

DP II - 2022-2023

ANALYSIS REPORT



<https://github.com/pedlopruz/C1.04.12>

Miembros:

- Carolina Carrasco Díaz
- Pedro López Ruz
- Ángel Lorenzo Casas
- Manuel Navarro Sicre
- Manuel Ortíz Blanco

Tutor: RAFAEL CORCHUELO GIL

GRUPO - C1.04.12

1 ÍNDICE

Resumen Ejecutivo	2
Tabla de Versionado	2
Introducción	2
Contenido	2
Conclusión	13
Bibliografía	14

2 RESUMEN EJECUTIVO

A continuación la finalidad del presente documento será la de tener documentado por escrito cualquier cambio realizado en el proyecto en curso. Reflejándose en todo momento la aprobación o desestimación del cliente ante los cambios propuestos para realizar en dicho proyecto. Se proporcionará por tanto una breve descripción del problema u mejora, así como distintas soluciones a dicho cambio y la decisión final del cliente a dichas propuestas.

3 TABLA DE VERSIONADO

Versión	Fecha	Descripción
1.0	04/03/2023	Primera versión del documento
1.1	13/03/2023	Añadidos los contextos, problemas, alternativas y aprobaciones del cliente según cada observación o problemas encontrados.

4 INTRODUCCIÓN

La intención de este documento es la de recopilar y documentar el análisis de todos los problemas encontrados, los cuáles que van surgiendo a lo largo del desarrollo del proyecto) así como las propuestas de cambios proporcionadas al cliente en espera de su aprobación de cambio u correcciones necesarias, siempre y en todo momento de manera consensuada con el cliente.

5 CONTENIDO

A continuación se pone en contexto todas dudas y problemas encontrados a lo largo del siguiente entregable, con las propuestas sugeridas y la respuesta y/o aceptación del cliente en cada una de ella aprobando o desestimando dichos cambios propuestos.

#1 - Contexto: *The requirement in question states that the system must store the code attribute for each of the practicums.*

Problema: Este atributo de código debería haber sido el siguiente: patrón "[A-Z]{1,3}[0-9]{3}", no en blanco, sino único. Presentando por tanto un ligero cambio en el patrón proporcionado previamente. Como pide un número y luego otros 3 números de nuevo, el siguiente patrón significaría lo mismo, pero más directo: [A-Z]{1,3}[0-9]{4}.

→ Pros:

- Enfoque mejor y más directo.

→ Contras:

- No se encuentra inicialmente ningún contra.

Alternativas propuestas:

Por lo anteriormente expuesto, se propone implementar el patrón regex para el atributo code de la siguiente manera: [A-Z]{1,3}[0-9]{4} .

Respuesta Cliente:

"Por desgracia, hay un error en este requisito. Nuestro cliente ficticio confirma que el patrón correcto es el siguiente: [A-Z]{1,3}[0-9]{3} (Una hasta tres letras mayúsculas y luego tres dígitos)"

Conclusión: Cambio validado.

#2 - Contexto: *Produce assorted sample data to test your application informally. The data must include two student accounts with credentials "student1/student1" and "student2/student2".*

Problema: Si el cliente ha entendido como credenciales el identificador de los estudiantes, entonces el formato otorgado estaría incorrecto y generaría problemas, ya que los datos deben estar escritos en formato "kebab case", por lo que las credenciales que solicita el cliente deberían ser "student-01" y "student-02".

Alternativas propuestas:

Alternativa 1: El cliente se ha equivocado al explicar las credenciales y debería rectificar la información.

- **Ventajas:**
 - Para futuras entregas, el cliente ya sabrá formular requisitos relacionados con la generación de datos de prueba.
 - Entenderá mejor el formato utilizado para generar datos de prueba.
- **Desventajas:**
 - Será necesario que reformule el requisito

Alternativa 2: El cliente no se ha equivocado y las credenciales están correctas.

- **Ventajas:**
 - No hay necesidad de reformular el requisito
- **Desventajas:**
 - Se tendría que buscar una forma de que las credenciales solicitadas funcionen al generar los datos de prueba, pudiendo ocupar mucho tiempo.

