



**PUCP**

# Desarrollo Web con Python

*CETAM - PUCP*



# Sesión 4:

- Python
  - Variables
  - Condicionales e Iterativas
  - Funciones y procedimientos
  - POO
- Django Framework
  - Creación del proyecto y aplicaciones
  - Settings.py
  - urls.py
  - views.py
  - models.py



# Python

*Desarrollo Web con Python - Sesión 4*



# Hola Mundo - Python

```
print('Hola a todos')
```

# Python - Entornos de ejecución

- Scripts
- Terminal
- Notebooks

# Variables

- `a = 28` `#Variable numérica`
- `b = 1.5` `#Variable float`
- `c = 'Hola'` `#Variable String`
- `d = True` `#Variable Booleana`
- `e = None` `#Variable sin tipo`

# Input - Output

- `nombre = input('Ingrese su nombre')`
- `print(nombre)`



# Condicionales

```
x = 0
if x > 0:
    print('X es positivo')
if x < 0:
    print('X es negativo')
else:
    print('X es cero')
```

# Bucles

- `lista_conjunto = [2,3,4,5,6,7]`  
`for elemento in lista_conjunto:`  
 `print(elemento)`
- `i = 0`  
`while i < 52:`  
 `print(i)`  
 `i = i + 1`

# Funciones y módulos

- `def square(x):  
 return x**x`
- `import functions  
elemento = functions.square(24)  
print(elemento)`

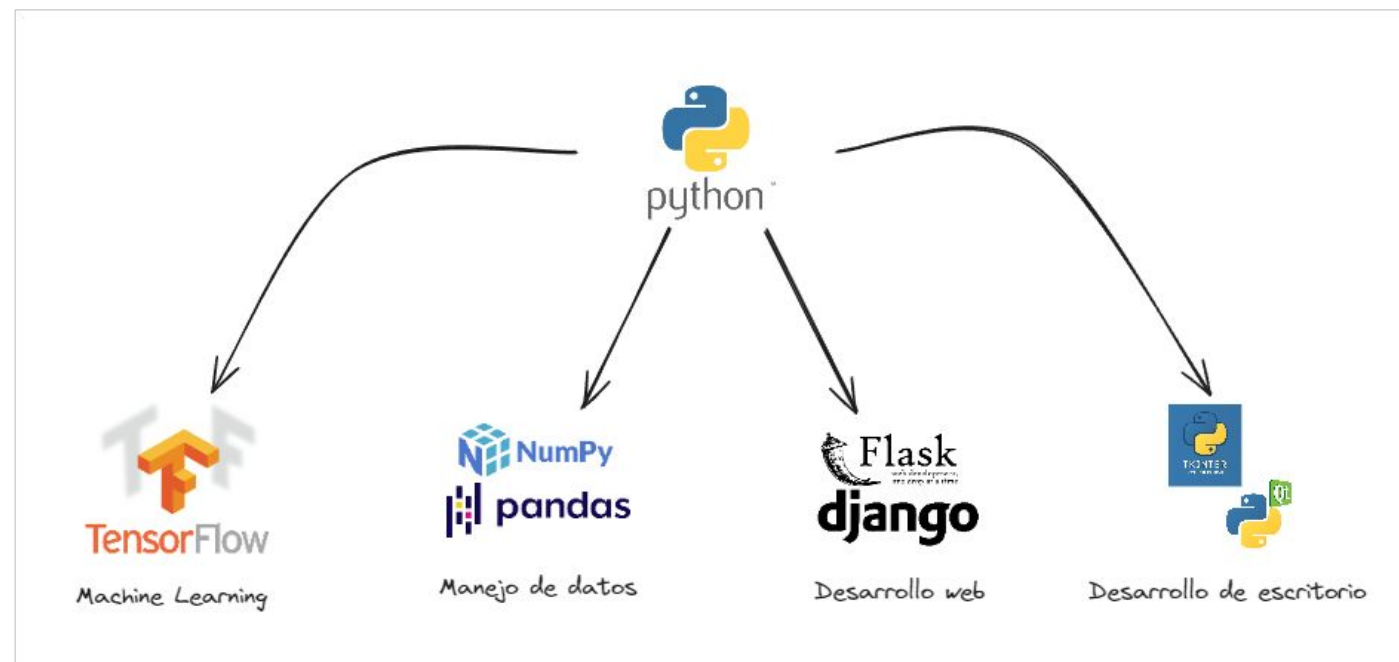
# Programación Orientada a Objetos

```
class flight:  
    def __init__(self,capacity):  
        self.capacity = capacity  
        self.passengers = []  
    def add_passenger(self,name):  
        self.passengers.append(name)  
        return True
```

