



## MANUAL - DJANGO

NOMBRE:

GUALA EDISON

DOCENTE:

ING. RENE QUISAGUANO



UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

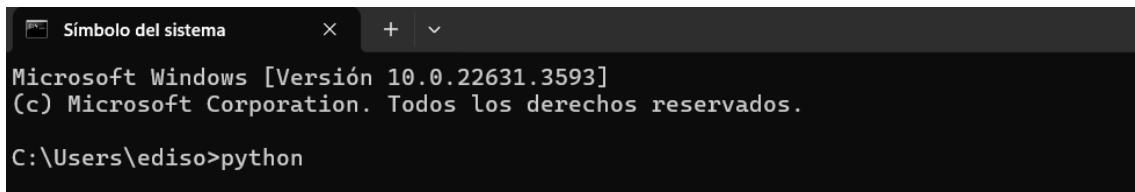
DESARROLLO Y APLICACIONES WEB

ABRIL 2024 – AGOSTO 2024

## MANUAL PARA LA INSTALACIÓN DEL FRAMEWORK DJANGO

### PASO 1

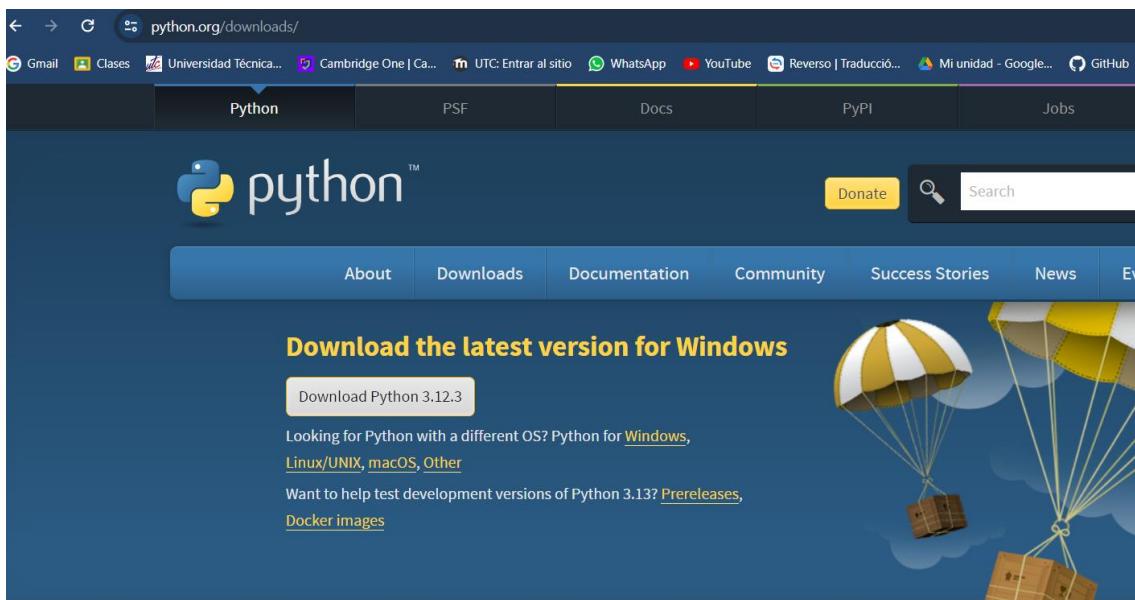
Abrimos un cmd para verificar si está instalado Python, nos vamos al cmd y verificamos escribiendo **Python**.



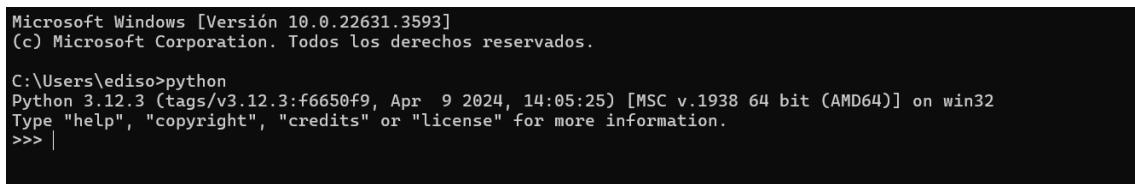
```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.3593]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ediso>python
```

En el caso de que no esté instalado Python, entonces nos dirigimos a la página oficial de Python y descargamos el instalador e instalamos Python.



Una vez hecho eso volvemos al cmd y verificamos nuevamente si está instalado Python, y si todo está correcto, entonces nos debería aparecer la versión de Python, tal como se ve en la imagen:

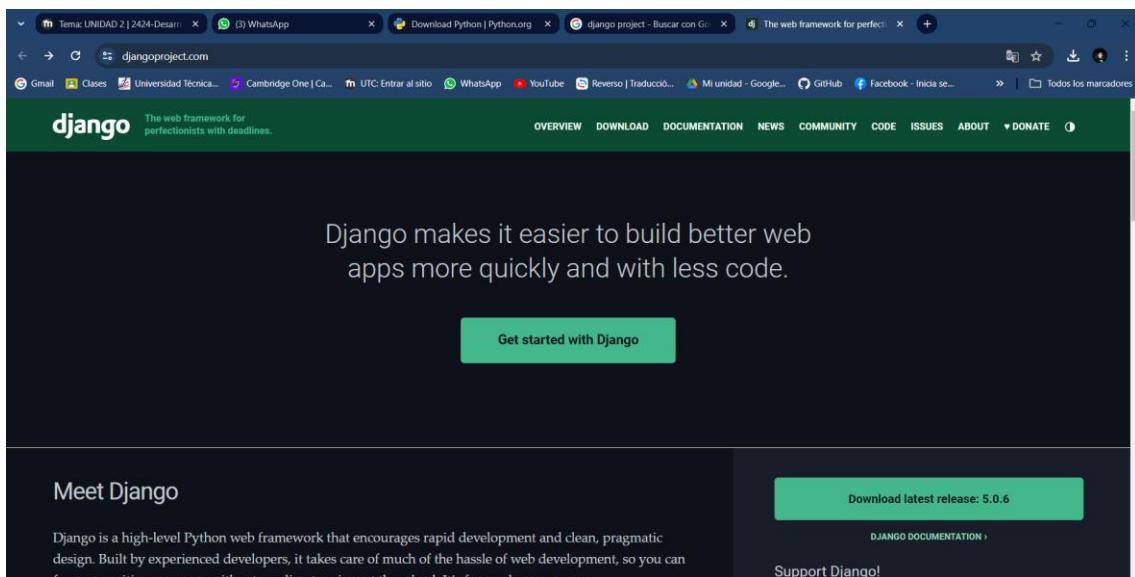


```
Microsoft Windows [Versión 10.0.22631.3593]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ediso>python
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

### PASO 2

Vamos a la página de django y damos clic en Download latest reléase para descargar.



Se nos desplegará una nueva pantalla, y a continuación, revisamos los comandos de instalación, los cuales nos servirán para aplicarlos en nuestra máquina a través del cmd.

A screenshot of a web browser displaying the 'How to get Django' documentation page. The page title is 'How to get Django'. It contains text about the availability of Django under the BSD license and its support for Python 3. Below this, there's a section titled 'Option 1: Get the latest official version' with instructions to read the 5.0.6 release notes and install it with pip. It provides separate commands for Linux/macOS and Windows. The browser's address bar shows 'djangoproject.com'.

### PASO 3

Abrimos el cmd y hacemos uso de los comandos para instalar django, en este caso instalaremos la versión 4.0.6 para que sea más estable, entonces digitamos ***pip install Django==4.0.6***.

```
Microsoft Windows [Versión 10.0.22631.3593]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ediso>pip install Django==4.0.6
Collecting Django==4.0.6
  Downloading Django-4.0.6-py3-none-any.whl.metadata (4.0 kB)
Collecting asgiref<4,>=3.4.1 (from Django==4.0.6)
  Downloading asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.2.2 (from Django==4.0.6)
  Downloading sqlparse-0.5.0-py3-none-any.whl.metadata (3.9 kB)
Collecting tzdata (from Django==4.0.6)
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
  Downloading Django-4.0.6-py3-none-any.whl (8.0 MB)
    8.0/8.0 MB 2.8 MB/s eta 0:00:00
  Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
  Downloading sqlparse-0.5.0-py3-none-any.whl (43 kB)
    44.0/44.0 kB ? eta 0:00:00
  Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
    345.4/345.4 kB 10.8 MB/s eta 0:00:00
Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-4.0.6 asgiref-3.8.1 sqlparse-0.5.0 tzdata-2024.1

C:\Users\ediso>
```

#### PASO 4

Importamos django, para eso abrimos cmd, primero colocamos ***Python***, me debe aparecer la versión, y luego escribimos ***import django***.

```
C:\Users\ediso>python
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25)
Type "help", "copyright", "credits" or "license" for more
>>> import django
```

Si todo está bien entonces nos debe aparecer otra línea solo con 3 diamantes de cierre **>>>**.

```
>>> |
```

#### PASO 5

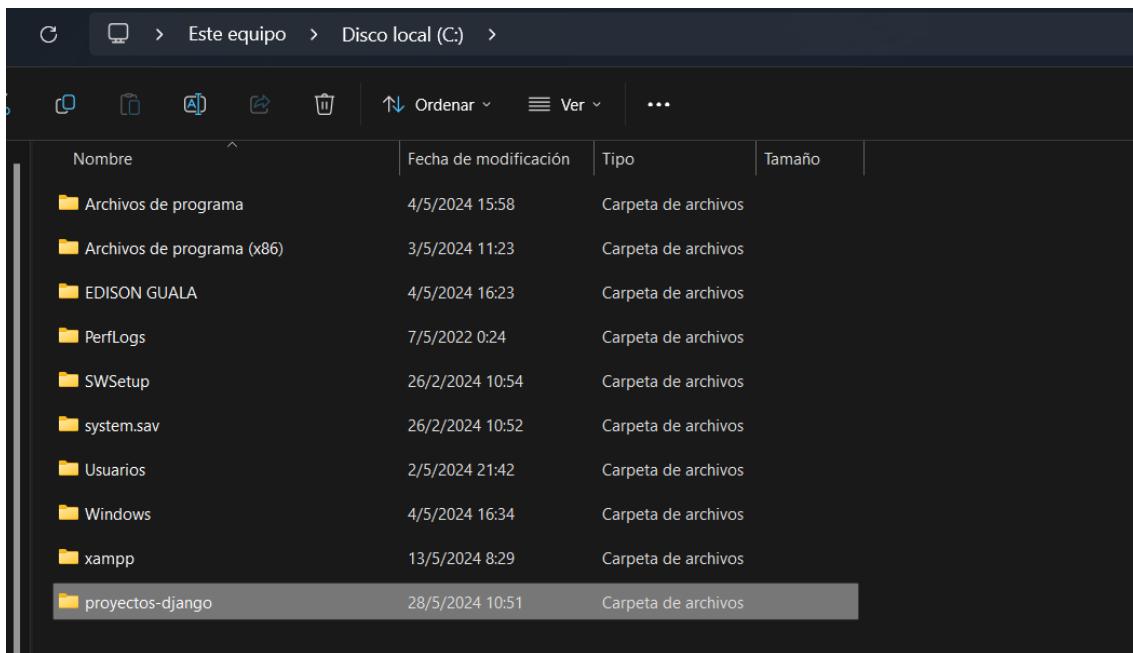
A continuación, verificamos la versión instalada de django, para eso colocamos el comando

***django.VERSION***

```
>>> django.VERSION
(4, 0, 6, 'final', 0)
>>> |
```

## PASO 6

Creamos una carpeta en el **disco C** llamada **proyectos-django**



## A PARTIR DE ESTE PASO SE CREARÁN LOS PROYECTOS

## PASO 7

A continuación, vamos a crear una carpeta a través de comandos, para eso abrimos el cmd y colocamos **cd /** para redirigir a otra ubicación, entonces escribimos la nueva ruta donde se guardarán los proyectos de django, en este caso será **cd proyectos-django**

```
C:\Windows\system32\cmd.exe + ^

Microsoft Windows [Versión 10.0.22631.3593]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ediso>cd /
C:\>cd proyectos-django
C:\proyectos-django>django-admin startproject cineUTC
C:\proyectos-django>
```

## PASO 8

Le damos un nombre al proyecto, en este caso ***django-admin startproject cineUTC***

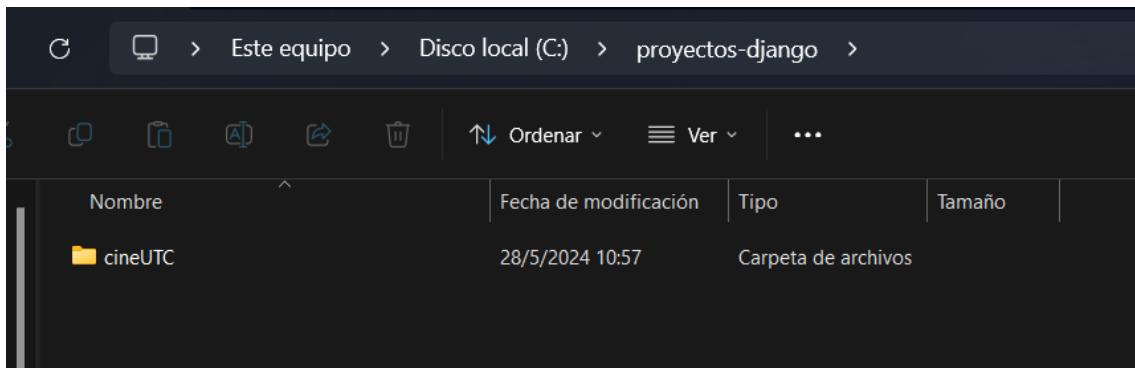
```
C:\proyectos-django>django-admin startproject cineUTC
```

Y si damos en ejecutar nos debe aparecer la siguiente línea, lo que significa que todo está correcto.

```
C:\proyectos-django>django-admin startproject cineUTC
```

```
C:\proyectos-django>
```

Y para verificar podemos ir al explorador de archivos y se puede ver que se ha creado correctamente una nueva carpeta.



## PASO 9

Vamos a ingresar a la carpeta cineUTC utilizando el comando ***cd cineUTC***

```
C:\proyectos-django>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 22CC-EFB2

Directorio de C:\proyectos-django

28/05/2024  10:57    <DIR>          .
28/05/2024  10:57    <DIR>          cineUTC
                           0 archivos           0 bytes
                           2 dirs   303.220.477.952 bytes libres
```

```
C:\proyectos-django>cd cineUTC
```

```
C:\proyectos-django\cineUTC>
```

## PASO 10

Vamos a crear una carpeta para guardar las aplicaciones, entonces colocamos ***mkdir Aplicaciones***

```
C:\proyectos-django>cd cineUTC  
C:\proyectos-django\cineUTC>mkdir Aplicaciones  
C:\proyectos-django\cineUTC>
```

Verificamos la creación por modo consola o modo gráfico.

```
C:\proyectos-django\cineUTC>dir  
El volumen de la unidad C no tiene etiqueta.  
El número de serie del volumen es: 22CC-EFB2  
  
Directorio de C:\proyectos-django\cineUTC  
  
28/05/2024 11:06 <DIR> .  
28/05/2024 10:57 <DIR> ..  
28/05/2024 11:06 <DIR> Aplicaciones  
28/05/2024 10:57 <DIR> cineUTC  
28/05/2024 10:57 685 manage.py  
    1 archivos          685 bytes  
    4 dirs   303.220.682.752 bytes libres
```

```
C:\proyectos-django\cineUTC>
```

📁 cineUTC	28/5/2024 10:57	Carpeta de archivos
📄 manage	28/5/2024 10:57	Archivo de origen ... 1 KB
📁 Aplicaciones	28/5/2024 11:06	Carpeta de archivos

## PASO 11

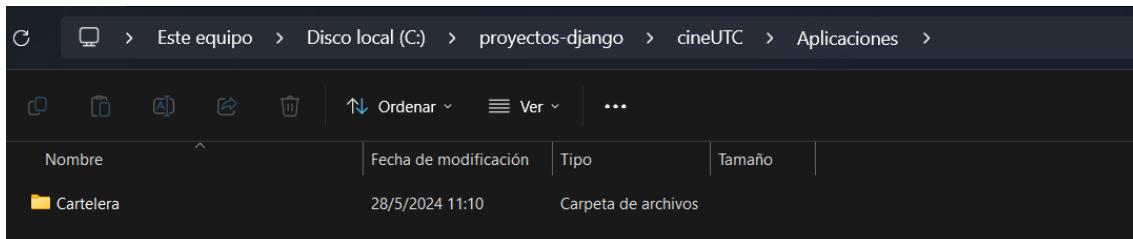
Ingresamos a la carpeta Aplicaciones, colocamos ***cd Aplicaciones***

```
C:\proyectos-django\cineUTC>cd Aplicaciones  
C:\proyectos-django\cineUTC\Aplicaciones>
```

Creamos una carpeta llamada Cartelera, para eso escribimos ***django-admin startapp Cartelera***

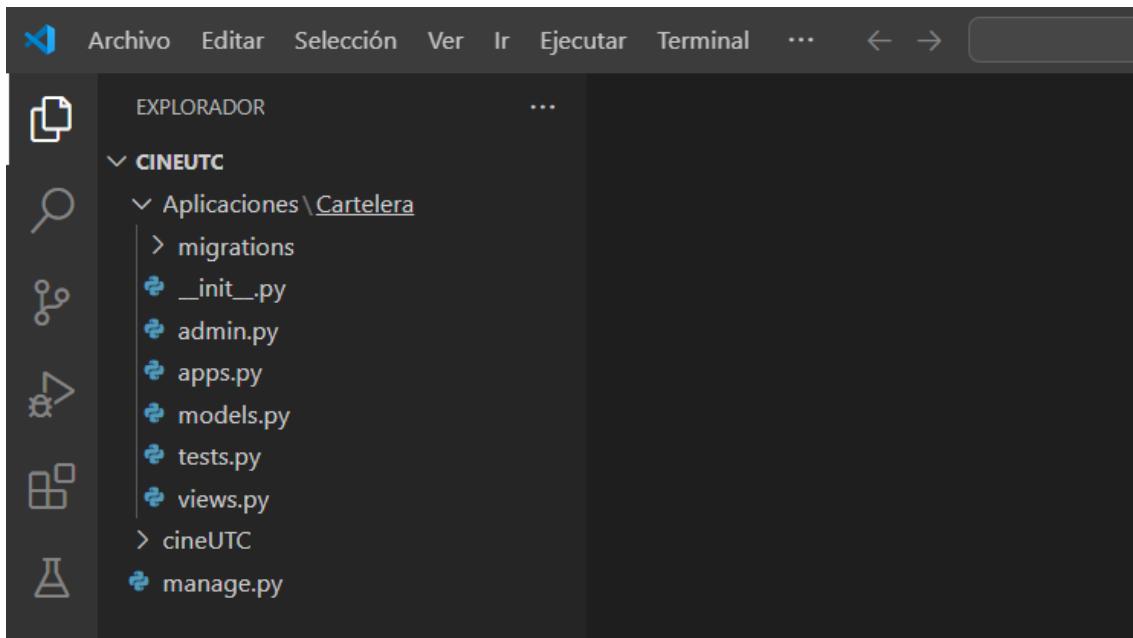
```
C:\proyectos-django\cineUTC>cd Aplicaciones  
C:\proyectos-django\cineUTC\Aplicaciones>django-admin startapp Cartelera  
C:\proyectos-django\cineUTC\Aplicaciones>
```

Verificamos que se haya creado correctamente.



## PASO 12

Abrimos Visual Studio Code y hacemos la apertura de la carpeta llamada cineUTC



## PASO 13

Trabajaremos en **settings.py**, esta es la configuración general, es decir afectara a todas las aplicaciones. Nos ubicamos en la línea 33

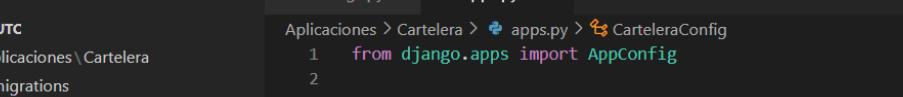
```
cineUTC > settings.py > ...
23 SECRET_KEY = 'django-insecure-@+y6qgc_5a^--cp/8j!1^g1'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40 ]
41
42 MIDDLEWARE = [
43     'django.middleware.security.SecurityMiddleware',
```

A continuación, registramos la aplicación cartelera dentro de nuestro proyecto

```
cineUTC > settings.py > ...
23 SECRET_KEY = 'django-insecure-@+y6qgc_5a^--cp/8j!1^g1'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
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     'Aplicaciones.Cartelera',
41 ]
42
```

## PASO 14

Nos ubicamos en este apartado **apps.py**, esto es la configuración solo de Cartelera



The screenshot shows the PyCharm interface with the following details:

- File Menu:** ARCHIVO, Editar, Selección, Ver, Ejecutar, Terminal.
- SIDE BAR:** EXPLORADOR (selected), showing the project structure:
  - CINEUTC
    - Aplicaciones\Cartelera
      - migrations
      - \_\_init\_\_.py
      - admin.py
      - apps.py** (selected)
      - models.py
      - tests.py
      - views.py
    - cineUTC
      - \_\_init\_\_.py
- EDITOR:** The current file is `apps.py` under `Aplicaciones\Cartelera`. The code content is:

```
from django.apps import AppConfig

class CarteleraConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'Cartelera'
```

Colocamos lo siguiente dentro del archivo *apps.py*

The screenshot shows the PyCharm interface with the following details:

- File Structure (EXPLORADOR):** The project tree shows the `CINEUTC` application under `Aplicaciones\Cartelera`. Inside `Cartelera`, there are files: `migrations`, `__init__.py`, `admin.py`, `apps.py` (selected), `models.py`, `tests.py`, and `views.py`. There is also a `cineUTC` folder containing `__init__.py`.
- Current File (apps.py):** The code editor displays the `apps.py` file from the `Cartelera` application. The code defines a class `CarteleraConfig` that inherits from  `AppConfig` . It sets the `default_auto_field` to `'django.db.models.BigAutoField'` and the `name` to `'Aplicaciones.Cartelera'`.
- Toolbar:** The top bar includes icons for File, Edit, Selection, View, Go, Execute, Terminal, and Help.
- Status Bar:** The status bar on the right shows the project name `cineUTC`.

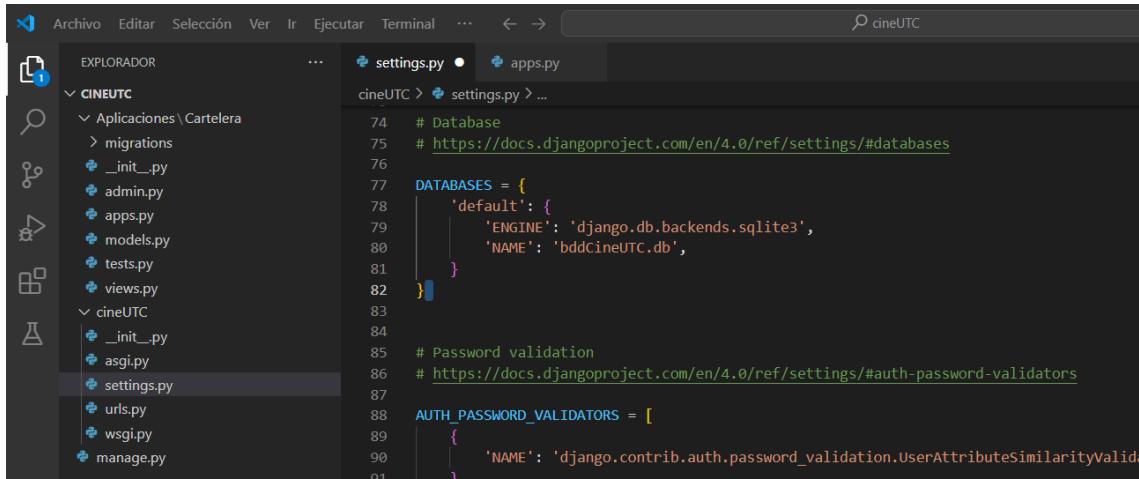
## PASO 15

Configuración de una conexión a una base de datos, nos ubicamos en la línea 77

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists the project structure under 'CINEUTC'. The 'settings.py' file in the 'cineUTC' directory is currently selected and highlighted in the Explorer. The main editor area displays the contents of 'settings.py', which includes configuration for databases and password validation. The status bar at the bottom shows the file path 'cineUTC > settings.py > ...'.

