

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

D02 – Analysis Report



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas II

Curso 2023 – 2024

Fecha	Versión
08/03/2024	v1r1

Grupo de prácticas: C1.002	
Autores	Correo Corporativo
Rubén Pérez Garrido	rubpergar@alum.us.es

Link repositorio: <https://github.com/Joserra24/Acme-SF-D02-24.1.0>



Índice

1. Tabla de versiones.....	3
2. Resumen ejecutivo.....	4
3. Introducción	5
4. Contenido	6
5. Conclusiones.....	9
6. Bibliografía.....	10



1. *Tabla de versiones*

Fecha	Versión	Descripción
02/03/2024	v1r0	Creación del documento
08/03/2024	v1r1	Entrega



2. Resumen ejecutivo

El reporte indica que se han cumplido todos los puntos establecidos a cumplir por el miembro del equipo cuyas tareas son relativas a las del student #2. Aunque se ha progresado en las tareas designadas, no se ha llegado a tomar ninguna decisión importante hasta ahora.



3. Introducción

En este documento, se realiza un listado para el registro de análisis. Para cada entrada, se especifica el requisito sobre el que se han tenido complicaciones, así como las conclusiones y las decisiones que se tomaron con respecto a dicho requisito.

Este proceso de registro y análisis es fundamental para asegurar la integridad y la calidad de nuestro trabajo. Al documentar claramente cada paso y las acciones tomadas, no solo se cumple con los requisitos establecidos, sino que también se proporciona una base sólida para futuras revisiones y mejoras.

Al mantener un registro detallado, podemos identificar áreas de mejora y tomar decisiones informadas sobre cómo abordar cualquier desafío o requisito específico. Además, este enfoque nos permite comunicar de manera efectiva nuestros hallazgos y recomendaciones a todas las partes interesadas involucradas en el proceso.

Se han realizado en esta entrega las tareas:

- Requisito 2 Mandatory - Entidad contract
- Requisito 3 Mandatory - Entidad Progress log
- Requisito 4 Mandatory - Dashboards para el rol client
- Requisito 5 Mandatory - Sample data
- Requisito 13 Supplementary - Rol client
- Requisito 14 Supplementary - Modelo de dominio UML
- Requisito 15 Supplementary - Reporte de análisis
- Requisito 16 Supplementary - Reporte de progreso y planificación

4. Contenido

Requisito 2: *A contract is one or several agreements between the stakeholders involved in the development 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 instantiation moment (in the past), a provider name (not blank, shorter than 76 characters), a customer name (not blank, shorter than 76 characters), some goals (not blank, shorter than 101 characters), and a budget (less than or equal to the corresponding project cost).*

Problema	Solución
La anotación @Past podría ocasionar problemas en el atributo ‘instantiationMoment’	La validación para la fecha de este atributo se llevará a cabo utilizando la anotación @PastOrPresent. Esto se debe a que Java actualiza las fechas cada 1 milisegundo, lo que podría generar complicaciones al determinar si la fecha de instanciación del contrato ha ocurrido en el pasado.
Doble comprobación de un campo nulo.	Se eliminan las anotaciones @NotNull en aquellas propiedades que ya tengan implementadas las anotaciones @NotBlank, puesto que estas últimas ya cubren esa validación.
Tipo confuso para la propiedad budget.	En este atributo, el cliente ha cometido un error de redacción ya que el atributo cost se mide en horas enteras mientras que el budget se mide en cantidad de dinero. Por esto, el tipo designado para esta propiedad será Money (validado en el foro en la cadena “[Análisis] D02-Student#2-002”).

Requisito 3: *Every contract has an evolution that is composed of progress logs. The system must store the following data about them: a record id (pattern “PG-[A-Z]{1,2}-[0-9]{4}”, not blank, unique), a percentage of completeness (positive), a comment on the progress (not blank, shorter than 101 characters), a registration moment (in the past), and a responsible person for the registration (not blank, shorter than 76 characters)*

Problema	Solución
La anotación @Past podría ocasionar problemas en el atributo ‘registrationMoment’	La validación para la fecha de este atributo se llevará a cabo utilizando la anotación @PastOrPresent. Esto se debe a que Java actualiza las fechas cada 1 milisegundo, lo que podría generar complicaciones al determinar si la fecha de instanciación del contrato ha ocurrido en el pasado.
Doble comprobación de un campo nulo.	Se eliminan las anotaciones @NotNull en aquellas propiedades que ya tengan implementadas las anotaciones @NotBlank, puesto que estas últimas ya cubren esa validación.
Uso de @NotNull redundante en la propiedad	En aquella clase que use un tipo wrapper y la anotación @NotNull, se sustituirá dicho tipo por el mismo, pero en su



completeness.

forma primitiva.

Requisito 4: *The system must handle client dashboards with the following data: total number of progress logs with a completeness rate below 25%, between 25% and 50%, between 50% and 75%, and above 75%; average, deviation, minimum, and maximum budget of the contracts.*

Problema	Solución
El tipo empleado en los contadores de la completitud de los progress log no tienen en cuenta los valores nulos.	Se ha cambiado el tipo de los contadores de los progress log con un cierto porcentaje de completitud de Integer (wrapper) a int (primitivo) para salvarnos de los valores nulos y comenzar siempre estos contadores desde el 0.
Tipo confuso para la propiedad budget.	En este atributo, el cliente ha cometido un error de redacción ya que el atributo cost se mide en horas enteras mientras que el budget se mide en cantidad de dinero. Por esto, el tipo designado para los presupuestos de los contratos será Money (validado en el foro en la cadena "[Análisis] D02-Student#2-002").

Requisito 13: *There is a new project-specific role called clients, which has the following profile data: identification (pattern "CLI-[0-9]{4}", not blank, unique), a company name (not blank, shorter than 76 characters), type (not blank, either "company" or "individual"), an email (not blank), and an optional link with further information.*

Problema	Solución
Doble comprobación de un campo nulo.	Se eliminan las anotaciones @NotNull en aquellas propiedades que ya tengan implementadas las anotaciones @NotBlank, puesto que estas últimas ya cubren esa validación.



Diseño y Pruebas II

Rubén Pérez Garrido | Student #2

C1.002

Analysis report



5. Conclusiones

Durante el proceso de desarrollo de esta segunda entrega, nos encontramos con ciertas incertidumbres que surgieron en el camino. Sin embargo, gracias a la colaboración de compañeros en el foro, quienes compartieron los obstáculos y a los valiosos consejos proporcionados por el profesor sobre ciertas anotaciones en el código, pudimos abordar y superar estas dificultades.

Cada solución se ha implementado meticulosamente en el código de esta entrega específica. Además, es importante destacar que el alumno ha tomado nota de estas soluciones, asegurándose de tenerlas en consideración para futuras entregas que puedan presentar desafíos similares. Este proceso de aprendizaje continuo y colaborativo nos permite no solo resolver problemas inmediatos, sino también crecer y mejorar constantemente en nuestro trabajo.



6. *Bibliografía*

- 08 Annexes – Material proporcionado en la asignatura *Diseño y Pruebas II* por la Universidad de Sevilla.
- Foro: Tutorials – Foro proporcionado por la asignatura *Diseño y Pruebas II* para resolver dudas relativas al proyecto.