



Django do Zero - Como Criar um Aplicativo Web

01 - Frameworks WEB e Django

Wanderson Reis



@VamosCodar
Inscreva-se!

Framework na programação !?

É um conjunto de bibliotecas especialmente organizadas que servem de **arcabouço** (estrutura, esqueleto) para projetos de software.

Um framework de programação permite encapsular implementações complexas em um formato mais fácil e adaptável para uso. Por exemplo: Bootstrap (para CSS).

Mais detalhes:

<https://pt.stackoverflow.com/questions/17501/qual-é-a-diferença-de-api-biblioteca-e-framework>

Vantagens dos Frameworks

A principal vantagem é agregar rotinas, funções, modelos e recursos já implementados. Este fato acelera muito o desenvolvimento de soluções simples ou complexas.

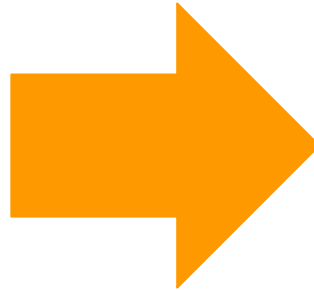
Desvantagens dos Frameworks

Direcionam o estilo, exigem seguir um **padrão rigoroso**.

Convenções e configurações se sobrepõem à criatividade do programador na solução de determinados problemas.

O desenvolvedor deve conhecer como ele funciona, e pode limitar determinadas necessidades que o framework não suporta nativamente.

