

## Investigación: ¿Cómo hacer un proyecto en Django?

Django es sencillamente una colección de bibliotecas escritas en el lenguaje de programación Python. Para desarrollar un sitio usando Django escribes código Python que utiliza esas bibliotecas.

Como primer paso para crear un **proyecto** se debe tener instalado **Python** en la versión apropiada, enseguida se debe instalar **Django** siguiendo lo siguiente.

- ✓ **python setup.py install**
- ✓ tener instalado **pip**, **sudo apt-get install python\_pip**
- ✓ **pip install virtualenv**

Este proyecto deberá ubicarse dentro de un **directorio**. Es importante que una vez creando el directorio deberá generar un ambiente virtual **\$virtualenv nombreEspacioVirtual** y enseguida activarlo **\$source nombreEspacioVirtual/bin/activate**, para después instalar Django con **\$pip install Django**.

Ahora se deberán seguir los siguientes pasos:

### 1. **Django-admin.py startproject misitio**

El comando **startproject** crea un directorio de trabajo que contiene varios archivos:

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

Estos archivos son los siguientes:

**misitio/:** El directorio de trabajo externo misitio/, es solo un contenedor, es decir una carpeta que contiene nuestro proyecto. Por lo que se le puede cambiar el nombre en cualquier momento sin afectar el proyecto en sí.

**manage.py:** Una utilidad de línea de comandos que te permite interactuar con un proyecto Django de varias formas. Usa `manage.py help` para ver lo que puede hacer. No deberías editar este archivo, ya que este es creado en el directorio convenientemente para manejar el proyecto.

**misitio/misitio/:** El directorio interno misitio/ contiene el paquete Python para tu proyecto. El nombre de este paquete Python se usará para importar cualquier cosa dentro del.

**\_\_init\_\_.py:** Un archivo requerido para que Python trate el directorio misitio como un paquete o como un grupo de módulos. Es un archivo vacío y generalmente no necesitarás agregarle nada.

**settings.py:** Las opciones/configuraciones para nuestro proyecto Django.

**urls.py:** Declaración de las URLs para este proyecto de Django.

**wsgi.py:** El punto de entrada WSGI para el servidor Web, encargado de servir nuestro proyecto.

2. **python manage.py migrate** (Debes entrar primero al proyecto creado en el paso anterior)

El comando **migrate** busca la variable `INSTALLED_APPS` y crea las tablas necesarias de cada una de las aplicaciones registradas en el archivo `settings.py`, que contiene todas las aplicaciones.

3. **python manage.py runserver**

**runserver** inicia el servidor de desarrollo en el puerto 8000, escuchando sólo conexiones locales. Ahora que el servidor está corriendo, visita la dirección `http://127.0.0.1:8000/` con tu navegador Web.

4. **python manage.py startapp biblioteca**

Este comando no produce ninguna salida, pero crea un directorio llamado `biblioteca` dentro del directorio `misitio` y dentro de este crea otro directorio más, llamado `migrations`. Echemos un vistazo al contenido:

```
biblioteca/  
  __init__.py  
  admin.py  
  models.py  
  tests.py  
  views.py  
  migrations/  
    __init__.py
```

Estos archivos contendrán los modelos, las vistas, las pruebas, y las migraciones para esta aplicación.

## Bibliografía

García M., S. (2015). *La Guía definitiva de Django*. Celayita México: Django Software Corporation.