





## DESARROLLO CON PYTHON

### Contenido de formación



**Programación**



**Pyhton**



**Paradigma POO**



**Bases de datos SQL**



**Tkinter**



**Django**



**Flask**

1. Primeros Instalación de SW
2. Variables y tipos de datos
3. Operadores (aritméticos, asignación)
4. Entrada y salida de datos
5. Estructuras de control
  - 5.1 Condicionales (operadores lógicos, operadores de comparación)
  - 5.2 Bucles (estructuras interactivas for, while)
6. Bloque de ejercicios de aplicación de conceptos
7. Funciones (parámetros, return, invocación, lambda)
  - 7.1 Variables locales y globales, funciones y métodos predefinidos
8. Lista y tuplas (creación, índices, recorrer y mostrar listas, listas multidimensionales)
9. Diccionarios y sets
10. Bloque de ejercicios aplicación de conceptos
11. Módulos y paquetes (creación y funcionalidad)
12. Sistemas de archivos y directorios
13. Manejo de errores (captura de excepciones, errores personalizados)
14. Programación orientada a objetos
15. Bases de datos SQLite
16. Bases de datos MySQL
17. Proyecto con Python
18. Interfaces graficas con Tkinter (aplicación de escritorio Python – Tkinter)
19. Desarrollo Web con Django
20. Interfaces graficas con Flask (aplicación de escritorio Python – Flask)



## Django

Python es el quinto lenguaje de programación en popularidad en estos momentos. Se trata de un lenguaje que se emplea, sobre todo, en la programación para las distintas aplicaciones de análisis de datos e IA, pero que también es importante para la programación web. En este contexto, el trabajo de los programadores es mucho más sencillo con herramientas como Django, un **framework** específico para el trabajo con Python.

Django es un framework de desarrollo para Python que se emplea para la creación de páginas web. Se trata de una herramienta de código abierto y gratuita que cuenta con una comunidad amplia y que comparte recursos constantemente. Además, Django también cuenta con funciones de pago que pueden facilitar más el trabajo de los desarrolladores.

Django es una herramienta que se puede usar para el desarrollo full-stack de aplicaciones y páginas web, así como para el desarrollo de servidores. Está considerado como el mejor framework para el desarrollo de aplicaciones web con Python y es uno de los marcos de desarrollo más demandados por los programadores que trabajan con este lenguaje en el desarrollo web.

Creado por Adrian Holovaty y Simon Willison mientras estaban trabajando en PHP y necesitaban algo que les facilitase actualizar una web de manera rápida para cumplir con los plazos. En ese momento se pasaron a Python y, en 2005, lanzaron Django para que la programación fuese aún más sencilla.

Estas son algunas de las características por las que Django es uno de los frameworks más usados para el desarrollo web:

### **Completo**

Django proporciona casi todo lo que los programadores necesitan y pueden querer usar. Se trata de una herramienta que sigue unos principios de diseño consistentes y que cuenta con una buena base de documentación para facilitar el trabajo de los desarrolladores.

### **Escalable**

Django funciona por componentes sustituibles e intercambiables. Eso significa que se puede escalar con bastante facilidad. En este sentido, un ejemplo de uso de Django lo tenemos en Instagram o Disqus, plataformas que lo han empleado para mejorar sus servidores gracias a la escalabilidad del entorno de desarrollo.

### **Versátil**

Este entorno de desarrollo se ha empleado para la creación de todo tipo de páginas web. Desde sistemas que son puramente para administración de contenidos, como puede ser, por ejemplo, una wiki; hasta redes sociales o páginas webs de noticias.

Y es que, Django es compatible con cualquier framework que se emplea del lado del cliente, por lo que puede mandar contenidos en cualquier formato.

### **Seguro**

Django facilita la detección y solución de posibles fallos en la seguridad en las páginas web del lado del servidor. Proporciona una administración segura de usuarios y contraseñas y evita errores que son comunes en el diseño y desarrollo back-end.

### **Portátil**

Django se puede usar en cualquier sistema y plataforma y, además, está respaldado por muchos de los proveedores de hosting que, además, suelen proporcionar la documentación necesaria para implementarlo.

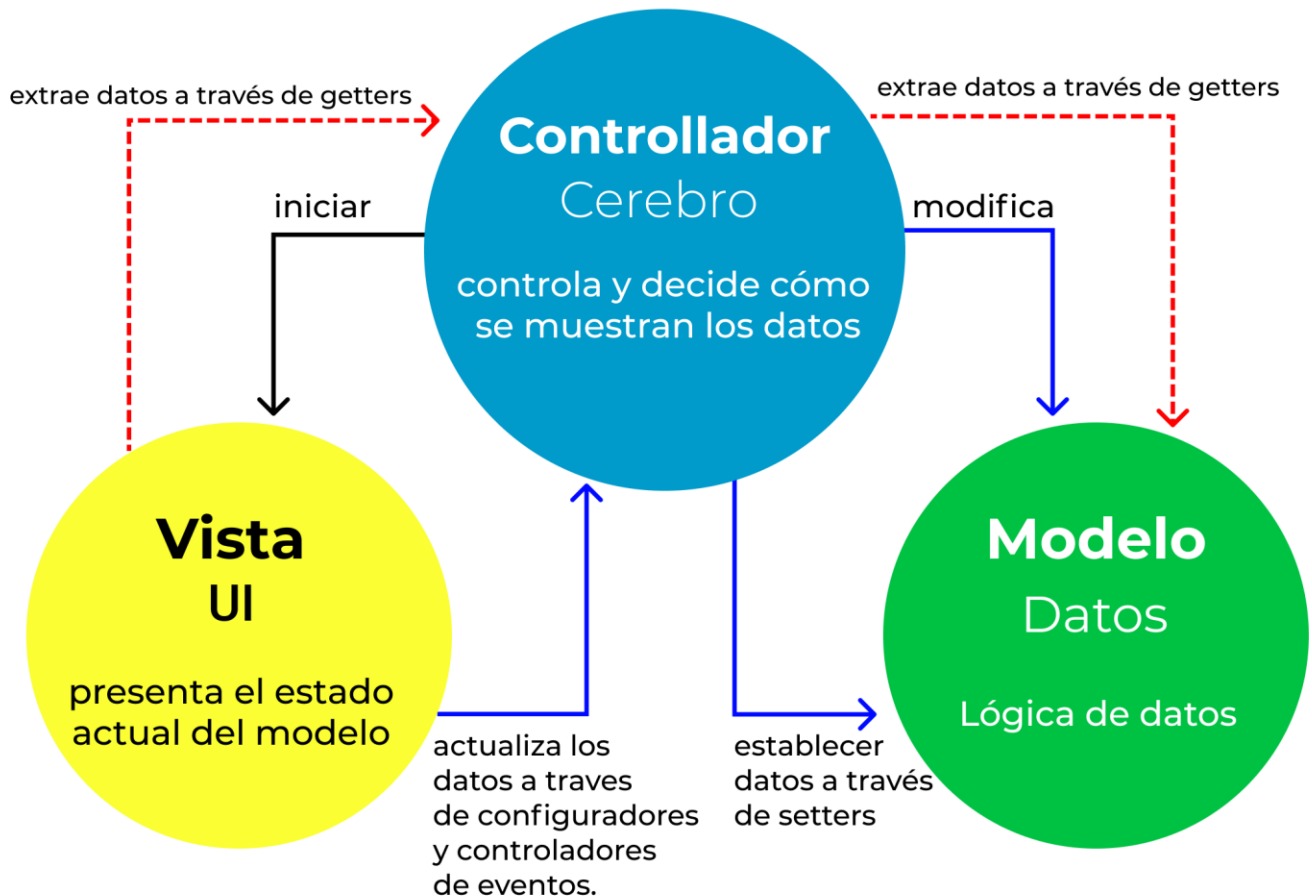


### Mantenimiento

Django facilita el mantenimiento y la velocidad del mismo en el desarrollo web. Fomenta la creación de código reutilizable, lo cual simplifica, en gran medida, el trabajo de los desarrolladores web.

