

**TECNOLOGIA EM SISTEMAS PARA INTERNET**

**Kálita Rodrigues de Souza**

**DOCUMENTO DE IMPLANTAÇÃO  
PROJETO EXPERIENCEIN**

**Brasília - DF**

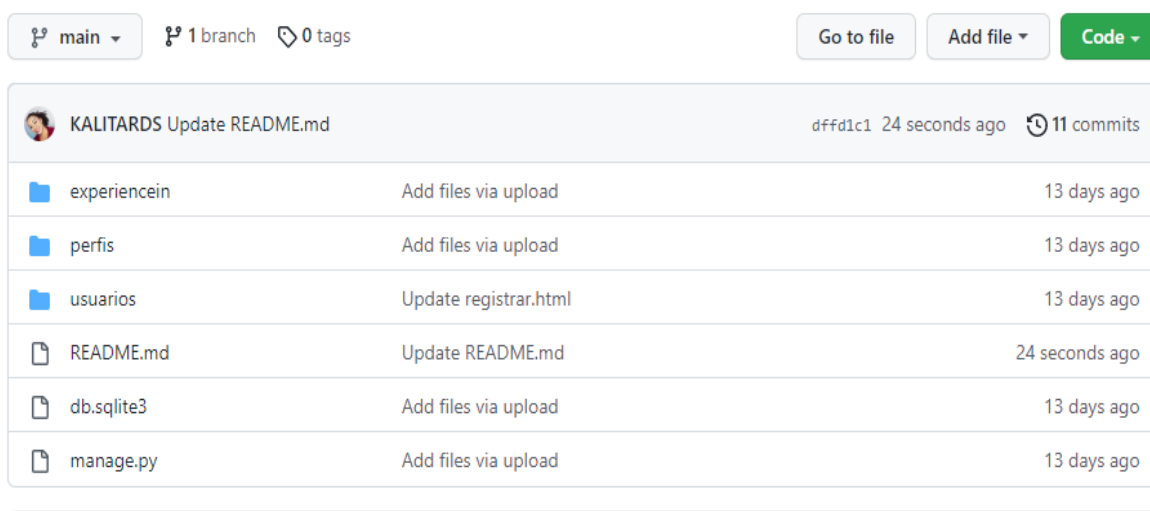
**14/07/2022**

# Sumário

1. Projeto no GitHub	3
2. Configuração da aplicação web no pythonnanywhere.	3
3. Vincular o ambiente virtual com a aplicação web criada.	6
4. Configuração do WSGI.	6
5. Edição do <i>settings.py</i> / <i>configuração dos arquivos estáticos</i>	9
5. Links	11

# 1. Projeto no GitHub

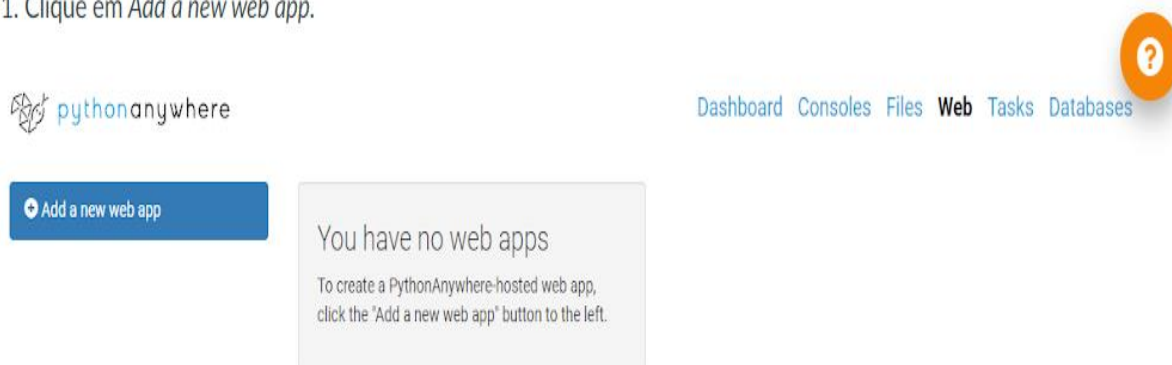
Para inicialização enviei todos os arquivos necessários para o GitHub.



## 2. Configuração da aplicação web no pythonanywhere.

No pythonanywhere comecei criando minha conta <https://www.pythonanywhere.com/>, e configurei a aplicação web da seguinte forma:

1. Clique em *Add a new web app*.



2.

