**Sistema de Registro de Visitantes**

**Este proyecto es una aplicación web completa para gestionar visitas en una empresa o institución. Permite registrar visitas de manera manual o autogestionada, y administrar usuarios, empleados y sedes. Está diseñada para dos tipos de usuario (administrador y estándar), e incluye funcionalidades móviles.**

**Requisitos del Proyecto**

**Roles del sistema:**

* **Administrador: acceso completo a todas las funciones.**
* **Usuario Estándar: acceso limitado a funcionalidades esenciales.**
* **Visitante (sin cuenta): puede registrar una visita desde un enlace público (autogestión).**

**Funcionalidades Generales**

**1. Inicio de Sesión**

* **Ruta: /login**
* **Campos: correo, contraseña**
* **Seguridad: autenticación JWT**
* **Redirección según rol:**
  + **Administrador: /admin/dashboard**
  + **Estándar: /usuario/dashboard**

**2. Panel de Administración**

* **Ver visitas (todas)**
* **Ver visitas activas**
* **Consultar visitas finalizadas**
* **Ver y aprobar usuarios pendientes**
* **Ver y administrar empleados**
* **Añadir nueva visita**
* **Cambiar estado de visitas (finalizar, rechazar)**
* **Acceder al enlace de visita autogestionada**
* **Aprobar/rechazar visitas autogestionadas**

**3. Panel de Usuario Estándar**

* **Ver sus propias visitas**
* **Ver visitas activas**
* **Añadir visita**
* **Cambiar estado de visitas**
* **Acceder al enlace de autogestión**
* **Aprobar/rechazar visitas autogestionadas**
* **Restricciones: no puede ver empleados ni visitas antiguas ajenas**

**4. Registro Manual de Visitas**

* **Ruta: /api/visitas**
* **Método: POST**
* **Campos requeridos:**
  + **nombre\_visitante**
  + **cedula**
  + **empresa**
  + **motivo**
  + **foto**
  + **tipo\_registro: 'manual'**
  + **estado: por defecto 'activa'**
  + **empleado\_id, sede\_id, usuario\_id**

**5. Estado de Visita**

* **Puede ser:**
  + **activa**
  + **finalizada**
  + **rechazada**
  + **pendiente (solo en autogestión)**

**6. Autogestión de Visitas (sin login)**

* **Ruta: /autogestion**
* **Desde el móvil, cualquier visitante puede:**
  + **Completar un formulario con sus datos**
  + **Enviar una solicitud de visita autogestionada**
* **Almacena como tipo\_registro = autogestionado y estado = pendiente**
* **Aparece en dashboards para aprobación o rechazo**

**7. Gestión de Usuarios (solo admin)**

* **Ver usuarios pendientes (activo = false)**
* **Aprobar usuarios (activar cuenta)**
* **Eliminar usuarios**

**8. Gestión de Empleados (solo admin)**

* **Ver empleados**
* **Añadir nuevos empleados**
* **Eliminar empleados**

**Requisitos Técnicos**

**Backend (Node.js + Express + PostgreSQL)**

* **Autenticación JWT**
* **Bcrypt para hash de contraseñas**
* **Validaciones de campos**
* **Rutas protegidas por middleware**

**Frontend (React.js)**

* **Interfaz responsive**
* **Paneles diferenciados por rol**
* **Formularios de registro y gestión**
* **Botones y vistas para cambiar estados de visita**
* **Vista especial para autogestión (móvil)**

**Base de Datos**

**Tablas requeridas:**

* **usuarios: id, nombre, correo, contraseña, rol, activo, sede\_id**
* **visitas: id, nombre\_visitante, cedula, empresa, motivo, foto, tipo\_registro, estado, fecha\_ingreso, fecha\_salida, empleado\_id, sede\_id, usuario\_id**
* **empleados: id, nombre, cargo, sede\_id**
* **sedes: id, nombre**

**Flujo de Trabajo Esperado**

1. **Inicio de sesión (admin o estándar)**
2. **Carga del panel según el rol**
3. **Registro de visitas desde panel o enlace de autogestión**
4. **Visualización y administración de visitas y usuarios**
5. **Aprobación/rechazo de visitas autogestionadas**
6. **Finalización manual de visitas activas**

**Consideraciones Finales**

* **Código debe estar organizado por carpetas (controllers, routes, models, middleware)**
* **Código frontend en React debe tener rutas protegidas y navegación según rol**
* **Se debe garantizar la seguridad y validación de datos en cada paso**
* **Las funciones deben estar bien documentadas para facilitar mantenimiento**