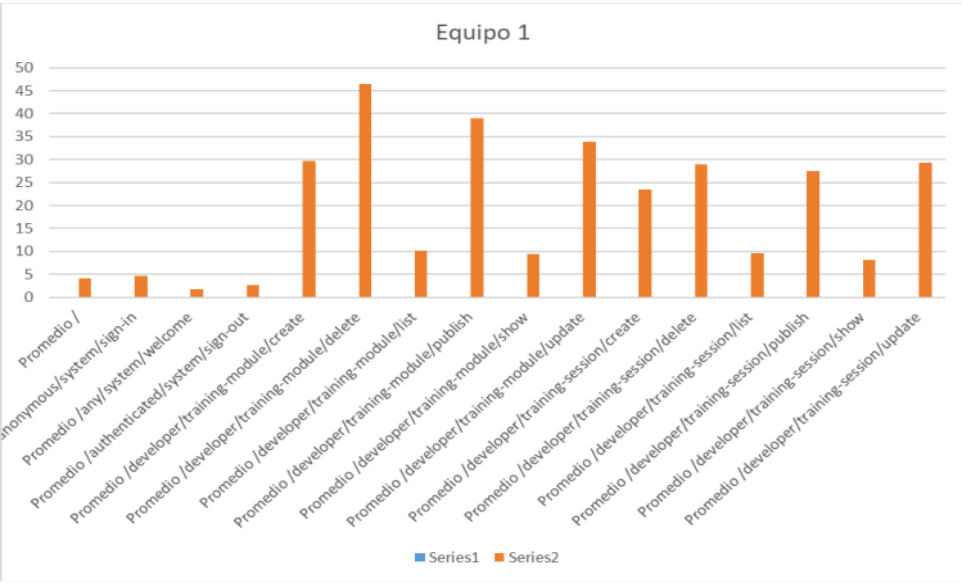
- **Descripción:** Se prueba si el sistema muestra correctamente los detalles de la TrainingSession en el formato esperado.
- **Efectividad en la Detección de Errores:** Este caso de prueba es efectivo en la detección de errores, ya que verifica que los detalles de la TrainingSession se muestren en el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



