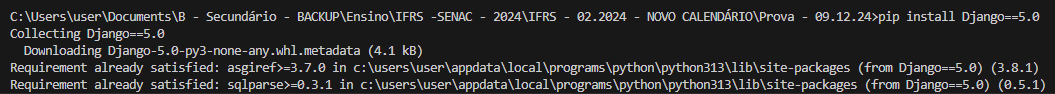
***ROTEIRO PROVA 09.12.24***

***COMANDOS NO LABORATÓRIO DO IFRS***

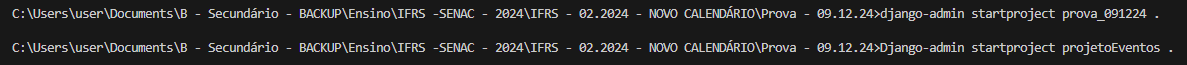
1. **pip install Django==5.0**

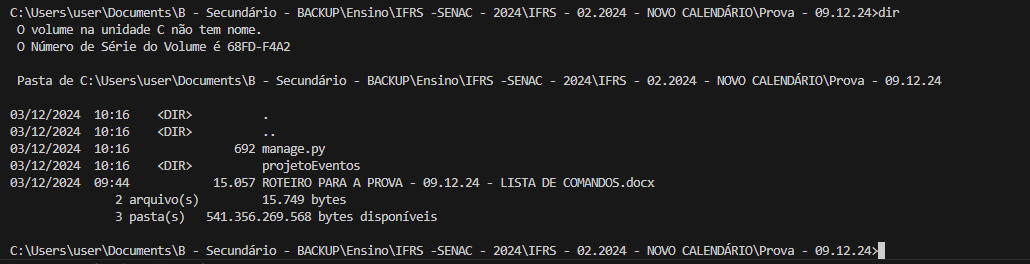
**2 vezes para instalar o django atualizado (só no PC do if, pois reseta ao ser reiniciado)**

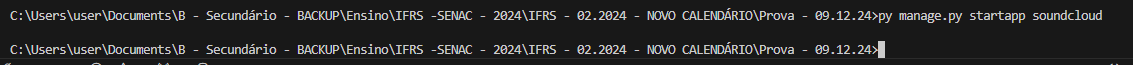


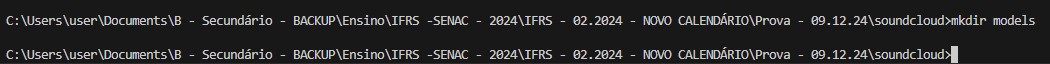
1. **Python manage.py runserver**

**(roda o servidor)**

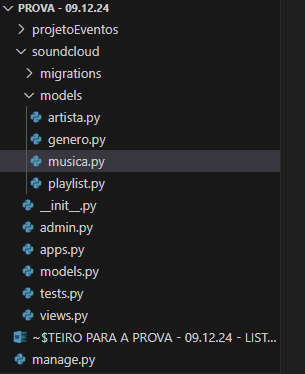
1. Criar uma pasta para o PROJETO
2. Abrir no VS Code
3. Abrir view/prompt de comando
4. mkdir - Cria um diretório **prova\_091224**
5. **Dentro do diretório digite Django-admin startproject projetoEventos .**
6. 
7. Nota: Ele também cria o **MANAGE.PY**



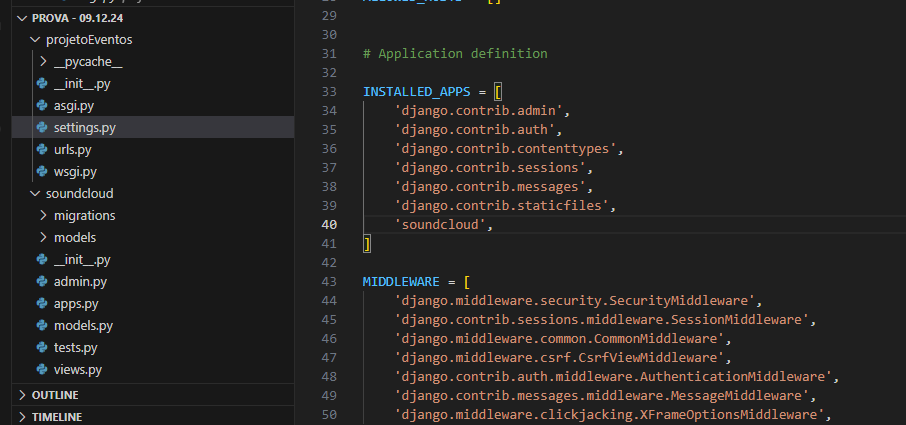
1. **py manage.py startapp soundcloud** 
2. C:\Users\user\Documents\B - Secundário - BACKUP\Ensino\IFRS -SENAC - 2024\IFRS - 02.2024 - NOVO CALENDÁRIO\Prova - 09.12.24\soundcloud>**mkdir models**



