

# Lección 3: Actividad para portafolios nro. 3

## “Despliegue de un blog con contenido dinámico en Django”

Nicolás Jofré Andrade - Desarrollo Full-stack Python

04-2024

**Objetivo:** Desarrollar un blog simple utilizando Django, implementando despliegue de páginas con contenido estático y dinámico.

### 1. Instalación de Bootstrap y configuración estática:

#### a. Instala Bootstrap usando CDN en tu plantilla HTML principal ('base.html')

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>{% block title %}Tu título aquí{% endblock %}</title>
5     <!-- Bootstrap CSS -->
6     <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
7 </head>
8 <body>
```

#### b. Agrega un Navbar, un Jumbotron y un Footer de Bootstrap en tu página principal.

```
<!-- Navbar -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container">
        <a class="navbar-brand" href="#">Historias de animales</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="{% url 'post_list' %}">Home</a>
                </li>
            </ul>
        </div>
    </div>
</nav>
```

```

<!-- Jumbotron -->
<div class="jumbotron container py-4">
  <div class="container h-100 p-5 text-bg-dark rounded-3">
    <h1 class="display-5 fw-bold">Historias de ficción de perros y gatos</h1>
    <p class="lead">Este es un blog de historias</p>
  </div>
</div>

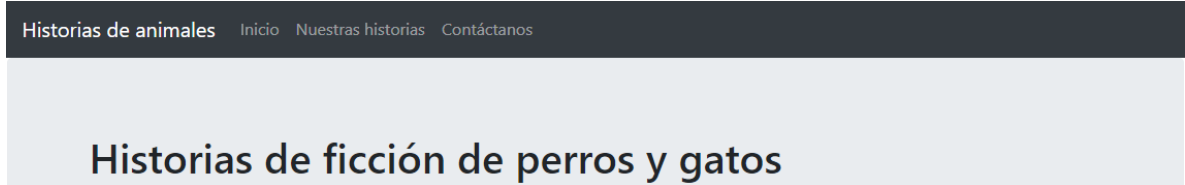
40 <!-- Footer -->
41 <footer class="footer bg-dark text-light">
42   <div class="container text-center">
43     <span class="text-muted">© 2024 Portafolio 3 - njofredev - Django</span>
44   </div>
45 </footer>
46
47 <!-- Bootstrap y jQuery JavaScript -->
48 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
49 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>
50 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
51 </body>
52 </html>
53

```

2. Creación de páginas responsivas y navegación
  - a. Asegúrate de que tu sitio sea responsivo utilizando las clases de Bootstrap.



- b. Implementa un menú de navegación con links a la página principal y al listado de publicaciones.



3. Modelo MVC en Django y Renderización dinámica:

- a. Define un modelo 'Post' en 'models.py'

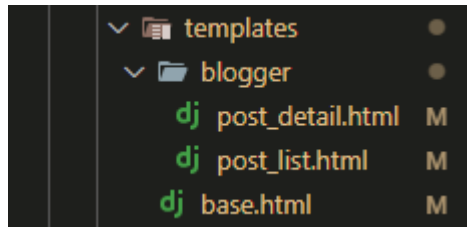
```
models.py M X
1  from django.db import models
2  from django.utils import timezone
3
4  class Post(models.Model):
5      title = models.CharField(max_length=200)
6      content = models.TextField()
7      date_posted = models.DateTimeField(default=timezone.now)
8
9      def __str__(self):
10    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

- b. Crea vistas en 'views.py' para mostrar la lista de publicaciones y detalles de una publicación individual.

```
views.py M X
1  from django.shortcuts import render
2  from .models import Post
3
4  def post_list(request):
5      posts = Post.objects.all()
6      return render(request, 'blogger/post_list.html', {'posts': posts})
7
8  def post_detail(request, pk):
9      post = Post.objects.get(pk=pk)
10     return render(request, 'blogger/post_detail.html', {'post': post})
11
```

4. Templates en Django y variables:

- a. Crea plantillas ('post\_list.html' y 'post\_detail.html') para mostrar la lista de publicaciones y los detalles de una publicación individual.



- b. Utiliza variables en las plantillas para mostrar dinámicamente el contenido.

```
1  {% extends 'base.html' %}
2
3  {% block title %}Lista de Posts{% endblock %}
4
5  {% block content %}
6      <h2>Lista de Posts</h2>
7      <div class="row">
8          {% for post in posts %}
9              <div class="col-md-6">
10                 <div class="card mb-3">
11                     <div class="card-body">
12                         <h5 class="card-title">{{ post.title }}</h5>
13                         <p class="card-text">{{ post.content }}</p>
14                         <a href="{% url 'post_detail' post.id %}" class="btn btn-primary">Ver Detalles</a>
15                     </div>
16                 </div>
17             </div>
18         {% endfor %}
19     </div>
20 {% endblock %}
21
```

## 5. Herencia de plantillas:

- a. Implementa una plantilla base ('base.html') con bloques para el contenido dinámico.

```
40  <!-- Contenido dinámico -->
41  <div class="container">
42      {% block content %}
43      {% endblock %}
44  </div>
45
46  <!-- Footer -->
47  <footer class="footer bg-dark text-light">
48      <div class="container text-center">
49          <span class="text-muted">© 2024 Portafolio 3 - njofredev - Django</span>
50      </div>
51  </footer>
52
53  <!-- Bootstrap y jQuery JavaScript -->
54  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
55  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>
56  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
57  </body>
58  </html>
59
```

b. Extiende esta plantilla en las plantillas de listado y detalles.

```
post_detail.html x
1 {% extends 'base.html' %}
2
3 {% block title %}{{ post.title }}{% endblock %}
4
5 {% block content %}
6     <div class="card">
7         <div class="card-body">
8             <h5 class="card-title">{{ post.title }}</h5>
9             <p class="card-text">{{ post.content }}</p>
10            <p class="card-text"><small class="text-muted">Ingresado el: {{ post.date_posted }}</small></p>
11        </div>
12    </div>
13 {% endblock %}
14

post_list.html x
1 {% extends 'base.html' %}
2
3 {% block title %}Lista de Posts{% endblock %}
4
5 {% block content %}
6     <h2>Lista de Posts</h2>
7     <div class="row">
8         {% for post in posts %}
9             <div class="col-md-6">
10                <div class="card mb-3">
11                    <div class="card-body">
12                        <h5 class="card-title">{{ post.title }}</h5>
13                        <p class="card-text">{{ post.content }}</p>
14                        <a href="{% url 'post_detail' post.id %}" class="btn btn-primary">Ver Detalles</a>
15                    </div>
16                </div>
17            </div>
18        {% endfor %}
19    </div>
20 {% endblock %}
21
```

6. Manejo de errores y generadores en plantillas:

a. Implementa manejo de errores en las plantillas

- No enseñado -

b. Utiliza generadores y la etiqueta 'for' para iterar sobre elementos en las plantillas.

```
<!-- Contenido de Posts dinámicos -->
<div class="container">
    {% block content %}
        <h2>Últimos posts</h2>
        <ul>
            {% for post in Posts %}
                <li>
                    <a href="{% url 'post_detail' %}">{{ post.title }}</a> - {{ post.created_date }}
                </li>
            {% endfor %}
        </ul>
    {% endblock %}
</div>
```

7. Creación de modelos y despliegue:

- a. Crea migraciones y aplica los cambios de para incorporar el modelo 'Post'

```
TERMINAL

> python manage.py makemigrations
● No changes detected
> python manage.py migrate
● Operations to perform:
  Apply all migrations: admin, auth, blogger, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
```

- b. Despliega tu aplicación en un entorno de desarrollo.

