

# **REPORTE DE ANÁLISIS**

**C1-03-03**

---

**DP II**

---

**Grupo:** C1.03.03

**Repositorio:** <https://github.com/rauhernaan/C1.03.03>

**Estudiante #1**

**DNI:** 32099110B

**UVUS:** pabpinmat

**Nombre:** Pino Mateo, Pablo

**Rol:** Developer

**Estudiante #2**

**DNI:** 47349483G

**UVUS:** ferbaqfer

**Nombre:** Baquero Fernández, Fernando

**Rol:** Developer

**Estudiante #3**

**DNI:** 77939845K

**UVUS:** dancorfon

**Nombre:** Cortés Fonseca, Daniel

**Rol:** Analista

**Estudiante #4**

**DNI:** 17475514E

**UVUS:** raumerbas

**Nombre:** Mérida Bascón, Raúl Hernán

**Rol:** Manager

**Estudiante #5**

**DNI:** 77930171F

**UVUS:** gonriblun

**Nombre:** Ribas Luna, Gonzalo

**Rol:** Developer

**Fecha:** 17/03/2023

# Índice

1.-Executive summary .....	3
2.-Tabla de revisiones.....	3
3.-Introducción .....	3
4.-Contenido.....	4
4.1.- Listado de requisitos .....	4
4.2.- Conclusión y decisiones sobre los requisitos .....	6
5.-Conclusiones .....	8
6.-Bibliografía .....	8

## 1.-Executive summary

Durante la realización del análisis de esta entrega no hemos tenido prácticamente ningún problema o complicación, al ser bastante simple y haber hecho antes los requisitos individuales.

Solo hemos tenido alguna complicación al realizar los diagramas UML individuales y grupales.

## 2.-Tabla de revisiones

Fecha	Versión	Descripción
13/03/2023	1.0	Creación del documento y listado de los requisitos
14/03/2023	1.1	Elaboración de las decisiones
17/03/2023	1.2	Finalización del documento, realizando el executive summary y la conclusión final

## 3.-Introducción

En este documento realizaremos un listado de los distintos requisitos que se nos piden para este entregable a nivel grupal únicamente, así mismo se proporciona un análisis individual para cada requisito o tarea a partir del cual obtenemos una conclusión y una o un conjunto de decisiones a tomar para la realización de dicho requisito.

Este análisis lo dividiremos en dos boques, uno en el que mostraremos un listado de los requisitos proporcionados por el “cliente”, traducido al español, y otro bloque en el que daremos la conclusión a la que se ha llegado, de forma grupal, acerca del requisito que se analiza y las decisiones a tomar para el mismo.

## 4.-Contenido

### 4.1.- Listado de requisitos

En este entregable encontramos 11 requisitos, que no han supuesto grandes ni complejos problemas:

1. La configuración del sistema debe incluir los siguientes datos iniciales: Una moneda del sistema, que debe inicializarse en "EUR". Una lista de monedas aceptadas, que debe inicializarse en "EUR", "USD" y "GBP".
2. Un "peep" es un mensaje publicado por cualquier usuario. El sistema debe almacenar los siguientes datos sobre ellos: el momento de la publicación (en el pasado), un título (no vacío, más corto que 76 caracteres), un apodo (no vacío, más corto que 76 caracteres), un mensaje (no vacío, más corto que 101 caracteres), una dirección de correo electrónico opcional y un enlace opcional.
3. Un "boletín" es un mensaje publicado por un administrador. El sistema debe almacenar los siguientes datos sobre ellos: el momento de la publicación (en el pasado), un título (no vacío, más corto que 76 caracteres), un mensaje (no vacío, más corto que 101 caracteres), una marca para indicar si es crítico o no y un enlace opcional con más información.
4. Una "oferta" es un registro en el que un administrador anuncia algo. El sistema debe almacenar los siguientes datos sobre ellos: el momento de la publicación (en el pasado), un encabezado (no vacío, más corto que 76 caracteres), un resumen (no vacío, más corto que 101 caracteres), un período de disponibilidad (al menos un día después de la publicación de la oferta y que debe durar al menos una semana), un precio (positivo, posiblemente cero) y un enlace opcional con más información.
5. Una "nota" es un mensaje publicado por un usuario autenticado. El sistema debe almacenar los siguientes datos sobre ellos: el momento de la publicación, un título (no vacío, más corto que 76 caracteres), un autor (no vacío, más corto que 76 caracteres), un mensaje (no vacío, más corto que 101 caracteres), una dirección de correo electrónico opcional y un enlace opcional. El autor debe ser calculado de la siguiente manera: "<nombre de usuario> - <apellido, nombre>",

donde "<nombre de usuario>" denota el nombre de usuario del usuario que publicó la nota y "<apellido, nombre>" denota su nombre completo.