```
EXPLORADOR
...
settings.py x apps.py
cineUTC > settings.py > ...
76
77 DATABASES = {
78     'default': {
79         'ENGINE': 'django.db.backends.sqlite3',
80         'NAME': BASE_DIR / 'db.sqlite3',
81     }
82 }
83
84
85 # Password validation
86 # https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-
87
88 AUTH_PASSWORD_VALIDATORS = [
89     {
90         'NAME': 'django.contrib.auth.password_validation.UserAttribute-
91     },
92     {
93         'NAME': 'django.contrib.auth.password_validation.MinimumLength-
94     }
95 ]
```

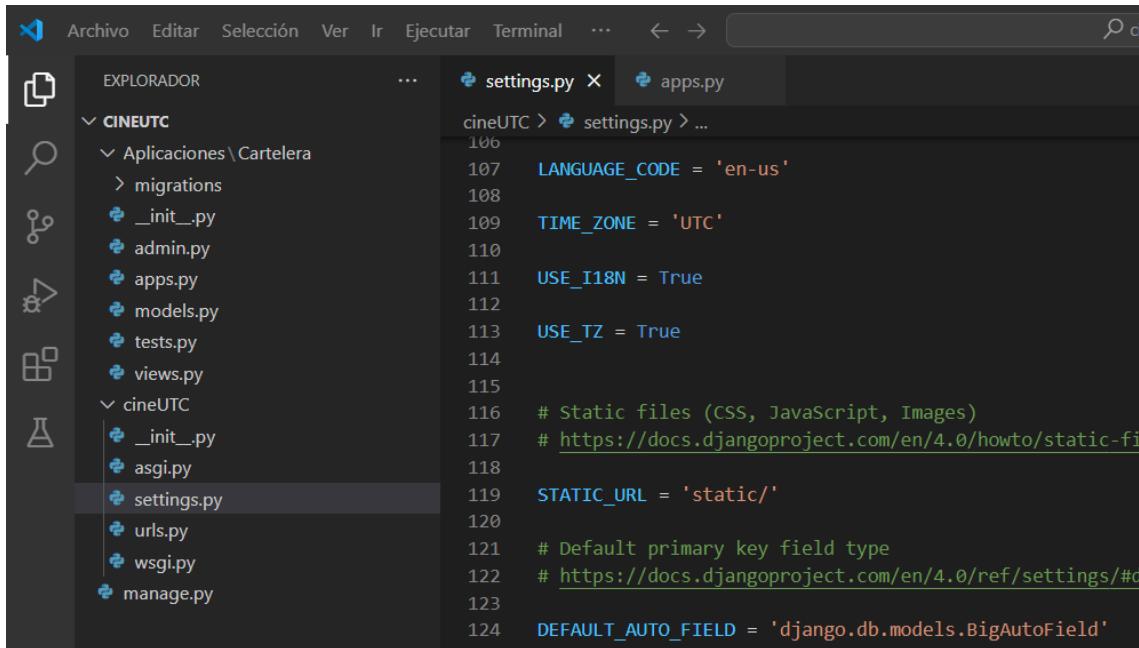
Cambiamos el nombre para nuestra base de datos



```
74 # Database
75 # https://docs.djangoproject.com/en/4.0/ref/settings/#databases
76
77 DATABASES = {
78     'default': {
79         'ENGINE': 'django.db.backends.sqlite3',
80         'NAME': 'bddCineUTC.db',
81     }
82 }
83
84
85 # Password validation
86 # https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validators
87
88 AUTH_PASSWORD_VALIDATORS = [
89     {
90         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator'
91     },
92     {
93         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator'
94     },
95     {
96         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator'
97     },
98     {
99         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator'
100    }
101]
```

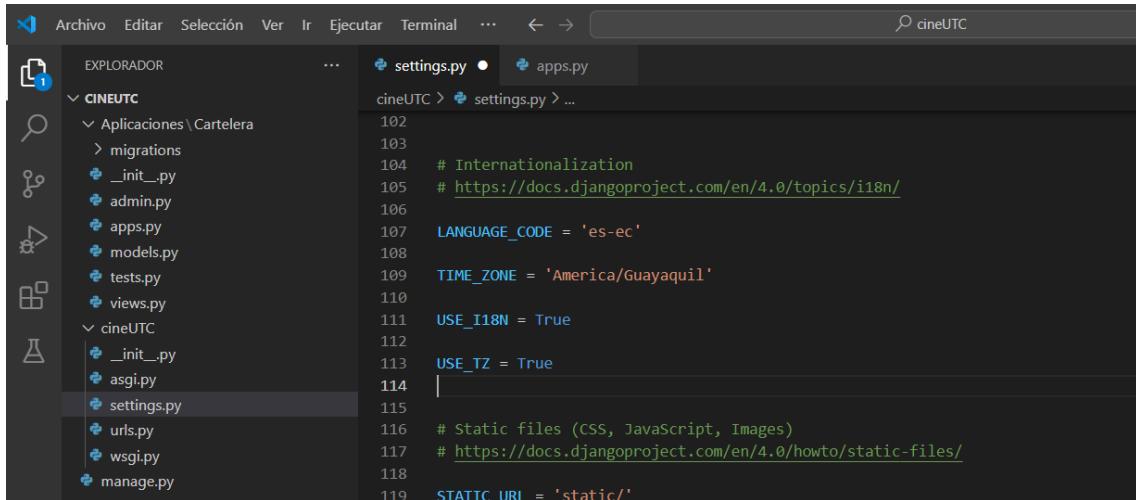
## PASO 16

Configuraremos el idioma y la zona horaria



```
106
107 LANGUAGE_CODE = 'en-us'
108
109 TIME_ZONE = 'UTC'
110
111 USE_I18N = True
112
113 USE_TZ = True
114
115
116 # Static files (CSS, JavaScript, Images)
117 # https://docs.djangoproject.com/en/4.0/howto/static-files/
118
119 STATIC_URL = 'static/'
120
121 # Default primary key field type
122 # https://docs.djangoproject.com/en/4.0/ref/settings/#db-primary-key
123
124 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

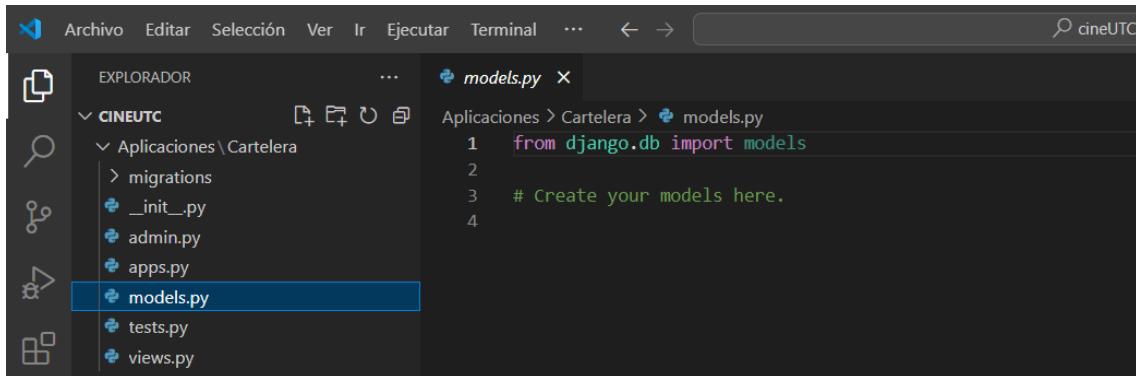
Colocamos en español ecuador para el idioma, y America/Guayaquil para la zona horaria



```
cineUTC > settings.py > ...
102
103
104 # Internationalization
105 # https://docs.djangoproject.com/en/4.0/topics/i18n/
106
107 LANGUAGE_CODE = 'es-ec'
108
109 TIME_ZONE = 'America/Guayaquil'
110
111 USE_I18N = True
112
113 USE_TZ = True
114
115
116 # Static files (CSS, JavaScript, Images)
117 # https://docs.djangoproject.com/en/4.0/howto/static-files/
118
119 STATIC_URL = 'static/'
```

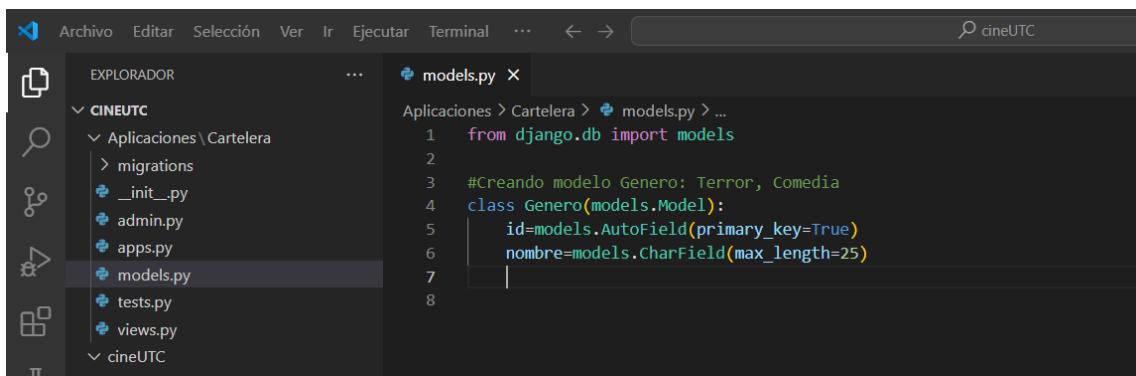
## PASO 17

Ahora vamos a este apartado **models.py**



```
Aplicaciones > Cartelera > models.py
1 from django.db import models
2
3 # Create your models here.
4
```

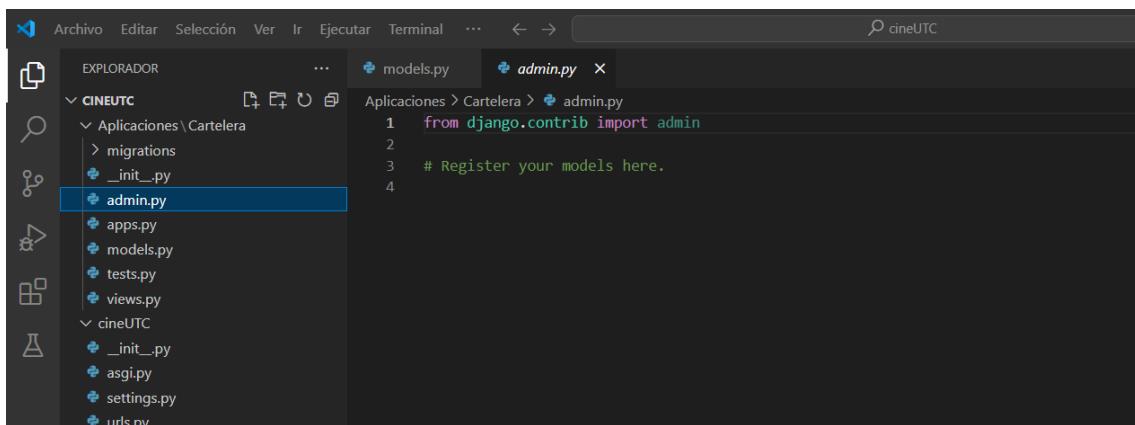
A continuación, creamos un modelo.



```
Aplicaciones > Cartelera > models.py > ...
1 from django.db import models
2
3 #Creando modelo Genero: Terror, Comedia
4 class Genero(models.Model):
5     id=models.AutoField(primary_key=True)
6     nombre=models.CharField(max_length=25)
7
8
```

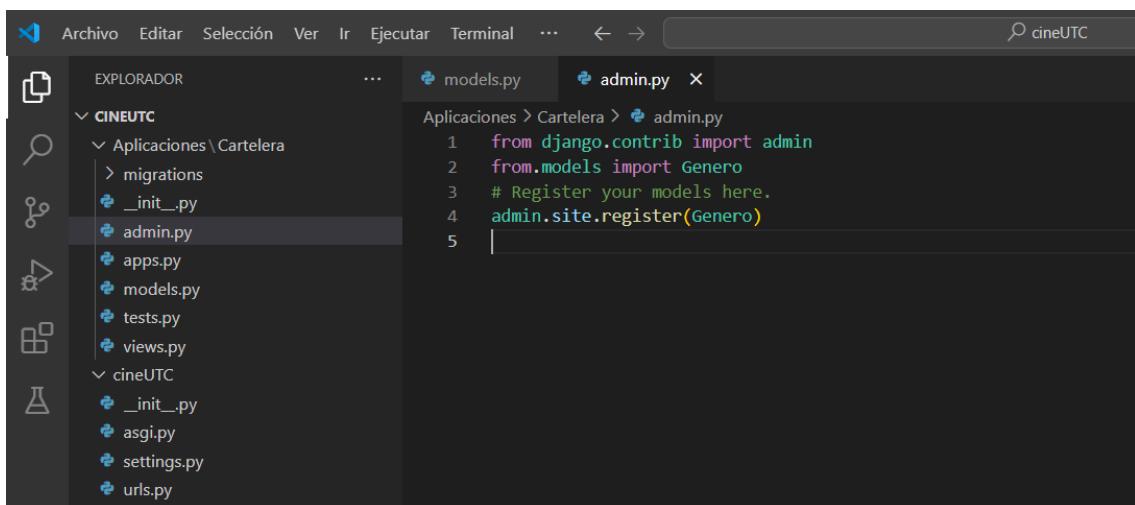
## PASO 18

Nos ubicamos en **admin.py**



```
from django.contrib import admin
from .models import Genero
# Register your models here.
```

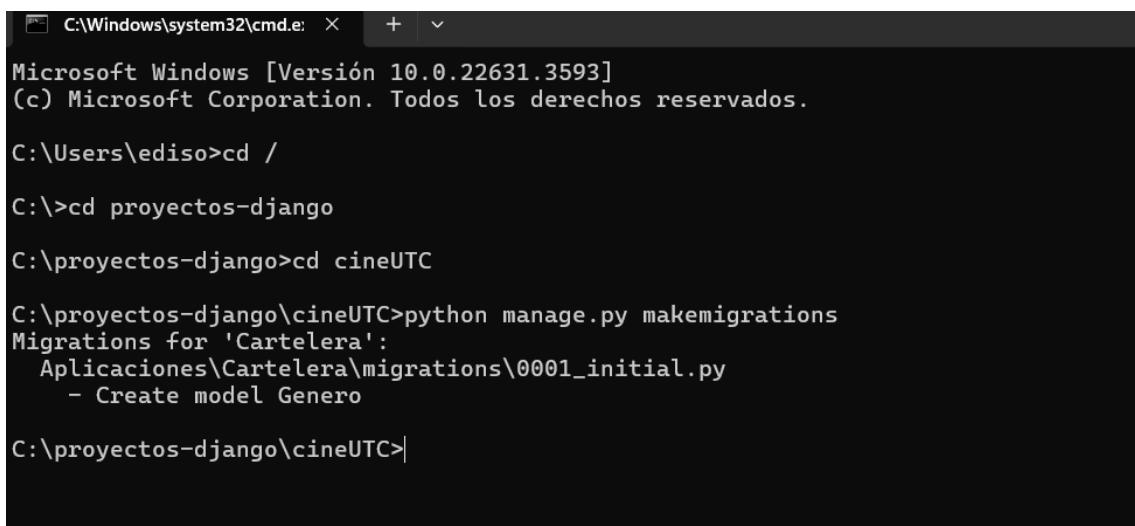
Añadimos los siguientes códigos para el CRUD



```
from django.contrib import admin
from .models import Genero
# Register your models here.
admin.site.register(Genero)
```

## PASO 19

Ahora nos vamos al cmd y colocamos el siguiente comando para crear un modelo: **python manage.py makemigrations**



```
C:\Windows\system32\cmd.exe: × + ▾
Microsoft Windows [Versión 10.0.22631.3593]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ediso>cd /
C:\>cd proyectos-django
C:\proyectos-django>cd cineUTC
C:\proyectos-django\cineUTC>python manage.py makemigrations
Migrations for 'Cartelera':
  Aplicaciones\Cartelera\migrations\0001_initial.py
    - Create model Genero

C:\proyectos-django\cineUTC>
```

A continuación, ejecutamos este comando para hacer las migraciones: ***python manage.py migrate***

```
C:\proyectos-django\cineUTC>python manage.py makemigrations
Migrations for 'Cartelera':
  Aplicaciones\Cartelera\migrations\0001_initial.py
    - Create model Genero

C:\proyectos-django\cineUTC>python manage.py migrate
Operations to perform:
  Apply all migrations: Cartelera, admin, auth, contenttypes, sessions
Running migrations:
  Applying Cartelera.0001_initial... OK
  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:\proyectos-django\cineUTC>
```

A continuación, debemos crear un super usuario, para eso hacemos uso del siguiente comando: ***python manage.py createsuperuser***, una vez ejecutado ese comando nos pedirá cierta información, tal como se muestra en la imagen.

```
Applying sessions.0001_initial... OK

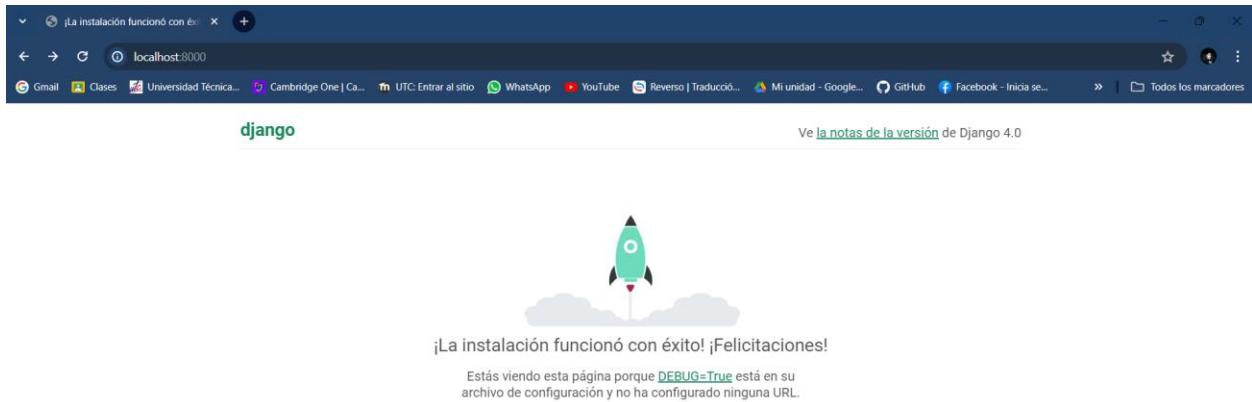
C:\proyectos-django\cineUTC>python manage.py createsuperuser
Nombre de usuario (leave blank to use 'ediso'): javi
Dirección de correo electrónico: edison.guala2040@utc.edu.ec
Password:
Password (again):
Esta contraseña es demasiado corta. Debe contener al menos 8 caracteres.
Esta contraseña es demasiado común.
Esta contraseña es completamente numérica.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

C:\proyectos-django\cineUTC>
```

Colocamos un último comando para activar el servidor, entonces escribimos ***python manage.py runserver***, ejecutamos y nos aparecerá un link ***http://127.0.0.1:8000/***, a continuación, copiamos ese link en cualquier navegador de preferencia.

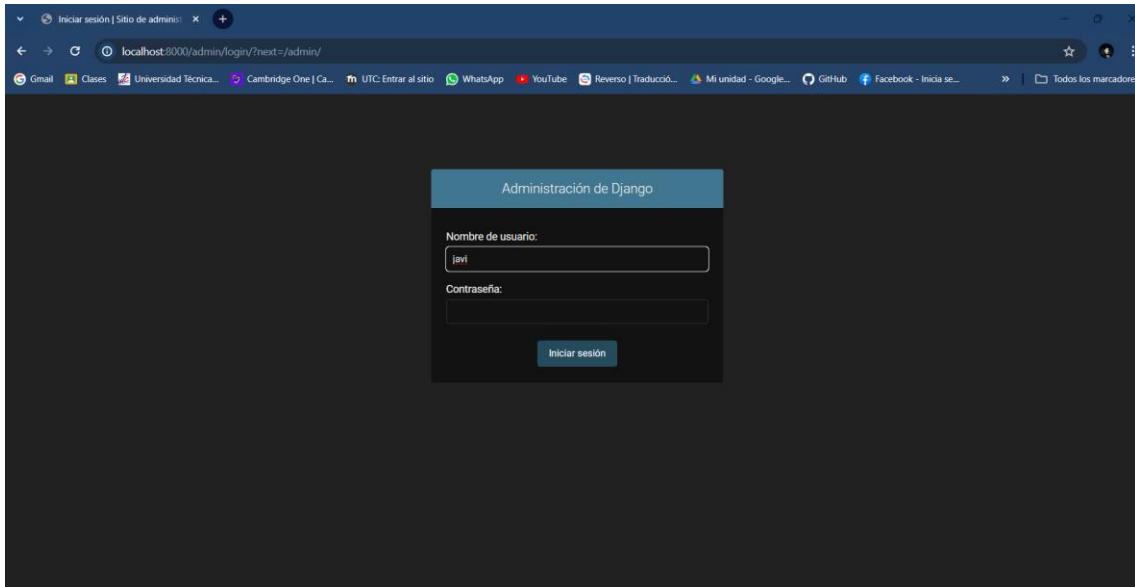
```
Esta contraseña es completamente numérica.  
Bypass password validation and create user anyway? [y/N]: y  
Superuser created successfully.  
  
C:\proyectos-django\cineUTC>python manage.py runserver  
Watching for file changes with StatReloader  
Performing system checks...  
  
System check identified no issues (0 silenced).  
May 28, 2024 - 12:08:38  
Django version 4.0.6, using settings 'cineUTC.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.  
[28/May/2024 12:08:53] "GET / HTTP/1.1" 200 10749  
[28/May/2024 12:08:53] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423  
[28/May/2024 12:08:53] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 86184  
[28/May/2024 12:08:53] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85876  
[28/May/2024 12:08:53] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692  
Not Found: /favicon.ico  
[28/May/2024 12:08:54] "GET /favicon.ico HTTP/1.1" 404 2111
```

Nos aparecerá la siguiente pantalla, lo que significa que todo va bien.



## PASO 20

En el navegador escribimos: **localhost:8000/admin/**, nos aparecerá una pantalla donde debemos iniciar sesión con los datos correspondientes al superusuario creado anteriormente.



Una vez iniciado sesión, finalmente llegamos a esta interfaz donde podremos hacer registros

Sitio administrativo

AUTENTICACIÓN Y AUTORIZACIÓN

Grupos	+ Añadir	Modificar
Usuarios	+ Añadir	Modificar

CARTELERA

Generos	+ Añadir	Modificar
---------	----------	-----------

BIENVENIDOS, JAVI. VER

Acciones recientes

Mis acciones

Ninguno disponible

Los registros quedarían así:

Bien, ahora para que la parte de los registros no nos aparezca “Genero object()”, entonces debemos crear una función def el cual nos permitirá darle diseño a esas filas

```

EXPLORADOR
... models.py admin.py
Aplicaciones > Cartelera > models.py > Director
1   from django.db import models
2
3   #Creando modelo Genero: Terror, Comedia
4   class Genero(models.Model):
5       id=models.AutoField(primary_key=True)
6       nombre=models.CharField(max_length=25)
7       #cuando se desee añadir un nuevo campo a un modelo que ya fue creado y donde ya se hizo registros,
8       #debemos colocar default=' ' para que no nos arroje errores y nos acepte el nuevo campo
9       descripcion=models.CharField(max_length=150, default='')
10
11      def __str__(self):
12          fila="{0}: {1} - {2}"
13          return fila.format(self.id, self.nombre, self.descripcion)
14

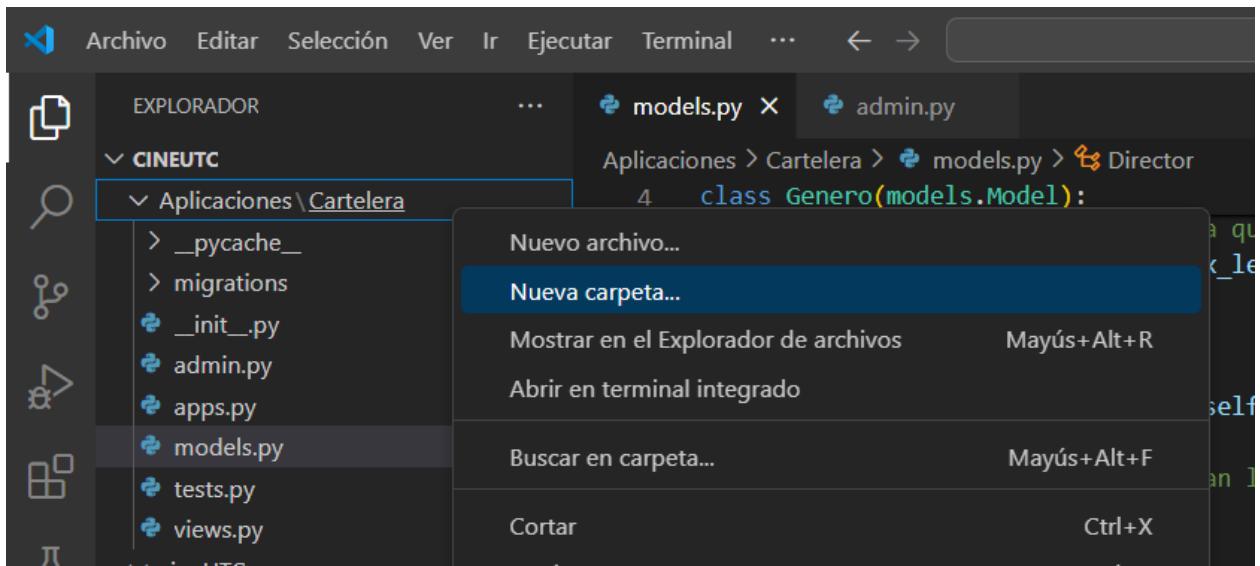
```

Y nos quedaría de la siguiente manera

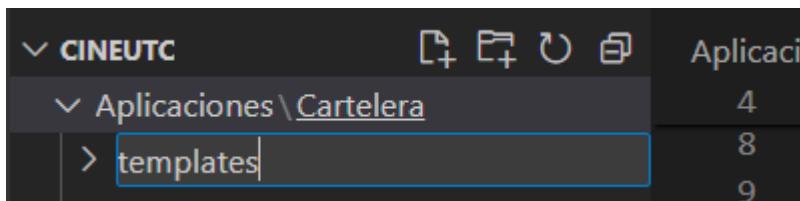
## PASO

### AHORA VAMOS A CREAR EL CRUD MANUALMENTE

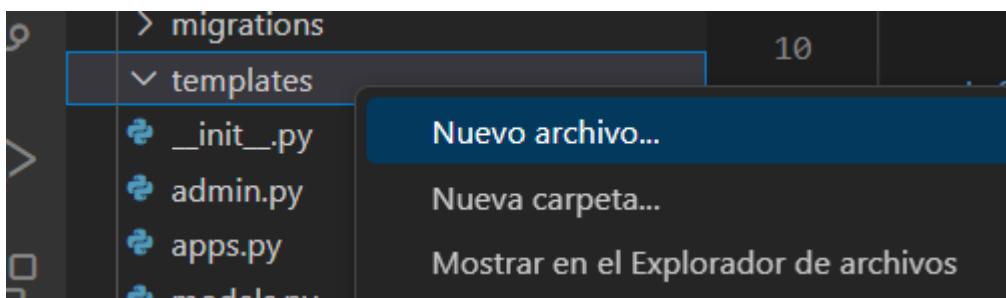
Creamos una nueva carpeta



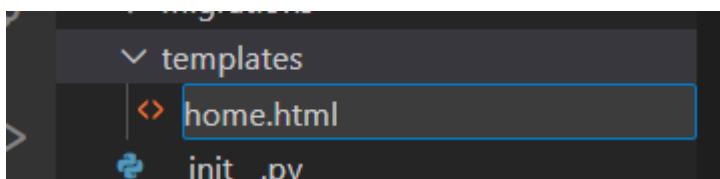
El nombre de la carpeta será templates , todo minuscula



