



django  
UnCHAINED

# Quem sou eu ?

## Paola Katherine Pacheco

- Graduanda em Análise de Sistemas pela Unesa
- Ex-graduanda de Estatística na Uerj
- Apaixonada por números , viagens e chocolate
- Desenvolvedora Back End Python
- Membro do Django Girls Rio, Pyladies Rio
- GDG Organizer Rio de Janeiro
- Women TechMakers Rio de Janeiro

[about.me/pkcpweb](https://about.me/pkcpweb)





**Python** é uma linguagem de programação criada por Guido van Rossum em 1991. Os objetivos do projeto da linguagem eram: produtividade e legibilidade. Em outras palavras, **Python** é uma linguagem que foi criada para produzir código bom e fácil de manter de maneira rápida.

**Instalando o Python :**

[http://tutorial.djangogirls.org/pt/python\\_installation/](http://tutorial.djangogirls.org/pt/python_installation/)

# Meu primeiro comando em Python

Para testar o Python, digite no seu terminal :

*Python*

Com isso seu terminal se tornará iterativo e poderá testar o código abaixo com enter no final:

```
>>>2+3
```

```
5
```

```
>>>len("Ola")
```

```
3
```



**Django** é um framework gratuito e de código aberto para a criação de aplicações web, escrito em Python. É um framework web, ou seja, é um conjunto de componentes que ajuda a desenvolver sites de forma mais rápida e mais fácil.

# Virtual Env Wrapper

Para facilitar, usaremos o Virtual Env Wrapper.

O Virtual Env é um ambiente virtual que fará com que todas as dependências do projeto fiquem em um diretório só.

Para instalar digite o comando abaixo:

```
$ pip install virtualenvwrapper
```

*(criar PastaDoProjeto)*

E criaremos uma virtual env:

```
$ mkvirtualenv NomeDoProjeto
```

# Instalando o Django

E agora sim instalaremos o Django:

*\$ pip install Django*

\*Sem definir a versão o pip sempre irá instalar a mais recente.

```
→ projeto_hangout git:(develop) ✗ mkvirtualenv proj-hang
New python executable in /Users/Hadronstech/.virtualenvs/proj-hang/bin/python2.7
Also creating executable in /Users/Hadronstech/.virtualenvs/proj-hang/bin/python
Installing setuptools, pip, wheel...done.
virtualenvwrapper.user_scripts creating /Users/Hadronstech/.virtualenvs/proj-hang/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/Hadronstech/.virtualenvs/proj-hang/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/Hadronstech/.virtualenvs/proj-hang/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/Hadronstech/.virtualenvs/proj-hang/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/Hadronstech/.virtualenvs/proj-hang/bin/get_env_details
(proj-hang) → projeto_hangout git:(develop) ✗ pip install django
Collecting django
  Using cached Django-1.9.6-py2.py3-none-any.whl
Installing collected packages: django
Successfully installed django-1.9.6
```

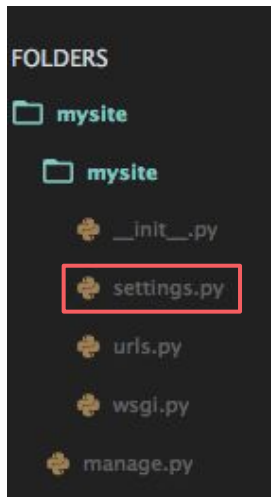
# Começando seu projeto



```
$ django-admin startproject meuprojeto
```



# Começando o seu projeto...



Precisamos alterar algumas configurações no *settings.py*

```
LANGUAGE_CODE = 'pt-BR'

TIME_ZONE = 'America/Sao_Paulo'
```

```
STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static')
```

*LANGUAGE\_CODE*="pt-BR"

*TIME\_ZONE*='America/Sao\_Paulo'

*STATIC\_URL* = '/static/'

*STATIC\_ROOT* = *os.path.join(BASE\_DIR, 'static')*

## Banco de dados

O banco de dados que usaremos é o padrão sqlite3.

