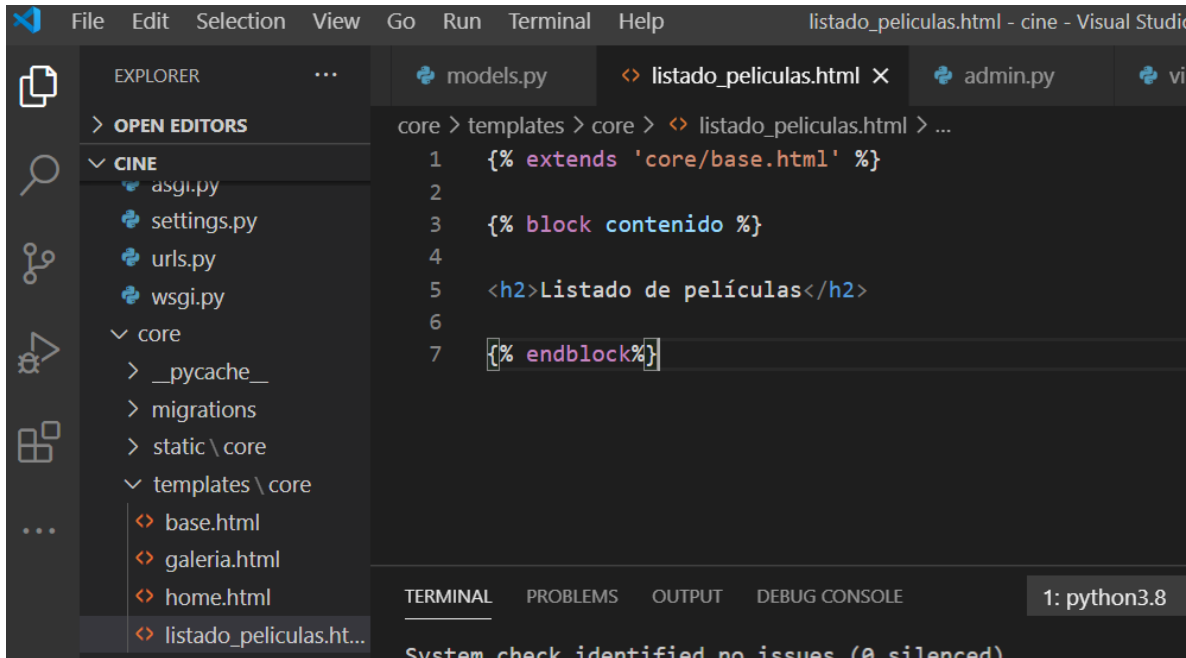


Creando CRUD con páginas HTML

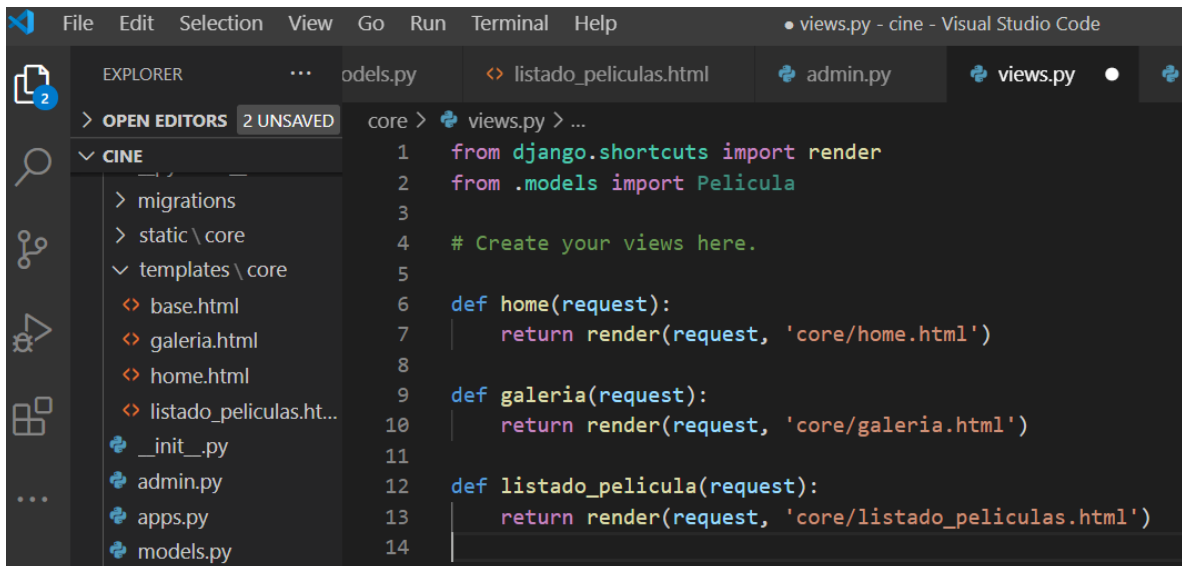
Para listar películas: dentro de `core/templates/core/listado_peliculas.html`



