



Documento de implantação-
Projeto de Programação para Internet II

Versão 0.1

Brasília- DF

21/06/2022

Projeto simulação de uma rede social			
Histórico de Revisão do Documento			
Versão	Data	Responsável	Descrição
0.1	21/06/2022	Camila Carvalho	Etapa de deploy da aplicação

SUMÁRIO

1.Introdução.....	3
2.Escopo.....	3
3.Não escopo.....	3
4.Recursos.....	3
4.1- Hardware.....	3
4.2- Tecnologias utilizadas.....	3
4.3 Realização do deploy da aplicação.....	4
a) O projeto deve estar no repositório do github com as pastas do projeto, como a figura 1:.....	4
b) Criar uma conta na plataforma pythonanywhere, como ilustrado na figura 2 abaixo:.....	4
c) Clonar o projeto na plataforma.....	5
d) Criando o ambiente virtual para aplicarmos as configurações específicas do projeto experiencein.....	6
e) Instalando o django.....	7
f) Configurando a aplicação web com WSGI.....	7

1. Introdução

O documento tem por finalidade apresentar detalhadamente como foi realizado o deploy da aplicação.

2. Escopo

O projeto visa descrever o passo a passo de como realizar a implementação do projeto experiencein que tem como foco simular uma rede social, com o cadastro de perfis e convite entre os usuários.

3. Não escopo

O projeto não terá como foco desenvolver uma aplicação de rede social completa com a possibilidade de enviar mídias sociais.

4. Recursos

Para o desenvolvimento da aplicação foi necessário utilizar alguns recursos de hardware e software, que serão listados nos próximos tópicos.

4.1- Hardware

Para desenvolver o projeto foram utilizados os seguintes recursos de hardware:

Sistema Operacional	Memória	Processador	Tipo de sistema	Disco Rígido
Linux	8GB	AMD Fx 8300	64bits	HD
Windows	8GB	Intel 11 ^a Geração Core i5	64bits	SSD

4.2- Tecnologias utilizadas

Python versão 3.7

Django versão 2.2

Pip

HTML5

CSS3

Bootstrap 3.3

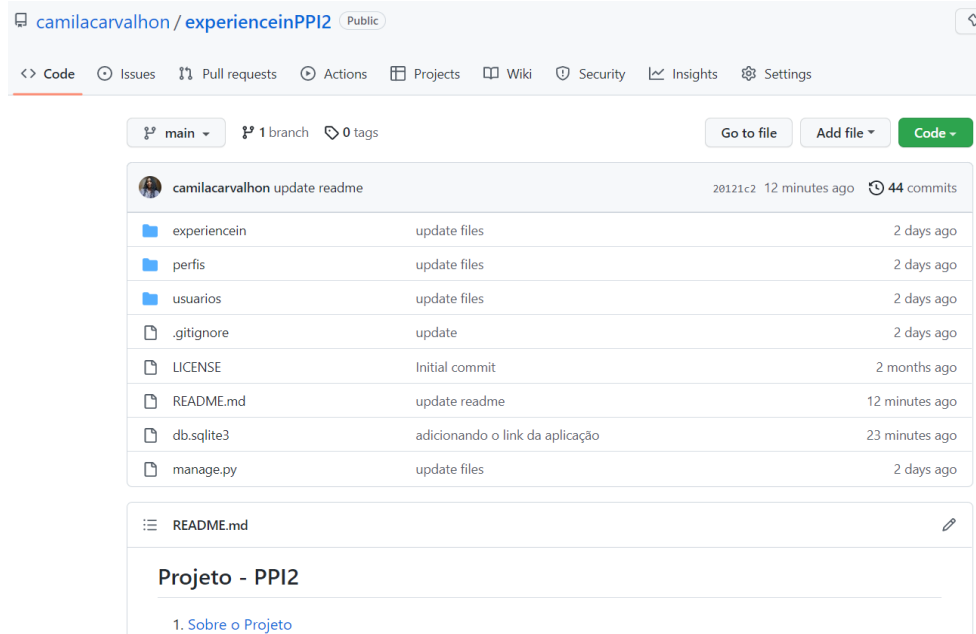
Virtualenv

4.3 Realização do deploy da aplicação

Para realizar esta etapa foi necessário seguir os passos enumerados abaixo:

- a) O projeto deve estar no repositório do github com as pastas do projeto, como a figura 1:

Figura 1 - Repositório



[Link do repositório](#)

- b) Criar uma conta na plataforma pythonanywhere, como ilustrado na figura 2 abaixo:

Figura 2 - Criando uma conta na plataforma

Send feedback Forums Help Blog Pricing & sign up Log in

pythonanywhere

Create your account

Username:

Email:

Password:

Password (again):

☐ I agree to the [Terms and Conditions](#) and the [Privacy and Cookies Policy](#), and confirm that I am at least 13 years old.

