# Despliegue de nuestra aplicación

En la pestaña Consoles de PythonAnywhere podemos abrir consolas con los distintos interpretes python, con bash o con mysql, vamos a abrir una consola bash para comenzar el despliegue:



Consoles Files Web Schedule Databases

## Start a new console:

Python: 3.6/3.5/3.4/3.3/2.7 IPython: 3.6/3.5/3.4/3.3/2.7 PyPy: 2.7

Other: Bash | MySQL

Custom: 0

### Your consoles:

Bash console 5962090 X



#### Bash console 5962090



Vamos a clonar nuestro repositorio, crear un repositorio e instalar las dependencias:

\$ git clone <a href="https://github.com/gibinaghi/cientificas-de-datos.git">https://github.com/gibinaghi/cientificas-de-datos.git</a> -b pythonanywhere \$ mkvirtualenv --python=/usr/bin/python3.9 flask

```
E Bash console 28941283

19:41 ~ $ git clone https://github.com/gibinaghi/cientificas_de_datos.git
Cloning into 'cientificas_de_datos'...
remote: Enumerating objects: 159, done.
remote: Counting objects: 100% (159/159), done.
remote: Counting objects: 100% (19/159), done.
remote: Counting objects: 100% (19/159), done.
remote: Counting objects: 100% (19/159), done.
remote: Cotal 159 (delta 53), reused 119 (delta 22), pack-reused 0
Receiving objects: 100% (59/159), 3.26 MiB | 34.40 MiB/s, done.
Resolving deltas: 100% (53/53), done.
19:42 ~ $ mkwirtualenv - python=/usr/bin/python3.4 flask
fileNotFoundError: [Errno 2] No such file or directory: '/usr/bin/python3.4'
19:43 ~ $ mkwirtualenv - python=/usr/bin/python3 p flask
created virtual environment CPython3, 9.13. final 0.64 in 29040ms
created virtual environment CPython3, 9.13. final 0.64 in 29040ms
created virtual environment CPython4 pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/florcorvalan26/.virtualenvy
added seed packages: pip==22.1.2, setuptools==62.6.0, wheel==0.37.1
activators BashActivator, CShellActivator, FishActivator, NushellActivator, PowerShellActivator, PythonActivator
virtualenvwrapper.user_scripts creating /home/florcorvalan26/.virtualenvs/flask/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/florcorvalan26/.virtualenvs/flask/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/florcorvalan26/.virtualenvs/flask/bin/postactivate
virtualenvwrapper.user_scripts creating /home/florcorvalan26/.virtualenvs/flask/bin/postactivate
virtualenvwrapper.user_scripts creating /home/florcorvalan26/.virtualenvs/flask/bin/peteactivate
virtualenvwrapper.user_scripts creating /home/florcorvalan26/.virtualenvs/flask/bin/peteactivate
virtualenvwrapper.user_scripts creating /home/florcorvalan26/.virtualenvs/flask/bin/peteactivate
virtualenvwrapper.user_scripts creating /home/florcorvalan26/.virtualenvs/flask/bin/peteactivate
```

#### \$ workon flask

(flask)\$ pip install -r cientificas de datos/practicas profesionalizantes/requirements.txt

