

Analysis report



Diseño y pruebas II

Sprint 2

Versión	Fecha	descripción
1.0	07/03/2024	Creación del documento y adición de la estructura básica
1.1	07/03/2024	Adición del análisis perteneciente a los requisitos de información obligatorios
1.2	08/03/2024	Adición del análisis de los requisitos restantes y revisión del documento

Fecha 08/03/2024

Preparado por:
Benito Hidalgo, Daniel(C1.028)

Índice:

Índice:	2
1. Introducción	2
2. Análisis y ejecución de los distintos requisitos	2
2.1. Requisito 2) Creación de la entidad Contract	2
2.2. Requisito 3) Creación de la entidad PorgressLog	3
2.3. Requisito 4) Creación del dashboard para Client	4
2.4. Requisito 5) Desarrollo de datos de ejemplo para entidades realizadas y rol Client	4
2.5. Requisito 13) Creación del rol Client	5
2.6. Requisito 14) Desarrollo del UML	5
2.7. Requisito 15) Desarrollo del documento de análisis	6
2.8. Requisito 16) Desarrollo del documento de planificación y progreso	6

1.Introducción

En este documento se tiene por objetivo plasmar el proceso de análisis que se ha llevado a cabo para la realización de los diferentes requisitos de esta entrega, así como explicar y justificar las diferentes decisiones tomadas durante el desarrollo de estas.

2.Análisis y ejecución de los distintos requisitos

Para el desarrollo de este punto, se hará una división por requisitos, para cada uno, se dará la descripción proporcionada por el documento de requisitos, se explicara el análisis realizado por el alumno y la forma en la que se ha realizado el requisito

2.1. Requisito 2) Creación de la entidad Contract

Descripción: “ 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)”

De este enunciado podemos sacar en claro que será necesario crear una entidad Contrato, la cual tendrá todos los atributos que se han mencionado junto con las restricciones a estos que sea posible introducir en el modelo, además de las relaciones necesarias con otras entidades o roles.

La decisión tomada en este caso es la de crear la entidad siguiendo el enunciado dado, creando un paquete específico “contracts” y dentro de este una clase que contendrá la entidad. Además, se han añadido las relaciones necesarias hacia project y el rol Client, esta última gracias al feedback aportado por el profesor.

Los problemas principales a resolver en esta tarea han sido aquellos relacionados a los tipos de los atributos, los más notorios han sido los siguientes:

Para el atributo budget se ha llegado a la conclusión de que el tipo que mejor le corresponde es Money, un tipo personalizado que nos proporciona el framework, debido a esto, la validación de la restricción de este atributo (budget debe ser menor o igual que el atributo cost de project, que es un tipo entero al tratarse de horas) se deberá realizar en un servicio ante la falta de anotaciones que nos permitan comprobar algo de semejante complejidad.

El atributo instantiationMoment también planteaba problemas referentes al tipo y su validación, ya que debía ser una fecha anterior al momento presente, Como resolución al problema que esto planteaba, se estableció el tipo del atributo como un tipo date, usando la anotación @temporal(temporalType.TIMESTAMP) para especificar que el atributo almacenaría fecha y hora, y la anotación @Past para validar que los datos aportados no superasen una fecha “presente” establecida en el archivo “applicationProperties” como base-moment, con valor “2022/07/30 00:00”.

Como datos extra, se estuvo estudiando la necesidad de establecer rangos para atributos de los que no nos den una información específica en relación a sus valores máximos o mínimos, y gracias a una publicación del foro de la asignatura se nos dieron algunos extremos para diferentes tipos de atributos, la aplicación de estas será realizada cuando sea posible en forma de anotación en el modelo, pero en el caso del tipo Money esta restricción será introducida en los servicios, debido a la dificultad de parametrizar esta mediante anotaciones

2.2. Requisito 3) Creación de la entidad PorgressLog

Descripción: “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).”

Tras analizar el requisito, se llegó a la misma conclusión que con la creación de Contract, crear un paquete específico, y dentro de este la clase de java que contendrá a la entidad, realizada siguiendo los atributos, restricciones y relaciones establecidas en el enunciado.

Para este requisito, se ha tomado la decisión de establecer el parámetro completeness como un float que podrá tener valores desde 0.00 hasta 100.00, la razón de esto es que al usar un tipo primitivo se ahorraría memoria y eliminaría la necesidad de introducir la anotación @NotNull.

