

Python Django

Para la creación de un proyecto con Django se debe realizar lo siguiente:

Ingresar al símbolo del sistema y crear un nuevo directorio sobre el cual se generará un proyecto con Django.

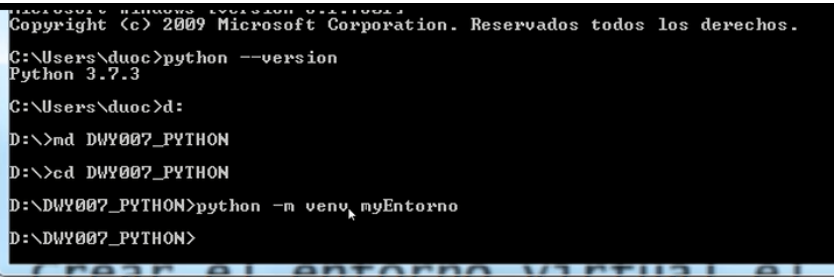


```
Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\duoc>python --version
Python 3.7.3

C:\Users\duoc>_
```

Para desarrollar el proyecto es recomendable crear un **entorno virtual** sobre el cual se instalarán y generarán los elementos necesarios para el desarrollo del proyecto.

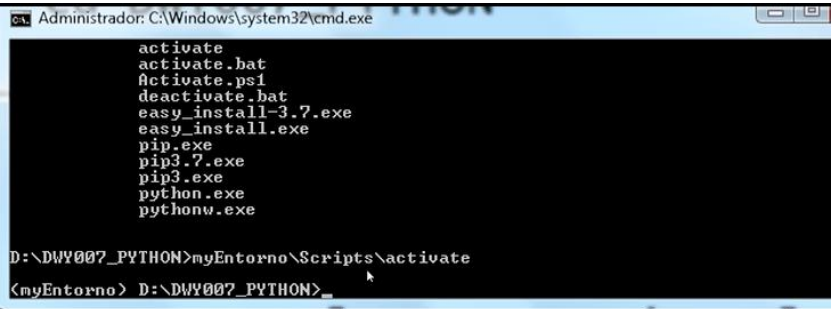


```
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\duoc>python --version
Python 3.7.3

C:\Users\duoc>d:
D:\>md DWY007_PYTHON
D:\>cd DWY007_PYTHON
D:\DWY007_PYTHON>python -m venv myEntorno
D:\DWY007_PYTHON>
```

Para ingresar se debe generar **myEntorno\Scripts\activate**

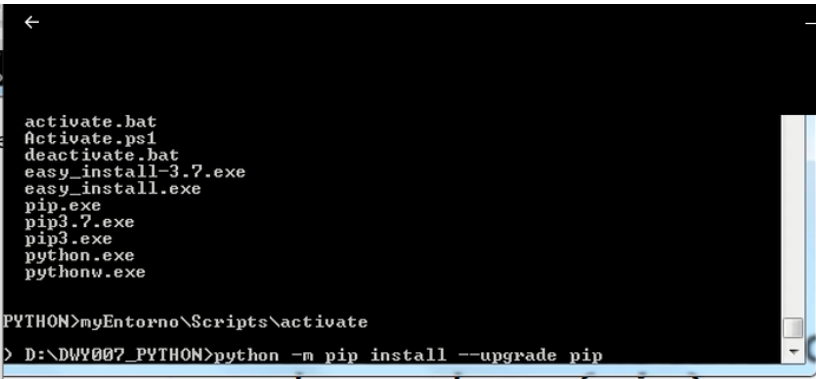


```
Administrador: C:\Windows\system32\cmd.exe

activate
activate.bat
Activate.ps1
deactivate.bat
easy_install-3.7.exe
easy_install.exe
pip.exe
pip3.7.exe
pip3.exe
python.exe
pythonw.exe

D:\DWY007_PYTHON>myEntorno\Scripts\activate
(myEntorno) D:\DWY007_PYTHON>_
```

