

TESTING REPORT

Testing Report producido por Rubén López Expósito (Student #5) para el proyecto Acme-ANS, trabajando con el grupo C1.056.

- **Índice:**

1. **Resumen**
2. **Tabla de Revisiones**
3. **Formal Testing**
 - 3.1. **Tasks**
 - 3.2. **Maintenance Records**
 - 3.3. **Involves**
 - 3.4. **Coverage**
4. **Performance Testing**
 - 4.1. **Antes de los índices**
 - 4.2. **Después de los índices**
 - 4.3. **Comparación y conclusiones**
5. **Conclusiones**

1 RESUMEN

Este informe realiza un análisis sobre los tests realizados sobre las funcionalidades realizadas por un technician sobre las entidades task, maintenance record e involves, tanto antes de añadir índices, como después de añadirlos.

2 TABLA DE REVISIONES

Fecha	versión	Cambios
26/05/2025	1.0.0	Versión Final
04/07/2025	2.0.0	Segunda Convocatoria

3 FORMAL TESTING

El Formal Testing consiste en una forma de grabar pruebas sobre nuestro proyecto de forma que más tarde sean reproducibles por otras personas. En el caso de Acme-ANS, para realizar esta función se utiliza un launcher llamado “testing-tester#record”, lo que te permite grabar un test case, hasta que llegas a tener todos los test case, formando una test suite, la cual se puede reproducir a través del launcher llamado “testing-tester#replay”, que además sirve para comprobar el buen funcionamiento de todos los test.

A continuación, voy a exponer un resumen de las pruebas hechas para cada funcionalidad, clasificándolas según la entidad:

3.1 TASKS:

- **Create:**
 - **En casos legales:** Probé a crear una task, iterando por todos los campos dejándolos vacíos y rellenándolos uno a uno con todos los valores posibles, tanto erróneos, como válidos, hasta subir una task creada de forma correcta.
 - **En casos ilegales (hacking):** Probé a hackear el campo TaskType, el cual es un campo navegable, introduciendo los valores “hacked” y “123123”, devolviéndome en ambos casos una excepción por no estar autorizado.
- **Delete:**
 - **En casos legales:** Probé a eliminar una task, tanto una que estuviese asociada a un maintenance record, para comprobar que se elimina bien la relación intermedia, como una que no esté asociado a ninguno.
 - **En casos ilegales (hacking):** Probé a hackear el campo TaskType, el cual es un campo navegable, introduciendo los valores “hacked” y “123123”, devolviéndome en ambos casos una excepción por no estar autorizado. Además, probé a intentar borrar a través de la url una task ya publicada, lo cual tampoco se permitió.
- **List:**
 - **En casos legales:** Probé a abrir la lista de tasks de un technician, así como probar las funcionalidades ofrecidas, como el buscador y el cambio de página.
 - **En casos ilegales (hacking):** Probé a meterme en la funcionalidad desde un realm no permitido, lo que dio lugar a una excepción de no autorización. También probé a introducir acciones en la url no permitidas, que dio lugar al mismo tipo de excepción.
- **Publish:**
 - **En casos legales:** Probé a publicar una task, iterando por todos los campos dejándolos vacíos y rellenándolos uno a uno con todos los valores posibles, tanto erróneos, como válidos, hasta poder publicar una task con todos los valores introducidos de forma correcta.

