



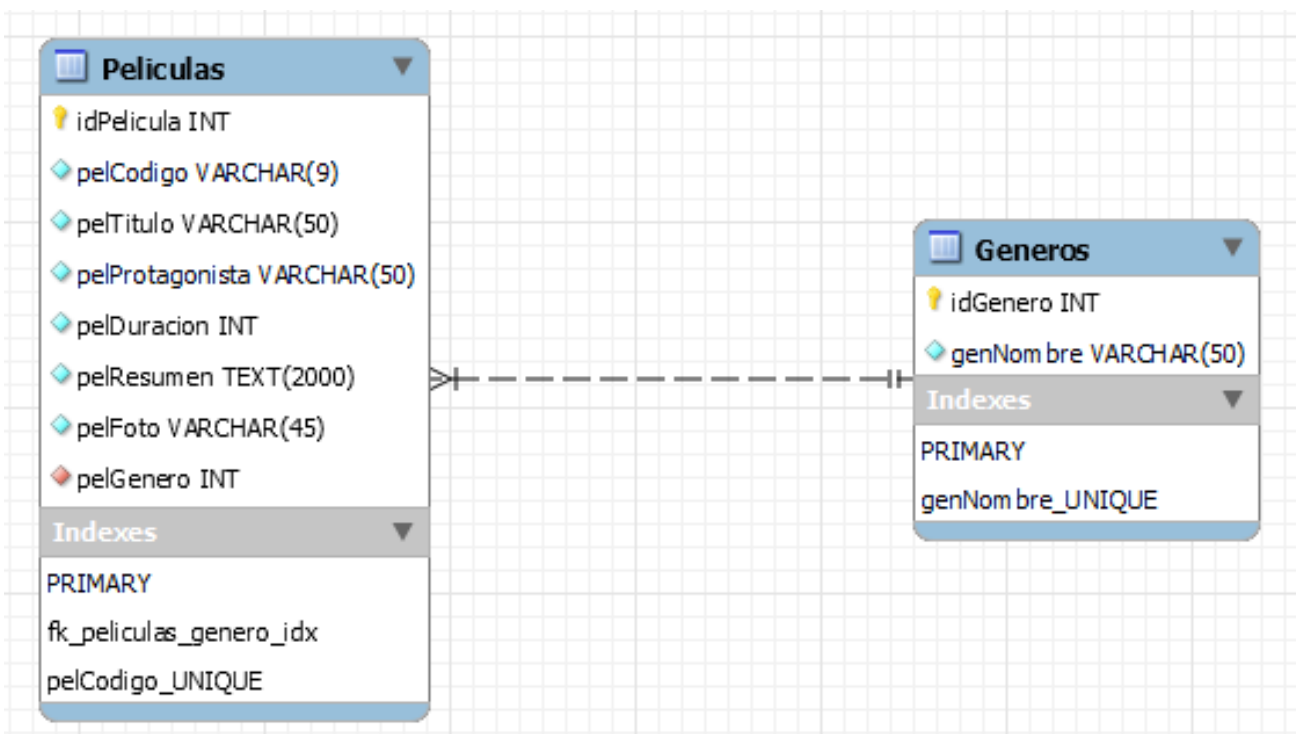
PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL
FORMATO GUÍA DE APRENDIZAJE – MATERIAL DE APOYO
DESARROLLO APLICACIONES WEB EN PYTHON

CREACIÓN DE APLICACIONES WEB EN PYTHON CON EL FRAMEWORK DJANGO



Para iniciar primero crear una carpeta donde se va a crear el proyecto. Después abrir visual studio code y abrir dicha carpeta. En mi caso la carpeta la he llamado **Proyecto Inicial Django**.

