

## Lección 2: Actividad para portafolios nro. 2

### “Desarrollo de un blog con Django”

Nicolás Jofré Andrade - Desarrollo Full-stack Python  
04-2024

**Objetivo:** Crear un proyecto de blog utilizando Django, abordando temas específicos solicitados.

#### 1. Instalación y configuración:

##### a. Crea un entorno virtual llamado 'miBlogEnv'

```
TERMINAL

> python -m venv miBlogEnv
> dir

Directory: C:\Users\njofre\Desktop\dev\portafolio-py-mod6\actividad2

Mode                LastWriteTime         Length Name
----                -
d-----          22-04-2024    21:02             miBlogEnv
-a---          16-04-2024    21:46          3236 .gitignore
```

##### b. Activa el entorno virtual

```
> .\miBlogEnv\scripts\activate
(miBlogEnv) njofre ~\Desktop\dev\portafolio-py-mod6\actividad2 main ≡*?1
> 
```

##### c. Instalar Django con pip

```
> pip install django

> pip list
Package Version
-----
asgiref 3.8.1
Django  5.0.4
pip      23.2.1
sqlparse 0.5.0
tzdata  2024.1

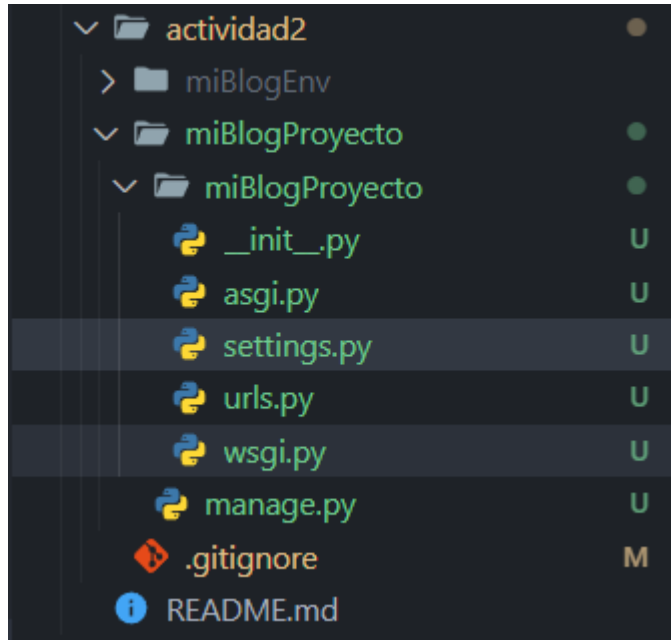
[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
(miBlogEnv) njofre ~\Desktop\dev\portafolio-py-mod6\actividad2 main ≡*~1 740ms
```

#### 2. Creación del proyecto:

- a. Crea un proyecto llamado 'miBlogProyecto'

```
> django-admin startproject miBlogProyecto
● (miBlogEnv) njofre ~\Desktop\dev\portafolio-py-mod6\actividad2 main ≡*?1 ~1 592ms
○ > |
```

- b. Examina la estructura de carpetas generada y explora el archivo 'settings.py'



3. Gestión de comandos con 'manage.py':

- a. Ejecuta 'runserver' y verifica que el servidor se inicie correctamente en localhost

```
> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly without
Run 'python manage.py migrate' to apply them.
April 22, 2024 - 21:08:55
Django version 5.0.4, using settings 'miBlogProyecto.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[22/Apr/2024 21:08:57] "GET / HTTP/1.1" 200 10629
Not Found: /favicon.ico
[22/Apr/2024 21:08:57] "GET /favicon.ico HTTP/1.1" 404 2118
```

127.0.0.1:8000

Servicio Tools Tailwind

django

[View release notes](#) for Django 5.0



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

## b. Utiliza el comando para crear un superusuario

```
> python manage.py createsuperuser
Username (leave blank to use 'njofre'): njofre
Email address: n@n.cl
Password:
Password (again):
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
(miBlogEnv) njofre ~\Desktop\dev\portafolio-py-mod6\actividad2\miBlogProyecto main ≡*?1 ~1 13.735s
>
```

#### 4. Creación de una aplicación de blog:

- a. Crea una nueva aplicación llamada 'blog' con 'manage.py startapp'

```
> python manage.py startapp blog
• > dir
•
Directory: C:\Users\njofre\Desktop\dev\portafolio-py-mod6\actividad2\miBlogProyecto

Mode                LastWriteTime         Length Name
----                -
d-----          22-04-2024    21:12           blog
d-----          22-04-2024    21:08       miBlogProyecto
-a----          22-04-2024    21:11       131072 db.sqlite3
-a----          22-04-2024    21:07         692 manage.py
```