Creamos un archivo dentro de la carpeta template



El nombre será home.html

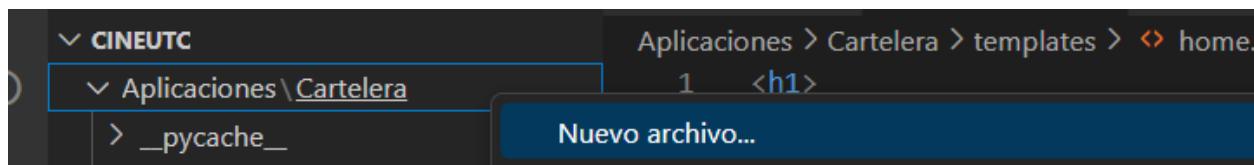


Y colocamos un mensaje **hola mundo desde django** dentro del archivo **home.html**

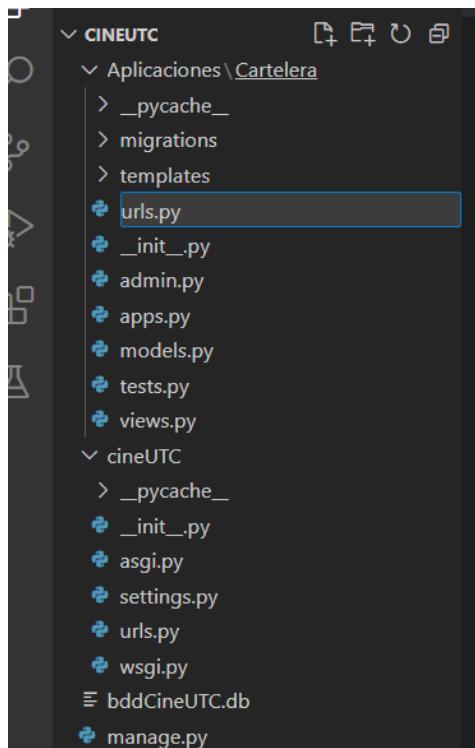
The screenshot shows the Visual Studio Code interface. The left sidebar displays the project structure under 'CINEUTC' with 'Aplicaciones\Cartelera' expanded, showing files like '\_pycache\_\_', 'migrations', 'templates', '\_init\_\_.py', 'admin.py', 'apps.py', 'models.py', and 'tests.py'. The 'templates' folder is selected. The right pane shows the file 'home.html' with the following content:

```
1 <h1>
2 |   Hola mundo desde django :)
3 </h1>
```

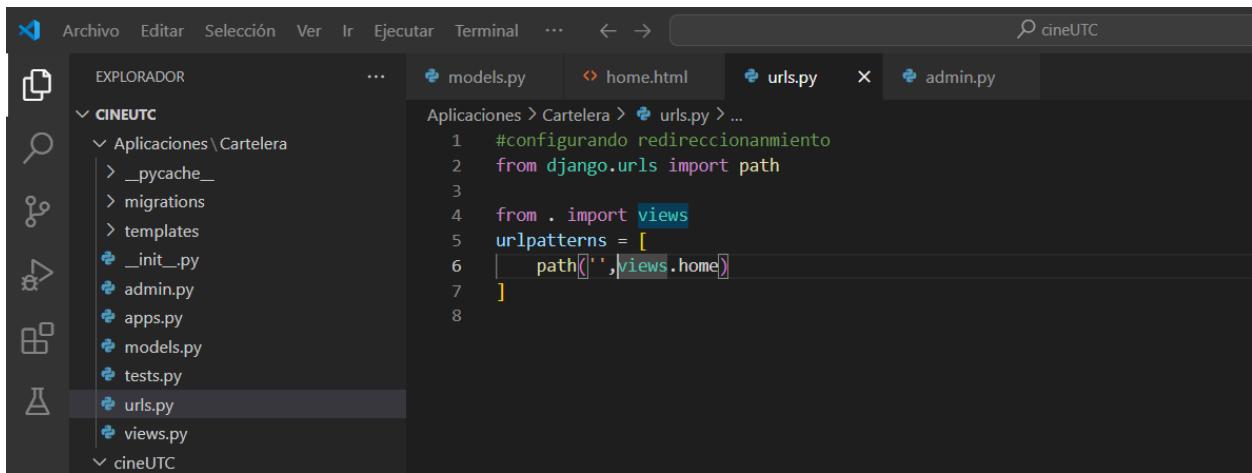
Creamos un archivo dentro de aplicaciones/cartelera



Y le daremos el nombre urls.py



## Configuramos el urls de la aplicación cartelera



The screenshot shows a code editor with the following file structure:

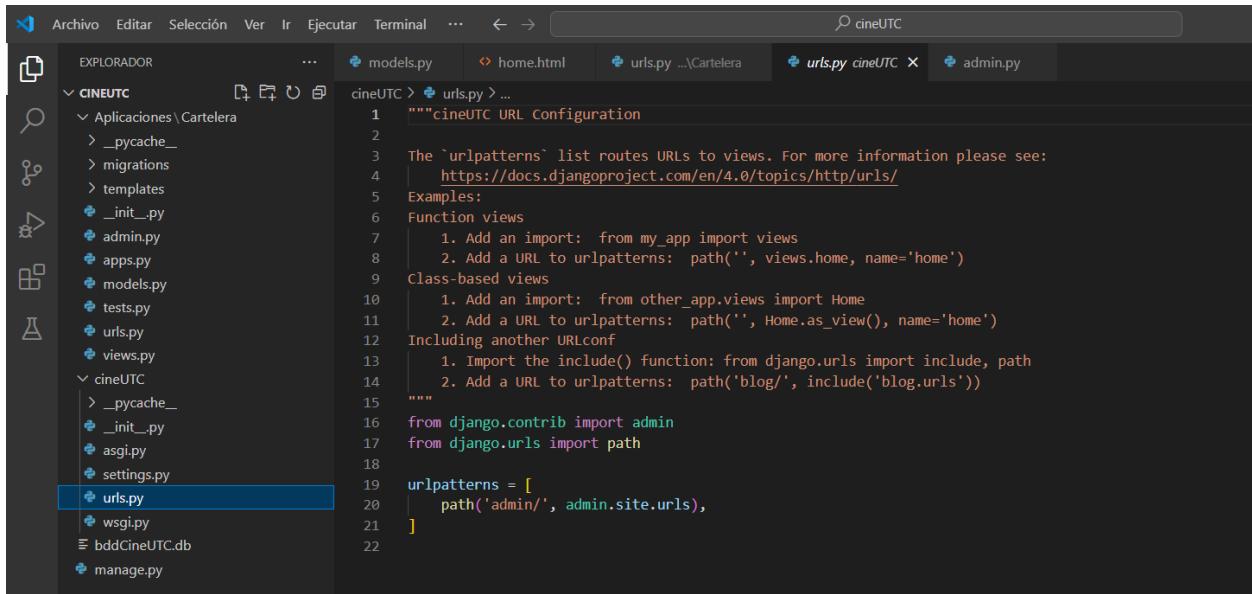
- EXPLORADOR
- CINEUTC
  - Aplicaciones\Cartelera
    - \_pycache\_
    - migrations
    - templates
    - \_\_init\_\_.py
    - admin.py
    - apps.py
    - models.py
    - tests.py
    - urls.py
    - views.py
  - cineUTC
    - \_pycache\_
    - \_\_init\_\_.py
    - asgi.py
    - settings.py
    - urls.py
    - wsgi.py

The current file is `urls.py` from the `Cartelera` application. The code contains:

```
#configurando redireccionamiento
from django.urls import path
from . import views
urlpatterns = [
    path('',views.home)
```

## PASO

Ahora configuraremos el otro url del proyecto en general



The screenshot shows a code editor with the following file structure:

- EXPLORADOR
- CINEUTC
  - Aplicaciones\Cartelera
    - \_pycache\_
    - migrations
    - templates
    - \_\_init\_\_.py
    - admin.py
    - apps.py
    - models.py
    - tests.py
    - urls.py
    - views.py
  - cineUTC
    - \_pycache\_
    - \_\_init\_\_.py
    - asgi.py
    - settings.py
    - urls.py
    - wsgi.py

The current file is `urls.py` at the project root. The code contains:

```
"""cineUTC URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/4.0/topics/http/urls/
Examples:
Function views
    1. Add an import: from my_app import views
    2. Add a URL to urlpatterns: path('', views.home, name='home')
Class-based views
    1. Add an import: from other_app.views import Home
    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path

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

Importamos las urls de la aplicación dentro del proyecto

```
cineUTC > urls.py > ...
1     """cineUTC URL Configuration
2
3     The `urlpatterns` list routes URLs to views. For more information please see:
4         https://docs.djangoproject.com/en/4.0/topics/http/urls/
5     Examples:
6         Function views
7             1. Add an import: from my_app import views
8             2. Add a URL to urlpatterns: path('', views.home, name='home')
9         Class-based views
10            1. Add an import: from other_app.views import Home
11            2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
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('Aplicaciones.Cartelera.urls'))
22 ]
```

Nos vamos a este apartado

```
Aplicaciones > Cartelera > views.py
1     from django.shortcuts import render
2
3     # Create your views here.
```

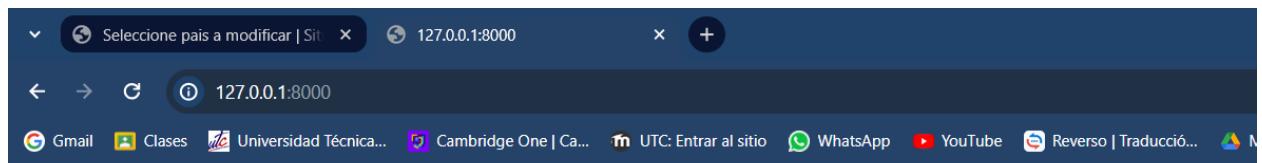
Ahora creamos la función home

The screenshot shows a code editor interface with the following details:

- EXPLORADOR (Left Panel):** Shows the project structure under "CINEUTC".
  - Aplicaciones\Cartelera
    - \_pycache\_
    - migrations
    - templates
      - home.html
    - \_\_init\_\_.py
    - admin.py
    - apps.py
    - models.py
    - tests.py
    - urls.py
- Views.py Content (Right Panel):**

```
1  from django.shortcuts import render
2
3  # Create your views here.
4  def home(request):
5      return render(request, "home.html")
```

Ahora copiamos este link <http://127.0.0.1:8000/> en el navegador y nos debe aparecer lo siguiente



## Hola mundo desde django :)

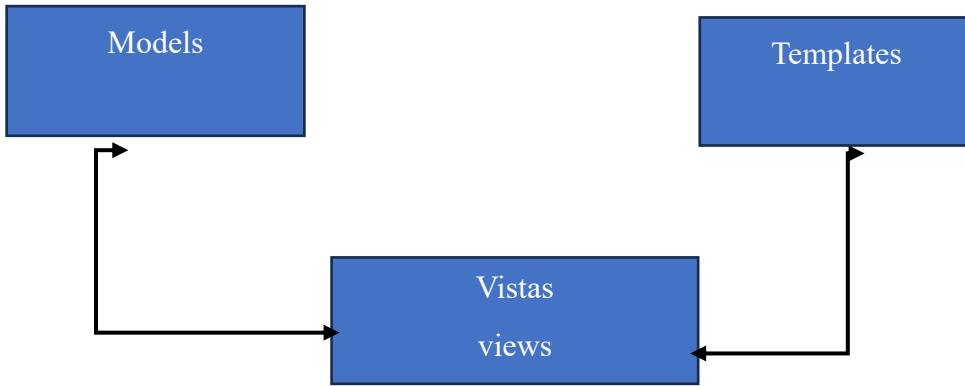
## CLAVES FORÁNEAS

Añadimos el siguiente código para los fk

```
#Creando un nuevo modelo Pelicula
class Pelicula(models.Model):
    id=models.AutoField(primary_key=True)
    titulo=models.CharField(max_length=250)
    duracion=models.TimeField(null= True)
    sinopsis=models.TextField()
    genero=models.ForeignKey(Genero, on_delete=models.CASCADE) #aqui se hace el llamado a las claves foraneas
    director=models.ForeignKey(Director, on_delete=models.CASCADE)

    def __str__(self):
        fila="{0}: {1}"
        return fila.format(self.id,self.titulo)
```

## ARQUITECTURA MTV



## CONTINUAMOS CON LA CREACIÓN DEL CRUD DE MANERA MANUAL

Creamos un nuevo archivo llamado plantilla.html

La captura de pantalla muestra la estructura de directorios de un proyecto Django. En el explorador de archivos, se ve la carpeta 'CINEUTC' que contiene 'Aplicaciones', 'templates' y otros archivos como \_\_init\_\_.py, admin.py, apps.py, models.py, tests.py, urls.py y views.py. Dentro de 'Aplicaciones', hay una carpeta 'Cartelera' que contiene 'migrations', 'templates' y 'urls.py'. El archivo 'plantilla.html' se encuentra dentro de la carpeta 'templates' de 'Cartelera'. La terminal muestra el contenido del archivo 'home.html':

```
<h1>
| Hola mundo desde django :)
</h1>
```

Vamos estructurando el html, para activar la búsqueda y utilizar html utilizamos el comando :

presionamos las teclas **CONTROL+K**, soltamos y luego presionamos solo la tecla **m**

The screenshot shows a code editor interface with a dark theme. The left sidebar displays the project structure under 'EXPLORADOR'. The main pane shows the content of 'plantilla.html'.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cine UTC</title>
</head>
<body>
    <h3>MENU</h3>
    |
    <h2>PIE</h2>
</body>
</html>
```

A continuación colocamos las dos siguientes líneas , las cuales nos permitirá añadir el contenido entre el menú y el pie

The screenshot shows the same code editor interface as before, but with additional code added to 'plantilla.html'.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cine UTC</title>
</head>
<body>
    <h3>MENU</h3>
    <!--Estas dos lineas nos permitiran añadir el contenido entre el men
    {% block body %}
    ...
    {% endblock %}
    <h2>PIE</h2>
</body>
</html>
```

Aplicaremos la herencia con jinja2

The screenshot shows a code editor interface with the following details:

- File Explorer (Left):** Shows the project structure under "CINEUTC". The "templates" folder contains "home.html" and "plantilla.html". Other files include "models.py", "admin.py", "apps.py", and "models.py".
- Code Editor (Right):** Displays the content of "home.html".

```

1  {% extends "./plantilla.html" %} <!--Con esto heredamos los archivos de html-->
2  {% block body %}
3  <h1>
4  |   Hola mundo desde django :) <!-- Poner aqui un banner-->
5  </h1>
6
7  <!-- Poner una mision y vision-->
8  {% endblock %}

```

A continuación, agregamos bootstrap a nuestra pagina, importamos los códigos y arreglamos el código

The screenshot shows a code editor interface with the following details:

- File Explorer (Left):** Shows the project structure under "CINEUTC". The "templates" folder contains "home.html" and "plantilla.html". Other files include "models.py", "admin.py", "apps.py", and "models.py".
- Code Editor (Right):** Displays the content of "home.html", which now includes Bootstrap imports and a navbar structure.

```

1  <!DOCTYPE html>
2  <html lang="es">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Cine UTC</title>
8      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
9          integrity="sha384-T3cCOI6uLrA9TnEo7RknatzjcdSCmGIXXSR1GAsXEV/DwykC2MPK8M2IN" crossorigin="anonymous">
10     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js"
11         integrity="sha384-BBt+EGRqQAUmxJ7phWbEyER41lg+O15P+i6Ep7Q9Q+qX6gbd85u4mG4QZxI"
12         crossorigin="anonymous"></script>
13
14  <body>
15      <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
16          <div class="container-fluid">
17              <a class="navbar-brand" href="#">CINE UTC</a>
18              <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
19                  data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
20                  aria-label="Toggle navigation">
21                  <span class="navbar-toggler-icon"></span>
22              </button>
23              <div class="collapse navbar-collapse" id="navbarSupportedContent">
24                  <ul class="navbar-nav me-auto mb-2 mb-lg-0">
25                      <li class="nav-item">
26                          <a class="nav-link active" aria-current="page" href="#">Inicio</a>
27                      </li>
28                      <li class="nav-item">
29                          <a class="nav-link" href="#">Géneros</a>
30                      </li>
31                  </ul>
32              </div>
33          </div>
34      </nav>
35      <!--Estas dos líneas nos permitirán añadir el contenido entre el menú y el pie de página-->
36
37

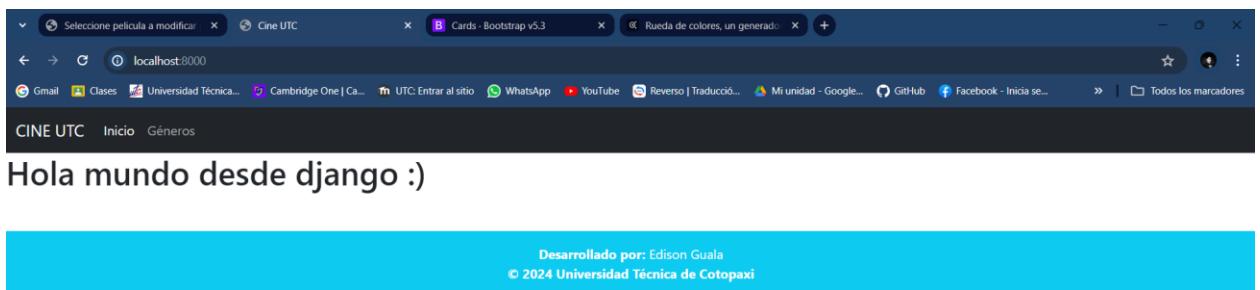
```

The screenshot shows a code editor interface with the following details:

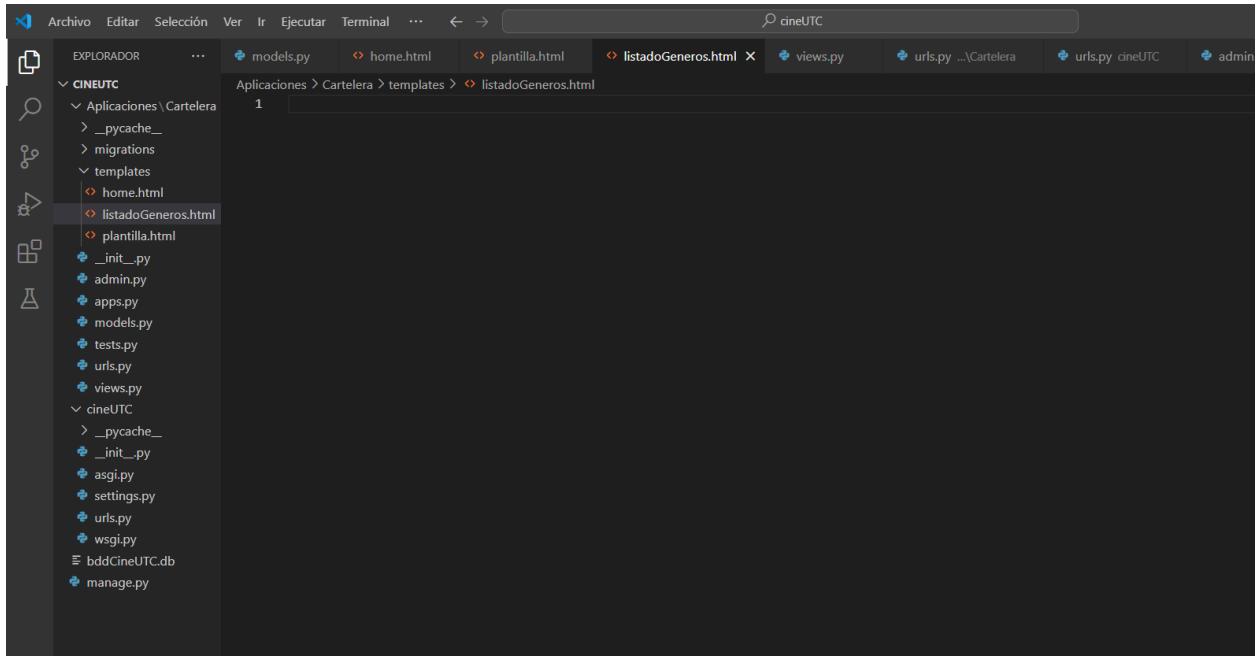
- File Explorer (Left):** Shows the project structure under "CINEUTC".
  - Aplicaciones\Cartelera
  - migrations
  - templates
    - home.html
    - plantilla.html
  - \_\_init\_\_.py
  - admin.py
  - apps.py
  - models.py
  - tests.py
  - urls.py
  - views.py
  - cineUTC
    - \_\_pycache\_\_
    - \_\_init\_\_.py
    - asgi.py
    - settings.py
    - urls.py
    - wsgi.py
  - bddCineUTC.db
  - manage.py
- Code Editor (Right):** Displays the content of "plantilla.html".

```
<html lang="es">
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <div class="container-fluid">
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li><a class="nav-link" href="#">Géneros</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
    <!--Estas dos líneas nos permitirán añadir el contenido entre el menú y el pie de página-->
    {% block body %}
    {% endblock %}
    <footer class="footer mt-5 py-3 bg-info text-white">
      <div class="container">
        <p class="text-center mb-0"><b>Desarrollado por:</b> Edison Guala</p>
        <p class="text-center mb-0">© 2024 Universidad Técnica de Cotopaxi</p>
      </div>
    </footer>
  </body>
</html>
```

Y se verá así



Ahora creamos un nuevo archivo listadoGeneros.html



Vamos a crear una nueva vista, para eso nos ubicamos en el archivo views.py

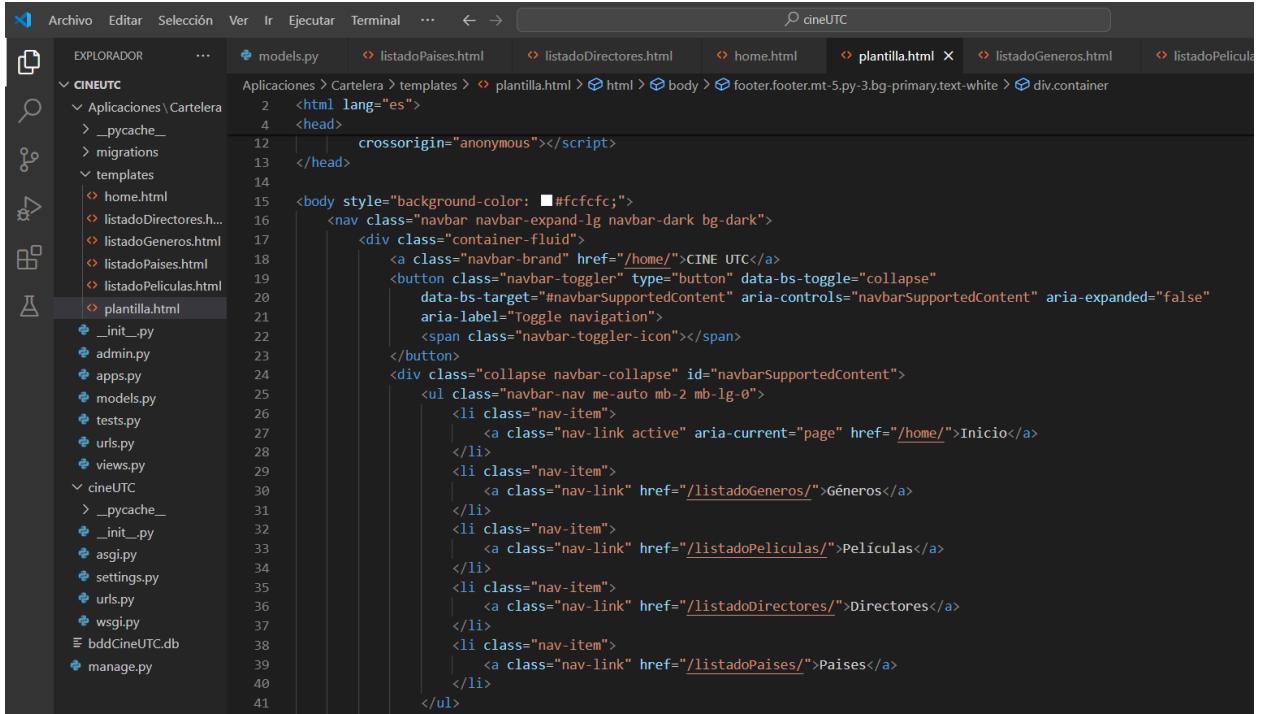
```
from django.shortcuts import render
# Create your views here.
def home(request):
    return render(request,"home.html")
#Renderizando el template de ListadoGeneros
def ListadoGenero(request):
    return render(request,"listadoGeneros.html")
```

Ahora vamos al archivo urls.py , y añadimos un nuevo path

```
#configurando redireccionamiento
from django.urls import path
from . import views
urlpatterns = [
    path('',views.home),
    path('listadoGeneros/',views.listadoGeneros)
]
```

## CLASE 6/6/2024

Vamos a crear los botones, aquí se va a rutear es decir colocar los links de las páginas para las visualizaciones

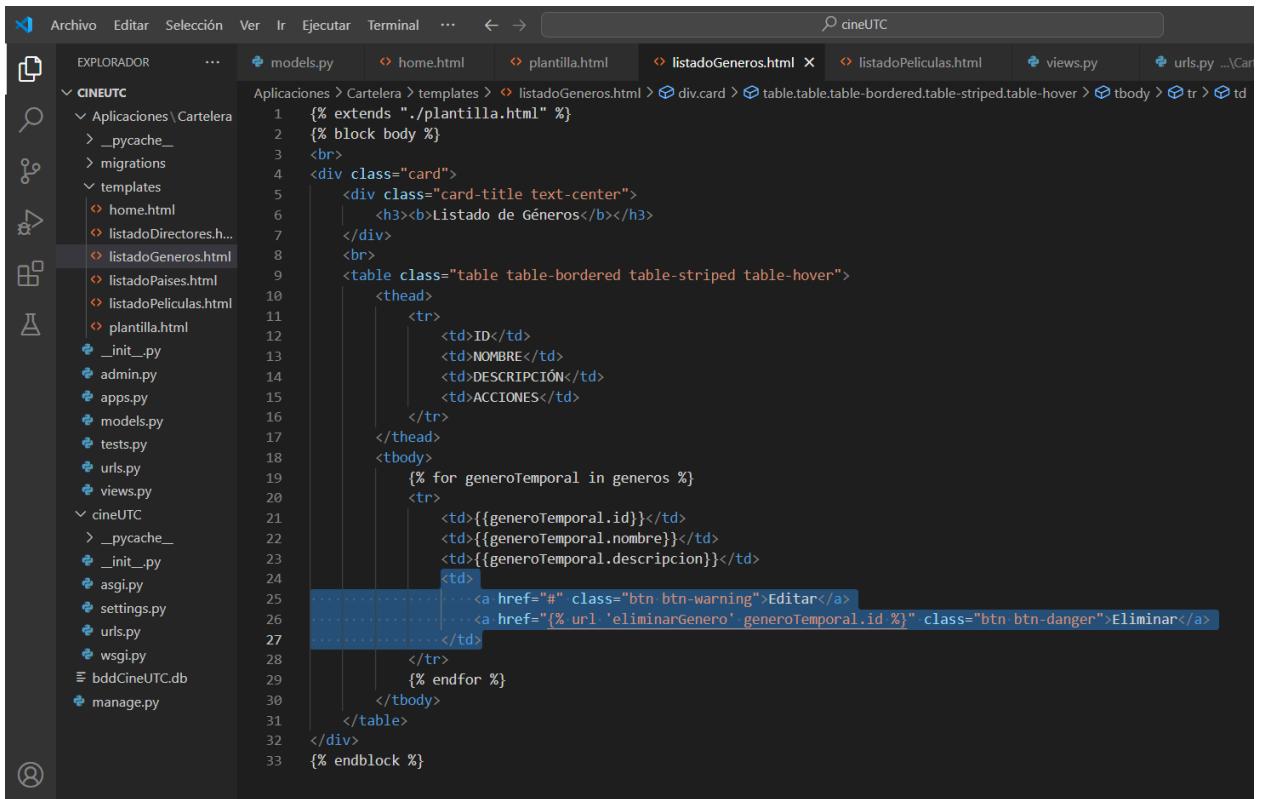


