



# Django Framework do Zero

## Como Criar um Aplicativo Web

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

## 20 - Paginação Simples de Resultados

### Requisitos

- 1) Aulas 01 a 19 - Django Framework do Zero - Playlist:

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

### Introdução

Paginar uma lista de resultados muito grande é requisito em todos os sistemas. Usando as views genéricas do Django a implementação da paginação usando ListView já vem pronta, basta configurar como faremos a seguir.

#### Nesta aula:

- 1) Modificar a view genérica que lista todas as perguntas
- 2) Incluir o paginador no template renderizado pela ListView

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

## Passo 1: Modificar a view genérica que lista todas as perguntas

Editar o arquivo **django-polls/polls/views.py** e modificá-lo para acrescentar o bloco de código conforme indicado na próxima imagem.

```
polls > views.py > ...  
  
class QuestionListView(ListView):  
    model = Question  
    template_name = 'polls/question_list.html'  
    context_object_name = 'questions'  
    paginate_by = 5 # quantidade de itens por página  
    ordering = ['-pub_date'] # ordenar pela data de publicação de forma inversa - decrescente
```

(salve o arquivo).

## Passo 2: Incluir o paginador no template renderizado pela ListView

É necessário editar o template `question_list.html` dentro da pasta de `templates/polls` (conforme o seu ambiente). (**atenção apenas ao conteúdo novo**):

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

```
{% endfor %}  
</ul>
```

```
{# exibe paginador somente se tiver mais de uma página #}
```

```
<p class="count font-weight-bold">
```

```
    Total de {{ page_obj.paginator.count }} pergunta(s).
```

```
</p>
```

```
{% if page_obj.paginator.num_pages > 1 %}
```

```
<nav aria-label="...">
```

```
    <span class="current">
```

```
        Exibindo página {{ page_obj.number }} de {{ page_obj.paginator.num_pages }}.
```

```
    </span>
```

```
    <ul class="pagination">
```

```
        {% if page_obj.has_previous %}
```

```
        <li class="page-item">
```

```
            <a class="page-link" href="?page=1">&laquo; Primeira</a>
```

```
        </li>
```

```
        <li class="page-item">
```

```
            <a class="page-link" href="?page={{ page_obj.previous_page_number }}">
```

```
                Anterior
```

```
            </a>
```

```
        </li>
```

```
        {% else %}
```

```
        <li class="page-item disabled">
```

```
            <a class="page-link" href="#">&laquo; Primeira</a>
```

```
        </li>
```

```
        <li class="page-item disabled">
```

```
            <a class="page-link" href="#">Anterior</a>
```

```
        </li>
```

```
        {% endif %}
```

```
{% if page_obj.has_next %}
```

```
    <li class="page-item">
```

```
        <a class="page-link" href="?page={{ page_obj.next_page_number }}">
```

```
            Próxima
```

```
        </a>
```

```
    </li>
```

```
    <li class="page-item">
```

```
        <a class="page-link" href="?page={{ page_obj.paginator.num_pages }}">
```

```

        Última &raquo;
    </a>
</li>
{% else %}
<li class="page-item disabled">
    <a class="page-link" href="#">Próxima</a>
</li>
<li class="page-item disabled">
    <a class="page-link" href="#">Última &raquo;</a>
</li>
{% endif %}
</ul>
</nav>
{% endif %} {# fim do paginador #}

```

(salve o arquivo).

Em seguida suba o servidor de desenvolvimento e acesse a rota que exibirá o paginador.

`python manage.py runserver` (ENTER)

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