

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

TestingReport



Acme ANS

OUR FIRST PROJECT IN D&T

Grado en Ingeniería Informática – Ingeniería del Software


Diseño y Pruebas 2

Curso 2024 – 2025

Grupo de prácticas: C2-055


Autores por orden alfabético

Alvaro Carrera Bernal

	Diseño y Pruebas II Acme-Software-Factory
	Testing Report


Índice de contenido

1. Resumen ejecutivo	3
2. Tabla de revisiones	4
3. Introducción.....	5
4. Contenido	6
Pruebas funcionales	6
Pruebas de rendimiento.....	13
Análisis de los resultados obtenidos	17
5. Conclusiones	19
6. Bibliografía	20

	Diseño y Pruebas II Acme-Software-Factory
	Testing Report


1. Resumen ejecutivo

En este informe se presenta un análisis detallado de las pruebas y los resultados obtenidos para las funcionalidades correspondientes al módulo Student 3.

	Diseño y Pruebas II Acme-Software-Factory
	Testing Report

2. Tabla de revisiones

Fecha	Versión	Descripción
22/05/2025	1.0	Primera versión del documento
25/05/2025	1.1	Revisión del análisis de rendimiento
03/07/2025	1.2	Corrección del documento


	Diseño y Pruebas II Acme-Software-Factory
	Testing Report

3. Introducción

El informe se estructura en dos secciones principales: pruebas funcionales y pruebas de rendimiento. Su propósito es ofrecer un análisis exhaustivo de los casos de prueba y del desempeño asociado a estos.

La primera sección se enfoca en las pruebas funcionales, destinadas a evaluar el comportamiento de dos entidades específicas: assignment y activity log. Se detallan los casos de prueba ejecutados, organizados por entidad, con descripciones claras y concisas para cada uno, lo que facilita una evaluación precisa y eficiente de las funcionalidades relacionadas.

La segunda sección presenta gráficos detallados que ilustran los tiempos de respuesta del sistema durante las pruebas funcionales descritas. Se compara el rendimiento del proyecto en dos entornos diferentes, proporcionando un análisis profundo de su desempeño. Esta comparación permite identificar diferencias en la eficiencia y la capacidad de gestión de carga entre los sistemas evaluados.

	Diseño y Pruebas II Acme-Software-Factory
	Testing Report

4. Contenido

Pruebas funcionales

Requirement 8

Para este requisito se han implementado las siguientes pruebas:

```

v > assignment
  > create.safe
  > create-wrongRealm.hack
  > delete.safe
  > delete-rightRealm-rightUser-wrongAction.hack
  > delete-rightRealm-wrongUser.hack
  > delete-wrongRealm.hack
  > list.hack
  navigation-attributes.hack
  publish.safe
  publish-rightRealm-rightUser-wrongAction.hack
  publish-rightRealm-wrongUser.hack
  publish-wrongRealm.hack
  > show.safe
  > show-rightRealm-wrongUser.hack
  > show-wrongRealm.hack
  > update.safe
  > update-rightRealm-rightUser-wrongAction.hack
  > update-rightRealm-wrongUser.hack
  > update-wrongRealm.hack

