**TaskMaster Pro**

Manual de Instalación

Versión: 0100

[Versión del Producto]

**HOJA DE CONTROL**

| **Organismo** | SENA | | |
| --- | --- | --- | --- |
| **Proyecto** | TaskMaster Pro | | |
| **Entregable** | Manual de Instalación | | |
| **Autor** | Johan Garcia  Nikole Bernal  Andrés Garzón  Erika Triana | | |
| **Aprobado por** |  | **Fecha Aprobación** | 12/06/2025 |
|  |  | **Nº Total de Páginas** |  |

REGISTRO DE CAMBIOS

| **Versión** | **Causa del Cambio** | **Responsable del Cambio** | **Fecha del Cambio** |
| --- | --- | --- | --- |
| 0100 | Versión inicial | Johan Felipe Garcia Salazar | 12/06/2025 |
| 0200 | Finalización punto 4 | Johan Felipe Garcia Salazar | 18/06/2025 |
| 0300 | Correción puntos | Johan Felipe Garcia Salazar | 27/06/2025 |

CONTROL DE DISTRIBUCIÓN

| **Nombre y Apellidos** |
| --- |
| Johan Felipe Garcia Salazar |
| Nikole Camila Bernal Ávila |
| Andrés Julián Garzón Perea |
| Erika Daniela Triana Bustos |

[1 INTRODUCCIÓN 4](#_heading=h.1ysnuo4sdpfb)

[1.1 Objetivo 4](#_heading=h.sbs155ptf7xo)

[1.2 Alcance 4](#_heading=h.4ccdsn1nplbv)

[2 DESCRIPCIÓN DEL SISTEMA 5](#_heading=h.vtriwxl2fhq5)

[2.1 Antecedentes y descripción funcional del sistema 5](#_heading=h.4cev0c875ecs)

[2.2 Componentes fundamentales 6](#_heading=h.7bg2f86x4wnf)

[2.3 Relación con otros sistemas 6](#_heading=h.umfb6i7i89te)

[3 RECURSOS HARDWARE 7](#_heading=h.mmfkfltercvc)

[3.1 Servidores 7](#_heading=h.nnoc8oen6c2m)

[3.2 Estaciones cliente 7](#_heading=h.jfw3bc9lkewv)

[3.3 Conectividad 8](#_heading=h.sanrjzelcqd1)

[3.4 Restricciones 8](#_heading=h.xow0ofk05e7)

[4 RECURSOS SOFTWARE 9](#_heading=h.kfc0byki20xb)

[4.1 Matriz de certificación 9](#_heading=h.ngr7r7tb8v9h)

[4.2 Restricciones técnicas del sistema 9](#_heading=h.z7p7e9cf8avf)

[5 INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE BASE 10](#_heading=h.e26v6ho5w110)

[6 CONFIGURACIÓN DEL SISTEMA 12](#_heading=h.v4pn8ykavqol)

[6.1 Configuración del sistema 12](#_heading=h.wwch9zir9nq4)

[6.2 Configuración de otros sistemas 12](#_heading=h.rcap2rzstlg)

[7 COMPILACIÓN DEL SISTEMA 14](#_heading=h.bo3ckv2n06i2)

[8 INSTALACIÓN DEL SISTEMA 16](#_heading=h.mpa2oxhj9h4y)

[8.1 Requisitos previos 16](#_heading=h.gwcjjgv78pw9)

[8.2 Procedimiento de instalación 17](#_heading=h.spy7ch4oyvvf)

[9 VERIFICACIÓN DEL PROCESO DE INSTALACIÓN 20](#_heading=h.79x0e290dfgl)

[10 MARCHA ATRÁS DE LA INSTALACIÓN Y CONFIGURACIÓN 21](#_heading=h.3w963gal8u3e)

[10.1 Requisitos previos 21](#_heading=h.1vabx06ohmn7)

[10.2 Marcha atrás del sistema 21](#_heading=h.ryi9rut3j21o)

[10.3 Marcha atrás del software base 22](#_heading=h.3h5wz9ogv4nw)

[11 ANEXOS 23](#_heading=h.as0rcqs1e1pv)

[11.1 Resumen de tareas de configuración 24](#_heading=h.dwzuf2tb2lrv)

[12 GLOSARIO 26](#_heading=h.gc2cq6vclduz)

[13 BIBLIOGRAFÍA Y REFERENCIAS 27](#_heading=h.e1uho3jbtp9s)

# INTRODUCCIÓN

## Objetivo

Brindar una guía clara y detallada al administrador del sistema o desarrollador para la correcta y segura instalación y configuración del software en el entorno donde será ejecutado.

## Alcance

El manual de instalación abarca los puntos más importantes para la instalación del sistema.

Algunos de estos puntos son:

* Recursos de hardware necesarios y mínimos para el correcto funcionamiento del sistema.
* Recursos de software (instalación de herramientas necesarias como: NodeJS, MySQL, etc).
* Clonación o descarga del proyecto.
* Instalación de dependencias.
* Configuración de archivos.
* Creación o importación de base de datos.
* Ejecución del sistema y verificación básica.

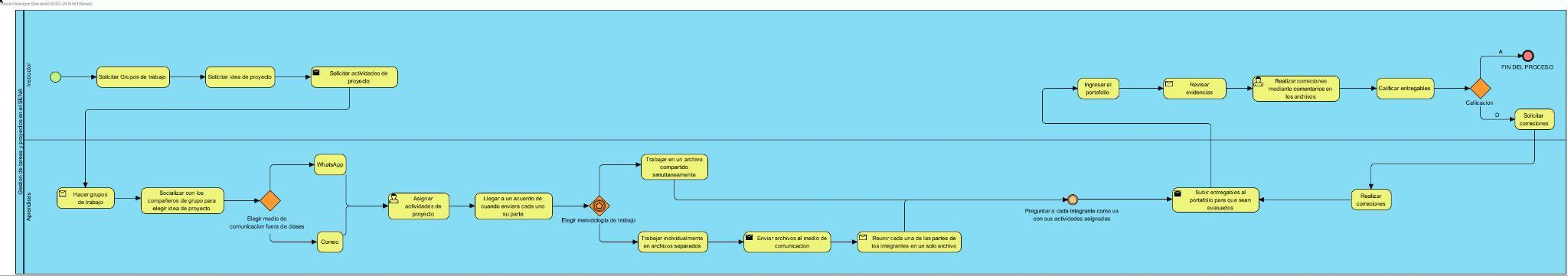
# DESCRIPCIÓN DEL SISTEMA

## Antecedentes y descripción funcional del sistema

El SENA, en el centro CEET, es una organización educativa que ofrece formación gratuita con programas técnicos, tecnológicos y complementarios enfocados a las ramas de estudio de electricidad, electrónica y telecomunicaciones.