1. Criar dentro da pasta **models: genero.py, playlist.py, artista.py e musica.py**



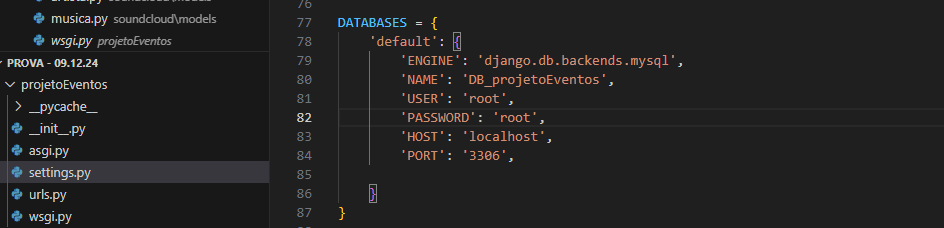
1. Adicionar o app **soundcloud** criado em settings.py no projeto criado pelo comando **Django-admin startproject projetoEventos**



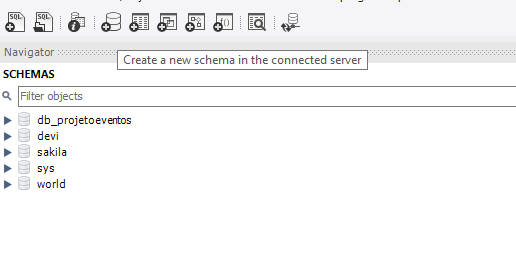
**TESTAR: ACESSAR PHPMYADMIN --> <https://php-myadmin.net/>**

**OBS: Configurar o SQLite base de dados no DJANGO**

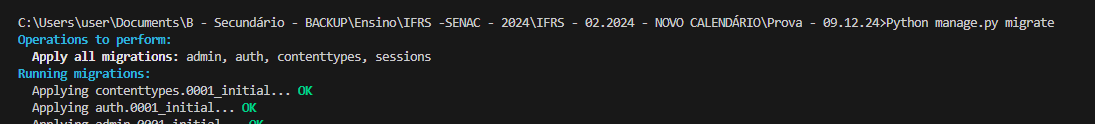
**OBS: Configurar a base de dados no DJANGO**

1. 

