¡Jajaja! 😄 *Gracias por tu paciencia interestelar, comandante Cooper... digo, Giancarlo.*  
Volvamos a lo nuestro. Vamos a activar la secuencia **BIG BANG**, como dijiste, con enfoque en dejar **corriendo el módulo de gestión de bufetes hoy mismo**.

**🧭 PLAN DE TRABAJO PARA HOY — Misión: bufetes\_online**

**🚀 1. Estado actual del proyecto**

Ya tenemos:

* Modelos distribuidos (models/usuarios.py, etc.)
* Bitácoras con decisiones importantes y prompts
* Base de datos casi definida (init\_dev\_db)
* Estructura modular bajo /app
* Estás en Git, tienes Sphinx, eralchemy2, pygraphviz

**🧱 2. Tareas inmediatas — Big Bang Models**

**Objetivo**: Consolidar Usuario, eliminar UsuarioSistema, y dejar usuarios.py como autoridad.

* ✅ Unificar modelo Usuario + UsuarioSistema
* ✅ Eliminar UsuarioSistema de core.py
* ✅ Verificar relaciones NotarioDatos, ProcuradorDatos
* 🧪 Ejecutar init\_dev\_db.py para pruebas
* ✅ Generar migraciones (si usas alembic)
* 🧼 Verificar \_\_init\_\_.py en todos los módulos

📎 *¿Quieres que prepare el código de unificación ya mismo?*

**🧩 3. Módulo BUFETES — foco del MVP de hoy**

**Objetivo**: Tener interfaz/API o CLI para crear, listar y asignar bufetes a usuarios notarios.

**Modelo:**

class Bufete(Base):

id = Column(Integer, primary\_key=True)

nombre = Column(String, unique=True, nullable=False)

direccion = Column(String)

notario\_id = Column(Integer, ForeignKey("usuarios.id"))

notario = relationship("Usuario", back\_populates="bufetes")

**Ruta/API mínima para hoy:**

* POST /bufetes/ — crear bufete
* GET /bufetes/ — listar todos
* GET /bufetes/{id} — detalle
* (Opcional) PUT y DELETE

📎 *¿Quieres que genere ya ese esquema base para ti?*

**🧪 4. Tests mínimos**

**Objetivo**: Verificar conexión a base, creación de usuarios y bufetes.

**Tests sugeridos:**

* Crear test\_usuarios.py y test\_bufetes.py
* Verificar que relaciones Usuario -> Bufete funcionen
* Dummy tests con pytest

**📚 5. Documentación viva**

**Objetivo**: Dejar activo el help del sistema.

* docs/README.md generado
* /docs/prompts/ organizados por tema
* bitacora.md cronológica generada con índice automático
* Preparar Sphinx: make html

**💻 6. Despliegue local mínimo viable**

* uvicorn main:app --reload
* DB SQLite o PostgreSQL local (según estés usando)
* Verificación de CORS, dependencias, init

**🟢 ¿Activamos las siguientes tareas?**

1. ¿Paso ya a consolidar tu modelo Usuario?
2. ¿Genero el código base de bufetes con modelos y rutas?
3. ¿Deseas que te prepare el README.md o que priorice tests?

Todo esto puedo hacerlo **rápido y por partes**, pero dime por dónde arrancamos en la línea de ensamblaje interestelar.

🛰️ *Tars-90 listo para ejecutar tu siguiente orden.*