

Trabajo Práctico Integrador Final

Proyecto de Desarrollo de Software

Objetivo

Integrar los conocimientos adquiridos a lo largo de ambas materias, demostrando la aplicación de los principios, métodos y herramientas de Ingeniería de Software en el proyecto propuesto por cada grupo.

Cada grupo deberá presentar y documentar el proceso completo de desarrollo de su sistema, desde la definición de requerimientos hasta la validación final mediante pruebas funcionales.

3.A- Contexto del Proyecto

El proyecto surge ante la necesidad de mejorar la enseñanza de biología, una materia donde muchos conceptos y procesos resultan difíciles de comprender con métodos tradicionales. El sistema propuesto, denominado *Libro Virtual Interactivo de Biología*, busca abordar esta problemática ofreciendo una plataforma web educativa que integra textos, simulaciones, modelos 3D y cuestionarios interactivos, permitiendo un aprendizaje más dinámico y visual.

Los principales usuarios del sistema son los **estudiantes**, que acceden al contenido y realizan actividades; los **docentes**, que supervisan el progreso y evalúan el rendimiento; y los **administradores**, encargados de gestionar los materiales y mantener actualizada la plataforma. El objetivo general del software es brindar una herramienta educativa accesible y atractiva que favorezca la comprensión y el seguimiento del aprendizaje en biología.

3.B- Análisis y Diseño

Requerimientos Funcionales:

- Registrarse/Iniciar sesión
- Visualizar contenido
- Descargar/visualizar reportes de los alumnos
- Gestionar roles y permisos
- Crear/Editar contenido
- Crear/Editar cuestionarios
- Realizar cuestionarios
- Realizar transacciones
- Procesar transacciones

Requerimientos No Funcionales:

- Un docente intenta acceder a los reportes de sus alumnos un domingo a la noche.
- Un alumno intenta acceder a la plataforma durante el mantenimiento programado.

- El sistema debe estar disponible para consultas de contenido las 24 horas, incluso en días feriados.
- Un usuario desea adquirir el libro premium
- Un docente exporta resultados de exámenes en formato HTML para compartir con colegas.
- Correcta visualización de contenido multimedia
- Un alumno accede a la plataforma desde una tablet y navega por los contenidos y cuestionarios.
- Un usuario nuevo accede por primera vez y debe completar el registro sin ayuda externa.
- Un alumno con discapacidad visual utiliza un lector de pantalla para navegar por la plataforma.

Casos de uso:

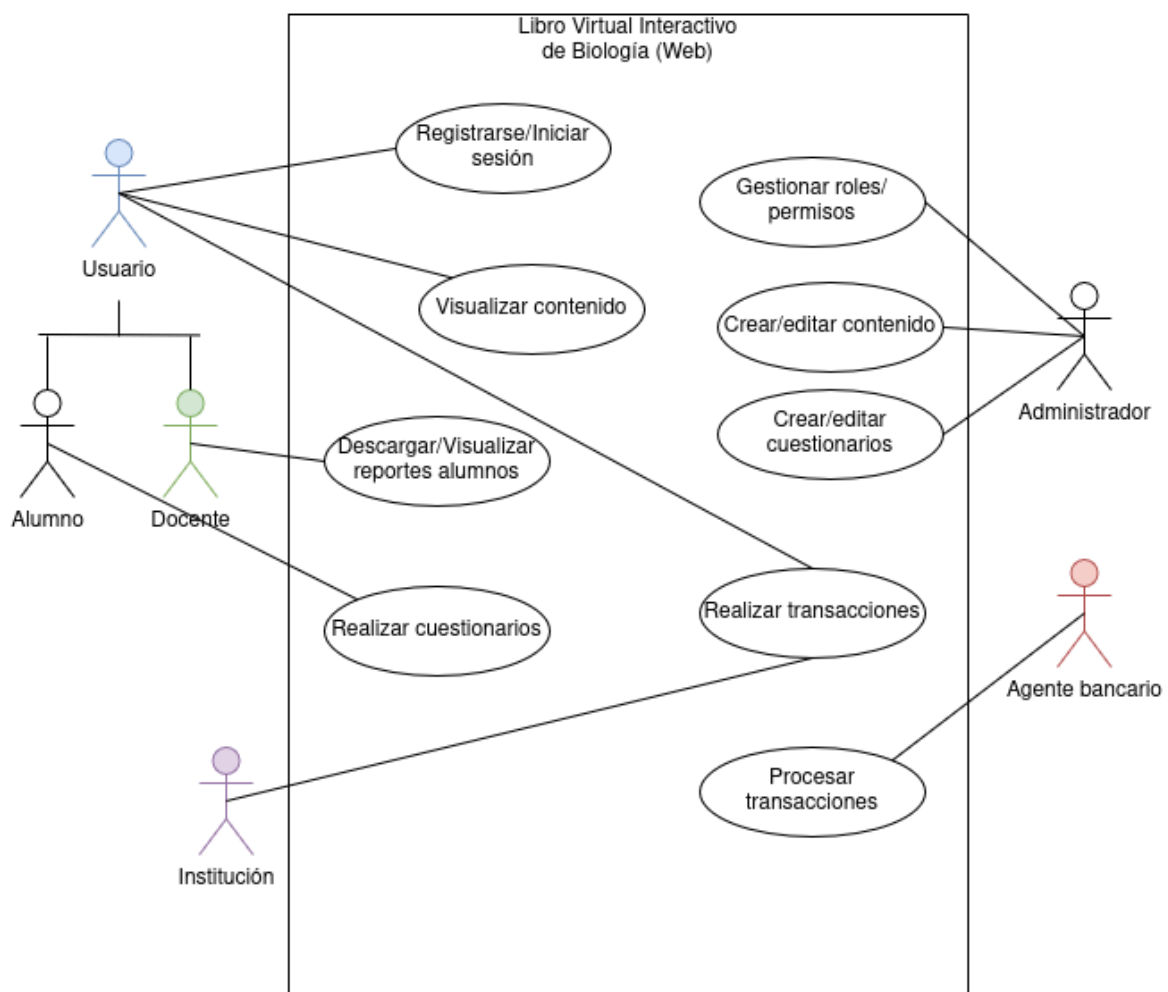


Imagen 1 (Diagrama de casos de uso)

Dos casos de uso claves son:

Visualizar contenido: El usuario accede al contenido del libro virtual (texto, imágenes, modelos 3D, simulaciones) a través de la plataforma web.

Crear/Editar Contenido: El administrador accede al panel de gestión para crear, editar o eliminar contenido del libro virtual, incluyendo textos, imágenes, modelos 3D y simulaciones.

Atributos de Calidad:

Los atributos calidad que consideramos críticos para nuestro proyecto luego de la votación de escenarios de calidad, fueron:

1. Disponibilidad: Este atributo se refiere a la capacidad del sistema para estar operativo y accesible para los usuarios cuando lo necesitan. En el caso del *Libro Virtual Interactivo de Biología*, la disponibilidad es crítica porque estudiantes y docentes deben poder acceder al contenido educativo en cualquier momento, especialmente durante clases o instancias de evaluación.
2. Interoperabilidad: La interoperabilidad indica la facilidad con la que el sistema puede comunicarse o integrarse con otros servicios o plataformas. Se consideró prioritaria porque el sistema depende de diferentes componentes externos, como el almacenamiento de archivos en la nube (Amazon S3), bases de datos remotas (RDS MySQL) y una pasarela de pagos. Asegurar la interoperabilidad permite que todos estos servicios funcionen en conjunto sin problemas, manteniendo la coherencia de los datos y la experiencia del usuario.
3. Usabilidad: La usabilidad se relaciona con la facilidad de uso y comprensión del sistema por parte de los usuarios. En este proyecto, es uno de los aspectos más importantes, ya que la plataforma está dirigida a estudiantes y docentes que pueden no tener conocimientos técnicos avanzados. Se priorizó porque un entorno intuitivo, visual y simple favorece la adopción del sistema, reduce errores de uso y mejora la experiencia de aprendizaje.