[Register](#)

We promise not to spam or pass your details on to anyone else.

Copyright © 2011-2022 PythonAnywhere LLP -- [Terms](#) -- [Privacy & Cookies](#)

[Link da plataforma](#)

Após criar a conta, confirmar o e-mail cadastrado por meio da sua conta email e efetuar o login.

c) Clonar o projeto na plataforma

1º Acessar o bash como ilustrado na figura 3 e 4:

Figura 3 - Localizando o bash

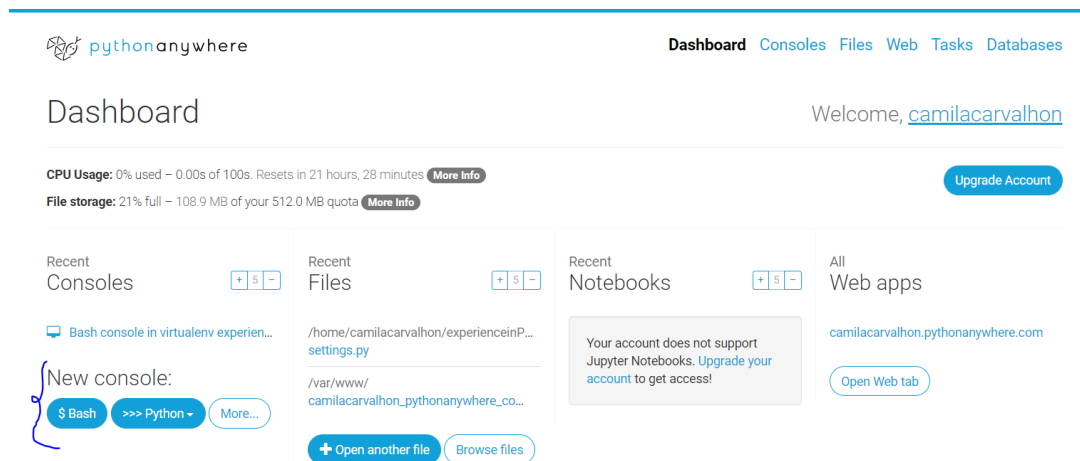
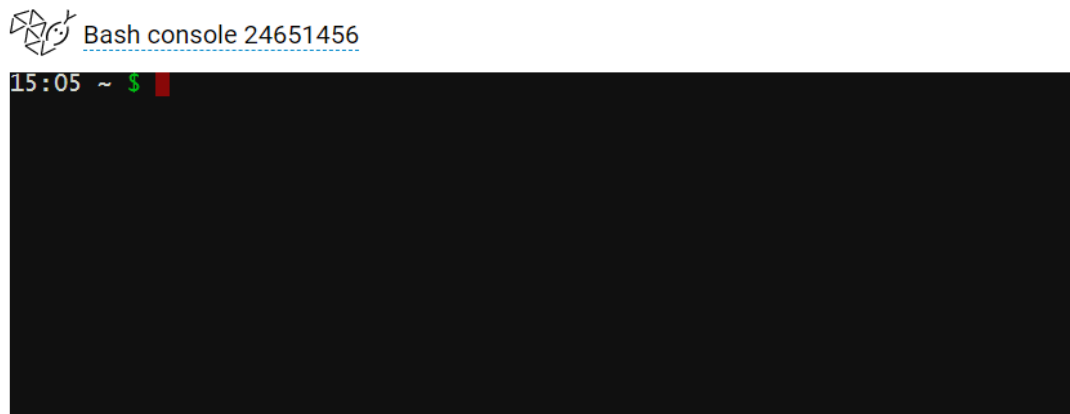
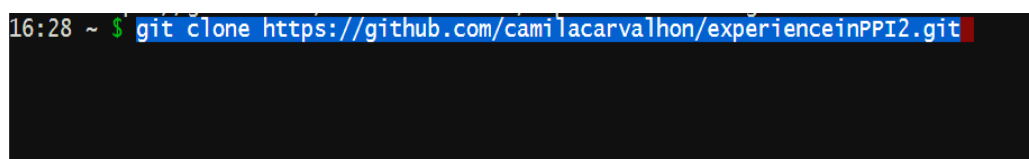