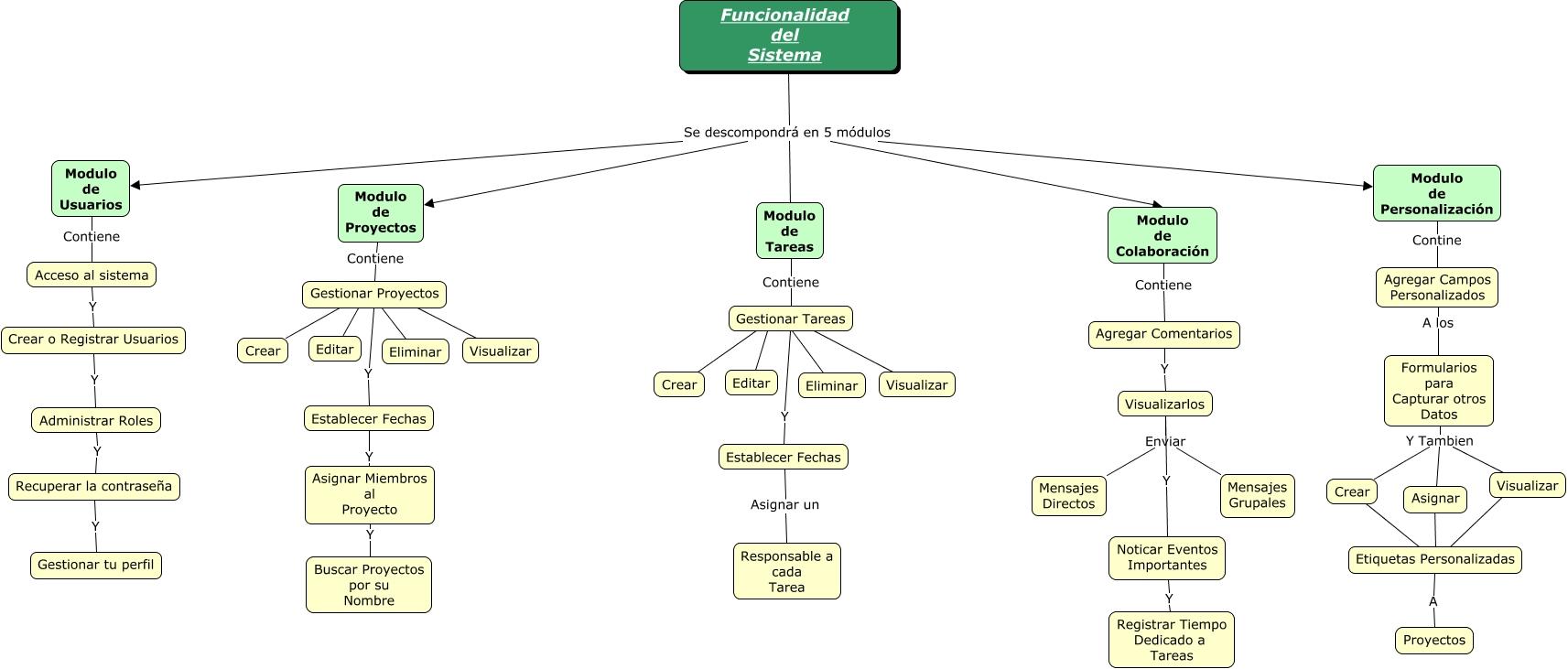
En el SENA la metodología de estudio está basada en proyectos donde se imparten los conocimientos de los programas académicos a partir de una necesidad y que a partir de esa necesidad, se conforman los diferentes grupos de trabajo para darle una solución.

A partir de las actividades de levantamiento de información (entrevista, encuesta, análisis de competencia y observación) realizadas en esta organización, se obtuvo como resultado la identificación de una problemática en la gestión y ejecución de proyectos de los aprendices, que se desarrollan dentro del ciclo del conocimiento de manera grupal en los programas que ofrece el centro (CEET). Esto debido a que la forma de organizar, repartir, entregar y monitorear los avances de las actividades o entregables de estos proyectos no es óptima ni centralizada y puede causar retraso en la presentación de resultados o el no cumplimiento de objetivos propuestos.

[](https://github.com/FGFERNAN/TaskMasterPro/blob/main/docs/trim01/02_mapa_procesos/BPMN%20Proyecto%20(Gestion%20de%20proyectos%20y%20actividades).jpg)

Investigaciones realizadas han demostrado que el uso de sistemas de información (software) en la gestión de proyectos trae consigo bastantes beneficios, mejora en los resultados, mayor productividad, organización e información centralizada.

De este modo se propone el desarrollo de un sistema de información web que sirva como herramienta de apoyo al seguimiento de la gestión de proyectos y actividades de los aprendices, pretendiendo generar una optimización de los procesos y mayor facilidad a la hora de realizar sus proyectos formativos.

[](https://github.com/FGFERNAN/TaskMasterPro/blob/main/docs/trim05/04_Manuales/Diagramas%20de%20Apoyo%20Manuales/Funcionalidad%20del%20Sistema.jpg)

## Componentes fundamentales

| **Módulo** | **Descripción** |
| --- | --- |
| Módulo de Usuarios | Permite la autenticación al sistema y gestión de usuarios (CRUD), recuperación de contraseña, asignación de roles para controlar el acceso a las funcionalidades del sistema. |
| Módulo de Gestión de Proyectos | Permite gestionar todo lo que tenga que ver con proyectos como, crear, editar, eliminarlos, asignar miembros, definir fechas, estados y también utilizar plantillas prediseñadas para no crear un proyecto desde 0. |
| Módulo de Gestión de Tareas | Permite gestionar tareas (CRUD), asignarles un responsable miembro del proyecto, cambiar su estado y fechas y buscarlas por su nombre. |
| Módulo de Colaboración | Permite comunicación entre miembros de proyecto, ya sea directa o grupal, adjuntar archivos a las distintas tareas que hagan parte del proyecto y registrar el tiempo dedicado a cada tarea. |
| Módulo de Personalización | Permite agregar campos personalizados en los formularios de creación de proyectos, esto con el fin de capturar datos adicionales específicos para proyectos únicos. También eliminar editar estos campos y crear, editar, asignar y ver etiquetas personalizadas que se agregan a los proyectos para clasificarlos. |

