

C1.04.06

Informe de análisis

Estudiante #2

Repositorio: <https://github.com/C1-04-06/Acme-L3-D02.git>

Fecha: 11 de marzo de 2023



Integrantes

Diego García Linares diegarlin@alum.us.es

María Márquez Soldán marmarsol4@alum.us.es

Juan Carlos Morato Navarro juamornav3@alum.us.es

Olegario Morato Navarro olemornav@alum.us.es

Úrsula Garrucho Sánchez ursgarsan@alum.us.es

ÍNDICE

Resumen de la entrega e introducción	2
Tabla de revisiones	2
Análisis	2
Conclusiones	7
Bibliografía	7

Resumen de la entrega e introducción

Todo este documento es referente al estudiante #2: María Márquez Soldán con UVUS marmarsol4.

En esta entrega se han realizado las entidades de Enrolment, Activity y EnrolmentDashboard y he creado el rol Student. También se han introducido datos de prueba de cada entidad y rol mencionados. Además, se han elaborado 2 reportes, uno de planificación y este mismo.

Tabla de revisiones

Nº	Fecha	Descripción
1	09/03/2023	Primera versión completa del documento
2	11/03/2023	Cambio estético para mayor comprensión e introducción de enlaces en referencias al foro de la asignatura

Análisis

R04 Student: There is a new project-specific role called student, which has the following profile data: statement (not blank, shorter than 76 characters), list of strong features (not blank, shorter than 101 characters), list of weak features (not blank, shorter than 101 characters), and an optional link with further information.

El requisito no presenta ninguna ambigüedad, incompletitud o incorrección.

R05 Enrolment: *An enrolment is a registration of a student in a course. The system must store the following data about them: a code (pattern "[A-Z]{1,3}[0-9][0-9]{3}", not blank, unique), a motivation (not blank, shorter than 76 characters), some goals (not blank, shorter than 101 characters), and a work time (in hours, computed from the corresponding activities).*

Este requisito lo he considerado como incorrecto y ambiguo, realizando los siguientes análisis.

Análisis patrón

Dado el feedback en las clases de follow-up se ha indicado que el patrón mostrado en el requisito es incorrecto, dado que presenta errores y adicionalmente no se aplica a toda la cadena, siendo el correcto el siguiente patrón: `^[A-Z]{1,3}\\d{3}$` (donde los extremos `^` y `$` fuerzan que el patrón se cumpla sobre toda la cadena). El primer fallo del patrón quedó reflejado en un mensaje del foro [\[D&T\] Analysis on Practicum requirement](#).

Análisis sobre Workbook time

El análisis que tuvo que realizarse fue sobre cómo almacenar el cómputo de tiempo total del Enrolment, la cual es una propiedad derivada que aún no sabemos implementar. Tras estudiar la publicación del foro [Ambigüedad en la estimación de tiempo de una lección](#), se dejó claro que la cantidad de horas puede ser representada como un Double, que representaría las horas como una fracción y por lo tanto ha sido la opción escogida al ser la más cómoda de tratar.

R06 Activity: *Every enrolment has a workbook that is composed of activities. The system must store the following data about them: a title (not blank, shorter than 76 characters), an abstract (not blank, shorter than 101 characters), an indication on whether it can be considered a theory activity or a hands-on activity, a time period (either in the past or the future), and an optional link with further information.*

Este requisito lo he considerado como incompleto y ambiguo, realizando los siguientes análisis. La causa de que lo considere incompleto es que en ningún lado (aparte del foro) se hace explícito que las actividades son exclusivas del curso y que no pueden ser compartidas.

Análisis Activity Nature

El análisis realizado fue cómo implementar la naturaleza de una actividad que, tras estudiar la publicación del foro [\[D&T\] Theoretical/hands-on type](#), se optó por utilizar un enumerado.

Observando además que este enumerado era común a la entidad Course del estudiante 1 pero con un valor más, se optó por introducirlo en un paquete aislado para evitar repetir código.

Análisis Time Period

El tercer análisis realizado fue referente a cómo almacenar un periodo de tiempo para la entidad de Activity. Para lo cual se puso el siguiente análisis (que se encuentra [aquí](#) y ya fue respondida por el profesor) en el foro de la asignatura:

Estimado profesor,

Soy María Márquez Soldán, del grupo de laboratorio C1.04.06. Le escribo para consultarle respecto a un requisito referente al estudiante #2 que me ha planteado dudas sobre su almacenamiento.

Contexto:

El requisito es el siguiente:

- *Every enrolment has a workbook that is composed of activities. The system must store the following data about them: a title (not blank, shorter than 76 characters), an abstract (not blank, shorter than 101 characters), an indication on whether it can be considered a theory activity or a hands-on activity, a time period (either in the past or the future), and an optional link with further information.*

Pregunta:

¿A qué hace referencia ese periodo de tiempo? Dada mi interpretación, he entendido que se refiere al tiempo en el cual está disponible la actividad (una especie de tiempo límite, un deadline), pero da a entender que se trata de un tipo Date. Si esta fuese incorrecta, ruego que me corrija.

También, partiendo de que no hay ninguna anotación que considere simultáneamente el pasado y el futuro, quería pedirle consejo sobre cómo proceder sobre esa restricción, porque pienso que va a dar conflicto si se usa @Past y @Future a la vez.

