

Diseño y Pruebas II



ANALYSIS REPORT

Grupo: C1.02.07

Repositorio: <https://github.com/DP2-C1-02-07/Acme-L3>

Autor/es:

- Javier Nunes Ruiz - **javnunrui@alum.us.es**
- Manuel Palacios Pineda - **manpalpin@alum.us.es**
- Manuel Carnero Vergel - **mancarver1@alum.us.es**
- Pablo Martínez Valladares - **pabmarval@alum.us.es**
- Julio Navarro Rodríguez - **julnavrod@alum.us.es**

Fecha: 16/02/2023

1. Índice

1. Índice	2
2. Tabla de versiones	3
3. Resumen del documento	3
4. Introducción	3
5. Contenidos	4
5.1. Requisitos de información	4
5.2. Requisitos funcionales	5
5.3. Requisitos no funcionales	5
5.4. Requisitos de testing	5
5.5. Requisitos de gestión	6
6. Conclusiones	6
7. Bibliografía	7

2. Tabla de versiones

Versión	Fecha	Descripción
1.0.0	21/02/2023	Documento creado
1.0.1	22/02/2023	Modificaciones en análisis
1.0.2	23/02/2023	Documento completado

3. Resumen del documento

En este documento se recopila información sobre los requisitos que han requerido un análisis exhaustivo para su desarrollo.

En el análisis de cada requisito incluimos información sobre la descripción que se nos da del requisito, una conclusión tras analizarlo en la que se incluye las decisiones tomadas para llevar a cabo el requisito y, en algunos casos, un link a la validación del profesor sobre la decisión tomada.

4. Introducción

Un informe de análisis es un documento que recoge información sobre el proceso de análisis, en este caso, de los requisitos que han requerido dicho análisis

En ese análisis se plantean diversas formas de afrontar el problema que supone el requisito, las cuales se comparan para tomar una decisión final, decisión de la cual se debe explicar en qué consiste y por qué se ha elegido, teniendo en cuenta que esta decisión debe estar validada.

Para este documento se ha optado por usar una estructura clásica a la hora de realizarlo. Esta estructura está compuesta por una portada con información breve del grupo, así como un resumen del documento , una introducción, el contenido principal, una conclusión y la bibliografía consultada para su realización.

5. Contenidos

5.1. Requisitos de información

REQUISITO 9): The system configuration must include the following initial data:

- A system currency, which must be initialised to “EUR”.
- A list of accepted currencies, which must be initialised to “EUR”, “USD”, and “GBP”.

ANÁLISIS: Este primer requisito que encontramos en el entregable se puede realizar de forma relativamente sencilla. Para ello, debemos crear una entidad `SystemConfiguration` con un atributo “system currency” de tipo `Money`, que representará la moneda principal del sistema; y un atributo “accepted currencies” de tipo `String`, que representa la lista de divisas aceptadas por el sistema, separadas por una coma sin espacios. Esta decisión se ha tomado a futuro para que sea más fácil modificar la lista de divisas aceptadas en el sistema.

REQUISITO 12): An offer is a record in which an administrator advertises something. The system must store the following data about them: an instantiation moment (in the past), a heading (not blank, shorter than 76 characters), a summary (not blank, shorter than 101 characters), an availability period (at least one day after the offer is instantiated and must last for at least one week), a price (positive, possibly nought), and an optional link with further information.

ANÁLISIS: Para este requisito, pensamos que donde radica la mayor dificultad es en el atributo “availability period”, ya que no es un atributo que podamos implementar directamente con algún tipo temporal de Java compatible con Acme-Framework. Para tratar este inconveniente, decidimos dividir este atributo en dos: “start period” y “finish period”. De esta manera, cuando implementemos las operaciones CRUD para esta entidad podremos validar de forma más sencilla la duración de este periodo comparando estos dos atributos (concretamente, esta validación se hará en el Create Service y en el Update Service).

REQUISITO 13): A note is a message posted by an authenticated principal. The system must store the following data about them: an instantiation moment, a title (not blank, shorter than 76 characters), an author (not blank, shorter than 76 characters), a message (not blank, shorter than 101 characters), an optional email address, and an optional link. The author must be computed as follows: “⟨username⟩ - ⟨surname, name⟩”, where “⟨user-name⟩” denotes the username of the principal who has posted the note and “⟨surname, name⟩” denotes his or her full name.