Executaremos apenas o comando: `$ python manage.py migrate`

```
DATABASES = {  
    ... 'default': {  
        ... 'ENGINE': 'django.db.backends.sqlite3',  
        ... 'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
    ... }  
}
```

```
(proj-hang) → mysite git:(develop) * python manage.py migrate
```

```
Operations to perform:
```

```
  Apply all migrations: admin, contenttypes, auth, sessions
```

```
Running migrations:
```

```
  Rendering model states... DONE
```

```
  Applying contenttypes.0001_initial... OK
```

```
  Applying auth.0001_initial... OK
```

```
  Applying admin.0001_initial... OK
```

```
  Applying admin.0002_logentry_remove_auto_add... OK
```

```
  Applying contenttypes.0002_remove_content_type_name... OK
```

```
  Applying auth.0002_alter_permission_name_max_length... OK
```

```
  Applying auth.0003_alter_user_email_max_length... OK
```

```
  Applying auth.0004_alter_user_username_opts... OK
```

```
  Applying auth.0005_alter_user_last_login_null... OK
```

```
  Applying auth.0006_require_contenttypes_0002... OK
```

```
  Applying auth.0007_alter_validators_add_error_messages... OK
```

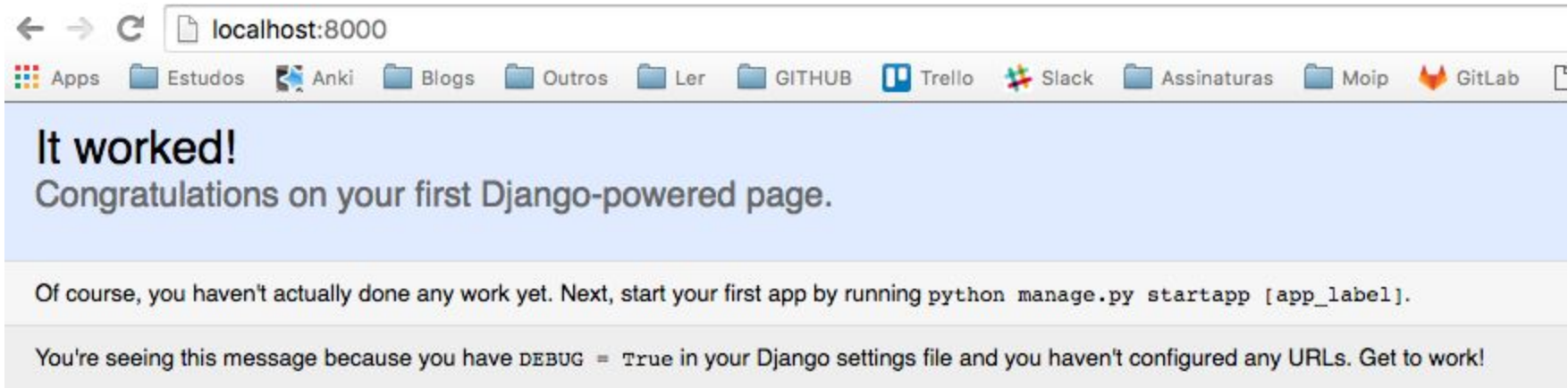
```
  Applying sessions.0001_initial... OK
```

# Testando

```
(proj-hang) → mysite git:(develop) ✕ python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
May 30, 2016 - 21:36:17
Django version 1.9.6, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

*\$ python manage.py runserver*





Git é um sistema de controle de versão de arquivos. Através deles podemos desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas.



O Github é um serviço web que oferece diversas funcionalidades extras aplicadas ao git. Resumindo, você poderá usar gratuitamente o github para hospedar seus projetos pessoais.

*Além disso, quase todos os projetos/frameworks/bibliotecas sobre desenvolvimento open source estão no github, e você pode acompanhá-los através de novas versões, contribuir informando bugs ou até mesmo enviando código e correções.*

# GitHub

O que você precisa para começar a usar o GitHub:

- Ter uma conta no GitHub
- Autenticação via SSH pública ( para que você possa commitar )

<http://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de-perguntar/>


<https://git-scm.com/book/pt-br/v1/Git-no-Servidor-Gerando-Sua-Chave-P%C3%BAblica-SSH>

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name

 PKpacheco ▾

/


meu-portfolio ✓


Great repository names are short and memorable. Need inspiration? How about **animated-octo-happiness**.

Description (optional)

App em Django para criação de portfólio pessoal.

- ☒

 **Public**  
Anyone can see this repository. You choose who can commit.
- ☐

 **Private**  
You choose who can see and commit to this repository.

- ☐

**Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾ ⓘ

Create repository

[Code](#)[Issues](#) 0[Pull requests](#) 0[Wiki](#)[Pulse](#)[Graphs](#)[Settings](#)

## Quick setup — if you've done this kind of thing before

[Set up in Desktop](#)

or

[HTTPS](#)[SSH](#)[git@github.com:PKpacheco/meu-portfolio.git](#)

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

## ...or create a new repository on the command line