```




Diseño y Pruebas II Acme-Software-Factory

















Testing Report

- **create.safe:** Valida que un usuario autorizado pueda crear una asignación correctamente, cumpliendo con las reglas del sistema.
- **create-wrongRealm.hack:** Asegura que usuarios anónimos no puedan crear asignaciones, protegiendo el sistema contra accesos no autorizados.
- **delete.safe:** Confirma que una asignación en estado "borrador" pueda ser eliminada por un usuario autorizado.
- **delete-rightRealm-rightUser-wrongAction.hack:** Verifica que un usuario autorizado ("member1") no pueda eliminar una asignación ya publicada.
- **delete-rightRealm-wrongUser.hack:** Como "member2", asegura que no se pueda eliminar una asignación que pertenece a "member1".
- **delete-wrongRealm.hack:** Garantiza que usuarios anónimos no puedan eliminar asignaciones, sin importar su estado.
- **list.hack:** Evalúa si el sistema previene intentos no autorizados o incorrectos de listar asignaciones.
- **list-mine.safe:** Comprueba que un usuario autorizado pueda listar sus propias asignaciones sin problemas.
- **navigation-attribute.hack:** Se comprueba que el sistema reacciona correctamente ante accesos no autorizados. En concreto, se verifica que el formulario de actualización y creación de asignaciones no permita seleccionar vuelos que no existan, estados que no existan o cargos (Duty) que no existan.
- **show.safe:** Verifica que un usuario autorizado pueda ver los detalles de una asignación específica.
- **show-rightRealm-wrongUser.hack:** Como "member2", asegura que no se pueda ver una asignación de "member1".
- **show-wrongRealm.hack:** Confirma que usuarios anónimos no puedan visualizar ninguna asignación.
- **update.safe:** Valida que un usuario autorizado pueda actualizar una asignación respetando las restricciones del sistema.
- **update-rightRealm-rightUser-wrongAction.hack:** Como "member1", verifica que no se pueda realizar una acción no permitida (como actualizar una asignación publicada).
- **update-rightRealm-wrongUser.hack:** Como "member2", asegura que no se pueda actualizar una asignación de "member1".
- **update-wrongRealm.hack:** Garantiza que usuarios anónimos no puedan actualizar asignaciones, protegiendo el sistema.

Asimismo, se ha verificado mediante el test de navigation-attributes.hack que los atributos de solo lectura (readOnly) no permiten modificaciones, impidiendo cualquier intento de manipulación.

	Diseño y Pruebas II Acme-Software-Factory
	Testing Report

Tras ejecutar el replayer obtenemos la siguiente cobertura.

✓  acme.features.crew.assignment	 67,9 %
>  CrewAssignmentUpdateService.java	 0,5 %
>  CrewAssignmentPublishService.java	 93,9 %
>  CrewAssignmentCreateService.java	 94,3 %
>  CrewAssignmentDeleteService.java	 94,7 %
>  CrewAssignmentListLegsService.java	 98,2 %
>  CrewAssignmentShowService.java	 99,5 %
>  CrewAssignmentController.java	 100,0 %


Como vemos, la mayoría de la funcionalidad está cubierta, aunque estas excepciones:

- En el método validate del servicio CrewAssignmentPublishService, no se ha ejecutado la validación que verifica si el leg ya está completado, pero se ha comprobado manualmente, impidiendo que se pueda publicar una asignación.
- Ocurre lo mismo con el servicio CrewAssingmentCreateService, donde no se puede crear una asignación con un tramo de vuelo (leg) ya completado, ni se ha comprobado que un tramo ya tiene un piloto, aunque si manualmente.
- En el caso del servicio de CrewAssignmentUpdateService, no se ha ejecutado debido a un problema con el framework que hace que se pare automáticamente la ejecución del tester replayer, sin motivo alguno. En primer momento se pensó que era problema del test creado, pero al realizar este procedimiento de nuevo durante varias veces y ver que el problema persistía, se declinó la opción de que fuera problema del test.
- Quedándonos con esto último, ocurre lo mismo con los métodos de hacking, aunque ejecutándolo por separados cada uno, llegamos a una cobertura de tests de al menos el 97%.

Errores localizados tras la realización de los test:


























- Se han corregido los validadores en los métodos create, update y publish, ya que anteriormente no estaban vinculados correctamente a los campos correspondientes del formulario.
- Se ha ajustado el comportamiento de los menús desplegables en el selector de tramos de vuelos, al miembro de la tripulación y estado actual de una asignación, con el objetivo de ofrecer una experiencia de usuario más clara e intuitiva.
- Han permitido detectar varios intentos de uso indebido del sistema, como por ejemplo:
 - Selección de tramos de vuelos con fecha pasada, inexistente o no publicados desde el selector de tramos de vuelos.
 - Envío de datos inválidos en el campo correspondiente al momento de creación de una asignación.
- Se adjunta foto del testing-tester#replayer, mencionado anteriormente y que hace que los tests no se ejecuten correctamente:

```
ERROR com.fasterxml.jackson.databind.JsonMappingException: from(JsonMappingException.java:276): N/A
at [Source: (StringReader); line: 1, column: 12] (through reference chain: acme.client.testing.Oracle["request-method"])
acme.client.testing.Oracle.setRequestMethod(Oracle.java:137): java.lang.AssertionError
The system is shutting down...
Exiting application with status 0.
```


