

2 Creando las páginas web de la app

Para este tutorial puedes bajar el sitio estático **cine proyecto html.rar** desde Classroom en la **semana08, lo descomprimes y luego dentro de la carpeta cine, haces click sobre index.html para comenzar a navegar**, lo necesitarás para copiar información, ya que trataremos de traspasar ese sitio como proyecto django desde cero en tu computador.

Dentro de nuestra app 'core', necesitamos crear una subcarpeta denominada: templates y dentro de esta otra subcarpeta denominada igual que la app, en nuestro caso core

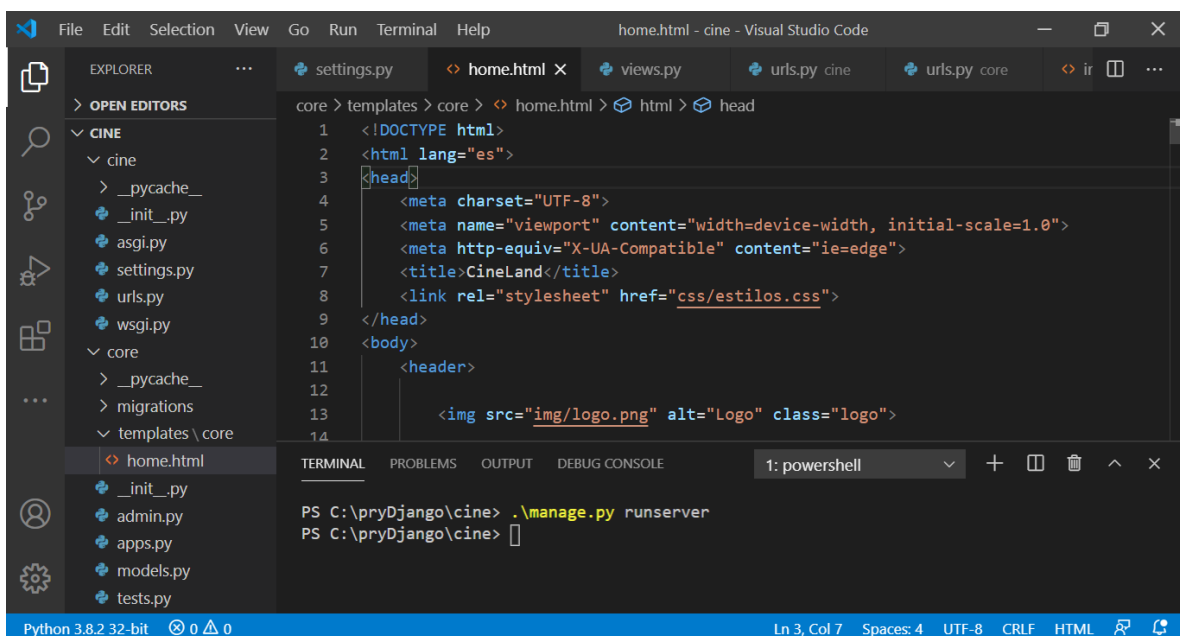
En esta carpeta quedarán alojados los archivos web del sitio (html, css, js, imágenes, etc).

Vamos a extraer los códigos del sitio estático de ejemplo, para este caso incorporaremos el index.html de nuestro sitio para que lo reconozca django como parte de la app.

La primera página

Creamos un nuevo archivo que llamaremos home.html y copiaremos y pegaremos todo el código fuente del archivo index.html de nuestro sitio estático de ejemplo, quedando la ruta:

cine/core/templates/core/home.html



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the project structure with folders 'CINE' and 'core', and a file 'home.html' selected under 'core/templates/core'. The Editor panel shows the content of 'home.html' with the following code:

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>CineLand</title>
8   <link rel="stylesheet" href="css/estilos.css">
9 </head>
10 <body>
11   <header>
12
13     
14
```

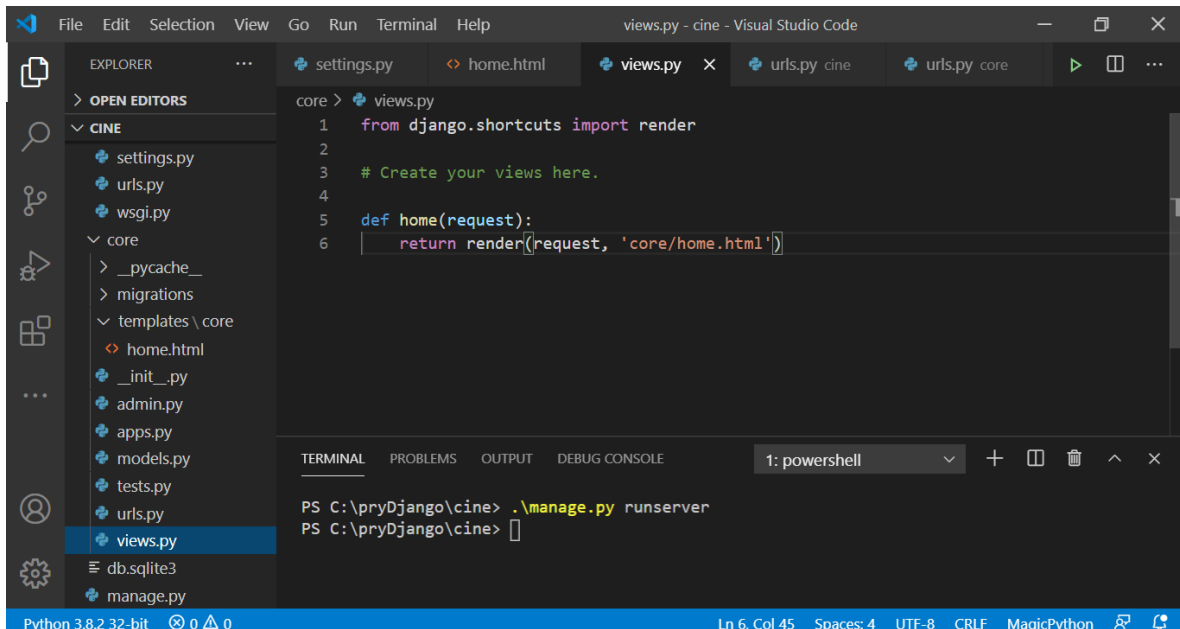
The Terminal panel at the bottom shows the command prompt with the following commands and output:

```
PS C:\pryDjango\cine> .\manage.py runserver
PS C:\pryDjango\cine>
```

The status bar at the bottom indicates the Python version (3.8.2 32-bit), the current file (home.html), and the encoding (UTF-8).

Para que se pueda ver esta página, django requiere 2 capas más: vista y el template ¿Cómo funciona el llamado interno?, la URL --> view --> template

Primero modificamos el archivo *core/views.py* y le definimos una función (líneas 5 y 6 de la imagen) para que al llamar al home(página principal) se vaya a buscar donde estarán nuestras vistas



The screenshot shows the Visual Studio Code interface with the Django project structure on the left and the *core/views.py* file open in the editor. The Explorer pane shows the project structure with the *core* folder expanded, showing files like *__init__.py*, *admin.py*, *apps.py*, *models.py*, *tests.py*, *urls.py*, *views.py*, *db.sqlite3*, and *manage.py*. The *core* folder is expanded, showing subfolders like *_pycache_*, *migrations*, and *templates*, and files like *home.html*. The *views.py* file is open in the editor, showing the following code:

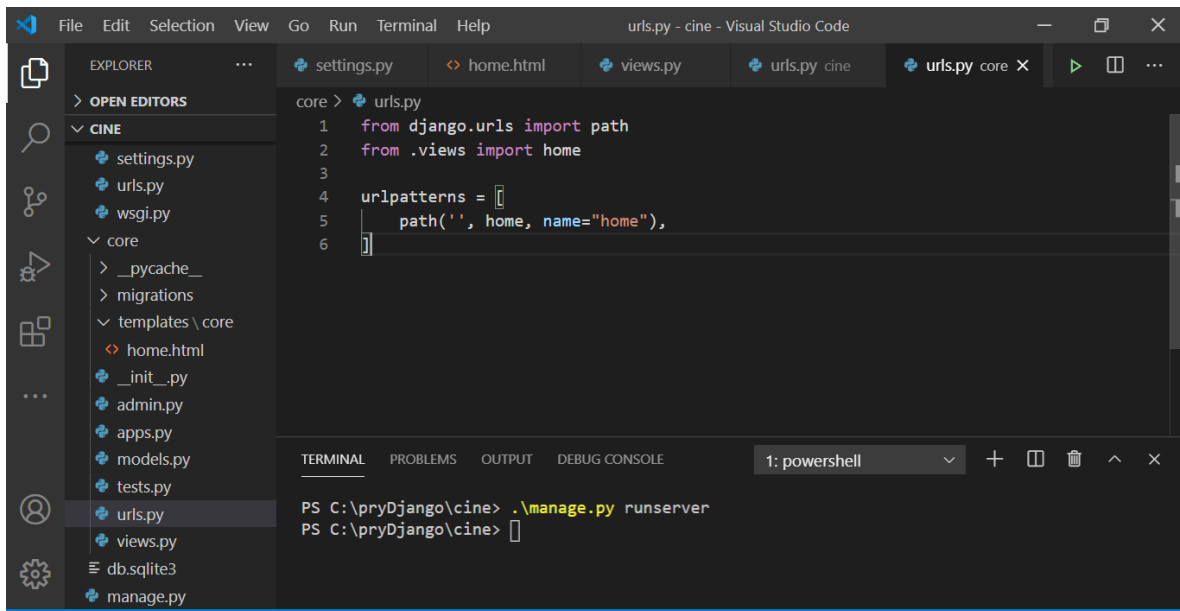
```
core > views.py
1  from django.shortcuts import render
2
3  # Create your views here.
4
5  def home(request):
6      return render(request, 'core/home.html')
```

The Terminal pane at the bottom shows the command prompt with the following commands and output:

```
PS C:\pryDjango\cine> .\manage.py runserver
PS C:\pryDjango\cine>
```

Segundo: necesitamos dar el permiso para la URL, por lo tanto, necesitamos crear el archivo *urls.py* dentro de *core* de nuestro proyecto, o sea: *cine/core/urls.py* y le añadimos las líneas que están a continuación (se pueden copiar parte de ellas desde archivo *cine/urls.py*).

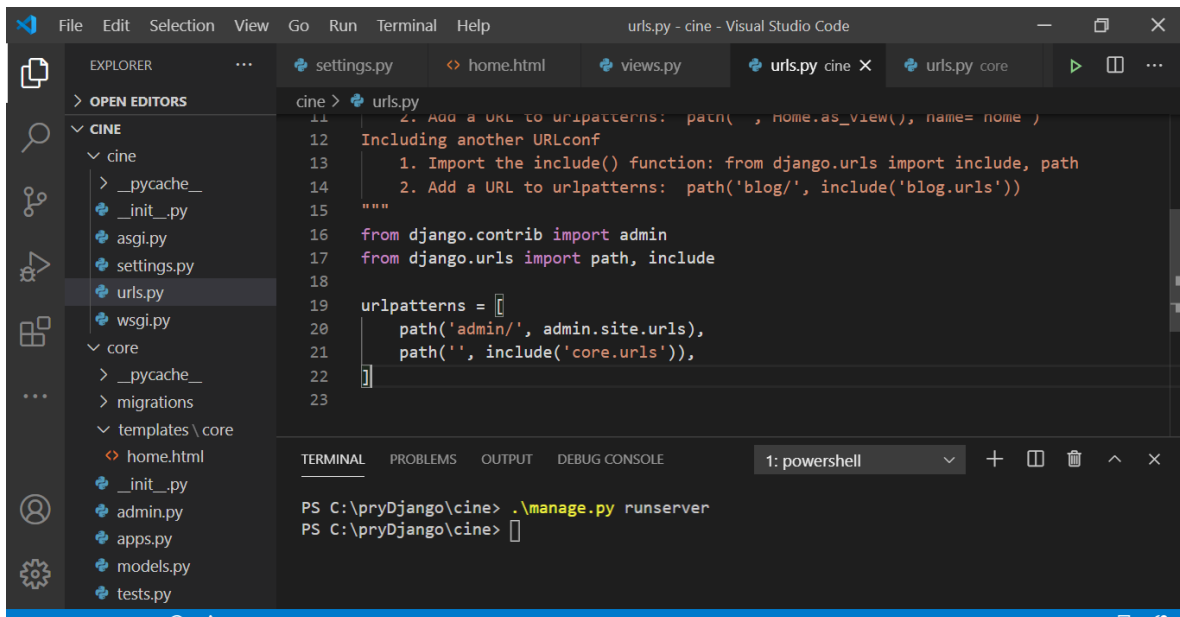
Esto es una especie de controlador, donde se indica que cuando llame al home, estará en el directorio principal



```
core > urls.py
1 from django.urls import path
2 from .views import home
3
4 urlpatterns = [
5     path('', home, name="home"),
6 ]
```

```
PS C:\pryDjango\cine> .\manage.py runserver
PS C:\pryDjango\cine>
```

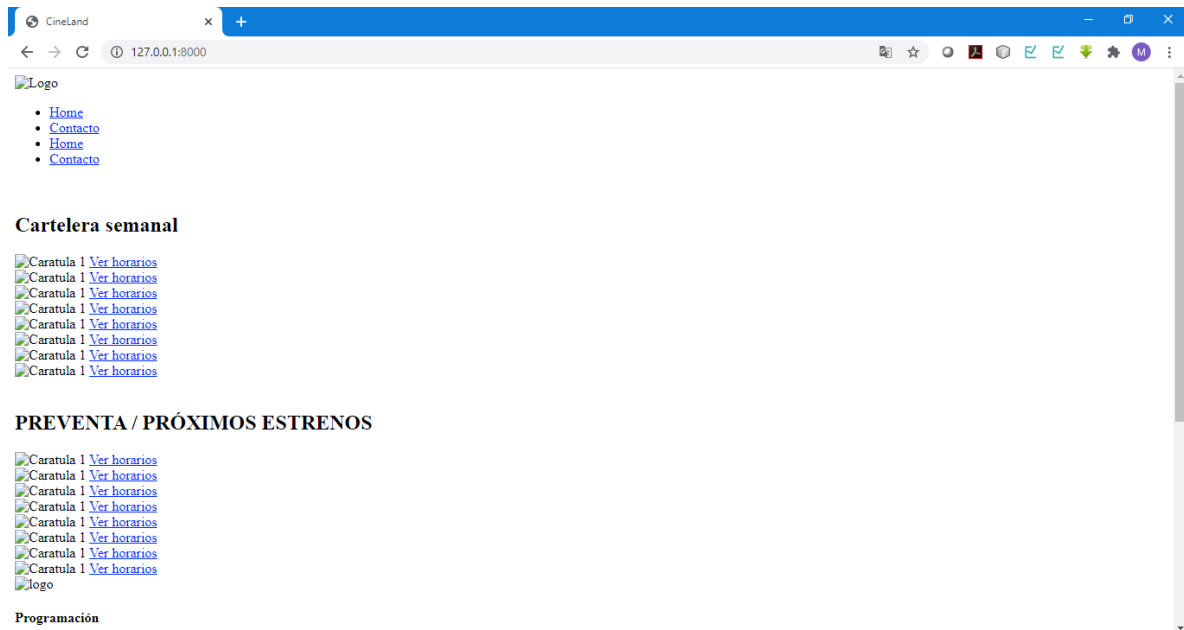
Ahora modificamos el archivo *cine/urls.py* para que pueda ejecutar el template y agregar el nuevo path



