## Estrutura dos arquivos app/

### main.py

* Usa lifespan moderno com asynccontextmanager
* Conecta e inicializa o banco corretamente com create\_db\_and\_tables()
* Inclui as rotas /mfa via app.include\_router(mfa.router)  
  main

### db.py

* Define o engine usando sqlmodel.create\_engine()
* Usa os.getenv("DATABASE\_URL", "..."), o que permite facilmente alternar para .env
* Executa SQLModel.metadata.create\_all(...) para criar as tabelas  
  db

### models.py

* Define o modelo MFASecret com SQLModel
* Inclui Field(primary\_key=True) corretamente

### routes/mfa.pymfa

* Tem os 4 endpoints essenciais:
  + /setup → cria e salva secret + QR base64
  + /verify → valida token consultando o banco
  + /qrcode-image → retorna PNG direto
  + /demo → mostra QR com HTML
* Todos persistem corretamente com Session(engine) via SQLModel
* ISSUER\_NAME vem do .env → ótimo para reutilizar

### schemas/mfa.pymfa

* Modelos GenerateRequest, GenerateResponse, VerifyTokenRequest muito bem definidos
* Inclui todos os campos úteis, como qrcode\_preview

### utils/totp.pytotp

* Todas as funções estão isoladas e bem escritas
* verify\_totp\_token() já considera valid\_window=1, evitando falsos negativos