 Add a new web app

Create new web app

### Your web app's domain name

Your account doesn't support custom domain names, so your PythonAnywhere web app will live at `olive1rafhm.pythonanywhere.com`

Want to change that? [Upgrade now!](#)

Otherwise, just click "Next" to continue.

Cancel

« Back

Next »



3.

 Add a new web app

Create new web app

### Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

- » Django
- » web2py
- » Flask
- » Bottle
- » **Manual configuration (including virtualenvs)**

What other frameworks should we have here? Send us some feedback using the link at the top of the page!

Cancel

« Back

Next »



4.

Create new web app

Select a Python version

» Python 2.7

» Python 3.4

» Python 3.5

» Python 3.6

» Python 3.7

Cancel

« Back

Next »

5.

Create new web app

Manual Configuration

Manual configuration involves editing your own WSGI configuration file in `/var/www/`. Usually this imports a WSGI-compatible application which you've stored elsewhere

When you click "Next", we will create a WSGI file for you, including a simple "Hello World" app which you can use to get started, as well as some comments on how to use other frameworks.

You will also be able to specify a *virtualenv* to use for your app.

Cancel

« Back

Next »

KalitaSouza1.pythonanywhere.com

➔ Add a new web app

Configuration for KalitaSouza1.pythonanywhere.com

Reload:

↻ Reload KalitaSouza1.pythonanywhere.com

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Saturday 01 October 2022**

Run until 3 months from today

[Paying users'](#) sites stay up forever without any need to log in to keep them running.

Aqui a nossa aplicação web já foi criada.

### 3. Vincular o ambiente virtual com a aplicação web criada.

Para isso, fui na seção ilustrada na imagem abaixo na página de configuração da web app e entrei com o nome do ambiente virtual que criei (*experiencein-virtualenv*).

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here.](#) You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

Enter path to a virtualenv, if desired

### 4. Configuração do WSGI.

Eu abri o arquivo de configuração WSGI da aplicação web em uma aba nova. Bastou clicar no Link *WSGI configuration file* na seção, igual mostra a imagem abaixo.

Code:

What your site is running.

Source code: [Enter the path to your web app source code](#)

Working directory: [/home/KalitaSouza1/](#)

[Go to directory](#)

WSGI configuration file: [/var/www/kalitasouza1\\_pythonanywhere\\_com\\_wsgi.py](#)

Python version: 3.9 

Ao clicar no link aparece assim:

 [/var/www/kalitasouza1\\_pythonanywhere\\_com\\_wsgi.py](#) Keyboard shortcuts: ☐ No

```
1 # web application at http://KalitaSouza1.pythonanywhere.com/
2 # It works by setting the variable 'application' to a WSGI handler of some
3 # description.
4 #
5
6 # ++++++ GENERAL DEBUGGING TIPS ++++++
7 # getting imports and sys.path right can be fiddly!
8 # We've tried to collect some general tips here:
9 # https://help.pythonanywhere.com/pages/DebuggingImportError
10
11
12 # ++++++ HELLO WORLD ++++++
13 # A little pure-wsgi hello world we've cooked up, just
14 # to prove everything works. You should delete this
15 # code to get your own working.
16
17
18 HELLO_WORLD = """<html>
19 <head>
20   <title>PythonAnywhere hosted web application</title>
21 </head>
22 <body>
23   <h1>Hello, World!</h1>
24   <p>
25     This is the default welcome page for a
26     <a href="https://www.pythonanywhere.com/">PythonAnywhere</a>
27     hosted web application.
28   </p>
29   <p>
30     Find out more about how to configure your own web application
31     by visiting the <a href="https://www.pythonanywhere.com/web_app_setup/">web app setup</a> page
32   </p>
33 </body>
34 </html>"""
35
```