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the project structure, with the file `listado_peliculas.html` selected under the path `core > templates > core`. The main editor area shows the content of this file, which is a Django template. The terminal at the bottom indicates that the system check identified no issues.

```
core > templates > core > <> listado_peliculas.html > ...
1  {% extends 'core/base.html' %}
2
3  {% block contenido %}
4
5  <h2>Listado de películas</h2>
6
7  {% endblock %}
```

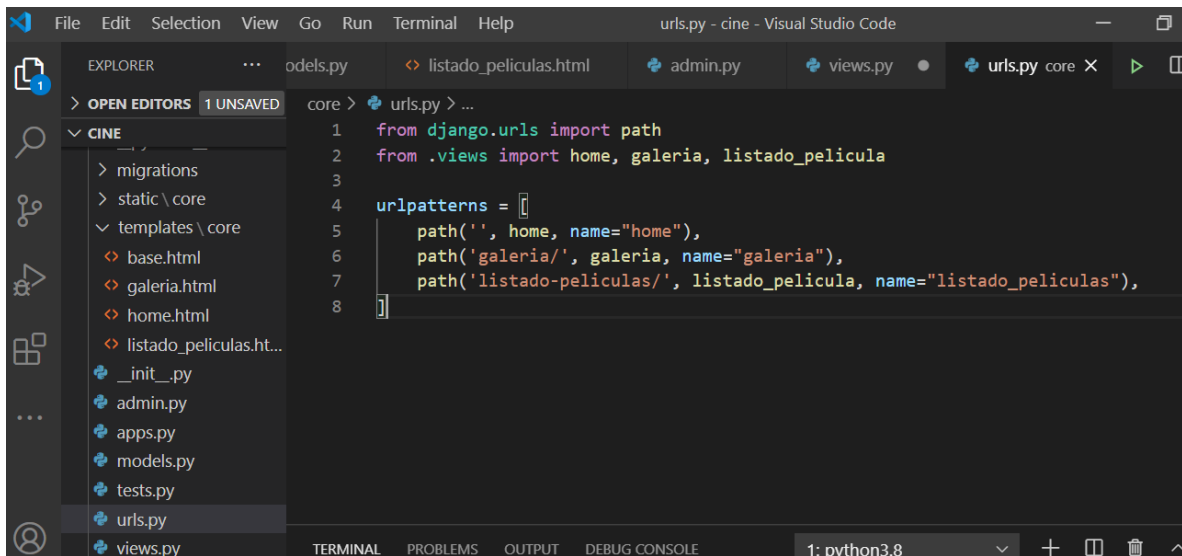
Luego seguimos los pasos: vamos a `core/views.py`



The screenshot shows the Visual Studio Code interface with the file `views.py` open in the main editor. The Explorer panel on the left shows the project structure, with `views.py` selected under the path `core > templates > core`. The main editor area shows the content of `views.py`, which includes imports for `render` from `django.shortcuts` and `Pelicula` from `.models`, and three view functions: `home`, `galeria`, and `listado_pelicula`.

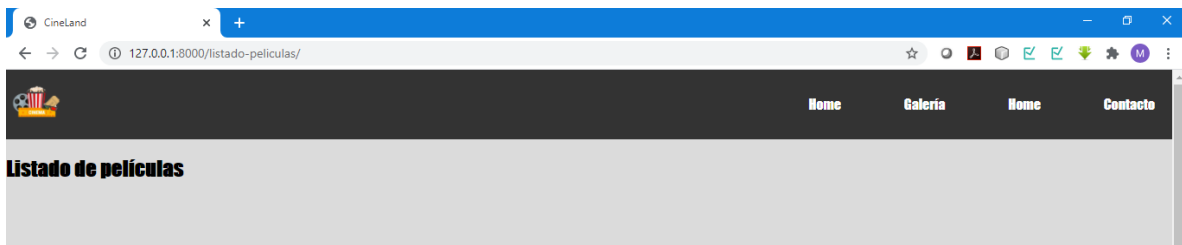
```
core > views.py > ...
1  from django.shortcuts import render
2  from .models import Pelicula
3
4  # Create your views here.
5
6  def home(request):
7      return render(request, 'core/home.html')
8
9  def galeria(request):
10     return render(request, 'core/galeria.html')
11
12  def listado_pelicula(request):
13     return render(request, 'core/listado_peliculas.html')
14
```

Ahora vamos a `core/urls.py`



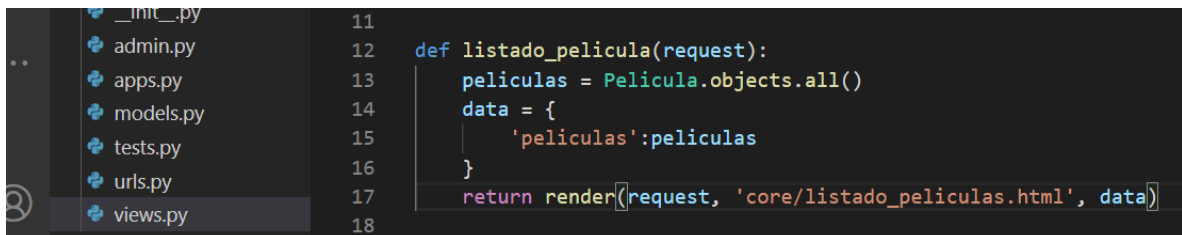
```
1 from django.urls import path
2 from .views import home, galeria, listado_pelicula
3
4 urlpatterns = [
5     path('', home, name="home"),
6     path('galeria/', galeria, name="galeria"),
7     path('listado-peliculas/', listado_pelicula, name="listado_peliculas"),
8 ]
```

