

# Back-End Frameworks

Aula 5 - 10/02/2023



[https://bit.ly/23\\_1\\_backend\\_n](https://bit.ly/23_1_backend_n)



## O que temos para **hoje**?

- Popular nosso banco;
- Criar urls com parâmetros;
- Mostrar o resultado em um arquivo html;
- Adicionar urls automáticas aos htmls;
- Reaproveitamento de código;
- Usar arquivos estáticos;
- Continuação do trabalho;



# Popular nosso banco

---



## Popular nosso **banco**

- Nosso banco tem duas tabelas: Question e Choices:
- Crie 3 itens na tabela Question;
- Para cada Question, adicione 3 Choices;



# Criar urls com parâmetros



## Criar urls com **parâmetros**

🕒 No arquivo `polls/urls.py`, deixar assim:

```
polls > 🗄️ urls.py > ...
1  from django.urls import path
2
3  from . import views
4
5  urlpatterns = [
6      # ex: /polls/
7      path('', views.index, name='index'),
8      # ex: /polls/5/
9      path('<int:question_id>/', views.detail, name='detail'),
10     # ex: /polls/5/results/
11     path('<int:question_id>/results/', views.results, name='results'),
12     # ex: /polls/5/vote/
13     path('<int:question_id>/vote/', views.vote, name='vote'),
14 ]
```



## Criar urls com **parâmetros**

🕒 No arquivo polls/views.py, deixar assim:

```
polls > views.py > index
1  from django.shortcuts import render
2
3  # Create your views here.
4  from django.http import HttpResponseRedirect
5
6
7  def index(request):
8      return HttpResponseRedirect("Olá, mundo. Esta é a página inicial.")
9
10 def detail(request, question_id):
11     return HttpResponseRedirect("Essa é a página da questão %s." % question_id)
12
13 def results(request, question_id):
14     response = "Esta é a página de resultados da questão %s."
15     return HttpResponseRedirect(response % question_id)
16
17 def vote(request, question_id):
18     return HttpResponseRedirect("Esta é a página de votação da questão %s." % question_id)
```



## Criar urls com **parâmetros**

### ● Acessar as urls:

`http://127.0.0.1:8000/polls/`

`http://127.0.0.1:8000/polls/34/`

`http://127.0.0.1:8000/polls/34/results/`

`http://127.0.0.1:8000/polls/34/vote/`





# **Mostrar o resultado em um arquivo html**



## Mostrar o resultado em um arquivo **html**

- No arquivo `polls/views.py`, modificar `index` e deixar da seguinte forma:

```
7  ✓ def index(request):  
8      context = {}  
9      return render(request, 'index.html', context)
```



## Mostrar o resultado em um arquivo **html**

- ⦿ Acessar:  
`http://127.0.0.1:8000/polls/`
- ⦿ O que acontece?



## Mostrar o resultado em um arquivo **html**

TemplateDoesNotExist at /polls/

polls/index.html

Request Method: GET

Request URL: http://127.0.0.1:8000/polls/

Django Version: 4.1.3

Exception Type: TemplateDoesNotExist

Exception Value: polls/index.html

Exception Location: /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/django/template/loader.py, line 19, in get\_template

Raised during: polls.views.index

Python Executable: /Library/Frameworks/Python.framework/Versions/3.10/bin/python3

Python Version: 3.10.2

Python Path: [' /Users/fabricioaraujo/Library/CloudStorage/Dropbox/Aulas/Unama\_Alcindo/2023\_1/ADS\_BackEndFrameworks\_Manha/2023\_1\_ads\_backend\_frameworks\_manha/aula4/projeto/mysite',  
'/Library/Frameworks/Python.framework/Versions/3.10/lib/python310.zip',  
'/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10',  
'/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/lib-dynload',  
'/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages']

Server time: Wed, 08 Mar 2023 21:05:13 +0000



## Mostrar o resultado em um arquivo **html**

- Na pasta polls, criar uma pasta templates;
- Dentro da pasta templates, criar um arquivo index.html;



## Mostrar o resultado em um arquivo **html**

- Acessar:  
[https://github.com/fabriciopa/2023\\_1\\_ads\\_backend\\_frameworks\\_noite/blob/main/aula5/projeto/index.html](https://github.com/fabriciopa/2023_1_ads_backend_frameworks_noite/blob/main/aula5/projeto/index.html)
- Copiar o conteúdo e colar no arquivo index.html;



## Mostrar o resultado em um arquivo **html**

- ⦿ Acessar:  
`http://127.0.0.1:8000/polls/`
- ⦿ O que acontece?



