🚀✨ ¡Perfecto, navegante!  
Empezamos ahora mismo.

Lo haré **en bloques ordenados**.  
👉 *Por favor, ve copiando cada bloque a tu Word/Markdown/TXT* antes de que sigamos, así nada se pierde.  
Cuando diga **“✔️ Siguiente bloque?”**, respóndeme *“sí”* y te mando el siguiente.

**📜 🧭 Bitácora & Blueprint: Software\_Notarios**

**Autor:** Tars-90 & Navegante (Giancarlo)  
**Fe:** En el nombre de Dios 🙏✨

**☑️ 1. Visión general y objetivo**

Desarrollar un software pensado *por notarios, para notarios*, que combine:

* Gestión de bufetes jurídicos
* Manejo de notarios, procuradores, asistentes
* Automatización de documentos (actas, escrituras, poderes, avisos)
* Librería de cláusulas y puntos reutilizables
* Control, auditoría, dashboards
* Feature flags para activar o desactivar funciones por bufete
* Planes de contratación escalables
* Exportación de documentos (Word, PDF)
* Machine Learning futuro basado en datos históricos
* Soporte a firma electrónica y timbres digitales
* CRM / ERP básico para crecimiento
* Flexibilidad multi-bufete y multirrol
* Documentación técnica y funcional clara, para mantenimiento y crecimiento sostenible

**☑️ 2. Roles y usuarios**

* **Superadmin (central):**
  + Crea bufetes, asigna planes y activa feature flags
  + Puede hacer backups, ver auditorías globales
* **Administrador interno de bufete:**
  + Crea/edita notarios, procuradores, asistentes
  + Asigna tareas y gestiona usuarios internos
* **Notarios:**
  + Redactan y gestionan documentos
  + Firman electrónicamente (cuando se implemente)
* **Procuradores:**
  + Llevan trámites y seguimiento en entidades externas
* **Asistentes:**
  + Apoyo administrativo (citas, borradores, adjuntar documentos)

⚖️ Todos con *auditoría, trazabilidad y control de accesos*.

**☑️ 3. Planes y feature flags**

**Planes (Starter, Pro, Enterprise, etc.):**

* Número máximo de:
  + Escrituras
  + Actas
  + Notarios
  + Procuradores
  + Asistentes
* Límites mensuales (max\_actas\_mensuales, etc.)
* Posibilidad de hacer promociones o ampliaciones

**Feature flags:**

* Activación/desactivación de:
  + Firma electrónica
  + Timbres electrónicos
  + CRM
  + ERP
  + Storage externo (documentos, PDFs, etc.)
  + Machine Learning (futuro)
* Se asocian a cada bufete por id.

**☑️ 4. Documentos y librería**

* Librería central de cláusulas y puntos (con bulk upload en CSV/Excel)
* Librería de documentos base (actas, escrituras, poderes, avisos)
* Plantillas personalizables
* Posibilidad de importar documentos existentes (PDF, Word)
* Campos dinámicos y generación automática
* Guardar imágenes (DPI, etc.) y adjuntos por documento

**☑️ 5. Auditoría & dashboards**

* Logs de acciones (crear, editar, borrar lógico, login)
* Trazabilidad completa por documento
* Reportes:
  + Documentos creados por mes/año
  + Documentos pendientes vs. firmados
  + Uso por usuario y dispositivo
* Dashboards dinámicos para bufetes y superadmin

**✔️ ¿Listo para el siguiente bloque?**  
Escribe: **“sí”** y te mando el bloque 2 (sobre storage, ML, login, firma electrónica, etc.).

¡Perfecto, navegante! 🌊⚓  
Aquí va el **📦 Bloque 2**: detalles técnicos, ideas avanzadas y visión a futuro.  
(Copia todo este bloque en tu Word o notas, y avísame para seguir con el próximo).

**📦 6. Storage, ML & crecimiento**

✅ **Storage:**

* Almacenamiento en nube (o hosting dedicado)
* Costos asociados según plan
* Control de adjuntos:
  + PDFs de documentos oficiales
  + Imágenes (DPI anverso/reverso)
  + Informes del RGP, timbres, etc.

✅ **Machine Learning (fase 2):**

* Datos históricos de documentos y cláusulas
* Sugerencias automáticas:
  + Cláusulas similares usadas antes
  + Errores comunes detectados
* Dashboard predictivo: “qué documentos usarás el próximo mes”

✅ **Backup & recovery:**

* Copias automáticas configurables
* Restore manual por superadmin o soporte

**🔐 7. Login & control de dispositivos**

* Usuario y contraseña con email verificado
* Login simultáneo permitido los primeros meses (fase de adopción)
* Después, control y reporte de:
  + Usuarios conectados
  + Dispositivos
  + Ubicaciones (opcional)

📊 **Reportes de uso** para el bufete:

