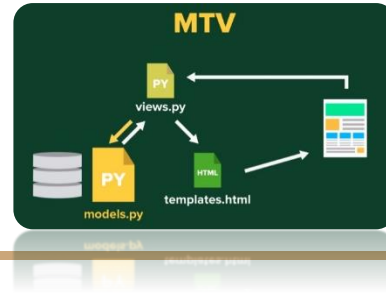


Instalación Django

Requisitos:



1. Tener Instalado Python 3.6.4



2. Instalar Gestor de Contenido PIP

En Windows usaremos los comandos:

pip install

pip install requests

```
C:\Users\Julio>pip install
You must give at least one requirement to install (see "pip help install")


C:\Users\Julio>pip install requests
Collecting requests
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
    100% |#####| 92kB 1.6MB/s
Collecting idna<2.7,>=2.5 (from requests)
  Downloading idna-2.6-py2.py3-none-any.whl (56kB)
    100% |#####| 61kB 1.9MB/s
Collecting certifi>=2017.4.17 (from requests)
  Downloading certifi-2017.11.5-py2.py3-none-any.whl (330kB)
    100% |#####| 337kB 946kB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests)
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
    100% |#####| 143kB 1.6MB/s
Collecting urllib3<1.23,>=1.21.1 (from requests)
  Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)
    100% |#####| 133kB 1.9MB/s
Installing collected packages: idna, certifi, chardet, urllib3, requests
Successfully installed certifi-2017.11.5 chardet-3.0.4 idna-2.6 requests-2.18.4 urllib3-1.22

C:\Users\Julio>
```

3. Instalar Virtualenv 15.1.0

Lo descargamos de <https://pypi.python.org/pypi/virtualenv>

| File | Type | Py Version | Uploaded on | Size |
|---|--------------|------------|-------------|------|
| virtualenv-15.1.0-py2.py3-none-any.whl (md5, pgp) | Python Wheel | py2.py3 | 2016-11-16 | 1MB |
| virtualenv-15.1.0.tar.gz (md5, pgp) | Source | | 2016-11-16 | 1MB |

 <https://pypi.python.org/packages/d4/0c/9840c08189e030873387a73b90ada981885010dd9aea134d6de30cd24cb8/virtualenv-15.1.0.tar.gz#md5=44e19f4134906fe2d75124427dc9b716>

- Vamos a la ruta de la descarga de virtualenv-15.1.0 ejecutamos el comando:

python setup.py install

```
C:\Users\Julio\Desktop\virtualenv-15.1.0>python setup.py install
running install
running bdist_egg
running egg_info
writing virtualenv.egg-info\PKG-INFO
writing dependency_links to virtualenv.egg-info\dependency_links.txt
writing entry points to virtualenv.egg-info\entry_points.txt
writing top-level names to virtualenv.egg-info\top_level.txt
reading manifest file 'virtualenv.egg-info\SOURCES.txt'
```

- Ahora crearemos un Entorno Virtual:

python -m virtualenv myenv

```
C:\Users\Julio>python -m virtualenv myenv
Using base prefix 'C:\\Users\\Julio\\AppData\\Local\\Programs\\Python\\Python36'
New python executable in C:\Users\Julio\myenv\Scripts\python.exe
Installing setuptools, pip, wheel...done.
```

- Activamos el Entorno Virtual:

entramos en Scripts dentro de **myenv** ejecutamos el archivo **activate**

```
C:\Users\Julio>.\myenv\Scripts\activate
```

4. Instalamos Django

Después de tener nuestro entorno virtual activado ejecutamos el comando:

pip install django

```
(myenv) C:\Users\Julio>pip install django
Collecting django
  Downloading Django-2.0.1-py3-none-any.whl (7.1MB)
    100% |#####| 7.1MB 107kB/s
Collecting pytz (from django)
  Downloading pytz-2017.3-py2.py3-none-any.whl (511kB)
    100% |#####| 512kB 717kB/s
Installing collected packages: pytz, django
Successfully installed django-2.0.1 pytz-2017.3
```

- Creamos nuestro proyecto:

Antes de ejecutar dicho comando iremos al directorio donde lo queramos tener alojado.