6. Un "banner" permite a los administradores anunciar productos, servicios u organizaciones. El sistema debe almacenar los siguientes datos sobre ellos: el momento de la creación/actualización (en el pasado), un período de visualización (que debe comenzar en cualquier momento después del momento de creación/actualización y debe durar al menos una semana), un enlace a una imagen que debe almacenarse en otro lugar, un eslogan (no vacío, más corto que 76 caracteres) y un enlace a un documento web objetivo.
7. El sistema debe manejar paneles de control de administrador con los siguientes indicadores: número total de usuarios con cada rol; relación de "peeps" con tanto una dirección de correo electrónico como un enlace; relaciones de boletines críticos y no críticos; promedio, mínimo, máximo y desviación estándar del presupuesto en las ofertas agrupadas por moneda; promedio, mínimo, máximo y desviación estándar del número de notas publicadas en las últimas 10 semanas.
8. Produce datos iniciales para arrancar en frío su proyecto; estos deben incluir una cuenta de administrador con credenciales "administrator/administrator". Produzca datos de ejemplo variados para probar sus requisitos; estos deben incluir dos cuentas de administrador con credenciales "administrator1/administrator1" y "administrator2/administrator2".
9. Elabora un reporte de análisis.
10. Elabora un reporte de planificación.
11. Produce un modelo de dominio UML.

#### 4.2.- Conclusión y decisiones sobre los requisitos

En este entregable no hemos tenido grandes problemas a la hora de analizar los requisitos:

1. En este primer requisito hemos concluido, tras analizarlo que teníamos que crear una clase System Configuration y añadir las monedas en un systemConfiguration.csv.

No hemos tenido complicaciones con este requisito.

2. En este requisito concluimos que teníamos que crear una clase Peep, con los atributos especificados en el requisito.

No hemos tenido complicaciones con este requisito.

3. En este requisito concluimos que teníamos que crear una clase Bulletin, con los atributos especificados en el requisito.

No hemos tenido complicaciones con este requisito.

4. En este requisito concluimos que teníamos que crear una clase Offer, con los atributos especificados en el requisito.

No hemos tenido complicaciones con este requisito.

5. En este requisito concluimos que teníamos que crear una clase Note, con los atributos especificados en el requisito.

No hemos tenido complicaciones con este requisito.

6. En este requisito concluimos que teníamos que crear una clase Banner, con los atributos especificados en el requisito.

No hemos tenido complicaciones con este requisito.

7. Para este requisito llegamos a la conclusión de hacer la clase AdministratorDashboard con los datos que me especifican el requisito.

No hemos tenido complicaciones con este requisito.



8. Para este requisito llegamos a la conclusión de que teníamos que crear los csv con datos de prueba de todas las clases creadas anteriormente.

No hemos tenido complicaciones con este requisito.

9. Para este requisito no ha habido ninguna complicación.

10. Para este requisito no ha habido ninguna complicación.

11. Hemos llegado a la conclusión de que teníamos que realizar dos diagramas UML, uno grupal con las clases de los requisitos grupales y otro con las clases de los requisitos individuales.

Para este requisito hemos tenido una complicación con el UML individual, ya que las clases de algunos integrantes tenían relación con una clase de un integrante en concreto, por lo que llegamos a la conclusión de que este integrante tenía que hacer su diagrama UML primero, y después lo completamos los demás.

Para el UML grupal hemos tenido otra complicación, no sabíamos si incluir las restricciones en el diagrama, y al final hemos decidido incluirlas porque al no haber relaciones entre las clases y ser bastante simple, poner las restricciones no enrevesaba mucho el diagrama.

## 5.-Conclusiones

Como conclusión acerca de este informe podríamos decir que nos ha servido de ayuda para la realización de algunas tareas en concreto ya que el trabajo de estas se facilita a la hora de analizarlas previamente y al tomar ciertas decisiones.

Igualmente, en esta entrega quizás no ha sido todo lo útil que nos esperábamos ya que los requisitos no tenían mucha dificultad, al ser sobre todo creación de clases sin relaciones entre ellas y al haber primero elaborado los requisitos individuales, que eran muy parecidos.

## 6.-Bibliografía

Intentionally blank