Figura 4 - Bash aberto



2º Digitar o seguinte comando para clonar o projeto <git clone + link do seu repositório>, veja o exemplo abaixo figura 5 e repita o processo:

Figura 5 - Digitando no bash



Comando digitado: `git clone https://github.com/camilacarvalhon/experienceinPPI2.git`

d) Criando o ambiente virtual para aplicarmos as configurações específicas do projeto experiencein.

1º Verificamos se o pacote Python virtualenv já está instalado, através do seguinte comando:

`pip3 show virtualenv`

Figura 6 - Verificando se tem o pacote virtualenv

```
16:28 ~ $ 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: six, appdirs, filelock, distlib
Required-by: virtualenvwrapper
16:52 ~ $
```

Observação: Caso não esteja instalado, poderá instalar a partir do comando:


`pip3 install virtualenv`

2º Criar o ambiente virtual digitando a instrução abaixo:

1º Etapa: Você deve estar fora da pasta do projeto, ou seja na pasta de usuário

‘/home/seu nome de usuário/’, no meu caso é ‘/home/camilacarvalhon/’. Primeiro liste através do comando “ls” para ficar a pasta e o readme do projeto, e depois verifique o diretório local que você está, como apresentado na figura 7:

Figura 7 - Acessando o console para verificar o diretório local

```
 Bash console 24697068
18:19 ~ $ ls
README.txt  experienceinPPI2
18:20 ~ $ pwd
/home/camilacarvalhon
18:20 ~ $
```

2º Etapa: Você deve digitar a seguinte instrução para criar o ambiente virtual:

`mkvirtualenv --python=/usr/bin/python3.7 nomedoambientevirtual.`

Exemplo: `mkvirtualenv --python=/usr/bin/python3.7 experiencein-virtualenv`

Veja o exemplo da figura 8 abaixo:

Figura 8 - Criando o ambiente virtual

```
18:19 ~ $ ls
18:20 ~ $ mkvirtualenv --python=/usr/bin/python3.7 experiencein-virtualenv
created virtual environment CPython3.7.10.final.0-64 in 9548ms
creator CPython3Posix(dest=/home/camilacarvalhon/.virtualenvs/experiencein-virtualenv, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/camilacarvalhon/.local/share/virtualenv)
added seed packages: Django==2.2, pip==21.1.2, pytz==2022.1, setuptools==57.0.0, sqlparse==0.4.2, wheel==0.36.2, wheel==0.37.1
activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
(experiencein-virtualenv) 18:27 ~ $
```

Observação: Sobre a versão do python vocês podem colocar a que estiver utilizando no projeto, no nosso caso é a versão 3.7.

e) Instalando o django

1º Digite o seguinte comando `pip install django==2.2`, como ilustrado na figura 9:

Figura 9 - Instalação do django

```
(experiencein-virtualenv) 18:27 ~ $ pip install django==2.2
Looking in links: /usr/share/pip-wheels
Requirement already satisfied: django==2.2 in ./virtualenvs/experiencein-virtualenv/lib/python3.7/site-packages (2.2)
Requirement already satisfied: sqlparse in ./virtualenvs/experiencein-virtualenv/lib/python3.7/site-packages (from django==2.2) (0.4.2)
Requirement already satisfied: pytz in ./virtualenvs/experiencein-virtualenv/lib/python3.7/site-packages (from django==2.2) (2022.1)
(experiencein-virtualenv) 18:29 ~ $
```

2º Podemos verificar se a instalação foi realizada corretamente digite `python -m django version`, como exemplo a figura 10:

Figura 10 - Verificação da versão instalada do django

```
(experiencein-virtualenv) 18:29 ~ $ python -m django version
2.2
(experiencein-virtualenv) 18:30 ~ $
```

f) Configurando a aplicação web com WSGI

1º - Criamos uma aplicação web, acessando o menu web, posteriormente clicamos em “Add a new web app”, como apresentado na figura 11:

Figura 11 - Criando uma nova aplicação web

 pythonanywhere

[Dashboard](#) [Consoles](#) [Files](#) **[Web](#)** [Tasks](#) [Databases](#)

[camilacarvalhon.pythonanywhere.c...](#)

[Add a new web app](#)

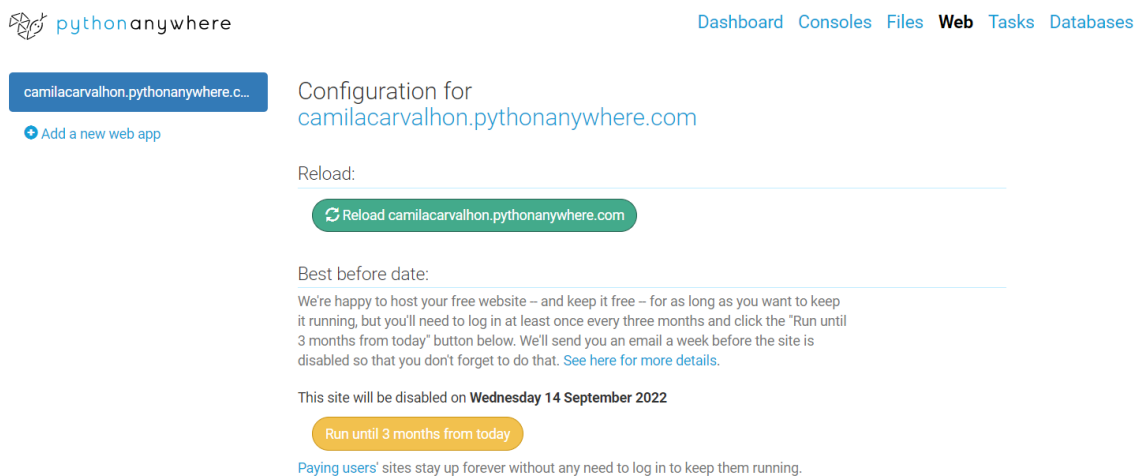
Configuration for
[camilacarvalhon.pythonanywhere.com](#)

2º Após isso, aparecerá uma janela para selecionar algumas configurações.

- 1-Selecionar nome do domínio padrão
- 2- Selecionar a opção de “manual de configuração”
- 3- Escolha a versão 3.7 do Python (a que estamos usando no projeto)
- 4- Click em próximo quando aparecer configuração manual

Pronto! Deverá aparecer isso como apresentado na figura 12, agora, vamos vincular o ambiente virtual com a aplicação web criada.

Figura 12 - Tela após a criação da aplicação web



3º Ao rolar o scroll, você deve clicar em “Enter path to a virtualenv, if desired”, para adicionar o caminho do ambiente virtual que criamos anteriormente, como ilustrado abaixo nas figuras 13 e 14.

Observação: no projeto em questão será

`/home/camilacarvalhon/.virtualenvs/experiencein-virtualenv`, nisso você substitui pelo seu nome de usuário e pelo nome do ambiente virtual que você nomeou ao gerar o ambiente virtual que deverá ficar assim `/home/nomedeusuario/.virtualenvs/nomedoambientevirtual`.

Figura 13 - Inserir o path do virtualenv

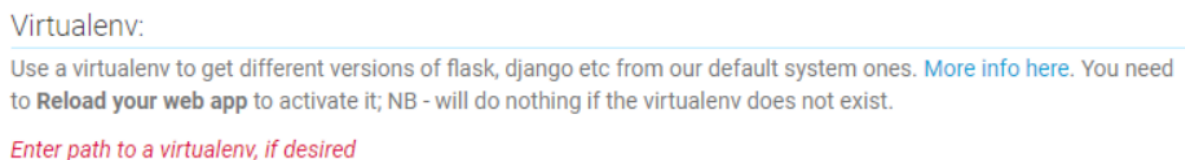


Figura 14 - Após inserir o caminho do ambiente virtual

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.

</home/camilacarvalhon/.virtualenvs/experiencein-virtualenv>

[Start a console in this virtualenv](#)

4º Nessa etapa, acessaremos o arquivo de configuração WSGI. Para isso, é necessário clicar no link como ilustrado na figura 15:

Figura 15 - Acessando o arquivo de configuração WSGI

Code:

What your site is running.