## Relación con otros sistemas

| **Sistema** | **Relación** |
| --- | --- |
| Looker Studio | La relación con esta herramienta de google es especialmente para el tema de reportes de los registros de tiempo dedicados a tarea, ya que por medio de una integración que se hizo del sistema TaskMaster con esta herramienta, se pueden ver reportes gráficos y tabulados detalladamente de estos registros, lo cual facilita su entendimiento. |
| Gmail | La relación con esta aplicación es que para el tema de recuperar contraseña se necesita un servicio de correo electrónico gratuito y el elegido fue gmail, entonces siempre los correos ingresados para recuperar contraseña, el sistema buscará enviarlos a este cliente de correo y si no encuentra esta dirección o se da cuenta que no hace parte de un correo de Gmail, sencillamente rebotara el correo al remitente es decir TaskMaster |

# RECURSOS HARDWARE

## Servidores

| **Servidor de Aplicaciones** | | |
| --- | --- | --- |
| **Dato** | **Valor mínimo** | **Valor recomendado** |
| Procesador | 4 núcleos | 8 núcleos |
| Memoria RAM | 4 GB | 16 GB |
| Tamaño Almacenamiento | 100 GB SSD | 256 GB SSD |
| Sistema Operativo | Ubuntu Server 20.04 | Ubuntu Server 22.04 LTS |

| **Servidor de Base de Datos** | | |
| --- | --- | --- |
| **Dato** | **Valor mínimo** | **Valor recomendado** |
| Procesador | 4 núcleos | 8 núcleos |
| Memoria RAM | 4 GB | 16 GB |
| Tamaño Almacenamiento | 20 GB SSD | 64 GB SSD |
| Sistema Operativo | Ubuntu Server 20.04 | Ubuntu Server 22.04 LTS |
| Puerto | 3306 | 3306 |

El servidor de archivos no necesita un servidor dedicado ya que al manejar servicios de almacenamiento en la nube como Google Drive no siempre los archivos quedarán subidos a un servidor. Sin embargo se recomienda un servidor local (una carpeta uploads en el backend) que será el mismo de servidor de aplicaciones, por ende sus características.

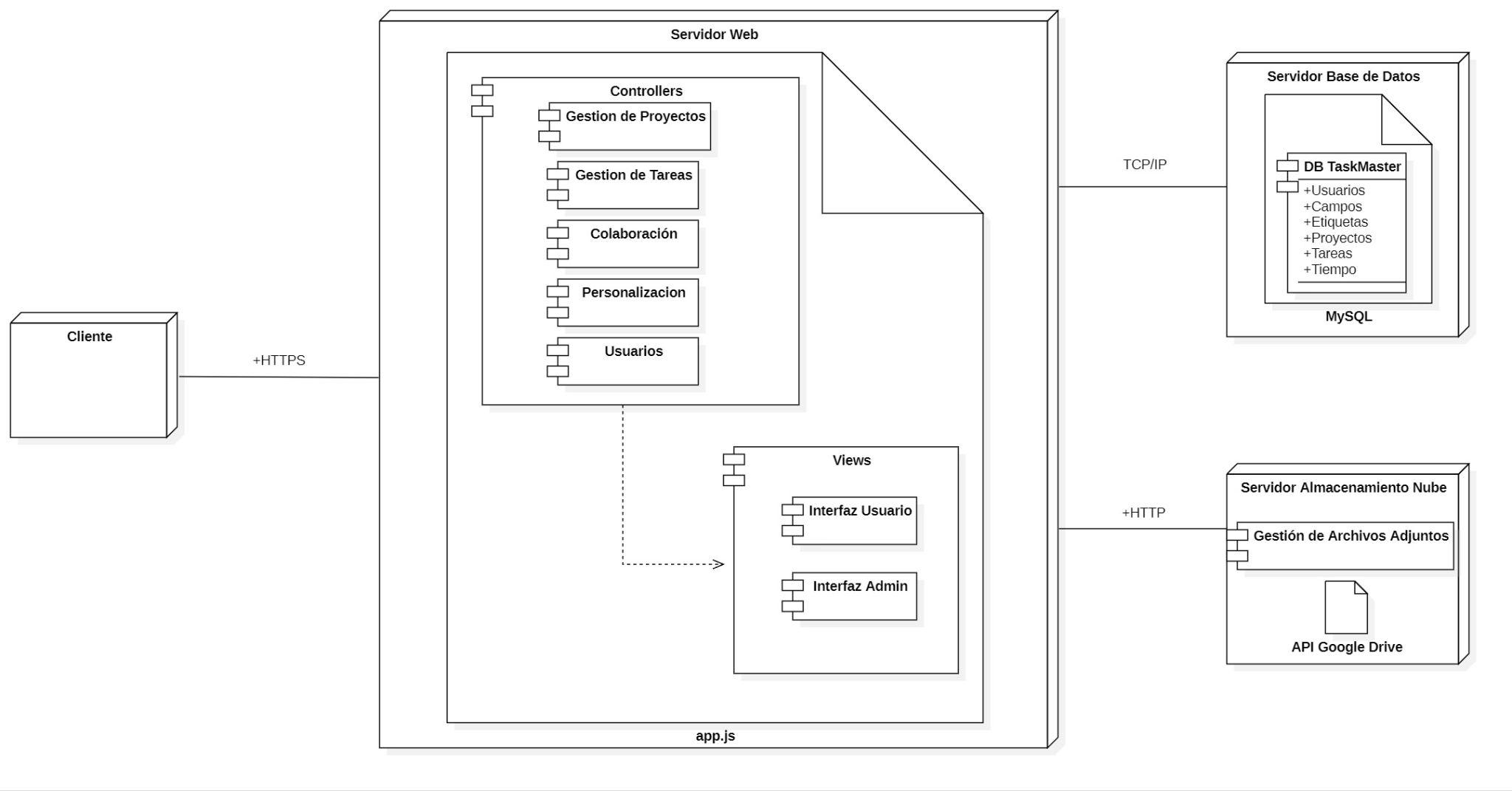
## Estaciones cliente

| **Dato** | **Valor mínimo** | **Valor recomendado** |
| --- | --- | --- |
| Procesador | Intel Core I3 o Ryzen 3 | Intel Core I5 o Ryzen 5 |
| Memoria RAM | 8 GB | 16 GB |
| Tamaño Almacenamiento | 128 GB SSD | 512 GB |
| Sistema Operativo | Windows 10 | Windows 11 |
| Navegador | Google Chrome on Microsoft Edge | Google Chrome |
| Pantalla | 15.6 Pulgadas | 24 Pulgadas |
|  |  |  |