django-admin.py startproject nombreproyecto

```
(myenv) C:\Users\Julio>django-admin.py startproject web_jcontreras
```

```
Directorio de C:\Users\Julio\web_jcontreras

16/01/2018  20:41    <DIR>          .
16/01/2018  20:41    <DIR>          ..
16/01/2018  20:41                561 manage.py
16/01/2018  20:41    <DIR>          web_jcontreras
                1 archivos          561 bytes
                3 dirs  72.393.158.656 bytes libres
```

- Arrancamos el servidor:

python manage.py runserver

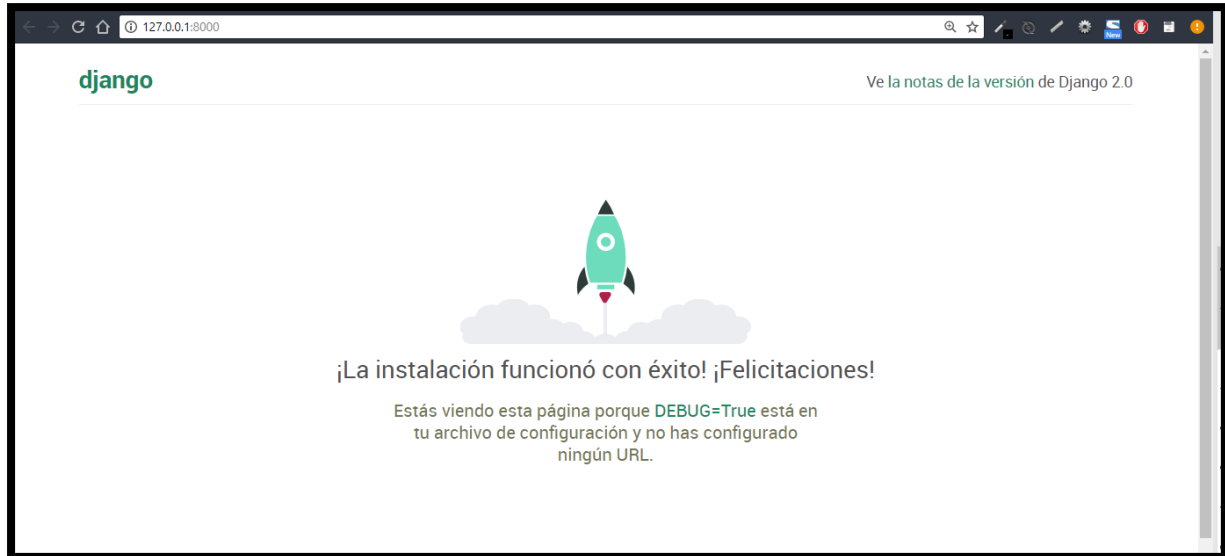
```
(myenv) C:\Users\Julio\web_jcontreras>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 14 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
January 16, 2018 - 21:15:05
Django version 2.0.1, using settings 'web_jcontreras.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[16/Jan/2018 21:15:35] "GET / HTTP/1.1" 200 16559
[16/Jan/2018 21:15:35] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
[16/Jan/2018 21:15:36] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 80304
[16/Jan/2018 21:15:36] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 82564
[16/Jan/2018 21:15:36] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 81348
```

5. ¡Ya estamos corriendo Django!

Podemos comprobar que nuestro servidor está activo en: <http://127.0.0.1:8000/>



- Paramos el servidor Control + C:

Ahora migraremos todo lo que no se haya instalado correctamente con el comando:

python manage.py migrate

```
(myenv) C:\Users\Julio\web_jcontreras>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
```

6. Creamos nuestra APP

- Dentro del proyecto creado ejecutamos:

`python manage.py startapp nombre`

```
(myenv) C:\Users\Julio\web_jcontreras>python manage.py startapp juliocontreras
```