Source code:	Enter the path to your web app source code	
Working directory:	/home/camilacarvalhon/	Go to directory
WSGI configuration file:	/var/www/camilacarvalhon_pythonanywhere_com_wsgi.py	
Python version:	3.7	

Após isso, o arquivo será aberto e deixaremos apenas as configurações referentes ao django, as outras podem ser deletadas, antes de salvar as modificações, vamos alterar duas linhas: a primeira referente ao path que deve ser adicionado o caminho completo para o projeto django, e a segunda deverá conter o nome_do_projeto.settings, lembrando que é o mesmo nome que colocamos ao criar o projeto.

Na figura 16 é possível verificar como o arquivo deve ficar:

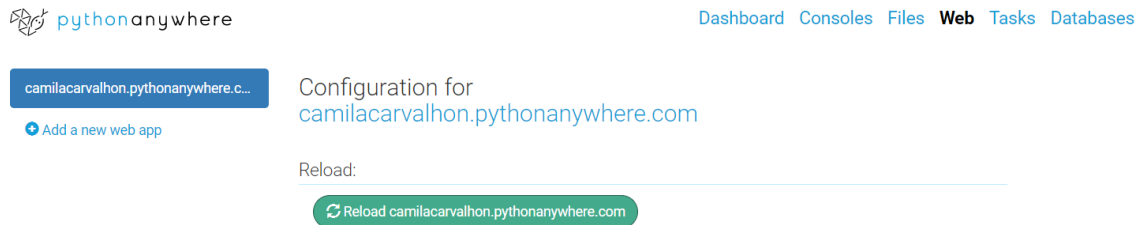
Figura 16 - Organização do arquivo de configuração WSGI

```
3 # It works by setting the variable "application" to a WSGI handler of some
4 # description.
5 #
6 #
7 # ++++++ GENERAL DEBUGGING TIPS ++++++
8 # getting imports and sys.path right can be fiddly!
9 # We've tried to collect some general tips here:
10 # https://help.pythonanywhere.com/pages/DebuggingImportError
11 #
12 #
13 # Below are templates for Django and Flask. You should update the file
14 # appropriately for the web framework you're using, and then
15 # click the 'Reload /yourdomain.com/' button on the 'Web' tab to make your site
16 # live.
17 #
18 #
19 # ++++++ DJANGO ++++++
20 # To use your own django app use code like this:
21 import os
22 import sys
23 #
24 ## assuming your django settings file is at '/home/camilacarvalhon/mysite/mysite/settings.py'
25 ## and your manage.py is at '/home/camilacarvalhon/mysite/manage.py'
26 path = '/home/camilacarvalhon/experienceinPPI2'
27 if path not in sys.path:
28     sys.path.append(path)
29 #
30 os.environ['DJANGO_SETTINGS_MODULE'] = 'experiencein.settings'
31 #
32 ## then:
33 from django.core.wsgi import get_wsgi_application
34 application = get_wsgi_application()
35
36
```

Agora, podemos salvar o arquivo clicando em “SAVE”.

5º Para aplicar as modificações devemos recarregar a aplicação web, clicando no botão reload, como ilustrado na imagem 17 abaixo:

Figura 17 - Recarregar a aplicação web



6º Vamos navegar até o arquivo de configuração settings. Para acessar podemos clicar no menu Files e navegar até o arquivo settings.py do projeto que se encontra dentro da pasta do projeto django, como ilustram as figuras 18, 19 e 20:

Figura 18 - Passo 1 acessar arquivo settings.py

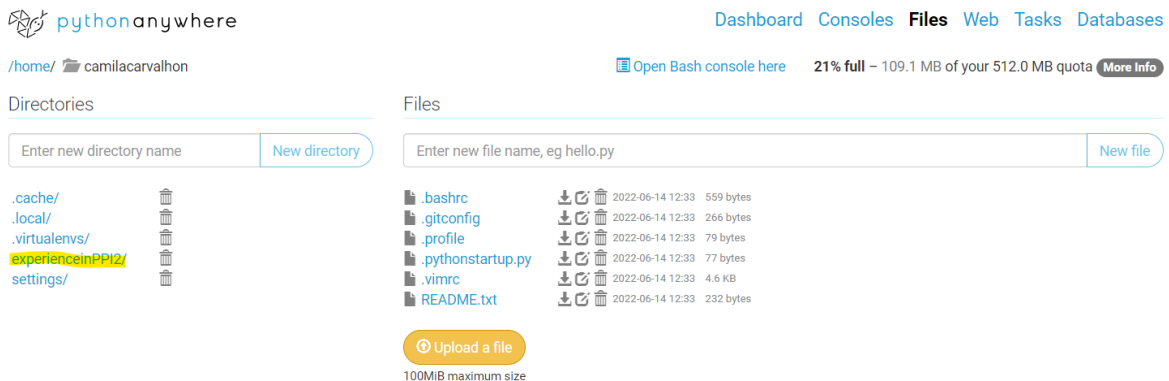


Figura 19 - Passo 2 acessar arquivo settings.py

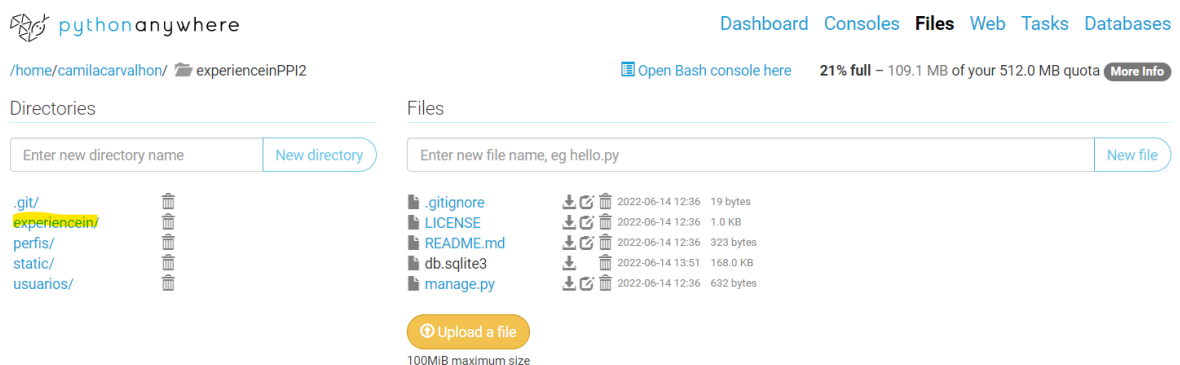
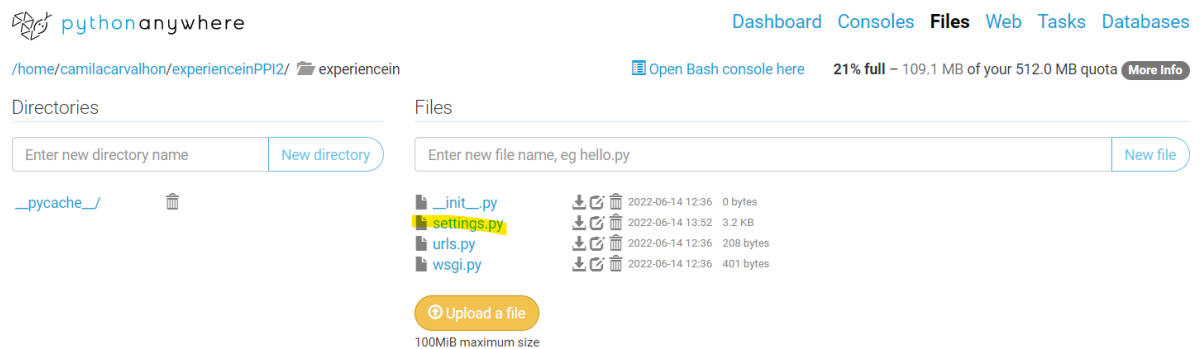
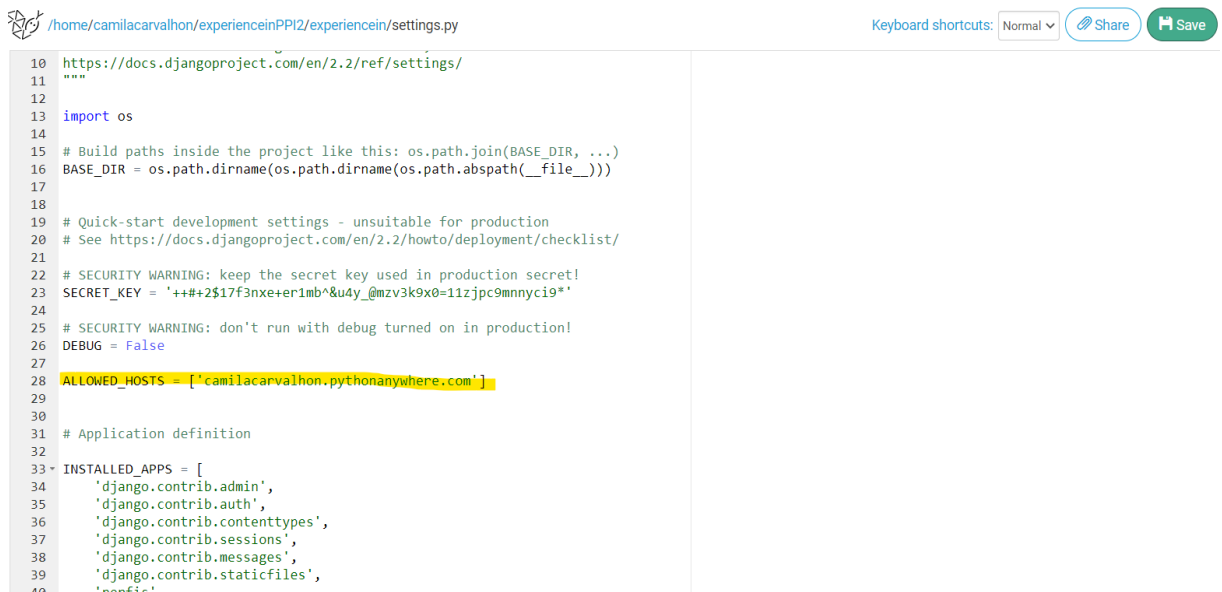


