

# INFORME DE ANÁLISIS

---

**GRUPO: C1.02.11**

María José Ruiz Vázquez ([marruivaz1@alum.us.es](mailto:marruivaz1@alum.us.es))

## Índice

<b>Resumen ejecutivo</b>	<b>1</b>
<b>Historial de versiones</b>	<b>2</b>
<b>1. Introducción</b>	<b>2</b>
<b>2. Contenidos</b>	<b>3</b>
<b>a. Listado de registros de análisis</b>	<b>3</b>
i. Errores Actualización del Framework	3
ii. Operación CRUD auditoría e incorporación de un desplegable con CURSOS	4
iii. Operación CRUD registros de auditorías	4-5
iv. Operación Estadísticas de una auditoría	5
v. Operación CRUD banner	5
<b>3. Conclusiones</b>	<b>6</b>
<b>4. Bibliografía</b>	<b>6</b>

## **1. Resumen ejecutivo**

En este reporte se desarrollará toda la información referente al análisis de los requisitos más complejos de cada uno de los entregables.

Para ello, contaremos con una copia literal del requisito al que se refiere el registro y daremos las conclusiones detalladas del análisis y decisiones tomadas para subsanar dicho requisito, si se diera el caso, aportamos un enlace a la validación realizada por un profesor.

## 2. Historial de versiones

Fecha	Versión	Descripción de los cambios	Sprint
15/02/2023	1.0	Creación de los documentos para el “Deliverable 1” e inicialización del proyecto	1
1/03/2023	2.0	Deliverable 2 :Data models	2
21/04/2023	3.0	Deliverable 3: Implementing Features	3

## 3. Introducción

En primer lugar, decir que en este documento se analizarán y explicarán cada uno de los requisitos abarcados por los entregables .

Los pasos a seguir para afrontar este análisis será leer detenidamente todos los requisitos del nuevo delivery, hecho esto, tendremos que distinguir entre dos tipos de requisitos:

Por una parte, los requisitos que por su simplicidad o bien por ser de los primeros en el comienzo de este proyecto no necesitaran un análisis previo.

Por otra parte, los requisitos de código que por su dificultad necesitaran de un análisis para su comprensión y valoración de cómo empezar a desarrollarlos. Es evidente que cuando vayamos avanzando en el proyecto los requisitos de las entregas empezarán a complicarse cada vez más.

El análisis exhaustivo de estos requisitos será muy importante ya que este nos dará una idea previa de lo que tendrá que hacer el desarrollador que le toque realizar esa tarea, viéndose beneficiado el tiempo como el coste de la misma.

Además, nos permitirá realizar una repartición más equilibrada de los requisitos ya que al conocer la dificultad, balanceamos la carga de trabajo para que unos compañeros no realicen más trabajos que otros.

En definitiva, creo que este documento nos ayudará a todos los integrantes del grupo tanto a estar en contexto de los requisitos más difíciles del sprint, como a conocer las conclusiones detalladas del análisis y las decisiones tomadas para subsanar el requisito.

## **4.Contenidos**

En este punto encontraremos dos partes diferenciadas del requisito que será analizado:

- Una copia literal del requisito al que se refiere el registro.
- Conclusiones detalladas del análisis y decisiones tomadas para subsanar el requisito.

### **4.1. Listados registro de análisis**

Comenzamos con un entregable de mayor magnitud donde vamos a visualizar todas aquellas entidades que previamente habíamos creado y donde vamos a poder interactuar con ellas dándole todas las funcionalidades que nos pide en los requerimientos. En primer lugar, hemos tenido que crear las features compuesta por controlador, servicio y repositorio. Los servicios nos van a permitir añadir aquellas restricciones que anteriormente nos iban anunciando en los requerimientos, aumentando más la dificultad de esta segunda entrega. Por otro lado, en las vistas implementaremos un sistema de carpetas que nos servirá para que el Framework reconozca las peticiones que les estamos mandando, esto nos ha ayudado a conocer más cómo funciona este.

A continuación voy a explicar todos los problemas que he tenido en este entregable que no han sido pocos.

#### **4.1.1.Errores Actualización del Framework**

A lo largo del curso los profesores nos van anunciando en la plataforma las actualizaciones de este, para ayudarnos a implementar toda la funcionalidad que el entregable que vamos a empezar a realizar funcione correctamente. Pues en la actualización 23.3.0 fue fatídica ya que no era capaz de saber que problema me estaba ocasionando, pero no podía arrancar mi proyecto. Esto me cogió justo en semana santa por lo que el avance de mi proyecto fue prácticamente nulo, y el follow up siguiente a esto no fue nada aprovechado ya que yo no tenía dudas sobre nada, puesto que todo el código que empecé a hacer no lo podía probar. Ese Martes antes habían anunciado una nueva actualización y con la ayuda de un compañero y los videos de apoyo del foro pude actualizarlo con éxito lo que me permitió avanzar pero con el tiempo muy ajustado.

Da comienzo a los errores propios de la programación.

#### **4.1.2. Operación CRUD auditoría e incorporación de un desplegable con CURSOS**

Task 10-D03: [Obligatorio] Operaciones de los auditores en las auditorías:

- Enumerar las auditorías que han creado.
- Mostrar los detalles de sus auditorías.
- Crear, actualizar o eliminar sus auditorías. Las auditorías se pueden actualizar o eliminar siempre que no se hayan publicado.

#### **Análisis y toma de decisiones:**

En este requerimiento tenía que diferenciar entre dos clases de auditorías, las que estaban publicadas o las que seguían siendo borradores, ya que esto te permitía distintas funcionalidades. En primer lugar, para hacer esta distinción lo que me planteé es incorporar un nuevo atributo a la clase auditoría de tipo Boolean.

