Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas II

Curso 2023 – 2024

**Grupo:** C1.014

**Repositorio**:

**Fecha**:

|  |  |
| --- | --- |
| **Integrante del grupo** | **Correo** |
| Pérez Fernández, Ibai | ibaperfer@alum.us.es |

**Historial de versiones**

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Descripción de los cambios** |
| V1.0 | 07/03/2024 | Creación del documento |
| V1.1 | 08/03/2024 | Adición de la información relacionada con los requisitos 2, 3, 4 y 5 |
| V1.2 | 08/03/2024 | Adición de la información relacionada con los requisitos 13, 14, 15 y 16 |
|  |  |  |
|  |  |  |

**Tabla de revisiones**

|  |  |  |
| --- | --- | --- |
| **Número de revisión** | **Fecha** | **Descripción** |
| 1.0 | 08/03/2024 | Revisión general del documento en busca de fallos o pequeños errores. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Índice de contenido**

[**1.** **Resumen ejecutivo** 2](#_Toc158831141)

[**2.** **Introducción** 2](#_Toc158831142)

[**3.** **Contenido** 2](#_Toc158831143)

[**3.1.** **Contenido1** 2](#_Toc158831144)

[**4.** **Conclusiones** 2](#_Toc158831145)

[**Bibliografía** 2](#_Toc158831146)

# **Resumen ejecutivo**

Este documento recoge la información requerida y necesaria para la realización de un reporte de análisis.

# **Introducción**

Un informe de análisis es una herramienta fundamental para el desarrollo y mejora continua. Este tipo de informe proporciona una visión detallada de los requisitos del sistema, evaluando su cumplimiento, identificando áreas de mejora y proponiendo soluciones para optimizar su funcionalidad y eficiencia.

A continuación, se describirán cada uno de los requisitos individuales del proyecto y se expondrán las conclusiones obtenidas después de la realización de las mismas. Además se añadirán notas adicionales cuando se crea oportuno para obtener la mayor claridad y calidad posible.

# **Contenido**

# **Requisito 2**

**Requisito:** Las auditorias de código son una parte esencial para asegurar la calidad de un proyecto. El sistema debe almacenar la siguiente información sobre estas: un código (patrón “[A-Z]{1,3}-[0-9]{3}”, no en blanco, único), una fecha de ejecución ( en el pasado), un tipo (“Static”, “Dynamic”), una lista de acciones de corrección propuestas (no en blanco, con menos de 101 caracteres), una nota ( calculada como la moda de las notas de sus anotaciones), y un link opcional con más información.

**Conclusión del análisis:** Este requisito nos indica la necesidad de almacenar la información correspondiente a las auditorias de código en la base de datos, la cual debe tener unas características y satisfacer unas restricciones.

**Decisión:** Para cumplir con el requisito se ha creado una nueva entidad llamada CodeAudit. Para ello se ha creado una nueva clase dentro del paquete acme.entities.audit que cumplirá lo solicitado. Además, para modelar el tipo de auditoría se ha creado un enum llamado AuditType con los valores Static y Dynamic en el mismo paquete.

**Nota adicional:** Se han encontrado algunos problemas con respecto al tipo de algunos atributos de la entidad, tales como “executionDate” y “correctiveActions”. Tras un análisis de pros y contras se ha determinado que: “executionDate” será un tipo Date con TemporalType.TIMESTAMP, y “correctiveActions” será un String en el que cada acción se verá delimitada con dos puntos.

* 1. **Requisito 3**

**Requisito:** El resultado de cada auditoría de código se basa en los análisis de sus anotaciones de auditoría. El sistema debe almacenar la siguiente información sobre estas anotaciones: un código (patrón “AU-[0-9]{4}-[0-9]{3}”, no en blanco, único), el periodo durante el que se llevó a cabo (en el pasado, mayor que una hora), una nota (“A+”, “A”, “B”, “C”, “F”, o “F-”), y un link opcional con más información.

**Conclusión del análisis:** Este requisito nos indica la necesidad de almacenar la información correspondiente a los análisis de cada auditoria de código en la base de datos, la cual debe tener unas características y satisfacer unas restricciones. Además, esta debe estar relacionada como una composición con su auditoría.

**Decisión:** Para cumplir con el requisito se ha creado una nueva entidad llamada AuditRecord. Para ello se ha creado una nueva clase dentro del paquete acme.entities.audit que cumplirá lo solicitado. Además, para modelar el tipo de auditoría se ha creado un enum llamado Mark con los valores A\_PLUS, A, B, C, F Y F\_MINUS y su conversión a String en el mismo paquete.