Ahora cuando vamos al navegador y vemos listado-peliculas/



Se debiese ver la estructura que llevamos al momento, solo el <h2>Listado de películas</h2>

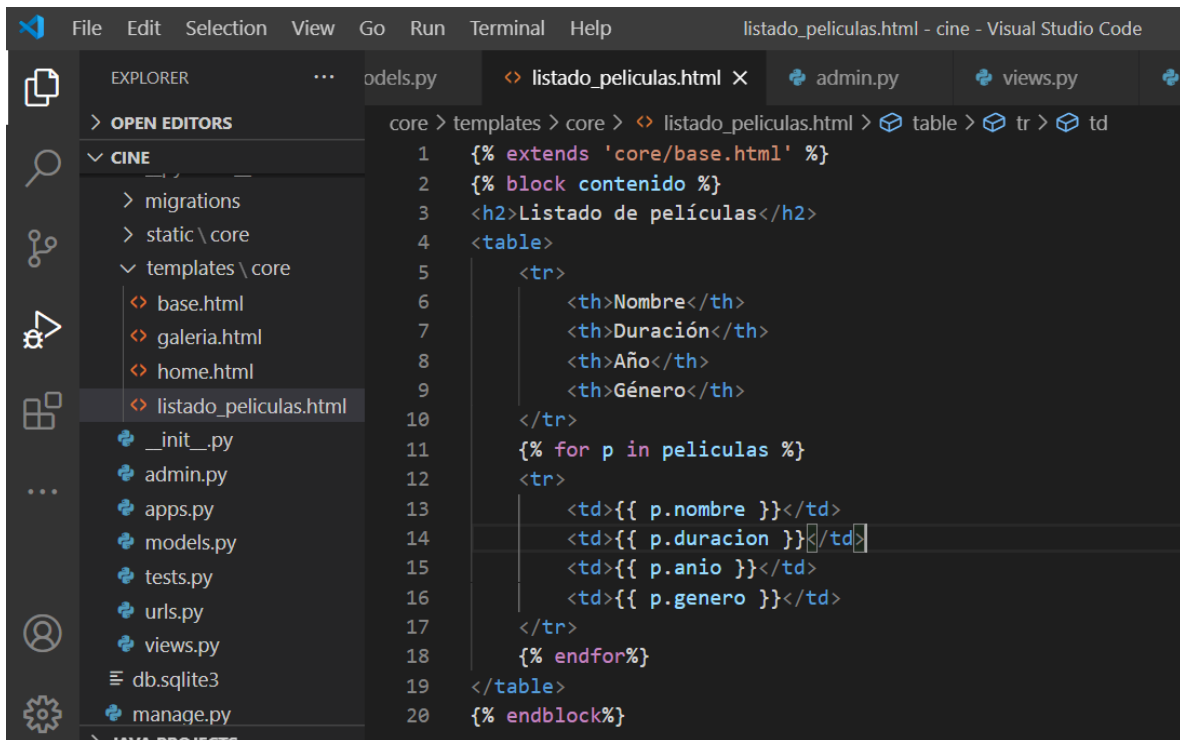
Como queremos mostrar las películas que están en la base de datos, debemos continuar modificando *views.py*



```
11
12 def listado_pelicula(request):
13     peliculas = Película.objects.all()
14     data = {
15         'peliculas':peliculas
16     }
17     return render(request, 'core/listado_peliculas.html', data)
18
```

