



Taller Deploy de un Proyecto Django Framework Python

https://github.com/dpineiden/taller_django

Flisol Santiago Chile

19 julio 2016

David Pineda Osorio

dahalpi@gmail.com



Introducción

Django es una herramienta para el desarrollo web, en este taller daremos a conocer los puntos de partida necesarios para crear un website.

Será necesario tener habilidades de programación, al menos en Python y comprender el idioma inglés.

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

django



Objetivos y Alcance



Objetivo

- Hacer un deploy para un proyecto Django

Alcance

- Solo hasta tener el runserver ok



¿Qué es el HTTP?

- Es una forma de intercambiar información llamado *protocolo de transporte de hipertexto*

Acciones

- GET: petición de datos que servidor entrega a cliente
- POST: entrega de información a servidor

PUT y +

https://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol





El lenguaje de programación Python

Lenguaje multiplataforma de script.

No compilado

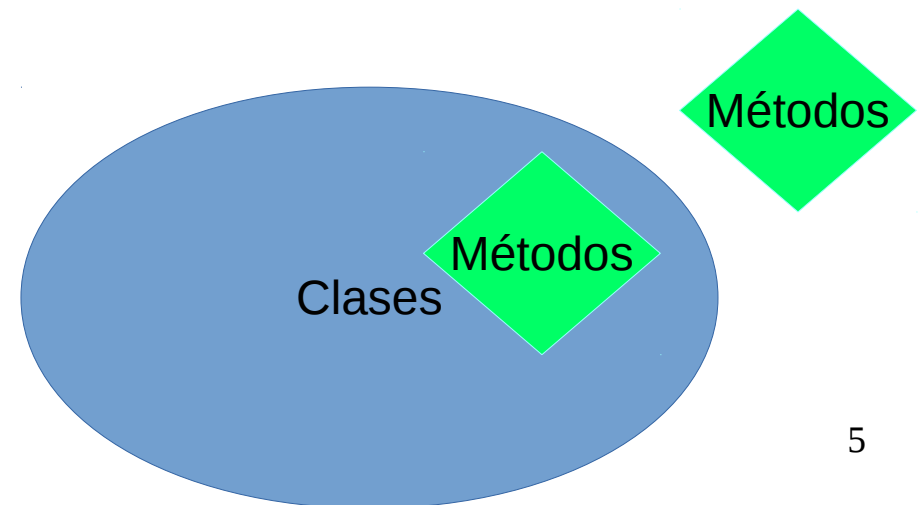
Característica **CLARIDAD DEL CÓDIGO**

{variables, listas, diccionarios, objetos, generadores, socket, input/output, línea de comando, scripts, etc}

Web:

- <https://www.python.org/>

- Clases: es la forma de definir objetos
- Métodos: forma de definir funcionalidades sobre elementos u objetos



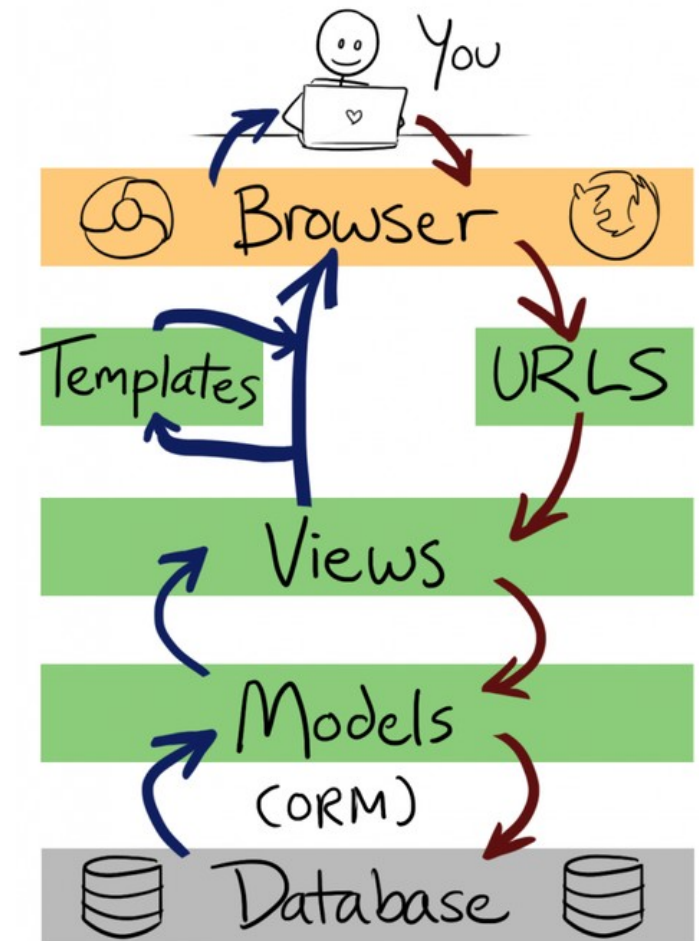


Estructura de un proyecto Django

Estructura de Carpetas

- |— apps
 - | |— accounts
 - | |— contactform
 - | |— localizacion
 - | |— pagina
 - | |— principal
 - | |— propiedad
 - | |— usuario
- |— files
 - | |— geo
 - | |— media
 - | |— static
 - | |— templates
- |— mjd
 - | |— fixtures
 - | |— __pycache__
 - | |— settings