Biblioteca -> Framework -> Solução



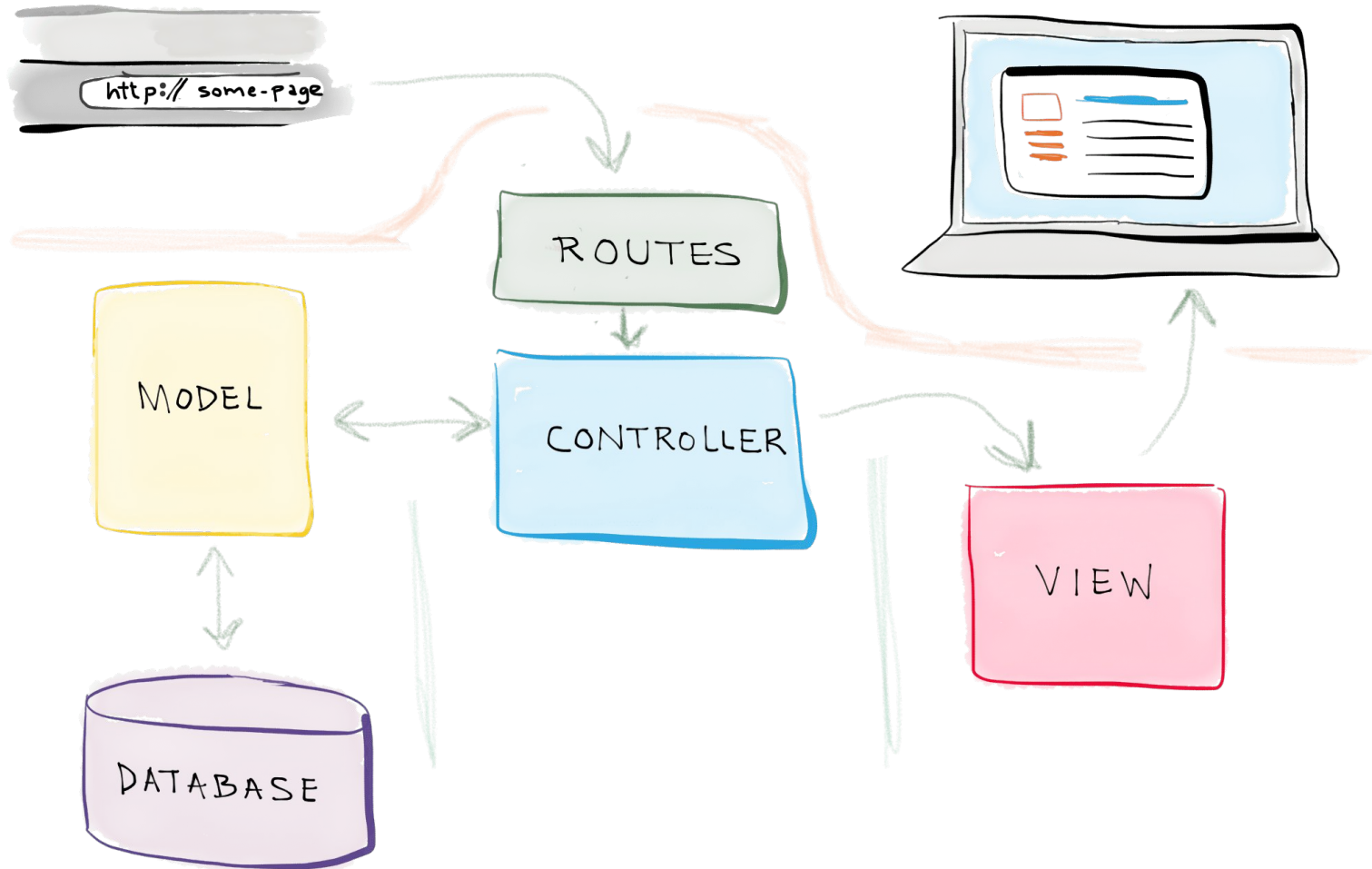
MVC - *Model, View e Controller*

MVC (*Model, View e Controller*) é a arquitetura adotada pela maior parte dos frameworks de desenvolvimento Web.

Divide o sistema em **camadas**, de forma que possam ***trabalhar em conjunto*** mas sendo independentes ao mesmo tempo.

Permite alto grau de **reaproveitamento** e a integração das equipes (programadores) especializados em cada camada.

