¡Bendecido día para ti también, Giancarlo! Aquí está el **estado actual** y el plan de trabajo **actualizado al 08/08/25**, retomando desde la órbita exacta en que nos quedamos anoche:

**✅ ESTADO ACTUAL**

| **Área** | **Estado actual** |
| --- | --- |
| **Login** | Funcional. Flask-Login operativo. Autenticación del Superadmin verificada. |
| **Dashboard Superadmin** | Activo. Mostrando estructura inicial. Blueprint registrado correctamente. |
| **Modelos (models/)** | Refactorizados, campos renombrados, importaciones corregidas. |
| **Base de Datos** | Sin cambios pendientes. Alembic confirmó que está sincronizada. |
| **Git** | Rama feature/superadmin activa, tag login-superadmin-v1 ya creado y subido. |

**🚀 PLAN DE TRABAJO BLOQUE 3: Superadmin + Bufetes + Usuarios**

Este bloque lo dividimos en **etapas**. Estás a punto de entrar en la **Etapa 2**.

**🔢 ETAPAS ACTUALES**

| **Etapa** | **Objetivo** |
| --- | --- |
| 1. ✅ | Login funcional para Superadmin. Tag: login-superadmin-v1 |
| 2. ⬅️ | CRUD de Bufetes desde el panel del Superadmin |
| 3. ⬜ | CRUD de Usuarios (Notario, Procurador, etc.) desde bufetes |
| 4. ⬜ | Redirección post-login según rol |
| 5. ⬜ | Auditoría mínima (registro de login, IP, etc.) |
| 6. ⬜ | Decoradores de acceso (roles\_required, etc.) |
| 7. ⬜ | UI limpia para gestión de bufetes y usuarios |

**🔁 ¿QUÉ CORRESPONDE AHORA?**

**🟦 Etapa 2: CRUD de Bufetes para el Superadmin**

Objetivo: El superadmin podrá:

* Ver lista de bufetes
* Crear un nuevo bufete (con plan y configuración básica)
* Editar un bufete existente
* Eliminarlo (borrado lógico)
* Todo esto desde su dashboard

**✅ Requisitos ya cumplidos para avanzar:**

* ✔️ Blueprint superadmin\_bp ya existe
* ✔️ Modelo BufeteJuridico listo y funcional
* ✔️ Seed inicial con al menos 1 bufete y 1 plan
* ✔️ Login Superadmin operativo

**📌 PASOS SUGERIDOS PARA HOY (08/08/25)**

1. **Crear vistas (routes) del superadmin:**
   * /superadmin/bufetes (listado)
   * /superadmin/bufetes/nuevo
   * /superadmin/bufetes/<id>/editar
   * /superadmin/bufetes/<id>/eliminar
2. **Crear formularios WTForms (forms.py)**
   * CrearBufeteForm y EditarBufeteForm
3. **Diseñar templates:**
   * templates/superadmin/bufetes.html
   * form\_bufete.html (parcial reutilizable)
4. **Conectar con modelo BufeteJuridico**
5. **Verificar permisos:**
   * Solo rol == SUPERADMIN puede acceder a esas rutas (usando @roles\_required o equivalente)

**🧠 OPCIONAL PARA HOY**

* Agregar campo de búsqueda por nombre del bufete
* Agregar paginación o filtros si hay muchos registros

**¿Empezamos?**

¿Te gustaría que preparemos de una vez los archivos para el CRUD de bufetes y sus formularios, o prefieres que vayamos paso a paso (primero rutas, luego formularios)?

Perfecto. Vamos a avanzar ordenadamente con la mejor estrategia para la **Etapa 2 del Bloque 3: CRUD de Bufetes desde el Superadmin**. Como ya tenemos:

* Login funcional
* Dashboard inicial del superadmin
* Modelo BufeteJuridico operativo
* Base de datos actualizada y sin errores de migración

**✅ Etapas para implementar el CRUD de Bufetes (Superadmin)**

Aquí el desglose paso a paso:

**Paso 1: Definir rutas en el Blueprint superadmin\_bp**

Vamos a agregar las siguientes rutas:

* /superadmin/bufetes → Ver listado de bufetes
* /superadmin/bufetes/nuevo → Crear nuevo bufete
* /superadmin/bufetes/<int:id>/editar → Editar bufete existente
* /superadmin/bufetes/<int:id>/eliminar → Borrado lógico del bufete

Estas rutas deben:

* Estar protegidas con @login\_required y @roles\_required(['SUPERADMIN'])
* Renderizar templates limpios, separados en templates/superadmin/bufetes/

**Paso 2: Crear archivo de rutas routes\_bufetes.py dentro de app/superadmin/**

Esto ayuda a mantener las rutas organizadas (como ya hicimos con dashboard).

**Paso 3: Crear templates base para cada vista**

* listar\_bufetes.html
* form\_bufete.html
* Usar base.html como layout
* Mostrar validaciones y feedback

**Paso 4: Formularios WTForms**

* Crear forms\_bufetes.py con campos:
  + nombre\_bufete
  + responsable
  + direccion
  + etc. (según el modelo)
* Agregar validaciones básicas

**Paso 5: Lógica de control**

* Insertar y actualizar correctamente usando SQLAlchemy
* Flash messages (flash("Bufete creado exitosamente", "success"))
* Redirección con redirect y url\_for

**Paso 6: Prueba funcional desde interfaz**

* Crear un bufete nuevo
* Editarlo
* Eliminarlo (lógicamente)
* Ver que no aparecen los eliminados
* Revisar desde flask shell si todo se guarda bien

**Paso 7: Confirmar avance con Git**

* Commit con mensaje: ✨ CRUD de bufetes operando desde Superadmin
* Tag: superadmin-crud-bufetes-v1

¿Listo para iniciar con el Paso 1 (rutas y blueprint)? Te genero de inmediato el archivo routes\_bufetes.py si confirmás.