



# MAPA PRODUCTIVO DE VENEZUELA

## Manual de Instalación

<b>Proceso</b>	<b>Tipo de documento</b>	<b>Versión del documento</b>	<b>Responsable</b>	<b>Fecha de elaboración</b>
Desarrollo de Aplicaciones de Software	Manual de Instalación	1,0	Erwin Paredes	30/10/2014



FUNDACIÓN CENTRO NACIONAL DE DESARROLLO E  
INVESTIGACIÓN EN TECNOLOGÍAS LIBRES

MINISTERIO DEL PODER POPULAR PARA LA  
CIENCIA Y TECNOLOGÍA

---



## Licencia de Uso

*Copyright (c), 2007, CENDITEL.*

*Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".*

Una copia de la licencia puede obtenerse en los ficheros llamados “copyright.txt” en inglés, “copyright.es.txt” en español o en los siguientes sitios en Internet:

- <http://www.gnu.org/copyleft/fdl.html>
- <http://www.fsf.org/licensing/licenses/fdl.html>



## Historial de Revisiones

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Responsable(s)</b>
30/10/2014	1,0	Versión inicial del documento.	Erwin Paredes

## Tabla de Contenido

### Índice de contenido

Licencia de Uso.....	3
Historial de Revisiones.....	4
Tabla de Contenido.....	5
Manual de Instalación.....	12
Instalar un ambiente virtual Python.....	13
Crear un entorno virtual, para Python version 2.7.....	13
Activar el entorno virtual.....	13
Descargar Django 1.6.8 desde el sitio oficial y descomprimir:.....	14
Clonar repositorio del sistema Observatorio.....	14
Añadir parámetros de conexión a observatorio/procesos/settings.py .....	14
Instalar Postgresql y Postgis.....	15
Crear template de postgis .....	15
Crear base de datos a partir de la plantilla .....	15
Cargar base de datos procesos, sigesic y geodb .....	16
Instalar módulos dependencias .....	16
Ejecutar la aplicación de apoyo para la instalación:.....	16
Instalar otras dependencias:.....	17
En un navegador probar con la dirección URL: <a href="http://localhost:8000">http://localhost:8000</a> .....	17
Pasos para instalar los servidores web.....	17
Configurar el servicio uwsgi:.....	17
Para ampliar la información puede consultar la dirección URL.....	19
Instalar Celery para el control de tareas asincrónicas:.....	19
Instalar SQLALchemy.....	20
Instalar celery como un demonio.....	20
Puedes consultar la dirección URL en caso de dudas sobre el celery:.....	22
Instalar el servidor web nginx:.....	22
En un navegador colocar la dirección <a href="http://direccion_ip:8000/">direccion_ip:8000/</a> .....	24
Enhorabuena, ya puede hacer uso del sistema!.....	24
Consultar los archivos del manual de usuario.....	24

## Manual de Instalación

Instrucciones para la instalación del Sistema Observatorio del proyecto Mapa Productivo de Venezuela, en un computador con sistema operativo Linux.

Para verificar el sistema operativo instalado en el computador, desde un terminal o consola de comandos, ejecutar la siguiente instrucción:

```
lsb_release -a
```

ejemplo:

```
usuario@computer:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 7.1 (wheezy)
Release:       7.1
Codename:      wheezy
```

Una vez comprobado el sistema instalado, seguimos los pasos para la instalación en el sistema Operativo Debian Wheezy:

**Instalar un ambiente virtual Python**, ejecutar los comandos:

```
usuario@computer:~$ aptitude install libc6-dev python-dev
usuario@computer:~$ aptitude install python-setuptools
usuario@computer:~$ easy_install virtualenv
```

**Crear un entorno virtual, para Python version 2.7:**

```
usuario@computer:~$ cd $HOME ; mkdir ./virtualenv ; cd virtualenv
```

```
usuario@computer:~$ virtualenv --python=/usr/bin/python2.7 python2.7
usuario@computer:~$ virtualenv --distribute --python=/usr/bin/python2.7
python2.7
```

### **Activar el entorno virtual**

```
usuario@computer:~$ source ./python2.7/bin/activate
```

El prompt cambia para indicar el uso del ambiente virtual

```
(python2.7)usuario@computer:~$
```

### **Descargar Django 1.6.8 desde el sitio oficial y descomprimir:**

```
https://www.djangoproject.com/download/
```

```
(python2.7)usuario@computer:~$ tar -xzvf Django-1.6.8.tar.gz
```

```
(python2.7)usuario@computer:~$ cd Django-1.6.8
```

```
(python2.7)usuario@computer:~$ python setup.py install
```

