Informe de análisis sobre tarea XX

Integrantes del grupo C1.026:

* Ignacio Blanquero Blanco ([ignblabla@alum.us.es](mailto:ignblabla@alum.us.es))
* Adrián Cabello Martín ([adrcabmar@alum.us.es](mailto:adrcabmar@alum.us.es))
* María de la Salud Carrera Talaverón ([marcartal1@alum.us.es](mailto:marcartal1@alum.us.es))
* Joaquín González Ganfornina ([joagongan@alum.us.es](mailto:joagongan@alum.us.es))
* Natalia Olmo Villegas ([natolmvil@alum.us.es](mailto:natolmvil@alum.us.es))

Fecha: Sevilla, 8 de Marzo 2024

Tabla de contenido

[Resumen del Informe 3](#_Toc160791645)

[Historial de Versiones 4](#_Toc160791646)

[Introducción 5](#_Toc160791647)

[Informe de Análisis – Acme-SF-D02 6](#_Toc160791648)

[Registro de Análisis 1 6](#_Toc160791649)

[Requisito 6](#_Toc160791650)

[Análisis y decisiones 6](#_Toc160791651)

[Validación por el profesor 6](#_Toc160791652)

[Registro de Análisis 2 7](#_Toc160791653)

[Requisito 7](#_Toc160791654)

[Análisis y decisiones 7](#_Toc160791655)

[Validación por el profesor 7](#_Toc160791656)

[Registro de Análisis 3 7](#_Toc160791657)

[Requisito 7](#_Toc160791658)

[Análisis y decisiones 8](#_Toc160791659)

[Validación por el profesor 8](#_Toc160791660)

[Registro de Análisis 4 8](#_Toc160791661)

[Requisito 8](#_Toc160791662)

[Análisis y decisiones 8](#_Toc160791663)

[Validación por el profesor 9](#_Toc160791664)

[Conclusiones 10](#_Toc160791665)

[Bibliografía 11](#_Toc160791666)

# Resumen del Informe

El objetivo de este informe es proporcionar detalles sobre el análisis de todos los tipos de requisitos especificados en esta segunda entrega del proyecto, así como las decisiones tomadas para llevarlos a cabo y enlazarlo, si procediera, a las validaciones y observaciones planteadas por el profesor encargado de la asignatura.

# Historial de Versiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Contenidos** | **Fecha** | **Contribuyente** |
| V0.1 | Actualizar Analysis Report D02 | 06/03/2024 | Natalia Olmo Villegas |
| V0.2 | Actualizar Analysis Report D02 | 08/03/2024 | Natalia Olmo Villegas  María de la Salud Carrera Talaverón |

# Introducción

En esta segunda entrega teníamos que implementar varios requisitos de información y un requisito de testing, a diferencia de la primera entrega, en la que sólo se tenía que implementar un requisito funcional. Por lo tanto, la carga de trabajo ha aumentado y resulta de vital importancia una buena gestión de las tareas a cumplimentar por parte del Project Manager, Joaquín González Ganfornina en este caso. A continuación, se describirán con más detalle todos los requisitos implementados por todos los integrantes del grupo.

# Informe de Análisis – Acme-SF-D02

Cabe destacar que a la hora de declarar los requisitos hemos decidido que aquellos atributos en los que en su requisito correspondiente no se especificaba si era obligatorio se les asignara @NotNull para obligarlos a ser obligatorios.

## Registro de Análisis 1

### Requisito

Una noticia es un mensaje publicado por un usuario autenticado. El Sistema debe guardar los siguientes datos sobre ella: un momento de instanciación (en el pasado), un título (no puede estar en blanco, con menos de 76 caracteres), un autor (no puede estar en blanco, con menos de 76 caracteres), un mensaje (no puede estar en blanco, con menos de 101 caracteres), un correo electrónico opcional y un link opcional. El autor debe ser completado de la siguiente forma: “〈nombre de usuario〉 - 〈apellido, nombre〉”, donde “〈nombre de usuario〉” denota el nombre de usuario de aquella persona que ha publicado la noticia y “〈apellido, nombre〉” denota su nombre completo.

### Análisis y decisiones

Dado que este requisito no presentaba mucha complejidad, no me ha surgido ningún problema, sólo se debían declarar los atributos siguiendo las instrucciones definidas en el requisito, las cuales eran bastante claras. Se ha creado un paquete “notices” y dentro de este se ha creado la entidad “Notice.java”.

Respecto al atributo “author”, en un primer momento pensamos que podría ser un datatype ya que en el requisito nos indica que debe seguir una estructura concreta “〈nombre de usuario〉 - 〈apellido, nombre〉”, sin embargo, hemos optado por declarar a esta entidad para que sea tipo String y ya implementaremos esta restricción próximamente.