Respuesta Cliente:

*"Las convenciones de nomenclatura tienen que ver con nombres de ficheros (./WEB-INF/views/authenticated/announcement/messages-en.i18n), con identificadores (AbstractController<Authenticated, UserAccount>), con URLs (http://localhost:8080/acme-l3/authenticated/user-account/show?id=99), ... pero no con datos. No hay nada malo en las credenciales "estudiante1/estudiante1"; **son correctas.**"*

Conclusión: Ningún cambio necesario.

#3 - Contexto: *A lecture is a document that a lecturer uses to get some knowledge across. 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 estimated learning time (in hours, positive, not nought), a body (not blank, shorter than 101 characters), an indication on whether it can be considered theoretical or hands-on, and an optional link with further information.*

Problema:El problema está en la ambigüedad que se presenta, debido a que el tiempo estimado al medirse como Double, podría significar por ejemplo, que 2.5 horas son dos horas y media, o dos horas y cincuenta minutos

Alternativas propuestas:

Alternativa 1: Que el parámetro tome valores decimales, donde la parte entera representa las horas y la parte decimal representa los minutos.

- Ventajas:
 - Facilidad a la hora de computar la duración de una lección
- Desventajas:
 - No es un formato propio del lenguaje natural y puede llevar a controversias

Alternativa 2: Que el parámetro tome valores decimales, donde la parte entera representa las horas y la parte decimal represente el porcentaje de una hora

- Ventajas:
 - Este formato representa un poco mejor el lenguaje natural.
- Desventajas:
 - Dificultad a la hora de calcular el porcentaje de una hora que representan los minutos.

Alternativa 3: Que el parámetro tome sólo valores enteros. Esta alternativa da lugar a dos nuevas posibilidades:

- Opción 1: Que el parámetro represente horas completas.
 - Ventajas:
 - Es la alternativa más fácil a la hora de representar la duración de una lección.
 - Desventajas:
 - Es muy probable que las lecciones no duren horas completas por lo que el parámetro no representaría bien la realidad por lo que sería muy impreciso.
- Opción 2: Que el parámetro represente minutos.
 - Ventajas:

- Precisión para representar las duraciones.
- Desventajas:
 - Podría presentar un poco más de complicación para representar tiempos ya que cuando se quieran representar horas, habrá que pasarlas a minutos.

Solución Propuesta: Tras analizar el requisito, nos decantamos por la tercera alternativa, concretamente por la segunda opción ya que nos parece que es la opción cuyo balance pros-contras, es mejor que tiene una gran ventaja como es la precisión y solo tiene la desventaja de que si queremos representar duraciones deberemos tener en cuenta que trabajamos con minutos y que para representar una hora utilizamos el valor 60.

Respuesta Cliente: *“El problema que se plantea es muy común: nuestro cliente habla de medir un tiempo en horas y no tenemos claro si el número de horas es un entero (habitual en muchos contextos relacionados con la formación y también en la prestación de servicios profesionales dado que se factura por horas completas en muchos) o decimal. En este último caso hay clientes que interpretan la parte decimal como un porcentaje (2.50 h = 2 horas 30 minutos) y otros que la interpretan como minutos (2:50 = 2 horas 50 minutos). La diferencia al mostrar los datos en pantalla es que en un caso se usa el “.” o la “,” mientras que en el otro se usa “:” para separar la parte entera de los decimales.*

Nuestro cliente ficticio nos comunica que el número de horas puede ser fraccionario y que la fracción representa porcentaje. Su mensaje nos confirma que podemos implementarlo como un atributo de tipo doble con las siguientes restricciones (entre otras): tres dígitos enteros, dos dígitos fraccionarios, y mínimo valor 0.01.”

Conclusión: Cliente acepta la propuesta, considerándose como un atributo tipo doble (3 dígitos enteros y 2 dígitos fraccionarios y mínimo valor 0.01..

