Para definir a versão do Python no local:

pyenv local versão

Criar virtual environment

python -m venv .venv

Ativar o venv

source .venv/Scripts/activate

instalar o front end

pip install streamlit

rodar o streamlit - página web

streamlit run nome.py

utilizado o streamlit para criar o dashboard

Necessário instalar a pydantic - pip install pydantic

para validar email, é necessário o cód: pip install pydantic[email]

Inicialmente, utilizado o render - banco de dados 24h - gratuito

Vinculado ao GitHub

em new, criar um PostgreSQL

CREATE TABLE IF NOT EXISTS vendas(

id SERIAL PRIMARY KEY,

email VARCHAR(255) NOT NULL,

data TIMESTAMP NOT NULL,

valor NUMERIC NOT NULL,

quantidade INTEGER NOT NULL,

produto VARCHAR(50) NOT NULL

)

instalar psycopg2

pip install psycopg2-binary

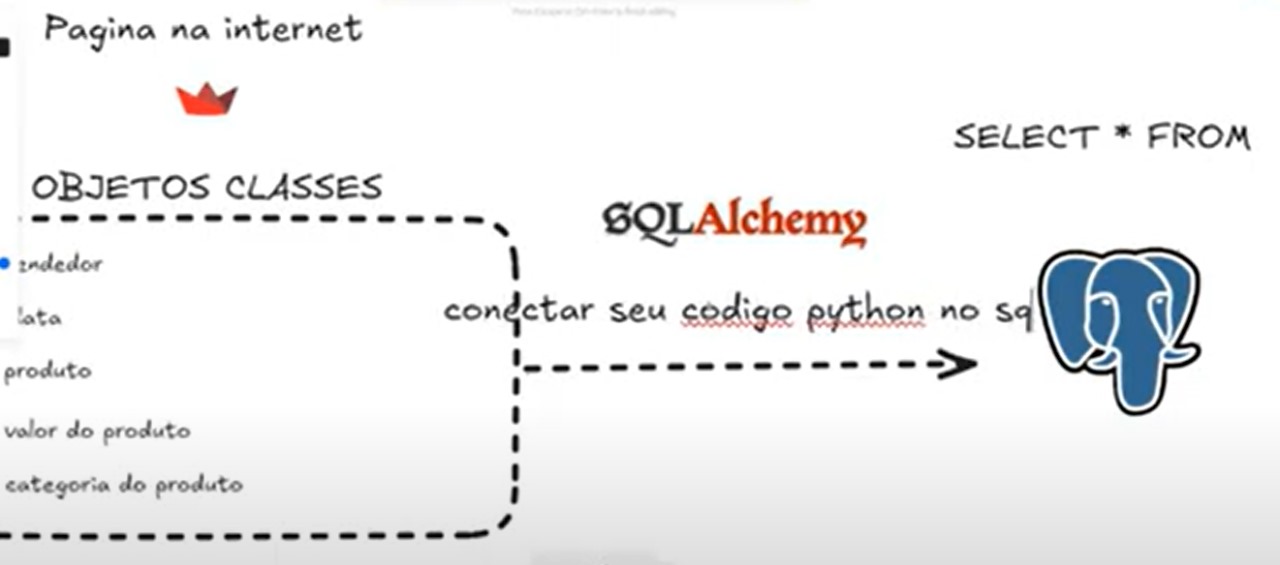
AqlAlchemy

pip install sqlalchemy

dotenv

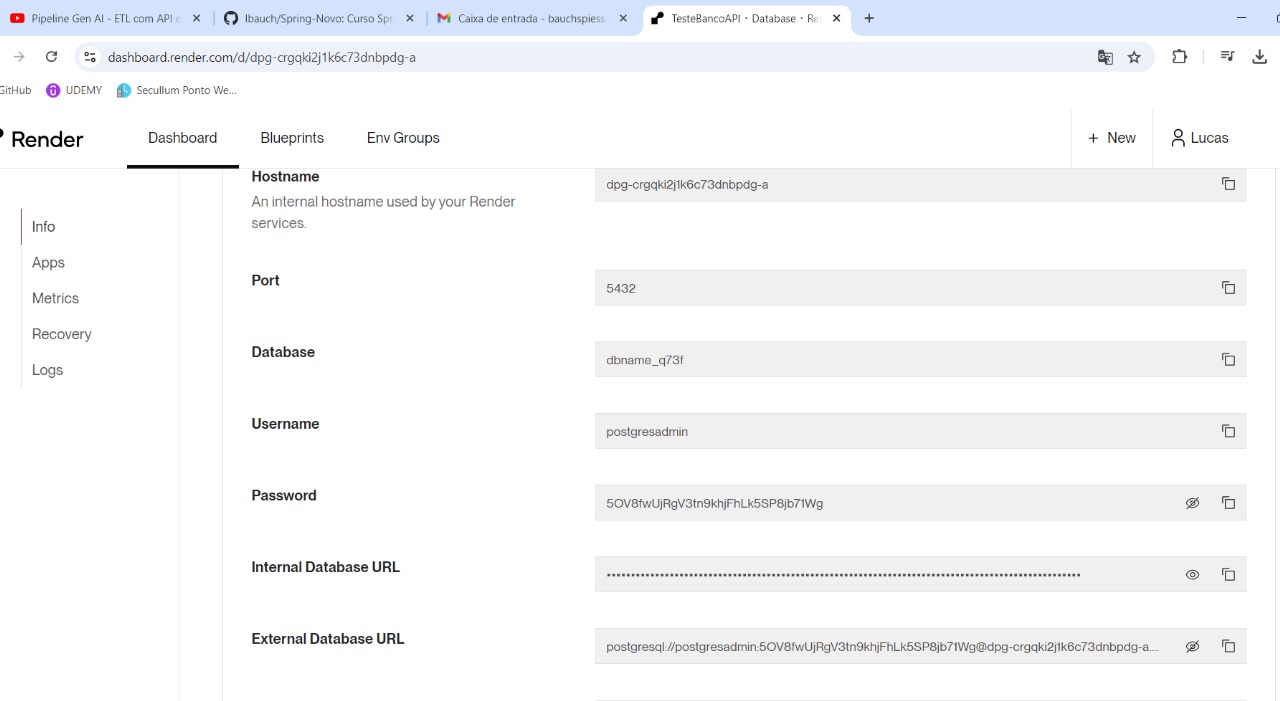
pip install python-dotenv

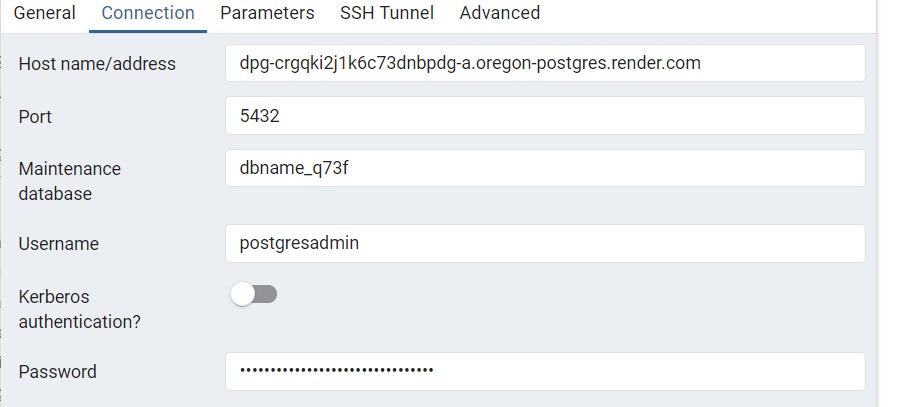
pip freeze > requirements.txt



Pydentic - valida os campos para verificar se estão corretos ou não. – datacontract

Mkdocs para documentar o projeto





No PostgreSQL, criar um servidor. O nome do host é oque aparece após o @ (excluindo o @) e até o .com (incluindo o .com) – Render

Para fazer o deploy do site:  
entrar no streamlit com github  
canto superior direito – deploy  
deploy now  
inserir URL  
variáveis de ambiente definidas – passar os valores entre aspas

Instalar o mkdocs – gerador de documentação  
pip install mkdocs

Instalar o mkdocs – material design do Google – gerador de documentação  
pip install mkdocs mkdocs-material mkdocstrings mkdocstring-python

No arquivo mkdocs.yml, passar os parâmetros:

site\_name: My Docs

theme:

  name: "material"

plugins:

  - search

  - mkdocstrings:

    default\_handler: python

Em index .md (dentro da pasta docs gerada)

Botar conforme modelo os métodos necessários:

::: contrato.Vendas

::: database.salvar\_no\_postgres

Gerar a documentação:

mkdocs build

mkdocs gh-deploy