* Número de accesos
* Documentos editados
* Descargas y exportaciones

**✒️ 8. Firma electrónica & timbres digitales**

* Notario puede tener firma electrónica asociada (proveedor, fecha de emisión, expiración)
* Guardar metadatos de la firma
* Timbres digitales: integrables con SAT/RGP (cuando estén disponibles)
* Permitir que cada bufete decida activar/contratar

**🏛 9. Datos de personas (clientes, partes en documentos)**

Campos recomendados:

* Nombre completo
* DPI / documento oficial
* Estado civil (soltero / casado; opcional viudo/divorciado como texto)
* Fecha de emisión & lugar
* Nacionalidad
* País de nacimiento
* Sexo / género
* Vecindad (domicilio actual)
* Adjuntar imágenes del DPI
* Fecha de vencimiento (opcional, para validación interna)

⚙️ *Muchos opcionales, pero muy útiles para borradores o librería.*

**📌 10. Documentación técnica & funcional**

* Generada con:
  + **Sphinx** → Documentación técnica en HTML/PDF
  + **MkDocs** → Manual de usuario online
* Diagrama ER:
  + **ERAlchemy2** → exporta a PDF o PNG
* Docstrings en modelos y funciones (útil para Sphinx)
* Tooltips y ayuda contextual para el usuario
* PDF descargable del manual

**✔️ ¿Listo para el bloque 3?**  
Ahí te contaré sobre:  
✅ Migraciones (init, migrate, upgrade)  
✅ Seed inicial  
✅ CLI para developers  
✅ Documentación de VS Code + control con Git

Pon “**sí**” y seguimos.

¡Excelente, navegante! 🌊⚓  
Aquí tienes el **📦 Bloque 3**: parte técnica, CLI, migraciones, seed, y organización del proyecto.

**🛠 11. Migraciones y flujo completo (Big Bang & cambios)**

**Fase inicial (Big Bang):**  
1️⃣ init\_dev\_db.bat

* Borra la base de datos software\_notarios
* Crea nueva DB en UTF-8 (usando template0)
* Ejecuta:
  + flask db init → crea carpeta migrations
  + flask db migrate -m "big bang initial creation"
  + flask db upgrade → aplica cambios
  + flask seed-cli init → crea plan Starter, bufete principal, superadmin y notario inicial

**Fase cambios incrementales:**

* Editas modelos en core.py
* Ejecutas:

python -m flask db migrate -m "Descripción del cambio"

python -m flask db upgrade

🔄 *Esto evita recrear todo; aplica solo el delta.*

**🌱 12. Seed inicial con CLI**

Creamos en app/cli.py:

* Comando seed-cli:
  + Crea plan Starter si no existe
  + Crea bufete principal
  + Crea notario inicial
  + Crea superadmin

✅ *Permite reiniciar DB rápido en desarrollo.*

**🧰 13. CLI para developers & auto\_migrate (idea)**

Podemos crear script auto\_migrate.bat (ejemplo):

echo Ejecutando migrate...

python -m flask db migrate -m "auto migrate"

echo Aplicando upgrade...

python -m flask db upgrade

*O crear comando custom en cli.py.*

**🧩 14. Organización del proyecto (estructura)**

/app

/models.py o core.py ← modelos SQLAlchemy

/cli.py ← seed y comandos CLI

/config.py ← config por entorno

/\_\_init\_\_.py ← create\_app, init db, migrate

/migrations ← generada por Flask-Migrate

/docs

/erd.pdf ← diagrama ER

/tests (opcional)

/init\_dev\_db.bat ← Big Bang inicial

/requirements.txt

**🧪 15. Configuración de VS Code**

* Instalar:
  + Python extension
  + Pylance
  + SQLTools (para ver DB)
  + GitLens (para ver cambios)
* Configurar:

{

"python.defaultInterpreterPath": "c:\\Users\\Usuario\\Mi unidad\\Software\_Notarios\\swnot\\Scripts\\python.exe",

"terminal.integrated.env.windows": {

"PYTHONUTF8": "1"

}

}

**🔄 16. Control de cambios con Git**

Cada vez que:

* Crees migraciones
* Cambies modelos o CLI
* Generes documentación (/docs)

git add .

git commit -m "Mensaje claro del cambio"

git push

**¿Listo para el 📦 Bloque 4?**  
Hablaremos de:  
✅ Sphinx + MkDocs  
✅ ERAlchemy2 (diagrama)  
✅ Tips finales para tener documentación técnica + funcional viva

Pon “**sí**” y seguimos.

¡🚀 Perfecto, navegante!  
Aquí viene el **📦 Bloque 4**: documentación técnica, diagramas, manual de usuario y cultura DevOps que hemos ido creando juntos.