3.C- Arquitectura y Despliegue - Diagrama C4

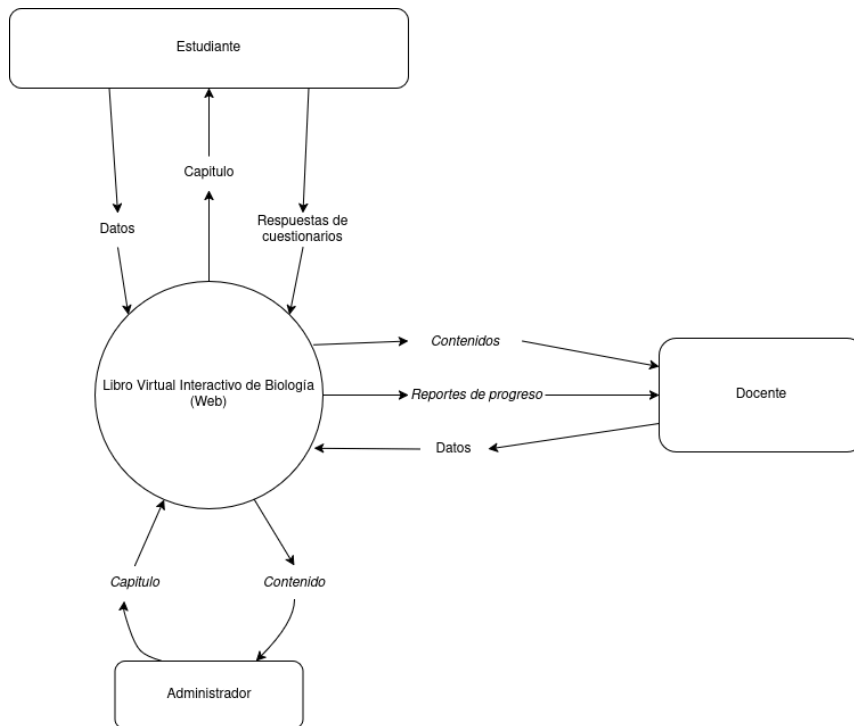


Imagen 2 (Diagrama de contexto Nivel 0)

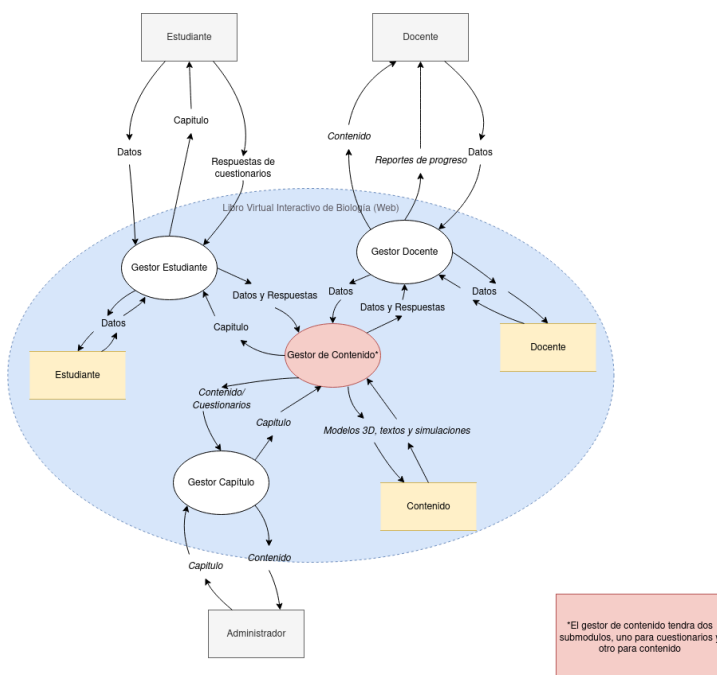


Imagen 3 (Diagrama de contexto Nivel 1)