Base de Datos a utilizar






Pasos:

Crear un entorno virtual de trabajo para el proyecto

```
D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django>python -m venv entorno
```




```
D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django>
```

Activar el entorno

```
D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django>cd entorno
```

```
D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django\entorno>cd scripts
```


```
D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django\entorno\Scripts>activate
```



```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django\entorno\Scripts>
```

Ahora nos ubicamos en la raíz de la carpeta para instalar django así:

```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django>pip install Django
```





```
tzdata-2024.1-py2.py3-none-any.whl.metadata
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Downloading Django-5.0.4-py3-none-any.whl (8.2 MB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.2/8.2 MB 4.2 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 5.3 MB/s eta 0:00:00
0Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-5.0.4 asgiref-3.8.1 sqlparse-0.5.0 tzdata-2024.1

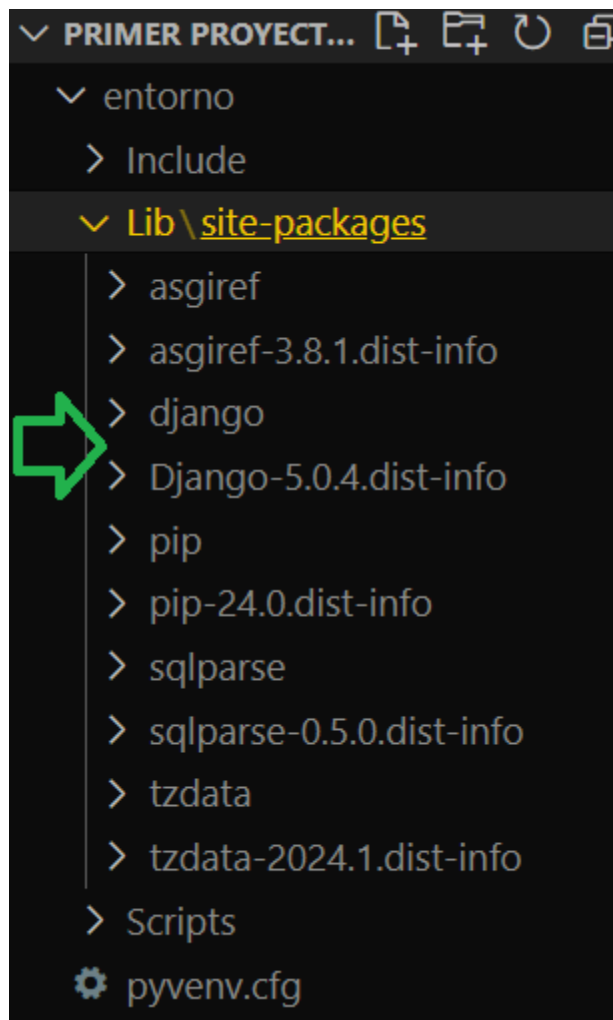
[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip

(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2
024\Primer Proyecto Django>
```

Si solicita actualizar pip hacer el siguiente comando:

```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2
024\Primer Proyecto Django>python -m pip install --upgrade pip
```

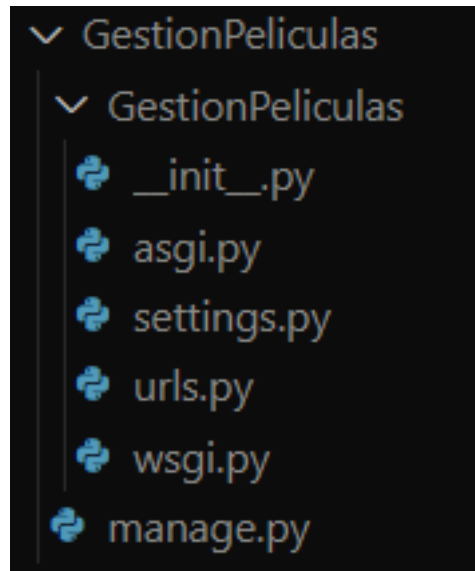
Podemos verificar en la lib dentro de la carpeta del entorno para verificar si aparece django.



Ahora vamos a crear el proyecto ubicados en la carpeta raíz, ejecutando el siguiente comando.

```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django>django-admin startproject GestionPelículas
```

Al ejecutarse se crea una Carpeta con ese Nombre y dentro de ella también se crea una carpeta con el mismo nombre y un archivo **manage.py**. Dentro de la subcarpeta se crean varios archivos como se muestra en la siguiente imagen.



Después vamos ir conociendo para que sirve cada uno de los archivos.

