

# **Analysis report**



## **Diseño y pruebas II**

**Sprint 4**

**Versión 1.0**

**Fecha 27/05/2024**

Preparado por:  
Fernando José de Celis Hurtado

# Índice

|   |          |
|---|----------|
| <b>1.) Introducción</b>   | <b>3</b> |
| <b>2.) Contenido</b>  | <b>3</b> |
| 2.1) Implementación entidad code audit (Information requirement 2)                          | 3        |
| 2.2) Implementación entidad audit record (Information requirement 3)                        | 4        |
| 2.3) Implementación auditor dashboard (Information requirement 4)                           | 4        |
| 2.4) Implementación del testing individual de la aplicación (Information requirement 5)     | 4        |
| 2.5) Operaciones de auditors en code audits (Information requirement 6)                     | 5        |
| 2.6) Operaciones de auditors en audit records (Information requirement 7)                   | 5        |
| 2.7) Operaciones de auditors en el panel de indicadores (Information requirement 8)         | 6        |
| 2.8) Testing de los requirements 6 y 7 (Information requirement 9)                          | 6        |
| 2.9) Hacer un testing report (Information requirement 10)                                   | 6        |
| 2.10) Implementación rol auditor (Information requirement 13)                               | 7        |
| 2.11) Hacer el UML (Information requirement 14)   | 7        |
| 2.12) Hacer el analysis report Sprint2 (Information requirement 15)                         | 7        |
| 2.13) Hacer el planning and progress report Sprint2 (Information requirement 16)            | 7        |
| 2.14) Hacerse auditor (Information requirement 17)  | 7        |
| 2.15) Actualizar los perfiles de auditor (Information requirement 18)                       | 8        |
| 2.16) Mostrar los code audits publicados con sus audit records (Information requirement 19) | 8        |
| 2.17) Hacer el planning and progress report Sprint3 (Information requirement 21)            | 8        |
| 2.18) Hacer el analysis report Sprint4 (Information requirement 23)                         | 8        |
| 2.19) Hacer el planning and progress report Sprint4 (Information requirement 24)            | 8        |
| <b>3.) Conclusiones</b>   | <b>9</b> |

# 1.) Introducción

Un análisis detallado resulta esencial para el progreso y la constante optimización. La elaboración de un informe de análisis desempeña un papel fundamental en este proceso, ofreciendo una perspectiva minuciosa sobre los requisitos del sistema. Este documento no solo evalúa el grado de cumplimiento de dichos requisitos, sino que también destaca áreas susceptibles de mejora, proponiendo soluciones que buscan potenciar la eficiencia y funcionalidad del sistema.

El propósito fundamental de este informe consiste en ofrecer una guía clara y estructurada para el perfeccionamiento del sistema. Se busca no solo asegurar el cumplimiento de los requisitos establecidos, sino también alcanzar los más altos estándares de calidad en cada uno de ellos.

## 2.) Contenido

### 2.1) Implementación entidad code audit (Information requirement 2)

Descripción: 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.

Este requisito nos pide almacenar en la base de datos la información correspondiente de los code audits, los cuales tienen ciertas características y deben satisfacer una serie de restricciones. Para cumplir con este requisito, se ha creado la entidad CodeAudit. Para ello se creará un paquete dentro de entities llamado codeAudits, y dentro de este una clase CodeAudit.

Para poder hacer este requisito correctamente, se ha tenido que crear otra clase enumerado CodeAuditType para poder hacer uno de los atributos de la entidad CodeAudit.

Uno de los atributos de la clase (Mark) debe ser calculado a partir de otro atributo de otra clase que se explicará posteriormente (atributo Mark de la entidad AuditRecord), este atributo se calculará en la clase AuditorCodeAuditRepository.java (creada en el information requirement 6) pasando las notas de los audit records pertenecientes al code audit a números, pasados a números se hacen una media y el resultado de la media se vuelve a transformar a nota, siendo esta la nota media del code audit en base a las notas de las clases hijas.

También el atributo execution debe ser inferior a todas las auditStartTime de sus entidades hijas audit records. Estas restricciones se han hecho en el sprint 3 en diferentes servicios (create, update y publish).

## 2.2) Implementación entidad audit record (Information requirement 3)

Descripción: 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.

Este requisito nos pide almacenar en la base de datos la información correspondiente de los audit records, los cuales tienen ciertas características y deben satisfacer una serie de restricciones. Para cumplir con este requisito, se ha creado la entidad AuditRecord. Para ello se creará un paquete dentro de entities llamado auditRecords, y dentro de este una clase AuditRecord.

