



**UNIVERSIDAD DE
CÓRDOBA**



LICENCIATURA EN
INFORMÁTICA

Acreditada de Alta Calidad
MEN Res. 10710 25/05/17

Documento
técnico para
proyectos de
Diseño de
Software

Documento de Propuesta de Diseño de Software I, II y III

Componente Gestor de Tareas

Antonio José Gómez Banda - Agomezbanda30@correo.unicordoba.edu.co

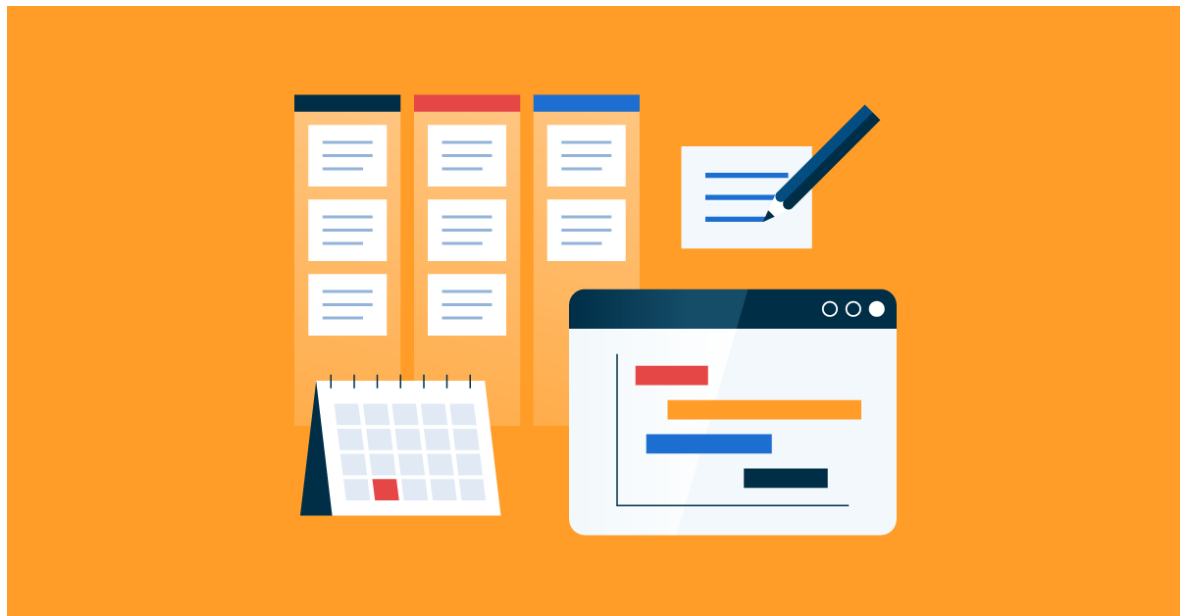
Camilo José Aviléz Guzmán - Cavilezguzman24@correo.unicordoba.edu.co

José Luis Mercado Ibarra - jmercadoibarra@correo.unicordoba.edu.co

Duván Steven Ruiz Benítez - druizbenitez@correo.unicordoba.edu.co

Sebastián López Sotelo – slopezsotelo19@correo.unicordoba.edu.co

Alexander Toscano Ricardo



Breve reseña

Se busca crear un componente de Gestor de Tareas, destinado a plataformas de contenido, que facilite la organización y el control de las responsabilidades diarias. El componente se centrará en ofrecer una interfaz intuitiva y funcionalidades avanzadas para organizar, asignar y monitorear tareas relacionadas con la planificación de actividades o proyectos. Desde la asignación automatizada de tareas hasta el seguimiento en tiempo real del progreso, se espera que este gestor de tareas facilite la colaboración entre equipos. Su desarrollo se presenta como una solución integral para mejorar la eficacia y la calidad en la gestión de contenidos en diversas plataformas.

| | |
|---|-----------|
| ETAPA 1 DISEÑO DE LA APLICACIÓN Y ANÁLISIS DE REQUISITOS | 6 |
| INTRODUCCIÓN | 6 |
| PROPÓSITO DEL DOCUMENTO | 6 |
| ALCANCE DEL PROYECTO | 6 |
| DEFINICIONES Y ACRÓNIMOS | 7 |
| DESCRIPCIÓN GENERAL | 9 |
| OBJETIVOS DEL SISTEMA | 12 |
| FUNCIONALIDAD GENERAL | 12 |
| USUARIOS DEL SISTEMA | 12 |
| RESTRICCIONES | 12 |
| REQUISITOS FUNCIONALES | 12 |
| CASOS DE USO | 12 |
| DESCRIPCIÓN DETALLADA DE CADA CASO DE USO | 12 |
| DIAGRAMAS DE FLUJO DE CASOS DE USO | 12 |
| PRIORIDAD DE REQUISITOS | 15 |
| REQUISITOS NO FUNCIONALES | 20 |
| REQUISITOS DE DESEMPEÑO | 20 |
| REQUISITOS DE SEGURIDAD | 20 |
| REQUISITOS DE USABILIDAD | 20 |
| REQUISITOS DE ESCALABILIDAD | 20 |
| MODELADO E/R | 20 |
| DIAGRAMA DE ENTIDAD-RELACIÓN | 20 |
| DESCRIPCIÓN DE ENTIDADES Y RELACIONES | 21 |
| REGLAS DE INTEGRIDAD | 21 |
| ANEXOS (SI ES NECESARIO) | 21 |
| DIAGRAMAS ADICIONALES | 21 |
| REFERENCIAS | 21 |
| ETAPA 2: PERSISTENCIA DE DATOS CON BACKEND | 22 |
| INTRODUCCIÓN | 22 |
| PROPÓSITO DE LA ETAPA | 22 |
| ALCANCE DE LA ETAPA | 22 |

| | |
|--|-----------|
| DEFINICIONES Y ACRÓNIMOS | 22 |
| DISEÑO DE LA ARQUITECTURA DE BACKEND | 22 |
| DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA | 22 |
| COMPONENTES DEL BACKEND | 22 |
| DIAGRAMAS DE ARQUITECTURA | 22 |
| ELECCIÓN DE LA BASE DE DATOS | 22 |
| EVALUACIÓN DE OPCIONES (SQL o NoSQL) | 22 |
| JUSTIFICACIÓN DE LA ELECCIÓN | 22 |
| DISEÑO DE ESQUEMA DE BASE DE DATOS | 23 |
| IMPLEMENTACIÓN DEL BACKEND | 23 |
| ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN | 23 |
| CREACIÓN DE LA LÓGICA DE NEGOCIO | 23 |
| DESARROLLO DE ENDPOINTS Y APIs | 23 |
| AUTENTICACIÓN Y AUTORIZACIÓN | 23 |
| CONEXIÓN A LA BASE DE DATOS | 23 |
| CONFIGURACIÓN DE LA CONEXIÓN | 23 |
| DESARROLLO DE OPERACIONES CRUD | 23 |
| MANEJO DE TRANSACCIONES | 23 |
| PRUEBAS DEL BACKEND | 23 |
| DISEÑO DE CASOS DE PRUEBA | 23 |
| EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN | 24 |
| MANEJO DE ERRORES Y EXCEPCIONES | 24 |
| ETAPA 3: CONSUMO DE DATOS Y DESARROLLO FRONTEND | 25 |
| INTRODUCCIÓN | 25 |
| PROPÓSITO DE LA ETAPA | 25 |
| ALCANCE DE LA ETAPA | 25 |
| DEFINICIONES Y ACRÓNIMOS | 25 |
| CREACIÓN DE LA INTERFAZ DE USUARIO (UI) | 25 |
| DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS | 25 |
| CONSIDERACIONES DE USABILIDAD | 25 |
| MAQUETACIÓN RESPONSIVA | 25 |

| | |
|---|-----------|
| PROGRAMACIÓN FRONTEND CON JAVASCRIPT (JS) | 25 |
| DESARROLLO DE LA LÓGICA DEL FRONTEND | 25 |
| MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS | 25 |
| USO DE BIBLIOTECAS Y FRAMEWORKS (SI APLICABLE) | 26 |
| CONSUMO DE DATOS DESDE EL BACKEND | 26 |
| CONFIGURACIÓN DE CONEXIONES AL BACKEND | 26 |
| OBTENCIÓN Y PRESENTACIÓN DE DATOS | 26 |
| ACTUALIZACIÓN EN TIEMPO REAL (SI APLICABLE) | 26 |
| INTERACCIÓN USUARIO-INTERFAZ | 26 |
| MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS | 26 |
| IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS | 26 |
| MEJORAS EN LA EXPERIENCIA DEL USUARIO | 26 |
| PRUEBAS Y DEPURACIÓN DEL FRONTEND | 26 |
| DISEÑO DE CASOS DE PRUEBA DE FRONTEND | 26 |
| PRUEBAS DE USABILIDAD | 26 |
| DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO | 27 |
| IMPLEMENTACIÓN DE LA LÓGICA DE NEGOCIO EN EL FRONTEND | 27 |
| MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO) | 27 |
| VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND | 27 |
| INTEGRACIÓN CON EL BACKEND | 27 |
| VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND | 27 |
| PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND | 27 |

Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

Introducción

En la actualidad, la organización y gestión del tiempo se han convertido en aspectos cruciales para la productividad y el bienestar personal. En este contexto, los gestores de tareas emergen como herramientas indispensables, facilitando la planificación, seguimiento y ejecución de actividades de manera eficiente.

Este componente se centra en la creación de un gestor de tareas que permita a los usuarios mantener un control efectivo sobre sus responsabilidades diarias. Desde la captura inicial de tareas hasta su finalización, este sistema ofrece una interfaz intuitiva y funcionalidades personalizables para adaptarse a las necesidades individuales de cada usuario.

A través de la combinación de diseño centrado en el usuario y tecnología de vanguardia, este gestor de tareas aspira a proporcionar una experiencia fluida y satisfactoria, ayudando a los usuarios a alcanzar sus objetivos de manera más eficaz y organizada.

Propósito del Documento

El presente documento tiene como finalidad documentar el proceso de diseño, análisis e implementación de software de tipo educativo, comercial, OVA, componente o módulo de aplicaciones. Se divide en tres etapas para facilitar el entendimiento y aplicación a gran escala en la asignatura de diseño de software.

- Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

Esta etapa cumple la tarea de recoger todas las competencias desarrolladas en todas las áreas de formación del currículo de la licenciatura en Informática y Medios Audiovisuales y ponerlas a prueba en el diseño y análisis de un producto educativo que se base en las teorías de aprendizaje estudiadas, articule las estrategias de enseñanza con uso de TIC y genere innovaciones en educación con productos interactivos que revelen una verdadera naturaleza educativa. Estos productos deben aprovechar las fortalezas adquiridas en las áreas de tecnología e informática, técnicas y herramientas, medios audiovisuales y programación y sistemas, para generar productos software interactivos que permitan a los usuarios disfrutar de lo que aprenden, a su propio ritmo. Todo esto en el marco de un proceso metodológico (metodologías de desarrollo de software como MODESEC, SEMLI, etc.) que aproveche lo aprendido en la línea de gestión y lo enriquezca con elementos de la Ingeniería de Software.

- Etapa 2: Persistencia de Datos con Backend – Servidor

En la etapa 2 se continúa con los lineamientos de la etapa 1, para seguir adicionando elementos de diseño e implementación de software, enfocados en el desarrollo de APIs, servidores o microservicios que permitan soportar aplicaciones cliente del software educativo; en este sentido, el curso presenta los conceptos de los sistemas de bases de datos, su diseño lógico, la organización de los sistemas manejadores

de bases de datos, los lenguaje de definición de datos y el lenguaje de manipulación de datos SQL y NoSQL; de tal manera que los estudiantes adquieran las competencias para analizar, diseñar y desarrollar aplicaciones para gestionar y almacenar grandes cantidades de datos, mediante el uso de técnicas adecuadas como el diseño y modelo lógico y físico de base datos, manejo de los sistemas de gestión de bases de datos, algebra relacional, dominio del lenguaje SQL como herramienta de consulta, tecnología cliente / servidor; igualmente, se definirán los elementos necesarios para el acceso a dichas bases de datos, como la creación del servidor API, utilizando tecnologías de vanguardia como node.js, express, Nest.js, Spring entre otros; para, finalmente converger en el despliegue de la API utilizando servicios de hospedaje en la nube, preferiblemente gratuitos. También podrá implementar servidores o API 's con inteligencia artificial o en su defecto crear una nueva capa que consuma y transforme los datos obtenidos de la IA. El desarrollo del curso se trabajará por proyectos de trabajo colaborativo que serán evaluados de múltiples maneras, teniendo en cuenta más el proceso que el resultado.

Etapas 3: Consumo de Datos y Desarrollo Frontend – Cliente

La etapa 3 el estudiante está en capacidad de establecer la mejor elección de herramientas de consumo de datos y técnicas en aras de lograr el mejor producto a nivel de software o hardware acorde a los requerimientos funcionales y no funcionales del problema a solucionar. En este punto el estudiante puede consumir los datos a través de un cliente que puede ser una aplicación de celular, una aplicación de escritorio, una página web, IoT(internet de las cosas) o incluso, artefactos tecnológicos. El diseño gráfico es de los requisitos esenciales en la capa de presentación, por lo tanto, se requieren los cursos de diseño gráfico vistos previamente. Los elementos anteriores nos permiten elegir el paradigma y tecnología para desarrollar nuestras aplicaciones, teniendo en cuenta que podríamos desarrollar aplicaciones de tipo cliente

Alcance del Proyecto

El alcance que se pretende tener con el gestor de tareas se centra en desarrollar un componente digital que facilite la gestión, organización, creación, realización y filtración de tareas de manera eficiente, para generar en el usuario la mejor experiencia posible.

- Creación de una tarea.
- Búsqueda de una tarea: Permite al usuario buscar tareas según diferentes criterios, como palabras clave, fechas de vencimiento, etiquetas o categorías, facilitando la organización y la localización de tareas específicas dentro del sistema.
- Asignación de tareas: Posibilita asignar tareas específicas a diferentes usuarios o equipos, facilitando la colaboración y el trabajo en equipo.
- Modificación de una tarea.
- Eliminación de una tarea.