Actualizaremos el **paquete de instalación de Python**

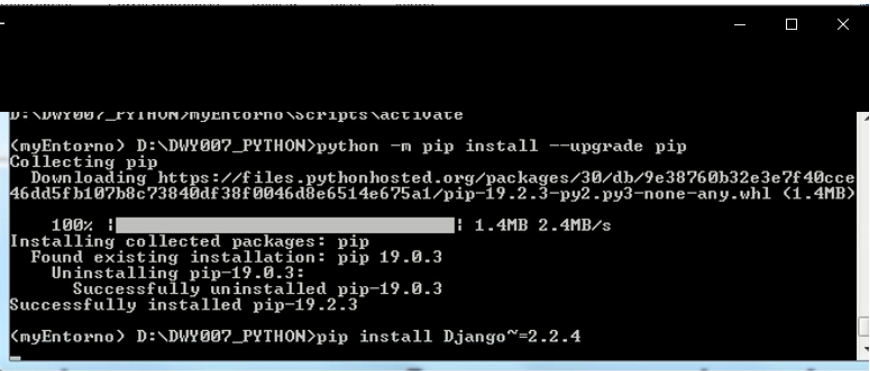


```
←

activate.bat
Activate.ps1
deactivate.bat
easy_install-3.7.exe
easy_install.exe
pip.exe
pip3.7.exe
pip3.exe
python.exe
pythonw.exe

PYTHON>myEntorno\Scripts\activate
> D:\DWY007_PYTHON>python -m pip install --upgrade pip
```

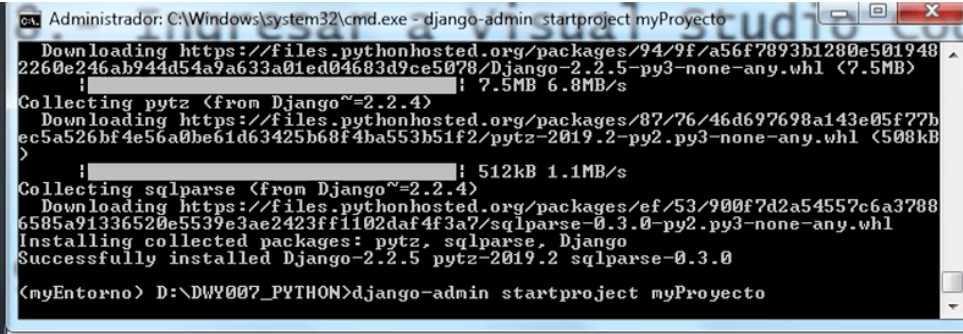
Descargamos e instalamos **Django**



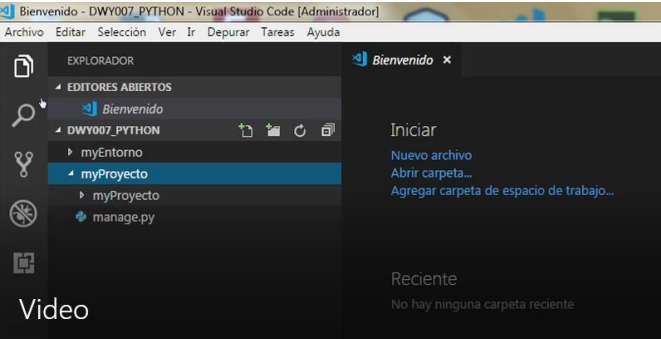
```
D:\DWY007_PYTHON>myEntorno\Scripts\activate
(myEntorno) D:\DWY007_PYTHON>python -m pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/30/db/9e38760b32e3e7f40cce46dd5fb107b8c73840df38f0046d8e6514e675a1/pip-19.2.3-py2.py3-none-any.whl (1.4MB)
    100% |#####| 1.4MB 2.4MB/s
Installing collected packages: pip
  Found existing installation: pip 19.0.3
  Uninstalling pip-19.0.3:
    Successfully uninstalled pip-19.0.3
  Successfully installed pip-19.2.3
(myEntorno) D:\DWY007_PYTHON>pip install Django~=2.2.4
```

Instalar en **Visual Studio Code** extensiones de Django y Django Templates

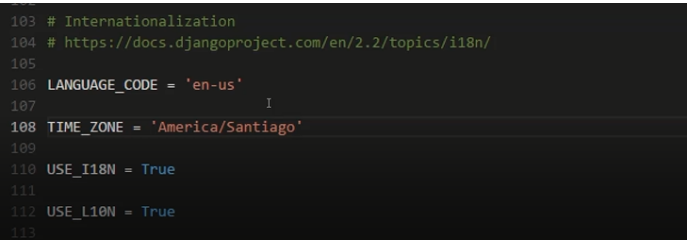
Creamos un proyecto Django (Solucion) **django-admin startproject myProyecto**