Además, “Notice.java” es una entidad aislada, por lo que no está relacionada con nada. En el requisito nos piden que la noticia sea publicada por un usuario autenticado, esta funcionalidad la implementaremos en la siguiente entrega, ya que ahora mismo no disponemos de los conocimientos necesarias para desarrollarla.

### Validación por el profesor

Tras la segunda sesión de revisión en la clase de laboratorio, el profesor ha comentado que no era necesario relacionar a esta entidad con nada, por ser una entidad aislada. Además, ha indicado que no era necesario la creación de ningún datatype para lo que se nos requería.

## Registro de Análisis 2

### Requisito

El Sistema de configuración debe incluir los siguientes datos iniciales:

* Un sistema monetario, que debe ser inicializado a “EUR”
* Una lista de las monedas aceptadas, las cuales deben ser inicializadas a “EUR”, “USD” y “GBP”

### Análisis y decisiones

Este requisito ha resultado un gran reto para María de la Salud Carrera Talaverón y Natalia Olmo Villegas, pues no encontrábamos demasiada información acerca de qué podíamos hacer para satisfacer dicho requisito.

En un primer momento, pensamos que sería buena idea crear un datatype basado en el datatype “Money.java” que se nos proporcionaba en el framework. No obstante, el profesor indicó que en esta entrega no era necesario la implementación de datatypes nuevos.

Entonces, pensamos dos alternativas, modificar “application.properties” o crear una entidad nueva. Finalmente, llegamos a la conclusión de que sería mejor crear una entidad nueva, debido a que se sabe que modificar este archivo “application.properties” podría generar problemas.

La entidad creada se llama “SystemConfiguration.java” y se encuentra dentro del paquete “systemConfigurations”. Sus atributos serán la moneda por defecto y la lista de monedas que pueden utilizarse. Para cumplir con el requisito por completo, creamos y nuevo archivo denominado “system-configuration.csv” poblándolo con los datos necesarios para satisfacer las restricciones.

### Validación por el profesor

En la última sesión de laboratorio antes de esta segunda entrega, el profesor indicó que no sería necesario un datatype nuevo. Sin embargo, tampoco nos proporcionó mucha más información acerca de cómo debíamos desarrollar esta tarea, por lo que decimos crear una nueva entidad para ello por las razones comentadas anteriormente.

## Registro de Análisis 3

### Requisito

Un banner permite a los administradores publicitas productos, servicios u organizaciones. El sistema debe almacenar la siguiente información sobre ellos: el momento de creación o actualización de la instancia (en el pasado), el período en el que se muestra (debe comenzar en cualquier momento tras la creación o actualización y durar al menos una semana), un enlace a una imagen almacenada externamente, un eslogan (no en blanco, 75 caracteres o menos) y un enlace a un documento web.

### Análisis y decisiones

Este requisito, implementado por María de la Salud Carrera Talaverón, no ha supuesto grandes complicaciones más allá del período de muestra del banner. Se ha creado el paquete “acme.banners” y dentro se ha añadido la entidad Banner.java.

Todos los atributos pedidos se han implementado con las restricciones solicitadas. En el caso del período, se ha decidido implementar mediante dos atributos no nulos: startDate y finishDate. Debido a que las restricciones para estos atributos eran personalizadas y no se pueden indicar mediante las anotaciones habituales disponibles en Java, se ha dejado su implementación para el siguiente entregable, donde se hará en la clase servicio adecuada. Se pensó añadir un atributo derivado llamado displayPeriod, que calculase el período de tiempo transcurrido entre startDate y finishDate, pero finalmente se descartó como sugerencia por parte del profesor.

Del mismo modo, inicialmente se relacionó la entidad con el rol Administrator.java, pero esto también fue descartado como parte de la validación del profesor, ya que todas las entidades a implementar en la parte grupal debían estar aisladas.

### Validación por el profesor

Habiendo incluido los cambios ya mencionados, el profesor validó esta implementación el día 6 de marzo.

## Registro de Análisis 4

### Requisito

Producir un diagrama UML.

### Análisis y decisiones

Esta tarea fue asignada a María de la Salud Carrera Talaverón y Natalia Olmo Villegas. Su desarrollo fue relativamente sencillo, puesto que, como ya se ha mencionado, las entidades a representar no estaban relacionadas entre ellas ni con ninguna otra. Se ha utilizado el programa UMLet como se indica en los requisitos de la asignatura. El resultado se puede encontrar en la carpeta “src/reports/Group” bajo el nombre de UML-Group.uxf.

### Validación por el profesor

No hizo falta que el profesor validase esta tarea debido a su sencillez. El único feedback aportado se deduce de las sugerencias hechas en la implementación de cada una de las clases, ya que este diagrama es un reflejo de estas.

# Conclusiones

# Bibliografía

En blanco a propósito.