```
echo "# meu-portfolio" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:PKpacheco/meu-portfolio.git
git push -u origin master
```



## ...or push an existing repository from the command line

```
git remote add origin git@github.com:PKpacheco/meu-portfolio.git
git push -u origin master
```



## ...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)



# GitHub

```
(proj-hang) → mysite git:(master) X echo "# meu-portfolio" >> README.md
```

```
(proj-hang) → mysite git:(develop) X git init  
Initialized empty Git repository in /Users/Hadronstech/git.hadrons.io/projetos/projeto_hangout/mysite/.git/
```

```
(proj-hang) → mysite git:(master) X git add README.md  
(proj-hang) → mysite git:(master) X git commit -m "first commit"  
[master (root-commit) 4713688] first commit  
1 file changed, 1 insertion(+)  
create mode 100644 README.md
```

```
(proj-hang) → mysite git:(master) X git remote add origin git@github.com:PKpacheco/meu-portfolio.git  
(proj-hang) → mysite git:(master) X git push -u origin master  
Counting objects: 6, done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (6/6), 457 bytes | 0 bytes/s, done.  
Total 6 (delta 0), reused 0 (delta 0)  
To git@github.com:PKpacheco/meu-portfolio.git  
* [new branch]      master -> master  
Branch master set up to track remote branch master from origin.
```

```
$ echo "# hangout_teste">> README.md
```


```
$ git init
```


```
$ git add README.md
```

```
$ git commit -m "first commit"
```


```
$ git remote add origin git@github.com:PKpacheco/meu-portfolio.git
```

```
$ git push -u origin master
```

 Paola Katherine first commit

 README.md

first commit

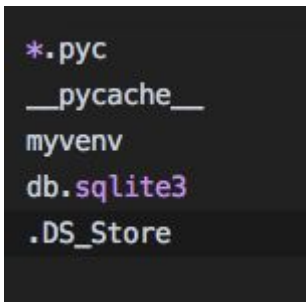
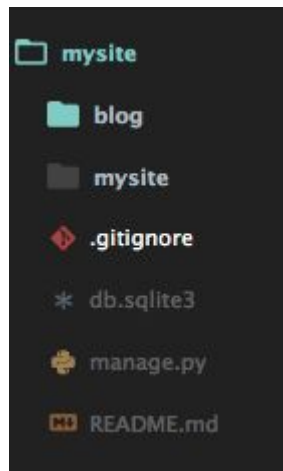
 README.md

# meu-portfolio

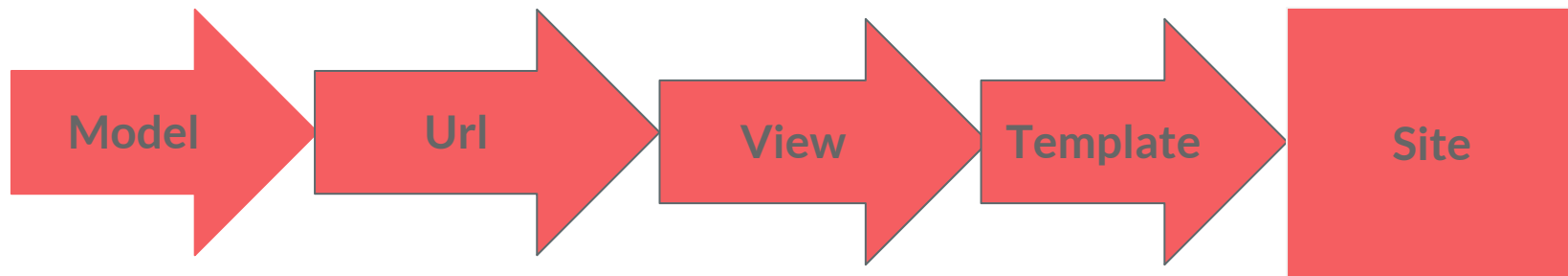
---

Crie um arquivo dentro do app principal chamado **.gitignore**  
Coloque nele arquivos que serão ignorados pelo Git

*\*\*'Commite' novamente*



# Estrutura do Django



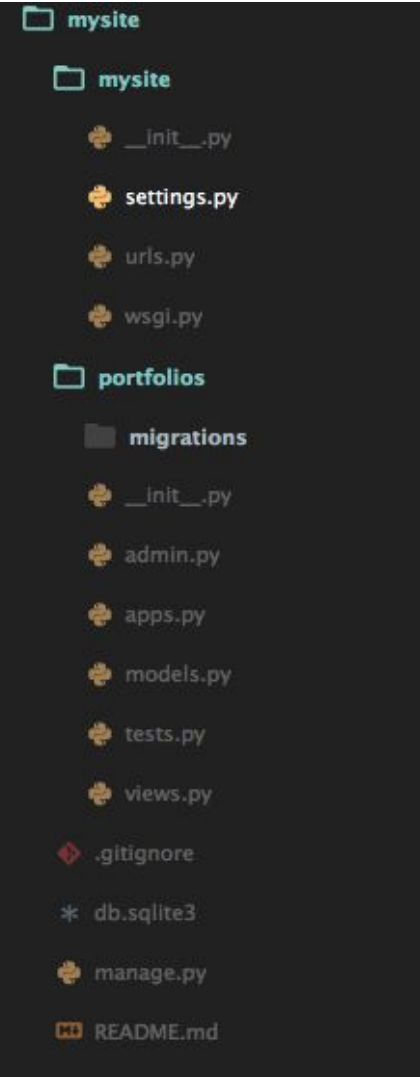
# App Django

- Vamos criar um app diferente dentro do nosso projeto.
- `$ python manage.py startapp portfolios`

