

# INFORME DE ANÁLISIS

---

**GRUPO: C1.02.11**

Francisco Javier de la Prada Prados ([fraprapra1@alum.us.es](mailto:fraprapra1@alum.us.es))

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Contenidos</b>	<b>2</b>
<b>2.1. Listado registro de análisis</b>	<b>2</b>
<b>2.1.1 Tareas grupales</b>	<b>3</b>
<b>2.1.2 Tareas individuales</b>	<b>4</b>
<b>3. Conclusión</b>	<b>8</b>
<b>4. Bibliografía</b>	<b>8</b>

## Resumen ejecutivo

En este reporte se desarrollará toda la información referente al análisis de los requisitos individuales de cada uno de los entregables que tengo que realizar

Para ello, procederé a presentar los requisitos que se me han propuesto para la entrega y ofreceré unas conclusiones detalladas del resultado de realizar un análisis de este, además de exponer las decisiones tomadas para realizar correctamente el requisito en cuestión. En caso de que el profesor realice una validación de los requisitos previo a la entrega se aportará un enlace a la validación realizada (foro de la ev o correo electrónico).

## 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
20/02/2023	2.0	Realización de los documentos para el “Deliverable 2”	2
23/03/2023	3.0	Realización de los documentos para el “Deliverable 3”	3

## **1. Introducción**

En primer lugar, en este documento se analizarán y explicarán cada uno de los requisitos que me han sido asignados en los entregables, tanto grupales como individuales .

Los pasos a seguir para afrontar este análisis será leer detenidamente todos los requisitos del nuevo delivery y posteriormente realizar una división entre dos tipos de requisitos:

- Los requisitos que por su simplicidad o bien por ser de los primeros en el comienzo de este proyecto no necesitaran un análisis previo.
- Los requisitos de código que por su dificultad necesitarán 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, llegado el punto en el que será necesario un análisis de todos los requisitos que me sean asignadas, así como irá aumentando el análisis sobre estos para asegurarme de realizar un buen trabajo y seguir las buenas prácticas de la asignatura.

El análisis exhaustivo de estos requisitos será muy importante ya que este me dará una idea previa de lo que debo realizar para llevar a cabo con éxito la tarea, viéndose beneficiado tanto el tiempo como el coste de la misma.

En definitiva, este documento me ayudará 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.

## **2. 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.

### **2.1. Listado registro de análisis**

Voy a dividir este punto entre los requisitos grupales que tengo asignados y los propios del documento de requisitos individuales.

### 2.1.1 Tareas grupales

#### **D3-21: Operaciones de cualquier persona en los “Peeps”**

- Listar los peeps y mostrar sus detalles
- Publicar un peep. Por defecto, el apodo debe mantenerse en blanco si la cuenta es anónima; en otro caso, debe estar relleno con el nombre completo del usuario; de todos modos, puede cambiarse.

#### Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 3-Implementing features: Session 1-Display features**”, donde se explica claramente cómo realizar operaciones de List y Show con los controladores y servicios.

Con estas operaciones puedo listar todos los cursos mediante un servicio usando consultas de SQL en un repositorio y en caso de querer acceder a los detalles de los cursos listados puedo hacer uso de un servicio con la operación Show que y una consulta SQL para mostrar los detalles del curso seleccionado.

También será necesario revisar la “**Lesson 3-Implementing features: Session 2-Edition features**”, donde se explica claramente cómo realizar operaciones CRUD (Create, Read, Update y Delete) siguiendo las pautas establecidas por el framework.

Para esta tarea en concreto necesito la operación Create que me permitirá crear un “peep” y publicarlo al momento en la página para que todo el mundo pueda verlo.

Además en el workspace (actualizado) proporcionado por los profesores vienen ejemplos prácticos de cómo realizar correctamente estas operaciones de List y Show además de ejemplos de consultas SQL, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo la tarea de forma correcta.

Por tanto, bastará con seguir las lecciones teóricas y ejemplos prácticos para realizar esta tarea.

#### **D3-30: Cambios de dinero**

El sistema debe mostrar los montos de dinero tal como los ingresan los usuarios, pero también sus intercambios de dinero correspondientes según la moneda del sistema si sus ratios de cambio conocidos son recientes. Es responsabilidad de los estudiantes encontrar el servicio de tipo de cambio adecuado ¡La Universidad de Sevilla o sus afiliados individuales no cubrirán ninguna responsabilidad implícita o explícita si los estudiantes contratan servicios de pago por uso! Este requisito debe cumplirse en esta y en cada entrega posterior de grupo o individual para que se considere cumplido.