#4 - Contexto: *El requisito 5 indica lo siguiente: "A tutorial provides additional support to a course by means of one or more sessions. 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 title (not blank, shorter than 76 characters), an abstract (not blank, shorter than 101 characters), some goals (not blank, shorter than 101 characters), and an estimated total time. "*

Y el requisito 6 indica lo siguiente: "The system must store the following data about the sessions: 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 session or a hands-on session, a time period (at least one day ahead, from one up to five hour long), and an optional link with further information."

Problema: El requisito número 5 deja claro que un tutorial tendrá una o más sesiones. Sin embargo, mi duda corresponde al caso inverso. ¿La sesión de un tutorial puede pertenecer a varios tutoriales distintos o sólo a un tutorial?

Alternativas propuestas:

Alternativa 1: Las sesiones pertenecen a un único tutorial.

- **Ventajas:**
 - Las sesiones pueden ser más específicas, y pueden contener mayor información relevante para el tutorial en cuestión.
- **Desventajas:**
 - Puede haber información redundante en dos tutoriales distintos.

Alternativa 2: Las sesiones pueden pertenecer a varios tutoriales

- **Ventajas:**
 - Si dos tutoriales tienen gran parte del contenido en común, se puede llegar a reutilizar la sesión de uno de estos.
- **Desventajas:**
 - Puede haber información en una sesión que sea necesaria para un tutorial pero no para otro relacionado, por lo que las sesiones tendrían que ser más genéricas.

Solución Propuesta: Limitar que las sesiones pertenezcan a un único tutorial, debido a que la generalización podría dar lugar a errores no contemplados anteriormente.

Respuesta Cliente:

"Los tutoriales son sesiones que un asistente organiza para proporcionar ayuda complementaria en el contexto de un curso particular. Muchos tutoriales se esperan que tengan una sola sesión, pero el

sistema debe abrir la posibilidad a que un tutorial especialmente complicado pueda dividirse en varias sesiones. Las sesiones documentan qué se va a hacer en el contexto de un determinado tutorial, por lo que son sus elementos constituyentes; se trata de una composición (strong aggregation, composition) y no de una agregación (weak aggregation, aggregation)."

Conclusión: duda aclarada.

#5 - Contexto: *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.*

Problema: El problema recae en cómo interpretar ese periodo de tiempo, ya que según la forma de interpretarlo afectará a todas las clases que tengan atributos de navegación con esta entidad. Por ejemplo, la entidad Enrolment debe tener el un atributo derivado que indique el tiempo de trabajo total , en horas, de las actividades que pertenecen a esa inscripción. Y aunque el requisito explica que ese periodo de tiempo puede ser tanto pasado como futuro, esto no afecta al problema porque decir que un periodo de tiempo sea válido tanto en el pasado como en el futuro he interpretado de que es equivalente a decir que cualquier periodo de tiempo es válido.

Alternativas propuestas:

Alternativa 1: Considerar periodo de tiempo como la diferencia de fechas de inicio y fin , ambas fechas teniendo como anotaciones @Temporal(TemporalType.TIMESTAMP), considerando el periodo como un Double, haciendo que las horas serán la parte entera del resultado y la parte fraccionaria como un porcentaje de los minutos:

Ej: 1.5 serían 1h y 30 minutos

- Ventajas:
 - Facilita el cálculo del atributo derivado de Enrolment, ya que se tendrían que sumar todos los periodos de tiempo ya calculados en la entidad Activity.
 - Al utilizar la parte fraccionaria como un porcentaje de los minutos, no hay que aplicar ninguna regla sobre la parte fraccionaria.
 - Cómo utilizas TemporalType.TIMESTAMP, se consideran los días en el cálculo del periodo.
- Desventajas:
 - El periodo de tiempo sería un atributo derivado que devuelve la diferencia entre las fechas de inicio y de fin.
 - -Se necesitaría una restricción customizada sobre las fechas de inicio y fin para evitar que la fecha de inicio sea superior a la fecha de fin y que la fecha de fin sea inferior a la fecha de inicio.
 - No se cumpliría con la cantidad de atributos solicitados por el cliente, ya que tendrías que añadir los atributos de fecha de inicio y fecha de fin

