Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática



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

Curso 2023 – 2024

**Informe de análisis**

**Grupo:** C1.014

**Número de estudiante dentro del grupo:** 1

**Repositorio**: <https://github.com/Jaime-Linares/Acme-SF-D02>

**Fecha**: 07/03/2024

|  |  |
| --- | --- |
| **Alumno** | **Correo** |
| Linares Barrera, Jaime | jailinbar@alum.us.es |

**Historial de versiones**

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Descripción de los cambios** |
| V1.0 | 21/02/2024 | Creación del informe |
| V1.1 | 07/03/2024 | Elaboración y revisión del informe |
|  |  |  |

**Índice de contenido**

[**1.** **Resumen ejecutivo** 2](#_Toc160751224)

[**2.** **Introducción** 2](#_Toc160751225)

[**3.** **Contenido** 3](#_Toc160751226)

[**3.1.** **Segundo entregable** 3](#_Toc160751227)

[**4.** **Conclusiones** 6](#_Toc160751228)

[**Bibliografía** 7](#_Toc160751229)

# **Resumen ejecutivo**

Este documento se centra en examinar aquellos requisitos individuales del estudiante 1 que requieren un análisis detallado. Cada registro de análisis incluye el requisito en cuestión, conclusiones detalladas y decisiones para corregir problemas. Se destaca la validación por parte de un profesor designado para garantizar la calidad. El informe promueve la colaboración al permitir la revisión y aportes adicionales en un foro específico.

# **Introducción**

Este documento establece un enfoque estructurado para llevar a cabo y documentar análisis sobre requisitos específicos del proyecto en la parte individual del estudiante 1. El objetivo del informe de análisis es proporcionar una comprensión integral de los requisitos, articular conclusiones detalladas extraídas del proceso de análisis y proponer acciones apropiadas para abordar cualquier problema identificado.

El informe de análisis consta de una serie de registros de análisis, cada uno abordando requisitos específicos que requieren un examen exhaustivo. Estos registros incluyen una copia textual del requisito en cuestión, junto con conclusiones detalladas del análisis y decisiones para corregir discrepancias.

Para garantizar la calidad y validación de los resultados del análisis, cada registro incluye un enlace a la validación realizada por un profesor designado. Esta validación sirve como una medida de aseguramiento de la calidad, verificando la precisión y pertinencia de las conclusiones del análisis. Además, para fomentar la colaboración y obtener ideas adicionales, los registros de análisis están destinados a ser publicados en un foro designado. Esto permite al profesor revisar los hallazgos y proporcionar aportes adicionales si es necesario, enriqueciendo así el proceso de análisis y garantizando su exhaustividad.

# **Contenido**

# **Segundo entregable**

Requisito #2 - Estudiante 1:

*“A project aggregates several user stories elicited by the same manager. The system must store the following data about them: a code (pattern “[A-Z]{3}-[0-9]{4}”, not blank, unique), a title (not blank, shorter than 76 characters), an abstract (not blank, shorter than 101 characters), an indication on whether it has fatal errors, e.g., panics, a cost (positive or nought), and an optional link with further information. Projects containing fatal errors must be rejected by the system.”*

Se nos indica la necesidad de almacenar en la base de datos la información relacionada con los proyectos, los cuales deben tener unas características y satisfacer una serie de restricciones.

Para cumplir este requisito se requería la creación de la entidad *Project*. Para ello se creará un nuevo paquete dentro de *acme.entities* llamado *projects*, y dentro una clase java llamada *Project* que cumpla lo solicitado en el requisito.

Tras consultarlo con el cliente, este nos aclaró que el coste se refería al coste estimado en horas enteras y que este no derivaba de las historias de usuarios incluidas en tal proyecto.

[[Link al foro]](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board_entry&conf_id=_405265_1&forum_id=_234042_1&message_id=_404105_1)

Además, se nos aclaró que un proyecto se guarda en la base de datos aunque tenga errores fatales pero este no podrá publicarse, por lo que decidimos añadir la propiedad *published* para

dejar constancia en la base de datos de si el proyecto estaba o no publicado.

[[Link al foro]](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board_entry&conf_id=_405265_1&forum_id=_234042_1&message_id=_403856_1)

Por último, se fijaron ciertas restricciones que serían útiles como el número de caracteres máximos para un link y para el máximo valor que podría tomar el coste en horas enteras. Estos se tendrán en cuenta en todos los requisitos de aquí en adelante.

[[Link al foro]](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board_entry&conf_id=_405265_1&forum_id=_234042_1&message_id=_405561_1)

Requisito #3 - Estudiante 1:

*“A user story is a document that a manager uses to represent the smallest unit of work in a project. The system must store the following data about them: a title (not blank, shorter than 76 characters), a description (not blank, shorter than 101 characters), an estimated cost (in hours, positive, not nought), the acceptance criteria (not blank, shorter than 101 characters), a priority (“Must”, “Should”, “Could”, or “Won’t”), and an optional link with further information.”*

Se nos indica la necesidad de almacenar en la base de datos la información relacionada con los historias de usuario, las cuales deben tener unas características y satisfacer una serie de restricciones.

