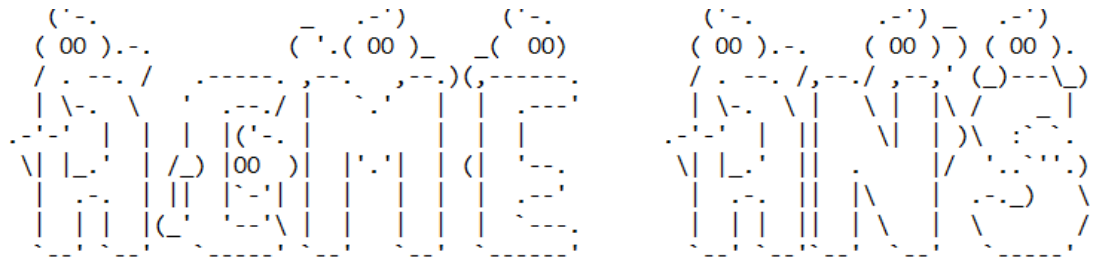


REPORTE DE ANÁLISIS

Acme-ANS-D02



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

13 de Marzo 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 he tomado en la segunda entrega del proyecto. A comparación de la entrega anterior, sí existirán documentos individuales ya que en esta entrega sí se han tomado más decisiones por cuenta propia que no repercuten a la globalidad 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 4 individual / *A booking is a reservation made by a customer to purchase a flight, guaranteeing some seats on a specific itinerary and associating some passengers' details with the trip. The system must manage the following information for each booking: a locator code (unique, pattern "[A-Z0-9]{6,8}\$"), a purchase moment (in the past), a travel class ("ECONOMY", "BUSINESS"), and a price. Optionally, the system should record the last nibble of the credit card used for payment.*

Problema encontrado: no saber cómo poner las relaciones con flight, customer y passenger.

Soluciones posibles valoradas:

1. Booking tendrá una relación ManyToOne con las tres entidades.
 - a. Pros:
 - i. Simplicidad de código.
 - ii. A partir de una reserva, se podría obtener de qué vuelo es, quién la compró y para quién es.
 - b. Contras
 - i. Desde la entidad flight no podremos obtener (directamente) las reservas que contiene.
 - ii. Desde la entidad customer no podemos ver las reservas que ha hecho.
 - iii. Desde la entidad passenger no podemos ver las reservas que tiene.
2. Flight, Customer y Passenger tendrán una relación OneToMany con Bookings.
 - a. Pros:
 - i. Desde la entidad flight podremos obtener (directamente) las reservas que contiene.
 - ii. Desde la entidad customer podemos ver las reservas que ha hecho.
 - iii. Desde la entidad passenger podemos ver las reservas que tiene.
 - b. Contras
 - i. El código se vuelve más complejo (creación de tablas intermedias debido al uso de listas en las relaciones).
 - ii. A partir de una reserva, no se podría obtener de qué vuelo es, quién la compró ni para quién es.
 - iii. Es probable que se repitan entidades de Bookings, debido a que tendrán que ser guardadas simultáneamente

Solución adoptada: Booking tendrá una relación ManyToOne con Flight, Customer y Passenger (es decir, la opción 1).

Validación del profesor: Sin validación necesaria.

Decisión #2:

Requisitos asociados: Requisito 38 grupal / *The system must store the following data: a full name (up to 50 characters), date of birth (in the past), a passport number (pattern “[A-Z0-9]{6,9}\$”), nationality (up to 50 characters), the reason for the ban (up to 255 characters), the date the ban was issued (in the past), and an optional lift date to indicate whether the passenger is still banned or if the prohibition has been lifted.*

Requisito 5 individual / *A passenger is an individual who takes a flight and he or she must be registered in the corresponding booking. The system must store the following data about passengers: a full name (shorter than 256 characters), an email, a passport number (pattern “[A-Z0-9]{6,9}\$”), a date of birth, and, optionally, his or her special needs (shorter than 51 characters).*

Problema encontrado: no saber si repetir dos roles casi idénticos.

Soluciones posibles valoradas:

1. Crear passenger y banned passenger como viene explícitamente en los requisitos.
 - a. Pros:
 - i. Mayor relación con los requisitos.
 - b. Contras
 - i. Tenemos dos roles casi idénticos.
2. Crear passenger y una entidad aparte que solo tengan los banned passengers.
 - a. Pros:
 - i. No crearemos roles casi idénticos.
 - ii. Evitaremos repetición de datos.
 - iii. Optimización de código.
 - b. Contras
 - i. No viene explícitamente así en los requisitos.

Solución adoptada: se ha optado por la segunda opción, es decir, hemos creado la entidad Passenger y la entidad Ban, la cual tiene una relación ManyToOne con Passenger y contiene datos respecto a dicho pasajero que tiene (o tuvo) prohibido volar.

Validación del profesor: Proporcionada en clase presencial.

Conclusiones

Como conclusión, no me han surgido demasiadas dudas durante la realización del proyecto, lo cuál me ha llevado a tener que valorar distintas alternativas pocas veces, o entre las valoradas la cantidad de pros y contras eran tan distintas que no había que debatir demasiado al respecto, pues era clara la solución óptima. Aparte de eso, considero que he hecho una labor satisfactoria como analista.

Bibliografía

Intencionalmente en blanco