



INFORME DE ANÁLISIS

Grupo 1-C1.020 | <https://github.com/PDJ6975/Acme-ANS-D02-25.2.0>



Nombre	Correo Corporativo
Antonio Rodríguez Calderón	antrodcal@alum.us.es
Adrián Ramírez Gil	adrramgil@alum.us.es
Jianwu Hu	jiahu3@alum.us.es
Pablo Castrillón Mora	pabcasmor1@alum.us.es
Pablo Olivencia Moreno	pabolimor@alum.us.es

Tabla de Contenido

- 1. Resumen Ejecutivo 2**
- 2. Tabla de Revisión 2**
- 3. Introducción 2**
- 4. Registros de Análisis 3**
 - 4.1 Relacionados con la primera entrega 3
 - 4.1.1 Requisito 2: 3
 - 4.2.2 Requisito 24/25:..... 4
 - 4.2 Relacionados con la segunda entrega..... 5
 - 4.2.1 Requisito 5: 5
 - 4.2.2 Requisito 26:..... 6
 - 4.2.3 Requisito 6: 7
 - 4.2.4 Requisito 4: 9
- 5. Conclusión..... 10**
- 6. Bibliografía..... 11**

1. Resumen Ejecutivo

Este informe detalla los análisis realizados sobre los requisitos individuales del estudiante número tres del proyecto, abordando las dudas que surgieron durante su desarrollo, las posibles soluciones evaluadas y las decisiones finales adoptadas.

A través de este informe, se busca reflejar el proceso llevado a cabo para garantizar el cumplimiento adecuado de los requisitos, minimizando ambigüedades y asegurando la coherencia con la metodología de trabajo establecida.

2. Tabla de Revisión

Versión	Fecha	Descripción de los cambios
1.0	18/02/2025	Creación inicial del documento
1.1	19/02/2025	Corregir números de contenido
2.0	28/02/2025	Actualizar contenido para el segundo entregable
2.1	09/03/2025	Introducir más decisiones

3. Introducción

Durante el desarrollo del proyecto, algunos requisitos individuales crearon dudas sobre su correcta implementación y documentación. Este informe tiene como objetivo comentar dichas dudas, analizar las alternativas consideradas y justificar la elección final de la solución adoptada.

Cada análisis presentado sigue una estructura clara: primero, se expone el requisito específico que generó la duda; luego, se describen las opciones contempladas para abordarlo, analizando ventajas e inconvenientes; finalmente, se detalla la decisión tomada y su justificación (generalmente en base a la respuesta del profesor).

4. Registros de Análisis

En este apartado se aborda en primera persona el análisis introducido en los apartados anteriores de forma completa.

4.1 Relacionados con la primera entrega

4.1.1 Requisito 2:

“Provide a link to your planning dashboard in GitHub to review the tasks, their current status, and your schedule”

Al leer este requisito, me surgió la duda sobre dónde debía colocar el enlace solicitado. Al revisar el documento **“On Your Deliverables”**, observé que el espacio debajo del requisito solo debía usarse para indicar con una "X" si estaba completado o no, pero no se especificaba dónde debía incluirse el enlace. A partir de esto, consideré dos opciones posibles:

- **Opción 1: Colocar el enlace en la entrega de Enseñanza Virtual.**

Pensé en incluirlo en el campo de la entrega donde se adjunta el proyecto. Sin embargo, el documento de instrucciones no menciona esta posibilidad explícitamente, por lo que no estaba seguro de si sería la mejor opción.

- **Opción 2: Incluir el enlace en el fichero “README” del proyecto.**

En un principio, esta opción me generaba dudas, ya que no veía claro que el README fuera el lugar adecuado para esta información. Además, no estaba seguro de si los profesores iban a revisar este fichero.

Para resolver esta pregunta, en una sesión de seguimiento de laboratorio, un compañero planteó esta misma duda al profesor. Su respuesta confirmó que colocar el enlace en el README era una opción válida.

Conclusión: Finalmente, decidí incluir el enlace en el README del proyecto, ya que cumple con el requisito y fue validado como una solución correcta en clase.

4.2.2 Requisito 24/25:

“Produce an analysis report”/“Produce a planning and progress report”

Cuando leí estos requisitos, me surgió la duda de cómo debía abordarlos, ya que ambos estaban definidos tanto como requisitos individuales suplementarios como requisitos grupales suplementarios. No tenía claro cuál era la diferencia entre ambas versiones ni cómo debía enfocarlos en cada caso.

A partir de esto, consideré dos posibles interpretaciones:

- **Opción 1: Tomar una versión global completa en el informe grupal y una individual completa para los individuales**

En el informe grupal se debían tratar todos los requisitos del proyecto, tanto grupales como individuales, realizados por todos los miembros del equipo. Mientras que en el informe individual debía hablar únicamente de los requisitos grupales e individuales en los que yo hubiera trabajado personalmente.