Figura 20 - Passo 3 acessar arquivo settings.py



Ao abrir o arquivo podemos adicionar o domínio da aplicação no meu caso será “**camilacarvalhon.pythonanywhere.com**” e clicar em “SAVE”, como ilustra a figura 21 abaixo:

Figura 21 - Adicionando domínio da aplicação



Pronto! Podemos acessar a url da nossa aplicação e ver o resultado, se você fez tudo certo apresentará uma página simples, sem estilos na página.

Observação: Não feche ainda o arquivo de configuração

7 - Coletando e configurando os arquivos estáticos

1º No arquivo de configuração settings.py vamos criar uma constante de configuração chamada “STATIC_ROOT”, ela deverá apresentar o path completo para acessar o arquivo static que será criado dentro do diretório do projeto, como apresentado na figura 22:

Figura 22- Adicionando a configuração da url do arquivo Static

```
104
105 # Internationalization
106 # https://docs.djangoproject.com/en/2.2/topics/i18n/
107
108 LANGUAGE_CODE = 'en-us'
109
110 TIME_ZONE = 'UTC'
111
112 USE_I18N = True
113
114 USE_L10N = True
115
116 USE_TZ = True
117
118
119 # Static files (CSS, JavaScript, Images)
120 # https://docs.djangoproject.com/en/2.2/howto/static-files/
121
122 STATIC_URL = '/static/'
123
124 STATIC_ROOT = '/home/camilacarvalhon/experienceinPPI2/static'
125
126 LOGIN_URL = '/login/'
127 LOGOUT_URL = '/logout/'
128 LOGIN_REDIRECT_URL = '/'
129 LOGOUT_REDIRECT_URL = LOGIN_URL
```

2º Após adicionar, vamos salvar a modificação e abrir o bash e navegar até a pasta do projeto, onde está o arquivo manage.py. Em seguida, vamos digitar o comando python manage.py collectstatic que irá coletar todos os arquivos estáticos vinculados ao projeto e organizá-los na pasta static que será criada, como mostra as figuras 22:

Figura 23 - Criando e adicionando o arquivo de configuração static


```
(experiencein-virtualenv) 13:29 ~/experienceinPPI2 (main)$ pwd
/home/camilacarvalhon/experienceinPPI2
(experiencein-virtualenv) 13:49 ~/experienceinPPI2 (main)$ ls
LICENSE README.md db.sqlite3 experiencein manage.py perfis usuarios
(experiencein-virtualenv) 13:49 ~/experienceinPPI2 (main)$ python manage.py collectstatic
132 static files copied to '/home/camilacarvalhon/experienceinPPI2/static'.
(experiencein-virtualenv) 13:49 ~/experienceinPPI2 (main)$ ls
LICENSE README.md db.sqlite3 experiencein manage.py perfis static usuarios
(experiencein-virtualenv) 13:49 ~/experienceinPPI2 (main)$
```

3º A próxima etapa será inserirmos a url e o diretório. Para isso, vamos navegar até o menu web e ir em Static files, nele adicionamos o caminho da pasta static, como mostra a figura 24 abaixo:

Figura 24 - Inserindo o caminho do arquivo static

Static files:

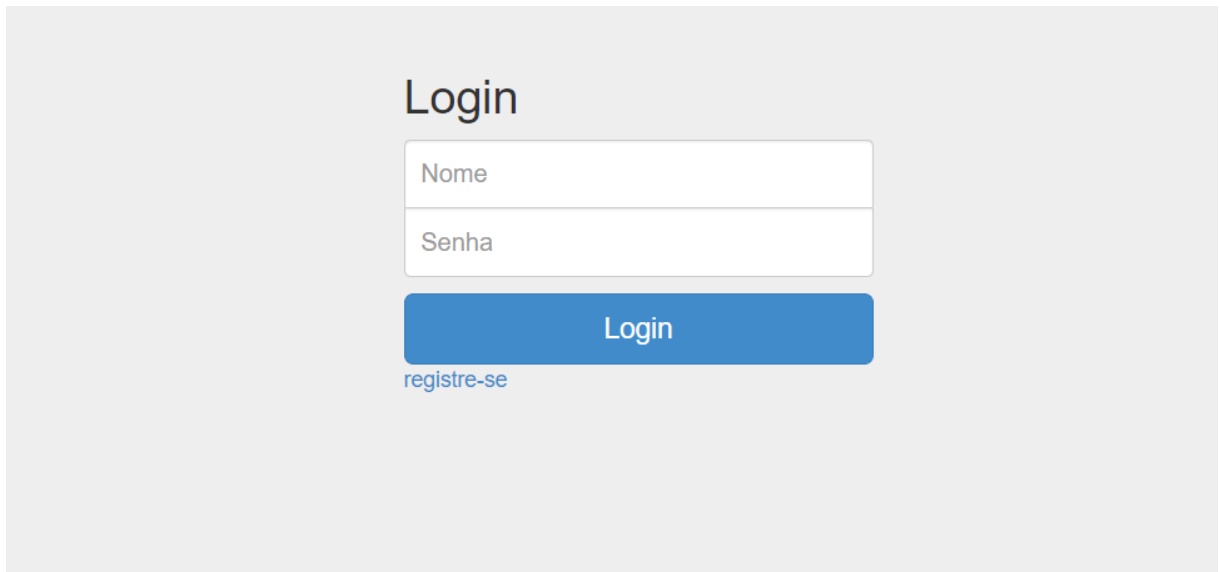
Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, can be served much faster straight off the disk if you specify them here. You need to **Reload your web app** to activate any changes you make to the mappings below.

URL	Directory	Delete
/static/	/home/camilacarvalhon/experienceinPPI2/static	
Enter URL	Enter path	

4º Antes de finalizar, vamos acessar as configurações novamente e alterar o valor da variável debug para false, uma vez que não estamos mais em nível de desenvolvimento, e sim de produção.

Após isso clicamos em reload e acessamos a url da aplicação. Se seguiu os mesmos passos a página web acessada deverá apresentar a seguinte tela como na imagem 25:

Figura 25 - Tela de Login

A imagem mostra uma interface de login com o título "Login" em uma fonte grande e escura. Abaixo do título, há dois campos de entrada de texto empilhados verticalmente. O primeiro campo é rotulado "Nome" e o segundo "Senha". Ambos os campos são retangulares com bordas arredondadas e uma borda cinza. Abaixo dos campos, há um botão de login com o texto "Login" em branco sobre um fundo azul. Logo abaixo do botão, há um link "registre-se" em uma fonte menor e cor azul.