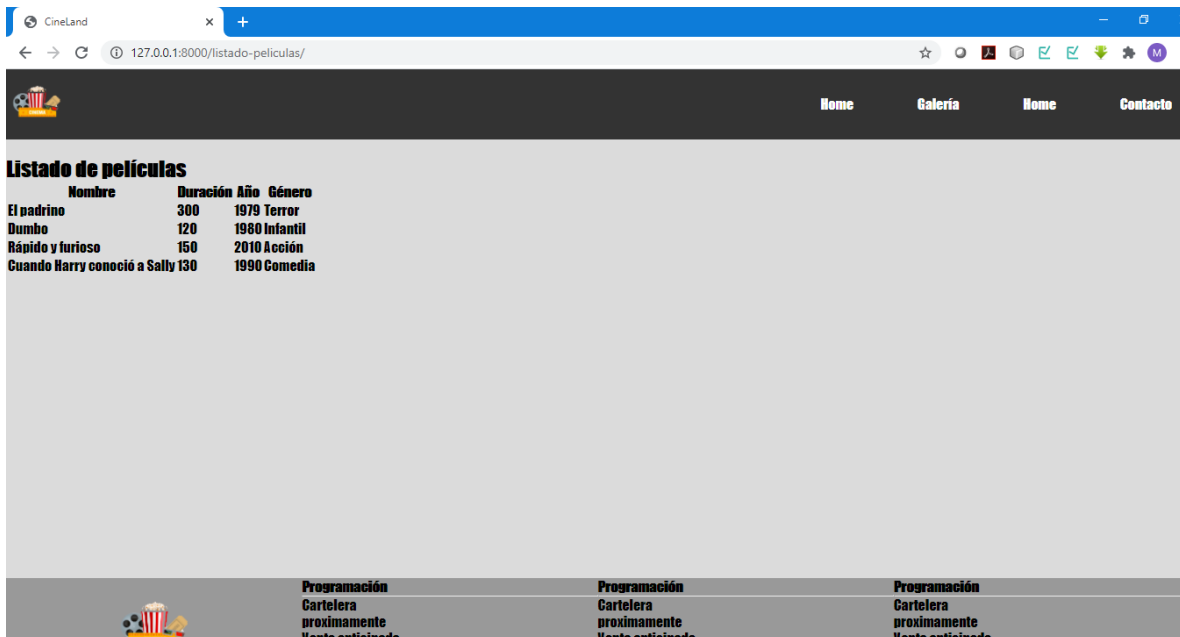
Le dijimos a Django que trajera todas las películas de la tabla y la pasara al template

Editamos el *templates/core/listado_peliculas.html*



```
core > templates > core > listado_peliculas.html > table > tr > td
1 {% extends 'core/base.html' %}
2 {% block contenido %}
3 <h2>Listado de películas</h2>
4 <table>
5     <tr>
6         <th>Nombre</th>
7         <th>Duración</th>
8         <th>Año</th>
9         <th>Género</th>
10    </tr>
11    {% for p in peliculas %}
12    <tr>
13        <td>{{ p.nombre }}</td>
14        <td>{{ p.duracion }}</td>
15        <td>{{ p.anio }}</td>
16        <td>{{ p.genero }}</td>
17    </tr>
18    {% endfor %}
19 </table>
20 {% endblock %}
```

Si actualizo el navegador



Ya se comienzan a visualizar las películas que están en la tabla

Ahora crearemos un formulario para agregar una nueva película

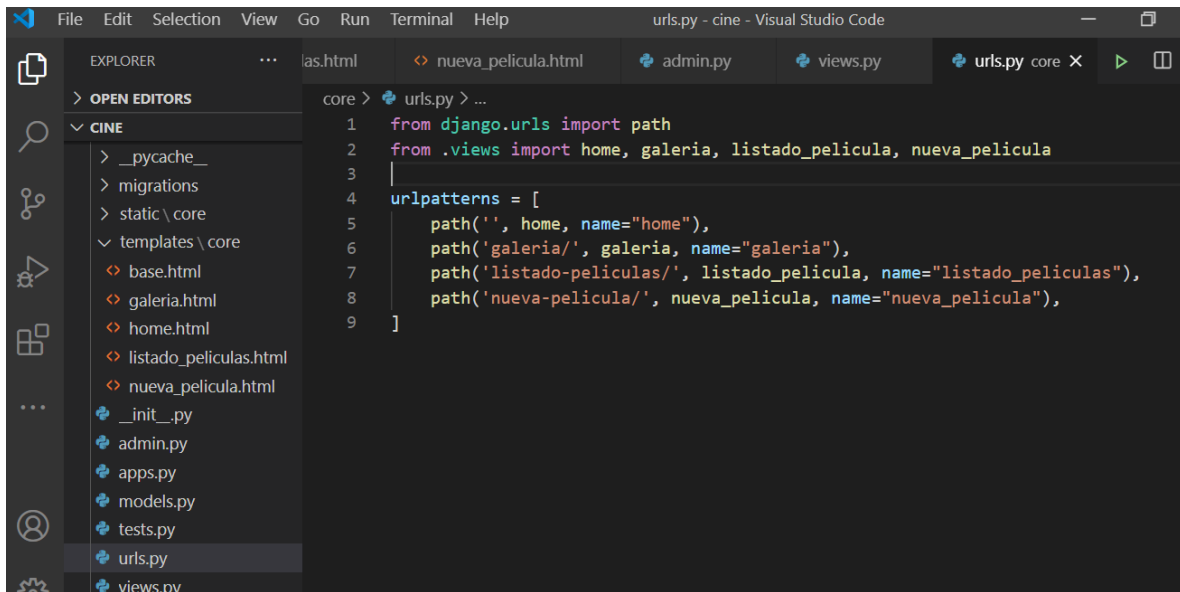
Primero Templates, luego views y finalmente urls

This screenshot shows the Visual Studio Code interface with the 'nueva_pelicula.html' file open. The Explorer sidebar on the left shows the project structure, including the 'core' directory and its subdirectories. The main editor area displays the content of the template file, which includes a Jinja2 template structure with a title 'Nueva Película'.

```
core > templates > core > nueva_pelicula.html > ...
1  {% extends 'core/base.html' %}
2
3
4  {% block contenido %}
5
6  <h2>Nueva Película</h2>
7
8  |
9
10 {% endblock %}
```

This screenshot shows the Visual Studio Code interface with the 'views.py' file open. The Explorer sidebar on the left shows the project structure, including the 'core' directory and its subdirectories. The main editor area displays the content of the 'views.py' file, which includes a Python script for handling web requests and rendering templates.

