



# INFORME DE ANÁLISIS

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

20 DE FEBRERO DE 2025  
PABLO OLIVENCIA | STUDENT #2

## Tabla de Contenido

1. Resumen Ejecutivo .....	1
2. Tabla de Revisión .....	2
3. Introducción .....	2
4. Registros de Análisis.....	2
11. Conclusión .....	3
12. Bibliografía .....	5

## 1. Resumen Ejecutivo

Este informe detalla los análisis realizados sobre ciertos **requisitos individuales** del proyecto, abordando las **dudas** que surgieron durante su desarrollo, las **posibles soluciones** evaluadas y las **decisiones** finales adoptadas.

A través de este informe, se busca reflejar el proceso llevado a cabo para garantizar el cumplimiento adecuado de los requisitos, minimizando ambigüedades y asegurando la coherencia con la metodología de trabajo establecida.

## 2. Tabla de Revisión

Versión	Fecha	Descripción de los cambios
1.0	20/02/2025	Creación del documento
2.0	24/03/2025	Actualización D02

## 3. Introducción

Durante el desarrollo del proyecto, algunos requisitos individuales plantearon dudas sobre su correcta implementación y documentación. Este informe tiene como objetivo exponer dichas dudas, analizar las alternativas consideradas y justificar la elección final de la solución adoptada.

Cada análisis presentado sigue una estructura clara: primero, se expone el requisito específico que generó la duda; luego, se describen las opciones contempladas para abordarlo, analizando ventajas e inconvenientes; finalmente, se detalla la decisión tomada y su justificación (generalmente en base a la respuesta del profesor).

El análisis incluye aspectos como la ubicación adecuada de ciertos elementos dentro de la documentación del proyecto y la diferencia entre enfoques de informes individuales y grupales, cuestiones que fueron aclaradas a través de la consulta directa con el profesor en clase de laboratorio. Este documento busca proporcionar una visión clara del proceso de toma de decisiones.

## 4. Registros de Análisis

### 4.1 Entregable D01

Para este primer entregable, no se han presentado problemas que requieran un análisis de posibles soluciones. Se ha creado el documento como constancia de

ello y como plantilla para futuras iteraciones.

## 4.2 Entregable D02

### Decisión de Diseño 1: Modelado de BannedPassenger

#### Opción 1: Entidad BannedPassenger independiente

- **Definición:**  
Se crea una entidad independiente que incluye todos los atributos relevantes del pasajero (por ejemplo, fullName, dateOfBirth, passportNumber) junto con los atributos específicos del baneo (nationality, reason, banDate, liftDate).
- **Ventajas:**
  - **Integridad Histórica:** Se conserva de forma inmutable la información del pasajero en el momento del baneo, evitando que cambios posteriores en la entidad Passenger afecten el registro del baneo.
  - **Auditoría y Trazabilidad:** Permite un seguimiento claro y preciso del estado del baneo, útil para auditorías o consultas históricas.
- **Desventajas:**
  - **Duplicación de Datos:** Se replican ciertos atributos ya presentes en Passenger, lo que puede generar redundancia en el modelo de datos.

#### Opción 2: Relación entre BannedPassenger y Passenger (por ejemplo, ManyToOne)

- **Definición:**  
Se modela el baneo como una relación con la entidad Passenger, derivando algunos atributos del pasajero relacionado.
- **Ventajas:**
  - **Evita la Duplicación de Datos:** Se utiliza la información actualizada de Passenger sin necesidad de replicarla en otra entidad.
- **Desventajas:**
  - **Pérdida de Información Histórica:** Si los datos del pasajero cambian con el tiempo, el registro del baneo también se actualizará, lo que podría dificultar el seguimiento de la situación en el momento del baneo.
  - **Dependencia de la Entidad Original:** El registro del baneo queda fuertemente acoplado a la entidad Passenger, lo que puede no ser deseable en ciertos contextos (por ejemplo, auditorías).

## Elección Justificada: Entidad BannedPassenger Independiente

- **Preservación** **Histórica:**  
Al almacenar de forma independiente los datos del pasajero en el momento del baneo, se asegura que el registro se mantenga inalterable incluso si la información en Passenger cambia posteriormente.
- **Mejor** **Trazabilidad:**  
La entidad independiente permite un seguimiento claro y confiable de cuándo y por qué un pasajero fue baneado, así como de si y cuándo se levantó el baneo.
- **Cumplimiento** **de** **Requisitos:**  
La entidad BannedPassenger cumple con el requisito de modelar todos los atributos solicitados (nombre, fecha de nacimiento, pasaporte, nacionalidad, motivo, fecha de baneo y fecha de levantamiento opcional) de forma completa y autocontenida.

Esta decisión favorece la integridad y la trazabilidad de la información en el sistema, asegurando que los registros históricos no se vean afectados por futuras modificaciones en la entidad Passenger.

## Decisión de Diseño 2: Modelado de atributos estadísticos en CustomerDashboard

### 1) Incluir atributos directamente en la clase CustomerDashboard

- **Definición:**  
Los atributos estadísticos (como bookingCostAverage, bookingCostMinimum, passengerCount, etc.) se incluyen directamente en la clase CustomerDashboard en lugar de crear una clase específica para estadísticas.
- **Ventajas:**
  - **Sencillez:** Evita crear una clase adicional que podría resultar innecesaria si no se va a reutilizar en otro contexto.
  - **Claridad:** Todos los datos relevantes del dashboard están definidos claramente en un único lugar.
- **Desventajas:**
  - **Poca Reusabilidad:** Si en el futuro fuera necesario reutilizar estas estadísticas, sería necesario refactorizar el código.
  - **Mayor Complejidad de la clase:** Puede incrementar la complejidad visual o cognitiva de la clase al incluir muchos atributos relacionados.

### 2) Crear una clase específica Statistics

- **Definición:**  
Crear una clase separada Statistics que agrupe los atributos estadísticos y reutilizar esta clase dentro del CustomerDashboard.

- **Ventajas:**
  - **Reusabilidad:** Permite reutilizar fácilmente estas estadísticas en otros contextos o dashboards futuros.
  - **Mejor Modularidad:** Facilita la gestión y mantenimiento del código.
- **Desventajas:**
  - **Complejidad adicional:** Se crea una clase más, que puede ser innecesaria si no se reutilizan estos atributos en ningún otro sitio.

#### **Elección Justificada: Atributos Directos en CustomerDashboard**

- **Contexto Específico:** Debido a que estos atributos no serán reutilizados en otros contextos, la inclusión directa en la clase actual simplifica el modelo y reduce la complejidad.
- **Simplicidad del diseño:** Mantener los atributos estadísticos en la misma clase contribuye a mantener un diseño sencillo y comprensible.

Esta decisión se basa en el principio de simplicidad y en la evaluación de la necesidad real del proyecto, que no anticipa el uso futuro de estos atributos fuera del contexto inmediato del dashboard actual.

## **5. Conclusión**

Este informe ha documentado las dudas surgidas durante el desarrollo del proyecto, analizando distintas alternativas y justificando las decisiones tomadas en cada caso. Se ha seguido un enfoque estructurado para evaluar las opciones disponibles y validar la solución más adecuada en cada situación.

El análisis realizado permite garantizar una interpretación precisa de los requisitos y una correcta aplicación de los criterios establecidos. Además, contribuye a mejorar la toma de decisiones y optimizar el proceso de desarrollo, asegurando coherencia y alineación con los objetivos del proyecto.

## **6. Bibliografía**

Intencionalmente en blanco.