	Diseño y Pruebas II Acme-Software-Factory
	Testing Report

Requirement 9

Para este requisito se han implementado las siguientes pruebas:

 > activityLog
 > create.safe
 > create-rightRealm-rightUser-wrongAction.hack
 > create-rightRealm-wrongUser.hack
 > create-wrongRealm.hack
 > delete.safe
 > delete-rightRealm-rightUser-wrongAction.hack
 > delete-rightRealm-wrongUser.hack
 > delete-wrongRealm.hack
 > list-mine.safe
 > list-rightRealm-wrongUser.hack
 > list-wrongRealm.hack
 navigation-attributes.hack
 > publish.safe
 > publish2.safe
 > publish-rightRealm-rightUser-wrongAction.hack
 > publish-rightRealm-wrongUser.hack
 > publish-wrongRealm.hack
 > show.safe
 > show-rightRole-wrongUser.hack
 > show-wrongRealm.hack
 > update.safe
 > update-rightRealm-rightUser-wrongAction.hack
 > update-rightRealm-wrongUser.hack
 > update-wrongRealm.hack

- **create.safe:** Valida que un usuario autorizado pueda crear un registro de actividad correctamente, respetando las reglas del sistema.
- **create-rightRealm-rightUser-wrongAction.hack:** Como usuario autorizado, verifica que no se pueda realizar una acción no permitida (como crear un registro en un estado inválido).
- **create-rightRealm-wrongUser.hack:** Como un usuario diferente (ej. "member2"), asegura que no se pueda crear un registro de actividad que no le corresponda.
- **create-wrongRealm.hack:** Confirma que usuarios anónimos no puedan crear registros de actividad, protegiendo el sistema.
- **delete.safe:** Comprueba que un usuario autorizado pueda eliminar un registro de actividad cuando sea permitido.
- **delete-rightRealm-rightUser-wrongAction.hack:** Verifica que un usuario autorizado no pueda eliminar un registro de actividad en un contexto no permitido.




Diseño y Pruebas II Acme-Software-Factory












Testing Report

- **delete-rightRealm-wrongUser.hack:** Como "member2", asegura que no se pueda eliminar un registro que pertenece a "member1".
- **delete-wrongRealm.hack:** Garantiza que usuarios anónimos no puedan eliminar registros de actividad.
- **list-mine.safe:** Confirma que un usuario autorizado pueda listar sus propios registros de actividad.
- **list-rightRealm-wrongUser.hack:** Como "member2", valida que no se pueda listar registros de actividad de "member1".
- **list-wrongRealm.hack:** Asegura que usuarios anónimos no puedan listar registros de actividad.
- **publish.safe:** Verifica que un usuario autorizado pueda publicar un registro de actividad cuando sea aplicable.
- **publish2.safe:** Evalúa un segundo escenario de publicación de un registro de actividad, asegurando su correcto funcionamiento.
- **publish-rightRealm-rightUser-wrongAction.hack:** Como usuario autorizado, comprueba que no se pueda realizar una publicación no permitida.
- **publish-rightRealm-wrongUser.hack:** Como "member2", asegura que no se pueda publicar un registro de actividad de "member1".
- **publish-wrongRealm.hack:** Confirma que usuarios anónimos no puedan publicar registros de actividad.
- **show.safe:** Valida que un usuario autorizado pueda visualizar los detalles de un registro de actividad.
- **show-rightRealm-wrongUser.hack:** Como "member2", asegura que no se pueda ver un registro de actividad de "member1".
- **show-wrongRealm.hack:** Garantiza que usuarios anónimos no puedan visualizar registros de actividad.
- **update.safe:** Comprueba que un usuario autorizado pueda actualizar un registro de actividad con datos válidos.
- **update-rightRealm-rightUser-wrongAction.hack:** Como usuario autorizado, verifica que no se pueda realizar una actualización no permitida.
- **update-rightRealm-wrongUser.hack:** Como "member2", asegura que no se pueda actualizar un registro de actividad de "member1".
- **update-wrongRealm.hack:** Confirma que usuarios anónimos no puedan actualizar registros de actividad.

