**Informe de análisis sobre tarea 053**

Integrantes del grupo C1.026:

* Ignacio Blanquero Blanco ([ignblabla@alum.us.es](mailto:ignblabla@alum.us.es))
* Adrián Cabello Martín ([adrcabmar@alum.us.es](mailto:adrcabmar@alum.us.es))
* María de la Salud Carrera Talaverón ([marcartal1@alum.us.es](mailto:marcartal1@alum.us.es))
* Joaquín González Ganfornina ([joagongan@alum.us.es](mailto:joagongan@alum.us.es))
* Natalia Olmo Villegas ([natolmvil@alum.us.es](mailto:natolmvil@alum.us.es))

Fecha: Sevilla, 8 de Marzo 2024

Contenido

[Resumen del Informe 3](#_Toc160789386)

[Historial de versiones 4](#_Toc160789387)

[Introducción 5](#_Toc160789388)

[Informe de análisis – Acme-SF-D02 6](#_Toc160789389)

[Registro de análisis 1 6](#_Toc160789390)

[Requisito 6](#_Toc160789391)

[Análisis y decisiones 6](#_Toc160789392)

[Validación por el profesor 6](#_Toc160789393)

[Registro de análisis 2 7](#_Toc160789394)

[Requisito 7](#_Toc160789395)

[Análisis y decisiones 7](#_Toc160789396)

[Validación por el profesor 7](#_Toc160789397)

[Registro de análisis 3 8](#_Toc160789398)

[Requisito 8](#_Toc160789399)

[Análisis y decisiones 8](#_Toc160789400)

[Validación por el profesor 8](#_Toc160789401)

[Registro de análisis 4 9](#_Toc160789402)

[Requisito 9](#_Toc160789403)

[Análisis y decisiones 9](#_Toc160789404)

[Validación por el profesor 9](#_Toc160789405)

[Registro de análisis 5 10](#_Toc160789406)

[Requisito 10](#_Toc160789407)

[Análisis y decisiones 10](#_Toc160789408)

[Validación por el profesor 10](#_Toc160789409)

[Conclusiones 11](#_Toc160789410)

[Bibliografía 12](#_Toc160789411)

# Resumen del Informe

El propósito primordial de este informe consiste en ofrecer una exposición detallada del análisis efectuado sobre todos los tipos de requisitos establecidos en la segunda fase del proyecto, abordando asimismo las decisiones adoptadas para su implementación. Además, se pretende establecer conexiones pertinentes, de ser aplicable, con las validaciones y observaciones emitidas por el profesor a cargo de la asignatura, con el propósito de garantizar la cohesión y excelencia del trabajo realizado.

# Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Contenidos** | **Fecha** | **Contribuyente** |
| V0.1 | Estructura inicial Analysis Report D02 | 01/03/2024 | Ignacio Blanquero Blanco |
| V0.2 | Actualización Analysis Report D02 | 05/03/2024 | Ignacio Blanquero Blanco |
| V0.3 | Actualización Analysis Report D02 | 07/03/2024 | Ignacio Blanquero Blanco |
| V0.4 | Revisión y finalización del informe | 08/03/2024 | Ignacio Blanquero Blanco |

# Introducción

En esta segunda entrega, se nos encomendó la implementación de varios requisitos de información, así como un requisito de prueba, a diferencia de la primera entrega, que se limitaba a un único requisito funcional. Este cambio implica un incremento notable en la carga de trabajo, haciendo imperativa una gestión eficiente por parte de Joaquín González Ganfornina, quien ha desempeñado el rol de Project Manager durante esta fase. Por consiguiente, la labor de coordinación y distribución de tarea se vuelve esencial para asegurar el éxito del proyecto.

A continuación, se procederá a detallar minuciosamente cada uno de los requisitos implementados individualmente por Ignacio Blanquero Blanco, con el fin de proporciona una comprensión exhaustiva del trabajo realizado en la segunda etapa del proyecto.

# Informe de análisis – Acme-SF-D02

## Registro de análisis 1

### Requisito

Un módulo de formación consiste en una o varias actividades formativas de corta duración destinadas a ampliar o actualizar conocimientos y habilidades relacionados con el tema de un proyecto. El sistema debe almacenar los siguientes datos sobre ellos: un código (patrón "[A-Z]{1,3}-[0-9]{3}", no en blanco, único), un momento de creación (en el pasado), algunos detalles que describan el módulo de formación (no en blanco, inferior a 101 caracteres), un nivel de dificultad ("Básico", "Intermedio" o "Avanzado"), un momento de actualización opcional (en el pasado, después del momento de creación), un enlace opcional con más información y un tiempo total estimado.