Alternativa 2: Considerar periodo de tiempo como la diferencia de fechas de inicio y fin , ambas fechas teniendo como anotaciones @Temporal(TemporalType.TIMESTAMP), considerando el periodo como un Double, haciendo que las horas serán la parte entera del resultado y la parte fraccionaria como minutos

Ej: 1.3 serían 1h y 30 minutos

- Ventajas:
 - Facilita el cálculo del atributo derivado de Enrolment, ya que se tendrían que sumar todos los periodos de tiempo ya calculados en la entidad Activity.
 - Cómo utilizas TemporalType.TIMESTAMP, se consideran los días en el cálculo del periodo.
- Desventajas:
 - El periodo de tiempo sería un atributo derivado que devuelve la diferencia entre las fechas de inicio y de fin.
 - Se necesitaría una restricción customizada sobre las fechas de inicio y fin para evitar que la fecha de inicio sea superior a la fecha de fin y que la fecha de fin sea inferior a la fecha de inicio.
 - Sería necesario aplicar una regla sobre este valor Double para incrementar la parte entera cuando los minutos sean superiores a 59.
 - No se cumpliría con la cantidad de atributos solicitados por el cliente, ya que tendrías que añadir los atributos de fecha de inicio y fecha de fin

Alternativa 3: Considerar como periodo de tiempo un único atributo tipo Date y con la anotación @Temporal(TemporalType.TIME) y luego cambiar el formato a Double para que sea interpretable por Enrolment:

Ej: 01:30:00 como 1h y 30 minutos

- Ventajas:
 - Cumpliríamos con la cantidad de atributos solicitados por el cliente, ya que el propio atributo marcaría una duración al ser del tipo TemporalType.TIME
 - No sería necesario crear una restricción customizada ya que sólo es un atributo

- Desventajas:
 - Sería necesario pasar este atributo a un formato numérico que pueda interpretar la entidad Enrolment
 - Para pasarlo a Double, habría que establecer que entiendo como parte fraccionaria(si es un porcentaje de los minutos o si son la cantidad de minutos)
 - Cómo sería un TemporalType.TIME, no se tendrían en cuenta los días, por lo que habría que aplicar una regla que guardase los días del periodo

Solución Propuesta: Tras analizar las alternativas, la solución por la que nos hemos decantado es la primera, ya que aunque para aplicar el atributo solicitado será necesario crear 2 atributos que indiquen la fecha de inicio y la de fin y también será necesario crear una restricción que evitase los conflictos entre estas fechas, facilitará el cálculo del tiempo total de la entidad Enrolment.

Respuesta Cliente:

“La opción 1 que ustedes plantean me parece buena para tratar el problema. La única cuestión es que esto no es un problema de análisis. El análisis de los requisitos está relacionado con identificar incorrecciones, incompletitudes, ambigüedades, y misclassifications en los requisitos y en todos los casos debemos informar al cliente dado que los requisitos forman parte de nuestro contrato con dicho cliente. El problema que aquí hemos tratado es cómo implementar un requisito, lo que supone una decisión de diseño que incumbe tan sólo al equipo de desarrollo. Lo único que afectaría a nuestro cliente es cómo se muestran en pantalla los períodos y sus duraciones, pero entiendo que ahí no hay ninguna duda: el framework ya muestra los objetos de tipo date con un formato estándar en español e inglés y las duraciones pueden Udes. mismos decidir teniendo en cuenta lo que su cliente les ha dicho: 2.5h = 2 horas, 30 min.

Conclusión: cliente aprueba la propuesta 1; propuesta validada.

#6 - Contexto: A course aggregates several lectures by the same lecturer. The system must store the following data about them: a code (pattern "[A-Z]{1,3}[0-9]{3}", not blank, unique), 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 course or a hands-on course (depending on the lectures that it aggregates), a retail price (positive or nought), and an optional link with further information. Purely theoretical courses must be rejected by the system.

