

Django

The web framework for perfectionists with deadlines.



Definición

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–template. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt. En junio de 2008 fue anunciado que la recién formada Django Software Foundation se haría cargo de Django en el futuro.

Página web del proyecto: <https://www.djangoproject.com/>

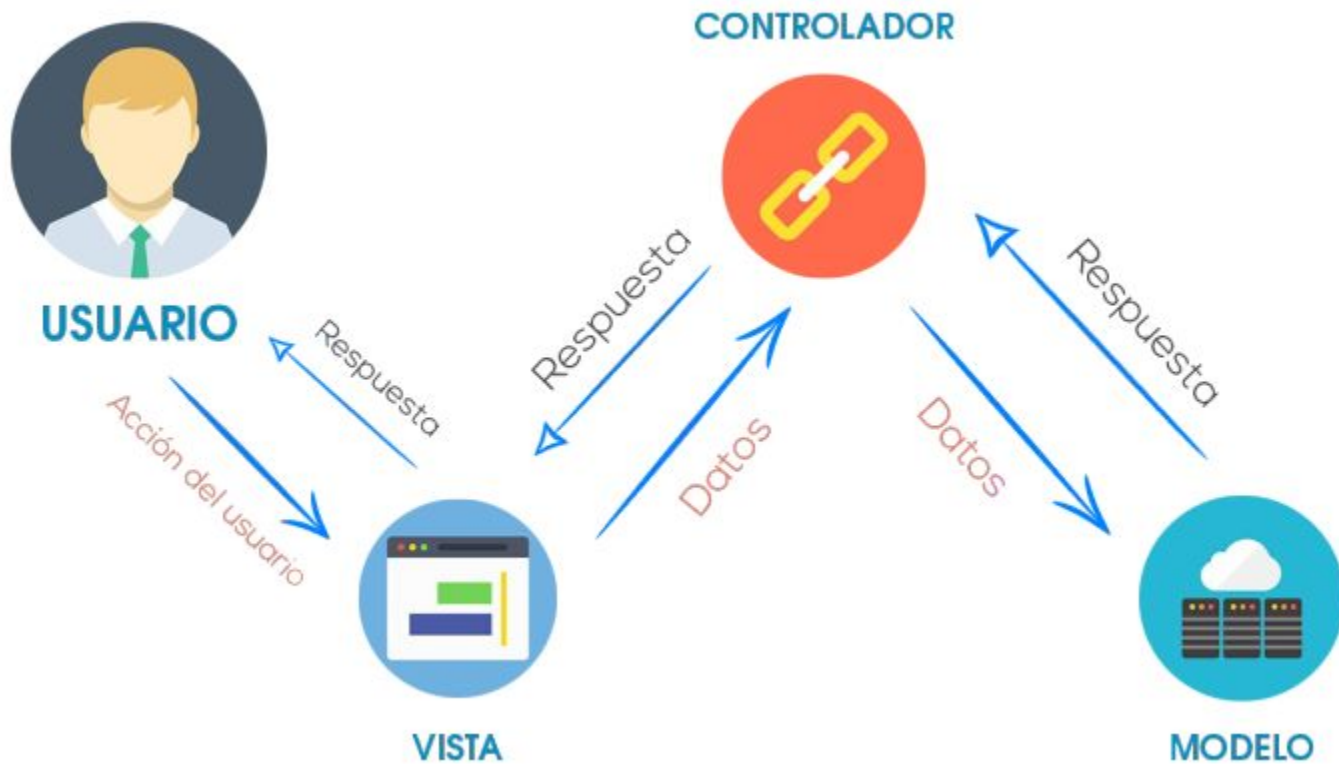
Qué es un framework?

Es un entorno de trabajo, un ecosistema que nos facilita crear algo, en este caso un sitio web, sin mayores esfuerzos.

WEB2PY



Patrón: Modelo – Vista – Controlador (MVC)



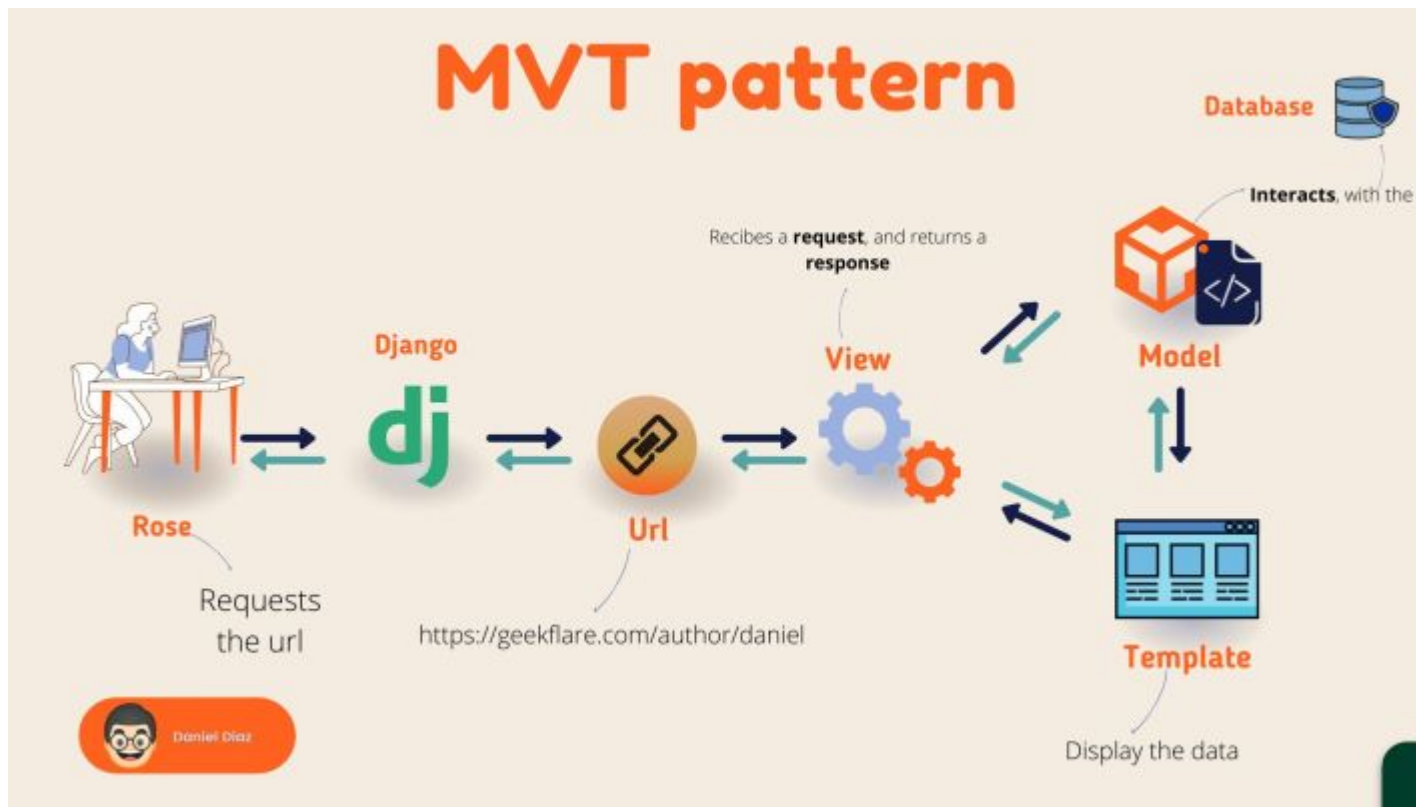
Patrón: Modelo – Vista – Controlador (MVC)

Modelo: Representa la lógica de negocio y los datos de la aplicación. Se encarga de acceder a la base de datos y procesar la información.

Vista: Se encarga de la presentación de los datos al usuario. Es la interfaz que el usuario ve y con la que interactúa.

Controlador: Actúa como intermediario entre el modelo y la vista. Recibe las entradas del usuario, procesa las solicitudes y actualiza la vista.

Patrón: Modelo - Vista - Template



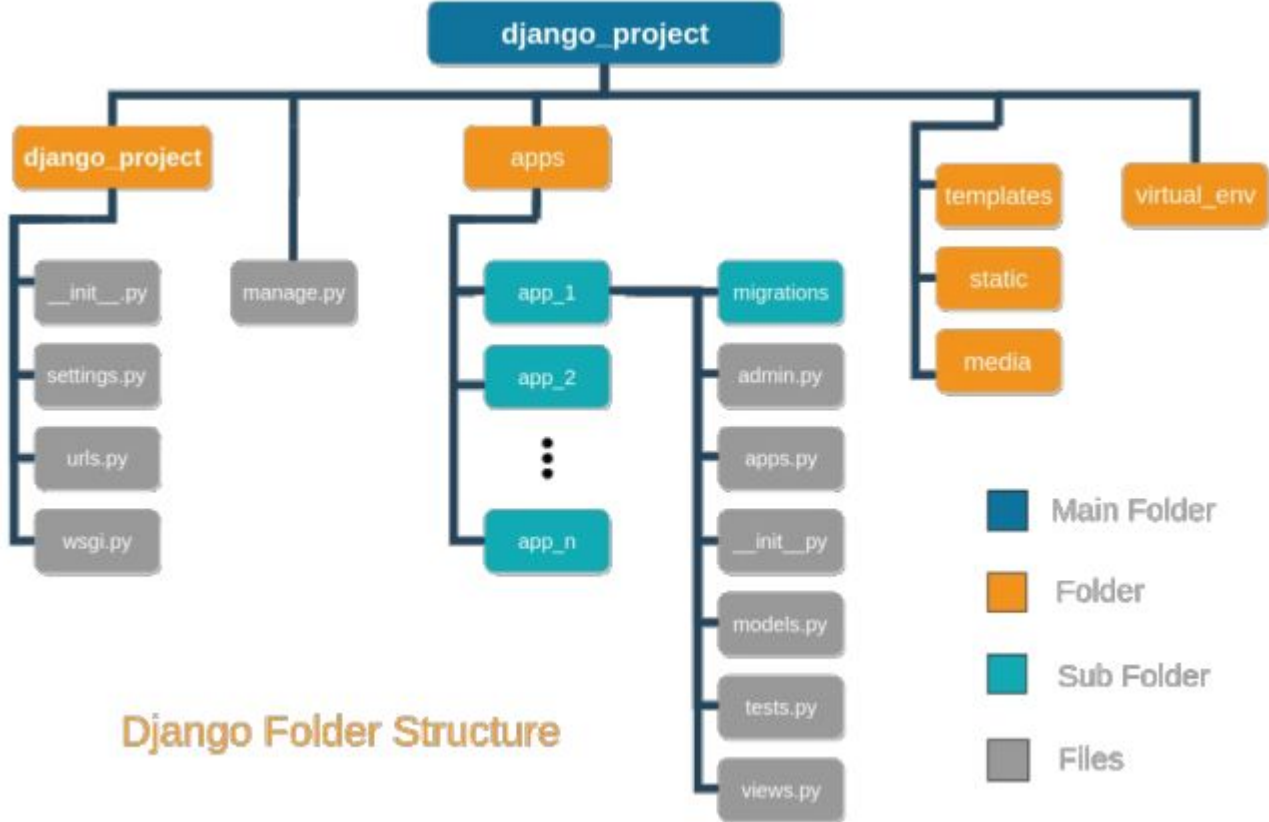
Patrón: Modelo – Vista – Template

Modelo: Al igual que en MVC, maneja la lógica de negocio y los datos.

Vista: En MVT, se refiere a la lógica de presentación y la gestión de las solicitudes. Aquí, la vista se encarga de definir qué datos se envían a la plantilla, pero no incluye la presentación en sí.

Template: Es la parte responsable de la presentación. Contiene el HTML y puede incluir marcadores de posición para los datos dinámicos que se envían desde la vista.

Scaffolding



Creando un proyecto I

Ejecutar **django-admin startproject <nombre_del_proyecto>**

```
$ django-admin startproject sitio
```

Se crea

```
sitio/  
  manage.py  
  sitio/  
    __init__.py  
    settings.py  
    urls.py  
    asgi.py  
    wsgi.py
```

Creando un proyecto II

- **manage.py**: Una utilidad de la línea de comandos que le permite interactuar con este proyecto Django de diferentes formas. Puede leer todos los detalles sobre **manage.py** en [django-admin and manage.py](#).
- En interior del directorio **sitio/** es el propio paquete de Python para su proyecto. Su nombre es el nombre del paquete de Python que usted tendrá que utilizar para importar todo dentro de este (por ejemplo, **sitio.urls**).
- **sitio/__init__.py**: Un archivo vacío que le indica a Python que este directorio debería ser considerado como un paquete Python. Si usted es un principiante lea [más sobre los paquetes](#) en la documentación oficial de Python.
- **sitio/settings.py**: Ajustes/configuración para este proyecto Django. [Django settings](#) le indicará todo sobre cómo funciona la configuración.
- **sitio/urls.py**: Las declaraciones URL para este proyecto Django; una «tabla de contenidos» de su sitio basado en Django. Puede leer más sobre las URLs en [URL dispatcher](#).
- **sitio/asgi.py**: An entry-point for ASGI-compatible web servers to serve your project. See [How to deploy with ASGI](#) for more details.
- **sitio/wsgi.py**: Un punto de entrada para que los servidores web compatibles con WSGI puedan servir su proyecto. Consulte `:doc:~/howto/deployment/wsgi/index`` para más detalles.

Instalación && Verificación

```
python3 -m venv django_env
```

```
source django_env/bin/activate
```

```
pip install django
```

```
python
```

```
Python 3.10.12 (main, Sep 11 2024, 15:47:36) [GCC 11.4.0] on linux
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import django
```

```
>>> django.get_version()
```

```
'5.1.1'
```

Servidor de desarrollo

```
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 until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.

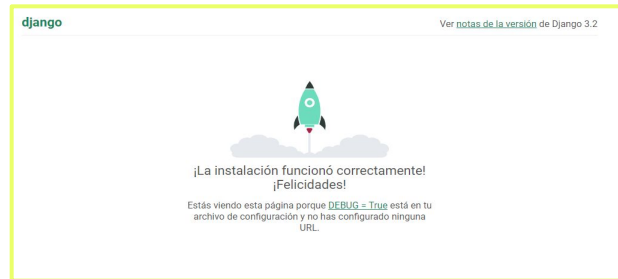
September 23, 2024 - 00:10:10

Django version 5.1.1, using settings 'sitio.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CONTROL-C.

http://localhost:8000/



Primera vista (views)

```
sitio/  
  manage.py  
  sitio/  
    __init__.py  
    settings.py  
    urls.py  
    asgi.py  
    wsgi.py  
    views.py
```

En el archivo views.py

```
from django.http import HttpResponse
```



```
def saludo(request):  
    return HttpResponse('IADES -- Programacion III')
```

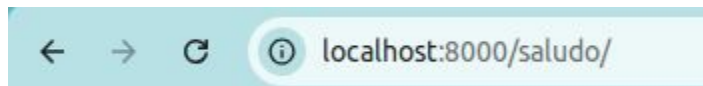
Primera vista (urls)

En el archivo urls.py

```
from django.contrib import admin
from django.urls import path

# Importamos nuestra vista
from .views import saludo

urlpatterns = [
    path("admin/", admin.site.urls),
    path("saludo/", saludo),
]
```



Agreguemos otra vista (views - urls)

```
def saludo dos(request):  
    return HttpResponse("<p>Es nuestra segunda vista<p>")
```

Podemos poner comandos HTML 🕶️

```
from django.contrib import admin  
from django.urls import path
```

```
from .views import saludo, saludo dos
```

```
urlpatterns = [  
    path("admin/", admin.site.urls),  
    path('saludo/', saludo),  
    path('path-url/', saludo dos),  
]
```

Retornando valores dinámicos

Muchas veces queremos mostrar en una vista el resultado de algún proceso interno realizado en Python, esto es básicamente enviar parámetros por medio de la vista.

Hacer una vista que retorne la fecha y la hora

Mapeo urls – rutas

```
from django.urls import path

from . import views

urlpatterns = [

    path('articles/2003/', views.special_case_2003),
    path('articles/<int:year>', views.year_archive),
    path('articles/<int:year>/<str:name>', views.year_name),

]
```

Son archivos que nos permiten separar la vista de la estética, es decir guardar en un archivo separado, para así enviar por la HttpResponse.

Entonces, ¿podríamos decir que así se crea un template? 😊

¡Exacto! 🙌

Recordemos que Django se basaba en Modelo, Vista, Template.