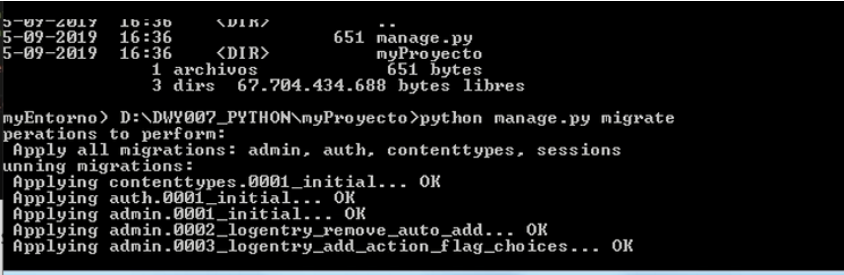
Abriremos el proyecto sobre Visual Studio Code



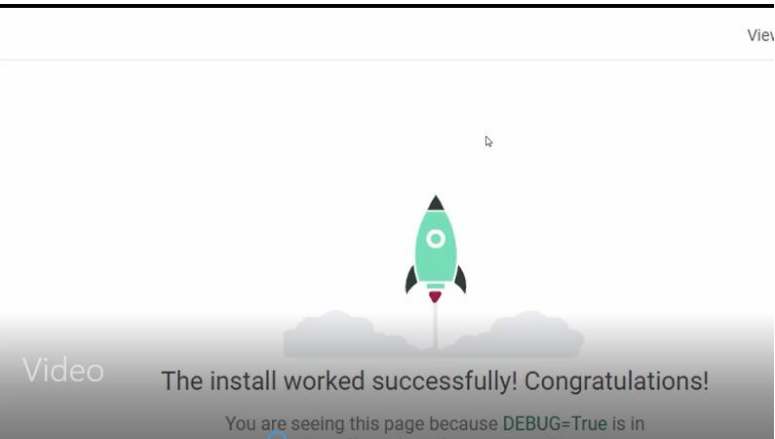
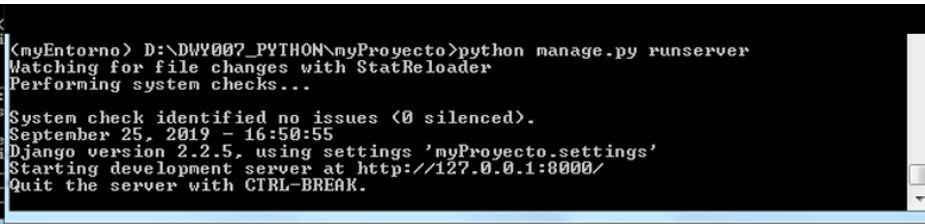
Modificar desde el archivo **setting.py** del proyecto en Python en **TIME_ZONE** por **America/Santiago**



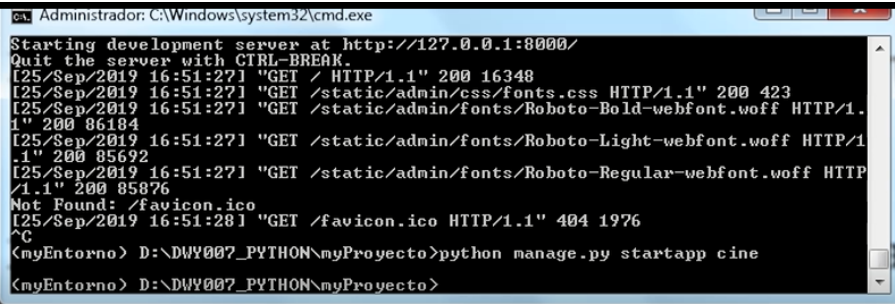
Crear los archivos de administración de base de datos **Python manage.py migrate**



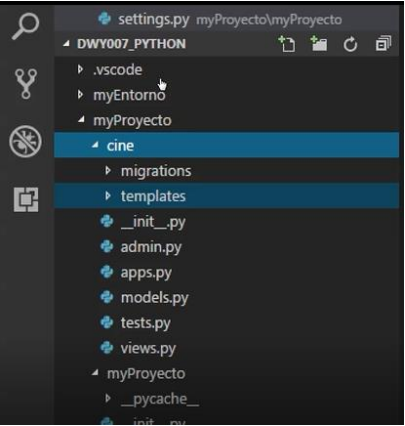
Podemos Iniciar el Servidor **Python manage.py runserver**



