

## Tarea 2 - Proyectos en Django

Django es un framework de aplicaciones web gratuito y de código abierto (open source) escrito en Python. Un framework web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.

Cuando construyes un sitio web, siempre necesitas un conjunto de componentes similares: una manera de manejar la autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para tu sitio web, formularios, una forma de subir archivos, etc.

### Creando un proyecto

Un proyecto es un conjunto de configuraciones y aplicaciones para un sitio web determinado. Un proyecto puede contener aplicaciones múltiples.

La primera vez que utiliza Django, tendrá que hacerse cargo de ciertas configuraciones iniciales. Concretamente, tendrá que autogenerar un código que establezca un Django project – un conjunto de ajustes para una instancia de Django, incluida la configuración de la base de datos, opciones específicas de Django y configuraciones específicas de la aplicación.

Desde la línea de comandos, cambie a un directorio donde le gustaría almacenar su código, luego, ejecute el siguiente comando:

```
$ django-admin startproject mysite
```

Esto creará un directorio mysite en su directorio actual

Veamos lo que el comando startproject creó:

```
mysite/  
  manage.py  
  mysite/  
    __init__.py  
    settings.py  
    urls.py  
    wsgi.py
```

Estos archivos son:

- **El directorio raíz externo mysite/** solo es un contenedor de su proyecto. Su nombre no es relevante para Django; usted puede cambiarle el nombre a lo que quiera.
- **manage.py:** Una utilidad de la línea de comandos que le permite interactuar con este proyecto Django de diferentes formas.
- **En interior del directorio mysite/** es el propio paquete de Python para su proyecto. Su nombre es el nombre del paquete de Python que usted tendrá que utilizar para importar todo dentro de este (por ejemplo, mysite.urls).\
- **mysite/\_\_init\_\_.py:** Un archivo vacío que le indica a Python que este directorio debería ser considerado como un paquete Python.
- **mysite/settings.py:** Ajustes/configuración para este proyecto Django. Django settings le indicará todo sobre cómo funciona la configuración.
- **mysite/urls.py:** Las declaraciones URL para este proyecto Django; una «tabla de contenidos» de su sitio basado en Django.
- **mysite/wsgi.py:** Un punto de entrada para que los servidores web compatibles con WSGI puedan servir su proyecto.

El servidor de desarrollo

Comprobemos que su proyecto Django funciona. Cambie al directorio externo mysite, si todavía no lo ha hecho, y ejecute los siguientes comandos:

```
$ python manage.py runserver
```

Verá la siguiente salida en la línea de comandos:

```
Performing system checks...
System check identified no issues (0 silenced).
You have unapplied migrations; your app may not work properly
until they are applied.
Run 'python manage.py migrate' to apply them.

octubre 21, 2018 - 15:50:53
Django version 2.1, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Con esto se ha iniciado el servidor en desarrollo de Django, El servidor de desarrollo es un servidor web ligero escrito puramente en Python. se ha incluido con Django para que pueda desarrollar cosas rápidamente sin tener que lidiar con la configuración de un servidor en producción, como Apache, hasta que esté listo para la producción.

Se debe tener en cuenta que no se debe utilizar este servidor en algo parecido a un entorno de producción. Está pensado sólo para usarse durante el desarrollo del proyecto.

#### Bibliografía:

<https://docs.djangoproject.com/es/2.1/intro/>

<https://tutorial.djangogirls.org/es/django/>