```
cine > urls.py
11 2. Add a URL to urlpatterns: path('', home.as_view(), name= name )
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('', include('core.urls')),
22 ]
23
```

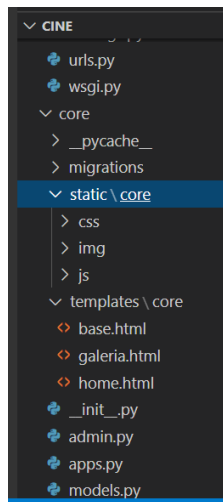
```
PS C:\pryDjango\cine> .\manage.py runserver
PS C:\pryDjango\cine>
```

Grabas y luego actualizas el navegador, donde debería mostrar la página de esta forma, pero con un problema ya que no carga el css.

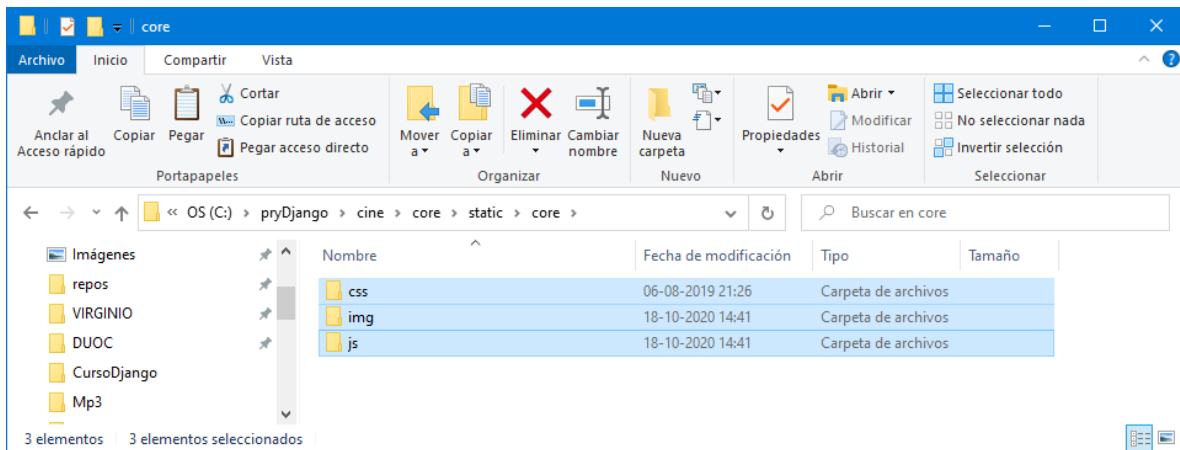


Carga de archivos static

¿Dónde cargamos los archivos css?, dentro del mismo módulo core dentro de una subcarpeta de nombre: *static* y dentro de esta otro con el nombre del módulo *core*



Una vez creadas las subcarpetas, copiamos y pegamos las carpetas necesarias (css, js, img) del proyecto estático en la ruta *cine/core/static/core*



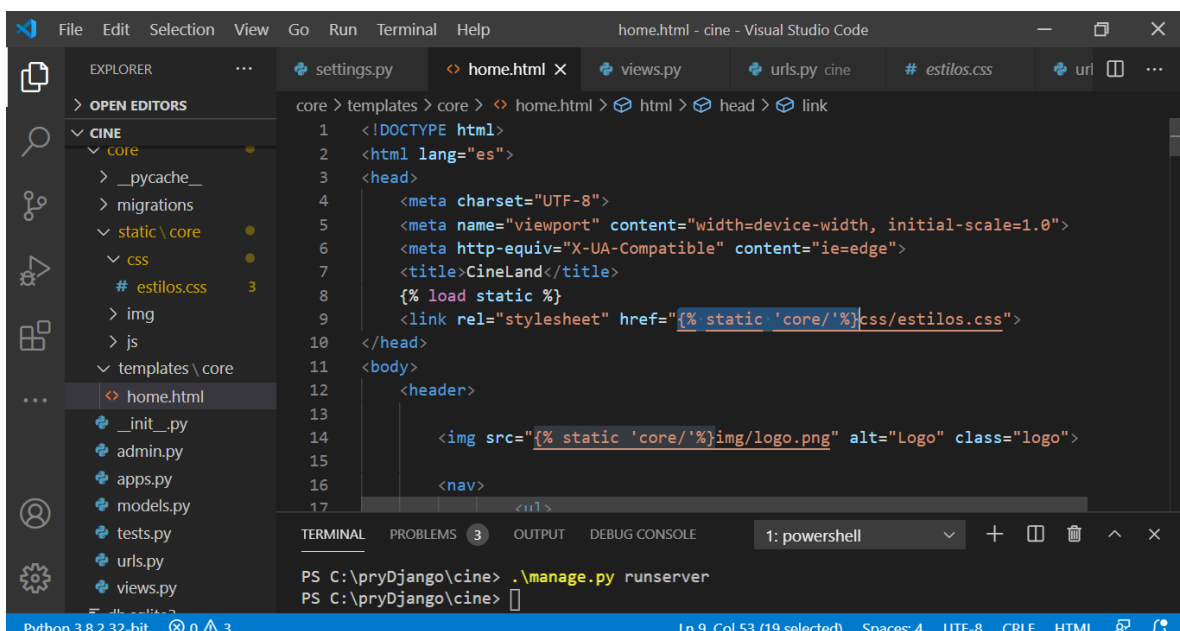
Una vez copiados los elementos de css, js, img necesitamos modificar home.html en las líneas que llama a los archivos externos, *por ejemplo* `<link rel="stylesheet" href="css/estilos.css">`

Lo mismo con las llamadas a las imágenes desde el html, ya que ahora hay que hacerlos según la ruta del proyecto y para eso necesitamos incorporar un tag de django `{% load static %}` donde éstos pasan a ser variables que contienen información de texto y reemplazan y/o insertan en tiempo de ejecución el texto almacenado.

Te recomiendo que en este caso lo reemplaces por lo siguiente:

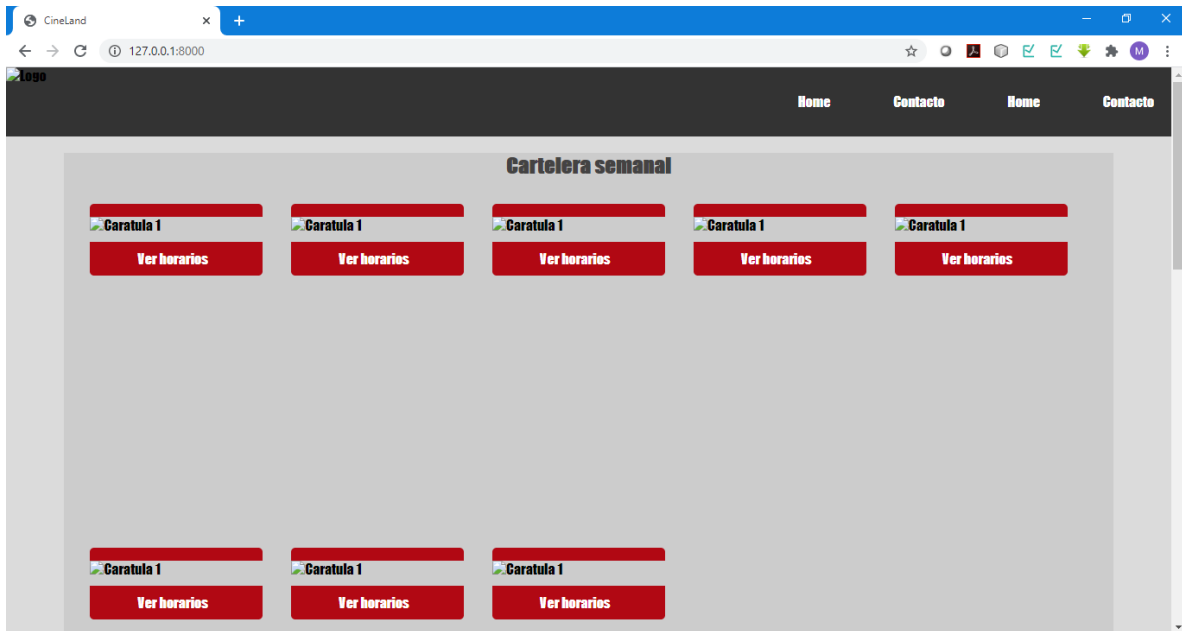
```
{% static 'core/' %}
```

Eso permite que donde se encuentre ese tag, este indique la ruta desde donde se obtiene la información, en este caso: desde `static/core/` ya que ahora es donde tenemos nuestras carpetas con css, js y las img (fíjate en las líneas 9 y 14 de la imagen)

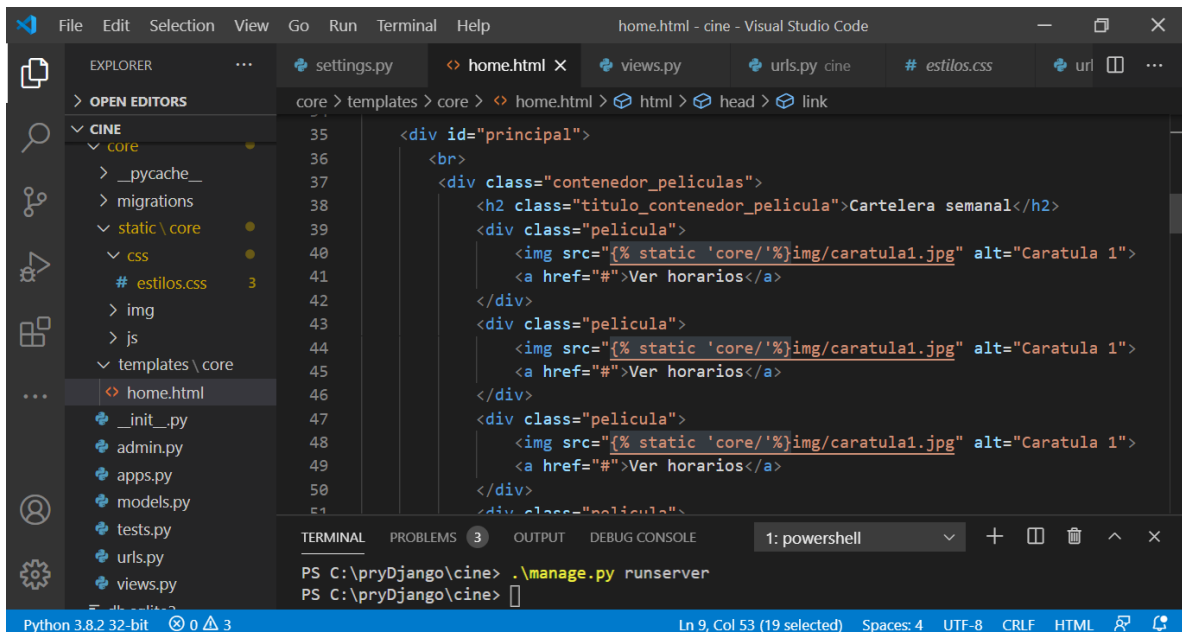


Ahora, necesitamos reiniciar el servidor para que tome los cambios y encuentre las carpetas añadidas: CTRL C sobre la ventana del servidor y luego `.\manage.py runserver`

Vuelves a abrir la url del servidor y ya se debería ver algo así:



Necesitamos cambiar el resto de `home.html` para que las imágenes se vean, por lo tanto, debemos hacer algo parecido en la ruta de las imágenes incrustando el tag de django `{% static 'core/' %}`



The screenshot shows the Visual Studio Code interface with a Django project named 'cine'. The Explorer sidebar on the left shows the project structure: 'core' (containing 'static' and 'templates'), 'css' (containing 'estilos.css'), 'img', 'js', and 'models.py'. The 'core' directory is expanded, showing 'static' and 'templates'. The 'templates' directory is also expanded, showing 'home.html'. The 'home.html' file is open in the editor, showing the following HTML code:

```
35 <div id="principal">
36   <br>
37   <div class="contenedor_peliculas">
38     <h2 class="titulo_contenedor_pelicula">Cartelera semanal</h2>
39     <div class="pelicula">
40       
41       <a href="#">Ver horarios</a>
42     </div>
43     <div class="pelicula">
44       
45       <a href="#">Ver horarios</a>
46     </div>
47     <div class="pelicula">
48       
49       <a href="#">Ver horarios</a>
50     </div>
51     <div class="pelicula">
```

The terminal at the bottom shows the command prompt for PowerShell, with the command `PS C:\pryDjango\cine> .\manage.py runserver` and the output `PS C:\pryDjango\cine>` .

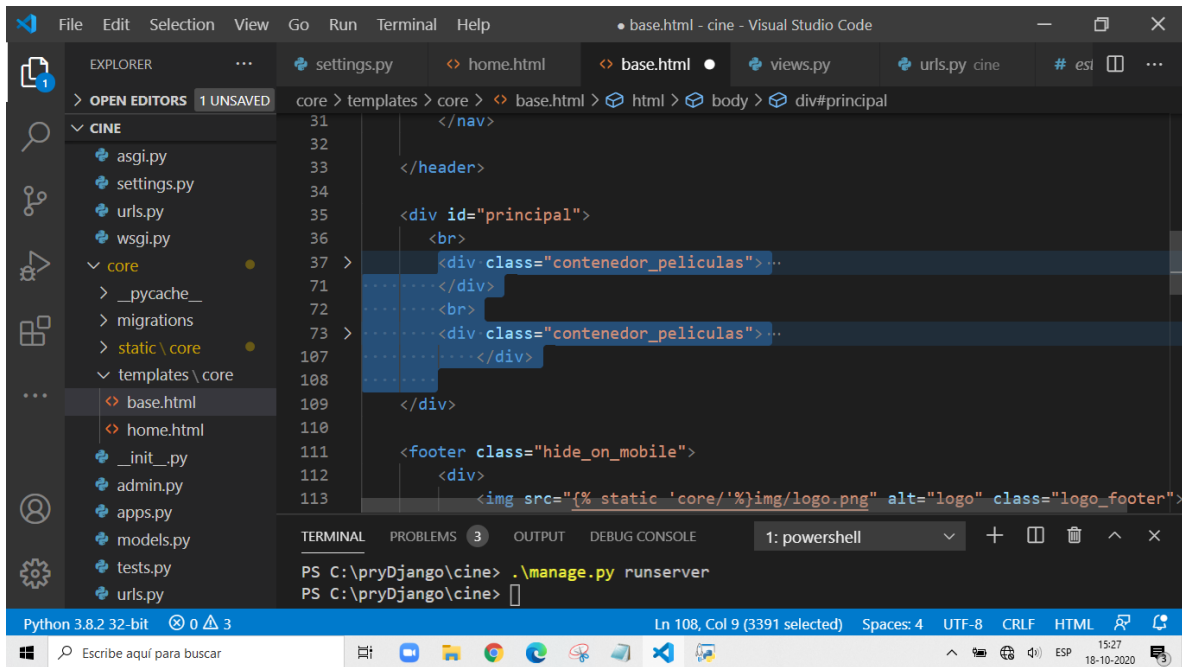
Si actualizas el navegador, debieses ver el contenido general con todas las imágenes como estaban escritas en el archivo estático index.html.

Si bien es cierto se ve bien, pero el contenido es estático, por lo tanto si quisiera mostrar más imágenes, debiese incorporar más código html estático, por lo que conviene utilizar un template dividido por secciones, por ejemplo, un bloque para el encabezado, un bloque para el contenido, un bloque para el footer, lo que permitirá cambiar en tiempo de ejecución lo requerido. Para ello utilizaremos template.

3. Creando el template o página base

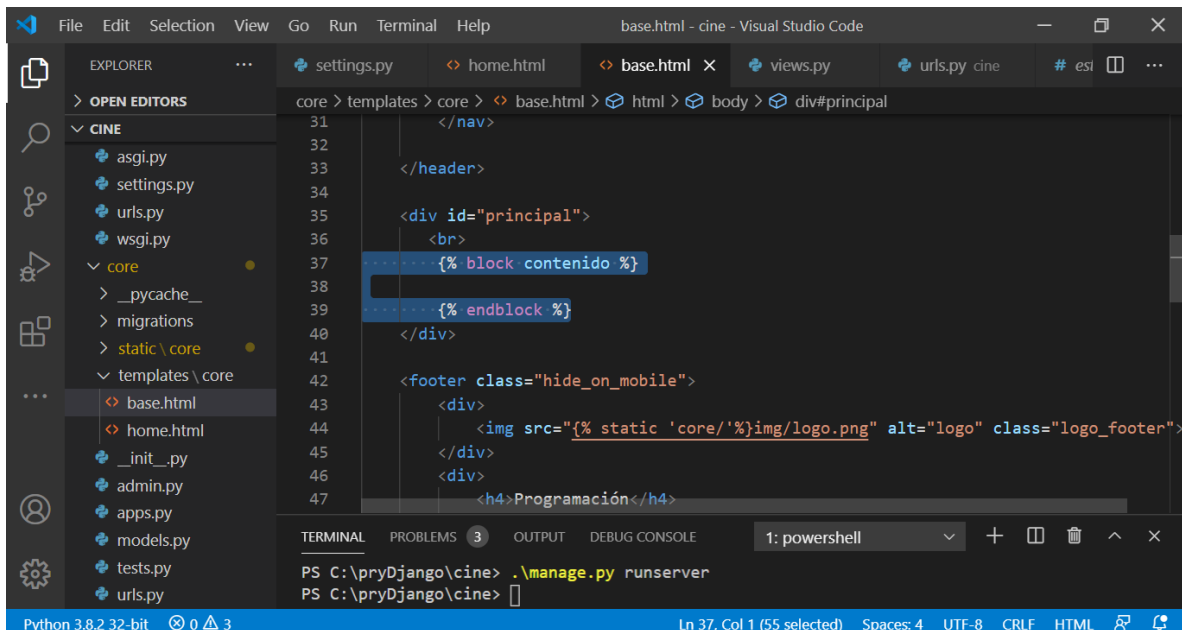
En el mismo directorio donde tenemos `home.html`, creamos una página nueva llamada `base.html`

Copiamos y pegamos el contenido completo de `home.html` en `base.html` pero dejamos todo lo que se mantiene en la página y eliminamos las secciones que variarán.



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying a project structure. The main editor window shows the `base.html` file, which has been created by copying content from `home.html`. The code in `base.html` includes a navigation bar, a header, a main content area with two `<div class="contenedor_peliculas">` blocks, and a footer with a logo. The terminal at the bottom shows the command `PS C:\pryDjango\cine> .\manage.py runserver` being executed.

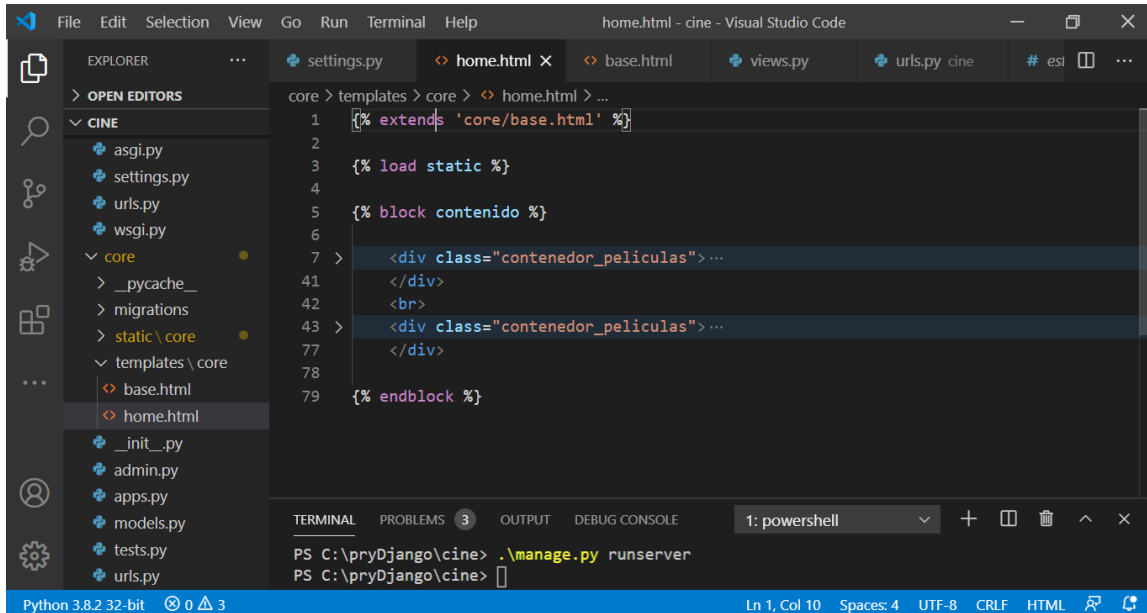
Y lo reemplazamos por los tag `{% block contenido %}` y `{% endblock %}`



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The main editor window shows the `base.html` file, where the content of the `<div class="contenedor_peliculas">` blocks has been replaced with the Django template tags `{% block contenido %}` and `{% endblock %}`. The terminal at the bottom shows the command `PS C:\pryDjango\cine> .\manage.py runserver` being executed.

Ahora el *home.html* ya no necesita todo el código html anterior ya que lo llamaremos desde nuestra página *base.html*

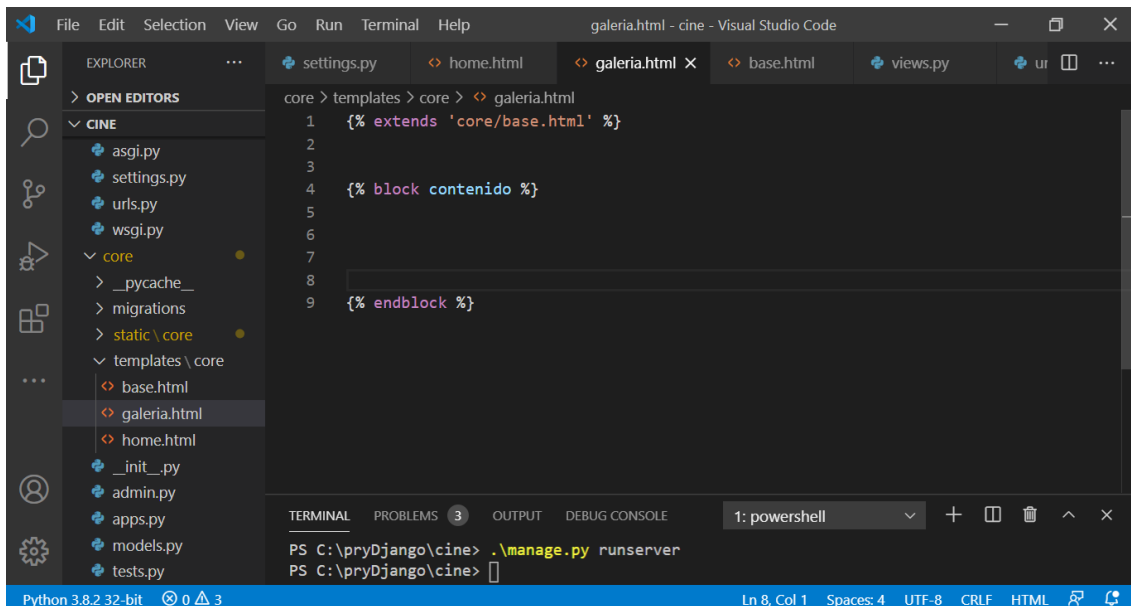
Donde *base.html* queda al revés de lo que muestra *home.html*, dejando como contenido el html que muestra en este caso los div contenedor_peliculas



