

DP2 2023-2024

Analysis report

Acme-SF-D04

Repositorio: <https://github.com/andpizcer/DP2-C1.032>

Miembro:

- Avilés Carrera, Francisco, fraavicar1@alum.us.es

Tutor: Patricia Jiménez Aguirre

GRUPO C1.032 (G3 teoría)

27/05/2024

Historial de versiones

Fecha	Versión	Descripción de los cambios	Sprint
15/02/2024	V1	<ul style="list-style-type: none">Creación del documento.	1
06/03/2024	V2	<ul style="list-style-type: none">Análisis de los requisitos del entregable 2.	2
25/04/2024	V3	<ul style="list-style-type: none">Análisis de los requisitos del entregable 3.	3
25/05/2024	V4	<ul style="list-style-type: none">Conclusiones.	4

Índice

1. Resumen ejecutivo	4
2. Introducción	4
3. Entregable 01	4
4. Entregable 02.....	4
5. Entregable 03.....	6
6. Entregable 04.....	9
7. Conclusiones	9
8. Bibliografía	9

1. Resumen ejecutivo

En este documento se detallarán los requisitos individuales que así lo requieran. Está organizado en entregables, conteniendo cada uno de ellos los requisitos que necesiten análisis.

2. Introducción

El informe de análisis contendrá registros específicos que aborden requisitos particulares, detallando la copia literal del requisito, conclusiones del análisis y decisiones tomadas para abordar el requisito. Además, podrá tener un enlace a la validación realizada por un profesor en caso de que sea necesario.

3. Entregable 01

Intencionalmente en blanco.

4. Entregable 02

REQUISITO ANALIZADO N°1

[D02-S02-13]: 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: El atributo en cuestión que me genera confusión sobre como implementarlo es **type**. Esto se debe a que a priori es un atributo Enumerado, ya que sólo puede tomar los valores COMPANY o INDIVIDUAL. Por ello, la restricción “not blank” es una restricción que no puede ir con un atributo de tipo Enumerado.

DECISIÓN TOMADA: Tras el análisis anterior, como realmente ese atributo sólo puede tomar los dos valores mencionados, la mejor decisión sería implementar el tipo Enumerado y añadirle la restricción “not null”. Con esto, me aseguro de que ese atributo sólo puede tomar esos dos valores únicamente (nulo no). El requisito quedaría redactado de la siguiente manera:

[D02-S02-13]*: 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 null, either “company” or “individual”)*, **an email** (not blank), and an **optional link** with further information.

REQUISITO ANALIZADO N°2

[D02-S02-02]: 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: El atributo en cuestión que me genera confusión sobre como implementarlo es **budget**. Las opciones valoradas:

- **Opción 1:** Implementarlo con el tipo int.
Ventajas: Fácil de implementar e intuitivo.
Desventajas: Únicamente se puede tener en cuenta un tipo de moneda, además de que habría que agregar demasiadas restricciones sin saber cuánto es el valor máximo que puede tomar ese valor realmente.
- **Opción 2:** Implementarlo con el DataType Money.
Ventajas: Resuelve las desventajas que genera la opción 1.
Desventajas: Es algo menos intuitivo y no se le pueden añadir restricciones adicionales a la cantidad (debe ser igual mayor a cero).

DECISIÓN TOMADA: Tras el análisis anterior y habiendo consultado el foro, dónde ya estaba esta cuestión resulta, la conclusión es que la mejor decisión es la opción 2. De esta manera, se podría almacenar la cantidad del presupuesto junto con su correspondiente moneda. La restricción de que debe ser mayor o igual a cero tiene que ser tratada en el servicio. El requisito quedaría redactado de la siguiente manera:

[D02-S02-02]*: 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** (zero or positive amount of money)*.

5. Entregable 03

REQUISITO ANALIZADO N°1

[D03-S02-07]: 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.

PROBLEMA: La problemática en cuestión viene dada por si es necesario que los contratos a los que se asocian los progress logs estén publicados antes de que los progress logs asociados a ellos sean creados. Las opciones valoradas:

- **Opción 1:** No es necesario que los contratos estén publicados.
Ventajas: Fácil de implementar.
Desventajas: No tiene mucho sentido crear registros de progreso si dicho contrato no está publicado, ya que no sabemos si el contrato cumple con los requisitos del proyecto al que está asociado.
- **Opción 2:** Sí es necesario que los contratos estén publicados.
Ventajas: En esta opción, ya sabemos que dicho progress log está asociado a un contrato publicado, por lo que ha sido aceptado y cumple con los requisitos del proyecto.
Desventajas: Es algo más difícil de implementar pues hay que añadir más restricciones.

DECISIÓN TOMADA: Tras el análisis anterior y habiendo consultado a la profesora de la asignatura, la mejor decisión es la opción 2. Esto es debido a que no tiene mucho sentido tratar registros de la completitud de un contrato si este todavía no hay sido publicado.

REQUISITO ANALIZADO N°2

[D03-S02-07]: 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.

