

# DP2 2023-2024

## Analysis report

### Acme-SF-D03

Repositorio: <https://github.com/andpizcer/DP2-C2.032>

#### Miembro:

- Pizzano Cerrillos, Andrés, [andpizcer@alum.us.es](mailto:andpizcer@alum.us.es)

**Tutor:** Patricia Jiménez Aguirre

GRUPO C2.032 (G3 teoría)

26/04/2024

## Historial de versiones

Fecha	Versión	Descripción de los cambios	Sprint
15/02/2024	V1	Creación del documento.	1
07/02/2024	V2	Requisitos segunda entrega	2
26/04/2024	V3	Requisitos tercera entrega	3
20/06/2024	V4	Conclusiones	4

## Índice

<b>Resumen ejecutivo.....</b>	<b>4</b>
<b>Introducción.....</b>	<b>4</b>
<b>Entregable01 .....</b>	<b>4</b>
<b>Entregable02 .....</b>	<b>4</b>
<b>Entregable03 .....</b>	<b>7</b>
<b>Entregable04 .....</b>	<b>8</b>
<b>Conclusiones .....</b>	<b>8</b>
<b>Bibliografía .....</b>	<b>9</b>

## 1. Resumen ejecutivo

El presente documento detalla aquellos requisitos que han necesitado análisis a lo largo del desarrollo del proyecto.

## 2. Introducción

Con el fin de poner en práctica lo anteriormente mencionado, se agruparán los requisitos por entregables. En cada uno de los apartados se realizará un análisis de los requisitos y las decisiones asociadas para su implementación. Posteriormente, se hará referencia a estos procesos y las conclusiones obtenidas en el apartado de Conclusiones.

## 3. Entregable 01

Intencionalmente en blanco.

## 4. Entregable 02

### **ANÁLISIS 1**

#### **2) Creación de la entidad Project**

*“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.”*

#### **3) Creación de la entidad UserStory**

*“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.”*

### **PROBLEMA**

Tras la lectura de los requisitos, se observa que las entidades UserStory y Project tienen una relación de muchos a muchos (ManyToMany). Según la metodología empleada, esta multiplicidad debe evitarse. Las relaciones de muchos a muchos pueden complicar la gestión de datos y las consultas en la base de datos, además de introducir potenciales problemas de integridad y consistencia.

### **SOLUCIÓN**

Para resolver este problema, se ha decidido crear una tabla intermedia llamada Assignment. Esta tabla intermedia tendrá relaciones de muchos a uno (ManyToOne) con las entidades Project y UserStory, lo que permitirá mantener una estructura de datos más manejable y eficiente.

Además, dado que tanto las entidades Project como UserStory deben ser gestionadas por un gestor, ambas tendrán una relación de muchos a uno (ManyToOne) con el rol de Manager. Esto asegura que cada proyecto y cada historia de usuario estén claramente asignados a un único gestor, facilitando la trazabilidad y responsabilidad en la gestión de proyectos.

## ENLACE DE VALIDACIÓN



PATRICIA JIMENEZ AGUIRRE

RE: D03 - Student 1 [Actividad 7 - Operaciones con Historias de usuario]

## ANÁLISIS 2

### 2) Creación de la entidad Project

*“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.”*

## PROBLEMA

Tras la lectura del requisito, me doy cuenta de que el costo del proyecto crea ambigüedad y barajo las siguientes opciones de implementación:

- **Opción 1:** Implementarlo como un tipo int.  
Ventajas: Fácil de implementar e intuitivo.  
Desventajas: Tendría que existir un consenso para utilizar en todos los proyectos un solo tipo de divisa. Todos los usuarios tendrían que convertir la divisa en la que está financiada su proyecto a la divisa del sistema.
- **Opción 2:** Implementarlo con el DataType Money.  
Ventajas: Resuelve las desventajas de la opción 1.  
Desventajas: Añade complejidad y debemos añadir restricciones personalizadas para que no tome valores incoherentes.

