

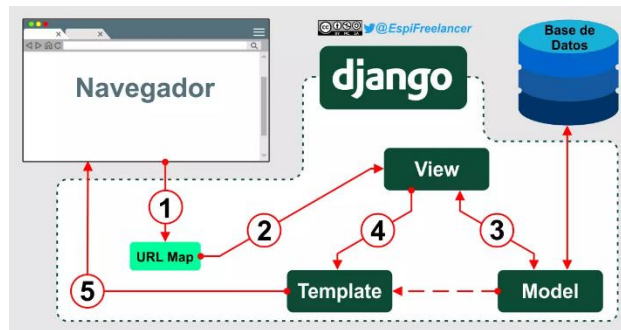
## 2- Django (MTV)

### Introducción

#### Model Template Vista (MTV)

En Django, el controlador sigue estando presente, nada más que de una manera intrínseca, ya que todo el framework Django es el controlador.

- Modelo: Maneja todo lo relacionado con la información, esto incluye como acceder a esta, la validación, relación entre los datos y su comportamiento.
- Vista: Es un enlace entre el modelo y el template. Decide qué información sera mostrada y por cual template.
- Template: Decide como sera mostrada la información.



Descargar el framework de Django:

1- Ejecuta e siguiente comando: **pip install Django**

```
C:\Users\viane>pip install Django
Collecting Django
  Downloading Django-4.2.2-py3-none-any.whl (8.0 MB)
    ----- 8.0/8.0 MB 153.0 kB/s eta 0:00:00
Collecting asgiref<4,>=3.6.0 (from Django)
  Downloading asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting sqlparse>=0.3.1 (from Django)
  Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)
Collecting tzdata (from Django)
  Using cached tzdata-2023.3-py2.py3-none-any.whl (341 kB)
Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-4.2.2 asgiref-3.7.2 sqlparse-0.4.4 tzdata-2023.3
```

Este mismo comando te permite actualizar la versión de Django. Por ejemplo, si tuvieras una versión anterior a esta, ejecuta:

**pip install Django==4.2.2**

Para comprobar que django se instaló exitosamente o si quieres conocer la versión que estás usando ejecuta los siguientes comandos:

```
C:\Users\viane>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59)
Type "help", "copyright", "credits" or "license" for more in
>>> import django
>>> django.VERSION
(4, 2, 2, 'final', 0)
>>>
```

- 2- Para crear un nuevo proyecto, primero ubica la ruta donde quieres crear el proyecto y ahí ejecuta el comando:

**django-admin startproject nombreProyecto**






```
C:\>django-admin startproject miapp
```

Y puedes comprobar que se creó el proyecto ubicando la carpeta en la ubicación correspondiente.

Disco local (C:)	
Nombre	Tipo
miapp	Carpeta de archivos
Disco local (C:) > miapp	
Nombre	Tipo
miapp	Carpeta de archivos
manage.py	Archivo de origen ...

**Nota:** manage.py es un script que ayuda con la administración del sitio. Con él podremos iniciar un servidor web en nuestro ordenador sin necesidad de instalar nada más, entre otras cosas.

Y en la carpeta mi app encontrarás:

Disco local (C:) > miapp > miapp	
Nombre	Tipo
 <code>_init_.py</code>	Archivo de origen ...
 <code>asgi.py</code>	Archivo de origen ...
 <code>settings.py</code>	Archivo de origen ...
 <code>urls.py</code>	Archivo de origen ...
 <code>wsgi.py</code>	Archivo de origen ...

