



# Django Framework do Zero

## Como Criar um Aplicativo Web

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

### 14 - CRUD Básico - READ

#### Requisitos

- 1) Aulas 01 a 13 - Django Framework do Zero - Playlist:

<https://www.youtube.com/playlist?list=PLFOqHo8oljzewcT23HCxJV0xWO451CTJe>

#### Nesta aula:

- 1) Implementar a view genérica para exibir uma pergunta
- 2) Criar a rota para view QuestionDetailView
- 3) Implementar a view genérica para listar todas as perguntas
- 4) Implementar uma view genérica sem Model

#### Passo 1: Implementar a view genérica para exibir uma pergunta

Edite o arquivo **django-do-zero/polls/views.py** e acrescente a view baseada em classe da seguinte forma (**atenção apenas ao conteúdo novo**):

**# (...manter tudo o que já existe...)**

**# na parte superior do arquivo**

`from django.views.generic import DetailView`

**# na parte inferior do arquivo**

`class QuestionDetailView(DetailView):`

`model = Question`

`template_name = 'polls/question_detail.html'`

`context_object_name = 'question'`

(salve o arquivo).

É necessário criar o template **question\_detail.html** dentro da pasta de **templates/polls** (conforme o seu ambiente). O arquivo deverá ter uma estrutura semelhante ao conteúdo abaixo:

`{% extends '_layout1.html' %} {# herda tudo que já tem em _layout1.html #}`

```
{% block content %} {# substitui o bloco content em _layout1.html #}
<div class="card">
  <div class="card-header">
    <h5 class="card-title">Pergunta</h5>
  </div>
  <div class="card-body">
    <p>
      {{ question.question_text }}
      <a href="{% url 'question_edit' pk=question.id %}" class="link-secondary">
        Editar
      </a> &nbsp;
      <a href="{% url 'question_delete' pk=question.id %}" class="link-secondary">
        Excluir
      </a>
    </p>
    <p>Data de publicação: {{ question.pub_date }}</p>
  </div>
</div>
{% endblock %} {# fecha o bloco content #}
```

(salve o arquivo).

## Passo 2: Criar a rota para view QuestionDetailView

Edite o arquivo `django-do-zero/polls/urls.py` e acrescente a nova rota seguindo o exemplo abaixo (**atenção apenas ao conteúdo novo**):

```
from polls import views # vou passar a importar desta maneira

urlpatterns = [
  # (...manter tudo o que já existe...)
  path('pergunta/<int:question_id>/remove', question_delete, name="question_remove"),
  path('enquete/<int:pk>/show',
    views.QuestionDetailView.as_view(),
    name="question_detail"
  ),
]
```

(salve o arquivo).

Em seguida suba o servidor de desenvolvimento e acesse a nova rota.

`python manage.py runserver` (ENTER)

**Substituir o ID-EXISTENTE por um ID de pergunta já cadastrada no seu ambiente.**

No navegador acesse <http://127.0.0.1:8000/enquete/ID-EXISTENTE/show>

É necessário editar o template **questions.html** dentro da pasta **django-do-zero/templates/polls** e incluir uma tag similar ao exemplo abaixo:

Observe que o trecho de código abaixo fica dentro do laço FOR pois a cada nova pergunta tem seu próprio ID e link para **visualizar**.

```
<a href="{% url 'question_detail' pk=question.id %}" class="link-secondary">
    Ver
</a>
```

(salve o arquivo).

Observe que o link da rota será gerada/traduzida automaticamente pelo comando de template **url** que recebe como argumento **'poll\_show'** mais o valor do parâmetro.

Suba o servidor (se já não estiver em execução) e acesse a página/rota onde estará o link criado.

## Passo 3: Implementar a view genérica para listar todas as perguntas

Edite o arquivo **django-do-zero/polls/views.py** e acrescente a view baseada em classe da seguinte forma (**atenção apenas ao conteúdo novo**):

```
# (...manter tudo o que já existe...)
```

```
# na parte superior do arquivo
```

```
from django.views.generic import DetailView, ListView
```

```
# na parte inferior do arquivo
```

```
class QuestionListView(ListView):
    model = Question
    template_name = 'polls/question_list.html'
    context_object_name = 'questions'
```

(salve o arquivo).