En este caso no ha sido necesario realizar test relacionados a navigation-attributes puesto que el formulario de creación, actualización y publicación de actividades no cuenta con este elemento.

	Diseño y Pruebas II Acme-Software-Factory
	Testing Report

Tras ejecutar el replayer obtenemos la siguiente cobertura.

✓  acme.features.crew.activityLog	 76,1 %
>  CrewActivityLogUpdateService.java	 1,1 %
>  CrewActivityLogPublishService.java	 83,6 %
>  CrewActivityLogCreateService.java	 97,3 %
>  CrewActivityLogListService.java	 97,1 %
>  CrewActivityLogDeleteService.java	 97,1 %
>  CrewActivityLogShowService.java	 98,1 %
>  CrewActivityLogController.java	 100,0 %

Como vemos, la mayoría de la funcionalidad está cubierta, a excepción del CrewActivityLogDeleteService, aunque encontramos dos excepciones:

- En el servicio CrewActivityLogPublishService, encargado de gestionar la publicación de registros de actividad (ActivityLog), no se ha probado el método perform debido a la falta de un leg con una fecha disponible que permita realizar la publicación.
- En el caso del servicio de CrewActivityLogUpdateService, no se ha ejecutado debido a un problema con el framework que hace que se pare automáticamente la ejecución del tester replayer, sin motivo alguno. En primer momento se pensó que era problema del test creado, pero al realizar este procedimiento de nuevo durante varias veces y ver que el problema persistía, se declinó la opción de que fuera problema del test.
- Quedándonos con esto último, ocurre lo mismo con los métodos de hacking, aunque ejecutándolo por separados cada uno, llegamos a una cobertura de tests de al menos el 97%.

Errores localizados tras realizar los test:

- Se han corregido los validadores en los métodos create, update y publish, ya que anteriormente no estaban vinculados correctamente a los campos correspondientes del formulario.
- Se adjunta foto del testing-tester#replayer, mencionado anteriormente y que hace que los tests no se ejecuten correctamente:

```
ERROR com.fasterxml.jackson.databind.JsonMappingException.from(JsonMappingException.java:276): N/A
at [Source: (StringReader); line: 1, column: 12] (through reference chain: acme.client.testing.Oracle["request-method"])
acme.client.testing.Oracle.setRequestMethod(Oracle.java:137): java.lang.AssertionError
The system is shutting down...
Exiting application with status 0.
```



Pruebas de rendimiento

Las pruebas de rendimiento se realizaron con la herramienta "tester-replayer", que reproduce los casos de prueba grabados para medir el tiempo que se tarda en procesar las peticiones. Las pruebas de rendimiento se ejecutaron dos veces:

1. En mi PC personal.
2. En el PC de mi compañero.

Métricas para el caso 1

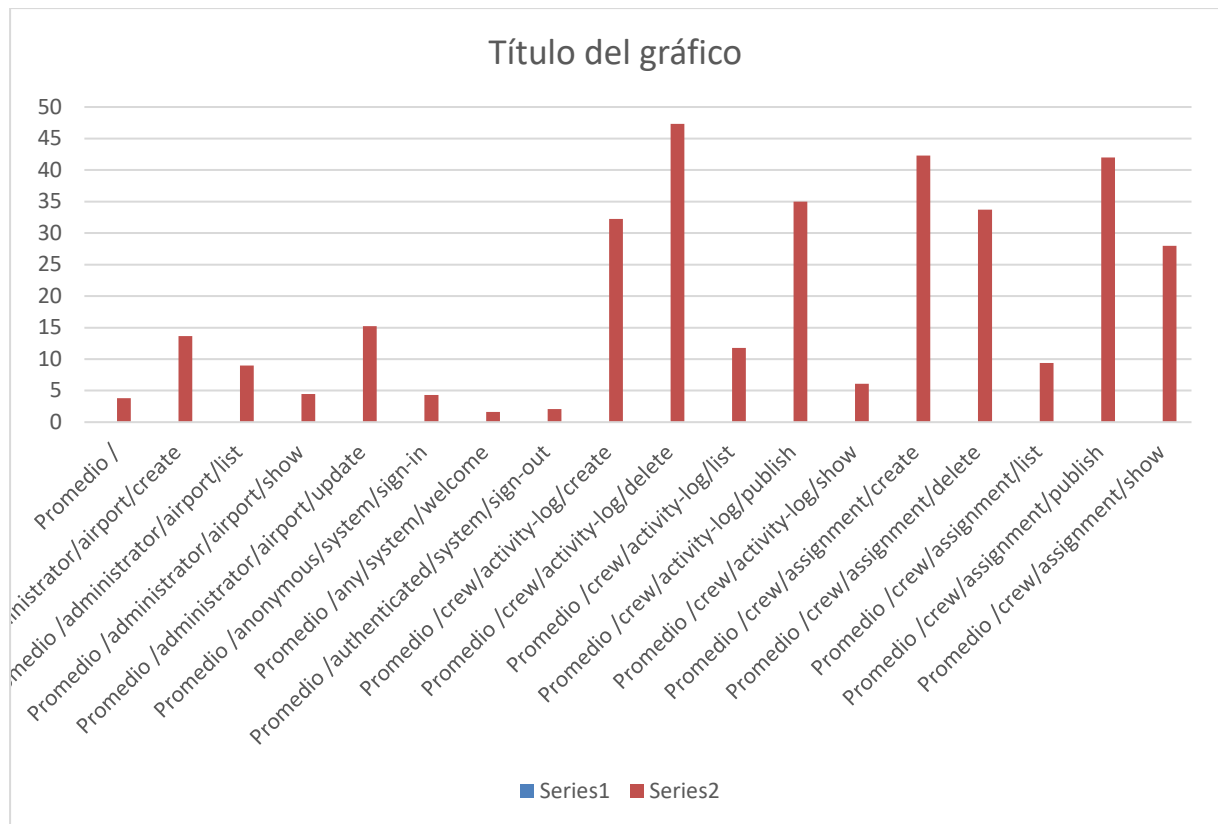
Estos han sido los resultados obtenidos:

Promedio /	3,78188571
Promedio /administrator/airport/create	13,6625777
Promedio /administrator/airport/list	8,96115
Promedio /administrator/airport/show	4,47176667
Promedio /administrator/airport/update	15,2140573
Promedio /anonymous/system/sign-in	4,33113636
Promedio /any/system/welcome	1,64208182
Promedio /authenticated/system/sign-out	2,08811111
Promedio /crew/activity-log/create	32,238195
Promedio /crew/activity-log/delete	47,3465
Promedio /crew/activity-log/list	11,8013818
Promedio /crew/activity-log/publish	34,9735719
Promedio /crew/activity-log/show	6,113675
Promedio /crew/assignment/create	42,2869846
Promedio /crew/assignment/delete	33,6978
Promedio /crew/assignment/list	9,41304444
Promedio /crew/assignment/publish	42,0058583
Promedio /crew/assignment/show	27,96885
Promedio general	16,6458807



Diseño y Pruebas II Acme-Software-Factory

Testing Report



También se ha calculado el intervalo de confianza del 95 % para el tiempo que tarda en atender las solicitudes anteriores mi ordenador personal. Obteniendo como resultado:

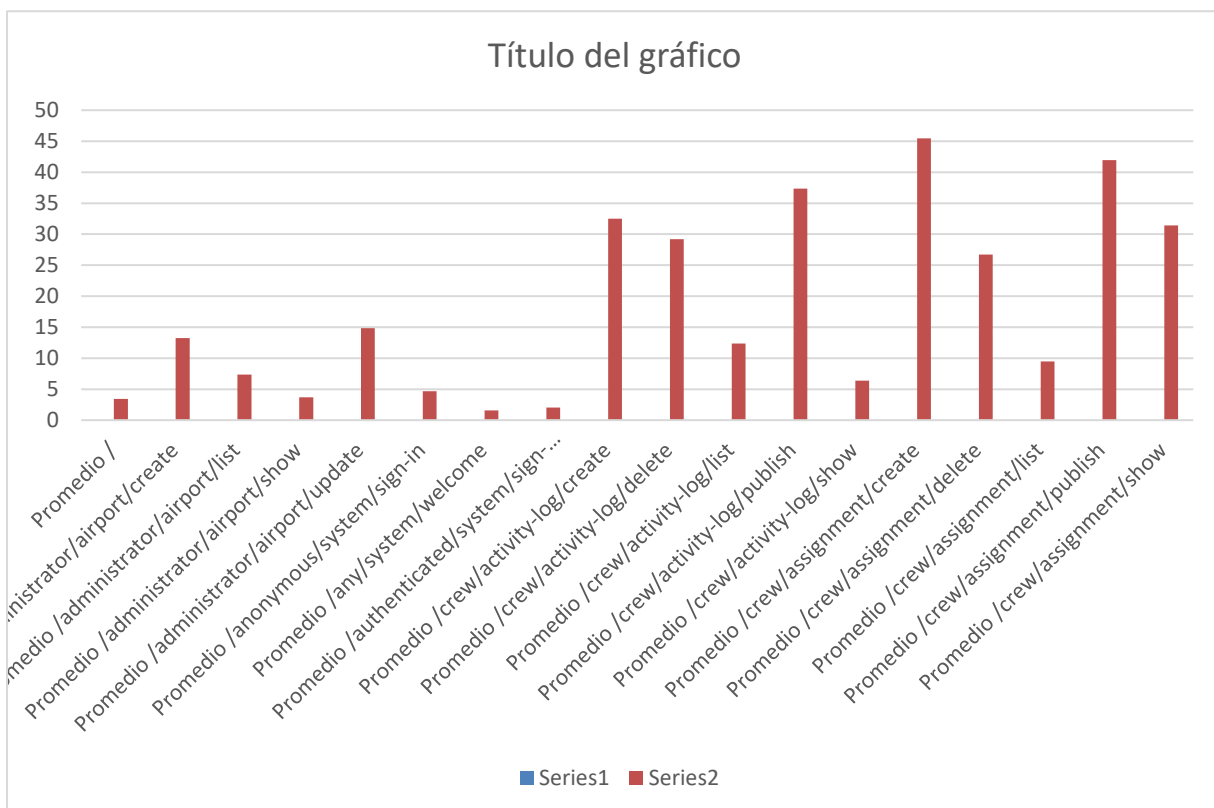
Interval(ms)	16,2700636	18,9497314
Interval(s)	0,01627006	0,01894973




Métricas para el caso 2

Estos han sido los resultados obtenidos:


Promedio /	3,450294286
Promedio /administrator/airport/create	13,23810851
Promedio /administrator/airport/list	7,3580625
Promedio /administrator/airport/show	3,692966667
Promedio /administrator/airport/update	14,8515875
Promedio /anonymous/system/sign-in	4,689438636
Promedio /any/system/welcome	1,58625
Promedio /authenticated/system/sign-out	2,0438
Promedio /crew/activity-log/create	32,50063167
Promedio /crew/activity-log/delete	29,202
Promedio /crew/activity-log/list	12,36272121
Promedio /crew/activity-log/publish	37,35386563
Promedio /crew/activity-log/show	6,38235
Promedio /crew/assignment/create	45,48028846
Promedio /crew/assignment/delete	26,6952
Promedio /crew/assignment/list	9,474
Promedio /crew/assignment/publish	41,958825
Promedio /crew/assignment/show	31,4113625
Promedio general	16,8521536