### Análisis y toma de decisiones

Para realizar esta tarea puedo basarme en la tarea de “MoneyExchange” que está realizada en el proyecto de Acme-Jobs 23.3.0 que viene en los archivos del framework proporcionado en la asignatura, lo cual será de bastante ayuda.

Por tanto, siguiendo el ejemplo de dichos archivos no debería encontrar problema en realizar esta tarea de forma correcta.

### 2.1.2 Tareas individuales

#### **D3-11: Operaciones de los usuarios anónimos en las cuentas de usuarios**

Registrarse en el sistema y convertirse en un “lecturer”.

### Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 3-Implementing features: Session 2-Edition features**”, donde se explica claramente cómo realizar operaciones CRUD (Create, Read, Update y Delete) con los controladores y servicios siguiendo las pautas establecidas por el framework.

Para esta tarea en concreto necesito la operación Create que me permitirá convertir a cualquier usuario que se registre en la aplicación en un “lecturer” creando la nueva cuenta dentro de un servicio.

Además en el workspace (actualizado) proporcionado por los profesores vienen ejemplos prácticos de cómo convertir a un usuario en un rol que se haya creado en la aplicación, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo la tarea de forma correcta.

Por tanto, bastará con seguir las lecciones teóricas y ejemplos prácticos para realizar esta tarea.

#### **D3-12: Operaciones de los “lecturers” en las cuentas de usuario**

Actualizar sus perfiles.

### Análisis y toma de decisiones

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 3-Implementing features: Session 2-Edition features**”, donde se explica claramente cómo realizar operaciones CRUD (Create, Read, Update y Delete) con los controladores y servicios siguiendo las pautas establecidas por el framework.

Para esta tarea en concreto necesito la operación Update que me permitirá modificar los datos que ya se han establecido en la aplicación mediante los archivos .csv . Esto se realizará en un Servicio al que irá la consulta de actualización del perfil.

Además en el workspace (actualizado) proporcionado por los profesores vienen ejemplos prácticos de cómo crear servicios con operaciones de actualización de datos, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo la tarea de forma correcta.

Por tanto, bastará con seguir las lecciones teóricas y ejemplos prácticos para realizar esta tarea.

### **D3-13: Operaciones de los usuarios registrados en los “courses”**

- Listar los cursos del sistema que estén publicados.
- Mostrar los detalles de los cursos que se pueden listar (exceptuando sus “lectures”)

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

Para realizar esta tarea debo seguir los pasos se han explicado en las clases de teoría, concretamente la “**Lesson 3-Implementing features: Session 1-Display features**”, donde se explica claramente cómo realizar operaciones de List y Show con los controladores y servicios siguiendo las pautas establecidas por el framework.

Con estas operaciones puedo listar todos los cursos mediante un servicio usando consultas de SQL en un repositorio y en caso de querer acceder a los detalles de los cursos listados puedo hacer uso de un servicio con la operación Show que y una consulta SQL para mostrar los detalles del curso seleccionado.

Además en el workspace (actualizado) proporcionado por los profesores vienen ejemplos prácticos de cómo realizar correctamente estas operaciones de List y Show además de ejemplos de consultas SQL, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo la tarea de forma correcta.

He de remarcar también que será necesario realizar una modificación a la entidad “Course” ya que necesito establecer una variable que me indique si el curso ha sido o no publicado. Crearé una nueva propiedad de tipo boolean llamada “publish” que me permita así listar sólo aquellos cursos que tienen “true” en esta variable.

Por tanto, bastará con seguir las lecciones teóricas y ejemplos prácticos para realizar esta tarea.

### **D3-14: Operaciones de los “Lecturers” en “Courses”**

- Listar los cursos que han creado.
- Mostrar los detalles de sus cursos.
- Crear, actualizar o borrar sus cursos. Los cursos pueden ser actualizados y borrados siempre que no hayan sido publicados. Para que un curso sea publicado, todas sus clases deben estar publicadas.

#### Análisis y toma de decisiones

Para realizar esta tarea la voy a dividir en dos partes:

- **Las dos primeras subtareas (List y Show)** que las realizaré siguiendo los pasos de la clase de teoría “**Lesson 3-Implementing features: Session 1-Display features**” donde se explica claramente cómo realizar operaciones de List y Show con los controladores y servicios siguiendo las pautas establecidas por el framework.
- **La última subtask (Create, Update, Delete)** que la realizaré siguiendo los pasos de la clase de teoría “**Lesson 3-Implementing features: Session 2-Edition features**” donde se explica claramente cómo realizar operaciones CRUD (Create, Read, Update y Delete) con los controladores y servicios siguiendo las pautas establecidas por el framework.