```
core > templates > core > home.html > ...
1  {% extends 'core/base.html' %}
2
3  {% load static %}
4
5  {% block contenido %}
6
7  <div class="contenedor_peliculas">...
41 </div>
42 <br>
43 <div class="contenedor_peliculas">...
77 </div>
78
79 {% endblock %}
```

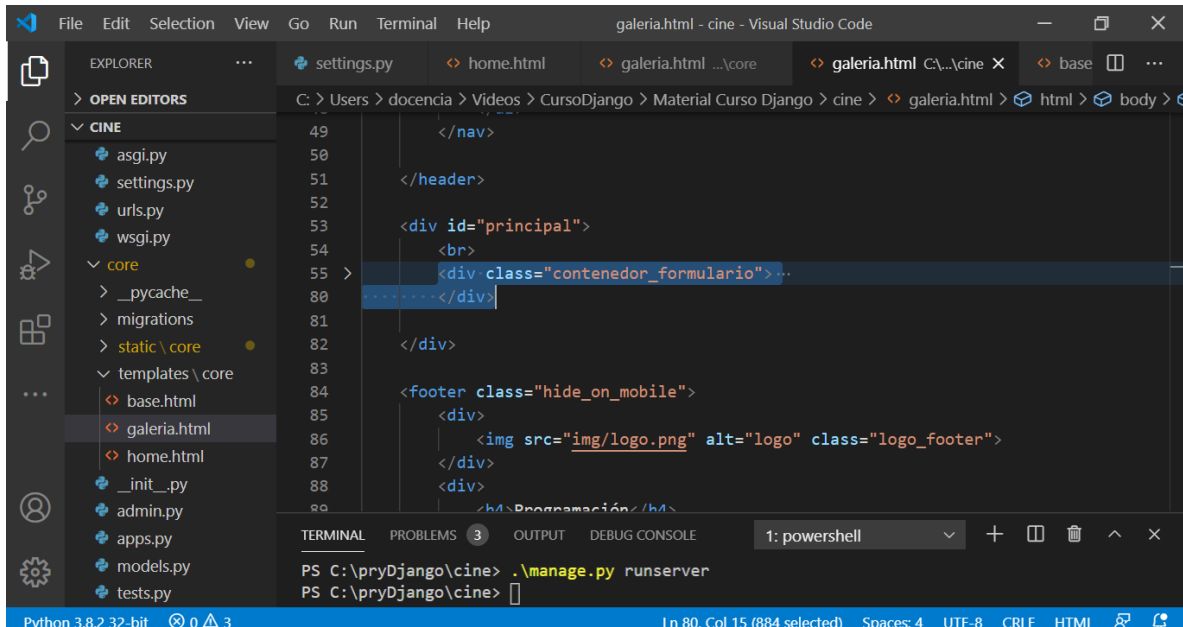
Ahora agreguemos otra página, galeria.html (recuerda los pasos, crear el archivo, crear la vista y crear la url)

Vamos a *templates/core/* y agregamos un nuevo archivo *galeria.html*



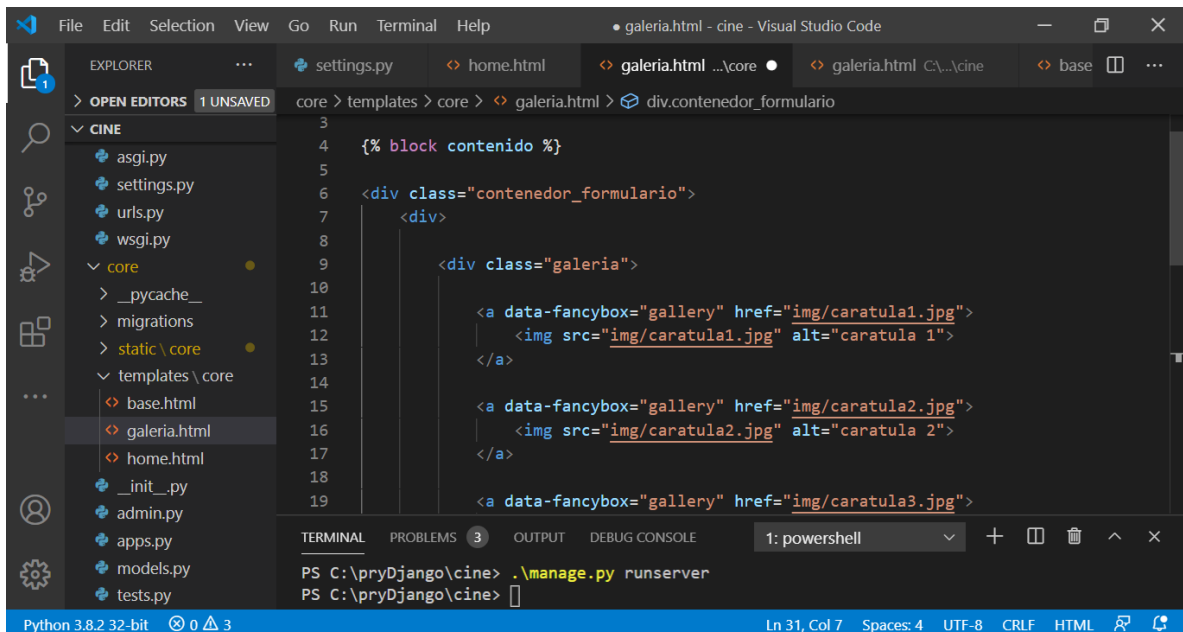
```
core > templates > core > galeria.html
1  {% extends 'core/base.html' %}
2
3
4  {% block contenido %}
5
6
7
8
9  {% endblock %}
```

Abrimos `galeria.html` del sitio original y copiamos sólo el contenido que necesito dejar



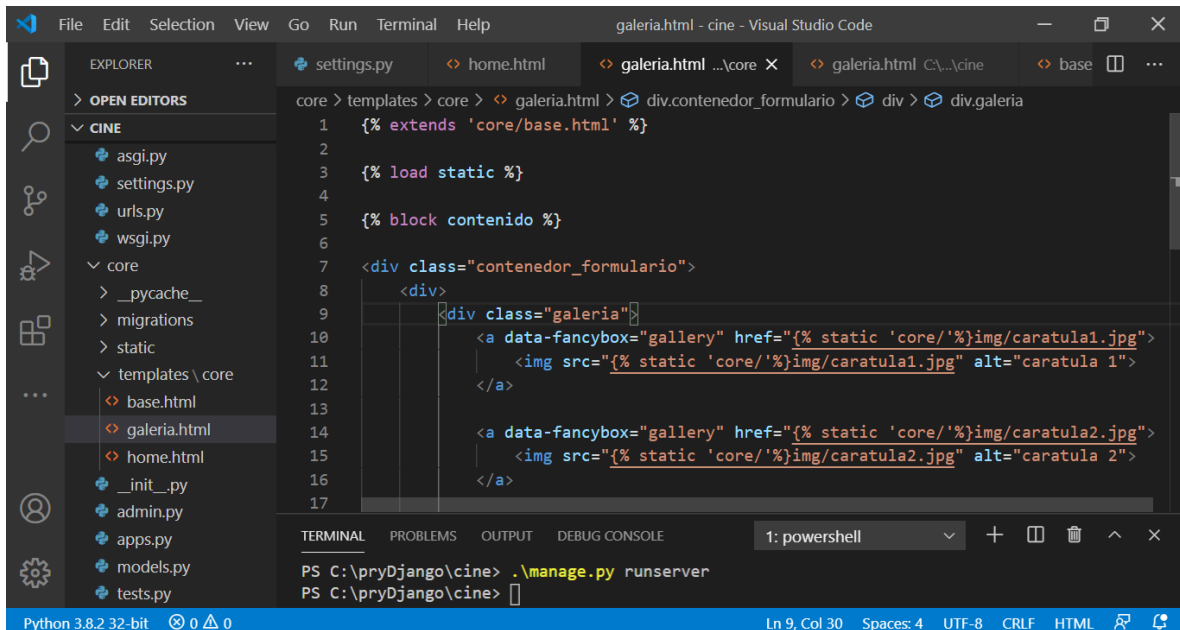
```
49     </nav>
50
51   </header>
52
53   <div id="principal">
54     <br>
55     <div class="contenedor_formulario">...
56     ...</div>
57
58   </div>
59
60   <footer class="hide_on_mobile">
61     <div>
62       
63     </div>
64     <div>
65       <h4>Programación</h4>
66     </div>
67   </footer>
68 </div>
69 </body>
70 </html>
```

Y lo copio en `templates/core/galeria.html`



```
3
4 {% block contenido %}
5
6 <div class="contenedor_formulario">
7   <div>
8
9     <div class="galeria">
10
11       <a data-fancybox="gallery" href="img/caratula1.jpg">
12         
13       </a>
14
15       <a data-fancybox="gallery" href="img/caratula2.jpg">
16         
17       </a>
18
19       <a data-fancybox="gallery" href="img/caratula3.jpg">
```

Recuerda de colocar el `{% load static %}` para abreviar y cambiar los link asociados a css e img



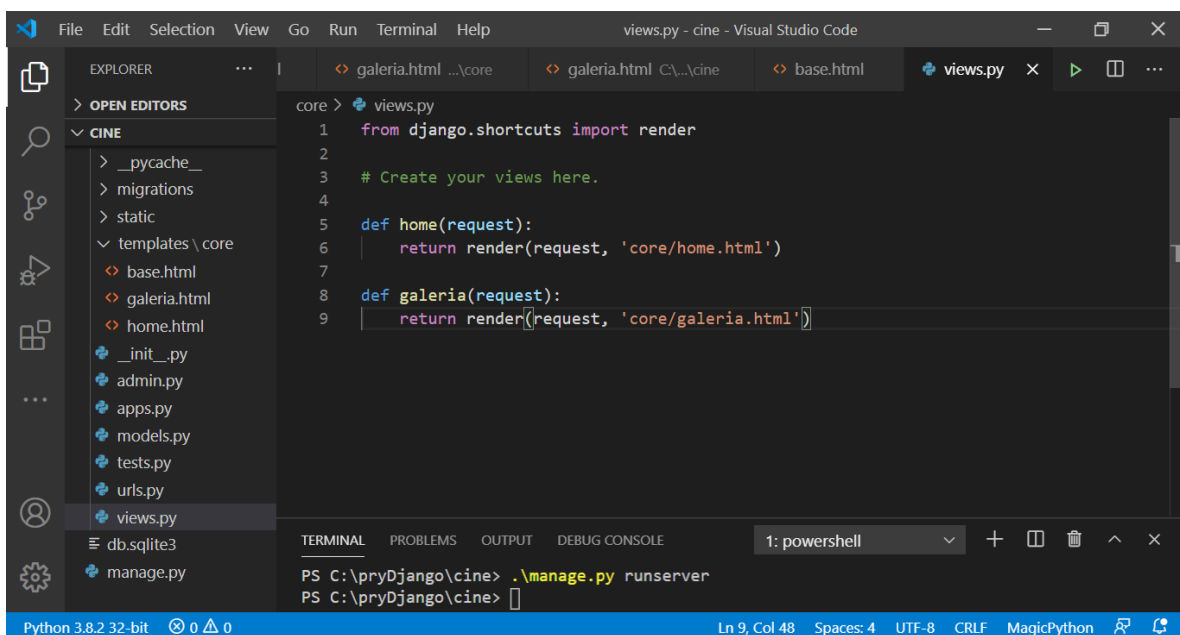
```
File Edit Selection View Go Run Terminal Help
galeria.html - cine - Visual Studio Code