# Try - Except

```
x = 0
y = 24
try:
    a = y / x
    print('Se ha ejecutado la division')
except:
    print('Error encontrado')
```

# Aplicaciones en python



# django

## Django Framework

*Desarrollo Web con Python - Sesión 4*



# ¿Qué es Django?

Django es un marco de trabajo que permite crear aplicaciones web con rapidez y robustez. Utiliza la estructura MVC y gracias a que se encuentra basado en python se tiene acceso a una gran cantidad de librerías y recursos.

# django



# Cliente - Servidor



# Respuestas Cliente - Servidor

```
GET / HTTP/1.1  
Host: www.example.com  
...
```

```
HTTP/1.1 200 OK  
Content-Type: text/html  
...
```

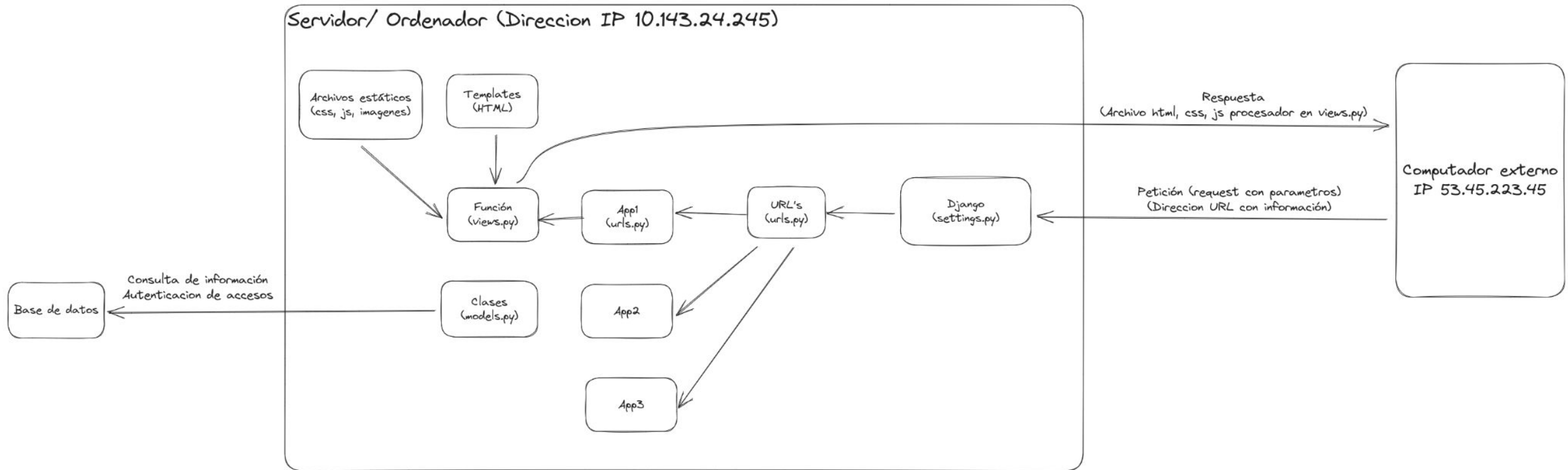
# Instalar Django

- Actualizar pip  
`python -m pip install --upgrade pip`
- Instalar django  
`pip3 install django`

