DJANGO

DJANGO ES UN FRAMEWORK WEB DE ALTO NIVEL QUE FOMENTA EL DESARROLLO RÁPIDO Y EL DISEÑO LIMPIO Y PRAGMÁTICO.

M6_Clase 04 . Creación de un sitio web (Parte 1)

Requerimiento

Queremos crear un sitio web con Django.

- Para ello, abrimos el VScode en nuestra carpeta general de proyectos.
- Abra el CMD y active el ambiente de Python adecuado (que tenga Django instalado).

  workon env_django

- Cree el proyecto a trabajar:

  django-admin startproject misitio

- Cámbiese a la carpeta del proyecto: cd misitio
- (Paso opcional) Active el servidor de Django python manage.py runserver

  y pruebe la conexión en un navegador. ¡Debe ver la imagen del cohete!

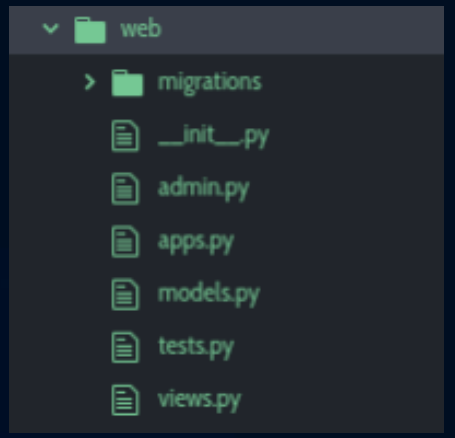
Requerimiento

- Ahora debe crear la aplicación llamada “web”. Para ello, se requiere ejecutar el siguiente comando, estando el terminal posicionado dentro de la carpeta del proyecto:

```
python manage.py startapp web
```

- Esto creará la estructura de archivos necesaria para la nueva aplicación dentro de nuestro proyecto.

- Si todo sale bien, no se verá ningún error al ejecutar el comando, y se creará una nueva carpeta llamada web dentro de nuestro proyecto, la cual contendrá los siguientes archivos:



Requerimiento

Configurando el settings.py

- Para agregar esta app a la lista de aplicaciones instaladas, debemos abrir el archivo settings.py presente en la carpeta de la app principal de nuestro proyecto, llamada misitio y agregarla a la lista de INSTALLED_APPS

```
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      'web',
41  ]
42
```

Requerimiento

Creando los templates

- Luego, dentro de la carpeta (de la app) web se debe crear una subcarpeta llamada templates y a continuación dentro de ella crear los archivos index.html, about.html y welcome.html

web/templates/index.html

```
<html>
  <body>
    indice
  </body>
</html>
```

web/templates/about.html

```
<html>
  <body>
    acerca
  </body>
</html>
```

web/templates/welcome.html

```
<html>
  <body>
    bienvenido cliente
  </body>
</html>
```

Requerimiento 2

Creando el archivo views.py

- Para habilitar las 3 urls distintas se debe crear en el archivo views.py dentro de la carpeta web una vista por cada url, cada una de las cuales retornará un render con los datos necesarios para desplegar cada uno de los archivos html creados anteriormente.

web/views.py

```python
from django.shortcuts import render

def indice(request):
    return render(request, 'index.html', {})

def acerca(request):
    return render(request, 'about.html', {})

def bienvenido(request):
    return render(request, 'welcome.html', {})

```

Requerimiento 2

Registrar vistas en urls.py

- Posteriormente debes registrar las vistas en el archivo urls.py para que dirijan a las rutas /, /acerca y /bienvenido respectivamente

