Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Analysis report D03**



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas 2.

Curso 2024 – 2025

|  |  |
| --- | --- |
| **Fecha** | **Versión** |
| 03/07/2025 | V2.0.0 |

|  |  |  |
| --- | --- | --- |
| **Grupo de prácticas: C2.005** | | |
| **Autores por orden alfabético** | **Rol** | **Descripción del rol** |
| González Benito, Claudio – clagonben@alum.us.es | Project Manager  Developer  Tester | Persona encargada de tomar decisiones de diseño y vigilar el correcto desarrollo  Persona encargada de desarrollar el código.  Persona encargada de realizar pruebas sobre el código. |
| Ramos Vargas, Alba – albramvar1@alum.us.es | Operador Developer  Tester | Encargado de las tareas de campo, de las instalaciones y del mantenimiento de los sistemas de la empresa.  Persona encargada de desarrollar el código.  Persona encargada de realizar pruebas sobre el código. |

**Control de Versiones**

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 14/02/2025 | v1.0.0 | Desarrollo de la primera versión. |
| 03/07/2025 | V2.0.0 | Actualización de la convocatoria 2 |
|  |  |  |
|  |  |  |

Repositorio: <https://github.com/Manuelgithuv/Acme-ANS-C2>

**Índice de contenido**

Contenido

[Resumen ejecutivo 3](#_Toc202429045)

[**Introducción** 4](#_Toc202429046)

[Registro de Análisis para los Requisitos 5](#_Toc202429047)

[Conclusiones 11](#_Toc202429048)

[Bibliografía 12](#_Toc202429049)

## Resumen ejecutivo

El objetivo de este documento es cumplir con el criterio extra establecido para el alumno 4 en los objetivos individuales de los alumnos en concreto los del D03. El objetivo de esta entrega es implementar los requisitos funcionales usando interfaces para poder hacer las operaciones que se definen en los requisitos

## Introducción

Se redactan en este documento los requisitos que el estudiante 4 no ha tenido del todo claros para los requisitos relacionados con el entregable D03, para ello se ha puesto el requisito, las dudas o problemas que genera, las conclusiones del análisis, decisión tomada y verificación con contenido del foro.

## Registro de Análisis para los Requisitos

**Requisito:**

**D3-R9) Operations by assistance agents on tracking logs:**

**• List and show the tracking logs associated to their claims.**

**• Create, update, publish, and delete a tracking log. A tracking log cannot be published until its corresponding claim is published. Once published, tracking logs cannot be updated or deleted. In exceptional cases, a new tracking log may be created even after the last one has been published (the one with a 100% resolution percentage). This additional tracking log is generated when the customer expresses dissatisfaction, prompting agents to review their claims.**

**Problemas Identificados:**

Tras analizar este requisito y darse cuenta de la necesidad tener que ordenar los logs de una claim para poder validar que el porcentaje de un nuevo log es igual o superior al porcentaje de uno nuevo, surge la siguiente duda **¿Cómo deberían ordenarse los tracking logs para poder saber cual es el ultimo?**

**Alternativa 1: Ordenarlos por lastUpdateMoment**

* **Ventajas: es un valor que ya existe, y ordenar por el resultaría fácil**
* **Contras: es un valor que podría cambiar o ser editado, cambiando el orden, lo cual puede complicar la lógica de algunas restricciones**

**Alternativa 2: Crear un nuevo valor y usarlo crear un orden fijo que no sea editable**

* **Ventajas: Al tener un orden fijo, es fácil de sacar el orden. Como ese orden nunca va a cambiar, se podría fácilmente implementar restricciones**
* **Contras: requeriría pensar un nuevo valor para poder obtener dicho orden**

**Conclusiones del Análisis:**

**Tras un análisis de las alternativas, concluyo que la mejor alternativa es la segunda, ya que aunque hace falta crear un valor nuevo, las restricciones mas complejas se facilitan teniendo un orden fijo. Tras realizar esta análisis, se ha verificado con una respuesta en el foro que es valido [1].**

**Decisión:**

Entonces, se ha añadido el atributo creationMoment, que cuando se crea una entidad por interfaz se fija usando el reloj de la aplicación. (La misma lógica se ha implementado para registrationMoment en claim), para lastUpdateMoment se ha hecho que se actualice cuando se cambia la entidad. Entonces, se ha creado un método de repositorio que devuelve los logs asociados a una claim ordenados por el creationMoment mas reciente. En caso de empate (que en producción seria mas raro pero podría ocurrir), el segundo criterio de ordenación serían las id de mayor a menor, ya que estas son inmutables cuando se crea un objeto y se van generando de forma ascendiente. Entonces, con este metodo de repositorio, se implementa en claim el método transient getLastTrackingLog(), el cual saca esta lista ordenada, y si no esta vacía devuelve su primer elemento (el ultimo tracking log para esa claim). Usando estos dos recursos se han implementado las siguientes restricciones sobre las operaciones:

* Create: Un tracking log siempre tendría que tener un porcentaje mayor o igual al del tracking log anterior, si tiene uno
* Update y publish: Si un tracking log tiene un log anterior, el nuevo porcentaje actualizado deberá ser mayor igual al de ese log anterior. Si un tracking log tiene un log posterior, el nuevo porcentaje actualizado deberá ser menor igual al de ese log posterior.