```

<html lang="es">
  <head>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-9�
    crossorigin="anonymous"></script>
  </head>
  <body style="background-color: #fcfcfc;">
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <div class="container-fluid">
        <a class="navbar-brand" href="/home/">CINE UTC</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="/home/">Inicio</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/listadoGeneros/">Géneros</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/listadoPeliculas/">Películas</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/listadoDirectores/">Directores</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/listadoPaises/">Países</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
    <div class="row justify-content-center" style="margin-top: 20px;">
      <div class="col-md-8">
        <h2 style="text-align: center;">Listado de GénerosIDNOMBREDESCRIPCIÓNACTIONESEditar</a>
                <a href="{% url 'eliminarGenero' generoTemporal.id %}" class="btn btn-danger" href="#">Eliminar</a>
              </td>
            </tr>
            <% endfor %>
          </tbody>
        </table>
      </div>
    </div>
  </body>
</html>

```

Nos ubicamos en el archivo `listadoGeneros`. Y añadimos el siguiente código para los botones

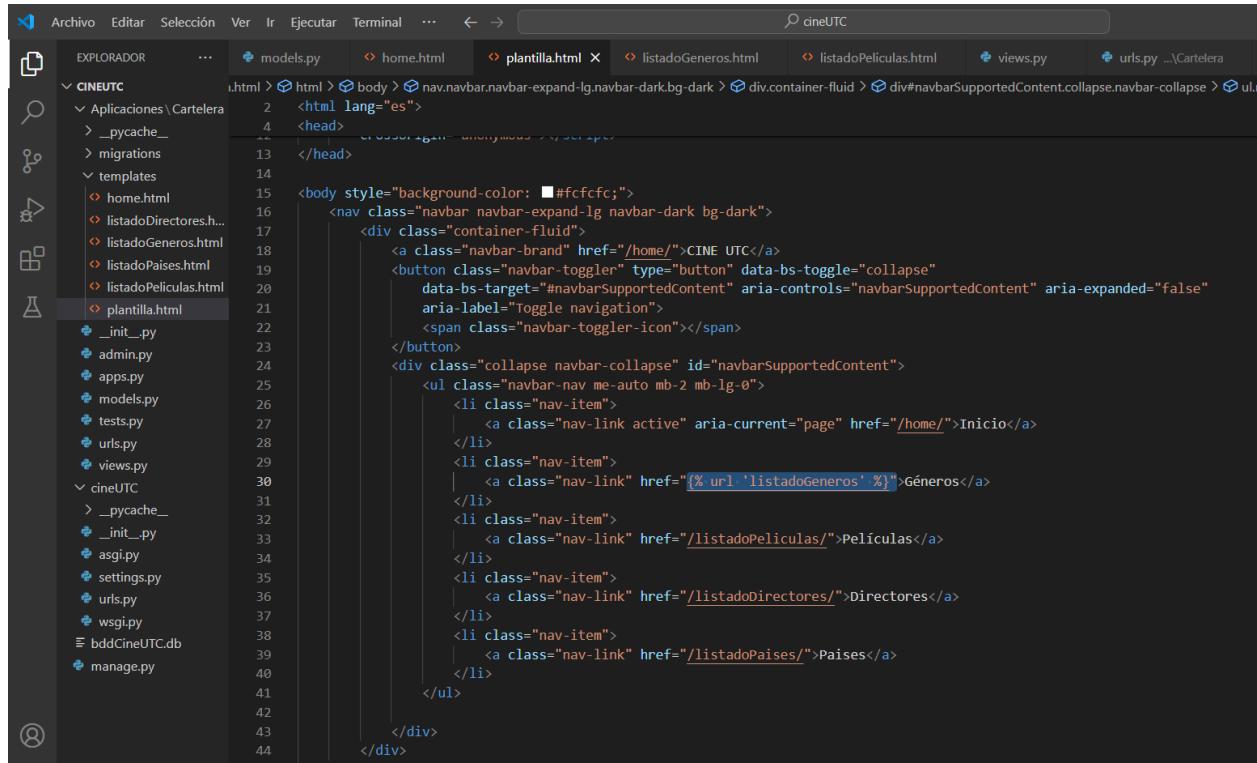


```

<% extends "./plantilla.html" %>
<% block body %>
<br>
<div class="card">
  <div class="card-title text-center">
    <h3><b>Listado de Géneros</b></h3>
  </div>
  <br>
  <table class="table table-bordered table-striped table-hover">
    <thead>
      <tr>
        <th>ID</th>
        <th>NOMBRE</th>
        <th>DESCRIPCIÓN</th>
        <th>ACTIONS</th>
      </tr>
    </thead>
    <tbody>
      <% for generoTemporal in generos %>
      <tr>
        <td>{{generoTemporal.id}}</td>
        <td>{{generoTemporal.nombre}}</td>
        <td>{{generoTemporal.descripcion}}</td>
        <td>
          <a href="#" class="btn btn-warning" href="#">Editar</a>
          <a href="{% url 'eliminarGenero' generoTemporal.id %}" class="btn btn-danger" href="#">Eliminar</a>
        </td>
      </tr>
      <% endfor %>
    </tbody>
  </table>
</div>
<% endblock %>

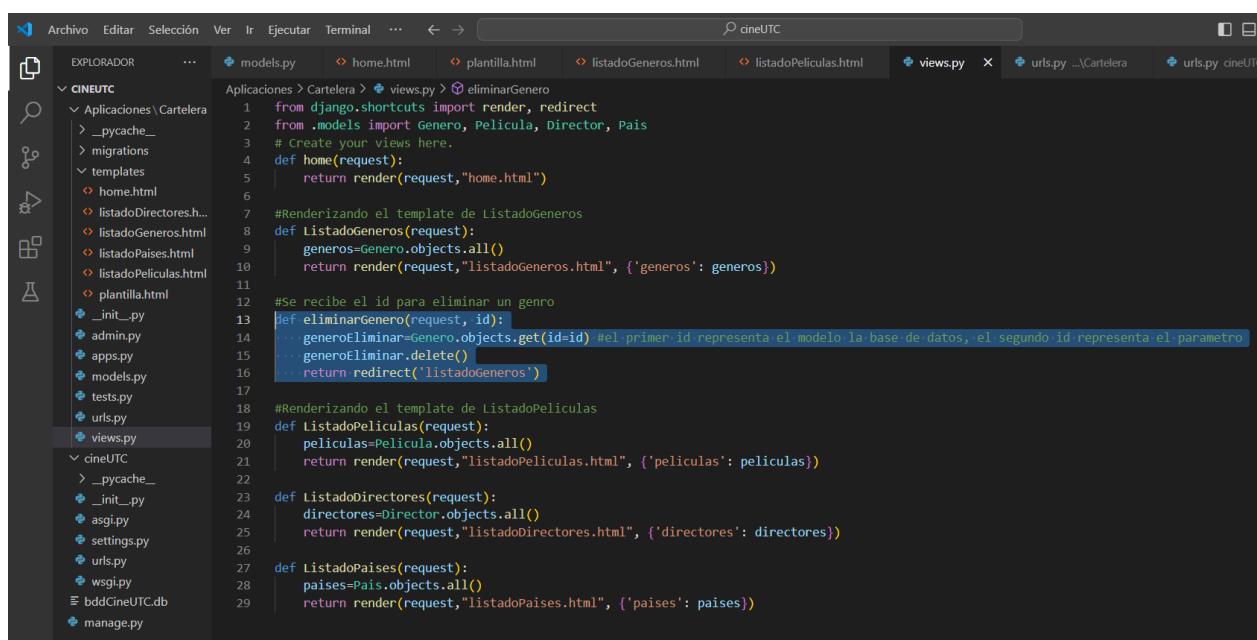
```

## Nos ubicamos en el archivo plantilla y redireccionamos



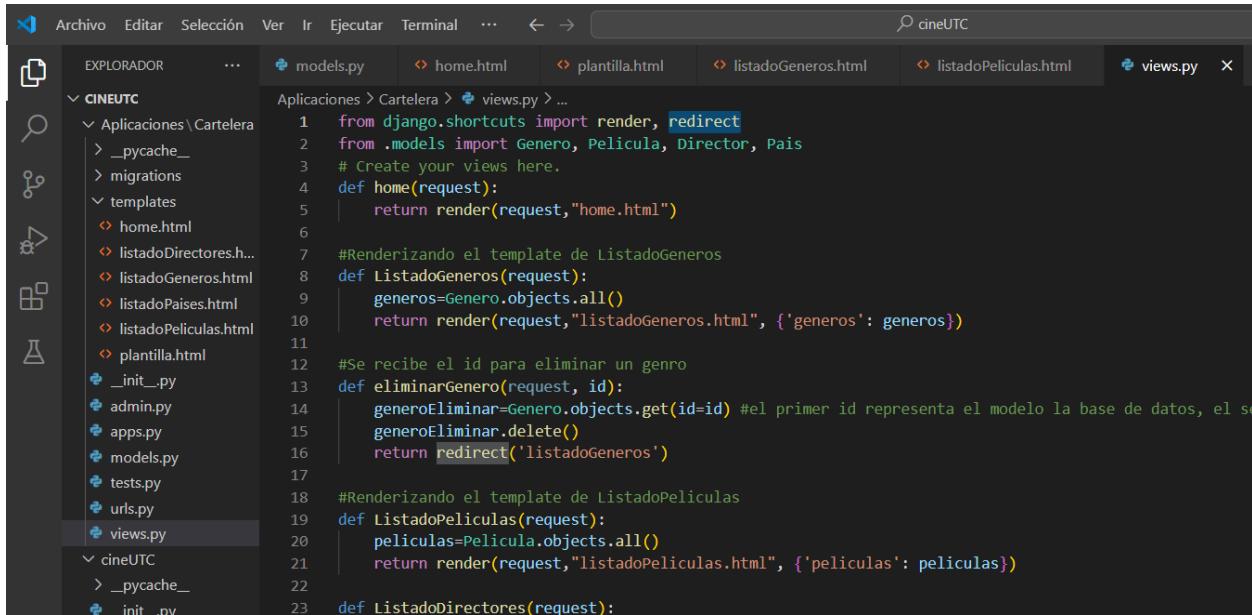
```
<html>
    <body lang="es">
        <head>
            <meta charset="UTF-8" />
            <title>CINE UTC</title>
        </head>
        <body style="background-color: #fcfcfc;">
            <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
                <a class="navbar-brand" href="/home/">CINE UTC</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                        <li class="nav-item">
                            <a class="nav-link active" aria-current="page" href="/home/">Inicio</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="{% url 'listadoGeneros' %}">Géneros</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/listadoPeliculas/">Películas</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/listadoDirectores/">Directores</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/listadoPaises/">Países</a>
                        </li>
                    </ul>
                </div>
            </nav>
        </body>
    </html>
```

Vamos a darle funcionalidad al botón eliminar, para eso nos ubicamos en el archivo views.py, y añadimos la función para eliminar



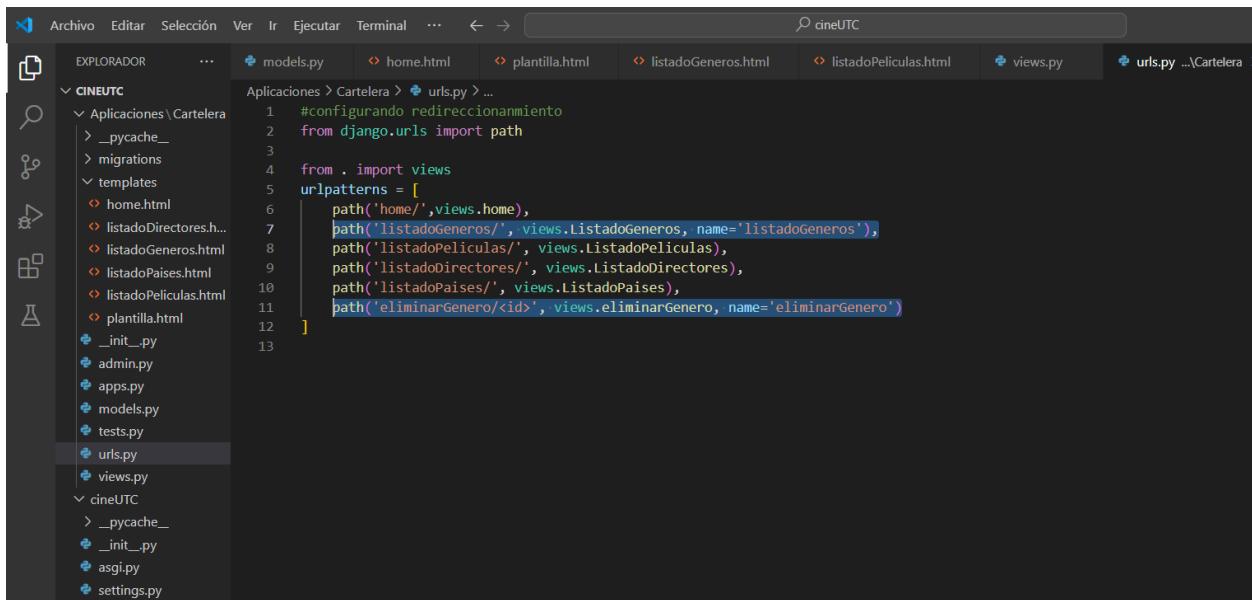
```
from django.shortcuts import render, redirect
from .models import Genero, Pelicula, Director, Pais
# create your views here.
def home(request):
    return render(request, "home.html")
# Renderizando el template de ListadoGeneros
def ListadoGeneros(request):
    generos=Genero.objects.all()
    return render(request,"listadoGeneros.html", {'generos': generos})
#Se recibe el id para eliminar un género
def eliminarGenero(request, id):
    genero=Genero.objects.get(id=id) #el primer id representa el modelo la base de datos, el segundo id representa el parámetro
    genero.delete()
    return redirect('listadoGeneros')
# Renderizando el template de ListadoPeliculas
def ListadoPeliculas(request):
    peliculas=Pelicula.objects.all()
    return render(request,"listadoPeliculas.html", {'peliculas': peliculas})
def ListadoDirectores(request):
    directores=Director.objects.all()
    return render(request,"listadoDirectores.html", {'directores': directores})
def ListadoPaises(request):
    paises=Pais.objects.all()
    return render(request,"listadoPaises.html", {'paises': paises})
```

Hay que tener en cuenta que debemos importar el **redirect**



```
from django.shortcuts import render, redirect
from .models import Genero, Pelicula, Director, Pais
# Create your views here.
def home(request):
    return render(request,"home.html")
#Renderizando el template de ListadoGeneros
def ListadoGeneros(request):
    generos=Genero.objects.all()
    return render(request,"listadoGeneros.html", {'generos': generos})
#Se recibe el id para eliminar un genero
def eliminarGenero(request, id):
    generoEliminar=Genero.objects.get(id=id) #el primer id representa el modelo la base de datos, el se
    generoEliminar.delete()
    return redirect('listadoGeneros')
#Renderizando el template de ListadoPeliculas
def ListadoPeliculas(request):
    peliculas=Pelicula.objects.all()
    return render(request,"listadoPeliculas.html", {'peliculas': peliculas})
def ListadoDirectores(request):
```

Ahora debemos crear un url, nos ubicamos en el archivo urls.py



```
#configurando redireccionamiento
from django.urls import path
from . import views
urlpatterns = [
    path('home/',views.home),
    path('listadoGeneros/', views.ListadoGeneros, name='listadoGeneros'),
    path('listadoPeliculas/', views.ListadoPeliculas),
    path('listadoDirectores/', views.ListadoDirectores),
    path('listadoPaises/', views.ListadoPaises),
    path('eliminarGenero/<id>', views.eliminarGenero, name='eliminarGenero')
```

Añadimos un mensaje de confirmación al momento de eliminar

```

EXPLORADOR      models.py  home.html  plantilla.html  listadoGeneros.html  listadoPeliculas.html  views.py  urls.py ...\\Cartelera  urls.py cineUTC
CINEUTC
  Aplicaciones\\Cartelera
    > __pycache__
    > migrations
    > templates
      > home.html
      > listadoDirectores.h...
      > listadoGeneros.html
      > listadoPaises.html
      > listadoPeliculas.html
      > plantilla.html
    > __init__.py
    > admin.py
    > apps.py
    > models.py
    > tests.py
    > urls.py
    > views.py
  > cineUTC
    > __pycache__
    > __init__.py
    > asgi.py
    > settings.py
    > urls.py
    > wsgi.py
  bddCineUTC.db
  manage.py

Aplicaciones > Cartelera > templates > listadoGeneros.html > div.card > div.table.table-bordered.table-striped.table-hover > tbody > tr > td > a.btn.btn-danger
1  {% extends "./plantilla.html" %}
2  {% block body %}
3  <br>
4  <div class="card">
5    <div class="card-title text-center">
6      <h3><b>Listado de Géneros</b></h3>
7    </div>
8    <br>
9    <table class="table table-bordered table-striped table-hover">
10      <thead>
11        <tr>
12          <td>ID</td>
13          <td>NOMBRE</td>
14          <td>DESCRIPCIÓN</td>
15          <td>ACCIONES</td>
16        </tr>
17      </thead>
18      <tbody>
19        {% for generoTemporal in generos %}
20          <tr>
21            <td>{{generoTemporal.id}}</td>
22            <td>{{generoTemporal.nombre}}</td>
23            <td>{{generoTemporal.descripcion}}</td>
24            <td>
25              <a href="#" class="btn btn-warning">Editar</a>
26              <a href="{% url 'eliminarGenero' generoTemporal.id %}" onclick="return confirm('Estas seguro?');" class="btn btn-danger">Eliminar</a>
27            </td>
28          </tr>
29        {% endfor %}
30      </tbody>
31    </table>
32  </div>
33  {% endblock %}

```

Agregamos un nuevo botón

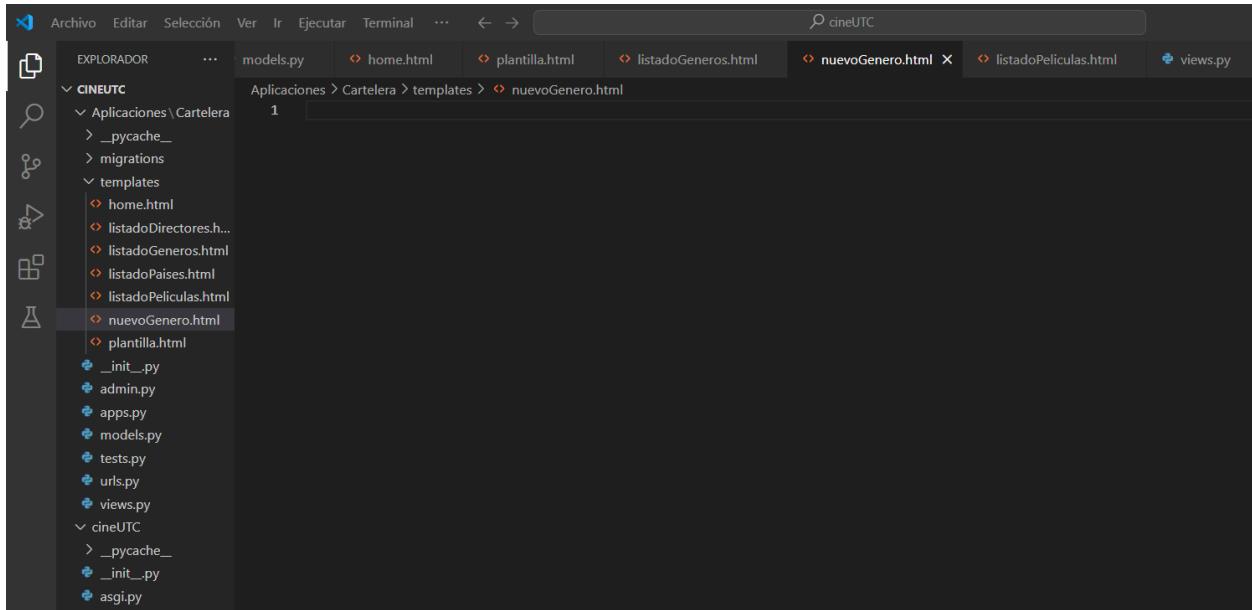
```

EXPLORADOR      models.py  home.html  plantilla.html  listadoGeneros.html  listadoPeliculas.html  views.py  urls.py ...\\Cartelera  urls.py cineUTC
CINEUTC
  Aplicaciones\\Cartelera
    > __pycache__
    > migrations
    > templates
      > home.html
      > listadoDirectores.h...
      > listadoGeneros.html
      > listadoPaises.html
      > listadoPeliculas.html
      > plantilla.html
    > __init__.py
    > admin.py
    > apps.py
    > models.py
    > tests.py
    > urls.py
    > views.py
  > cineUTC
    > __pycache__
    > __init__.py
    > asgi.py
    > settings.py
    > urls.py
    > wsgi.py
  bddCineUTC.db
  manage.py

Aplicaciones > Cartelera > templates > listadoGeneros.html > div.card > div.text-end
1  {% extends "./plantilla.html" %}
2  {% block body %}
3  <br>
4  <div class="card">
5    <div class="card-title text-center">
6      <h3><b>Listado de Géneros</b></h3>
7    </div>
8    <div class="text-end">
9      <a href="#" class="btn btn-success">Agregar Género</a>
10     </div>
11   <br>
12   <table class="table table-bordered table-striped table-hover">
13     <thead>
14       <tr>
15         <td>ID</td>
16         <td>NOMBRE</td>
17         <td>DESCRIPCIÓN</td>
18         <td>ACCIONES</td>
19       </tr>
20     </thead>
21     <tbody>
22       {% for generoTemporal in generos %}
23         <tr>
24           <td>{{generoTemporal.id}}</td>
25           <td>{{generoTemporal.nombre}}</td>
26           <td>{{generoTemporal.descripcion}}</td>
27         </tr>
28       {% endfor %}
29     </tbody>
30   </table>
31 </div>
32 {% endblock %}

```

Creamos un nuevo archivo dentro de templates



Añadimos el siguiente código en el archivo nuevoGenero