- Listar tareas.
- Listar tareas por usuarios.
- Listar tareas por fecha.
- Listar tareas por vencimiento.
- Listar tareas pendientes.
- Editar estados de la tarea: los estados de la tarea serian iniciado, en proceso, terminado, en revisión.
- Gestión de tareas basada en listas y calendarios: Permite al usuario organizar sus tareas tanto en forma de listas como en un calendario, lo que proporciona flexibilidad para visualizar y planificar las actividades de manera adecuada según sus preferencias y necesidades.
- Priorización de tareas con niveles de importancia: Permite al usuario asignar niveles de importancia a las tareas, lo que ayuda a priorizarlas en función de su urgencia o relevancia, facilitando la gestión del tiempo y la atención en las actividades más importantes.
- Integraciones con aplicaciones de calendario y correo electrónico: Permite al usuario sincronizar el gestor de tareas con aplicaciones de calendario y correo electrónico, lo que facilita la coordinación de actividades y la recepción de notificaciones en diferentes plataformas.
- Notificaciones usando sonidos, vibraciones, síntesis de voz: Permite al usuario recibir notificaciones de nuevas tareas, recordatorios o eventos importantes a través de diferentes medios, como sonidos, vibraciones o síntesis de voz, garantizando que no se pierdan información importante y se mantenga al tanto de las actividades planificadas.
- Permite agregar archivos adjuntos como imágenes, PDF, vídeos, documentos de Office: Permite al usuario adjuntar archivos multimedia o documentos a las tareas, lo que facilita el acceso a la información relevante y la colaboración en proyectos que requieran documentos o recursos adicionales.
- Recordatorios y notificaciones: Envía recordatorios y notificaciones automáticas sobre tareas próximas a vencer o completadas.
- Comentarios y discusiones: Permite a los usuarios comentar y discutir sobre las tareas dentro de la plataforma, facilitando la comunicación y la colaboración.

Para futuras versiones, se proponen características adicionales como:

- Generación de informes y análisis: Ofrece herramientas para generar informes y análisis sobre el progreso y rendimiento de las tareas.
- Adición de tareas a través de la voz: Permite al usuario agregar nuevas tareas mediante comandos de voz, lo que proporciona una forma rápida y conveniente de ingresar información al gestor de tareas sin necesidad de escribir.

- Creación de hábitos y recordatorios recurrentes: Permite al usuario establecer hábitos o tareas recurrentes y configurar recordatorios automáticos para realizarlas en intervalos específicos, lo que ayuda a mantener la consistencia y la disciplina en la realización de actividades repetitivas.
- Personalización de la interfaz: Ofrece opciones de personalización de la interfaz para adaptarse a las preferencias y necesidades de los usuarios.

Definiciones y Acrónimos

API: Interfaz de Programación de Aplicaciones (Application Programming Interface).

DBMS: Sistema de Gestión de Bases de Datos (Database Management System).

SQL: Lenguaje de Consulta Estructurada (Structured Query Language).

HTTP: Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol).

REST: Transferencia de Estado Representacional (Representational State Transfer).

JSON: Notación de Objetos de JavaScript (JavaScript Object Notation).

JWT: Token de Web JSON (JSON Web Token).

CRUD: Crear, Leer, Actualizar y Borrar (Create, Read, Update, Delete).

ORM: Mapeo Objeto-Relacional (Object-Relational Mapping).

MVC: Modelo-Vista-Controlador (Model-View-Controller).

API RESTful: API que sigue los principios de REST.

CI/CD: Integración Continua / Entrega Continua (Continuous Integration / Continuous Delivery).

SaaS: Software como Servicio (Software as a Service).

SSL/TLS: Capa de sockets seguros/Seguridad de la Capa de Transporte (Secure Sockets Layer/Transport Layer Security).

HTML: Lenguaje de Marcado de Hipertexto (Hypertext Markup Language).

CSS: Hojas de Estilo en Cascada (Cascading Style Sheets).

JS: JavaScript. DOM: Modelo de Objeto del Documento (Document Object Model).

UI: Interfaz de Usuario (User Interface). UX: Experiencia del Usuario (User Experience).

SPA: Aplicación de Página Única (Single Page Application).

AJAX: Asíncrono JavaScript y XML (Asynchronous JavaScript and XML).

CMS: Sistema de Gestión de Contenido (Content Management System).

CDN: Red de Distribución de Contenido (Content Delivery Network).

SEO: Optimización de Motores de Búsqueda (Search Engine Optimization).

IDE: Entorno de Desarrollo Integrado (Integrated Development Environment).

CLI: Interfaz de Línea de Comandos (Command Line Interface).

PWA: Aplicación Web Progresiva (Progressive Web App).

CRUD: Operaciones básicas de creación (Create), lectura (Read), actualización (Update) y eliminación (Delete) de datos en una base de datos.

AJAX: Asincrónico JavaScript y XML (Asynchronous JavaScript and XML). Permite actualizar partes de una página web sin necesidad de recargarla completamente, lo que mejora la experiencia del usuario.

OAuth: Protocolo de autorización que permite a una aplicación obtener acceso limitado a recursos en un servicio en nombre del propietario de los recursos, sin necesidad de compartir sus credenciales.

GUI: Interfaz Gráfica de Usuario (Graphical User Interface). Permite a los usuarios interactuar con el software a través de elementos visuales como ventanas, botones y menús.

GESTOR DE TAREAS: es una herramienta que te ayuda a organizar y administrar tus actividades diarias, proyectos y responsabilidades. Puede ser tanto una aplicación digital como una técnica de gestión personal.

INTERFAZ INTUITIVA: Una interfaz de usuario diseñada de manera que sea fácil de entender y utilizar, sin requerir una explicación extensa o capacitación previa por parte del usuario.

PERSONALIZABLE: La capacidad de adaptar o modificar una herramienta o aplicación de acuerdo con las preferencias y necesidades individuales de cada usuario.

DISEÑO CENTRADO EN EL USUARIO: Un enfoque de diseño que prioriza las necesidades, deseos y habilidades del usuario final durante el proceso de desarrollo de productos o servicios.

EFICIENCIA: La capacidad de realizar una tarea o alcanzar un objetivo utilizando la menor cantidad de recursos posibles, como tiempo, esfuerzo o dinero.

PRODUCTIVIDAD: La medida en que se logran los resultados deseados de manera efectiva y eficiente, maximizando el rendimiento y minimizando el desperdicio.

PLANIFICACIÓN: El proceso de establecer objetivos, identificar acciones y recursos necesarios, y establecer un cronograma para alcanzar esos objetivos de manera sistemática.

SEGUIMIENTO: La acción de monitorear y controlar el progreso de las tareas o actividades para asegurarse de que se estén llevando a cabo de acuerdo con el plan establecido.

INNOVACIÓN: La introducción de nuevas ideas, métodos o productos que generan cambios positivos y mejoras significativas en la forma en que se realizan las actividades o se satisfacen las necesidades.

EXPERIENCIA DEL USUARIO: La percepción general y la satisfacción del usuario al interactuar con un producto o servicio, que incluye aspectos como la facilidad de uso, la eficiencia y la utilidad percibida.

BACKEND: La parte de un sistema informático que se encarga del procesamiento y almacenamiento de datos, así como de la lógica de negocio, generalmente no visible para el usuario final.

PERSISTENCIA DE DATOS: La capacidad de almacenar y recuperar datos de manera permanente en un sistema informático, incluso después de que se haya cerrado la aplicación o se haya apagado el dispositivo.

Descripción General

Objetivos del Sistema

Desarrollar un módulo de gestión de tareas para plataformas de contenidos, que permita a los usuarios organizar, asignar y dar seguimiento a las actividades de manera colaborativa y eficiente. Este componente estará diseñado para mejorar la coordinación entre equipos, facilitando la asignación de tareas, el seguimiento del progreso y la gestión de plazos.

Conceptos de las entidades

Funcionalidad General

Crear nueva tarea

Titulo de la tarea

Descripción:
.....
.....

Asignar a

Fecha de inicio

| Ene | Feb | Mar |
|-----|-----|-----|
| 1 | 2 | 3 |
| | | |
| | | |
| | | |

Fecha de cierre

| Ene | Feb | Mar |
|-----|-----|-----|
| 1 | 2 | 3 |
| | | |
| | | |
| | | |

Prioridad

☐

Alta

☒

Media

☐

Baja

Estado

☐ En progreso

☐ En revisión

☐ Hecho

Guardar

Cancelar

Asignar tarea

Titulo de la tarea

Descripción:
.....
.....

Asignar a

rubena20@gmail.es

maríabaltazar30@gmail.es

Fecha de inicio

| Ene | Feb | Mar |
|-----|-----|-----|
| 1 | 2 | 3 |
| | | |
| | | |
| | | |

Fecha de cierre

| Ene | Feb | Mar |
|-----|-----|-----|
| 1 | 2 | 3 |
| | | |
| | | |
| | | |

Prioridad

☒

Alta

☐

Media

☐

Baja

Estado

☐ En progreso

☐ En revisión

☐ Hecho

Guardar

Cancelar

Modificar tarea

Título de la tarea

Descripción:

Asignar a

rubena20@gmail.es

maríabaltazar30@gmail.es

Fecha de inicio

Dom

Feb

Mar

| | | |
|---|---|---|
| 1 | 2 | 3 |
| | | |
| | | |
| | | |

Fecha de cierre

Dom

Feb

Mar

| | | |
|---|---|---|
| 1 | 2 | 3 |
| | | |
| | | |
| | | |

Prioridad

☐
☐
☐

Alta

Media

Baja

Estado

☐ En progreso
☐ En revisión
☐ Hecho

Guardar Cambios

Eliminar

Buscar tarea

Nombre de la tarea

Bu...

Bue

Busto

Buseta

Bueno

Buenavista

Buenos días

Buscar

Modificar

Lista de tareas

1

2

3

4

5

Tarea 1

Fecha de vencimiento

Dom

Feb

Mar

| | | |
|---|---|---|
| 1 | 2 | 3 |
| | | |
| | | |
| | | |

Tarea 2

Fecha de vencimiento

Dom

Feb

Mar

| | | |
|---|---|---|
| 1 | 2 | 3 |
| | | |
| | | |
| | | |

Tarea 3

Fecha de vencimiento

Dom

Feb

Mar

| | | |
|---|---|---|
| 1 | 2 | 3 |
| | | |
| | | |
| | | |

Editar

Eliminar

Eliminar tarea

Título de la tarea

¿Está seguro que desea eliminar esta tarea?