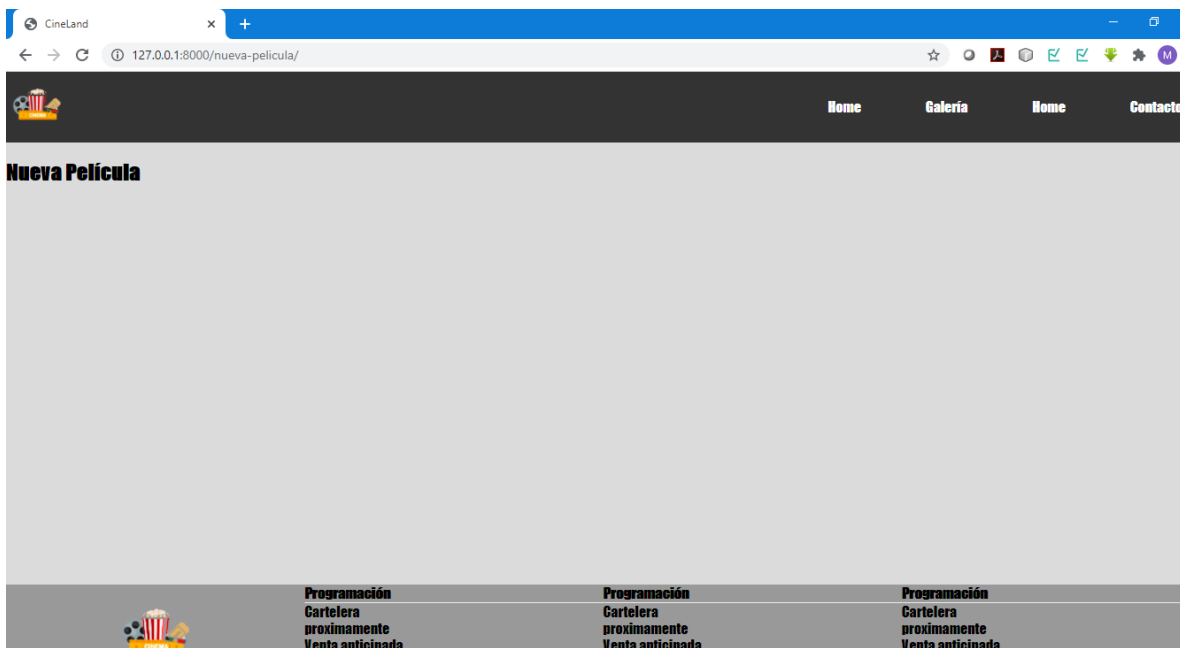
```
core > views.py > nueva_pelicula
1 from django.shortcuts import render
2
3
4 # Create your views here.
5
6 def home(request):
7     return render(request, 'core/home.html')
8
9 def galeria(request):
10    return render(request, 'core/galeria.html')
11
12 def listado_pelicula(request):
13     peliculas = Pelicula.objects.all()
14     data = {
15         'peliculas':peliculas
16     }
17     return render(request, 'core/listado_peliculas.html', data)
18
19
20 def nueva_pelicula(request):
21     return render(request, 'core/nueva_pelicula.html')
22
```



The screenshot shows the Visual Studio Code interface with the Django project structure on the left and the `urls.py` file open in the editor. The Explorer sidebar shows the project structure under the 'CINE' directory, including `__pycache__`, `migrations`, `static\core`, `templates\core`, and several HTML files. The `urls.py` file is selected, and its content is displayed in the editor. The code defines the URL patterns for the application, including the home page, gallery, list of movies, and the new movie page.

```
core > urls.py > ...
1 from django.urls import path
2 from .views import home, galeria, listado_pelicula, nueva_pelicula
3
4 urlpatterns = [
5     path('', home, name="home"),
6     path('galeria/', galeria, name="galeria"),
7     path('listado-peliculas/', listado_pelicula, name="listado_peliculas"),
8     path('nueva-pelicula/', nueva_pelicula, name="nueva_pelicula"),
9 ]
```