- **En casos ilegales (hacking):** Probé a hackear el campo TaskType, el cual es un campo navegable, introduciendo los valores “hacked” y “123123”, devolviéndome en ambos casos una excepción por no estar autorizado. Además, intenté publicar una task ya publicada a través de la url, lo cual dio lugar a una excepción.
- **Show:**
 - **En casos legales:** Probé a abrir una task de un technician, así como probar las funcionalidades ofrecidas, como desplegar los atributos navegables.
 - **En casos ilegales (hacking):** Probé a meterme a través de la url a una task de otro technician, así como a introducir la id de una task que no exista. También probé a introducir la url no estando logueado como un technician, así como también cambiar la acción show por “update”, “delete” y “publish”, dando todo esto lugar a excepciones del tipo not authorised.
- **Update:**
 - **En casos legales:** Probé a actualizar una task, iterando por todos los campos dejándolos vacíos y rellenándolos uno a uno con todos los valores posibles, tanto erróneos, como válidos, hasta poder actualizar una task con todos los valores introducidos de forma correcta.
 - **En casos ilegales (hacking):** Probé a hackear el campo TaskType, el cual es un campo navegable, introduciendo los valores “hacked” y “123123”, devolviéndome en ambos casos una excepción por no estar autorizado. Además, intenté actualizar una task ya publicada a través de la url, lo cual dio lugar a una excepción.

3.2 MAINTENANCE RECORDS:

- **Create:**
 - **En casos legales:** Probé a crear un maintenance record, iterando por todos los campos dejándolos vacíos y rellenándolos uno a uno con todos los valores posibles, tanto erróneos, como válidos, hasta subir un maintenance record creado de forma correcta. Una de las claves de probar la creación de un maintenance record ha sido probar la validación que no permite que su fecha de próxima inspección sea anterior a la de mantenimiento actual.
 - **En casos ilegales (hacking):** Probé a hackear los campos Maintenance Status y aircraft, los cuales son campos navegables, introduciendo los valores “hacked” y “123123” en el primero, y “-123” en el segundo, ya que al ser el segundo un campo con valores numéricos el hackeo con un valor de otro tipo daría lugar a otra excepción que no sería not authorised, la cual no podemos controlar.
- **Delete:**
 - **En casos legales:** Probé a eliminar un maintenance record, tanto uno que estuviese asociada a una task, para comprobar que se elimina bien la relación intermedia, como uno que no esté asociado a ninguna.
 - **En casos ilegales (hacking):** Probé a hackear los campos Maintenance Status y aircraft, los cuales son campos navegables, introduciendo los valores “hacked” y “123123” en el primero, y “-123” en el segundo, ya que al ser el segundo un campo con valores numéricos el hackeo con un valor de otro tipo daría lugar a otra excepción que no sería not authorised, la cual no podemos controlar. Además, probé a intentar borrar a través de la url un maintenance record ya publicado, lo cual tampoco se permitió.
 -
- **List:**
 - **En casos legales:** Probé a abrir la lista de maintenance records de un technician, así como probar las funcionalidades ofrecidas, como el buscador y el cambio de página.
 - **En casos ilegales (hacking):** Probé a meterme en la funcionalidad desde un realm no permitido, lo que dio lugar a una excepción de no autorización. También probé a introducir acciones en la url no permitidas, que dio lugar al mismo tipo de excepción.
- **Publish:**
 - **En casos legales:** Probé a publicar un maintenance record, iterando por todos los campos dejándolos vacíos y rellenándolos uno a uno con todos los valores posibles, tanto erróneos, como válidos, hasta poder publicar un maintenance record con todos los valores introducidos de forma correcta. Previo a esto, tuve que actualizar uno de los maintenance record al estado “Completed” y asociarle una acción que estuviera pública, ya que en caso contrario no podría haber realizado la publicación de este maintenance record.
 - **En casos ilegales (hacking):** Probé a hackear los campos Maintenance Status y aircraft, los cuales son campos navegables, introduciendo los

valores “hacked” y “123123” en el primero, y “-123” en el segundo, ya que al ser el segundo un campo con valores numéricos el hackeo con un valor de otro tipo daría lugar a otra excepción que no sería not authorised, la cual no podemos controlar. Además, probé a intentar publicar a través de la url un maintenance record ya publicado, lo cual tampoco se permitió.

○

- **Show:**

- **En casos legales:** Probé a abrir un maintenance record de un technician, así como probar las funcionalidades ofrecidas, como desplegar los atributos navegables.
- **En casos ilegales (hacking):** Probé a meterme a través de la url a un maintenance record de otro technician, así como a introducir la id de un maintenance record que no exista. También probé a introducir la url no estando logueado como un technician, así como también cambiar la acción show por “update”, “delete” y “publish”, dando todo esto lugar a excepciones del tipo not authorised.