# Manejo de App's en Django

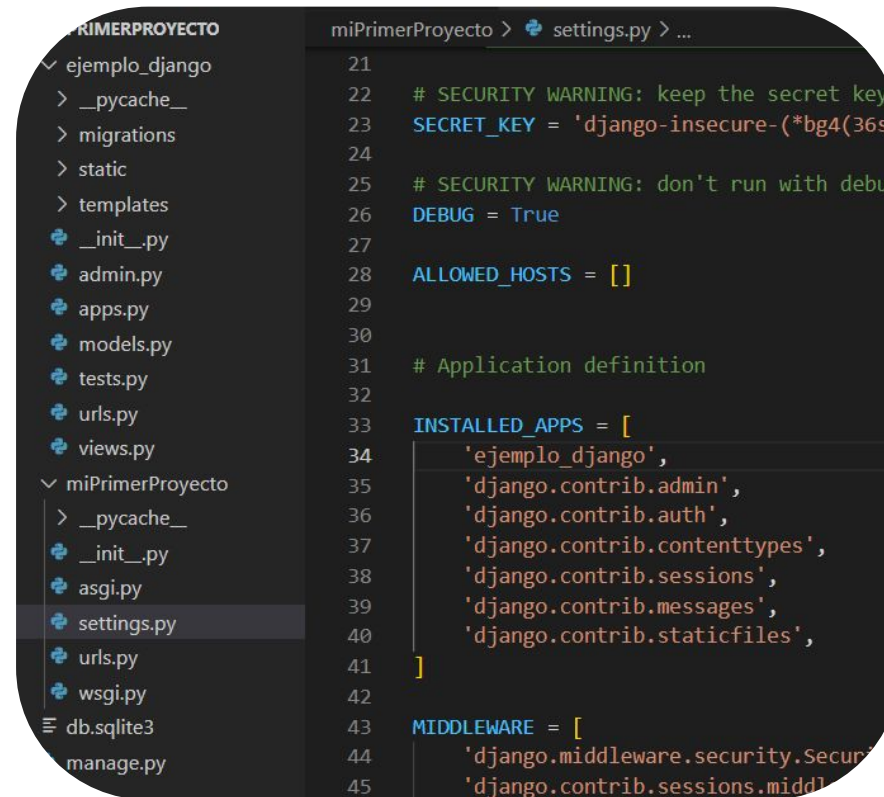
- Crear un proyecto en django  
`django-admin startproject miPrimerProyecto`
- Ejecutar el servidor  
`python manage.py runserver`
- Crear una aplicación en django  
`python manage.py startapp ejemplo_django`

# Funcionamiento del framework



# Registro de aplicaciones

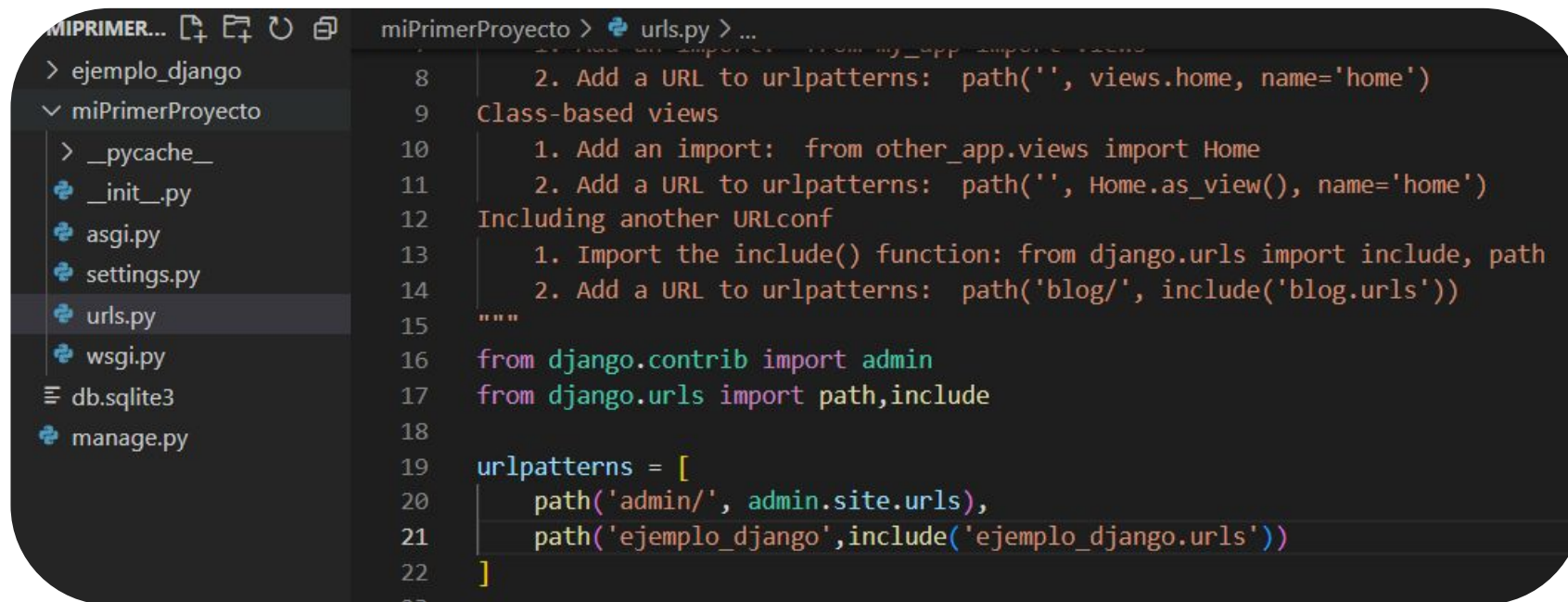
En el archivo 'settings.py' de la carpeta miPrimerProyecto agregar el nombre de la aplicación en la sección 'installed apps':



```
21
22 # SECURITY WARNING: keep the secret key
23 SECRET_KEY = 'django-insecure-(*bg4(36s=
24
25 # SECURITY WARNING: don't run with debug
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'ejemplo_django',
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
```