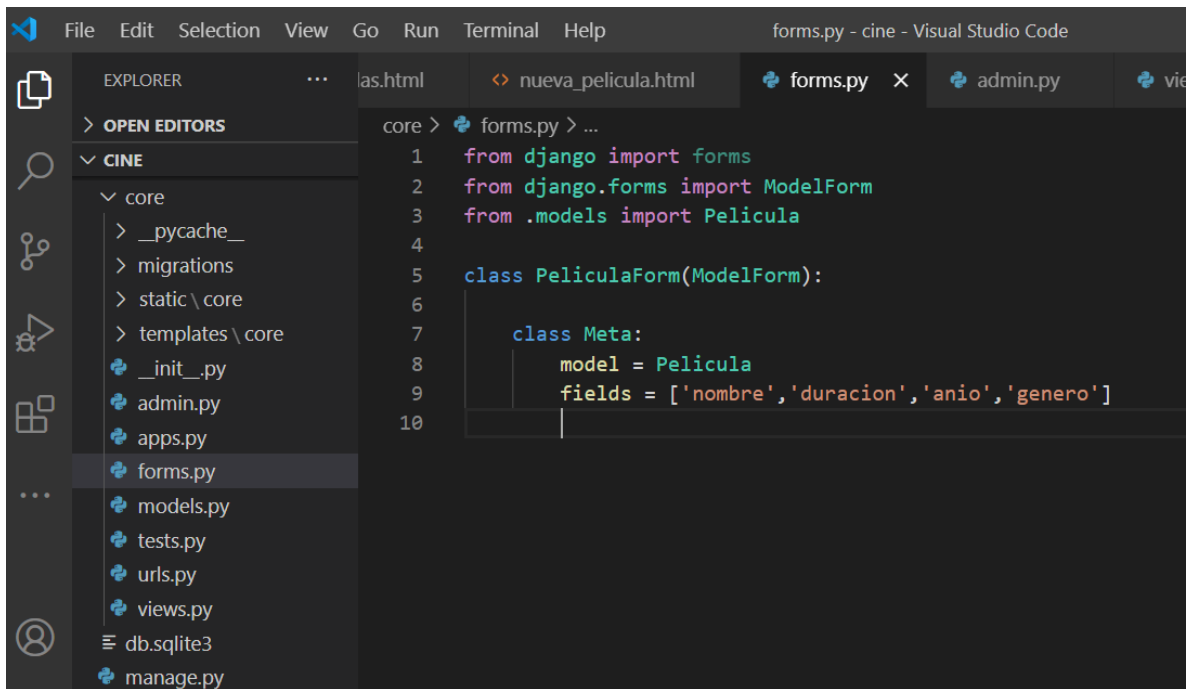
Si actualizo el navegador con nueva-pelicula/



Nos falta el formulario

Dejemos que Django se haga cargo de los campos

Creamos un archivo dentro de `core/forms.py`



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

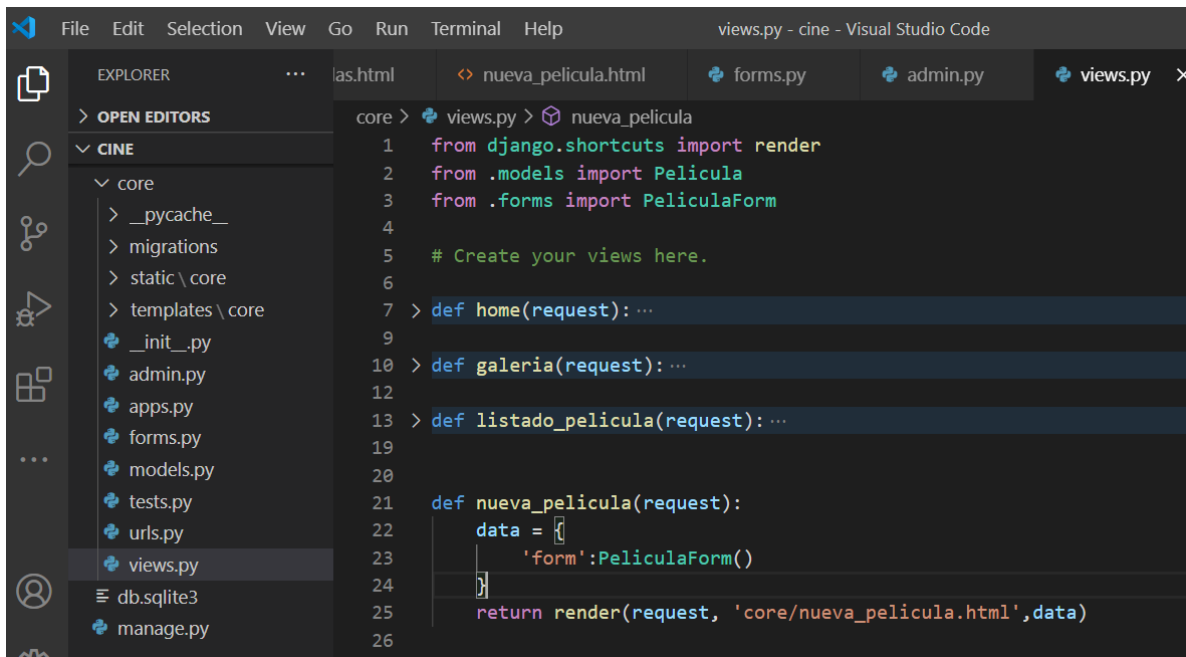
EXPLORER
> OPEN EDITORS
CINE
  core
    > __pycache__
    > migrations
    > static \ core
    > templates \ core
    __init__.py
    admin.py
    apps.py
    forms.py
    models.py
    tests.py
    urls.py
    views.py
    db.sqlite3
    manage.py

core > forms.py > ...
1 from django import forms
2 from django.forms import ModelForm
3 from .models import Pelicula
4
5 class PeliculaForm(ModelForm):
6
7     class Meta:
8         model = Pelicula
9         fields = ['nombre', 'duracion', 'anio', 'genero']
10
```

Ahora enviamos al views para que pase el formulario

Ojo, donde están los import debes importar también