Usamos el modelo, vista controlador

# Patrones de Arquitectura MVC



### Instalación de Django

Cmd

1. Comprobar si tenemos Python

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19042.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\MGV>python --version
Python 3.10.7

C:\Users\MGV>
```





## 2. Instalar Django

```
Símbolo del sistema
C:\Users\MGV>pip install Django==4.1.6
Collecting Django==4.1.6
  Downloading Django-4.1.6-py3-none-any.whl (8.1 MB)
    ----- 8.1/8.1 MB 1.4 MB/s eta 0:00:00
Collecting asgiref<4,>=3.5.2
  Downloading asgiref-3.6.0-py3-none-any.whl (23 kB)
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.3-py3-none-any.whl (42 kB)
    ----- 42.8/42.8 2.0 MB/s eta 0:00:00
    kB
Collecting tzdata
  Downloading tzdata-2022.7-py2.py3-none-any.whl (340 kB)
    ----- 340.1/340.1 1.8 MB/s eta 0:00:00
    kB
Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-4.1.6 asgiref-3.6.0 sqlparse-0.4.3 tzdata-2022.7

[notice] A new release of pip available: 22.3.1 -> 23.0
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\MGV>
```

Creamos en VsCode una carpeta 22-django luego vamos al cmd para ubicarnos en la carpeta creada

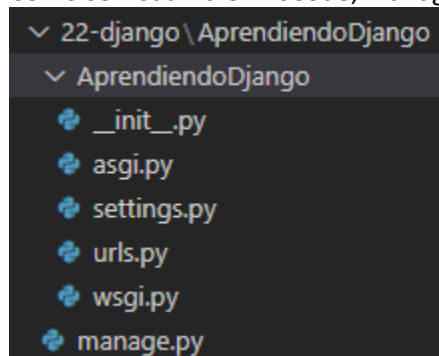
```
c:\xampp\htdocs\sena-python>cd 22-django

c:\xampp\htdocs\sena-python\22-django>
```

Iniciamos un nuevo proyecto en Django **AprendiendoDjango**

```
Símbolo del sistema
c:\xampp\htdocs\sena-python\22-django>django-admin startproject AprendiendoDjango
c:\xampp\htdocs\sena-python\22-django>
```

Como se visualiza en VsCode, manage.py es un archivo de comandos



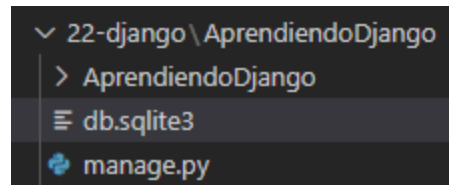
Vamos a visualizar el proyecto un navegador web, para lo cual debo migrar el proyecto, y lo que hace es generar una base de datos con la funcionalidad de las aplicaciones que vienen por defecto dentro de Django.



```
c:\xampp\htdocs\sena-python\22-django\AprendiendoDjango>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, 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
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK

c:\xampp\htdocs\sena-python\22-django\AprendiendoDjango>
```

Y visualizamos el cambio en VsCode



Ahora arrancamos el servidor local de Django, esta consola siempre la debemos tener abierta en segundo plano

```
Seleccionar Símbolo del sistema - python manage.py runserver

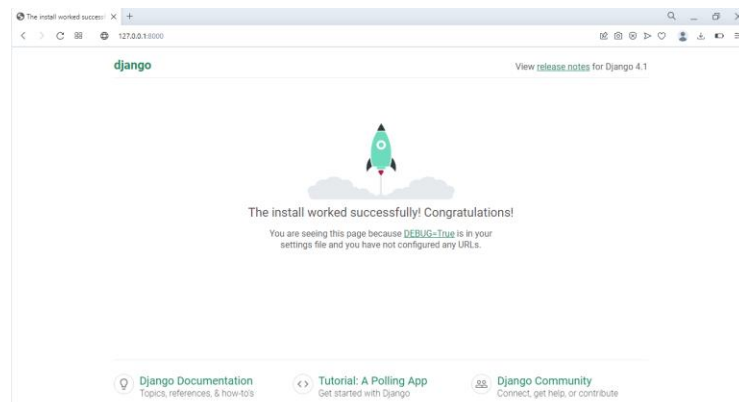
c:\xampp\htdocs\sena-python\22-django\AprendiendoDjango>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 11, 2023 - 13:01:19
Django version 4.1.6, using settings 'AprendiendoDjango.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

En donde se ejecuta

Dirección del servidor

Ahora lo corremos en el navegador **<http://127.0.0.1:8000/>**:



Instructor: Ing. Moisés García Vargas





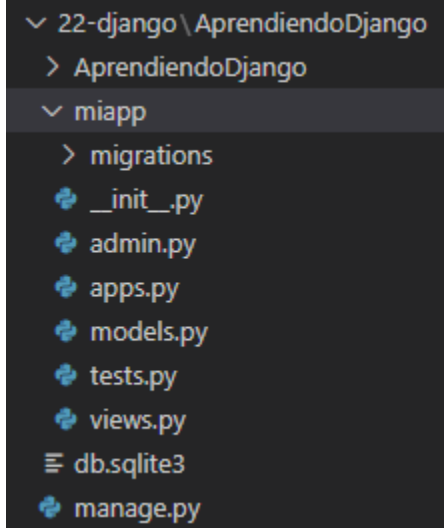


### 3. Crear apps con Django

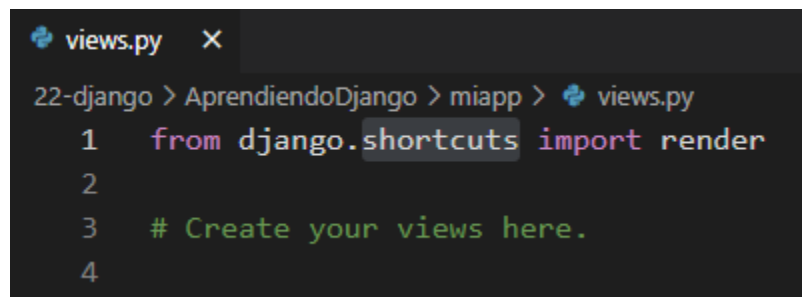
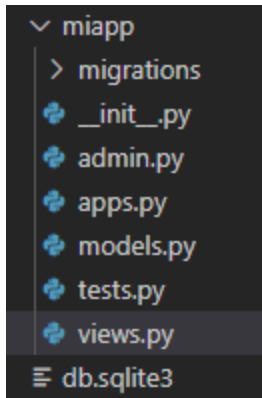
Simbolo del sistema

```
c:\xampp\htdocs\sena-python\22-django\AprendiendoDjango>python manage.py startapp miapp
c:\xampp\htdocs\sena-python\22-django\AprendiendoDjango>
```

Como de visualiza en VsCode



### 4. Creación de vistas y rutas, esto es dentro del fichero **views.py**



### Diferencia entre MVC y MVT

MVC = MODELO

VISTA

CONTROLADOR

**MVT = MODELO**

**TEMPLATE**

**VISTA** → así se maneja en Django

Las siglas en ingles MTV corresponden a Model (Modelo), Template (Plantilla) y View (Vista). En este post me referiré a las plantillas como templates, ya que su uso en el español es bastante común.

