

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2023 – 2024

Reporte de análisis estudiante 1

David Fuentelsaz Rodríguez

davfuerod@alum.us.es

Repositorio: <https://github.com/DP2-C1-013/Acme-SF-D01-24.1.0>

Fecha	Versión
07/03/2024	v1.0

1. Resumen.....	2
2. Control de versiones.....	3
3. Introducción.....	3
4. Contenidos.....	4
4.1. Modificación del menú de anónimos.....	4
5. Conclusión.....	4
6. Bibliografía.....	4

1. Resumen

Este reporte tiene como objetivo realizar un listado del análisis de los distintos requisitos individuales asignados al estudiante #1 para el proyecto de la asignatura. Para cada requisito se dará información sobre las conclusiones a las que ha llegado el equipo y de las decisiones tomadas.

2. Control de versiones

Nº de revisión	Fecha	Descripción
1.0	14/02/2024	Desarrollo inicial del reporte de análisis
2.0	07/03/2024	

3. Introducción

El reporte de análisis está centrado en el análisis de los requisitos asignados al estudiante #4. En esta versión del documento corresponde al entregable 2 del proyecto, pero siguen apareciendo los análisis de entregables anteriores. Se analizarán los requisitos más complejos, centrándose principalmente en aquellos elementos que hayan requerido la valoración de distintas opciones. Los requisitos triviales no serán tratados o se hará de forma muy superficial si hicieran falta para el desarrollo del análisis de otro más complejo. Todos los requisitos han sido revisados en las sesiones de Follow Up.

El contenido del reporte de análisis está estructurado en los siguientes puntos:

1. Resumen ejecutivo: resumen muy general del contenido principal del reporte.
 2. Control de versiones del documento.
 3. Introducción: a diferencia del resumen ejecutivo, se realiza un resumen más detallado del contenido principal al mismo y se explica la estructura del documento.
 4. Contenidos: punto principal del reporte en el que se entra con todo detalle en el análisis de los distintos requisitos.
 5. Conclusión.
 6. Bibliografía.
- 1.

4. Contenidos

4.1 Entregable 1

En el entregable 1 no ha sido necesario el análisis de ninguno de los requisitos debido a la trivialidad de los mismos.

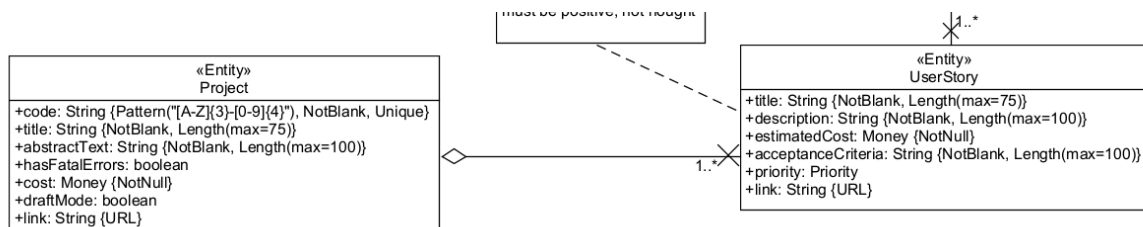
4.2. Entregable 2

4.2.1 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.

Este requisito de información implica la creación de la entidad *Project*, la cual se relaciona con la entidad *UserStory* mediante un conglomerado. Se planteó la cuestión sobre qué tipo de conglomerado sería más óptimo en este caso. Tras un análisis, se consideró que la mejor opción era modelar esta relación como una composición, principalmente debido a dos ventajas sobre la agregación. La primera ventaja es la flexibilidad, ya que los objetos relacionados pueden existir de manera independiente y pueden ser agregados o eliminados dinámicamente. Además, este tipo de relación presenta menos acoplamiento, lo que reduce las dependencias y facilita la modificación y mantenimiento del sistema. La única desventaja encontrada fue que las relaciones de agregación presentan una complejidad adicional respecto a las de composición. Sin embargo, debido a que las ventajas superan a las desventajas, se ha decidido modelar esta relación como una de agregación.

También se ha considerado que, cualquier instancia de la entidad proyecto, debe tener, como mínimo, una historia de usuario asociada.

