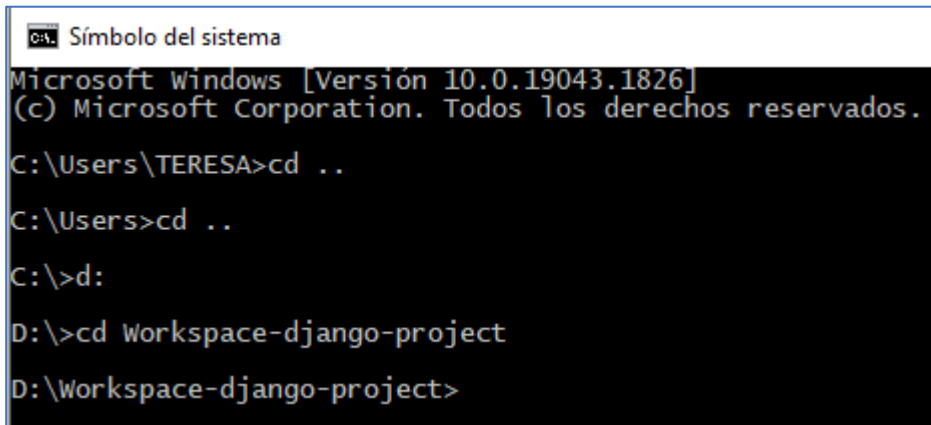


Laboratorio N°7: CREAR PROYECTO CON TEMPLATES

Crear otro Proyecto **TiendaVirtualTemplateDjango** y una aplicación **firstappTemplate**

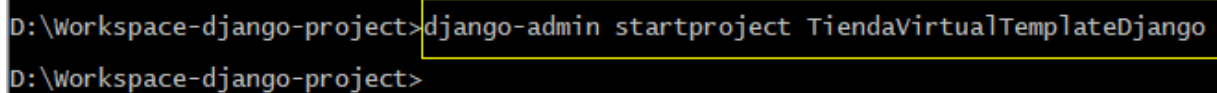
Pasos:

1. Abrir CMD
2. Ubicarse dentro del espacio de trabajo **workspace-django-project**



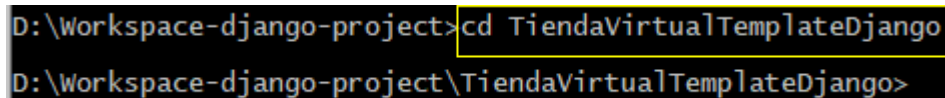
```
CA. Símbolo del sistema
Microsoft Windows [Versión 10.0.19043.1826]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\TERESA>cd ..
C:\Users>cd ..
C:\>d:
D:\>cd Workspace-django-project
D:\Workspace-django-project>
```

3. Crear el proyecto **TiendaVirtualTemplateDjango** ejecutando la siguiente instrucción:
django-admin startproject TiendaVirtualTemplateDjango



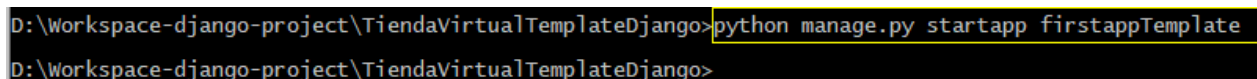
```
D:\Workspace-django-project>django-admin startproject TiendaVirtualTemplateDjango
D:\Workspace-django-project>
```

4. Entrar al proyecto **TiendaVirtualTemplateDjango**



```
D:\Workspace-django-project>cd TiendaVirtualTemplateDjango
D:\Workspace-django-project\TiendaVirtualTemplateDjango>
```

5. Crear la aplicación **firstappTemplate** ejecutando la siguiente instrucción:
python manage.py startapp firstappTemplate

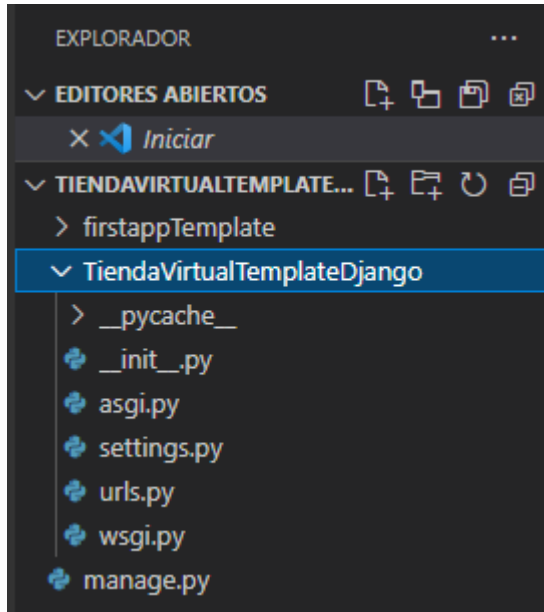


```
D:\Workspace-django-project\TiendaVirtualTemplateDjango>python manage.py startapp firstappTemplate
D:\Workspace-django-project\TiendaVirtualTemplateDjango>
```