PROBLEMA: La problemática en cuestión viene dada por el valor que puede tomar el atributo completeness de un progress log. Las opciones valoradas:

- **Opción 1:** Puede tomar cualquier valor.
Ventajas: Fácil de implementar.
Desventajas: Es algo irreal ya que cuando se publican registros, el valor completeness, el cual indica qué porcentaje del contrato se ha completado, si es menor a uno anterior publicado significaría que hemos retrocedido en la completeness, algo que no puede ser cierto.
- **Opción 2:** Mayor al mayor progress log publicado de ese contrato.
- *Ventajas:* Le más sentido a que la completeness de un contrato siempre vaya en aumento a medida que se van publicando sus progress logs.
Desventajas: Es algo más difícil de implementar pues hay que añadir una validación más en los controladores crear, actualizar y publicar, que requieren de una consulta al repositorio para obtener la completeness mayor publicada de un contrato en cuestión.

DECISIÓN TOMADA: Tras el análisis anterior, la mejor decisión es la opción 2. En un sistema real no tendría sentido que la completeness de un progress log de un contrato fuera menor a la última publicada (que es la mayor).

REQUISITO ANALIZADO N°3

[D02-S03-08]: Operations by **clients** on client **dashboards** :

- Show their **client** dashboards.

PROBLEMA: La problemática en cuestión es si debemos considerar en el dashboard únicamente los registros de progreso y contratos publicados, o también debemos incluir aquellos que están en modo borrador:

- **Opción 1:** Considerar únicamente los datos de las entidades publicadas.
Ventajas: Tiene más sentido en un sistema real, pues los contratos y registros que no están publicados pueden cambiar.
Desventajas: No tiene ninguna desventaja significativa, sólo la adición de un parámetro en el filtrado de las consultas del repositorio.
- **Opción 2:** Considerar todas las entidades, incluso las de modo borrador.
- *Ventajas:* No tiene ninguna ventaja con respecto a la anterior opción.
Desventajas: Es algo irreal en un sistema real porque los datos mostrados no se corresponden con la realidad de los contratos y progress logs aceptados.

DECISIÓN TOMADA: Tras el análisis anterior y habiendo consultado el foro, dónde ya estaba esta cuestión resulta, la conclusión es que la mejor decisión es la opción 1. De esta manera sólo se tienen en cuenta las entidades publicadas, ya que las demás son susceptibles a cambios y pueden generar una interpretación de los datos que no es real.

ENLACE DE VALIDACIÓN:



7/04/24
18:50

Incoherencia con
atributo cost en
Projects.



RAMON
GAVIRA
SANCHEZ

REQUISITO ANALIZADO N°4

[D02-S03-08]: Operations by **clients** on client **dashboards** :

- Show their **client** dashboards.

PROBLEMA: La problemática en cuestión es si debemos mostrar los atributos del dashboard referidos a las medidas de los contratos agrupados por moneda o no:

- **Opción 1:** No agrupar por monedas los atributos.
Ventajas: Mas fácil de implementar.
Desventajas: No muestra realmente ningún valor real, pues si hubiera por ejemplo dos monedas, calcularía las medidas mezclando ambas divisas sin tener en cuenta el valor de una con respecto a la otra.
- **Opción 2:** Agrupar por monedas los atributos.
- *Ventajas:* Es más realista mostrar las medidas agrupadas por monedas.
Desventajas: Es más complejo de implementar. Esto se debe a que hay que añadir 5 atributos nuevos en el dashboard. 4 maps que agrupan la divisa con la medida en cuestión y una colección que contiene el nombre de todas las divisas.

DECISIÓN TOMADA: Tras el análisis anterior y habiendo consultado el foro, dónde ya estaba una cuestión parecida resuelta sobre los dashboards de los managers, la conclusión es que la mejor decisión es la opción 2. De esta manera las medidas de los contratos se muestran agrupadas por moneda. Además, teniendo en cuenta la implementación de nuestro sistema, es la mejor opción porque al no tener monedas predefinidas en el sistema cualquier divisa es aceptada.

ENLACE DE VALIDACIÓN:



19/03/24
19:42

D03 - Operaciones
relacionadas a Money



LUCAS
MANUEL
HERENCIA SOLIS

6. Entregable 04

Intencionalmente en blanco.

7. Conclusiones

Tras el análisis de requisitos, se puede observar que no hay ninguno analizado en los entregables 1 y 4. En cuanto a los requisitos analizados en los entregables 2 y 3, se centran en las entidades Contract, ProgressLog y ClientDashboard, además de un requisito analizado para la entidad Client.

Para la entidad Client, se ha decidido que el atributo type se implementará como un tipo Enumerado con los valores COMPANY o INDIVIDUAL", y se le aplicará la restricción "not null" para asegurar que no pueda ser nulo.

De la entidad Contrato, se ha determinado que el atributo budget será de tipo Money, lo cual permite almacenar la cantidad del contrato junto con su correspondiente moneda, solucionando las limitaciones de usar un tipo de dato más simple como int.

Para la entidad ProgressLog, se ha concluido que los logs de progreso solo pueden ser creados si el contrato asociado está publicado. También se ha decidido que el atributo completeness debe ser mayor al del último progress log publicado para reflejar un progreso continuo y coherente.

Por último, en cuanto a la entidad ClientDashboard, se ha decidido que solo se considerarán las entidades publicadas para evitar inconsistencias en los datos mostrados. Además, las medidas de los contratos se agruparán por moneda para proporcionar una visión realista de los datos.

8. Bibliografía

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