

# Análisis de Requisitos - Proyecto CTPGA Manager

## 1. Descripción General del Sistema

El proyecto **CTPGA Manager** es una aplicación web integral diseñada para facilitar la gestión de guías, reportes, eventos y actividades en la institución **SENA**. Su objetivo principal es optimizar la organización académica y administrativa mediante un sistema intuitivo que permite a instructores, administradores y superadministradores coordinar eficientemente sus responsabilidades.. El sistema incluye dos paneles principales:

- **Panel de Administración:** Gestionado por administradores y superadministradores.
- **Panel de Instructores:** Para que los instructores puedan gestionar sus actividades y contenido.

## 2. Roles y Permisos

### Super admin

- Acceso total a todas las funcionalidades.
- Puede ver todas las actividades, reportes y eventos.
- Capacidad para crear, leer, actualizar y eliminar (CRUD) datos.
- Puede visualizar información de los administradores, incluyendo:
  - Quiénes están en línea.
  - A qué área pertenecen.

### Administrador

- Recibe solicitudes de registro de instructores.
- Corroborar la información de los instructores y aprueba o rechaza sus solicitudes de acceso.

### Instructor

- Envía una solicitud de registro que debe ser aprobada por un administrador.
- Una vez aprobado, puede:
  - Crear actividades.
  - Agregar contenido a las guías.

## 3. Flujos de Trabajo

### Registro y Validación de Instructores

1. El instructor se registra en la plataforma.
2. El administrador recibe la solicitud de registro.
3. El administrador revisa la información y aprueba o rechaza el acceso.
4. Si se aprueba, el instructor accede a su panel de control.

## Gestión de Actividades y Guías

1. El instructor crea una nueva actividad.
2. El instructor agrega contenido a la guía asociada, organizando el material en secciones claramente definidas:
  - **Introducción:** Explicación general del propósito de la guía.
  - **Objetivos:** Desglose de los objetivos de aprendizaje.
  - **Materiales Requeridos:** Listado de recursos necesarios.
  - **Desarrollo:** Explicación paso a paso del contenido práctico o teórico.
  - **Evaluación:** Criterios para evaluar el cumplimiento de la actividad.
3. Cada guía creada pasa por un proceso de aprobación automatizado que valida que todos los campos requeridos estén completos. En caso de inconsistencias, se notifica al instructor para realizar las correcciones pertinentes.
4. Una vez aprobada, la guía se publica y queda disponible para los usuarios autorizados.
5. El instructor crea una nueva actividad.
6. El instructor agrega contenido a la guía asociada.
7. Los datos se almacenan en la base de datos MongoDB.

## Gestín de Administradores por el Superadmin

1. El superadmin accede al panel de administración.
2. Visualiza la información de cada administrador:
  - Estado en línea.
  - Área a la que pertenece.
3. Puede realizar acciones CRUD sobre cualquier dato del sistema.

## 4. Requisitos Funcionales

- Sistema de autenticación seguro para cada rol.
- Paneles diferenciados para Superadmin, Administrador e Instructor.
- Sistema de gestión de solicitudes de registro.
- Capacidad de gestionar (crear, editar y eliminar) guías, actividades, eventos y reportes.
- Sistema de notificaciones en tiempo real para alertar sobre solicitudes, aprobaciones o cambios importantes.
- Generación de reportes automatizados en formato Excel para facilitar la gestión documental.
- Historial de actividades para registrar todas las acciones realizadas en el sistema.
- Sistema de autenticación seguro para cada rol.
- Paneles diferenciados para Superadmin, Administrador e Instructor.

- Sistema de gestión de solicitudes de registro.
- Capacidad de gestionar (crear, editar y eliminar) guías, actividades, eventos y reportes.

## 5. Requisitos No Funcionales

- Alta disponibilidad para garantizar el acceso en todo momento.
- Interfaz intuitiva y fácil de usar.
- Seguridad en la transmisión de datos con cifrado SSL.
- Sistema escalable para soportar un alto número de usuarios concurrentes.

## 6. Tecnologías Propuestas

- **Pre-optimizador:** Vite para optimizar el entorno de desarrollo y mejorar el rendimiento del frontend.
- **Backend:** Node.js con Express.js.
- **Base de Datos:** MongoDB.
- **Frontend:** React.js con Context API para la gestión del estado.
- **WebSockets:** Implementación con Socket.io para ofrecer actualizaciones en tiempo real.
- **Estilo:** Tailwind CSS para mejorar la consistencia del diseño y agilizar el desarrollo.
- **CDN:** Implementación de un CDN para optimizar la velocidad de carga de la aplicación.
- **Autenticación:** JWT (JSON Web Tokens) para la gestión segura de sesiones.
- **Despliegue:** Servicios en la nube como AWS, Vercel o DigitalOcean.
- **Pre-optimizador:** Vite para optimizar el entorno de desarrollo y mejorar el rendimiento del frontend.
- **Backend:** Node.js con Express.js.
- **Base de Datos:** MongoDB.
- **Frontend:** React.js con Context API para la gestión del estado.
- **Autenticación:** JWT (JSON Web Tokens) para la gestión segura de sesiones.
- **Despliegue:** Servicios en la nube como AWS, Vercel o DigitalOcean. **Backend:** Node.js con Express.js.
- **Base de Datos:** MongoDB.
- **Frontend:** React.js con Context API para la gestión del estado.
- **Autenticación:** JWT (JSON Web Tokens) para la gestión segura de sesiones.
- **Despliegue:** Servicios en la nube como AWS, Vercel o DigitalOcean. **Backend:** Node.js con Express.js.
- **Base de Datos:** MongoDB.
- **Frontend:** React.js.
- **Autenticación:** JWT (JSON Web Tokens) para la gestión segura de sesiones.
- **Despliegue:** Servicios en la nube como AWS, Vercel o DigitalOcean.

## 7. Diagrama de Flujo

El siguiente diagrama de flujo representa el proceso de registro, aprobación de instructores y gestión de actividades por parte de cada rol (se incluirá el diagrama visual).