```
(proj-hang) → mysite git:(master) X python manage.py startapp portfolios
```

- Precisamos colocar o novo app dentro do `settings.py`

```
INSTALLED_APPS = (  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'portfolios',  
)
```



# Modelos

models.py

```
# coding: utf-8
from django.db import models

class DadosPessoais(models.Model):
    ... name = models.CharField(max_length=50, verbose_name='Nome')
    ... adress = models.CharField(max_length=100, verbose_name='Endereço')
    ... city = models.CharField(max_length=50, verbose_name='Cidade')
    ... cep = models.CharField(max_length=50, verbose_name='Cep')
    ... phone = models.CharField(max_length=20, verbose_name='Telefone')
    ... mobile = models.CharField(max_length=20, verbose_name='Celular')

    ... about = models.TextField(max_length=255, verbose_name='Sobre')
    ... data_nascimento = models.CharField(max_length=255, default='01 de janeiro de 2000', verbose_name='Data Nascimento')
    ... email = models.EmailField(verbose_name='Email')

    ... conhecimentos = models.TextField(max_length=255, verbose_name='Conhecimentos')
    ... github = models.CharField(max_length=100, default='http://github.com/SeuGit', verbose_name='Github')
    ... current_position = models.CharField(max_length=100, verbose_name='Cargo atual')
    ... empresa = models.CharField(max_length=100, verbose_name='Empresa')

    ... def __str__(self):
    ...     return self.name
```

```
class DadosPessoais(models.Model):
```

class -> definição de objeto

DadosPessoais -> nome do modelo ( que é um objeto)

models.Model -> modelo do Django que será salvo no banco.

```
def __str__(self):
    ... return self.Name
```

método/função

# Modelos

*Precisamos aplicar as alterações ao banco de dados:*

*\$ python manage.py makemigrations*

*\$ python manage.py migrate*

```
(proj-hang) → mysite git:(master) ✖ python manage.py makemigrations
Migrations for 'portfolios':
  0001_initial.py:
    - Create model DadosPessoais
```

```
(proj-hang) → mysite git:(master) ✖ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, portfolios, contenttypes, auth, sessions
Running migrations:
  Rendering model states... DONE
  Applying portfolios.0001_initial... OK
```

# Admin

```
# coding: utf-8
from django.contrib import admin
from models import DadosPessoais

admin.site.register(DadosPessoais)
```

```
(proj-hang) → mysite git:(master) python manage.py createsuperuser
```

*\$ python manage.py createsuperuser*

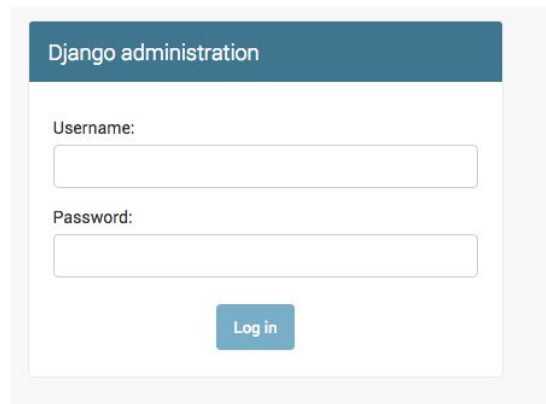
*\$ python manage.py runserver*

```
(proj-hang) → mysite git:(master) python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
June 01, 2016 - 18:33:58
Django version 1.9.6, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```



# Admin



The image shows the Django administration login page. It has a dark blue header with the text 'Django administration'. Below the header, there are two input fields: 'Username:' and 'Password:'. At the bottom, there is a blue button labeled 'Log in'.

