



ACME-L3

Informe de Análisis

GRUPO C1.02.02

Repositorio: <https://github.com/antcardia/Acme-L3-Do2>

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, 16 de marzo 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

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 empezado a organizarnos en la primera semana del Sprint tras una reunión por vía llamada. Nos hemos asignado las tareas grupales y dejado o recalcado claros los aspectos y guías a la hora de realizar código y subirlo a GitHub. No hemos encontrado ningún impedimento a la hora de repartirnos las tareas y ajustar el proyecto a la entrega.
- Moneda del sistema
 - Análisis previo: En un principio tenemos pensado incluir las monedas y el cambio de las monedas de una manera parecida a cómo está implementado el cambio de idioma en el sistema.
 - Análisis final: Finalmente, para simplificar el requisito, hemos creado una entidad para la configuración del sistema donde incluimos lo que el requisito nos pedía. Además, hemos tenido que poblarlo con los datos correspondientes.
- Entidad peep
 - Análisis previo: Para completar esta tarea, será necesario usar entidades ya creadas del starter acme jobs además de usar las transparencias de clase para hacer un buen uso de las anotaciones necesarias.
 - Análisis final: Durante la creación de la entidad peep no se han tomado decisiones de diseño significativas que requirieron ningún análisis más allá de la lectura del documento de requisitos y la identificación de las anotaciones a usar en cada atributo.
- Diagrama UML grupal
 - Análisis: Podemos observar gracias a las descripciones y tras la implementación que es un diagrama muy compacto y consistente que está lleno de relaciones y enumerados.
 - Análisis: se han corregido pequeñas minucias, como entidades vacías que simplemente no aportan nada
- Entidad bulletin
 - Análisis previo: Para completar la tarea, usaremos como ejemplo entidades ya creadas del starter acme jobs y adaptarlas a nuestras necesidades, aplicando siempre los conocimientos adquiridos en la asignatura.
 - Análisis final: Durante la creación de la entidad bulletin no se han tomado decisiones de diseño significativas que requirieran un análisis más allá de la lectura del documento de requisitos y la identificación de las anotaciones a usar en cada atributo.

- Entidad note
 - Análisis previo: Para crear la entidad, tenemos pensado coger de referencia otras clases y adaptarlas para completar el requisito.
 - Análisis final: A la hora de implementar la entidad, ha sido como esperaba, excepto uno de los atributos, que tiene una validación semántica que no conocía y no sabía cómo incluirla, en concreto el atributo email.
- Crear la entidad offer
 - Análisis: como las demás entidades, tras añadir sus atributos con sus validaciones indicadas, se ha visto que se ha integrado de forma correcta a la base de datos y no ha dado ningún tipo de problema.
 - Análisis final: el commit se rechazó debido a que se necesitaba ultimar ciertos cambios que se solucionaron bastante rápido, todos relacionados con la anotación y el periodo.
- Crear la entidad banner
 - Análisis previo: Se procede a recogida de información de código ya implementado así como las explicaciones de la asignatura para tener una “plantilla” de implementación y ahorrar en tiempo y dificultad.
 - Análisis final: Una vez entendido el concepto de periodo y cómo implementarlo correctamente gracias a la información de los debates de la asignatura. El resto de la implementación fue sencilla.
- Cambiar los precios de double a money
 - Análisis: tras hablar con el product owner, hemos tenido que cambiar todos los precios por el tipo Money implementado en el framework. Las validaciones se harán de forma diferente a las implementadas en otros atributos como por ejemplo @Min(o) que no sirve.
- Interpretación de los periodos:
 - Análisis: los periodos tras hablar con el product owner, hemos decidido que sean dos atributos en vez de uno solo, por consistencia y coherencia del proyecto, donde un periodo serán siempre dos atributos que contengan el día de inicio y el del final de este periodo.
- Dashboard, decisión con la clase Statistic
 - Análisis previo: hemos decidido meter los atributos comunes de los dashboards de todos los students dentro de la clase Statistic, que será implementado en todos los dashboards, y quedará reflejado en el UML.
 - Análisis final: tras el follow up, hemos decidido por la posibilidad de alguna persistencia en un futuro, así que se volverá al modelo original de UML con su correspondiente código.

- **Balanced y la clase Nature**
 - Análisis: tras consultar el foro de debate, nos hemos percatado de que para determinar un desempate es necesario añadir dentro del enumerado el valor balanced, [aquí](#).
- **Uso de Draw.io**
 - Análisis: en nuestro proyecto hemos decidido, por comodidad y el uso de esta en otros proyectos, usar esta aplicación online que se puede integrar en Google drive y sus servicios y como ya describimos, estamos gestionando documentación mediante Google drive. No hemos usado UMLet, debido al desconocimiento de este, tras el visto bueno del product owner, se nos ha permitido usar esta aplicación siempre y cuando esté incluido en el proyecto en GitHub. A la hora de entregar los modelos, haremos capturas de estos pasados al formato PDF.
- **Cambio de la palabra abstract:**
 - Análisis : como la palabra abstract en java es una palabra reservada, es decir, una palabra que tiene implicaciones en el código, y no se puede usar como cadena suelta. Hemos decidido usar otra nomenclatura como summary.
 - Análisis final: tras consultarlo en el follow up con el product owner, hemos decidido entre el equipo y el llamar a los atributos abstract como abstract\$, para respetar el nombre original.

Conclusión

En este segundo Sprint, al realizar un análisis previo y cuando ha sido necesario un análisis posterior en tiempo real, ha resultado muy cómodo, ya que se ha ido rellenando el documento a medida que se realizaban las tareas y las decisiones de diseño, pudiendo reflejar las situaciones lo más real posible.