Para el atributo de tipo Date se ha seguido la misma política que se comentó para el atributo instantiationMoment, ya que ambos se comportan de la misma forma.

2.3. Requisito 4) Creación del dashboard para Client

Descripción: “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**.”

Del análisis del requisito sacamos en claro la necesidad existente de guardar información relacionada a datos estadísticos sobre los progressLogs y contracts.

Ante esto se decidió hacer uso de estructuras de tipo Form para la realización de este requisito, creando uno de nombre ClientDashboard y estableciendo los tributos y tipos necesarios.

A la hora de los tipos usados en este requisito, para aquellos relacionados al porcentaje de completitud de los progressLog, se ha optado por usar el tipo float, ya que no permite el uso de valores null, algo que atendiendo a la descripción se vió que no debía mostrarse nunca. Por último, para aquellos atributos que provendrán del presupuesto de los contratos se ha decidido usar el tipo Money para mantener el mismo tipo del atributo a usar para calcular el valor a mostrar.

2.4. Requisito 5) Desarrollo de datos de ejemplo para entidades realizadas y rol Client

Descripción: “Produce assorted sample data to test your application informally. The data must include two **client** accounts with credentials “**client1/client1**” and “**client2/client2**”.”

Del análisis de este requisito sacamos la necesidad de crear una serie de datos de muestra sobre las entidades que se han pedido anteriormente, incluyendo 2 cuentas de usuario pertenecientes a clientes para realizar un testeo informal sobre la aplicación.

Para el cumplimiento de este requisito se ha decidido seguir la política que se explicó en teoría, para atributos definidos en rangos, se desarrollarían 1 ejemplo en cada extremo, otro en el centro del rango, y 2 ejemplos con valores más centrales. A parte, se debería probar que las validaciones sean correctas, buscando siempre la introducción de datos siempre válidos, teniendo en cuenta probar aquellos atributos que puedan ser nulos introduciendo datos de este tipo, en total, los datos de ejemplo introducidos han sido:

Para Contract se han creado 30 datos de ejemplo, para progressLog 25 y para client se han realizado las dos cuentas que se pedía, aparte, para la entidad opcional client se han realizado los 2 datos exigidos

2.5. Requisito 13) Creación del rol Client

Descripción: “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.”

A partir de la información dada, llegamos a la conclusión de que en el sistema es necesario almacenar una serie de datos en relación a un nuevo rol, Client.

Con este fin, se ha tomado la decisión de seguir el patrón que siguen los diferentes roles de la aplicación y crear una clase donde establecer los atributos y restricciones necesarias.

En relación a otras decisiones tomadas en el transcurso de la creación del rol, se han añadido las anotaciones @email y @URL al atributo email y link respectivamente con el fin de restringir qué datos son tomados como válidos para estos atributos. Otra adición ha sido la creación de un Enum, “CompanyType”, dentro del paquete donde se aloja Client para añadir así el atributo “type” de la forma más cercana a como se define en la descripción del requisito.

2.6. Requisito 14) Desarrollo del UML

Descripción: “Produce a UML domain model”

A partir de la información aportada se aprecia la necesidad de generar un diagrama UML donde establezcamos las distintas relaciones, atributos y restricciones que las distintas entidades y roles que hemos ido creando a lo largo de esta entrega.

Para su correcta realización se ha decidido generar el diagrama a partir de la información y ejemplos dados en las sesiones de teoría.

Debido a la poca complejidad de esta tarea no se ha tomado ninguna otra decisión que implique un cambio sustancial del formato o contenido del UML

2.7. Requisito 15) Desarrollo del documento de análisis

Descripción: "Produce an analysis report."

Para este requisito se nos insta a generar el documento que ahora mismo está leyendo, debido a la poca complejidad de este no tiene sentido hablar sobre las decisiones tomadas en su realización

2.8. Requisito 16) Desarrollo del documento de planificación y progreso

Descripción: "Produce a planning and progress report."

En este enunciado no hay mucha información que analizar, pues se puede ver claramente que el objetivo de este es que se cree un documento donde se analice la fase de planificación de los diferentes requisitos, y el proceso de desarrollo de estos.

Dada la poca complejidad de este requisito, no ha sido necesario tomar ninguna decisiones base a cómo realizar el documento.