**ETAPA 6**

**1- Criando o models**

Antes de qualquer coisa , o processo do models é o mesmo do processo da etapa 3, primeiro va para a parte de models, e crie um objeto que tenha os mesmos atributos que teria o create table, mas deve ser levado em consideração o modulo de choices, que serve justamente para selecionar algo que teria multipla escolha e não digitar , Dentro do file App/models.py coloque esse seguinte codigo

import uuid

import re

#from stdimage.models import StdImageField

from django.utils import timezone

from pathlib import Path, os

from django.db.models import signals

from django.core.exceptions import ValidationError

from django.db import models

from django.core.validators import MinLengthValidator, MaxLengthValidator,MinValueValidator,MaxValueValidator

class Doador(models.Model):

options\_choices = (

('A', ' tipo A '),

('B', 'tipo B '),

('AB','tibo AB '),

('O', 'tipo O '),

)

CHOICES = (

('+', 'RH + '),

('-', 'RH - '),

)

codigo = models.AutoField("alto completo",null=False,primary\_key=True)

nome = models.CharField("Nome",blank=True,max\_length=200)

cpf = models.CharField("CPF",blank=True,max\_length=11)

contato = models.CharField("numero",blank=True,max\_length=12)

tipo\_sanguineo = models.CharField("tipo sanguineo",blank=True,null =False ,max\_length=4 ,choices = options\_choices)

rh = models.CharField("RH",blank=True,max\_length=1 ,choices= CHOICES)

tipo\_rh\_corretos = models.BooleanField("RH corretos",blank=True)

situacao = models.CharField("situcao", blank=True)

class Meta:

ordering = ['codigo']

verbose\_name = 'Doador'

def get\_detalhe(self):

return f'/teste/Doador/{self.codigo}/'

def get\_modifica(self):

return f'/teste/altera\_doador/{self.codigo}/'

def get\_exclui(self):

return f'/teste/deleta\_Doador/{self.codigo}/'

def \_\_str\_\_(self):

return self.nome

class Doacao(models.Model):

codigo = models.AutoField("alto completo",null=False,primary\_key=True)

Data = models.DateField("Data",blank=True,null=False)

Hora = models.TimeField("time",blank=True,null=False)

volume = models.DecimalField("volume",decimal\_places = 2 , max\_digits = 7)

situacao = models.CharField("situcao", blank=True)

codigo\_doador = models.ForeignKey(Doador,db\_column='codigo\_doador',related\_name='codigo\_do',on\_delete=models.CASCADE)

class Meta:

ordering = ['codigo']

verbose\_name = 'Doador'

verbose\_name\_plural = 'Doadores'

Agora va para a pasta principal, e procure por settings.py e insire essa parte do codigo

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.postgresql',

'NAME': 'teste',

'USER': 'postgres',

'PASSWORD': '12345',

'HOST': 'localhost',

'PORT': '5432',

}

}

Essa parte faz justamente a conexão do codigo principal com ele , e isso pode se dizer que está pronto para conectar com o banco, antes de rodar o código

deve abrir o prompt, ou usar o próprio prompt do views e rodar o seguinte comando

python manage.py makemigrations APP

python manage.py migrate APP

isso vai criar o diretorio migrations e migra-los direto para o banco de dados