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)

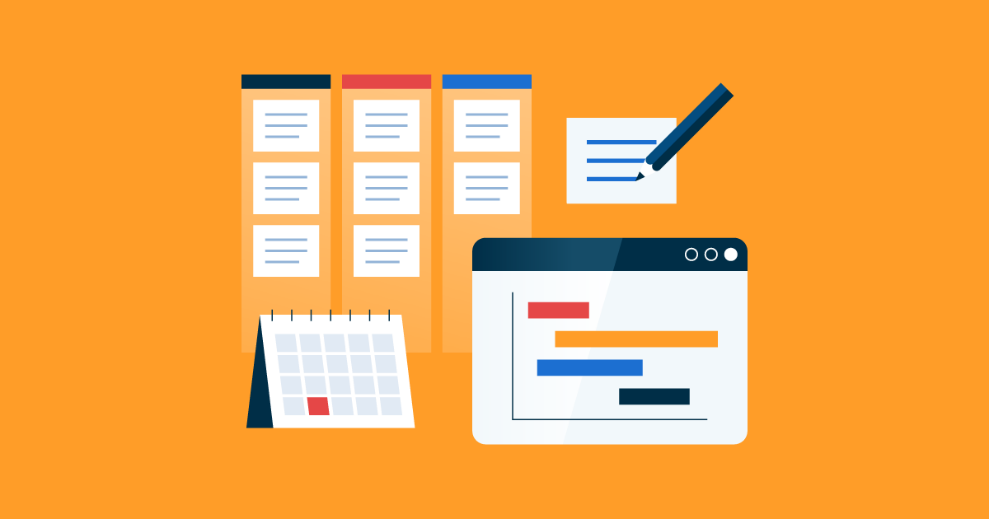
Antonio José Gómez Banda - [agomezbanda30@correo.unicordoba.edu.co](mailto:agomezbanda30@correo.unicordoba.edu.co)

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)

Tutor: 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 5](#_Toc148193298)

[Introducción 5](#_Toc148193299)

[Propósito del Documento 5](#_Toc148193300)

[Alcance del Proyecto 5](#_Toc148193301)

[Definiciones y Acrónimos 5](#_Toc148193302)

[Descripción General 5](#_Toc148193303)

[Objetivos del Sistema 5](#_Toc148193304)

[Funcionalidad General 5](#_Toc148193305)

[Usuarios del Sistema 5](#_Toc148193306)

[Restricciones 5](#_Toc148193307)

[Requisitos Funcionales 5](#_Toc148193308)

[Casos de Uso 5](#_Toc148193309)

[Descripción detallada de cada caso de uso 5](#_Toc148193310)

[Diagramas de Flujo de Casos de Uso 5](#_Toc148193311)

[Prioridad de Requisitos 5](#_Toc148193312)

[Requisitos No Funcionales 5](#_Toc148193313)

[Requisitos de Desempeño 5](#_Toc148193314)

[Requisitos de Seguridad 6](#_Toc148193315)

[Requisitos de Usabilidad 6](#_Toc148193316)

[Requisitos de Escalabilidad 6](#_Toc148193317)

[Modelado E/R 6](#_Toc148193318)

[Diagrama de Entidad-Relación 6](#_Toc148193319)

[Descripción de Entidades y Relaciones 6](#_Toc148193320)

[Reglas de Integridad 6](#_Toc148193321)

[Anexos (si es necesario) 6](#_Toc148193322)

[Diagramas Adicionales 6](#_Toc148193323)

[Referencias 6](#_Toc148193324)

[Etapa 2: Persistencia de Datos con Backend 7](#_Toc148193325)

[Introducción 7](#_Toc148193326)

[Propósito de la Etapa 7](#_Toc148193327)

[Alcance de la Etapa 7](#_Toc148193328)

[Definiciones y Acrónimos 7](#_Toc148193329)

[Diseño de la Arquitectura de Backend 7](#_Toc148193330)

[Descripción de la Arquitectura Propuesta 7](#_Toc148193331)

[Componentes del Backend 7](#_Toc148193332)

[Diagramas de Arquitectura 7](#_Toc148193333)

[Elección de la Base de Datos 7](#_Toc148193334)

[Evaluación de Opciones (SQL o NoSQL) 7](#_Toc148193335)

[Justificación de la Elección 7](#_Toc148193336)

[Diseño de Esquema de Base de Datos 7](#_Toc148193337)

[Implementación del Backend 7](#_Toc148193338)

[Elección del Lenguaje de Programación 7](#_Toc148193339)

[Creación de la Lógica de Negocio 7](#_Toc148193340)

[Desarrollo de Endpoints y APIs 7](#_Toc148193341)

[Autenticación y Autorización 7](#_Toc148193342)

[Conexión a la Base de Datos 8](#_Toc148193343)