- **Update:**

- **En casos legales:** Probé a actualizar un maintenance record, iterando por todos los campos dejándolos vacíos y rellenándolos uno a uno con todos los valores posibles, tanto erróneos, como válidos, hasta poder actualizar un maintenance record con todos los valores introducidos de forma correcta.
- **En casos ilegales (hacking):** Probé a hackear los campos Maintenance Status y aircraft, los cuales son campos navegables, introduciendo los valores “hacked” y “123123” en el primero, y “-123” en el segundo, ya que al ser el segundo un campo con valores numéricos el hackeo con un valor de otro tipo daría lugar a otra excepción que no sería not authorised, la cual no podemos controlar. Además, probé a intentar actualizar a través de la url un maintenance record ya publicado, lo cual tampoco se permitió.

3.3 INVOLVES:

- **Create:**

- **En casos legales:** Probé a añadir una task a un maintenance record, tanto siendo una task de mi propio technician, como una de otro technician que estuviese publicada.
- **En casos ilegales (hacking):** Probé a introducir la url en un realm que no fuera el que yo estaba asociado, es decir, que no fuera mi technician, lo cual dio lugar a una excepción not authorised. Además, intenté meterme en un maintenance record que no le pertenecía a mi technician a través de la url, así como a uno que no existía. Probé también a realizar la acción show a través de la url, lo cual daba lugar a una excepción debido a que el show se hace sobre un objeto involve y no un maintenance record. Por último, probé a intentar añadirle una tarea ya añadida anteriormente, una tarea de otro technician que no estuviera publicada, y, una tarea con un id que no existiera, dando lugar a una excepción en los tres casos.

- **Delete:**

- **En casos legales:** Probé a eliminar una task de un maintenance record.
- **En casos ilegales (hacking):** Probé a introducir la url en un realm que no fuera el que yo estaba asociado, es decir, que no fuera mi technician, lo cual dio lugar a una excepción not authorised. Además, intenté meterme en un maintenance record que no le pertenecía a mi technician a través de la url, así como a uno que no existía. Probé también a realizar la acción show a través de la url, lo cual daba lugar a una excepción debido a que el show se hace sobre un objeto involve y no un maintenance record. Por último, probé a intentar eliminar una tarea que no estuviera asociada anteriormente al maintenance record, así como una que no existiera, dando lugar a una excepción.

- **List:**

- **En casos legales:** Probé a abrir la lista de tasks asociadas a un maintenance record de un technician, así como probar las funcionalidades ofrecidas, como el buscador.
- **En casos ilegales (hacking):** Probé a meterme en la funcionalidad desde un realm no permitido, lo que dio lugar a una excepción de no autorización. También probé a introducir acciones en la url no permitidas, que dio lugar al mismo tipo de excepción.

- **Show:**

- **En casos legales:** Probé a abrir una relación involves de un technician, comprobando que se podía visualizar la tarea asociada a un maintenance record.
- **En casos ilegales (hacking):** Probé a meterme a través de la url a un involves que asociara un maintenance record de otro technician, así como a introducir la id de un involves que no exista. También probé a introducir la url no estando logueado como un technician, así como también cambiar la acción show por “list”, “delete” y “create”, dando todo esto lugar a excepciones del tipo not authorised.

