

# REPORTE DE ANÁLISIS

**Grupo:** C1.03.03

**Repositorio:** <https://github.com/ferbaqfer/Acme-L3-D02>

**Estudiante #5**

**DNI:** 77930171F

**UVUS:** gonriblun

**Nombre:** Ribas Luna, Gonzalo

**Rol:** Developer

**Fecha:** 17/03/2023

# Índice

1.-Executive summary .....	2
2.-Tabla de revisiones.....	2
3.-Introducción .....	2
4.-Contenido.....	3
4.1.- Listado de requisitos .....	3
4.2.- Decisiones sobre los requisitos .....	4
5.-Conclusiones .....	5
6.-Bibliografía .....	5

## 1.-Executive summary

Durante la realización del análisis de este entregable no han surgido muchos problemas, y los que han surgido no han sido graves o complicados, por lo que no han supuesto un excesivo incremento de las horas planeadas.

## 2.-Tabla de revisiones

Fecha	Versión	Descripción
15/03/2023	1.0	Creación del documento y listado de los requisitos
17/03/2023	1.1	Elaboración de la conclusión y las decisiones a tomar

## 3.-Introducción

A continuación, se detallarán los requisitos exigidos para este entregable, así como un análisis individual de cada requisito o tarea a partir de la cual he obtenido una conclusión, y por tanto, una o varias decisiones para su realización.

Este análisis se dividirá en dos bloques, uno en el que se mostrará un listado de los requisitos exigidos y otro bloque en el que se dará la conclusión a la que se ha llegado, de forma individual o grupal, acerca del requisito que se analiza y las decisiones a tomar para el mismo.

## 4.-Contenido

### 4.1.- Listado de requisitos

En este entregable encontramos más requerimientos que en la entrega anterior, concretamente 7. No han conllevado mucha dificultad al estar todavía empezando el proyecto y ser la mayoría de los requisitos crear nuevas clases con sus respectivas relaciones. Estos requisitos son:

1. Hay un nuevo rol específico del proyecto llamado "auditor", que tiene los siguientes datos de perfil: empresa (no en blanco, más corta que 76 caracteres), identificación profesional (no en blanco, más corta que 26 caracteres), una lista de certificaciones (no en blanco, más corta que 101 caracteres) y un enlace opcional con información adicional.
2. Una auditoría es un documento con registros de auditoría relacionados con un curso publicado. El sistema debe almacenar los siguientes datos sobre ellos: un código (patrón "[A-Z]{1,3}[0-9][0-9]{3}", no en blanco, único), una conclusión (no en blanco, más corta que 101 caracteres), algunos puntos fuertes (no en blanco, más cortos que 101 caracteres), algunos puntos débiles (no en blanco, más cortos que 101 caracteres) y una nota (calculada como la moda de las notas en los registros de auditoría correspondientes; los empates deben resolverse arbitrariamente si es necesario).
3. El sistema debe almacenar los siguientes datos sobre los registros de auditoría de un curso: una asignatura (no en blanco, más corta que 76 caracteres), una evaluación (no en blanco, más corta que 101 caracteres), el período durante el cual se realizó la auditoría del tema (en el pasado, al menos de una hora de duración), una calificación ("A+", "A", "B", "C", "F" o "F-"), y un enlace opcional con información adicional.
4. El sistema debe manejar los paneles de control de los auditores con los siguientes datos: número total de auditorías que han realizado para cursos teóricos y prácticos; promedio, desviación, mínimo y máximo de registros de auditoría en sus auditorías; promedio, desviación, mínimo y máximo tiempo de duración de los períodos en sus registros de auditoría.
5. Produce datos de muestra variados para probar su aplicación de manera informal. Los datos deben incluir dos cuentas de auditor con credenciales "auditor1 / auditor1" y "auditor2 / auditor2".
6. Elabora un reporte de análisis.
7. Elabora un reporte de planificación.

#### 4.2.- Decisiones sobre los requisitos

No he encontrado mucha dificultad en los requisitos, ya que son cosas simples, aunque al no estar familiarizado con este framework hay cosas a las que les he dedicado mas tiempo del que me esperaba, por ejemplo, con los datos de prueba en los csv.

1. Para el primer requisito llegué a la conclusión de que tenía que crear un nuevo rol llamado Auditor, donde recogería la información expresada en el requisito, y poniendo restricciones encima de cada atributo utilizando la notación @.

No hubo complicaciones con este requisito.

2. Para el segundo requisito cree la entidad Audit, con los atributos especificados en el requisito y sus respectivas restricciones. Además, tendría una relación ManyToOne con auditor y otra con course (esta entidad la hizo otro compañero) y una relación OneToMany con auditingRecord, ya que un mismo auditor y un mismo curso pueden tener varias auditorías, pero una auditoría va sobre un único auditor y un único curso, y un registro de una auditoría trata sobre una auditoría en concreto pero una auditoría puede tener varios registros.

El atributo “mark” o “nota” es un atributo derivado, por lo que no se pondrá como un atributo como los demás, sino elaborando una función que la calcule.

En este requisito ha surgido un problema, ya que al estar Audit relacionada con Course, y Course ser requisito de otro compañero, tuve que esperar a que él la crease para que no diese fallo mi parte del código.

3. En este requisito concluí que tenía que crear una nueva entidad llamada AuditingRecord, con los atributos especificados en el requisito y una clase Mark que sería un enumerado con las notas especificadas.

Tuve un problema de decisión con el requisito del periodo:

- a. Hacer una función para comprobar que el periodo estaba bien.
- b. Dejar ese requisito como una validación para siguientes entregables.

Al final opte, con la ayuda del profesor, por escoger la opción b, pero haciendo la opción a y dejándola comentada por si me pudiese servir en un futuro.

Otro problema de decisión que me surgió es si hacía doble la relación entre Audit y AuditingRecord:

- a. Poner la relación solo en Audit
- b. Poner la relacion tanto en Audit como en AuditingRecord

Elegí la solución b, porque en los requisitos no está especificado si tiene que ser solo de Audit a AuditingRecord o puede ser en las dos direcciones, y puede ser útil en un futuro poder acceder a una Audit desde una AuditingRecord.

4. Para el cuarto requisito llegué a la conclusion de hacer la clase AuditingDashboard con los datos que me especifican el requisito.

No hubo ninguna complicación con este requisito.

5. Para este requisito surgió la duda de si tendría que crear datos de prueba para todas las clases que habia creado o no:
  - a. Crear datos de prueba solo para Auditor.
  - b. Crear datos de prueba para Auditor, Audit y AuditingRecord.

Al final decidí que tenía más sentido hacer la opción b, ya que si solo pruebo auditor estaría dejando muchos casos posibles en los que la aplicación podría estar fallando.

6. Para este requisito no ha habido ninguna complicación.
7. Para este requisito no ha habido ninguna complicación.

## 5.-Conclusiones

He tenido algún problema al elaborar este entregable, aunque ninguno ha sido grave o muy complicado.

A pesar de haber tenido 4 problemas, estos no han supuesto un excesivo incremento del tiempo planificado, al ser problemas que se resolvieron preguntando la duda al profesor o planificándome mejor con el compañero.

## 6.-Bibliografía

Intentionally blank