	Diseño y Pruebas II Acme-Software-Factory
	Testing Report

También se ha calculado el intervalo de confianza del 95 % para el tiempo que tarda en atender las solicitudes anteriores el ordenador de mi compañero. Obteniendo como resultado:

Interval(ms)	16,4205703	19,1994241
Interval(s)	0,01642057	0,01919942

	Diseño y Pruebas II Acme-Software-Factory
	Testing Report

Análisis de los resultados obtenidos

Análisis estadístico:

Para validar estadísticamente las mejoras observadas, se calculó un intervalo de confianza del 95% para los tiempos medios de respuesta para los resultados de ambos PC. Además, se realizó un contraste de hipótesis.

Columna1	
Media	17,60989751
Error típico	0,681978989
Mediana	12,0049
Moda	47,3465
Desviación estándar	15,41634584
Varianza de la muestra	237,6637192
Curtosis	3,985201759
Coeficiente de asimetría	1,650779428
Rango	107,9856
Mínimo	0,7429
Máximo	108,7285
Suma	8998,657628
Cuenta	511
Nivel de confianza(95,0%)	1,339833903

Ilustración 2 Mi PC

Columna1	
Media	17,80999717
Error típico	0,707221958
Mediana	12,3224
Moda	12,4312
Desviación estándar	15,98697096
Varianza de la muestra	255,5832406
Curtosis	3,176819573
Coeficiente de asimetría	1,656122161
Rango	96,7017
Mínimo	0,7849
Máximo	97,4866
Suma	9100,908553
Cuenta	511
Nivel de confianza(95,0%)	1,389426906

Ilustración 12. PC Compañero




Diseño y Pruebas II Acme-Software-Factory

Testing Report

Contraste de hipótesis al 95% de confianza:

Prueba z para medias de dos muestras		
	<i>Before</i>	<i>After</i>
Media	16,64588068	16,8521536
Varianza (conocida)	242,225008	260,130841
Observaciones	528	528
Diferencia hipotética de las medias	0	
z	-0,211472276	
P(Z<=z) una cola	0,416259382	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,832518764	
Valor crítico de z (dos colas)	1,959963985	

Ambos PCs presentan un rendimiento estadísticamente similar en términos de tiempos de respuesta. Aunque el PC de mi compañero ("After") tiene una media ligeramente mayor (16,8521536 frente a 16,64588068 de tu PC "Before"), la diferencia no es significativa con un 95% de confianza ($P = 0,832518764 > 0.05$), indicando que no hay evidencia de mejora o deterioro notable en el rendimiento entre ambos.

	Diseño y Pruebas II Acme-Software-Factory
	Testing Report


5. Conclusiones

Después de realizar las pruebas funcionales y de rendimiento, se concluye que el sistema satisface de forma adecuada los requisitos funcionales definidos para las entidades assignment y activity log. Las pruebas han validado con éxito tanto los casos positivos como los negativos, garantizando que se cumplen las restricciones establecidas y que el sistema opera de acuerdo con la lógica de negocio esperada.

Asimismo, se detectaron y resolvieron diversos errores y vulnerabilidades críticas, como validadores incorrectamente implementados y validaciones insuficientes en los formularios, lo que ha mejorado notablemente la estabilidad y seguridad de la aplicación. En relación con el análisis de rendimiento, los datos obtenidos reflejan que no hay diferencias estadísticamente significativas entre los dos entornos de ejecución evaluados, lo que demuestra que el sistema ofrece un desempeño uniforme independientemente del entorno, un indicativo positivo de su robustez y capacidad de escalado.

Además, no se pudo concluir correctamente el tester#replayer debido a un problema del framework comentado anteriormente. Aun así, individualmente en cada servicio el porcentaje de cobertura es bastante alto.

En definitiva, el proceso de pruebas ha resultado eficaz para confirmar la funcionalidad, identificar y corregir fallos, y evaluar el rendimiento del sistema, asegurando un producto más confiable y listo para su implementación.

	Diseño y Pruebas II Acme-Software-Factory
	Testing Report

6. Bibliografía

<https://ev.us.es/>