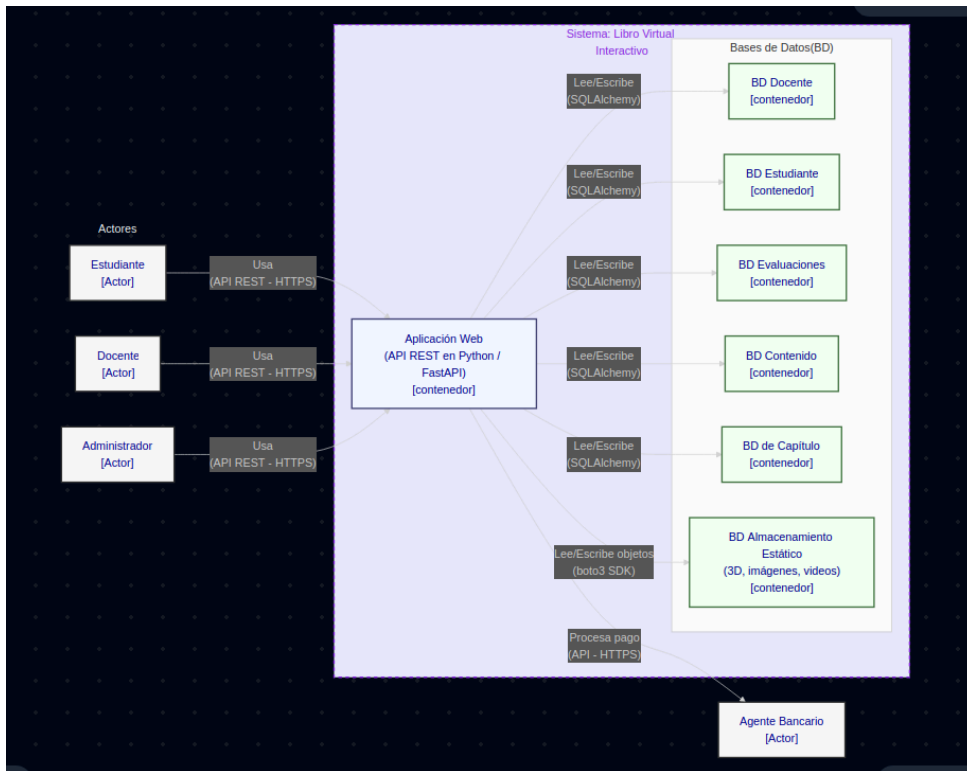


Imagen 4 (Diagrama de contenedores)