3.4 COBERTURA DE CÓDIGO

▼	acme.features.technician.maintenanceRecord	95,3 %	1.288	63	1.351
>	TechnicianMaintenanceRecordDeleteService.java	78,3 %	224	62	286
>	TechnicianMaintenanceRecordPublishService.java	99,7 %	328	1	329
>	TechnicianMaintenanceRecordController.java	100,0 %	35	0	35
>	TechnicianMaintenanceRecordCreateService.java	100,0 %	220	0	220
>	TechnicianMaintenanceRecordListService.java	100,0 %	65	0	65
>	TechnicianMaintenanceRecordShowService.java	100,0 %	136	0	136
>	TechnicianMaintenanceRecordUpdateService.java	100,0 %	280	0	280
▼	acme.features.technician.task	95,1 %	905	47	952
>	TechnicianTaskDeleteService.java	75,3 %	143	47	190
>	TechnicianTaskController.java	100,0 %	35	0	35
>	TechnicianTaskCreateService.java	100,0 %	159	0	159
>	TechnicianTaskListService.java	100,0 %	65	0	65
>	TechnicianTaskPublishService.java	100,0 %	212	0	212
>	TechnicianTaskShowService.java	100,0 %	121	0	121
>	TechnicianTaskUpdateService.java	100,0 %	170	0	170
>	acme.entities.claims	75,0 %	84	28	112
>	acme.components	84,9 %	45	8	53
>	acme.features.customer.passengerRecord	99,2 %	526	4	530
▼	acme.features.technician.involves	99,5 %	768	4	772
>	TechnicianInvolvesCreateService.java	99,3 %	270	2	272
>	TechnicianInvolvesDeleteService.java	99,1 %	232	2	234
>	TechnicianInvolvesController.java	100,0 %	24	0	24
>	TechnicianInvolvesListService.java	100,0 %	117	0	117
>	TechnicianInvolvesShowService.java	100,0 %	125	0	125

Se puede observar que en todas las funcionalidades de technician se obtiene más de un 95% de coverage, siendo el problema general que baja el porcentaje de este tanto en task como en maintenance record, el servicio dedicado a la eliminación de estos. Esto se debe a que en las pruebas no se llega a cargar la funcionalidad unbind, pudiendo llegar a esconder algún bug que esconda algún problema. El coverage se mantiene igual en la versión 2.0.0 ya que no ha cambiado nada, por lo que las conclusiones siguen siendo las mismas.

```

82  @Override
83  public void unbind(final Task task) {
84      Dataset dataset;
85      SelectChoices typeChoices;
86
87      typeChoices = SelectChoices.from(TaskType.class, task.getTaskType());
88
89      dataset = super.unbindObject(task, "taskType", "description", "priority", "estimatedDuration", "technician");
90      dataset.put("types", typeChoices);
91      dataset.put("draftMode", task.getDraftMode());
92
93      super.getResponse().addData(dataset);
94  }

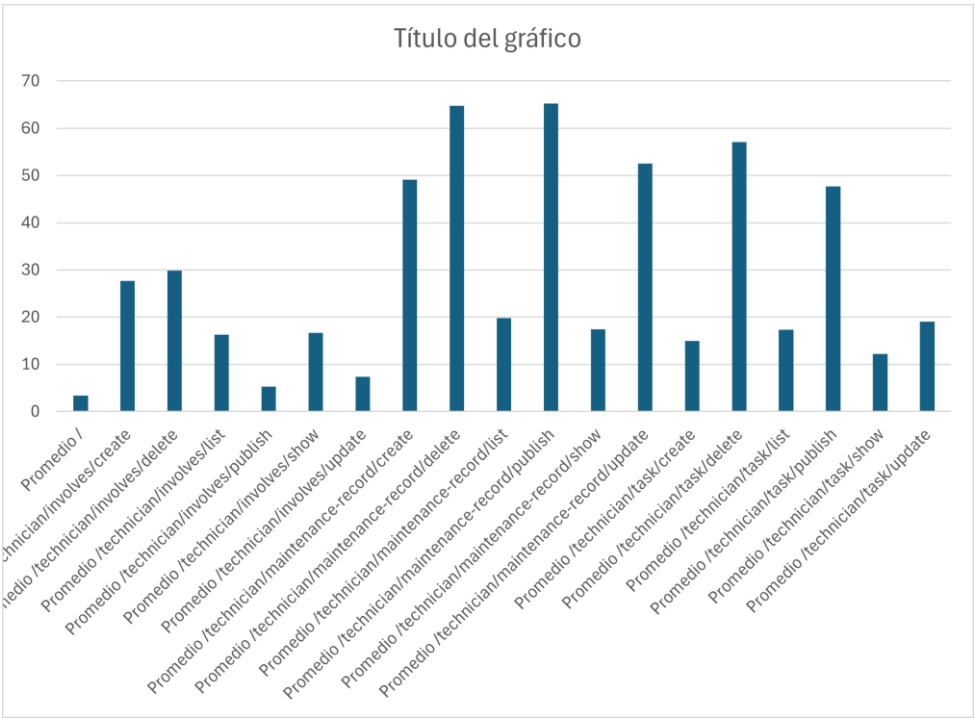
107  @Override
108  public void unbind(final MaintenanceRecord maintenanceRecord) {
109      Dataset dataset;
110      SelectChoices statusChoices;
111      SelectChoices aircraftChoices;
112      Collection<Aircraft> aircrafts;
113
114      aircrafts = this.repository.findAllAircrafts();
115
116      aircraftChoices = SelectChoices.from(aircrafts, "registrationNumber", maintenanceRecord.getAircraft());
117      statusChoices = SelectChoices.from(MaintenanceStatus.class, maintenanceRecord.getMaintenanceStatus());
118
119      dataset = super.unbindObject(maintenanceRecord, "maintenanceMoment", "nextInspectionDate", "estimatedCost", "notes", "technician");
120      dataset.put("statuses", statusChoices);
121      dataset.put("draftMode", maintenanceRecord.getDraftMode());
122      dataset.put("aircrafts", aircraftChoices);
123
124      super.getResponse().addData(dataset);
125  }