Estructura de Sistema





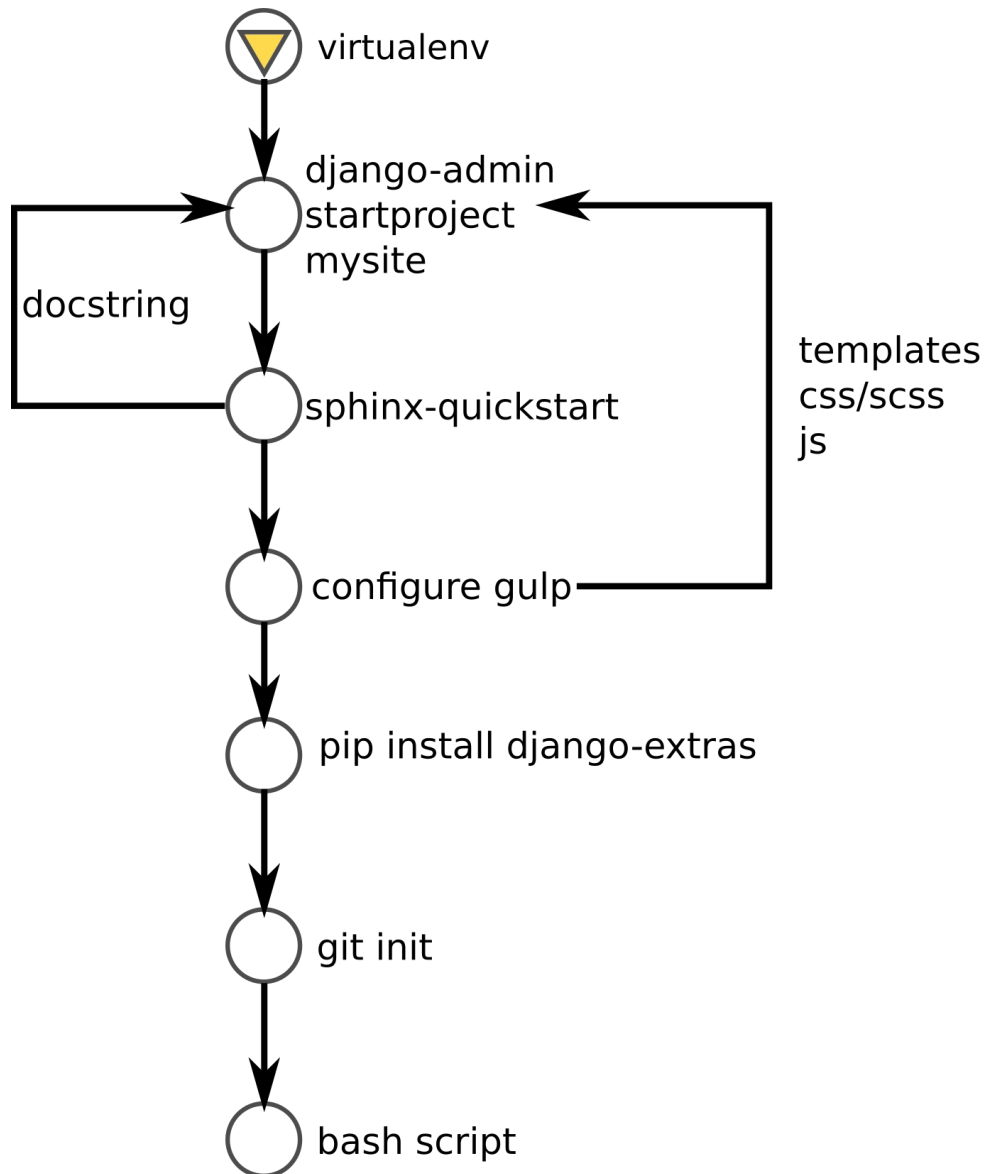
Herramientas para Desarrollo

- GIT
- GULP
- SPHINX
- BASH
- CONSOLA Python
- Elasticsearch (búsquedas)





Procedimiento





I.- El ambiente virtual

- Instalar:

```
pip3 install virtualenv
```

```
pip3 install virtualenvwrapper
```

- En `.profile` o `.bash_rc` poner:

```
export WORKON_HOME=$HOME/.virtualenvs
```

```
export PROJECT_HOME=$HOME/Proyectos
```

```
export VIRTUALENV_PYTHON=/usr/bin/python3.5
```

```
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3.5
```

```
source /usr/local/bin/virtualenvwrapper.sh
```

- En `<mi_ambiente>/bin/postactivate` agregar variables de ambiente



Crear proyecto base django

- En ambiente <mi_ambiente>
- Ejecutar:
`django-admin startproject <mi_proyecto>`
- Adentro, crear carpeta apps
- En `mi_proyecto/mi_proyecto` modificar `settings.py` a carpeta `settings/*`
`settings.py → base.py`
`touch development.py`
- Variables de django: en `.virtualenvs/mjd/bin/postactivate`