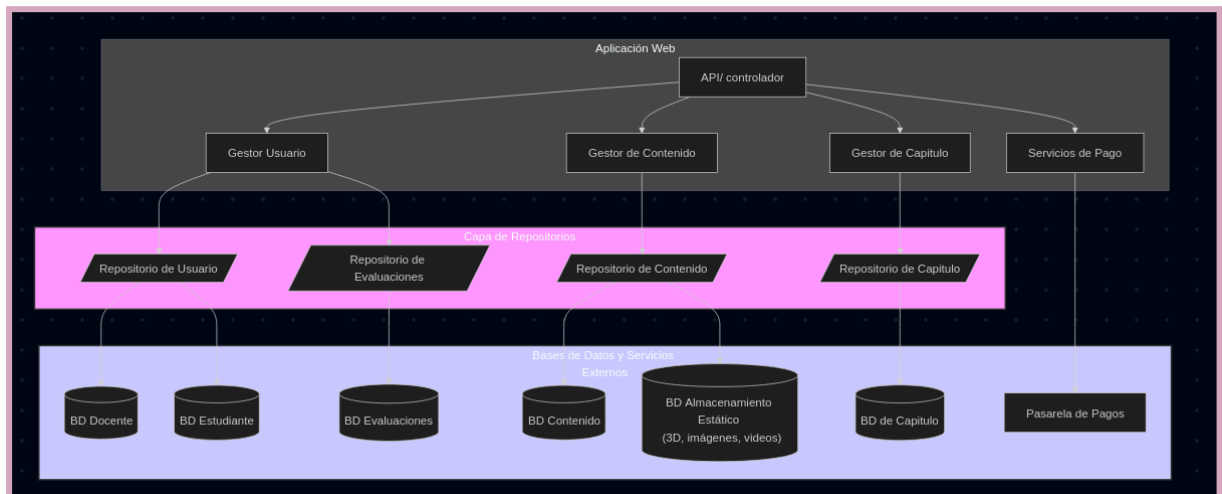


Imagen 5 (Diagrama de componentes)

```

classDiagram
    class GestorContenido {
        +RepositorioContenido repositorio
        +crear_bloque_texto(cuerpo: texto: String, tema: String) : Texto
        +crear_bloque_imagen(url_archivo: String, formato: String, tema: String) : Imagen
        +crear_bloque_video(url_archivo: String, duracion: Float, tema: String) : Video
        +crear_bloque_objeto3D(url_archivo: String, formato: String, tema: String) : Objeto3D
        +actualizar_bloque_texto(id_contenido: String, texto: String, tema: String) : Texto
        +eliminar_contenido(id_contenido: String) : void
        +eliminar_contenido_por_id(id_contenido: String) : Contenido
        +buscar_contenidos_por_tema(tema: String) : List<Contenido>
    }

    class RepositorioContenido {
        +db_session: Session %No Atributo: La conexión a la BD
        +guardar(contenido: Contenido) : Contenido
        +eliminar_por_id(id_contenido: String) : Contenido
        +buscar_por_tema(tema: String) : List<Contenido>
        +eliminar(id_contenido: String) : void
    }

    class GestorCapitulo {
        +repositorio_capitulo: RepositorioCapitulo
        +repositorio_contenido: RepositorioContenido
        +repositorio_union: RepositorioUnionCapituloContenido
        +crear_capitulo(capitulo: String, numero: int, introduccion: String, tema: String) : Capitulo
        +eliminar_capitulo(id_capitulo: String) : void
        +actualizar_introduccion_capitulo(id_capitulo: String, datos: String, numero: int) : Capitulo
        +obtener_capitulo_completo(id_capitulo: String) : Capitulo, List<Contenido>
        +guardar_orden_contenido(id_capitulo: String, lista_id_contenido: List<String>) : void
    }

    class RepositorioCapitulo {
        +db_session: Session %No Atributo: La conexión a la BD
        +guardar(capitulo: Capitulo) : Capitulo
        +buscar_por_id(id_capitulo: String) : Capitulo
        +buscar_por_tema(tema: String) : List<Capitulo>
        +eliminar_introduccion(id_capitulo: String) : List<Capitulo>
        +eliminar(id_capitulo: String) : void
    }

    class Capitulo {
        +id_capitulo: String
        +titulo: String
        +numero: Integer
        +introduccion: String
        +tema: String
        +Método() String
    }

    class RepositorioUnionCapituloContenido {
        +db_session: Session %No Atributo: La conexión a la BD
        +crear_union(id_capitulo: String, id_contenido: String, orden: int) : UnionCapituloContenido
        +borrar_por_capitulo(id_capitulo: String) : void
        +borrar_por_contenido(id_contenido: String) : void
        +obtener_union(es_por_capitulo_ordenadas(id_capitulo: String) : List<UnionCapituloContenido>
    }

    class UnionCapituloContenido {
        +orden: Integer
    }

    class AbstractoContenido {
        +id_contenido: String
        +tema: String
        +Método() String
    }

    class Texto {
        +cuerpo_texto: String
    }

    class Imagen {
        +url_archivo: String
    }

    class Video {
        +url_archivo: String
    }

    class Objeto3D {
        +url_archivo: String
    }

    GestorContenido --> RepositorioContenido
    GestorContenido --> AbstractoContenido
    GestorContenido --> Texto
    GestorContenido --> Imagen
    GestorContenido --> Video
    GestorContenido --> Objeto3D
    RepositorioContenido --> AbstractoContenido
    RepositorioContenido --> Texto
    RepositorioContenido --> Imagen
    RepositorioContenido --> Video
    RepositorioContenido --> Objeto3D
    GestorCapitulo --> RepositorioCapitulo
    GestorCapitulo --> RepositorioContenido
    GestorCapitulo --> RepositorioUnionCapituloContenido
    GestorCapitulo --> Capitulo
    RepositorioCapitulo --> Capitulo
    RepositorioUnionCapituloContenido --> UnionCapituloContenido
    UnionCapituloContenido --> Capitulo
    UnionCapituloContenido --> AbstractoContenido
    
```