## 

## Conectividad

| **Dato** | **Valor mínimo** | **Valor recomendado** |
| --- | --- | --- |
| Tarjeta de Red | Ethernet 10/100 Mbps o Wi-Fi 4 (802.11n) | Ethernet 1 Gbps o Wi-Fi 5/6 (802.11ac/ax) |
| Tipo de Red | Red local cableada o Wi-Fi con señal estable | Red cableada dedicada (LAN) o Wi-Fi 5/6 con señal fuerte |
| Otros | Latencia < 100 ms, 300 Mbps | Latencia < 50 ms, 500 Mbps |

[](https://github.com/FGFERNAN/TaskMasterPro/blob/main/docs/trim02/05_diagrama_de_despliegue/Diagrama_de_Despliegue_TaskMaster_Pro_02.jpg)

## Restricciones

| **Restricción** | **Detalle** |
| --- | --- |
| Dependencia de conexión a internet | Al ser una aplicación web depende 100% de una conexión estable a internet para acceder al sistema y sus funcionalidades. |
| Compatibilidad con navegador | La interfaz del sistema ha sido probada y optimizada para Google Chrome y Microsoft Edge. Otros navegadores pueden presentar inconsistencias.  No compatible con navegadores sin soporte para ES6+ (como Internet Explorer). |
| No compatible con sistemas operativos antiguos | El sistema no garantiza funcionamiento correcto en versiones obsoletas de Windows (como Windows 7) |
| Versión de Node.js | El sistema depende de node.js v20.18.0 y no funciona con versiones anteriores |
| Versión MySQL | El sistema requiere MySQL versión 8.0 y disponibilidad del puerto 3306 |

# RECURSOS SOFTWARE

## Matriz de certificación

| **Componente** | **Versión** | **Observaciones** |
| --- | --- | --- |
| Node.JS (Backend) | 20.18.0 | Probado en esta versión, compatible con versiones superiores. |
| NPM (Gestor de Paquetes) | 10.9.0 |
| MySQL (Base de Datos) | 8.0.38 |
| Navegador Web | Google Chrome 137 / Edge 137 |
| Sistema Operativo Servidor | Ubuntu 22.04 LTS |
| Git | 2.46.2 |
| API | RESTful (JSON) |

## Restricciones técnicas del sistema

| **Elemento** | **Descripción** |
| --- | --- |
| Sistema operativo | El servidor debe contar con un sistema operativo estable basado en Linux o Windows Server, con acceso de administrador o privilegios sudo. |
| Servidor de aplicaciones | Node.js y NPM deben estar instalados. Express es usado como framework principal. Puerto de ejecución: 4000. |
| Servidor de base de datos | MySQL instalado en el mismo servidor o en uno remoto. El puerto 3306 debe estar abierto para conexiones externas. |
| Servicios de Red | Protocolo de comunicación HTTP/HTTPS.  Puertos requeridos: 4000 (backend), 8080 (frontend), 443 (HTTPS). |
| Control de Versiones | Git para clonar el repositorio desde GitHub. |
| Entorno de Desarrollo | Visual Studio Code versión o superior, con extensiones sugeridas: Prettier, ESLint. |
| Ubicación del Servidor | Se puede desplegar en servidor físico, máquina virtual o túnel seguro (ej. DevTunnels) con acceso desde los clientes. |

# INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE BASE

En esta sección se describen los componentes de software base necesarios para la correcta instalación y funcionamiento del sistema TaskMaster Pro. Cada uno de estos componentes es esencial para compilar, ejecutar o conectar el sistema, ya sea en entornos de desarrollo o producción.

El objetivo de esta sección es asegurar que cualquier técnico, administrador o desarrollador pueda preparar el entorno base de manera segura y coherente.

| **<Software Base 1>:Visual Studio Code** | | |
| --- | --- | --- |
| Descripción | Visual Studio Code (VS Code) es un editor de código multiplataforma que permite a los desarrolladores inspeccionar, editar y ejecutar el proyecto TaskMaster de forma eficiente. | |
| Localización | <https://code.visualstudio.com/> | |
| Procedimiento de instalación | | |
| Paso 1 | Ingresar a:https://code.visualstudio.com/ | |
| Paso 2 | Descargar la versión para Windows e iniciar el instalador. | |
| Paso 3 | Aceptar las opciones por defecto e instalar. | |
| Procedimiento de configuración | | |
| Paso 1 | Instalar extensiones recomendadas: ESLint, Prettier, MySQL, Node.js Extension Pack. | |
| Paso 2 | Abrir la carpeta del proyecto TaskMaster con Archivo > Abrir Carpeta. | |
| Paso 3 | Usar la terminal integrada (Ctrl + ñ) para ejecutar comandos como npm install, npm start | |
| Parámetros a configurar | | |
| Extensiones útiles | Marketplace de extensiones de VS Code | ESLint, Prettier, Node.js, GitLens,Live server |