MVC - Conceitual WEB



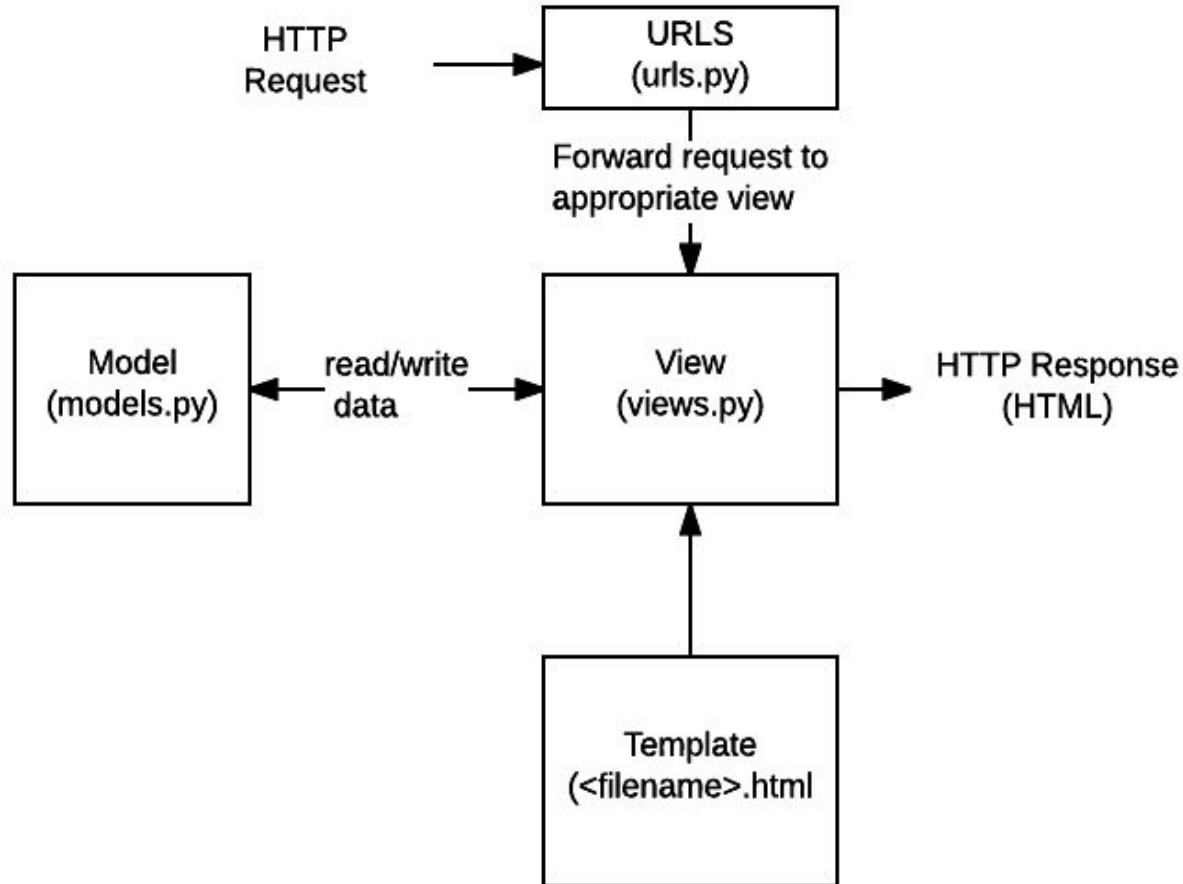
Django - um Framework Web

O Django é um framework web full stack open source (código aberto) escrito em Python, gratuito e de alto nível.

O Django habilita o programador a construir de forma rápida sistemas que usam pouco código.

Um projeto Django se divide em pequenas aplicações baseadas no modelo MVT (*Model-View-Template*).

MVT - Model, View e Template



Principais Características - Django

- Desenvolvimento rápido de aplicações WEB;
- Incluí toda a base para o sistema: autenticação, área administrativa, formulários, rotas, sistema de templates, etc;
- Poderoso ORM (*Object Relational Mapper*);
- Robustez comprovada pelo tempo;
- Focado em segurança e manutenibilidade;
- Portável pois usa a linguagem Python;

Feito com Django

Sites Using Django

Disqus

MacArthur Foundation

Open Knowledge Foundation

Instagram

Mozilla

Pinterest

Knight Foundation

National Geographic

Open Stack

<https://www.djangoproject.com/start/overview/>

Princípios - Django

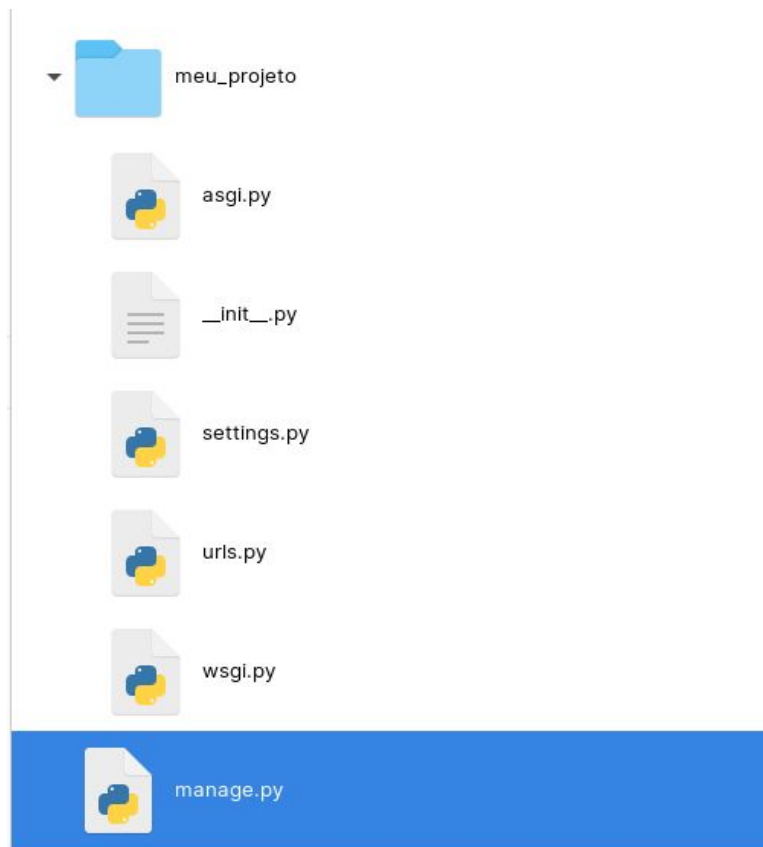
- **KISS*** (*keep it simple and stupid*);
- **DRY** (*don't repeat yourself*);
- Models “encorpado”, e views “enxutas”;
- Colocar o mínimo possível da **lógica de negócio** no template;
- Criar apps “pequenos” e reutilizáveis;

Um app no Django é um módulo Python independente que contém seus próprios models, views, templates e testes.