Problema: En dicho requisito, se plantea la creación de un curso que contiene varias clases impartidas por el mismo profesor. Para ello, se guardarán varios campos. Entre ellos, se pide un parámetro que indique si una clase se considera teórica o práctica, dependiendo de las clases que contenga. Lo único que se concreta en este requisito sobre este parámetro es que no pueden ser puramente teóricas, pero no el criterio a utilizar para determinar si es o no teórica.

Alternativas propuestas:

Alternativa 1: Usar como criterio que se considerará teórica o practica dependiendo de cuál predomina en total

- Ventajas:
 - Fácil de computar.
- Desventajas:
 - Normalmente, los cursos pueden ser prácticos pero tener más clases de teoría para que se puedan aplicar correctamente a las prácticas

Alternativa 2: Usar un criterio que considerará que si un porcentaje de las clases son prácticas, el curso se considerará práctico

- Ventajas:
 - Aumenta la personalización y es más realista.
- Desventajas:
 - Hay que fijar un valor para dicho porcentaje

Alternativa 3: Usar un criterio que, a partir de un número fijo de clases prácticas, se considera práctico.

- Ventajas:
 - Aumenta la personalización y es fácil de implementar.
- Desventajas:
 - No tiene en cuenta el tamaño del curso en el criterio.

Solución Propuesta: Tras analizar las alternativas, no hemos podido decantarnos por ninguna de las opciones anteriormente comentadas, ya que es cuestión de las políticas del cliente con respecto al tema. Aun así, creemos que las dos primeras opciones son las más beneficiosas y debería elegirse una de ellas. Por tanto, le pedimos al cliente que nos de un criterio concreto que usar.

Respuesta Cliente:

“Una interpretación bastante intuitiva es su alternativa 1: usando la moda, la mayoría. El problema al usar la moda es que un curso podría tener el mismo número de clases teóricas que prácticas y entonces tendría dos modas: ambas opciones serían válidas, pero Ud. sólo puede asignar una naturaleza al curso. Si optamos por esta opción, sería conveniente hablar de una naturaleza intermedia que podríamos denominar BALANCED. Un curso sería tendría la misma naturaleza que la mayoría (moda simple) de sus clases o de naturaleza BALANCED en caso de empate (doble moda).

Su segunda interpretación es razonable, aunque tiene un inconveniente. Se podría dejar al administrador que fijará un porcentaje frontera: por ejemplo, que a partir de un 40% de clases de teoría, un curso se considere teórico y si no es práctico. El problema que le veo a esta interpretación es que no tengo claro que un parámetro de configuración pueda satisfacer a todos los profesores que usan su plataforma.

La tercera opción la veo un poco más problemática. De nuevo requiere fijar un parámetro de configuración del sistema y seguramente será difícil encontrar un valor que venga bien a todo el mundo. Pero además, es necesario un número mínimo en valor absoluto: ¿10? Esto nos obligará a que todos los cursos tengan al menos ese número de lecciones para ser considerados de una u otra forma. ¿2? Entonces un curso con sólo una lección de teoría y su correspondiente práctica sería ... ¿teórico? ¿práctico?

Nuestro trabajo como ingenieros del software es hacer un buen análisis y llevar las conclusiones al cliente para que sea él/ella quien nos indique qué debemos hacer. En este caso, nuestro cliente ficticio nos comunica que la naturaleza de las lecciones, actividades y otras entidades más puede ser: THEORETICAL, HANDS-ON, o BALANCED. En el caso de agregados o composiciones de entidades con naturaleza, ésta se computará como la moda y en caso de empate será BALANCED.”

Conclusión: cliente aclara la duda.

6 CONCLUSIÓN

Actualmente todos los cambios llevados a cabo durante esta entrega se han consensuado en todo momento con nuestro cliente, no habiendo ningún cambio realizado sin la aprobación del mismo. Sírvase además este documento de análisis como prueba de que todos los cambios realizados han sido aprobados y queda documentado en dicho reporte.

7 BIBLIOGRAFÍA

Intencionalmente en blanco.