



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

---

Componente Gestor de Tareas

Pier Paolo Chima Durango – [Pchimadurango57@correo.unicordoba.edu.co](mailto:Pchimadurango57@correo.unicordoba.edu.co)

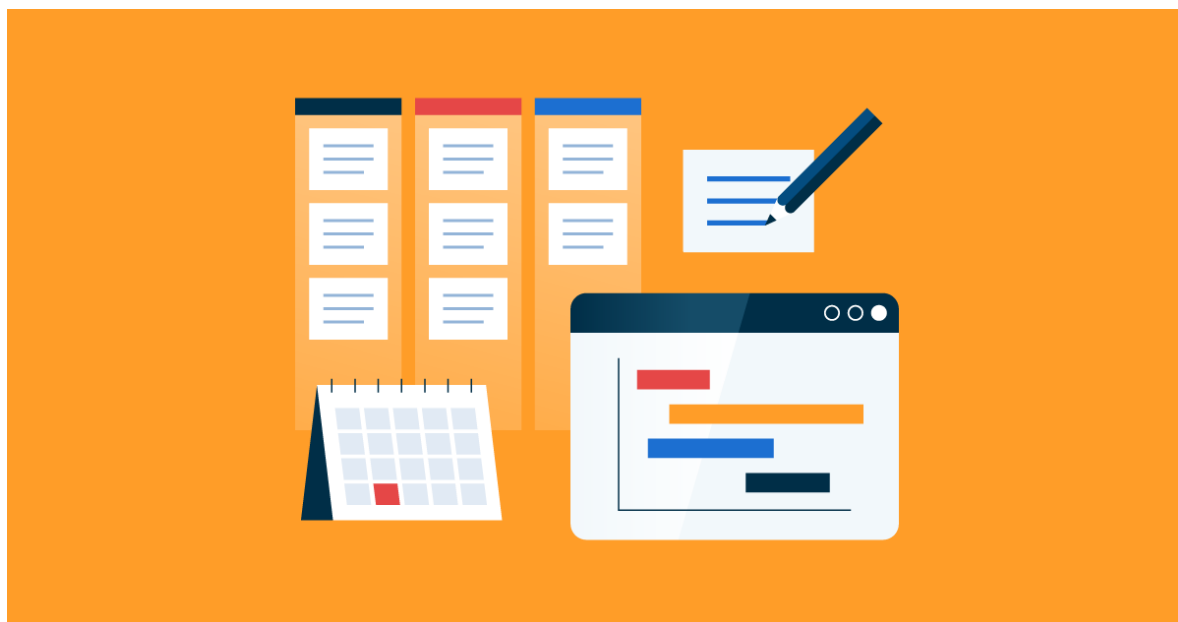
Antonio José Gómez Banda - [Agomezbanda30@correo.unicordoba.edu.co](mailto:Agomezbanda30@correo.unicordoba.edu.co)

Camilo José Aviléz Guzmán - [Cavilezguzman24@correo.unicordoba.edu.co](mailto:Cavilezguzman24@correo.unicordoba.edu.co)

Rosa Elvira Herrera Peña - [rherrerapena99@correo.unicordoba.edu.co](mailto:rherrerapena99@correo.unicordoba.edu.co)

Sebastián López Sotelo - [slopezsotelo19@correo.unicordoba.edu.co](mailto: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.

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

<b>ETAPA 2: PERSISTENCIA DE DATOS CON BACKEND .....</b>	<b>22</b>
<b>INTRODUCCIÓN .....</b>	<b>22</b>
PROPÓSITO DE LA ETAPA .....	22
ALCANCE DE LA ETAPA .....	22
DEFINICIONES Y ACRÓNIMOS .....	22
<b>DISEÑO DE LA ARQUITECTURA DE BACKEND.....</b>	<b>22</b>
DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA .....	22
COMPONENTES DEL BACKEND.....	22
DIAGRAMAS DE ARQUITECTURA .....	22
<b>ELECCIÓN DE LA BASE DE DATOS .....</b>	<b>22</b>
EVALUACIÓN DE OPCIONES (SQL o NoSQL).....	22
JUSTIFICACIÓN DE LA ELECCIÓN.....	22
DISEÑO DE ESQUEMA DE BASE DE DATOS .....	23
<b>IMPLEMENTACIÓN DEL BACKEND.....</b>	<b>23</b>
ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN.....	23
CREACIÓN DE LA LÓGICA DE NEGOCIO .....	23
DESARROLLO DE ENDPOINTS Y APIS.....	23
AUTENTICACIÓN Y AUTORIZACIÓN .....	23
<b>CONEXIÓN A LA BASE DE DATOS .....</b>	<b>23</b>
CONFIGURACIÓN DE LA CONEXIÓN .....	23
DESARROLLO DE OPERACIONES CRUD.....	23
MANEJO DE TRANSACCIONES .....	23
<b>PRUEBAS DEL BACKEND .....</b>	<b>23</b>
DISEÑO DE CASOS DE PRUEBA.....	23
EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN.....	24
MANEJO DE ERRORES Y EXCEPCIONES .....	24

<b>ETAPA 3: CONSUMO DE DATOS Y DESARROLLO FRONTEND .....</b>	<b>25</b>
<b>INTRODUCCIÓN .....</b>	<b>25</b>
PROPÓSITO DE LA ETAPA .....	25
ALCANCE DE LA ETAPA .....	25
DEFINICIONES Y ACRÓNIMOS .....	25
<b>CREACIÓN DE LA INTERFAZ DE USUARIO (UI) .....</b>	<b>25</b>
DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS .....	25
CONSIDERACIONES DE USABILIDAD .....	25
MAQUETACIÓN RESPONSIVA .....	25
<b>PROGRAMACIÓN FRONTEND CON JAVASCRIPT (JS).....</b>	<b>25</b>
DESARROLLO DE LA LÓGICA DEL FRONTEND .....	25
MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS .....	25
USO DE BIBLIOTECAS Y FRAMEWORKS (SI APLICABLE) .....	26
<b>CONSUMO DE DATOS DESDE EL BACKEND.....</b>	<b>26</b>
CONFIGURACIÓN DE CONEXIONES AL BACKEND .....	26
OBTENCIÓN Y PRESENTACIÓN DE DATOS .....	26
ACTUALIZACIÓN EN TIEMPO REAL (SI APLICABLE) .....	26
<b>INTERACCIÓN USUARIO-INTERFAZ .....</b>	<b>26</b>
MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS .....	26
IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS .....	26
MEJORAS EN LA EXPERIENCIA DEL USUARIO .....	26
<b>PRUEBAS Y DEPURACIÓN DEL FRONTEND .....</b>	<b>26</b>
DISEÑO DE CASOS DE PRUEBA DE FRONTEND .....	26
PRUEBAS DE USABILIDAD.....	26
DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO.....	27
<b>IMPLEMENTACIÓN DE LA LÓGICA DE NEGOCIO EN EL FRONTEND .....</b>	<b>27</b>
MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO) .....	27
VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND .....	27
<b>INTEGRACIÓN CON EL BACKEND .....</b>	<b>27</b>
VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND .....	27
PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND .....	27

# **Etapas 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: Asíncrono 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.