## Mostrar o resultado em um arquivo **html**

Navbar Index Detail Results Vote

# Index





## Mostrar o resultado em um arquivo **html**

- Exercício:
  - No arquivo polls/views.py, modificar detail, results e vote para ficar igual a index;
  - Cada uma das páginas deve apontar para um arquivo html diferente;
  - Cada arquivo html pode ser uma cópia de index.html, modificando o texto para diferenciar;



# Adicionar urls automáticas aos htmls



## Adicionar urls automáticas aos **htmls**

- No arquivo polls/templates/index.html, modificar o link de Index para:

```
<li class="nav-item">  
  <a class="nav-link active" aria-current="page" href="{% url 'index' %}">Index</a>  
</li>
```



## Adicionar urls automáticas aos **htmls**

- No arquivo polls/templates/index.html, modificar o link de Detail para:

```
<li class="nav-item">  
  <a class="nav-link" href="{% url 'detail' 34 %}">Detail</a>  
</li>
```



## Adicionar urls automáticas aos **htmls**

- Exercício:
- No arquivo polls/templates/index.html, modificar o link de Results e Vote para o padrão definido em urls.py;
- Replicar o menu para os arquivos: detail.html, results.html e vote.html;



# Reaproveitamento de código



## Reaproveitamento de código

- Na pasta polls/template, criar um arquivo base.html;
- Acessar:  
[https://github.com/fabriciopa/2023\\_1\\_ads\\_backend\\_frameworks\\_noite/blob/main/aula5/projeto/base.html](https://github.com/fabriciopa/2023_1_ads_backend_frameworks_noite/blob/main/aula5/projeto/base.html)
- Copiar o conteúdo e colar no arquivo base.html;



## Reaproveitamento de código

- Modificar o arquivo `polls/templates/index.html` para ficar assim:

```
1  {% extends "base.html" %}
2
3  {% block body %}
4  <h1>
5      Index
6  </h1>
7  {% endblock %}
```





## Reaproveitamento de código

- Exercício:
- Modificar os arquivos detail.html, results.html e vote.html para ficar igual ao arquivo index.html, modificando os textos necessários;



## Reaproveitamento de código

- Exercício:
- No arquivo base.html, adicionar o block body dentro de um container, row e col-12;



# Usar arquivos estáticos



## Usar arquivos **estáticos**

- Na pasta polls, criar uma pasta chamada static. Nesta pasta, criar uma outra pasta chamada img;
- Acessar:  
[https://github.com/fabriciopa/2023\\_1\\_ads\\_backend\\_frameworks\\_noite/blob/main/aula5/projeto/unama.jpeg](https://github.com/fabriciopa/2023_1_ads_backend_frameworks_noite/blob/main/aula5/projeto/unama.jpeg)
- Baixar e salvar o arquivo na pasta img;



## Usar arquivos **estáticos**

- Modificar o arquivo index.html para ficar igual a:

```
1  {% extends "base.html" %}
2  {% load static %}
3  {% block body %}
4  <h1>
5  Index
6  </h1>
7  <p>
8  
9  </p>
10 {% endblock %}
```



# Continuação do trabalho



## Continuação do **trabalho**

- Acessar:  
[https://github.com/fabriciopa/2023\\_1\\_ads\\_backend\\_frameworks\\_noite/blob/main/aula5/projeto/blog.zip](https://github.com/fabriciopa/2023_1_ads_backend_frameworks_noite/blob/main/aula5/projeto/blog.zip)
- Baixar o arquivo e extrair em uma pasta separada;



## Continuação do **trabalho**

- O arquivo descomprimido tem um arquivo `blog-template.html` e 4 outras pastas;
- Na sua aplicação (blog), você deve adicionar um template base com o código `blog-template.html`;
- Você deve adequar o código para que os arquivos estáticos que são usados sejam utilizado dentro do seu app;





# Obrigado!

*Alguma pergunta?*

Contato:

☉ 040601692@prof.unama.br