Claro, aquí tienes un **documento completo** en estilo técnico-profesional, que combina **tu prompt**, **mis recomendaciones**, **buenas prácticas**, y un **plan de trabajo general** para que sirva de guía para el Bloque 3 de tu proyecto **Software Notarios**.

Lo redacté para que puedas **guardarlo como .md o .docx**, y usarlo como **prompt inicial mañana** para seguir avanzando sin perder contexto.

**Proyecto: Software Notarios | Bloque 3 - Planes, Bufetes y Roles/Usuarios**

**Usuario:** Giancarlo Figueroa ([pinedayfigueroa@gmail.com](mailto:pinedayfigueroa@gmail.com))  
**Fecha:** 05/08/2025  
**Versión del documento:** 1.0

**Contexto**

* Desarrollo del software **Software Notarios**, orientado a la gestión integral de bufetes notariales.
* El sistema debe permitir **gestión multi-bufete**, con roles y permisos granulares para Superadmin, Admin Local, Notario, Procurador y Asistente.
* Se busca implementar **login seguro**, asignación de usuarios a bufetes y un flujo controlado de creación de planes y bufetes.
* La restauración de un Git anterior provocó pérdida de datos y cambios en modelos; se requiere reorganizar para evitar errores previos (como circulares, UTF-8, y campos desincronizados).

**Alcance del Bloque 3**

1. **Gestión de Planes**
   * Tabla planes definida y cargada con seeds.
   * Soporte para:
     + Plan Demo
     + Plan Profesional
     + Plan Ilimitado
     + Planes personalizados (bandera es\_personalizado)
   * Validación de límites: notarios, procuradores, asistentes, documentos y almacenamiento.
   * Asociar cada BufeteJuridico a un Plan.
2. **Gestión de Bufetes Jurídicos**
   * Tabla bufetes\_juridicos con datos:
     + Nombre (respeta mayúsculas y acentos según ingresado por usuario)
     + Dirección, teléfono, NIT, correo, forma de contacto
     + Feature flags:
       - usa\_control\_inventario
       - usa\_auditoria
       - usa\_digital
   * Relación 1:N con Usuario.
   * Relación N:1 con Plan.
   * CRUD desde el dashboard del **Superadmin**:
     + Crear bufetes
     + Editar información
     + Activar/desactivar bufetes
     + Asignar plan
   * Validaciones:
     + NIT opcional pero validable si se ingresa.
     + Nombre único para evitar duplicidades internas.
3. **Gestión de Usuarios y Roles**
   * Modelo Usuario centralizado, con extensión para roles específicos (NotarioDatos, ProcuradorDatos).
   * Roles definidos en RolUsuarioEnum:
     + SUPERADMIN, ADMIN\_LOCAL, NOTARIO, PROCURADOR, ASISTENTE
   * Atributos principales:
     + username (único, estilo sugerido B1\_N1\_jperez para trazabilidad)
     + nombre\_completo
     + correo (opcional, no único en esta etapa)
     + telefono
     + password\_hash con werkzeug.security
     + estado (activo/inactivo/bloqueado)
     + ultimo\_login, intentos\_fallidos
   * Un usuario solo puede pertenecer a un bufete a la vez (para trazabilidad).
4. **Proceso de Login y Seguridad**
   * Implementar con **Flask-Login** + **Werkzeug Security**:
     + Login y logout seguro
     + Password hash PBKDF2 con sal
   * Flujo:
     + Superadmin inicia sesión con contraseña inicial 123456 (hash en seed).
     + Primer login fuerza cambio de contraseña (con validación de complejidad).
     + Cada usuario es redirigido al **dashboard de su bufete** según rol.
   * Control de acceso:
     + @login\_required en rutas protegidas
     + @roles\_required() decorador custom para filtrar por rol
   * Auditoría mínima desde el inicio (quién, cuándo, IP):
     + Registro de intentos fallidos
     + Bloqueo tras X intentos
     + Logs en tabla de auditoría si el feature flag está activo.
5. **Seeds iniciales (CLI)**
   * Crear planes base
   * Crear bufete principal
   * Crear superadmin y un notario principal asignado a ese bufete
   * Proceso flask seed-cli init renovado:
     + Mensajes claros con emojis ✅⚠️❌
     + Hash real para passwords
     + Un solo commit final para eficiencia
6. **Dashboard inicial**
   * Superadmin:
     + Ver bufetes
     + Crear bufetes nuevos
     + Asignar planes
   * Notario Admin Local:
     + Ver su bufete, usuarios y documentos
   * Navbar dinámico:
     + Nombre real del usuario
     + Rol y bufete actual
     + Botón logout y ayuda

**Plan de Trabajo Técnico**

1. **Revisión y limpieza de modelos**
   * Consolidar Usuario como modelo principal
   * NotarioDatos / ProcuradorDatos como tablas extendidas
   * Evitar dependencias circulares usando:
     + back\_populates
     + Importaciones diferidas si es necesario
2. **Refactor del seed CLI (app/cli.py)**
   * Crear cli\_seed\_refactor.py con:
     + Planes
     + Bufete principal
     + Superadmin y notario
     + Passwords con hash
   * Generar logs de lo creado o ya existente
3. **Login seguro**
   * Crear blueprint auth con:
     + login.html, logout y register (solo superadmin)
   * Implementar Flask-Login con UserMixin en Usuario
   * Decoradores para control de roles
4. **Dashboard Superadmin**
   * Permitir crear bufetes desde el login inicial
   * Listar planes, bufetes y usuarios
   * Acceso condicional según RolUsuarioEnum
5. **Pruebas y validación**
   * Test de seeds (flask seed-cli init)
   * Test de login/logout y redirecciones
   * Test de creación de bufetes y asignación de usuarios
   * Verificación de auditoría básica

**Buenas Prácticas y Consideraciones**

* Mantener nombres de bufetes **tal cual** ingresados, para documentos y reportes.
* Evitar relaciones circulares:
  + Importaciones modulares
  + back\_populates siempre que haya 1:1 o 1:N bidireccional
* Preparar desde ya las tablas de auditoría para quienes activen el módulo.
* Establecer políticas de contraseñas y cambios periódicos.
* Logs claros durante seeds para evitar confusiones si se pierde información.
* Dejar hooks para que en versiones futuras el **chatbot** de asistencia guíe a usuarios en la creación de bufetes, usuarios y manejo de documentos.

**Próximos pasos inmediatos**

1. Revisar modelos Usuario y BufeteJuridico para consolidación definitiva.
2. Generar cli\_seed\_refactor.py y probar seeds iniciales sin errores.
3. Implementar blueprint auth y login funcional con Flask-Login.
4. Habilitar dashboard Superadmin con opción de **crear bufetes**.
5. Integrar auditoría mínima y rutas protegidas por rol.

¿Quieres que ahora genere este documento directamente como **docs/PLAN\_BLOQUE3\_BUFETES\_ROLES.md** para que quede listo para tu Git y como base del próximo prompt?