```
%> <div class="container-fluid">
    <div class="row">
        <div class="col-md-12">
            <form action="#">
                <label for="nombre"><b>Nombre:</b></label>
                <input type="text" name="nombre" class="form-control" id="nombre" placeholder="Ingrese el nombre del genero">
                <label for="descripcion"><b>Descripción:</b></label>
                <textarea name="descripcion" class="form-control" id="descripcion" rows="3" placeholder="Ingrese la descripción"></textarea>
                <br>
                <button type="submit" class="btn btn-primary">Guardar</button>
                &nbsp;&nbsp;&nbsp;&nbsp;
                <a href="{% url 'listadoGeneros' %}">Cancelar</a>
            </form>
        </div>
    </div>
</div>
[&#128; endblock %]
```

**CLASE MARTES 11/06/2024**

Vamos a añadir la opción para buscar

Ingresamos a la página datatables

The screenshot shows the DataTables homepage. At the top, there are three tabs: 'DataTables', 'Editor', and 'Advanced Components'. Below the tabs, there's a search bar and a small advertisement for EthicalAds. The main content area features a large heading: 'Add advanced interaction controls to your HTML tables *the free & easy way*'. Below the heading, there are two numbered steps: '1 - Include these two files' and '2 - Initialise your DataTable:'. Step 1 shows code snippets for CSS and JS. Step 2 shows a preview of a DataTable with columns for Name, Position, Office, Age, and Start date. The table data is as follows:

Name	Position	Office	Age	Start date
Sonya Frost	Software Engineer	Edinburgh	23	12/12/2008
Timothy Mooney	Office Manager	London	37	10/12/2008
Airi Satou	Accountant	Tokyo	33	27/11/2008
Shad Decker	Regional Director	Edinburgh	51	12/11/2008
Gavin Cortez	Team Leader	San Francisco	22	25/10/2008

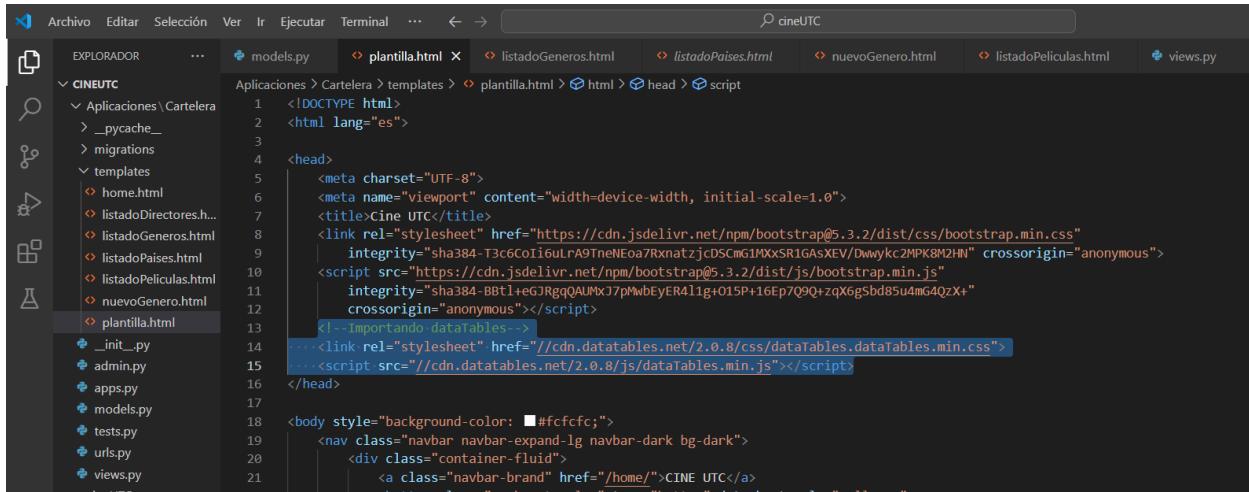
Copiamos el css

This screenshot shows the first step of the DataTables setup process. It displays the instruction '1 - Include these two files' followed by a downward arrow. Below this, there is a 'CSS' button with the URL: `//cdn.datatables.net/2.0.8/css/dataTables.dat`.

Importamos el js

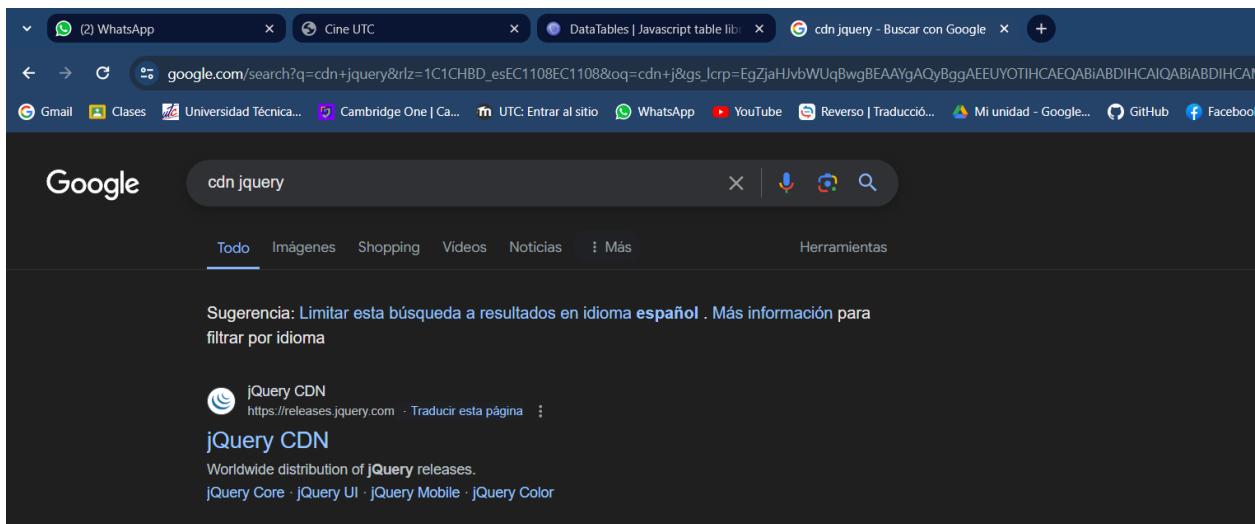
This screenshot shows the second step of the DataTables setup process. It displays the instruction '2 - Initialise your DataTable:' followed by a downward arrow. Below this, there is a 'JS' button with the URL: `//cdn.datatables.net/2.0.8/js/dataTables.min.`

## Lo pegamos en el archivo plantilla.html



```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cine UTC</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" integrity="sha384-T3c6CoI6UlRA9TneFoa7RxatzjcdSCmgIMXSR1GAsXEV/DwykC2MP8M2HN" crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js" integrity="sha384-BBt+eGJRgQAUbxJ7pMwbEyER4lg+015P+16Ep7Q9Q+zqX6gSbd85u4mG4QzX+" crossorigin="anonymous"></script>
    <!-- Importando dataTables -->
    <link rel="stylesheet" href="https://cdn.datatables.net/2.0.8/css/dataTables.dataTables.min.css">
    <script src="https://cdn.datatables.net/2.0.8/js/dataTables.min.js"></script>
</head>
<body style="background-color: #f0f0f0;">
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
            <a class="navbar-brand" href="/">CINE UTC</a>
        </div>
    </nav>
    <div class="container">
        <h1>Listado de Películas</h1>
        <table border="1" class="table">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Nombre</th>
                    <th>Año de Estreno</th>
                    <th>Genero</th>
                    <th>País de Origen</th>
                    <th>Acciones</th>
                </tr>
            <tbody>
                <tr>
                    <td>1</td>
                    <td>El Padrino</td>
                    <td>1972</td>
                    <td>Acción</td>
                    <td>EE.UU.</td>
                    <td><a href="#">Ver</a> <a href="#">Actualizar</a> <a href="#">Eliminar</a></td>
                </tr>
                <tr>
                    <td>2</td>
                    <td>Star Wars: Episodio IV - Una Nueva Esperanza</td>
                    <td>1977</td>
                    <td>Ciencia Ficción</td>
                    <td>EE.UU.</td>
                    <td><a href="#">Ver</a> <a href="#">Actualizar</a> <a href="#">Eliminar</a></td>
                </tr>
                <tr>
                    <td>3</td>
                    <td>El Señor de los Anillos: La Comunidad de la Oscuro</td>
                    <td>1978</td>
                    <td>Fantasía</td>
                    <td>Nueva Zelanda</td>
                    <td><a href="#">Ver</a> <a href="#">Actualizar</a> <a href="#">Eliminar</a></td>
                </tr>
                <tr>
                    <td>4</td>
                    <td>El Padrino: Parte II</td>
                    <td>1974</td>
                    <td>Acción</td>
                    <td>EE.UU.</td>
                    <td><a href="#">Ver</a> <a href="#">Actualizar</a> <a href="#">Eliminar</a></td>
                </tr>
                <tr>
                    <td>5</td>
                    <td>El Padrino: Parte III</td>
                    <td>1990</td>
                    <td>Acción</td>
                    <td>EE.UU.</td>
                    <td><a href="#">Ver</a> <a href="#">Actualizar</a> <a href="#">Eliminar</a></td>
                </tr>
            </tbody>
        </table>
    </div>
</body>
```

Ahora buscamos en el navegador lo siguiente



Seleccionamos minified

The screenshot shows a browser window with multiple tabs open. The active tab is 'releases.jquery.com'. The page content is titled 'jQuery CDN – Latest Stable Versions' and is powered by 'fastly'. It features a section for 'jQuery Core' showing the latest stable release for 'jQuery 3.x'. A note mentions using 'uncompressed', 'minified', 'slim', and 'slim minified' versions.

The screenshot shows a code editor with a 'Code Integration' panel. The panel displays a script tag for jQuery 3.7.1:

```
<script
  src="https://code.jquery.com/jquery-3.7.1.min.js"
  integrity="sha256-/JqT3SQfawRcv/BiHPTbkBvs0OEvtFFmqPF/lYI/Cxo="
  crossorigin="anonymous"></script>
```

A tooltip on the right says 'Copy to clipboard!'. Below the panel, a note explains the use of 'integrity' and 'crossorigin' attributes for Subresource Integrity (SRI) checking.

Copiamos el código y añadimos en el archivo plantilla.html

The screenshot shows a code editor displaying the 'plantilla.html' file from a Django project. The file content includes a script tag for jQuery 3.7.1 with integrity and crossorigin attributes:

```
<script
  src="https://code.jquery.com/jquery-3.7.1.min.js" integrity="sha256-/JqT3SQfawRcv/BiHPTbkBvs0OEvtFFmqPF/lYI/Cxo="
  crossorigin="anonymous"></script>
```

Nos ubicamos en el archivo listadoGeneros.html y añadimos lo siguiente

```

EXPLORADOR      models.py      plantilla.html      listadoGeneros.html      listadoPaises.html      nuevoGenero.html      listadoPeliculas.html      views.py      urls.py
CINEUTC
  Aplicaciones\Cartelera
    > __pycache__
    > migrations
    templates
      > home.html
      > listadoDirectores.h...
      > listadoGeneros.html
      > listadoPaises.html
      > listadoPeliculas.html
      > nuevoGenero.html
      > plantilla.html
    > __init__.py
    > admin.py
    > apps.py
    > models.py
    > tests.py
    > urls.py
    > views.py
  cineUTC
    > __pycache__
    > __init__.py
    > asgi.py
    > settings.py
    > urls.py
    > wsgi.py
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
  32
  33
  34
  35
  36
  37
  38

```

Ahí mismo en ese archivo añadimos un script, y ahí debemos el código del punto 2 de la imagen

The screenshot shows a browser window with three tabs: WhatsApp, Cine UTC, and DataTables | Javascript table library. The main content area displays the DataTables.js documentation. The title is "Add advanced interaction to your HTML tables the first time". Below the title, there are two steps: 1. Include these two files (with links to CSS and JS CDN URLs) and 2. Initialise your DataTable (with a snippet of JavaScript code).

**1 – Include these two files ↓**

**CSS** //cdn.datatables.net/2.0.8/css/dataTables.dataTables.css

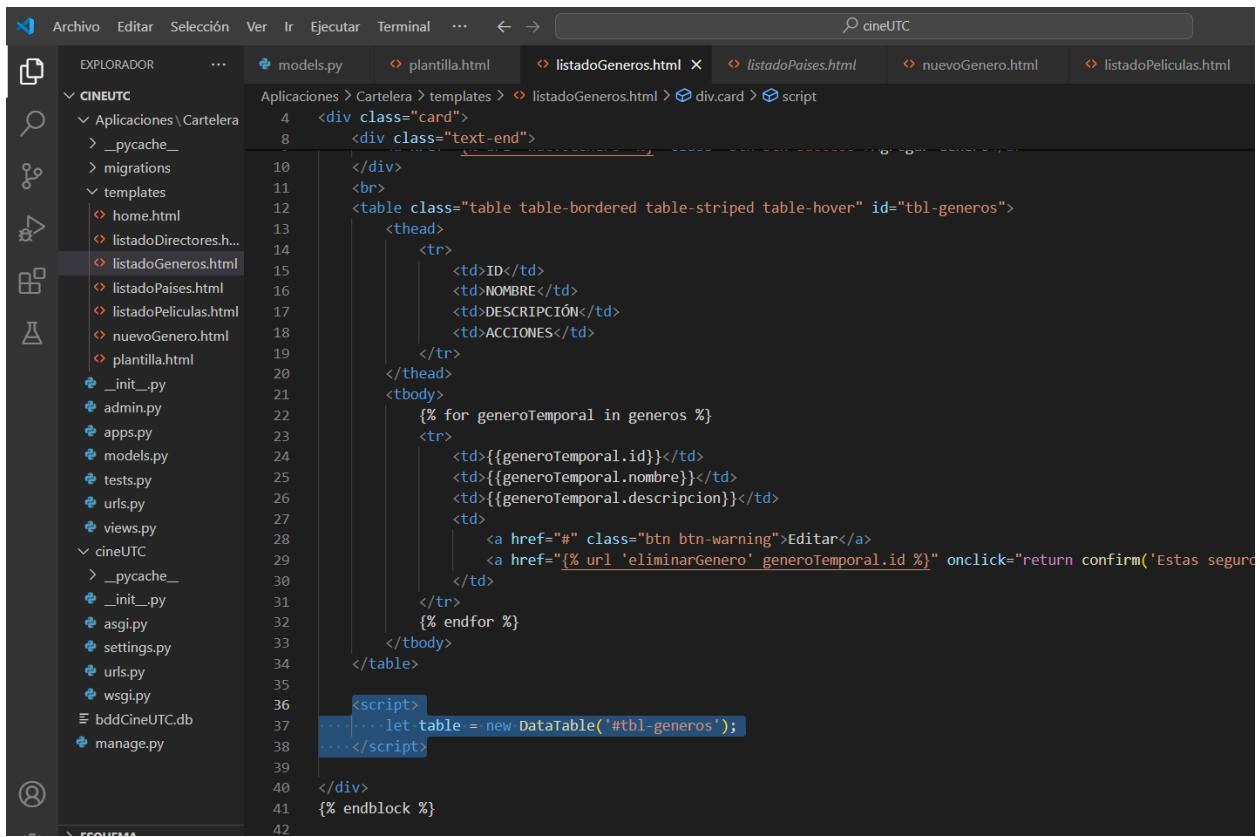
**JS** //cdn.datatables.net/2.0.8/js/dataTables.min.js

**2 – Initialise your DataTable: ↓**

```
1 let table = new DataTable('#myTable');
```

On the right side of the page, there is a dark-themed preview of a table with 10 entries per page, showing names like Sonya Frost, Timothy Moon, Airi Satou, and Shad Decker.

Lo pegamos dentro del endblock



```
EXPLORADOR ... models.py plantilla.html listadoGeneros.html listadoPaises.html nuevoGenero.html listadoPelículas.html
CINEUTC
Aplicaciones > Cartelera > templates > listadoGeneros.html > div.card > script
4   <div class="card">
8     <div class="text-end">
10    </div>
11    <br>
12    <table class="table table-bordered table-striped table-hover" id="tbl-generos">
13      <thead>
14        <tr>
15          <td>ID</td>
16          <td>NOMBRE</td>
17          <td>DESCRIPCIÓN</td>
18          <td>ACCIONES</td>
19        </tr>
20      </thead>
21      <tbody>
22        {% for generoTemporal in generos %}
23        <tr>
24          <td>{{generoTemporal.id}}</td>
25          <td>{{generoTemporal.nombre}}</td>
26          <td>{{generoTemporal.descripcion}}</td>
27          <td>
28            <a href="#" class="btn btn-warning">Editar</a>
29            <a href="{% url 'eliminarGenero' generoTemporal.id %}" onclick="return confirm('Estas seguro')> Eliminar</a>
30          </td>
31        {% endfor %}
32      </tbody>
33    </table>
34
35    <script>
36      let table = new DataTable('#tbl-generos');
37
38    </script>
39
40  </div>
41  {% endblock %}
```

Y nos quedaría así:



CINE UTC Inicio Géneros Películas Directores Paises

### Listado de Géneros

10 entries per page

ID	NOMBRE	DESCRIPCIÓN	ACCIONES
1	Terror	Películas para mayores de 15 años	<button>Editar</button> <button>Eliminar</button>
6	Novela	Películas para mayores de 16 años	<button>Editar</button> <button>Eliminar</button>
2	Comedia	Películas para mayores de 14 años	<button>Editar</button> <button>Eliminar</button>
3	Ciencia Ficción	Películas para mayores de 12 años	<button>Editar</button> <button>Eliminar</button>

Showing 1 to 4 of 4 entries

1

Podemos añadir estilos a ez atbla

```

EXPLORADOR      ...      models.py      plantilla.html      listadoGeneros.html X      listadoPaises.html      nuevoGenero.html
Aplicaciones > Cartelera > templates > listadoGeneros.html > div.card > style
    4   <div class="card">
    12     <table class="table table-bordered table-striped table-hover" id="tbl-generos">
    13       <thead>
    20         </thead>
    21       <tbody>
    22         {% for generoTemporal in generos %}
    23           <tr>
    24             <td>{{generoTemporal.id}}</td>
    25             <td>{{generoTemporal.nombre}}</td>
    26             <td>{{generoTemporal.descripcion}}</td>
    27             <td>
    28               <a href="#" class="btn btn-warning">Editar</a>
    29               <a href="{% url 'eliminarGenero' generoTemporal.id %}" onclick="return confirm('¿Desea eliminar este genero?')">Eliminar</a>
    30             </td>
    31           </tr>
    32         {% endfor %}
    33       </tbody>
    34     </table>
    35
    36     <script>
    37       let table = new DataTable('#tbl-generos');
    38     </script>
    39
    40     <style>
    41       .dt-input{
    42         border:3px solid red !important;
    43         border-radius: 20px !important ;
    44       }
    45       .dt-search label{
    46         color: green !important;
    47       }
    48     </style>
    49
    50   </div>
    51   {% endblock %}

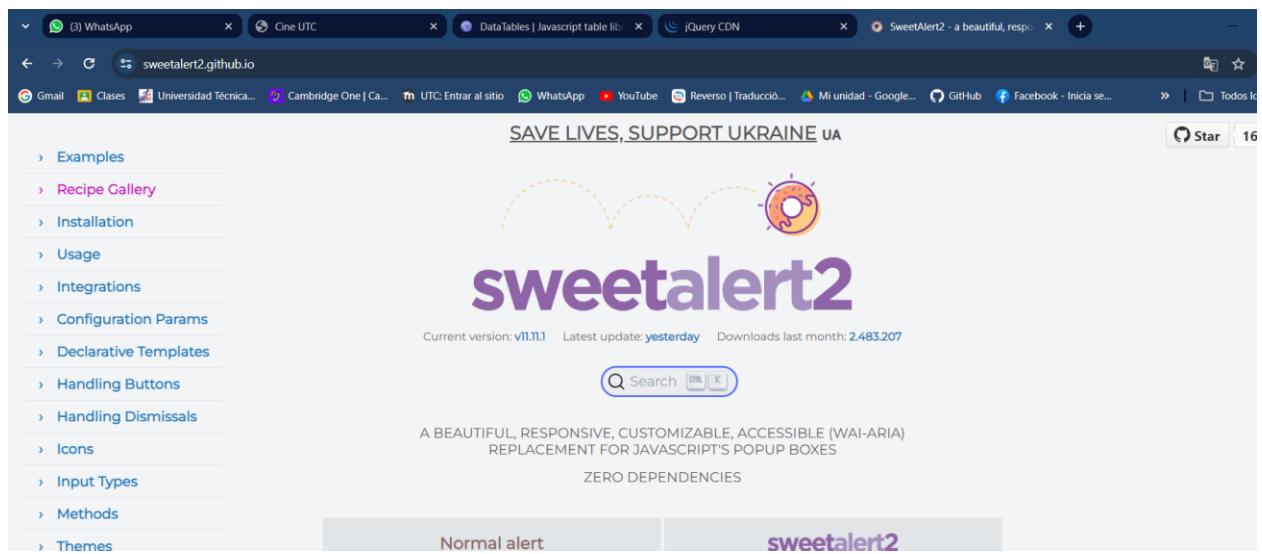
```

Así se vería:

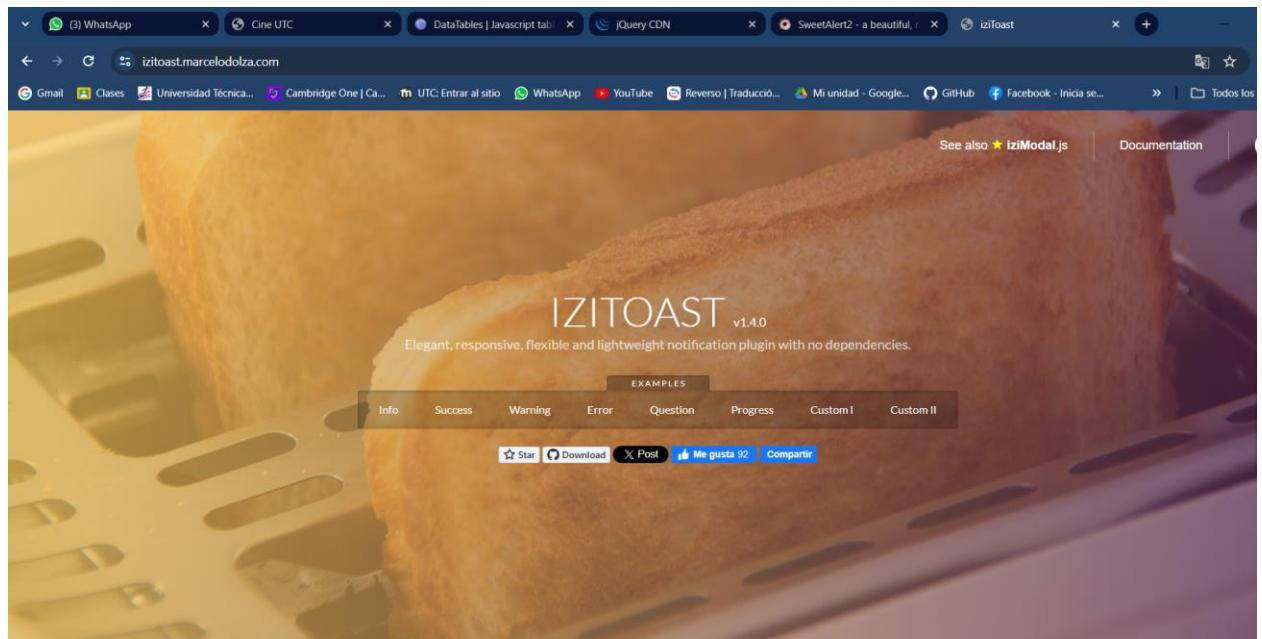
ID	NOMBRE	DESCRIPCIÓN	ACCIONES
1	Terror	Películas para mayores de 15 años	<a href="#">Editar</a> <a href="#">Eliminar</a>

Ahora ingresamos a estos sitios para ver las alertas, lo que podemos importar en nuestro proyecto

OPCION 1:



## OPCION 2:



Copiamos el siguiente código

The screenshot shows a browser window with several tabs open at the top, including WhatsApp, Cine UTC, DataTables | Javascript table, jQuery CDN, SweetAlert2 - a beautiful, ..., and iziToast. The main content area is titled "DOWNLOAD & INSTALL". On the left, there's a sidebar with a navigation menu:

- > Examples
- > Recipe Gallery
- > Installation
- > Usage
- > Integrations
- > Configuration Params
- > Declarative Templates
- > Handling Buttons
- > Handling Dismissals
- > Icons
- > Input Types
- > Methods

In the center, there's a terminal-like box containing the command `$ npm install sweetalert2`. Below it, text says "Or grab from jsDelivr CDN" with a link. A code snippet box contains the CDN URL: `<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>`. At the bottom, a note states: "sweetalert2 is the 23rd most popular package on jsDelivr, with 1.407.329.846 CDN hits in the last month".

Nos vamos a plantilla.html e importamos ahí

The screenshot shows a code editor with the file "plantilla.html" selected in the sidebar. The code in the editor is as follows:

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cine UTC</title>
    <!-- Importando jQuery -->
    <script src="https://code.jquery.com/jquery-3.7.1.min.js" integrity="sha256-/JqT3SOfaRcv/BIHPTHkBvs00EvtFFm0" crossorigin="anonymous"></script>
    <!-- Importando bootstrap -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatjzjDSCmG1MXXSR1GAsXEV/DwyKc2MPK8M2HN" crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js" integrity="sha384-BBtl+eGJRgQQUUmJ7pMwbEyER4l1g+015P+16Ep7Q9Q+zqX6gSbd85u4mG4QzX+" crossorigin="anonymous"></script>
    <!-- Importando dataTables -->
    <link rel="stylesheet" href="https://cdn.datatables.net/2.0.8/css/dataTables.dataTables.min.css">
    <script src="https://cdn.datatables.net/2.0.8/js/dataTables.min.js"></script>
    <!-- Importando SweetAlert -->
    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
</head>

<body style="background-color: #fcfcfc;">
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
            <a class="navbar-brand" href="/">CINE UTC</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

```

Ahora copiamos el siguiente código de ejemplo

The screenshot shows the official SweetAlert2 documentation page at [sweetalert2.github.io/#examples](https://sweetalert2.github.io/#examples). On the left, there's a sidebar with links to Examples, Recipe Gallery, Installation, Usage, Integrations, Configuration Params, Declarative Templates, Handling Buttons, Handling Dismissals, Icons, Input Types, and Methods. The main content area is titled "EXAMPLES". It displays two examples. The first example, "A basic message", shows a simple button labeled "Try me!" which triggers the JavaScript code `Swal.fire("SweetAlert2 is working!");`. The second example, "A title with a text under", shows a similar setup with a "Try me!" button and the corresponding JavaScript code `Swal.fire({ title: "The Internet?", text: "That thing is still around?", icon: "question" });`.

Y lo pegamos dentro de un script en el archivo de plantilla.html, hacemos unas modificaciones del texto

The screenshot shows a code editor interface with a dark theme. On the left is a file explorer showing a project structure for "CINEUTC" with files like `models.py`, `plantilla.html`, `listadoGeneros.html`, etc. The main pane shows the content of `plantilla.html`. A specific section of the code is highlighted in blue, representing the addition of SweetAlert2 confirmation logic. The code snippet is:

```

    ...
    <script>
        Swal.fire({
            title: "CONFIRMACIÓN",
            text: "Género guardado exitosamente",
            icon: "success"
        });
    </script>

```

Se verá así:

The screenshot shows a web application interface. At the top, there are several tabs: 'tsApp', 'Cine UTC', 'DataTables | Javascript tab...', 'jQuery CDN', 'SweetAlert2 - a beautiful, ...', and 'iziToast'. Below the tabs, the URL 'localhost:8000/listadoGeneros/' is visible. The main content area has a navigation bar with links: 'Inicio', 'Géneros', 'Películas', 'Directores', and 'Países'. The title 'Listado de Géneros' is centered above a table. A modal dialog box is overlaid on the page, containing a green checkmark icon, the word 'CONFIRMACIÓN', the message 'Género guardado exitosamente', and a blue 'OK' button. To the left of the modal, a table lists genres with columns 'NOMBRE' and 'DESCRIPCION'. To the right, there is a section titled 'ACCIONES' with 'Editar' and 'Eliminar' buttons for each row. A red box highlights the search input field on the right.