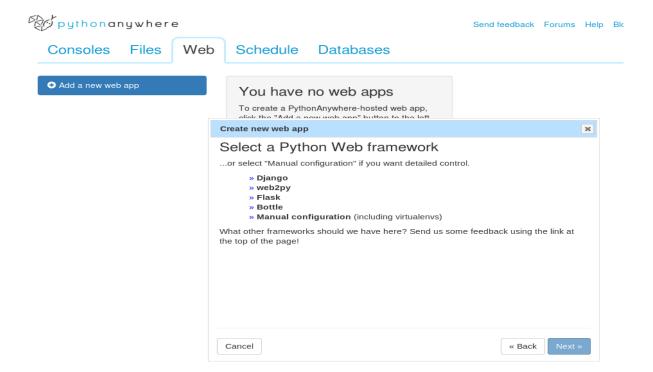
```
## Share with others

| Collecting bleach=6.00 | Downloading certifi-2023.5.7-py3-none-any.whl (156 kB) | Downloading defusedxml=e7.1 | Downloading defusedxml=e7.2 | Downloading defusedxml=e7.1 | Downloading defusedxml=e7.1 | Downloading defusedxml=e7.1 | Downloading defusedxml=e7.2 | Downloading defusedxml=e7.2 | Downloading defusedxml=e7.2 | Downloading defusedxml=e7.2 | Downloading defusedxml=e7.1 | Downloading defusedxml=e7.1 | Downloading defusedxml=e7.1 | Downloading defusedxml=e7.1 | Downloading defusedxml=e7.2 | Downloading defusedxml=e7.3 | Downloading defusedxml=e7.2 | Downloading defusedxml=e7.2 | Downloading defusedxml=e7.3 | Downloading defusedxml=e7.4 | Downloading defusedxml=e7.4 | Downloading defusedxml=e7.6 | Downloading defusex
```

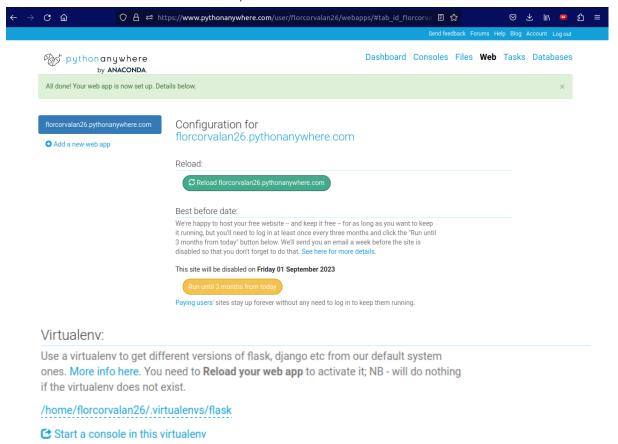
# Creando una nueva aplicación

Por último en la pestaña Web tenemos que crear una nueva aplicación:

Elegimos la opción "Manual configuration" para poder indicar el entorno virtual que hemos creado:



Elegimos la versión de python (en este caso la misma con la que hemos creado el entorno virtual, python 3.9) y ya tenemos la aplicación creada. Sólo nos queda indicar la ruta donde se encuentra nuestro entorno virtual, en la sección **Virtualenv**:



Y modificar el fichero `<u>/var/www/florcorvalan26\_pythonanywhere\_com\_wsgi.py</u> `en el apartado "WSGI configuration file:" de la sección "Code":

```
Code:
      What your site is running.
                               Source code:
                                                      Enter the path to your web app source code
                        Working directory:
                                                      /home/florcorvalan26/
                                                                                                                                   →Go to directory
                 WSGI configuration file:
                                                      /var/www
                                                      /florcorvalan26_pythonanywhere_com_wsgi.py
                            Python version:
                                                      3.9 🧪
      Virtualenv:
     Hea a virtualany to not different versions of flack diagno eto from our default exetem
 91
 92
      # +++++++++ FLASK +++++++++
 94 # Flask works like any other WSGI-compatible framework, we just need
95 # to import the application. Often Flask apps are called "app" so we
96 # may need to rename it during the import:
99 import sys
100 #
## The "/home/florcorvalan26" below specifies your home

## directory -- the rest should be the directory you uploaded your Flask

## code to underneath the home directory. So if you just ran

## "git clone git@github.com/myusername/myproject.git"
105 ## ...or uploaded files to the directory "myproject", then you should
106 ## specify "/home/florcorvalan26/myproject"
107 path = '/home/florcorvalan26/cientificas_de_datos/practicas_profesionalizantes/spam_detection'
108 - if path not in sys.path:
109
            sys.path.append(path)
```

Si surgen problemas de librerías ejecutar los siguientes comandos:

pip install flask

pip install pandas

pip install scikit-learn