Para poder hacer este requisito correctamente, se ha tenido que crear otra clase enumerado Mark para poder hacer uno de los atributos de la entidad AuditRecord.

En esta entidad, el atributo auditEndTime debe ser superior al atributo auditStartTime como mínimo en una hora y también auditStartTime debe ser superior al atributo execution de su entidad padre code audit. Estas restricciones se han hecho en el sprint 3 en diferentes servicios (create, update y publish)..

## 2.3) Implementación auditor dashboard (Information requirement 4)

Descripción: The system must handle auditor dashboards with the following data: total number of code audits for "Static" and "Dynamic" types; average, deviation, minimum, and maximum number of audit records in their audits; average, deviation, minimum, and maximum time of the period lengths in their audit records.

Este requisito nos pide almacenar una serie de información relacionada con los auditors, que se irá actualizando a medida que se vayan añadiendo code audits y audit records. Para ello se creará una clase dentro de forms llamada AuditorDashboard.

Posteriormente en el sprint 3, se arregló un bug el cual daba incoherencias cuando no había code audits y/o audit records sin publicar el cual era que si me pedían el máximo o mínimo de algo y devolvía 0 cuando debería devolver null o vacío ya que el máximo de un conjunto vacío no existe.

## 2.4) Implementación del testing individual de la aplicación (Information requirement 5)

Descripción: Produce assorted sample data to test your application informally. The data must include two auditor accounts with credentials "auditor1/auditor1" and "auditor2/auditor2".

Este requisito nos indica la necesidad de generar datos para poder realizar un testing informal sobre nuestra aplicación. Para cumplir con este requisito, se realizarán datos de prueba siguiendo las buenas prácticas de la asignatura. Se añadirán datos del rol auditor, de la entidad CodeAudit y de la entidad AuditRecord.

En este requisito surgieron diversos problemas puesto que la estructura de los csv son muy delicadas y si no haces a la perfección las tablas de datos, te salta error.

En el sprint 4 se cambiaron por completo los csv creados en el sprint 2 ya que gracias al testing formal, me percaté de la escasez de ejemplos que tenía y de varios errores que había cometido, es por ello que se actualizaron posteriormente.

## 2.5) Operaciones de auditors en code audits (Information requirement 6)

Descripción: List the code audits that they have created. Show the details of their code audits. Create, update, or delete their code audits. Code audits can be updated or deleted as long as they have not been published. For a code audit to be published, the mark must be, at least, "C".

Este requisito nos pide realizar diferentes acciones con las code audits, estas acciones son listarlas, mostrarlas, crearlas, borrarlas, publicarlas y actualizarlas con ciertas restricciones. Para poder realizar esta tarea se han tenido que crear diferentes servicios (uno por acción), un controller y un repository en el paquete codeAudit dentro del paquete auditor dentro del paquete features.

Gracias a esta tarea se pudo hacer en el repository el cálculo de su nota en base a las notas de sus entidades hijas la cual fue explicada en el requirement 2

## 2.6) Operaciones de auditors en audit records (Information requirement 7)

Descripción: List the audit records in their code audits. Show the details of their audit records. Create and publish an audit record. Update or delete an audit record as long as it is not published.

Este requisito nos pide realizar diferentes acciones con los audit records estas acciones son listarlos, mostrarlos, crearlos, borrarlos, publicarlos y actualizarlos con ciertas restricciones. Para poder realizar esta tarea se han tenido que crear diferentes servicios (uno por acción), un controller y un repository en el paquete auditRecord dentro del paquete auditor dentro del paquete features.

## 2.7) Operaciones de auditors en el panel de indicadores (Information requirement 8)

Descripción: Show their auditor dashboards.

Este requisito nos pide mostrar el panel con diferentes parámetros implementado en el requirement 4. Para poder realizar esta tarea se han tenido que crear un servicio para poder hacer el show, un controller y un repository en el paquete dashboard dentro del paquete auditor dentro del paquete features.

Gracias a este requirement me percaté de un error a la hora de representar el máximo/mínimo de algunos atributos ya que mostraba 0 cuando el conjunto de entidades está vacío no debería mostrar 0, debería mostrarse vacío o nulo.

## 2.8) Testing de los requirements 6 y 7 (Information requirement 9)

Descripción: Produce a test suite for Requirements #6 and #7.

Este requisito nos pide realizar el testing formal de los requisitos 6 y 7, los cuales son code audit y audit record respectivamente. Para realizar esta actividad se hizo uso de diferentes herramientas que proporciona el framework. Fui grabando cada caso de test generandose un archivo tester.safe y moviendolo a sus carpetas correspondientes de code audit o audit record cambiando sus nombres a .safe o a .hack, según el test grabado.

