



INFORME DE ANÁLISIS GRUPAL

Grupo 1-C1.020 | <https://github.com/PDJ6975/Acme-ANS-D02-25.2.0>



Nombre	Correo Corporativo
Antonio Rodríguez Calderón	antrodcal@alum.us.es
Adrián Ramírez Gil	adrramgil@alum.us.es
Jianwu Hu	jiahu3@alum.us.es
Pablo Castrillón Mora	pabcasmor1@alum.us.es
Pablo Olivencia Moreno	pabolimor@alum.us.es

11 DE MARZO DE 2025
DISEÑO Y PRUEBAS II

Tabla de Contenido

1. Resumen Ejecutivo	2
2. Tabla de Revisión	2
3. Introducción	2
4. Registro de Análisis.....	3
4.1 Durante el primer entregable.....	3
4.1.1 Requisito 2: Enlace al Dashboard de Planificación en GitHub.....	3
4.1.2 Requisito 3: Versionado del Proyecto en el Nombre del Repositorio	3
4.1.3 Requisito 4: Configuración de Privacidad del Repositorio	4
4.2 Durante el segundo entregable	4
4.2.1 Respecto a todos los requisitos exigidos.....	4
4.2.2 Requisito 9: Generar datos iniciales	5
5. Conclusión.....	6
6. Bibliografía.....	6

1. Resumen Ejecutivo

Este informe de análisis tiene como objetivo evaluar los requisitos grupales establecidos en el proyecto, identificando posibles ambigüedades, inconsistencias y áreas de mejora. A través del análisis detallado de los requisitos más relevantes, se proponen correcciones y ajustes que permitirán una implementación más precisa y alineada con los objetivos del proyecto.

El análisis se presenta en secciones organizadas donde cada requisito identificado es evaluado, detallando las acciones necesarias para garantizar su correcta aplicación en el desarrollo del sistema.

2. Tabla de Revisión

Versión	Fecha	Descripción de los cambios
1.0	19/02/2025	Creación inicial del documento
2.0	11/03/2025	Ajustes para la segunda entrega

3. Introducción

Este informe de análisis es parte del proceso de validación y ajuste de los requisitos grupales del proyecto. Su propósito es garantizar que los requisitos documentados sean claros, factibles y correctamente implementados dentro del sistema en desarrollo.

El análisis de requisitos es una etapa fundamental para detectar posibles fallos en la interpretación de las especificaciones iniciales y prevenir problemas de implementación en etapas posteriores del proyecto. Para ello, se examinan los requisitos clave con el fin de identificar puntos críticos que requieran ajustes, aclaraciones o validaciones adicionales.

El presente documento está estructurado de manera que cada sección cubra un requisito específico, proporcionando su descripción, análisis detallado y decisiones propuestas para su correcta implementación. Se espera que este informe sirva como referencia para futuras revisiones y ajustes en la planificación del proyecto.

4. Registro de Análisis.

En este apartado se presentan los requisitos que generaron dudas en el desarrollo del proyecto dividido por entregables, las distintas posibilidades consideradas y la decisión tomada en cada caso.

4.1 Durante el primer entregable

4.1.1 Requisito 2: Enlace al Dashboard de Planificación en GitHub.

Descripción:

“Proporcionar un enlace al dashboard de planificación en GitHub para revisar las tareas, su estado actual y el cronograma.”

Análisis:

Al leer el requisito surgió la duda sobre la mejor ubicación para incluir el enlace dentro del proyecto. Se consideraron las siguientes opciones:

- **Opción 1:** Colocar el enlace en el propio proyecto como un nuevo apartado dentro de la página principal.
- **Opción 2:** Incluir el enlace en el archivo **README** del repositorio.
- **Opción 3:** Adjuntar el enlace en la entrega de la **Enseñanza Virtual**.

Se consultó al profesor durante una práctica de laboratorio, quien confirmó que había múltiples formas de cumplir con este requisito.

Decisión:

Se decidió de forma grupal que una única persona añadiese el enlace al dashboard dentro del **README del proyecto**, asegurando que todos los miembros del equipo tuvieran acceso a la planificación en GitHub.

4.1.2 Requisito 3: Versionado del Proyecto en el Nombre del Repositorio

Descripción:

“El nombre del repositorio debe seguir un esquema de versionado que refleje la versión del entregable.”

Análisis:

Este requisito generó dudas sobre la estructura de nombres que debía seguir el repositorio. Se evaluaron las siguientes opciones:

- **Opción 1:** Usar un esquema como **Acme-ANS-D01-25.1.0** para reflejar la versión explícitamente. 3

- **Opción 2:** Usar solo **Acme-ANS-D01** y manejar el versionado dentro de los commits y releases.
- **Opción 3:** Incluir la fecha en el nombre, por ejemplo, **Acme-ANS-D01-2025**.

Decisión:

Se optó por la opción 1, tras una aclaración en el laboratorio con nuestro profesor, se comunicó que el nombre del repositorio debía incluir un identificador de versión claro y estructurado como **Acme-ANS-D01-25.1.0**, facilitando su identificación y seguimiento.