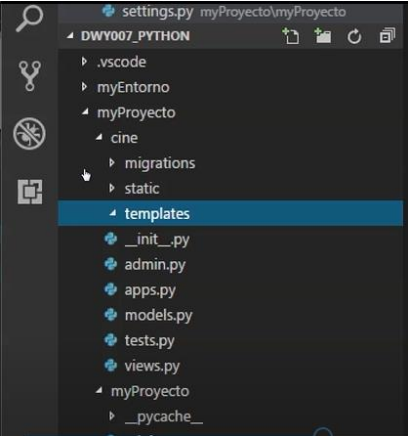
Crearemos una aplicación en Django **Python manage.py startapp myAplicacion**



Crearemos una carpeta con el nombre de **templates** en el interior de la aplicación en donde agregaremos las paginas web.



También debemos crear la carpeta **static** en el interior también de la aplicación

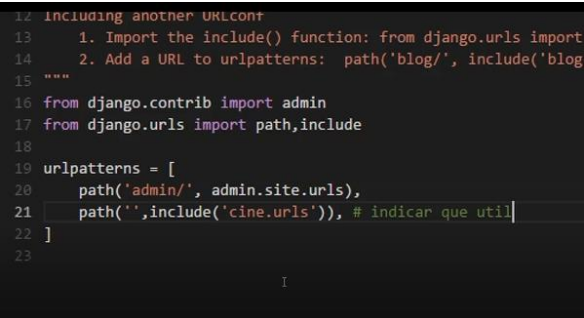


Generalmente en el interior de **templates** podemos crear la carpeta **core** (o centro de la aplicacion)

Incluir en el archivo de **setting.py** la aplicación creada.



En el archivo **urls.py** del proyecto debemos agregar la ruta hacia un archivo **urls.py** de la aplicacion, recuerde incluir **include** en los import y agregar al final el **path("",include('cine.urls'))**,



```

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 #agregue la libreria de 'include'
18
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('',include('cine.urls')), # indicar que utilice el archivo URL.py de Cine
22 ]

```

Luego debe crear un archivo **urls.py** en el interior de la aplicación. Copiar y pegar el contenido del archivo **urls.py** que posee el proyecto, esta será su base para este archivo.

```

2 from django.urls import path,include #agregue la libreria de 'include'
3
4 urlpatterns = [
5     path('', admin.site.urls),
6     path('',include('cine.urls')), # indicar que utilice el archivo URL.py de Cine
7 ]
8

```

Recuerde que deberá primero crear una **View** en donde renderizara una pagina web de las que posee en la carpeta **templates**

```

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

```

Se debe modificar el archivo el archivo **urls.py** de la aplicación.

```

settings.py  urls.py ...\myProyecto  urls.py ...\cine x
1 from django.contrib import admin
2 from django.urls import path,include
3 from .views import index
4
5 urlpatterns = [
6     path('', index,name='IND'),
7 ]
8

```

Puede volver a iniciar la aplicación.

Podemos crear un controlador para una galería

```

settings.py  urls.py ...\myProyecto  urls.py ...\cine  galeria.html  views.py
1 from django.shortcuts import render
2
3 # Create your views here.
4 def index(request):
5     return render(request,'core/index.html')
6
7 def gale(request):
8     return render(request,'core/galeria.html')

```

Creación de la url hacia la vista

```

settings.py  urls.py ...\myProyecto  urls.py ...\cine x
1 from django.contrib import admin
2 from django.urls import path,include
3 from .views import index, gale
4
5 urlpatterns = [
6     path('', index,name='IND'),
7     path('galeria/',gale,name='GAL'),
8 ]
9

```

Se puede realizar la unión con enlaces empleando `{% url 'ind' %}`

```

2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Document</title>
8 </head>
9 <body>
10     <h1>Galeria</h1>
11
12     <a href="{% url 'IND' %}"> ir a principal</a>
13 </body>
14 </html>

```

Para crear los enlaces hacia los archivo que estarán en **static** (css, javascript y img). Se debe agregar **{% load static %}** y luego se pueden llamar a los objetos **{% static 'img/romeo.jpg' %}**

```
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 <title>Document</title>
8 </head>
9 <body>
10     {% load static %}
11     <h1>Galeria</h1>
12     
13     <a href="{% url 'IND' %}"> ir a principal</a>
14 </body>
15 </html>
```