```


4 PERFORMANCE TESTING

En este apartado vamos a observar estadísticas respecto a las pruebas formales realizadas anteriormente, tanto antes de añadir índices que deberían mejorar estas estadísticas, como después

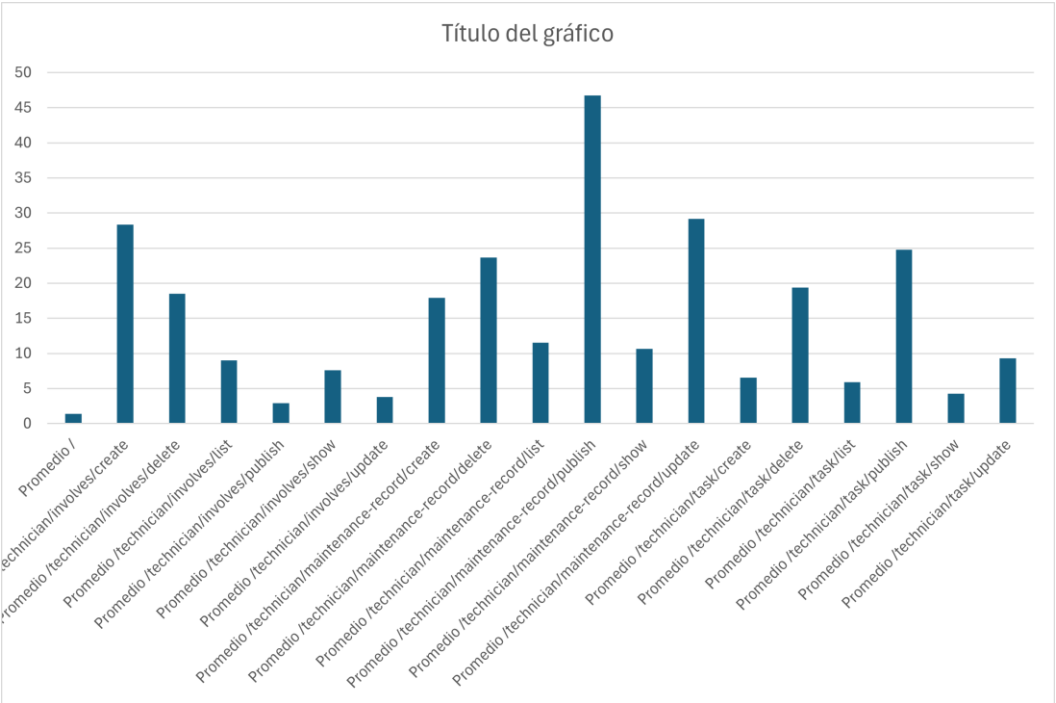
4.1 ANTES DE AÑADIR ÍNDICES



Tiempo de respuesta promedio por cada request antes de añadir los índices

	A	B	C	D	E	F
1	<i>Antes de los índices</i>					
2				Confidence Interval (ms):	10,6837472	12,166356
3	Media	11,4250516		Confidence Interval (s):	0,01068375	0,01216636
4	Error típico	0,37799389				
5	Mediana	3,9991				
6	Moda	1,9723				
7	Desviación estándar	16,8832572				
8	Varianza de la muestra	285,044372				
9	Curtosis	8,03168891				
10	Coeficiente de asimetría	2,62743756				
11	Rango	135,5408				
12	Mínimo	1,2074				
13	Máximo	136,7482				
14	Suma	22792,9779				
15	Cuenta	1995				
16	Nivel de confianza(95,0%)	0,74130439				

4.2 DESPUÉS DE AÑADIR ÍNDICES



Tiempo de respuesta promedio por cada request después de añadir los índices

	A	B	C	D	E	F
1	<i>Después de los índices</i>					
2				Confidence Interval (ms):	8,26172714	10,6736787
3	Media	9,467702916		Confidence Interval (s):	0,00826173	0,01067368
4	Error típico	0,61449996				
5	Mediana	4,1968				
6	Moda	0,8002				
7	Desviación estándar	18,69938625				
8	Varianza de la muestra	349,667046				
9	Curtosis	80,92839198				
10	Coefficiente de asimetría	7,720203752				
11	Rango	235,9229				
12	Mínimo	0,652				
13	Máximo	236,5749				
14	Suma	8767,0929				
15	Cuenta	926				
16	Nivel de confianza(95,0%)	1,205975777				

4.3 COMPARACIÓN Y CONCLUSIONES

Se puede observar que el tiempo de respuesta promedio, tanto en cada request, como en total, es menor después de añadir los índices, lo que corrobora los índices ayudan a mejorar la velocidad de algunas consultas, lo que lleva a un mejor tiempo de respuesta.