Con estas restricciones se espera evitar contrasentidos y cumplir los requisitos obligatorios

**Validación:**

**[1]Respuesta del foro: "... A2: “Añadir un nuevo atributo creationMoment para reflejar la fecha de creación real, y usarlo para validar el orden de los porcentajes.” Es una de las alternativas que se han comentado en los mensajes anteriores para resolver este problema; si tiene Ud. que mantener un orden en un conjunto de objetos, en este caso parece que dicho orden podría implementarse fácilmente usando el momento de creación de los objetos. ... Por lo tanto, entre las tres alternativas que Ud. valora, la segunda puede ser adecuada..."**

**Enlace de la discusión:** [**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=\_471048\_1**](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=_471048_1)

**Requisito:**

**D3-R9) Operations by assistance agents on tracking logs:**

**• List and show the tracking logs associated to their claims.**

**• Create, update, publish, and delete a tracking log. A tracking log cannot be published until its corresponding claim is published. Once published, tracking logs cannot be updated or deleted. In exceptional cases, a new tracking log may be created even after the last one has been published (the one with a 100% resolution percentage). This additional tracking log is generated when the customer expresses dissatisfaction, prompting agents to review their claims.**

**Problemas Identificados:**

Tras leer el siguiente requisito, surgen la siguientes dudas sobre El trackin log adicional que se menciona que se puede crear después de que una claim publicada tenga un tracking log publicado al 100%:

**¿Qué valores son validos para este tracking log?**

**Alternativa 1: Son validos cualquier estado que sea accepted o rejected, ya que como el porcentaje debe de ser creciente y ya es 100, no podrá ser pending**

* **Ventajas: Habría que implementar menos restricciones**
* **Contras: El estado final de una claim publicada con su ultimo tracking log publicado podría verse alterado**

**Alternativa 2: Solo es valido que tenga el mismo estado que el log anterior que esta publicado**

* **Ventajas: Se mantiene el mismo estado en la claim publicada**
* **Contra: habría que implementar una restricción mas**

**Conclusiones del Análisis:**

Tras el análisis de las alternativas, se concluye que, bajo las condiciones del requisito, solo se podrá crear un único tracking log, que tendrá que tener porcentaje 100 y como se indica en la alternativa 2 tener el mismo estado que el log publicado anterior con el 100%.

**Decisión:**

Se han implementado estas restricciones en el validate del create de tracking logs.

**Validación:**

**"...Pregunta Ud. si “puede bajar el porcentaje de resolución de este nuevo Tracking Log con respecto al anterior”. La respuesta es no; el porcentaje de resolución indica el porcentaje de trabajo realizado para resolver una reclamación; cuando se completa debe estar al 100%; cómo se comenta, ese registro de seguimiento adicional simplemente sirve para recoger que el cliente no está satisfecho con la resolución de su reclamación, nada más. Por lo tanto, dicho porcentaje tiene que estar al 100%.**

**También pregunta Ud. que “[cómo] tratamos el estado de este nuevo Tracking Log”. El estado de la reclamación no cambia; si el último registro de seguimiento concluyo que era necesario aceptar la reclamación o rechazarla, ese último registro excepcional debe mantener dicho estado. Si el cliente decide abrir una nueva reclamación, será un proceso de reclamación nuevo. ..."**

**Enlace de la discusión:** [**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**](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)

**Requisito:**

**D03-R8) Operations by assistance agents on claims:**

**• List their completed claims, that is, the ones that have been accepted or rejected and show their details.**

**• List the undergoing claims and show their details including the leg to which they are linked.**

**• Create, update, publish, and delete their claims. Claims must be linked to legs that occurred. Additionally, claims can be updated or deleted as long as they have not been published.**

**Problemas Identificados:**

Tras leer dicho requisito, surge la siguiente duda: **¿debe una claim ser registrada después de que ocurra una leg o puede ser registrada antes?**

**Alternativa 1: Si puede ser registrada antes**

* **Ventajas: No hay que implementar ninguna validación extra**
* **Contras: Un cliente podría poner una reclamación de una leg que aun no ha ocurrido, lo cual puede no tener sentido**

**Alternativa 2: No puede ser registrada antes**

* **Ventajas: Habría un proceso mas lógico y definido. Una reclamación solo se podría registrar sobre una claim que ya ha ocurrido**
* **Contras: Hay que implementar ninguna validación extra**

**Conclusiones del Análisis:**

**Tras un análisis me decanto por la alternativa 2, ya que es un proceso mas estructurado y la implementación seria simplemente comprobar que el momento de registro de la claim es igual o posterior al momento de llegada. Se ha verificado este análisis con la siguiente respuesta de un profesor en el foro**

**Decisión:**

**Implementar dicha restricción sobre la temporalidad de las claims**

**Validación:**

**"... Su segunda alternativa es “Implementar una validación que impida registrar claims sobre legs futuros”. Correcto, hemos llegado a esa conclusión. ..."**

**Enlace de la discusión:** [**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=\_465589\_1**](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=_465589_1)

## Conclusiones

Se tuvieron varias dudas sobre como seria el proceso de registro y de creación de logs, así como sus distintas restricciones de como debía hacerse correctamente. Se plantearon estas dudas, se analizaron y se verifico el análisis con post ya respondidos por los profesores y se ha explicado las acciones que se han seguido

## Bibliografía

Intencionalmente en blanco.