- b. Define un modelo 'POST' en 'models.py' con campos como título, contenido y fecha de publicación

```
models.py U X
1  from django.db import models
2  from django.utils import timezone
3
4
5  class Post(models.Model):
6      TituloPost = models.CharField(max_length=200)
7      Contenido = models.TextField()
8      FechaPost = models.DateTimeField(default=timezone.now)
9
10     def __str__(self):
11         return self.TituloPost
```

- c. Realiza las migraciones y aplica los cambios a la base de datos

```
> python manage.py makemigrations
• No changes detected

> python manage.py migrate
• Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
  Running migrations:
    No migrations to apply.
```

#### 5. Configuración del proyecto:

- a. Configura la aplicación 'blog' en 'INSTALLED\_APPS' en 'settings.py'

```
settings.py U X
25 # SECURITY WARNING: don't run with
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     "django.contrib.admin",
35     "django.contrib.auth",
36     "django.contrib.contenttypes",
37     "django.contrib.sessions",
38     "django.contrib.messages",
39     "django.contrib.staticfiles",
40     "blog",
41 ]
```

- b. Configura el archivo 'urls.py' del proyecto para incluir las URLs de la aplicación 'blog'

```
urls.py U X
1 # Urls proyecto
2 from django.contrib import admin
3 from django.urls import path, include
4
5 urlpatterns = [
6     path("admin/", admin.site.urls),
7     path("", include("blogapp.urls")),
8 ]
9
```

## 6. Vistas y plantillas:

- a. Crea una vista que muestre la lista de publicaciones en 'views.py'

```
views.py U X
1  from django.shortcuts import render
2  from .models import Post
3
4
5  def post_list(request):
6      posts = Post.objects.all()
7      return render(request, "blogapp/post_list.html", {"posts": posts})
8
```

- b. Crea una plantilla HTML en la carpeta 'templates/blog/' para mostrar la lista de publicaciones.

```
post_list.html U X
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Lista de Posts</title>
5      <script src="https://cdn.tailwindcss.com"></script>
6  </head>
7  <body>
8      <section>
9          <div class="mt-5 mx-5 p-10 text-center">
10             <h1 class="ml-6 pt-1 text-base text-gray-800 leading-normal">Lista de Posts</h1>
11             {% for post in posts %}
12                 <div class="ml-6 pt-1">
13                     <h2 class="text-4xl">{{ post.TituloPost }}</h2>
14                     <p class="text-3xl">{{ post.Contenido }}</p>
15                     <p class="text-1xl">Publicado el: {{ post.FechaPost }}</p>
16                 </div>
17                 <hr>
18             {% endfor %}
19         </div>
20     </section>
21 </body>
22 </html>
```

- c. Configura las URLs para que apunten a la vista recién creada

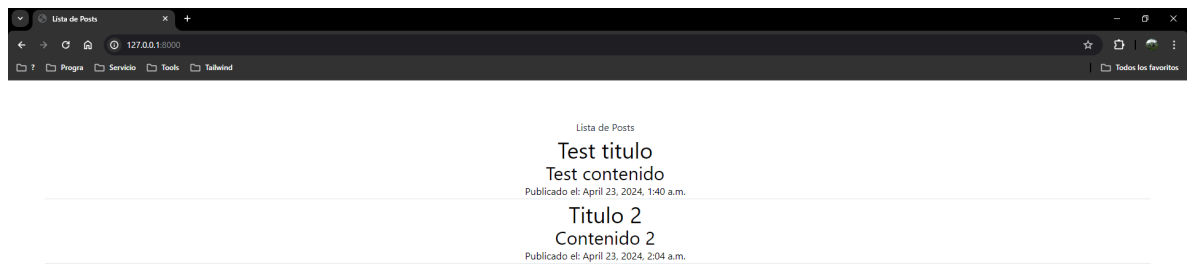
```
urls.py U X
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path("", views.post_list, name="post_list"),
6  ]
7
```

## 7. Creación de detalles de publicación:

### a. Agrega una vista para mostrar detalles de una publicación individual

```
views.py U X
1  from django.shortcuts import render
2  from .models import Post
3
4
5  def post_list(request):
6      posts = Post.objects.all()
7      return render(request, "blogapp/post_list.html", {"posts": posts})
8
```

### b. Crea una plantilla HTML para mostrar detalles de una publicación



8. Personalización del administrador de django:
- a. Personaliza el administrador de Django para que sea más amigable con los posts.

**Test título**


**TítuloPost:**


Test título

**Contenido:**

Test contenido

**FechaPost:**

**Date:** 2024-04-23 Today | 

**Time:** 01:40:38 Now | 

Note: You are 4 hours behind server time.



- 9. Despliegue en producción (opcional):**
  - a. Investiga y documenta los pasos básicos para desplegar tu aplicación django**