

# Introdução ao Framework Django

Prof. Me. Thiago Ferauche  
thiago.ferauche@unisantos.br

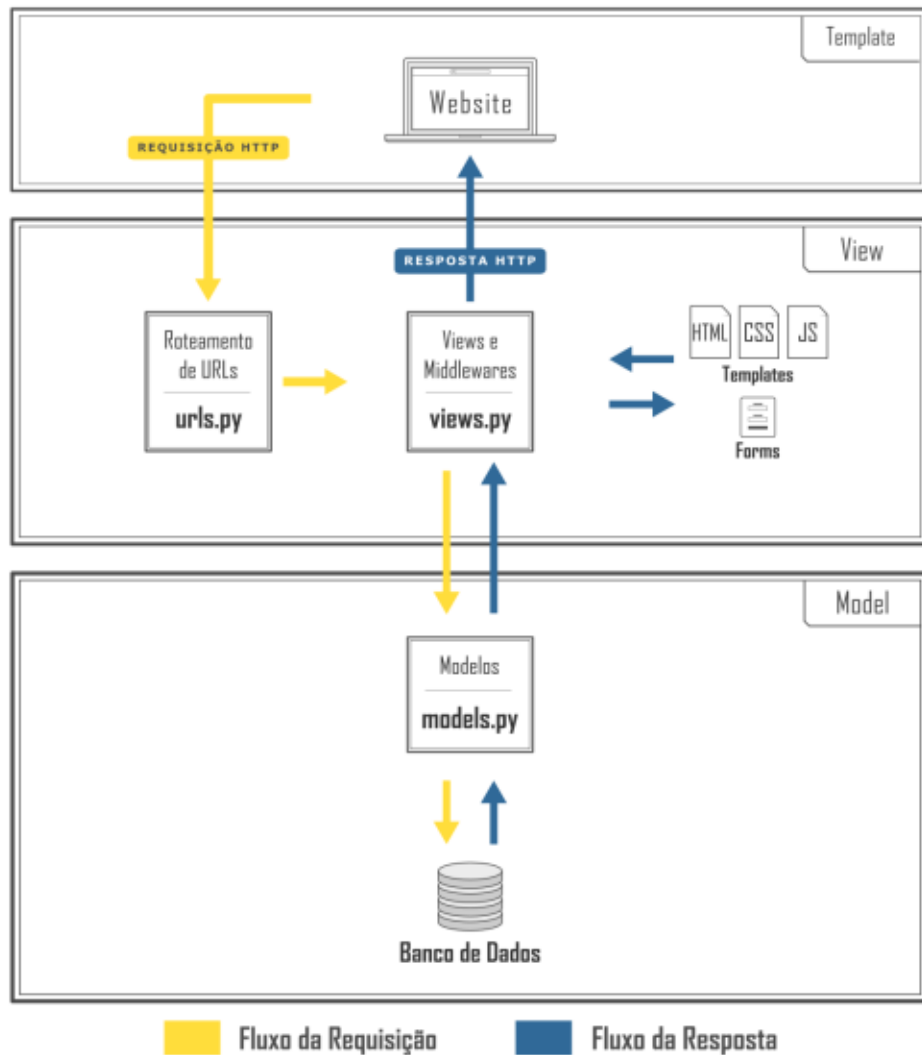
# Django o que é?

- Django é um framework de código-aberto escrito na linguagem Python, mantido pela Django Foundation;
- Django possui seu foco no desenvolvimento de aplicações para a web;
- Django vai além do MVC – Model View Control, conforme sua documentação os desenvolvedores o declaram como MTV – Model Template View



