

# Iniciar y configurar un nuevo proyecto de Django

agosto 20, 2023 by [Recursos Python](#) [1 comentario](#)

Desarrollar un nuevo sitio web en Django requiere de una serie de pasos iniciales que pueden resultar algo tediosos y confusos cuando nos estamos introduciendo en el *framework*: crear el proyecto, crear la aplicación, configurar las URL, etc. He aquí una guía rápida y fácil de seguir que podremos usar a modo de referencia cada vez que necesitemos iniciar un nuevo proyecto.

1. Crear una nueva carpeta y ubicarse allí con la terminal (usuarios de Windows pueden ver [cómo abrir la terminal en una carpeta](#)).

2. [Crear un entorno virtual](#) e instalar Django:

- Linux/macOS

- Windows

1. `> py -m venv env`
2. `> env\Scripts\activate`
3. `(env) > pip install django`

Este paso puede omitirse si se desea usar el intérprete de Python del sistema y Django ya está instalado.

3. Crear el nuevo proyecto de Django (en este ejemplo, con el nombre `myproject`; cámbiese a gusto) usando el comando `django-admin` e ingresar a la carpeta del proyecto:

- Linux/macOS

- Windows

1. `(env) > django-admin startproject myproject`
2. `(env) > cd myproject`

4. Crear una nueva aplicación de Django (con el nombre `myapp`) dentro del proyecto:

- Linux/macOS

- Windows

1. `(env) > python manage.py startapp myapp`

Recuérdese que Django distingue entre proyectos y aplicaciones. Una aplicación es un paquete de Python con una estructura determinada (modelos, vistas, plantillas, etc.). Un proyecto es —además de un paquete de Python— un conjunto de aplicaciones con una configuración común (por ejemplo, todas comparten la(s) misma(s) base(s) de datos). Un proyecto de Django debe tener al menos una aplicación configurada.

5. Abrir el archivo `myproject/settings.py` con nuestro editor de código favorito e instalar la aplicación recién creada en el proyecto:

```
1. INSTALLED_APPS = [  
2. 'django.contrib.admin',  
3. 'django.contrib.auth',  
4. 'django.contrib.contenttypes',  
5. 'django.contrib.sessions',  
6. 'django.contrib.messages',  
7. 'django.contrib.staticfiles',  
8. "myapp.apps.MyappConfig", # <-- Agregar este elemento.  
9. ]
```

6. Crear alguna vista de prueba en la nueva aplicación, escribiendo lo siguiente dentro de `myapp/views.py`:

```
1. from django.http import HttpRequest, HttpResponse  
2.  
3.  
4. def index(request: HttpRequest) -> HttpResponse:  
5. return HttpResponse("¡Hola, mundo!")
```

6. Crear el archivo `myapp/urls.py` y asociar la vista recién creada con alguna ruta:

```
1. from django.urls import path  
2.  
3. from . import views  
4.  
5.  
6. urlpatterns = [  
7. # Usar una cadena vacía para que sea la URL de inicio.  
8. path("", views.index, name="index")  
9. ]
```

7. En el archivo `myproject/urls.py` (¡que no es lo mismo que `myapp/urls.py`!) probablemente nos encontremos con algo como:

```
1. from django.contrib import admin  
2. from django.urls import path  
3.  
4. urlpatterns = [  
5. path('admin/', admin.site.urls),  
6. ]
```

Agregar las URL definidas por nuestra aplicación:

```
1. from django.contrib import admin
```

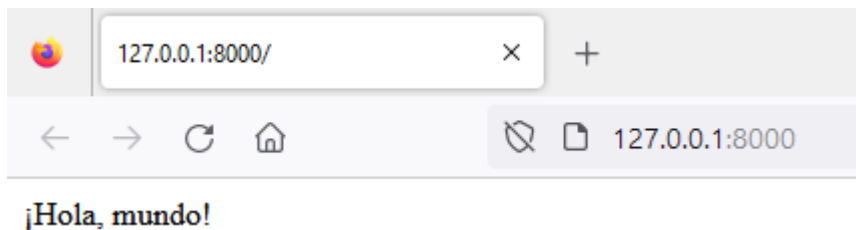
```
2. from django.urls import path, include
3.
4. urlpatterns = [
5.     path('admin/', admin.site.urls),
6.     path("", include("myapp.urls"))
7. ]
```

7. Iniciar el servidor de desarrollo de Django:

- Linux/macOS
- Windows

```
1. (env) > python manage.py runserver
```

8. Visitar <http://127.0.0.1:8000/> y ver el resultado de nuestra vista:



¡Eso es todo! Ya tenemos un proyecto de Django completamente configurado para empezar a desarrollar.