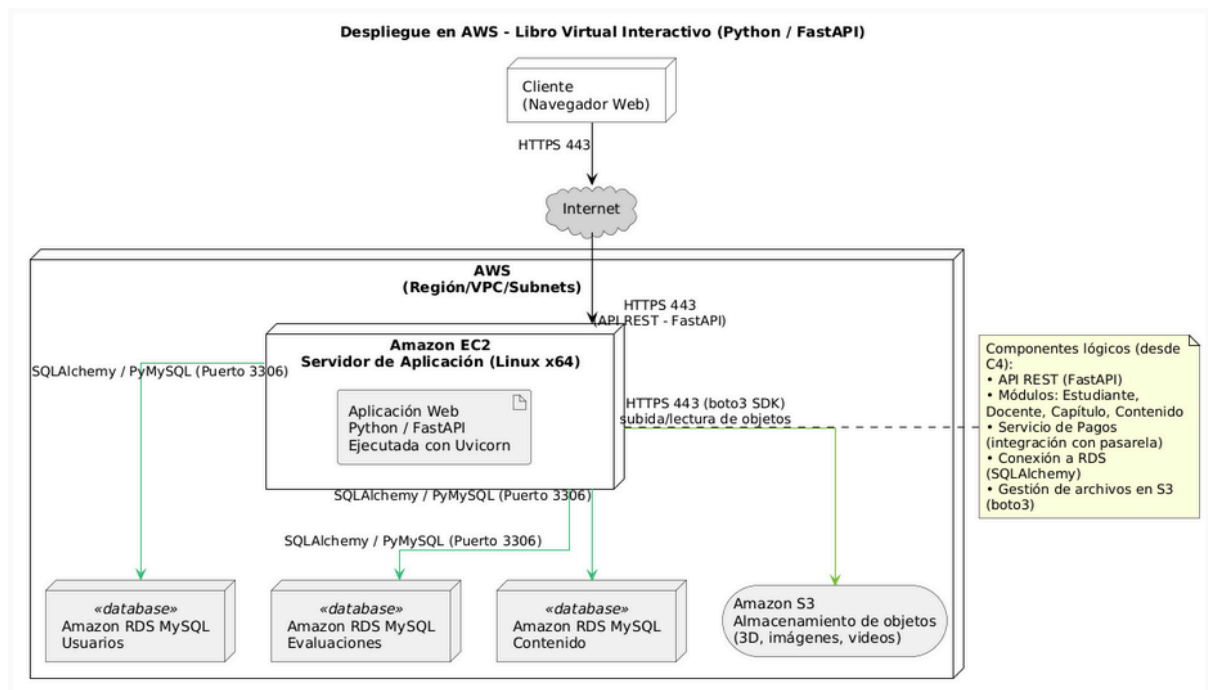


Imagen 7 (Diagrama de despliegue)

Navegadores permitidos

Como el sistema es una **aplicación web moderna** que usa **FastAPI** (que entrega contenido en formato HTML y JSON), es compatible con todos los navegadores actuales que soportan estándares web recientes.

Los navegadores recomendados serían:

- **Google Chrome** (versión 100 o superior)
- **Mozilla Firefox** (versión 100 o superior)
- **Microsoft Edge** (últimas versiones basadas en Chromium)
- **Safari** (para dispositivos Apple, versión 15 o superior)

3.D- Testing

CP01_01 — Visualizar capítulo existente

- **Id:** CP01_01
- **Caso de Uso Relacionado:** CU_01 Visualizar contenido
- **Caso de Prueba:** Visualizar capítulo existente
- **Descripción:** El usuario selecciona un capítulo que ya fue creado y almacenado en la base de datos.
- **Fecha:** 3/11/2025
- **Área Funcional / Subproceso:** Contenidos
- **Funcionalidad / Característica:** Visualización de capítulos

- **Datos / Acciones de Entrada:** Ingresar un ID de capítulo existente y ejecutar la función de visualización.
- **Resultado Esperado:** El sistema muestra el título, número, introducción y tema del capítulo correctamente.
- **Requerimientos de Ambiente de Pruebas:** Base de datos con al menos un capítulo creado.
- **Precondiciones:** El ID del capítulo debe existir en la base de datos.
- **Dependencias:** Ninguna.

CP01_02 — Intento de visualizar capítulo inexistente

- **Id:** CP01_02
- **Caso de Uso Relacionado:** CU_01 Visualizar contenido
- **Caso de Prueba:** Capítulo inexistente o eliminado
- **Descripción:** El usuario intenta visualizar un capítulo que no se encuentra en la base de datos.
- **Fecha:** 3/11/2025
- **Área Funcional / Subproceso:** Contenidos
- **Funcionalidad / Característica:** Manejo de errores en visualización
- **Datos / Acciones de Entrada:** Ingresar un ID que no exista en la base de datos y ejecutar la función de visualización.
- **Resultado Esperado:** El sistema muestra un mensaje de error indicando que el capítulo no existe, sin interrumpir la ejecución.
- **Requerimientos de Ambiente de Pruebas:** Base de datos sin registros para el ID indicado.
- **Precondiciones:** Ninguna.
- **Dependencias:** Ninguna.

