

Investigación: Comparación entre patrón arquitectónico MVC y MTV

Django sigue el patrón MVC tan al pie de la letra que puede ser llamado un framework **MVC**. Someramente, la M, V y C se separan en Django de la siguiente manera:

- M, la porción de acceso a la base de datos, es manejada por la capa de la base de datos de Django.
- V, la porción que selecciona qué datos mostrar y cómo mostrarlos, es manejada por la vista y las plantillas.
- C, la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el framework mismo siguiendo tu URLconf y llamando a la función apropiada de Python para la URL obtenida.

Debido a que la “C” es manejada por el mismo framework y la parte más emocionante se produce en los modelos, las plantillas y las vistas, Django es conocido como un Framework MTV. En el patrón de diseño **MTV**.

- M significa “Model” (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
- T significa “Template” (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web o otro tipo de documento.
- V significa “View” (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: se puede pensar en esto como un puente entre los modelos y las plantillas.

En la interpretación de Django de MVC, la “vista” describe los datos que son presentados al usuario; no necesariamente el cómo se mostrarán, pero sí cuáles datos son presentados.

Ninguna de las interpretaciones es más “correcta” que otras. Lo importante es entender los conceptos subyacentes.

models.py

```
from django.db import models
"""Las tablas de la base de datos"""

class Libro(models.Model):
    nombre = models.CharField(max_length=50)
    fecha = models.DateField()
```

views.py

```
from django.shortcuts import render_to_response
from models import Libro

def ultimos_libros(request):

    """La parte lógica"""
    lista_libros = Libro.objects.order_by('-fecha')[:10]
    return render_to_response('ultimos-libros.html', {'lista_libros': lista_libros})
```

urls.py

```
from django.conf.urls import url
import views

# La configuración de la URL
urlpatterns = [
    url(r'^ultimos_libros/$', views.ultimos_libros),
]
```

ultimos_libros.html

```
{# La plantilla #}
<html><head><title>Libros</title></head>
<body>
    <h1>Libros</h1>
    <ul>
        {% for libro in lista_libros %}
            <li>{{ libro.nombre }}</li>
        {% endfor %}
    </ul>
</body></html>
```

Bibliografía

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