



django
UnCHAINED

Python



Python es un lenguaje de programación creada por Guido van Rossum en 1991. Los objetivos del proyecto del lenguaje eran: productividad y legibilidad.

En otras palabras, **Python** es un lenguaje que se creó para producir código bueno y fácil de mantener de manera rápida.

Instalando o Python :

http://tutorial.djangogirls.org/es/python_installation/

Mi primer comando en Python

Para probar Python, escriba en su terminal:

Python

Con eso tu terminal se volverá iterativo y podrás probar el código abajo con enter al final:

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

```
5
```

```
>>>len("Hola")
```

```
4
```



Django es un Framework gratuito y de código abierto para la creación de aplicaciones web, escrito en Python.

Framework web, es un conjunto de componentes que ayuda a desarrollar sitios de forma más rápida y fácil.

Virtual Env Wrapper

Para facilitar, usaremos Virtual Env Wrapper.

Virtual Env es un entorno virtual que hace que todas las dependencias del proyecto queden en un solo directorio.

Para instalar utilice el comando siguiente:

```
$ pip install virtualenvwrapper
```

Python2 x Python3

(crear CarpetaDelProyecto)

Y crearemos una virtual env:

```
$ mkvirtualenv NombreDelProyecto
```

```
$ mkvirtualenv --python=python3 NombreDelProyecto
```

```
→ mi_primer_app mkvirtualenv --python=python3 mi_proyecto
```

```
(mi_proyecto) → mi_primer_app python --version  
Python 3.6.4
```

Instalación de Django

Y ahora sí instalaremos el Django:

\$ pip install Django

* Sin establecer la versión el pip siempre instalará la última.

```
(mi_proyecto) → mi_primer_app pip install django
Collecting django
  Downloading Django-2.0.2-py3-none-any.whl (7.1MB)
    100% |████████████████████████████████████████| 7.1MB 148kB/s
Collecting pytz (from django)
  Using cached pytz-2018.3-py2.py3-none-any.whl
Installing collected packages: pytz, django
Successfully installed django-2.0.2 pytz-2018.3
```

Comenzando su proyecto

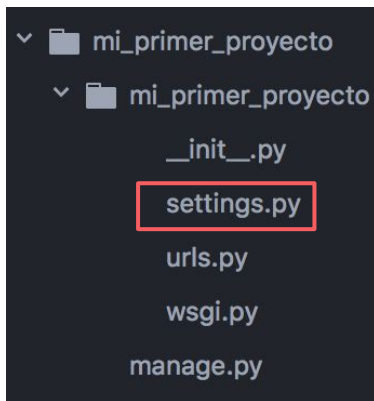


```
$ django-admin startproject mi_proyecto
```

```
(mi_proyecto) → mi_primer_app django-admin startproject mi_primer_proyecto
```


Comenzando su proyecto ...

Necesitamos cambiar algunas configuraciones en *settings.py*



```
LANGUAGE_CODE = 'es-ar'
```

```
TIME_ZONE = 'America/Argentina/Mendoza'
```

```
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')
```

Base de datos

La base de datos que utilizamos es el estándar sqlite3.

Ejecutaremos sólo el comando: *\$ python manage.py migrate*

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

```
(mi_proyecto) → mi_primer_proyecto python manage.py migrate
```

Operations to perform:

Apply all migrations: admin, auth, contenttypes, sessions

Running migrations:

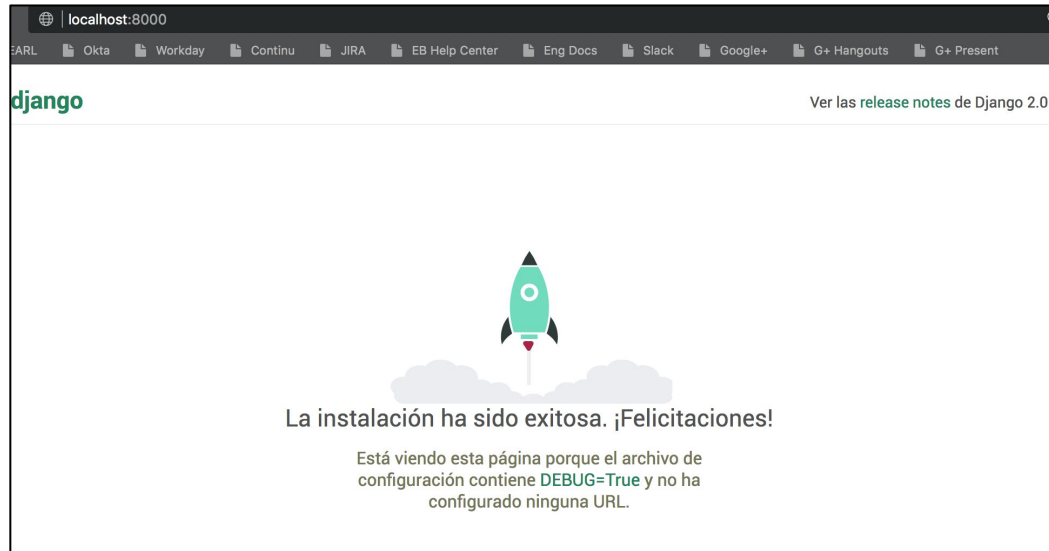
```
Applying contenttypes.0001_initial... OK  
Applying auth.0001_initial... OK  
Applying admin.0001_initial... OK  
Applying admin.0002_logentry_remove_auto_add... OK  
Applying contenttypes.0002_remove_content_type_name... OK  
Applying auth.0002_alter_permission_name_max_length... OK  
Applying auth.0003_alter_user_email_max_length... 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 auth.0008_alter_user_username_max_length... OK  
Applying auth.0009_alter_user_last_name_max_length... OK  
Applying sessions.0001_initial... OK
```

Probando

```
(mi_proyecto) → mi_primer_proyecto python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
February 24, 2018 - 19:31:27
Django version 2.0.2, using settings 'mi_primer_proyecto.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

\$ python manage.py runserver





Git es un sistema de control de versiones de archivos. A través de ellos podemos desarrollar proyectos en la cual varias personas pueden contribuir simultáneamente en el mismo, editando y creando nuevos archivos y permitiendo que los mismos puedan existir sin el riesgo de que sus alteraciones sean sobrescritas.



Github es un servicio web que ofrece varias funcionalidades adicionales aplicadas al git. Resumiendo, usted podrá usar gratuitamente el github para alojar sus proyectos personales.

Además, casi todos los proyectos / frameworks / bibliotecas sobre desarrollo de código abierto están en github, y usted puede acompañar a través de nuevas versiones, contribuir a informar de errores o incluso enviar código y correcciones.

Lo que usted necesita para empezar a usar GitHub:

- Tener una cuenta en GitHub
- Generar una clave SSH
- Autenticación vía SSH pública (para que usted pueda commitar)

[Search](#)[All](#)[Public](#)[Private](#)[Sources](#)[Forks](#)[Mirrors](#)[New](#)

Create a new repository

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

Owner



Repository name



Great repository names are short and memorable. Need inspiration? How about **congenial-system**.

Description (optional)



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

 PKpacheco / mi_primer_proyecto

 Unwatch  1  0  Fork  0

 Code  Issues 0  Pull requests 0  Projects 0  Wiki  Insights  Settings


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

 Set up in Desktop or  HTTPS  SSH 

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


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

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




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

```
git remote add origin git@github.com:PKpacheco/mi_primer_proyecto.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

```
(mi_proyecto) → mi_primer_proyecto git:(master) ✕ echo "# mi_primer_proyecto" >> README.md
```

```
(mi_proyecto) → mi_primer_proyecto git init
```

```
Initialized empty Git repository in /Users/ppacheco/Dev/mi_primer_app/mi_primer_proyecto/.git/
```

```
(mi_proyecto) → mi_primer_proyecto git:(master) ✕ git add README.md
```

```
(mi_proyecto) → mi_primer_proyecto git:(master) ✕ git commit -m "first commit"
```

```
[master (root-commit) 3e7a509] first commit
```

```
1 file changed, 2 insertions(+)
```

```
create mode 100644 README.md
```

```
(mi_proyecto) → mi_primer_proyecto git:(master) ✕ git remote add origin git@github.com:PKpacheco/mi_primer_proyecto.git
```

```
(mi_proyecto) → mi_primer_proyecto git:(master) ✕ git push -u origin master
```

```
Counting objects: 3, done.
```

```
Writing objects: 100% (3/3), 234 bytes | 234.00 KiB/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0)
```

```
To github.com:PKpacheco/mi_primer_proyecto.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 REPO_GITHUB
```