**📑 17. Documentación técnica y funcional**

**🧰 a) Sphinx**

Para documentar la parte técnica (modelos, CLI, módulos):  
1️⃣ Instala:

pip install sphinx

2️⃣ Inicializa:

sphinx-quickstart docs

3️⃣ En docs/conf.py:

* Agrega ../app a sys.path para que detecte módulos
* Cambia idioma a 'es' si quieres en español

4️⃣ Genera doc automática:

sphinx-apidoc -o docs/source ../app

5️⃣ Compila HTML:

make html

Resultado: /docs/build/html/index.html

**📚 b) MkDocs (opcional)**

Ideal para manual de usuario funcional:

pip install mkdocs mkdocs-material

* mkdocs.yml define estructura
* Docs en docs/

mkdocs serve

**📊 18. ERAlchemy2 (diagrama ER)**

Ya lo probaste:

eralchemy2 -i "postgresql://postgres:12345678@localhost:5432/software\_notarios" -o docs/erd.pdf

🧩 *Importante:*

* Diagrama vive en /docs
* Regenera cada vez que cambien modelos
* Súbelo a Git

**🧠 19. Manual de usuario (idea)**

Desde el inicio:

* Capturas de pantalla
* Explicaciones paso a paso
* Incluye tooltips y ejemplos
* Se va armando junto con desarrollo

**⚙ 20. Documentación viva = + valor**

* Docstring en modelos y CLI (explican qué hace cada clase / método)
* Readme claro en /docs/README.md o en raíz
* Descripción del flujo de Big Bang:
  + init\_dev\_db.bat
  + migraciones
  + seed
  + run

**🏗 21. Cultura DevOps light**

* Cada cambio documentado → commit
* Cada commit relacionado a:
  + cambio en modelo
  + cambio en CLI
  + migración generada
  + doc actualizada

✅ Esto permite que cualquiera pueda reconstruir el proyecto.

**✏ 22. Próximos pasos sugeridos**

✅ Pulir modelo Personas (dpi, nacionalidad, etc.)  
✅ Prototipo frontend (admin, login, crear bufete, etc.)  
✅ Dashboards iniciales  
✅ Machine learning futuro: preparar librerías / bulk

**✨ 23. Resumen técnico**

* Base: Flask + SQLAlchemy
* Migraciones: Flask-Migrate / Alembic
* CLI: seed inicial
* Doc técnica: Sphinx
* Doc funcional: MkDocs (opcional)
* Diagrama ER: ERAlchemy2
* Control de cambios: Git
* Feature flags, roles, planes: pensados desde el modelo

🙏 **Bendición final del bloque:**

“Lo que hagas, hazlo de corazón, como para el Señor” ✨

¿Listo para que te arme **el bloque final** con comandos resumidos + guías + checklist para Big Bang?  
Pon "**sí**" y te lo mando, navegante. 🌊⚓

🚀✨ **Perfecto, navegante Tars-90 reportándose con el 📦 Bloque final**:  
👉 *Checklist técnico + comandos + pasos Big Bang + mini roadmap*  
(💡 Copia/guarda en Word, Notion o Markdown para tenerlo siempre)

**✅ Bloque 5: Big Bang / Checklist + comandos**

**🛠 1) Entorno virtual**

python -m venv swnot

swnot\Scripts\activate

pip install -r requirements.txt

**⚙ 2) Base de datos**

**PostgreSQL 17+**

* DB name: software\_notarios
* Encoding: UTF8
* Owner: postgres  
  *(puedes crear otro user seguro después)*

**🌱 3) Big Bang: init\_dev\_db.bat**

:: Ejecuta todo:

:: - drop/create bdd

:: - flask db init / migrate / upgrade

:: - seed inicial

init\_dev\_db.bat

📦 Crea:

* Plan Starter
* Bufete principal
* Notario principal
* Super admin

**🧬 4) Cambios futuros**

Para cambios en modelos:

python -m flask db migrate -m "Describe change"

python -m flask db upgrade

**🧰 5) Diagrama ER**

eralchemy2 -i "postgresql://postgres:12345678@localhost:5432/software\_notarios" -o docs/erd.pdf

**📑 6) Documentación técnica (Sphinx)**

pip install sphinx

cd docs

sphinx-quickstart

sphinx-apidoc -o source ../app

make html

**📚 7) Manual usuario (opcional, MkDocs)**

pip install mkdocs mkdocs-material

mkdocs new .

mkdocs serve

**🧠 8) Recomendación diaria**

* pip freeze > requirements.txt
* git add .
* git commit -m "feat: cambios X"
* git push

**📊 Mini roadmap *(futuro)***

✅ Probar login / CRUD  
✅ Crear planes, feature flags, roles  
✅ Dashboards iniciales  
✅ Subir docs  
✅ Probar diagrama ER  
✅ Empezar frontend

**✨ Resumen ultra-corto**

✅ init\_dev\_db.bat  
✅ eralchemy2 → ER  
✅ sphinx → doc técnica  
✅ mkdocs → manual  
✅ pip freeze + git

**🙏 Bendición final**

“Todo lo puedo en Cristo que me fortalece” ✝️  
Con visión, orden y documentación, este proyecto va a bendecir a muchos 🙌

💡 **¿Quieres que te empaquete TODO esto en un README.md completo + carpeta /docs lista?**  
Responde "**sí**" y lo armo navegante ⚓🚀