

# Django Views



Prof. Bruno Gomes



@profbrunogomes



# Aula de Hoje

2

- » **Django:**
  - » **VIEWS**



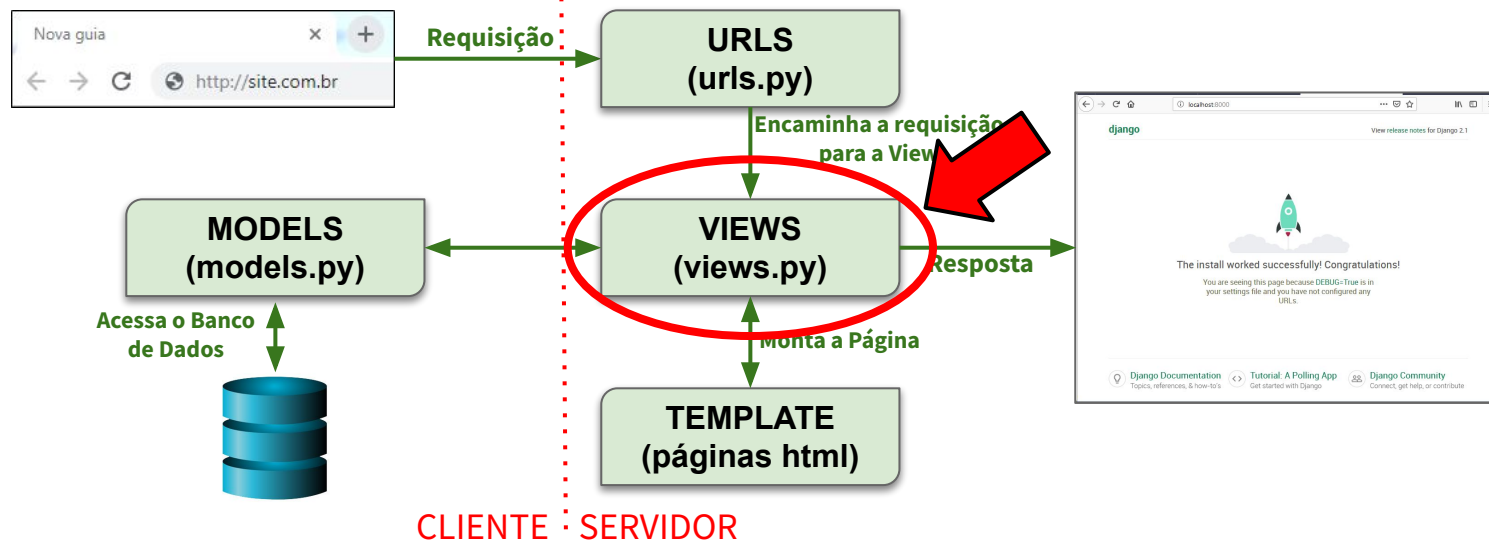
# 1.

## VIEWS

# VIEWS

4

» No Ciclo de Vida do Django:



# Documentação

5

» <https://docs.djangoproject.com/en/4.1/topics/http/views/>



# Views

6

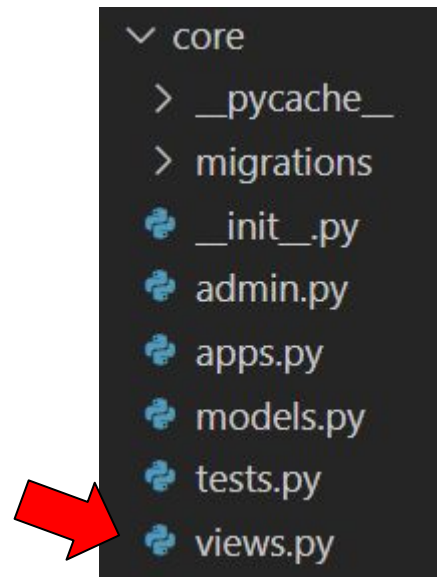
- » Onde são criadas funções que têm por parâmetro um objeto de requisição (request) e por retorno um objeto de resposta (HttpResponse);
- » É nela que informamos:
  - » Qual modelo deve ser instanciado, o que ele deve fazer;
  - » Qual template deve ser importado, como o valor deve ser exibido nele;
  - » Qual resposta deve ser enviada para o internauta (um HTML, um XML, um SVG, um redirecionamento, um erro 404, um erro 500, etc.).



# VIEWS

7

- » Fica no arquivo views.py
  - » Se criar uma aplicação, automaticamente o arquivo views.py é criado;
  - » Se não tiver aplicação, deve-se criar o arquivo views.py na pasta do projeto (no nosso exemplo, PrimeiroSite).



## Revisando 01 - Função Simples

8

```
from django.http import HttpResponse

def olamundo(request):
    return HttpResponse('Olá Mundo')
```

core/views.py



## Revisando 02 - Passando Parâmetro

9

```
def cadastro(request, idade):  
    return HttpResponse('Idade: '+str(idade))
```

core/views.py

- » Obs.: Sempre request deve ser o primeiro parâmetro, e esta variável pode ter qualquer nome, contanto que entenda que esta primeira variável contém um objeto request;

## Revisando 03 - Vários Parâmetros

10

```
def cadastro(request, nome, idade):  
    return HttpResponse('Idade: '+str(idade)+' Nome: '+nome)
```

[core/views.py](#)

# Funções Complementares

11

- » É possível também ter funções extras na view que não são chamadas pelas URL;
- » Elas servem de complemento às funções mapeadas.
- » Exemplo:
  - » Função para o cálculo da média;
  - » Função para realizar alguma operação no Banco de dados;
  - » Entre outras.

# Exemplo - Função para Calcular a Média

12

```
def media(num1, num2):  
    media = 0  
    media = (num1+num2)/2  
    return media  
  
def situacao(request, n1, n2):  
    if media(n1,n2) >= 6:  
        return HttpResponse("Aprovado")  
    else:  
        return HttpResponse("Reprovado")
```

Esta é a função que será  
mapeada pela urls.py

core/views.py

# Testando o exemplo anterior

13

» Em `urls.py`:

```
from core.views import olamundo, home, pagina1, pagina2, cadastro, situacao  
path('situacao/<int:n1>/<int:n2>/', situacao, name='situacao'),
```

PrimeiroSite/urls.py

» No navegador:

```
http://localhost:8000/situacao/8/10/
```

Navegador

# Funções com Parâmetros Opcionais

14

- » Exemplo: nome é opcional, apenas o CPF é obrigatório;
- » Acrescentar um valor padrão ao parâmetro nome:

```
def cadastro(request, nome, idade="Não informado"):  
    return HttpResponse('Nome: '+nome+'; Idade: '+str(idade))
```

core/views.py



## Testando o exemplo anterior

15

- » Em **urls.py**, adicionar o mesmo path sem o parâmetro nome:

```
path('dados/<str:nome>/<int:idade>/', cadastro),  
path('dados/<str:nome>/', cadastro),
```

PrimeiroSite/urls.py

- » No **navegador**, testar as 2 URLs abaixo:

```
http://localhost:8000/dados/Bruno/20  
http://localhost:8000/dados/Bruno
```

Navegador

# Dúvidas?