# Rutear las url's

Por defecto las url's del proyecto se dirigirán a la carpeta raíz, por lo que para redireccionar las url's de nuestra aplicación debemos de agregar el ruteo en el archivo urls.py



The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the project structure with the following files and folders:

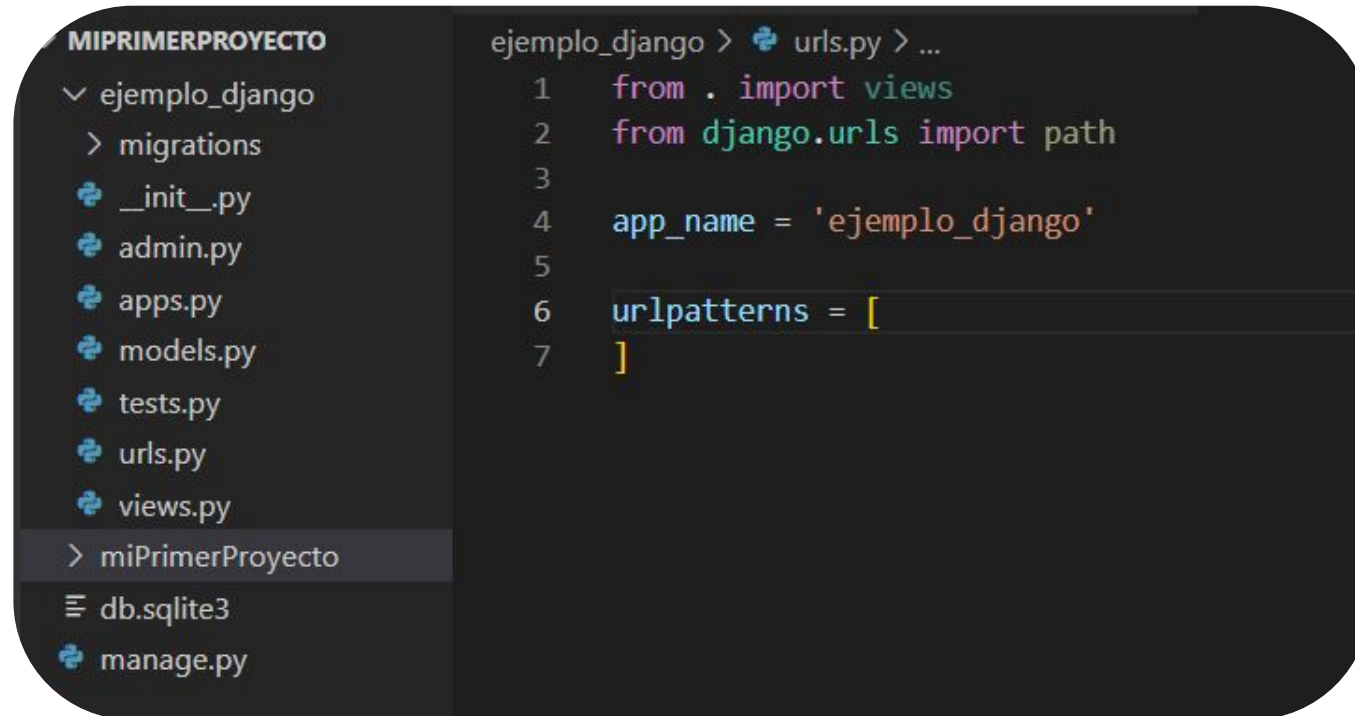
- ejemplo\_django
- miPrimerProyecto
  - \_\_pycache\_\_
  - \_\_init\_\_.py
  - asgi.py
  - settings.py
  - urls.py (selected)
  - wsgi.py
  - db.sqlite3
  - manage.py