EXPLORER
OPEN EDITORS
CINE
  asgi.py
  settings.py
  urls.py
  wsgi.py
  core
    _pycache_
    migrations
    static
    templates
      core
        galeria.html
        home.html
      _init_.py
      admin.py
      apps.py
      models.py
      tests.py

core > templates > core > galeria.html > div.contenedor_formulario > div > div.galeria
1  {% extends 'core/base.html' %}
2
3  {% load static %}
4
5  {% block contenido %}
6
7  <div class="contenedor_formulario">
8    <div>
9      <div class="galeria">
10        <a data-fancybox="gallery" href="{% static 'core/' %}img/caratula1.jpg">
11          
12        </a>
13
14        <a data-fancybox="gallery" href="{% static 'core/' %}img/caratula2.jpg">
15          
16        </a>
17
TERMINAL
PROBLEMS
OUTPUT
DEBUG CONSOLE
1: powershell
PS C:\pryDjango\cine> .\manage.py runserver
PS C:\pryDjango\cine>

Python 3.8.2 32-bit 0 0 0 Ln 9, Col 30 Spaces: 4 UTF-8 CRLF HTML
```

Ahora agregamos la view en `views.py`



```
File Edit Selection View Go Run Terminal Help
views.py - cine - Visual Studio Code

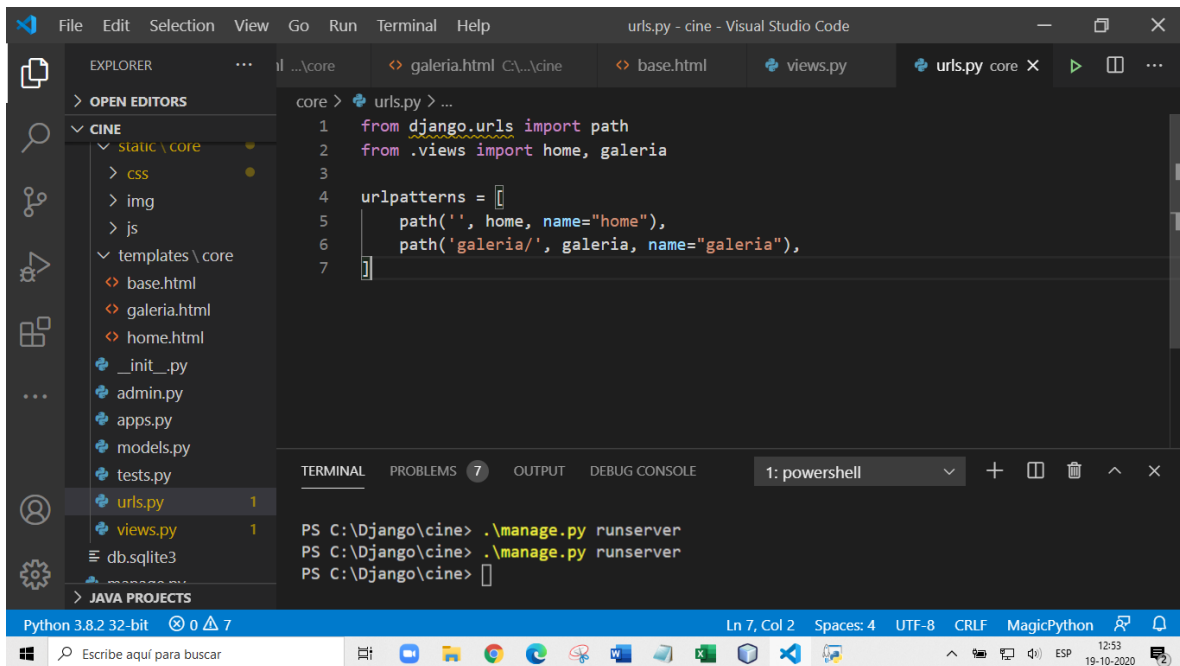
EXPLORER
OPEN EDITORS
CINE
  _pycache_
  migrations
  static
  templates
    core
      base.html
      galeria.html
      home.html
    _init_.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  db.sqlite3
  manage.py

core > views.py
1  from django.shortcuts import render
2
3  # Create your views here.
4
5  def home(request):
6    return render(request, 'core/home.html')
7
8  def galeria(request):
9    return render(request, 'core/galeria.html')

TERMINAL
PROBLEMS
OUTPUT
DEBUG CONSOLE
1: powershell
PS C:\pryDjango\cine> .\manage.py runserver
PS C:\pryDjango\cine>

Python 3.8.2 32-bit 0 0 0 Ln 9, Col 48 Spaces: 4 UTF-8 CRLF MagicPython
```

Ahora agregamos la url



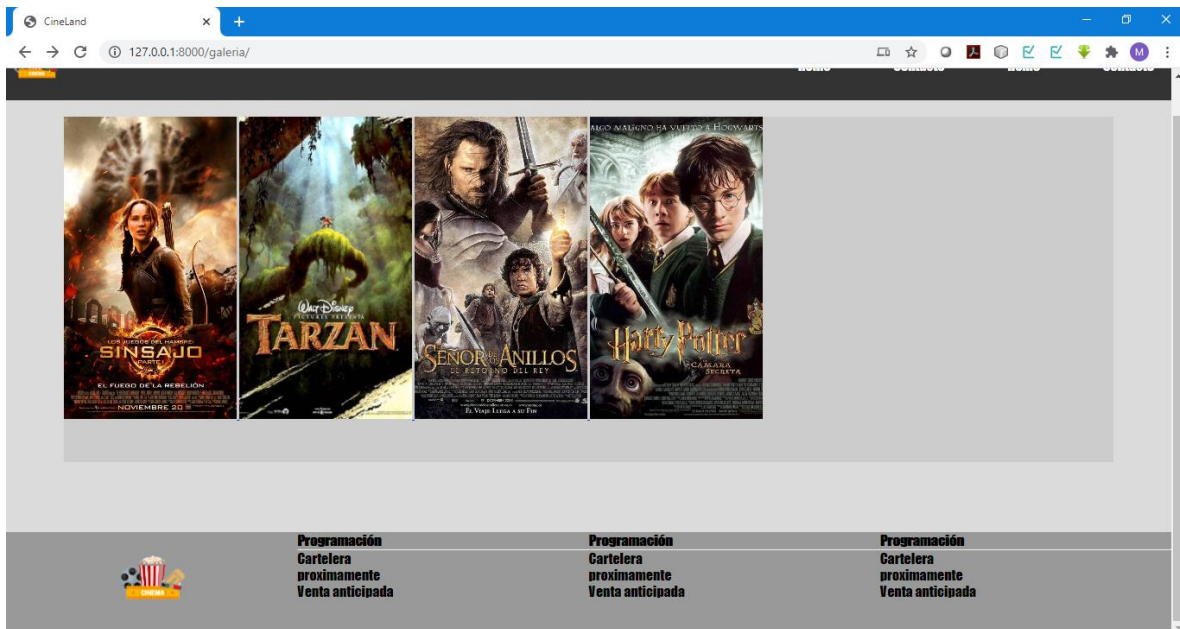
The screenshot shows the Visual Studio Code interface with a Django project named 'cine'. The Explorer panel on the left shows the project structure, including the 'core' app with files like 'urls.py', 'views.py', and 'models.py'. The main editor shows the 'urls.py' file with the following code:

```
1 from django.urls import path
2 from .views import home, galeria
3
4 urlpatterns = [
5     path('', home, name="home"),
6     path('galeria/', galeria, name="galeria"),
7 ]
```

The terminal at the bottom shows the command prompt running the Django server:

```
PS C:\Django\cine> .\manage.py runserver
PS C:\Django\cine> .\manage.py runserver
PS C:\Django\cine>
```

En el navegador prueba colocando la dirección: <http://127.0.0.1:8000/galeria/>

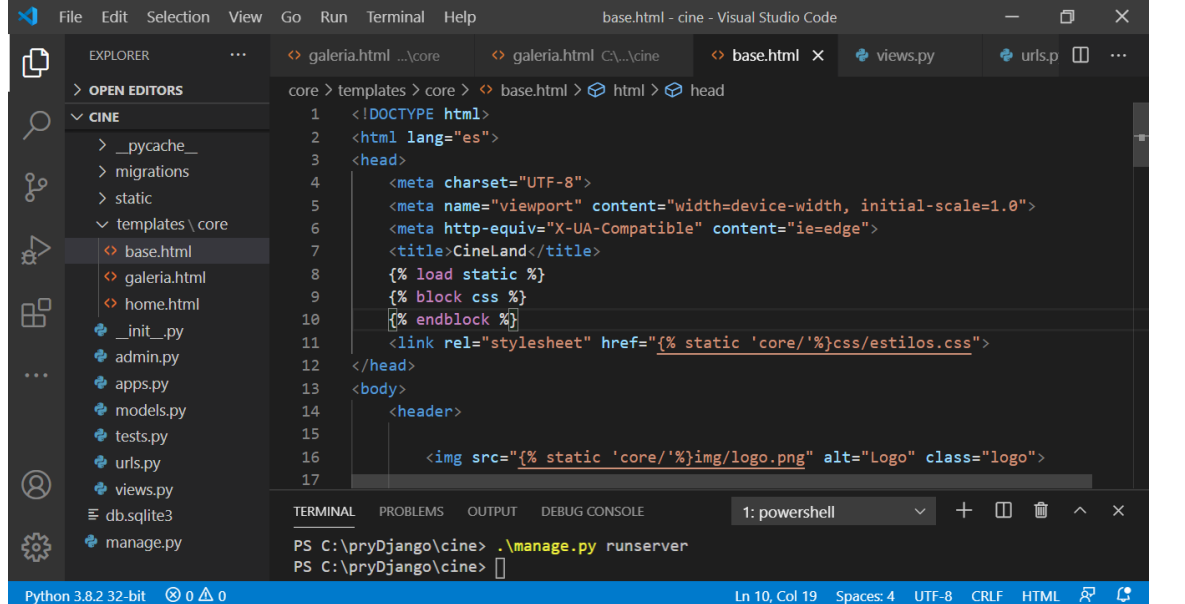


Al momento, cargan las imágenes, pero faltan asociar los plugins de fancybox y jquery para que pueda mostrar la galería como el efecto correspondiente.

Para eso, en la *galeria.html* original se cargan los plugins al principio en el head y otros al final del body, por lo que en la base, deberemos realizar los bloques para que se carguen como corresponden.

Modificamos *base.html* para agregar los tag

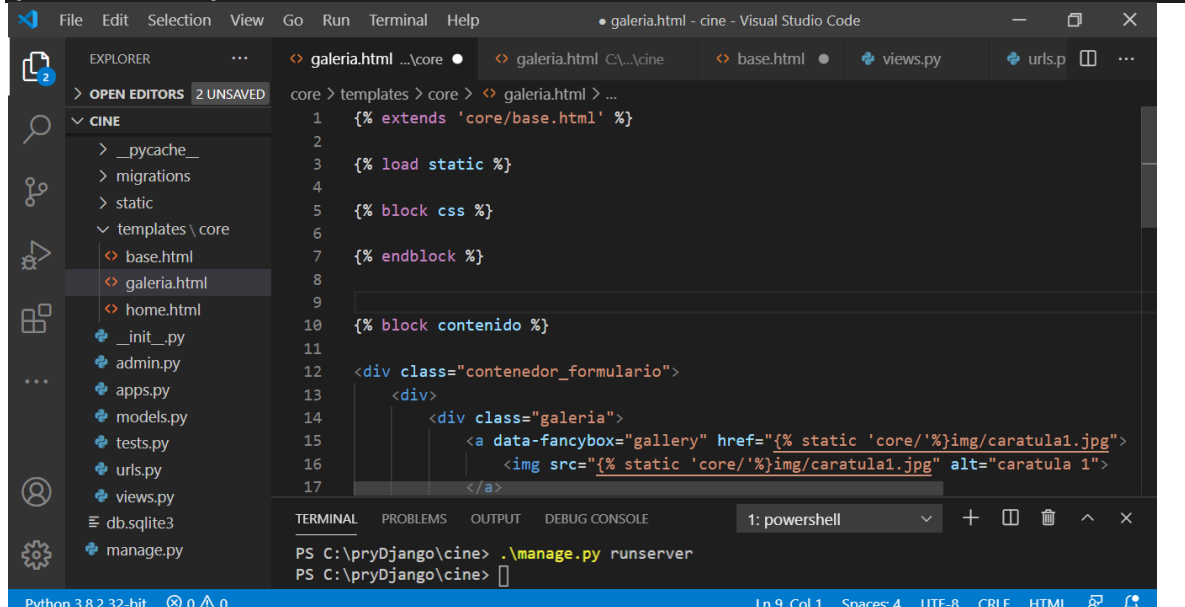
```
{% load static %}
{% block css %}
{% endblock %}
```



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>CineLand</title>
  {% load static %}
  {% block css %}
  {% endblock %}
  <link rel="stylesheet" href="{% static 'core/' %}css/estilos.css">
</head>
<body>
  <header>
    
  </header>
</body>
```

Y luego modificamos *galeria.html* para que se puedan cargar los bloques donde corresponde

```
{% block css %}
{% endblock %}
```



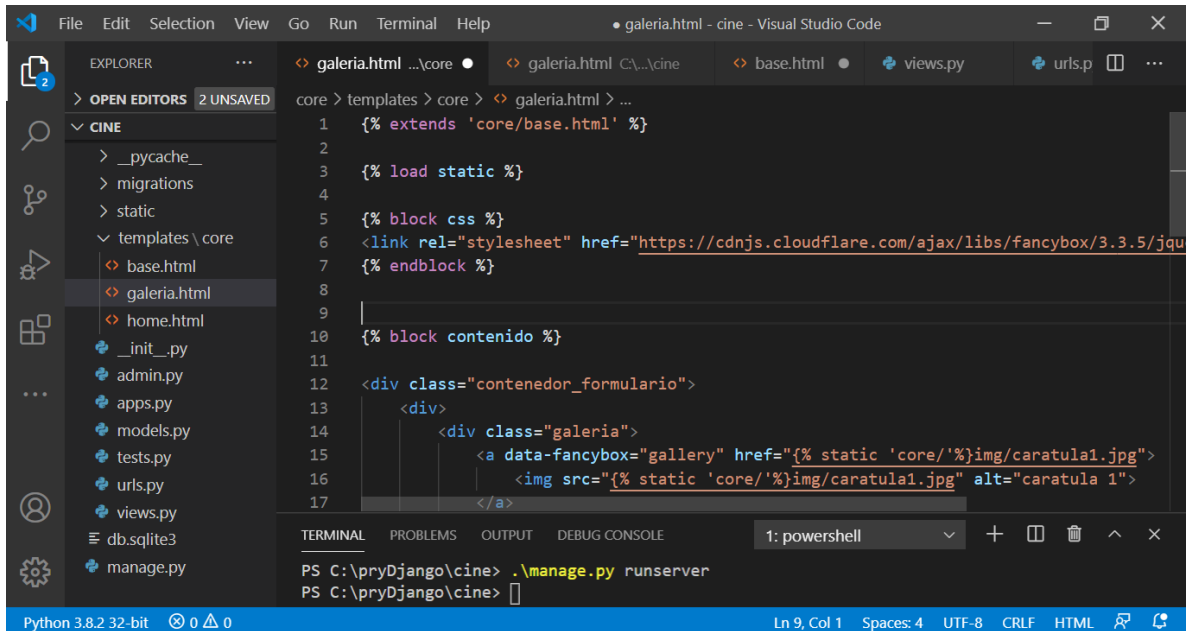
```
{% extends 'core/base.html' %}

{% load static %}
{% block css %}
{% endblock %}

{% block contenido %}
  <div class="contenedor_formulario">
    <div>
      <div class="galeria">
        <a data-fancybox="gallery" href="{% static 'core/' %}img/caratula1.jpg">
          
        </a>
      </div>
    </div>
  </div>
{% endblock %}
```

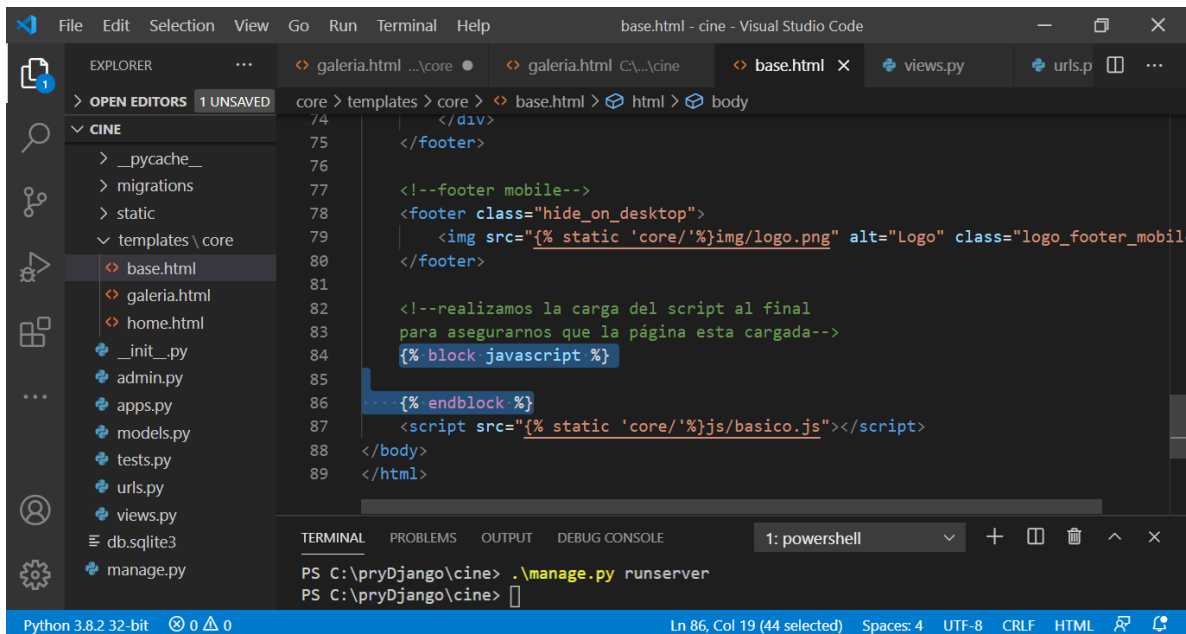
Desde el archivo `galeria.html` original copiamos la línea del plugin en el bloque css de `templates/core/galeria.html`