| **<Software Base 1>:GitHub** | | |
| --- | --- | --- |
| Descripción | Git es un sistema de control de versiones que permite gestionar los cambios realizados en el código fuente. GitHub es una plataforma de alojamiento de repositorios remotos donde está alojado el proyecto TaskMaster. | |
| Localización | |  | | --- |  | <https://git-scm.com> y<https://github.com> | | --- | | |
| Procedimiento de instalación | | |
| Paso 1 | Ingresar a <https://git-scm.com> | |
| Paso 2 | Descargar la versión estable recomendada para Windows: **Git 2.45.1** o superior | |
| Paso 3 | Ejecutar el instalador y aceptar las opciones por defecto | |
| Paso 4 | Configurar Git con tu nombre y correo:  git config --global user.name "TuNombre"  git config --global user.email "tucorreo@dominio.com" | |
| Procedimiento de configuración | | |
| Paso 1 | Iniciar sesión en GitHub y acceder al repositorio del proyecto TaskMaster. | |
| Paso 2(Opción 1) | **Clonar desde Visual Studio Code**:  Presiona Ctrl + Shift + P → escribe “**Git: Clonar**” → pega la URL ejemplo:https://github.com/usuario/proyecto.git) → selecciona una carpeta local → abre el proyecto. | |
| Paso 3 (Opción 2) | **Clonar desde la terminal**:  1. Abre la terminal y ve al directorio deseado con cd  2. Ejecuta:  git clone https://github.com/usuario/proyecto.git  3. Entra al proyecto:  cd proyecto | |
| Paso 4 | Usar la terminal integrada o el panel de control de código fuente para ejecutar comandos:  git add .  git commit -m "mensaje"  git push | |
| Parámetros a configurar | | |
| Git en VS Code | Uso del panel de control de código fuente para commits y sincronización | Panel "Control de código fuente" |
| GitLens | Extensión para ver historial, autores, ramas y contribuciones en los archivos | Extensión GitLens instalada desde el Marketplace |

| **<Software Base 2>:Node.js** | | |
| --- | --- | --- |
| Descripción | Node.js es un entorno de ejecución para JavaScript que permite ejecutar código del lado del servidor. Task Master utiliza Node.js para levantar el servidor y ejecutar las funcionalidades backend. Incluye npm, el gestor de paquetes para instalar dependencias del proyecto. | |
| Localización | <https://nodejs.org> | |
| Procedimiento de instalación | | |
| Paso 1 | Ingrese al sitio oficial de Node.js:https://nodejs.org | |
| Paso 2 | Descargar la versión LTS recomendada (ej Node 18.x LTS) | |
| Paso 3 | Ejecutar el instalador y seguir los pasos por defecto | |
| Procedimiento de configuración | | |
| Paso 1 | Verificar instalación en la consola : node -v y npm -v | |
| Parámetros a configurar | | |
| PATH | Variables del entorno operativo | C:\Program Files\nodejs\ |

| **<Software Base 2>:XAMPP** | | |
| --- | --- | --- |
| Descripción | XAMPP es un entorno de desarrollo que instala Apache, MariaDB (compatible con MySQL), PHP y phpMyAdmin. Permite ejecutar servidores locales para aplicaciones web como Task Master sin necesidad de instalar cada componente por separado. | |
| Localización | <https://www.apachefriends.org/index> | |
| Procedimiento de instalación | | |
| Paso 1 | Ingrese al sitio oficial de Xampp:<https://www.apachefriends.org/index.html> | |
| Paso 2 | Descargar el instalador para Windows (recomendado: versión 8.x o superior) | |
| Paso 3 | Ejecutar el instalador y seguir los pasos por defecto. | |
| Paso 4 | Al finalizar, abrir el Panel de control de XAMPP. Iniciar los servicios Apache y MySQL (botón "Start").. | |
| Procedimiento de configuración | | |
| Paso 1 | Verificar que el servicio MySQL esté corriendo desde el panel de control. | |
| Paso 2 | Validar que el usuario root esté disponible y que el puerto usado sea el 3306. | |
| Parámetros a configurar | | |
| Puerto base de datos | Panel de control de XAMPP → Config (MySQL | 3306 |
| Usuario base de datos | phpMyAdmin → Usuarios | root |
| Contraseña | (por defecto vacía) | (puede establecerse si se desea mayor seguridad) |

# CONFIGURACIÓN DEL SISTEMA

## Configuración del sistema

| **Configuración: Conexión a la base de datos** | | |
| --- | --- | --- |
| **Efecto** | | Permite que el backend de Taskmaster se conecte correctamente con la base de datos para leer y guardar datos**.** |
| **Fase** | | Inicial |
| **Ubicación** | | TaskMaster\_Pro\backend\src\config\dbConnection.js |
| **Paso** | **Descripción** | |
| **1º** | Navegar a la rut: TaskMaster\_Pro/backend/src/config/dbConnection.js | |
| **2º** | Verificar que los datos estén definidos como:  host:'localhost', user:'root', password:'', database: 'task\_master', port: 3306 | |
| **3º** | Guardar y ejecutar npm start para confirmar que el backend se conecta correctamente a la base de datos | |

## Configuración de otros sistemas

En esta sección se describen herramientas adicionales que, aunque no forman parte del software base del sistema TaskMaster, son fundamentales para el trabajo de desarrollo y pruebas.

Estos sistemas permiten a los desarrolladores interactuar, verificar o visualizar componentes del sistema como bases de datos, APIs o control de versiones. Su configuración adecuada facilita la correcta ejecución, depuración y mantenimiento del proyecto.

| **<Otro sistema 1> Postman** | | |
| --- | --- | --- |
| Descripción | Postman es una herramienta que permite realizar pruebas de las API REST del sistema Task Master sin necesidad del frontend. Facilita el envío de peticiones HTTP y la validación de respuestas. | |
| Localización | <https://www.postman.com/downloads> | |
| Procedimiento de instalación | | |
| Paso 1 | Acceder al sitio oficial de Postman: postman.com/downloads | |
| Paso 2 | Descargar el instalador para Windows e instalar siguiendo los pasos por defecto | |
| Paso 3 | Ejecutar Postman y crear una cuenta gratuita o iniciar sesión | |
| Procedimiento de configuración | | |
| Paso 1 | Crear una nueva colección para agrupar las peticiones de la API Task Master | |
| Paso 2 | Configurar una petición GET, POST, PUT o DELETE apuntando a http://localhost:3000/ (o el puerto configurado) | |
| Paso 3 | Agregar en "Body" el formato raw en JSON para probar endpoints como creación de tareas o login de usuario | |
| Parámetros a configurar | | |
| URL de la API | Campo URL de la petición | http://localhost:3000/ |
| Tipo de petición | Menú desplegable del request | GET, POST, PUT, DELETE |
| Formato del cuerpo | Body > Raw > JSON | {"nombre": "tarea de prueba"} |

