

## Guía paso a paso para crear templates en Django

### Parte 1: Crear entorno virtual e instalar Django

Asegúrate de estar ubicado en la carpeta donde quieres crear tu proyecto Django. Una vez ahí se debe crear el entorno virtual a través de los siguientes comandos utilizando la consola:

```
pip install virtualenv
virtualenv venv
```

Al crear el entorno virtual llamado 'venv' se debe activar ejecutando el archivo 'activate':

```
.\venv\Scripts\activate
```

Una vez dentro de nuestro entorno virtual se instalará Django con el siguiente comando en consola:

```
pip install django
```

Por último, verifiquemos qué versiones de Django y Python están instaladas en nuestro entorno virtual, con ello te aseguras su correcta instalación:

```
django-admin --version
python --version
```

### Parte 2: Crear proyecto Django

Crea un nuevo proyecto Django, para efectos de práctica, nombraremos al proyecto 'mysite' utilizando los comandos:

```
django-admin startproject mysite
```

### Parte 3: Crear aplicación Django

Crear una aplicación dentro del proyecto, para efectos de práctica, nombraremos a la aplicación 'myapp' utilizando los comandos:

```
python manage.py startapp myapp
```

### Parte 4: Configurar la aplicación

Agrega la aplicación 'myapp' a la configuración del proyecto. Para ello abre el archivo settings.py y encuentra la lista 'INSTALLED\_APPS', por último, agrega el nombre de la aplicación a la lista.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myapp',
]
```

## Parte 5: Crear una vista

Para crear una vista abre el archivo `views.py` en el directorio de la aplicación y crea una vista, función que procesará una solicitud y devolverá una respuesta. Como ejemplo, crearemos la vista 'index' el que va a renderizar un template llamado 'index.html'

```
myapp > views.py > ...
1  from django.shortcuts import render
2
3  # Create your views here.
4  def index(request):
5      return render(request, 'index.html')
6
```

## Paso 6: Crear un template

Crea una carpeta 'templates' dentro del directorio de la aplicación. En la carpeta templates crearás las plantillas HTML, para efectos de este ejemplo, crea el archivo 'index.html'

El archivo "index.html" contendrá el código HTML de tu página. Puedes utilizar etiquetas de plantilla de Django para incluir datos dinámicos.

Hasta ahora, deberás tener un directorio como este:

## Paso 7: Configurar los templates de Django

Para configurar el sistema de plantillas de Django, ve al archivo `settings.py` y actualiza la variable `DIRS` con la ruta de la carpeta de plantillas. Es necesario importar el módulo `os` para acceder a funcionalidades dependientes del Sistema Operativo.

```
import os
# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'myapp', 'templates', 'myapp')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

```

└─ myapp
   ├── __pycache__
   ├── migrations
   └─ templates\myapp
      └─ index.html
├─ __init__.py
├─ admin.py
├─ apps.py
├─ models.py
├─ tests.py
├─ urls.py
├─ views.py
└─ mysite
   ├── __pycache__
   ├── __init__.py
   ├── asgi.py
   ├── settings.py
   ├── urls.py
   ├── wsgi.py
   ├── venv
   ├── db.sqlite3
   ├── manage.py
   └─ requirements.txt
```

## Paso 8: Definir URLs

Abre el archivo `urls.py` en el directorio de la aplicación y define una URL para la vista que creaste anteriormente.

```
myapp > 📄 urls.py > ...
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path('', views.index)
6  ]
```

## Paso 9: Ejecutar el servidor de desarrollo

Ejecuta el servidor de desarrollo de Django usando el comando:

```
python manage.py runserver
```

Esto iniciará el servidor de desarrollo y podrás acceder a tu aplicación en <http://localhost:8000/>. ¡Eso es todo! Ahora has creado un template en Django y configurado una vista para renderizarlo. Puedes agregar más templates y vistas según tus necesidades para desarrollar tu aplicación web.