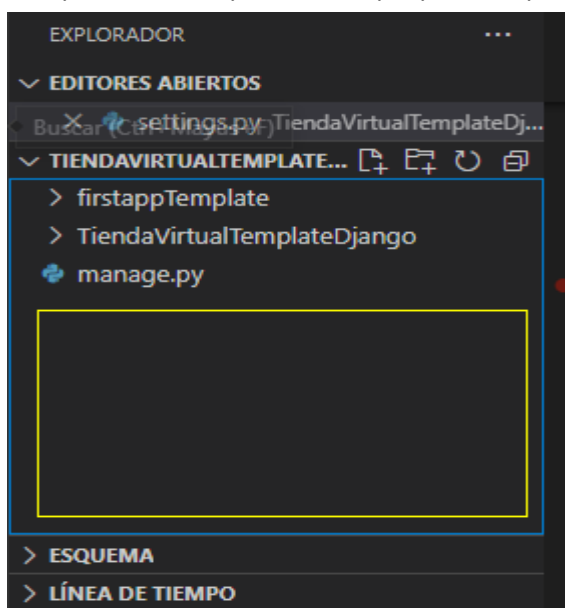
6. Abrir visual code

```
D:\Workspace-django-project\TiendaVirtualTemplateDjango>code .  
D:\Workspace-django-project\TiendaVirtualTemplateDjango>
```

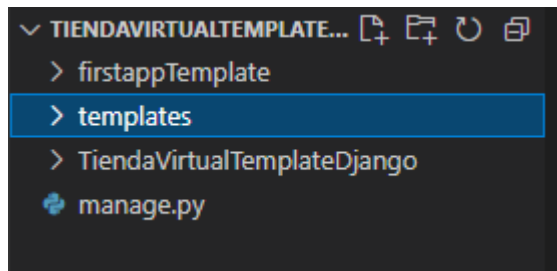
En la siguiente imagen, se puede visualizar el proyecto `TiendaVirtualTemplateDjango` y la aplicación `firstappTemplate`



7. En el proyecto `TiendaVirtualTemplateDjango`, crear la carpeta `templates` que almacenará los templates de las aplicaciones que posee el proyecto. Revisar la creación en la imagen siguiente.

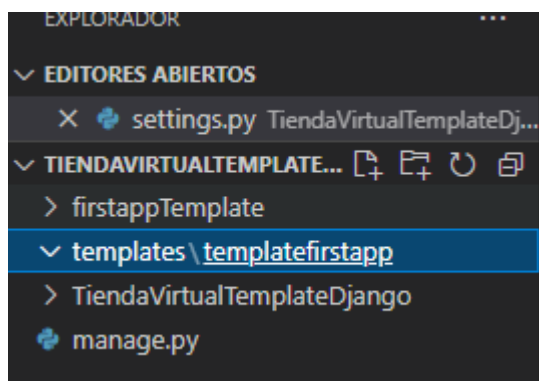
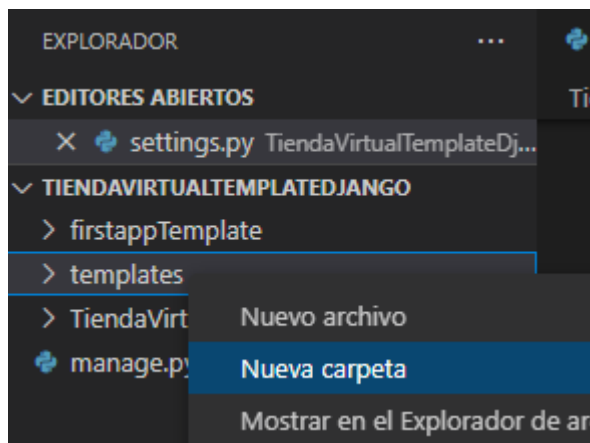
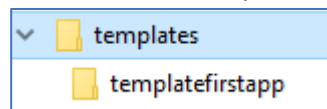


A partir de la imagen, hacer click derecho del mouse en la zona del recuadro amarillo y seleccionar Nueva carpeta para crear la carpeta templates.

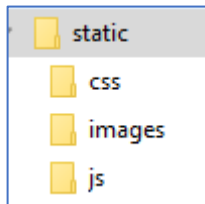


Nota: también se puede crear desde el CMD con la instrucción `mkdir`

8. Crear la carpeta **templatefirstapp** dentro de la carpeta **templates**. La carpeta **templatefirstapp** almacenará los templates de la aplicación **firstappTemplate**

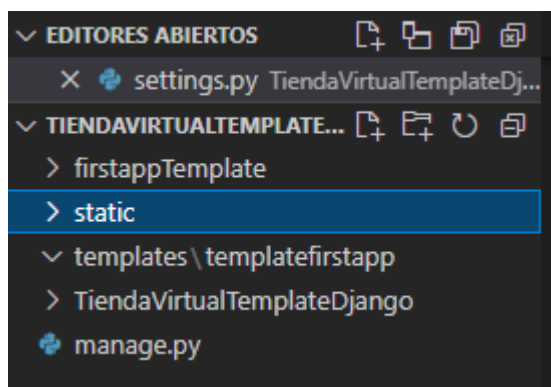
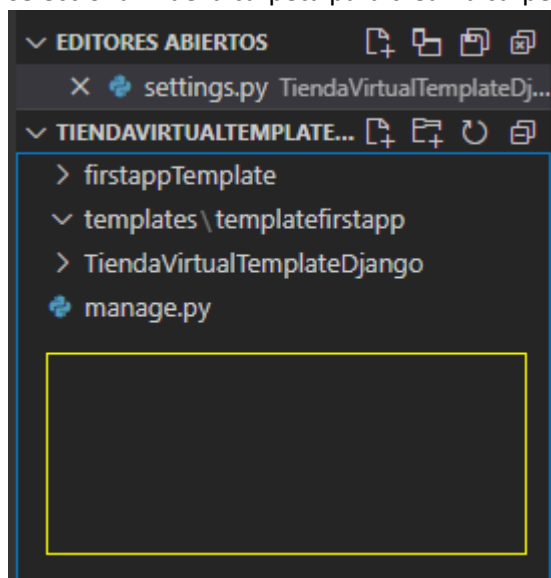


