

Analysis Report – Student #1

Group: L3-D01

Repository: <https://github.com/adrrf/acme-life-long-learning>

Student #1:

UVUS: manvazmar3

Name: Vázquez Martín, Manuel

Roles: Developer, Tester

Email: manvazmar3@alum.us.es

Student #2:

UVUS: angmunpri

Name: Muñoz Prieto, Ángel

Roles: Developer, Analyst

Email: angmunpri@alum.us.es

Student #3:

UVUS: adrromflo

Name: Romero Flores, Adrián

Roles: Developer, Manager

Email: adrromflo@alum.us.es

Student #4:

UVUS: jorromlim

Name: Romero Limón, Jorge

Roles: Developer

Email: jorromlim@alum.us.es

Student #5:

UVUS: alematcap

Name: Mateo Capilla, Alejandro

Roles: Developer

Email: alematcap@alum.us.es

Date: Sevilla a 26 de Mayo de 2023

Índice:

Introducción	3
Contenido	4
Entregable 1:	4
Entregable 2:	4
Entregable 3:	6
Entregable 4:	7
Bibliografía.....	8

Introducción

En este documento se detallarán las decisiones de análisis tomadas por el estudiante 1, Manuel Vázquez Martín, respecto de las tareas individuales que se le asignan; siendo en este caso el rol 'Lecturer', las entidades de 'Lecture', 'Course', el formulario 'LecturerDashboard' y el poblado de los mismos, con sus pertinentes opciones a evaluar y una conclusión explicando por qué se ha escogido dicha solución.

Contenido

Entregable 1:

Debido a la naturaleza de las tareas del entregable, no es necesario realizar tareas de análisis para su realización, luego el equipo del proyecto ha decidido no realizar el informe relacionado con esto.

Entregable 2:

- 1) *There is a new project-specific role called lecturer, which has the following profile data: alma mater (not blank, shorter than 76 characters), a résumé (not blank, shorter than 101 characters), list of qualifications (not blank, shorter than 101 characters), and an optional link with further information.*

No se ha encontrado ninguna ambigüedad, incompletitud o incorrección en este requisito.

- 2) **[MANDATORY]** *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.*

En el siguiente requisito encontramos una ambigüedad, debido a que en este no se especifica si una *lecture* podía pertenecer a varios cursos a la vez o pertenecer únicamente a uno. En clase el profesor comentó que la decisión correcta era que una *lecture* podría pertenecer a varios cursos. Por lo tanto, se plantearon las siguientes soluciones:

- Alternativa 1: Una relación N:N entre *lecture* y *course*. Opción que nos daría muchos problemas en futuras entregas.
- Alternativa 2: Crear una clase intermedia *courseLecture*, con relación N:1 hacia *lecture*, y relación N:1 hacia *course*. Menos complejo que la relación N:N, que es una entidad abstracta con la utilidad de relacionar las dos clases anteriores.

En conclusión, se ha implementado la segunda alternativa, ahorrando así problemas futuros y siendo menos complejo.

La siguiente ambigüedad la encontramos en la propiedad *isTheory*, siendo esta una propiedad derivada proveniente de la propiedad *isTheory* de la entidad *lecture*. Planteamos las siguientes alternativas:

- Alternativa 1: Propiedad no persistente, que se calculara con las propiedades de las asignaturas cada vez que se muestre un curso.
- Alternativa 2: Propiedad persistente, recogiendo la propiedad en la base de datos.
- Alternativa 3: Un curso es teórico si todas sus *lectures* son teóricas.
- Alternativa 4: Un curso es teórico o práctico dependiendo del mayor número de *lectures* que tenga de cada tipo.

En conclusión, se ha seleccionado la segunda y cuarta alternativa, debido a que si la propiedad se persistía en la base de datos, en un futuro nos facilitaría los cálculos de las estadísticas de dichos cursos. Y, en cuanto a la cuarta alternativa, se ha seleccionado para poder disponer de cursos tanto teóricos como prácticos en el sistema. En caso de coincidir el número de *lectures* que tenga el curso, se considerará práctico. En el requisito nos dice "*Purely theoretical courses must be rejected by the system*", donde

consideramos un curso puramente teórico aquel que no disponga de ninguna *lecture* práctica, por lo tanto, dicho curso será rechazado por el sistema.

La siguiente ambigüedad la encontramos en la propiedad *retail Price*, en la que no se le especificaban ninguna restricción.

- Alternativa 1: Propiedad sin restricciones.
- Alternativa 2: Propiedad con restricciones.

En conclusión, se ha seleccionado la segunda alternativa, debido a que la primera alternativa es más compleja a la hora de calcular estadísticas, tener cursos con la propiedad igual a *null*, etc. Por lo tanto, se ha decidido poner la restricción de *@NotNull*, teniendo todos los cursos *retail Price*.

3) **[MANDATORY]** *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.*

En el siguiente requisito encontramos una ambigüedad con el atributo que nos indica las horas estimadas (*stimatesLearningTime*), donde nos indica que tiene que estar expresado en horas, sin indicar el formato a seguir. En clase de laboratorio se planteó la duda de que si hacía falta estimar los minutos, dando libertad a la hora de implementarlo. Las soluciones planteadas son las siguientes:

- Que el atributo fuera de tipo *date* usando el formato de *HH:MM:ss*.
- Que el atributo fuera de tipo *integer*.