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



PKpacheco change readme

Latest commit 4cdc94b 12 seconds ago



mi_primer_proyecto

change readme

11 seconds ago



README.md

change readme

11 seconds ago



db.sqlite3

change readme

11 seconds ago



manage.py

change readme

11 seconds ago

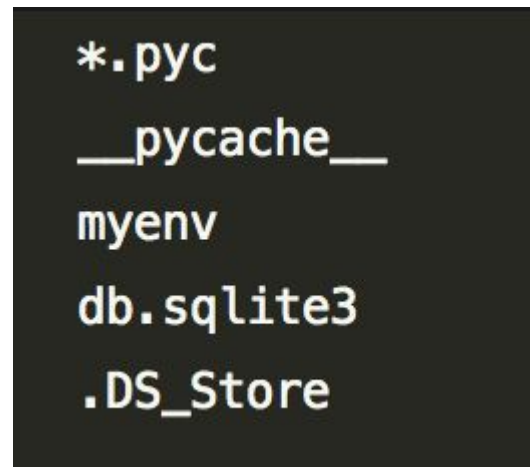
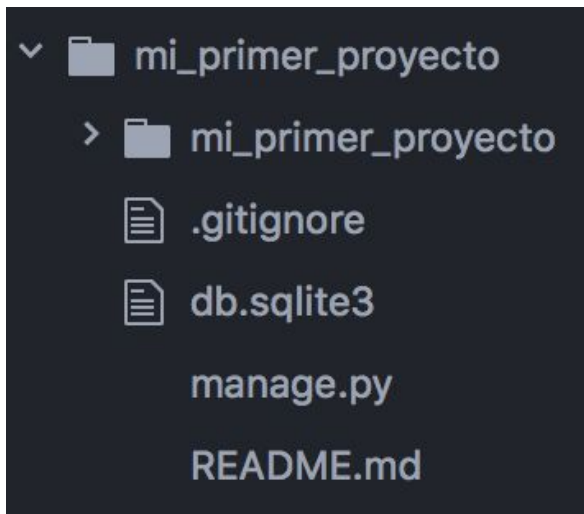


README.md

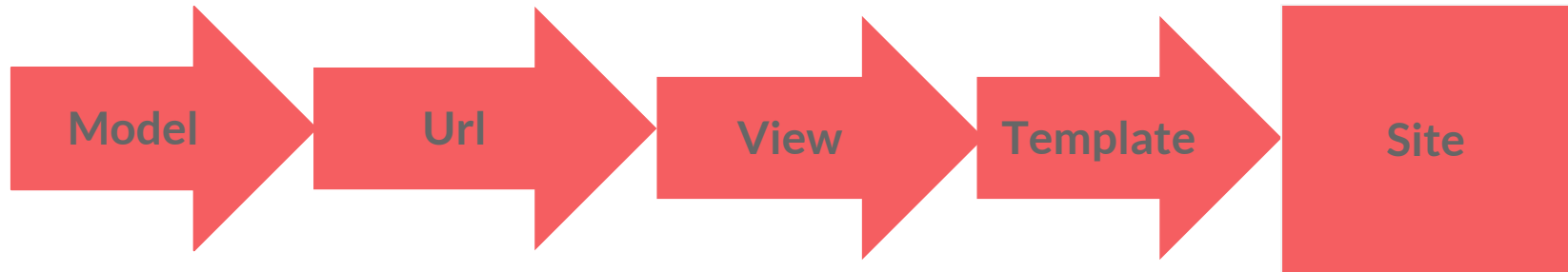
mi_primer_proyecto

Crear un archivo dentro de la aplicación principal llamada .gitignore
Ponga en él archivos que serán ignorados por Git