```
from .forms import PeliculaForm
```

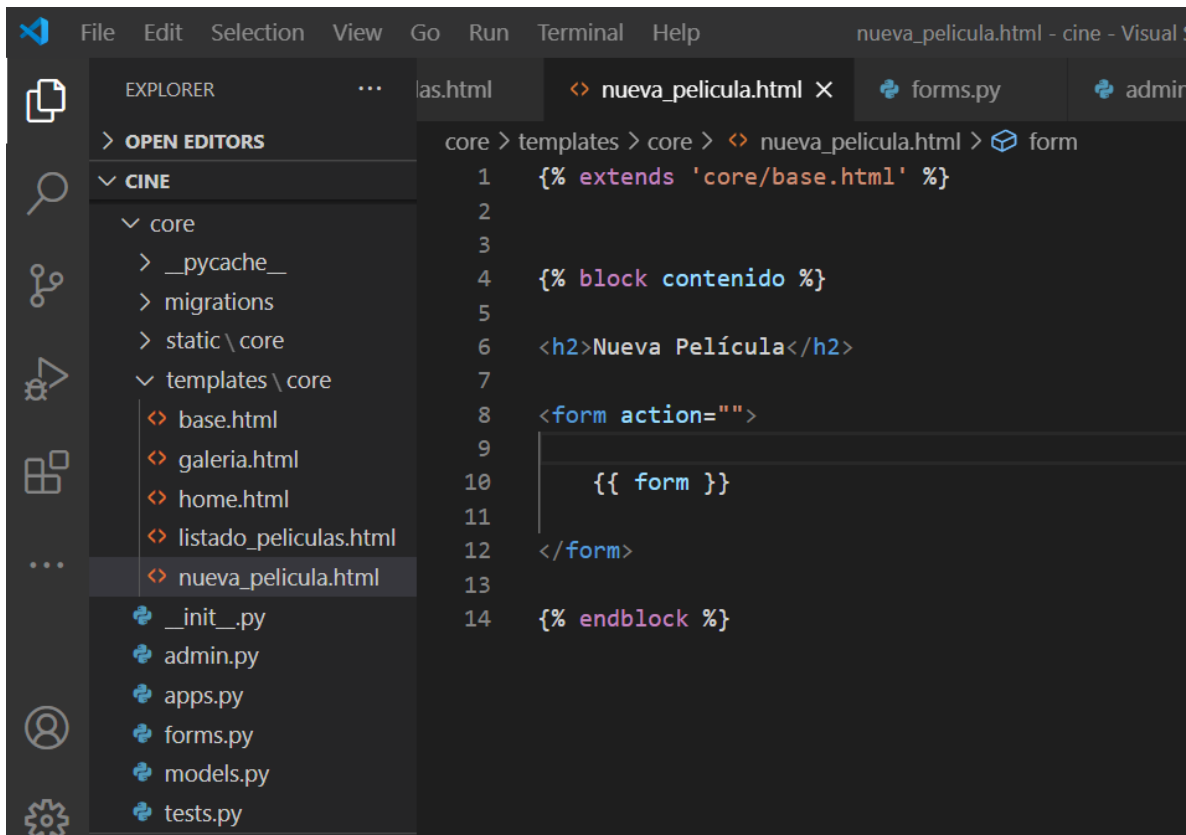


```
File Edit Selection View Go Run Terminal Help views.py - cine - Visual Studio Code

EXPLORER
> OPEN EDITORS
CINE
  core
    > __pycache__
    > migrations
    > static \ core
    > templates \ core
    __init__.py
    admin.py
    apps.py
    forms.py
    models.py
    tests.py
    urls.py
    views.py
    db.sqlite3
    manage.py

