

# Django Model

Prof. Me. Thiago Ferauche  
thiago.ferauche@unisantos.br

# O arquivo settings.py

- No arquivo settings.py teremos as variáveis que serão utilizadas no framework django para comunicação com banco de dados, além de módulos adicionais que queira utilizar.
- Por padrão, o SQLite já vem configurado e já instalado com o framework Django, mas é possível alterá-lo para acessar qualquer banco de dados.

# O arquivo settings.py

- Dentro do arquivo settings.py de seu projeto já vem o SQLite

```
# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

- Pode alterar, ou configurar outros banco de dados, adicionando ao dicionário DATABASES, colocando:
  - ENGINE : 'django.db.backends.postgresql' ou 'django.db.backends.mysql'
  - NAME : nome do DATABASE, além de outros parâmetros como usuário e senha, etc

## O arquivo `settings.py`, `INSTALLED_APPS`

- `INSTALLED_APPS` é a variável que contém módulos do framework já instaladas, e podemos também incluir outros módulos. Dentro dos já instalados destacamos:
  - `django.contrib.admin` – Site de Administração
  - `django.contrib.auth` – Um sistema de Autenticação
  - `django.contrib.contenttypes` – framework para tipos de conteúdo
  - `django.contrib.sessions` – framework de controle de sessão
  - `django.contrib.messages` – framework de envio de mensagens
  - `Django.contrib.staticfiles` – gerenciamento de arquivos estáticos

# Comando `manage.py migrate`

- O comando migrate irá verificar o que está em `INSTALLED_APPS` e cria as alterações necessárias nas tabelas do Banco de Dados do seu projeto.
  - `python manage.py migrate`
- Na primeira vez irá criar toda a estrutura de tabelas das aplicações instaladas.
- Caso tenha o `sqlite3` instalado na sua máquina pode acessar direto o banco de dados e visualizar as tabelas criadas. Pode usar também o DB Browser SQLite (<https://sqlitebrowser.org/>)

# Criando modelos da sua aplicação

- Modelo é a camada de software que irá carregar e levar a informação do banco de dados ao usuário, e vice-versa.
- Para isso cada modelo é uma classe filha da class `moldels.Model`, setando os atributos e relacionamento entre os modelos criados.
- Estes modelos, posteriormente serão traduzidos para Banco de Dados Relacionais.

# Alterando o polls/models.py

- Em nossa aplicação polls, iremos criar o Modelo Questions e Choices, relacionados entre si de forma que se apagar uma questão, as escolhas dessas questão serão apagadas automaticamente.

```
from django.db import models
# Create your models here.

class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
```

# Adicionando polls em INSTALLED\_APPS

- Para que manage.py funcione, e as tabelas referentes a minha aplicação sejam gerenciadas pelo DJANGO é necessário incluir a classe PollsConfig já criada em apps.py da minha aplicação na lista INSTALLED\_APPS, dentro de settings.py

mysite/settings.py

```
INSTALLED_APPS = [  
    'polls.apps.PollsConfig',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```





# Criando os modelos no BD

- Executando o comando makemigrations gera os scripts
  - `python manage.py makemigrations polls`
- Os scripts de banco ficarão guardados em `polls/migrations`

```
mysite$ python manage.py makemigrations polls
```

Migrations for 'polls':

`polls/migrations/0001_initial.py`

- Create model Question
- Create model Choice

# Criando os modelos no BD

- É possível visualizar os comandos SQLs que serão executados usando o comando abaixo :
- `python manage.py sqlmigrate polls 0001`

```
BEGIN;
--
-- Create model Question
--
CREATE TABLE "polls_question" (
    "id" bigint NOT NULL PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY,
    "question_text" varchar(200) NOT NULL,
    "pub_date" timestamp with time zone NOT NULL
);
--
-- Create model Choice
--
CREATE TABLE "polls_choice" (
```

# Criando os modelos no BD

- Vamos agora efetivamente gerar as tabelas no Banco
  - `python manage.py migrate`

```
mysite$ python manage.py migrate
```

Operations to perform:

Apply all migrations: admin, auth, contenttypes, polls, sessions

Running migrations:

Applying polls.0001\_initial... OK

# Resumo Django Models

- Toda a vez que criar, ou alterar modelos, deverá seguir os passos abaixo:
  - 1) Salve as alterações em models.py
  - 2) Execute `python manage.py makemigration <sua_app>`
  - 3) Execute `python manage.py migrate`