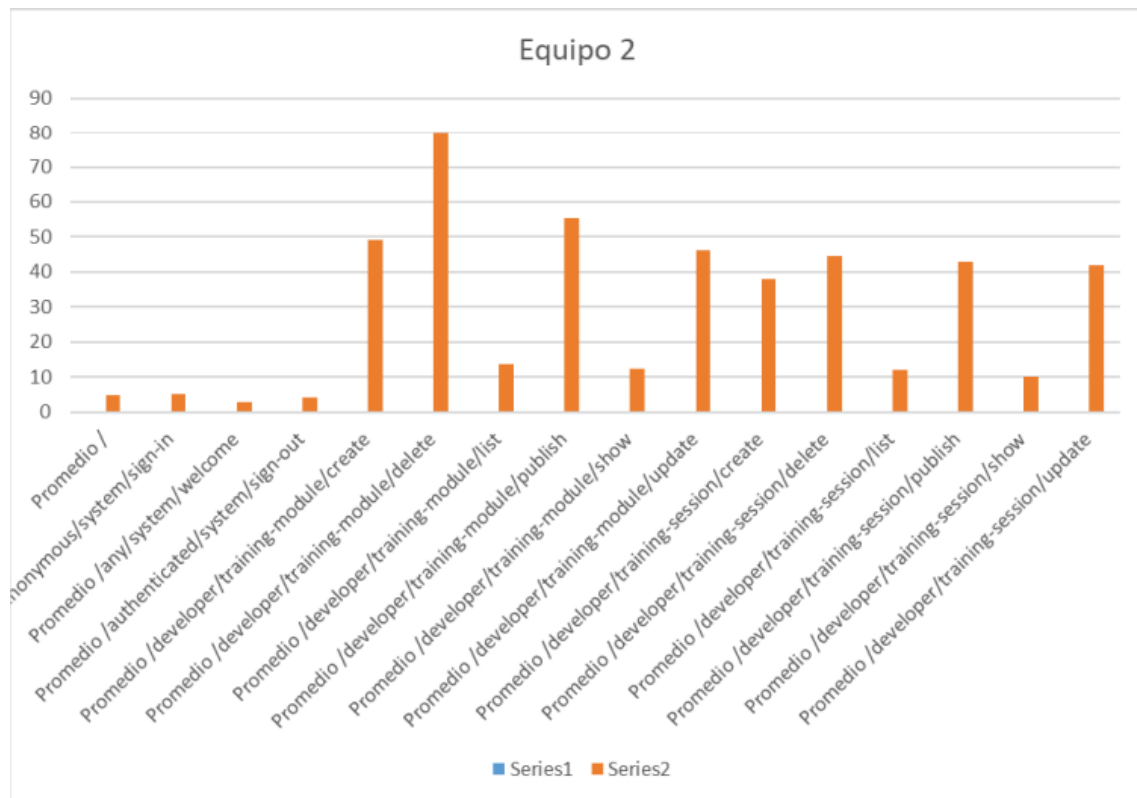
# 6.Pruebas de Rendimiento

## 6.1. Gráficos e Intervalos de Confianza

### Ordenador 1



### Ordenador 2



## 6.2. Contraste de Hipótesis

**Ordenador 1:**

<i>Equipo 1</i>		
Media	15.34808952	
Error típico	0.402003183	
Mediana	9.29275	
Moda	28.9814	
Desviación estándar	13.93739877	
Varianza de la muestra	194.2510846	
Curtosis	4.575910459	
Coefficiente de asimetría	1.682045491	
Rango	130.1944	
Mínimo	0.7408	
Máximo	130.9352	
Suma	18448.4036	
Cuenta	1202	
Nivel de confianza(95.0%)	0.788706604	
Interval (ms)	14.55938291	16.13679612
Interval (s)	0.014559383	0.016136796

## Ordenador 2:

<i>Equipo 2</i>		
Media	22.69702146	
Error típico	0.625080309	
Mediana	12.2737	
Moda	11.3709	
Desviación estándar	22.41590629	
Varianza de la muestra	502.4728549	
Curtosis	2.038091479	
Coefficiente de asimetría	1.503019097	
Rango	168.2307	
Mínimo	1.0641	
Máximo	169.2948	
Suma	29188.3696	
Cuenta	1286	
Nivel de confianza(95.0%)	1.226289939	
Interval (ms)	21.47073152	23.9233114
Interval (s)	0.021470732	0.02392331

	<i>Equipo 1</i>	<i>Equipo 2</i>
Media	15.34808952	22.6970215
Varianza (conocida)	194.2510846	502.472855
Observaciones	1202	1286
Diferencia hipotética de las medias	0	
z	-9.888357098	
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	

### Conclusiones:

En conclusión, tras realizar el proceso de testing durante este Sprint, se ha podido comprobar que lo desarrollado en los sprints anteriores cumple con los requisitos en términos generales. Sin embargo, el análisis estadístico muestra un p-valor de 0, lo cual indica diferencias significativas en algunos aspectos del rendimiento del sistema. Esto significa que algunos cambios introducidos sí han tenido un impacto relevante. Este resultado nos ha permitido identificar áreas específicas de mejora y realizar ajustes críticos necesarios para optimizar el rendimiento del sistema.

## 7.Conclusión

En conclusión, tras realizar todo el proceso de testing durante este Sprint se ha podido comprobar que lo desarrollado en los anteriores cumple en términos generales con lo requerido, rindiendo bien y respondiendo correctamente ante casos tanto válidos como inválidos. Ha servido además para solucionar algunos errores críticos que no habían sido detectados durante el testing informal

## **8.Bibliografía**

En blanco intencionalmente.