

# Informe de análisis individual Alejandro Gallardo Pelayo

---

## **Datos del grupo**

**Group:** C2.03.06

**Repository:** <https://github.com/alejandromd/acme-L3-D02>

## **Student #2**

**ID Number:** 54183271D

**UVUS:** alegalpel  
([alegalpel@alum.us.es](mailto:alegalpel@alum.us.es))

**Name:** Gallardo Pelayo,  
Alejandro

**Roles:** tester, desarrollador

**Date:** Sevilla, a 6 de julio de 2023

## **Índice**

1. Executive summary
2. Tabla de versiones
3. Introducción
4. Contenido
5. Conclusión

## **Executive summary**

El objetivo de este documento es realizar un análisis de los requisitos que se han implementado durante este segundo entregable, en el que se incluye una copia exacta del propio requisito en sí, así como las decisiones y conclusiones tras haber completado la implementación de este requisito.

## **Tabla de versiones**

<u>Versión</u>	<u>Descripción</u>	<u>Fecha</u>
v1.0	Creación y primera versión del documento	15/03/23
v2.0	Mejor del documento para la convocatoria de julio	06/07/23

## Introducción

En este documento realizaremos un análisis de los requisitos que hemos desarrollado durante el entregable. En este entregable, se han implementado un total de dos entidades, asociadas a una entidad rol. También, se ha implementado un formulario para ese mismo rol, así como un poblado de la base de datos para probar dichas entidades. Por tanto, se han implementado 4 requisitos de información y un requisito de prueba para el poblado de la base de datos.

## Contenido

- Análisis 1: atributo código de la entidad “enrolment”.

El requisito en cuestión es el requisito número 5:

*“An enrolment is a registration of a student in a course. The system must store the following data about them: a code (pattern “[A-Z]{1,3}[0-9][0-9]{3}”, not blank, unique), a motivation (not blank, shorter than 76 characters), some goals (not blank, shorter than 101 characters), and a work time (in hours, computed from the corresponding activities).”*

Como se puede observar, el patrón que nos proporcionan para el atributo *code* puede dar lugar a una confusión, debido a la forma en la que viene expresado. El patrón que nos indica el cliente, indica que el atributo puede tener de 1 a 3 letras seguidas de 3 dígitos del 1 al 9. Sin embargo, la forma en la que el cliente expresa el patrón no es errónea, pero puede lugar a confusión, ya que podría expresarse directamente como “[A-Z]{1,3}[0-9]{3}”. Después, pude confirmar que efectivamente se trata de un error del cliente, ya que pude encontrar en el foro una pregunta relacionada con este tema en cuestión. [link](#)

En esta pregunta se indica que, efectivamente la forma correcta de expresar el patrón es “[A-Z]{1,3}[0-9]{3}”. Finalmente, me di cuenta que el patrón podía expresarse como “[A-Z]{1,3}\\d{3}”, ya que la expresión *d{3}* indica que tiene que contener 3 dígitos. Por tanto, tomé la decisión de expresar el patrón de esa forma, ya que sería una forma más simplificada de expresarlo.

- Análisis 2: atributo derivado en la entidad “enrolment”

El requisito en cuestión es el requisito número 5:

*“An enrolment is a registration of a student in a course. The system must store the following data about them: a code (pattern “[A-Z]{1,3}[0-9][0-9]{3}”, not blank, unique), a motivation (not blank, shorter than 76 characters), some goals (not blank, shorter than 101 characters), and a work time (in hours, computed from the corresponding activities).”*

El problema con este requisito surge a la hora de implementar el atributo derivado *work time*, que se calcula en base a la duración de las actividades de ese *enrolment*. El problema es que hay direccionalidad desde la entidad *Activity* hacia la entidad *Enrolment* pero no al contrario, por tanto pensé dos posibles alternativas:

- 1) La primera alternativa sería crear una relación bidireccional entre ambas entidades, lo que resolvería el problema de la direccionalidad. Sin embargo, esta alternativa tiene un gran inconveniente, ya que como hemos visto en la asignatura, estas relaciones son mucho más complejas y propensas a tener errores.
- 2) Otra alternativa sería realizar el cálculo del atributo derivado directamente en el servicio de la entidad y accediendo a los datos a través del repositorio, sin necesidad de usar relaciones. La gran ventaja sería que podríamos seguir manteniendo la relación unidireccional entre ambas entidades. En el foro, pude encontrar diferentes soluciones para esto y viendo la siguiente cuestión [link](#), me di cuenta que los atributos derivados no pueden ser implementados por ahora y se explicarán en la siguiente lección de teoría. Por ello, pude suponer que sería en el servicio probablemente.

Tras analizar ambas alternativas, he tomado la decisión de escoger la alternativa 2) y esperar a futuras lecciones de teoría para implementar dicho atributo derivado, ya que considero que es la que tiene más ventajas. El atributo se ha incluido en el modelo de dominio, pero aún no ha sido implementado.

## **Conclusión**

En resumen, considero que he tenido algunas dificultades al implementar los requisitos de información sobre la entidad *enrolment* y *activity*. En especial, me han surgido problemas al implementar dicha entidad *enrolment* ya que había varias alternativas para implementar esta entidad. Al final, se ha llegado a una conclusión y ha sido posible implementar dicho requisito con éxito.

## **Bibliografía**

intentionally blank

