Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Diseño y Pruebas II**

**Analysis Report D03 – Student #1**



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

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Repositorio** |
| 26/05/2025 | 1.0 | <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](#_Toc199155467)

[**2.** **Tabla de revisión**: 3](#_Toc199155468)

[**3.** **Introducción**: 4](#_Toc199155469)

[**4.** **Contenidos**: 5](#_Toc199155470)

[**4.1** **Registro de análisis**: 5](#_Toc199155471)

[**5.** **Conclusiones**: 7](#_Toc199155472)

[**6.** **Bibliografía**: 8](#_Toc199155473)

### **Resumen Ejecutivo**:

Este informe detalla el análisis de los requisitos abordados en la tercera 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 llevado acabo.

### **Tabla de revisión**:

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 26/05/2025 | 1.0 | Creación y finalización del documento |
|  |  |  |
|  |  |  |

### **Introducción**:

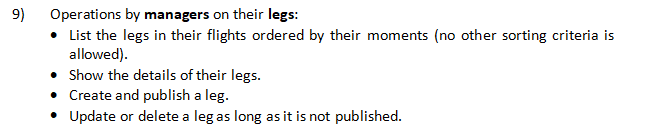
En este documento se explicará el análisis realizado de todos los requisitos individuales de esta entrega. Para ello, se proporcionará un registro de análisis de cada requisito problemático, que incluirá los problemas encontrados y las soluciones planteadas. Además, en los casos en los que había varias 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.

Cabe destacar que se realizaron cambios respecto a la entrega anterior, especialmente en cuanto a la realización de algunas de las validaciones.

### **Contenidos**:

### **Registro de análisis**:

La primera duda surge sobre el siguiente requisito:



En esta tercera entrega, surgieron varios problemas respecto a dónde se debían realizar ciertas validaciones, es decir, si en los propios servicios o en los validadores custom que se crearon en la entrega pasada.

En primer lugar, surgió un problema respecto a la validación de la fecha de salida y llegada de los tramos, ya que si se realizaban en el validador, es decir, forzábamos a que se comprobaran todos los datos de prueba al popular la base de datos, surgiría un conflicto con las fechas de las entidades de los estudiantes 3 y 4. Nos centraremos en el conflicto del estudiante 3, aunque todo lo que se diga aplica al 4. La situación fue la siguiente: Un leg debe tener el momento de salida y llegada en futuro. En nuestro entorno de pruebas, fijamos ese momento al 01/01/2025 00:00. Sin embargo, los activity logs necesitan estar asignados a flight assignments cuyos tramos ya se encuentren en el pasado, es decir, hayan aterrizado antes de la fecha anteriormente mencionada. Esto, sin embargo, no es posible conseguirlo en los datos de prueba de los .csv, ya que si se pone alguna fecha de los tramos en pasado se incumpliría la restricción, lo que resultaría en un error de validación. Ante esto, realizamos un intercambio de mensajes en el foro con uno de los profesores de la asignatura, Rafael Corchuelo. Tras ello, y con la ayuda de nuestro profesor de laboratorio, llegamos a las siguientes alternativas:

* **Solución 1**: el estudiante 3 no crea datos de prueba para los activity logs*.*
* **Solución 2**: añadir una nueva propiedad de tipo Date que permita elaborar datos de prueba válidos.
* **Solución 3**: el estudiante 1 elimina del LegValidator la validación que no permite crear tramos pasados y la añade a los servicios de creación, actualización y publicación de tramos, lo que permite incluir tramos pasados en los datos de prueba, que pueden ser usados por el estudiante 3, sin permitir su creación desde la aplicación.

En nuestro optamos por la alternativa 3, que fue aceptada por José González, nuestro profesor de laboratorio, además de la profesora de teoría del grupo 2. Esto se debe a que no generar datos de prueba para los activity logs sería problemático de cara a las futuras pruebas, y no teníamos demasiado claro cómo implementar la segunda solución, de manera que la tercera fue la opción natural: [Enlace al hilo 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=_463447_1).

La siguiente duda también surge sobre el mismo requisito, aunque esta vez está relacionada con la publicación de tramos de vuelo. Surgió un post en el foro en el que se indicó que para poder asignar un avión a un tramo es necesario que no esté en uso en algún otro tramo en el mismo intervalo de tiempo. De tal manera, resulta claro que antes de publicar, se debe verificar que esto no ocurra. Únicamente se verifica en este momento ya que antes de ello no hay problema con que haya este tipo de inconsistencias, siempre y cuando esté en modo borrador. [Enlace al hilo en cuestión](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_89154_1&conf_id=_426211_1&forum_id=_253522_1&message_id=_465652_1&nav=discussion_board).

Por otra parte, cabe destacar que la validación de que los tramos de un mismo vuelo no estén solapados en el tiempo se realizará en el servicio de publicación de los tramos, ya que, cómo se mencionó en anteriores documentos de análisis, hacerlo de esta manera permite evitar la situación en la que un vuelo no se puede publicar ya que varias de sus legs ya publicadas están solapadas, y al estar publicadas, no se podría corregir el error. En concreto, la validación verifica que el leg a publicar no se solape con ningún leg del vuelo que ya esté publicado. De esta manera, no incumplimos ningún requisito, a la par que ofrecemos la mejor experiencia posible al usuario. Si bien no hay hilo en cuestión sobre este enfoque, esto se validó durante las sesiones de follow up.

### **Conclusiones**:

En conclusión, en esta entega surgieron ciertos problemas de complicada interpretación, pero gracias al foro y a las sesiones de seguimiento, se tomaron decisiones que llevaron a implementar el los servicios sin incumplir ningún requisito, asegurando la calidad de la aplicación para el usuario final.

### **Bibliografía**:

Intencionalmente en blanco.