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/3.3.5/jquery.fancybox.min.css" />
```



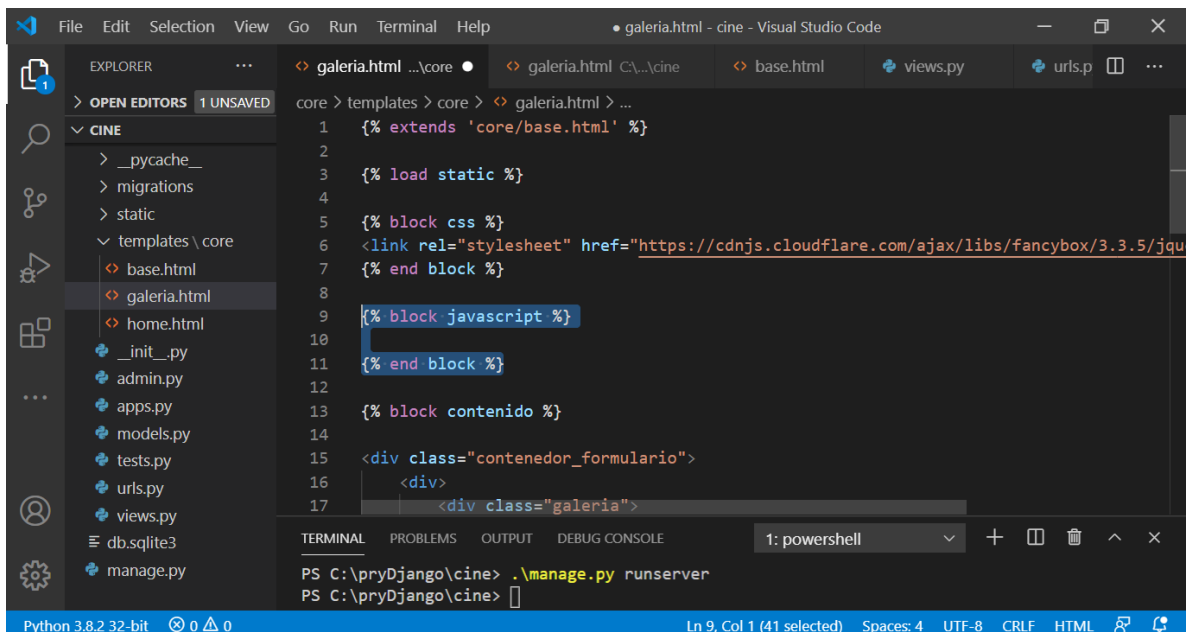
Ahora preparamos la carga del plugin en la parte inferior, donde se carga el jquery

Editamos `base.html` y le añadimos el bloque de carga utilizando los tag



```
base.html - cine - Visual Studio Code
EXPLORER
core > templates > core > base.html > html > body
74 </div>
75 </footer>
76
77 <!--footer mobile-->
78 <footer class="hide_on_desktop">
79 
81
82 <!--realizamos la carga del script al final
83 para asegurarnos que la página esta cargada-->
84 {% block javascript %}
85
86 {% endblock %}
87 <script src="{% static 'core/' %}js/basico.js"></script>
88 </body>
89 </html>
TERMINAL
1: powershell
PS C:\pryDjango\cine> .\manage.py runserver
PS C:\pryDjango\cine>
```

Y en *templates/core/galeria.html* le añadimos el bloque de carga para los javascript



```
galeria.html - cine - Visual Studio Code
EXPLORER
core > templates > core > galeria.html > ...
1 {% extends 'core/base.html' %}
2
3 {% load static %}
4
5 {% block css %}
6 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/3.3.5/jqu
7 {% end block %}
8
9 {% block javascript %}
10
11 {% end block %}
12
13 {% block contenido %}
14
15 <div class="contenedor_formulario">
16 <div>
17 <div class="galeria">
```

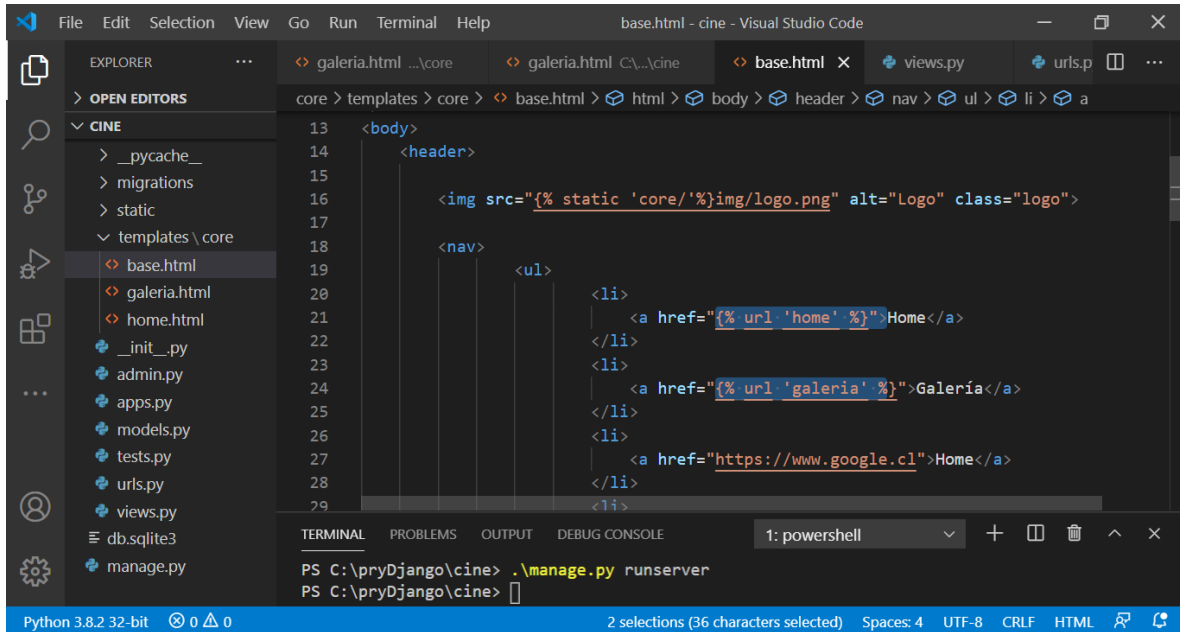
Copiamos desde el archivo antiguo las 2 líneas (casi al final) de carga de los script

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.
js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/fancybox/3.3.5/jquery.fa
ncybox.min.js"></script>
```

Y las pegamos dentro del bloque javascript

Modifiquemos los enlaces interiores para que al hacer click en galería, nos lleven a nuestra página galeria.html

Modificamos *base.html* y cambiamos href de los enlaces



```
base.html - cine - Visual Studio Code

EXPLORER
> OPEN EDITORS
  > CINE
    > _pycache_
    > migrations
    > static
    > templates \ core
      > base.html
      > galeria.html
      > home.html
    > _init_.py
    > admin.py
    > apps.py
    > models.py
    > tests.py
    > urls.py
    > views.py
    > db.sqlite3
    > manage.py

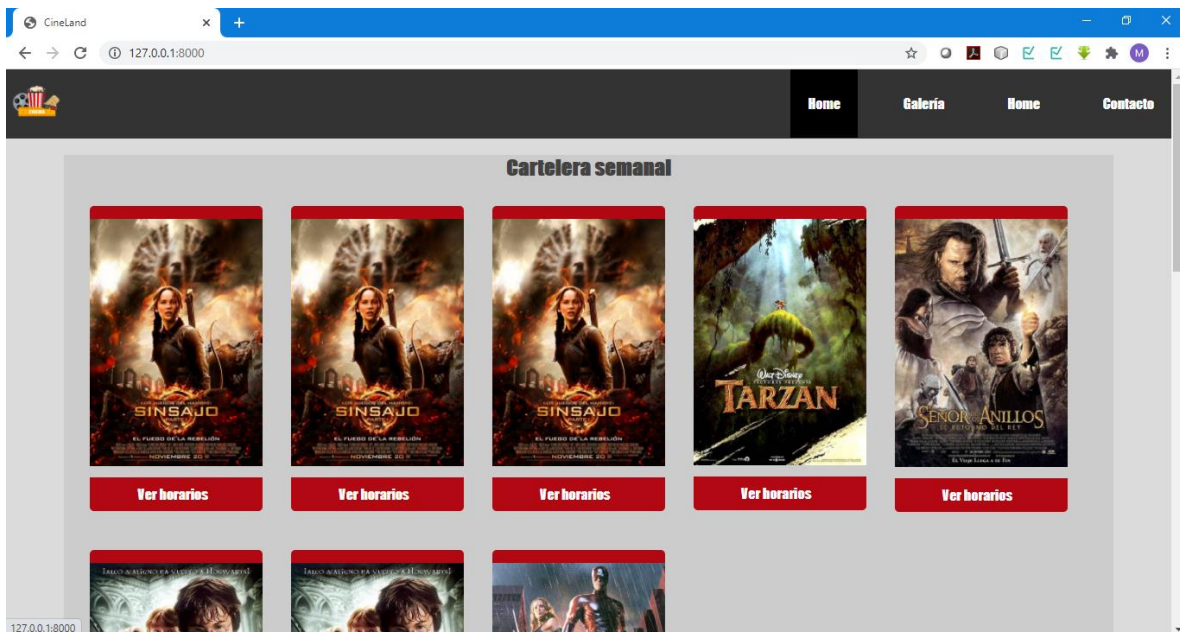
core > templates > core > > base.html > html > body > header > nav > ul > li > a

13 <body>
14   <header>
15     
16   <nav>
17     <ul>
18       <li>
19         <a href="{% url 'home' %}">Home</a>
20       </li>
21       <li>
22         <a href="{% url 'galeria' %}">Galería</a>
23       </li>
24       <li>
25         <a href="https://www.google.cl">Home</a>
26       </li>
27     </ul>
28   </nav>
29 </body>

TERMINAL
PS C:\pryDjango\cine> .\manage.py runserver
PS C:\pryDjango\cine>
```

Para que acceda con los alias definidos en el archivo urls.py

y probamos desde el navegador en los link superiores de Home y Galería



CineLand



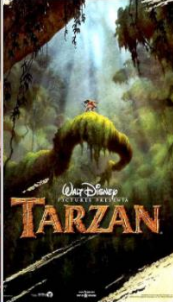

127.0.0.1:8000/galeria/


Home

Galería

Home

Contacto





127.0.0.1:8000/galeria/

Programación

Cartelera

proximamente

Venta anticipada

Programación

Cartelera

proximamente

Venta anticipada

Programación

Cartelera

proximamente

Venta anticipada