

Manual de Instalación de DjangoLab V.2.3.1

DjangoLab es un sistema de administración y difusión de investigación, orientada a centros de investigación, laboratorios y grupos de investigación

Ejemplos (examples) de los Web Sites con DjangoLab:

- [LascaWeb] (<https://www.lasca.ic.unicamp.br/>) - Desarrollado con DjangoLab 1.0
- [IIMAT-FCPN] (<http://iimat.fcpn.edu.bo/>) - Desarrollado con DjangoLab 2.2
- [III-FCPN] (<http://iii.fcpn.edu.bo/>) - Desarrollado con DjangoLab 2.2
- [IIF-FCPN] (<http://iif.fcpn.edu.bo/>) - Desarrollado con DjangoLab 2.3.1
- [IBMB-FCPN] (<http://ibmb.fcpn.edu.bo/>) - Desarrollado con DjangoLab 2.3.1
- [IE-FCPN] (<http://ie.fcpn.edu.bo/>) - Desarrollado con DjangoLab 2.3.1
- [IETA-FCPN] (<http://ieta.fcpn.edu.bo/>) - Desarrollado con DjangoLab 2.3.1
- [IIQ-FCPN] (<http://iiq.fcpn.edu.bo/>) - Desarrollado con DjangoLab 2.3.1

Pre-requisitos

DjangoLab fue desarrollado sobre python 3.6 y Ubuntu 16.04. Los requerimientos mínimos son:

- [Ubuntu 16.04 LTS o +/- Debian 8.0 o +] (<https://ubuntu.com/about/release-cycle>) - Sistemas Operativos usados en el proyecto, aunque soporta Windows.
- [Django 2.2 LTS] (<https://www.djangoproject.com/download/>) - Framework de desarrollo.
- [Python 3.6] (<https://python3statement.org/>) - Lenguaje de programación Python 3.6.8 (soporta desde Python 3.5 o +).

Otros requerimientos técnicos mínimos:

- Procesador: Intel Atom o Intel Core i3
- Disco de 10GB
- Memoria de 2GB (Para Ubuntu o Debian)

Más información en: <https://software.intel.com/en-us/distribution-for-python/system-requirements>

Instalación

##0. Comandos de ayuda:

0.1. Versión de linux instalada.

```
$ lsb_release -a  
OR  
$ cat /etc/os-release
```

0.2. Verificar tamaño de disco:

```
$ df -h
```

0.3. Actualizar el Sistema Operativo:

Para ubuntu:

```
# apt-get update
# apt-get upgrade
```

Para Debian:

```
# sudo apt-get update
# sudo apt-get -y upgrade
```

0.4. Dependencias básicas: Para Debian usar: <https://pythonchile.cl/post/2018/02/07/python3-en-debian-8-y-9/>

```
$ sudo apt install python3.6 // Python 3.5 o +
$ python3 --version
    > Python 3.6.8
$ sudo apt install python3-pip
$ pip3 --version
    > pip 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)
```

Para Debian:

```
$ sudo pip3 install --upgrade setuptools
```

```
$ pip3 install Pillow
```

Pillow es una librería para abrir, manipular y guardar muchos formatos de archivo de imagen diferentes.

1. Configuración de virtualenv:

1.1. Instalar dependencias necesarias:

```
$ sudo apt-get install apache2 libapache2-mod-wsgi-py3
```

Para Debian:

```
$ sudo apt-get install apache2
$ sudo apt-get install libapache2-mod-wsgi-py3
```

1.2 Configuración de Python Virtual Environment:

```
$ pip3 install virtualenv
    > Successfully installed virtualenv-16.6.2
```

Para Debian instalar:

```
$ sudo apt install python3-venv
```

\$ Verificar la posición de instalación:

```
$ pwd
    > home/user
```

Crear la carpeta de trabajo

```
$ mkdir.djangolab
$ cd.djangolab
```

Dentro de la carpeta django, crear el ambiente virtual Python:

```
$ virtualenv.djangolabenv
    > done.
```