**Nota adicional:** Se han encontrado problemas con respecto a algunos atributos de la identidad, tales como “period” y “mark”. Tras un análisis de pros y contras se ha determinado que: “period” constará de dos fechas “startDate” y “endDate” ambas de tipo Date con TemporalType.TIMESTAMP, y para “mark” al no poder guardar caracteres especiales como + o – se decidió escribir su significado para que quede representado.

* 1. **Requisito 4**

**Requisito:** El sistema debe encargarse de los paneles de los auditores/as con la siguiente información: número total de auditorías para los tipos “Static” y “Dynamic”: media, desviación, mínimo y máximo de anotaciones en sus auditorias; media, desviación, mínimo y máximo de duración de los periodos de sus anotaciones.

**Conclusión del análisis:** Este requisito nos indica la necesidad de almacenar la información correspondiente a los auditores/as de código que se irá actualizando cuando se vayan añadiendo más auditorías.

**Decisión:** Para cumplir con este requisito, se requerirá de la creación de un Form AuditorDashboard, debido a que según nos describen, la información irá variando y no será persistente.

**Nota adicional:** Debido a que no es un requisito excesivamente complejo, no se han encontrado problemas en la realización de este.

* 1. **Requisito 5**

**Requisito:** Produce información de ejemplo para probar tu aplicación de manera informal. La información debe incluir dos cuentas de auditores/as con las credenciales “**auditor1/auditor1**” y “**auditor2/auditor2**”.

**Conclusión del análisis:** Este requisito nos indica la necesidad de generar datos para poder realizar un testing informal sobre nuestra aplicación.

**Decisión:** Para cumplir con este requisito, se realizará la inserción de numerosos datos de prueba siguiendo las buenas prácticas de la asignatura, para testear informalmente la aplicación. Se añadirán datos del rol Auditor (26 casos), de la entidad CodeAudit (30 casos) y de la entidad AuditRecord (35 casos).

**Nota adicional:** Debido a que no es un requisito excesivamente complejo, no se han encontrado problemas en la realización de este.

* 1. **Requisito 13**

**Requisito:** Hay un nuevo rol específico de un proyecto llamado auditor, el cual tiene la siguiente información de perfil: agencia (no en blanco, con menos de 76 caracteres), un identificador profesional (no en blanco, con menos de 26 caracteres), una lista de certificaciones (no en blanco, con menos de 101 caracteres), y un link opcional con más información.

**Conclusión del análisis:** Este requisito nos indica la necesidad de almacenar en la base de datos la información correspondiente de los auditores/as, los cuales deben tener unas características y satisfacer una serie de restricciones mencionadas anteriormente. Para ello se creará una nueva clase dentro de acme.roles llamada Auditor que cumpla con lo solicitado en el requisito.

**Decisión:** Para cumplir con este requisito, se requerirá de la creación del rol Auditor.

**Nota adicional:** Debido a que no es un requisito excesivamente complejo y se han tenido que implementar entidades con atributos parecidos a este caso, no se han encontrado problemas en la realización de este.

* 1. **Requisito 14**

**Requisito:** Produce un diagrama UML de dominio.

**Conclusión del Análisis:** Es un requisito claro y conciso, por lo que no ha requerido de análisis.

**Nota adicional:** Debido a que no es un requisito excesivamente complejo, no se han encontrado problemas en la realización de este.

* 1. **Requisito 15**

**Requisito:** Produce un informe de análisis.

**Conclusión del Análisis:** Es un requisito claro y conciso, por lo que no ha requerido de análisis.

**Decisión:** Se han revisado los anexos que detallan la información requerida para el informe, y se ha elaborado el documento teniendo en consideración dicha información.

* 1. **Requisito 16**

**Requisito:** Produce un informe de planificación y progreso.

**Conclusión del Análisis:** Es un requisito claro y conciso, por lo que no ha requerido de análisis.

**Decisión:** Se han revisado los anexos que detallan la información requerida para el informe, y se ha elaborado el documento teniendo en consideración dicha información.

# **Conclusiones**

En este entregable se han realizado múltiples tareas de implementación de entidades que han dado lugar a dudas, pero con la ayuda del foro, las reuniones con el cliente y los propios miembros del equipo de desarrollo he sabido solventarlas satisfactoriamente.

# **Bibliografía**

Intencionalmente en blanco