Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática



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

Curso 2023 – 2024

|  |  |
| --- | --- |
| **Fecha** | **Versión** |
| 13/02/24 | 1.0 |

|  |  |
| --- | --- |
| **Grupo de Prácticas: C1.012** | |
| **Repositorio:** [https://github.com/DP2-C1-012/Acme-SF](mailto:https://github.com/DP2-C1-012/Acme-SF) | |
| **Autores por orden alfabético** | **Correo** |
| Bernal Caunedo, Álvaro Jesús | [alvbercau@alum.us.es](mailto:alvbercau@alum.us.es) |
| Caballero Hernández, Jaime | [jaicabher1@alum.us.es](mailto:jaicabher1@alum.us.es) |
| Casamitjana Benítez, Juan José | [juacasben@alum.us.es](mailto:juacasben@alum.us.es) |
| Herrera Lobo, Nicolás | [nicherlob@alum.us.es](mailto:nicherlob@alum.us.es) |
| Montoya Albitres, Ronald Alexander | [ronmonalb@alum.us.es](mailto:ronmonalb@alum.us.es) |

**Índice de contenido**

[1. Resumen Ejecutivo 2](#_Toc474589782)

[2. Tabla de revisiones 2](#_Toc179714428)

[3. Introducción 3](#_Toc1184956288)

[4. Contenido 4](#_Toc647377246)

[5. Conclusiones 6](#_Toc201131074)

[6. Bibliografía 7](#_Toc1190488549)

# **Resumen Ejecutivo**

El informe de análisis individual contiene un resumen de los requisitos contiene los requisitos implementados en el sprint correspondiente y una explicación detallada de como se han llevado a cabo si es necesario aclarar algún punto.

# **Tabla de revisiones**

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Revisión** | **Descripción** |
| 01/03/2024 | 1.0 | Redacción inicial |
| 27/05/2024 | 2.0 | Cambio del análisis a tabla con pros y contras. |

# **Introducción**

El análisis de los requisitos es una parte esencial de la ejecución de un sprint, nos permite identificar errores en la redacción de estos y tratarlos con los responsables para desarrollar la funcionalidad correspondiente de forma satisfactoria. En estos casos es necesario dejar constancia de estas decisiones, si se toma alguna.

# **Contenido**

|  |  |  |
| --- | --- | --- |
| Funcionalidad | Decisión | Pros y contras |
| **Mandatory:** 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. | *Se han creado relaciones específicas unidireccionales hacia Project y Auditor, Type se ha considerado como un enumerado, lo mismo que Mark, que además se guarda en base de datos a pesar de ser computado, ya que hace que este valor sea constante a medida que se hace un nuevo cálculo porque al usarse la moda hay que romper los empates de forma arbitraria y hay condiciones sobre Mark en requisitos futuros para publicar o no un Code Audit, por esto se ha optado a guardar en base de datos.* | Pros:   * Aumenta la simplicidad y la consistencia de los datos a la hora de guardarlos en base de datos   Contra:   * Ocupa más espacio en base de datos, sobre todo al guardar Mark. |
| **Mandatory:** 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. | *En este caso el período se ha considerado como dos atributos distintos, la fecha de inicio y la fecha de fin, la restricción de 1h se realizará en la próxima iteración en la capa de aplicación, Mark, como se ha dicho antes, consistirá en un enumerado.* | Pros:   * Es más fácil aplicar restricciones sobre un período definido por dos fechas que usando un dato alternativo no cubierto por el framework   Contra:   * Hay que validar ambos datos individualmente |
| **Supplementary:** There is a new project-specific role called auditor, which has the following profile data: firm (not blank, shorter than 76 characters), professional ID (not blank, shorter than 26 characters), a list of certifications (not blank, shorter than 101 characters), and an optional link with further information. | *En este rol se ha considerado el professionalID como unique porque no existe ningún otro campo junto que el que se pueda distinguir unicidad de forma contingente, list of certifications se ha considerado como un único campo de cadena de texto porque facilita el cálculo del límite de caracteres y reduce la complejidad de la base de datos.* | Pros:   * Puede ser una forma alternativa de identificación.   Contra:   * Puede no ser un campo único en realidad. |
| **Supplementary:** Produce a UML domain model. | *El UML se ha generado teniendo en cuenta las observaciones anteriores y procurando que las relaciones sean Many to One unidireccionales para reducir la complejidad de la base de datos y seguir las recomendaciones de la asignatura.* | Pros:   * Simplifica la direccionalidad de los datos y la complejidad de las relaciones   Contra:   * Puede dificultar la navegabilidad a la hora de trabajar con los datos. |

*.*

# **Conclusiones**

En conclusión, en este entregable se han tenido que aclarar algunos aspectos sobre los atributos de las entidades a implementar y como se integran con la aplicación, hay que añadir que la mayoría de estas aclaraciones provienen de las sesiones de Follow up.

# Bibliografía

Intencionalmente en blanco.