Sistema de Documentación

```
> pip install sphinx
```

```
> sphinx-autobuild
```

web: <http://www.sphinx-doc.org/en/stable/>

- Crear documentación de proyecto

```
> sphinx-quickstart
```

- Añadir detección de docstrings [PEP 257,287], en sources/conf.py:

```
import sys
```

```
import os
```

```
import django
```

```
os.environ['DJANGO_SETTINGS_MODULE'] = 'mjd.settings.development'
```

```
sys.path.insert(0, os.path.abspath('../..mjd'))
```

```
from django.conf import settings
```

```
settings.configure()
```

```
django.setup()
```

ref: <http://www.marinamele.com/2014/03/document-your-django-projects.html>



GULP: gestor de tareas dev

- <http://gulpjs.com/>
- Necesario: node y comando npm
 - >npm install --global gulp-cli
 - >npm init
 - >npm install --save-dev gulp
 - >touch gulpfile.js
 - >npm install gulp bower
 - >npm install gulp-sass gulp-minify-css gulp-rename gulp-gzip gulp-livereload
 - >bower install foundation



Contenido gulpfile.js





Instalar extras (requeriments)

>pip install django-extensions:

- Permite correr consola python con proyecto django y sus modelos cargados, crear esquema de modelos, correr servidor con extras

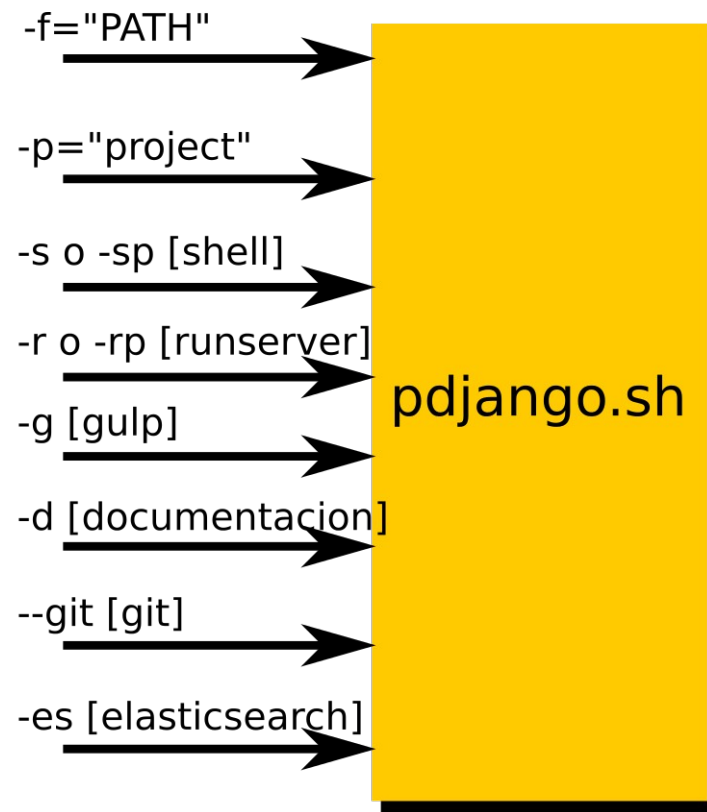
>pip install -r requeriments.txt

- Permite instalar el paquete de requerimientos de un proyecto



La magia de bash

```
>pdjango -f="PROJECT" -p="project"  
-sp -g -d -rp --git -es
```





GIT, para cerrar el círculo

```
>git init
```

```
>nano .gitignore
```

```
# Byte-compiled / optimized  
/ DLL files
```

```
__pycache__/
```

```
*.py[cod]
```

```
*migrations/*
```

```
!__init__.py
```

```
*node_modules/*
```

```
>git add . --all
```

```
>git commit -m "se  
inicia proyecto  
django"
```

```
>git push -u origin  
master
```




Método de Desarrollo

¡Todo parte con una idea!



Escoger método de desarrollo (ej: agile, scrum, etc)

Bocetos

{UML, casos de uso, diagramas relacionales}

django

En apps/<aplicacion>/

Modelos
{models.py}

Formularios
{forms.py}

Vistas
{views.py}

Templates
{templates/*.html}

Vistas
{urls.py}



La base de datos

- **El lugar digital** en dónde tus datos del sistema Django son almacenados: ***sqlite, mysql, postgres***
- Una vez definido un modelo (clase que se traduce en una tabla para la BBDD), actualizar la estructura:
 - 1) `python manage.py makemigrations`
 - 2) `python manage.py migrate`





Material de apoyo

- Documentación Django

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

- Marina Melee Taskbuster

<http://www.marinamele.com/taskbuster-django-tutorial>

- Web de View Class

<https://ccbv.co.uk>

- Readthedocs

<https://readthedocs.org/>

- Stackoverflow

<http://stackoverflow.com/questions/tagged/django>

- Github

<https://github.com/>

- Django Two Scoops

<https://www.twoscoopspress.com/products/two-scoops-of-django-1-8>

- Django by examples

<http://it-ebooks.info/book/6597/>

- Django Girls

<https://djangogirls.org/>