- **Opción 2: Realizar una distinción clara entre grupales e individuales**

En el informe grupal solo debían abordarse los requisitos grupales, mientras que en el informe individual solo se debían tratar los requisitos individuales propios de cada miembro.

Mi duda principal era si los miembros del equipo debíamos considerar los requisitos individuales de los demás, o si cada uno era responsable de su parte individual.

Para aclararlo, planteé la cuestión al profesor en una clase de laboratorio, y su respuesta fue que nos podíamos organizar como quisiéramos, pero que los requisitos individuales eran responsabilidad exclusiva de cada persona. Así, una buena manera de estructurar los informes sería siguiendo la Opción 2.

Conclusión: Es por ello por lo que al final me decanté por seguir la segunda opción, realizando una clara separación entre los tipos de requisitos.

4.2 Relacionados con la segunda entrega

4.2.1 Requisito 5:

“An activity log records incidents that occur during a flight. They are logged by any of the flight crew members assigned to the corresponding leg and after the leg has taken place. The incidents include weather-related disruptions, route deviations, passenger issues, or mechanical failures, to mention a few. Each log entry includes a registration moment (in the past), a type of incident (up to 50 characters) a description (up to 255 characters), and a severity level (ranging from 0 to 10, where 0 indicates no issue and 10 represents a highly critical situation).”

La duda con este requisito está relacionada principalmente con la necesidad de crear la entidad “ActivityLog (nombre de ejemplo)”. Como se menciona, una etapa de vuelo va a tener asociado un registro de actividad, de manera que cada entrada del mismo tendrá asociado un incidente registrado por cualquier miembro de la tripulación asignado en esa etapa.

Bien, el problema viene en que entonces un registro de actividad no es más que un listado de incidentes, de manera que se puede cuestionar la existencia de la entidad.

A partir de esto, consideré dos posibles interpretaciones:

- **Opción 1: Mantener la existencia de la entidad “Activity Log”.**

En esta opción se busca mantener la entidad, de forma que desde ella se puede acceder unidireccionalmente a la etapa de vuelo “FlightStage” mediante una relación @OneToOne. Del mismo modo, un registro tiene una relación con un listado de incidentes, pero se accede unidireccionalmente desde los incidentes mediante una relación @ManyToOne.

Esta opción puede generar el problema de que al final tienes una entidad prácticamente vacía y con una relación @OneToOne que no es muy utilizada en la asignatura, ya que se insiste mucho en las relaciones @ManyToOne.

- **Opción 2: Eliminar la entidad “Activity Log”**

La otra opción contemplada sería eliminar la entidad y mantener una relación directa @ManyToOne entre “FlightStage” y la lista de incidentes (cada uno es una entidad “LogEntry”). En un principio, parece la solución más viable ya que estás eliminando una entidad y una relación que en un principio no son necesarias.

Conclusión: Al final, me decanté por utilizar la segunda opción. Así, nos va a quedar una relación @ManyToOne directa y un diagrama UML más limpio.

4.2.2 Requisito 26:

“The system is required to provide crew members with information about visa requirements. A web service must be used to populate this entity with information about visa requirements. Thus, the exact data to store depends on the chosen service, and it is the students' responsibility to define them accordingly. It is also the students' responsibility to find the appropriate service; no implicit or explicit liabilities shall be covered by the University of Seville or their individual affiliates if the students hire pay-per-use services! The students are strongly advised to ensure that the service they choose is free of charge.”

La duda con este requisito ha sido principalmente con qué API decantarme, ya que había encontrado dos opciones bastante viables, una API abierta de GitHub gratuita y una API con un plan gratuito de 600 peticiones mensuales.

A partir de esto, consideré dos posibles interpretaciones:

- **Opción 1: Utilizar la API gratuita.**

El problema de esta opción es que la API devolvía muy poca información sobre los requisitos de visa, generalmente indicando solo el tipo de visa necesario, lo que podía ser insuficiente para cumplir con la tarea, pues al final tendríamos una sola entidad con uno o dos atributos a almacenar.

- **Opción 2: Utilizar la API de pago con el plan gratuito (Travel Buddy)**

La otra opción contemplada sería utilizar la API de pago con el plan gratuito de 600 peticiones. Lo bueno de esta opción es que la API devolvía mucha más información relevante e interesante para cumplir con la tarea. Me preocupaba que consumiese más de una petición por llamada, pero no era así.

Conclusión: Al final, me decanté por utilizar la segunda opción, pues pienso que 600 peticiones al mes son más que suficientes para cumplir con la tarea de la asignatura sin llegar a su límite, manteniendo así una tarea interesante.

4.2.3 Requisito 6:

“Produce assorted sample data to test your application informally. The data must include four flight crew member accounts with credentials “memberX/memberX” with X ranging from 1 to 4 (and different duties each). Additionally, create a fifth member account member/member, representing a new member with no flight assignment, as if the account had just been created.”

