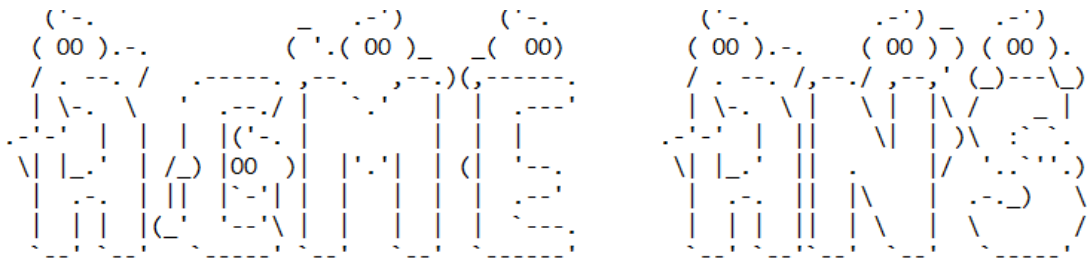


REPORTE DE ANÁLISIS

Acme-ANS-D03



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

3 de abril de 2025

Índice

Portada.....	1
Índice.....	2
Resumen ejecutivo.....	3
Tabla de revisiones.....	4
Introducción.....	5
Contenidos.....	6
Decisión #1:.....	6
Decisión #2:.....	7
Conclusiones.....	8
Bibliografía.....	9

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

[illegible]

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:

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.*

Problema encontrado: no saber si la relación entre booking y passenger estará bien.

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
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: se ha optado por la primera opción, creando una relación entre customer y passenger para poder manejar el requisito 9.

Validación del profesor: Proporcionada en clase presencial.

Decisión #2:

Requisito asociado: Requisito 26 / *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: He de escoger una api que proporciona recomendaciones en una ciudad.

Soluciones posibles valoradas:

1. Escoger una api de pago.
 - a. Pros:
 - i. Peticiones ilimitadas.
 - b. Contras
 - i. Habría que pagar, y la Universidad de Sevilla no lo cubre.
 - ii. Seguramente no hagan falta peticiones ilimitadas.
2. Escoger una api gratuita.
 - a. Pros:
 - i. No hay que pagar.
 - ii. El requisito lo recomienda.
 - b. Contras
 - i. El número de peticiones seguramente será limitado.

Solución adoptada: se ha optado por la segunda opción, escogeré una api gratuita (en principio, la que he encontrado en:

<https://developers.amadeus.com/self-service/category/destination-experiences/api-doc/tours-and-activities>).

Validación del profesor: validación no necesaria.

Conclusiones

Como conclusión, al igual que en el entregable anterior, no he tomado apenas decisiones de diseño. Sí considero que estas decisiones han sido aún más importantes, y gran parte de la lógica de la aplicación dependerá de ellas. Aparte de eso, considero que he hecho una labor satisfactoria como analista.

Bibliografía

Intencionalmente en blanco