En la practica el patrón MTV es muy similar al MVC a tal punto que se puede decir que Django es un framework MVC. Realmente este no se desvía demasiado del patrón Modelo Vista Controlador, simplemente lo implementa de una manera distinta y para evitar confusiones es llamado MTV.





## Modelo Vista Controlador (MVC)

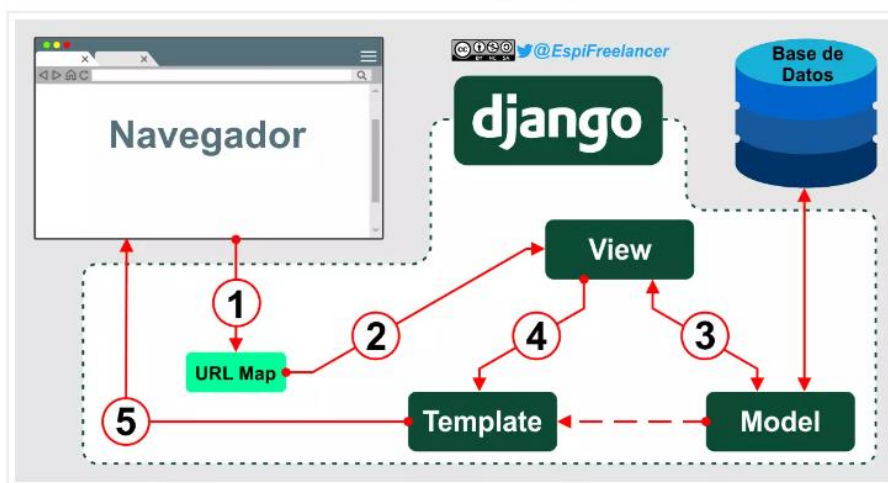
- Modelo: Es el que se encarga de manipular la información de la aplicación, la cual usualmente esta almacenada en la base de datos.
- Vista: Decide qué información mostrar y cómo mostrarla.
- Controlador: Es quien responde a las peticiones, decide que vista usar y si es necesaria información accede al modelo.

## 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 será mostrada y por cual template.
- Template: Decide como será mostrada la información.

## Funcionamiento MTV de Django



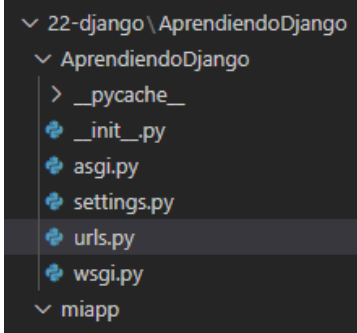
## Creación de nuestra primera vista

```
views.py X
22-django > AprendiendoDjango > miapp > views.py
1 from django.shortcuts import render, HttpResponseRedirect
2
3 # Create your views here.
4 #request es un parametro que permite recibir datos de una URL
5 #Se le pasa a cada una de nuestras vistas
6
7 def hola_mundo(request):
8     #respuesta HTTP que se debe importar HttpResponseRedirect
9     return HttpResponseRedirect("Hola Mundo con Django!!!")
10
```



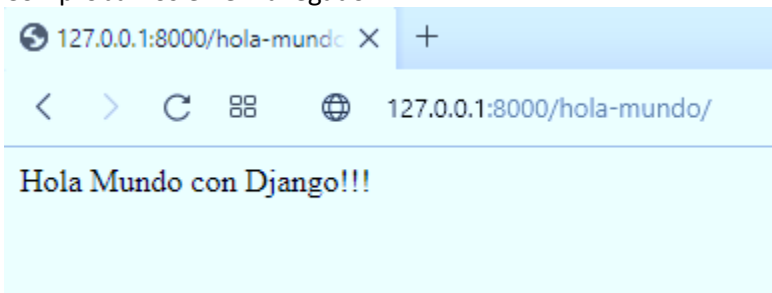


## Configuración de la URL



```
urls.py x
22-django > AprendiendoDjango > AprendiendoDjango > urls.py
12 including another URLconf
13     1. Import the include() function: from django.urls import include
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
18
19 #Importar app con mis vistas
20 from miapp import views
21
22 urlpatterns = [
23     path('admin/', admin.site.urls),
24     #creacion de ruta
25     path('hola-mundo/', views.hola_mundo, name="hola_mundo")
26 ]
```

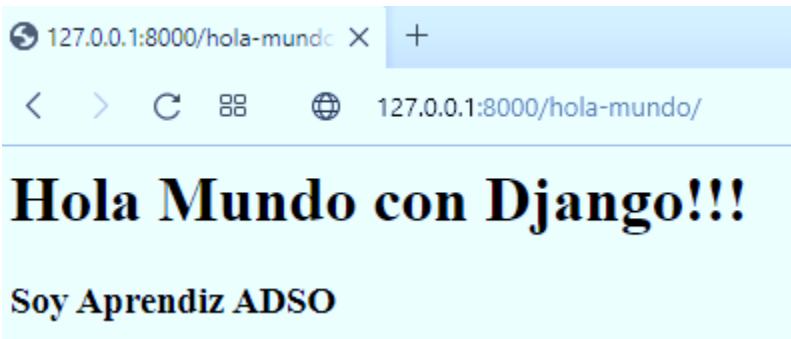
## Comprobamos en el navegador



## Algunos cambios

```
views.py x
22-django > AprendiendoDjango > miapp > views.py
5 #Se le pasa a cada una de nuestras vistas
6
7 def hola_mundo(request):
8     #respuesta HTTP que se debe importar HttpResponse
9     return HttpResponse("""
10     <h1>Hola Mundo con Django!!!</h1>
11     <h3>Soy Aprendiz ADSO</h3>
12     """)
13
```



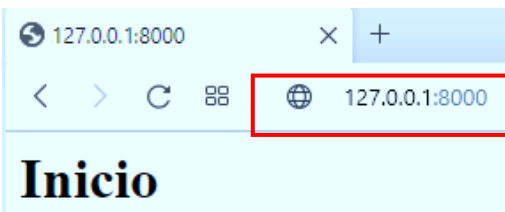


Creación de un método Inicio → *index*

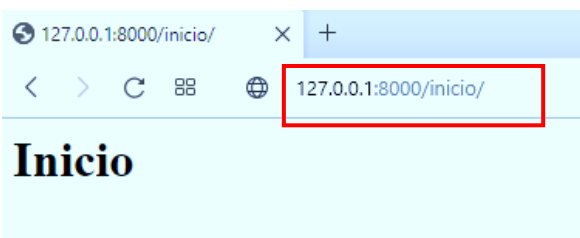
```
views.py X
22-django > AprendiendoDjango > miapp > views.py
1 from django.shortcuts import render, HttpResponse
2
3 def index(request):
4     return HttpResponse("""
5         <h1>Inicio</h1>
6     """)
7
8 # Create your views here.
9 #request es un parametro que permite recibir datos de una URL
10 #Se le pasa a cada una de nuestras vistas
```