La duda principal en este requisito ha sido al poblar el rol “FlightCrewMember”, pues en la tarea se exige crear cinco cuentas de usuario para estos miembros en

total. El problema está en que no se habla de poder incluir más cuentas para este rol, de forma que el número de filas en el csv de “FlightCrewMember” puede verse limitado.

Si bien la aplicación permite como tal que una cuenta de usuario pueda estar asociada a más de una instancia de un rol por una relación @ManyToOne, es algo que carece de sentido en la práctica.

Así, si se mantienen solo las cinco cuentas, podríamos crear cinco filas como máximo en el csv de FlightCrewMember, en donde asociemos un miembro a cada cuenta, por lo que no se puede probar de forma exacta y completa todos los atributos de la entidad puesto que el tamaño limitado no permite probar todas las variaciones.

A partir de esto, consideré dos posibles interpretaciones:

- **Opción 1: Asociar cada cuenta de usuario con más de un miembro**

El problema de esta opción es que en la práctica carece de sentido, aunque la ventaja es que se podría probar de forma completa la cobertura de la entidad.

- **Opción 2: Asociar cada cuenta a un solo miembro**

Si bien con esta opción la cobertura disminuye, seguiría el ejemplo de roles establecidos en Acme Job como “Employer”, que asocia un solo empleado a cada cuenta.

Conclusión: Al final, me decanté por utilizar la segunda opción, pues creo que basarse en Acme Job es lo más viable, y si bien la cobertura puede disminuir, en tan solo cinco filas creo que se pueda abordar de forma bastante completa todas las variaciones, puesto que la entidad tiene pocos atributos que verdaderamente haya que probar.

4.2.4 Requisito 4:

“A flight assignment represents the allocation of a flight crew member to a specific leg of a flight. Each assignment specifies the flight crew's duty in that leg ("PILOT", "COPILOT", "LEAD ATTENDANT", "CABIN ATTENDANT"), the moment of the last update (in the past), the current status of the assignment ("CONFIRMED", "PENDING", or "CANCELLED"), and some remarks (up to 255 characters), if necessary.”

La duda principal con este requisito viene con el atributo “remarks”, puesto que se especifica en plural y por tanto puede hacer dudar de si debe ser una lista de comentarios (lista de String), o un único String.

En este sentido, si se establece como una lista, no se podría validar directamente a través del framework que cada comentario tenga una longitud máxima de 255 caracteres, mientras que si se usa directamente un String sí se podría, utilizando la notación @ValidString.

A partir de esto, consideré dos posibles interpretaciones:

- **Opción 1: Dejar una lista y de alguna manera en el servicio limitar la longitud de cada comentario al querer crear o editar una asignación de vuelo**

El problema de esta opción es que es bastante más compleja que simplemente validar en la entidad y va a quedar un servicio menos limpio.

- **Opción 2: Dejar el atributo como un único String en el que se utilice la notación @ValidString**

Con esta opción el código sería mucho más limpio y sencillo de entender, de forma que se promueve la separación de responsabilidades y la validación a nivel de entidad.

Conclusión: Al final, me decanté por utilizar la segunda opción, pues pienso que es la mejor a la hora de llevar a cabo el desarrollo y creo que cumple con lo que la tarea quiere especificar.

Por otra parte, este atributo también ha generado otra duda, y es que, al ser opcional, aunque limites el `@ValidString` a “min=1”, permitiría la inserción de valores vacíos que no se convertirían a nulo. Esto no ocurre cuando es obligatorio porque la notación directamente lo convertiría a nulo.

A partir de esto, consideré dos posibles interpretaciones:

- **Opción 1: Introducir algún tipo de notación como `@NotBlank` o crear un validador personalizado**

El problema de esta opción es que puede ser tediosa y no cumplir a la perfección con lo exigido en la inmensa mayoría de entidades que se desarrollan con el framework.

- **Opción 2: Hacer un setter propio para ese atributo**

En vez de dejar que lombok genere un setter para ese atributo, se puede crear uno personalizado de manera que, si el comentario no es nulo y al hacer el “trim” está vacío, se establezca a nulo.

Conclusión: Al final, me decanté por utilizar la segunda opción, pues la más sencilla y va a añadir un bloque de seguridad adicional a la aplicación.

5. Conclusión

Este informe ha documentado las dudas surgidas durante el desarrollo del proyecto, analizando distintas alternativas y justificando las decisiones tomadas en cada caso. Se ha seguido un enfoque estructurado para evaluar las opciones disponibles y validar la solución más adecuada en cada situación.

El análisis realizado permite garantizar una interpretación precisa de los requisitos y una correcta aplicación de los criterios establecidos.

6. Bibliografía

Intencionalmente en blanco.