Una vez listada todas las auditorías que pertenecieran al auditor que estaba logueado en ese momento solo aquellas que fueran borrador tendrán la opción de ser actualizadas, eliminadas o publicadas, y ambas compartirán el poder ver sus registros de auditoría.

Otra de las dificultades que tenía este requerimiento, es que ya en la entrega pasada nos pedían que las auditorías irían asociadas a un curso, por lo que la funcionalidad de crear una auditoría debía de incorporar un curso asociado, para ello pensé en incorporar al formulario un “choise” con el parámetro id del curso que previamente habíamos conseguido en las consulta del repo. Para que al auditor logueado pudiera ver todos aquellos cursos, ya que este no tiene porque saber que cursos existen e incorporar un dato a mano no lo veía lógico.

#### **4.1.3. Operación CRUD registros de auditorías**

Task 11-D03: [Obligatorio] Operaciones de los auditores en los registros de auditoría:

- Enumerar los registros de auditoría en sus auditorías.
- Mostrar los detalles de sus registros de auditoría.
- Crear un nuevo registro de auditoría para sus auditorías mientras no se hayan publicado. En raras ocasiones, un auditor puede añadir registros de auditoría de corrección a una auditoría que ya ha sido publicada; esto requiere confirmación; los registros de auditoría de corrección deben estar algo resaltados cuando se muestren.
- Actualizar o eliminar los registros de sus auditorías mientras no se hayan publicado.

#### **Análisis y toma de decisiones:**

Este es el requisito que más me ha costado hacer no por el código en sí, que también, sino por la cantidad de ERRORES que me han salido en casi todas las funcionalidades y que no identificaba a que estaba haciendo alusión.

Comenzamos con la “simple” lista de registro de auditoría, aunque iba con la confianza que ya había realizado la lista de auditoría por lo que esto iba a ser pan comido, pero fue sin duda el error más frustrante que he tendido en todo el proyecto hasta ahora, cuando parecía que todo estaba correcto me decidí a levantar la aplicación y me encuentro con un ERROR 404 not found. Comprobé que en la vista estuviera pasando el endpoint bien, que coincidiera con el nombre de las carpetas que le había pasado y todo estaba bien, pero aquello seguía igual. Seguí avanzando con las diferentes funcionalidades de ver, actualizar, etc. Aunque al estar incorporadas dentro de la lista pues no podía comprobar que funcionará y cuando la intentaba meter a mano en el navegador tampoco, así que después de días, lo vi con el profesor en la clase de seguimiento. Donde él me explicó la importancia de que la carpeta de mi feature se llamará igual que mi entidad y como no era así, pues posiblemente ese sería el error por el cual no estaba encontrando mi lista. Aunque no fue el único error que tenía, pasé de tener un 404 a un 500, no sabía que tener un ERROR 500 era una maravilla ya que eso solo podía ser que algún dato le están pasando mal, pero oye ya me entraba la ruta. Así que ahora, al descubrir ese dato erróneo, resultó ser el parámetro de la request que no era el mismo que le estaba pasando en la vista por lo que no lo encontraba.

Una vez teniendo acceso a la lista empecé a comprobar si los demás servicios me funcionaban y me vi con el mismo problema que acabo de explicar pero como ya reconoció el error me fue más fácil solucionarlo.

#### **4.1.4. Operación Estadísticas de una auditoría**

Task 11-D03: Operaciones de los auditores en los cuadros de mando de auditores:

- Mostrar sus cuadros de mando de auditores.

#### **Análisis y toma de decisiones:**

Este fue el último requerimiento en implementar y empecé poco a poco con el número total de auditoría porque aunque las consultas del repo, en la “query” tienes opciones a usar distintos “métodos de contar, max, min, etc” no estaba segura de como hacerlo. Y al correr el proyecto me encuentro con un error de que no se podía resolver “totalNumberOfAudit” y después de muchas comprobaciones, me di cuenta que los atributos que estaba pasándole al “bin” no eran iguales que los que había declarado en mi entidad previamente. Al final, despistes de este tipo hacen que se pierda mucho tiempo.

#### **4.1.5. Operación CRUD banner**

Task 28-D03: [OBLIGATORIO] Operaciones de los administradores sobre los banners:

- Listar los banners y mostrar sus detalles.
- Crear, actualizar o borrar un banner

Este requisito lo había implementado cuando no tenía acceso al proyecto debido al problema del framework pero como fue el último en corregir, después de todos los de mi parte individual, aunque me encontré con errores pero fue más fácil solventarlo, ya que con diferencia a la primera vez, no sabía identificar los distintos errores. Al igual que como explico en el individual los problemas que tuve fue relacionado con los parámetros del id del

banner que está en la request, para ello realice un printl para saber cual era y lo modifique en la vista.

## **5. Conclusión**

Conclusivamente este documento nos va servir de apoyo para contar cómo hemos realizado los requisitos más complejos de los sprint, y detallar qué decisiones importantes hemos tomado y como se han solventado los posibles errores.

Además nos ayudará a conocer como el resto de desarrolladores se organizan, analizan sus tareas y qué métodos usan para corregir sus errores. Sirviendo de referente a los demás miembros del grupo.

Aunque en esta primera entrega al ser tareas muy sencillas, en la reunión grupal que tuvimos al comienzo entre todos fuimos revisando las tareas y tomando la decisión de cómo se iban a desarrollar. Siendo luego un miembro del grupo el que se encarga de plasmar dichas decisiones en este documento.

Pero el acuerdo final, es que para las próximas entregas cada desarrollador después de la asignación y realización de sus tareas, tendrá que analizar aquellas tareas que le han resultado de mayor complejidad y contar paso a paso como ha realizado la tarea y con qué errores se ha encontrado.

## **6. Bibliografía**

Intencionadamente en blanco.