Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Diseño y Pruebas II**

**Analysis Report D03**



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

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Repositorio** |
| 1/05/2025 | 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:** 01/05/2025

**Tabla de contenidos**

[**1.** **Resumen Ejecutivo**: 2](#_Toc192792692)

[**2.** **Tabla de revisión**: 3](#_Toc192792693)

[**3.** **Introducción**: 4](#_Toc192792694)

[**4.** **Contenidos**: 5](#_Toc192792695)

[**4.1** **Registro de análisis**: 5](#_Toc192792696)

[**5.** **Conclusiones**: 7](#_Toc192792697)

[**6.** **Bibliografía**: 8](#_Toc192792698)

### **Resumen Ejecutivo**:

Este informe documenta el análisis de los requisitos del proyecto. Los tres primeros requisitos no representaron un problema en una primera versión. Sin embargo, a medida que avanzaba la teoría de la asignatura, decidimos realizar algunos cambios en las restricciones de los atributos. Esto nos llevó a crear validadores personalizados, lo que requirió cierta investigación. Por otro lado, el requisito de pruebas fue muy similar al implementado en *AcmeJobs*.

Este informe documenta las decisiones tomadas en relación con casos límite y ambigüedades detectadas durante la implementación del proyecto. En este entregable, nos centramos en interpretar con precisión ciertos requisitos poco específicos, especialmente aquellos relacionados con la lógica de negocio del seguimiento (*tracking logs*). Ante la falta de claridad, se analizaron varias opciones para cada situación y, en algunos casos, se consultó el foro de la asignatura.

Se tomaron decisiones clave como permitir la creación de múltiples borradores de *tracking logs* tras uno publicado con 100%, o mantener la resolución como opcional mientras el estado sea *pending*, priorizando siempre la flexibilidad sin comprometer la coherencia del sistema. También se ajustaron criterios sobre cómo deben mantenerse los estados entre distintos *trackings logs* relacionados.

Este proceso nos ayudó a reforzar el diseño del modelo y a anticipar posibles problemas de validación y usabilidad. Las decisiones fueron razonadas, justificadas y orientadas a ofrecer un comportamiento predecible y consistente para el usuario.

### **Tabla de revisión**:

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 1/05/2025 | 1.0 | Creación del documento y primeros cambios |
|  |  |  |
|  |  |  |
|  |  |  |

### **Introducción**:

Este informe detalla el análisis de los requisitos del proyecto. El primer requisito no especificaba con claridad si el patrón requería un validador personalizado o si bastaba con utilizarlo directamente. Además, en el siguiente requisito, relacionado con los datos de los *claims*, no quedaba claro si estos debían vincularse con los *legs* o con los *flights*.

El documento está estructurado de manera que el contenido principal es el registro de análisis, el cual comienza con la descripción del requisito, seguido de las opciones consideradas y la decisión tomada.

Este informe detalla el análisis de varios aspectos del comportamiento del sistema que no estaban completamente definidos en los requisitos. En particular, surgieron dudas sobre la gestión de los *trackings logs*, como la posibilidad de crear múltiples borradores tras uno publicado con resolución del 100%, la obligatoriedad del campo de resolución en estado *pending*, o la coherencia del estado en registros excepcionales.

El documento está estructurado de forma que cada entrada del registro de análisis comienza con la descripción del problema detectado, seguida de las opciones consideradas y finalizando con la decisión tomada y su justificación.

### **Contenidos**:

### **Registro de análisis**:

La tarea analizada será la octava obligatoria la cual dice así:

Imagen de la pantalla de un celular de un mensaje en letras blancas

El contenido generado por IA puede ser incorrecto.

Imagen 1 – Octavo requisito obligatorio

Se investigó el significado de la lista de “undergoings”, que hacía referencia a la necesidad de crear una lista para representar los elementos en estado *pending* y otra para los estados restantes (*accepted* o *rejected*). A partir de esto, se valoraron dos opciones:

* **Opción 1:** Crear dos listas separadas
  + **Pros:** Mayor orden.
  + **Contras:** Podría resultar innecesario
* **Opción 2:** Crear una única lista con todos los tipos
  + **Pros:** Mayor simplicidad.
  + **Contras:** Podría resultar más abrumador para el cliente al mostrar todo junto

Tras revisar el foro, encontré un mensaje del profesor Rafael Corchuelo aprobando una opción similar a la segunda. Por ello, decidí implementar la opción 2.

