

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

ANALYSIS REPORT D02



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas II

Curso 2024 – 2025

Grupo C1.021		
Autor	Correo	Rol
Paula Sánchez Gómez	pausangom1@alum.us.es	Desarrollador/Tester/Manager

Repositorio: <https://github.com/DP2-C1-021/Acme-ANS>

Control de versiones

Versión	Fecha	Descripción
v1.0	13/03/2025	Primera versión del documento

Índice

1. Introducción	3
2. Requisitos Funcionales	3
3. Conclusión	3

1. Introducción

Este documento ha sido elaborado con el propósito de clarificar aquellos requisitos que, pese a haber pasado por un proceso de especificación y desarrollo, aún presentan ciertas ambigüedades que podrían generar inconvenientes durante su implementación.

El objetivo principal de este análisis es identificar las razones que han llevado a la revisión de estos requisitos, así como presentar las conclusiones obtenidas a lo largo del análisis. Estas conclusiones servirán para precisar los requisitos de manera más clara. Además, se incluirán soluciones propuestas para abordar las ambigüedades detectadas y, finalmente, se proporcionará un enlace a la validación realizada por un profesor.

La estructura de este documento estará conformada por un título que hará referencia al requisito analizado, seguido de un apartado con las conclusiones del análisis y, posteriormente, se presentarán las propuestas de mejora más adecuadas. Al finalizar, se incluirá el enlace a la validación correspondiente.

2. Análisis requisitos individuales

Requisito 3:

- El requisito establece lo siguiente: *"The **assistance agents** are the people responsible for recording and managing post-flight incidents reported by passengers. The system must store the following data about them: an **employee code** (unique, pattern "[A-Z]{2-3}\d{6}\$", where the first two or three letters correspond to their initials), a list of **spoken languages** (no longer than 255 characters), the **airline** for which they work, the **moment** on which they began to work for that airline (in the past), and optionally, a **brief bio** (up to 255 characters), their **salary**, and a link to a **photo** that should be stored anywhere else."*

Duda: En relación con el campo *spokenLanguages*, surgió la cuestión de cómo implementar una lista de idiomas hablados. Debido a la falta de una especificación detallada sobre la gestión de esta lista, se planteó la duda de si debía usarse una lista como tal o si era posible almacenarla como una cadena de texto.

Conclusión: Tras analizar la discusión en el foro, se determinó que la mejor opción es utilizar una cadena de texto con un delimitador específico para separar los idiomas. Este enfoque permite a los usuarios ingresar los idiomas hablados de manera libre, empleando un delimitador como la coma o el punto y coma. Dado que no se requiere un procesamiento complejo de la lista, esta solución resulta adecuada y se ajusta al requisito sin generar complicaciones adicionales. Esta decisión fue validada en el foro: https://ev.us.es/webapps/discussionboard/do/message?conf_id=426211&forum_id=253522&course_id=89154&action=list_messages&nav=discussion_board&message_id=460593_1.

Requisito 4:

- “A **claim** is a formal request or complain made by a passenger or customer due to a problem or inconvenience experienced during a flight. They are registered by the **assistance agents**, and the data to store when registering a **claim** is the following: the **registration moment** (in the past), the **passenger email**, a **description** (up to 255 characters), a **type** (“FLIGHT ISSUES”, “LUGGAGE ISSUES”, “SECURITY INCIDENT”, “OTHER ISSUES”) and an **indicator** of whether the claim is accepted or not. “

Duda: Surgió la duda sobre cómo debe manejarse la relación entre los claims y los assistant agents. En concreto, la pregunta era si los pasajeros pueden registrar directamente sus claims en el sistema o si, por el contrario, deben comunicárselo a un assistant agent para que sea este quien lo registre formalmente. El problema surge porque el requisito establece que los reclamos deben ser registrados por los assistant agents, pero no aclara si el pasajero tiene la posibilidad de ingresarlo directamente en la plataforma.

Conclusión: Tras revisar las discusiones en el foro, se concluyó que los *claims* solo pueden ser registrados por los *assistant agents*. Es decir, los pasajeros no tienen acceso directo al sistema para registrar sus reclamos, sino que deben informar su problema a un agente de asistencia, quien posteriormente se encargará de formalizar el reclamo en la plataforma.

Esta decisión se tomó con base en las discusiones en el foro, en el siguiente enlace: https://ev.us.es/webapps/discussionboard/do/message?conf_id=426211_1&forum_id=25352_2_1&course_id=89154_1&action=list_messages&nav=discussion_board&message_id=461513_1

Requisito 5:

- “Claims need to be tracked through **tracking logs**. A tracking log records each step in the procedure followed to resolve or reject a claim, ensuring that all actions and decisions are documented. The system must store the following data about **tracking logs**: the **last update moment**, the **step** undergoing (up to 50 characters), a **resolution percentage**, and an **indicator** on whether the claim was finally accepted or not. When a claim is accepted or rejected, the system must store its **resolution** indicating the reason why was rejected or the compensation to offer (up to 255 characters).”

Duda: Surgió una confusión respecto a cómo estructurar el campo indicator. En la entidad Claim, el campo indicator ya había sido definido como un valor booleano (true/false), lo que representa directamente si el reclamo fue aceptado o rechazado. Sin embargo, en la especificación del requisito se mencionaba la posibilidad de utilizar un enum con valores "ACCEPTED" y "REJECTED". La duda principal era si realmente tenía sentido implementar un enum en este caso, considerando que "ACCEPTED" y "REJECTED" son equivalentes a true y false. La inclusión de "PENDING" parecía ser la única justificación para usar un enum, ya que un booleano solo permite dos estados y "PENDING" representaría un tercer estado posible.

Conclusión: Se decidió mantenerlo como un booleano (true/false), ya que efectivamente "ACCEPTED" y "REJECTED" son equivalentes a true y false. Sin embargo, si en el futuro se considera necesario un estado intermedio como "PENDING", entonces sí podría justificarse cambiarlo a un enum. Por ahora, al no ser un requisito obligatorio manejar un estado pendiente, el booleano es la opción más simple y eficiente.

3. Conclusión

El análisis realizado en este documento ha permitido esclarecer varias ambigüedades presentes en la especificación de los requisitos, asegurando una interpretación más precisa para su correcta implementación. A lo largo del estudio, se han identificado y resuelto diversas dudas sobre la gestión de ciertos campos clave en el sistema, lo que ha llevado a la adopción de soluciones más eficientes y alineadas con los objetivos del proyecto.

En primer lugar, en relación con el campo spokenLanguages, se determinó que la mejor forma de almacenar los idiomas hablados es mediante una cadena de texto con un delimitador específico, en lugar de una lista estructurada. Esta solución facilita la entrada de datos por parte de los usuarios sin añadir complejidad innecesaria al procesamiento de la información.

Por otro lado, se abordó la gestión de los claims en el sistema, aclarando que estos deben ser registrados exclusivamente por los assistant agents. Se confirmó que los pasajeros no tienen acceso directo al sistema para generar un claim, sino que deben notificar su problema a un agente, quien será el encargado de formalizar el registro en la plataforma.

Finalmente, se resolvió la duda sobre el campo indicator dentro de los claims. Inicialmente, se planteó la posibilidad de utilizar un enum para representar los valores "ACCEPTED", "PENDING" y "REJECTED", pero tras el análisis se concluyó que, dado que los estados "ACCEPTED" y "REJECTED" son equivalentes a true y false, un booleano es la opción más sencilla y eficiente. No obstante, si en el futuro fuera necesario incorporar un estado intermedio como "PENDING", se podría reevaluar la implementación del enum.