

Analysis report



Diseño y pruebas II

Sprint 4

Versión	Fecha	Descripción
1.0	15/05/2024	Creación del documento.
1.1	17/04/2024	Adición del análisis de los requisitos de entrega 2
1.2	23/05/2024	Revisión del documento y adición de los análisis de la entrega 3 y 4
1.3	25/05/2024	Revisión final y adición de la conclusión

<https://github.com/DP2-c1-028/Acme-SF-D04>

Fecha 15/05/2024

Preparado por:
Benito Hidalgo, Daniel (C1.028) (danbenhid@alum.us.es)

Índice

Índice.....	2
Introducción.....	2
1. Análisis de los requisitos.....	2
Entrega 1:.....	2
Entrega 2:.....	3
Entrega 3:.....	5
Entrega 4:.....	7
Conclusiones.....	7

Introducción

En este documento se revisarán las diferentes decisiones de diseños realizadas a lo largo de todas las entregas individuales de este proyecto, describiendo en primer lugar el requisito a implementar, detallando las posibles alternativas a la resolución y justificando la elección de una u otra implementación.

La estructura del documento consiste en

1.Análisis de los requisitos

Entrega 1:

1. Modify the anonymous menu so that it shows an option that takes the browser to the home page of your favourite web site. The title must read as follows: “(id-number): (surname), (name)”, where “(id-number)” denotes your DNI, NIE, or passport number, “(surname)” denotes your surname/s, and “(name)” denotes your name/s.

Teniendo en cuenta la simpleza de este requisito, considero que no es necesario hacer un análisis en profundidad sobre la realización de esta tarea en concreto.

Entrega 2:

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).

Tras analizar el requisito y consultar en los foros, más exactamente la entrada “[Análisis Rangos de atributos no especificados ” de Rafael Castillo Cebolla, se llegó a la conclusión de que era conveniente añadir rangos para determinados atributos cuya definición no aclaraba si la limitación de los posibles valores de este era necesaria. Debido a que la introducción de estos rangos requería conocimientos que no se tenían en aquel momento se dejó por hacer para entregas futuras.

Gracias al feedback del profesor a lo largo de los follow up de la segunda entrega se vio que se podía mejorar la eficiencia del sistema introduciendo tipos primitivos (los cuales no permiten valores nulos y resultan ser menos pesados) en los atributos de las entidades donde las circunstancias y restricciones a aplicar lo permitieran (por ejemplo el atributo draftMode)

Respecto a otros aspectos de este requisito, no se ha visto necesario analizar nada más ya que se basa en la creación de la entidad, como se verá en entregas posteriores, se han añadido validaciones complejas en los servicios de las entidades que se explicarán en las siguientes entregas

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).

Siguiendo con la entidad secundaria, se realizó el mismo proceso detallado anteriormente, se revisó qué atributos podrían ser declarados con tipos primitivos y se realizó la sustitución.

Durante uno de los follow up se nos explicó que esta entidad también debía ser relacionada al rol de cliente, por lo que fue corregido una vez realizado el rol de Client. Al igual que con la entidad Contract, se han ido introduciendo validaciones complejas en los servicios, debido a que no se ha añadido nada nuevo en las propias entidades, el análisis de estas implementaciones se harán en sus respectivos requisitos

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.

Tras un primer análisis de este requisito se vió que no sería necesario tomar ninguna decisión de diseño en su realización, sin embargo en entregas posteriores obtuvimos el conocimiento necesario para mostrar y calcular los datos del dashboard y se tuvo que replantear su funcionamiento.

Debido a los conocimientos obtenidos en la entrega 3 y al feedback aportado por el profesor durante las sesiones de follow up, se llegó a la conclusión de que el dashboard necesitaría mostrar los datos relacionados al tipo money de forma que se expresaran en las distintas divisas del sistema.