### Análisis y decisiones

Dado el nivel de complejidad relativamente bajo asociado a este requisito, no se han presentado contratiempos significativos en su implementación. El proceso consistió principalmente en la declaración de atributos conforme a las instrucciones estipuladas en la descripción, las cuales se mostraron claras y precisas. Con el objetivo de implementar este requisito, ha sido creado el paquete “trainingModules”, dentro del cual se ha establecido la entidad “TrainingModule.java”.

Los atributos “code”, “details” y “link” son atributos de tipo String, mientras que los atributos “creationMoment” y “updateMoment” son de tipo Date. Asimismo, para declarar el atributo “difficultyLevel” se ha tenido que crear un tipo enumerado dentro del paquete “trainingModules” donde definir los posibles valores a tomar. Por último, se ha optado por definir el atributo “totalTime” como un tipo Integer, que hace referencia al tiempo total estimado (en horas). Cada uno de los atributos ha sido implementado siguiendo las restricciones descritas en el enunciado del requisito. Es importante resaltar que, al momento de definir los requisitos, he tomado la decisión de marcar como obligatorios aquellos atributos para los cuales el requisito correspondiente no especificaba su obligatoriedad. Esto se logró asignándoles la etiqueta @NotNull, garantizando así su carácter obligatorio en el sistema.

La única funcionalidad que no ha podido ser incorporada en esta entrega es la restricción que impone que el momento de actualización (atributo “updateMoment”) debe darse después del momento de creación. Esta limitación requiere un servicio adicional y la definición de la lógica de negocio asociada a los servicios, aspectos que aún no han sido tratados en profundidad en la asignatura. En consecuencia, se ha decidido postergar la implementación de esta característica para futuras entregas.

En adición, se han establecido dos relaciones de Tipo “@ManyToOne”. La primera de ellas apunta hacia “Project”, dado que se especifica que un módulo de formación implica la realización de una o múltiples actividades de formación a corto plazo relacionadas con el proyecto en cuestión. La segunda relación de este tipo se dirige hacia “Developer”, ya que se da a entender que en estos módulos de formación están involucrados los desarrolladores.

### Validación por el profesor

El profesor ha revisado este requisito en la segunda clase de laboratorio y le ha parecido correcto.

## Registro de análisis 2

### Requisito

Cada módulo de formación se compone de sesiones de formación. El sistema debe almacenar los siguientes datos sobre ellas: un código (patrón "TS-[A-Z]{1,3}-[0-9]{3}", no en blanco, único), un periodo de tiempo (al menos una semana antes del momento de creación del módulo de formación, al menos una semana de duración), una ubicación (no en blanco, inferior a 76 caracteres), un instructor (no en blanco, inferior a 76 caracteres), un correo electrónico de contacto obligatorio y un enlace opcional con más información.

### Análisis y decisiones

Dado el carácter relativamente sencillo de este requisito, su implementación transcurrió sin contratiempos destacables. La tarea principal consistió en la declaración de los atributos siguiendo las instrucciones definidas en el requisito, las cuales se mostraron claras y comprensibles. En este sentido, se ha creado un paquete denominado “trainingSessions”, dentro del cual se ha establecido la entidad “TrainingSession.java”.

Los atributos “code”, “location”, “instructor”, “contactEmail” y “link” han sido declarados de tipo String, teniendo en cuenta las restricciones de cada uno de ellos. Asimismo, para definir el atributo “period” se han tenido que definir dos fechas (ambas de tipo Date), haciendo referencia la primera de ellas al comienzo del período y la otra al final de este.

Además, se ha tomado la decisión de establecer una relación @ManyToOne hacia la entidad “TrainingModule.java”, en consonancia con la especificación que indica que los módulos de formación están compuestos por sesiones de entrenamiento. Esta relación se establece para vincular de manera efectiva las sesiones de formación con sus respectivos módulos de formación.

### Validación por el profesor

En las sesiones de revisión de laboratorio, el profesor evaluó este requisito y destacó la importancia de revisar el atributo "period". Asimismo, recomendó buscar orientación en el foro de Enseñanza Virtual para abordar las correcciones necesarias en dicho atributo.

## Registro de análisis 3

### Requisito

El sistema debe manejar los paneles para desarrolladores con los siguientes datos: número total de módulos de formación con un momento de actualización; número total de sesiones de formación con un enlace; tiempo medio, desviación, mínimo y máximo de los módulos de formación.

### Análisis y decisiones

Dado el nivel de complejidad relativamente bajo asociado a este requisito, su implementación se llevó a cabo sin mayores dificultades. En el paquete “forms”, se ha creado la clase “DeveloperDashboard.java” de acuerdo con las especificaciones requeridas.

En cuanto a la elección de los tipos de datos para los métodos correspondientes, se ha optado por utilizar tipos primitivos, dado que estos no admiten valores nulos, lo que simplifica su implementación y mejora la integridad de los datos.