[**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=_464910_1)

La tarea analizada será la novena obligatoria la cual dice así:

Texto

El contenido generado por IA puede ser incorrecto.

Imagen 2 – Noveno requisito obligatorio

Surgió la duda sobre si la resolución debe estar vacía cuando el estado es *pending*:

Las opciones consideradas fueron:

* **Opción 1:** Obligar a que la resolución esté vacía mientras el estado sea *pending*
  + **Pros:** Se asegura que no haya resoluciones visibles antes de una decisión final.
  + **Contras:** Restringe la flexibilidad del sistema.
  + No permite guardar comentarios o resoluciones preliminares.
* **Opción 2:** Permitir que la resolución sea opcional mientras el estado sea *pending*.
  + **Pros:** Aumenta la flexibilidad en el manejo de datos.
  + Permite registrar comentarios, notas internas o resoluciones.
  + **Contras:** Ninguna.

Gracias al mensaje consultado a continuación, decidí optar por la opción 2, al considerar que aporta una mayor utilidad sin comprometer la coherencia del sistema.

[**Enlace**](https://ev.us.es/webapps/discussionboard/do/message?conf_id=_426211_1&forum_id=_253522_1&course_id=_89154_1&action=list_messages&nav=discussion_board&message_id=_461426_1)

Surgió la duda sobre si el estado de un *tracking log* excepcional puede ser diferente al del primer *tracking log* que tenga un porcentaje de resolución del 100% y publicado.

Las opciones consideradas fueron:

* **Opción 1:** Permitir que el estado cambie.
  + **Pros:** Ofrece cierta flexibilidad para corregir errores en el seguimiento.
  + **Contras:** Puede generar incoherencias entre los distintos *trackings logs* de un mismo proceso.
* **Opción 2:** Mantener el mismo estado que el primer *tracking log* con 100% de resolución.
  + **Pros:** Asegura coherencia en el historial del proceso.
  + Reduce la posibilidad de errores o ambigüedades en los registros.
  + **Contras:** Menor flexibilidad ante posibles correcciones.

En una primera instancia se optó por la opción 1, pero tras consultar el foro y revisar el criterio expuesto, decidí cambiar a la opción 2, al considerarla más robusta y coherente con la lógica del sistema.

[**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=_465569_1)

Surgió la duda sobre si, tras publicar un *tracking log* con porcentaje de resolución del 100%, se podrían crear más *tracking logs*. El requisito menciona la existencia de un único *tracking log*, pero no especifica si esto aplica únicamente a los publicados o también a los que están en modo borrador. Tras consultar el foro y no encontrar una respuesta concluyente, se valoraron las siguientes opciones:

Las opciones consideradas fueron:

* **Opción 1:** Solo se puede crear un único *tracking log* (ya sea en modo borrador o publicado)
  + **Pros:** Asegura un control estricto del flujo de creación y publicación.
  + **Contras:** No permite guardar versiones borrador si ya hay uno creado.
* **Opción 2:** Se pueden crear varios *trackings logs* en borrador tras el último publicado con 100%, pero solo se podrá publicar uno de ellos.
  + **Pros:** El usuario puede preparar varias versiones antes de decidir cuál publicar
  + Permite hacer cambios sin tener que editar borradores*.*
  + **Contras:** Ninguna.

Después de valorar ambas opciones, decidí quedarme con la opción 2 porque es más flexible para el usuario y sigue cumpliendo con el requisito de que solo haya un *tracking log* publicado después del último con 100%.

### **Conclusiones**:

En conclusión, este segundo entregable no presentó muchas dificultades, ya que, de las cuatro tareas, solo dos requirieron una breve investigación en el foro. Tras evaluar las opciones, las respuestas a escoger eran bastante claras. Además, los archivos de datos no representaron ningún inconveniente.

En conclusión, este tercer entregable requirió un análisis cuidadoso debido a la ambigüedad de varios requisitos. La mayoría de las decisiones se tomaron basándose en la información y criterios aportados en el foro de la asignatura. Gracias a ello, pudimos definir soluciones coherentes y flexibles que permiten un correcto funcionamiento del sistema, asegurando al mismo tiempo la usabilidad y la consistencia en el manejo de los *trackings logs* y estados.

### **Bibliografía**:

Intencionalmente en blanco.