Ahora para mostrarla y convertirla en la página de inicio

```
22 urlpatterns = [
23     path('admin/', admin.site.urls),
24     #ruta para pagina de inicio -> index
25     path('', views.index, name="index"),
26     #creacion de ruta
27     path('hola-mundo/', views.hola_mundo, name="hola_mundo")
28 ]
```



```
22 urlpatterns = [
23     path('admin/', admin.site.urls),
24     #ruta para pagina de inicio -> index
25     path('', views.index, name="index"),
26     #Otro cambio
27     path('inicio/', views.index, name="inicio"),
28     #creacion de ruta
29     path('hola-mundo/', views.hola_mundo, name="hola_mundo")
30 ]
```





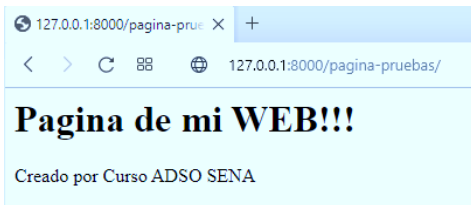
Ahora vamos a crear una que se llame pagina

```
17 def pagina(request):
18     return HttpResponse("""
19         <h1>Pagina de mi WEB!!!</h1>
20         <p>Creado por Curso ADSO SENA</p>
21     """)
```

views

```
27 path('inicio/', views.index, name="inicio"),
28 #creacion de ruta
29 path('hola-mundo/', views.hola mundo, name="hola mundo")
30 path('pagina-pruebas/', views.pagina, name="pagina")
31 ]
```

urls



## Navegación entre rutas

### Creación de plantilla Layout

#### Definición del Views

```
1 from django.shortcuts import render, HttpResponse
2
3 layout = """
4 <h1>Sitio WEb con Django | Aprendices ADSO</h1>
5 <hr/>
6 <ul>
7     <li>
8         <a href="/inicio">Inicio</a>
9     </li>
10    <li>
11        <a href="/hola-mundo">Hola Mundo</a>
12    </li>
13    <li>
14        <a href="/pagina-pruebas">Página de Pruebas</a>
15    </li>
16 </ul>
17 <hr/>
18 """
19
20 def index(request):
```

Y esto se le concatena a todos losHttpResponse(layout+

```
20 def index(request):
21     return HttpResponse(layout+"""
22     <h1>Inicio</h1>
```





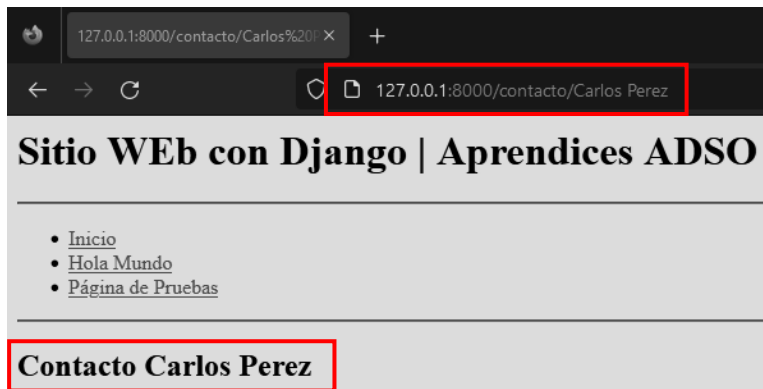
### Parámetros en rutas:

Creamos este método

```
40 def contacto(request, nombre):
41     return HttpResponse(layout+f"<h2>Contacto {nombre}</h2>")
```

Pasar parámetro

```
30 path('pagina-pruebas/', views.pagina, name="pagina"),
31 path('contacto/<str:nombre>', views.contacto, name="contacto")
```



### Plantillas y templates en Django

Creamos dentro de **miapp** una carpeta llamada **templates** y creamos dentro un fichero que se llamara **index.html** igual que el método.

```
< index.html x views.py
22-django > AprendiendoDjango > miapp > templates > < index.html
1 <h1> Inicio</h1>
2 <p>ADSO SENA 2023 - Regional CAUCA:</p>
```

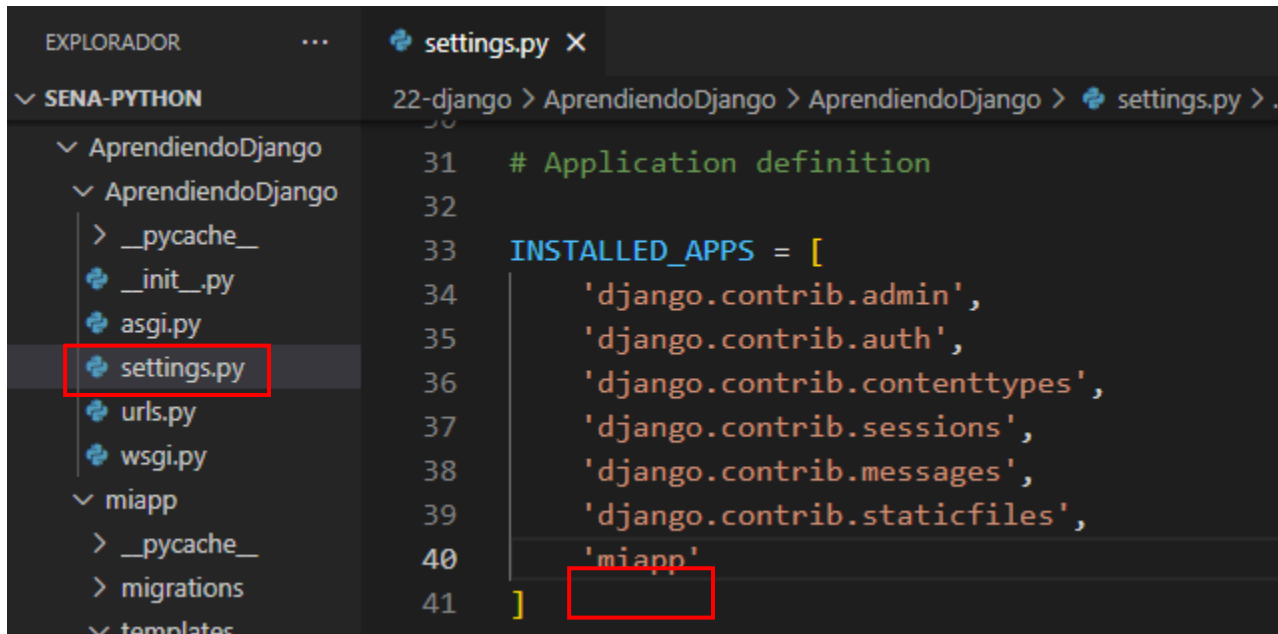


Vinculamos la vista al template en el archivo **views.py**

Verificamos que este importado el **render** en los **shortcuts**

```
22-django > AprendiendoDjango > miapp > views.py > ...  
1 from django.shortcuts import render, HttpResponseRedirect  
2
```

Luego **settings.py** en la línea de **INSTALLED\_APPS**



The screenshot shows the VS Code interface with the **settings.py** file open. The left sidebar shows the project structure with **settings.py** highlighted. The main editor shows the **INSTALLED\_APPS** list, which includes **'miapp'** at the end. A red box highlights the **'miapp'** entry.

```
31 # Application definition  
32  
33 INSTALLED_APPS = [  
34     'django.contrib.admin',  
35     'django.contrib.auth',  
36     'django.contrib.contenttypes',  
37     'django.contrib.sessions',  
38     'django.contrib.messages',  
39     'django.contrib.staticfiles',  
40     'miapp',  
41 ]
```

Después vamos **views.py** configuramos la siguiente línea para utilizar el **render** aquí es donde se separan las vistas del controlador

```
20 def index(request):  
21     html = """ return HttpResponseRedirect(layout+  
22         <h1>Inicio</h1>  
23     )"""  
24     year = 2023  
25  
26     return render(request, 'index.html')
```