(*) <https://uxdesign.blog.br/a-origem-do-keep-it-simple-stupid-kiss-b24085dc1327>

Estrutura de um Projeto Django

```
/django$ django-admin startproject meu_projeto
```



Arquivos Importantes do Projeto

settings.py

É o arquivo de configuração do sistema / projeto Django: conexão com banco de dados, apps instalados e ativos, métodos e formatos de cache, logs, etc.

urls.py

O arquivo urls.py do projeto mapeia as rotas do sistema

manage.py

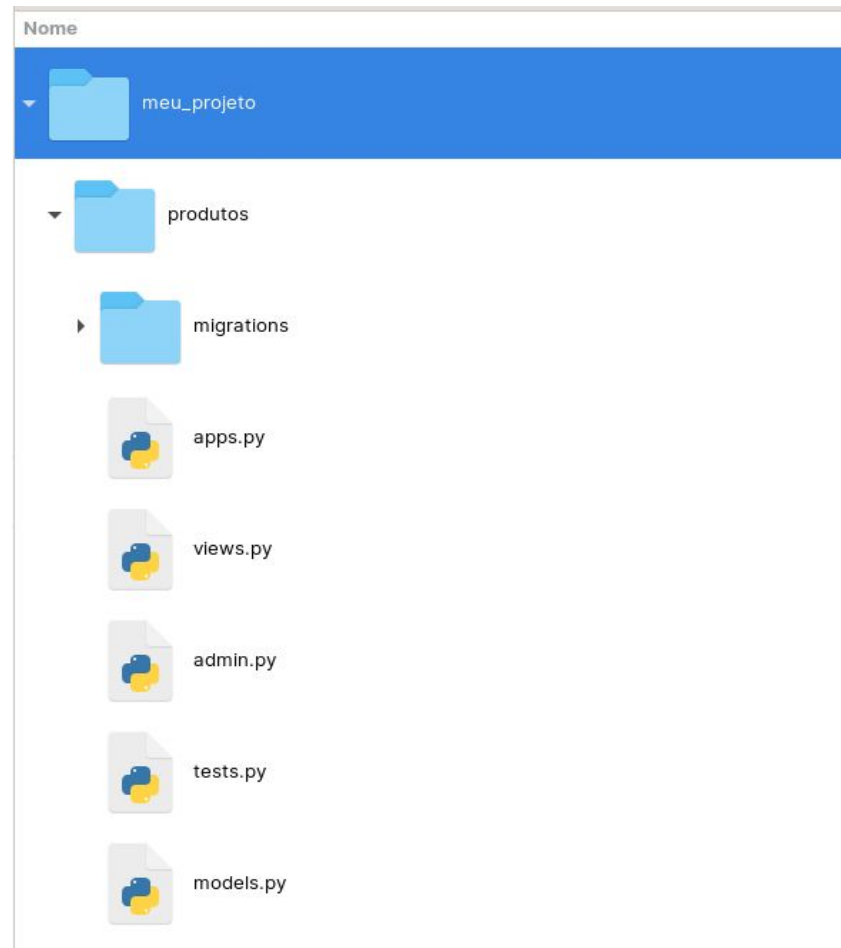
É o script administrativo do sistema / projeto Django. Através dele que se executa manutenção dos dados e diversas outras tarefas importantes como criação de usuário administrador.

Estrutura de um App Django

```
o/meu_projeto$ django-admin startapp produtos
```

Tipicamente dentro da pasta do App existirá uma pasta **templates** com os arquivos HTML (com as marcações específicas do Django).

Em geral, cada **módulo** ou **funcionalidade** terá um App próprio, Exemplo: produtos, pedidos, clientes, etc.



Arquivos Importantes do App

models.py

É o arquivo onde será descrita no formato de classes os modelos de dados do App, incluindo as regras de negócio.

views.py

Contém as funções que transformam uma requisição HTTP em uma resposta (HTML, JSON, xls, txt, pdf, etc).

migrations

É a pasta onde o Django manterá o versionamento dos modelos de dados do App. É gerenciado automaticamente com pouca ou nenhuma intervenção do desenvolvedor.

Dúvidas ?

- 1) Escreva sua dúvidas nos comentários
- 2) Envie sugestões de tópicos sobre o Django

Continuação: Preparação do ambiente