

Django

Django

- Django é um framework para desenvolvimento rápido para web, escrito em Python, que utiliza o padrão model-template-view
- Pip é um sistema de gerenciamento de pacotes padrão de fato usado para instalar e gerenciar pacotes de software escritos em Python

Instalando o Django

- Utilize o comando

```
pip install django
```

Criando o projeto

```
python -m django startproject NOME_DO_PROJETO
```

- Mas utilizando o ponto não cria uma pasta dentro da outra

```
python -m django startproject NOME_DO_PROJETO .
```

Para iniciar o servidor

```
python manage.py runserver NUMERO_DA_PORTA
```

- Para desligar o servidor

Ctrl + C

Ctrl + break

Criando App

- Um app é responsável por um módulo da aplicação e permite uma maior organização e legibilidade do projeto.
- Para criar o app, utilizamos o comando
`python manage.py startapp NOME_DO_APP`

Visão (View)

- Quando falamos de camada view ou visão, estamos falando da tela onde o usuário visualizará o nosso sistema.
- Então para começo de conversa, vamos abrir o arquivo "urls.py" do seu app e coloque o código a seguir:

```
from django.urls import path  
from . import views
```

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

Visão (View)

- Repare que na nossa variável urlpatterns, temos um "views.index"

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

- Isso significa que devemos ter um método no arquivo "views.py" do seu app.

```
from django.http import HttpResponse
```

```
def index(request):  
    return HttpResponse("Olá Mundo!")
```


Visão (View)

- Para finalizar devemos configurar a vinculação do app no projeto:
 1. Incluir a linha do app na variável "INSTALLED_APPS" no arquivo "settings.py" do projeto
 2. Incluir uma linha de acesso do seu app na variável "urlpatterns" do arquivo "views.py" do projeto

```
path('NOME_DO_APP', include('NOME_DO_APP.urls'))
```

Models (Modelos)

- Configurando o banco de dados:
 - Na variável "DATABASES" em "settings.py" do projeto
 - O Django trabalha com outros banco de dados, como MySQL, Oracle e outros, mudando apenas o tipo de engine

<http://docs.djangoproject.com/en/4.1/ref/settings/#databases>

Models (Modelos)

- Configurando o banco de dados:

```
python manage.py migrate
```

- Logo criaremos as nossas classes, no arquivo "models.py" do seu app.

- Após as classes criadas

```
python manage.py makemigrations
```

```
python manage.py migrate
```

Criando usuários ao administrador

```
python manage.py createsuperuser
```

Para adicionar o seu modelo no admin

- Basta adicionar as seguintes linhas no arquivo "admin.py" do seu app

```
from .models import NOME_DA_CLASSE  
admin.site.register(NOME_DA_CLASSE)
```