Si

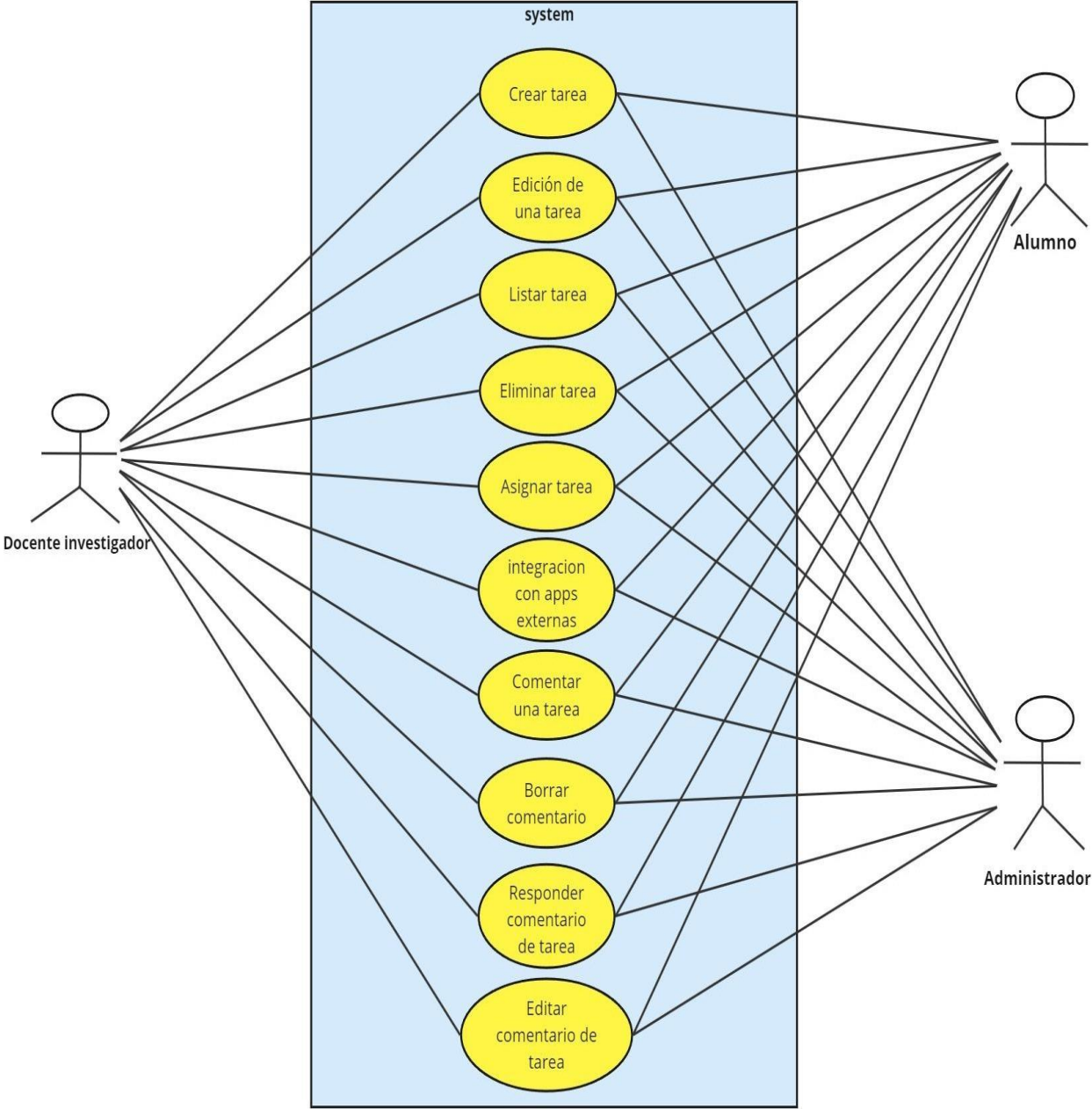
No

Guardar Cambios

Cancelar

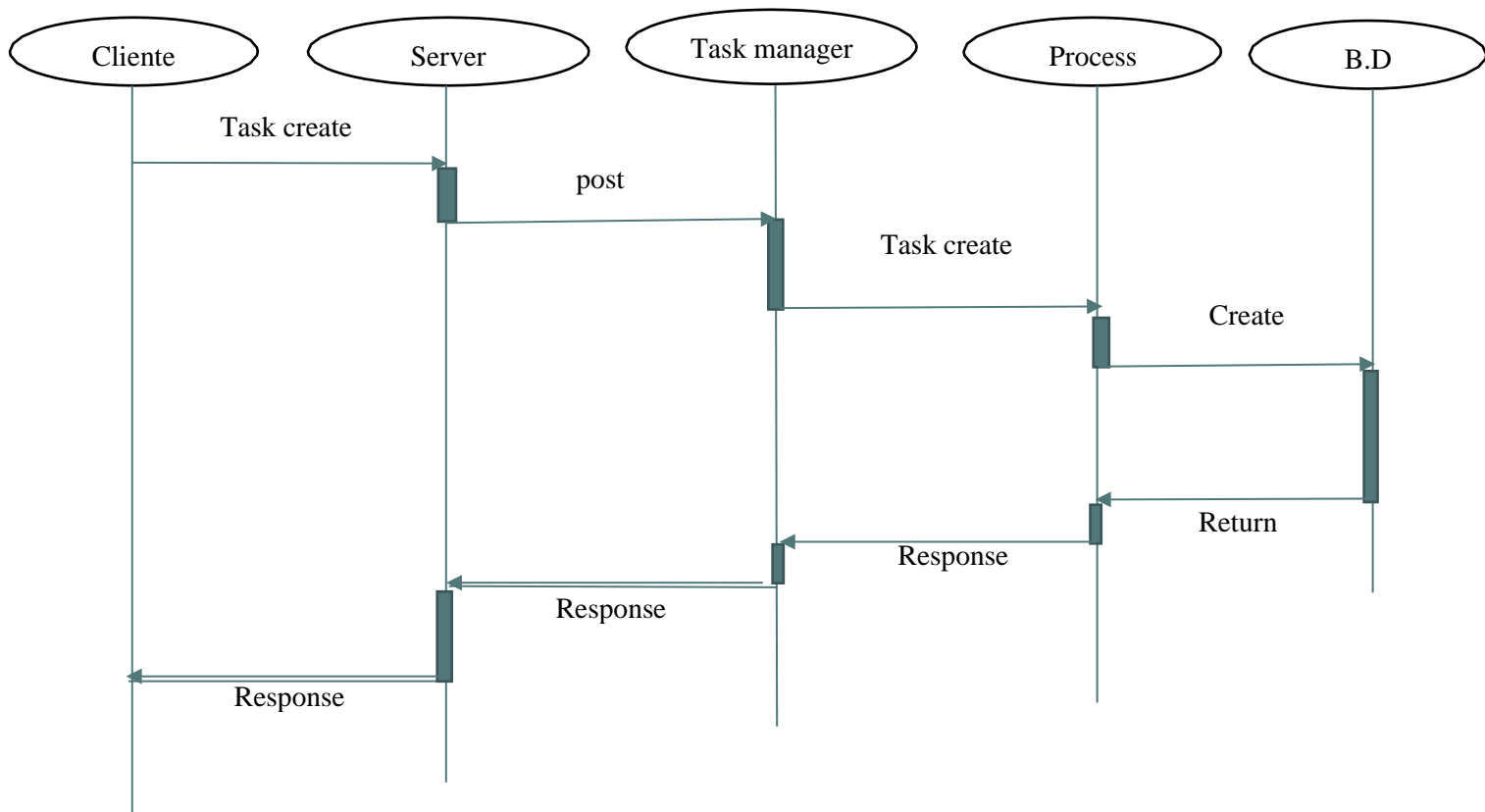
14

Casos de Uso

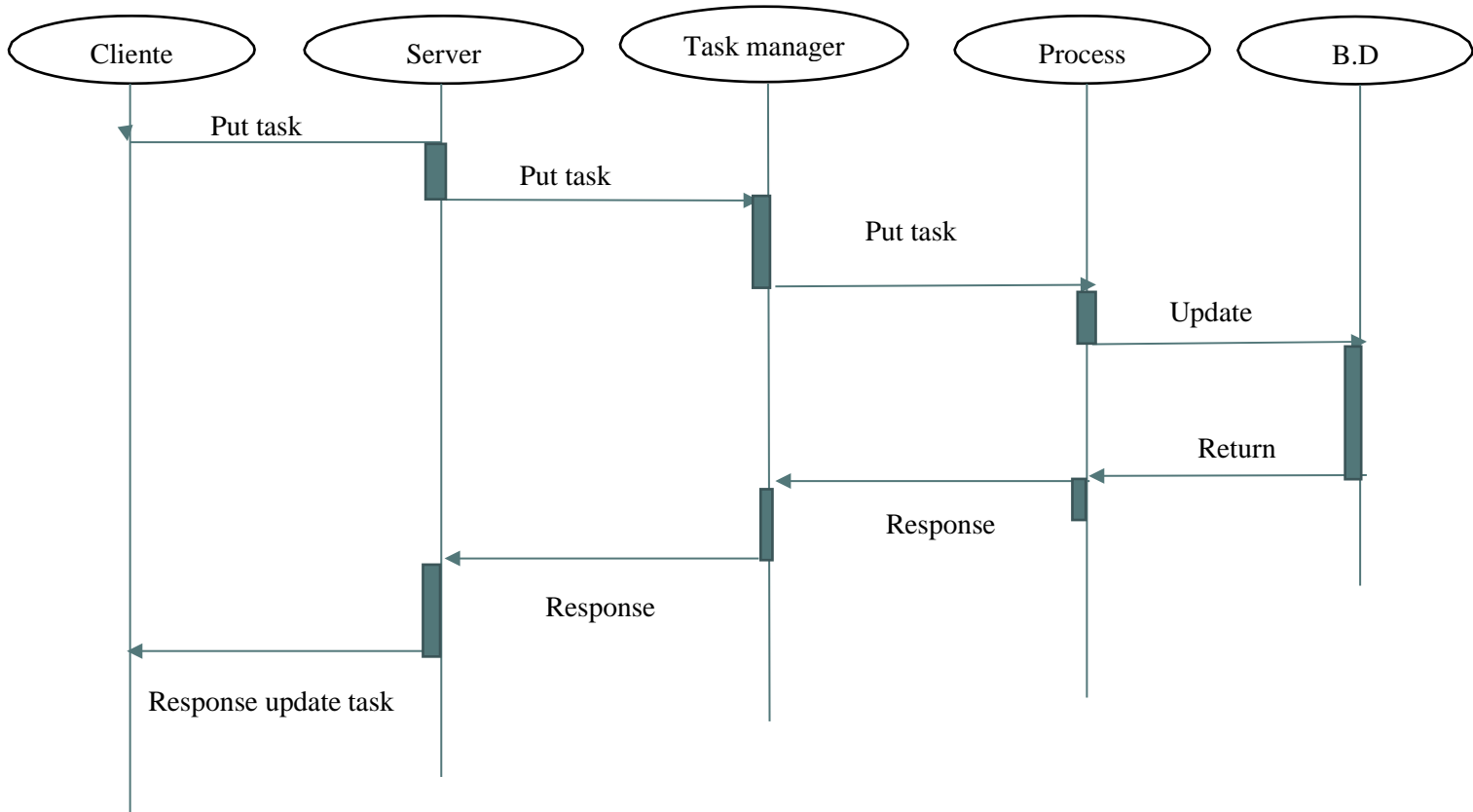


Diagramas de Secuencia

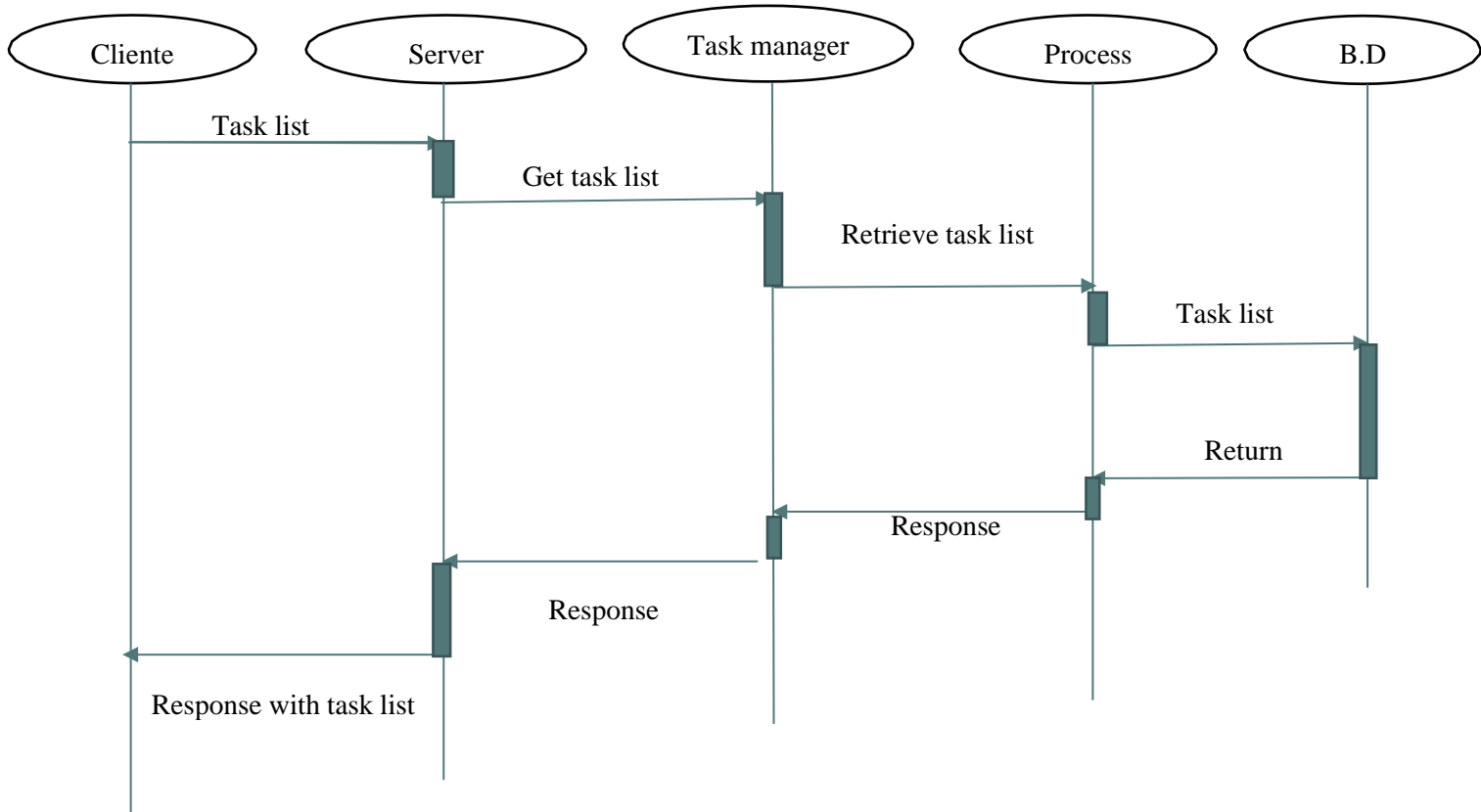
Crear tarea:



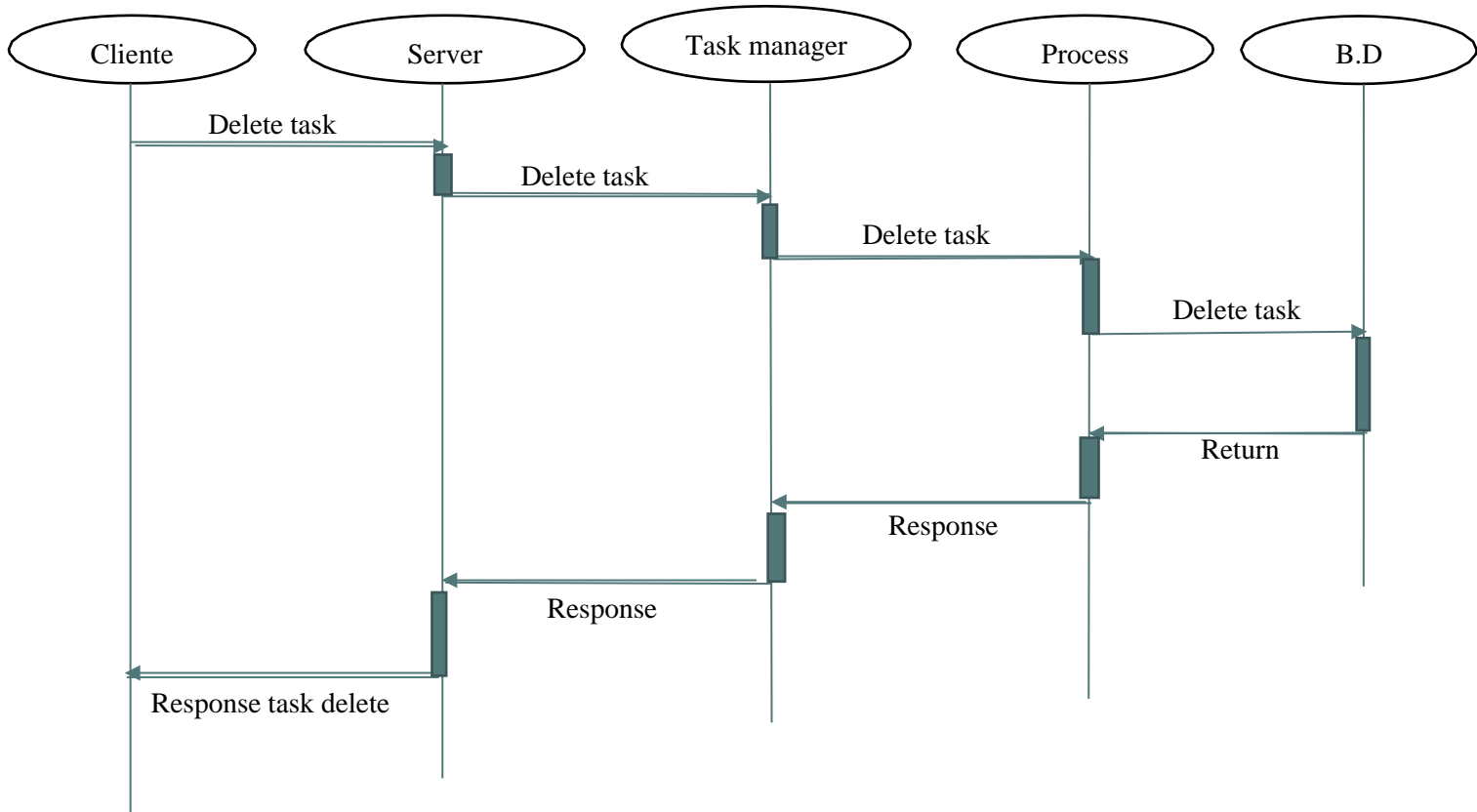
Editar tarea:



