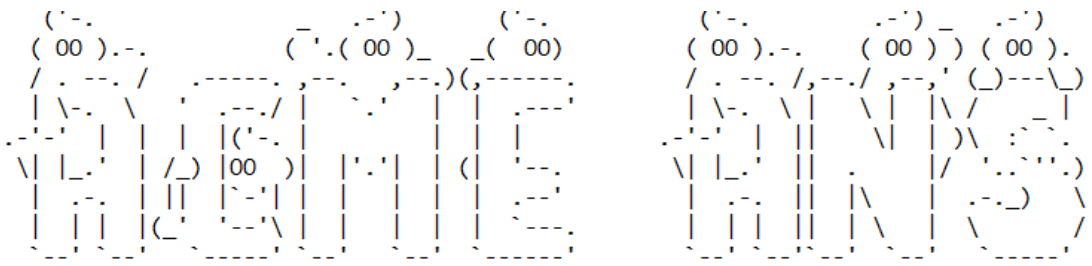
REPORTE DE ANÁLISIS

Acme-ANS-D04



Repositorio: https://github.com/FranciscoFernandezN/Acme-ANS

Creado por el grupo C1.022, del G1

Participantes	
Nombres	Correos
Gutiérrez Arazo, Beatriz	beagutara@alum.us.es

23 de mayo de 2025

Índice

Resumen ejecutivo

Este es el proyecto del grupo C1.022 sobre Acme AirNav Solutions, S.A. (Acme ANS, S.A. abreviado), la cual es una compañía ficticia especializada en ayudar a aeropuertos a organizar y coordinar sus operaciones a partir de soluciones desarrolladas en software. La logística de los vuelos (la programación de los vuelos, la organización de reservas y de tripulación, etc.) se gestionan mediante el desarrollo de un WIS. Con esto queda implícito que no solo será necesario desarrollar la aplicación en sí, sino también una documentación apropiada en la que se refleje la evolución de esta.

En este caso, se tratará de un reporte de análisis del proyecto, donde se describe toda la información relacionada con las decisiones de diseños que ha tomado el estudiante 2 en esta entrega del proyecto.

Tabla de revisiones

Número	Fecha	Descripción
v1.0.0	23/05/2025	Versión finalizada del documento para el entregable 4

Introducción

El análisis de un proyecto consiste en entender los requisitos dados para tener la capacidad de resolverlos. Esto conlleva tomar decisiones debido a que puede haber más de una alternativa a la hora de desarrollar soluciones, cada una con sus pros y sus contras, y no siempre habrá una correcta, sino una que se adaptará mejor al proyecto que debe de ser identificada y escogida.

Este reporte está organizado de la siguiente forma:

1. **Resumen ejecutivo**: Introducción breve sobre el reporte.
2. **Tabla de revisiones**: Historial de revisiones del documento.
3. **Introducción**: Contextualización del análisis además de su importancia.
4. **Contenidos**: Descripción de algunas decisiones tomadas.
5. **Conclusiones**: Resumen de los hallazgos y la importancia de este reporte.
6. **Bibliografía**: Fuentes consultadas durante la investigación.

Contenidos

Decisión #1: Cambios en las relaciones entre customer, booking y passenger.

Requisito asociado: Requisito 8 individual / *Operations by customers on bookings:*

- *List their bookings.*
- *Show the details of their bookings and the associated passengers, if any.*
- *Create or update their bookings. Bookings can be updated as long as they have not been published. A booking can be published only when the last credit card nibble has been stored*

Requisito 9 individual / *Operations by customers on passengers:*

- *List the passengers in their bookings.*
- *Show the details of their passengers.*
- *Create a passenger and record the information related to that passenger.*
- *Update a passenger as long as it has not been published.*

Requisito 15 individual / *The system must handle customers dashboards with the following indicators:*

- *The last five destinations.*
- *The money spent in bookings during the last year.*
- *Their number of bookings grouped by travel class.*
- *Count, average, minimum, maximum, and standard deviation of the cost of their bookings in the last five years.*
- *Count, average, minimum, maximum, and standard deviation of the number of pas-sengers in their bookings*

Problema encontrado: las operaciones para contar, hacer la media, el mínimo y el máximo de pasajeros por reserva no tienen sentido si solo hay un pasajero en cada reserva.

Soluciones posibles valoradas:

1. Mantenerlo como está ahora (oneToMany de booking a passenger).
    a. Pros:
        i. Simplicidad de código (no hay que cambiar nada de antes).
        ii. Familiaridad con el código.
    b. Contras
        i. Dificultad para añadir nuevas funcionalidades
        ii. No resuelve el nuevo problema encontrado.
2. Poner una manyToMany entre booking y passenger.
    a. Pros:
        i. Mejor adaptación a los requisitos.
    b. Contras
        i. Complejidad de código (habrá que editar código creado previamente).
        ii. Necesidad de crear una tabla intermedia y sus operaciones necesarias.

Solución adoptada: esta vez, se optó por la segunda opción, creando una tabla intermedia entre booking y passenger. La entidad para manejar esto se le llamará “BelongsTo”, y tendrá una relación ManyToOne hacia booking y hacia passenger, de tal forma que una reserva pueda ser de varios pasajeros, y que un pasajero pueda tener varias reservas.

Validación del profesor: proporcionada en clase presencial.

Decisión #2: Tipo de recomendación

Requisito asociado: Requisito 26 individual / *The system must include a board to recommend something in the city and/or country of a given airport. Recommendations can be about experiences, activities, restaurants, accommodation or any other thing that a person may find interesting at the destination. A web service must be used to populate this entity with information about recommendations. Thus, the exact data to store depends on the chosen service, and it is the students' responsibility to define them accordingly. It is also the students' responsibility to find the appropriate service; no implicit or explicit liabilities shall be covered by the University of Seville or their individual affiliates if the students contract pay-per-use services! The students are strongly advised to ensure that the service they choose is free of charge.*

Problema encontrado: la API escogida anteriormente no es funcional, por lo que he de buscar otra. los requisitos no especifican qué tipo de recomendación ha de usarse.

Soluciones posibles valoradas:

1. Restaurantes en la ciudad.
    a. Pros:
        i. Aparece en el propio requisito.
        ii. Existe una amplia variedad de APIs de este tipo.
    b. Contras
        i. Si bien es importante para los turistas, estos no suelen viajar a sitios basándose únicamente en los restaurantes.
2. Puntos de interés.
    a. Pros:
        i. Existe una amplia variedad de APIs de este tipo.
        ii. Es información muy interesante para los turistas.
    b. Contras
        i. No aparece explícitamente en el requisito.
3. Actividades en la ciudad.
    a. Pros:
        i. Aparece en el propio requisito.
        ii. Es información muy interesante para los turistas.
    b. Contras
        i. Las APIs existentes tienen cuotas de peticiones muy bajas o son de pago.

Solución adoptada: se ha optado por la segunda opción. A partir de la API Places de Google, se han obtenido los puntos de interés en función de las ciudades de los aeropuertos de destino de los vuelos.

Validación del profesor: proporcionada en clase presencial.

Conclusiones

Como conclusión, de nuevo no he tomado apenas decisiones de diseño, aunque sí han sido cruciales realizarlas. Si bien haber decidido mal una relación entre entidades es algo que ha llevado bastante tiempo corregir, es mejor haberlo cambiado ahora que dejarlo mal para la entrega final . Aparte de eso, considero que he hecho una labor satisfactoria como analista.

Bibliografía

Intencionalmente en blanco