Ahora crearemos las plantillas para las diferentes funciones:

Creamos en la carpeta **templates** otro fichero llamado **hola\_mundo.html** y **CORTAMOS** el siguiente código desde **views.py** y lo pegamos en el **hola\_mundo.html**



```
views.py x hola_mundo.html settings.py urls.py
22-django > AprendiendoDjango > miapp > views.py > index
28 #request es un parametro que permite recibir datos de una URL
29 #Se le pasa a cada una de nuestras vistas
30 def hola_mundo(request):
31     #respuesta HTTP que se debe importarHttpResponse
32     return HttpResponse(layout+"""
33         <h1>Hola Mundo con Django!!!</h1>
34         <h3>Soy Aprendiz ADSO</h3>
35     """)
```

```
views.py x hola_mundo.html x settings.py urls.py
22-django > AprendiendoDjango > miapp > templates > hola_mundo.html
1 <h1>Hola Mundo con Django!!!</h1>
2 <h3>Soy Aprendiz ADSO</h3>
```

Luego utilizamos el método render para reemplazar en viwes.py

```
views.py x hola_mundo.html settings.py urls.py
22-django > AprendiendoDjango > miapp > views.py > hola_mundo
28 #request es un parametro que permite recibir datos de una URL
29 #Se le pasa a cada una de nuestras vistas
30 def hola_mundo(request):
31     #le pasamos el render, la respuesta request y el nombre de la template
32     return render(request, 'hola_mundo.html')
33
34 def pagina(request):
```

Probamos en el navegador:







Ahora creamos una para **pagina** de la misma manera que el anterior:

- Creamos el fichero en el templates que se llama pagina.html
- Corregimos el views.py en la función página.
- Correr en el navegador.

## Layout, bloques y herencia de plantillas

Creación de Layout

En **templates** crear un fichero llamado **layout.html**, el cual servirá para cargar todas las templates después creamos una estructura html básica. Copiamos y pegamos en el body el menú que ya tenemos en views.py

```
<> layout.html X
22-django > AprendiendoDjango > miapp > templates > <> layout.html
1  <!DOCTYPE html>
2  <html>
3      <html lang="es">
4      <head>
5          <meta charset="utf-8">
6          <title>Sitio WEB con Django - ADSO SENA</title>
7      </head>
8      <body>
9          <h1>Sitio WEB con Django | Aprendices ADSO</h1>
10         <hr/>
11         <ul>
12             <li>
13                 <a href="/inicio">Inicio</a>
14             </li>
15             <li>
16                 <a href="/hola-mundo">Hola Mundo</a>
17             </li>
18             <li>
19                 <a href="/pagina-pruebas">Página de Pruebas</a>
20             </li>
21         </ul>
22         <hr/>
23         <div id="content">
24         </div>
25         <footer>
26             CURSO DE DJANGO ADSO 2023 &copy; Instructor Moisés García
27         </footer>
28     </body>
29 </html>
```



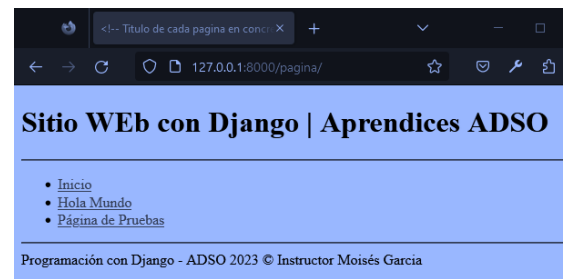
Ahora creamos diferentes bloques que es cuando sustituimos un trozo de pantalla por otro en las diferentes templates. Definimos un bloque dentro de **content** y en **title**

```
layout.html x pagina.html views.py urls.py
22-django > AprendiendoDjango > miapp > templates > layout.html
1  <!DOCTYPE html>
2  <html>
3      <html lang="es">
4      <head>
5          <meta charset="utf-8">
6          <title>
7              {% block title %}
8              <!-- Titulo de cada pagina en concreto-->
9              {% endblock %}
10             |- ADSO SENA</title>
11     </head>
12     <body>
13         <h1>Sitio WEb con Django | Aprendices ADSO</h1>
14         <hr/>
15         <ul>
16             <li>
17                 <a href="/inicio">Inicio</a>
18             </li>
19             <li>
20                 <a href="/hola-mundo">Hola Mundo</a>
21             </li>
22             <li>
23                 <a href="/pagina-pruebas">Página de Pruebas</a>
24             </li>
25         </ul>
26         <hr/>
27         <div id="content">
28             {%block content %}
29             <!-- Viene codigo de templates, que sera diferente para cada caso-->
30             {% endblock %}
31         </div>
32         <footer>
33             Programación con Django - ADSO 2023 &copy; Instructor Moisés Garcia
34         </footer>
35     </body>
36 </html>
37
```

Ahora vamos a pagina.html y utilizamos el **extends** para heredar el layout principal en el template y correr pagina.html



```
layout.html | pagina.html x | views.py | urls.py
22-django > AprendiendoDjango > miapp > templates > < pagina.html
1  {% extends 'layout.html' %}
2
3  <h1>Pagina de mi WEB</h1>
4  <p>Creado por ADSO SENA - 2023 </p>
```



Como cargar el contenido de cada página en concreto. Ya que en este momento se pasa todo lo del layout.

```
layout.html | < pagina.html x | views.py | urls.py
22-django > AprendiendoDjango > miapp > templates > < pagina.html
1  {% extends 'layout.html' %}
2
3  {% block title %}
4  Pagina de prueba para heredar
5  {% endblock %}
6
7  {% block content %}
8  <h1>Pagina de mi WEB</h1>
9  <p>Creado por ADSO SENA - 2023 </p>
10 {% endblock %}
```



Realizar el mismo procedimiento de heredar en index, hola\_mundo - para colocar el layout y titulo en cada página.

Ahora realizaremos la actualización de contenido por defecto en un bloque, para este caso vamos a **layout.html** y colocamos y luego aplicamos la herencia del padre en **hola\_mundo.html**:

```
27 <div id="content">
28     {%block content %}
29     <strong>Contenido del bloque original!!</strong>
30     {% endblock %}
31 </div>
```

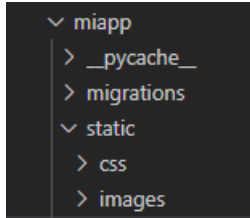
```
layout.html | pagina.html | index.html | < hola_mundo.html x
22-django > AprendiendoDjango > miapp > templates > < hola_mundo.html
1  {% extends 'layout.html' %}
2
3  {% block title %}
4  Hola Mundo!!!
5  {% endblock %}
6
7  {% block content %}
8  {{block.super}} <!--Heredamos del padre-->
9  <h1>Hola Mundo con Django!!!</h1>
10 <h3>Soy Aprendiz ADSO</h3>
11 {% endblock %}
```



