

Manual Técnico de Gestión de Expedientes

Introducción

Este manual tiene como objetivo proporcionar una guía detallada para el desarrollo, despliegue y mantenimiento del sistema de Gestión de Expedientes. El sistema está diseñado para facilitar la administración de clientes, citas y expedientes legales dentro de una organización jurídica, asegurando que los usuarios puedan gestionar eficientemente la información y acceder a los documentos legales asociados.

Plataforma de Ejecución

Entorno de Desarrollo

- **Node.js:** Plataforma para construir la API del backend utilizando JavaScript/TypeScript.
- **React con Vite y TypeScript:** Utilizado para la creación del frontend, proporcionando una interfaz de usuario dinámica.
- **MySQL:** Sistema de gestión de bases de datos relacional utilizado para almacenar toda la información del sistema.
- **Docker:** Utilizado para contenerizar y desplegar tanto el backend como la base de datos.
- **Express:** Framework utilizado para construir la API RESTful.

Requisitos del Sistema

- **Node.js:** Versión 18 o superior.
- **MySQL:** Versión 8.0 o superior.
- **Docker:** Versión 20.10 o superior.

Estructura del Proyecto

El proyecto está organizado en múltiples capas, cada una responsable de diferentes aspectos del sistema:

Frontend

- **Ubicación:** ./Frontend
- **Tecnologías:** React, Vite, TypeScript, React Router.
- **Responsabilidad:** Interfaz de usuario para la interacción con los clientes y abogados, incluyendo la visualización de documentos y la gestión de citas.

Backend

- **Ubicación:** ./Backend
- **Tecnologías:** Node.js, Express, TypeScript
- **Responsabilidad:** Proveer la API RESTful para gestionar los expedientes, citas, y clientes. Controlar la lógica de negocio y la interacción con la base de datos.

Base de Datos

- **Ubicación:** ./Database
- **Tecnologías:** MySQL.
- **Responsabilidad:** Almacenar la información de los clientes, citas, y expedientes. La base de datos incluye múltiples tablas para gestionar los diferentes aspectos del sistema.

Componentes Principales

Clientes

- **Funcionalidad:** Permite la gestión de la información de los clientes, como su creación, actualización y consulta.
- **Código Fuente:** ./Backend/controllers/ClientController.js

Citas

- **Funcionalidad:** Permite la programación, actualización y consulta de citas para los clientes.
- **Código Fuente:** ./Backend/controllers/CitaController.js

Expedientes

- **Funcionalidad:** Permite la creación y gestión de expedientes legales. Incluye la generación de códigos únicos para que los clientes puedan consultar sus documentos.
- **Código Fuente:** ./Backend/controllers/AbogadoController.js

Diccionario de Métodos

Cientes

1. **getClient(id: int)**: Recupera la información de un cliente específico.
2. **createClient(data: object)**: Crea un nuevo cliente en la base de datos.
3. **updateClient(id: int, data: object)**: Actualiza la información de un cliente existente.

Citas

1. **crearCita(data: object)**: Crea una nueva cita para un cliente.
2. **actualizarCita(id: int, data: object)**: Actualiza los detalles de una cita existente.
3. **citasPorCliente(id: int)**: Consulta todas las citas asociadas a un cliente específico.

Expedientes

1. **agregarExpediente(data: object)**: Crea un nuevo expediente y genera un código único para el cliente.
2. **generarDocumentos(expedientId: int)**: Recupera el contenido de un expediente.
3. **actualizarEstado(expedientId: int, codigo_Estado: int)**: Actualiza el estado de un expediente.

Ejecución y Despliegue

Docker Compose

El proyecto se despliega utilizando Docker Compose. Asegúrese de tener Docker instalado y configurado antes de continuar.

1. **Configuración:** Revise y modifique los archivos de entorno () con las configuraciones adecuadas para su entorno local o de producción.
2. Construcción:

Python

```
docker-compose build
```

3. Despliegue:

Python

```
docker-compose up -d
```

Comprobación de Salud

El backend y la base de datos están configurados con health checks automáticos para asegurar que ambos servicios estén funcionando correctamente antes de aceptar conexiones.

