

Prueba Técnica: Full Stack Developer (Node.js & Vue 3)

Descripción General

El objetivo de esta prueba es evaluar tus habilidades en el desarrollo de aplicaciones web de extremo a extremo (Full Stack). Deberás construir una herramienta de gestión de documentos CSV que incluya autenticación, roles de usuario y persistencia de datos.

Condiciones de Entrega

- **Plazo máximo:** 4 días naturales a partir de la recepción de este documento.
 - **Tiempo estimado de desarrollo:** Se sugiere dedicar un máximo de 1.5 horas al Backend y 1.5 horas al Frontend (3 horas en total de ejecución activa).
 - **Entregable:** Enlace a un repositorio (GitHub/GitLab) con el código fuente y las instrucciones necesarias para ejecutar el proyecto.
-

Stack Tecnológico (Obligatorio)

- **Backend:** Node.js con Express.
 - **Base de Datos:** Relacional (PostgreSQL o MySQL). **No se permite SQLite.**
 - **ORM:** De libre elección.
 - **Frontend:** Vue 3 utilizando **Option API** (por compatibilidad con estándares internos).
 - **Infraestructura:** Todo el ecosistema debe funcionar dentro de un entorno **Docker** (utilizando `docker-compose`).
-

Requerimientos del Backend (1.5h aprox.)

El servidor debe gestionar la lógica de negocio, la validación de archivos y la seguridad.

1. **Autenticación:**
 - Implementar registro de usuarios con los campos: `nombre`, `contraseña`, `confirmar contraseña` y `rol` (opciones: `user`, `admin`).
 - Implementar Login que retorne un token **JWT**.
2. **Gestión de Archivos CSV:**

- Crear un modelo de base de datos para almacenar la información de los documentos.
 - Implementar un endpoint para la carga de CSV que valide los siguientes campos internos:
 - **correo**: String (formato email obligatorio).
 - **nombre**: String-Correo (obligatorio).
 - **telefono**: Número-Telefono (obligatorio).
 - **ciudad**: String (obligatorio).
 - **notas**: String (opcional).
 - **Validación**: Si el CSV contiene datos erróneos (ej. un texto en el campo teléfono), el backend debe retornar un error detallado.
3. **Control de Acceso (RBAC)**:
- Cualquier usuario autenticado puede cargar y descargar archivos.
 - **Solo el rol admin** tiene permisos para eliminar registros de la tabla de documentos cargados.
-

Requerimientos del Frontend (1.5h aprox.)

La interfaz debe ser intuitiva y funcional, desarrollada en **Vue 3 (Option API)**.

1. **Vistas de Acceso**:
 - **Registro**: Formulario con validación de coincidencia de contraseñas y selección de rol.
 - **Login**: Almacenamiento seguro del JWT.
 2. **Vista Principal (Dashboard)**:
 - **Carga de Archivos**: Implementar un área de *Drag & Drop* o selector de archivos para subir el CSV.
 - **Tabla de Datos**: Mostrar un listado de los documentos procesados con:
 - Nombre del documento.
 - Usuario que lo cargó.
 - Fecha de carga.
 - Numero de registros en la tabla.
 - **Acciones**:
 - Botón para **Descargar** el documento original.
 - Botón para **Eliminar** (visible/habilitado solo para usuarios con rol **admin**).
 3. **Feedback**: Mostrar un alert o notificación clara si la carga falla por errores de validación en los datos del CSV.
-

Infraestructura y Despliegue

- El proyecto del backend debe estar en un contenedor de docker-Compose file junto a la base de datos.
 - El proyecto del frontend es apreciable si se puede ejecutar en un contenedor de docker(no obligatorio).
-

Criterios de Evaluación

- **Dominio de la herramienta:** Correcta implementación de Option API en Vue 3 y middlewares en Express.
 - **Seguridad:** Manejo correcto de JWT.
 - **Validación de Datos:** Capacidad para procesar y validar el contenido de un archivo externo (CSV).
 - **Dockerización:** Facilidad para levantar el entorno sin configuraciones manuales adicionales.
 - **Limpia de código:** Estructura de carpetas organizada y código legible.
-

Anexos