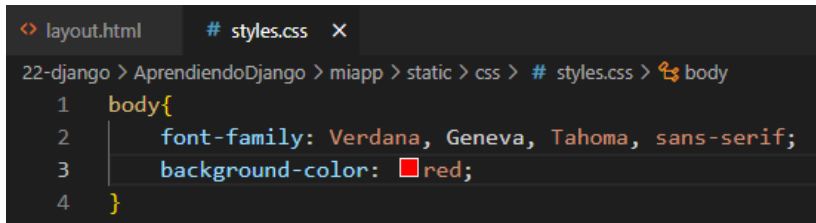


## Vincular hojas de estilos CSS en Django

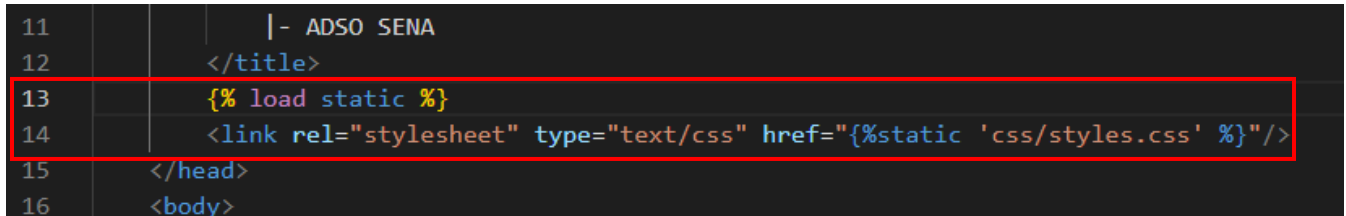
1. Creamos una carpeta **miapp** llamada **static** y dentro creamos otra carpeta llamada **css** y otra que se llama **images**



2. En **css** creamos un archivo llamado **styles.css**



3. Cargar los estilos estáticos en layout.html



Probamos el cambio





## Maquetación estructura HTML5 mejorada:

Cambios en *layout.html*

```
layout.html x # styles.css
22-django > AprendiendoDjango > miapp > templates > layout.html
1 {% load static %}
2 <!DOCTYPE html>
3 <html>
4     <html lang="es">
5     <head>
6         <meta charset="utf-8">
7         <title>
8             {% block title %}
9             <!-- Titulo de cada pagina en concreto-->
10            {% endblock %}
11            |- ADSO SENA
12        </title>
13        {% load static %}
14        <link rel="stylesheet" type="text/css" href="{%static 'css/styles.css' %}" />
15    </head>
16    <body>
17        <!-- Cabecera de la Página -->
18        <header>
19            <div id="logotipo"> <!--se usa id porque sera algo que estara una vez en la pagina-->
20                <!--Logo Django-->
21                <h1> Sitio WEB con Django </h1>
22            </div>
23        </header>
24        <div id="nav">
25            <ul>
26                <li>
27                    <a href="/hola mundo">Hola Mundo</a>
28                </li>
29            </ul>
30        </div>
31        <div id="content">
32            <h2 class="title">Titulo Prueba</h2>
33            {%block content %}
34                <strong>Contenido del bloque original!!</strong>
35            {% endblock %}
36        </div>
37        <div id="footer">
38            Programación con Django - ADSO 2023 &copy; Instructor Moisés Garcia
39        </div>
40    </body>
41 </html>
```

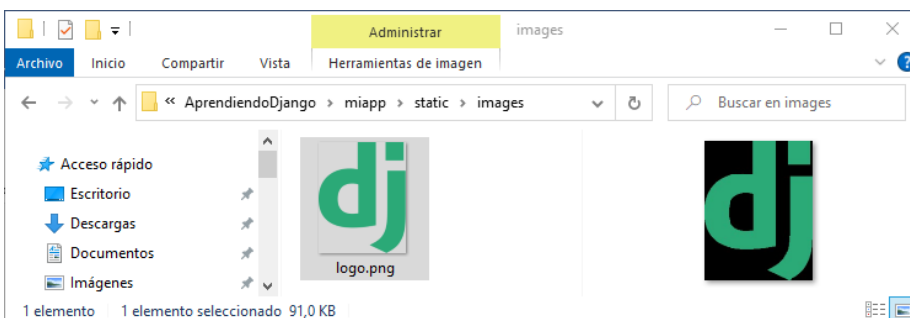


## Estilos – css iniciales:

```
layout.html # styles.css X
22-django > AprendiendoDjango > miapp > static > css > # styles.css > #logotipo h1
1  /* Estilos Generales*/
2  *{
3      margin: 0px;
4      padding: 0px;
5      font-family: Arial, Helvetica, sans-serif;
6      text-decoration: none;
7  }
8
9  body{
10     background-color: #f2f2f2;
11 }
12
13 /*Estilos Cabecera*/
14 header{
15     width: 1212px;
16     height: 140px;
17     background-color: #23282b;
18     margin: 0px auto;
19 }
20 #logotipo h1{
21     display: block;
22     letter-spacing: 2px;
23     text-transform: uppercase;
24     font-weight: normal;
25     color: #42a096;
26 }
```

**Cargar imagen en template de Django:** en la carpeta **images** -> **mostrar en el explorador de archivos** y se pega la imagen con el logo.png

```
18 <header>
19     <div id="logotipo"> <!--se usa id porque sera algo que estara una vez en la pagina-->
20         <!--Logo Django-->
21         
22         <h1> Sitio WEB con Django </h1>
```





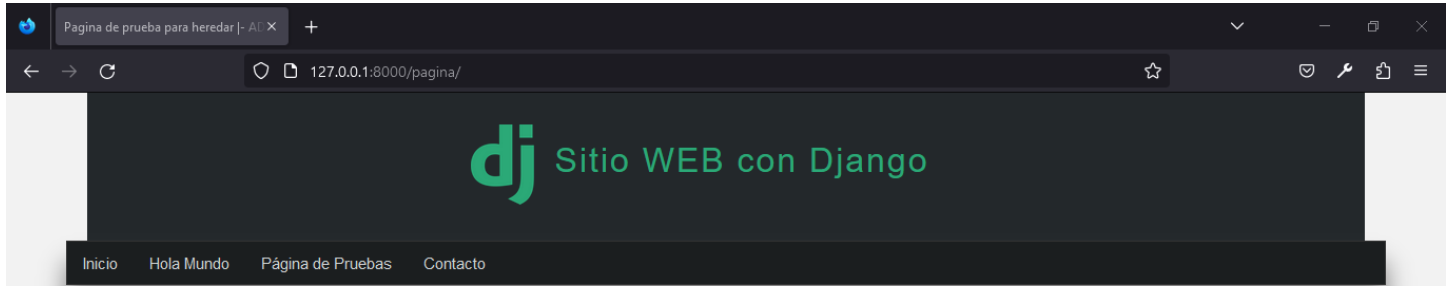


Estilos para la ubicación de logo y estilos: verificar los cambios realizados en el navegador

```
layout.html # styles.css X
22-django > AprendiendoDjango > miapp > static > css > # styles.css > #logotipo img
1  /* Estilos Generales*/
2  *{
3      margin: 0px;
4      padding: 0px;
5      font-family: Arial, Helvetica, sans-serif;
6      text-decoration: none;
7  }
8  body{
9      background-color: #f2f2f2;
10 }
11 /*Estilos Cabecera*/
12 header{
13     width: 1212px;
14     height: 140px;
15     background-color: #23282b;
16     margin: 0px auto;
17 }
18 /*Centrar texto y logo*/
19 #logotipo{
20     width: 40%;
21     height: 130px;
22     margin: 0px auto;
23     padding-top: 10px;
24 }
25 /*Tamaño y ubicacion de la imagen logo*/
26 #logotipo img{
27     display: block;
28     width: 60px;
29     float: left;
30     margin-top: 20px;
31 }
32 /*Tamaño y ubicacion del texto logo*/
33 #logotipo h1{
34     display: block;
35     float: left;
36     margin-top: 35px;
37     margin-left: 20px;
38     letter-spacing: 2px;
39     font-weight: lighter;
40     color: #2ba977;
41 }
```