# 

| **<Otro sistema 1> MySQL Workbench** | | |
| --- | --- | --- |
| Descripción | MySQL Workbench es una herramienta visual que permite conectarse a la base de datos MySQL del sistema Taskmaster para gestionar tablas, ejecutar consultas SQL y hacer seguimiento de la estructura del sistema. | |
| Localización | <https://dev.mysql.com/downloads/workbench/> | |
| Procedimiento de instalación | | |
| Paso 1 | Acceder al sitio oficial de MySQL Workbench y descargar la versión Community para Windows. | |
| Paso 2 | Ejecutar el instalador y seguir los pasos de instalación por defecto. | |
| Paso 3 | Abrir MySQL Workbench desde el menú de inicio. | |
| Procedimiento de configuración | | |
| Paso 1 | Crear una nueva conexión a la base de datos | |
| Paso 2 | Configurar la conexión con estos datos: host localhost, puerto 3306, usuario root, contraseña *(vacía si usas XAMPP por defecto)*. | |
| Paso 3 | Probar la conexión y guardar. Una vez abierta, puedes ejecutar sentencias SQL o explorar las tablas del sistema task\_master | |
| Parámetros a configurar | | |
| Host | Dirección del servidor MySQL | localhost |
| Puerto | Puerto de conexión | 3306 |
| Usuario | Credenciales de acceso | root |
| Contraseña | Contraseña del usuario | (vacia) |
| Base de datos | Nombre de la BD | task\_master |

# COMPILACIÓN DEL SISTEMA

| **Requisitos de compilación** | |
| --- | --- |
| **Requisito** | **Descripción** |
| **Ubicación Fuentes** | Nuestro proyecto principal está alojado en el repositorio de Github y en la ruta designada en la clonación del repositorio. |
| **Configuración 1** | Descargar los sistemas como Visual Studio Code y XAMPP. |
|  |  |
| **Repositorio 1** | El repositorio se encuentra en la página de GitHub a través de este direccionamiento https://github.com/FGFERNAN/TaskMasterPro.git. |
| **...** |  |
| **Dependencia 1** | Instalación de paquetes NPM y dependencias de axios. |
| **...** |  |
| **Producto final** | El entorno donde se despliega el software y se muestra funcional. |

| **Procedimiento de compilación** | |
| --- | --- |
| **Paso** | **Descripción** |
| **1** | Realizar la clonación del repositorio a través del enlace |
| **2** | Descargar directamente desde las páginas homologadas el sistema de edición de código Visual Studio Code y XAMPP en tu respectivo sistema operativo.  Adicional Instalar Node.js (incluye npm), Vue CLI globalmente y XAMPP para servidor Apache y MySQL.  Configurar los aplicativos y ejecutarlos. |
| **3** | Al abrir ambos sistemas primero tendrás que abrir el proyecto desde Visual Studio Code en la ruta donde se realizó la clonación del repositorio en GitHub.  En el caso esta es la ruta D:\TaskMasterPro  Se verá algo así: |
| **4** | En el paso siguiente vamos a crear nuestra base datos por lo que vamos a ejecutar el aplicativo de XAMPP y activaremos el módulo de apache y MySQL: |
| **5** | En el módulo de MySQL nos vamos al sistema de Administrador en el botón “Admin” |
| **6** | Crear la base de datos task\_master\_pro e importar el archivo task\_master\_pro.sql. |
| **7** | Ya configurada la base de datos y nuestros sistemas principales en proceso, vamos a Visual Studio Code y abrimos dos terminales. |
| **8** | En la primera terminal nos vamos a la carpeta del Backend Web ejecutando el comando “cd backend\_web” |
| **9** | En esta misma terminal del Backend vamos a ejecutar el comando “npm install -g npm” ejecutamos y se nos instalaran los paquetes del sistema de compilación. |
| **10** | Se ejecuta dentro de la terminal del backend el comando para la compilación “npm run dev” |
| **11** | En la otra terminal nos vamos a la carpeta del Frontend Web ejecutando el comando “cd frontend\_web” |
| **12** | En esta misma terminal del Frontend vamos a ejecutar el comando “npm install -g npm” ejecutamos y se nos instalaran los paquetes del sistema de compilación. |
| **13** | Se ejecuta dentro de la terminal del frontend el comando para la compilación “npm run serve” |

# INSTALACIÓN DEL SISTEMA

En este apartado se detalla el proceso necesario para instalar y configurar correctamente el sistema **TaskMaster Pro** en un entorno local. Se describen los requisitos previos necesarios, tanto de software como de configuración, así como los pasos secuenciales para llevar a cabo la instalación completa del backend, frontend y base de datos del sistema.

## Requisitos previos

| **Recurso o Requisito** | **Descripción** |
| --- | --- |
| Sistema Operativo | Windows 10 o superior |
| Node.js y npm | Versión 18.x o superior |
| Vue CLI | Instalado globalmente con npm install -g @vue/cli |
| Visual Studio Code | Editor de código recomendado para trabajar con el proyecto |
| XAMPP | Para ejecutar el servidor Apache y MySQL |
| Navegador web moderno | Chrome, Firefox u otro compatible con aplicaciones Vue.js |
| Git | Para clonar el repositorio desde GitHub |
| Conexión a internet | Para descarga de librerías y dependencias |
| Base de datos MySQL | Para el almacenamiento de los datos del sistema |

Para la instalación y ejecución del sistema **TaskMaster Pro** en entorno local, es necesario contar con acceso al código fuente, ya que:

* Se deben ejecutar comandos como npm install y npm run serve, que requieren los archivos originales del frontend y backend.
* Es necesario configurar archivos como .env, que no están disponibles en un sistema ya desplegado.
* La base de datos debe ser importada manualmente desde un archivo SQL incluido en el proyecto.

En un entorno de producción, este acceso puede limitarse entregando solo:

* Archivos compilados del frontend (npm run build)
* Backend desplegado en un servidor
* Interfaz accesible mediante navegador sin necesidad de modificar el código fuente

Estas decisiones dependen del objetivo del despliegue: si es para desarrollo, pruebas, uso interno o distribución final.