Para Debian:

```
$ python3 -m venv.djangolabenv
```

#IMPORTANTE# Caso de ERRORES con venv usar este manual: <https://vitux.com/install-python3-on-ubuntu-and-set-up-a-virtual-programming-environment/>

```
$ sudo apt-get install build-essential libssl-dev libffi-dev python-dev
$ pip3 -V
$ sudo apt-get update
```

```
$ sudo apt install python3-pip
$ pip3 install virtualenv
$ sudo apt install -y python3-venv
$ python3 -m venv.djangolabenv
```

```
$ Activar el entorno virtual desde la misma posición:
$ source ~/djangolab/djangolabenv/bin/activate
> (djangolabenv) user@server:~/djangolab$
```

2. Instalar Django y configuración en el Entorno Virtual (no salir):

Link de versiones: <https://www.djangoproject.com/download/>

Instalar los siguientes paquetes:

```
$ sudo apt install python3.6
```

Instalar una version LTS de Django 2.2, se recomienda revisar la última version de Django 2.2 en

<https://www.djangoproject.com/download/>

```
$ pip3 install django==2.2.8
```

Verificar la version de Django instalada

```
$ python3 -m django --version
> 2.2.8 o superior
```

OR

```
$ python3
```

```
>>> import django
```

```
>>> django.VERSION
```

```
(2, 2, 8, 'final', 0)
```

2.1. Instalar otros programas necesarios

```
$ sudo apt install python-django
```

```
$ sudo apt install python-django-common
```

2.2. Revisa todas las versiones de paquetes python instalados

```
$ pip3 freeze
```

```
Django==2.2.8
```

```
pytz==2019.3
```

```
sqlparse==0.3.0
```

Validar la versión de django-admin instalado

```
$ django-admin --version
```

```
> 2.2.8
```

2.3. Verificar los paquetes instalados por django-admin

```
$ django-admin
```

```
[django]
```

```
check
```

```
compilemessages
```

```
createcachetable
```

```
...
```

3. Instalar PostgreSQL y configuración dentro del entorno virtual (no salir):

Links: <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-ubuntu-18-04> <https://www.fullstackpython.com/blog/postgresql-python-3-psycopg2-ubuntu-1604.html>

Adicionar estos pasos para Debian:

```
$ wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -O - | sudo apt-key add -
```

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ stretch-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'
```

3.1. Instalar PostgreSQL y sus dependencias

```
$ sudo apt-get install postgresql postgresql-contrib
```

3.2. Instalar dependencias de PostgreSQL para Django psycopg2

```
$ sudo apt-get install postgresql-client postgresql-client-common
$ sudo apt install libpq-dev python-dev
$ pip3 install psycopg2
```

3.3. Acceder al Prompt de Postgres y cambiar la contraseña:

```
$ sudo -u postgres psql
# \l   retornará la lista de bases de datos en postgres
Cambiamos la contraseña
# \password postgres  # new password: password
# \q   salir de postgres
```

3.4. Creamos la base de datos labdb:

```
$ sudo -u postgres createdb labdb
OR
$ sudo -u postgres psql
# create database labdb
# \l   Y vemos la BD labdb
# \q
```

4. Creación y configuración del proyecto djangolab, y app academic desde el entorno virtual (no salir).

4.1. Creación y primera ejecución

```
(djangolabenv) $ django-admin.py startproject djangolab ~/djangolab
```

Configurar settings.py

```
$ nano djangolab/settings.py
```

Modificar:

```
ALLOWED_HOSTS = ['*']
```

Salvar

Verifica si el proyecto se está ejecutando correctamente:

```
$ python3 manage.py runserver 0:8080
$ python3 manage.py runserver 0:8000
```

4.2. Creación de la app academic

```
$ python3 manage.py startapp academic
```

4.3. Copiar los siguientes archivos:

```
academic/admin.py
academic/forms.py
academic/models.py
academic/tests.py
academic/urls.py
academic/views.py
djangolab/settings.py
djangolab/urls.py
media/*
static/*
templates/*
```

4.4. Reemplazar los siguientes archivos por tus imágenes:

```
media/images/logo.png          <-- logo del institutos en formato png
media/images/no-img.png         <-- logo del institutos en formato png
static/academic/img/logo.ico   <-- logo del institutos en formato ico
static/academic/img/logo.png   <-- logo del institutos en formato png
```

```
static/academic/img/logocarrera.png<-- logo del institutos en formato png
```

5. Instalación y Configuración de Django Publications desde el entorno virtual (no salir):

Link: <https://github.com/lucastheis/django-publications>

5.1. Instalando & configurando Django Publications

```
$ pip3 install django-publications
> Successfully installed Pillow-6.2.1 django-publications-1.0.0
```

5.2. Reemplazar los siguientes archivos a los siguientes posiciones de publications a `djangoenv/lib/python3.6/site-packages/publications/`

```
templates/base.html
templates/publications/publications.html
templates/publications/publication.html
```

5.3. Verificar errores:

```
$ python3 manage.py check
> System check identified no issues (0 silenced).
```

6. Completar la configuración inicial del proyecto desde el entorno virtual (no salir):

6.1. Migración de la base de datos:

```
$ python3 manage.py makemigrations
$ python3 manage.py migrate
```

6.2. Creamos un superusuario para el administrador:

```
$ python3 manage.py createsuperuser
> user: nombre_usuario
> email: email@usuario.com
> password: xxxxxxxx
```

6.3. Ingresar a settings y habilitar la opción para collectstatic

```
$ nano djangoenv/settings.py
Linea 136 adicionar comentarios ""
```

```
"""
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static/"),
]
"""
```

AND en la línea 134 quitar comentario (#):

```
STATIC_ROOT = os.path.join(BASE_DIR, 'static/')
_6.4. Recolectar archivos estáticos en /static/_0
```

```
$ python3 manage.py collectstatic
> yes
> 121 static files copied to '/home/marcelo/djangoenv/static'.
```

Podrá observar que se creó nuevas carpetas y archivos en static

6.5. Ingresar a settings a la opción por defecto

```
$ nano djangoenv/settings.py
Linea 136 quitar comentarios:
```

```
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static/"),
]
```

AND en la línea 134 adicionar comentario (#):

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

6.6. Configurar el UFW firewall para la puerta 8000 o 8080:

6.6.1. Verificar estado del Firewall

```
$ sudo ufw status
```

6.6.2. (OPCIONAL) En caso de activar el Firewall, ejecutar los siguientes comandos con cuidado:

```
(OPCIONAL) $ sudo ufw default deny
```

```
$ sudo ufw enable
```

```
$ sudo ufw allow 8000
```

```
$ sudo ufw allow 2223
```

```
$ sudo ufw allow ssh
```

```
$ sudo ufw allow 'Apache Full'
```

```
(OPCIONAL) $ sudo ufw allow 8800
```

```
Rules updated
```

```
Rules updated (v6)
```

```
$ sudo ufw status
```

```
(OPCIONAL) $ sudo ufw disable
```

```
(OPCIONAL) $ sudo ufw reset ==> Resetear con la configuración inicial.
```

Link 1: <https://linuxize.com/post/how-to-setup-a-firewall-with-ufw-on-ubuntu-18-04/> Link

2: <https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-16-04>

```
$ sudo iptables -I INPUT -p tcp --dport 8000 -j ACCEPT
```

7. Iniciar servicio de DjangoLab V2.3.1

```
$ python3 manage.py runserver 0:8080
```

```
$ python3 manage.py runserver 0:8000
```

7.1. Ingresar al 127.0.0.1:8000/admin y acceder con las credenciales creadas en 6.2:

Ingresar a Academic > D. Cargos/Puestos del Instituto y añadir los siguientes puestos en este orden:

1) Director(a)

2) Coordinador(a)

3) Docente Investigador

4) Auxiliar de Investigación

5) Estudiante Investigador Junior

5) Investigador Asociado

6) Personal Administrativo

* RESPETA EL ORDEN, PUES INFLUENCIARÁ EN LA SECCIÓN DE PERSONAL Y REPORTES

7.2. Optimización de Postgres:

```
$ sudo -u postgres psql
```

```
postgres=# ALTER ROLE postgres SET client_encoding TO 'utf8';
```

```
> ALTER ROLE
```

```
postgres=# ALTER ROLE postgres SET default_transaction_isolation TO 'read committed';
```

```
> ALTER ROLE
```

```
postgres=# ALTER ROLE postgres SET timezone TO 'UTC';
```

```
> ALTER ROLE
```

```
postgres=# \q
```

8. Configuración de Apache Server por mod_wsgi:

_ Concluido con la configuración, el sistema estará funcionando correctamente. Desactivamos el entorno virtual: _

```
(djangelabenv) $ deactivate
```

_8.1. Configurar WSGI. Editar el archivo de host virtual predeterminado:_0

```
$ sudo nano /etc/apache2/sites-available/000-default.conf
```

```
===== 000-default.conf =====
```

```
<VirtualHost *:80>
```

```
    ServerAdmin marcelopalma@fcpn.edu.bo
```

```
    DocumentRoot /var/www/html
```

```
    LogLevel warn
```

```
    ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
    CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
    Alias /static/ /home/marcelo/djangolab/static/
```

```
    <Directory /home/marcelo/djangolab/static>
```

```
        Require all granted
```

```
    </Directory>
```

```
    Alias /media/ /home/marcelo/djangolab/media/
```

```
    <Directory /home/marcelo/djangolab/media>
```

```
        Require all granted
```

```
    </Directory>
```

```
    # Next, add the following directory block
```

```
    <Directory /home/marcelo/djangolab/djangolab>
```

```
        <Files wsgi.py>
```

```
            Require all granted
```

```
        </Files>
```

```
    </Directory>
```

```
    WSGIDaemonProcess djangolab python-home=/home/marcelo/djangolab/djangolabenv python-path=/home/marcelo/djangolab/
```

```
    WSGIProcessGroup djangolab
```

```
    WSGIScriptAlias / /home/marcelo/djangolab/djangolab/wsgi.py
```

```
</VirtualHost>
```

```
=====
```

En caso de error:

Error: Django [Errno 13] Permission denied: '/var/www/media/animals/user_uploads'

<https://stackoverflow.com/questions/21797372/django-errno-13-permission-denied-var-www-media-animals-user-uploads>

8.2. (NO RECOMENDADO) Dar permisos a www-data para acceder a djangolab:

```
$ chmod 775 ~/djangolab
```

```
$ sudo chown :www-data ~/djangolab
```

8.3. (OPCIONAL) Configuración adicional del Firewall:

```
$ sudo ufw delete allow 8000
```

```
$ sudo ufw allow 'Apache Full'
```

```
$ sudo iptables -D INPUT -p tcp --dport 8000 -j ACCEPT
```

```
$ sudo iptables -I INPUT -p tcp --dport 80 -j ACCEPT
```

8.4. Verificación de configuración Apache2:

```
$ sudo apache2ctl configtest
```

```
> Syntax OK
```

```
# Caso exista un error (Debian):  
$ sudo a2enmod wsgi  
$ sudo apt-get install python3-pip apache2 libapache2-mod-wsgi-py3
```

8.5. Puesta en marcha del servidor:

```
$ sudo systemctl restart apache2
```

Auxiliares:

Trabajar con el shell del proyecto djangolab:

```
$ python3 manage.py shell  
>>> import django  
>>> from academic.models import *
```

Trabajar con el dbshell del proyecto djangolab para revisar la base de datos del proyecto y su contenido.

Agradecimientos:

Agradezco a @lucastheis por el desarrollo de Django Publications 1.0.0, el cual contribuyo al desarrollo de este DjangoLab