

ACME AirNav Solutions



ANALYSIS REPORT - Student 1 D03

Grupo: C1.050

Miembros: Cristina Fernández Chica (criferchi@alum.us.es), Ángel Amo Sánchez (angamosan@alum.us.es), Candela Jazmín Gutiérrez González (cangutgon@alum.us.es), Marta Aguilar Morcillo (maragumor@alum.us.es) y Luis Emmanuel Chávez Malavé (luichamal@alum.us.es)

Repositorio: <https://github.com/Cristinafernandezchica/Acme-ANS>

Planning dashboard: <https://github.com/users/Cristinafernandezchica/projects/1/views/1>

Sevilla 18 mayo, 2025

TABLA DE CONTENIDOS

Resumen Ejecutivo	3
Tabla de Revisiones	3
Introducción	4
Conflictos	4
Conflicto 1: Borrado de Tramos Publicados	4
Conflicto 2: Validaciones a nivel de Entidad o de Servicio	5
Conflicto 3: Update del estado de un tramo tras su publicación	6
Conflicto 4: No permitir que un tramo despegue y aterrice en un mismo aeropuerto	7
Conclusión	7
Bibliografía	7

Resumen Ejecutivo

En este documento se identifican aquellos aspectos que han generado problemas a lo largo de la realización del proyecto. Se analizarán los conflictos, aportando una breve descripción, las distintas alternativas evaluadas, los consejos e instrucciones ofrecidas por nuestro tutor o cualquier otro responsable de la asignatura durante las revisiones o a través del foro de la asignatura. Además, se indicará la alternativa finalmente escogida.

Tabla de Revisiones

Número de revisión	Fecha	Descripción de revisión	Autor
1.0	25/05/2025	Primera versión del documento. Se añaden los distintos conflictos surgidos durante la realización de las tareas, su análisis, y la solución escogida debidamente justificada.	Cristina Fernández Chica
2.0	26/05/2025	Finalización del documento.	Cristina Fernández Chica

Introducción

Para la documentación de todos aquellos conflictos surgidos durante la realización del proyecto se realizará la descripción de los mismos, proporcionando las alternativas que se barajaron para solucionarlo, añadiendo las recomendaciones proporcionadas por los profesores tutores. Por último, se justificará la alternativa escogida.

Conflictos

Conflicto 1: Borrado de Tramos Publicados

Descripción del conflicto

Implementando el borrado de Flight surgió la duda al borrar un vuelo, no quedaba claro si se debería borrar también sus tramos a pesar de que estos estuviesen publicados. Esta duda surgió porque en los requisitos dice que no se pueden borrar tramos publicados, pero analizando la situación no encontraba sentido a borrar únicamente el vuelo y dejar los tramos sin ningún vuelo asociado, convirtiéndose en tramos “fantasma” pues estarían en la base de datos, pero no aparecerían en ningún vuelo (cosa que no es posible), y nunca llegarían a llevarse a cabo.

Esto se preguntó en clase al profesor para de esta forma aclarar las dudas, se propusieron las siguientes alternativas:

Alternativas de solución

1. Borrar el vuelo pero no borrar sus tramos asociados si están publicados.

Pros:

- Cumpliría estrictamente con lo descrito en los requisitos proporcionados, eliminando un vuelo no publicado, pero no los tramos publicados que estén asociados al mismo.

Contras:

- La relación Leg - Flight se vuelve opcional, cosa que no tiene sentido para la aplicación que estamos creando, los tramos deben estar asociados siempre a un vuelo.

2. Eliminar tanto el vuelo como sus tramos independientemente de si estos están publicados o no.

Pros:

- Evitamos esa inconsistencia en la relación Leg - Flight, teniendo siempre los tramos relacionados a un vuelo.
- Es una opción que para el sistema que estamos implementando es la más lógica, para qué queremos los tramos de un vuelo que queremos borrar, si lo queremos borrar es porque no nos sirve en su totalidad, si nos sirviese lo actualizaríamos.

Contras:

- Incumple a medias el requisito de que no se puede borrar un tramo que ya está publicado, pues si solo se quiere borrar el tramo no será posible, pero si se quiere borrar el vuelo al completo se llevará a cabo.

3. No permitir borrar un vuelo con tramos publicados.

Pros:

- De nuevo se evita la inconsistencia en la relación Leg - Flight, los tramos siempre estarán relacionados a un vuelo.

Contras:

- Es una opción que provocaría que quedasen vuelos inútiles en la base de datos, pues si los queremos borrar pero no podemos por haber publicado alguna leg nunca llegaríamos a publicar el vuelo y sería información inútil almacenada.

Justificación de la solución

Tras valorar las 3 soluciones propuestas, junto a nuestro profesor tutor y todos los Students 1 presentes en la sesión de seguimiento, llegamos a la conclusión de que la mejor opción era borrar los tramos junto al vuelo independientemente de si están publicados o no, ya que de esta manera no quedan tramos sueltos en la base de datos ni se producen incongruencias en la relación Leg - Flight.