### **Clonar repositorio del sistema Observatorio**

```
(python2.7)usuario@computer:~$ export GIT_SSL_NO_VERIFY=true
```

```
(python2.7)usuario@computer:~$ git clone
```

```
https://miv.cenditel.gob.ve/cadenas/scm/git/observatorio.git
```

### **Añadir parámetros de conexión a observatorio/procesos/settings.py**

```
(python2.7)usuario@computer:~$ nano observatorio/procesos/settings.py
```

Modificar las variables de la Base de datos:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.contrib.gis.db.backends.postgis',
        'NAME': 'procesos',
        'USER': 'administrador',
        'PASSWORD': 'clave',
        'HOST': 'localhost',
        'PORT': '5432',
    },
    'sigesic': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'sigesic_03162014',
        'USER': 'administrador',
        'PASSWORD': 'clave',
        'HOST': 'localhost',
        'PORT': '5432',
    },
    'geocadena_db': {
        'ENGINE': 'django.contrib.gis.db.backends.postgis',
        'NAME': 'geocadena',
        'USER': 'administrador',
        'PASSWORD': 'clave',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

Presione las teclas Ctrl+x, para guardar los cambios.

## Instalar PostgreSQL 9.3 y Postgis 2.1



```
(python2.7)usuario@computer:~$ aptitude install postgresql-9.3
(python2.7)usuario@computer:~$ aptitude install postgis postgresql-9.3-
postgis-2.1
```

### Crear las Base de datos

```
(python2.7)usuario@computer:~$ sudo -u postgres createdb procesos
(python2.7)usuario@computer:~$ sudo -u postgres createdb sigesic
(python2.7)usuario@computer:~$ sudo -u postgres createdb geocadena
```

### Crear template de postgis

```
(python2.7)usuario@computer:~$ su postgres
(python2.7)usuario@computer:~$ createdb template_postgis
(python2.7)usuario@computer:~$ createlang plpgsql template_postgis
(python2.7)usuario@computer:~$ psql -d template_postgis -f
/usr/share/postgresql/9.3/contrib/postgis-2.1/postgis.sql
(python2.7)usuario@computer:~$ psql -d template_postgis -f
/usr/share/postgresql/9.3/contrib/postgis-2.1/spatial_ref_sys.sql
```

### Crear base de datos a partir de la plantilla

```
(python2.7)usuario@computer:~$ createdb -T template_postgis geodb
```

### Cargar base de datos procesos, sigesic y geodb

```
(python2.7)usuario@computer:~$ cd observatorio/procesos
(python2.7)usuario@computer:~$ python manages.py syncdb
```

### Instalar módulos dependencias

```
(python2.7)usuario@computer:~$ cd observatorio/procesos/extapps
(python2.7)usuario@computer:~$ tar zxvf django-tastypie-0.11.0.tar.gz
```

```
(python2.7)usuario@computer:~$ cd django-tastypie-0.11.0/  
(python2.7)usuario@computer:~$ python setup.py install  
(python2.7)usuario@computer:~$ cd ..
```

```
(python2.7)usuario@computer:~$ tar zxvf pycaptcha-0.3-pre.tar.gz  
(python2.7)usuario@computer:~$ cd pycaptcha  
(python2.7)usuario@computer:~$ python setup.py install  
(python2.7)usuario@computer:~$ cd ..
```

```
(python2.7)usuario@computer:~$ tar zxvf libxml2-python-2.6.21.tar.gz  
(python2.7)usuario@computer:~$ cd libxml2-python-2.6.21/  
(python2.7)usuario@computer:~$ python setup.py install  
(python2.7)usuario@computer:~$ cd ..
```

```
(python2.7)usuario@computer:~$ tar zxvf Imaging-1.1.7.tar.gz  
(python2.7)usuario@computer:~$ cd Imaging-1.1.7/  
(python2.7)usuario@computer:~$ python setup.py install  
(python2.7)usuario@computer:~$ cd ..
```

### **Ejecutar la aplicación de apoyo para la instalación:**

```
(python2.7)usuario@computer:~$ python install.py
```

Presionar las teclas 1: Para instalar los requerimientos complementarios

Presionar las teclas 2: Para instalar las bases de datos

Presionar las teclas 3: Para instalar cargar los datos

Presionar las teclas 4: Para comprobar los requerimientos

Presionar las teclas 5: Para Salir de la ejecución

### **Instalar otras dependencias:**

```
(python2.7)usuario@computer:~$ easy_install django_databrowse
(python2.7)usuario@computer:~$ easy_install psycopg2
(python2.7)usuario@computer:~$ easy_install pygraphviz
(python2.7)usuario@computer:~$ python manage.py runserver
```