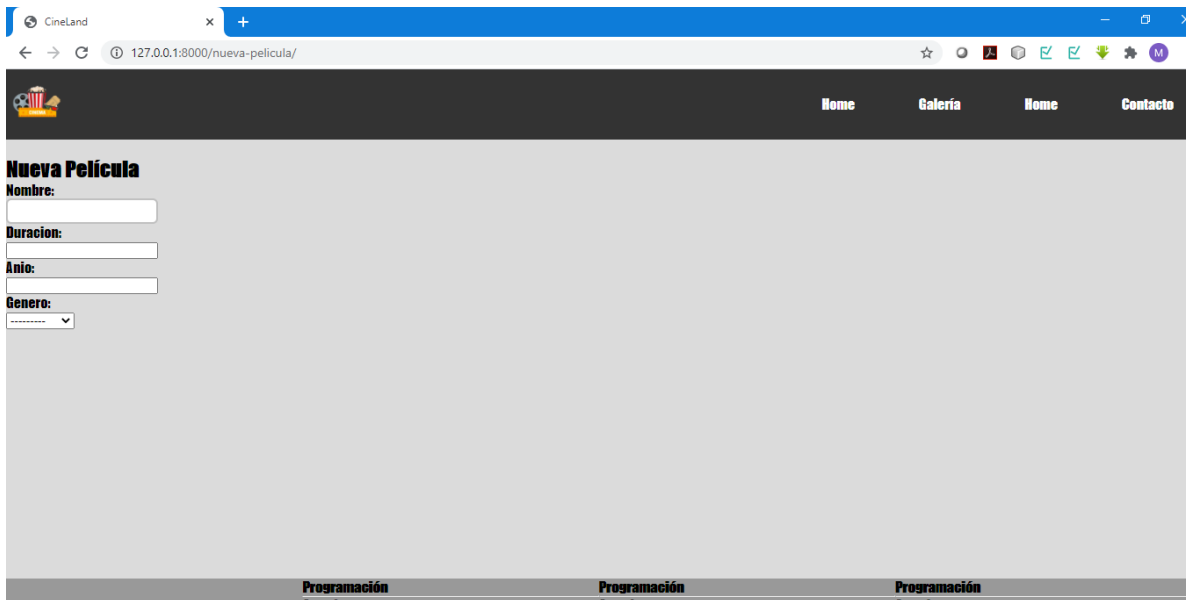
core > views.py > nueva_pelicula
1 from django.shortcuts import render
2 from .models import Pelicula
3 from .forms import PeliculaForm
4
5 # Create your views here.
6
7 > def home(request): ...
8
9
10 > def galeria(request): ...
11
12
13 > def listado_pelicula(request): ...
14
15
16
17
18
19
20
21 def nueva_pelicula(request):
22     data = {}
23     'form': PeliculaForm()
24
25     return render(request, 'core/nueva_pelicula.html', data)
26
```

Y ahora modificamos el template



```
core > templates > core > <> nueva_pelicula.html > form
1  {% extends 'core/base.html' %}
2
3
4  {% block contenido %}
5
6  <h2>Nueva Película</h2>
7
8  <form action="">
9
10     {{ form }}
11
12 </form>
13
14 {% endblock %}
```

Si actualizas el navegador



Trajo automáticamente los campos incluyendo el listado de Géneros,

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the project structure. The 'CINE' folder is expanded, showing subfolders like 'core' and 'templates', and files like 'base.html', 'galeria.html', 'home.html', 'listado_peliculas.html', and 'nueva_pelicula.html'. The 'nueva_pelicula.html' file is selected and its content is displayed in the main editor. The code is a Django template that extends 'core/base.html' and contains a form for adding a new movie.

```
core > templates > core > <nueva_pelicula.html> form
1  {% extends 'core/base.html' %}
2
3
4  {% block contenido %}
5
6  <h2>Nueva Película</h2>
7
8  <form action="" method="POST">
9      {% csrf_token %}
10     <table>
11         {{ form.as_table }}
12     </table>
13     <input type="reset" value="Limpiar">
14     <input type="submit" value="Guardar">
15
16 </form>
17
18 {% endblock %}
```

Modificamos el views.py para que envíe los datos a la BBDD

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The 'views.py' file is selected and its content is displayed in the main editor. The code defines several Django view functions: 'home', 'galeria', 'listado_pelicula', and 'nueva_pelicula'. The 'nueva_pelicula' function is the focus, showing the logic for handling a POST request, validating the form, saving it to the database, and returning a success message.

```
core > views.py > nueva_pelicula
6
7 > def home(request): ...
9
10 > def galeria(request): ...
12
13 > def listado_pelicula(request): ...
19
20
21 def nueva_pelicula(request):|
22     data = {
23         'form': PeliculaForm()
24     }
25
26     if request.method == 'POST':
27         formulario = PeliculaForm(request.POST)
28         if formulario.is_valid():
29             formulario.save()
30             data['mensaje'] = "Guardado correctamente"
31
32     return render(request, 'core/nueva_pelicula.html', data)
```

Y en nueva_pelicula.html le agregamos el mensaje de salida

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

EXPLORER ... las.html < nueva_pelicula.html x forms.py admin.py views.py

> OPEN EDITORS

▼ CINE

- base.html
- galeria.html
- home.html
- listado_peliculas.html
- nueva_pelicula.html

__init__.py admin.py apps.py forms.py models.py tests.py urls.py views.py db.sqlite3 manage.py

> JAVA PROJECTS

core > templates > core > < nueva_pelicula.html > form

```
1 {% extends 'core/base.html' %}
2
3
4 {% block contenido %}
5
6 <h2>Nueva Película</h2>
7
8 <form action="" method="POST">
9     {% csrf_token %}
10     <table>
11         {{ form.as_table }}
12     </table>
13     <input type="reset" value="Limpiar">
14     <input type="submit" value="Guardar">
15
16 </form>
17
18 {{ mensaje }}
19
20 {% endblock %}
```