Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Diseño y Pruebas II**

**Analysis Report D02 – Student #1**



Grado en Ingeniería Informática – Ingeniería del Software  
Curso 2024 – 2025

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Repositorio** |
| 26/05/2025 | 2.0 | <https://github.com/javsorbla/Acme-ANS-D02> |

|  |  |  |
| --- | --- | --- |
| **Grupo de prácticas: C1.032** | |  |
| **Autores por orden alfabético** | **Roles** | **Correo Corporativo** |
| Castilla Rodríguez, Javier | Tester, Developer | javcasrod1@alum.us.es |
| De los Reyes Pérez, Alejandro | Operator, Tester, Developer | alereyper@alum.us.es |
| Del Pino Escalante, Nuno José | Analyst Tester, Developer | nundelesc@alum.us.es |
| Gutiérrez Pastor, Javier | Tester, Developer | javgutpas@alum.us.es |
| Soria Blanco, Javier | Manager, Tester, Developer | javsorbla@alum.us.es |

**Fecha:** 26/05/2025

**Tabla de contenidos**

[**1.** **Resumen Ejecutivo**: 2](#_Toc190960910)

[**2.** **Tabla de revisión**: 3](#_Toc190960911)

[**3.** **Introducción**: 4](#_Toc190960912)

[**4.** **Contenidos**: 5](#_Toc190960913)

[**4.1** **Registro de análisis**: 5](#_Toc190960914)

[**5.** **Conclusiones**: 9](#_Toc190960915)

[**6.** **Bibliografía**: 10](#_Toc190960916)

### **Resumen Ejecutivo**:

Este informe detalla el análisis de los requisitos abordados en la segunda entrega del proyecto, destacando los principales desafíos identificados durante el proceso. Además, se describe el proceso de análisis y resolución de los problemas encontrados, proporcionando una visión clara del procedimiento seguido y de las soluciones implementadas, justificando por qué se han escogido dichas soluciones.

El objetivo de este informe es por tanto mejorar la comprensión del proceso de análisis de requisitos y explicar las acciones que se han tomado.

### **Tabla de revisión**:

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 13/03/2025 | 1.0 | Creación y finalización del documento |
| 26/05/2025 | 2.0 | Corrección de decisiones tomadas |
|  |  |  |
|  |  |  |

### **Introducción**:

En este documento se explicará el análisis realizado de todos los requisitos individuales de esta entrega. Para ello, se proporcionará un registro de análisis de cada requisito problemático, que incluirá una copia o captura de pantalla del requisito en cuestión, los problemas encontrados y las soluciones planteadas. Además, en los casos en los que había varias alternativas posibles en un principio, se explicará por cuál de ellas se ha optado y el motivo de ello. De esta manera, se ofrece una visión general del proceso de análisis de requisitos, facilitando su comprensión.

### **Contenidos**:

### **Registro de análisis**:

Esta segunda entrega ha sido más compleja que la anterior, de manera que hay varios requisitos que han causado problemas de entendimiento, haciendo dudar sobre la forma correcta de implementarlos, tal que se satisficieran de la mejor manera posible las necesidades del cliente. En primer lugar, el primer requisito es claro y no presenta ninguna ambigüedad, de manera que no fue necesario realizar un análisis detallado, pasando directamente a la planificación e implementación de este.

Antes de enumerar los problemas encontrados, cabe destacar que la mayoría de rangos en los que están limitados los atributos de las entidades y roles solicitados se han basado en un **Excel** compartido por un compañero de otro grupo por el foro, en el que el profesor Rafael Corchuelo aclaró la gran mayoría de valores límite a establecer. En concreto, se tomó como base la última versión de dicho archivo, compartido en respuesta a la alumna María Quirós.

Enlace al foro en cuestión: <https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_89154_1&nav=discussion_board&conf_id=_426211_1&forum_id=_253522_1&message_id=_460989_1>

Dicho esto, se empezará a comentar el primer requisito:

Texto

El contenido generado por IA puede ser incorrecto.

Hubo varios atributos que resultaron ambiguos:

Entre ellos, se encuentra el identificador, puesto a que no se tenía claro en un principio cómo realizar la comprobación de las iniciales. Sin embargo, tras un [hilo](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_89154_1&nav=discussion_board&conf_id=_426211_1&forum_id=_253522_1&message_id=_460881_1) en el foro y avanzar en el contenido de teoría, esta duda se resolvió de forma natural, llegando a la conclusión de que sería necesario implementar un validador personalizado.

Así mismo, el profesor José González nos aconsejó, durante la última sesión de seguimiento de este entregable, implementar las comprobaciones de patrones creando un validador que pudiera ser reutilizado en otros atributos, de manera que se realizó en todos los casos en los que aplicaba, incluido el identificador. Por tanto, abandonamos la anotación @ValidString, nuestra primera opción, para esos casos.

