



Informe de Análisis

GRUPO C1.02.02

Repositorio: <https://github.com/antcardia/Acme-L3-Do3>

Autores:

Antonio Carretero Díaz (antcardia4@alum.us.es)

Guillermo Galeano de Paz (guigalde@alum.us.es)

Eduardo Pizarro López (edupizlop@alum.us.es)

David Reyes Alés (davreyale@alum.us.es)

Juan Romero González (juaromgon@alum.us.es)

Sevilla, 20 de abril de 2023

TABLA DE CONTENIDOS

1. Tabla de revisión [1]

2. Introducción [1]

3. Contenido [2]

4. Conclusión [4]

Hay un total de 4 páginas

Resumen ejecutivo

Este es un informe básico y simple de análisis del grupo, dentro del segundo sprint, donde realizaremos un seguimiento de este. Este informe se ha ido completando a lo largo del sprint donde primero hemos hecho un análisis previo a los requisitos y, una vez acabados todos los requisitos, hemos hecho un análisis final.

Tabla de revisión

Fecha	Versión	Descripción de los cambios	Sprint
14/02/2023	23.1	Creación del documento	1
22/02/2023	23.2	Primer análisis del sprint 2	2
16/03/2023	23.2	Actualización del informe de análisis	2
23/03/2023	23.3	Primer análisis del sprint 3	3
20/03/2023	23.3	Análisis final del sprint 3	3

Introducción

Este es un proyecto dedicado al desarrollo de un sistema de información web para ayudar a la organización Acme Life-Long Learning, S.A (Acme-L3) a gestionar su empresa. Esta organización es ficticia y se especializa en ayudar a los estudiantes en una gran variedad de cuestiones con la ayuda de profesores de renombre.

Para comenzar, hemos utilizado el proyecto Hello World como plantilla para dar los primeros pasos en el desarrollo del sistema. Además, vamos a hacer uso de diferentes herramientas como MariaDB y DBeaver, para la creación y gestión de base de datos, Eclipse, como nuestro espacio de trabajo, Java, como nuestro lenguaje principal de programación y Acme Framework, como nuestro framework.

Contenido

- Sprint Planning
 - Análisis: Hemos hecho la repartición de tareas grupales de manera equitativa siguiendo la siguiente regla: Cada uno se encargará de implementar repositorio, servicio, controlador y vistas de su respectiva entidad implementada en el Sprint Anterior.
- Operaciones sobre la entidad “Banner”:
 - Análisis previo: Tenemos el problema de no conocer bien el framework, y por tanto, no tener claro cómo y dónde colocar los banners. En cuanto al listado por parte de los administradores, todo correcto.
 - Análisis final: Después de aclararlo con el product owner e investigar como se podría implementar en “Acme Framework”, hemos decidido que los banners se muestren en el footer para todo el mundo, aleatoriamente por cada recarga de página, respetando la mínima intrusividad.
- Operaciones sobre la entidad “Note”
 - Análisis previo: Para realizar las operaciones CRUD y listado de Note, tenemos que esperar a que en teoría se explique la implementación del código. Por tanto, suponemos que se necesitará un repositorio, servicio y controlador para cada una de las operaciones CRUD junto a las vistas necesarias.
 - Análisis final: Tras haber atendido a todas las clases de teoría, hemos podido implementar el requisito tal y como hemos explicado en el anterior análisis.
- Vistas create, list y show para cualquier usuario:
 - Análisis previo: para la implementación de este requisito se pueden seguir las transparencias sin problemas además de contar con los ejemplos de acme-jobs. El único análisis necesario es el segundo punto del requisito, el cual he interpretado que al crear un “peep”, se pase al formulario como dato por defecto el nombre completo del usuario que lo cree, y en caso de ser un usuario anónimo dejarlo en blanco, siendo este campo modificable a voluntad.
 - Análisis final: tras la realización de la tarea, considero que mi análisis previo ha sido acertado y se pudo realizar la implementación sin mayor problema que algún error sin mucha importancia.
- Vista de Bulletin show y list para autenticados
 - Análisis previo: se nos pide que los usuarios autenticados puedan ver los distintos bulletin y sus detalles. Esto podría implementarse siguiendo las diapositivas de teoría y fijándonos en ejemplos de acme-jobs.