Además, en la imagen siguiente podemos ver la “prueba z para medias de dos muestras”, la cual compara las dos muestras de datos que tenemos para antes y después de añadir los índices, y expone si las medias de las dos muestras son comparables o no. Para que sean comparables, el Valor crítico de z (dos colas), que es en este caso el que vemos en la fila 11, debe estar entre 0 y alfa, siendo alfa en este caso un índice que mide el nivel significativo entre las muestras, que vale 0.05. Como podemos observar, en este caso su valor es 0, por lo que las muestras son perfectamente comparables. Podemos llegar a la conclusión de que en estas pruebas se ha mejorado la media del tiempo de respuesta al añadir algunos índices a las entidades.

	A	B	C
1	Prueba z para medias de dos muestras		
2			
3		136,7482	56,8568
4	Media	18,44643903	9,416471459
5	Varianza (conocida)	469,7366149	349,667046
6	Observaciones	925	925
7	Diferencia hipotética de las medias	0	
8	z	9,594185406	
9	P(Z<=z) una cola	0	
10	Valor crítico de z (una cola)	1,644853627	
11	Valor crítico de z (dos colas)	0	
12	Valor crítico de z (dos colas)	1,959963985	

5 CONCLUSIONES

Al observar los resultados de las pruebas realizadas se puede observar un muy buen rendimiento al realizar la prueba con los índices, el cual ha decaído al quitar los índices, ya que los índices ayudan a hacer más rápidas algunas consultas.

Por tanto, se puede concluir que las pruebas realizadas sobre las funcionalidades de las entidades asociadas a technician, y que por tanto, son las pruebas realizadas por el student #5, son exitosas, ya que tienen un coverage mayor del 95% en cada entidad, así como un muy buen rendimiento con índices.