

Passo a Passo Alembic + SQLAlchemy – Autor & Livro

- **Inicialização do Alembic**

- *alembic init alembic*

- **No alembic.ini, altere a string de conexão:**

- *sqlalchemy.url =*
DRIVER://USER:PASSWORD@localhost/DBNAME
- Troque para:
 - *sqlalchemy.url = %(DATABASE_URL)s*
- E garanta que o Alembic vai ler da variável de ambiente. (Ou deixe vazio, que ele pega do [env.py](#).)

- **No alembic/env.py, altere para importar o metadata do SQLAlchemy:**

- Substitua:
 - *# from myapp import mymodel*
 - *# target_metadata = mymodel.Base.metadata*
- Por:

```
import sys
import os
sys.path.append(os.path.abspath(os.path.join(os.path.dirname(__file__), '..')))
from app.models import SQLAlchemyModel

target_metadata = SQLAlchemyModel.metadata
```

- E na função *run_migrations_online()*, troque a *get_url* para ler do *.env*:

```
from app.database import DATABASE_URL

def get_url():
    return DATABASE_URL
```

- **Criação da Primeira Migração**

- alembic revision --autogenerate -m "criação tabelas autor e livro"
- alembic upgrade head
 - Veja o SQL gerado no alembic/versions/xxxx_criacao_tabelas_autor_e_livro.py.
 - Isso cria as tabelas no banco configurado (PostgreSQL).

- **Testando e Conferindo o Banco**

- Abra seu gerenciador de banco ou use psql/dbeaver para checar se as tabelas estão criadas.
- O Alembic gera a tabela de controle alembic_version.

- **Alterando o Schema – Exemplo: Adicionar Campo 'email' ao Autor**

- **Altere o modelo em app/[models.py](#):**
 - Adicione
 - email: Optional[str] = Field(default=None, index=True)
- **Gera nova migração:**
 - alembic revision --autogenerate -m "add campo email ao autor"
 - alembic upgrade head
- **Cheque no banco:**
 - A coluna 'email' foi criada em autor.

- **Trocando o Banco: De PostgreSQL para SQLite**

- Altere o .env:
 - DATABASE_URL=sqlite:///./[livros.db](#)
- **Se desejar, limpe o banco SQLite anterior (apague o arquivo livros.db se existir):**

- rm [livros.db](#)

- **Execute novamente as migrações:**
 - alembic upgrade head

- **Resumo do Fluxo**

- Crie os modelos e configure Alembic.
- Gere a migração inicial e aplique no banco (PostgreSQL).
- Modifique o modelo (ex: adiciona campo), gere nova migração, e aplique.
- Altere o banco para SQLite, execute novamente as migrações → schema será replicado no novo banco!
- Continue o desenvolvimento com FastAPI.