Manual de instalación

Indice .

En este manual se detalla lo necesario para poner en producción el proyecto. Se partirá desde un sistema operativo Linux como Ubuntu Server.

1. Instalación de dependencias.

Para que la aplicación funcione es necesario instalar el paquete pip y unzip.

```
$ apt-get install python-pip unzip apache2 libapache2-
wsgi git
```

Una vez instalado estos dos paquetes, instalamos las siguientes dependencias:

Pip :	Apt-get:
Django==1.11.11	pdftk
django-extensions==2.0.5	poppker-utils
django-shell-plus==1.1.7	
djangorestframework==3.8.2	
pdfminer==20140328	
Pillow==5.0.0	
psycopg2==2.7.4	
PyPDF2==1.26.0	
xhtml2pdf==0.2.1	

2. Copia del proyecto.

Nos descargamos el proyecto de GitHub :

```
$ git clone  https://github.com/edtlopez/ApercibimientosAuto
```

Copiamos el proyecto dentro del directorio /var/www de apache. Y eliminamos los archivos html por defecto.

```
$ rm /var/www/*
$ cp -r ApercibimientosAuto/DjangoProyecto/* /var/www
```

3. Configuración de apache.

Modificamos el fichero 000-default.conf que se encuentra en el directorio /etc/apache2/sites-available. Y añadimos las siguientes lineas.

```
WSGIScriptAlias / /var/www/WebGada/wsgi.py
WSGIDaemonProcess gada python-home=/var/www
WSGIProcessGroup gada



Alias /static /var/www/pdf/static

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

 <Directory /var/www/pdf/static>
 Require all granted
 </Directory>
```

Una vez configurado podemos ver la linea WSGIProcessGroup la cual indica el nombre del grupo que tendrá acceso a las carpetas por lo que es necesario crear un dicho grupo y poner los permisos correspondientes a estos.

```
$ groupadd gada
$ chown www-data:gada -R /var/www
```

3.2 Configuración SSL.

Para activar SSL en apache habrá que activar el modulo SSL de apache y configurar el fichero **ssl-default.conf** con la configuración del fichero anterior a la que se le añadirá las siguientes lineas.

```
 SSLEngine on
 SSLCertificateFile      /etc/apache2/cert/server.crt
 SSLCertificateKeyFile /etc/apache2/cert/server.key
```

Donde indicaremos la ubicación de nuestro certificado y nuestra clave privada. También es necesario modificar el usuario del proceso WSGI este corre con el usuario “gada” y habrá que ponerlo como “www-data”.

```
 WSGIDaemonProcess www-data python-home=/var/www/
 WSGIProcessGroup www-data
```

4. Modificar variables de entorno.

Apache posee un fechero en /etc/apache2 llamado envvars este contiene unas variables las cuales se carga cuando se ejecuta el proceso de apache, dentro de este archivo es necesario añadir estas dos.

```
export LANG='es_ES.UTF-8'
export LC_ALL='es_ES.UTF-8'
```

Esto ara que el interprete de python funcione de manera predeterminada con UTF-8 y no tengamos problemas con los caracteres raros.

5. Fin de configuración de apache.

Una vez echo esto ya tenemos configurado nuestro servidor web solo es necesario reiniciar el servicio apache2.

```
$ service apache2 restart
```

6. Configuración e instalación Base dedatos (Postgres).

Instamos desde el repositorio la base de datos postgres. Una vez instalado accedemos mediante psql y creamos el usuario y la base de datos en la que el usuario de la aplicación tendrá permisos al ser el propietario.

```
CREATE USER gada WITH PASSWORD '{PASSWORD}' ;
CREATE DATABASE docker with owner gada;
```

Configuramos el fichero setings.py de la aplicación web con las credenciales para que se pueda conectar la aplicación web.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'gada',
        'USER': 'gada',
        'PASSWORD': 'gadapasswd',
        'HOST': '127.0.0.1',
        'PORT': '',
    }
}
```

Una vez configurado la conexión a la base de datos nos dirigimos al directorio de la aplicación y ejecutamos .

```
$ python manage.py migrate
```

Esto creara las tablas que necesitamos para que nuestra aplicación funcione. Por ultimo creamos un usuario administrador con el cual poder acceder a la aplicación.

```
$ python manage.py createsuperuser
```

7. Instalación pgmyadmin.

Para poder administrar la base de datos y limpiar los registros de la misma de una manera cómoda se propuso la instalación de esta aplicación para administrar el servidor postgres.

Para esto Instalamos la aplicación desde el repositorio.

```
$ sudo apt-get install phppgadmin
```

Una vez instalado accedemos a la dirección.

```
http:{ipdelservidor}/phppgmyadmin
```

Accedemos a la base de datos usando el usuario y la contraseña que pusimos para la aplucación web. Esta cuenta no tiene privilegios de administrador pero tiene los permisos necesarios para modificar las tablas de la aplicación web.