Durante la realización de esta actividad me percaté de diferentes errores como por ejemplo que no había tenido en cuenta las validaciones que relacionan las fechas de las code audits con las de los audit records. También se arreglaron los csv de las dos entidades siguiendo al pie de la letra las recomendaciones de las diapositivas y cogiendo los datos de ejemplo del csv que viene incluido en el workspace.

## 2.9) Hacer un testing report (Information requirement 10)

Descripción: Produce a testing report.

Este requisito nos pide realizar un documento en base al testing realizado, dicho documento se divide en dos puntos principales. En el primer punto se ha analizado el testing realizado y explicando el coverage resultante, explicando todos los casos en los que salen “líneas amarillas”, documentándose y demostrándose. En la segunda parte podemos comparar el rendimientos en diferentes ordenadores, lo cual puede ser interesante ya que de primera vista parecen cosas incoherentes ya que en ciertos casos un ordenador más potente saca tiempos peores. Y también podemos comparar los tiempos en el mismo ordenador con y sin índices incluidos.

## 2.10) Implementación rol auditor (Information requirement 13)

Descripción: Produce assorted sample data to test your application informally. The data must include two auditor accounts with credentials “auditor1/auditor1” and “auditor2/auditor2”.

Aquí se nos pide crear un rol auditor, que será el rol que se encargue de los code audits. Para ello se ha creado una clase en la carpeta roles llamada Auditor.

### 2.11) Hacer el UML (Information requirement 14)

Descripción: Produce a UML domain model.

Este requisito nos pide crear un UML que represente todas las entidades junto al rol y al dashboard creados que se han indicado previamente, así como las relaciones entre ellas.

El UML se actualizó en el sprint 4 con dos nuevas restricciones.

### 2.12) Hacer el analysis report Sprint2 (Information requirement 15)

Descripción: Produce an analysis report.

Para este requisito no se han encontrado dificultades, ya que es relativamente sencillo. No ha sido necesario analizar este requisito.

### 2.13) Hacer el planning and progress report Sprint2 (Information requirement 16)

Descripción: Produce a planning and progress report.

Para este requisito no se han encontrado dificultades, ya que es relativamente sencillo. No ha sido necesario analizar este requisito.

### 2.14) Hacerse auditor (Information requirement 17)

Descripción: Sign up to the system and become an auditor.

Para este requisito no se han encontrado dificultades, ya que es relativamente sencillo. No ha sido necesario analizar este requisito.

### 2.15) Actualizar los perfiles de auditor (Information requirement 18)

Descripción: Update their profiles.

Para este requisito no se han encontrado dificultades, ya que es relativamente sencillo. No ha sido necesario analizar este requisito.

### 2.16) Mostrar los code audits publicados con sus audit records (Information requirement 19)

Descripción: List the code audits in the system that are published. Show the details of the code audits that they can list (including their audit records).

Durante la realización de este requirement se creó una nueva query en el repositorio ya que hasta el momento de hacer esta actividad no necesitaba una query que me devolviera los code audits publicados. Primeramente se realizó sin tener en cuenta sólo las code audits pero posteriormente se corrigió.

### 2.17) Hacer el planning and progress report Sprint3 (Information requirement 21)

Descripción: Produce a planning and progress report.

Para este requisito no se han encontrado dificultades, ya que es relativamente sencillo. No ha sido necesario analizar este requisito.

### 2.18) Hacer el analysis report Sprint4 (Information requirement 23)

Descripción: Produce an analysis report.

Para este requisito no se han encontrado dificultades, ya que es relativamente sencillo. No ha sido necesario analizar este requisito.

### 2.19) Hacer el planning and progress report Sprint4 (Information requirement 24)

Descripción: Produce a planning and progress report.

Para este requisito no se han encontrado dificultades, ya que es relativamente sencillo. No ha sido necesario analizar este requisito.

## **3.) Conclusiones**

En conclusión, este proyecto ha sido un camino lleno de piedras con las que me he tropezado. Gracias a las reuniones follow-up y actividades en posteriores sprints se han descubierto errores implementados previamente y se han arreglado satisfactoriamente. Se



han tenido pocos inconvenientes en cuanto a la hora de trabajar. Se corrigió el uso de las ramas y en el sprint 3 se empezaron a realizar una rama por actividad, siendo esta forma mucho más efectiva en comparación a la metodología de trabajo realizada en los dos primeros sprints. El trabajo en equipo ha ayudado mucho también para arreglar fallos y en el avance del cuatrimestre se ha visto una mejoría de química en el equipo.