O Django coloca um 's' no model informando plural!!!

Cole esse código no seu model , assim você força seu model a aceitar o nome no singular e no plural.

```
class Meta:
    verbose_name = 'Dados Pessoais'
    verbose_name_plural = 'Dados Pessoais'
```

## Django administration

WELCOME, **ADMIN**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

### Site administration

#### AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#) [Change](#)

Users [+ Add](#) [Change](#)

#### PORTFOLIOS

Dados pessoaiss [+ Add](#) [Change](#)

#### Recent Actions

##### My Actions

None available

# Url- Endereço da Web

*mysite/urls.py*

```
from django.conf.urls import include, url
from django.contrib import admin

urlpatterns = [
    ...url(r'^admin/', include(admin.site.urls)),
    ...url(r'', include('portfolios.urls')),
]
```

*Um pouco de Regex:*

^ para o início do texto

\$ para o final do texto

\d para um dígito

+ para indicar que o item anterior deve ser repetido pelo menos uma vez

() para capturar parte do padrão

Criaremos o arquivo:  
*portfolios/urls.py*

```
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.portfolio_exibir),
]
```

Próximo passo → Criar a View para **portfolio\_exibir**

# Views

Uma View pega informações do Model e exibe em um Template.  
Nela temos a "lógica" de negócio, da sua aplicação.

*portfolio/views.py*

```
from django.shortcuts import render

def portfolio_exibir(request):
    ... return render(request, 'portfolios/portfolio_exibir.html', {})
```

Temos um método *def*  
Renderizando para *portfolio\_exibir.html*

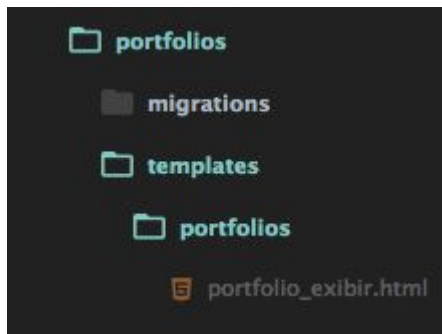
Executando *python manage.py runserver*

**TemplateDoesNotExist** at /  
portfolio/portfolio\_exibir.html

# Templates

Criaremos uma pasta **templates**, dentro dela uma outra pasta **portfolios** e um arquivo **portfolio\_exibir.html**

```
portfolios>template>portfolios>portfolio_exibir.html
```



# Templates Teste

*portfolio\_exibir.html*

```
<!DOCTYPE html>
<html>
<head>
  <title>Teste</title>
</head>
<body>
  <h1>Hello World</h1>

</body>
</html>
```

*python manage.py runserver*

**Hello World**

# Git

```
(proj-hang) → mysite git:(master) $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   mysite/settings.py
        modified:   mysite/urls.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        portfolios/

no changes added to commit (use "git add" and/or "git commit -a")
```

```
(proj-hang) → mysite git:(master) $ git add -A
(proj-hang) → mysite git:(master) $ git commit -m "criacao das urls e templates"
[master 628783a] criacao das urls e templates
13 files changed, 126 insertions(+)
create mode 100644 portfolios/__init__.py
create mode 100644 portfolios/admin.py
create mode 100644 portfolios/apps.py
create mode 100644 portfolios/migrations/0001_initial.py
create mode 100644 portfolios/migrations/0002_auto_20160603_1413.py
create mode 100644 portfolios/migrations/__init__.py
create mode 100644 portfolios/models.py
create mode 100644 portfolios/templates/portfolios/portfolio_exibir.html
create mode 100644 portfolios/tests.py
create mode 100644 portfolios/urls.py
create mode 100644 portfolios/views.py
(proj-hang) → mysite git:(master) $ git push origin master
Counting objects: 18, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (16/16), done.
Writing objects: 100% (18/18), 2.89 KiB | 0 bytes/s, done.
```