Es importante señalar que al realizar los cálculos como la media, desviación y tiempos máximo y mínimo, la presencia de un conjunto vacío podría generar problemas. Por lo tanto, para los métodos “averageTrainingModuleTime”, “deviationTrainingModuleTime”, “minimumTrainingModuleTime” y “maximumTrainingModuleTime”, se ha determinado que devuelvan un valor de tipo Double. Esta elección se justifica en que el tipo Double admite valores nulos, lo que puede prevenir complicaciones en situaciones donde se encuentre un conjunto vacío.

### Validación por el profesor

Después de la evaluación inicial del requisito durante la segunda sesión de revisión del proyecto de laboratorio, el profesor recomendó el uso de tipos primitivos como int y double en lugar de Integer y Double, con el fin de evitar la posibilidad de recoger resultados nulos. Asimismo, destacó la importancia de considerar adecuadamente el tipo de dato al calcular medias, ya que la media de un conjunto vacío podría ocasionar problemas si no se declara correctamente.

## Registro de análisis 4

### Requisito

Produzca datos de muestra variados para probar su aplicación de manera informal. Los datos deben incluir dos cuentas de desarrollador con credenciales "desarrollador1/desarrollador1" y "desarrollador2/desarrollador2".

### Análisis y decisiones

Dado que este requisito no implicaba una gran complejidad, su implementación se realizó sin contratiempos significativos. Se ha generado un archivo llamado “developer.csv” y se han realizado modificaciones en el archivo “user-account.csv”, ambos ubicados en la ruta “src/main/webapp/WEB-INF/resources/sample-data”.

En relación al archivo “user-account.csv”, se ha optado por mantener la misma sintaxis al declarar las nuevas credenciales. Una vez completados ambos archivos, se procedió a poblar la base de datos utilizando mariaDB y Dbeaver. Posteriormente, se ejecutaron los lanzadores correspondientes para verificar la finalización exitosa de la tarea, y los resultados obtenidos fueron conforme a lo esperado.

Para poblar las demás entidades, se siguió una lógica específica: se creó una nueva fila en la tabla de la base de datos por cada prueba de los valores de los atributos. Se consideraron cinco valores para cada atributo, incluyendo límites, valores cercanos a los límites y valores intermedios. Este proceso se llevó a cabo sin mayores dificultades, creando nuevos archivos CSV para cada entidad.

### Validación por el profesor

En esta ocasión, debido a la limitación de tiempo en las sesiones de laboratorio, el profesor no ha podido realizar una revisión exhaustiva de esta sección. Sin embargo, estamos seguros de haber completado esta parte de manera adecuada, ya que al revisar el registro en Dbeaver, hemos constatado que las tablas se han poblado correctamente.

## Registro de análisis 5

### Requisito

Hay un nuevo rol específico de proyecto llamado desarrollador, que tiene los siguientes datos de perfil: titulación (no en blanco, más corto que 76 caracteres), una especialización (no en blanco, más corto que 101 caracteres), lista de habilidades (no en blanco, más corto que 101 caracteres), un correo electrónico y un enlace opcional con más información.

### Análisis y decisiones

Debido a la naturaleza relativamente sencilla de este requisito, su implementación transcurrió sin mayores complicaciones. La tarea primordial implicaba la declaración de atributos, considerando las restricciones definidas en el requisito, todos ellos de tipo String. Como resultado, se ha generado una clase denominada "Developer.java", la cual ha sido ubicada dentro del paquete "roles".

### Validación por el profesor

Durante la segunda sesión de laboratorio no hubo tiempo para revisar dicha tarea.

# Conclusiones

En esta segunda fase del proyecto, nos hemos enfrentado a un incremento notable de la carga de trabajo al implementar múltiples requisitos de información y un requisito de prueba, como parte de las responsabilidades individuales de cada miembro del grupo. La gestión efectiva de estas tareas ha sido liderada por Joaquín González Ganfornina, quien ha desempeñado el rol de Project Manager durante esta fase del proyecto.

En general, la implementación de todos los requisitos asignados para esta entrega se ha realizado de manera eficiente y sin mayores complicaciones. Entre las decisiones más destacadas, se encuentra la postergación de la restricción relacionada con el atributo “updateMoment” de la clase “TrainingModule.java”, que impone que el momento de actualización debe darse después del momento de creación. Asimismo, la restricción del atributo “period” de la clase “TrainingSession.java” también será implementada más adelante, pues se requiere el uso de un servicio para ello. Además, se ha optado por utilizar tipos primitivos siempre y cuando no haya que hacer cálculos con conjuntos vacíos.

En resumen, hemos logrado una exitosa implementación de todos los requisitos necesarios para esta entrega, y los comentarios proporcionados por el profesor durante las sesiones de revisión han sido favorables.

# Bibliografía

Intencionalmente en blanco.