Por ello se planteó un sistema en el que el dashboard mostraría los datos expresados solo en las divisas que han estado involucradas en el cálculo (por ejemplo, si se han creado contratos con un presupuesto en USD o GBP, el dashboard devolvería las estadísticas de los presupuestos en GBP y USD).

Para la realización de este sistema se usó una entidad de utilidad que se denominó como ClientMoneyStatistics que sustituiría a los atributos relacionados a cálculos de dinero en el dashboard (es decir, el dashboard ahora tiene como atributo un objeto de tipo clientMoneyStatistics el cual contiene los diversos cálculos que se realizan con los presupuestos de los contratos).

Este cambio fue comprobado por el profesor de prácticas durante una de las sesiones de follow up y le dio el visto bueno a la solución realizada.

5. Produce assorted sample data to test your application informally. The data must include two client accounts with credentials "client1/client1" and "client2/client2".

Para la realización de este requisito no fue necesario ningún análisis ya que la metodología propuesta en las sesiones de teoría fueron entendidas sin problema.

Entrega 3:

6. Operations by clients on contracts:

- List the contracts that they have created.
- Show the details of their contracts.
- Create, update, or delete their contracts. Contracts can be updated or deleted as long as they have not been published. For a contract to be published, the sum of all budgets cannot exceed the total cost of the corresponding project.

Tras un análisis del requisito y teniendo en cuenta la información que se tenía sobre las restricciones a aplicar se implementaron las siguientes soluciones:

Se introdujeron las validaciones y restricciones de rangos de atributos que se anotaron durante la primera entrega, todo esto fue realizado en los correspondientes servicios y fue revisado por el profesor en los follow ups.

Además de las validaciones dadas por la descripción del requisito (límites de tipo money, fechas y que el presupuesto no supere el coste del proyecto asociado), gracias al feedback recibido se anotaron algunas validaciones a implementar en la entidad de contract:

- Se añadió una validación en los servicios de creación, actualización y publicación del contrato para que se comprobara que el proyecto al que está asociado estuviera publicado.
- Se crearon validaciones en todos los servicios que permitían la modificación de los datos del contrato, con el fin de comprobar que el código del proyecto (que debía ser único) no se pudiera repetir, avisando al usuario con un mensaje de error.
- En todo servicio que permitía la modificación de los datos del contrato se introdujo una validación que comprobase que no se estaba haciendo uso de una divisa invalida, devolviendo un mensaje de error en caso de que se introdujese alguna, esto se realiza gracias a la introducción del sistema de divisas y creación de monedas que se nombrará más adelante, ya que este sistema surgió de la necesidad de resolver el requisito relacionado a dashboards de la entrega 3.
- Para el servicio de publicación se estableció gracias al feedback del profesor que en la condición de publicación del contrato (que la suma de todos los contratos asociados al proyecto no superará el coste de este) solo se tuvieran en cuenta para este cálculo aquellos contratos ya publicados. La implementación de esta validación se realiza principalmente gracias al método de repositorio creado que nos permite encontrar los presupuestos de aquellos contratos publicados que están asociados al mismo proyecto que el contrato a publicar. Una vez con los valores se llama al servicio de gestión de divisas creado y se realiza una conversión a la misma moneda(se ha elegido dólares como moneda estándar para realizar estas conversiones) y se suman todos los presupuestos junto al del contrato a publicar, una vez con el valor, se establece la condición de que la suma no supere al coste del proyecto y el error que debe lanzar en caso de que no se cumpla.

A parte de las validaciones, no fue necesario indagar más en los análisis realizados al requisito debido a la claridad de lo que se pedía y la información aportada durante las

sesiones de teoría sobre la creación de vistas y el resto de elementos necesarios en backend.