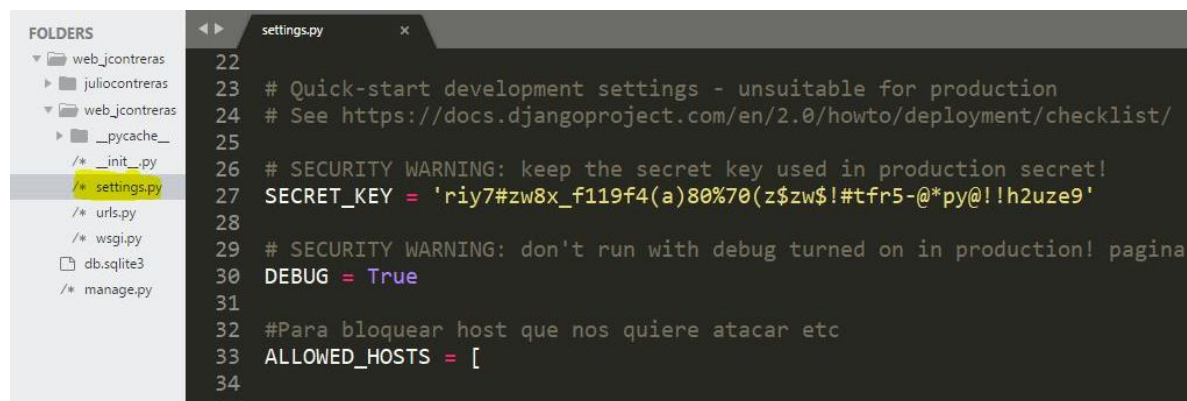
- Esta es la estructura de nuestra APP:

```
Directorio de C:\Users\Julio\web_jcontreras\juliocontreras
16/01/2018 22:34 <DIR> .
16/01/2018 22:34 <DIR> ..
16/01/2018 22:34      66 admin.py
16/01/2018 22:34     108 apps.py
16/01/2018 22:34 <DIR> migrations
16/01/2018 22:34      60 models.py
16/01/2018 22:34      63 tests.py
16/01/2018 22:34      66 views.py
16/01/2018 22:34      0 __init__.py
      6 archivos      363 bytes
      3 dirs 71.925.755.904 bytes libres
```

- Y esta la estructura del proyecto:

```
Directorio de C:\Users\Julio\web_jcontreras\web_jcontreras
16/01/2018 21:14 <DIR> .
16/01/2018 21:14 <DIR> ..
17/01/2018 01:45   3.588 settings.py
17/01/2018 01:47     889 urls.py
16/01/2018 20:41     421 wsgi.py
16/01/2018 20:41      0 __init__.py
17/01/2018 01:47 <DIR> __pycache__
      4 archivos     4.898 bytes
      3 dirs 71.893.516.288 bytes libres
```

- Archivo de configuración más importante de Django: **settings.py**



```
22
23 # Quick-start development settings - unsuitable for production
24 # See https://docs.djangoproject.com/en/2.0/howto/deployment/checklist/
25
26 # SECURITY WARNING: keep the secret key used in production secret!
27 SECRET_KEY = 'riy7#zw8x_f119f4(a)80%70(z$zw$!#tfr5-@*py@!!h2uze9'
28
29 # SECURITY WARNING: don't run with debug turned on in production! pagina
30 DEBUG = True
31
32 #Para bloquear host que nos quiere atacar etc
33 ALLOWED_HOSTS = [
34
35
```

- En **settings.py** tenemos **ROOT_URLCONF** aquí enrutaremos nuestro archivo de direcciones del proyecto:

```
60
61 ( ROOT_URLCONF = 'web_jcontreras.urls' )
62
```

Podemos ver como dentro del fichero. urls tenemos la ruta a la página de Administrador

A la cual tenemos acceso: <http://127.0.0.1:8000/admin/login/?next=/admin/>

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
]
```

- **Settings.py** sección **DATABASES** enlazaremos nuestra base de datos, por defecto viene instalada **sqlite3**.

```
84
85 DATABASES = {
86     'default': {
87         'ENGINE': 'django.db.backends.sqlite3',
88         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
89     }
90 }
91
```

- **Settings.py** en la sección **INSTALLED_APPS** agregaremos la APP que hemos creado **‘nombre.apps.NombreConfig’**

Ahora levantamos nuestro servidor: **python manage.py runserver**

```
1
2 INSTALLED_APPS = [
3     'django.contrib.admin',
4     'django.contrib.auth',
5     'django.contrib.contenttypes',
6     'django.contrib.sessions',
7     'django.contrib.messages',
8     'django.contrib.staticfiles',
9     'juliocontreras.apps.JuliocontrerasConfig',
10 ]
11
```

- Dentro de nuestra **APP** en el directorio **migrations** archivo **apps.py** podemos ver la ruta que hemos escrito anteriormente:



```
1 from django.apps import AppConfig
2
3 class JuliocontrerasConfig(AppConfig):
4
5     #configuracion a nivel app conviene dej
6
7     name = 'juliocontreras'
8
9
```