The code editor shows the contents of the `urls.py` file:

```
miPrimerProyecto > urls.py > ...
1 1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path('', views.home, name='home')
8
9 Class-based views
10 1. Add an import: from other_app.views import Home
11 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13 1. Import the include() function: from django.urls import include, path
14 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path,include
18
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('ejemplo_django',include('ejemplo_django.urls'))
22 ]
23
```

# Rutear las url's

Crear el archivo `urls.py` al interior de la carpeta de nuestra aplicación `ejemplo_django`; en dicho archivo se irán definiendo las rutas hacia las funciones que utilizaremos en nuestra aplicación web



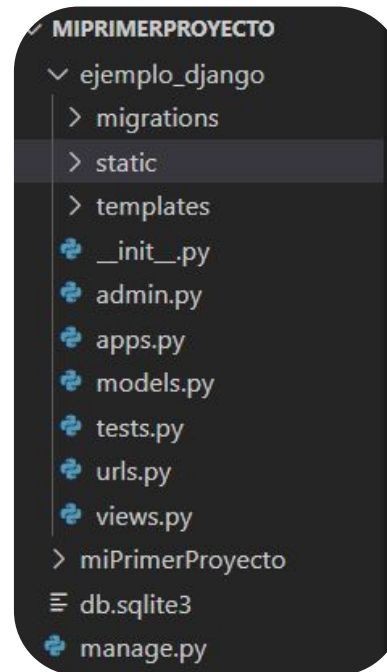
The screenshot shows a code editor with a dark theme. On the left, a file explorer pane displays the project structure under the name 'MIPRIMERPROYECTO'. The 'ejemplo\_django' directory is expanded, showing files like `migrations`, `__init__.py`, `admin.py`, `apps.py`, `models.py`, `tests.py`, `urls.py`, and `views.py`. The `urls.py` file is selected. On the right, the editor shows the content of `ejemplo_django > urls.py` with the following code:

```
ejemplo_django > urls.py > ...
1  from . import views
2  from django.urls import path
3
4  app_name = 'ejemplo_django'
5
6  urlpatterns = [
7  ]
```



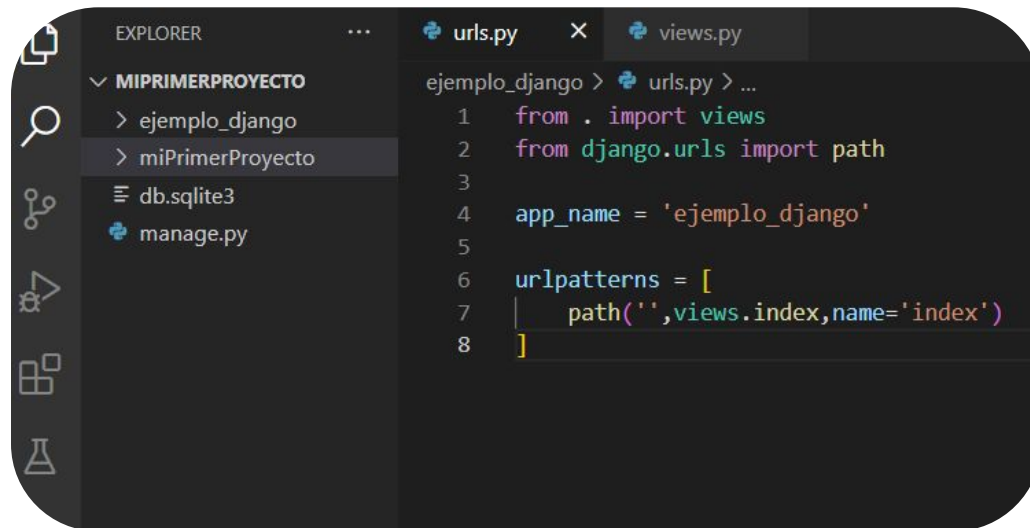
# Crear los directorios en donde se almacenarán las vistas

