

Proyecto Módulo 2

Manual de Instalación

Cabrera Balderas Carlos Eduardo
Gonzalez Vargas Andrea Itzel

Índice

1. Requerimientos	2
2. Instrucciones	2
2.1. Instalación de dependencias	2
2.2. Creación de ambiente virtual	3
2.3. Configuraciones de la aplicación	3
2.4. Creación de la base de datos	4
2.5. Creación de sitio de Apache 2	4
2.6. GoAccess	5

El siguiente manual está pensado para la instalación de la aplicación en un equipo con **Debian** que utilice **Apache 2** como servidor web. Más específicamente, se probaron los pasos en un equipo con **Debian 8.10** y **Apache 2.4.10**

1. Requerimientos

Antes de continuar con la instalación de la aplicación web, es necesario tener instaladas las siguientes herramientas en el servidor:

- `python3` (`>= 3.4`)
- `apache2`
- `git`
- `sudo`

Es necesario, a demás, crear un usuario que será utilizado para correr la aplicación (en este caso se utiliza al usuario `djangouser`). Este usuario debe tener permisos para correr como administrador (sin necesitar ingresar contraseña) los siguientes comandos:

- | | |
|-------------------------|---------------------|
| ▪ <code>netstat</code> | ▪ <code>lsof</code> |
| ▪ <code>crontab</code> | ▪ <code>gzip</code> |
| ▪ <code>passwd</code> | ▪ <code>tac</code> |
| ▪ <code>ps</code> | ▪ <code>test</code> |
| ▪ <code>iptables</code> | |

A demás se debe de agregar al usuario `djangouser` al grupo `adm` para que pueda tener acceso a las bitácoras en `/var/log`.

2. Instrucciones

2.1. Instalación de dependencias

Una vez cubiertos los requerimientos anteriores, es posible continuar con la instalación. El primer paso es clonar el repositorio del proyecto:

```
$ git clone https://github.com/andreagonz/pmod2-pbsi
```

dentro del directorio creado (`pmod2-pbsi`) se encuentra el script de instalación de dependencias (`pmod2-pbsi/instalacion.sh`). Para instalar las dependencias iniciales se debe de correr:

```
$ ./instalacion.sh -d
```

2.2. Creación de ambiente virtual

El siguiente paso es la creación del ambiente virtual que contendrá los ejecutables de `python` que utilizará la aplicación, para lo cual basta ejecutar:

```
$ ./instalacion.sh -v /home/djangousr/env
```

donde `/home/djangousr/env` será la ubicación del ambiente virtual.

Se debe entonces copiar el directorio `pmod2-pbsi/proyecto` al lugar donde se quiera tener los archivos del proyecto, por ejemplo:

```
$ sudo cp -r pmod2-pbsi/proyecto /var/www
```

El usuario `djangousr` debe ser el dueño de todo el directorio:

```
$ sudo chown -R djangousr:djangousr /var/www/proyecto
```

2.3. Configuraciones de la aplicación

Dentro del archivo de configuración `/var/www/proyecto/proyecto/settings.py` es *muy importante* modificar la variable `SECRET_KEY` por una llave generada aleatoriamente, para lo cual se puede utilizar la página <https://www.miniwebtool.com/django-secret-key-generator/>.

Dentro de este mismo archivo se puede cambiar la ubicación de la base de datos, que por defecto es `.../proyecto/db.sqlite3`, para lo cual se debe modificar la siguiente sección:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

donde `'NAME'` indica la ubicación de la base de datos.

De igual manera, se puede modificar la ubicación del archivo de configuración donde se especificarán las bitácoras que leerá la aplicación, junto con el script que al ser ejecutado permite obtener la información que se mostrará (por defecto estos se encuentran en `.../proyecto/conf`). Para este fin basta modificar las variables `ARCHIVO_CONFIGURACION`¹ y `SCRIPT_SECCIONES`, que se encuentran al final del archivo.

¹Para más información del archivo de configuración, ver el manual de usuario.

2.4. Creación de la base de datos

A continuación se debe crear la base de datos siguiendo los siguientes pasos:

Activar el ambiente virtual

```
$ su djangousr
$ source /home/djangousr/env/bin/activate
```

Verificar que la variable de ambiente PATH incluye al directorio /home/djangousr/env/bin.

```
$ echo $PATH
```

Dentro del directorio /var/www/proyecto correr los siguientes comandos:

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

El último comando creará la base de datos en la ubicación que se indicó en el archivo de configuración, junto con las tablas necesarias. Se debe verificar que el dueño de la base de datos sea el usuario djangousr.

Para crear el primer usuario de la aplicación se debe correr el comando

```
$ python3 manage.py createsuperuser
```

con el cual se pedirá que se ingresen los datos del nuevo usuario.

2.5. Creación de sitio de Apache 2

Finalmente, se debe crear un nuevo sitio de Apache 2 para la aplicación, en este se especificará el directorio del proyecto junto con el del ambiente virtual, a demás del usuario con el que se correrá la aplicación.

A continuación se muestra un ejemplo del archivo de configuracion del sitio:

```
<VirtualHost *:80>
    Alias /favicon.ico /var/www/proyecto/static/favicon.ico
    Alias /static/ /var/www/proyecto/static/

    WSGIScriptAlias / /var/www/proyecto/proyecto/wsgi.py
    WSGIDaemonProcess localhost user=djangousr python-home=/home/djangousr/env \
        python-path=/var/www/proyecto/
    WSGIProcessGroup localhost

    ErrorLog ${APACHE_LOG_DIR}/error_proyecto.log
    CustomLog ${APACHE_LOG_DIR}/acceso_proyecto.log combined

    <Directory /var/www/proyecto/static>
```

```

        Require all granted
    </Directory>

    <Directory /var/www/proyecto/proyecto>
        <Files wsgi.py>
            Require all granted
        </Files>
    </Directory>
</VirtualHost>

```

Basta entonces con habilitar el sitio creado y reiniciar **Apache 2**, tras lo cual se podrá ingresar a la aplicación.

```

# a2ensite sitio.conf
# systemctl restart apache2

```

2.6. GoAccess

Para poder mostrar las gráficas de alguna bitácora de acceso de **Apache 2** en la aplicación, se debe de hacer la instalación de la herramienta **GoAccess**, para lo cual basta correr el script de instalación de la siguiente manera

```
$ ./instalacion.sh -g /home/djangouser
```

donde `/home/django` será el lugar donde se descargarán los archivos de la herramienta.

Una vez terminada la instalación se debe de correr el comando

```

$ goaccess /var/log/apache2/access.log \
  -o /var/www/proyecto/secciones/templates/secciones/graficas.html \
  --real-time-html --daemonize

```

lo cual correrá el demonio de **GoAccess** que se encargará de interpretar la información de la bitácora `/var/log/apache2/access.log`, y escribirá la información gráficamente en el archivo `/var/www/proyecto/secciones/templates/secciones/graficas.html`, que es leído y mostrado por la aplicación.