Aquí ya podemos iniciar el servidor para probar si quedo bien creado el proyecto. Ejecutar el siguiente comando verificando que estemos en la carpeta raíz **GestiónPelículas**.

```
python manage.py runserver
```

```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Pro
yecto Django\GestionPelículas>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
?: (urls.W002) Your URL pattern '/' has a route beginning with a '/'. Remove this slash as it
is unnecessary. If this pattern is targeted in an include(), ensure the include() pattern has
a trailing '/'.

System check identified 1 issue (0 silenced).
April 20, 2024 - 19:52:59
Django version 5.0.4, using settings 'GestionPelículas.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Verificamos e ingresamos a la url indicada y debe aparecer:



django

View [release notes](#) for Django 5.0



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

Ahora vamos a crear una aplicación en el proyecto, para ello vamos a ejecutar el siguiente comando verificando que estemos en la carpeta raíz **GestiónPelículas**.

```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django\GestionPelículas>
```

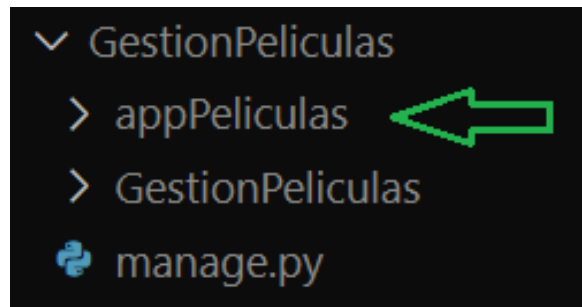
```
python manage.py startapp appPelículas
```

Ahora escribimos el siguiente comando para crear una aplicación llamada **appPelículas**:

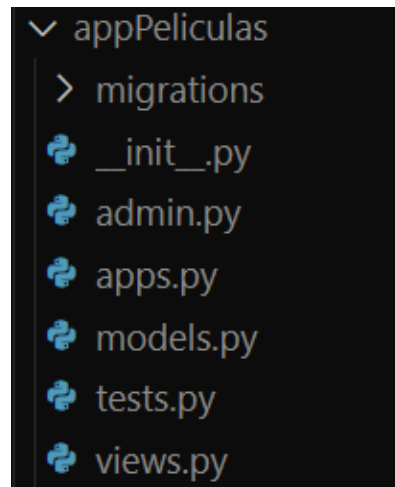
```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django\GestionPelículas>python manage.py startapp appPelículas
```



Al ejecutar dicho comando se creo una carpeta con el nombre de la aplicación **appPeliculas** así:

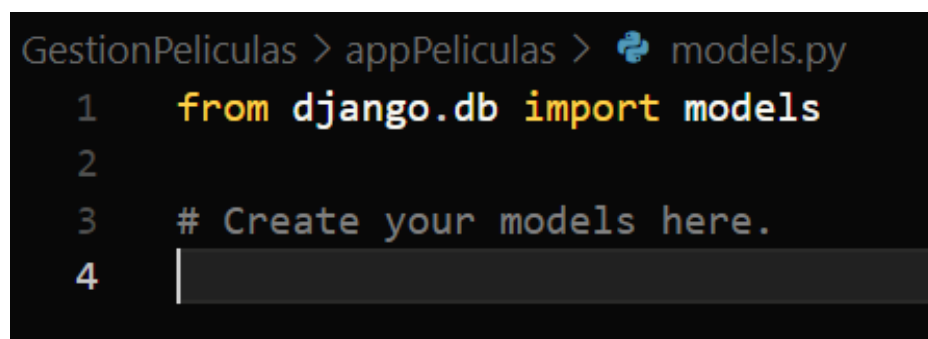