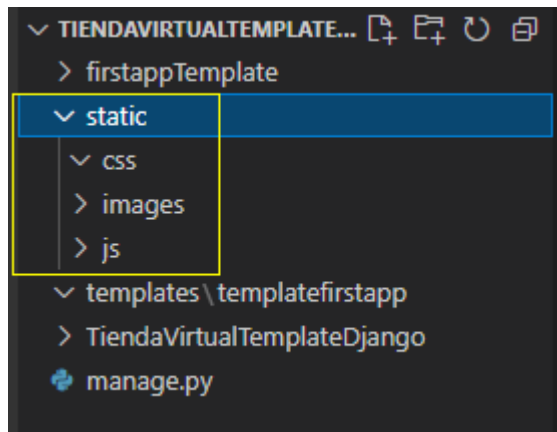
9. Para almacenar contenido estático crear la siguiente estructura de carpetas dentro del proyecto **TiendaVirtualTemplateDjango**



Nota: la carpeta static está a la misma altura de la carpeta templates. Se usan los mismos pasos de creación de carpetas del punto anterior.

A partir de la siguiente imagen, hacer click derecho del mouse en la zona del recuadro amarillo y seleccionar Nueva carpeta para crear la carpeta static.





Nota: para crear las carpetas css, js e images hacer click derecho del mouse sobre la palabra static.

10. Modificar el archivo de configuración `settings.py` del proyecto `TiendaVirtualTemplateDjango`.

Las modificaciones son:

a) Incluir la aplicación `firstappTemplate` al Proyecto `TiendaVirtualTemplateDjango`

```
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     'firstappTemplate'
41 ]
```

b) Indicar la ruta de los templates del Proyecto `TiendaVirtualTemplateDjango`

Las instrucciones son:

i. Incluir import os

```
12 import os
13 from pathlib import Path
14
```

ii. Crear la variable `TEMPLATE_DIR` con la ruta de la carpeta templates del Proyecto y agregarla

```
15 # Build paths inside the project like this: BASE_DIR / 'subdir'.
16 BASE_DIR = Path(__file__).resolve().parent.parent
17 TEMPLATE_DIR=os.path.join(BASE_DIR, 'templates')
```

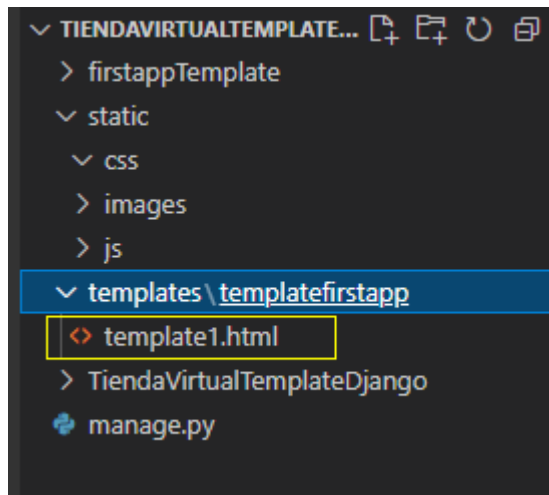
- iii. Especificar la variable `TEMPLATE_DIR` en el código que se indica en el recuadro amarillo

```
55  ∨ TEMPLATES = [  
56  ∨      {  
57      'BACKEND': 'django.template.backends.django.DjangoTemplates',  
58      'DIRS': [],  
59      'APP_DIRS': True,  
60      'OPTIONS': {  
61      'context_processors': [  
62          'django.template.context_processors.debug',  
63          'django.template.context_processors.request',  
64          'django.contrib.auth.context_processors.auth',  
65          'django.contrib.messages.context_processors.messages',  
66      ],  
67      },  
68  },  
69  ]
```

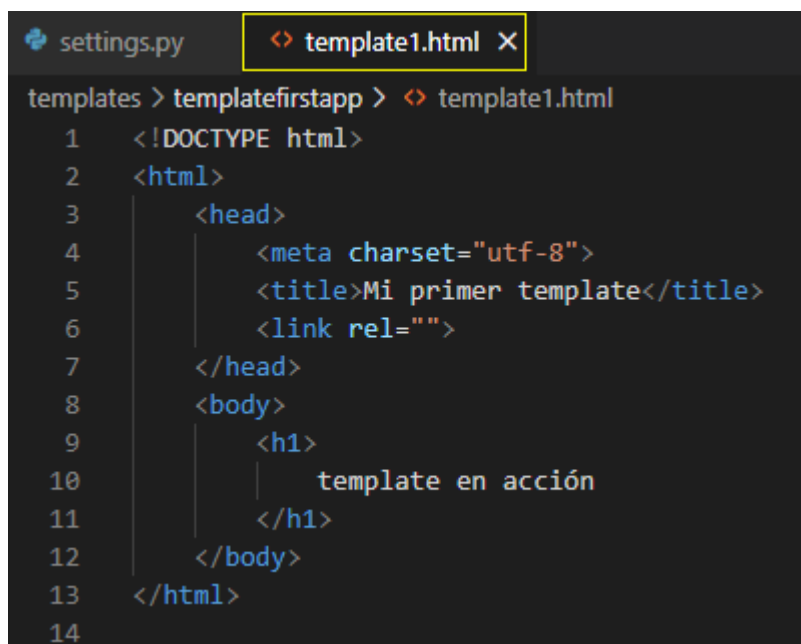
El resultado es:

```
55  TEMPLATES = [  
56  {  
57      'BACKEND': 'django.template.backends.django.DjangoTemplates',  
58      'DIRS': [TEMPLATE_DIR],  
59      'APP_DIRS': True,  
60      'OPTIONS': {  
61      'context_processors': [  
62          'django.template.context_processors.debug',  
63          'django.template.context_processors.request',  
64          'django.contrib.auth.context_processors.auth',  
65          'django.contrib.messages.context_processors.messages',  
66      ],  
67      },  
68  },  
69  ]
```