Como ya pasa con las tareas anteriores, en el workspace (actualizado) proporcionado por los profesores vienen ejemplos prácticos de cómo realizar correctamente estas operaciones de List, Show, Create, Update y Delete además de ejemplos de consultas SQL, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo la tarea de forma correcta.

Para la anotación “*Para que un curso sea publicado, todas sus clases deben estar publicadas*” será necesario añadir un nuevo atributo “publish” a lecture y course (como ya he explicado anteriormente).

### **D3-15: Operaciones de los “Lecturers” en “Lectures”**

- Listar las clases de sus cursos.
- Mostrar los detalles de sus clases.
- Crear y publicar una clase.
- Actualizar o borrar sus clases que no estén publicadas.

#### Análisis y toma de decisiones

Para realizar esta tarea la voy a dividir en dos partes:

- **Las dos primeras subtareas (List y Show)** que las realizaré siguiendo los pasos de la clase de teoría “**Lesson 3-Implementing features: Session 1-Display features**” donde se explica claramente cómo realizar operaciones de List y Show con los controladores y servicios siguiendo las pautas establecidas por el framework.
- **Las dos últimas subtareas (Create, Update, Delete)** que la realizaré siguiendo los pasos de la clase de teoría “**Lesson 3-Implementing features: Session 2-Edition features**” donde se explica claramente cómo realizar operaciones CRUD (Create, Read, Update y Delete) con los controladores y servicios siguiendo las pautas establecidas por el framework.

Como ya pasa con las tareas anteriores, en el workspace (actualizado) proporcionado por los profesores vienen ejemplos prácticos de cómo realizar correctamente estas operaciones de List, Show, Create, Update y Delete además de ejemplos de consultas SQL, por lo que puedo revisar dichos ejemplos para asegurar que estoy haciendo la tarea de forma correcta.

Remarcar también que, aunque no lo dice explícitamente, revisando el foro de dudas de la asignatura pude ver que era necesario que el “Lecturer” pudiera crear clases al margen de un curso ya que “Lecture” es una entidad que existe aparte de “Course”.

Aquí el enlace con la validación:

[https://ev.us.es/webapps/discussionboard/do/message?action=list\\_messages&course\\_id=630091&nav=discussion\\_board\\_entry&conf\\_id=3039641&forum\\_id=2062151&message\\_id=3647201](https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=630091&nav=discussion_board_entry&conf_id=3039641&forum_id=2062151&message_id=3647201)

Por tanto, será necesario crear una vista donde el “Lecturer” pueda ver todas sus clases (independientemente de si están o no en un curso) y ahí poder crear clases que no estén asociadas a un curso. Si luego queremos añadir dichas clases a un curso, puedo crear un botón dentro de la vista “show” de cursos que me de la opción de buscar aquellas clases del profesor que no estén en ese curso y que hayan sido publicadas (con una consulta SQL).

### **D3-16: Operaciones de los “Lecturers” en “LecturerDashboard”**

Mostrar sus tableros

#### *Análisis y toma de decisiones*

Esta tarea se centra principalmente en realizar consultas SQL a la base de datos para obtener los campos necesarios del tablero, que son:

- Número de clases teóricas
- Número de clases prácticas
- Media, desviación, máximo y mínimo del tiempo de las clases
- Media, desviación, máximo y mínimo del tiempo de los cursos

Después será tan sencillo como hacer lo que llevamos haciendo todo el entregable, que es realizar un Show siguiendo los pasos de la clase de teoría **“Lesson 3-Implementing features: Session 1-Display features”** donde se explica claramente cómo realizar operaciones de List y Show con los controladores y servicios siguiendo las pautas establecidas por el framework.

### **3. Conclusión**

Este documento me ha sido de gran utilidad para tener claro cómo voy a realizar las tareas asignadas además de para dar contexto al profesor que me vaya a realizar la corrección de mis entregables. Realizar un análisis previo a la realización de los requisitos más complejos de los distintos entregables me es de gran utilidad ya que puedo trazar un plan de ruta claro y conciso sobre mis resoluciones de los requisitos.

Además queda claramente detallado cuales son las decisiones más importantes que he tomado sobre el cómo desarrollar las diferente tareas tanto colectivas como individuales y como he solucionado los posibles errores que han ido apareciendo durante el desarrollo de las tareas.

De esta forma podré llevar a cabo un aprendizaje continuo en cuanto al análisis que realizó sobre las tareas que tengo asignadas, llevándome a realizar mejores análisis en las siguientes tareas que deba realizar, mejorando en mi rol como analista y dando mejores resultados a la hora de llevar a cabo las tareas futuras del proyecto.

### **4. Bibliografía**

Intencionadamente en blanco.