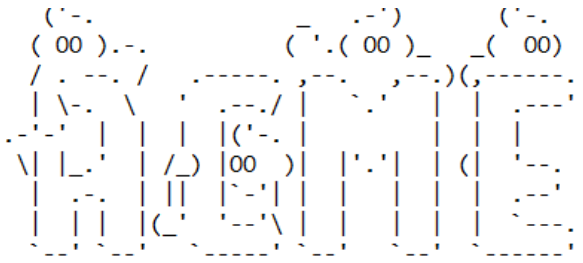
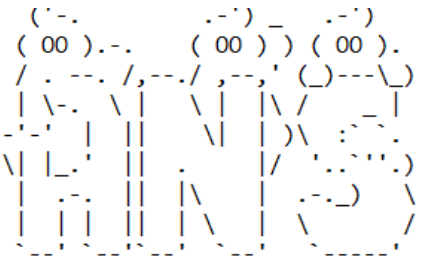
REPORTE DE ANÁLISIS

Acme-ANS-D04





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

Creado por el grupo C1.022, del G1

Participantes	
Nombres	Correos
Benito Merchán, Manuel Jesús	manbenmer1@alum.us.es
Fernández Noguerol, Francisco	frafernog@alum.us.es
Gómez Navarro, Esteban	estgomnav@alum.us.es
Gutiérrez Arazo, Beatriz	beagutara@alum.us.es
Varo Vera, Juan	juavarver@alum.us.es

25 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, para el correcto funcionamiento de la aplicación, se deberá escribir un reporte de testing donde se describan los valores de cobertura del código junto con los valores correspondientes al rendimiento de la aplicación, ambos, previos y posteriores a realizar mutaciones en el código.

Tabla de revisiones

Número	Fecha	Descripción
v1.0.0	25/05/2025	Versión finalizada del documento para el entregable 4

Introducción

En este documento se describe toda la información relacionada con las decisiones de diseños tomadas por el grupo para la última entrega del proyecto. Al tratarse de la última entrega, la cantidad de decisiones tomadas ya es más baja y solo han tenido lugar para modificaciones de los requisitos que ya estaban hechos.

Contenidos

Decisión #1:

Requisito asociado: Requisito 32 grupal / *Money amounts, Booleans, and moments must be internationalised when they are shown. Other kinds of data are not expected to be internationalised. Internally, all moments must be stored in GMT format. This requirement must be fulfilled in this and every other group or individual deliverable for it to be considered satisfied.*

Problema encontrado: Ver como internacionalizar las fechas

Soluciones posibles valoradas:

1. Mejorar el atributo fecha:
    a. Pros:
    Mayor versatilidad en el código
    Menor cantidad de cambios necesarios en el proyecto
    b. Contras:
    Habría que cambiar parte del framework
    Habría que normalizar todas las validaciones de las fechas en las features
2. Guardar los datos tal y como están y asumir que son GMT:
    a. Pros:
    0 necesidad de cambiar código
    Sistema sencillo que comprende cualquier persona
    b. Contras:
    Menos versatilidad para el usuario final

Solución adoptada: Guardar los datos tal y como ya estaban, y esto es, primeramente, debido a que el requisito no explica muy bien lo que quiere, por lo tanto, no haremos trabajo extra en vano, en segundo lugar, para una internacionalización de las fechas parecidas a la realizada con el idioma, habría que cambiar tanto mucha cantidad del proyecto como mucha cantidad del framework.

Validación del profesor: Sin validación necesaria.

Decisión #2:

Requisito asociado: Requisito 24 grupal / *I24) The system configuration must include the following initial data: a system currency, which must be initialised to "EUR" and a list of accepted currencies, which must be initialised to "EUR", "USD", and "GBP".*

Problema encontrado: En la anterior entrega se estableció como inamovible, la moneda por defecto del sistema, y esta no tenía repercusión en el proyecto.

Soluciones posibles valoradas:

1. Dejar la moneda por defecto inamovible y que todos los datos se guarden en la base de datos traduciéndose a esta moneda
    a. Pros:
    Coherencia dentro de la base de datos
    No haría falta cambiar el código de los dashboard
    b. Contras:
    Inutilización del atributo currency dentro del objeto Money
    Llamada a la api de conversión cada vez que se quiera modificar una entrada que tenga dinero
2. Poder cambiar la moneda por defecto, pero los datos en la base de datos se guardarían en la moneda inicialmente escrita
    a. Pros:
    Mayor versatilidad a la hora de usar el objeto Money
    Menos llamadas a la api, (solo a la hora de leer)
    b. Contras:
    Necesidad de cambiar todos los dashboard relacionados con el dinero
    Traducción obligatoria en todos las features relacionadas con el dinero

Solución adoptada: Poder cambiar la moneda por defecto, así hemos visto que es más versátil y útil, además de no tener que cambiar tantas partes el código como creíamos.

Validación del profesor: Sin validación necesaria.

Conclusiones

Como conclusión, a raíz de la revisión de tareas grupales que hemos realizado, se han detectado ciertas incongruencias que han tenido que ser debatidas y puestas en práctica.

Bibliografía

Intencionalmente en blanco