Podemos crear los **modelos** de datos (tablas) sobre la base de datos

```
1 from django.db import models
2
3 # Create your models here.
4 class Categoria(models.Model):
5     name=models.CharField(max_length=45,primary_key=True)
6     calificacion=models.IntegerField()
7
8     def __str__(self):
9         return self.name
10
11 class Pelicula(models.Model):
12     name=models.CharField(max_length=45,primary_key=True)
13     precio=models.IntegerField()
14     duracion=models.IntegerField()
15     descripcion=models.TextField()
16     categoria=models.ForeignKey(Categoria,on_delete=models.CASCADE)
17
18     def __str__(self):
```

```
1 from django.db import models
2
3 # Create your models here.
4 class Categoria(models.Model):
5     name=models.CharField(max_length=45,primary_key=True)
6     calificacion=models.IntegerField()
7
8     def __str__(self):
9         return self.name
10
11 class Pelicula(models.Model):
12     name=models.CharField(max_length=45,primary_key=True)
13     precio=models.IntegerField()
14     duracion=models.IntegerField()
15     descripcion=models.TextField()
16     categoria=models.ForeignKey(Categoria,on_delete=models.CASCADE)
17
18     def __str__(self):
19         return self.name
```

Para pedir a Python que cree las tablas solo debe escribir

- Python manage.py makemigrations cine
- Python manage.py migrate

```
<DIR>
<DIR>
<DIR>
131.072 db.sqlite3
651 manage.py
myProyecto
ivos 131.723 bytes
69.035.782.144 bytes libres
PYTHON\myProyecto>python manage.py makemigrations cine
```

```
2 archivos 131.723 bytes
4 dirs 69.035.782.144 bytes libres

myEntorno) D:\DWY008 PYTHON\myProyecto>python manage.py makemigrations for 'cine':
cine\migrations\0001_initial.py
- Create model Categoria
- Create model Pelicula

myEntorno) D:\DWY008 PYTHON\myProyecto>
```



```
D:\DWY008_PYTHON\myProyecto>python manage.py makemigrations cine
for 'cine':
migrations\0001_initial.py
e model Categoria
e model Pelicula

D:\DWY008_PYTHON\myProyecto>python manage.py migrate
to perform:
migrations: admin, auth, cine, contenttypes, sessions
ations:
cine.0001_initial... OK

D:\DWY008_PYTHON\myProyecto>
```

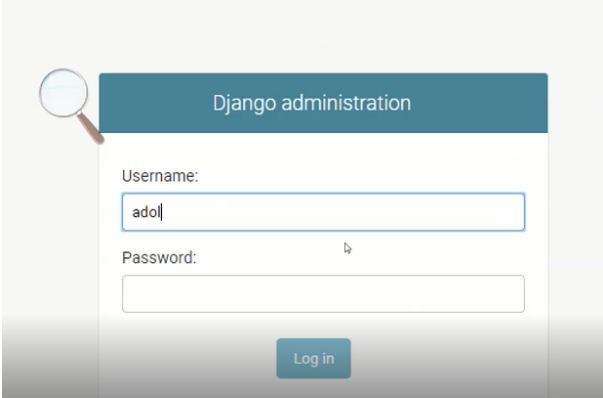
Crear un super usuario para acceder a la base de datos **Python manage.py createsuperuser**

```
2:14:29] "GET /static/admin/css/login.css HTTP/1.1" 200 1233
2:14:29] "GET /static/admin/css/responsive.css HTTP/1.1" 200 17944
2:14:43] "GET /admin/ HTTP/1.1" 302 0
2:14:43] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 1824

D:\DWY008_PYTHON\myProyecto>python manage.py createsuperuser
e blank to use 'duoc'):
```

```
(myEntorno) D:\DWY008_PYTHON\myProyecto>python manage.py createsuperuser
Username (leave blank to use 'duoc'): adolf
Email address: adolf@chicomalo.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
Bypass password validation and create user anyway? [y/N]: y
```

Puedes ingresar ingresando con <https://localhost/admin>



para observar estos modelos debemos agregarlos en el archivo **admin.py**

```
1 from django.contrib import admin
2 from .models import Categoria,Pelicula
3 # Register your models here.
4 admin.site.register(Categoria)
5
```