O arquivo contém templates para a configuração do WSGI, e como utilizei Django, precisei fazer algumas alterações no código como excluir arquivos desnecessários, remover os comentários do código da sessão Django e alterar duas linhas para a aplicação começar a funcionar, que no meu caso foram as linhas 80 e 84 como mostra a imagem abaixo.



```
59
60 # ++++++ CUSTOM WSGI ++++++
61 # If you have a WSGI file that you want to serve using PythonAnywhere, perhaps
62 # in your home directory under version control, then use something like this:
63 #
64 #import sys
65 #
66 #path = '/home/KalitaSouza1/path/to/my/app'
67 #if path not in sys.path:
68 #    sys.path.append(path)
69 #
70 #from my_wsgi_file import application # noqa
71
72
73 # ++++++ DJANGO ++++++
74 # To use your own django app use code like this:
75 import os
76 import sys
77 #
78 ## assuming your django settings file is at '/home/KalitaSouza1/mysite/mysite/settings.py'
79 ## and your manage.py is at '/home/KalitaSouza1/experiencein'
80 path = '/home/KalitaSouza1/experiencein'
81 if path not in sys.path:
82     sys.path.append(path)
83
84 os.environ['DJANGO_SETTINGS_MODULE'] = 'experiencein.settings'
85 #
86 ## then:
87 from django.core.wsgi import get_wsgi_application
88 application = get_wsgi_application()
89
90
```

A linha 80 deve conter o caminho completo para o projeto Django. Já a 84 tem que ser modificada inserindo o nome\_do\_projeto.settings. Para verificar e obter esse caminho fui no console, entrei na minha pasta e usei o comando pwd.

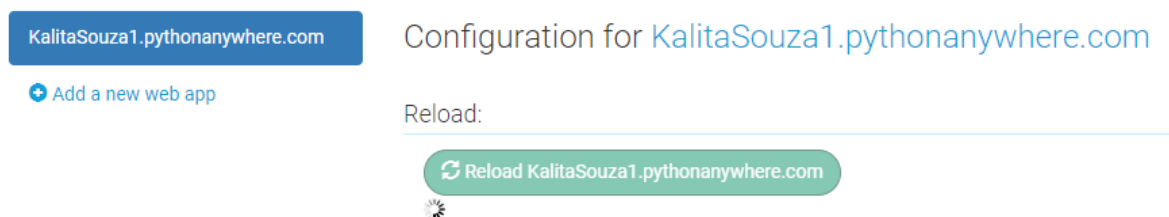


### Bash console 24812013

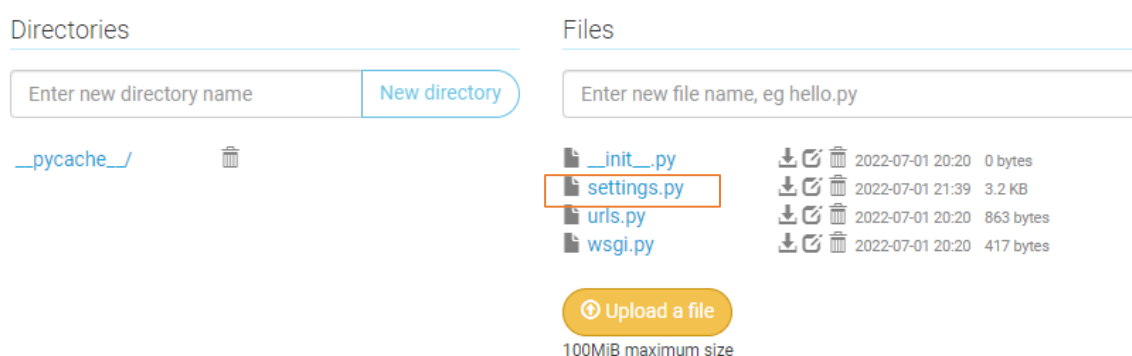
```
17:49 ~ $ pip3 show virtualenv
Name: virtualenv
Version: 20.4.7
Summary: Virtual Python Environment builder
Home-page: https://virtualenv.pypa.io/
Author: Bernat Gabor
Author-email: gaborjbernat@gmail.com
License: MIT
Location: /usr/local/lib/python3.9/site-packages
Requires: appdirs, distlib, filelock, six
Required-by: virtualenvwrapper
17:49 ~ $ ls
README.txt  experiencein
00:50 ~ $ cd experiencein
00:51 ~/experiencein (main)$ pwd
/home/KalitaSouza1/experiencein
00:51 ~/experiencein (main)$
```



Salvei as alterações e cliquei no botão verde “*Reload*” para carregar a aplicação web. Aguardei até o botão voltar ao normal.



Depois de salvo e carregado. Alterei outras configurações. Acessei o *setting.py* do projeto Django.