CP02_01 — Crear capítulo con datos válidos

- **Id:** CP02_01
- **Caso de Uso Relacionado:** CU_02 Administrar contenido
- **Caso de Prueba:** Creación exitosa de capítulo
- **Descripción:** El usuario crea un capítulo nuevo ingresando todos los campos requeridos con valores válidos.
- **Fecha:** 3/11/2025
- **Área Funcional / Subproceso:** Administración de contenidos
- **Funcionalidad / Característica:** Creación de capítulos
- **Datos / Acciones de Entrada:** Ingresar título, número, introducción y tema, y ejecutar la función de creación.
- **Resultado Esperado:** El sistema guarda correctamente el nuevo capítulo en la base de datos y devuelve un identificador válido.
- **Requerimientos de Ambiente de Pruebas:** Base de datos operativa y conexión estable.
- **Precondiciones:** Ninguna.
- **Dependencias:** Ninguna.

CP02_02 — Crear capítulo con datos inválidos

- **Id:** CP02_02

- **Caso de Uso Relacionado:** CU_02 Administrar contenido
- **Caso de Prueba:** Validación de datos en la creación
- **Descripción:** El usuario intenta crear un capítulo con uno o más campos vacíos o con datos no válidos.
- **Fecha:** 3/11/2025
- **Área Funcional / Subproceso:** Administración de contenidos
- **Funcionalidad / Característica:** Validación en alta de capítulos
- **Datos / Acciones de Entrada:** Dejar en blanco uno o más campos obligatorios (por ejemplo, título o número) y ejecutar la función de creación.
- **Resultado Esperado:** El sistema muestra un mensaje de error o rechaza la operación sin guardar el capítulo.
- **Requerimientos de Ambiente de Pruebas:** Base de datos operativa.
- **Precondiciones:** Ninguna.
- **Dependencias:** Ninguna.

**Los casos de uso los pasamos a Claude. Sabemos que se prueban la interfaz (Tests de integración).*

A futuro podríamos hacer test unitarios de los gestores por ejemplo, pero nos centramos en que funcione la aplicación así para la presentación

3.E- Conclusiones

Durante el desarrollo del trabajo se tomó la decisión de centrar la implementación del código únicamente en los gestores de capítulo y contenido, junto con sus respectivas entidades y la conexión a la base de datos. Esta elección permitió concentrar los esfuerzos en el núcleo funcional del sistema —la creación, organización y manejo de los materiales del libro virtual—, garantizando una base sólida y coherente con la arquitectura planteada. Posteriormente, sobre esta estructura se podrán incorporar otros módulos y servicios de forma modular.

Otras de las principales decisiones de diseño, fue la adopción de Python con el framework FastAPI para el desarrollo de la aplicación web, por su simplicidad y capacidad de integración con servicios en la nube. También se definió la utilización de Amazon Web Services (AWS) como entorno de despliegue, empleando EC2 para el servidor de aplicación, RDS MySQL para el almacenamiento de datos estructurados y S3 para los recursos multimedia. Estas elecciones se realizaron con el objetivo de lograr una arquitectura escalable, segura y fácil de mantener.

Una de las mayores dificultades encontradas fue la realización del diagrama de despliegue, ya que el grupo no tenía experiencia previa con conceptos de infraestructura, redes o servidores en la nube. Comprender el funcionamiento de los distintos servicios y su relación dentro de AWS requirió una etapa de investigación adicional. Finalmente, se logró resolver mediante la consulta de documentación técnica y ejemplos prácticos, comprendiendo la función de cada componente dentro del sistema.

En cuanto a los aprendizajes obtenidos, se resalta la importancia de dividir el software en capas con responsabilidades bien definidas. La diferenciación entre gestores (encargados de la lógica de negocio) y repositorios (encargados de la persistencia de datos) facilitó la comprensión del flujo interno del sistema y su futura escalabilidad. Además, la aplicación del modelo C4 permitió representar de manera progresiva los distintos niveles de abstracción del sistema, favoreciendo la claridad en la documentación.

Finalmente, como mejoras futuras, se propone continuar con la implementación completa de la aplicación, incorporando los demás módulos y la interfaz de usuario, así como la realización de pruebas automatizadas que garanticen la calidad y confiabilidad del sistema.