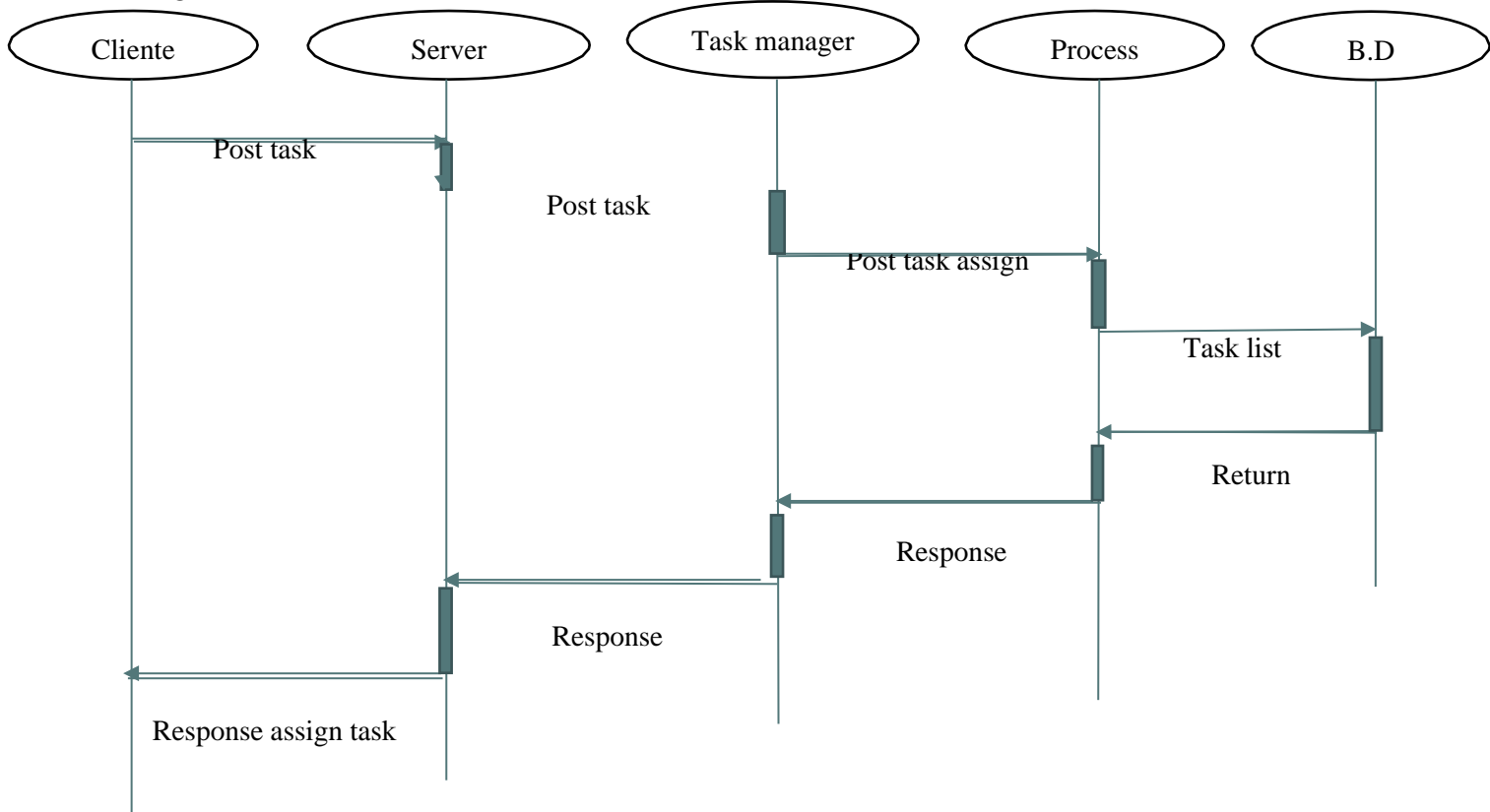
Listar tarea:



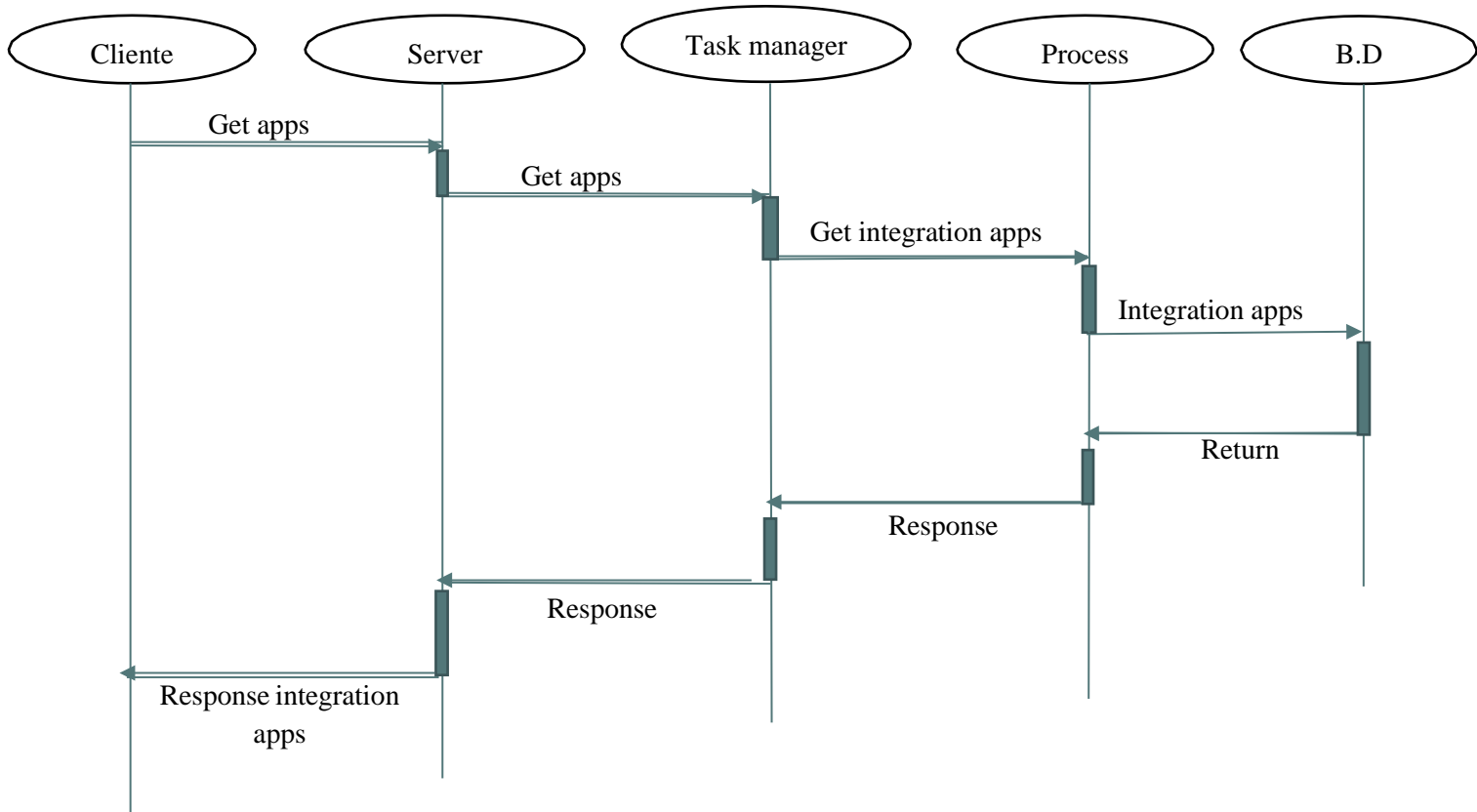
Eliminar tarea:



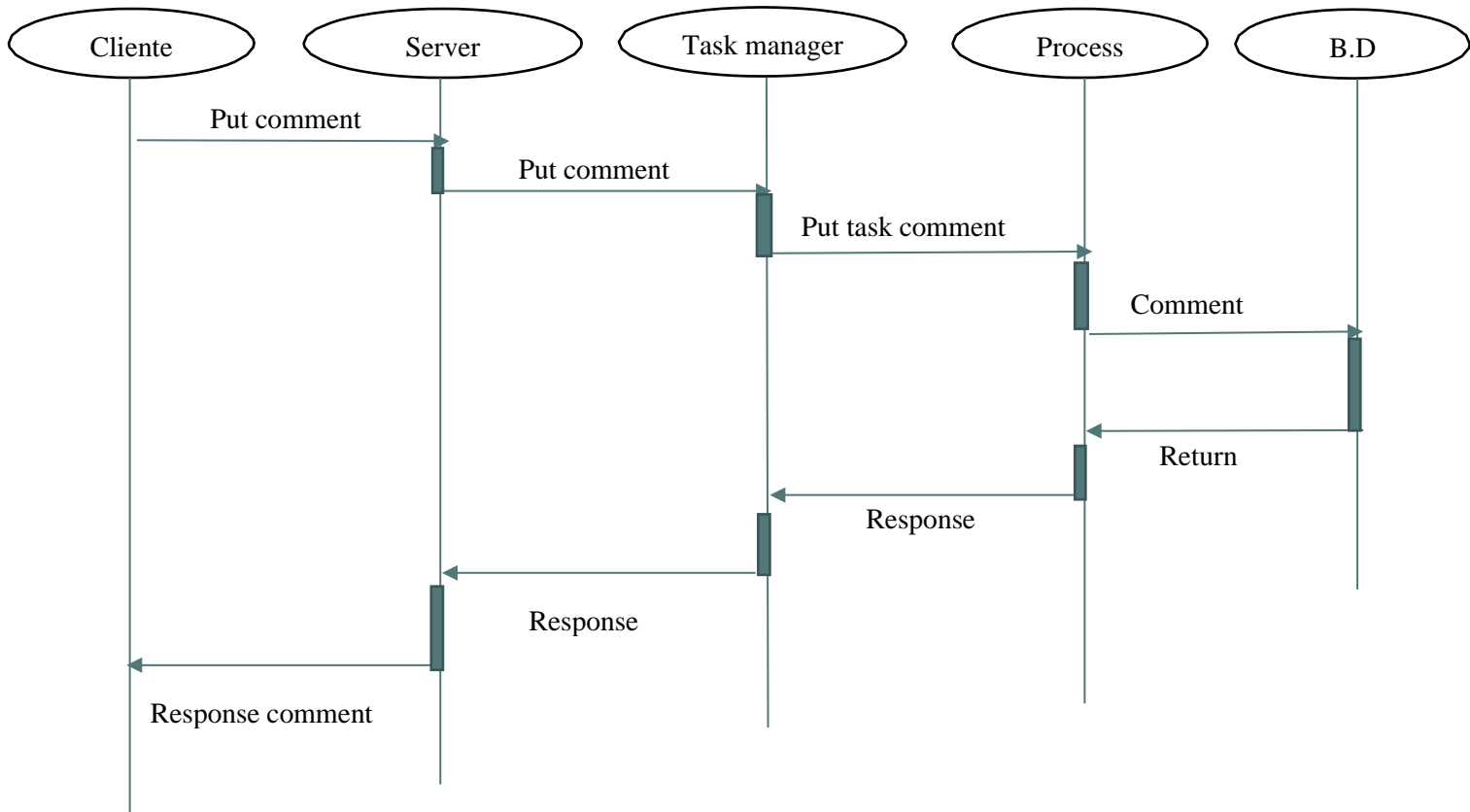
Asignar Tarea:



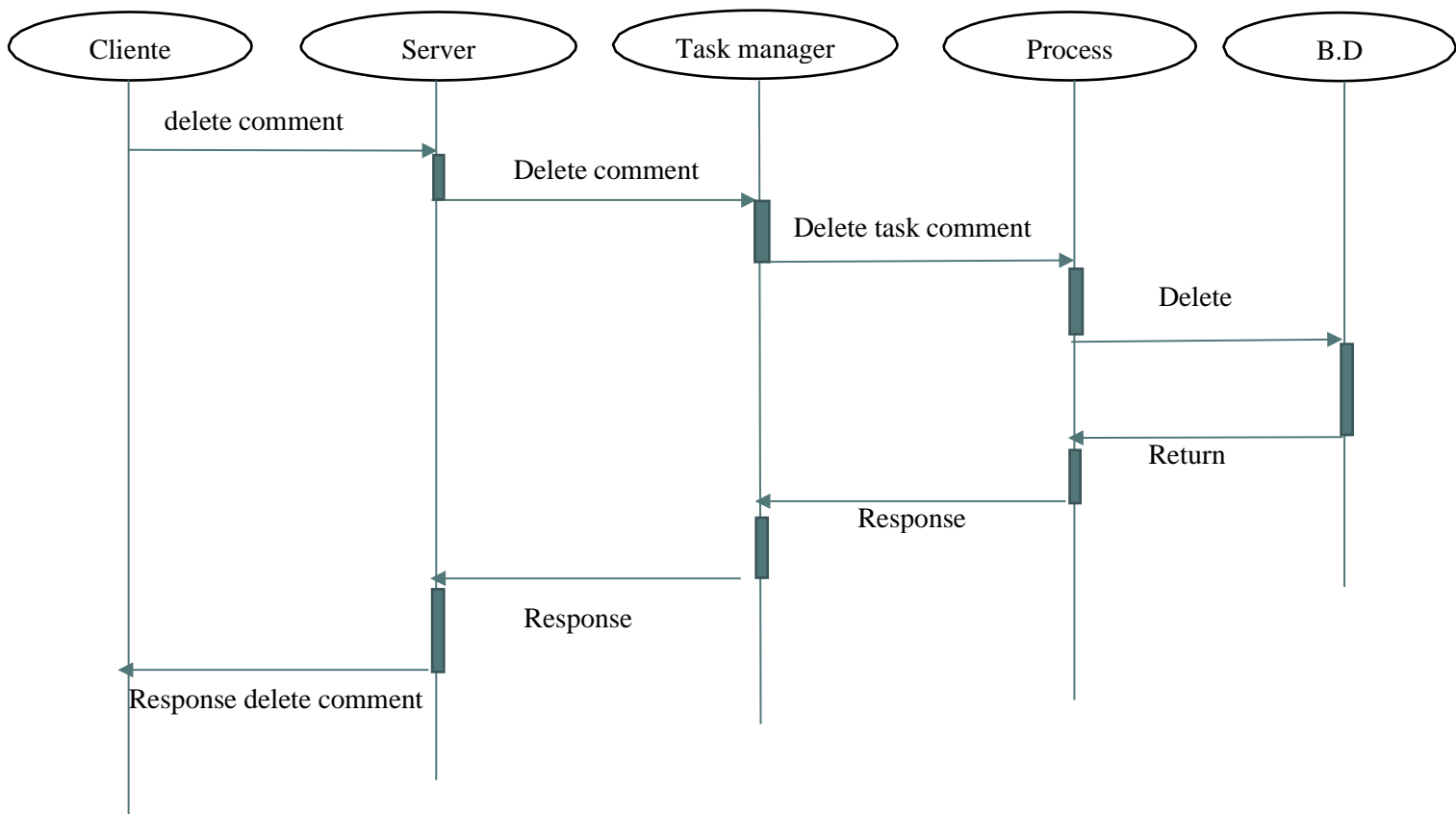
Integración con apps externas:



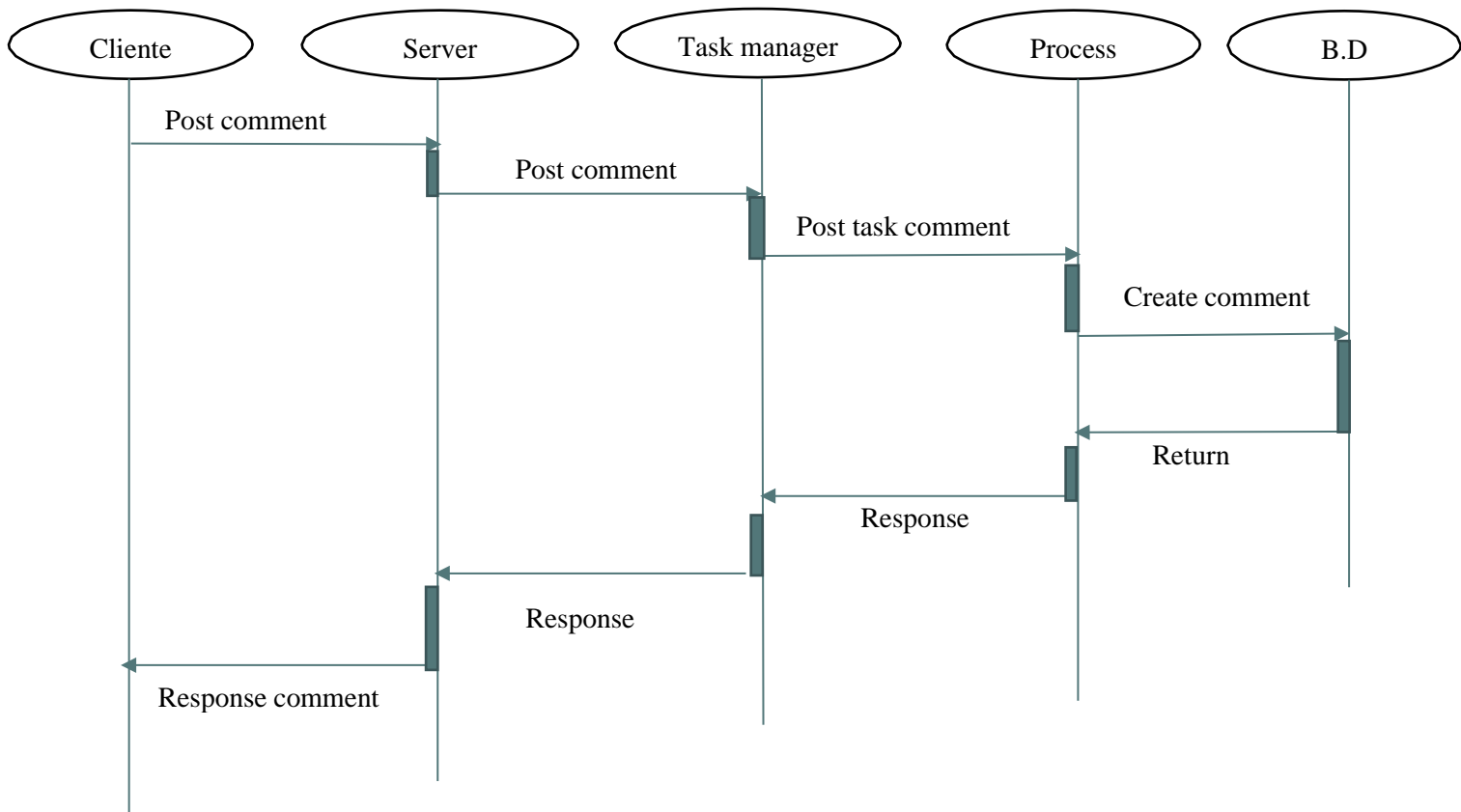
Comentar Tarea:



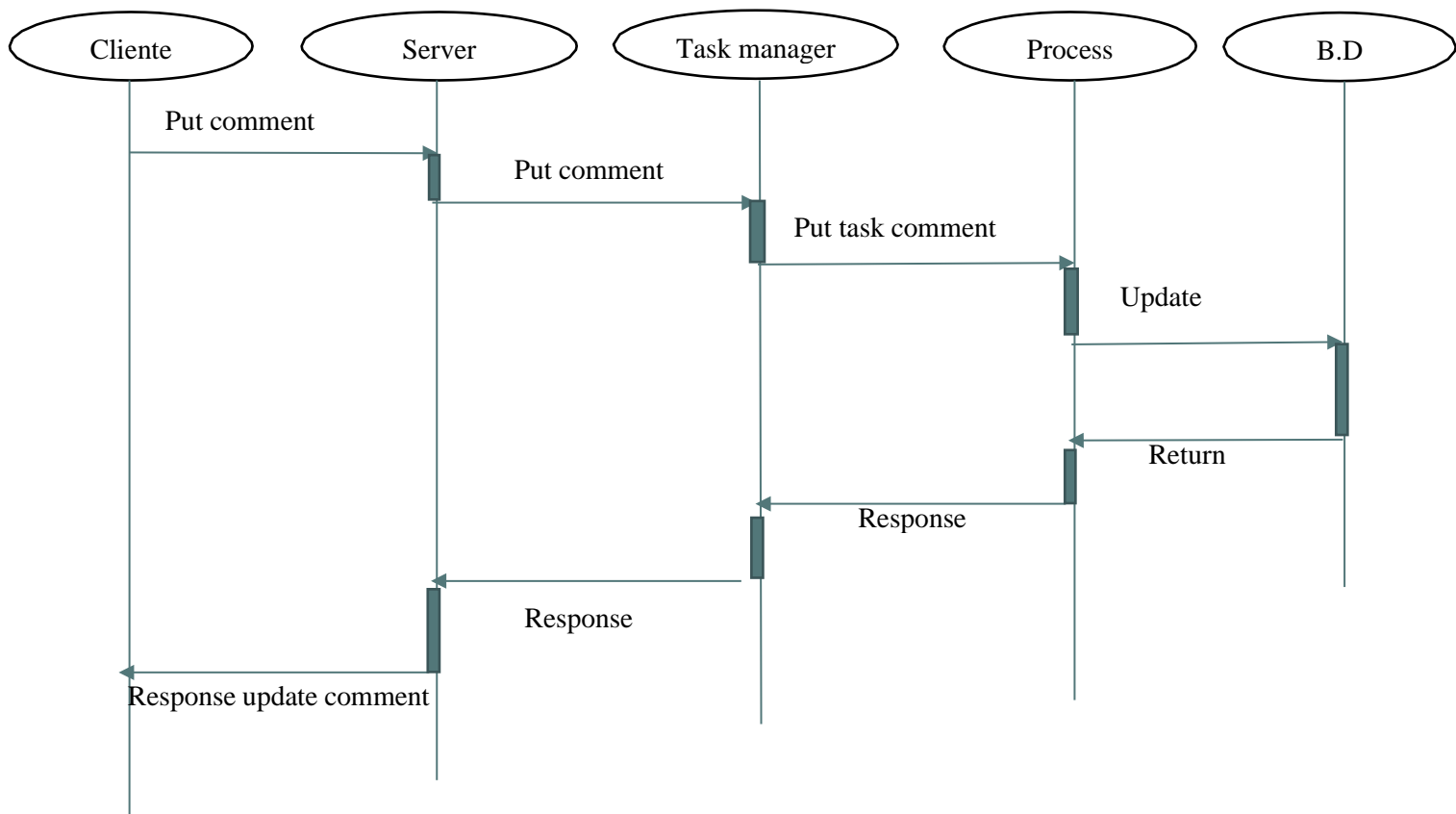
Borrar comentario tarea:



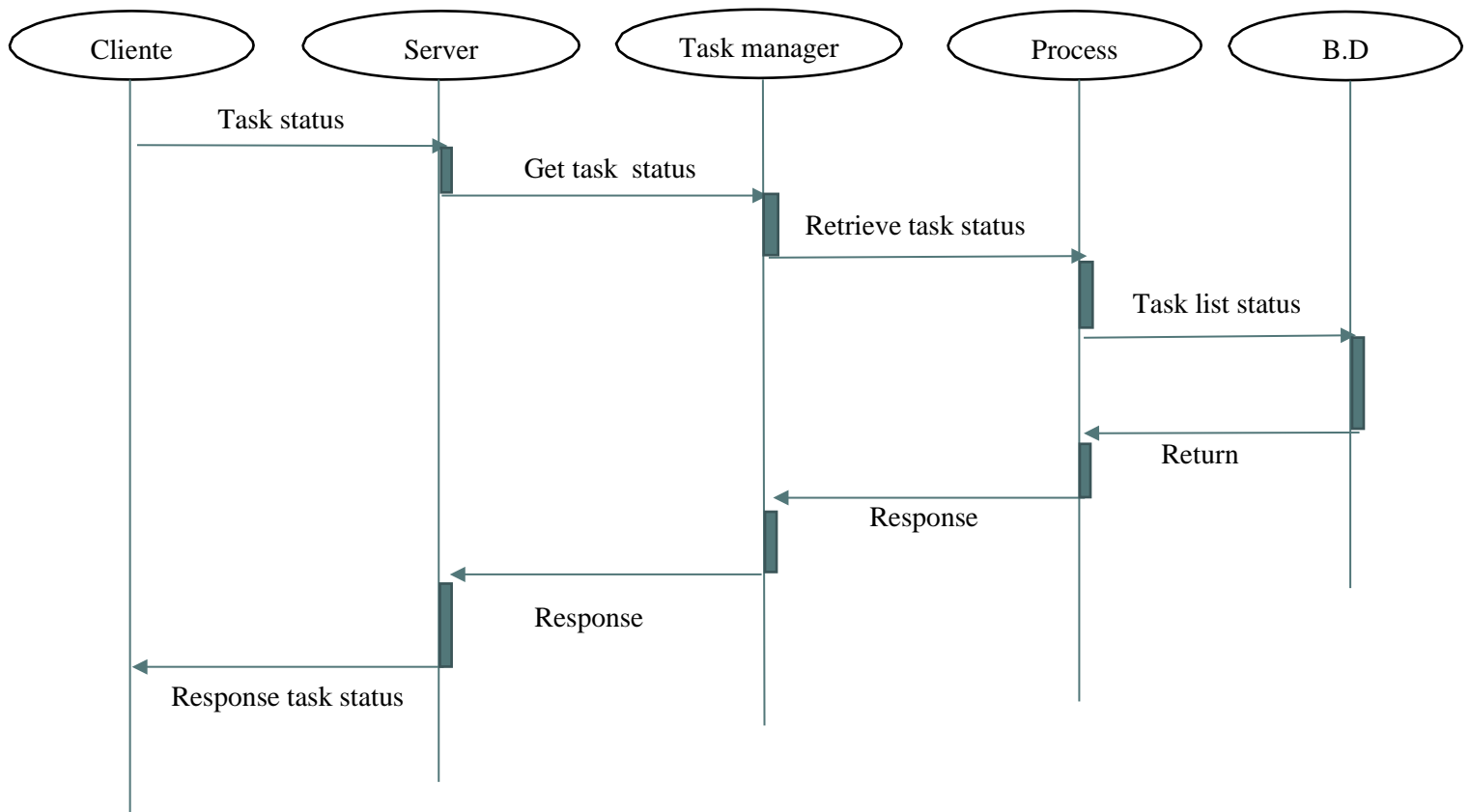
Responder comentario tarea:



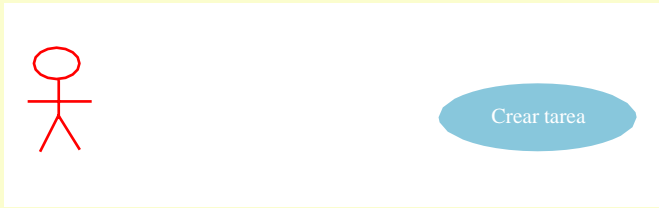
Editar comentario tarea:



Estado Tarea:



Diagramas de Flujo de Casos de Uso



Versión: 1
Urgencia: 5
Esfuerzo: 4

AI: Adicionar Item
NT: Nombre Tarea
CT: Crear Tarea

1. Crear tarea

Flujo: Crear tarea

Prueba: Seleccionar nuevo item, escribir nombre de tarea, crear tarea.

a. Crear tarea

Flujo: AI, NT, CT



Versión: 1
Urgencia: 5
Esfuerzo: 4

ET: Editar tarea
ST: Seleccionar tarea
AD: Actualizar Datos

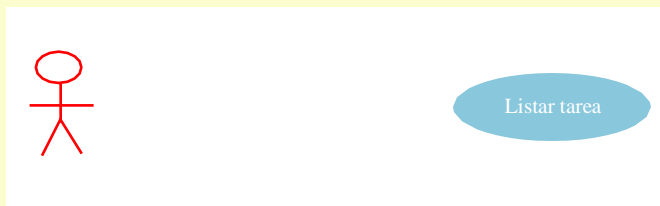
2. Edición de una tarea

Flujo: edición de tarea

Prueba: Seleccionar tarea, modificar tarea, guardar cambios.

a. Edición de tarea

Flujo: ST, ET, AD



Versión: 1
Urgencia: 3
Esfuerzo: 4

LT: listar tarea
FT: Filtrar tarea

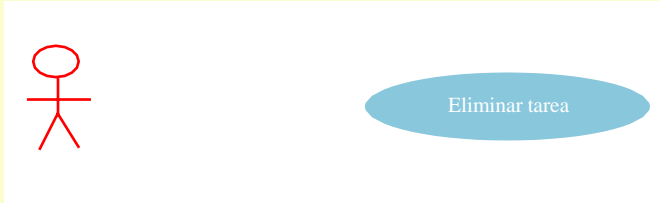
3. Listar Tarea

Flujo: Listar tarea

Prueba: escribir palabras claves, generar una lista.

a. Listar tarea.

