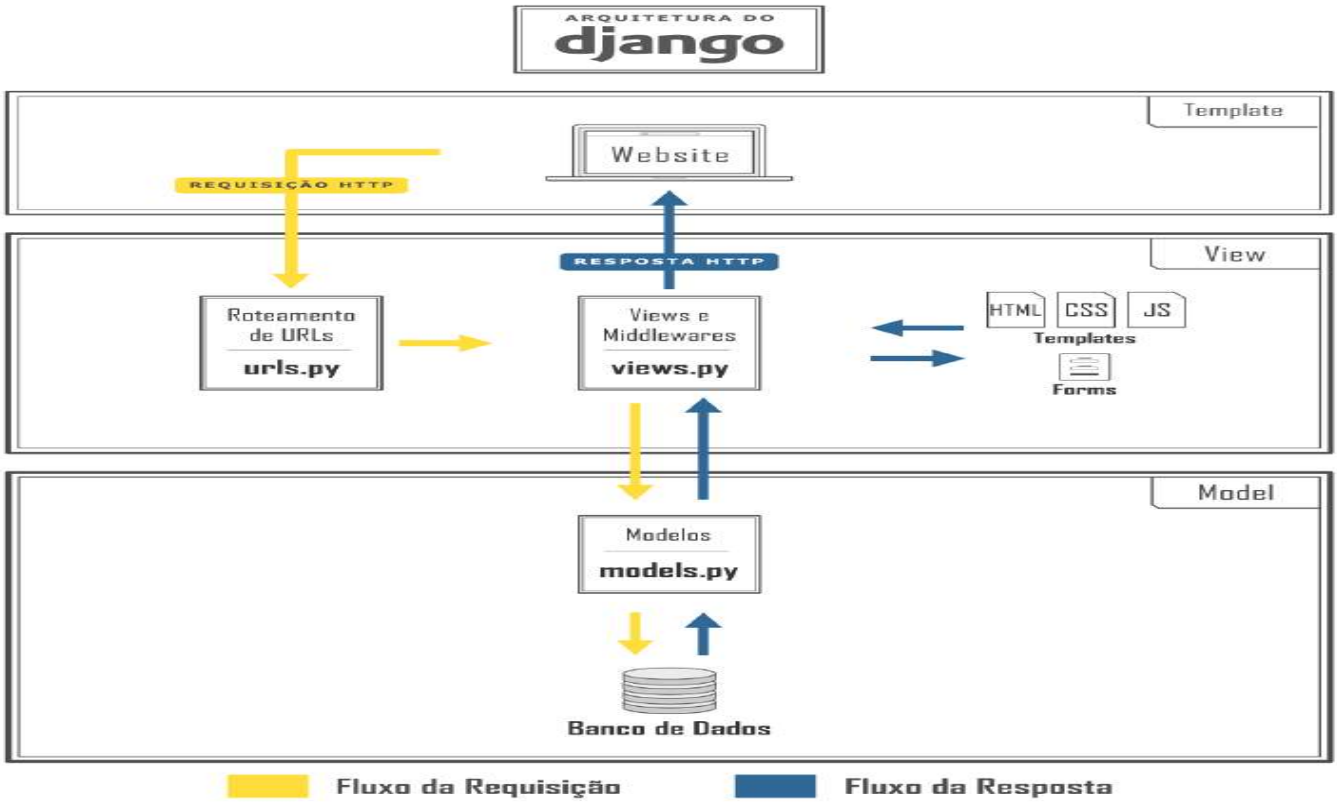
Projeto Django



1 - Criar/entrar na pasta do projeto

```
$ mkdir helloworld
$ cd helloworld
$ touch
```

2 - Criar ambiente virtual

```
$ python3 -m venv venv
```

With this command, I'm asking Python to run the venv package, which creates a virtual environment named venv. The first venv in the command is the name of the Python virtual environment package, and the second is the virtual environment name that I'm going to use for this particular environment. If you find this confusing, you can replace the second venv with a different name that you want to assign to your virtual environment.
In general I create my virtual environments with the name venv in the project directory, so whenever I cd into a project I find its corresponding virtual environment.

If you are using any version of Python older than 3.4 (and that includes the 2.7 release), virtual environments are not supported natively. For those versions of Python, you need to download and install a third-party tool called virtualenv before you can create virtual environments. Once virtualenv is installed, you can create a virtual environment with the following command:

```
$ virtualenv venv
```