## 5. Edição do *settings.py* / configuração dos arquivos estáticos

Na linha 28 adicionei o domínio da aplicação web (kalitasouza1.pythonanywhere.com) e ficou da seguinte forma:

```
27
28 ALLOWED_HOSTS = ['kalitasouza1.pythonanywhere.com']
29
```

Logo após fiz outra configuração ainda no mesmo arquivo. Criei uma constante de configuração chamada `STATIC_ROOT`, e nela coloquei o caminho completo para a pasta `static`.


```
123
124 STATIC_ROOT = '/home/KalitaSouza1/experiencein/static'
125 LOGIN_URL="/login/"
126 LOGOUT_URL="/logout/"
127 LOGIN_REDIRECT_URL="/"
128 LOGOUT_REDIRECT_URL=LOGIN_URL
```

Depois desse passo salvei o arquivo, acessei o console `bash` e naveguei até a pasta do projeto Django onde se encontra o arquivo `manage.py`. Na sequência eu executei o comando `python manage.py collectstatic` para coletar todos os arquivos estáticos vinculados ao projeto e organizá-los na pasta `static` que ele iria criar.

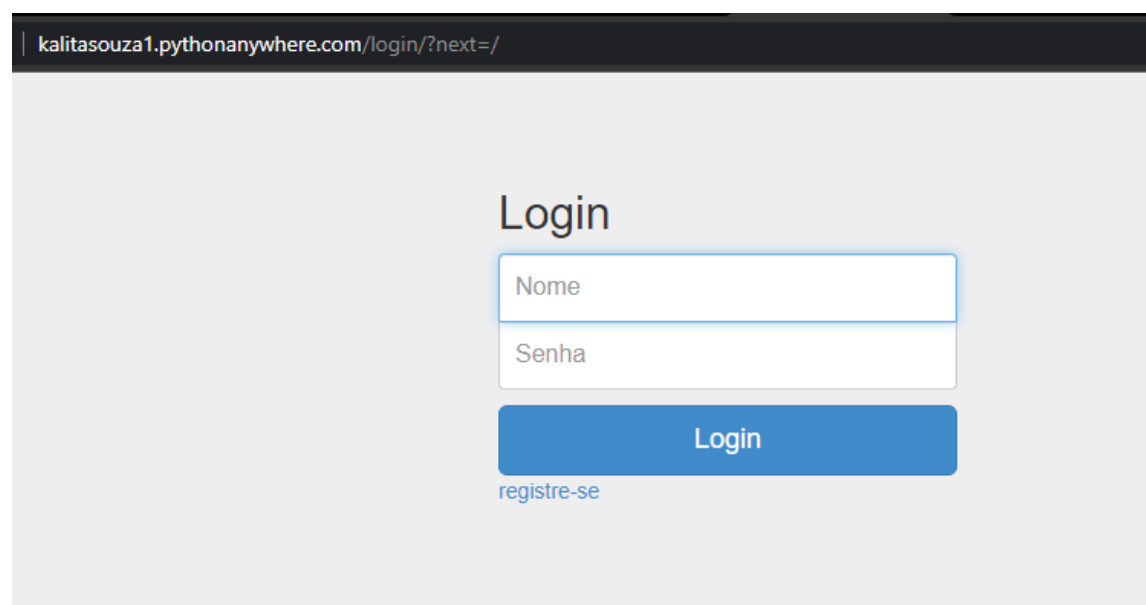
A próxima configuração eu fiz no `dashboard` da aplicação web. Naveguei até o menu Web, seção `Static files` e inseri no campo `URL` e `Directory` as mesmas configurações que fiz no arquivo `setting.py` e ficou dessa forma:

#### Static files:

Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, can be served much faster straight off the disk if you specify them here. You need to **Reload your web app** to activate any changes you make to the mappings below.

URL	Directory	Delete
<a href="#">/static/</a>	<a href="#">/home/KalitaSouza1/experiencein/static</a>	
<a href="#">Enter URL</a>	<a href="#">Enter path</a>	

E após essas configurações a aplicação já estava funcionando corretamente como na imagem abaixo:



The screenshot shows a web browser window with the address bar displaying `kalitasouza1.pythonanywhere.com/login/?next=/`. The main content area has a light gray background. In the center, the word "Login" is displayed in a large, dark font. Below it, there are two input fields: the first is labeled "Nome" and the second is labeled "Senha". Below these fields is a blue button with the text "Login" in white. At the bottom, there is a blue link that says "registre-se".

kalitasouza1.pythonanywhere.com/registrar/

### Crie o seu usuário

Email \*

Nome \*

Senha \*

Telefone

Empresa

Registrar

## 5. Links

GitHub do projeto: <https://github.com/KALITARDS/experiencein>

Aplicação web: <http://kalitasouza1.pythonanywhere.com/>