## SOLUCIÓN

Se ha llegado a la conclusión de que lo correcto es implementar la opción 2. El atributo costo, aunque no se especifica, se debe modelar con el tipo de dato Money, al referirse al costo económico del proyecto. Esta decisión se basa en la necesidad de manejar valores monetarios con precisión, incluyendo decimales, para reflejar correctamente los gastos y presupuestos del proyecto. Aunque añade cierta complejidad, esta es manejable y las restricciones personalizadas necesarias para asegurar la coherencia de los valores monetarios pueden ser implementadas y validadas en los servicios.

## ENLACE DE VALIDACIÓN



JOSE GONZALEZ ENRIQUEZ 🟢

RE: Incoherencia con atributo cost en Projects.

## ANÁLISIS 3

### 3) Creación de la entidad UserStory

*"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."*

## PROBLEMA

Tras la lectura del requisito, me doy cuenta de que el costo estimado de una historia de usuario crea ambigüedad y barajo las siguientes opciones de implementación:

- **Opción 1:** Implementarlo como un tipo int.  
Ventajas: Fácil de implementar e intuitivo. Solo admitiría horas.  
Desventajas: No tendría sentido si estamos trabajando con valores que pueden tomar decimales, como, por ejemplo: una hora y media.
- **Opción 2:** Implementarlo como un tipo double.  
Ventajas: Resuelve las desventajas de la opción 1.  
Desventajas: No se suelen usar decimales al estimar horas y sería innecesario en tal caso contar con valores decimales.

## SOLUCIÓN

Se ha llegado a la conclusión de que lo correcto es implementar la opción 1. El atributo costo estimado en horas, en la práctica, se establece como un entero, ya que las estimaciones se redondean y nunca se tienen en cuenta minutos o segundos. Debido a ello, se decide implementar el costo estimado como un tipo de dato entero. Esta decisión permite mantener la simplicidad en la implementación y asegura que las estimaciones sean coherentes con las prácticas habituales en la gestión de proyectos. Además, evita la complejidad innecesaria de manejar decimales para valores de tiempo que no suelen ser utilizados en la estimación de horas.

## ENLACE DE VALIDACIÓN



RAFAEL CORCHUELO GIL 🟢

RE: [Análisis] D02-Student#1-003

## 5. Entregable 03

### **ANÁLISIS 1**

#### **6) Operations by managers on projects**

- *List the projects that they have created.*
- *Show the details of their projects.*
- *Create, update, or delete their projects. Projects can be updated or deleted as long as they have not been published. For a project to be published, it must have at least one userstory, and all its user stories must have been published. Moreover, it must not have any fatal errors.*

#### **7) Operations by managers on user stories**

- *List the user stories in their projects.*
- *Show the details of their user stories.*
- *Create and publish a user story.*
- *Update or delete a user story as long as it is not published.*

### **PROBLEMA**


El principal problema encontrado en el desempeño de la tarea fue el concepto de publicar, concepto que el cliente no había introducido hasta ahora. Poniendo en contexto, no se puede publicar un proyecto si no tiene historias de usuario asociadas, y si las tiene, éstas deben estar publicadas también.

### **SOLUCIÓN**

Para solventar esta problemática, he decidido añadir la propiedad "published" a ambas entidades y he creado un controlador y servicio para crear "Assignments" (tabla ManyToMany que relaciona las entidades). Esto cubre los requisitos especificados porque, en primer lugar, permite asociar historias de usuario a proyectos y, por otro lado, gracias a la definición de la propiedad "published", podemos introducir las restricciones necesarias en el servicio de publicación para que no se puedan publicar proyectos sin tener historias de usuario asociadas ya publicadas. Esta solución garantiza que los proyectos solo se publiquen cuando cumplan con todos los requisitos necesarios, asegurando la integridad y coherencia de los datos.