misitio/urls.py

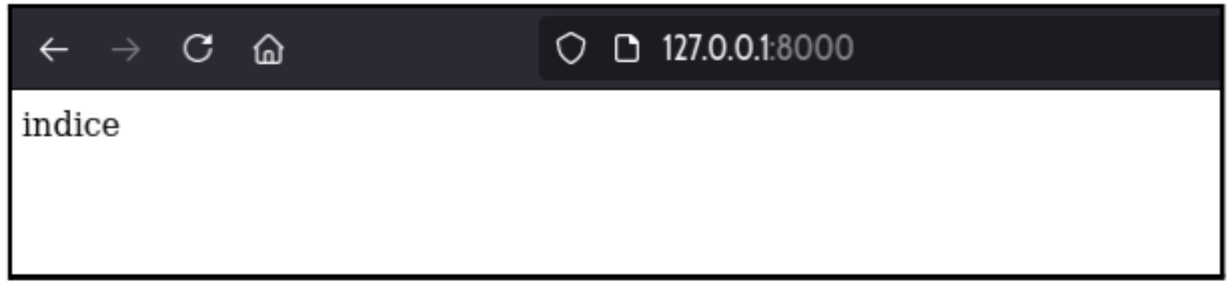
```
16  from django.contrib import admin
17  from django.urls import path
18  from web.views import indice, acerca, bienvenido
19
20  urlpatterns = [
21      path('admin/', admin.site.urls),
22      path('', indice, name="indice"),
23      path('acerca', acerca, name="acerca"),
24      path('bienvenido',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

Requerimiento

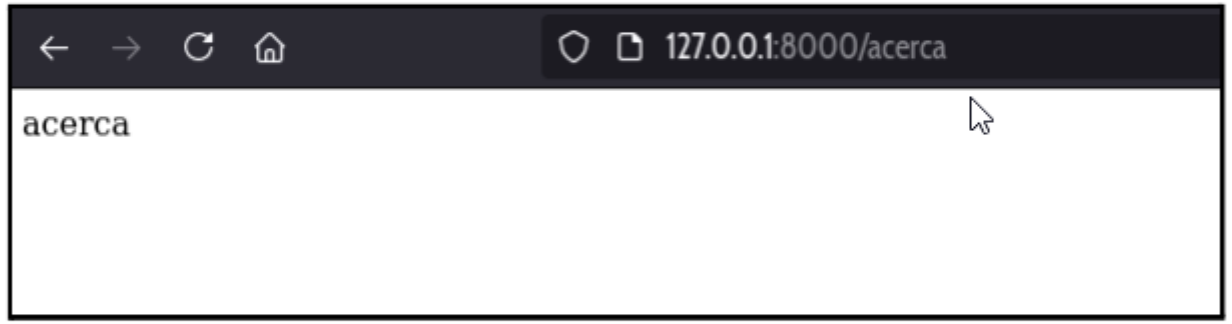
¿Cómo verificamos que las rutas funcionan?

- Al ejecutar python manage.py runserver se debe acceder a las rutas

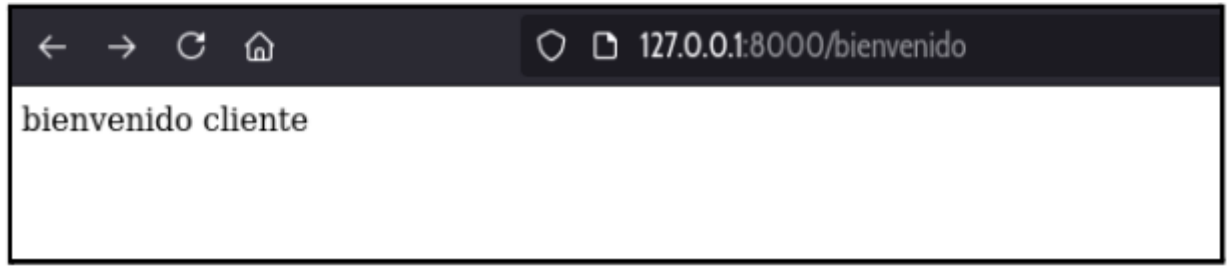
http://127.0.0.1:8000/



http://127.0.0.1:8000/acerca



http://127.0.0.1:8000/bienvenido



Requerimiento

Creando la plantilla base

- Para crear una plantilla base se debe crear un archivo llamado base.html en la carpeta web/templates que contenga la estructura general de la web a mostrar, usando blocks y marcando con fondos de colores (background) cada sección para mostrar de manera más explícita la separación entre éstas (opcional).

web/templates/base.html

```
<html>
  <body>

    {% block 'header' %}
    <div style="background: red;">
      header
    </div>
    {% endblock %}

    {% block 'navbar' %}
    <div style="background: yellow;">
      navbar
    </div>
    {% endblock %}

    {% block 'content' %}
    <div>
      contenido
    </div>
    {% endblock %}

    {% block 'footer' %}
    <div style="background: green;">
      footer
    </div>
    {% endblock %}
  </body>
</html>
```

Requerimiento

Conectando los templates con la plantilla base

- Una vez creado este archivo, debemos reemplazar el contenido de nuestros archivos index.html, about.html y welcome.html (como se muestra en las imágenes) de manera que extiendan de nuestra plantilla base (extends) y reemplacen el contenido (block content) por el contenido correspondiente a cada una de las vistas (mensajes).