## Estilos barra de navegación:



### Título Prueba

## Pagina de mi WEB

Creado por ADSO SENA - 2023

Programación con Django - ADSO 2023 © Instructor Moisés García

Cambio en **style.css** para lograr el cambio de arriba

```
40     color: #2ba977;
41 }
42 /*Estilos barra de navegación*/
43 nav{
44     width: 1250px;
45     height: 40px;
46     background-color: #1b1e1f;
47     border: 1px solid #333333;
48     margin: 0 auto;
49     box-shadow: 0px 22px 22px gray;
50     font-size: 15px;
51     margin-bottom: 45px;
52 }
53 nav ul{
54     list-style: none;
55     text-decoration: 40px;
56 }
57 nav ul li{
58     line-height: 40px;
59     float: left;
60 }
61 /*enlaces*/
62 nav ul li a{
63     display: block;
64     padding-left: 15px;
65     padding-right: 15px;
66     color: #d1d4d6;
67 }
68
69 nav ul li a:hover{
70     background: #2ba977;
71     box-shadow: 0px 0px 5px #444444 inset;
72     color: white;
73     transition: all 300ms;
74 }
```





### Estilos del contenido central:

```

73     transition: all 300ms;
74 }
75 /*Estilos contenido central*/
76 #content{
77     width: 1212px;
78     min-height: 930px;
79     margin: 0 auto;
80     margin-bottom: 30px;
81     margin-top: 30px;
82 }

```

### Creamos una caja para el contenido central en *layout.html*

```

40     </nav>
41     <div id="content">
42         <div class="box">
43             <h2 class="title">Titulo Prueba</h2>
44             {%block content %}
45                 <strong>Contenido del bloque original!!</strong>
46             {% endblock %}
47         </div>
48     </div>
49 <footer>

```

### Otros ajustes

```

74 }
75 /*Estilos contenido central*/
76 #content{ /*Posicionar*/
77     width: 1212px;
78     min-height: 930px;
79     margin: 0 auto;
80     margin-bottom: 30px;
81     margin-top: 40px;
82 }
83 .box{
84     background: white;
85     min-height: 930px;
86     width: 95%;
87     padding: 20px;
88     border: 1px solid #ddd;
89     border-radius: 2px;
90     margin: 0 auto;
91 }
92 /*Estilos para title*/
93 .title{
94     color: #444;
95     letter-spacing: 1px;
96     font-size: 30px;
97     margin-bottom: 10px;
98     margin-top: 5px;
99 }

```



Usamos la **class="title"** para nuestros títulos de cada página, realizarlos los cambios y probar en el navegador.

```

22-django > AprendiendoDjango > miapp > templates > < > pagina.html
1  {% extends 'layout.html' %}
2
3  {% block title %}
4  Pagina de prueba para heredar
5  {% endblock %}
6
7  {% block content %}
8  <h1 class="title">Pagina de mi WEB</h1>
9  <p>Creado por ADSO SENA - 2023 </p>
10 {% endblock %}

```

```

22-django > AprendiendoDjango > miapp > templates > < > index.html
1  {% extends 'layout.html' %}
2
3  {% block title %}
4  Inicio del Sistema
5  {% endblock %}
6
7  {% block content %}
8  <h1 class="title"> Inicio</h1>
9  <p>ADSO SENA 2023 - Regional CAUCA:</p>
10 {% endblock %}

```

Estilos para el footer: y probar en el navegador

```

99  }
100 /*Estilos para el footer*/
101 footer{
102     width: 1250px;
103     background-color: #1b1e1f;
104     border: 1px solid #333333;
105     color: #d1d4d6;
106     text-align: center;
107     padding-top: 20px;
108     margin: 0 auto;
109     padding-bottom: 20px;
110     box-shadow: 0px 0px 20px gray;
111 }

```

## Lenguaje de plantillas y templates tags en Django

*Comentarios en plantillas:*

```

9  <p>ADSO SENA 2023 - Regional CAUCA:</p>
10 {% endblock %}
11
12 <!--Esto es un comentario para HTML-->
13
14 {% comment "Nota para el comentario" %}
15     <h1>lo que esta aqui tambien es un comentario</h1>
16 {% endcomment %}

```





### **Pasar datos desde la vista y mostrarlos en la plantilla:**

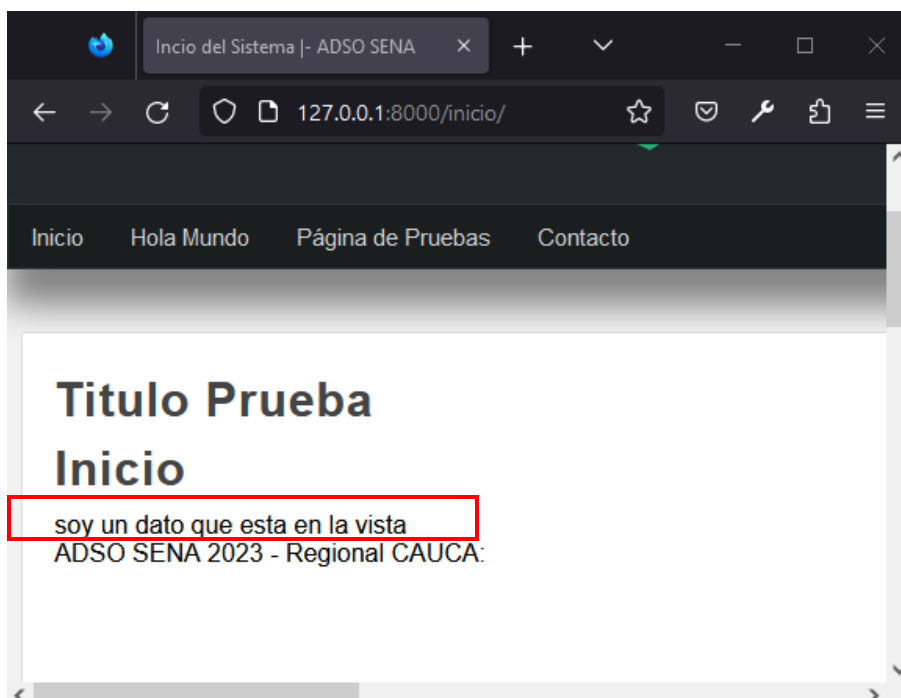
Desde el views.py se necesita pasar información al template para mostrarla y se necesita pasar desde el método o función

`return render(request, 'index.html', Aquí se pasa un diccionario con la información o las variables que quiero mostrar)`

```
30
31     return render(request, 'index.html', {
32         'mi_variable': 'soy un dato que esta en la vista'
33     })
34 # Create your views here.
35 #request es un parametro que permite recibir datos de una URL
```

Luego la llamamos en el index.html

```
7     {% block content %}
8     <h1 class="title"> Inicio</h1>
9     {% comment "interpolación" %}
10     mostrar los datos de una variable en un template
11     {% endcomment %}
12     {{mi_variable}}
13     <p>ADSO SENA 2023 - Regional CAUCA:</p>
14     {% endblock %}
```