Dentro de nuestra carpeta del proyecto crear 'templates' y 'static'. Templates nos servirá para generar nuestras vistas mientras que en static se guardaran los archivos .js y .css de nuestro proyecto



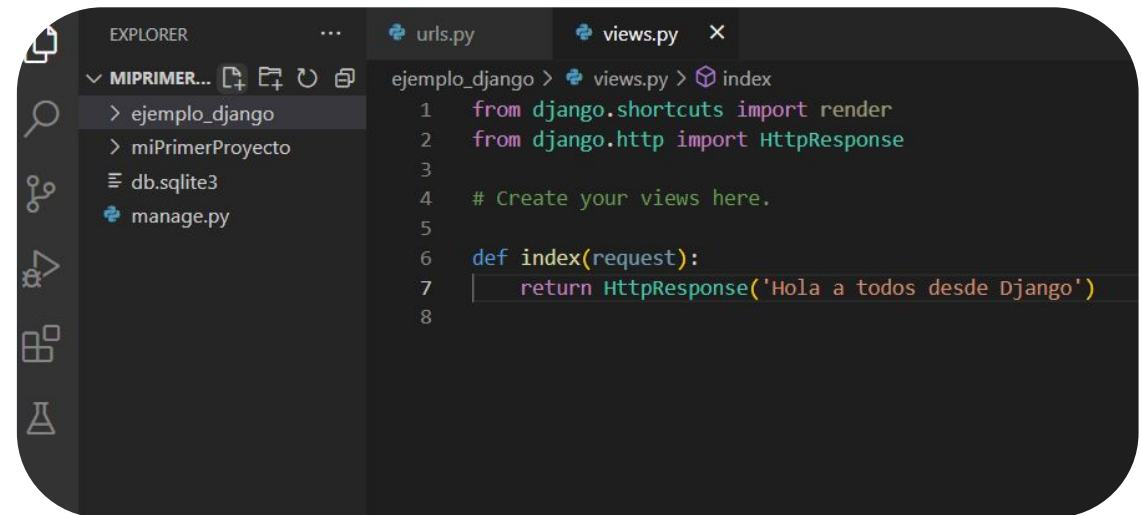
# El archivo views.py

Comenzar a programar las funciones de las rutas en views.py y rutearlas a través del registro en urls.py



The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer shows a project named 'MIPRIMERPROYECTO' with a subdirectory 'ejemplo\_django' containing 'miPrimerProyecto', 'db.sqlite3', and 'manage.py'. The main editor window displays the 'urls.py' file for 'ejemplo\_django'. The code in 'urls.py' is as follows:

```
ejemplo_django > urls.py > ...
1  from . import views
2  from django.urls import path
3
4  app_name = 'ejemplo_django'
5
6  urlpatterns = [
7      path('', views.index, name='index')
8  ]
```



The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer shows a project named 'MIPRIMER...' with a subdirectory 'ejemplo\_django' containing 'miPrimerProyecto', 'db.sqlite3', and 'manage.py'. The main editor window displays the 'views.py' file for 'ejemplo\_django'. The code in 'views.py' is as follows:

```
ejemplo_django > views.py > index
1  from django.shortcuts import render
2  from django.http import HttpResponse
3
4  # Create your views here.
5
6  def index(request):
7      return HttpResponse('Hola a todos desde Django')
8
```

# Diferentes casos

- Renderizar un template html
- Enlazar un estilo .css a un template html
- Enviar parametros en las direcciones
- Enviar información a través de formularios
- etc

Documentación: <https://docs.djangoproject.com/es/4.1/>

# Gracias