### **ENLACE DE VERIFICACIÓN**



**PATRICIA JIMENEZ AGUIRRE** 

**RE: [Análisis] D03 Alternativas con el draft mode en las clases secundarias**

### **ANÁLISIS 2**

#### **8) Operations by managers on manager dashboards**

- *Show their manager dashboards.*

## PROBLEMA

Tras la lectura del requisito, y el previo modelado de la entidad ManagerDashBoard, me doy cuenta de que tengo que realizar operaciones de máximo, mínimo, media, etc., del costo de proyectos que pueden estar financiados con diferentes divisas. Se barajan las siguientes opciones de implementación:

- **Opción 1:** Implementarlo sin tener en cuenta la divisa.  
Ventajas: Fácil de implementar. Aplicaríamos las métricas a todos los proyectos.  
Desventajas: Incoherente, puesto que las divisas tienen diferente valor y las métricas dejarían de tener sentido para el Manager.
- **Opción 2:** Agrupar los proyectos por el tipo de divisa y crear las métricas de estos proyectos  
Ventajas: Resuelve las desventajas de la opción 1. Las métricas serían coherentes.  
Desventajas: No tiene una desventaja real, no añadiría una complejidad significativa. Realizando un filtrado sobre las consultas agruparíamos los proyectos y realizaríamos las métricas sobre éstos.

## SOLUCIÓN

Se ha llegado a la conclusión de que lo correcto es implementar la opción 2. Para resolver este problema, se ha decidido agrupar los proyectos por divisa y realizar las estadísticas para los diferentes grupos de proyectos. Se ha utilizado un Map para organizar los proyectos según su divisa y posteriormente se ha cargado en el Frontend, que se ha encargado de recorrerlo y mostrarlo. Esta solución asegura que las métricas sean coherentes y reflejen adecuadamente el valor de los proyectos en sus respectivas divisas, manteniendo la integridad y utilidad de las estadísticas en el sistema.

## ENLACE DE VERIFICACIÓN



**PATRICIA JIMENEZ AGUIRRE** 🟢

**RE: D03 - Operaciones relacionadas a Money**

## 6. Entregable 04

Intencionalmente en blanco.

## 7. Conclusiones

Después de analizar los diferentes requisitos que han presentado desafíos, se ha determinado que no se requirió análisis para ninguna de las entregas 1 y 4. En cuanto a los entregables 2 y 3, el enfoque del análisis se centró en las entidades Assignment, UserStory y Project.

Para las entidades UserStory y Project, se ha decidido utilizar una entidad intermedia denominada Assignment debido a la existencia de una relación ManyToMany entre ellas.

Respecto a la entidad Project, se ha establecido que el costo de los proyectos será tratado utilizando el DataType Money, considerado apropiado para el dominio del problema y garantizando precisión.



En el caso de la entidad UserStory, se concluyó que el costo estimado debe ser modelado como un entero, dado que en la práctica las estimaciones de esfuerzo para una historia de usuario se expresan típicamente en horas, sin considerar minutos o segundos.

Además, tanto para la entidad Project como para UserStory, se ha argumentado que sería necesario añadir la propiedad adicional "published", la cual facilitaría la publicación de estas entidades y permitiría establecer restricciones tanto en los servicios como en el Frontend.

Finalmente, se ha determinado que, para asegurar la coherencia en las métricas de los DashBoard del rol Manager, los proyectos pueden estar financiados con diferentes divisas, y lo más adecuado será crear y agrupar las métricas de los proyectos según cada tipo de divisa.

Esta conclusión garantiza una implementación coherente y eficaz de los requisitos identificados, asegurando que el sistema sea capaz de gestionar adecuadamente las relaciones, los costos y las publicaciones dentro del contexto definido.

## 8. Bibliografía

[1] Discussion Board, Foro: Tutorials. En Enseñanza Virtual, Diseño y Pruebas II.