Para cumplir este requisito se requería la creación de la entidad *UserStory*. Para ello se creará un nuevo paquete dentro de *acme.entities* llamado *userStories*, y dentro una clase java llamada *UserStory* que cumpla lo solicitado en el requisito.

Al igual que en el coste relacionado con el proyecto, en este caso el coste (estimated cost) también está relacionado con horas enteras.

Además, hemos decidido añadirle la propiedad *published*, para dejar constancia en la base de datos si esa historia de usuario ha sido publicada o no, ya que a la hora de implementar la funcionalidad esto será un aspecto fundamental.

Por último, se nos aclaró que una historia de usuario podría estar en varios proyectos, de ahí que llevásemos a claro la creación de la entidad *AssociatedWith* para guardar la información de como se relacionan las historias de usuario y los proyectos.

[[Link al foro]](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board_entry&conf_id=_405265_1&forum_id=_234042_1&message_id=_405648_1)

Requisito #4 - Estudiante 1:

*“The system must handle manager dashboards with the following data: total number of “must”, “should”, “could”, and “won’t” user stories; average, deviation, minimum, and maximum estimated cost of the user stories; average, deviation, minimum, and maximum cost of the projects.”*

Se nos indica la necesidad de que el sistema trabaje con una serie de información (estadísticas) relacionada con los managers, que se irá actualizando a medida que un manager vaya creando y publicando historias de usuario y proyectos.

Para cumplir este requisito se requería la creación de un form *ManagerDashboard*. Para ello se creará dentro del paquete *acme.forms* una clase java llamada *ManagerDashboard* que cumpla lo solicitado en el requisito.

Este requisito es claro y conciso, por lo que no ha requerido ningún tipo de análisis.

Requisito #5 - Estudiante 1:

*“Produce assorted sample data to test your application informally. The data must include two manager accounts with credentials “manager1/manager1” and “manager2/manager2”.”*

Se nos indica la necesidad de que generar datos de prueba de manera ordenada para probar nuestra aplicación de manera informal.

Para ello es muy importante generar los datos de la manera más ordenada posible y probando los datos en los extremos (siempre válidos) y con caracteres extraños en el caso de las cadenas de caracteres.

Este requisito es claro y conciso, por lo que no ha requerido ningún tipo de análisis.

Requisito #13 - Estudiante 1:

*“There is a new project-specific role called manager, which has the following profile data: degree (not blank, shorter than 76 characters), an overview (not blank, shorter than 101 characters), list of certifications (not blank, shorter than 101 characters), and an optional link with further information.”*

Se nos indica la necesidad de crear un nuevo rol llamado *manager* y se nos pide que se guarden en la base de datos ciertas características sobre ellos.

Para cumplir este requisito se requería la creación de un rol *Manager*. Para ello se creará dentro del paquete *acme.roles* una clase java llamada *Manager* que cumpla lo solicitado en el requisito.

Lo único que cabe especificar en este requisito es que, con la aprobación del cliente, hemos guardado las certificaciones en una cadena de caracteres en vez de en una lista de cadena de caracteres ya que es una implementación más sencilla, eficiente y que no implica incumplir lo que el cliente nos pide.

[[Link al foro]](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board_entry&conf_id=_405265_1&forum_id=_234042_1&message_id=_407304_1)

Requisito #14 – Estudiante 1:

*“Produce a UML domain model.”*

Se pide elaborar un modelo UML del dominio de nuestro módulo del sistema.

Este requisito es claro y conciso, por lo que no requiere ningún tipo de análisis.

Requisito #15 – Estudiante 1:

*“Produce an analysis report.”*

Se pide elaborar un informe individual de análisis siguiendo las pautas del documento “Annexes”.

Este requisito es claro y conciso, por lo que no requiere ningún tipo de análisis.

Requisito #16 – Estudiante 1:

*“Produce a planning and progress report.”*

Se pide elaborar un informe individual de la planificación y el progreso siguiendo las pautas del documento “Annexes”.

Este requisito es claro y conciso, por lo que no requiere ningún tipo de análisis.

# **Conclusiones**

Este documento presenta un enfoque estructurado para analizar y documentar los requisitos específicos del proyecto para este segundo entregable, centrándose en las necesidades individuales del estudiante 1. Cada requisito se examina detalladamente, proporcionando conclusiones elaboradas y decisiones para abordar posibles problemas. La validación por parte de un profesor designado garantiza la calidad de los resultados.

El contenido se organiza en registros de análisis, cada uno dedicado a un requisito particular. Se destaca la importancia de la colaboración mediante la publicación de los análisis en un foro designado, permitiendo revisiones adicionales y aportes valiosos.

Los requisitos abordados incluyen la creación de entidades como Project y UserStory, así como la implementación de funcionalidades específicas como el dashboard para los managers y la generación de datos de prueba. Además, se establecen nuevas roles y se solicitan documentos como un modelo UML y varios informes.

En resumen, el documento proporciona una guía detallada para entender, analizar y abordar los requisitos del proyecto, asegurando su calidad y completitud mediante un proceso estructurado y colaborativo.

# **Bibliografía**

Intencionalmente en blanco.