- Análisis final: para implementar este requisito he aplicado los conocimientos adquiridos y me he fijado en el proyecto de ejemplo que tenemos.
- Vista de Bulletin create para administradores
 - Análisis previo: se nos pide que los administradores puedan crear bulletin. Para implementarse podría coger la vista de usuarios autenticados y adaptarla a administradores.
 - Análisis final: para completar este requisito he hecho uso de los conocimientos adquiridos en teoría y del proyecto de ejemplo. La única complicación que he tenido ha sido la confirmación que había que implementar para poder crear un bulletin, pero la pude resolver sin demasiado trabajo.
- Actualización del framework:
 - Análisis: A la vista del buen funcionamiento del sistema se nos ha cedido la nueva versión de Acme Framework utilizado en el mismo. Hemos actualizado todos nuestros entornos y estamos preparados para comenzar con el tercer sprint.
 - Análisis final: Tal y como se ha comentado anteriormente, siguiendo la clase teórica, hemos actualizado el framework sin problemas.
- Vista offer show y list para autenticados.
 - Análisis previo: tras leer el requisito, he podido observar que nos está hablando de querer implementar una vista para que el usuario autenticado pueda ver las ofertas y además sus detalles de cada una.
 - Análisis final: tras terminar, no he tenido muchas complicaciones, fuera de tener que corregir palabras reservadas mal escritas, pero sin complicaciones de implementación. Para el list he decidido poner el título de la oferta con su precio, que he interpretado como la información más relevante, y todos sus atributos/información, una vez ves los detalles de cada uno.
- Vista offer show, list, create, update y delete para administradores.
 - Análisis: se puede observar que nos pide implementar la misma funcionalidad en offer, de que una persona autenticada pueda ver la vista de list y show pero además de en vez de estar autenticado ser administrador, también nos pide poder crear eliminar o actualizar las ofertas. Seguramente podamos copiar la implementación pero cambiando la configuración para que solo el administrador pueda acceder a esas vistas.
 - Análisis final: efectivamente, el proceso de la implementación consistió en copiar la implementación similar, añadiendo las otras tres vistas/funcionalidades. Al implementarlo nos dio el error de que nos faltaba en el controlador añadir los comandos básicos para las nuevas vistas y

también cambiar bien los ficheros JSP y los controladores y servicios, para que se aplicarán a administrador en vez de autenticados. En conclusión, me parece una implementación donde se ha podido reutilizar código y comprobar ciertas reglas, como que un usuario que no es administrador no puede entrar en las nuevas vistas.

- Internacionalización de moments, money y booleanos.
 - Análisis previo: Según lo visto en clase, considero que para la realización de esta tarea será necesario crear clases formatters para cada uno de los tipos de datos, con sus funciones parse y print, tal y como se ha dado en teoría.
 - Análisis final: Tras varios intentos fallidos de implementación del requisito y estudiar el framework, observé que ya existían formateadores para los tipos money y moments. Además de no lograr poder internacionalizar el booleano. Por esto decidí preguntar por el foro de debate:

https://ev.us.es/webapps/discussionboard/do/message?action=list_message&course_id= 63009_1&nav=discussion_board&conf_id= 303964_1&forum_id= 206215_1&message_id= 365635_1

A lo que el profesor Rafael Corchuelo me respondió entre otras cosas indicándome [la respuesta que le dió a otro compañero](#), en la cual se dice que este requisito se debe marcar como completado ya que para los tipos money y moments ya está implementada la internacionalización por el propio framework y que para los booleanos no se haga ya esto no está soportado por el framework y no era posible implementarlo.

Conclusión

En este tercer Sprint, nos ha resultado útil tener en cuenta los análisis previos para saber afrontar mejor los problemas. Dado que la dificultad de este sprint ha sido bastante mayor al resto, nos ha sido de gran ayuda el análisis de ciertos aspectos antes de ir con la implementación de código.