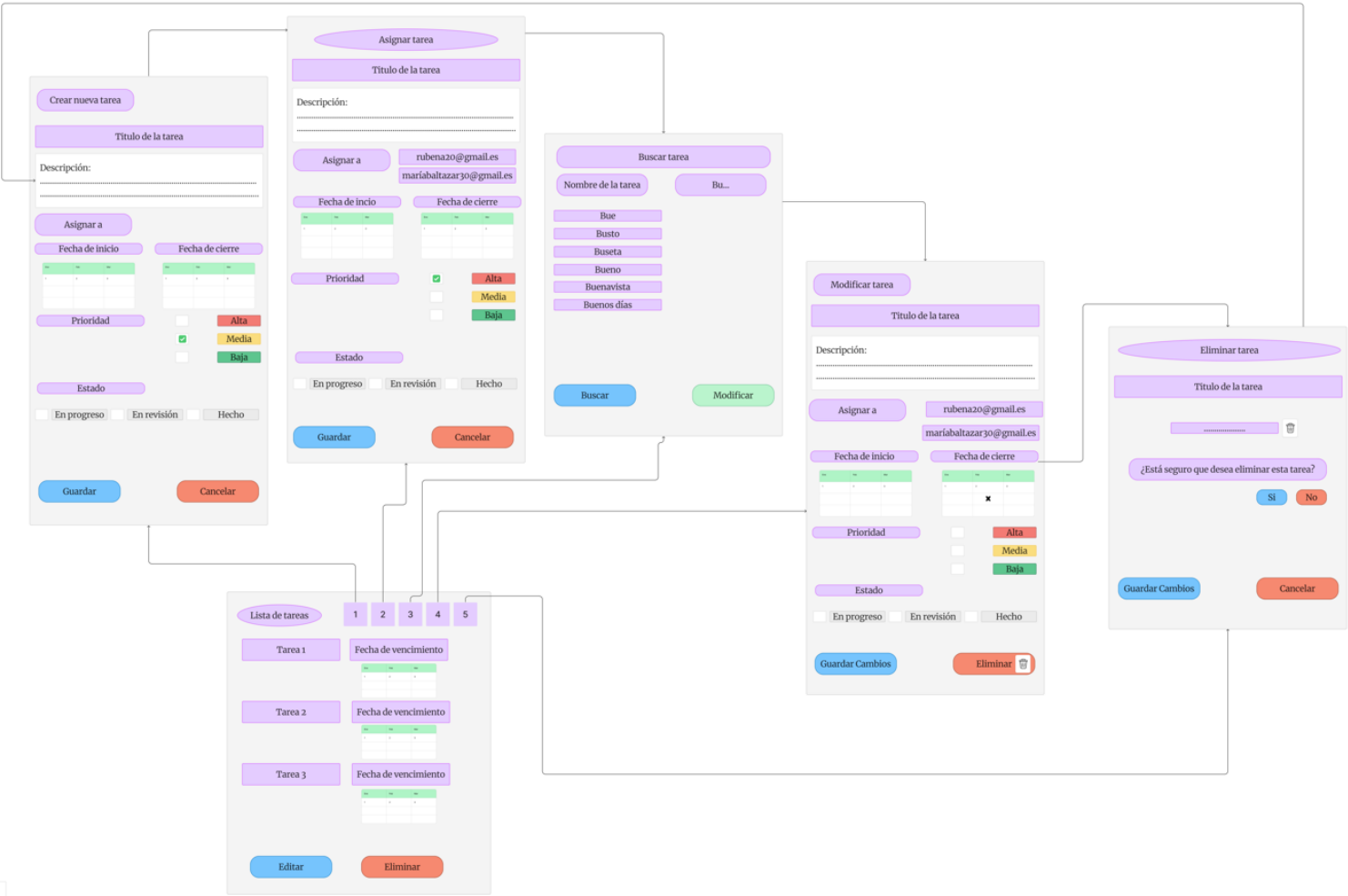
# Descripción General

## 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

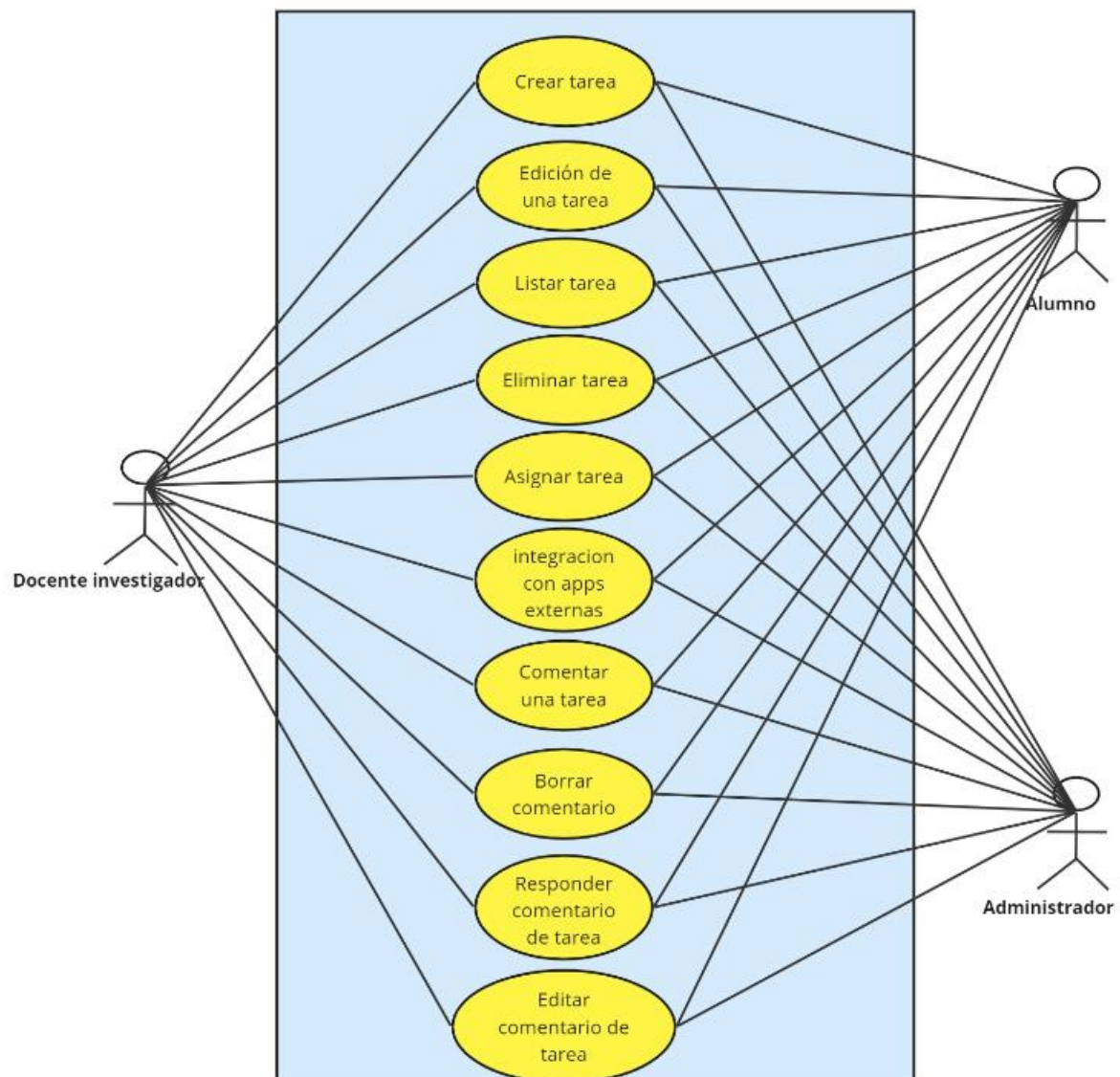


