Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

ANALYSIS REPORT INDIVIDUAL - D02



**C1.027**

**Repositorio**: <https://github.com/josrojrom1/DP2-G27-Acme>

**Autor:**

Rojas Romero, José Joaquín

josreimun@alum.us.es

**Fecha:** 19/02/2024

**Tabla de contenidos**

[**Resumen**](#_heading=h.gjdgxs) **2**

[**Tabla de revisión**](#_heading=h.30j0zll) **2**

[**Introducción**](#_heading=h.1fob9te) **2**

[**Contenidos**](#_heading=h.3znysh7) **3**

[**Bibliografía**](#_heading=h.3dy6vkm) **3**

# Resumen

Este documento se trata de un listado de registros de análisis, cada uno de los cuales deberá incluir los siguientes datos: copia textual del requerimiento a que se refiere el registro; conclusiones detalladas del análisis y decisiones tomadas para enmendar el requerimiento, en caso de ser necesario.

# Tabla de revisión

| **Versión** | **Fecha** | **Descripción** |
| --- | --- | --- |
| V1 | 19/02/2024 | Creación y desarrollo del documento |

# Introducción

En este documento se va a llevar a cabo el análisis de los requisitos implementados individualmente por el miembro José Joaquín Rojas Romero.

# Contenidos

## **Requisitos Funcionales**

**2) Code audits** are essential pieces to ensure the quality of a **project**. The system must store the following data about them: a **code** (pattern “[A-Z]{1,3}-[0-9]{3}”, not blank, unique), an **execution** date (in the past), a **type** (“Static”, “Dynamic”), a list of proposed **corrective actions** (not blank, shorter than 101 characters), a **mark** (computed as the mode of the marks in the corresponding auditing records; ties must be broken arbitrarily if necessary), and an **optional link** with further information.

La dificultad que he encontrado a la hora de realizar este requisito ha sido a la hora del desarrollo del atributo derivado ***mark***, debido a que en primera instancia no tenía claro cómo implementar la moda de dicho atributo para la clase entidad ***Code Audit*** (conglomerado). No se dieron explicaciones técnicas ni en clase de teoría ni en los *follow ups*, por lo que tuve que hablar con los compañeros de mi grupo para conseguir alguna información extra. Al buscar en el foro de la asignatura encontré que alguien justo estaba preguntando la misma duda que yo tenía por lo que al leer la solución aportada por el tutor entendí cómo afrontar dicho problema de desarrollo. Simplemente éste cálculo de la moda se implementa en la capa de servicios en el método bind/unbind y la teoría en relación a ello aún no se ha dado, pues pertenece al contenido L03.

Enlace a la cuestión en concreto en el foro de la asignatura: <https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board&conf_id=_405265_1&forum_id=_234042_1&message_id=_405313_1>

**3)** The result of each **code audit** is based on the analysis of their **audit records**.The system must store the following data about them: a **code** (pattern “AU-[0-9]{4}-[0-9]{3}”, not blank, unique), the **period** during which the subject was audited (in the past, at least one hour long), a **mark** (“A+”, “A”, “B”, “C”, “F”, or “F-”), and an **optional link** with further information.

La dificultad que he encontrado en este requisito es similar a la del anterior requisito, aunque esta vez se especifica en la teoría el cómo calcular la validación del atributo ***period***, donde su duración tiene que ser como mínimo de una hora, por lo que entiendo que su implementación dependerá de la futura teoría dada en L03 en la capa de servicio en el método que gestione las validaciones, por lo que dicha validación aún no podrá ser realizada. Aún así se ha dejado comentado un método en la clase ***AuditRecord.java*** el cual sirve de base para resolver este problema futuro, donde en cualquier caso se puede copiar y pegar donde haga falta haciendo mínimas modificaciones para que se cumpla la validación.

# Bibliografía

-Documento 08 Annexes, de los contenidos de la plataforma virtual de esta asignatura.