Para habilitar la **creación de bufetes desde el dashboard del Superadmin** y enlazarlo correctamente al flujo de login/roles, ya tengo el panorama completo.

Lo que necesito saber o confirmar antes de implementarlo:

**1. Dependencias y datos mínimos para crear un bufete**

Según tu modelo BufeteJuridico, los datos que requiere el formulario son:

1. **nombre** *(obligatorio)*
2. **dirección** *(obligatorio)*
3. **teléfono** *(obligatorio)*
4. **NIT** *(opcional, validable)*
5. **correo** *(opcional, puede repetirse)*
6. **forma\_contacto** *(opcional)*
7. **plan\_id** *(debe existir al menos 1 plan cargado previamente con el comando init\_dev\_db o similar)*
8. **feature flags**:
   * usa\_control\_inventario (boolean)
   * usa\_auditoria (boolean)
   * usa\_digital (boolean)

Con esto ya podemos crear un bufete, y automáticamente quedará disponible para asignar **usuarios**.

**2. Rutas y templates que propongo**

Usando Flask con Blueprints para la sección de administración:

app/admin/routes.py

* /admin/bufetes → listado de bufetes
* /admin/bufetes/nuevo → formulario agregar\_bufete.html
* /admin/bufetes/<int:id>/editar → formulario editar\_bufete.html

Dashboard del **superadmin** mostrará un botón **“➕ Crear Bufete”** si el rol del usuario logueado es Superadmin.

**3. Flujo de login + creación de bufetes**

1. **Superadmin** hace login.
2. Si es su **primer acceso** y no hay bufetes creados:
   * Redirigirlo directamente a /admin/bufetes/nuevo.
3. Si hay bufetes creados:
   * Redirigirlo a su **dashboard** con acceso a:
     + Listado de bufetes (listado\_bufetes.html)
     + Botón de **crear bufete**.
4. Una vez creado el bufete:
   * Se podrán crear usuarios asociados a ese bufete.

**4. Recomendaciones de implementación**

* **Evitar errores UTF-8 / nombres con espacios:**
  + Normalizar los nombres de bufetes (ej: "Mi Bufete Jurídico" → "Mi\_Bufete\_Juridico" como slug interno).
  + Usar secure\_filename de Flask si vas a manejar uploads.
* **Relaciones SQLAlchemy sin circulares:**
  + BufeteJuridico → usuarios = relationship("Usuario", back\_populates="bufete")
  + Usuario → bufete = relationship("BufeteJuridico", back\_populates="usuarios")
  + Crear tablas en orden: Plan, BufeteJuridico, Usuario para evitar problemas de migración.
* **Formulario protegido:**
  + CSRF activado con Flask-WTF.
  + Validar campos obligatorios y longitud.
  + Validar NIT con función en app/utils/validators.py.

**5. Próximo paso de desarrollo**

1. Crear el blueprint admin con las rutas anteriores.
2. Crear los templates usando los que ya tenías como referencia.
3. Integrar el botón en el **navbar del Superadmin**.
4. Agregar prueba inicial:
   * Crear un bufete desde cero.
   * Confirmar que se ve en el listado.
   * Crear usuario ligado a este bufete.

Si quieres, puedo generarte **ya mismo** un documento .md completo con:

* Explicación del flujo
* Dependencias
* Plantilla de rutas y formularios
* Mejoras sugeridas para evitar errores futuros
* Mini plan de implementación para Git

Así lo guardamos en docs/ y lo versionamos con Git.

¿Quieres que te lo genere de una vez?