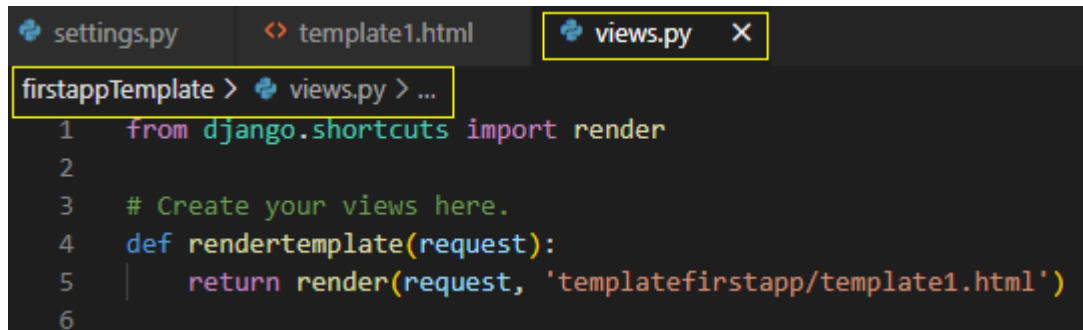
11. Crear un template en el Proyecto, específicamente en la carpeta `templates/templatefirstapp`. Esto es crear un archivo html con el nombre `template1.html` (puede ser cualquier nombre).



12. Editar el archivo `template1.html` e ingresar el siguiente código

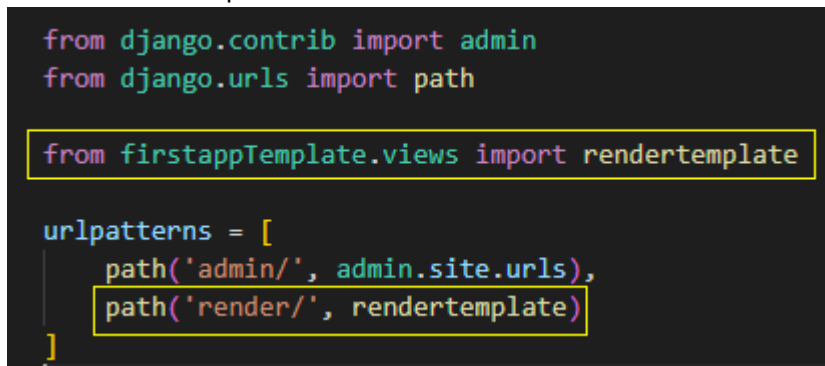


13. Crear la vista para el template1.html. Editar el archivo views.py de la aplicación `firstappTemplate` y agregar la función `rendertemplate` como se indica en la imagen.



```
settings.py  template1.html  views.py X
firstappTemplate > views.py > ...
1  from django.shortcuts import render
2
3  # Create your views here.
4  def rendertemplate(request):
5      return render(request, 'templatefirstapp/template1.html')
6
```

14. Modificar el archivo `urls.py` del proyecto `TiendaVirtualTemplateDjango` e incluir el path a la función `rendertemplate`-

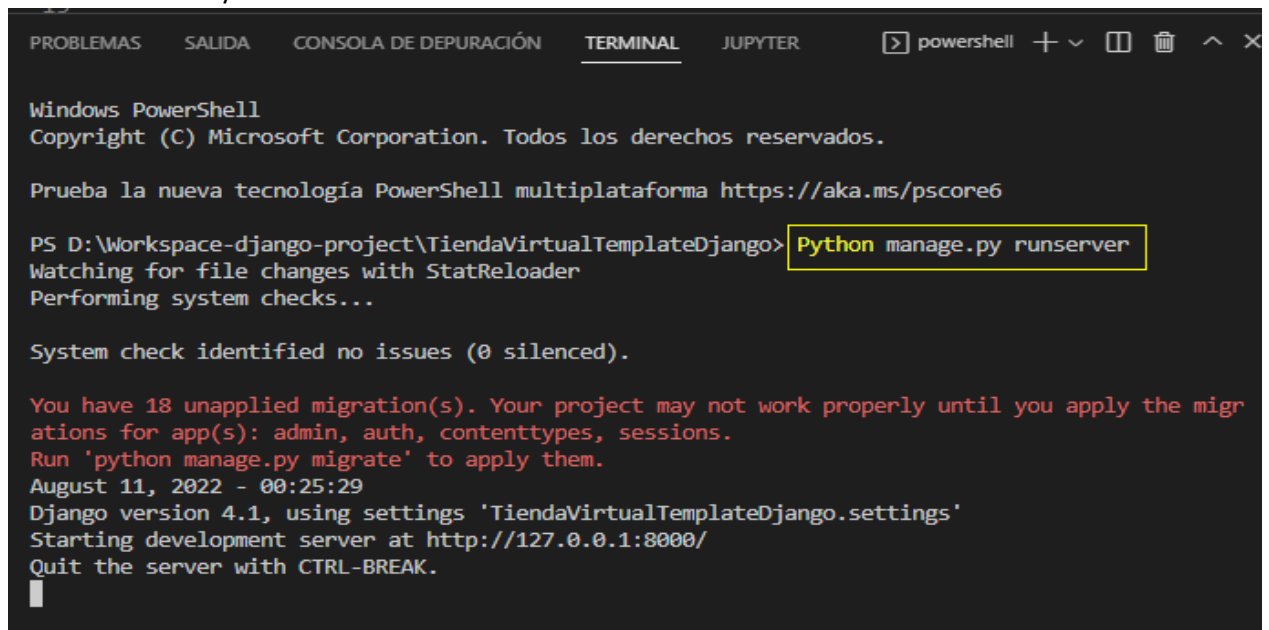


```
from django.contrib import admin
from django.urls import path

from firstappTemplate.views import rendertemplate

urlpatterns = [
    path('admin/', admin.site.urls),
    path('render/', rendertemplate)
]
```

15. Iniciar el servidor y revisar. Abrir el terminal del visual code



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  JUPYTER  powershell + v [ ] [X] ^ X
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS D:\Workspace-django-project\TiendaVirtualTemplateDjango> 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.
August 11, 2022 - 00:25:29
Django version 4.1, using settings 'TiendaVirtualTemplateDjango.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
█
```

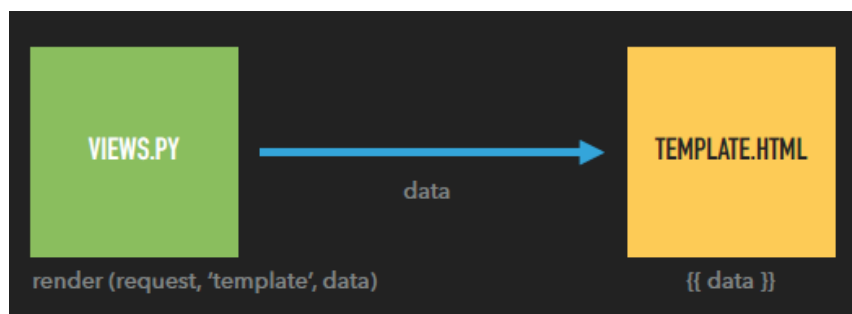

Digitar en el navegador render/



Laboratorio N°8: INCLUIR DATA DINAMICA EN EL TEMPLATE

USO DE TEMPLATES TAGS (DTL)

- ▶ `{{ data }}`
- ▶ `{% tagName %}`
 - ▶ `{% csrf_token %}`
 - ▶ `{% x extends y %}`
 - ▶ `{% if %} {% endif %}`
 - ▶ `{% for %} {% endfor %}`



1. Crear otro template en el Proyecto, específicamente en la carpeta `templates/templatefirstapp`. Esto es crear un archivo html con el nombre `template2.html` (puede ser cualquier nombre).

```
templates > templatefirstapp > <> template2.html
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Mi segundo template</title>
6          <link rel="">
7      </head>
8      <body>
9          <h1>
10         template en acción
11     </h1>
12 </body>
13 </html>
```

2. Editar el `template2.html` e incluir la etiqueta `{{nombre}}` donde nombre es una variable enviada desde views.

```
templates > templatefirstapp > <> template2.html
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Mi segundo template</title>
6          <link rel="">
7      </head>
8      <body>
9          <h1>
10         template en acción
11     </h1>
12     <h1>Hola {{nombre}} !!</h1>
13 </body>
14 </html>
```

3. Modificar el archivo `views.py` y agregar una nueva función `rendertemplatedata`

```
firstappTemplate > views.py > ...
1  from django.shortcuts import render
2
3  # Create your views here.
4  def rendertemplate(request):
5      return render(request, 'templatefirstapp/template1.html')
6
7  def rendertemplatedata(request):
8      data={"nombre": "Tere"}
9      return render(request, 'templatefirstapp/template2.html',data)
10
```

4. Modificar el archivo `urls.py` del proyecto

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

from firstappTemplate.views import rendertemplate, rendertemplatedata

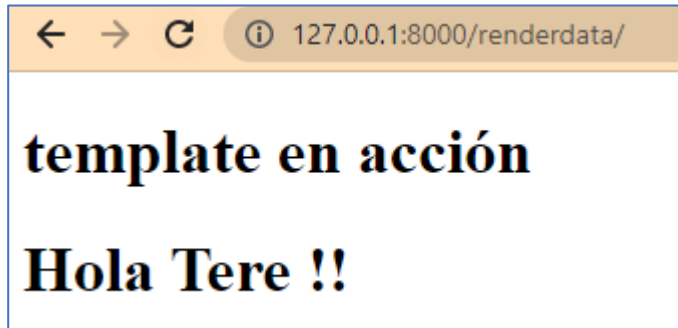
urlpatterns = [
    path('admin/', admin.site.urls),
    path('render/', rendertemplate),
    path('renderdata/', rendertemplatedata)
]
```

5. Revisar las vistas a través del navegador

/render



/renderdata



Laboratorio N°9: INCLUIR DATA DINAMICA EN EL TEMPLATE (parte 2)

EJERCICIO TEMPLATES

- ▶ Agregar una vista llamada **infoUsuario** que despliegue la información de un usuario utilizando un template llamado **userInfoTemplate.html** en la ruta <http://127.0.0.1:8000/info/> como se muestra en la imagen.



1. Crear otro template en el Proyecto, específicamente en la carpeta `templates/templatefirstapp`. Esto es crear un archivo html con el nombre `userInfoTemplate.html`

```
templates > templatefirstapp > <> userInfoTemplate.html
1
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta charset="utf-8">
6     <title>mi primer template</title>
7   </head>
8   <body>
9     <h1>
10      Información del usuario
11    </h1>
12    <table>
13      <tr>
14        <td>ID</td>
15        <td>:</td>
16        <td>{{id}}</td>
17      </tr>
18      <tr>
19        <td>Nombre</td>
20        <td>:</td>
21        <td>{{nombre}}</td>
22      </tr>
23      <tr>
24        <td>Email</td>
25        <td>:</td>
26        <td>{{email}}</td>
27      </tr>
28    </table>
29  </body>
30 </html>
```

2. Modificar el archivo `views.py` y agregar una nueva función `infoUsuario`

```
firstappTemplate > views.py > ...
1  from django.shortcuts import render
2
3  # Create your views here.
4  def rendertemplate(request):
5      return render(request, 'templatefirstapp/template1.html')
6
7  def rendertemplatedata(request):
8      data={"nombre": "Tere"}
9      return render(request, 'templatefirstapp/template2.html',data)
10
11  def infoUsuario(request):
12      data={"id": "123","nombre": "Teresa Tapia", "email": "teresa.tapia02@hotmail.com"}
13      return render(request, 'templatefirstapp/userInfoTemplate.html',data)
```

3. Modificar el archivo `urls.py` del proyecto

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

from firstappTemplate.views import infoUsuario, rendertemplate, rendertemplatedata

urlpatterns = [
    path('admin/', admin.site.urls),
    path('render/', rendertemplate),
    path('renderdata/', rendertemplatedata),
    path('info/', infoUsuario)
]
```

4. Inicial el servidor y revisar las vistas a través del navegador

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  JUPYTER
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS D:\Workspace-django-project\TiendaVirtualTemplateDjango> 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, contentt
ypes, sessions.
Run 'python manage.py migrate' to apply them.
August 11, 2022 - 08:27:47
Django version 4.1, using settings 'TiendaVirtualTemplateDjango.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Al revisar info/



Laboratorio N°10: INCLUIR CONTENIDO ESTÁTICO AL PROYECTO (IMAGES, CSS, JS, ETC)

1. Modificar el archivo de configuración `settings.py` del proyecto `TiendaVirtualTemplateDjango`. Las modificaciones son:

- a. Revisar la existencia del `import`, sino agregarlo

```
12  import os
13  from pathlib import Path
14
```

- b. Revisar la existencia de la variable `STATIC_URL` con la asignación de la carpeta `static` del Proyecto, sino agregarla

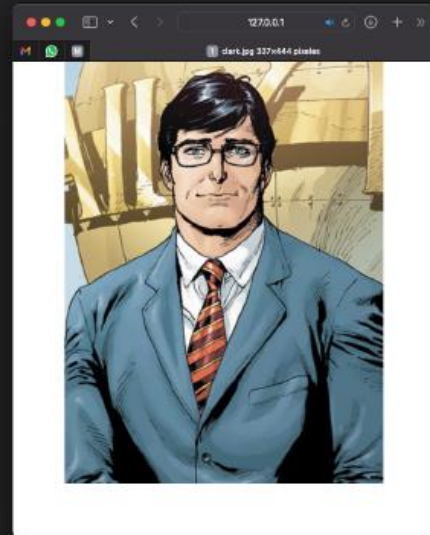
```
116  # Static files (CSS, JavaScript, Images)
117  # https://docs.djangoproject.com/en/4.1/howto/static-files/
118
119  STATIC_URL = 'static/'
```

- c. Especificar la variable `STATICFILES_DIRS` en el código como se indica en el recuadro amarillo. En otras palabras, agregar la variable `STATICFILES_DIRS`

```
116  # Static files (CSS, JavaScript, Images)
117  # https://docs.djangoproject.com/en/4.1/howto/static-files/
118
119  STATIC_URL = 'static/'
120  STATICFILES_DIRS=[os.path.join(BASE_DIR, 'static')]
```

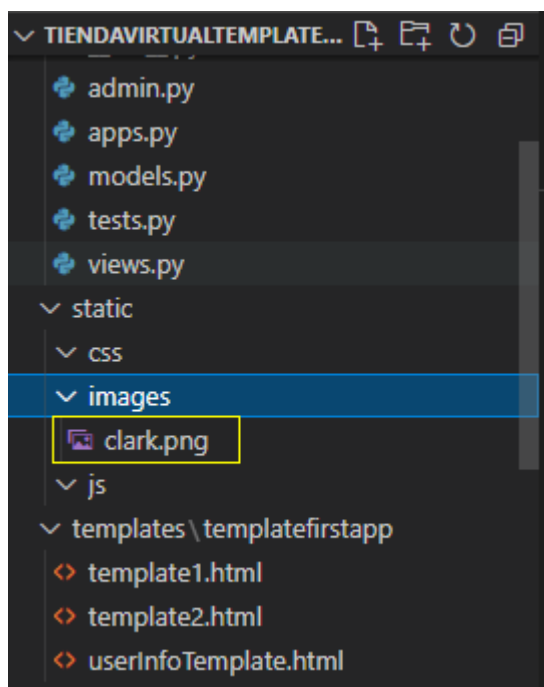
STATIC FILES

- ▶ Agreguemos una imagen a **static/images/**
- ▶ Podemos verificar el contenido accediendo a **<http://127.0.0.1:8000/static/images/clark.jpg>**

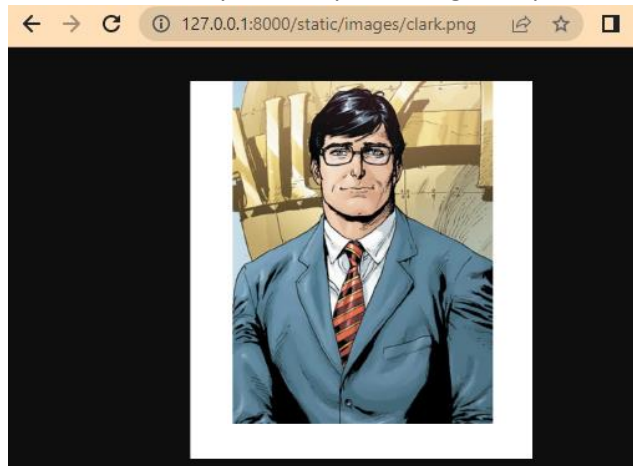


<http://127.0.0.1:8000/static/images/clark.jpg>

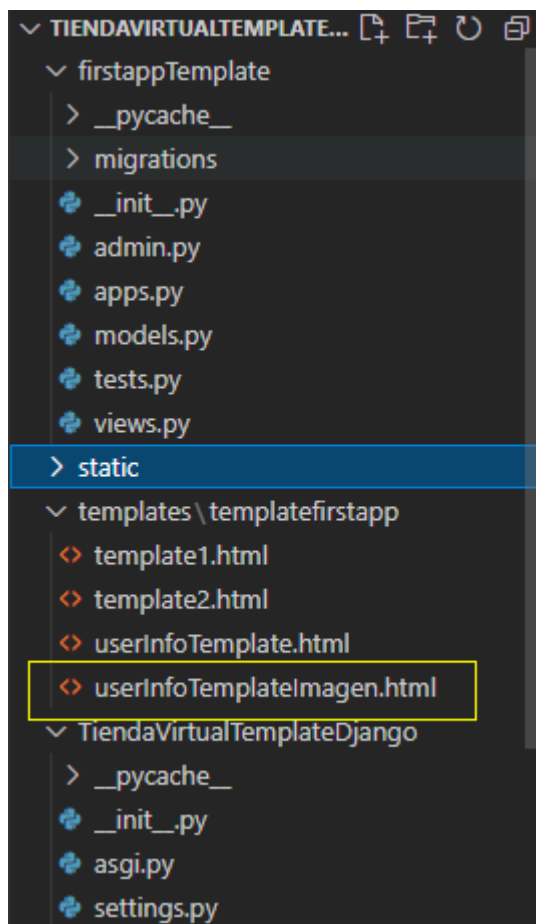
2. Agregar una imagen a la carpeta **static/images**



3. Guardar cambios y revisar que la imagen se puede ver por el navegador



5. Crear otro template en el Proyecto, específicamente en la carpeta `templates/templatefirstapp`. Esto es crear un archivo html con el nombre `userInfoTemplateImagen.html`



6. Editar el template `userInfoTemplateImagen.htm`. Incluir lo especificado en los recuadros amarillos.

```
templates > templatefirstapp > <> userInfoTemplateImagen.html
1  <!DOCTYPE html>
2  {% load static %}
3  <html>
4      <head>
5          <meta charset="utf-8">
6          <title>mi primer template</title>
7      </head>
8      <body>
9          <h1>
10             Información del usuario
11          </h1>
12          
13          <table>
14              <tr>
15                  <td>ID</td>
16                  <td>:</td>
17                  <td>{{id}}</td>
18              </tr>
19              <tr>
20                  <td>Nombre</td>
21                  <td>:</td>
22                  <td>{{nombre}}</td>
23              </tr>
24              <tr>
25                  <td>Email</td>
26                  <td>:</td>
27                  <td>{{email}}</td>
28              </tr>
29          </table>
30      </body>
31  </html>
```

Nota 1: {% load static%} se utiliza para incorporar contenidos estáticos en el html

Nota 2: para especificar la imagen utilizar {% static "images/clark.png" %}

STATIC FILES

- Para agregar la imagen al template agregamos:

```
<!DOCTYPE html>
{% load static %}

<html>
  <head>
    <meta charset="utf-8">
    <title>Información de Usuario</title>
  </head>
  <body>
    <h1>Información del Usuario</h1>
    
    <ul>
      <li>ID: {{id}}</li>
      <li>Nombre: {{nombre}}</li>
      <li>Email: {{email}}</li>
    </ul>
  </body>
</html>
```

← Incorporamos los
archivos estáticos

← Usamos la
referencia dentro
de la imagen



http://127.0.0.1:8000/info/

7. Modificar el archivo `views.py` y agregar una nueva función `infoUsuarioImagen`

```
firstappTemplate > views.py > infoUsuarioImagen
1  from django.shortcuts import render
2
3  # Create your views here.
4  def rendertemplate(request):
5      return render(request, 'templatefirstapp/template1.html')
6
7  def rendertemplatedata(request):
8      data={"nombre": "Tere"}
9      return render(request, 'templatefirstapp/template2.html',data)
10
11 def infoUsuario(request):
12     data={"id": "123","nombre": "Teresa Tapia", "email": "teresa.tapia02@hotmail.com"}
13     return render(request, 'templatefirstapp/userInfoTemplate.html',data)
14
15 def infoUsuarioImagen(request):
16     data={"id": "123","nombre": "Teresa Tapia", "email": "teresa.tapia02@hotmail.com"}
17     return render(request, 'templatefirstapp/userInfoTemplateImagen.html',data)
```

8. Modificar el archivo `urls.py` del proyecto

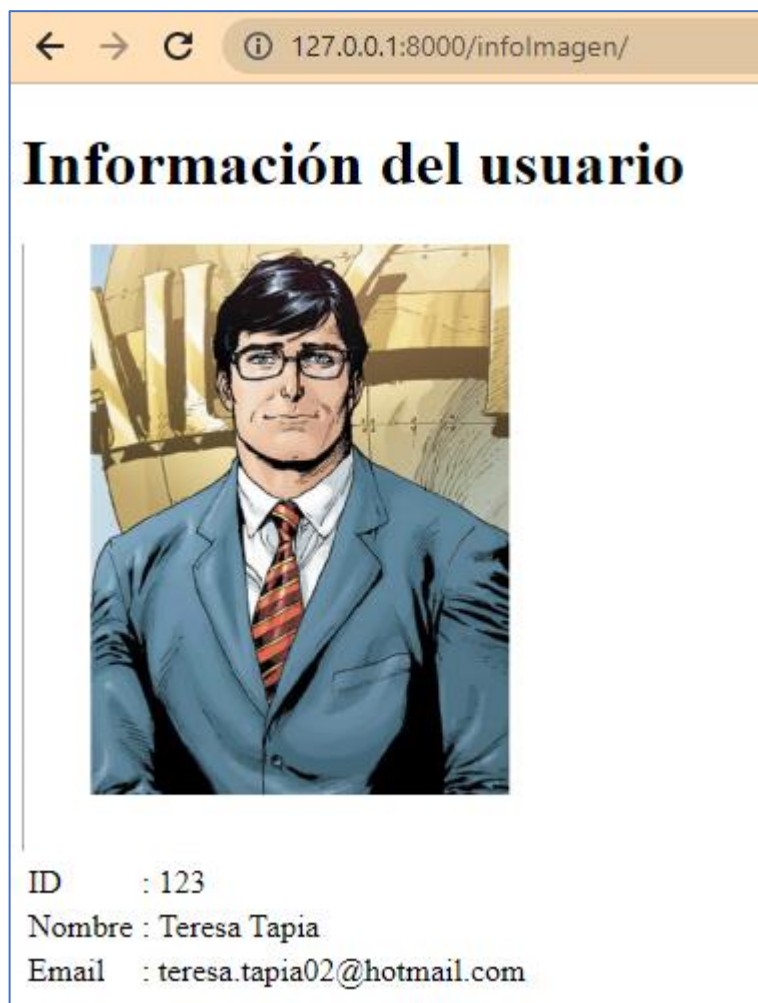
```
from django.contrib import admin
from django.urls import path

from firstappTemplate.views import infoUsuario, infoUsuarioImagen, rendertemplate, rendertemplatedata

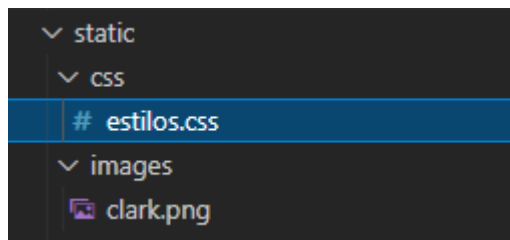
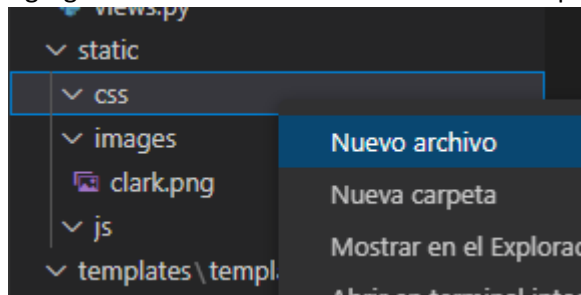
urlpatterns = [
    path('admin/', admin.site.urls),
    path('render/', rendertemplate),
    path('renderdata/', rendertemplatedata),
    path('info/', infoUsuario),
    path('infoImagen/', infoUsuarioImagen)
]
```

9. Iniciar el servidor y revisar las vistas a través del navegador

infolImagen/



10. Agregar un archivo **estilos.css** dentro de la carpeta **static/css** e incorporarlo al proyecto.



11. Editar el archivo **estilos.css** y agregar el estilo que se detalla en la imagen

```
static > css > # estilos.css > ...
1  body {
2    background-color: green;
3  }
4
5  .titulo{
6    color: white;
7    text-align: center;
8  }
9
10 .foto{
11   border: 1px solid black;
12   display: block;
13   margin:auto;
14 }
15
16 .data{
17   font-size: 2em;
18   color: aquamarine;
19 }
```

12. Editar el template `userInfoTemplateImagen.htm` e incluir lo especificado en los recuadros amarillos.

```
templates > templatefirstapp > <> userInfoTemplateImagen.html
1  <!DOCTYPE html>
2  {% load static %}
3  <html>
4      <head>
5          <meta charset="utf-8">
6          <title>mi primer template</title>
7          <link rel="stylesheet" href="{% static 'css/estilos.css' %}">
8      </head>
9      <body>
10         <h1 class="titulo">
11             Información del usuario
12         </h1>
13         
14         <table class="data">
15             <tr>
16                 <td>ID</td>
17                 <td>:</td>
18                 <td>{{id}}</td>
19             </tr>
20             <tr>
21                 <td>Nombre</td>
22                 <td>:</td>
23                 <td>{{nombre}}</td>
24             </tr>
25             <tr>
26                 <td>Email</td>
27                 <td>:</td>
28                 <td>{{email}}</td>
29             </tr>
30         </table>
31     </body>
32 </html>
```

13. Iniciar el servidor y revisar las vistas a través del navegador
infolmagen/