4.1.3 Requisito 4: Configuración de Privacidad del Repositorio

Descripción:

“Definir si el repositorio debe ser público o privado durante el desarrollo del proyecto.”

Análisis:

Se plantearon dos enfoques principales sobre la privacidad del repositorio:

- **Opción 1:** Mantener el repositorio público para garantizar transparencia y facilitar el acceso.
- **Opción 2:** Configurar el repositorio como privado para evitar plagios y accesos no autorizados.

Decisión:

Se decidió mantener el repositorio en modo privado durante el desarrollo del proyecto para evitar accesos externos no autorizados. Sin embargo, se acordó hacerlo público al finalizar el entregable, garantizando transparencia y acceso a la evaluación del código por parte de los revisores.

4.2 Durante el segundo entregable

4.2.1 Respecto a todos los requisitos exigidos

Análisis:

Una duda general grupal que hemos tenido en el proyecto ha sido con las relaciones exigidas tanto explícita como implícitamente en el proyecto. En la mayoría de requisitos de información se habla acerca de relaciones de una entidad con otra, sirva como ejemplo el requisito 8:

“A review is a short message posted by a user to share his or her experiences. The reviews may deal with aspects such as airport cleanliness, services (e.g., restaurants, customer service, or assistance), or experiences with airlines or

flights. The system must store the following data: a name or an alias of the person who posts it (up to 50 characters), the moment when the review is posted (in the past), a subject (up to 50 characters), a piece of text (up to 255 characters), and, optionally, a score (a double between 0 and 10), and whether the experience is recommended or not"

De este requisito se puede entender que una "review" puede ir dirigida a un aeropuerto, aerolínea, servicio o incluso vuelo. La pregunta aquí reside en si es necesario manejar esas relaciones.

Así, hemos considerado dos alternativas:

- **Opción 1: Incluir todas las relaciones que se pueden deducir de los requisitos sin importar que vayan a ser utilizadas o no.**

A simple vista, si bien parece una tarea más tediosa y menos eficiente, es cierto que sería lo mejor desde el punto de vista de la profesionalidad, puesto que así se abarcan todas las posibilidades y puede que a futuro necesitemos las relaciones.

- **Opción 2: Incluir solo aquellas relaciones necesarias en función de futuros requisitos eliminando con certeza aquellas que sepamos que no se van a utilizar.**

Esta solución nos permitiría tener un diagrama y entidades mucho más limpios, de forma que podríamos eliminar algunas relaciones que sepamos de seguro que no nos van a hacer falta, como ha sido el caso de las relaciones de la entidad "Review".

Conclusión: Nos hemos decantado por utilizar la idea de la segunda opción (puesto que el profesor ha dado a entender en el laboratorio que hay relaciones que no hay que poner si no es necesario), de manera que vamos a eliminar o a no tener en cuenta las relaciones de "Review" y el resto de relaciones que se pueden deducir de los requisitos grupales por ahora sí se van a dejar, de forma que si en el futuro vemos que verdaderamente no se utilizan, pues las eliminaremos. Eliminamos principalmente las de "Review" para no tener muchas relaciones opcionales "@ManyToOne", que van a empeorar la calidad del diagrama UML.

4.2.2 Requisito 9: Generar datos iniciales

Descripción:

"Produce initial data to cold start your project; it must include an administrator account with credentials "administrator/administrator". Produce sample data to test your requirements; it must include additional administrator accounts with credentials "administrator1/administrator1" and "administrator2/administrator2."

Análisis:

El problema con este requisito es que no teníamos claro si los datos deberían introducirse únicamente en el “sample” o si también deberíamos añadir algunos al “initial”.

Así, hemos considerado dos alternativas principales:

- **Opción 1:** añadir datos tanto al “initial” como al “sample”

En un principio parece la opción más viable ya que el initial sería representaría lo mínimo con lo que debe arrancar la aplicación, y añadir aunque sea una fila de datos para cada entidad podría estar bien.

- **Opción 2:** añadir datos únicamente al “sample”

Parece una idea fiel a la estructura seguida en Acme Job, donde solo se añaden datos para las entidades en el “sample”, mientras que el “initial” aparece por defecto como en nuestro proyecto.

Conclusión: Hemos decidido seguir la segunda opción tras preguntarle al profesor en la clase del follow-up y prefiriendo seguir la estructura de Acme Jobs.

5. Conclusión

Este informe ha identificado y analizado aspectos clave en los requisitos del Entregable D01. Se han propuesto soluciones para mejorar la claridad y aplicación de los requisitos en el desarrollo del proyecto.

En particular, se resolvieron dudas sobre la ubicación del enlace al dashboard de planificación, el esquema de versionado en el nombre del repositorio y la configuración de privacidad de este. La implementación de estas decisiones permitirá una gestión más eficiente del proyecto y garantizará el cumplimiento de los requisitos establecidos.

6. Bibliografía

Intencionalmente en blanco.