```
{% extends 'base.html' %}

{% block 'content' %}
<div>
  MENSAJE
</div>
{% endblock %}
```

- Observe la notación a utilizar

```
1  {% extends 'base.html' %}
2
3  {% block 'content' %}
4  <div>
5    indice
6  </div>
7  {% endblock %}
```

web/templates/index.html

```
1  {% extends 'base.html' %}
2
3  {% block 'content' %}
4  <div>
5    acerca
6  </div>
7  {% endblock %}
```

web/templates/about.html

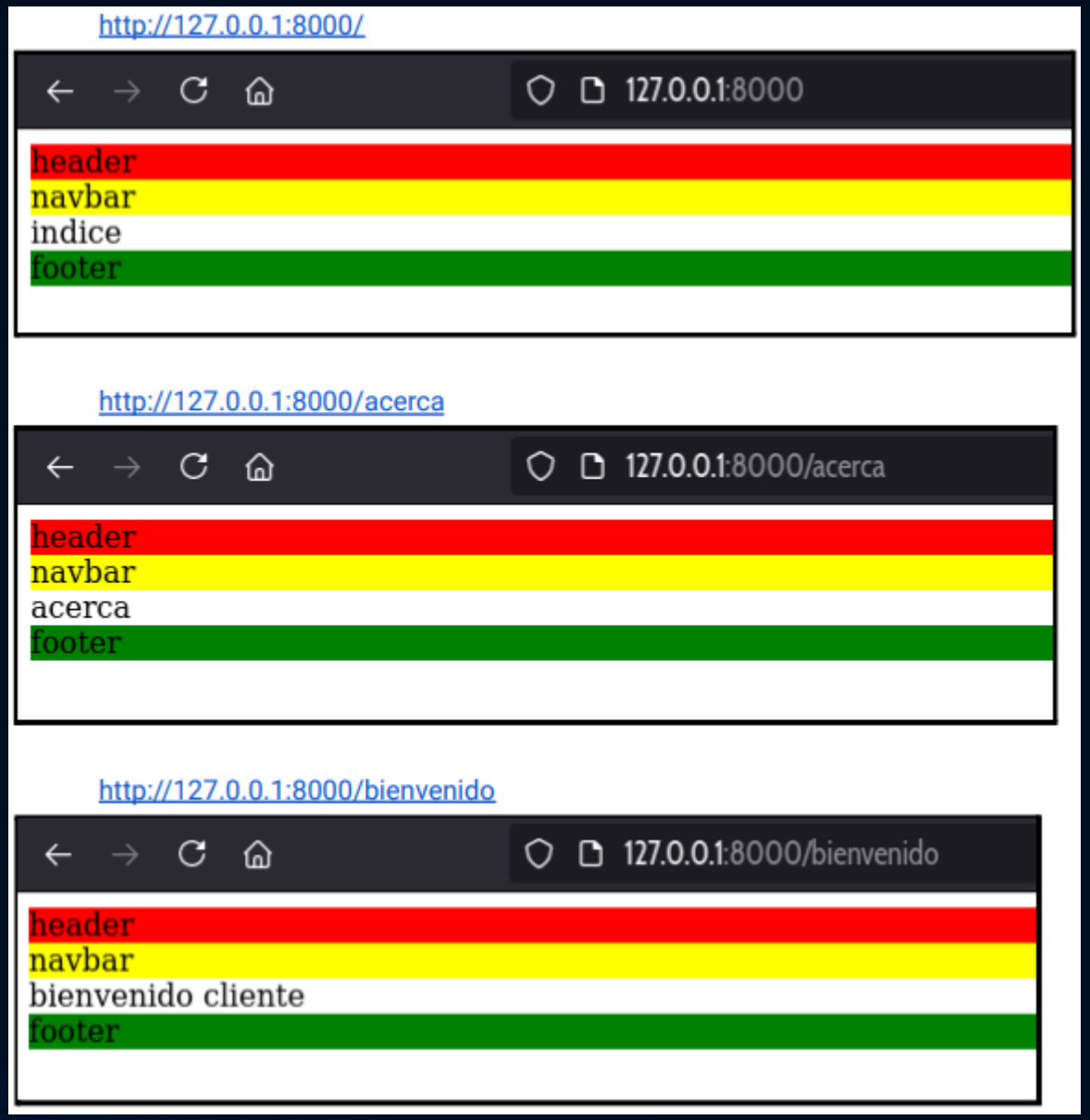
```
1  {% extends 'base.html' %}
2
3  {% block 'content' %}
4  <div>
5    bienvenido cliente
6  </div>
7  {% endblock %}
```

web/templates/welcome.html

Requerimiento

Testeando las rutas nuevamente

- Se debe acceder a las rutas y generar los “pantallazos” siguientes:



- No hay necesidad de volver a el comando runserver a no ser que hayamos cancelado o cerrado nuestra terminal, en tal caso podemos volver a ejecutar el proyecto utilizando el siguiente comando dentro de nuestro entorno virtual:

```
python manage.py runserver
```

Continuará ...