## Procedimiento de instalación

| **Procedimiento de instalación** | |
| --- | --- |
| **Paso 1** | |
| **Tipo** | Instalación de software |
| **Componente** | Node.js, Vue CLI, XAMPP |
| **Permisos** | Administrador |
| **Descripción** | Instalar los siguientes componentes:  • Node.js desde<https://nodejs.org>, que incluye **npm**.  • Instalar Vue CLI globalmente con el comando: npm install -g @vue/cli  • Descargar e instalar XAMPP desde<https://www.apachefriends.org> para disponer de Apache y MySQL. |
| **Paso 2** | |
| **Tipo** | Repositorio del sistema (GitHub) |
| **Componente** | Repositorio del sistema |
| **Permisos** | Ninguno |
| **Descripción** | El repositorio del sistema TaskMaster puede descargarse de dos formas:  **Desde Visual Studio Code:**  1. Presionar Ctrl + Shift + P  2. Escribir y seleccionar **“**Git: Clonar**”**  3. Pegar la URL del repositorio: https://github.com/usuario/taskmaster.git  4. Elegir carpeta local y abrir el proyecto.  **Desde Git Bash (por comandos)**:  1. Abrir Git Bash  2. Ir a la carpeta destino:  cd /c/Users/TuUsuario/Documents  3. Clonar el repositorio:  git clone https://github.com/usuario/taskmaster.git  4. Entrar al proyecto:  cd taskmaster  Con esto tendrás el proyecto TaskMaster listo para continuar la instalación. |

| **Paso 3** | |
| --- | --- |
| **Tipo** | Configuración de base de datos |
| **Componente** | MySQL (XAMPP) |
| **Permisos** | Acceso a phpMyAdmin |
| **Descripción** | 1.Abrir XAMPP y activar los servicios de Apache y MySQL.  2. Crear una nueva base de datos llamada: task\_master\_pro  3.Importar el archivo task\_master\_pro.sql |

| **Paso 4** | |
| --- | --- |
| **Tipo** | Instalación de dependencias |
| **Componente** | Backend |
| **Permisos** | Ninguno |
| **Descripción** | Abrir una terminal en la carpeta del backend:  cd taskmaster/backend  2. Ejecutar el siguiente comando para instalar todas las dependencias necesarias:  npm install  Esto instalará los paquetes definidos en el archivo package.json para que el backend funcione correctamente. |

| **Paso 5** | |
| --- | --- |
| **Tipo** | Configuración |
| **Componente** | Backend |
| **Permisos** | Ninguno |
| **Descripción** | 1 ️ Dentro de la carpeta del backend del proyecto (taskmaster/backend), verifica si existe un archivo llamado .env. Si no existe, créalo manualmente.  2 Abre el archivo .env y agrega las siguientes variables de entorno para configurar la conexión con la base de datos y el puerto del servidor:  env<br>PORT=3000<br>DB\_HOST=localhost<br>DB\_USER=root<br>DB\_PASSWORD=””l<br>DB\_NAME=task\_master\_pro<br>  Este archivo le indica al servidor backend en qué puerto ejecutarse y cómo conectarse a la base de datos. |

| **Paso 6** | |
| --- | --- |
| **Tipo** | Compilacion |
| **Componente** | Backend |
| **Permisos** | Ninguno |
| **Descripción** | 1 Abrir una terminal en la carpeta del backend:  cd taskmaster/backend  2 Ejecutar el siguiente comando para iniciar el servidor en modo desarrollo:  npm run dev  Esto pondrá en marcha el servidor backend utilizando el archivo .env para la configuración. Por defecto, se ejecuta en http://localhost:4000. |

| **Paso 7** | |
| --- | --- |
| **Tipo** | Instalación de dependencias |
| **Componente** | Frontend (Vue.js) |
| **Permisos** | Ninguno |
| **Descripción** | 1 Abrir una terminal y dirigirse a la carpeta del frontend del proyecto:  cd taskmaster/frontend  2 Ejecutar el siguiente comando para instalar las dependencias necesarias del proyecto Vue:  npm install  Esto descargará e instalará todos los paquetes definidos en el archivo package.json del frontend. Es un paso obligatorio antes de ejecutar la interfaz del sistema. |

| **Paso 8** | |
| --- | --- |
| **Tipo** | Compilación |
| **Componente** | Frontend |
| **Permisos** | Ninguno |
| **Descripción** | 1 Abrir una terminal en la carpeta del frontend:  cd taskmaster/frontend  2 Ejecutar el siguiente comando para iniciar la aplicación:  npm run serve  El sistema se abrirá en el navegador por defecto en http://localhost:8080. Aquí podrás ver y probar la interfaz de usuario del sistema TaskMaster. |

| **Paso 9** | |
| --- | --- |
| **Tipo** | Verificación |
| **Componente** | Sistema completo (frontend + backend) |
| **Permisos** | Acceso a navegador |
| **Descripción** | 1 Abrir el navegador e ingresar a la URL del frontend:  http://localhost:8080  2 Verificar que la interfaz cargue correctamente y que las funcionalidades estén operativas (crear tareas, consultar, editar, eliminar, etc.).  3 Asegurarse de que el backend esté en ejecución (http://localhost:4000) para validar el intercambio correcto entre frontend y backend.  Si todo funciona como se espera, la instalación fue exitosa. |

# VERIFICACIÓN DEL PROCESO DE INSTALACIÓN

Una vez finalizada la instalación del sistema Task Master Pro, se deben realizar las siguientes comprobaciones mínimas:

| **Verificación** | **Descripción** |
| --- | --- |
| Acceso al sistema | Confirmar que se puede ingresar a http://localhost:8080 . |
| Comunicación con backend | Validar que las funciones que requieren API (login, crear tarea) funcionen. |
| Conexión con base de datos | Verificar que los datos se guardan y consultan correctamente. |
| Funcionamiento básico | Probar inicio de sesión, creación de proyectos, tareas, etc. |
| Consola de errores | Revisar que no existan errores en la consola del navegador o terminal. |
| Integración completa | Confirmar que todos los componentes se comunican correctamente. |

# MARCHA ATRÁS DE LA INSTALACIÓN Y CONFIGURACIÓN

## Requisitos previos

Antes de ejecutar el proceso de marcha atrás, se debe:

* Realizar una copia de seguridad de la base de datos si se desea conservar la información.
* Detener los servicios que estén activos, como los servidores de frontend (npm run serve) o backend (npm run dev).
* Asegurarse de tener privilegios administrativos para eliminar archivos, carpetas y servicios si es necesario.