É necessário criar o template **question\_list.html** dentro da pasta de **templates/polls** (conforme o seu ambiente). O arquivo deverá ter uma estrutura semelhante ao conteúdo abaixo:

```
{% extends '_layout1.html' %} {# herda tudo que já tem em _layout1.html #}

{% block content %} {# substitui o bloco content em _layout1.html #}
<div class="card">
```

```

<div class="card-header">
  <h5 class="card-title">Listagem de Perguntas</h5>
</div>
<div class="card-body">
  <a href="{% url 'poll_add' %}" class="btn btn-primary mb-3">
    Cadastrar nova pergunta
  </a>
  <ul>
    {% for question in questions %}
      <li>
        <a href="{% url 'question_detail' pk=question.id %}">
          {{ question.question_text }}
        </a>
      </li>
    {% endfor %}
  </ul>
</div>
</div>
{% endblock %} {# fecha o bloco content #}

```

(salve o arquivo).

Criar a rota para a nova view QuestionListView. Edite o arquivo **django-do-zero/polls/urls.py** e acrescente a nova rota seguindo o exemplo abaixo (**atenção apenas ao conteúdo novo**):

```

from polls import views

urlpatterns = [
  # (...manter tudo o que já existe...)
  path('enquete/<int:pk>/show',
        views.QuestionDetailView.as_view(),
        name="poll_show"
    ),
  path('enquete/list',
        views.QuestionListView.as_view(),
        name="polls_list"
    ),
]

```

(salve o arquivo).

Em seguida suba o servidor de desenvolvimento e acesse a nova rota.

**python manage.py runserver (ENTER)**

No navegador acesse <http://127.0.0.1:8000/enquete/list>

É necessário editar o template `_layout1.html` dentro da pasta `django-do-zero/templates` e um novo item no menu usando uma tag similar ao exemplo abaixo:

**# (...manter tudo o que já existe...)**

**# DICA: Incluir antes do link "sobre"**

```
<a class="p-2 text-dark" href="{% url 'polls_list' %}">Enquetes</a>
```

(salve o arquivo).

Observe que o link da rota será gerada/traduzida automaticamente pelo comando de template `url` que recebe como argumento `'polls_list'` mais o valor do parâmetro.

Suba o servidor (se já não estiver em execução) e acesse a página/rota onde estará o link criado.

## Passo 4: Implementar uma view genérica sem Model

E quando eu precisar exibir uma página que não contém num modelo (model) associado ? Sendo apenas uma página estática ? Use a view genérica **TemplateView**. Vamos ao exemplo.

Edite o arquivo `django-do-zero/polls/views.py` e acrescente a view baseada em classe da seguinte forma (**atenção apenas ao conteúdo novo**):

**# (...manter tudo o que já existe...)**

**# na parte superior do arquivo**

```
from django.views.generic import DetailView, ListView, TemplateView
```

**# na parte inferior do arquivo**

```
class SobreTemplateView(TemplateView):
    template_name = 'polls/sobre.html'
```

(salve o arquivo).

Crie o template `polls/sobre.html`:

```
{% extends '_layout1.html' %}

{% block content %} {# substitui o bloco content em _layout1.html #}
<div class="card">
    <div class="card-header">
        <h5 class="card-title">Sobre nós</h5>
    </div>
    <div class="card-body">
        <p>
```

```
Somos ...  
</p>  
</div>  
</div>  
{% endblock %}
```

(salve o arquivo).

E por fim criar o rota editando o arquivo **django-do-zero/polls/urls.py** e acrescente a nova rota seguindo o exemplo abaixo (**atenção apenas ao conteúdo novo**):

```
from polls import views  
  
urlpatterns = [  
    # (...manter tudo o que já existe...)  
    path('about-us',  
        views.SobreTemplateView.as_view(),  
        name="about_page"  
    ),  
]
```

(salve o arquivo).

Com o servidor em execução, acesso no navegador <http://127.0.0.1:8000/about-us>

**# DICA:** editar o template `_layout1.html` e atualizar o link “sobre” para carregar esta nova rota.

```
<a href="{% url 'about_page' %}" class="nav-link px-2 link-dark">Sobre</a>
```