Propuestas para implementación:

Propuesta #1: Implementar dicho periodo como un Double mediante inicio y fin

La idea sería poner dos atributos Date (de tipo Timestamp), uno de inicio y otro de fin (considerando que el problema de la restricción estuviese solucionado). El periodo sería derivado

y sería implementado como la diferencia del fin y el inicio, para posteriormente ser convertido a un Double que representaría las horas de forma fraccionaria/porcentual.

Ej: 1.5 representaría 1 hora y 30 min

- Pros:
 - Facilitaría el cómputo de tiempo total de la colección de actividades en Enrolment.
 - Dado que el inicio y el fin quedan registrados en la entidad como atributos básicos, no se perderían las fechas sobre el periodo.
- Contras:
 - Al poner dos atributos básicos nuevos en la entidad, me estoy saliendo de lo que pide el cliente almacenar.
 - Comprobar la disponibilidad requeriría hacer un get al campo de fin.

Propuesta #2: Implementar dicho periodo como un @Temporal(TemporalType.TIME) mediante inicio y fin.

De nuevo, se haría uso de 2 atributos Date básicos (inicio y fin). El periodo sería derivado y sería implementado como la diferencia del fin y el inicio.

- Pros:
 - Dado que el inicio y el fin quedan registrados en la entidad como atributos básicos, no se perderían las fechas sobre el periodo.
- Contras:
 - Al poner dos atributos básicos nuevos en la entidad, me estoy saliendo de lo que pide el cliente.
 - Necesitaría una conversión a Double para satisfacer el cómputo de tiempo total de las actividades en la entidad Enrolment. Ya fuese implementado en la entidad de Activity o en el propio método de la entidad Enrolment.
 - Quizás TemporalType.TIME tenga alguna restricción que impida introducir un tiempo superior a 24h, en caso de que el periodo sea mayor a un día.
 - Comprobar la disponibilidad requeriría hacer un get al campo de fin.

Propuesta #3: Implementar dicho periodo como un @Temporal(TemporalType.TIME)

El atributo de periodo en este caso sería básico y representaría directamente el tiempo de disponibilidad.

Ej: 1:30:00 representaría 1 hora y 30 min

- Pros:
 - Me ajusto al número de atributos que pide el cliente
- Contras:

- Necesitaría una conversión a Double para satisfacer el cómputo de tiempo total de las actividades en la entidad Enrolment. Ya fuese implementado en la entidad de Activity o en el propio método de la entidad Enrolment.
- Quizás TemporalType.TIME tenga alguna restricción que impida introducir un tiempo superior a 24h, en caso de que el periodo sea mayor a un día.
- Se perdería información sobre cuándo esa actividad deja de estar disponible al no disponer de fechas

No he considerado implementarlo como @Temporal(TemporalType.TIMESTAMP) porque al tener un tipo fecha puede ser confuso para el usuario, habría que establecer unas pautas sobre la interpretación de la misma.

Solución: Sin conocer al completo las restricciones de TIME y teniendo en cuenta que pudiese dar problemas, pienso que la mejor solución es la que lo implementase como un Double, a pesar de añadir nuevos atributos a la entidad. Dado que en el resto o se pierde información o podría encontrarme con que no se puede añadir más de 24h.

Espero que pueda ayudarme a clarificar el asunto de la restricción @Past-@Future y sobre cuál piensa usted que es la mejor alternativa.

Muchas gracias de antemano y un saludo.

Tras la respuesta del profesor y apoyándome en la respuesta que le dieron a Antonio Rodríguez Ruiz en su publicación [\[D&T\] Analysis on activities requirement](#), implementé dicho periodo como dos tipo Date con TemporalType.TIMESTAMP.

Relación Enrolment-Activity

La relación entre Enrolment y Activity presentó dudas respecto a su multiplicidad, dado que en ningún lado se evidenció si las actividades podían ser compartidas entre Enrolments.

Referenciando el análisis realizado en la publicación del foro [\[D&T\] Analysis on enrolment requirement](#), he decidido implementar la relación entre ambas entidades como una ManyToOne, siendo el Many Activity y el One Enrolment, dado que una actividad no tiene sentido que exista sin un Enrolment.

La falsa entidad Workbook no planteó un problema desde el principio ya que no almacenaba ningún dato aparte de su identificador, por lo tanto se comprendía con facilidad que Workbook era la propia relación que he mencionado, no una entidad.

R07 EnrolmentDashboard: *The system must handle student dashboards with the following data: total number of theory and hands-on activities in his or her workbook; average, deviation,*

minimum, and maximum period of the activities in his or her workbook; average, deviation, minimum, and maximum learning time of the courses in which he or she has enrolled.

El requisito no presenta ninguna ambigüedad, incompletitud o incorrección.

Conclusiones

Los requisitos de esta entrega han resultado más ambiguos e incompletos que los de la primera entrega y por tanto algunos han necesitado un análisis completo, acompañado de un estudio de los mensajes del foro para sacar en claro la implementación para que sea coherente.

Sin embargo, no todos han sido así puesto que siguen habiendo requisitos muy claros que no presentan dificultades.

Bibliografía

Referencias
Foro de la asignatura
Transparencias de la asignatura