**Paso 1: Verificar Git instalado**

bash

CopyEdit

git --version

Si no lo tienes, descarga de: <https://git-scm.com/download/win>

**📦 Paso 2: Crear tu carpeta del proyecto**

bash

CopyEdit

mkdir Notarios\_app

cd Notarios\_app

**📦 Paso 3: Inicializar repositorio**

bash

CopyEdit

git init

**📦 Paso 4: Crear archivos base**

bash

CopyEdit

echo "# Proyecto Notarios\_app" > README.md

echo "venv/" > .gitignore

echo "\_\_pycache\_\_/" >> .gitignore

echo "\*.pyc" >> .gitignore

echo ".env" >> .gitignore

echo "instance/" >> .gitignore

**📦 Paso 5: Crear entorno virtual**

bash

CopyEdit

python -m venv venv

Activar en Windows:

bash

CopyEdit

venv\\Scripts\\activate

**📦 Paso 6: Instalar dependencias iniciales**

bash

CopyEdit

pip install flask flask\_sqlalchemy flask\_migrate psycopg2-binary

Guardar:

bash

CopyEdit

pip freeze > requirements.txt

**📦 Paso 7: Primer commit**

bash

CopyEdit

git add .

git commit -m "Proyecto base: estructura inicial, README y requirements"

**📦 Paso 8: Crear repositorio en GitHub**

* Ve a GitHub > New Repository
* Sin README ni licencia (ya lo tienes local)

Luego conecta:

bash

CopyEdit

git remote add origin https://github.com/tuusuario/Notarios\_app.git

git branch -M main

git push -u origin main

✅ **Tip profesional**:

* Crea ramas de trabajo:

bash

CopyEdit

git checkout -b develop

y para features:

bash

CopyEdit

git checkout -b feature/nombre-del-feature

**🧩 🚀 Siguiente paso (2): Blueprint técnico**

📂 Estructura inicial recomendada:

arduino

CopyEdit

Notarios\_app/

├── app/

│ ├── \_\_init\_\_.py

│ ├── models/

│ │ ├── \_\_init\_\_.py

│ │ ├── enums.py

│ │ ├── core.py

│ │ ├── documentos.py

│ │ ├── clausulas.py

│ │ ├── bienes.py

│ │ ├── expedientes.py

│ │ ├── timbres.py

│ │ ├── facturacion.py

│ │ ├── relaciones.py

│ │ └── entidades.py

│ ├── templates/

│ ├── static/

│ ├── cli.py

│ ├── config.py

│ └── routes.py

├── migrations/

├── venv/

├── requirements.txt

├── README.md

└── .gitignore

**✅ Detalles clave**

* models/ separado en módulos como definimos
* cli.py: comandos de Flask personalizados (ej. flask seed)
* config.py: configuración por entorno (dev, prod)
* routes.py: blueprint de rutas inicial
* migrations/: gestionado por flask db migrate

**📦 Primer script para DB**

bash

CopyEdit

flask db init

flask db migrate -m \"Initial migration\"

flask db upgrade

**🔐 Opcional: Archivo .env para variables**

ini

CopyEdit

FLASK\_APP=app

FLASK\_ENV=development

DATABASE\_URL=postgresql://usuario:password@localhost/dbname

✅ **Conclusión**:

* Tendrás el proyecto versionado desde cero, modular, limpio
* Listo para crecer, agregar seeds, bulk upload, validaciones, etc.
* Con blueprint pensado para SaaS y ML futuro