Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

ANALYSIS REPORT INDIVIDUAL - D02



**C1.04.02**

**Repositorio**: [https://github.com/josrojrom1/C1.04.02](https://github.com/josrojrom1/C1.04.02%C3%A7)

**Autor:**

Rojas Romero, José Joaquín

josrojrom1@alum.us.es

**Fecha:** 28/02/2023

**Tabla de contenidos**

[**Resumen**](#_4hmpm5uc85lj) **2**

[**Tabla de revisión**](#_90otg4vxm3ka) **2**

[**Introducción**](#_58vtmc9pw6vo) **2**

[**Contenidos**](#_tvy5edcq3b5t) **3**

[**Bibliografía**](#_ysgwrb1dc6k3) **6**

# Resumen

Este documento se trata de un listado de registros de análisis, cada uno de los cuales deberá incluir los siguientes datos: copia textual del requerimiento a que se refiere el registro; conclusiones detalladas del análisis y decisiones tomadas para enmendar el requerimiento.

# Tabla de revisión

| **Versión** | **Fecha** | **Descripción** |
| --- | --- | --- |
| V1 | 15/02/2023 | Desglose de contenidos |
| V2 | 17/02/2023 | Arreglo del documento + tareas D01 |
| V3 | 28/02/2023 | Letra más grande y mejora estética del apartado “Contenido” + tareas D02 |

# Introducción

En este documento se va a llevar a cabo el análisis de los requisitos implementados individualmente por el miembro José Joaquín Rojas Romero.

# Contenidos

# Para este entregable se han desarrollado las siguientes tareas individuales:

1. **Operations by anonymous principals on user accounts: Sign up to the system and become a lecturer.**

* Decisiones: tomé como ejemplo Acme-Jobs 23.3.1 y realicé los mismos pasos que realiza un *provider* o un *consumer* pero aplicado al rol *Lecturer*. No tuve que tomar decisiones complejas, tan solo elegir la ubicación del botón.
* Conclusión: Con esta tarea he comprendido las bases para convertir a un rol *any* en un rol creado por nosotros como es el caso de *Lecturer*. Quizás es una tarea más compleja de lo que parece y por eso tiene una estimación en tiempo algo superior de lo que debiera.

1. **Operations by lecturers on user accounts: Update their profiles.**

* Decisiones: En esta tarea dudé porque cuando fui a comprobar su implementación ya estaba hecha en su mayoría por lo que no he tenido que tomar decisiones importantes.
* Conclusión: Es útil comprobar que un rol *Lecturer* puede actualizar sus datos personales porque me sirve como base para entender el comando *update* en los servicios que he desarrollado posteriormente.

1. **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).**

* Decisiones: Esta entidad sigue un esquema parecido a la entidad anterior. En este caso tuve que elegir entre declarar el atributo “*time*” como tipo entero o como tipo “*date*”, siendo el primero el candidato final.
* Conclusión: Después de crear esta entidad he comprendido mejor como declarar cada tipo de atributo según el tipo de valor que queremos almacenar.

1. **[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.**

* Decisiones: Para poder crear un *course* es necesario tener añadidas una o varias *lectures* y para poder publicarlo es necesario que exista un mayor o igual número de *lectures* de tipo *HANDS\_ON* respecto al tipo *THEORY*. Estas acciones de añadir o eliminar *lectures* a un *course* pasan por la entidad intermedia *CourseOfLecture* la cual provee de estas acciones desde el formulario de creación de un *course*. En caso de que un *course* no disponga de las *lectures* necesarias para poder ser publicado, al hacer click en el botón de publicar un *course* nos aparecerá un mensaje avisándonos de que el *course* no puede ser publicado por el motivo comentado al incio.
* Conclusión: He dedicado demasiado tiempo a esta tarea debido a su complejidad pero también he aprendido suficiente a cerca de como guardar variables y crear atributos derivados como en el caso de *courseType* y luego poder usarlos en las vistas. Con una mejor interiorización de los conceptos teóricos quizá podría haber tardado menos tiempo en su implementación.

1. **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.**

* Decisiones: Al principio pensaba en poner el botón en el navegador para estar a un click de crear una *Lecture* pero tras obtener diversos errores se terminó implementando de una manera similar en la que se implementa un *duty* en el ejemplo de Acme-Jobs 23.3.31. Después de comprender como implementar dicho requisito tuve complicaciones a la hora de relacionar este requisito con el anterior y poder crear una *lecture* dentro de un *course*. Una vez vez publicada la *lecture*, ésta aparece disponible en el botón *Add lecture* dentro de un curso en proceso de creación.
* Conclusión: Complementando los conocimientos del anterior requisito, este me ha requerido menos tiempo debido a que ya conocía las bases sobre como implementar operaciones CRUD. Para el siguiente entregable espero arreglar el problema de que una vez creado un curso con sus respectivas asignaturas, cuando hacemos click en las asignaturas relacionadas al curso desde el *show* obtengo un error de tipo *AssertionError()*.

1. **Operations by lecturers on lecturer dashboards:**

* Decisiones: No sabía si emplear *Maps* o *Doubles* e *Integers* a la hora de crear los atributos del dashboard de *lecturer*. Por otro lado se tomó la decisión de crear una carpeta y servicios independientes para cada dashboard de la aplicación en vez de tener un dashboard general con los parámetros de todos los roles.
* Conclusión: Me ha faltado implementar todas las *querys* por falta de tiempo. No obstante se deja creada y funcionando a la espera de que en el siguiente entregable se terminen de implementar todas las consultas para que muestre lo que pide el enunciado.

1. **Produce an analysis report.**

* Decisiones: En este documento no se han tomado grandes decisiones más allá de cambiar el formato de las tareas desarrolladas a un formato más claro y concreto.
* Conclusión: He aprendido que tener una buena plantilla desde el comienzo de cada entregable facilita mucho su desarrollo y ayuda a emplear menos tiempo en mostrar la información y dedicar todo el tiempo a la calidad de esta. Además de que es muy positivo mostrar las conclusiones de los trabajos desarrollados para mejorar.

1. **Produce a planning report.**

* Decisiones: En este documento no se han tomado grandes decisiones más allá de cambiar el formato de las tareas desarrolladas a un formato tabular más comprensible que resume mejor la información.
* Conclusión: He aprendido que tener una buena plantilla desde el comienzo de cada entregable facilita mucho su desarrollo y ayuda a emplear menos tiempo en mostrar la información y dedicar todo el tiempo a la calidad de esta.

1. **Informal testing**

* Conclusión: Como no se han tenido que tomar decisiones importantes en el testing, informal más allá de emplear otro navegador distinto de Mozilla para visualizar los cambios en el proyecto, he obtenido como conclusión que es imprescindible hacer al menos una prueba de manera informal simulando ser un usuario medio para verificar que todo funciona, al menos, de la manera esperada.

# Bibliografía

* Documento 08 Annexes, de los contenidos de la plataforma virtual de esta asignatura.