*** Hacer otro commit*



Estructura Django



App Django

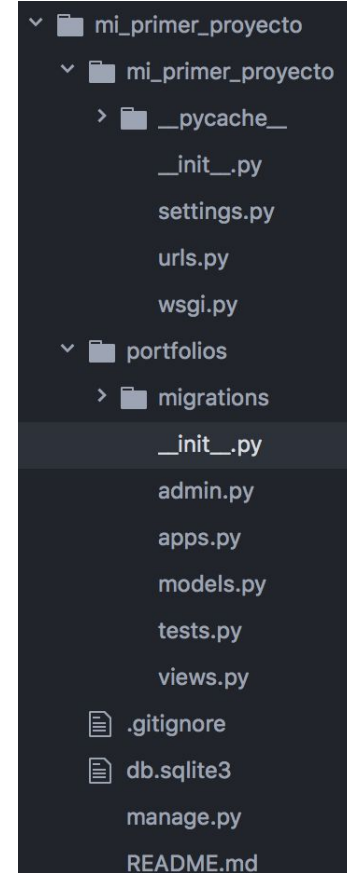
Vamos a crear una aplicación diferente dentro de nuestro proyecto.

- `$ python manage.py startapp portfolios`

```
(mi_proyecto) → mi_primer_proyecto git:(master) python manage.py startapp portfolios
```

- Necesitamos colocar el nuevo APP dentro de los `settings.py`

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



```
mi_primer_proyecto  
├── mi_primer_proyecto  
│   ├── __pycache__  
│   ├── __init__.py  
│   ├── settings.py  
│   ├── urls.py  
│   └── wsgi.py  
├── portfolios  
│   └── migrations  
└── __init__.py  
    ├── admin.py  
    ├── apps.py  
    ├── models.py  
    ├── tests.py  
    └── views.py  
    ├── .gitignore  
    ├── db.sqlite3  
    ├── manage.py  
    └── README.md
```

Modelos

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

class PersonalData(models.Model):
    ...name = models.CharField(max_length=50, verbose_name="Nombre")
    ...adress = models.CharField(max_length=100, verbose_name="Ubicación")
    ...city = models.CharField(max_length=50, verbose_name="Ciudad")
    ...postal_code = models.CharField(max_length=10, verbose_name="Postal Code")
    ...phone = models.CharField(max_length=20, verbose_name="Telefono")
    ...mobile = models.CharField(max_length=20, verbose_name="Celular")

    ...about = models.TextField(max_length=255, verbose_name="Sobre")
    ...birth_date = models.CharField(max_length=50, default='01 enero 2000', verbose_name="Fecha Nacimiento")
    ...email = models.EmailField(verbose_name="Email")

    ...knowledge = models.CharField(max_length=50, verbose_name="Conocimientos")
    ...github = models.CharField(max_length=100, default='https://github.com/YourGithub ', verbose_name="Github")
    ...current_position = models.CharField(max_length=50, verbose_name="Empleo Actual")
    ...companie = models.CharField(max_length=50, verbose_name="Empresa")

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

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

class -> definición de objeto

DatosPersonal -> nombre del modelo (que es un objeto)

models.Model -> modelo de Django que será guardado en el banco.

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

método/function

Modelos

Necesitamos aplicar los cambios a la base de datos:

\$ *python manage.py makemigrations*

\$ *python manage.py migrate*

```
(mi_proyecto) → mi_primer_proyecto git:(master) ✗ python manage.py makemigrations
Migrations for 'portfolios':
  portfolios/migrations/0001_initial.py
    - Create model PersonalData
```

```
(mi_proyecto) → mi_primer_proyecto git:(master) ✗ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, portfolios, sessions
Running migrations:
  Applying portfolios.0001_initial... OK
```