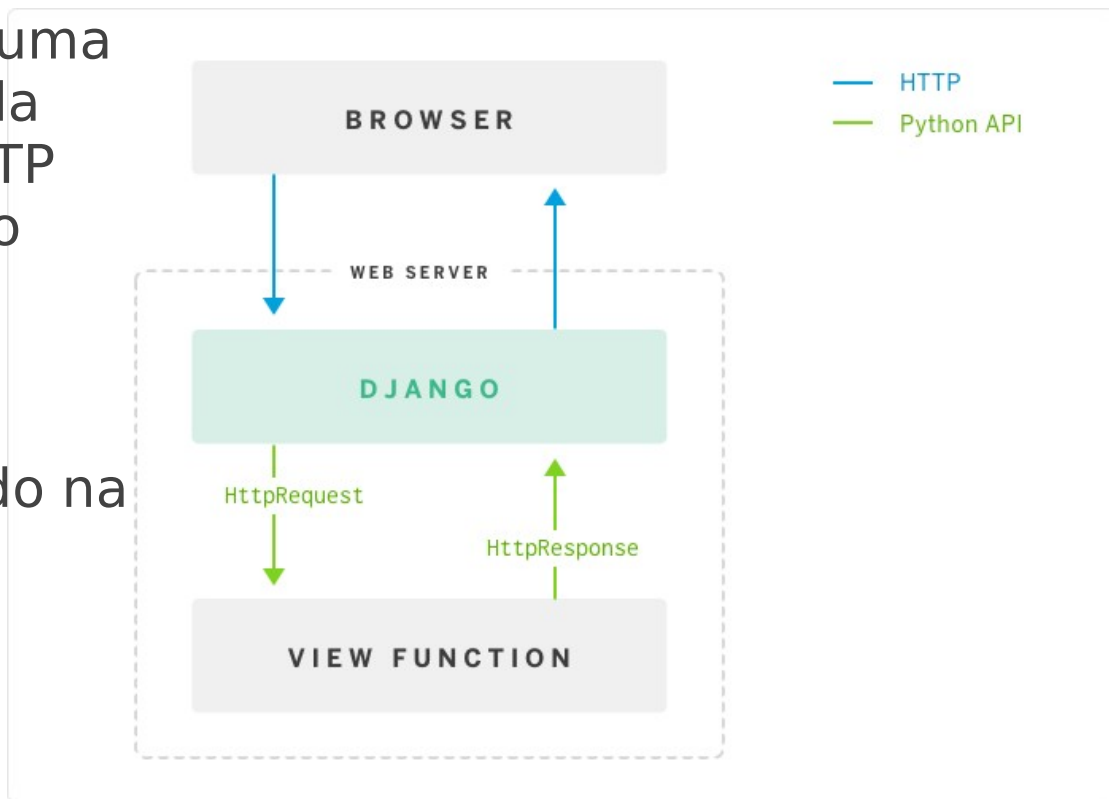
# MVT

## Model View Template



# Comunicação através da Web

- Toda a comunicação de uma aplicação web é realizada através do protocolo HTTP (camada de aplicação do TCP/IP);
- Protocolo HTTP é baseado na troca de requisições (requests) e respostas (response) HTTP;



# Verificação / Instalação ambiente Django

- Verificando se existe o django

```
python -m django --version
```

- Caso o comando acima não retorne a versão do django é necessário realizar a instalação do módulo (Requisito mínimo versão 3.8 do python).
- Recomenda-se a criação de um ambiente virtual python (venv) para seu projeto (Pycharm auxilia nisso). Com o ambiente virtual criado e ativado em seu projeto, a instalação pode ser feita:

```
python -m pip install Django
```

# Criando um projeto django

- Dentro do diretório de seu projeto, com o django previamente instalado, utilize a linha abaixo alterando o mysite pelo nome do projetoweb que deseja criar:

```
django-admin startproject mysite
```

- Serão gerados os arquivos iniciais de seu projeto

```
mysite/  
  manage.py  
  mysite/  
    __init__.py  
    settings.py  
    urls.py  
    asgi.py  
    wsgi.py
```

## Conhecendo os principais arquivos:

- **manage.py** arquivo usado em linha de comando para gerenciar seu projeto web.
- **settings.py** arquivo usado para manter as configurações de servidor web.
- **urls.py** arquivo que mantém os endereços a serem roteados em seu servidor web.

```
mysite/  
    manage.py  
mysite/  
    __init__.py  
    settings.py  
    urls.py  
    asgi.py  
    wsgi.py
```

# Iniciando servidor de desenvolvimento

- O Django já vem com seu servidor para desenvolvimento local, basta usar a linha de comando abaixo, estando no mesmo diretório de manage.py:

```
python manage.py runserver
```

- Abrir no navegador a URL `http://127.0.0.1:8000`, deve ser respondida a página de boas-vindas do Django.
- Caso precise alterar a porta de comunicação do servidor basta usar a linha abaixo:  

```
python manage.py runserver 8080
```
- URL do servidor será `http://127.0.0.1:8080`



# Criando uma aplicação web

- Dentro do mesmo diretório do arquivo manage.py, utilize o comando abaixo, onde polls é o nome da aplicação, pode ser alterado para o nome da sua aplicação:

```
python manage.py startapp polls
```

- Irá criar o diretório com o nome da sua aplicação com os arquivos conforme a figura ao lado.
- Para este momento iremos alterar o arquivo views.py

```
polls/  
    __init__.py  
    admin.py  
    apps.py  
    migrations/  
        __init__.py  
    models.py  
    tests.py  
    views.py
```

# Alterando o polls/views.py

- Alteraremos o arquivo polls/views.py para criarmos uma resposta HTTP em formato texto comum.

polls/views.py



```
from django.http import HttpResponseRedirect

def index(request):
    return HttpResponseRedirect("Hello, world. You're at the polls index.")
```

# Alterando o polls/urls.py

- Criaremos o arquivo polls/urls.py para vincular o padrão do nome da url com a view desejada.

polls/urls.py



```
from django.urls import path

from . import views

urlpatterns = [
    path('', views.index, name='index'),
]
```

# Alterando o mysite/urls.py

- Alteraremos o arquivo mysite/urls.py para criar um padrão na url para minha aplicação e incluir as regras de urls da minha aplicação.

mysite/urls.py



```
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('polls/', include('polls.urls')),
    path('admin/', admin.site.urls),
]
```

# Verificando as alterações

- Inicie o servidor, caso ele ainda não esteja iniciado  
`python manage.py runserver`
- Acesse no navegador `http://localhost:8000/polls`
  - Deverá retornar a mensagem de texto colocada no método `index`, dentro do arquivo `polls/view.py`

## **Exercícios**

Crie um projeto django, e uma aplicação web dentro deste projeto com uma view retornando uma página html customizada por você.