**En un navegador probar con la direccion URL: <http://localhost:8000>**

### **Pasos para instalar los servidores web**

```
(python2.7)usuario@computer:~$ apt-get install build-essential python
(python2.7)usuario@computer:~$ apt-get install python-dev
(python2.7)usuario@computer:~$ pip install uwsgi
(python2.7)usuario@computer:~$ pip install
http://projects.unbit.it/downloads/uwsgi-lts.tar.gz
```

### **Configurar el servicio uwsgi:**

```
(python2.7)usuario@computer:~$ cd /etc/uwsgi/apps-available
(python2.7)usuario@computer:~$ nano miv_uwsgi.ini
```

### **Copiar el contenido del archivo:**

```
# archivo para aplicacion MIV
[uwsgi]
uid = procesos
group = nogroup
# Django-related settings
# the base directory (full path)
chdir = /var/www/mvp/observatorio/procesos
# Django's wsgi file
module = wsgi
```

```
# the virtualenv (full path)
#home = /path/to/virtualenv
# process-related settings
# master
master = true
# maximum number of worker processes
processes = 1
# the socket (use the full path to be safe
#socket = /path/to/your/project/mysite.sock
#socket = /var/run/uwsgi/app/comun/socket
socket = 172.22.9.25:1597
# ... with appropriate permissions - may be needed
# chmod-socket = 664
# chmod-socket = 666
# clear environment on exit
vacuum = true
# max-requests - prueba con 1000
max-requests = 1000
# post-buffering - valor recomendado 4096
post-bufering = 4096
# threads - cantidad de hilos
threads = 10
daemonize=/var/log/uwsgi/app/miv_uwsgi.log
close-on-exec
log-x-forwarded-for
```

Guardar el archivo presionando las teclas Ctrl+X

### **Reiniciar el servicio:**

```
(python2.7)usuario@computer:~$ /etc/init.d/uwsgi restart
```

Nota: el código fuente del sistema se encuentra ubicado en el directorio: `/var/www/mvp/observatorio/procesos`, en caso contrario modificar la variable `chdir` en el archivo, colocando donde se encuentre ubicado.

Para ampliar la información puede consultar la dirección URL:

<http://uwsgi-docs.readthedocs.org/en/latest/index.html>

### **Instalar Celery para el control de tareas asincrónicas:**

```
(python2.7)usuario@computer:~$ apt-get install rabbitmq-server
(python2.7)usuario@computer:~$ pip install celery
(python2.7)usuario@computer:~$ rabbitmqctl add_user myuser mypassword
(python2.7)usuario@computer:~$ rabbitmqctl add_vhost myvhost
```

Cambiar las variables `myuser`, `mypassword` y `myvhost` por la de su preferencia, en este caso utilizaremos `administrador` como usuario y `clave` como password, así como `servidormvp` en `myvhost`, por lo que los comandos quedan:

```
(python2.7)usuario@computer:~$ rabbitmqctl add_user administrador clave
(python2.7)usuario@computer:~$ rabbitmqctl add_vhost servidormvp
(python2.7)usuario@computer:~$ rabbitmq-server
```

### **Instalar SQLALchemy**

```
(python2.7)usuario@computer:~$ pip install SQLALchemy
```

### **Instalar celery como un demonio:**

```
(python2.7)usuario@computer:~$ pip install celeryd
```

```
(python2.7)usuario@computer:~$ cd /etc/default/  
(python2.7)usuario@computer:~$ nano celeryd
```