Los años de experiencia también ocasionaron problemas, especialmente debido a que el límite superior no quedaba claro al analizar los requisitos, pero se acabó optando por un máximo de 120, tal y como se estableció en el archivo Excel. [Enlace](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_89154_1&nav=discussion_board&conf_id=_426211_1&forum_id=_253522_1&message_id=_460339_1) a un hilo con la explicación de por qué es válido escoger este valor.

El resto de los atributos son lo suficientemente explícitos como para no ocasionar mayores problemas.

Sin embargo, surgió una duda respecto a si era necesaria una relación entre el mánager y la aerolínea, pero analizando los requisitos de entregas futuras, rápidamente se llegó a la conclusión de que no sería necesaria, lo cuál fue confirmado en una sesión de follow-up.

En cuanto al segundo requisito:

Texto

El contenido generado por IA puede ser incorrecto.

El atributo que indica si hay auto conexión(*self-transfer*) generó problemas, al igual que todos los booleanos en un principio, ya que ningún miembro tenía claro si tenía sentido la anotación @Valid para los booleanos, especialmente primitivos. La respuesta fue sencilla, durante una sesión de follow-up se nos indicó que no se debían poner. El resto de los atributos, si bien al principio no se disponía de los conocimientos suficientes para implementar las propiedades derivadas, no presentaron problemas, especialmente tras avanzar en la teoría de la asignatura y adquirir los conocimientos necesarios.

El siguiente requisito dice lo siguiente:

Texto

El contenido generado por IA puede ser incorrecto.

Esta fue la entidad que más problemas generó. En un principio, no se sabía como validar el número de vuelo, aunque al igual que con el identificador de los mánagers, este problema se resolvió naturalmente, pues estaba claro que había que implementar un validador customizado, pero no se conocía la manera de hacerlo.

Las fechas de despegue y de aterrizaje también generaron problemas relacionados con los límites máximos y mínimos. Al principio se barajaban 2 alternativas: Validar que el aterrizaje ocurriera después del despegue, o sencillamente comprobar que ambas estuvieran en el futuro. Evidentemente, la solución correcta era la de implementar la validación, algo que se comprobó gracias al archivo Excel mencionado anteriormente. Además, gracias a ese archivo surgió una duda nueva, sobre si es necesario o no comprobar que los tramos de un vuelo no se solapen. Tras consultar el foro, se llegó a la conclusión de que sería necesario implementar dicha validación en los tramos, verificando que los tramos publicados no se solapen. Hacerlo de esta manera permite que si hay alguna inconsistencia se puedan solventar, ya que si se comprobara en el vuelo, no sería posible modificar las fechas de sus legs ya publicadas. De esta manera, el mánager publicará progresivamente las legs de un vuelo, asegurando que no estén solapadas, para después publicar el vuelo. El hilo consultado fue el [siguiente](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_89154_1&nav=discussion_board&conf_id=_426211_1&forum_id=_253522_1&message_id=_462236_1) (dentro del mismo hay un enlace a otro hilo relacionado que también fue de ayuda).

Por otra parte, se decidió que era necesario verificar que el aircraft que se usara en una leg estuviera en estado activo, ya que no tiene mucho sentido que se puedan programar y publicar tramos de vuelo con aviones que están en mantenimiento, de manera que se limita en el leg validator. Esto se verificó durante las clases de follow up.

Evidentemente, no tiene sentido que un tramo de vuelo aterrice en el mismo aeropuerto desde el que despegó, excepto en casos de emergencia, que no se contemplarán. Esto se verificó a través del siguiente post del foro: Así mismo, tampoco deben existir tramos de vuelo que aterricen en la misma ciudad en la que aterrizaron, ya que no realista, y no tiene demasiado sentido permitirlo. Tal y como se dice en el siguiente post del foro, no parece tener sentido que un tramo de vuelo vaya de Sevilla a Sevilla, por ejemplo. [Enlace al post donde se verifican ambas validaciones](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_89154_1&nav=discussion_board&conf_id=_426211_1&forum_id=_253522_1&message_id=_470352_1).

### **Conclusiones**:

En conclusión, los requisitos de esta entrega han supuesto más problemas que los de la anterior, pero gracias al foro y a las sesiones de seguimiento, se tomaron decisiones que llevaron a implementar el modelo de datos de una manera suficientemente robusta, creando una base que permita añadir nuevas validaciones en el futuro en caso de que las próximas entregas saquen a relucir defectos complicados de ver tras el análisis de requisitos realizado en esta entrega.

A diferencia de la anterior entrega, el hecho de analizar muchos de estos problemas de manera individual ha causado complicaciones inesperadas. Sin embargo, el resultado es lo suficientemente satisfactorio.

### **Bibliografía**:

Intencionalmente en blanco.