Nos ubicamos en el archivo `views.py`, e importamos lo siguiente

The screenshot shows a code editor with the file `views.py` open. The code is as follows:

```

EXPLORADOR          listadoGeneros.html  nuevoGenero.html  listadoPeliculas.html  views.py  urls.py ...\\Cartelera  urls.py cine...
CINEUTC
  Aplicaciones \\Cartelera
    > __pycache__
    > migrations
    > templates
      > home.html
      > listadoDirectores.html
      > listadoGeneros.html
      > listadoPaises.html
      > listadoPeliculas.html
      > nuevoGenero.html

Aplicaciones > Cartelera > views.py > ...
1  from django.shortcuts import render, redirect
2  from .models import Genero, Pelicula, Director, Pais
3  from django.contrib import messages
4  # Create your views here.
5  def home(request):
6      return render(request, "home.html")
7
8  # Renderizando el template de ListadoGeneros
9  def listadoGeneros(request):
10     generos=Genero.objects.all()
11     return render(request,"listadoGeneros.html", {'generos': generos})

```

Ahora nos ubicamos en la línea 27 y añadimos lo siguiente

```

    from django.shortcuts import render, redirect
    from .models import Genero, Pelicula, Director, Pais
    from django.contrib import messages

    # Create your views here.
    def home(request):
        return render(request, "home.html")

    # Renderizando el template de ListadoGeneros
    def ListadoGeneros(request):
        generos=Genero.objects.all()
        return render(request,"listadoGeneros.html", {"generos": generos})

    # Se recibe el id para eliminar un genero
    def eliminarGenero(request,id):
        generoEliminar=Genero.objects.get(id=id) #el primer id representa el modelo la base de datos, el segundo id representa el genero
        generoEliminar.delete()
        return redirect('listadoGeneros')

    # Renderizando formulario de nuevo genero
    def nuevoGenero(request):
        return render(request, 'nuevoGenero.html')

    # Insertando generos en la base de datos
    def guardarGenero(request):
        nom=request.POST["nombre"]
        des=request.POST["descripcion"]
        nuevoGenero=Genero.objects.create(nombre=nom, descripcion=des)
        messages.success(request,"Genero registrado exitosamente.")
        return redirect('listadoGeneros')

```

Ahora nos vamos al archivo plantilla.html y añadimos el siguiente código

```

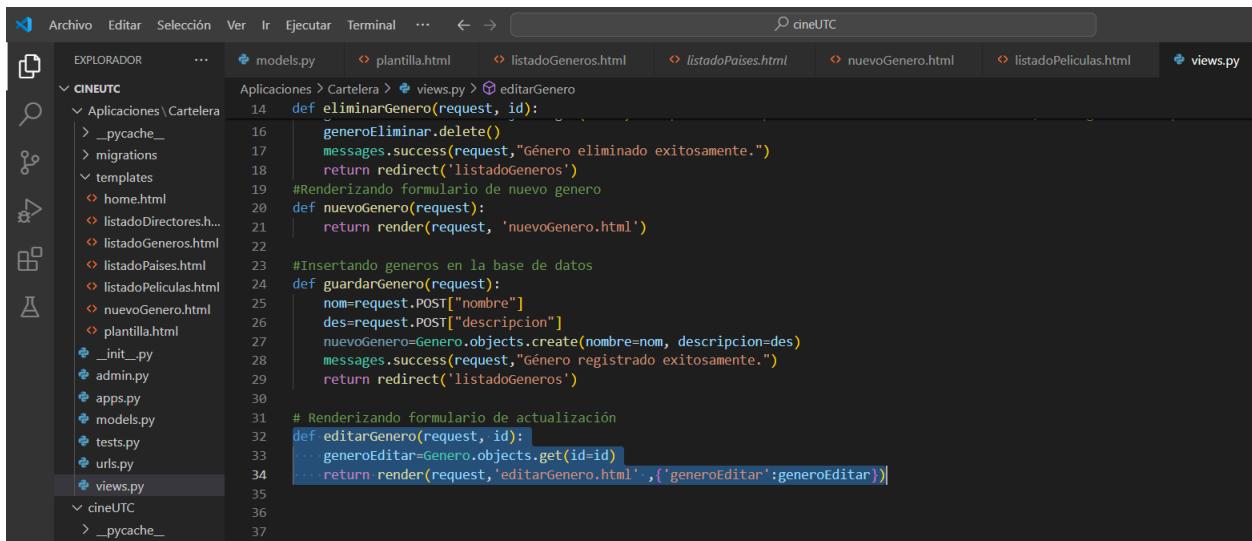
<html lang="es">
    <body style="background-color: #fcfcfc;">
        {% block body %}
        {% endblock %}
        <footer class="footer mt-5 py-3 bg-primary text-white">
            <div class="container" style="padding-top: 20px; padding-bottom: 20px;">
                <div class="row">
                    <div class="col-md-6">
                        <p class="text-center mb-0"><b>Desarrollado por:</b> Edison Guala</p>
                    </div>
                    <div class="col-md-6">
                        <p class="text-center mb-0"><b>© 2024 Universidad Técnica de Cotopaxi</b></p>
                    </div>
                </div>
            </div>
        </footer>

        [% if messages %]
            [% for mensaje in messages %]
                [% if mensaje.tags == 'success' %]
                    <script>
                        Swal.fire({
                            title: "CONFIRMACIÓN",
                            text: "{mensaje}",
                            icon: "success"
                        });
                    </script>
                [% endif %]
            [% endfor %]
        [% endif %]
    </body>
</html>

```

## CLASE JUEVES 13/06/2024

Vamos a crear la función para actualizar géneros, para eso nos ubicamos en el archivo de views.py



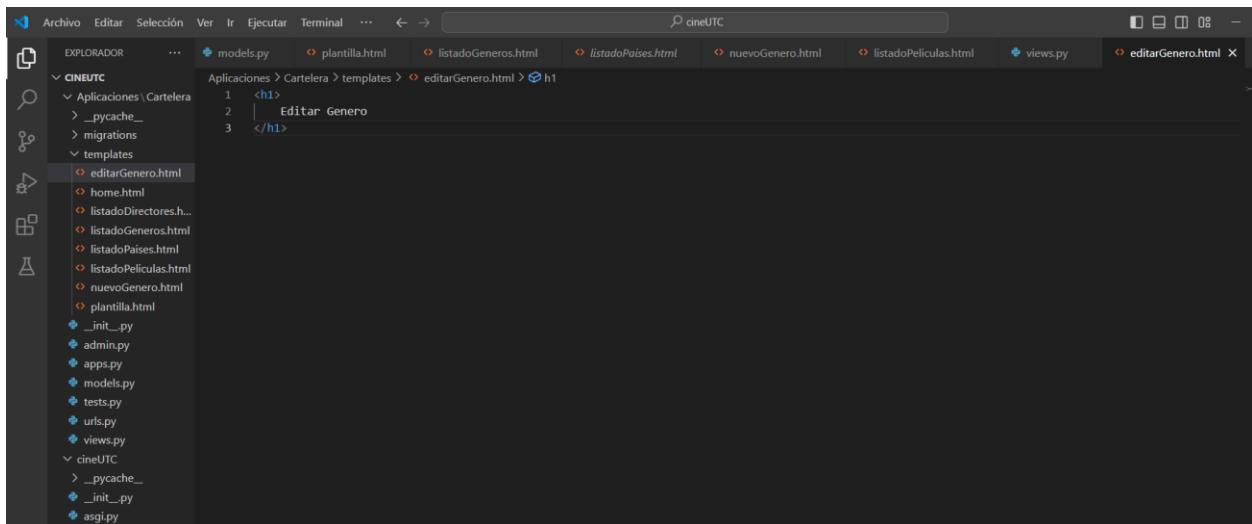
```
def eliminarGenero(request, id):
    generoEliminar.delete()
    messages.success(request, "Género eliminado exitosamente.")
    return redirect('listadoGeneros')

def nuevoGenero(request):
    return render(request, 'nuevoGenero.html')

# Insertando géneros en la base de datos
def guardarGenero(request):
    nom=request.POST['nombre']
    des=request.POST['descripcion']
    nuevoGenero=Genero.objects.create(nombre=nom, descripcion=des)
    messages.success(request, "Género registrado exitosamente.")
    return redirect('listadoGeneros')

# Renderizando formulario de actualización
def editarGenero(request, id):
    generoEditar=Genero.objects.get(id=id)
    return render(request, 'editarGenero.html', {'generoEditar':generoEditar})
```

Ahora vamos a crear un nuevo archivo llamado editarGenero.html dentro de la carpeta Templates



```
<h1>
|   Editar Genero
</h1>
```

Nos vamos a este apartado de urls.py de Cartelera y colocamos un nuevo path editar

```

    1 #configurando redireccionamiento
    2 from django.urls import path
    3
    4 from . import views
    5 urlpatterns = [
    6     path('home/',views.home),
    7     path('listadoGeneros/', views.ListadoGeneros, name='listadoGeneros'),
    8     path('listadoPelículas/', views.ListadoPelículas),
    9     path('listadoDirectores/', views.ListadoDirectores),
    10    path('listadoPaises/', views.ListadoPaises),
    11    path('eliminarGenero<id>', views.eliminarGenero, name='eliminarGenero'),
    12    path('nuevoGenero/', views.nuevoGenero, name='nuevoGenero'),
    13    path('guardarGenero/', views.guardarGenero, name='guardarGenero'),
    14    path('editarGenero<id>', views.editarGenero, name='editarGenero')]
    15 ]
    16

```

Y debemos ubicarnos en este apartado de listadoGeneros.html, para añadir el enlace en el botón editar

```

    4 <div class="card">
    5     <ul class="list-group list-group-flush">
    6         <li><a href="#">Nuevo Genero</a></li>
    7         <a href="#">Listado de Generos</a>
    8         <a href="#">Listado de Peliculas</a>
    9         <a href="#">Listado de Directores</a>
    10        <a href="#">Listado de Paises</a>
    11        <a href="#">Home</a>
    12    </ul>
    13
    14    <table class="table table-bordered table-striped table-hover" id="tbl-generos">
    15        <thead>
    16            <tr>
    17                <th>ID</th>
    18                <th>NOMBRE</th>
    19                <th>DESCRIPCIÓN</th>
    20                <th>ACCIONES</th>
    21            </tr>
    22        </thead>
    23        <tbody>
    24            {% for generoTemporal in generos %}
    25            <tr>
    26                <td>{{generoTemporal.id}}</td>
    27                <td>{{generoTemporal.nombre}}</td>
    28                <td>{{generoTemporal.descripcion}}</td>
    29                <td>
    30                    <a href="{% url 'editarGenero' generoTemporal.id %}" class="btn btn-warning">Editar</a>
    31                    <a href="{% url 'eliminarGenero' generoTemporal.id %}" onclick="return confirm('Estas seguro?');" class="btn btn-danger">Eliminar</a>
    32                </td>
    33            </tr>
    34        {% endfor %}
    35    </tbody>
    36 </table>
    37
    38 <script>
    39     let table = new DataTable('#tbl-generos');
    40 </script>
    41
    42 <style>
        .dt-input{
            border:3px solid red !important;

```

Ahora nos vamos al archivo editarGenero.html y colocamos todo esta línea de código

```
EXPLORADOR Archivo Editar Selección Ver Ir Ejecutar Terminal ... ← → ⌂ cineUTC
CINEUTC
  Aplicaciones\Cartelera
    > _pycache_
    migrations
    templates
      editarGenero.html
      home.html
      listadoDirectores.h...
      listadoGeneros.html
      listadoPaises.html
      listadoPelículas.html
      nuevoGenero.html
      plantilla.html
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      urls.py
      views.py
  cineUTC
    > _pycache_
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
bddCineUTC.db

Aplicaciones > Cartelera > templates > editarGenero.html > div.container-fluid > div.row > div.col-md-12 > form > button.

1  {% extends "./plantilla.html" %} 
2  {% block body %} 
3  <h1>EDITAR GENERO</h1> 
4  <div class="container-fluid"> 
5    <div class="row"> 
6      <div class="col-md-12"> 
7        <form action="#" method="post"> 
8          {% csrf_token %} 
9          <input type="hidden" name="id" value="{{generoEditar.id}}> 
10         <label for=""><b>Nombre:</b></label> 
11         <input type="text" name="nombre" class="form-control" 
12           value="{{generoEditar.nombre}}> 
13           id="nombre" placeholder="Ingrese el nombre del genero" 
14           <label for=""><b>Descripción:</b></label> 
15           <textarea name="descripcion" id="descripcion" 
16             class="form-control" rows="3" 
17               placeholder="Ingrese la descripcion" 
18               >{{generoEditar.descripcion}}</textarea> 
19             <br> 
20             <button type="submit" class="btn btn-warning"> 
21               Actualizar</button> 
22             &nbsp;&nbsp;&nbsp;&nbsp; 
23             <a href="{% url 'listadoGeneros' %}">Cancelar</a> 
24           </form> 
25         </div> 
26       </div> 
27     </div> 
28     <br> 
29   {% endblock %}
```

Ahora nos ubicamos en el archivo de `views.py` y creamos una nueva función para dar funcionalidad a la actualización

```

EXPLORADOR          ...      listadoPaises.html    nuevoGenero.html    listadoPeliculas.html    views.py    edit
CINEUTC
  Aplicaciones\Cartelera
    > __pycache__
    > migrations
    > templates
      editarGenero.html
      home.html
      listadoDirectores.h...
      listadoGeneros.html
      listadoPaises.html
      listadoPeliculas.html
      nuevoGenero.html
      plantilla.html
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  cineUTC
    > __pycache__
    __init__.py
    asgi.py
    settings.py

Aplicaciones > Cartelera > views.py > procesarActualizacionGenero
23 #Insertando generos en la base de datos
24 def guardarGenero(request):
25     nom=request.POST["nombre"]
26     des=request.POST["descripcion"]
27     nuevoGenero=Genero.objects.create(nombre=nom, descripcion=des)
28     messages.success(request,"Género registrado exitosamente.")
29     return redirect('listadoGeneros')
30
31 # Renderizando formulario de actualización
32 def editarGenero(request, id):
33     generoEditar=Genero.objects.get(id=id)
34     return render(request,'editarGenero.html' ,{'generoEditar':generoEditar})
35
36 #Actualizando los nuevos datos en la BDD
37 def procesarActualizacionGenero(request):
38     id=request.POST['id']
39     nombre=request.POST['nombre']
40     descripcion=request.POST['descripcion']
41     generoConsultado=Genero.objects.get(id=id)
42     generoConsultado.nombre=nombre
43     generoConsultado.descripcion=descripcion
44     generoConsultado.save()
45     messages.success(request,"Género actualizado exitosamente.")
46     return redirect('listadoGeneros')
47
48 #Renderizando el template de ListadoPeliculas
49 def ListadoPeliculas(request):
50     peliculas=Pelicula.objects.all()

```

Añadimos el path en urls.py de Cartelera

```

EXPLORADOR          ...      listadoPaises.html    nuevoGenero.html    listadoPeliculas.html    views.py    editarGenero.html    urls.py    ...
CINEUTC
  Aplicaciones\Cartelera
    > __pycache__
    > migrations
    > templates
      editarGenero.html
      home.html
      listadoDirectores.h...
      listadoGeneros.html
      listadoPaises.html
      listadoPeliculas.html
      nuevoGenero.html
      plantilla.html
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py

Aplicaciones > Cartelera > urls.py > ...
1 #configurando redireccionamiento
2 from django.urls import path
3
4 from . import views
5 urlpatterns = [
6     path('home/',views.home),
7     path('listadoGeneros/', views.ListadoGeneros, name='listadoGeneros'),
8     path('listadoPeliculas/', views.ListadoPeliculas),
9     path('listadoDirectores/', views.ListadoDirectores),
10    path('listadoPaises/', views.ListadoPaises),
11    path('eliminarGenero/<id>', views.eliminarGenero, name='eliminarGenero'),
12    path('nuevoGenero/', views.nuevoGenero, name='nuevoGenero'),
13    path('guardarGenero/', views.guardarGenero, name='guardarGenero'),
14    path('editarGenero/<id>', views.editarGenero, name='editarGenero'),
15    path('procesarActualizacionGenero/', views.procesarActualizacionGenero, name='procesarActualizacionGenero')
16 ]

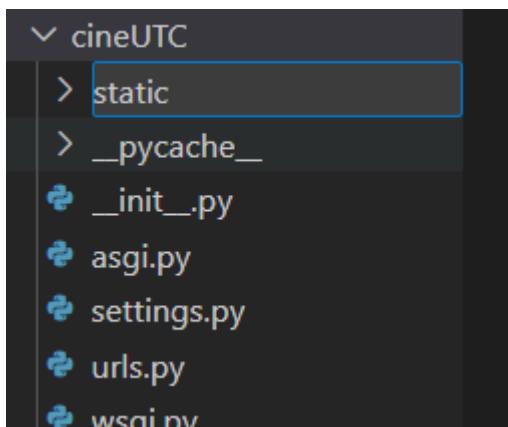
```

Nos ubicamos en el archivo editarGenero.html y colocamos lo siguiente

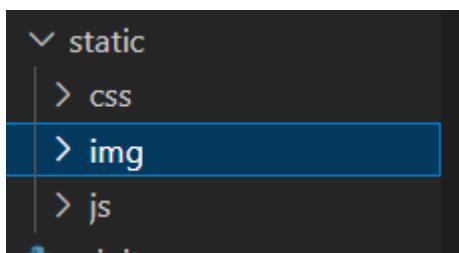
```
% extends "./plantilla.html"
% block body %
<h1>EDITAR GENERO</h1>
<div class="container-fluid">
    <div class="row">
        <div class="col-md-12">
            <form action="{% url 'procesarActualizacionGenero' %}" method="post">
                {% csrf_token %}
                <input type="hidden" name="id" value="{{generoEditar.id}}">
                <label for=""><b>Nombre:</b></label>
                <input type="text" name="nombre" class="form-control" value="{{generoEditar.nombre}}" id="nombre" placeholder="Ingrese el nombre del genero">
                <label for=""><b>Descripción:</b></label>
                <textarea name="descripcion" id="descripcion" value="">{{generoEditar.descripcion}}</textarea>
            </form>
        </div>
    </div>
</div>
```

## CLASE Martes 25/06/2024

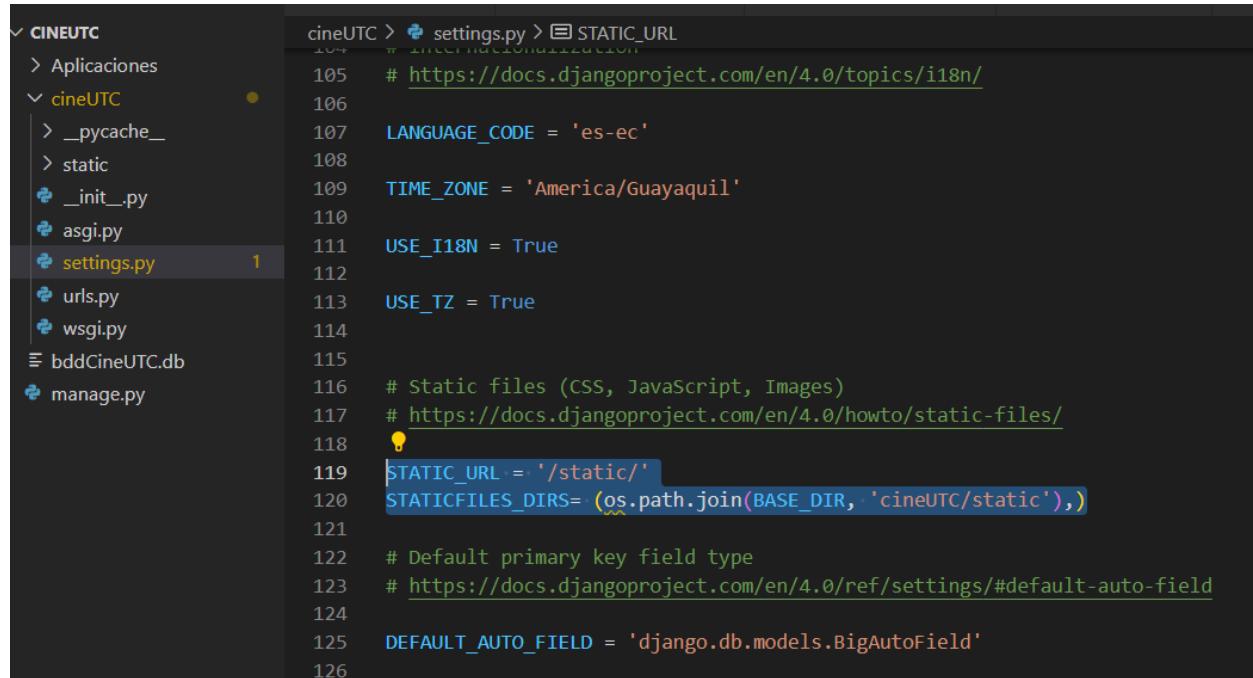
Vamos a crear una nueva carpeta **static**



Dentro de esa carpeta vamos a crear tres carpetas (**css – js – img**)

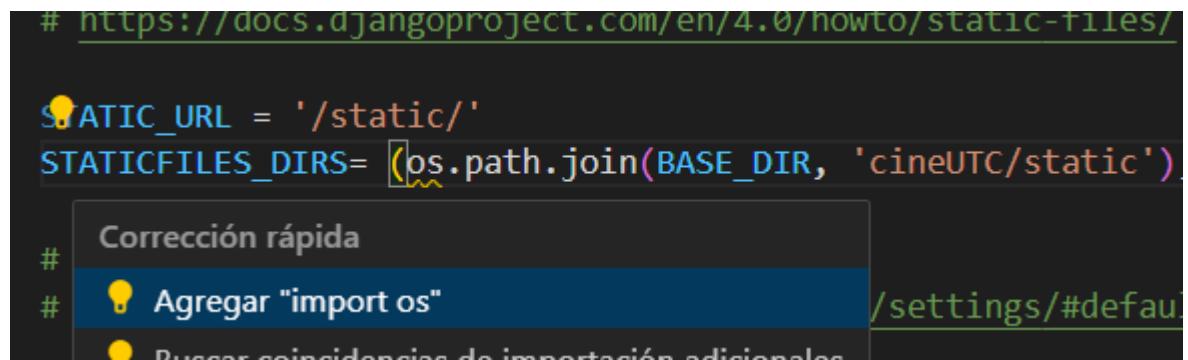


Nos ubicamos en **settings.py** en la **línea 119**



```
cineUTC > settings.py > STATIC_URL
105 # https://docs.djangoproject.com/en/4.0/topics/i18n/
106
107 LANGUAGE_CODE = 'es-ec'
108
109 TIME_ZONE = 'America/Guayaquil'
110
111 USE_I18N = True
112
113 USE_TZ = True
114
115
116 # Static files (CSS, JavaScript, Images)
117 # https://docs.djangoproject.com/en/4.0/howto/static-files/
118
119 STATIC_URL = '/static/'
120 STATICFILES_DIRS = (os.path.join(BASE_DIR, 'cineUTC/static'),)
121
122 # Default primary key field type
123 # https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field
124
125 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
126
```

Importamos os para que funcione

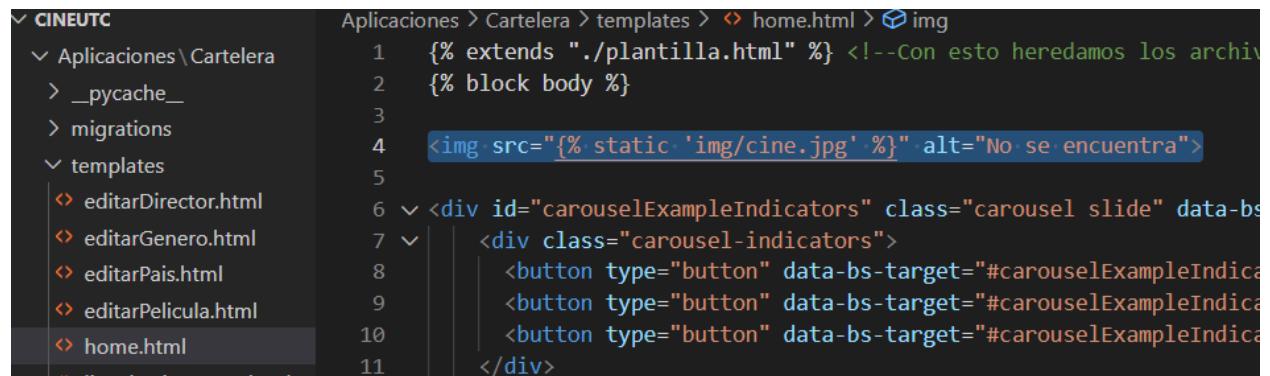


```
# https://docs.djangoproject.com/en/4.0/howto/static-files/
# STATIC_URL = '/static/'
# STATICFILES_DIRS = (os.path.join(BASE_DIR, 'cineUTC/static'))
```

Corrección rápida

- 💡 Agregar "import os"
- 💡 Buscar coincidencias de importación adicionales

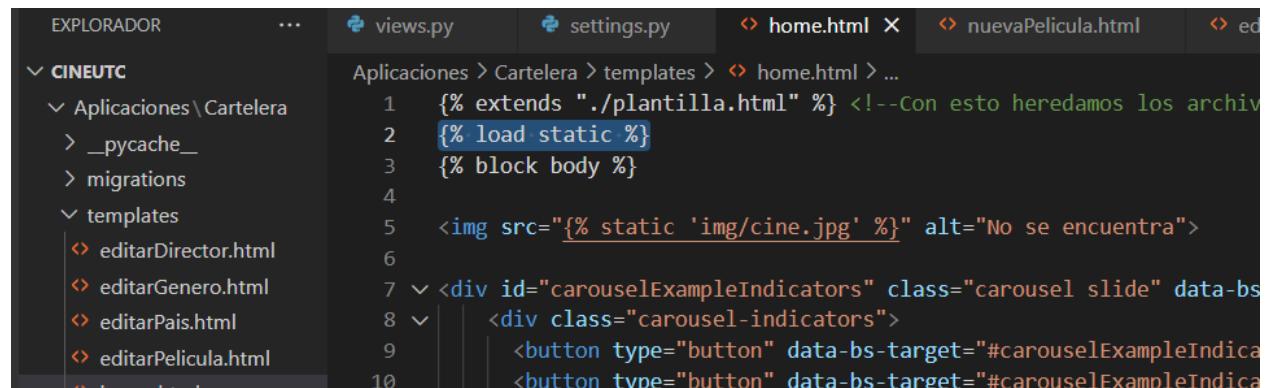
Ahora nos ubicamos en este apartado y añadimos el siguiente codigo



The screenshot shows a code editor with a file tree on the left and the content of `home.html` on the right. The file tree shows a project structure with `CINEUTC`, `Aplicaciones\Cartelera`, and several template files like `editarDirector.html`, `editarGenero.html`, etc. The `home.html` file is selected. The code in `home.html` is:

```
1  {% extends "./plantilla.html" %} <!--Con esto heredamos los archivos de la plantilla-->
2  {% block body %}
3
4      
5
6  </div>
7  <div id="carouselExampleIndicators" class="carousel slide" data-bs-...
8      <div class="carousel-indicators">
9          <button type="button" data-bs-target="#carouselExampleIndica...
10         <button type="button" data-bs-target="#carouselExampleIndica...
11     </div>
```

