

Primera parte

Proyecto: **Cinépolis**.

lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

1 Procedimiento para crear carpeta del Proyecto: **UIII_Cinepolis_0716**

2 procedimiento para abrir vs code sobre la carpeta **UIII_Cinepolis_0716**

3 procedimiento para **abrir terminal en vs code**

4 Procedimiento para crear carpeta entorno virtual **".venv"** desde terminal de vs code

5 Procedimiento para **activar el entorno virtual**.

6 procedimiento para **activar intérprete de python**.

7 Procedimiento para instalar **Django**

8 procedimiento para crear proyecto **backend_Cinepolis** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el **puerto 8716**

10 procedimiento para copiar y pegar el link en el navegador.

11 procedimiento para crear aplicacion **app_Cinepolis**

12 Aqui el modelo [models.py](#)

```
-----
```

```
from django.db import models
```

```
# =====
```

```
# MODELO: Sucursal
```

```
# =====
```

```
class Sucursal(models.Model):
```

```
    ESTADOS = [
```

```

        ('activo', 'Activo'),

        ('cerrado', 'Cerrado'),

        ('mantenimiento', 'Mantenimiento'),

    ]

    FORMATOS = [

        ('tradicional', 'Tradicional'),

        ('3D', '3D'),

        ('4DX', '4DX'),

        ('VIP', 'VIP'),

        ('IMAX', 'IMAX'),

        ('JUNIOR', 'Junior'),

    ]

    id_sucursal = models.AutoField(primary_key=True)

    nombre_cine = models.CharField(max_length=100)

    direccion = models.CharField(max_length=200)

    ciudad = models.CharField(max_length=100)

    telefono = models.CharField(max_length=20)

    numero_salas = models.IntegerField()

    estado = models.CharField(max_length=20, choices=ESTADOS, default='activo')

    formatos = models.CharField(max_length=20, choices=FORMATOS,
    default='tradicional')

    def __str__(self):

        return f"{self.nombre_cine} - {self.ciudad}"

# =====

```

```
# MODELO: Sala
```

```
# =====
```

```
class Sala(models.Model):
```

```
    TIPOS_SALA = [
```

```
        ('Tradicional', 'Tradicional'),
```

```
        ('3D', '3D'),
```

```
        ('4DX', '4DX'),
```

```
        ('VIP', 'VIP'),
```

```
        ('IMAX', 'IMAX'),
```

```
        ('JUNIOR', 'Junior'),
```

```
    ]
```

```
    ESTADOS_SALA = [
```

```
        ('Ocupada', 'Ocupada'),
```

```
        ('Desocupada', 'Desocupada'),
```

```
        ('En mantenimiento', 'En mantenimiento'),
```

```
    ]
```

```
    id_sala = models.AutoField(primary_key=True)
```

```
    numero_sala = models.IntegerField()
```

```
    tipo_sala = models.CharField(max_length=20, choices=TIPOS_SALA,  
default='Tradicional')
```

```
    capacidad = models.IntegerField()
```

```
    estado = models.CharField(max_length=20, choices=ESTADOS_SALA,  
default='Desocupada')
```

```
    fecha_ultimo_mantenimiento = models.DateField()
```

```
    asientos_especiales = models.IntegerField()
```

```

# Relación 1 a muchos: una sucursal puede tener muchas salas

sucursal = models.ForeignKey(Sucursal, on_delete=models.CASCADE,
related_name='salas')


def __str__(self):

    return f"Sala {self.numero_sala} - {self.tipo_sala} ({self.sucursal.nombre_cine})"

# =====

# MODELO: Pelicula

# =====

class Pelicula(models.Model):

    id_pelicula = models.AutoField(primary_key=True)

    titulo = models.CharField(max_length=200)

    genero = models.CharField(max_length=100)

    clasificacion = models.CharField(max_length=100)

    duracion = models.IntegerField(help_text="Duración en minutos")

    sinopsis = models.TextField()

    director = models.CharField(max_length=150)


# Relación muchos a muchos: una sala puede tener varias películas y viceversa

salas = models.ManyToManyField(Sala, related_name='peliculas')


def __str__(self):

    return self.titulo

==--==--==

```

12.5 Procedimiento para realizar las migraciones (makemigrations y migrate).

13 primero trabajamos con el MODELO: SUCURSAL

14 En view de **app_Cinepolis** crear las funciones con sus códigos correspondientes (inicio_cinepolis, agregar_sucursal, actualizar_sucursal, realizar_actualizacion_sucursal, borrar_sucursal)

15 Crear la carpeta "**templates**" dentro de "**app_Cinepolis**".

16 En la carpeta templates crear los archivos html (**base.html**, **header.html**, **navbar.html**, **footer.html**, **inicio.html**).

17 En el archivo base.html **agregar bootstrap** para css y js.

18 En el archivo navbar.html incluir las opciones ("Sistema de Administración Cinepolis", "Inicio", "sucursal", en submenu de categorias (Aregar sucursal, ver sucursal, actualizar sucursal, borrar sucursal), "Salas" en submenu de Salas (Aregar salas, ver salas, actualizar salas, borrar salas) "Peliculas" en submenu de Peliculas (Aregar peliculas, ver peliculas, actualizar peliculas, borrar peliculas), incluir iconos a las opciones principales, no en los submenu.

19 En el archivo **footer.html** incluir derechos de autor, fecha del sistema y "Creado por Jazmin Valtierra, Cbtis 128" y mantenerla fija al final de la página.

20 En el archivo inicio.html se usa para colocar información del sistema más una imagen tomada desde la red sobre cinepolis.

21 Crear la subcarpeta carpeta **sucursal** dentro de **app_Cinepolis\templates**.

22 crear los archivos html con su código correspondientes de (agregar_sucursal.html, ver_sucursales.html mostrar en tabla con los botones ver, editar y borrar, actualizar_sucursal.html, borrar_sucursal.html) dentro de **app_Cinepolis\templates\sucursal**.

23 No utilizar [forms.py](#).

24 procedimiento para crear el archivo [urls.py](#) en **app_Cinepolis** con el código correspondiente para acceder a las funciones de [views.py](#) para operaciones de crud en sucursales.

25 procedimiento para agregar **app_Cinepolis** en [settings.py](#) de **backend_Cinepolis**

26 realizar las configuraciones correspondiente a [urls.py](#) de **backend_Cinepolis** para enlazar **con app_Cinepolis**

27 procedimiento para registrar los modelos en [admin.py](#) y volver a realizar las migraciones.

27 por lo pronto solo trabajar con "sucursal" dejar pendiente # MODELO: SALA y # MODELO: PELÍCULA

28 Utilizar colores suaves, atractivos y modernos, el código de las páginas web sencillas.

28 No validar entrada de datos.

29 Al inicio **crear la estructura completa** de carpetas y archivos.

30 proyecto totalmente funcional.

31 finalmente ejecutar servidor en el puerto **puerto 8716**.