**OBS: Criar a base de dados no WORKBENCH**

1. 
2. **Python manage.py migrate**

**(para migrar)**

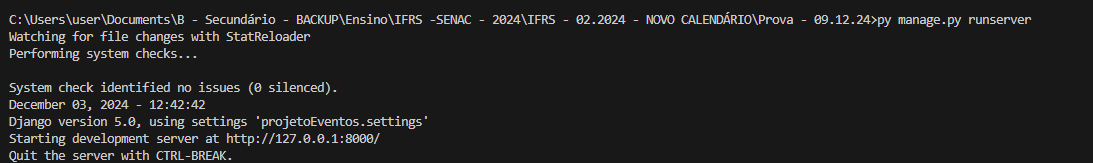


1. **Python manage.py makemigrations**

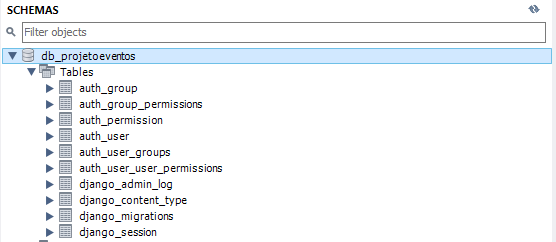
**(para fazer as migrações)**

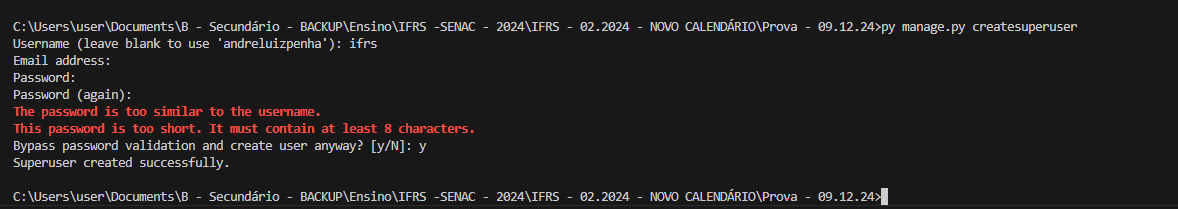


1. **Rodar com o comando  para testar**



**OBS: Após roda o MIGRATE na base de dados no WORKBENCH**

1. 
2. **py manage.py createsuperuser ifrs - ifrs - Cria um usuário como administrador**



**LISTA DE COMANDOS - EXTRAS**

**LISTA DE COMANDOS - EXTRAS**

**cd - Troca de diretório**

**cd.. - Volta diretório**

**mkdir - Cria um diretório**

**activate - Ativa um ambiente virtual**

**deactivate - Desativa um ambiente virtual**

**pip install Django - Instala o Django**

**py manage.py makemigrations - Cria toda a estrutura de banco de dados**

**Após determinar quais serão as entidades que utilizaremos em nosso projeto, precisamos criar as tabelas que representam estas entidades no banco de dados. Para isso, utilizamos o seguinte comando:**

**python mananage.py makemigrations**

**O comando acima irá obter a estrutura das classes definidas no arquivo models.py e criará os arquivos de migração de cada classe.**

**py manage.py migrate - Efetiva as alterações no worlbenck**

**Executar Migrações**

**Após criar os arquivos que definem a estrutura de cada entidade no banco de dados, precisamos executar estas migrações. Para isso, utilizamos o seguinte comando:**

**python manage.py migrate**

**Com isso, todas as classes definidas no arquivo models.py serão transformadas em tabelas no banco de dados.**

**py manage.py runserver - Roda a base de dados do DJANGO**

**py manage.py createsuperuser - Cria um usuário como administrador**

**django-admin startproject nome\_do\_projeto - Para criar um projeto no Django, utilizamos este comando**

**Após criar um projeto, precisamos criar uma (ou mais) aplicações para o projeto. Para isso, utilizamos o seguinte comando:**

**python manage.py startapp nome\_da\_app**

**py manage.py startapp eventos\_prova - Cria meu app eventos\_prova**

**py -m venv teste - Cria um ambiente virtual chamado teste**

**python.exe -m pip install --upgrade pip - Atualização do PYTHON**

**pip install python-decouple - PROTEGE AS INFORMAÇÕES**

**pip install dj-database-url - CRIAÇÃO DE URL PARA BANCO DE DADOS**

**pip install dj-static - GERENCIA ARQUIVOS STATICOS**

pip freeze > requirements-dev.txt - VAI CRIAR O ARQUIVO requirements-dev.txt COM UMA LISTA COM TUDO QUE JÁ FOI INSTALADO

pip install psycopg2 - INSTALAÇÃO DO PSYCOPG2

python.exe -m pip install --upgrade pip - INSTALAÇÃO DO PSYCOPG2

heroku plugins:install heroku-config - exporta as configurações do ENV

heroku config:push -a - (ESTE COMANDO NÃO FUNCIONOU) exporta as configurações do ENV para o ambiente remoto

**Limpar Banco de Dados**

**Quando estamos desenvolvendo nossa aplicação, é comum que queiramos limpar todos os dados do banco de dados para realizar novos testes. Para realizar este procedimento, utilizamos o seguinte comando:**

**python manage.py flush**

**Este comando irá limpar todos os dados salvos no banco de dados do projeto, mantendo sua estrutura.**

**Mapear BD existente para o projeto Django**

**Caso você possua um banco de dados já existente e queira mapeá-lo para seu projeto Django, o seguinte comando resolve esta necessidade:**

**python manage.py inspectdb > nome\_da\_app.models.py**

**Este comando irá mapear o banco de dados já existente no arquivo models.py da sua aplicação, conforme vimos**

**\* é obrigado o parâmetro**