Admin

```
# coding: utf-8

from django.contrib import admin
from .models import PersonalData

admin.site.register(PersonalData)
```

```
(mi_proyecto) → mi_primer_proyecto git:(master) ✕ python manage.py createsuperuser
```

\$ python manage.py createsuperuser

\$ python manage.py runserver

```
(mi_proyecto) → mi_primer_proyecto git:(master) ✕ python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
February 25, 2018 - 12:13:24
Django version 2.0.2, using settings 'mi_primer_proyecto.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```


Admin

Administración de Django

Nombre de usuario:

Contraseña:

Identificarse

El Django pone un 's' en el modelo informando plural !!

Pegue ese código en su modelo, así que usted fuerza a su modelo a aceptar el nombre en el singular y el plural.

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

Administración de Django

BIENVENIDO/A, **ADMIN**. [VER SITIO](#) / [CAMBIAR CONTRASEÑA](#) / [CERRAR SESIÓN](#)

Administración de sitio

AUTENTICACIÓN Y AUTORIZACIÓN		
Grupos	+ Agregar	✎ Modificar
Usuarios	+ Agregar	✎ Modificar
PORTFOLIOS		
Personal datas	+ Agregar	✎ Modificar

Acciones recientes

Mis acciones

Ninguna disponible

Url- Endereço da Web

mi_primer_proyecto/urls.py

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

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

Crearemos el archivo:
portfolios/urls.py

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

urlpatterns = [
    url(r'^$', views.portfolio_show, name='show_portfolios'),

]
```

Un poco 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

Siguiente paso → Crear la View para **portfolio_show**

Views

Una View toma información de Model y se muestra en un Template.
En una View tenemos la "lógica" de negocio, de su aplicación.

portfolio/views.py

```
from django.shortcuts import render

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

Temos um método *def*
Renderizando para *portfolio_exibir.html*

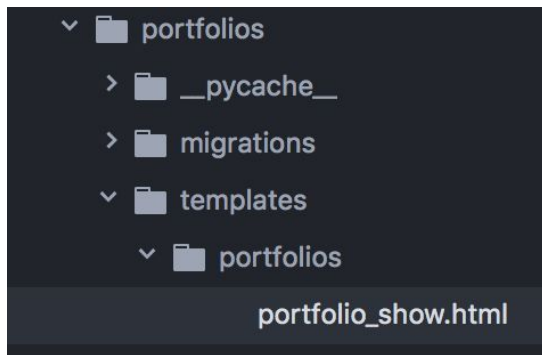
Executando *python manage.py runserver*

TemplateDoesNotExist at /
portfolios/portfolio_show.html

Templates

Vamos a crear una carpeta **templates**, dentro de ella otra carpeta **portfolios** y um archivo **portfolio_show.html**

```
portfolios>template>portfolios>portfolio_show.html
```



Templates Teste

portfolio_show.html

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

python manage.py runserver

Hello World

Git

```
(mi_proyecto) → mi_primer_proyecto 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:   db.sqlite3
        modified:   mi_primer_proyecto/__pycache__/settings.cpython-36.pyc
        modified:   mi_primer_proyecto/__pycache__/urls.cpython-36.pyc
        modified:   mi_primer_proyecto/settings.py
        modified:   mi_primer_proyecto/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")
```