ANÁLISIS: A diferencia del anterior requisito, cuya mayor dificultad se encontraba en un atributo de tipo temporal, en este requisito se encuentra en el atributo “author”, ya que depende de otros atributos ajenos a la propia entidad. Para poder efectuar la restricción del formato que debe tener el atributo, habrá que esperar a crear el Create Service y el Update Service de esta entidad en el próximo entregable.

REQUISITO 14): A banner allows administrators to advertise products, services, or organisations. The system must store the following data about them: an instantiation/update moment (in the past), a display period (must start at any moment after the instantiation/update moment and must last for at least one week), a link to a picture that must be stored somewhere else, a slogan (not blank, shorter than 76 characters), and a link to a target web document.

ANÁLISIS: Para este requisito, el atributo “display period” presenta la misma problemática que el atributo “availability period” del requisito 12. Es por esto que la resolución a este problema por la que hemos optado es prácticamente idéntica: dividir este atributo en dos, “start period” y “finish period”; y validar su duración en los servicios una vez comencemos con el tercer entregable.

REQUISITO 15): The system must handle administrator dashboards with the following indicators: total number of principals with each role; ratio of peeps with both an email address and a link; ratios of critical and non-critical bulletins; average, minimum, maximum, and standard deviation of the budget in the offers grouped by currency; average, minimum, maximum, and standard deviation of the number of notes posted over the last 10 weeks.

ANÁLISIS: Este requisito es el más complejo a simple vista entre todos los requisitos de información, ya que está relacionado con varias entidades y requiere de mucha información de las mismas entidades. Las partes más complicadas de implementar serían las de mostrar los valores estadísticos de los presupuestos y del número de notas.

Por una parte, la mayor complicación de las estadísticas de los presupuestos consiste en la agrupación por monedas. Como en un futuro la configuración del sistema puede cambiar, lo que proponemos es una lista de Double por cada valor estadístico (media, desviación, máximo y mínimo). En estas listas, cada posición representará una divisa del sistema, de tal manera que si se añade o se elimina una divisa no sea necesario replantear la clase SystemConfiguration.

Por otra parte, en las estadísticas de notas creemos que la dificultad radica en obtener las notas que se encuentran dentro del tiempo determinado en el requisito.

5.2. Requisitos funcionales

No ha sido necesario realizar un análisis de los requisitos funcionales debido a que este entregable no incorporaba requisitos de este tipo.

5.3. Requisitos no funcionales

No ha sido necesario realizar un análisis de los requisitos no funcionales debido a que este entregable no incorporaba requisitos de este tipo.

5.4. Requisitos de testing

No ha sido necesario realizar un análisis de los requisitos de testing debido a que el único requisito incluido en esta categoría no ameritaba un análisis en detalle del mismo.

5.5. Requisitos de gestión

REQUISITO 19): Producir un modelo de dominio UML

ANÁLISIS: En este último requisito incluido en el documento nos encontramos con una duda en cuanto a la inclusión de los requisitos individuales de cada uno. En el follow-up del día 23 de febrero, le preguntamos a nuestro profesor Pepe González si era necesario incluir las entidades derivadas de los requisitos individuales en el diagrama.

El profesor nos ofreció tres alternativas:

- Incluir todas las entidades derivadas de los requisitos individuales en el diagrama; junto con una leyenda para saber qué entidades son grupales y cuáles son las que corresponden a cada miembro en los requisitos individuales
- No incluir ninguna entidad individual para no mezclar requisitos grupales con individuales.
- Reflexionar y proponer una tercera vía en la que se llegue a una especie de punto medio entre las dos anteriores.

Decidimos optar por la tercera vía e incluir en el diagrama solamente los roles a implementar por cada estudiante como una única entidad llamada “Authenticated principal”, ya que estos son realmente los que intervienen con las entidades a desarrollar de manera grupal.

6. Conclusiones

Una vez finalizado este documento, con el que damos por terminado el segundo informe de análisis del grupo C1.02.07, se han adquirido nuevos conocimientos sobre cómo elaborar un modelo de dominio UML, conocimientos que nos servirán de ayuda en nuestro futuro en el mundo laboral.

Debido a conocimientos adquiridos previamente, varios requisitos han quedado fuera de este documento. Esto se debe a que gracias a estos conocimientos no ha sido necesario practicar un análisis sobre estos requisitos, ya que no mostraban demasiada complejidad. Entre estos requisitos se encuentran entidades simples y sencillas de implementar; y documentos previamente elaborados en esta misma asignatura.

7. Bibliografía

- “On your tutorials” - Material proporcionado en la asignatura DP2 en la Universidad de Sevilla.