Conflicto 2: Validaciones a nivel de Entidad o de Servicio

Descripción del conflicto

A la hora de implementar las validaciones de diferentes acciones, el equipo no estaba seguro de si debían de implementarse a nivel de entidad, a nivel de servicio o en ambos. En una de las sesiones de seguimiento se comentó con el profesor tutor, al que se le propusieron las siguientes alternativas.

Alternativas de solución

1. Implementar todas las validaciones a nivel de entidad.

Pros:

- Se asegura que los datos sean consistentes a nivel de base de datos.

Contras:

Para las validaciones asociadas a una acción (reglas de negocio que dependen de acciones del usuario) no es lo ideal.

- Puede ser complicado manejar validaciones que dependan de múltiples entidades.

2. Implementar las validaciones a ambos niveles.

Pros:

- Asegura la integridad de los datos en la base de datos a nivel de entidad y también cubre reglas de negocio.

Contras:

- Puede implicar mayor complejidad de implementación y mantenimiento.
- Se deben coordinar las validaciones de cada nivel para evitar redundancias.

3. Implementar validaciones de manera mixta.

Pros:

- Aporta flexibilidad, pudiendo dividir en aquellas que hace referencia al formato de los datos y las que dependen de una acción.

- Evita validaciones innecesarias a nivel de entidad que solo son relevantes en determinados escenarios.
- Se reduce la posibilidad de redundancia

Contras:

- Se deben planificar correctamente qué validaciones van en cada lado para evitar inconsistencias.
- Es posible que no se pueda evitar la duplicación de código.

Justificación de la solución

Tras consultarlo con el profesor tutor, nos dijo que lo mejor era la implementación mixta, teniendo siempre mucho cuidado con generar inconsistencias en la base de datos por no haber validado. Por tanto, algunas restricciones se han comprobado a nivel de entidad, como la unicidad del flightNumber, y otras a nivel de servicio (la mayoría) según fuese conveniente para nosotros.

Conflicto 3: Update del estado de un tramo tras su publicación

Descripción del conflicto

Implementando la actualización de los tramos, surgió una duda en cuanto a la actualización de ciertos atributos, independientemente de si el tramos estaba publicado. Esto surgió a raíz de que una de mis compañeras usaba el estado de los tramos para hacer determinada funcionalidad, el problema es que los tramos solo se pueden publicar con el estado ON_TIME, pues no tenía sentido que se pudiesen publicar con el resto de estados si ni siquiera han sucedido esos vuelos. Esto hacía que su funcionalidad no tuviera demasiado sentido, entonces se propuso al profesor que se pudiese actualizar el estado de los tramos de un vuelo una vez estuviese publicado y en vuelo.

Alternativas de solución

1. Permitir la actualización del estado de los tramos tras ser publicados.

Pros:

- Da sentido tanto a la funcionalidad de mi compañera como al sistema en sí, pues no tiene sentido que un tramos se cree como CANCELLED, por ejemplo.

Contras:

- No se detectaron aspectos negativos más allá de una implementación más compleja.

2. No permitir la actualización del estado de los tramos tras ser publicados.

Pros:

- Mayor simplicidad de implementación del servicio de actualización.

Contras:

- Hace que clasificar por estado sea complicado y no tenga demasiado sentido, quitándole el sentido a la funcionalidad de mi compañera.
- No estaría en concordancia con la realidad, un vuelo no puede cancelarse, por ejemplo, antes de ser publicado.

Justificación de la solución

Finalmente, se optó por permitir la actualización del estado de los tramos tras ser publicado el vuelo, pues era la única solución con un sentido real, tanto para el profesor tutor como para nosotros.

Conflicto 4: No permitir que un tramo despegue y aterrice en un mismo aeropuerto

Descripción del conflicto

A varios compañeros de clase nos surgió la duda sobre si tenía sentido que un avión despegue y aterrice en un mismo aeropuerto.

Alternativas de solución

1. Permitir el aterrizaje y despegue en un mismo aeropuerto.

Pros:

- Facilita la implementación.

Contras:

- No tiene un sentido demasiado lógico, ya que es como si el avión diera la vuelta.

2. No permitir el aterrizaje y despegue en un mismo aeropuerto.

Pros:

- Tiene sentido dentro de la aplicación.

Contras:

- No se detectan aspectos en contra más allá de hacer la implementación mínimamente más compleja.

Justificación de la solución

Tras proponerle la restricción al profesor tutor en uno de los follow ups nos dijo que la podíamos implementar si lo considerábamos, pero que no era estrictamente necesario. Finalmente, tomé la decisión de incluirla ya que consideraba que tenía sentido dentro del sistema que estamos implementando.

Conclusión

Se ha documentado la gestión de conflictos durante la realización del proyecto, proporcionando diversas alternativas, y justificando debidamente la alternativa escogida finalmente.

Bibliografía

Intencionalmente en blanco.