```
(mi_proyecto) → mi_primer_proyecto git:(master) ✖ git add -A
(mi_proyecto) → mi_primer_proyecto git:(master) ✖ git commit -m "create urls, views and templates"
[master d6a41ea] create urls, views and templates
15 files changed, 98 insertions(+), 1 deletion(-)
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/__init__.py
create mode 100644 portfolios/models.py
create mode 100644 portfolios/templates/portfolios/portfolio_show.html
create mode 100644 portfolios/tests.py
create mode 100644 portfolios/urls.py
create mode 100644 portfolios/views.py
(mi_proyecto) → mi_primer_proyecto git:(master) ✖ git push origin master
Counting objects: 21, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (19/19), done.
Writing objects: 100% (21/21), 4.50 KiB | 922.00 KiB/s, done.
Total 21 (delta 6), reused 0 (delta 0)
remote: Resolving deltas: 100% (6/6), completed with 5 local objects.
To github.com:PKpacheco/mi_primer_proyecto.git
642ce30..d6a41ea master -> master
```

4 commits

1 branch

0 releases

1 contributor

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

PKpacheco create urls, views and templates

Latest commit d6a41ea a minute ago

mi_primer_proyecto	create urls, views and templates	a minute ago
portfolios	create urls, views and templates	a minute ago
.gitignore	create gitignore	2 hours ago
README.md	change readme	17 hours ago
db.sqlite3	create urls, views and templates	a minute ago
manage.py	change readme	17 hours ago

README.md

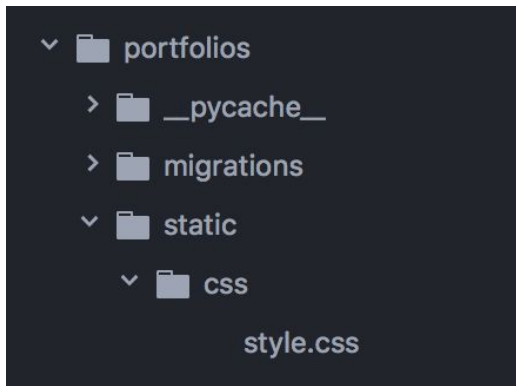
mi_primer_proyecto

git add .
git add -A

Vamos a crear un archivo **style.css** dentro de la carpeta **static** en **portfolios** u dentro de ella otra carpeta **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/mi_primer_proyecto/blob/master/portfolios/static/css/style.css

Alterações na View

Haremos alguns cambios en `portfolios/views.py` para facilitar nuestro Html

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

def portfolio_show(request):
    . . . person = PersonalData.objects.all()
    . . . context = {'person': person}

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

Htмл

```
{% load static %}
<!DOCTYPE html>
<html>
<head>

... <link href="http://pingendo.github.io/pingendo-bootstrap/themes/default/bootstrap.css"
... rel="stylesheet" type="text/css">

... <link href="{% static 'css/style.css' %}" rel="stylesheet" type="text/css">

</head>

<body>


... {% for person in person %}

... <div class="container-fluid">
... <div class="section">
... <div class="container" id="show">
... <div class="row">
... <div class="col-md-8">
... <h1>{{person.name}}</h1>
... <h3>{{person.current_position}}</h3>
... <p><a href="{{person.github}}"> <i>Conocer mi repositorio</i></a></p>
```

```
... <div class="section">
... <div class="container">
... <div class="row">
... <div class="panel panel-primary">
... <div class="panel-heading">
... <h2>Datos personales</h2>
... </div>
... <div class="panel-body">
... <p><span class="negrit">Fecha Nacimiento: </span>{{person.birth_date}}</p>
... <p><span class="negrit">Ubicacion : </span>{{person.adress}}</p>
... <p><span class="negrit">Ciudad :</span>{{person.city}} <span class="negrit">Postal Code: </span>
... <span>{{person.posta_code}}</p>
... <p><span class="negrit">Tel :</span> {{person.phone}} <span class="negrit"> Cel: </span>
... <span>{{person.mobile}}</p>
... <p><span class="negrit">E-mail :</span>{{person.email}}</p>
... </div>
... </div>
... </div>
... </div>
```

https://github.com/PKpacheco/mi_primer_proyecto/blob/master/portfolios/templates/portfolios/portfolio_show.html

Requirements.txt

 requirements.txt

Un **requirements.txt** guarda todas las dependencias instaladas en su aplicación.

Compruebe todas las dependencias con:

\$pip freeze

```
(mi_proyecto) → mi_primer_proyecto git:(master) x pip freeze
Django==2.0.2
pytz==2018.3
```

Em seguida, crie o arquivo:

requirements.txt

```
requirements.txt
```

```
Django==2.0.2
```

```
pytz==2018.3
```

Makefile

Un **Makefile** es un acceso directo a los comandos.



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
```

https://github.com/PKpacheco/mi_primer_proyecto/blob/master/Makefile

Gracias