## Usuarios del sistema

## Restricciones

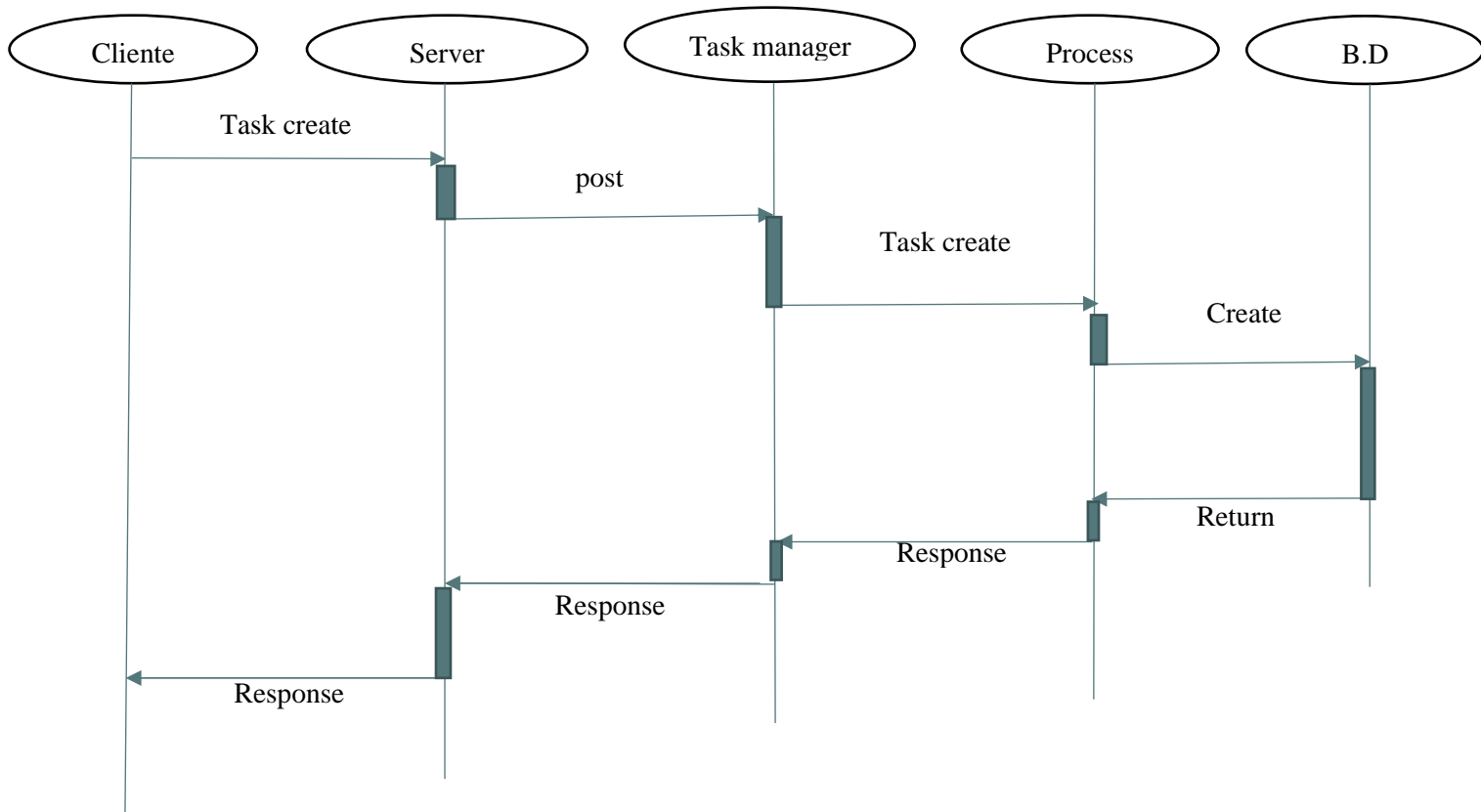
### 3. REQUISISTOS FUNCIONALES

#### Casos de Uso

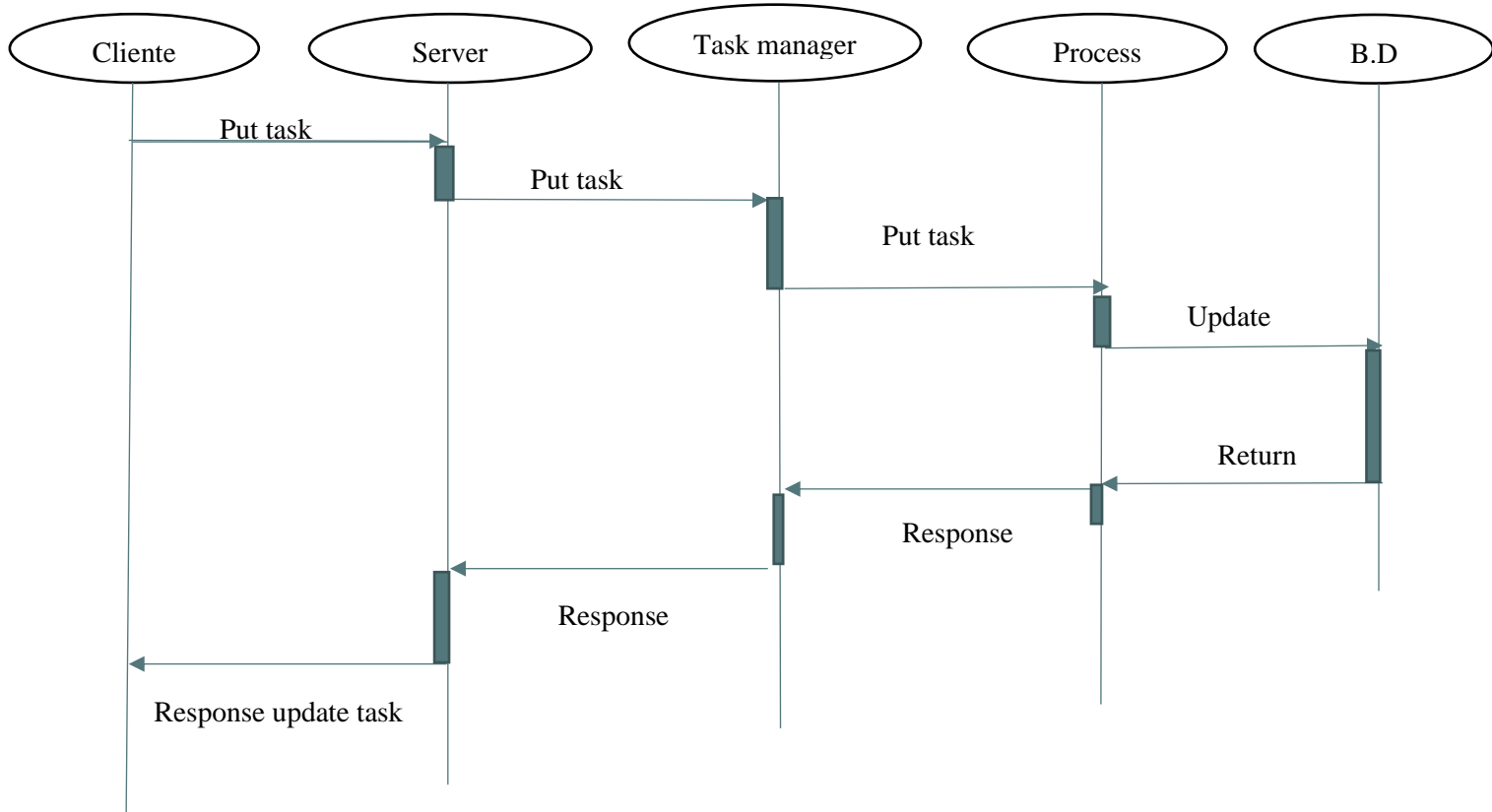