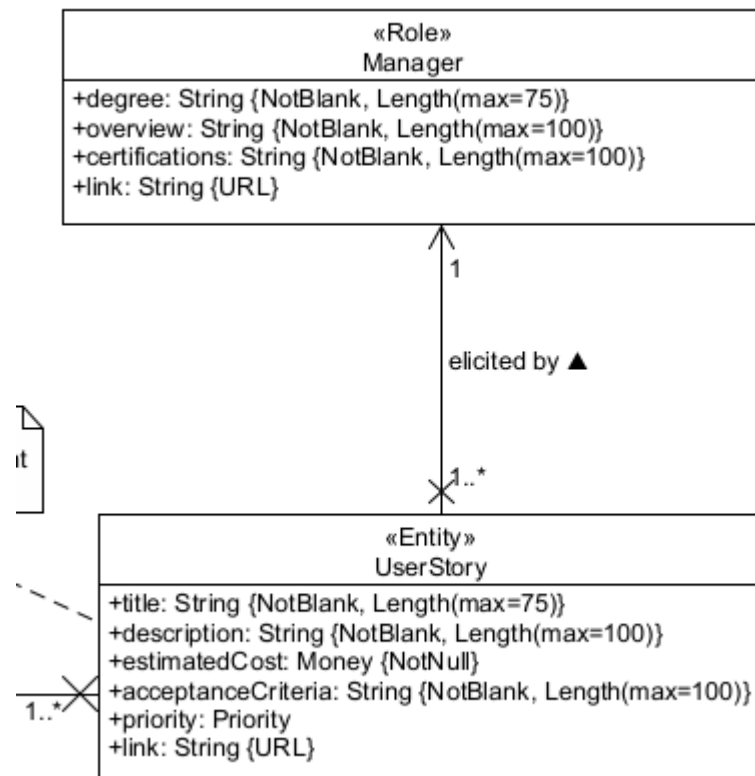


Para modelar los atributos de esta entidad se ha optado por incluir el atributo *draftMode* para indicar si la agregación que asocia las entidades *Project* y *UserStory* está completa, y representar el atributo *cost* con el tipo *Money*, en lugar de cualquier otro tipo numérico. Este tipo de dato personalizado viene incluido en el framework y permite expresar cantidades decimales y especificar el tipo de moneda usada.

4.2.2 Entidad User Story

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.

Este requisito de información implica la creación de la entidad *UserStory*, la cual se relaciona con la entidad *Project* mediante un conglomerado (como se razonó con anterioridad), y con el rol de *Manager*, en una relación many-to-one unidireccional, la cuál es la más óptima por razones de simplicidad, evitación de redundancia de la información y reducción del riesgo de dependencia circular. Se ha considerado que, el manager, tenga que tener al menos una historia de usuario elicitada, ya que, en caso contrario, el manager no tendría ningún proyecto asociado.



Dentro de los atributos, cabe destacar la presencia del enumerado *Priority*, cuyos posibles valores son *MUST*, *SHOULD*, *COULD* y *WON'T*, y, de nuevo, el modelado de la propiedad *estimatedCost* con el tipo *Money*.

4.2.3 ManagerDashboard

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**.

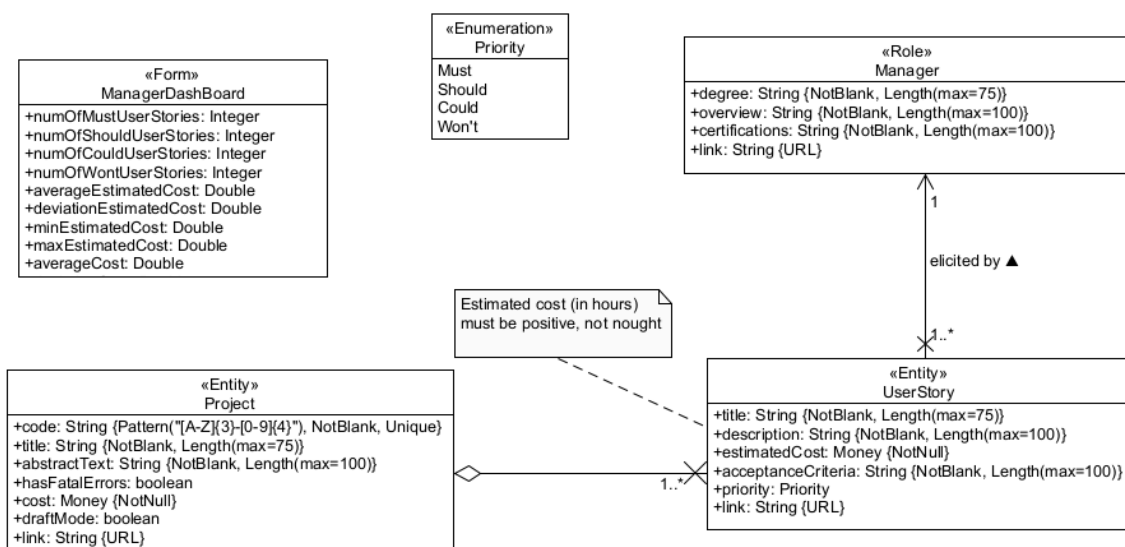
Este requisito expresa que el sistema debe manejar datos derivados de de las entidades *Project* y *UserStory*. Estos datos que se piden serán calculados mediante consultas relativamente simples a la base de datos, por lo que no será necesario persistir los datos calculados. Estas consultas se implementarán sobre el repositorio.

4.2.4 Rol Manager

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.

Este requisito implica implementar un nuevo rol en el sistema que represente a los managers de los proyectos. Por ello, se implementará usando una entidad que extienda a *AbstractRole*. Está relacionada con la entidad *UserStory* mediante una relación one-to-many unidireccional, como se mencionó anteriormente.

4.2.5 Diagrama UML resultante del análisis



5. Conclusión

Este documento está actualmente en su segunda versión, correspondiente al segundo entregable. Los requisitos recogidos en esta versión han sido más complejos que los de la anterior por lo que han requerido una análisis en profundidad.

6. Bibliografía

Intencionadamente en blanco.