[Configuración de la Conexión 8](#_Toc148193344)

[Desarrollo de Operaciones CRUD 8](#_Toc148193345)

[Manejo de Transacciones 8](#_Toc148193346)

[Pruebas del Backend 8](#_Toc148193347)

[Diseño de Casos de Prueba 8](#_Toc148193348)

[Ejecución de Pruebas Unitarias y de Integración 8](#_Toc148193349)

[Manejo de Errores y Excepciones 8](#_Toc148193350)

[Etapa 3: Consumo de Datos y Desarrollo Frontend 9](#_Toc148193351)

[Introducción 9](#_Toc148193352)

[Propósito de la Etapa 9](#_Toc148193353)

[Alcance de la Etapa 9](#_Toc148193354)

[Definiciones y Acrónimos 9](#_Toc148193355)

[Creación de la Interfaz de Usuario (UI) 9](#_Toc148193356)

[Diseño de la Interfaz de Usuario (UI) con HTML y CSS 9](#_Toc148193357)

[Consideraciones de Usabilidad 9](#_Toc148193358)

[Maquetación Responsiva 9](#_Toc148193359)

[Programación Frontend con JavaScript (JS) 9](#_Toc148193360)

[Desarrollo de la Lógica del Frontend 9](#_Toc148193361)

[Manejo de Eventos y Comportamientos Dinámicos 9](#_Toc148193362)

[Uso de Bibliotecas y Frameworks (si aplicable) 9](#_Toc148193363)

[Consumo de Datos desde el Backend 9](#_Toc148193364)

[Configuración de Conexiones al Backend 9](#_Toc148193365)

[Obtención y Presentación de Datos 9](#_Toc148193366)

[Actualización en Tiempo Real (si aplicable) 9](#_Toc148193367)

[Interacción Usuario-Interfaz 10](#_Toc148193368)

[Manejo de Formularios y Validación de Datos 10](#_Toc148193369)

[Implementación de Funcionalidades Interactivas 10](#_Toc148193370)

[Mejoras en la Experiencia del Usuario 10](#_Toc148193371)

[Pruebas y Depuración del Frontend 10](#_Toc148193372)

[Diseño de Casos de Prueba de Frontend 10](#_Toc148193373)

[Pruebas de Usabilidad 10](#_Toc148193374)

[Depuración de Errores y Optimización del Código 10](#_Toc148193375)

[Implementación de la Lógica de Negocio en el Frontend 10](#_Toc148193376)

[Migración de la Lógica de Negocio desde el Backend (si necesario) 10](#_Toc148193377)

[Validación de Datos y Reglas de Negocio en el Frontend 10](#_Toc148193378)

[Integración con el Backend 10](#_Toc148193379)

[Verificación de la Comunicación Efectiva con el Backend 10](#_Toc148193380)

[Pruebas de Integración Frontend-Backend 10](#_Toc148193381)

# 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 centrará 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 ofrecerá 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 buscará 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.

**Etapa 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. A continuación, se describirán una serie de características concebidas para la presente versión y otras que se proponen para versiones futuras:

* Creación de listas de tareas: Permite a los usuarios crear y organizar listas de tareas según sus necesidades y prioridades.
* Asignación de tareas: Posibilita asignar tareas específicas a diferentes usuarios o equipos, facilitando la colaboración y el trabajo en equipo.
* Programación de fechas límite: Permite establecer fechas límite para cada tarea, ayudando a mantener un seguimiento y cumplimiento de los plazos.
* Recordatorios y notificaciones: Envía recordatorios y notificaciones automáticas sobre tareas próximas a vencer o completadas.
* Seguimiento del progreso: Proporciona herramientas para realizar un seguimiento del progreso de las tareas, como barras de progreso o etiquetas de estado.
* Comentarios y discusiones: Permite a los usuarios comentar y discutir sobre las tareas dentro de la plataforma, facilitando la comunicación y la colaboración.
* Personalización de la interfaz: Ofrece opciones de personalización de la interfaz para adaptarse a las preferencias y necesidades de los usuarios.

**Para futuras versiones, se proponen características adicionales como:**

* Integración con calendarios externos: Permite sincronizar las tareas con calendarios externos, como Google Calendar o Outlook.
* Generación de informes y análisis: Ofrece herramientas para generar informes y análisis sobre el progreso y rendimiento de las tareas.
* Automatización de tareas recurrentes: Permite automatizar tareas recurrentes mediante la creación de plantillas o reglas predefinidas.

## 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: Asincrónico 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).

# 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

# 

# Requisitos Funcionales

## 

## Casos de Uso

## 

## Descripción detallada de cada caso de uso

## 

## Diagramas de Flujo de Casos de Uso Diagramas de Secuencia

## 

## 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

# Etapa 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

# Etapa 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