## Diagramas de Secuencia

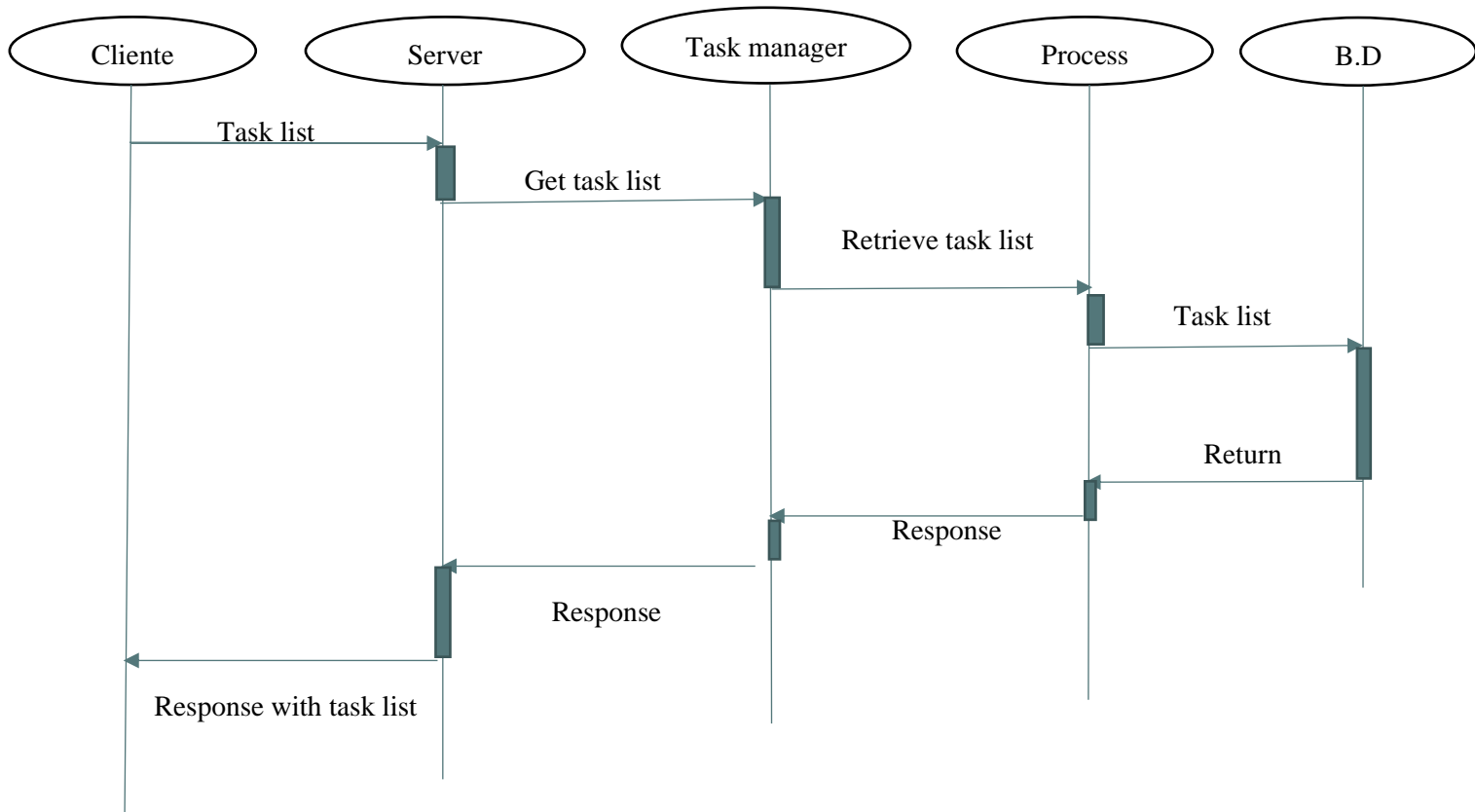
Crear tarea:



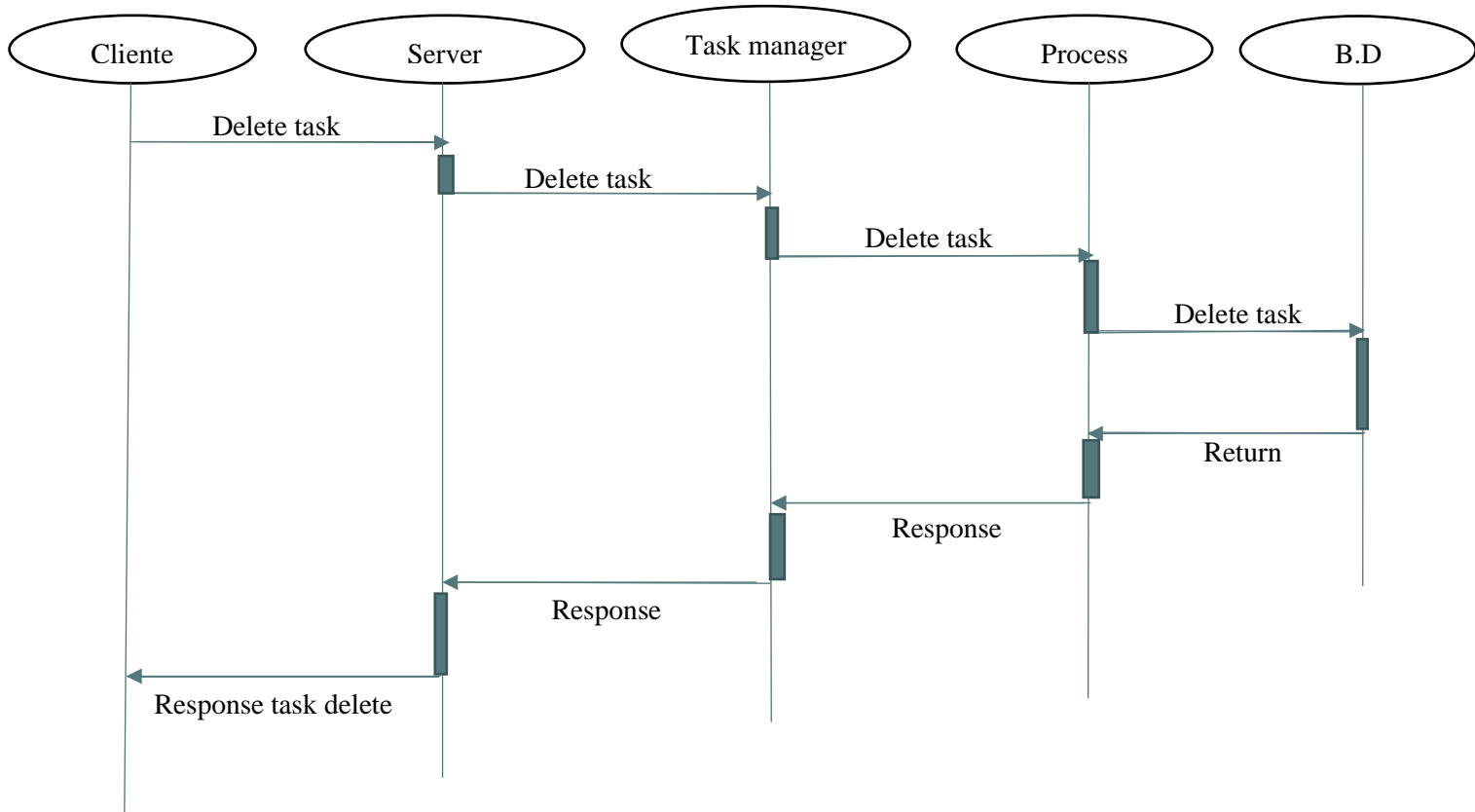
Editar tarea:



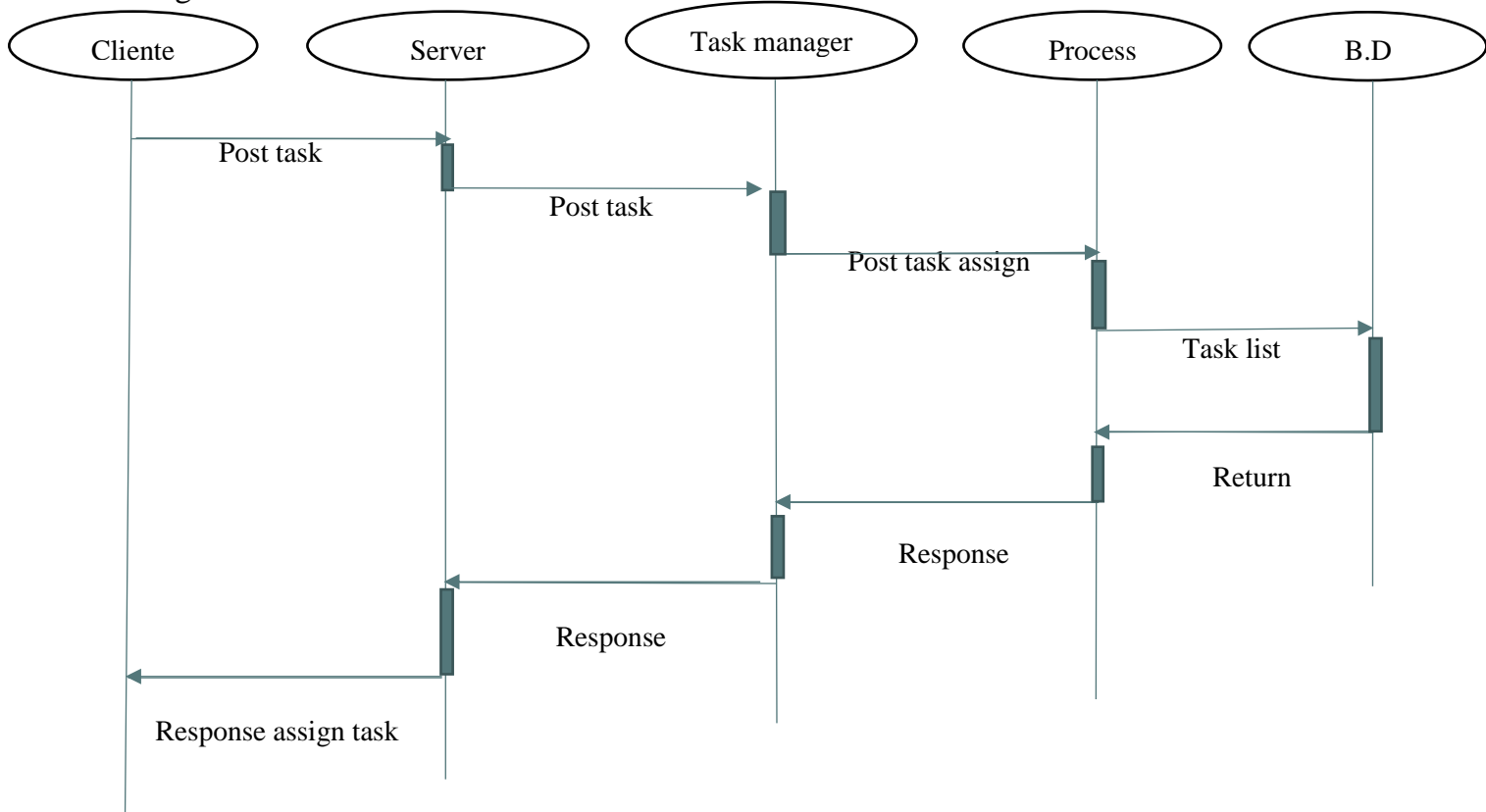
# Listar tarea:



### Eliminar tarea:

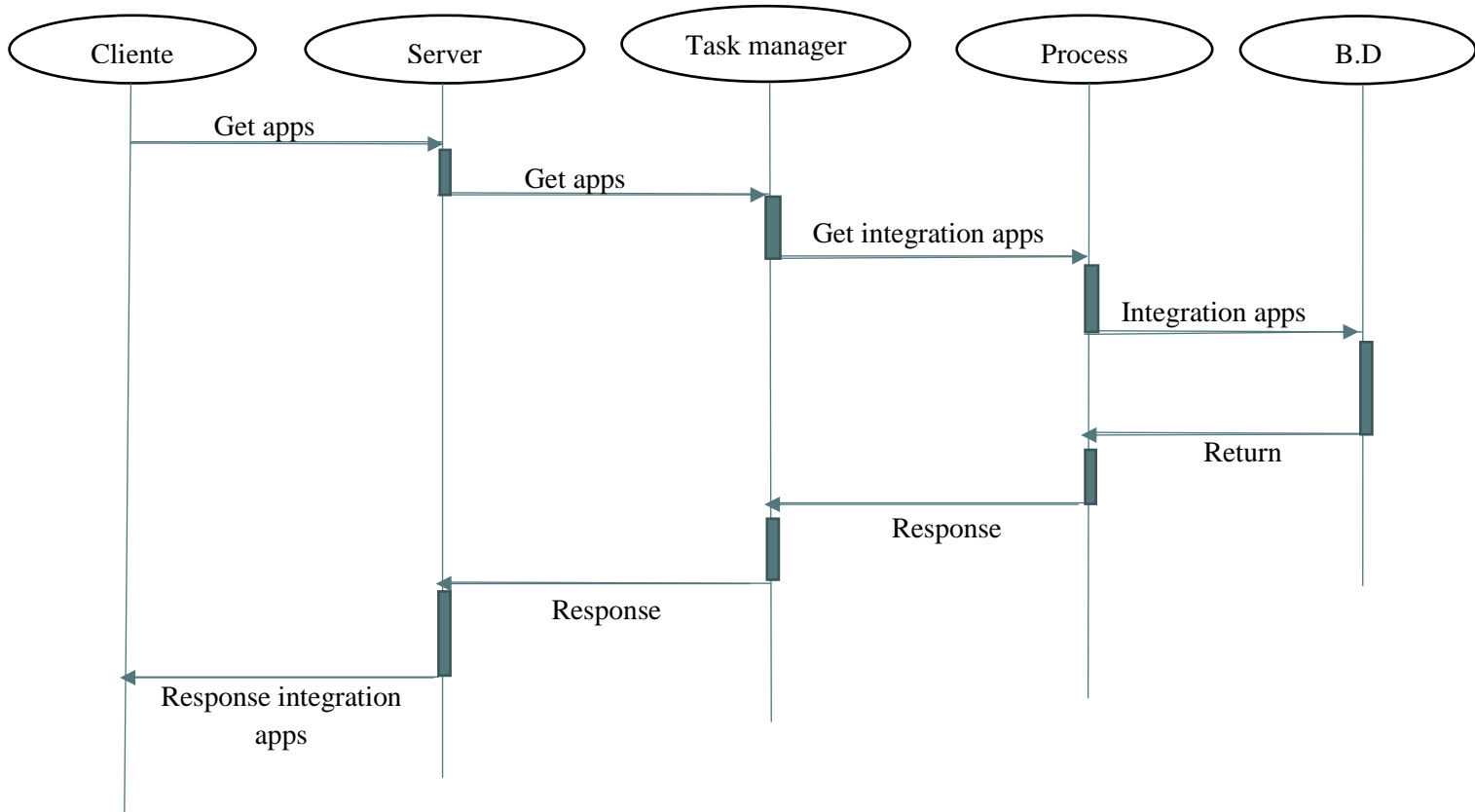


### Asignar Tarea:

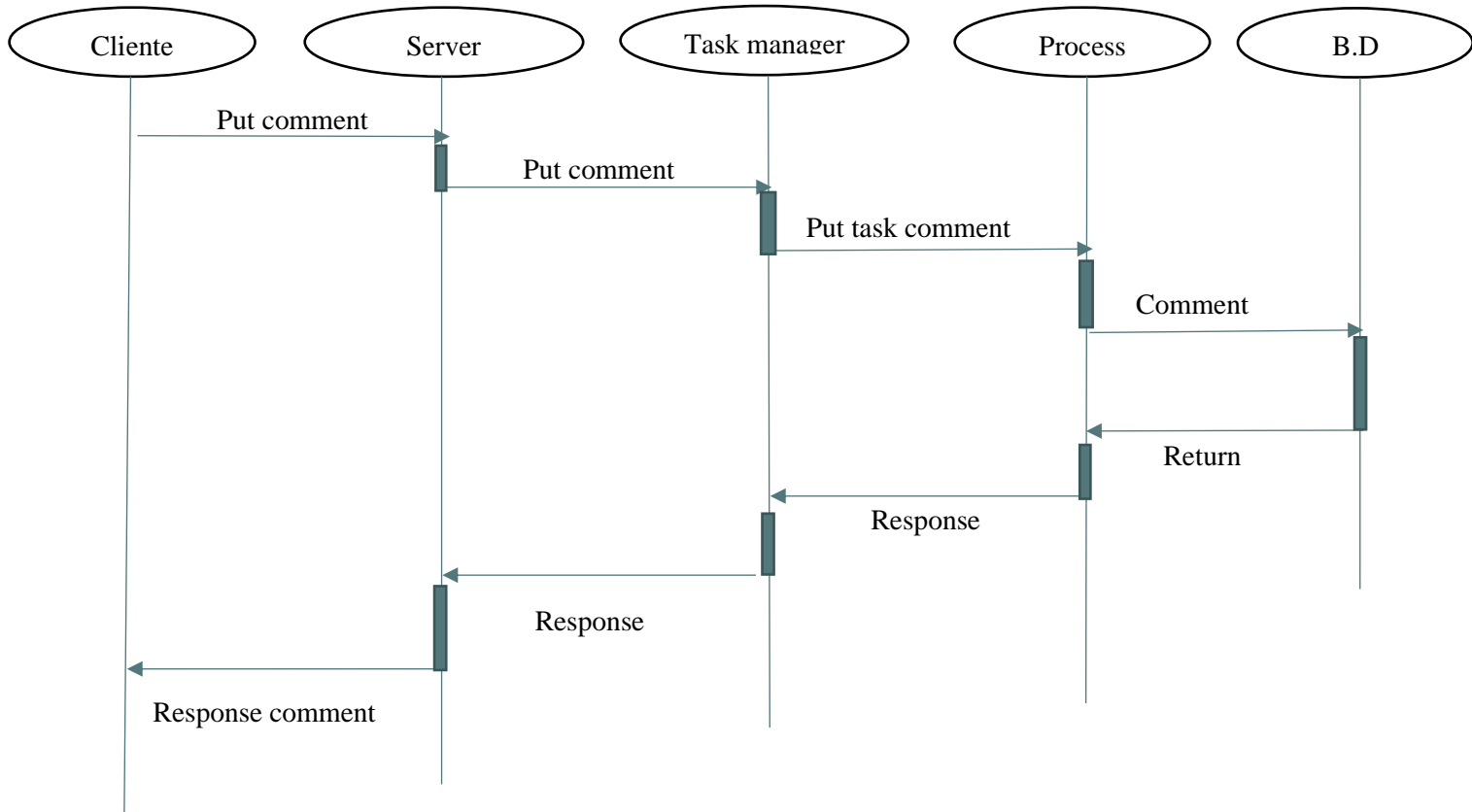




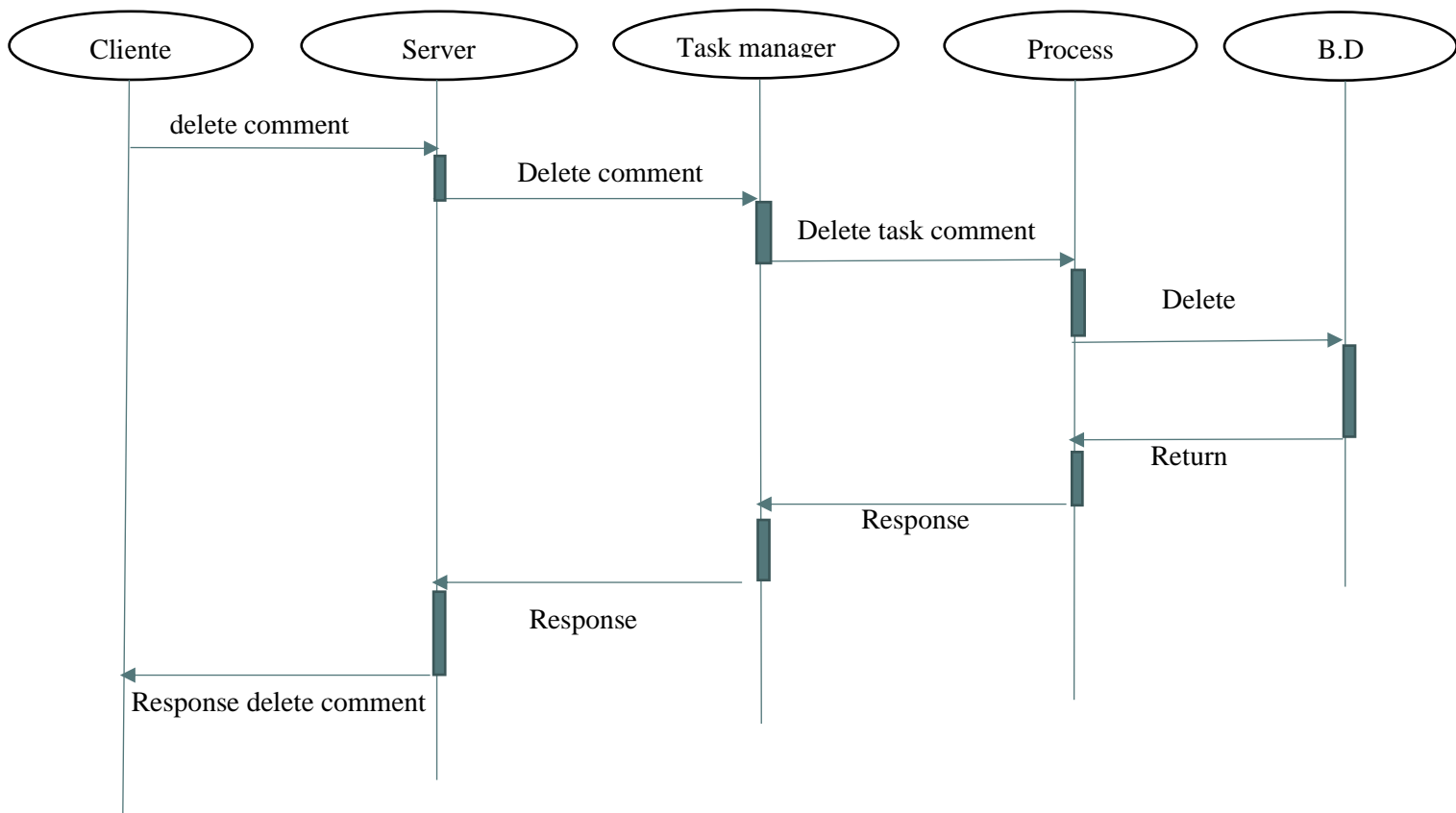
### Integración con apps externas:



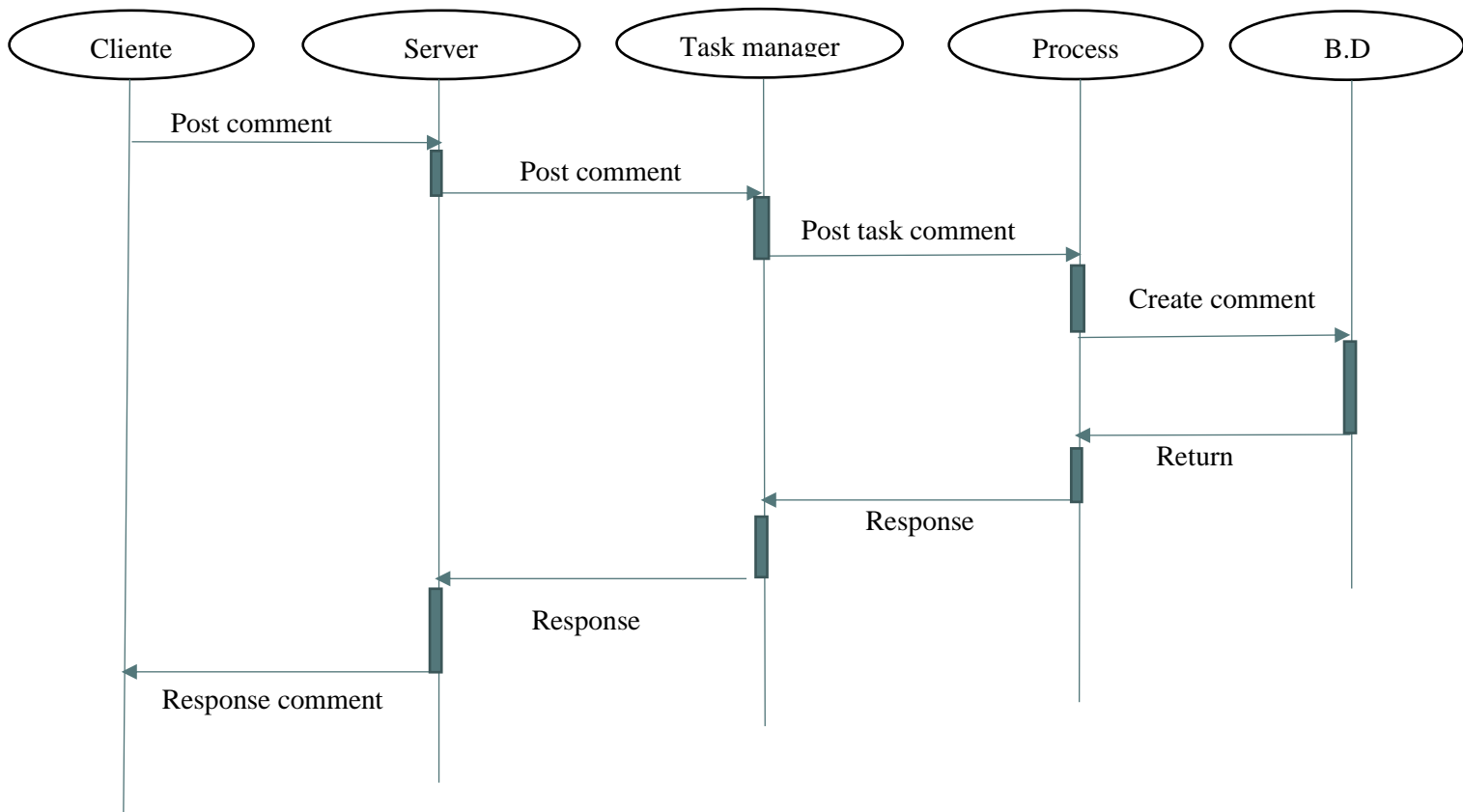
### Comentar Tarea:



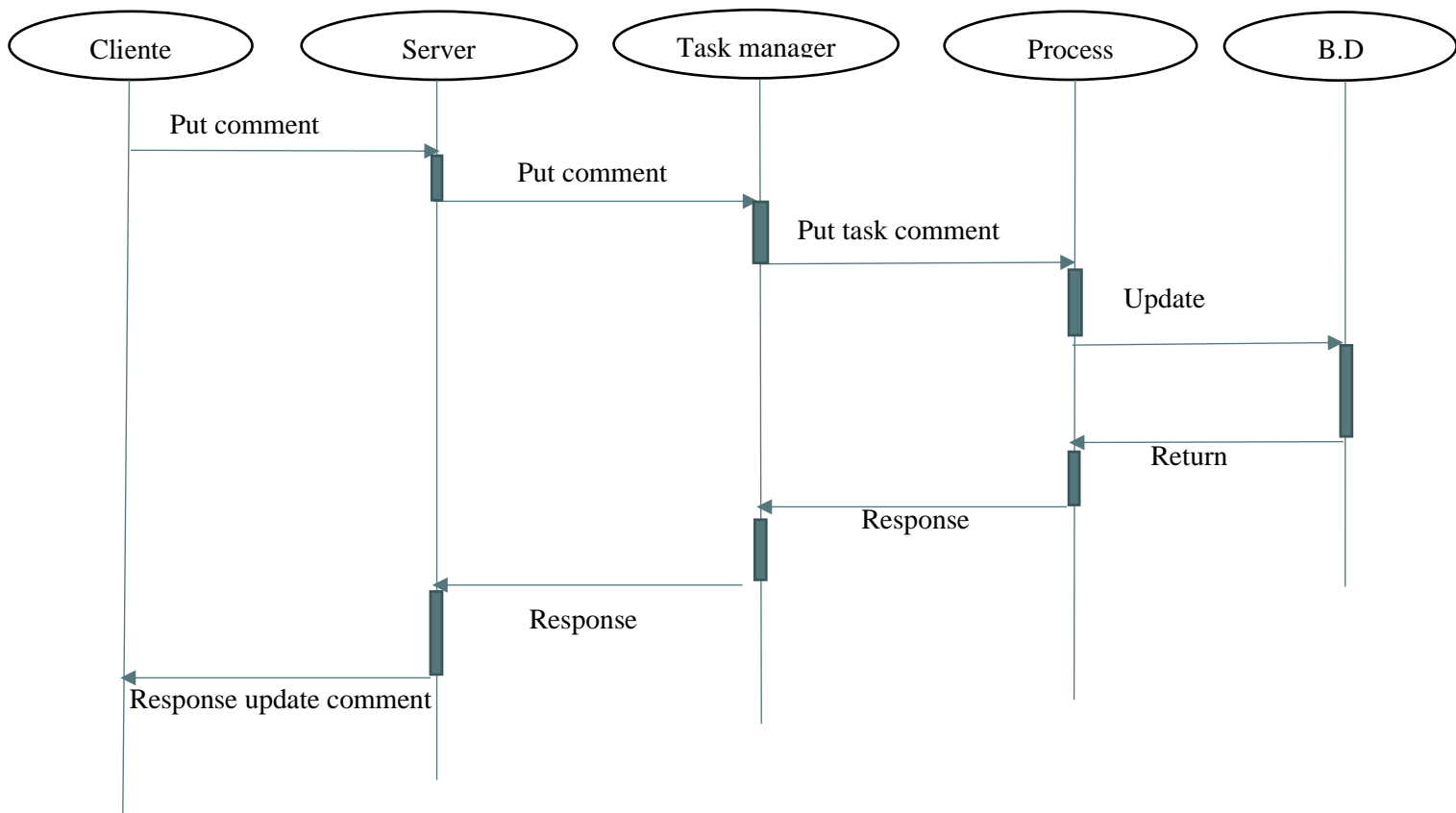
### Borrar comentario tarea:



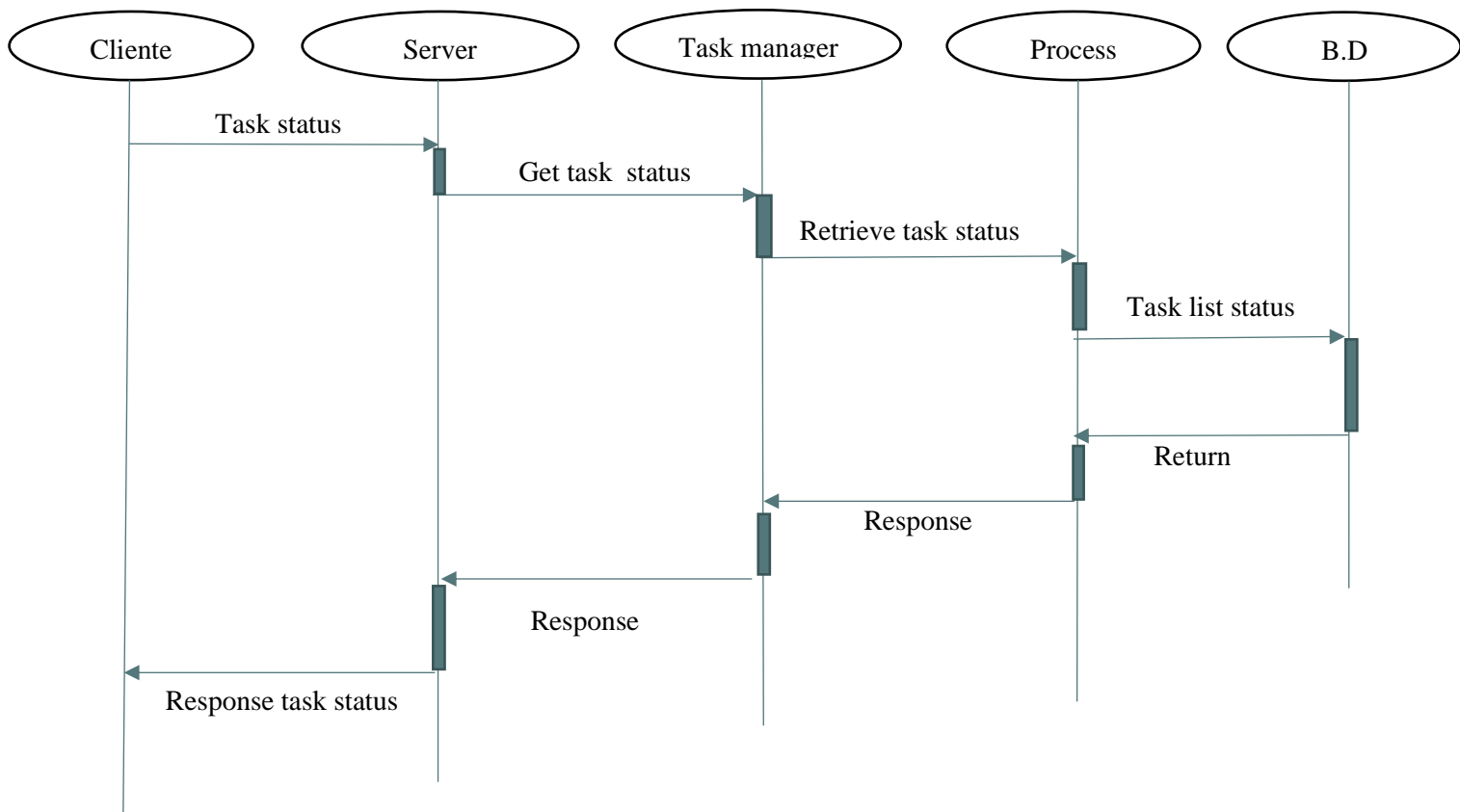
### Responder comentario tarea:



### Editar comentario tarea:



### Estado Tarea:



**Prioridad de Requisitos**

**Requisitos No Funcionales**

**Requisitos de Desempeño**

**Requisitos de Seguridad**

**Requisitos de Usabilidad**

**Requisitos de Escalabilidad**

**Modelado E/R**

**Caracterización de los datos**

**Diagrama de Entidad-Relación**

## **Diagrama relacional**

## **Descripción de Entidades y Relaciones**

## **Reglas de Integridad**

## **Anexos (si es necesario)**

## **Diagramas Adicionales**

## **Referencias**

# **Etapla 2: Persistencia de Datos con Backend**

## **Introducción**

## **Propósito de la Etapa**

## **Alcance de la Etapa**

## **Definiciones y Acrónimos**

## **Diseño de la Arquitectura de Backend**

## **Descripción de la Arquitectura Propuesta**

## **Componentes del Backend**

## **Diagramas de Arquitectura**

## **Elección de la Base de Datos**

## **Evaluación de Opciones (SQL o NoSQL)**

## **Justificación de la Elección**

**Diseño de Esquema de Base de Datos**

**Implementación del Backend**

**Elección del Lenguaje de Programación**

**Creación de la Lógica de Negocio**

**Desarrollo de Endpoints y APIs**

**Autenticación y Autorización**

**Conexión a la Base de Datos**

**Configuración de la Conexión**

**Desarrollo de Operaciones CRUD**

**Manejo de Transacciones**

**Pruebas del Backend**

**Diseño de Casos de Prueba**

## **Ejecución de Pruebas Unitarias y de Integración**

## **Manejo de Errores y Excepciones**



## **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**