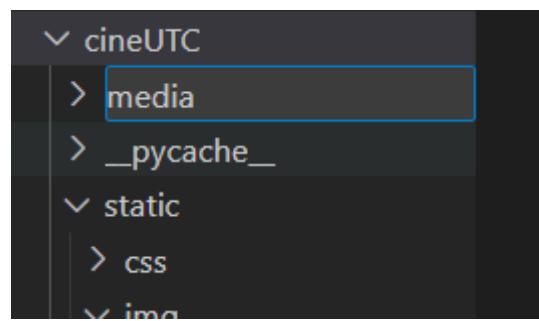
Debemos añadir lo siguiente



The screenshot shows a code editor with a file tree on the left and the content of `home.html` on the right. The file tree is identical to the previous screenshot. The `home.html` file now includes the `{% load static %}` directive at the top:

```
1  {% extends "./plantilla.html" %} <!--Con esto heredamos los archivos de la plantilla-->
2  {% load static %}
3  {% block body %}
4
5      
6
7  </div>
8  <div id="carouselExampleIndicators" class="carousel slide" data-bs-...
9      <div class="carousel-indicators">
10         <button type="button" data-bs-target="#carouselExampleIndica...
11     </div>
```

Vamos a crear una carpeta **media**



Ahora nos ubicamos en el archivo **settings.py** y debemos añadir las siguientes lineas

```
cineUTC > settings.py > MEDIA_URL
114 USE_LZ = true
115
116
117 # static files (css, JavaScript, Images)
118 # https://docs.djangoproject.com/en/4.0/howto/static-files/
119
120 STATIC_URL = '/static/'
121 STATICFILES_DIRS= (os.path.join(BASE_DIR, 'cineUTC/static'),)
122
123 # Default primary key field type
124 # https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field
125
126 MEDIA_URL='/media/'
127 MEDIA_ROOT=os.path.join(BASE_DIR, 'cineUTC/media')
128
129 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
130
131
132
```

Nos ubicamos aquí y añadimos lo siguiente

```
EXPLORADOR ... views.py settings.py urls.py cineUTC X home.html
cineUTC > urls.py > ...
11 | 1. Add a URL to urlpatterns:  path('blog/', include(
12 | Including another URLconf
13 |     1. Import the include() function: from django.urls
14 |     2. Add a URL to urlpatterns:  path('blog/', include(
15 |         ...
16 |         from django.contrib import admin
17 |         from django.urls import path, include
18 |
19 |         from django.conf import settings
20 |         from django.conf.urls.static import static
21 |
22 |         urlpatterns = [
23 |             path('admin/', admin.site.urls),
```

```

EXPLORADOR ... views.py settings.py urls.py cineUTC x home.html nuevaPelicula.html
CINEUTC
  > Aplicaciones
  < cineUTC
    > __pycache__
    > media
    > static
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  bddCineUTC.db
  manage.py

cineUTC > urls.py > ...
  20  FROM django.conf.urls.static import static
  21
  22  urlpatterns = [
  23      path('admin/', admin.site.urls),
  24      path('', include('Aplicaciones.Cartelera.urls'))
  25  ]
  26
  27  #Validacion para visualizar los archivos subidos en modo DEV
  28  if settings.DEBUG:
  29      urlpatterns+=static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)

```

Nos ubicamos aqui

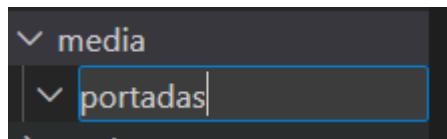
```

CINEUTC
  < Aplicaciones\Cartelera
    > __pycache__
    > migrations
    > templates
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  cineUTC
    > __pycache__
    > media
    > static
    __init__.py
    asgi.py

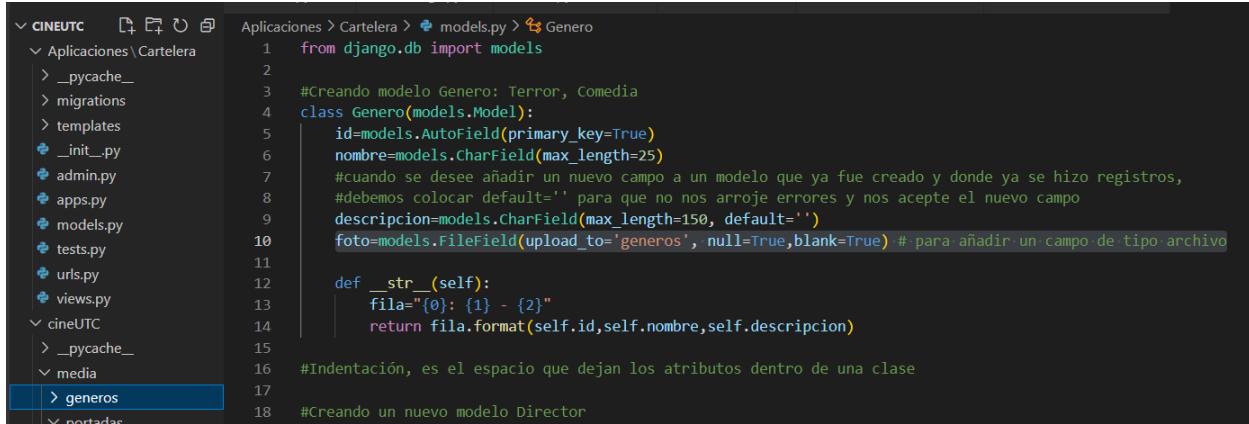
Aplicaciones > Cartelera > models.py > Pelicula > portada
  40  #creando un nuevo modelo Pelicula
  41  class Pelicula(models.Model):
  42      id=models.AutoField(primary_key=True)
  43      titulo=models.CharField(max_length=250)
  44      duracion=models.Timefield(null= True)
  45      sinopsis=models.TextField()
  46      genero=models.ForeignKey(Genero, on_delete=models.CASCADE) #aqui se hace el llamado a las claves foraneas
  47      director=models.ForeignKey(Director, on_delete=models.CASCADE)
  48      portada=models.FileField(upload_to='OJO', null=True, blank=True) # para añadir un campo de tipo archivo
  49
  50      def __str__(self):
  51          fila="{0}: {1}"
  52          return fila.format(self.id, self.titulo)
  53
  54

```

Ahora debemos crear una carpeta **portadas** dentro de media



Hacemos lo mismo para géneros, creamos una carpeta y modificamos el modelo



```

1  from django.db import models
2
3  #Creando modelo Genero: Terror, Comedia
4  class Genero(models.Model):
5      id=models.AutoField(primary_key=True)
6      nombre=models.CharField(max_length=25)
7      #cuando se desee añadir un nuevo campo a un modelo que ya fue creado y donde ya se hizo registros,
8      #debemos colocar default='' para que no nos arroje errores y nos acepte el nuevo campo
9      descripcion=models.CharField(max_length=150, default='')
10     foto=models.FileField(upload_to='generos', null=True, blank=True) # para añadir un campo de tipo archivo
11
12     def __str__(self):
13         fila="{0}: {1} - {2}"
14         return fila.format(self.id, self.nombre, self.descripcion)
15
16     #Indentación, es el espacio que dejan los atributos dentro de una clase
17
18     #Creando un nuevo modelo Director

```

Ahora vamos a la consola y ejecutamos las migraciones

```

C:\proyectos-django\cineUTC>python manage.py makemigrations
Migrations for 'Cartelera':
  Aplicaciones\Cartelera\migrations\0006_genero_foto_pelicula_portada_alter_pelicula_duracion.py
    - Add field foto to genero
    - Add field portada to pelicula
    - Alter field duracion on pelicula

C:\proyectos-django\cineUTC>python manage.py migrate
Operations to perform:
  Apply all migrations: Cartelera, admin, auth, contenttypes, sessions
Running migrations:
  Applying Cartelera.0006_genero_foto_pelicula_portada_alter_pelicula_duracion... OK

C:\proyectos-django\cineUTC>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 25, 2024 - 10:35:25
Django version 4.0.6, using settings 'cineUTC.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[25/Jun/2024 10:35:33] "GET /listadoDirectores/ HTTP/1.1" 200 4990
[25/Jun/2024 10:35:34] "GET /ListadoGeneros/ HTTP/1.1" 200 5911
[25/Jun/2024 10:37:12] "GET /listadoGeneros/ HTTP/1.1" 200 5911
[25/Jun/2024 10:37:15] "GET /nuevoGenero/ HTTP/1.1" 200 4522

```

Ahora nos ubicamos en **nuevoGenero.html** y debemos añadir las siguientes lineas

EXPLORADOR

- CINEUTC
  - Aplicaciones\Cartelera
    - migrations
    - 0005\_pelicula.py
    - 0006\_genero\_foto\_peli...
    - templates
      - editarDirector.html
      - editarGenero.html
      - editarPais.html
      - editarPelicula.html
      - home.html
      - listadoDirectores.html
      - listadoGeneros.html
      - listadoPaises.html
      - listadoPelículas.html
      - nuevaPelicula.html
      - nuevoDirector.html
      - nuevoGenero.html
      - nuevoPais.html
      - plantilla.html
    - \_init\_.py
    - admin.py
    - apps.py
    - models.py

```

    <div class="container-fluid">
        <div class="row">
            <div class="col-md-12">
                <form action="{% url 'guardarGenero' %}" enctype="multipart/form-data" method="post">
                    {% csrf_token %}
                    <label for=""><b>Nombre:</b></label>
                    <input type="text" name="nombre" class="form-control" id="nombre" placeholder="Ingrese e...
                    <label for=""><b>Descripción:</b></label>
                    <textarea name="descripcion" class="form-control" id="descripcion" rows="3" placeholder="...
                    <b for=""><br>Fotografía:</b></label>
                    <input type="file" name="foto" id="foto" accept="image/*" required class="form-control">
                    <br><br>
                    <button type="submit" class="btn btn-primary">
                        Guardar
                    </button>
                    &nbsp;&nbsp;&nbsp;&nbsp;
                    <a href="{% url 'listadoGeneros' %}>Cancelar</a>
                </form>
            </div>
        </div>
    </div>

```

Nos ubicamos en el archivo **views.py** debemos añadir el siguiente código

CINEUTC

- Aplicaciones\Cartelera
  - templates
    - listadoDirectores.html
    - listadoGeneros.html
    - listadoPaises.html
    - listadoPelículas.html
    - nuevaPelicula.html
    - nuevoDirector.html
    - nuevoGenero.html
    - nuevoPais.html
    - plantilla.html
  - \_init\_.py
  - admin.py
  - apps.py
  - models.py
  - tests.py
  - urls.py
  - views.py

```

    #Se recibe el id para eliminar un genero
    def eliminarGenero(request, id):
        generoEliminar=Genero.objects.get(id=id) #el primer id representa el modelo la base de datos
        generoEliminar.delete()
        messages.success(request,"Género eliminado exitosamente.")
        return redirect('listadoGeneros')
    #Renderizando formulario de nuevo genero
    def nuevoGenero(request):
        return render(request, 'nuevoGenero.html')

    #Insertando generos en la base de datos
    def guardarGenero(request):
        nom=request.POST["nombre"]
        des=request.POST["descripcion"]
        fot=request.FILES.get("foto")
        nuevoGenero=Genero.objects.create(nombre=nom, descripcion=des, foto=fot)
        messages.success(request,"Género registrado exitosamente.")
        return redirect('listadoGeneros')

    # Renderizando formulario de actualización
    def editarGenero(request, id):
        generoEditar=Genero.objects.get(id=id)
        return render(request,'editarGenero.html' ,{'generoEditar':generoEditar})

```

Nos ubicamos en **listadoGeneros.html** y debemos añadir las siguientes líneas

Tree view:

- ✓ CINEUTC
  - ✓ Aplicaciones \ Cartelera
    - > \_pycache\_
    - > migrations
    - ✓ templates
      - ▷ editarDirector.html
      - ▷ editarGenero.html
      - ▷ editarPais.html
      - ▷ editarPelicia.html
      - ▷ home.html
      - ▷ listadoDirectores.html
      - ▷ listadoGeneros.html
      - ▷ listadoPaises.html
      - ▷ listadoPelicias.html
      - ▷ nuevaPelicia.html
      - ▷ nuevoDirector.html
      - ▷ nuevoGenero.html
      - ▷ nuevoPais.html
      - ▷ plantilla.html
    - ❸ \_\_init\_\_.py
    - ❸ admin.py
    - ❸ apps.py
    - ❸ models.py
    - ❸ tests.py
    - ❸ urls.py

Code view:

```

Aplicaciones > Cartelera > templates > listadoGeneros.html > div.card > table#tbl-generos.table.table-bordered.table-striped
4   <div class="card">
11    <br>
12    <table class="table table-bordered table-striped table-hover" id="tbl-generos">
13      <thead>
14        <tr>
15          <td>ID</td>
16          <td>NOMBRE</td>
17          <td>DESCRIPCIÓN</td>
18          <td>FOTO</td>
19          <td>ACCIONES</td>
20        </tr>
21      </thead>
22      <tbody>
23        {% for generoTemporal in generos %}
24        <tr>
25          <td>{{generoTemporal.id}}</td>
26          <td>{{generoTemporal.nombre}}</td>
27          <td>{{generoTemporal.descripcion}}</td>
28          <td>
29            <a download href="{{generoTemporal.foto.url}}" target="_blank">
30              
31            </a>
32          </td>
33          <td>
34            <a href="{% url 'editarGenero' generoTemporal.id %}" class="btn btn-warning">Editar</a>
35            <a href="{% url 'eliminarGenero' generoTemporal.id %}" onclick="return confirm('¿Deseas eliminar este genero?')">Eliminar</a>
36          </td>

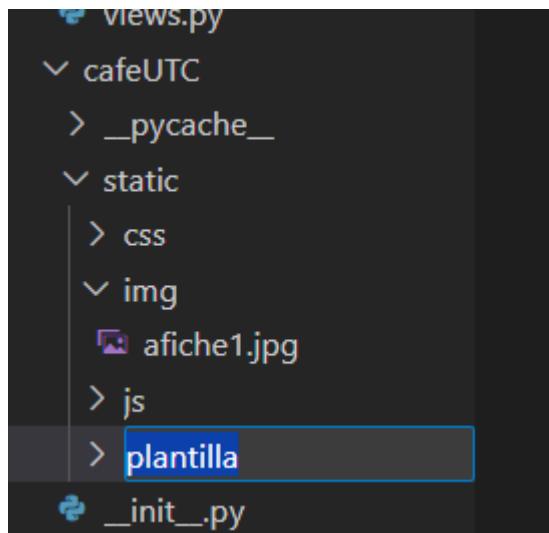
```

## USO DE PLANTILLAS

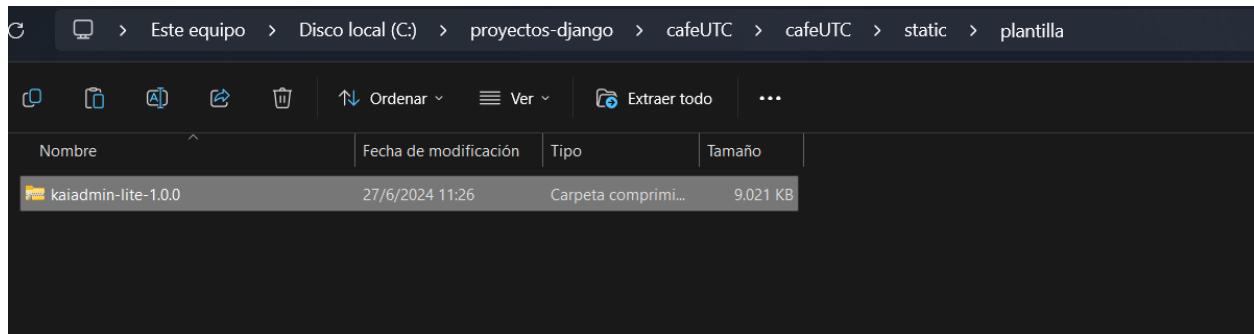
Ingresamos a esa página y descargamos

The screenshot shows the ThemeWagon website interface. At the top, there's a navigation bar with links like Gmail, Clases, Universidad Técnica, Cambridge One, UTC, WhatsApp, YouTube, Reverso, Mi unidad - Google, GitHub, Facebook, and Todos los marcadores. Below the navigation is a search bar with the placeholder "Search themes...". The main content area features a theme preview for "KaiAdmin – Free Responsive Bootstrap 5 Admin Dashboard Template". The preview includes a rating of 5 stars and 0 customer reviews. The theme itself is a dark-themed dashboard with various widgets: "Dashboard" (1,294 visitors, 1,003 subscribers, \$1,345 sales, 576 orders), "User Statistics" (line chart showing user growth from June to December), and "Daily Sales" (\$4,578.58). To the right of the preview, there's a "Free download" button, a "License" link, and a section stating "4691 Downloads". Below this, there's a testimonial about design and development tips and a note about ads via Carbon.

Vamos a crear una nueva carpeta plantilla

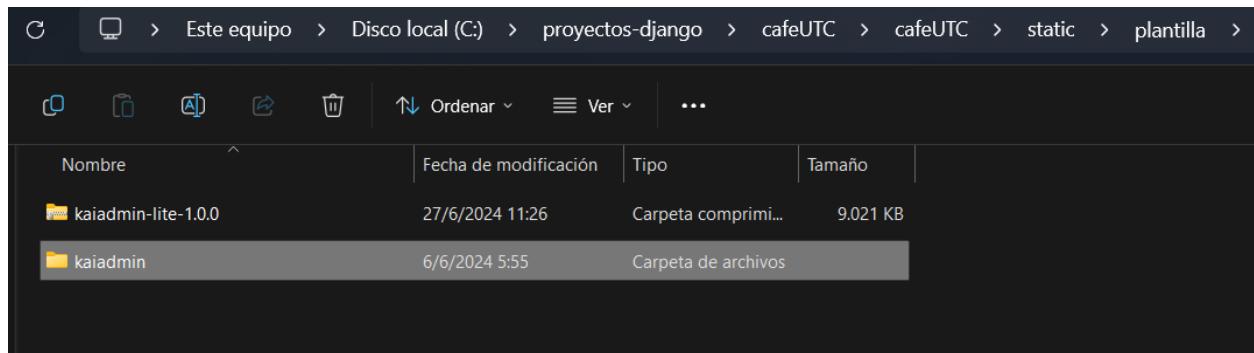


Ahora el archivo descargado en la siguiente ubicación, dentro de la carpeta plantilla



Damos en **Extraer aquí** y cambiamos de nombre de la carpeta extraída, algo más corto

**Kaiadmin**



Ahora vamos a ese archivo, ahora desde la línea 107 hacia arriba cortamos

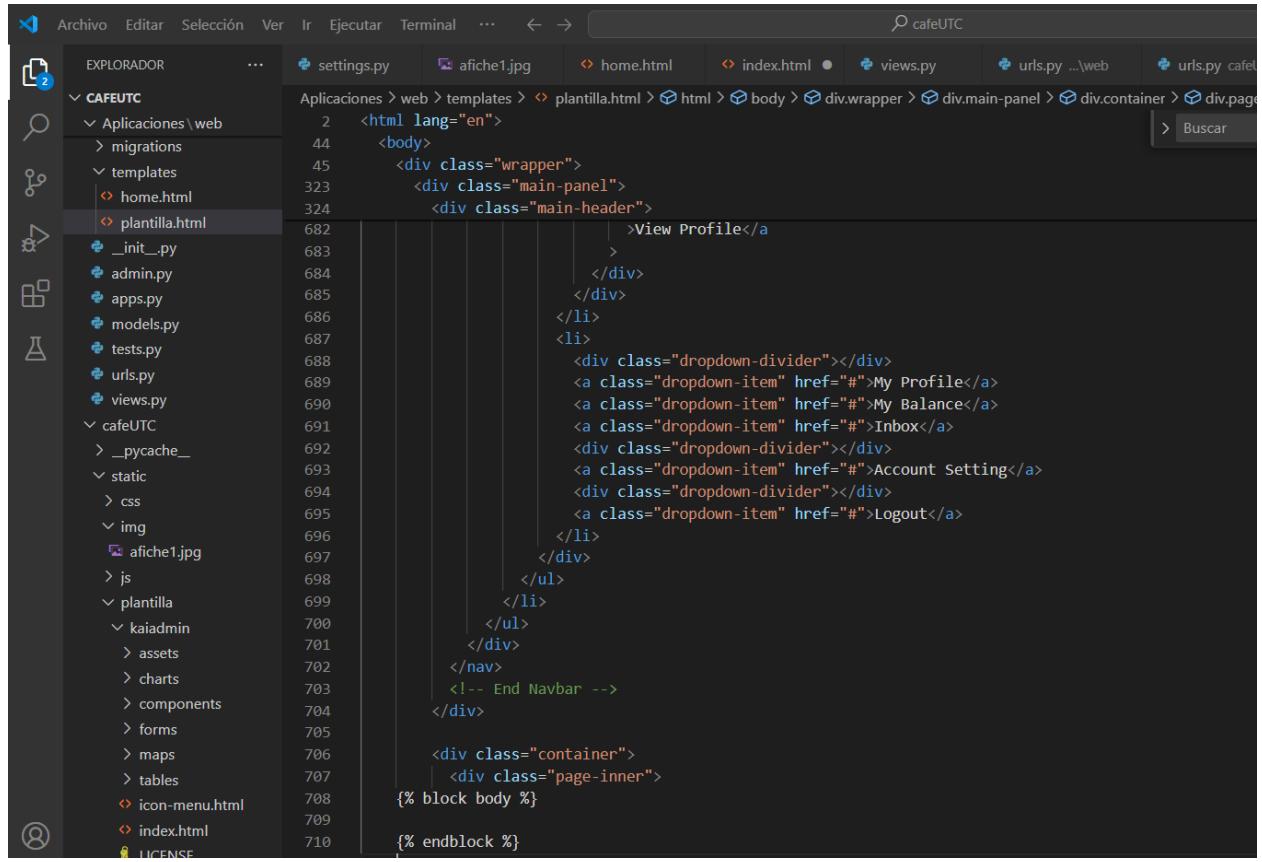
The screenshot shows a code editor interface with the following details:

- File Explorer (Left):** Shows the project structure for "CAFEUTC". The "templates" folder contains "index.html". Other files include "settings.py", "afiche1.jpg", "home.html", "plantilla.html", "init\_.py", "admin.py", "apps.py", "models.py", "tests.py", "urls.py", and "views.py".
- Code Editor (Right):** The file "index.html" is open. The code is a template for a dashboard or profile page. It includes a navigation bar with items like "View Profile", "My Profile", "My Balance", "Inbox", "Account Setting", and "Logout". The code uses Bootstrap classes such as "btn", "btn-xs", "btn-secondary", "dropdown-item", and "dropdown-divider".
- Toolbar:** Includes standard options like Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, and a search bar labeled "Buscar".