Flujo: FT, LT



Versión: 1

Urgencia: 3

Esfuerzo: 2

ST: Selección de tarea

CE: Confirmar eliminación

ET: eliminar tarea del sistema

4. Eliminar tarea

Flujo: eliminar una tarea

Prueba: Seleccionar tarea, eliminar tarea, confirmar eliminación de tarea.

b. Eliminar tarea

Flujo: ST, ET, CE



Versión: 1

Urgencia: 5

Esfuerzo: 5

AT: Asignar tarea

SU: Seleccionar Usuarios

CS: Confirmar Selección

6. Asignar Tarea

Flujo: Asignación de una Tarea

Prueba: Seleccionar tarea a asignar, seleccionar usuario a asignar, tarea asignada, Confirmar selección de usuarios.

d. Asignar Tarea

Flujo: AT, SU, CS



Versión: 1

Urgencia: 2

Esfuerzo: 5

PI: Proceso de integración

CE: Conexión con app externa

SD: Sincronización de datos

5. Integración con apps externa

Flujo: Integrar app externas

Prueba: Inicia proceso de integración, configurar la conexión con las app, sincronizar los datos.

c. Integración con app ecterna

Flujo: PI, CE, SD



Versión: 1
Urgencia: 1
Esfuerzo: 3

ST: Selección de tarea
EC: Escribir comentario
AC: Agregar comentario

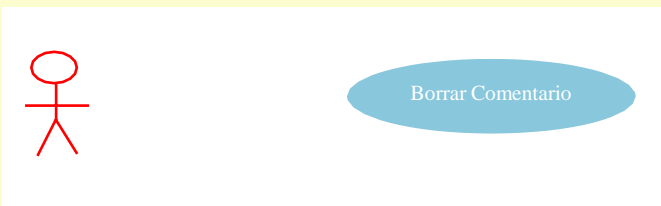
7. Comentar tarea

Flujo: Comentar una tarea

Prueba: Selección de tarea a comentar, usuario escribe el comentario, Agregar comentario a la tarea

e. Comentar tarea

Flujo: ST, EC, AC



Versión: 1
Urgencia: 2
Esfuerzo: 3

SC: selección de comentario
EC: Eliminar Comentario
CE: Confirmación de eliminación

8. Borrar Comentario

Flujo: Borrar un comentario

Prueba: seleccionar menú contextual, eliminar comentario de la tarea, confirmar eliminación de tarea.

f. Borrar comentario

Flujo: SC, EC, CE



Versión: 1
Urgencia: 1
Esfuerzo: 3

SC: Selección de comentario
RC: Respuesta comentario
AR: Agregar respuesta

9. Responder Comentario

Flujo: Responder un Comentario

Prueba: Selecciona el comentario a responder, escribe la respuesta al comentario, agrega la respuesta

g. Responder comentario

Flujo: SC, RC, AR



Versión: 1
Urgencia: 1
Esfuerzo: 3

SC: Selección de comentario
MC: Modificar Comentario
AC: Actualizar comentario

10. Editar comentario

Flujo: Editar un comentario

Prueba: Selección de comentario a editar, hacer modificación del comentario, actualizar comentario en la tarea

h. Editar Comentario

Flujo: SC, MC, AC

Descripción detallada de cada caso de uso:

| Caso N° - 1 Crear tarea | | |
|-------------------------|---|---|
| ID | CU-1 | |
| Nombre | Crear tarea | |
| Actores | Actores | |
| Objetivo | Este caso debe permitir crear una tarea | |
| Urgencia | 5 | |
| Esfuerzo | 4 | |
| Pre-condiciones | <ul style="list-style-type: none"> - El docente debe estar autenticado de forma correcta en el sistema. - El docente debe tener permisos para crear tareas. | |
| Flujo Normal | Docente | Sistema |
| | Selecciona crear una tarea | |
| | | Se despliega el formulario para crear una tarea |
| | Título de la tarea, Descripción de la tarea, fecha de inicio, fecha de cierre, Prioridad: Alta - Baja, Estado (En Progreso, en revisión, Hecho) | |
| | Guardar | |

| | | |
|----------------------------|--|--|
| | | Se crea la tarea |
| | Cerrar | |
| | | Se cierra el formulario |
| Flujo alternativo 1 | Si algún campo requerido no se completó y se selecciona Guardar. | Muestra mensaje de error y no permite crear tarea. |
| | | |

| Caso N° 2 - Edición de una tarea | | |
|---|---|---|
| ID | CU-2 | |
| Nombre | Edición de una tarea | |
| Actores | Actores | |
| Objetivo | Este caso debe permitir editar una tarea | |
| Urgencia | 5 | |
| Esfuerzo | 4 | |
| Pre-condiciones | <ul style="list-style-type: none"> - El docente debe estar autenticado de forma correcta en el sistema. - Debe existir una tarea previamente creada | |
| Flujo Normal | Docente | Sistema |
| | Selecciona la tarea a editar | |
| | | Despliega los detalles de la tarea a editar |
| | Título, Descripción, Asignar a un estudiantes, Fecha de inicio, Fecha de cierre, Prioridad: Alta - baja, Estado: (En Progreso, en revisión, Hecho) | |
| | Guardar los cambios | |
| | | Se modifica la tarea |
| | Eliminar | |
| | | Se eliminan las modificaciones |

| | | |
|----------------------------|-------------------------|-------|
| Flujo alternativo 1 | No se hizo modificación | Error |
| | | |

| | | |
|--------------------------------|--|--|
| Caso N°3 - Listar tarea | | |
| ID | CU-3 | |
| Nombre | Listar tarea | |
| Actores | Actores | |
| Objetivo | Este caso debe permitir listar todas las tareas | |
| Urgencia | 3 | |
| Esfuerzo | 4 | |
| Pre-condiciones | <ul style="list-style-type: none"> - El docente debe estar autenticado de forma correcta en el sistema. - Debe tener tareas para poder listar. | |
| Flujo Normal | Docente | Sistema |
| | Listar tareas | |
| | | Se despliega la lista de tareas |
| | Tarea, Fecha de vencimiento de la tarea. | |
| | Editar tarea | |
| | | Despliega el formulario para editar tarea |
| | Eliminar tarea | |
| | | Se elimina la tarea |
| Flujo alternativo 1 | Si el docente no tiene permisos para acceder al listado de tareas, muestra un mensaje de error | Si hay problemas técnicos para recuperar el listado de tareas, muestra un mensaje de error |

| | | |
|--|--|--|
| | | |
|--|--|--|

| | | |
|----------------------------------|--|---|
| Caso N°4 - Eliminar tarea | | |
| ID | CU-4 | |
| Nombre | Eliminar tarea | |
| Actores | Actores | |
| Objetivo | Permitir eliminar una tarea | |
| Urgencia | 3 | |
| Esfuerzo | 2 | |
| Pre-condiciones | <ul style="list-style-type: none"> - El actor debe estar autenticado en el sistema. - Deben existir tareas previamente creadas y visibles para el actor. | |
| Flujo Normal | Docente | Sistema |
| | Selecciona la opción Eliminar tarea. | |
| | | El sistema despliega la lista de tareas existentes. |
| | Selecciona la tarea que desea eliminar. | |
| | | tarea seleccionada |
| | eliminar | |
| | | El sistema muestra la tarea seleccionada con la opción de eliminar. |
| | Confirma la eliminación de la tarea. | |
| | | El sistema elimina la tarea seleccionada. |
| | | El sistema muestra un mensaje de confirmación indicando que la tarea ha sido eliminada. |
| Flujo alternativo 1 | | |
| | | |

| Caso N°5 - Asignar una tarea | | |
|------------------------------|--|--|
| ID | CU-5 | |
| Nombre | Asignar una tarea | |
| Actores | Actores | |
| Objetivo | Permitir asignar una tarea | |
| Urgencia | 5 | |
| Esfuerzo | 5 | |
| Pre-condiciones | <ul style="list-style-type: none"> - El docente debe estar autenticado en el sistema. - Deben existir tareas disponibles para asignar. - Deben existir actores a quienes se les puede asignar la tarea. | |
| Flujo Normal | Docente | Sistema |
| | Selecciona la opción Asignar tarea en el sistema. | |
| | | El sistema despliega la interfaz de asignación de tareas. |
| | Selecciona la tarea que desea asignar. | |
| | | El sistema muestra una lista de actores disponibles para asignar la tarea. |
| | Selecciona el actor al que se asignará la tarea. | |
| | | El sistema confirma la asignación de la tarea al actor seleccionado |
| Flujo alternativo 1 | | |
| | | |

| Caso N°6 - Integración con apps externas | |
|--|-------------------------------|
| ID | CU-6 |
| Nombre | Integración con apps externas |

| | | |
|----------------------------|---|---|
| Actores | Actores | |
| Objetivo | Permitir integrar Apps externas | |
| Urgencia | 2 | |
| Esfuerzo | 5 | |
| Pre-condiciones | <ul style="list-style-type: none"> - El usuario debe estar autenticado en el sistema. - Debe existir una API disponible para la integración. - El usuario debe tener permisos para configurar integraciones. | |
| Flujo Normal | Docente | Sistema |
| | Accede a la opción Integraciones en el sistema. | |
| | | El sistema despliega una lista de apps externas disponibles para integrar. |
| | Selecciona la app externa que desea integrar. | |
| | | El sistema solicita las credenciales y permisos necesarios para la integración. |
| | Introduce las credenciales y otorga los permisos necesarios. | |
| | | El sistema verifica las credenciales y los permisos. |
| | | El sistema confirma la integración exitosa de la app externa y muestra un mensaje de éxito. |
| Flujo alternativo 1 | | |

| | |
|---------------------------------------|-----------------------------|
| Caso N° 7 - Comentar una tarea | |
| ID | CU-7 |
| Nombre | Comentar una tarea |
| Actores | Actores |
| Objetivo | Permitir comentar una tarea |
| Urgencia | 1 |

| | | |
|----------------------------|--|--|
| Esfuerzo | 3 | |
| Pre-condiciones | El usuario debe estar autenticado en el sistema. La tarea a comentar debe existir en el sistema. | |
| Flujo Normal | Docente | Sistema |
| | El usuario selecciona la tarea a comentar. | |
| | El usuario escribe el comentario en el campo correspondiente. | |
| | | El sistema agrega el comentario a la tarea seleccionada. |
| | | El comentario se agrega correctamente a la tarea seleccionada. |
| Flujo alternativo 1 | | |
| | | |

| | |
|--------------------------------------|-------------------------------|
| Caso N° 8 - Borrar Comentario | |
| ID | CU-8 |
| Nombre | Borrar Comentario |
| Actores | Actores |
| Objetivo | Permitir borrar un comentario |
| Urgencia | 2 |
| Esfuerzo | 3 |

| | | |
|----------------------------|--|--|
| Pre-condiciones | El usuario debe estar autenticado en el sistema. El comentario a borrar debe pertenecer a una tarea existente en el sistema. | |
| Flujo Normal | Docente | Sistema |
| | El usuario selecciona el comentario que desea borrar. | |
| | El usuario confirma la eliminación del comentario. | |
| | | El comentario seleccionado se elimina correctamente de la tarea. |
| Flujo alternativo 1 | | |
| | | |

| Caso N° 9 - Responder comentario de una tarea | | |
|--|--|----------------|
| ID | CU-9 | |
| Nombre | Responder comentario de una tarea | |
| Actores | Actores | |
| Objetivo | Permitir responder un comentario | |
| Urgencia | 1 | |
| Esfuerzo | 3 | |
| Pre-condiciones | El usuario debe estar autenticado en el sistema. El comentario al que se va a responder debe existir en la tarea seleccionada. | |
| Flujo Normal | Docente | Sistema |
| | El usuario selecciona el comentario al que desea responder | |
| | El usuario escribe la respuesta en el campo correspondiente. | |

| | | |
|----------------------------|--|---|
| | | La respuesta se agrega correctamente al comentario seleccionado |
| | | |
| Flujo alternativo 1 | | |
| | | |

| Caso N° 10 - Editar comentario de tarea | | |
|--|--|--|
| ID | CU-10 | |
| Nombre | Editar comentario de tarea | |
| Actores | Actores | |
| Objetivo | Permitir editar el comentario de una tarea antrientemente creada | |
| Urgencia | 1 | |
| Esfuerzo | 3 | |
| Pre-condiciones | El usuario debe estar autenticado en el sistema. El comentario a editar debe existir en la tarea seleccionada. | |
| Flujo Normal | Docente | Sistema |
| | El usuario selecciona el comentario que desea editar. | |
| | | El sistema actualiza el comentario con los cambios realizados. |
| | El usuario modifica el contenido del comentario en el campo correspondiente. | |
| | | El comentario seleccionado se actualiza correctamente con los |

| | | |
|----------------------------|--|------------------------------------|
| | | cambios realizados por el usuario. |
| Flujo alternativo 1 | | |
| | | |

Prioridad de Requerimientos

A partir del análisis de requerimientos, funcionalidades y el proceso de design thinking, se concreta la siguiente matriz de prioridad de requerimientos.

Para la interpretación se tiene en cuenta la siguiente escala con sus valores.