copiar el siguiente contenido dentro del archivo:

```
# If enabled pid and log directories will be created if missing,  
# and owned by the userid/group configured.  
CELERY_CREATE_DIRS=1  
# Change this to true when done to enable the init.d script.  
# Default: false  
ENABLED="true"  
  
# Names of nodes to start  
# most will only start one node:  
CELERYD_NODES="worker"  
  
# but you can also start multiple and configure settings  
# for each in CELERYD_OPTS (see `celery multi --help` for examples).  
#CELERYD_NODES="worker1 worker2 worker3"  
  
# Absolute or relative path to the 'celery' command:  
CELERY_BIN="/usr/local/bin/celery"  
#CELERY_BIN="/virtualenvs/def/bin/celery"  
  
# App instance to use  
# comment out this line if you don't use an app  
CELERY_APP="celery_procesos"  
# or fully qualified:  
#CELERY_APP="proj.tasks:app"  
  
# Where to chdir at start.
```

```
CELERYD_CHDIR="/var/www/mvp/observatorio/procesos/"
```

```
# Extra command-line arguments to the worker
```

```
CELERYD_OPTS="--time-limit=3600 --concurrency=4"
```

```
# Name of the projects settings module
```

```
export DJANGO_SETTINGS_MODULE="settings"
```

```
# Path to celeryd_multi
```

```
#CELERYD_MULTI="$CELERYD_CHDIR/manage.py celeryd_multi"
```

```
# Name of the celery config module.
```

```
CELERY_CONFIG_MODULE="celeryconfig"
```

```
# %N will be replaced with the first part of the nodename.
```

```
CELERYD_LOG_FILE="/var/log/celery/%N.log"
```

```
CELERYD_PID_FILE="/var/run/celery/%N.pid"
```

```
# Workers should run as an unprivileged user.
```

```
# You need to create this user manually (or you can choose
```

```
# a user/group combination that already exists, e.g. nobody).
```

```
CELERYD_USER="procesos"
```

```
CELERYD_GROUP="procesos"
```

Nota: el código fuente del sistema se encuentra ubicado en el directorio: /var/www/mvp/observatorio/procesos, en caso contrario modificar la variable CELERYD\_CHDIR en el archivo, colocando donde se encuentre ubicado.

### Reiniciar el servicio celery:

```
(python2.7)usuario@computer:~$ /etc/init.d/celeryd restart
```

Puedes consultar la dirección URL en caso de dudas sobre el celery:  
<http://www.celeryproject.org/>

### Instalar el servidor web nginx:

```
(python2.7)usuario@computer:~$ aptitude install nginx
(python2.7)usuario@computer:~$ wget -O init-deb.sh
http://www.linode.com/docs/assets/1538-init-deb.sh
(python2.7)usuario@computer:~$ mv init-deb.sh /etc/init.d/nginx
(python2.7)usuario@computer:~$ chmod +x /etc/init.d/nginx
(python2.7)usuario@computer:~$ /usr/sbin/update-rc.d -f nginx defaults
(python2.7)usuario@computer:~$ cd /etc/nginx/sites-available
(python2.7)usuario@computer:~$ nano sigproductivo
```

copiar el contenido del archivo:

```
upstream django {
    server 127.0.0.1:8001;      # for a web port socket
}

server {
    listen 8000;
    server_name direccion_ip;
    access_log /var/log/nginx/sigproductivo.access.log;
    error_log /var/log/nginx/sigproductivo.error.log error;

    client_max_body_size 75M;
    client_body_buffer_size 256k;

    location /media{
```



```
    alias /var/www/mvp/observatorio/procesos/media;
}

location ~ ^/(static/admin)/.+ {
    alias /var/www/mvp/observatorio/procesos;
}

location ~ ^/sigproductivo(?<ruta>/.*)?$ {
    root /var/www/mvp/observatorio/procesos;
    try_files $uri @sigproductivo;
}

location / {
    uwsgi_pass 127.0.0.1:8001;
    include /etc/nginx/conf.d/uwsgi_params;
    #uwsgi_param UWSGI_SCHEME $http_x_forwarded_protocol;
    #uwsgi_param SCRIPT_NAME /geodjanfo/geodjango;
    #uwsgi_param PATH_INFO $ruta;
    #uwsgi_pass sigproductivo;
    #uwsgi_intercept_errors off;
    #uwsgi_read_timeout 300;
    error_page 404 media/404.html;
    # redirect server error pages to the static page /
50x.html
    #
    error_page 500 502 503 504 media/50x.html;
}
}
```

Sustituir la variable `direccion_ip` por la correspondiente (chequear con el comando `ifconfig`)

Guardar el archivo presionando las teclas `Ctrl+x`

Reiniciar el servidor web `nginx`

```
(python2.7)usuario@computer:~$ /etc/init.d/nginx restart
```

En un navegador colocar la dirección `ulr direccion_ip:8000/`, en mi caso:

<http://192.168.12.74:8000/>

Enhorabuena, ya puede hacer uso del sistema!.

Nota: Para la ejecución de algunos de estos comandos el usuario debe poseer permisos de superusuario o deben ser ejecutados como `root`.

Consultar los archivos del manual de usuario en el directorio:

`/var/www/mvp/observatorio/procesos/docs/Manual-Usuario/`