The code in "index.html" is as follows:

```
<html lang="en">
    <body>
        <div class="wrapper">
            <div class="main-panel">
                <div class="main-header">
                    <li>
                        <a href="profile.html" class="btn btn-xs btn-secondary btn-sm">View Profile</a>
                    </li>
                    <div class="dropdown-divider"></div>
                    <a class="dropdown-item" href="#">My Profile</a>
                    <a class="dropdown-item" href="#">My Balance</a>
                    <a class="dropdown-item" href="#">Inbox</a>
                    <div class="dropdown-divider"></div>
                    <a class="dropdown-item" href="#">Account Setting</a>
                    <div class="dropdown-divider"></div>
                    <a class="dropdown-item" href="#">Logout</a>
                </li>
            </div>
        </div>
    </body>
</html>
```

Y copiamos aquí en el archivo plantilla.html



```
Explorador de archivos
CAFETUC
  Aplicaciones\web
    migrations
    templates
      home.html
      plantilla.html
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  cafeUTC
    __pycache__
    static
      css
      img
        afiche1.jpg
      js
    plantilla
      kaiadmin
        assets
        charts
        components
        forms
        maps
        tables
        icon-menu.html
        index.html
    LICENSE

Archivos abiertos
settings.py
afiche1.jpg
home.html
index.html
views.py
urls.py
urls.py

Aplicaciones > web > templates > plantilla.html > html > body > div.wrapper > div.main-panel > div.container > div.page
  2   <html lang="en">
  44     <body>
  45       <div class="wrapper">
  323         <div class="main-panel">
  324           <div class="main-header">
  682             |   <a href="#">View Profile</a>
  683             |
  684             |   </div>
  685             </div>
  686             </li>
  687             <li>
  688               <div class="dropdown-divider"></div>
  689               <a class="dropdown-item" href="#">My Profile</a>
  690               <a class="dropdown-item" href="#">My Balance</a>
  691               <a class="dropdown-item" href="#">Inbox</a>
  692               <div class="dropdown-divider"></div>
  693               <a class="dropdown-item" href="#">Account Setting</a>
  694               <div class="dropdown-divider"></div>
  695               <a class="dropdown-item" href="#">Logout</a>
  696             |
  697             </li>
  698             </ul>
  699           </li>
  700           </ul>
  701         </div>
  702         </nav>
  703         <!-- End Navbar -->
  704       </div>
  705       <div class="container">
  706         |   <div class="page-inner">
  707           {% block body %}
  708           {% endblock %}
  709       </div>
```

Ahora para el footer desde la línea 623 copiamos todo hacia abajo

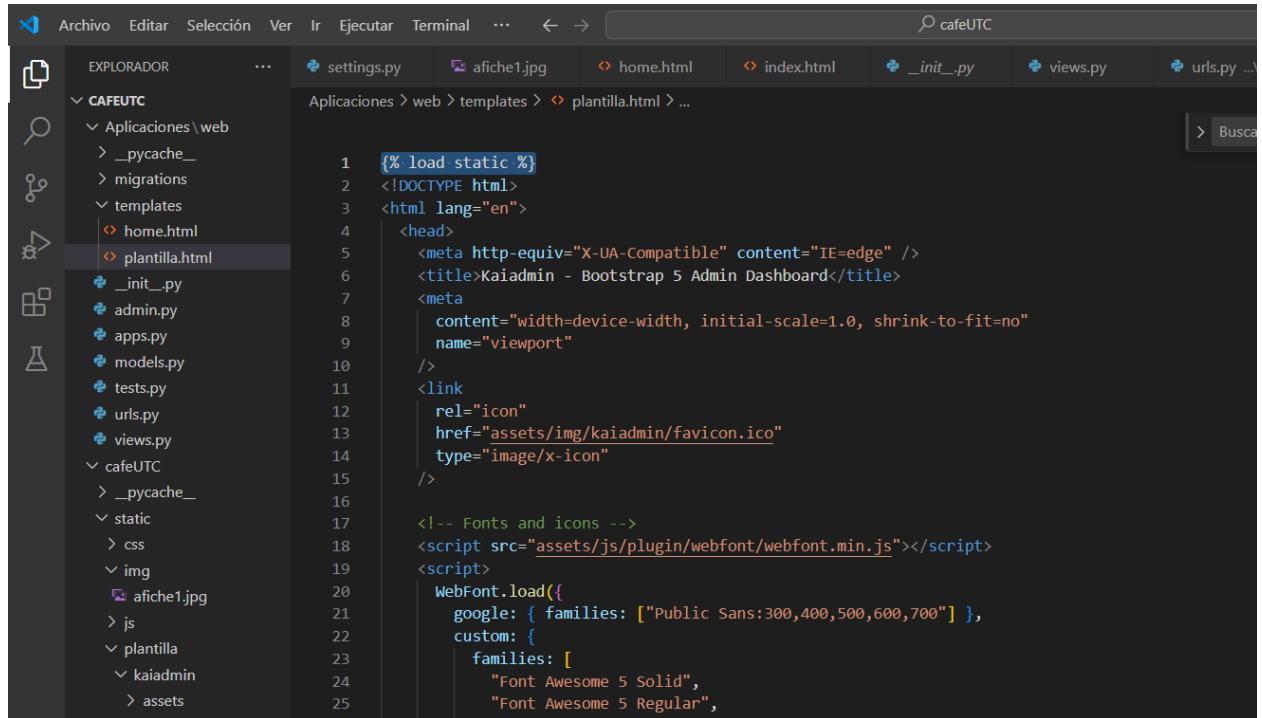
Control+F para buscar en las líneas Logout para el menú.. Y footer

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with various icons for navigation and file operations. The main area has a top navigation bar with buttons for Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, and a search bar. Below the navigation is a file tree titled 'EXPLORADOR'. The tree shows a project structure under 'CAFEUTC' with subfolders like 'Aplicaciones\web', 'templates', 'cafeUTC', 'static', 'img', 'js', and 'plantilla'. Under 'templates', 'plantilla.html' is selected. To the right of the file tree is a large code editor window displaying an HTML template file. The file contains code for a footer section, including navigation links and copyright information. A blue selection bar highlights several lines of code in the footer section, specifically lines 714 through 738. The code editor also shows line numbers from 3 to 738.

```
<html lang="en">
  <body>
    <div class="wrapper">
      <div class="main-panel">
        <div class="container">
          <div class="page-inner">
            {% block body %}

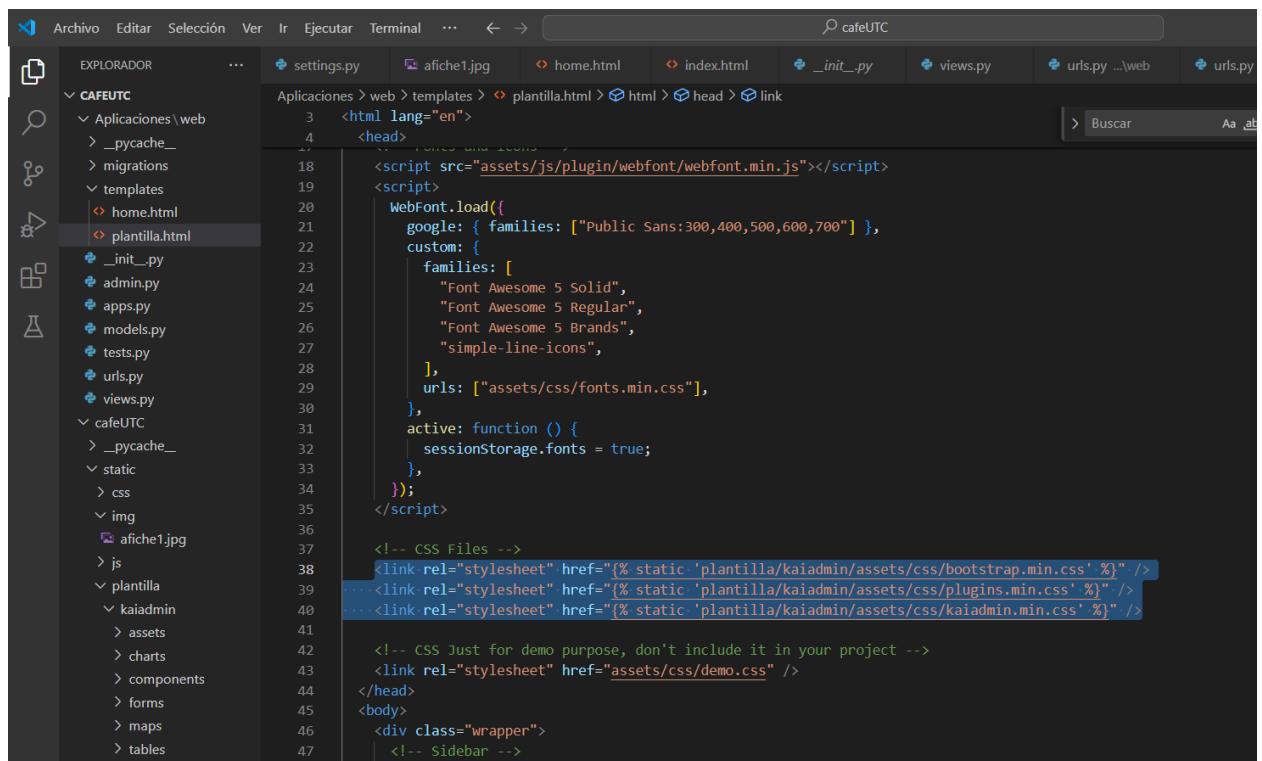
            {% endblock %}
            <footer class="footer">
              <div class="container-fluid d-flex justify-content-between">
                <nav class="pull-left">
                  <ul class="nav">
                    <li class="nav-item">
                      <a class="nav-link" href="http://www.themekita.com">
                        ThemeKita
                      </a>
                    </li>
                    <li class="nav-item">
                      <a class="nav-link" href="#"> Help </a>
                    </li>
                    <li class="nav-item">
                      <a class="nav-link" href="#"> Licenses </a>
                    </li>
                  </ul>
                </nav>
                <div class="copyright">
                  2024, made with <i class="fa fa-heart heart text-danger"></i> by
                  <a href="http://www.themekita.com">ThemeKita</a>
                </div>
                <div>
                  Distributed by
                  <a target="_blank" href="https://themewagon.com/">ThemeWagon</a>.
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

En el archivo plantilla modificamos esas líneas de código



The screenshot shows a code editor interface with a dark theme. The left sidebar displays a file tree for a project named 'CAFEUTC'. The 'templates' folder contains 'home.html' and 'plantilla.html'. The right pane shows the content of 'plantilla.html'. The code includes HTML tags like <head> and <link>, and a script block for loading fonts from 'webfont.min.js'.

```
% load static %>DOCTYPE html<html lang="en"><head><meta http-equiv="X-UA-Compatible" content="IE=edge" /><title>Kaiadmin - Bootstrap 5 Admin Dashboard</title><meta content="width=device-width, initial-scale=1.0, shrink-to-fit=no" name="viewport" /><link rel="icon" href="assets/img/kaiadmin/favicon.ico" type="image/x-icon" /><!-- Fonts and icons --><script src="assets/js/plugin/webfont/webfont.min.js"></script><script>WebFont.load({ google: { families: ["Public Sans:300,400,500,600,700"] }, custom: { families: [ "Font Awesome 5 Solid", "Font Awesome 5 Regular", "Font Awesome 5 Brands", "simple-line-icons" ], urls: ["assets/css/fonts.min.css"] }, active: function () { sessionStorage.fonts = true; } });</script><!-- CSS Files --><link rel="stylesheet" href="{% static 'plantilla/kaiadmin/assets/css/bootstrap.min.css' %}" /><link rel="stylesheet" href="{% static 'plantilla/kaiadmin/assets/css/plugins.min.css' %}" /><link rel="stylesheet" href="{% static 'plantilla/kaiadmin/assets/css/kaiadmin.min.css' %}" /><!-- CSS Just for demo purpose, don't include it in your project --><link rel="stylesheet" href="assets/css/demo.css" /></head><body><div class="wrapper"><!-- Sidebar -->
```

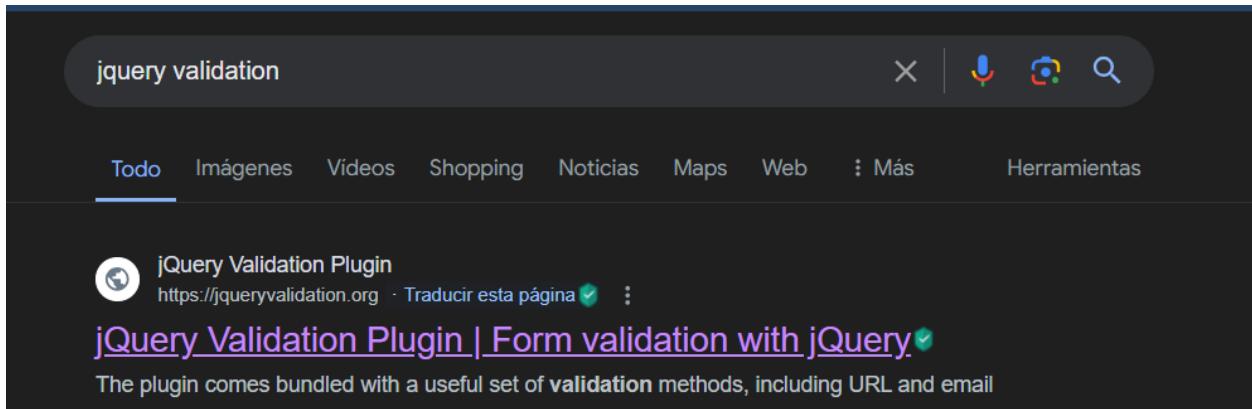


This screenshot shows the same code editor interface, but the 'head' section of the 'plantilla.html' file is selected. The code within the head section includes the font loading script and the CSS links mentioned above.

```
<html lang="en"><head><script src="assets/js/plugin/webfont/webfont.min.js"></script><script>WebFont.load({ google: { families: ["Public Sans:300,400,500,600,700"] }, custom: { families: [ "Font Awesome 5 Solid", "Font Awesome 5 Regular", "Font Awesome 5 Brands", "simple-line-icons" ], urls: ["assets/css/fonts.min.css"] }, active: function () { sessionStorage.fonts = true; } });</script><!-- CSS Files --><link rel="stylesheet" href="{% static 'plantilla/kaiadmin/assets/css/bootstrap.min.css' %}" /><link rel="stylesheet" href="{% static 'plantilla/kaiadmin/assets/css/plugins.min.css' %}" /><link rel="stylesheet" href="{% static 'plantilla/kaiadmin/assets/css/kaiadmin.min.css' %}" /><!-- CSS Just for demo purpose, don't include it in your project --><link rel="stylesheet" href="assets/css/demo.css" /></head>
```

# CLASE MARTES 2/07/2024

Vamos a trabajar con jquery-validation



Damos en download

The screenshot shows the homepage of jqueryvalidation.org. At the top, there is a navigation bar with links to Gmail, Clases, Universidad Técnica, Cambridge One, UTC, WhatsApp, YouTube, Reverso, Mi unidad, and Google Translate (set to English). The main content area has a heading "jQuery Validation Plugin" and a sub-heading "Form validation with jQuery". It mentions that the plugin was initially written by Jörn Zaefferer and maintained by the jQuery team. It notes that Markus Staab took over maintenance in 2016 and that Brahim Arki assisted him. The current version is 1.19.5 and the license is MIT. There are sections for "Files" (Download, Changelog, Demos, Documentation, GitHub Repository) and "Installation via Package Managers" (Bower command). A sidebar on the right lists categories: Methods, Plugin Methods, Selectors, Validator, and META (Log in, Entries RSS, Comments RSS, WordPress.org).

Y descargamos el Assets , el .zip

**Chore**

- Add CodeQL analysis [3d3c1fb](#)

**Core**

- Fixed jQuery .submit() event shorthand deprecation notice [#2430](#)
- Fixed ReDos vulnerability in url, and url2 validation [5bbd80d](#)

**Localisation**

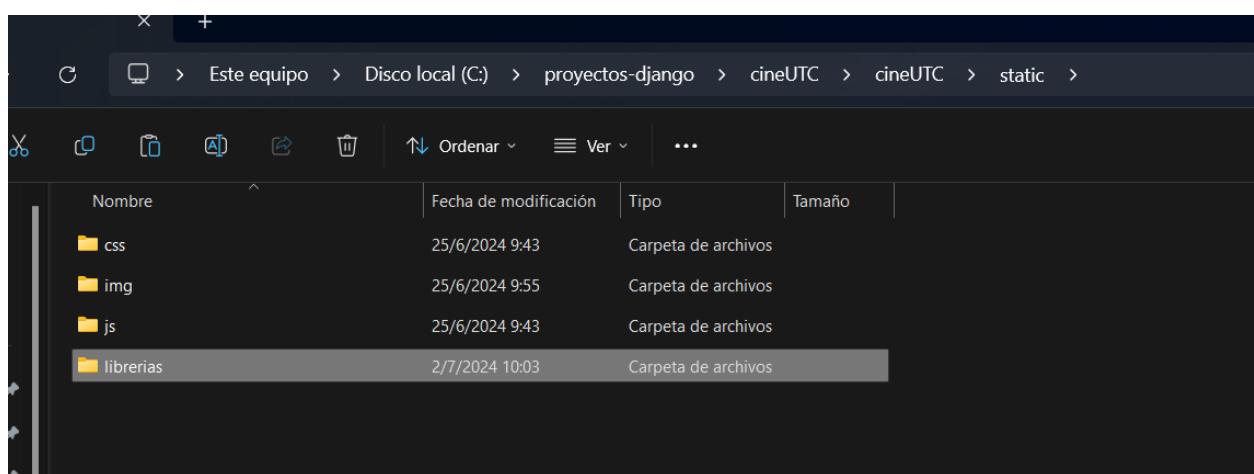
- Added periods to messages [#2266](#)

**Assets** 3

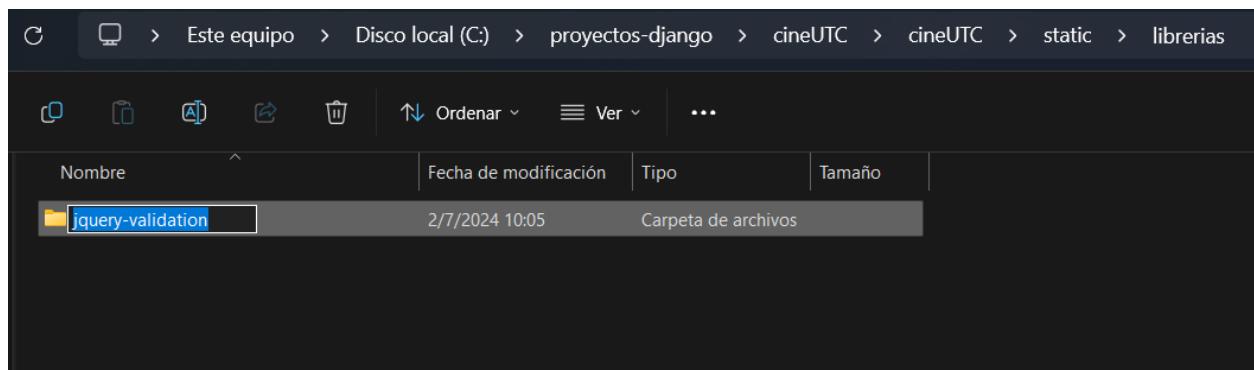
<a href="#">jquery-validation-1.19.5.zip</a>	1.35 MB	Jul 6, 2024	
<a href="#">Source code (zip)</a>		Jul 1, 2024	
<a href="#">Source code (tar.gz)</a>		Jul 1, 2024	

🕒 50 🌟 5 💬 5 🎉 5 ❤️ 9 🚀 6 📸 4 65 people reacted

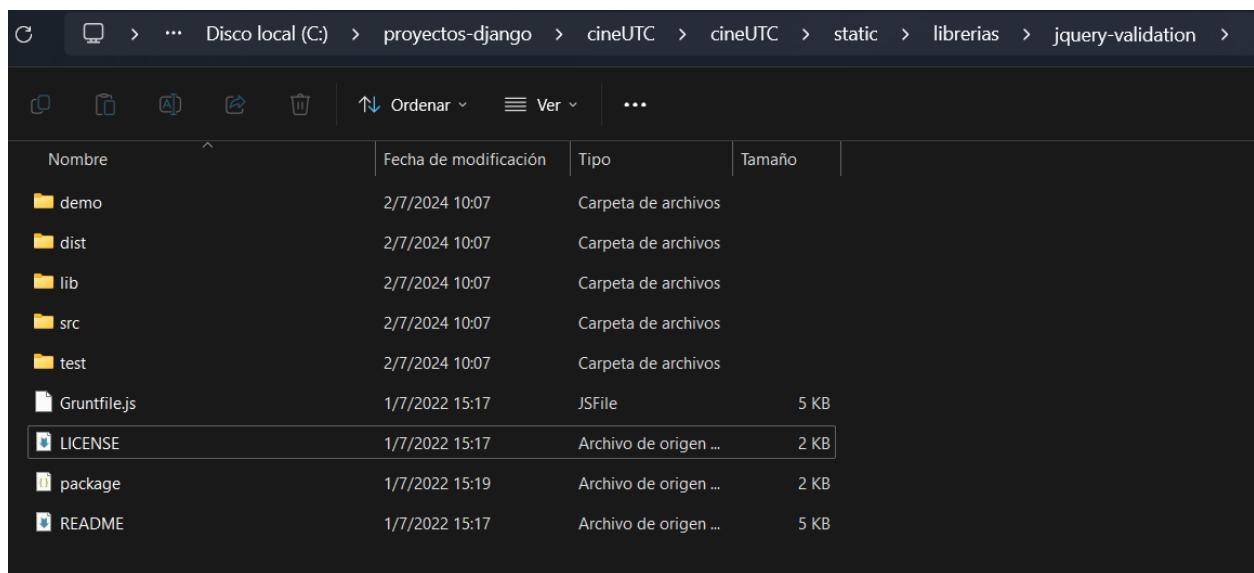
Ahora nos vamos a la siguiente ruta y debemos crear una carpeta **librerias**



Accedemos a esa carpeta y creamos una nueva carpeta **jquery-validation**



Y dentro de esa carpeta copiamos el archivo descargado y lo extraemos, nos quedara asi:



Vamos a visual, vamos a copiar el nombre de esa carpeta

The image shows a file explorer on the left and a code editor on the right.

**File Explorer (Left):**

- CINEUTC
  - cineUTC
    - media
      - generos
        - icono\_doctor.png
        - icono\_hospital.png
        - modulodecodigoci.png
        - novela.jpg
        - salud.png
        - terror.jpg
      - portadas
        - icono\_maternidad.png
        - it.jpg
        - titanic\_BX0ZL2I.jpg
        - titanic.jpg
        - venom.jpg
    - static
      - css
      - img
      - js
      - librerias\jquery-validation
        - demo
        - dist
          - localization
          - additional-methods.js**
          - additional-methods.min.js**
          - jquery.validate.js**
          - jquery.validate.min.js**
        - lib
        - src
        - test

Ahora nos ubicamos en el archivo plantilla.html, y añadimos lo siguiente

The screenshot shows the PyCharm interface with the following details:

- File Menu:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, Terminal, ...
- Toolbar:** Back, Forward, Refresh, Search (with placeholder "cineUTC")
- File Browser (Left):** EXPLORADOR, CINEUTC (expanded), Aplicaciones \ Cartelera, \_\_pycache\_\_, migrations, templates (expanded), editarDirector.html, editarGenero.html, editarPaís.html, editarPelícula.html, home.html, listadoDirectores.html, listadoGeneros.html, listadoPaíses.html, listadoPelículas.html, nuevaPelícula.html, nuevoDirector.html, nuevoGenero.html, nuevoPaís.html, plantilla.html.
- Code Editor (Right):** File: plantilla.html, Content:

```
Aplicaciones > Cartelera > templates > plantilla.html > ...
1 [% load static %]
2 <!DOCTYPE html>
3 <html lang="es">
4   <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Cine UTC</title>
8     <!-- Importando jQuery -->
9     <script src="https://code.jquery.com/jquery-3.7.1.min.js" integrity="sha256-JqT3SQfawRcv/BHPTkBvso0EvtFFmqP/lYI...">
10    <!-- Importando bootstrap -->
11    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" integrity="sha384-T3cCoI6ULRA97nNeNCo7RxnatzjDC5McM1XXSR16ASXEV/Dwmyk2MPK8M2HN" crossorigin="anonymous">
12    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js" integrity="sha384-B8t1+eGJgQUWxJ7pMwbEyER41lg+01SP+16Ep7Q9Q+zuX6gSbd85u4mG4QzX+" crossorigin="anonymous"></script>
13    <!-- Importando dataTables -->
14    <link rel="stylesheet" href="https://cdn.datatables.net/2.0.8/css/dataTables.dataTables.min.css" integrity="sha384-EGJgQUWxJ7pMwbEyER41lg+01SP+16Ep7Q9Q+zuX6gSbd85u4mG4QzX+" crossorigin="anonymous"></script>
15    <!-- Importando SweetAlert -->
16    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
17    <!-- Importando Jquery Validate -->
18    <script src="{% static 'librerias/jquery-validate/dist/jquery.validate.min.js' %}"></script>
19  </head>
20  <body style="background-color: #fcfcfc;">
21    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
22      <div class="container-fluid">
```

Vamos al archivo nuevoGenero y añadimos lo siguiente

EXPLORADOR

CINEUTC

- Aplicaciones \ Cartelera
  - \_pycache\_
  - migrations
  - templates
    - editarDirector.html
    - editarGenero.html
    - editarPais.html
    - editarPelcula.html
    - home.html
    - listadoDirectores.html
    - listadoGeneros.html
    - listadoPaises.html
    - listadoPelculas.html
    - nuevaPelcula.html
    - nuevoDirector.html
    - nuevoGenero.html
    - nuevoPais.html
    - plantilla.html
  - \_\_init\_\_.py
  - admin.py
  - apps.py
  - models.py
  - tests.py
  - urls.py
  - views.py

nuevoGenero.html

Aplicaciones > Cartelera > templates > nuevoGenero.html > div.container-fluid > div.col-md-12 > form#frm\_nuevo\_genero

```
1  {% extends "../plantilla.html" %}  
2  {% block body %}  
3  <div class="container-fluid">  
4      <div class="row">  
5          <div class="col-md-12">  
6              <form action="{% url 'guardarGenero' %}" enctype="multipart/form-data" id="frm_nuevo_genero" method="post">  
7                  {% csrf_token %}  
8                  <label for=""><b>Nombre:</b></label>  
9                  <input type="text" name="nombre" class="form-control" id="nombre" placeholder="Ingrese el nombre del genero">  
10                 <label for=""><b>Descripción:</b></label>  
11                 <textarea name="descripcion" class="form-control" id="descripcion" rows="3" placeholder="Ingrese la descripción"></text  
12                 <br>  
13                 <b><br>Fotografía:</b></label>  
14                 <input type="file" name="foto" id="foto" accept="image/*" required class="form-control">  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  {% endblock %}
```

Agregamos el siguiente código a nuevoGenero.html

```
CINEUTC Aplicaciones > Cartelera > templates > nuevoGenero.html > div.container-fluid > div.row > div.col-md-12
Aplicaciones \ Cartelera
  > __pycache__
  > migrations
  < templates
    > editarDirector.html
    > editarGenero.html
    > editarPais.html
    > editarPelicia.html
    > home.html
    > listadoDirectores.html
    > listadoGeneros.html
    > listadoPaises.html
    > listadoPelicias.html
    > nuevaPelicia.html
    > nuevoDirector.html
    < nuevoGenero.html
      > nuevoPais.html
      > plantilla.html
      < __init__.py
      < admin.py
      < apps.py
      < models.py
      < tests.py
      < urls.py
      < views.py
    < cineUTC
      > __pycache__
    < media
    < directores
      < ArcGIS_logo.png

Aplicaciones > Cartelera > templates > nuevoGenero.html > div.container-fluid > div.row > div.col-md-12
  30   <br><br><br><br><br>
  31   <!--Para que funcione java scrip se utiliza script, maneras de acceder al DOM-->
  32   <script>
  33     $("#frm_nuevo_genero").validate({
  34       rules:{
  35         nombre:{
  36           required:true,
  37           minlength: 3,
  38           maxlength: 10
  39         },
  40         descripcion:{
  41           required:true
  42         },
  43         foto:{
  44           required:true
  45         }
  46       },
  47       messages:{
  48         nombre:{
  49           required:"Por favor ingrese el nombre",
  50           minlength:"El nombre debe tener al menos 3 letras minimo",
  51           maxlength:"Máximo 10 caracteres"
  52         },
  53         descripcion:{
  54           required:"La descripción es obligatoria"
  55         },
  56         foto:{
  57           required:"Por favor seleccione la foto"
  58         }
  59       }
  60     });
  61   </script>
  62
  63   {% endblock %}
```

Agregamos estilos

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with icons for file operations like Open, Save, Find, and Refresh. The main area has tabs for 'Pais.html', 'views.py', 'settings.py', 'urls.py', 'cineUTC', and 'home.html'. A search bar at the top right contains the text 'cineUTC'. The central pane displays the content of 'plantilla.html':

```
Aplicaciones > Cartelera > templates > plantilla.html > html > body > style
3   <html lang="es">
25  <body style="background-color: #fcfcfc;">
    ...
97  </html>
```

The code includes Jinja2 templating with blocks like {% for %}, {% if %}, and {% endif %}. It also includes CSS styles within a style block:

```
/*letras de color rojo*/
label.error{
    color: red;
}
/*Borde en rojo*/
.form-control.error{
    border: 1px solid red;
}
```

El resultado es el siguiente:

**Nombre:**

hh

El nombre debe tener al menos 3 letras minimo

**Descripción:**

Ingrese la descripción

La descripción es obligatoria

**Fotografía:**

Seleccionar archivo

Ningún archivo seleccionado

Por favor seleccione la foto

[Guardar](#)

[Cancelar](#)