Se escogió la segunda opción siendo esta mucho más simple que la primera, aportando el mismo significado.

4) *The system must handle lecturer dashboards with the following data: total number of theory and hands-on lectures; average, deviation, minimum, and maximum learning time of the lectures; average, deviation, minimum, and maximum learning time of the courses.*

En el siguiente requisito no se especifica el tipo de estas propiedades, por lo que se han planteado las siguientes alternativas:

- Alternativa 1: Que la clase tuviera tantas propiedades como estadísticas a calcular.
- Alternativa 2: Que la clase tuviera cuatro propiedades, dos de ellas para el total de *lectures* y cursos, y las otras dos de tipo *Map<String,Double>* una para *lecture* y el otro para curso, en donde se le asigna a la clave la estadística a calcular, y el valor el resultado obtenido.

Por comodidad, se eligió la segunda alternativa simplificando la implementación de la clase.

Entregable 3:

5) *Operations by anonymous principals on user accounts:*

- *Sign up to the system and become a lecturer.*

No se ha encontrado ninguna ambigüedad, incompletitud o incorrección en este requisito.

6) *Operations by lecturers on user accounts:*

- *Update their profiles.*

No se ha encontrado ninguna ambigüedad, incompletitud o incorrección en este requisito.

7) *Operations by any principals on courses:*

- *List the courses in the system that are published.*
- *Show the details of the courses that they can list (excepting their lectures).*

No se ha encontrado ninguna ambigüedad, incompletitud o incorrección en este requisito.

8) **[MANDATORY]** *Operations by lecturers on courses:*

- *List the courses that they have created.*
- *Show the details of their courses.*
- *Create, update, or delete their courses. Courses can be updated or deleted as long as they have not been published. For a course to be published, all of its lectures must have been published.*

En la anterior entrega se consideró que un curso puramente teórico (no dispone de ninguna *lecture* práctica), debería ser rechazado por el sistema. Planteamos las siguientes alternativas:

- Alternativa 1: Un curso puramente teórico no se puede publicar.
- Alternativa 2: Un curso puramente teórico se puede publicar, pero no aparecería en el sistema.

Se ha seleccionado la primera alternativa, ya que en la segunda perderíamos el curso al rechazarlo por el sistema.

9) **[MANDATORY]** *Operations by lecturers on lectures:*

- *List the lectures in their courses.*
- *Show the details of their lectures.*
- *Create and publish a lecture.*
- *Update or delete a lecture as long as it is not published.*

En el siguiente requisito nos especifica que una *lecture* pueda actualizarse o ser borrada mientras esta no esté publicada. Según las transparencias de teoría, que esté publicada o no depende de la propiedad *draftMode*. Se han planteado las siguientes alternativas:

- Alternativa 1: Al crear una *lecture* esta se publicaría automáticamente, sin la posibilidad de ser editada o ser borrada.
- Alternativa 2: Al crear una *lectura* esta se publicaría automáticamente, pero con la posibilidad de ser editada o borrada.
- Alternativa 3: Añadir la propiedad *draftMode* a *lectura*.

Debido a la complejidad y en algunos casos la inconsistencia de la primera y segunda alternativa, se ha seleccionado la tercera alternativa, siendo esta más simple que las demás, y sin dar ocasión a errores futuros.

En segundo lugar, al ya haber contemplado las alternativas anteriores sobre publicar una *lecture*, ahora se nos abren dos posibilidades para añadir una *lecture* a un curso:

- Alternativa 1: Una *lecture* se podría añadir a un curso sin estar publicada.
- Alternativa 2: Una *lecture* solo se podría añadir a un curso si esta está publicada.

Ambas alternativas se plantearon en clase, y el profesor nos aconsejó que escogiéramos la segunda alternativa, ya que esta tiene más sentido, a la hora de que una *lecture* puede ser actualizada varias veces e incluso ser borrada, y que esta ya no tuviera sentido en el curso en el que estaba asociada.

En tercer lugar, con la posibilidad de añadir *lecture* cabía la posibilidad de que existieran varias *lectures* iguales, por lo tanto, se plantearon varias alternativas:

- Alternativa 1: Que pudiera existir varias *lectures* exactamente iguales.
- Alternativa 2: Que la propiedad *title* en las *lectures* tenga la restricción de que sea único.
- Alternativa 3: Añadir una propiedad código que sea único a las *lectures*.

Dado a la complejidad de tercera alternativa debido a el estado avanzado del proyecto, añadir una propiedad nueva llevaría mucho trabajo, pudiendo ser solucionada de una forma más simple con la segunda alternativa. Por lo tanto, se eligió la segunda alternativa, dejando la primera alternativa sin contemplar, debido a que daría muchos problemas de inconsistencia en un futuro.

10) *Operations by lecturers on lecturer dashboards:*

- *Show their lecturer dashboards.*

No se ha encontrado ninguna ambigüedad, incompletitud o incorrección en este requisito.

Entregable 4:

En este último entregable, debido a que no hay que implementar ni diseñar nada, no ha visto el estudiante necesario realizar ninguna tarea de análisis para ello, luego no será necesario un documento de análisis para este entregable..

Bibliografía

- Transparencias de teoría de la asignatura.