Eje de Urgencia:

- Obligatoria (5)
- Alta (4)
- Moderada (3)
- Menor (2)
- Baja (1)

Eje de Esfuerzo:

- Muy alto (5)
- Alto (4)
- Medio (3)
- Bajo (2)
- Muy bajo (1)

| Esfuerzo | Urgencia | | | | | |
|----------|------------|-----------------------|--------------|------------|--------|---------------|
| | | 1-Baja | 2-Menor | 3-Moderada | 4-Alta | 5-Obligatorio |
| | 5-Muy alto | 5 | 10 | 15 | 20 | 25 |
| | | | CU-6 | | | CU-5 |
| | 4-Alto | 4 | 8 | 12 | 16 | 20 |
| | | | | CU-3 | | CU-1 CU-2 |
| | 3-Medio | 3 | 6 | 9 | 12 | 15 |
| | | CU-7 CU-9 CU-10 | CU-4 CU-8 | | | |
| | 2-Bajo | 2 | 4 | 6 | 8 | 10 |
| | | | | | | |
| | 1-Muy bajo | 1 | 2 | 3 | 4 | 5 |
| | | | | | | |

Requisitos No Funcionales

Requisitos de Desempeño

Requisitos de Seguridad

Requisitos de Usabilidad

Requisitos de Escalabilidad

Modelado E/R

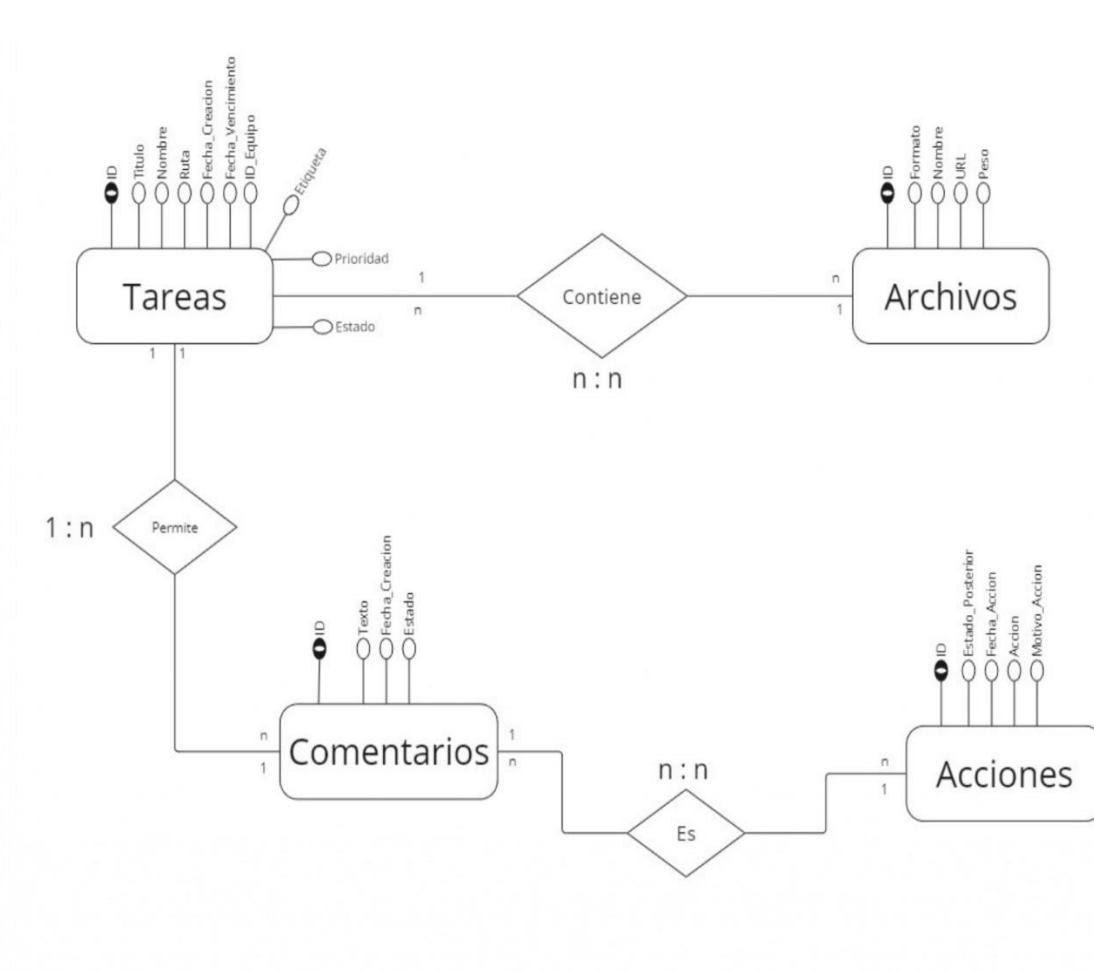
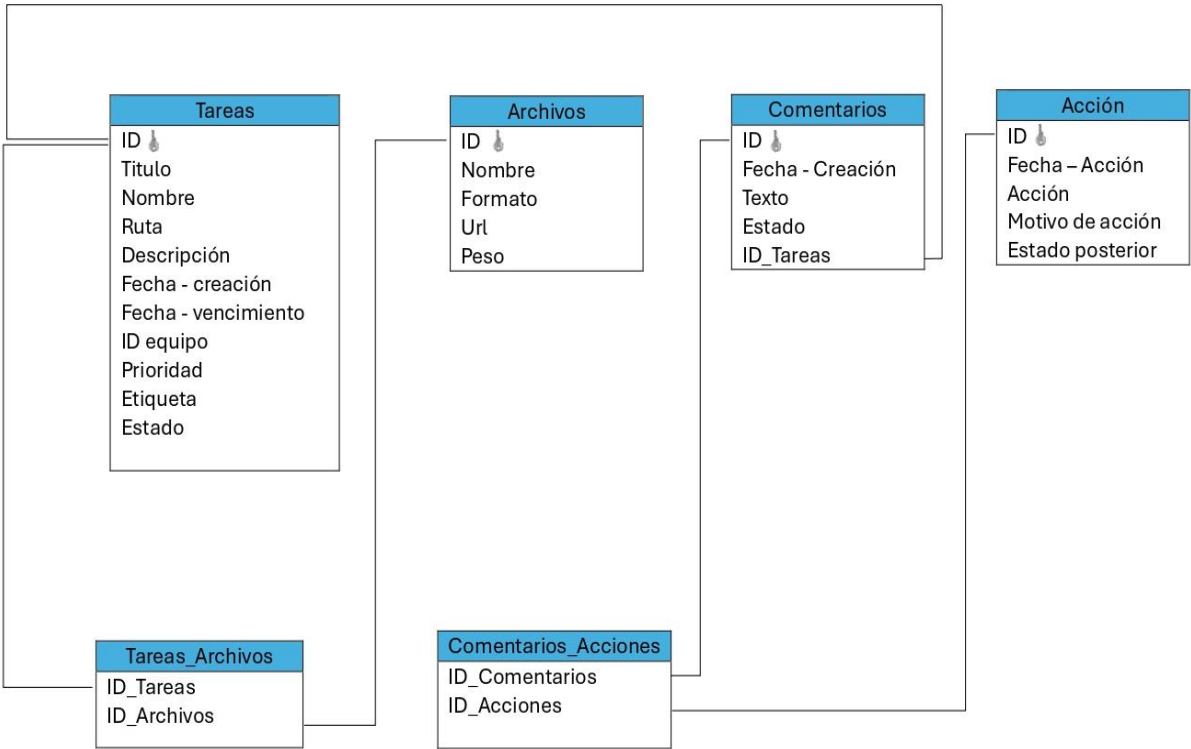


Diagrama Relacional



Script de modelo relacional

Tareas {

```
ID: number,  
Titulo: string,  
Nombre: string,  
Ruta: string,  
Descripción: string,  
Fecha – creación: timesdate,  
Fecha – vencimiento:  
timesdate,  
IDequipo: number,  
Prioridad: string,  
Etiqueta: string,  
Estado: string,  
ID_Archivo: array,  
}
```

Archivos{

```
ID: number,  
Nombre: string,  
Formato: string,  
Url: string,  
Peso: number,  
ID_tareas:array,  
}
```

Comentarios {

```
ID: number,  
Fecha-creacion: timesdate,  
Texto: string,  
Estado: string,  
ID_tareas: array,  
}
```

Acción {

```
ID: number,  
Fecha-accion: timesdate ,  
Acción: string ,  
Motivo de acción: string ,  
Estado posterios: string,  
}
```

Descripción de Entidades y Relaciones

Reglas de Integridad

Anexos (si es necesario)

Diagramas Adicionales

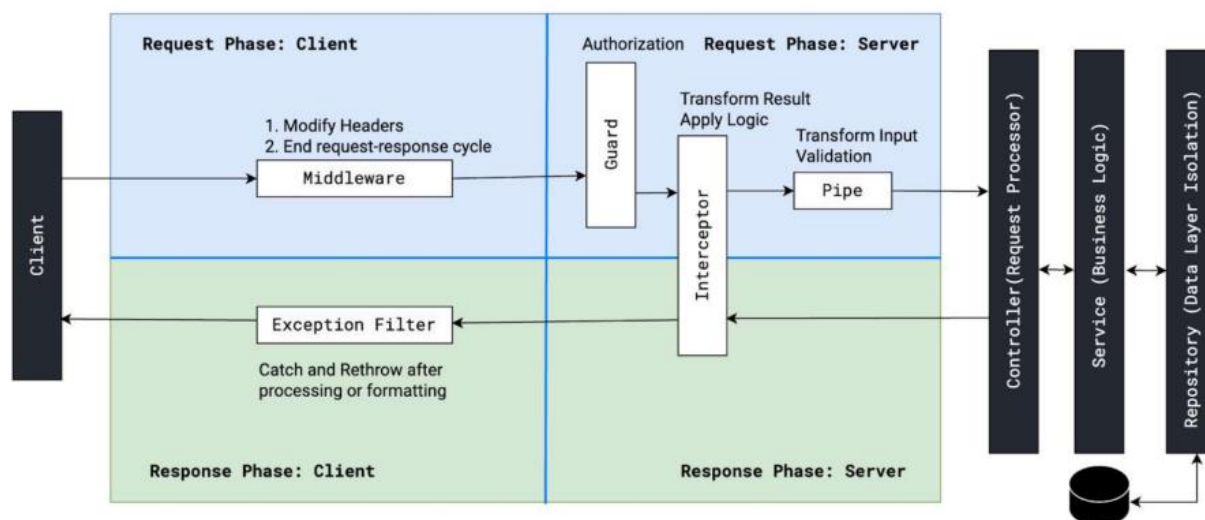
Referencias

Etapa 2: Persistencia de Datos con Backend

Descripción de la arquitectura propuesta

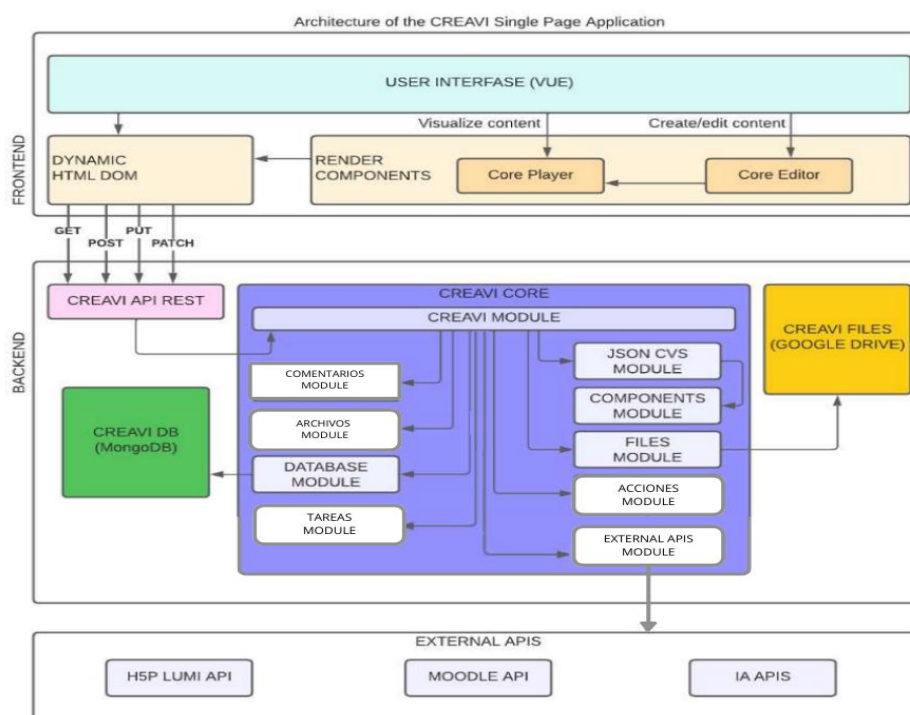
Este proyecto está basado en un sistema de Nest.js que cuenta con múltiples elementos, incluyendo el cliente, el interceptor que proporciona una validación al controlador, y una interacción fluida entre el controlador, el servidor y el repositorio, así como con la base de datos. El diseño de la arquitectura permite que cada capa del sistema se comunique de manera eficiente, asegurando que los datos se procesen correctamente antes de ser almacenados o enviados al usuario.

Diseño de la Arquitectura de Backend



Todo este proceso asegura que las solicitudes del cliente sean manejadas de manera ordenada y segura, facilitando la comunicación entre el cliente y el servidor. Cada componente tiene su rol específico para garantizar un funcionamiento eficiente y seguro de la aplicación.

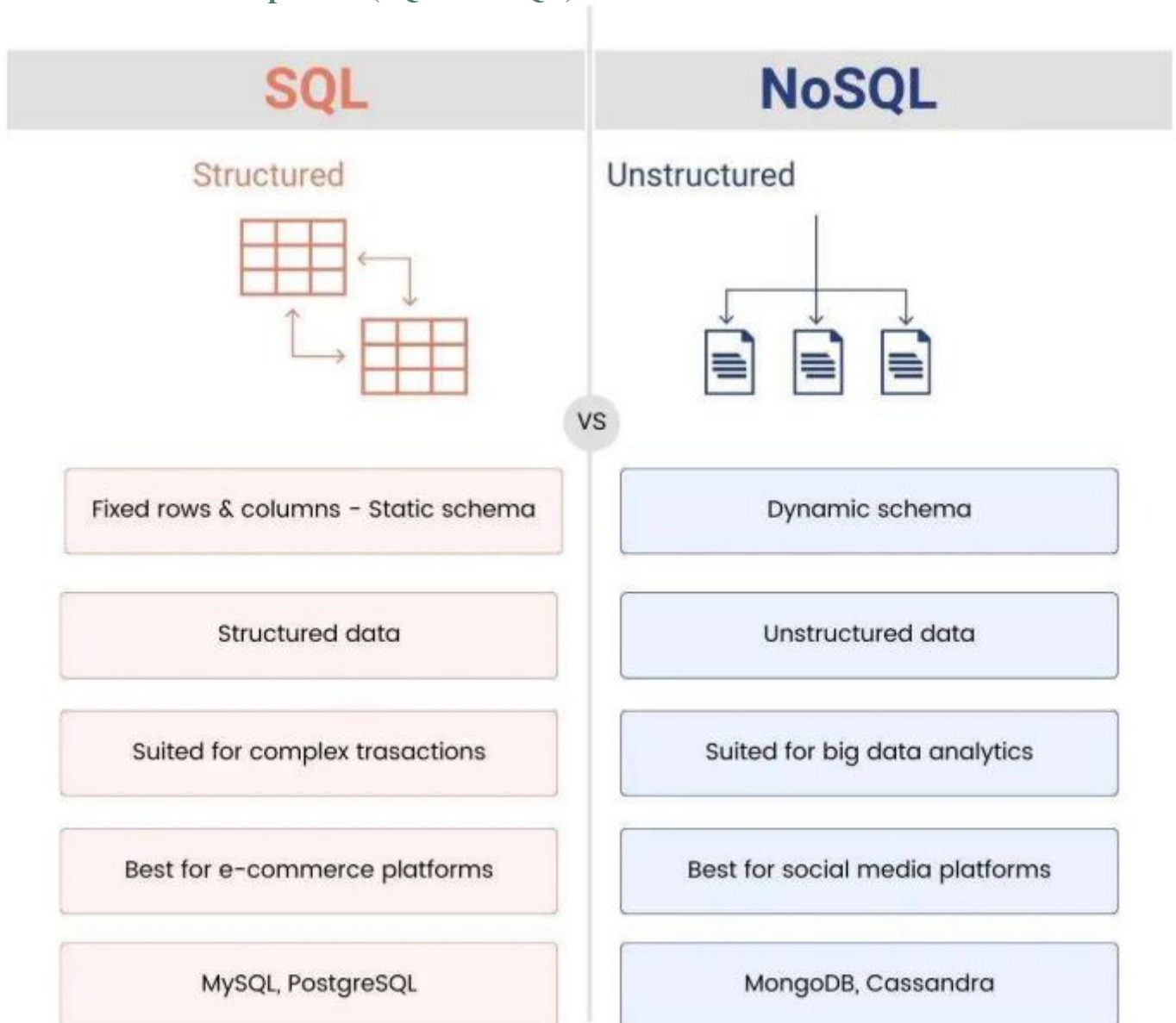
Diagramas de Arquitectura



Elección de la Base de Datos

Una de las decisiones más importantes para el desarrollo del backend de un proyecto es la elección de la base de datos. Entre las opciones más populares se encuentran SQL y NoSQL, y la selección dependerá de las necesidades específicas del proyecto.

