



# Django Framework do Zero

## Como Criar um Aplicativo Web

Canal [youtube.com/@VamosCodar](https://www.youtube.com/@VamosCodar) - Wanderson Reis

## 29 - Limitar Alternativas de Respostas no Cadastro

### Requisitos

- 1) Aulas 01 a 28 - Django Framework do Zero - Playlist:  
<https://www.youtube.com/playlist?list=PLFOqHo8oljzewcT23HCxJV0xWO451CTJe>

### Introdução

Nesta aula vamos aprender criar uma regra de negócio que permite cadastrar apenas uma quantidade pré-definida (personalizável) de alternativas (respostas) para um enquete.

#### Nesta aula:

- 1) Criar um item de configuração
- 2) Implementar a regra no model Choice
- 3) Alterar a view para tratar a exceção

**Observação:** Lembre-se de estar com o ambiente virtual ativado (venv).

### Passo 1: Criar um item de configuração

O objetivo deste passo é tornar a quantidade máxima de alternativas permitidas configurável. Editar o arquivo **django-do-zero/webapp/settings.py** e acrescente a nova chave de configuração conforme ilustrado na imagem abaixo:

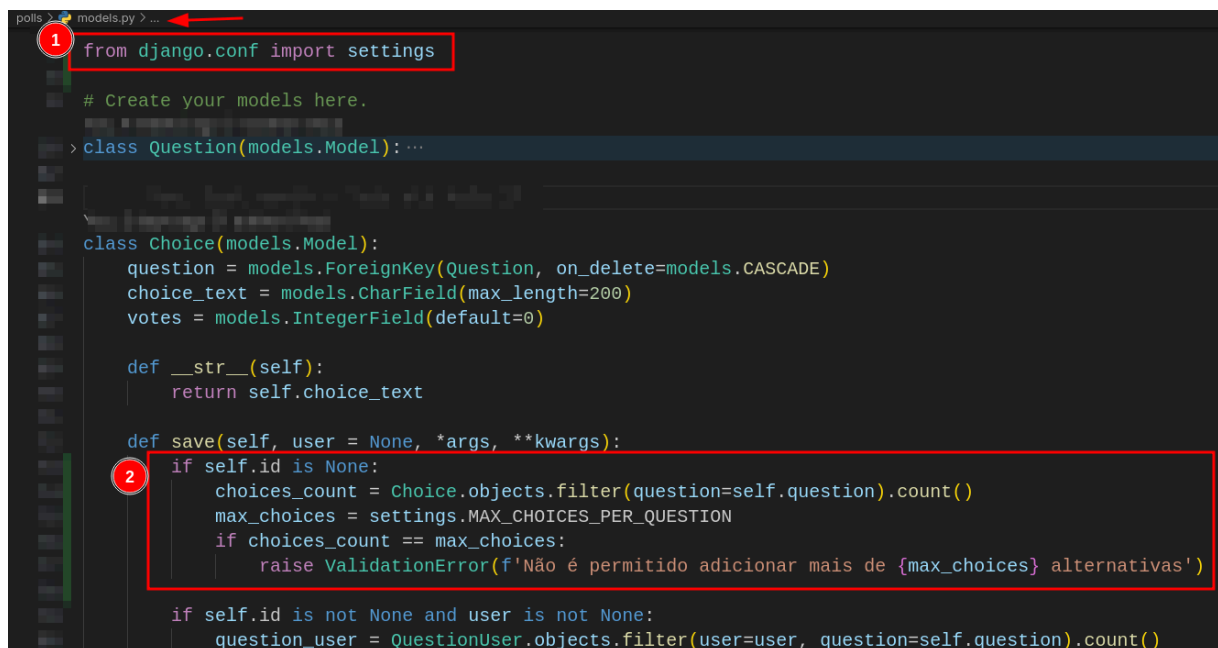
```
webapp > settings.py > ...  
EMAIL_USE_TLS = True  
EMAIL_HOST_USER = 'USUÁRIO DO EMAIL'  
EMAIL_HOST_PASSWORD = 'SENHA DO EMAIL'  
  
# Configurações Enquete  
MAX_CHOICES_PER_QUESTION = 3
```

(salve o arquivo).

## Passo 2: Implementar a regra no model Choice

Antes de criar uma nova alternativa de resposta para uma pergunta é necessário validar se estamos dentro da regra que limita a quantidade. É neste passo que faremos a validação.

Editar o arquivo **django-do-zero/polls/models.py** e modifique seguindo o exemplo mostrado na próxima imagem:



```
polls > models.py > ...  
1 from django.conf import settings  
  
# Create your models here.  
  
> class Question(models.Model): ...  
  
class Choice(models.Model):  
    question = models.ForeignKey(Question, on_delete=models.CASCADE)  
    choice_text = models.CharField(max_length=200)  
    votes = models.IntegerField(default=0)  
  
    def __str__(self):  
        return self.choice_text  
  
    def save(self, user = None, *args, **kwargs):  
2        if self.id is None:  
            choices_count = Choice.objects.filter(question=self.question).count()  
            max_choices = settings.MAX_CHOICES_PER_QUESTION  
            if choices_count == max_choices:  
                raise ValidationError(f'Não é permitido adicionar mais de {max_choices} alternativas')  
  
        if self.id is not None and user is not None:  
            question_user = QuestionUser.objects.filter(user=user, question=self.question).count()
```

(salve o arquivo).

## Passo 3: Alterar a view para tratar a exceção

As regras de negócios geralmente ficam implementadas nos models, contudo é preciso exibir corretamente um retorno para o usuário. Isso será feito através do tratamento da exceção gerada quando se tentar cadastrar mais alternativas de resposta do que o permitido.

Editar o arquivo **django-do-zero/polls/views.py** conforme indicado na imagem:

```
polls > views.py >
class ChoiceCreateView(LoginRequiredMixin, CreateView):
    model = Choice
    template_name = 'polls/choice_form.html'
    fields = ('choice_text', )
    success_message = 'Alternativa registrada com sucesso!'

    def dispatch(self, request, *args, **kwargs):...

    def get_context_data(self, **kwargs):...

    def form_valid(self, form):
        try:
            form.instance.question = self.question
            response = super(ChoiceCreateView, self).form_valid(form)
        except (ValidationError) as error:
            messages.error(self.request, error.message)
            return self.form_invalid(form)
        else:
            messages.success(self.request, self.success_message)
            return response
```

(salve o arquivo).

Em seguida suba o servidor de desenvolvimento.

`python manage.py runserver` (ENTER)

Para testar, acesse uma enquete estando em uma sessão logada, edite a pergunta e cadastre novas respostas até que o sistema limite o cadastro de novas com base na configuração realizada no Passo 1.

Observe que ao alterar a quantidade máxima de alternativas de respostas permitidas no arquivo settings.py é necessário reiniciar o servidor.