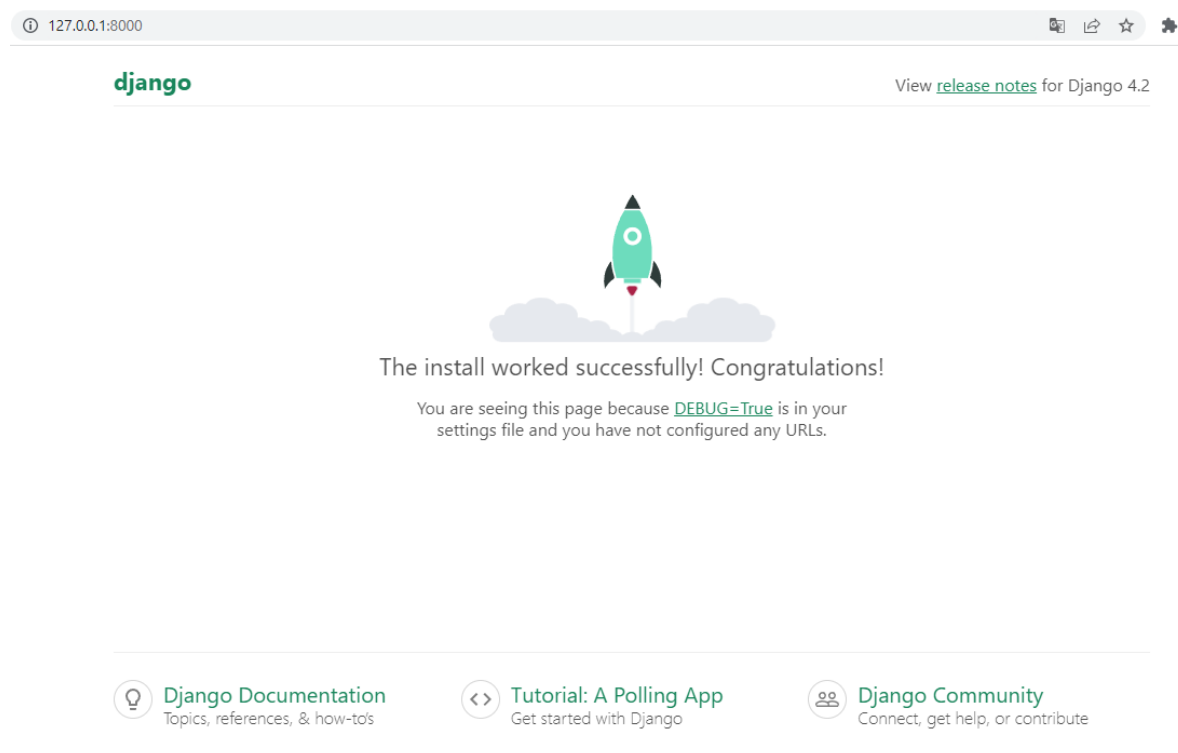
- `settings.py` contiene todos los ajustes del sitio. Es donde registramos todas las aplicaciones que creamos, la localización de nuestros ficheros estáticos, los detalles de configuración de la base de datos, etc.
  - `urls.py` define los mapeos url-vistas. A pesar de que éste podría contener todo el código del mapeo url, es más común delegar algo del mapeo a las propias aplicaciones, como verás más tarde.
  - `wsgi.py` se usa para ayudar a la aplicación Django a comunicarse con el servidor web. Puedes tratarlo como código base que puedes utilizar de plantilla.
  - El script `manage.py` se usa para crear aplicaciones, trabajar con bases de datos y empezar el desarrollo del servidor web.
- 3- Para ejecutar el servidor y poder visualizar tu proyecto en el navegador, accede a la carpeta del proyecto y ejecuta:

**`python manage.py runserver`**

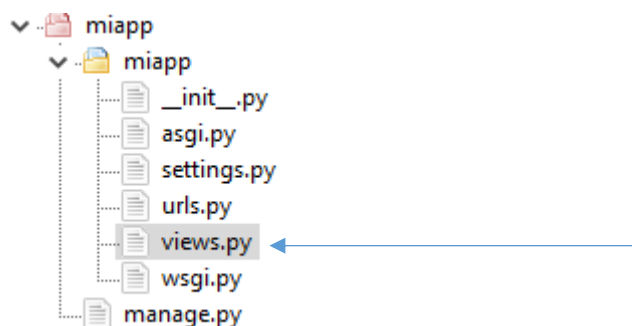
```
Django version 4.2.2, using settings 'miapp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Te proporcionará la ruta donde está corriendo el servidor

Abre esa ruta en el navegador y podrás visualizar lo siguiente:



- 4- Vamos a crear una primera vista, para ello crea un nuevo archivo llamado `views.py`



Django usa objetos de solicitud y respuesta para pasar el estado a través del sistema.

Cuando se solicita una página, Django crea un objeto `HttpRequest` que contiene metadatos sobre la solicitud. Luego, Django carga la vista apropiada, pasando `HttpRequest` como primer argumento a la función de vista. Cada vista es responsable de devolver un objeto `HttpResponse`.

Para usar ese objeto debemos importarlo.

En el archivo views.py agrega lo siguiente:

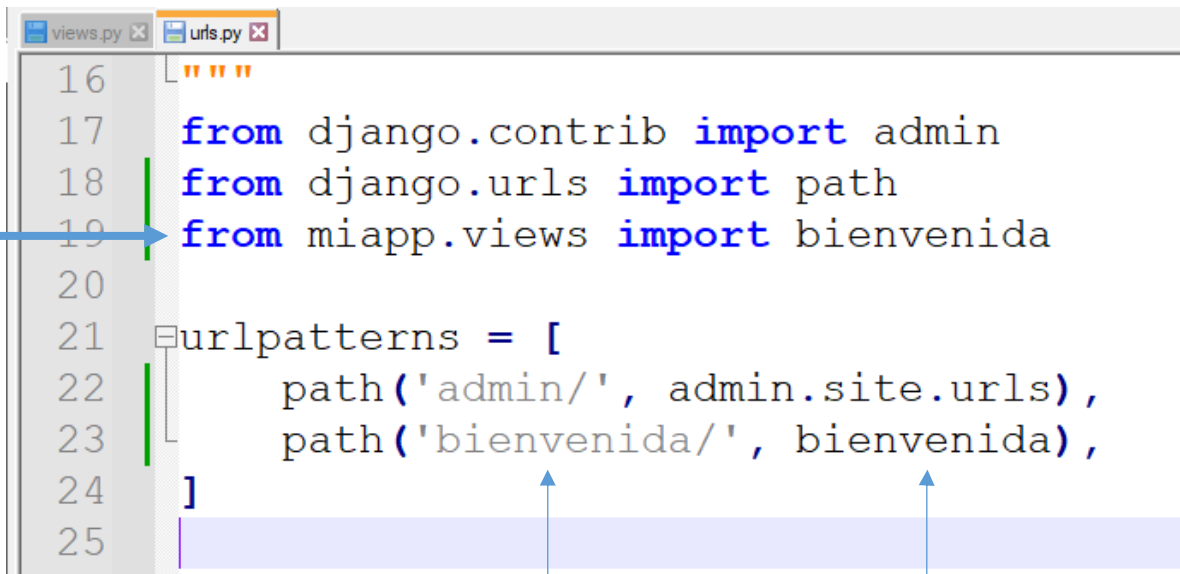
```
# Request: Para realizar peticiones
# HttpResponse: Para enviar la respuesta usando HTTP
from django.http import HttpResponse

def bienvenida(request):
    return HttpResponse("Bienvenido al curso de Django")
```

Vamos a crear una función sencilla que solo mostrará un letrero que diga "Bienvenido al curso de Django"

Para que se puedan mostrar las vistas, debemos configurar el archivo urls.py (es como un archivo de rutas)

Se debe importar la vista creada




```
16 """
17 from django.contrib import admin
18 from django.urls import path
19 from miapp.views import bienvenida
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('bienvenida/', bienvenida),
24 ]
25
```

Está sería la ruta (lo que va en la URL)

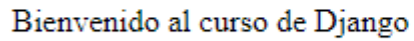
Esta sería la vista que quiero mostrar

5- Hacer la prueba en el navegador:

Colocamos la nueva URL



← → ↻ ⓘ 127.0.0.1:8000/bienvenida/



Bienvenido al curso de Django

Se muestra correctamente lo que tenemos en la vista.

## PASO DE PARÁMETROS POR URL

6- Crear una nueva vista en el archivo views.py

```
"""Esta vista recibe una variable, evalúa y luego envía respuesta"""
def categoriaEdad(request, edad):
    if edad < 18:
        categoria = "Menor de edad"
    else:
        categoria = "Mayor de edad"
    resultado = "<h1>De acuerdo a su edad, usted es: %s</h1>"
    % categoria
    return HttpResponse(resultado)
```

NOTA: Dentro de la cadena usamos %s para declarar "marcadores de posición". Después de la cadena ponemos % y después una tupla que contiene los valores que serán puestos en los marcadores de posición. Si solo tienes un marcador de posición entonces puedes omitir la tupla — e.j. 'Mi valor: %s' % variable.

Y agregamos la url



```
16  """
17  from django.contrib import admin
18  from django.urls import path
19  from miapp.views import bienvenida, categoriaEdad
20
21  urlpatterns = [
22      path('admin/', admin.site.urls),
23      path('bienvenida/', bienvenida),
24      path('edad/<int:edad>', categoriaEdad),
25  ]
```

Realizamos la prueba, si enviamos un 5

← → ↻ ⓘ 127.0.0.1:8000/edad/5

**De acuerdo a su edad, usted es: Menor de edad**

Y si enviamos un 25

← → ↻ ⓘ 127.0.0.1:8000/edad/25

**De acuerdo a su edad, usted es: Mayor de edad**

## ACTIVIDAD

Demuestra que has aprendido a manejar las rutas con las vistas agregando 2 vistas nuevas:

- Una vista muestra tu nombre completo
- Otra vista con parámetros, si mandas del 1-6 te dirá: "Tu carrera es TIDEM" y si mandas del 7-11 dirá: "Tu carrera es IGSOFT" y si el usuario manda cualquier otro número le dirá que no es válido.