7. Operations by clients on progress logs:

- List the progress logs in their contracts.
- Show the details of their progress logs.
- Create and publish a progress log.
- Update or delete a progress log as long as it is not published

Una vez realizado el análisis de este requisito se ha visto la necesidad de añadir algunas validaciones con el fin de que el proceso de creación y publicación de esta entidad:

- La primera validación añadida en todos los servicios donde es posible alterar los datos del registro de progreso ha sido relacionada a la fecha, con el fin de no permitir que esta entidad tenga una fecha de creación previa a la creación de la entidad principal, contrato. Para el desarrollo de esta validación se ha hecho uso de un método de repositorio que devuelve el contrato al que está asociado el registro de progreso, una vez con el contrato se comparan las fechas y se establece la condición para que la fecha de creación del registro sea posterior al contrato, en caso de que no se cumpla se mostrará un mensaje de error al usuario.
- Siguiendo con las validaciones añadidas, para todo servicio donde es posible alterar los datos de la entidad se añadieron validaciones que comprobaran la unicidad del código establecido en la entidad (atributo recordId), así como que el contrato sobre el que se está creando/editando/publicando el registro de progreso este publicado.
- Por último, atendiendo a la naturaleza de esta entidad se añadieron validaciones con el fin de mantener un orden en la publicación de esta, Para lograr esto se introducen validaciones que comprueben que tanto la fecha como la completitud del registro que se quiere publicar sean mayores que los 2 mismos atributos correspondientes al registro de progreso más reciente (o el que tiene el mayor porcentaje de completitud, que en este caso, debido a cómo se han introducido las validaciones funciona de forma similar)
Para realizar estas validaciones, se tuvo en cuenta que debido a cómo se realizaba la validación relacionada a la completitud (una vez publicado un registro, el nuevo a publicar debe tener un porcentaje mayor), se podía aprovechar ese comportamiento para introducir una validación relacionada a la fecha asignada al registro que permitiese un proceso de publicación más coherente, impidiendo que ocurran casos en los que la completitud del nuevo registro es mayor, pero pertenece a una fecha anterior.

Con todo esto acabaríamos de cubrir el análisis realizado sobre este requisito, el resto de implementaciones referentes al requisito no han resultado complejas ni han causado problemas, por ello su exclusión del documento.

8. Operations by clients on client dashboards:

- Show their client dashboards.

La realización de este requisito no ha necesitado un análisis exhaustivo debido a la información que se tenía sobre la implementación gracias a los proyectos de ejemplo. Si bien la tarea se entendió correctamente, durante uno de los follow ups se dió feedback referente a un detalle a implementar en los dashboards, no mostrar un 0 cuando no se pueda calcular un dato, ya que esto podría llevar a confusiones del cliente.

Entrega 4:

9. Produce a test suite for Requirements #6 and #7.

La realización de este requisito no ha necesitado un análisis debido a su complejidad y la información aportada por el profesorado en la teoría y proyectos de ejemplo, si bien se han tenido problemas a lo largo de la realización de este, la raíz de estos no procedía de un mal entendimiento del requisito sino de errores en la gestión de este.

10. Produce a testing report.

Para la realización de este requisito no ha sido necesario realizar un análisis ya que se tenía bastante información aportada por el profesor durante los follow ups y las transparencias de teoría

Las actividades suplementarias individuales no se han añadido a este documento debido a que ninguna ha presentado ninguna dificultad, ya que gran parte de estas han sido documentación o actividades muy relacionadas a los requerimientos obligatorios, lo que ha permitido aplicar lo aprendido con estos.

Conclusiones

La creación de este documento ha sido una buena oportunidad para repasar la correcta creación de los requisitos obligatorios y suplementarios individuales, como se ha podido leer a lo largo del documento, en términos generales las mayores dificultades se han producido durante la entrega 3, ya que esta consistió en la implementación de gran parte de la funcionalidad del proyecto, es por ello que ha sido necesario dedicar un gran tiempo solo a entender cómo debería funcionar el sistema, desde mi perspectiva, creo que la mayoría de estos requisitos han sido validados, o al menos, revisados por uno de los distintos profesores de teoría o prácticas.

Pero más importante considero que con este documento he podido comprobar la magnitud del proyecto que se estaba realizando y ver lo mucho que se ha aprendido a lo largo de la realización de este.