La carpeta **appPeliculas** contiene una carpeta llamada **migrations** y unos archivos como se muestra en la siguiente imagen:



Vamos ahora a proceder a crear las clases en el modelo para con ellas después crear la base de datos.

Abrimos el archivo **models.py**:





Nuestro modelo con las clases quedaría así:

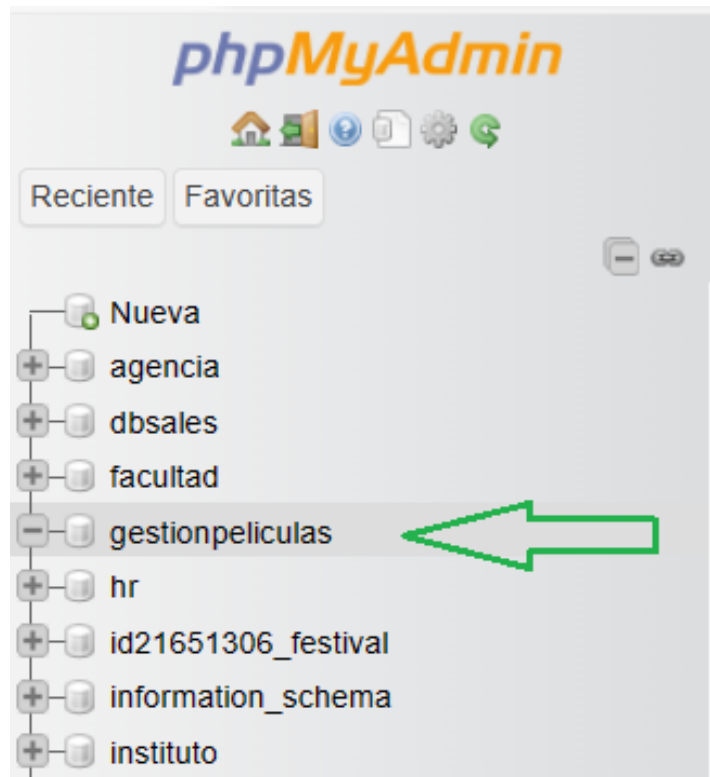
```
GestionPeliculas > appPeliculas > models.py > ...
1  from django.db import models
2  # Create your models here.
3
4  class Genero(models.Model):
5      |   genNombre = models.CharField(max_length=50, unique=True)
6
7  class Pelicula(models.Model):
8      |   pelCodigo = models.CharField(max_length=9)
9      |   pelTitulo = models.CharField(max_length=50)
10     |   pelProtagonista = models.CharField(max_length=50)
11     |   pelDuracion = models.IntegerField()
12     |   pelResumen = models.CharField(max_length=2000)
13     |   pelFoto = models.ImageField(upload_to=f"fotos/", null=True, blank=True)
14     |   pelGenero = models.ForeignKey(Genero, on_delete=models.PROTECT)
```

Como utilizamos el tipo de dato ImageField debemos instalar la librería **Pillow** para tratamiento de imágenes.

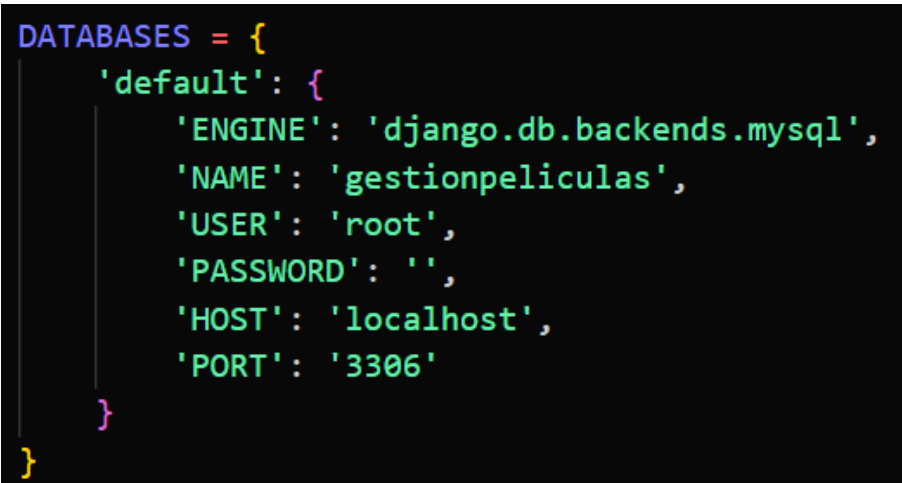
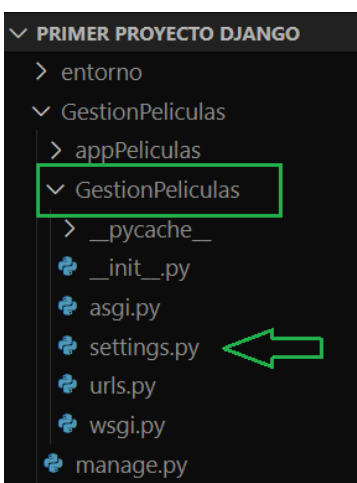
```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proye
cto Django\GestionPeliculas>pip install pillow
Collecting pillow
  Downloading pillow-10.3.0-cp312-cp312-win_amd64.whl.metadata (9.4 kB)
  Downloading pillow-10.3.0-cp312-cp312-win_amd64.whl (2.5 MB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.5/2.5 MB 3.3 MB/s eta 0:00:00
Installing collected packages: pillow
Successfully installed pillow-10.3.0