Branch: master ▾

New pull request



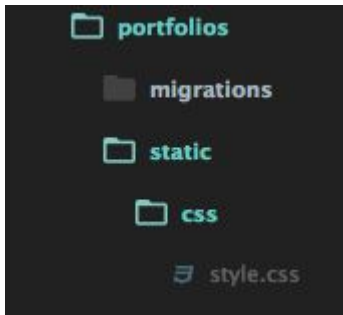
Paola Katherine criacao das urls e templates

mysite	criacao das urls e templates
portfolios	criacao das urls e templates
.gitignore	adicionando arquivo gitignore
README.md	first commit
manage.py	adicionando arquivo gitignore

Criaremos um arquivo **style.css** dentro da pasta **static** em **portfolios** e dentro dela outra pasta **css**.

**portfolios>static>css>style.css**

*style.css*



```
#show h1{
  font-family: 'Arial';
  font-weight: bold;
  font-size: 50px;
}

.negrit{
  font-weight: bold;
}

#show p{
  font-family: 'Arial';
}

#show a{
  font-family: 'Arial', cursive;
  text-decoration: none;
  color: #00BFB2;
  font-size: 20px;
}
```

<https://github.com/PKpacheco/meu-portfolio/blob/master/portfolios/static/css/style.css>



# Alterações na View

Faremos algumas alterações em **portfolios/views.py** para facilitar nosso Html

```
from django.shortcuts import render
from .models import DadosPessoais

def portfolio_exibir(request):
    ... pessoa = DadosPessoais.objects.all()
    ... context = {'pessoa': pessoa}

    ... return render(request, 'portfolios/portfolio_exibir.html', context)
```

# Html

```
<body>

...{% for pessoa in pessoa %}

.....<div class="container-fluid">
.....  <div class="section">
.....    <div class="container" id="show">
.....        <div class="row">
.....            <div class="col-md-8">
.....                <h1>{{pessoa.name}}</h1>
.....                <h3>{{pessoa.current_position}}</h3>
.....                <p><a href="{{pessoa.github}}"> <i>Conheça meu repositório</i></a></p>
.....            </div>
.....        </div>
.....    </div>
.....</div>

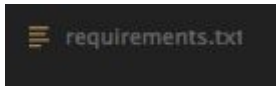
.....</div>
```

```
.....<div class="section">
.....  <div class="container">
.....    <div class="row">
.....        <div class="panel panel-primary">
.....            <div class="panel-heading">
.....                <h2 id="Resumo">Resumo</h2>
.....            </div>
.....            <div class="panel-body">
.....                <p><span class="negrit"></span>{{pessoa.about}}</p>
.....            </div>
.....        </div>
.....    </div>
.....</div>

.....</div>
```

[https://github.com/PKpacheco/meu-portfolio/blob/master/portfolios/templates/portfolios/portfolio\\_exibir.html](https://github.com/PKpacheco/meu-portfolio/blob/master/portfolios/templates/portfolios/portfolio_exibir.html)

# Requirements.txt



Um **requirements.txt** guarda todas as dependências instaladas na sua aplicação.

Verifique todas as dependências com :

**\$pip freeze**

```
(proj-hang) → meu-portfolio git:(master) X pip freeze  
Django==1.9.7
```

Em seguida, crie o arquivo:

***requirements.txt***

A dark rectangular box representing a file named "requirements.txt". It contains a single line of text: "Django==1.9.7".

```
requirements.txt  
  
Django==1.9.7
```

# Makefile

Um **Makefile** é um atalho para comandos.

A small icon representing a code editor window. It has a dark background with a light-colored border. Inside the window, there is a small icon of a code editor (a square with a diagonal line) followed by the text "Makefile".

```
clean:
—— find . -name "*.pyc" -exec rm -rf {} \;
run:
—— python manage.py runserver 0.0.0.0:8000
migrate:
—— python manage.py migrate
migrations:
—— python manage.py makemigrations
user:
—— python manage.py createsuperuser

shell:
—— python manage.py shell
```

# Comunidades



# Onde posso aprender?

## Python:

- <http://pycursos.com/python-para-zumbis/>
- Code Academy

## Django

- <http://welcometothedjango.com.br/>
- <https://osantana.me/curso-de-python-e-django/>
- [tutorial.djangogirls.org](http://tutorial.djangogirls.org)

# Obrigada

[about.me/pkcpweb](https://about.me/pkcpweb)