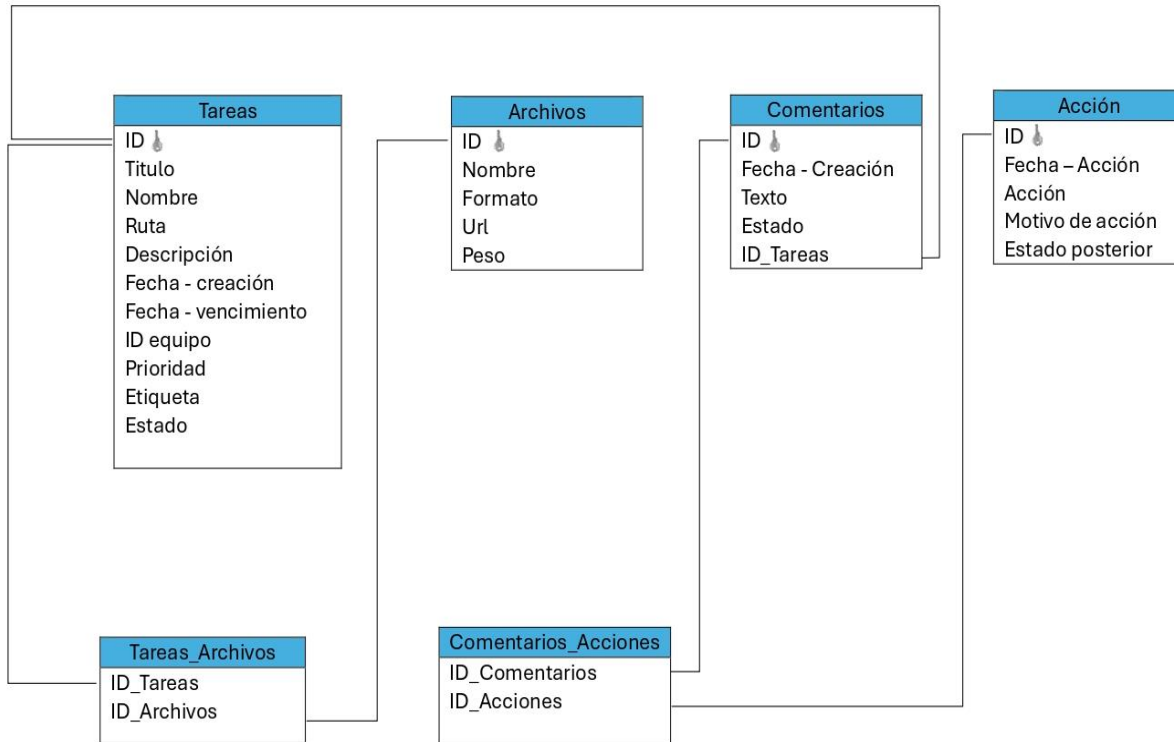
Evaluación de Opciones (SQL o NoSQL)



Justificación de la Elección

Para el desarrollo de este gestor de tareas se utilizará SQL debido a su capacidad para manejar una estructura más organizada. En este caso, las colecciones como tareas, archivos, acciones y comentarios requieren mantener unas conexiones claras entre ellas. SQL permite definir y gestionar estas relaciones con facilidad, asegurando consistencia en las operaciones del sistema, además de facilitar consultas complejas para el seguimiento y organización de la información.

Diseño de esquemas de bases de datos



El componente gestor de tareas organiza toda su información en cuatro colecciones principales: Tareas, que contiene los datos básicos de una tarea como título, descripción, fechas, prioridad, estado y etiquetas; Comentarios, donde se almacenan las opiniones o anotaciones vinculadas a cada tarea, con su texto, estado y fecha de creación del comentario; Archivos, que son los documentos adjuntos a las tareas, incluyendo su nombre, formato, URL, peso y Acciones, que registran los eventos realizados en las tareas, como creación, edición o eliminación, con detalles sobre la fecha, tipo de acción, motivo y estado posterior. Estas son las colecciones con las que cuenta el gestor de tareas y van a permitir que el usuario se le pueda garantizar una buena experiencia. Seguidamente, También encontramos una relación entre Tareas y Archivos, donde cada tarea puede tener múltiples archivos asociados que contienen documentos o recursos relacionados con la actividad, como imágenes, PDF o documentos de texto. También, existe una relación entre Comentarios y Acciones, ya que cada comentario puede generar una o varias acciones registradas en el sistema, como agregar, editar o eliminar, permitiendo mantener una vista clara sobre lo que se realiza dentro del gestor de tareas

Implementación del Backend

Elección del lenguaje de programación

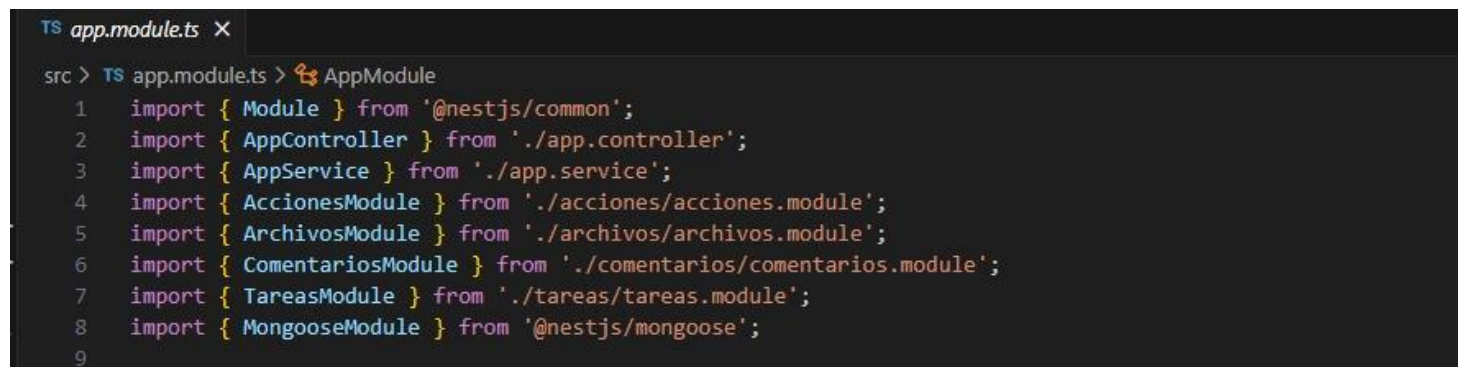
Para el desarrollo del backend se eligió TypeScript como lenguaje de programación, aprovechando la estructura y la robustez que ofrece en proyectos complejos. Esta elección está fundamentada en la capacidad de TypeScript para proporcionar tipado estático y un desarrollo más estructurado y escalable en aplicaciones de gran envergadura. TypeScript es compatible con Nestjs, el framework de Node.js seleccionado, que permite la construcción de aplicaciones modulares, fácilmente testeables y mantenibles. NestJS se destaca en el desarrollo de Apis rest y aplicaciones orientadas a microservicios, lo que facilita la integración y escalabilidad del sistema.

Conexión a la Base de Datos

La conexión a la base de datos MongoDB Atlas desde el componente se realiza mediante las credenciales de acceso, la URL de conexión, y una configuración específica para asegurar el sistema.

Al trabajar con MongoDB, es necesario instalar las dependencias correspondientes ejecutando:

```
npm install @nestjs/mongoose mongoose.
```

A screenshot of a code editor with a dark theme. The file name 'app.module.ts' is visible in the top left. The code shows imports for various modules: '@nestjs/common', './app.controller', './app.service', './acciones/acciones.module', './archivos/archivos.module', './comentarios/comentarios.module', './tareas/tareas.module', and '@nestjs/mongoose'.

```
TS app.module.ts X
src > TS app.module.ts > AppModule
1 import { Module } from '@nestjs/common';
2 import { AppController } from './app.controller';
3 import { AppService } from './app.service';
4 import { AccionesModule } from './acciones/acciones.module';
5 import { ArchivosModule } from './archivos/archivos.module';
6 import { ComentariosModule } from './comentarios/comentarios.module';
7 import { TareasModule } from './tareas/tareas.module';
8 import { MongooseModule } from '@nestjs/mongoose';
9
```

Luego, se realizó la configuración en el App Module. Aquí, el módulo MongooseModule se configuró para conectarse a la base de datos MongoDB usando la URI proporcionada, la cual incluye tanto el usuario y la contraseña como el nombre de la base de datos TaskManager_db. Esto permite el acceso seguro a la base de datos alojada en la nube.

```

9
10 @Module({
11   imports: [
12     MongooseModule.forRoot(
13       'mongodb://localhost:27020/just-a-mongo-db-net?authSource=admin&ssl=false',
14     ),

```

Por último, en el módulo principal (AppModule), se utiliza `MongooseModule.forRoot()` para establecer la conexión con MongoDB, especificando directamente la URI de conexión en el código.

Desarrollo de Operaciones CRUD

Las operaciones CRUD (Crear, Obtener todo, Obtener 1, Actualizar, Eliminar) son fundamentales para el funcionamiento óptimo del componente y permiten la interacción con las colecciones en MongoDB. A continuación, se detallan las implementaciones de estas operaciones para la colección de Acciones.

```

8 @Injectable()
9 export class AccionesService {
10   constructor(@InjectModel(acciones.name) private ACCIONESModel: Model<acciones>) {}
11   async create(createAccioneDto: CreateAccioneDto) {
12     const createAccione = new this.ACCIONESModel (createAccioneDto)
13     const result = await createAccione.save();
14     return result;
15   }
16   findAll() {
17     return this.ACCIONESModel.find();
18   }
19
20   findOne(id: string) {
21     return this.ACCIONESModel
22       .findById(id);
23   }
24
25   async update(id: string, updateAccioneDto: UpdateAccioneDto) {
26     try {
27       const updatedAcciones = await this.ACCIONESModel.findByIdAndUpdate(
28         id,
29         updateAccioneDto,
30         { new:true } );
31
32       return updatedAcciones;
33     }
34     catch (e) {
35       console.error(e)
36     }
37     finally{
38       console.log('actualización finalizada.');
```

Cada una de estas funciones implementa una operación CRUD específica:

- **create:** Inserta un nuevo documento en la colección de Acciones utilizando un DTO (Data Transfer Object) para manejar los datos de entrada.
- **findAll:** Devuelve un conjunto de documentos predefinidos que simulan el listado completo de acciones.
- **findOne:** Devuelve un documento específico según el id proporcionado.
- **update:** Actualiza los campos de un documento existente con los valores proporcionados en el `updateAccioneDto`.
- **remove:** Elimina un documento basado en el id especificado

API DOCUMENTACIÓN TASK MANAGER.

Este código implementa una aplicación NestJS configurada para utilizar Swagger como herramienta de documentación interactiva para una API. Comienza importando los módulos esenciales: NestFactory para inicializar la aplicación, AppModule como módulo principal y los componentes de @nestjs/swagger para la configuración de Swagger.

En la función bootstrap(), se inicializa la aplicación mediante NestFactory.create(AppModule). A continuación, se configura la documentación con DocumentBuilder, definiendo título, descripción y versión de la API. Luego, se genera el documento Swagger con SwaggerModule.createDocument, que combina la configuración creada y la instancia de la aplicación. Finalmente, se expone la documentación Swagger en la ruta /api mediante SwaggerModule.setup y se inicia el servidor escuchando en el puerto especificado (process.env.PORT o 3000 por defecto) con app.listen. Este enfoque garantiza una API documentada y lista para su consumo.

```
1 import { NestFactory } from '@nestjs/core';
2 import { AppModule } from './app.module';
3 import { DocumentBuilder, SwaggerModule } from '@nestjs/swagger';
4
5 async function bootstrap() {
6   const app = await NestFactory.create(AppModule);
7
8   const config = new DocumentBuilder()
9     .setTitle('API Documentation Task Manager')
10    .setDescription('Api description')
11    .setVersion('1.0')
12    .build();
13
14   const documentFactory = () => SwaggerModule.createDocument(app, config);
15   SwaggerModule.setup('api', app, documentFactory);
16
17   await app.listen(process.env.PORT ?? 3000);
18 }
19 bootstrap();
20 |
```

API Documentation Task Manager ^{1.0} OAS 3.0

Api description

App ^

| | | |
|-----|---|---|
| GET | / | ▼ |
|-----|---|---|

Acciones ^

| | | |
|--------|----------------|---|
| POST | /acciones | ▼ |
| GET | /acciones | ▼ |
| GET | /acciones/{id} | ▼ |
| PATCH | /acciones/{id} | ▼ |
| DELETE | /acciones/{id} | ▼ |

Archivos ^

| | | |
|---------------|-----------|--------|
| POST | /archivos | ▼ |
| GET | /archivos | ^ |
| Parameters | | Cancel |
| No parameters | | |

Etapas 3: Consumo de Datos y Desarrollo Frontend

Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

Creación de la Interfaz de Usuario (UI)

Diseño de la Interfaz de Usuario (UI) con HTML y CSS

Consideraciones de Usabilidad

Maquetación Responsiva

Programación Frontend con JavaScript (JS)

Desarrollo de la Lógica del Frontend

Manejo de Eventos y Comportamientos Dinámicos

Uso de Bibliotecas y Frameworks (si aplicable)

Consumo de Datos desde el Backend

Configuración de Conexiones al Backend

Obtención y Presentación de Datos

Actualización en Tiempo Real (si aplicable)

Interacción Usuario-Interfaz

Manejo de Formularios y Validación de Datos

Implementación de Funcionalidades Interactivas

Mejoras en la Experiencia del Usuario

Pruebas y Depuración del Frontend

Diseño de Casos de Prueba de Frontend

Pruebas de Usabilidad

Depuración de Errores y Optimización del Código

Implementación de la Lógica de Negocio en el Frontend

Migración de la Lógica de Negocio desde el Backend (si necesario)

Validación de Datos y Reglas de Negocio en el Frontend

Integración con el Backend

Verificación de la Comunicación Efectiva con el Backend

Pruebas de Integración Frontend-Backend