(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proye
cto Django\GestionPeliculas>
```

Como se requiere que desde django se cree las tablas en la base de datos, entonces primero vamos a ir a mysql y creamos una base de datos llamada **gestionpeliculas** sin tablas.



Ahora abrimos el archivo llamado **settings.py** que se encuentra dentro de la Carpeta interna llamada **GestionPeliculas** y modificamos lo relacionado con lo de la base de datos así:



Si la base de datos va a ser en sqlite sería así:



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'gestionpeliculas.db',
    }
}
```

Y en el mismo archivo registramos la aplicación creada llamada **appPeliculas** en el bloque **INSTALLED_APPS** como se muestra en la imagen.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'appPeliculas',
]
```

Ahora se debe instalar **mysqlclient** en el entorno así:

```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Pro
yecto Django\GestionPeliculas>pip install mysqlclient ←
Collecting mysqlclient
  Downloading mysqlclient-2.2.4-cp312-cp312-win_amd64.whl.metadata (4.6 kB)
  Downloading mysqlclient-2.2.4-cp312-cp312-win_amd64.whl (203 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 203.3/203.3 kB 1.5 MB/s eta 0:00:00
Installing collected packages: mysqlclient
Successfully installed mysqlclient-2.2.4

(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Pro
yecto Django\GestionPeliculas>
```

El siguiente paso sería crear las **migraciones** para poder después crear las tablas en la base de datos. Para crear las migraciones ejecutar el siguiente comando:



```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django\GestionPelículas>python manage.py makemigrations
```

```
Migrations for 'appPelículas':
  appPelículas\migrations\0001_initial.py
    - Create model Genero
    - Create model Pelicula
```

```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django\GestionPelículas>
```

Con el comando anterior **django** creó una migración con el código 0001. Dentro de la carpeta de la aplicación hay una carpeta llamada **migrations** y ahí podemos ver el archivo generado **0001_initial.py** así:

```
GestionPelículas > appPelículas > migrations > 0001_initial.py > ...
1  # Generated by Django 5.0.4 on 2024-04-21 00:24
2  import django.db.models.deletion
3  from django.db import migrations, models
4
5  class Migration(migrations.Migration):
6      initial = True
7      dependencies = [
8      ]
9
10     operations = [
11         migrations.CreateModel(
12             name='Genero',
13             fields=[
14                 ('id', models.BigAutoField(auto_created=True,
15                     primary_key=True, serialize=False, verbose_name='ID')),
16                 ('genNombre', models.CharField(max_length=50, unique=True)),
17             ],
18         ),
19         migrations.CreateModel(
20             name='Pelicula',
21             fields=[
22                 ('id', models.BigAutoField(auto_created=True,
23                     primary_key=True, serialize=False, verbose_name='ID')),
24                 ('pelCodigo', models.CharField(max_length=9)),
25                 ('pelTitulo', models.CharField(max_length=50)),
26                 ('pelProtagonista', models.CharField(max_length=50)),
27                 ('pelDuracion', models.IntegerField()),
28                 ('pelResumen', models.CharField(max_length=2000)),
29                 ('pelFoto', models.ImageField(
30                     blank=True, null=True, upload_to='fotos/')),
31                 ('pelGenero', models.ForeignKey(
32                     on_delete=django.db.models.deletion.PROTECT, to='appPelículas.genero')),
33             ],
34         ),
35     ]
```



El archivo de la **migración** si contiene el campo de llave primaria que por defecto los llamada **id**. A medida que se vayan realizando cambios y se vuelva a ejecutar **makemigrations** se van creando archivos nuevos que incluyen los cambios.

Y para terminar de crear las tablas en la base de datos ejecutamos el siguiente comando:

```
python manage.py migrate
```

```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django\GestionPelículas>python manage.py migrate
System check identified some issues:

WARNINGS:
?: (mysql.W002) MariaDB Strict Mode is not set for database connection 'default'
   HINT: MariaDB's Strict Mode fixes many data integrity problems in MariaDB, such as data truncation upon insertion, by escalating warnings into errors. It is strongly recommended you activate it. See: https://docs.djangoproject.com/en/5.0/ref/databases/#mysql-sql-mode
Operations to perform:
  Apply all migrations: admin, appPelículas, auth, contenttypes, sessions
Running migrations:
  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 appPelículas.0001_initial... 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

(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Proyecto Django\GestionPelículas>
```

Si vamos a mysql podemos ver que se crearon las dos tablas y adicionalmente otras tablas que utiliza **django** para administrar la aplicación:



Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño
<input type="checkbox"/> apppeliculas_genero	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	32.0 KB
<input type="checkbox"/> apppeliculas_pelicula	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	32.0 KB
<input type="checkbox"/> auth_group	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	32.0 KB
<input type="checkbox"/> auth_group_permissions	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	48.0 KB
<input type="checkbox"/> auth_permission	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	32	InnoDB	utf8_spanish_ci	32.0 KB
<input type="checkbox"/> auth_user	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	32.0 KB
<input type="checkbox"/> auth_user_groups	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	48.0 KB
<input type="checkbox"/> auth_user_user_permissions	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	48.0 KB
<input type="checkbox"/> django_admin_log	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	48.0 KB
<input type="checkbox"/> django_content_type	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8_spanish_ci	48.0 KB
<input type="checkbox"/> django_migrations	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	19	InnoDB	utf8_spanish_ci	16.0 KB
<input type="checkbox"/> django_session	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	32.0 KB

gestionpeliculas apppeliculas_pelicula
id : bigint(20)
pelCodigo : varchar(9)
pelTitulo : varchar(50)
pelProtagonista : varchar(50)
pelDuracion : int(11)
pelResumen : varchar(2000)
pelFoto : varchar(100)
pelGenero_id : bigint(20)

gestionpeliculas apppeliculas_genero
id : bigint(20)
genNombre : varchar(50)

Ahora vamos a crear dentro de la carpeta **appPeliculas** una carpeta llamada *templates* para colocar aquí los archivos html que se vayan a crear.

La ruta de la carpeta se debe registrar en el archivo **settings.py** en el bloque **TEMPLATES** como se muestra en la siguiente imagen:



```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': ['./appPeliculas/templates'],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    },  
]
```

Configuración de los templates y rutas

Aquí se crean los htmls necesarios para la aplicación, así como los archivos javascript y hojas de estilo. Se recomienda también crear la carpeta **static** dentro de la aplicación **appPeliculas**, y dentro de ella crear las otras carpetas como **css**, **js**, **imagenes** entre otros.

Debemos configurar en el archivo **settings.py** las siguientes variables que me permiten indicar cuál es la carpeta donde se van a ubicar los archivos estáticos.

Importar la librería os

```
from pathlib import Path  
import os
```

Crear la variable **STATIC_URL**

```
STATIC_URL = os.path.join(BASE_DIR, '/static/')
```

Para empezar vamos a crear una función que me represente la vista inicial en el archivo **views.py** que se encuentra dentro de la carpeta **appPeliculas**.



```
GestionPeliculas > appPeliculas > views.py > ...
1  from django.shortcuts import render
2
3  # Create your views here.
4
5
6  def inicio(request):
7      return render(request, "inicio.html")
8
```

Siempre que se vaya a crear una vista como la anterior se crea una función que recibe como parámetro un **request**. En el ejemplo anterior la vista muestra el html del archivo inicio.html.

Ahora debemos registrar la vista en el archivo llamado **urls.py** así:

```
17  from django.contrib import admin
18  from django.urls import path
19  from appPeliculas import views
20
21  urlpatterns = [
22      path('admin/', admin.site.urls),
23      path('/', views.inicio)
24  ]
```

El archivo **urls.py** ya tiene una vista registrada para el admin y ahora agregamos la del inicio. La primera parte del path no necesariamente debe ser igual a la función, pero el segundo parámetro **views.inicio** si hace referencia al nombre de la función. Recordemos que en **Flask** la primera parte es como el decorador.

No olvidar hacemos el **import** señalado en la imagen.

Ahora podemos probar ejecutando el servidor para ver si está funcionando.

En la consola ejecutamos el siguiente comando verificando que estemos en la carpeta donde se encuentra el archivo **manage.py**



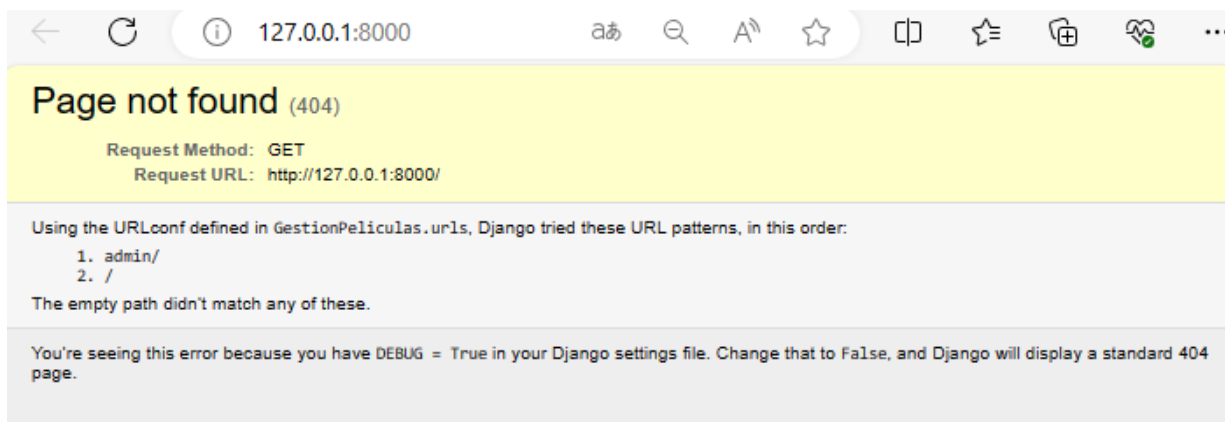
```
(entorno) D:\Documentos\SENA2024\FICHAS\ADSO 2669742\Actividades\Abril 22 de 2024\Primer Pro
yecto Django\GestionPeliculas>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
?: (urls.W002) Your URL pattern '/' has a route beginning with a '/'. Remove this slash as it
is unnecessary. If this pattern is targeted in an include(), ensure the include() pattern has
a trailing '/'.

System check identified 1 issue (0 silenced).
April 20, 2024 - 19:52:59
Django version 5.0.4, using settings 'GestionPeliculas.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

La respuesta nos dice que se ha lanzado el servidor web en el **puerto 8000**



El mensaje anterior sale porque no tenemos creada el archivo html llamado inicio.html.

Como en el proyecto se van a cargar y mostrar imágenes es necesario modificar el archivo **settings.py** donde se requiere indicar cual es la carpeta que almacena archivo multimedia.

Inicialmente hay que crear las siguientes variables en el archivo **settings.py**

```
# para almacenar los archivos multimedia
MEDIA_URL = '/media/'

# para recuperar los archivos multimedia
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```




Modificar el archivo **urls.py** para tener acceso a la carpeta donde se encuentran las imágenes.

```
from django.contrib import admin
from django.urls import path
from appPeliculas import views
from django.conf import settings
from django.conf.urls.static import static

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

if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL,
                           document_root=settings.MEDIA_ROOT)
```

Los siguientes pasos son los relacionados con el crud a la base de datos. Se anexa material de apoyo del ORM de Django.

Referencias:

1. Sitio oficial Framework Django: <https://www.djangoproject.com/>
2. Documentación oficial Django: <https://docs.djangoproject.com/es/4.1/>
3. Tutorial en la web: <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>
4. Qué es django: <https://aws.amazon.com/es/what-is/django/>
5. Tutorial Django: <https://www.w3schools.com/django/index.php>
6. Material Django <https://runebook.dev/es/docs/django/-index->

CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	César Marino Cuéllar Chacón	Instructor	CTPI-CAUCA	30-07-2025