



# Analysis report

**C3.03.14**

<https://github.com/c1-03-14/c1.03.14>

Victor Graván Biru  
vicgrabru@alum.us.es

Gonzalo Santiago Martín  
gonsanmar2@alum.us.es

Fernando Barroso Barroso  
ferbarbar1@alum.us.es

Alvaro Sánchez Flores  
alvsanf1o1@alum.us.es

# Tabla de contenidos

Tabla de contenidos	1
Resumen ejecutivo	2
Tabla de versiones	2
Introducción	3
Contenido	4
Conclusión	5
Bibliografía	5

# Resumen ejecutivo

El análisis de requisitos es un paso fundamental en el proceso de desarrollo de un sistema, ya que es el momento en que se identifican y documentan las necesidades y expectativas de los usuarios y las partes interesadas.

Un análisis de requisitos completo y preciso puede ayudar a evitar problemas y cambios costosos más adelante en el proceso de desarrollo del sistema. Es importante involucrar a los usuarios y partes interesadas en el proceso de análisis de requisitos para comprender sus necesidades y expectativas, y documentar estos requisitos de manera clara y concisa.

De esta manera, el equipo de desarrollo puede trabajar para diseñar e implementar un sistema que cumpla con estos requisitos y expectativas. En general, un análisis de requisitos efectivo es esencial para el éxito de un proyecto de desarrollo de sistemas.

## Tabla de versiones

Versión	Fecha	Descripción
1.0	28-09-2023	Primera versión del documento
1.1	30-09-2023	Modificaciones
1.2	20-10-2023	Versión final tras tutoría

# Introducción

La fase de análisis de requisitos es un paso fundamental en el proceso de desarrollo de un sistema, ya que es el momento en que se identifican y documentan las necesidades y expectativas de los usuarios y las partes interesadas. El análisis de requisitos se enfoca en comprender los objetivos del sistema, los procesos que se deben llevar a cabo, los datos que se deben capturar y los resultados que se esperan del sistema.

Para llevar a cabo un análisis de requisitos efectivo, es necesario involucrar a los usuarios y partes interesadas para entender sus necesidades y expectativas, y luego documentar estos requisitos de manera clara y concisa. De esta manera, el equipo de desarrollo puede trabajar para diseñar e implementar un sistema que cumpla con estos requisitos y expectativas.

Un análisis de requisitos completo y preciso puede ayudar a evitar problemas y cambios costosos más adelante en el proceso de desarrollo del sistema. Por lo tanto, es importante dedicar suficiente tiempo y recursos para el análisis de requisitos para garantizar que el sistema cumpla con las necesidades y expectativas de los usuarios y partes interesadas, y tenga éxito en su implementación.

# Contenido

## Requisito 6

Una clase es un documento que un profesor usa para dar conocimiento. El sistema debe almacenar los siguientes datos: título (no en blanco, menos de 76 caracteres), un resumen (no en blanco, menos de 101 caracteres), un tiempo estimado de aprendizaje (en horas, positivo), un cuerpo (no en blanco, menos de 101 caracteres), una indicación de si puede ser considerada teórica o práctica y un link opcional con más información.

En cuanto a este requisito, había cierta ambigüedad con el atributo de teórico o práctico, así que durante una sesión de seguimiento discutimos con la profesora cuál sería la mejor manera de implementar este atributo en la entidad.

Propusimos 2 alternativas:

### 1ª Alternativa:

Crear el atributo `isTheoretical` de manera booleana, siendo `true` si la clase es de tipo teórica, y `false` de tipo práctica.

#### PROS

Muy fácil de implementar y de manejar

#### CONTRAS

Puede dar lugar a ambigüedades en el cliente

Nos causará problemas a la hora de implementar este atributo en la entidad `Course`, puesto que es un atributo derivado del tipo de las clases.

### 2ª Alternativa:

Crear un enumerado de tipo `LectureType` para usarse tanto en `Course` como en `Lecture`, implementando el atributo `type` en ambas entidades como un tipo `LectureType`

#### PROS

Mayor claridad a la hora de crear clases

Facilidad a la hora de calcular el atributo derivado y mejor visualización en las vistas de la aplicación

#### CONTRAS

Algo más complejo de implementar que el tipo boolean

## CONCLUSIÓN

Finalmente nos decidimos por implementar el enumerado puesto que nos iba a resultar mucho más cómodo para trabajar tanto en el atributo derivado de la entidad `Course` como a la hora de visualizar las clases y los cursos.

En este requisito también tomamos la decisión de iniciar los cursos a tipo `theoretical` por defecto al crearlos y, además, al tener el mismo número de clases prácticas y teóricas el curso sería de tipo práctico siendo posible haberlos hecho que se fueran de tipo teórico o balanceado, optando por ponerlos de tipo práctico, ya que no nos influye en gran medida.

# Conclusión

En conclusión, todos los requisitos de este entregable estaban bastante bien definidos y no mostraban demasiada ambigüedad o completitud, excepto en el tipo de Lecture, aunque, gracias a la ayuda de la profesora pudimos resolver este problema con rapidez y sin grandes contratiempos. En el resto de requisitos no hemos encontrado ningún problema.

# Bibliografía

Intencionalmente en blanco.