## Marcha atrás del sistema

Este apartado detalla las acciones necesarias para revertir completamente la instalación del software base utilizado para el funcionamiento de Task Master Pro. Incluye los pasos para desinstalar herramientas como Node.js, Vue CLI y XAMPP, así como para eliminar cualquier configuración o archivo residual asociado. A continuación, se presentan los pasos organizados por tipo de software y ubicación.

| **Procedimiento de marcha atrás** | |
| --- | --- |
| **Paso 1** | |
| **Tipo** | Detención de servicios |
| **Componente** | Frontend y Backend |
| **Permisos** | Usuario con acceso a la terminal |
| **Descripción** | Detener los servicios en ejecución con Ctrl + C o cerrando las terminales activas. |
| **Paso 2** | |
| **Tipo** | Eliminación de archivos del proyecto |
| **Componente** | Carpeta del sistema |
| **Permisos** | Acceso al sistema de archivos |
| **Descripción** | Eliminar las carpetas que contienen código fuente del frontend y backend |

## Marcha atrás del software base

Este apartado detalla las acciones necesarias para revertir completamente la instalación del software base utilizado para el funcionamiento de Task Master Pro. Incluye los pasos para desinstalar herramientas como Node.js, Vue CLI y XAMPP, así como para eliminar cualquier configuración o archivo residual asociado. A continuación, se presentan los pasos organizados por tipo de software y ubicación.

| **Paso 1** | |
| --- | --- |
| **Tipo** | Desinstalación de software |
| **Componente** | Node.js, Vue CLI, XAMPP |
| **Permisos** | Administrador del sistema |
| **Descripción** | Desinstalar Node.js, XAMPP y Vue CLI desde el panel de control del sistema operativo o mediante comandos. |
| **Paso 2** | |
| **Tipo** | Limpieza de configuraciones residuales |
| **Componente** | Sistema operativo |
| **Permisos** | Administrador |
| **Descripción** | Eliminar carpetas residuales (como .npm, .vue, bases de datos en XAMPP, archivos temporales) que hayan sido creadas por el entorno de desarrollo. |

# ANEXOS

| **Anexo** | **Descripción** | **Ubicación del archivo generado** | **Formato** |
| --- | --- | --- | --- |
| **Reporte de tareas por usuario** | Generado desde el módulo de tareas; contiene el listado de tareas asignadas por usuario. | /backend\_web/reports/ | .pdf/.xlsx |
| **Exportación de proyectos** | Opción para exportar la información de un proyecto completo, incluyendo fechas y responsables. | /backend\_web/exports/projects/ | .csv |
| **Respaldo de base de datos** | Generado automáticamente desde el sistema o manualmente por el administrador. | /db/backups/ | .sql |
| **Informe de actividad del sistema** | Documento con logs o métricas sobre el uso del sistema (accesos, acciones realizadas, etc.). | /backend\_web/logs/ | .log/.txt |
| **Archivos adjuntos del usuario** | Documentos o archivos cargados por el usuario como soporte en tareas o proyectos. | /uploads/attachments/ | Variable |

## Resumen de tareas de configuración

Resumen de elementos de configuración a actualizar

| Este listado resume los elementos que requieren configuración para habilitar el sistema en el entorno local o productivo. Los detalles específicos y pasos de configuración se describen en el apartado "Configuración del Sistema". | |
| --- | --- |
| Elemento de Configuración | Descripción / Consideración |
| Ruta local del proyecto | Definir correctamente el directorio donde se clona el repositorio (ej. D:\TaskMasterPro). |
| XAMPP | Instalar y activar los módulos de Apache y MySQL. |
| Base de datos (MySQL) | Crear la base de datos requerida para el sistema. |
| Archivo de conexión a la base de datos | Actualizar las credenciales y el nombre de la base de datos si varían según el entorno. |
| npm y Node.js | Verificar la instalación y actualizar npm globalmente si es necesario (npm install -g npm). |
| Dependencias del backend | Ejecutar npm install en el directorio backend\_web para instalar paquetes necesarios. |
| Dependencias del frontend | Ejecutar npm install en el directorio frontend\_web para instalar paquetes del cliente. |
| Configuración de puertos (opcional) | Verificar que los puertos usados por el backend y frontend estén libres o correctamente definidos. |
| Axios (en frontend) | Asegurarse de que el endpoint configurado en axios apunte al backend correcto. |
| Configuración de entorno (.env) | (Si aplica) Crear o modificar archivos .env para definir variables como puerto, DB, host, etc. |

# GLOSARIO

| **Término** | **Descripción** |
| --- | --- |
| **Repositorio** | Espacio donde se almacena y gestiona el código fuente de un proyecto, comúnmente ubicado en plataformas como GitHub. |
| **Compilación del sistema** | Proceso de convertir el código fuente en un programa ejecutable, asegurando que todas las dependencias estén instaladas y configuradas correctamente. |
| **Visual Studio Code** | Editor de código fuente desarrollado por Microsoft, usado ampliamente para el desarrollo de software por su compatibilidad con múltiples lenguajes y extensiones. |
| **XAMPP** | Paquete de software que incluye Apache, MySQL, PHP y Perl, utilizado para crear un entorno local de desarrollo web. |
| **Apache** | Servidor web de código abierto que permite alojar y servir aplicaciones web en un entorno local o en línea. |
| **MySQL** | Sistema de gestión de bases de datos relacional usado para almacenar y administrar datos de aplicaciones. |
| **Terminal** | Interfaz de línea de comandos que permite ejecutar instrucciones directamente en el sistema operativo o entorno de desarrollo. |
| **Backend** | Parte del sistema que gestiona la lógica del sistema, la conexión a la base de datos y el procesamiento de datos. |
| **Frontend** | Parte visual del sistema con la que interactúa el usuario final, desarrollada usualmente con tecnologías como HTML, CSS y JavaScript. |
| **npm (Node Package Manager)** | Administrador de paquetes para Node.js que permite instalar bibliotecas y herramientas necesarias para ejecutar un proyecto. |
| **Dependencias** | Bibliotecas o módulos externos que un proyecto necesita para funcionar correctamente (por ejemplo, axios para hacer peticiones HTTP). |
| **Paquetes del sistema de compilación** | Conjunto de bibliotecas y dependencias necesarias para ejecutar y compilar correctamente el proyecto. |
| **Ruta del proyecto** | Ubicación en el sistema de archivos donde se encuentra alojado el código fuente clonado del repositorio. |

# BIBLIOGRAFÍA Y REFERENCIAS

| **Referencia** | **Título** | **Código** |
| --- | --- | --- |
| Ref. 1 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |