

# INFORME DE ANÁLISIS

---

**GRUPO: C1.02.11**

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

Pablo Quindós de la Riva ([pabquide@alum.us.es](mailto:pabquide@alum.us.es))

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

Juan Luis Ruano Muriedas ([juaruamur@alum.us.es](mailto:juaruamur@alum.us.es))

Santiago Zuleta de Reales Toro ([santizuleta11@gmail.com](mailto:santizuleta11@gmail.com))

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Contenidos</b>	<b>2</b>
<b>2.1. Listados registro de análisis</b>	<b>2</b>
<b>3. Conclusión</b>	<b>7</b>
<b>4. Bibliografía</b>	<b>7</b>

## **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.

## **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
21/04/2023	3.0	Deliverable 03: Implementing features	3

## **1. 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.

## **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. Listados registro de análisis**

#### **D3-20: Operaciones de los administradores en la configuración del sistema (Fraprapra1)**

- Mostrar la configuración del sistema.
- Actualizar la configuración del sistema.

### 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 la operación de Show con los controladores y servicios.

Al 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 Update que me permitirá actualizar los datos que haya en la configuración del sistema 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 Update 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.

### **D3-21: Operaciones de cualquier persona en los “Peeps” (Fraprapra1)**

- 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 rellenado con el nombre completo del usuario; de todos modos, puede cambiarse.

### Análisis y toma de decisiones

Para realizar esta tarea debe 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 puedes 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.

### **D03-22 Operaciones de usuarios autenticados sobre boletines (sanzultor):**

Los usuarios autenticados son cualquier usuario que haya introducido sus credenciales independientemente de su rol. Las operaciones que deben poder realizar son listar los boletines y mostrar los datos de cada uno.

#### Análisis y toma de decisiones:

Para llevar a cabo la tarea, se crean todas las clases necesarias;

- Controladores: se crean instancias y se inyectan los servicios y además se crean los comandos
- Servicios: se crean los métodos necesarios, (check, authorise, load, authenticate, bind, unbind y perform en caso de que sea necesario)
- Repositorio: se crea una instancia del repositorio a través del que se realizan todas las consultas a la base de datos

Además se crean el listado y formulario en los que se introduce cierta lógica dependiendo de la acción o requisito a implementar.

### **D03-23 Operaciones de administradores sobre boletines (sanzultor):**

Los administradores son los usuarios que tienen rol “administrator”. Las operaciones que deben poder realizar son crear, actualizar, eliminar, listar los boletines y mostrar los datos de cada uno.

#### Análisis y toma de decisiones:

Para llevar a cabo la tarea, se crean todas las clases necesarias;

- Controladores: se crean instancias y se inyectan los servicios y además se crean los comandos
- Servicios: se crean los métodos necesarios, (check, authorise, load, authenticate, bind, unbind y perform en caso de que sea necesario)

- Repositorio: se crea una instancia del repositorio a través del que se realizan todas las consultas a la base de datos

Además se crean el listado y formulario en los que se introduce cierta lógica dependiendo de la acción o requisito a implementar.

### **D3-24: Operaciones de mandantes autenticados sobre ofertas (pabquide):**

Listar las ofertas en el sistema y mostrar sus detalles.

#### *Análisis y toma de decisiones:*

Para llevar a cabo la tarea, se crean las clases de Java correspondientes:

- AuthenticatedOfferController: Donde metemos el controlador que te lleva los servicios a las vistas (list y show ).
- AuthenticatedOfferListService: Servicio de listado.
- AuthenticatedOfferRepository: Repositorio donde se ponen las Querys.
- AuthenticatedOfferShowService: Servicio del formulario con los detalles.

Y también se realizan las vistas siguientes:

- form.jsp: Vista del formulario.
- list.jsp: Vista del listado.

Finalmente se actualizan los mensajes en inglés y en español.

### **D3-25: Operaciones de administradores sobre ofertas (pabquide):**

Listar las ofertas en el sistema y mostrar sus detalles.

Create, update y delete de oferta.

#### *Análisis y toma de decisiones:*

Para llevar a cabo la tarea, se crean las clases de Java correspondientes:

- AdministratorOfferController: Donde metemos el controlador que te lleva los servicios a las vistas (list, show, create, update y delete ).
- AdministratorOfferListService: Servicio de listado.
- AdministratorOfferRepository: Repositorio donde se ponen las Querys.
- AdministratorOfferShowService: Servicio del formulario con los detalles.
- AdministratorOfferCreateService: Servicio de creación.
- AdministratorOfferUpdateService: Servicio de actualización.
- AdministratorOfferDeleteService: Servicio de borrado.

Y también se realizan las vistas siguientes:

- form.jsp: Vista del formulario.

- list.jsp: Vista del listado.

Finalmente se actualizan los mensajes en inglés y en español.

### **D3-26 Operacion de autenticados en notes (juaruamur):**

- Lista de notas en los recientes 30 días y que muestre los datos
- Postear una nota. Esta operación requiere confirmación porque las notas posteadas no se pueden borrar.

#### Análisis y toma de decisiones:

Para llevar a cabo la tarea, se crean las clases de Java correspondientes:

- AuthenticatedNoteController: Donde metemos el controlador que te lleva los servicios a las vistas (list, show, create).
- AuthenticatedNoteListService: Servicio de listado.
- AuthenticatedNoteRepository: Repositorio donde se ponen las Querys.
- AuthenticatedNoteShowService: Servicio del formulario con los detalles.
- AuthenticatedNoteCreateService: Servicio de creación.

Y también se realizan las vistas siguientes:

- form.jsp: Vista del formulario.
- list.jsp: Vista del listado.

Finalmente se actualizan los mensajes en inglés y en español.

Nos hemos basado en las diapositivas de “Display Features” de la asignatura así como el proyecto de Acme jobs, en concreto la clase Announcements ha sido de especial ayuda por su parecido a los requisitos que nos pedían.

### **D3-27 Operaciones de cualquier principal sobre banners(marruivaz1):**

- Mostrar cualquiera de ellos aleatoriamente, siempre que su periodo de visualización esté activo; los banners deben mostrarse en todas las páginas de la forma menos intrusiva posible.

#### Análisis y toma de decisiones:

Pendiente de analizar y de programar **esta tarea obligatoria.**

### **D3-28 Operaciones de los administradores sobre los banners(marruivaz1):**

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

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

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.

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.

### **D3-29: Cambios de dinero (Todos)**

Fechas, cantidades de dinero y booleans deben estar internacionalizados cuando se muestran. Otros tipos de datos no se esperan que estén internacionalizados. Este requisito debe estar completado en este y en todos los entregables grupales o individuales.

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

Pendiente de analizar y de programar **esta tarea obligatoria.**

### **D3-30: Cambios de dinero (Fraprapra1)**

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.

## **3. 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.

En esta segunda entrega ha habido diferentes cuestiones conforme al desarrollo de las entidades y la clase “dashboard” pero creemos que hemos aplacado los problemas con éxito y conseguido el resultado deseado.

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.

## **4. Bibliografía**

Intencionadamente en blanco.