3 - Ativar ambiente virtual

```
$ source venv/bin/activate
(venv) $
```

4 – Instalando DJANGO

pip install django

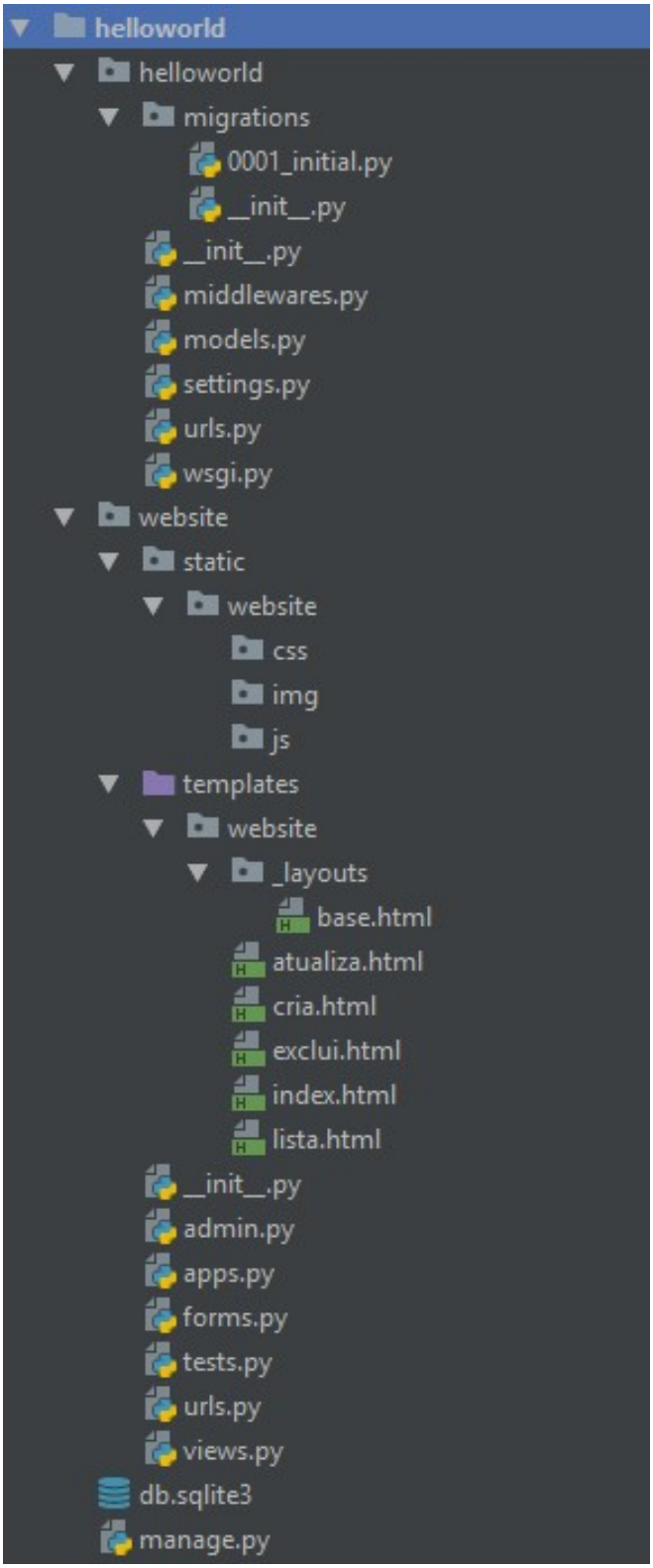
ou para copiar de outro projeto
na raiz do projeto original
pip freeze > requirements.txt
na raiz do novo projeto
pip install -r requirements.txt

5 – Criando a estrutura do projeto

django-admin.py startproject helloworld .

6 – Iniciar o projeto
na raiz do servidor
python manage.py runserver

7 – Criando o APP neste caso o app se chama website
django-admin.py startapp `website`
Agora, vamos criar algumas pastas para organizar a estrutura da nossa aplicação.
Primeiro, crie a pasta templates dentro de website.
Dentro dela, crie uma pasta website e dentro dela, uma pasta chamada _layouts. Crie também a pasta static dentro de website, para guardar os arquivos estáticos (arquivos CSS, Javascript, imagens, fontes, etc). Dentro dela crie uma pasta website, por questões de namespace. Dentro dela, crie: uma pasta css, uma pasta img e uma pasta js.



8 - Atualizando a configuração INSTALLED_APPS no arquivo de configuração helloworld/settings.py

```
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
```

```
'django.contrib.staticfiles',
'helloworld',
'website'
]
```

Agora, vamos fazer algumas alterações na estrutura do projeto para organizar e centralizar algumas configurações.
Primeiro, vamos passar o arquivo de modelos `models.py` de `/website` para `/helloworld`, pois os arquivos comuns ao projeto vão ficar centralizados no app `helloworld` (geralmente temos apenas um arquivo `models.py` para o projeto todo). Como não temos mais o arquivo de modelos na pasta `/website`, podemos, então, excluir a pasta `/migrations` e o `migrations.py`, pois estes serão gerados e gerenciados pelo app `helloworld`.

Estrutura inicial do projeto

```
/helloworld
    - __init__.py
    - settings.py
    - urls.py
    - wsgi.py
    - models.py
/website
    - __init__.py
    - admin.py
    - apps.py
    - tests.py
    - views.py
- manage.py
```