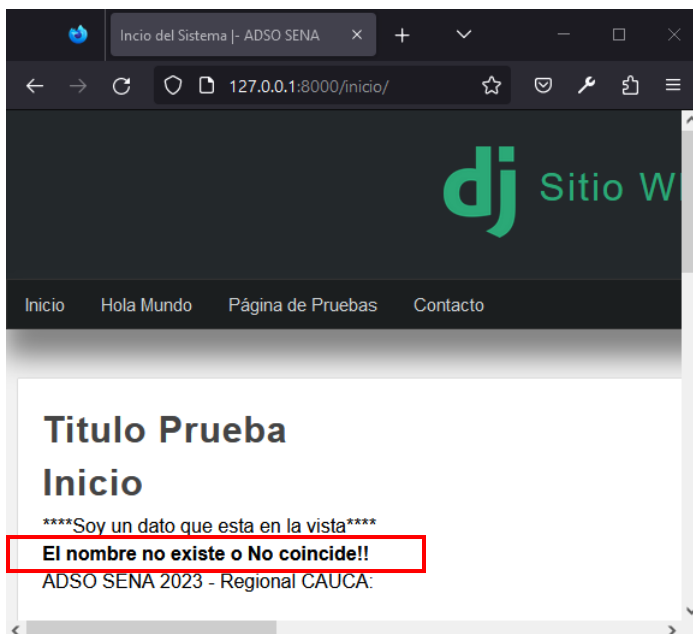


Condicionales – If templates Django: agregar en el `views.py`

```
28     year += 1
29     html += "</ul>"
30
31     nombre = 'Carlos Gomez'
32
33     return render(request, 'index.html', {
34         'mi_variable': '****Soy un dato que esta en la vista****',
35         'nombre': nombre
36     })
```

Y en `index.html` se genera el condicional para esa variable `nombre`

```
11 {% endcomment %}
12 <p>{{mi_variable}}</p>
13
14 <!--Condicionales-->
15 {% if nombre and nombre == 'Aprendiz ADSO' %}
16 <p>{{nombre}}</p>
17 {% else %}
18     <strong> El nombre no existe o No coincide!!</strong>
19 {% endif %}
20 <p>ADSO SENA 2023 - Regional CAUCA:</p>
```







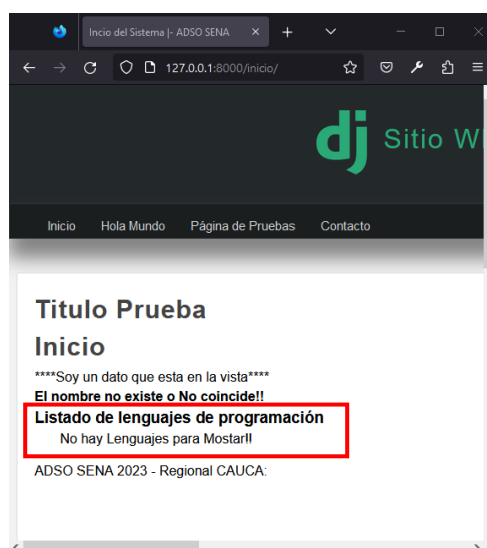
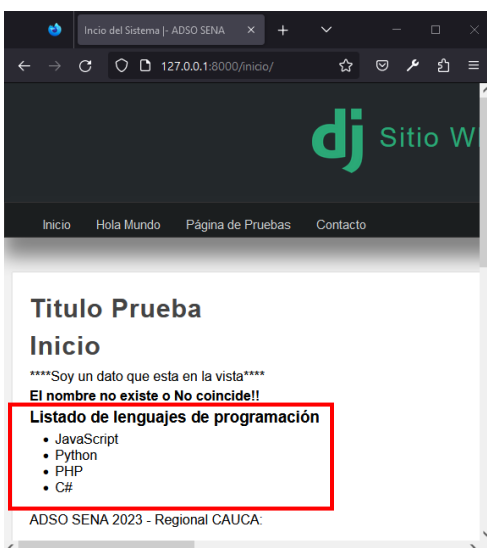
## Blucles – For template Django

Las variables de deben definir en las vistas para pasarlas luego a las templates, en **views.py**:

```
31     nombre = 'Carlos Gomez'
32
33     #Variable para bucles For utilizando listas
34     lenguajes = ['JavaScript', 'Python', 'PHP', 'C#']
35
36     return render(request, 'index.html', {
37         'mi_variable': '****Soy un dato que esta en la vista****',
38         'nombre': nombre,
39         'lenguajes': lenguajes,
40     })
41 # Create your views here.
```

En **index.html**

```
18     <strong> El nombre no existe o No coincide!!</strong>
19     {% endif %}
20     <!--Bucle For y recorrer la lista -->
21     <h3>Listado de lenguajes de programación</h3>
22     <ul>
23     {% for lenguaje in lenguajes %}
24         <li>{{lenguaje}}</li>
25     {% empty %}
26         <p>No hay Lenguajes para Mostar!!</p>
27     {% endfor %}
28     </ul>
29
30     <p>ADSO SENA 2023 - Regional CAUCA:</p>
```





### Includes dentro de templates Django:

Incluir una template dentro de otra template

Crear en templates un archivo **fecha-actual.html**

Y la incluyo en **index.html** y probar en el navegador

```

views.py  fecha-actual.html x  index.html
22-django > AprendiendoDjango > miapp > templates > fecha-actual.html
1  <br/>
2  <hr/>
3  <br/>
4  <h1> Bienvenido {{nombre}} </h1>
5  <h1> Estamos en el 2023 </h1>
6  <br/>
7  <hr/>
8  <br/>
9

```

```

views.py  fecha-actual.html  index.html x
22-django > AprendiendoDjango > miapp > templates > index.html
26  {% empty %}
27      <p>No hay Lenguajes para Mostar!!</p>
28  {% endfor %}
29  </ul>
30  <!--Esto es un include-->
31  {% include 'fecha-actual.html' %}
32
33  <p>ADSO SENA 2023 - Regional CAUCA:</p>

```

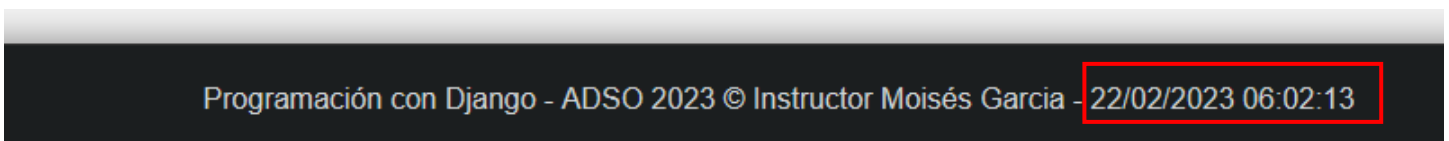
**URLs en Templates:** se abre el **layout.html** y se cambian las siguientes líneas de acuerdo a los **name** del archivo **urls.py**

```

layout.html x
22-django > AprendiendoDjango > miapp > templates > layout.html
21  
22  <h1> Sitio WEB con Django </h1>
23
24  </div>
25  </header>
26  <nav>
27  <ul>
28      <li>
29          <a href="{% url 'inicio' %}">Inicio</a>
30      </li>
31      <li>
32          <a href="{% url 'hola_mundo' %}">Hola Mundo</a>
33      </li>
34      <li>
35          <a href="{% url 'pagina' %}">Página de Pruebas</a>
36      </li>
37  </ul>

```

**Fechas:** colocar en el footer la fecha, abrimos la **layout.html** y colocamos la fecha que venga desde el servidor



```

layout.html x
22-django > AprendiendoDjango > miapp > templates > layout.html
46  </ul>
47  <footer>
48      Programación con Django - ADSO 2023 &copy; Instructor Moisés García - {% now "d/m/Y h:m:s"%}
49  </footer>
50  </body>
51  </html>

```