

Grupo: C1.015

Repositorio: <https://github.com/jormunrod/Acme-SF-24.1.1>

Analysis report

Ángel García Escudero (anggaesc1@alum.us.es)

Sevilla, 8-3-2024

Contenido

1.	Resumen ejecutivo	2
2.	Tabla de revisión	3
3.	Introducción.....	4
4.	Contenidos.....	5
4.1.	Requisito de información: Implementación de la entidad “Project”	5
4.1.1.	Descripción del requisito.....	5
4.1.2.	Análisis detallado	5
4.1.3.	Decisiones tomadas	5
4.1.4.	Enlace de validación	6
5.	Conclusiones.....	7
6.	Bibliografía.....	8

1. Resumen ejecutivo

El análisis se centró en el requisito de información para la implementación de la entidad "Project". Este requisito incluye la definición de los datos que deben almacenarse para cada proyecto. Además, se valoraron aspectos clave como la gestión de errores fatales, la representación del costo del proyecto y la relación con las historias de usuario. Para ello, se tomaron decisiones consensuadas con el cliente, como el uso de atributos booleanos para indicar errores fatales y la publicación del proyecto, así como la representación del costo como un entero positivo. Del mismo modo, se decidió implementar una entidad intermedia para relacionar proyectos e historias de usuario.

Estas decisiones se tomaron tras evaluar diversas opciones y considerar consultas realizadas al cliente a través del foro. Los detalles adicionales sobre las decisiones y sus fundamentos se encuentran detallados en el informe.

2. Tabla de revisión

Nº Revisión	Fecha	Descripción
1	8/03/2024	Versión inicial.

3. Introducción

El presente informe aborda el requisito de información para la implementación de la entidad "Project". El requisito especifica los datos que deben almacenarse para cada proyecto, incluidos el código, título, resumen, indicación de errores fatales, costo y un enlace opcional. La sección 4 detalla el análisis realizado para cumplir con estos requisitos, centrándose en la gestión de errores fatales, la representación del costo del proyecto y la relación con las historias de usuario. Cabe destacar que las decisiones tomadas se basan en una evaluación de diferentes opciones y consultas realizadas al cliente a través del foro. Para finalizar, se concluye con una visión general de las decisiones clave y su impacto en la implementación.

4. Contenidos

4.1. Requisito de información: Implementación de la entidad “Project”

4.1.1. Descripción del requisito

Un proyecto agrega varias historias de usuarios obtenidas por el mismo administrador. El sistema debe almacenar los siguientes datos sobre ellos: un código (patrón “[A-Z]{3}-[0-9]{4}”, no en blanco, único), un título (no en blanco, de menos de 76 caracteres), un resumen (no en blanco, de menos de 101 caracteres), una indicación de si tiene errores fatales, por ejemplo, pánicos, un costo (positivo o cero) y un enlace opcional con más información. El sistema debe rechazar los proyectos que contengan errores fatales.

4.1.2. Análisis detallado

Para implementar la entidad “Project” adecuadamente, es fundamental asegurarnos de que el diseño cumpla con los requisitos especificados, al tiempo que proporciona flexibilidad y robustez. El análisis detallado se centra en tres aspectos principales: la gestión de errores fatales y la publicación de proyectos, la representación del costo del proyecto y la relación con las historias de usuario.

La capacidad de identificar si un proyecto contiene errores fatales puede llevarse a cabo mediante un atributo de tipo *boolean*, al igual que la implementación de un atributo *boolean* para indicar si el proyecto está o no en modo borrador o completado o no completado. De esta forma, permite al cliente una visualización clara del estado de los dos atributos, así como de una forma simplificada para validar la restricción de que un proyecto con errores fatales no pueda ser publicado.

Por otro lado, a la hora de implementar el costo del proyecto puede considerarse el tipo de dicho atributo como *int*, *double* o *Money*. Si se considera que el costo del proyecto está asociado con el coste estimado en horas de las historias de usuario es conveniente declarar el atributo como el primer tipo propuesto para mantener la misma métrica en ambos aspectos. Por otro lado, puede tratarse como un *double* para trabajar con números reales o para indicar las horas (parte entera) y minutos (parte real), aunque pueden ocasionar dudas y conflictos si se considera que los minutos superan el valor 60 en la parte real. Por último, con el tipo *Money* se obtendría una representación más clara en cuanto al tipo de moneda y valor pero causaría una diferencia de tipos respecto al valor del coste estimado en horas de las historias de usuario.

Por último, para relacionar las entidades “Project” y “UserStory” es necesario considerar si una historia de usuario puede ser compartida o no por varios proyectos. En el caso de que una misma historia de usuario pueda formar parte de dos proyectos diferentes, se evitaría guardar información duplicada en la base de datos. Al mismo tiempo, cabe destacar la implementación manual (garantizando la accesibilidad a sus datos) de una entidad intermedia que relacione dichas entidades, teniendo una relación unidireccional *ManyToOne* hacia cada una de las entidades.

4.1.3. Decisiones tomadas

Tras evaluar diferentes opciones para cada uno de los puntos anteriores, se han tomado las siguientes decisiones:

- Indicación de errores fatales. Se ha decidido implementar un atributo de tipo *boolean* “hasFatalErrors” para marcar si el proyecto contiene errores fatales. Esta decisión permite una fácil identificación de los proyectos problemáticos y simplifica la lógica de validación para la publicación.
- Publicación de proyectos. Para garantizar que no se publiquen proyectos con errores fatales, se ha introducido un atributo de tipo *boolean* “isPublished”. Este enfoque facilita el control sobre qué proyectos son visibles puesto que ya han sido publicados, es decir, han pasado de un modo borrador o un estado no completado a uno finalizado previa comprobación de que el proyecto no contenga errores fatales.
- Costo del proyecto. Se ha optado por representar el costo del proyecto mediante un atributo de tipo *int* “cost” que esté limitado a valores cero o positivos. Esta decisión asegura que el costo se mantenga en unidades discretas (horas enteras) al igual que el coste estimado de las horas de cada una de las historias de usuario que componen un proyecto, simplificando el cálculo de recursos y horas estimadas.
- Relación con las historias de usuario. Se ha decidido implementar una entidad intermedia entre “Project” y “UserStory” que permita guardar qué historias de usuario están asociadas con qué proyectos. De esta forma, se garantiza que una misma historia de usuario pueda estar en varios proyectos, evitando almacenar información duplicada en la base de datos.

4.1.4. Enlace de validación

Nótese que las consultas realizadas al cliente a través del foro se han realizado por parte de otros alumnos que no forman parte del grupo de trabajo, pero se han tenido en cuenta a la hora de la implementación de los requisitos de información solicitados:

Información relacionada con la publicación de proyectos en el siguiente [link](#).

Información relacionada con el costo del proyecto en el siguiente [link](#).

Información relacionada con la relación de la entidad “Project” con las historias de usuario en el siguiente [link](#).

5. Conclusiones

El análisis detallado ha permitido definir con claridad los requisitos de información para la entidad "Project". Las decisiones tomadas garantizan la adecuada representación de los datos, con un enfoque en la gestión eficiente de errores fatales, la precisión en la representación del costo del proyecto y la flexibilidad en la relación con las historias de usuario. Estas decisiones se basaron en la evaluación de diversas opciones y consultas al cliente. En conjunto, proporcionan una sólida base para la implementación de la entidad "Project" de manera robusta y flexible.

6. Bibliografía

[Foro: Tutorials](#)