

Primera parte

Proyecto: **Cinépolis**.

lenguaje: **Python**.

Framework: **Django**.

Editor: VS code.

1 Procedimiento para crear carpeta del Proyecto: UIII_Cinepolis_0777

2 procedimiento para abrir vs code sobre la carpeta

UIII_Cinepolis_0777

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 8036

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: CATEGORÍA
```

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

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

```
    nombre = models.CharField(max_length=50, unique=True)
```

```
    descripcion = models.TextField(blank=True, null=True)
```

```
    edad_recomendada = models.CharField(max_length=10)
```

```
    fecha_creacion = models.DateField(auto_now_add=True)
```

```
    activo = models.BooleanField(default=True)
```

```
    popularidad = models.PositiveIntegerField(default=0)
```

```
    icono = models.CharField(max_length=100, blank=True, null=True)
```

```
    def __str__(self):
```

```
        return self.nombre
```

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

```
# MODELO: SALA
```

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

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

```
    nombre = models.CharField(max_length=50)
```

```
    numero_sala = models.PositiveIntegerField(unique=True)
```

```
    capacidad = models.PositiveIntegerField()
```

```
    tipo_pantalla = models.CharField(max_length=30, choices=[
```

```
        ('2D', '2D'),
```

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

```

('IMAX', 'IMAX')
])
sonido = models.CharField(max_length=30, default='Dolby Atmos')
disponible = models.BooleanField(default=True)
ubicacion = models.CharField(max_length=100)
def __str__(self):
    return f"Sala {self.numero_sala} - {self.tipo_pantalla}"

# =====
# MODELO: PELÍCULA
# =====
class Pelicula(models.Model):
    titulo = models.CharField(max_length=150)
    descripcion = models.TextField(blank=True, null=True)
    duracion = models.PositiveIntegerField(help_text="Duración en minutos")
    clasificacion = models.CharField(max_length=10)
    idioma = models.CharField(max_length=50)
    fecha_estreno = models.DateField()
    sala = models.ForeignKey(Sala, on_delete=models.CASCADE, related_name="peliculas")
    # Relación 1 a muchos: una sala puede tener varias películas proyectadas
    categorias = models.ManyToManyField(Categoria, related_name="peliculas")
    # Relación muchos a muchos: una película puede pertenecer a varias categorías
    def __str__(self):
        return self.titulo
=====

```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: CATEGORÍA

14 En view de app_Cinepolis crear las funciones con sus códigos correspondientes (inicio_cinepolis, agregar_categoria,

actualizar_categoria, realizar_actualizacion_categoria, borrar_categoria)

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”, “categoria”, en submenu de categorias(Aregar Categoria,ver categoria, actualizar categoria, borrar categoria), “Salas” en submenu de Salas(Aregar salas,ver salas, actualizar salas, borrar salas)

“Películas” 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 Ing. Eliseo Nava, 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 cinopolis.
- 21 Crear la subcarpeta carpeta categoria dentro de app_Cinopolis\templates.
- 22 crear los archivos html con su código correspondientes de (agregar_categoria.html, ver_categorias.html mostrar en tabla con los botones ver, editar y borrar, actualizar_categoria.html, borrar_categoria.html) dentro de app_Cinopolis\templates\categoria.
- 23 No utilizar forms.py.
- 24 procedimiento para crear el archivo urls.py en app_Cinopolis con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en categorias.
- 25 procedimiento para agregar app_Cinopolis en settings.py de backend_Cinopolis
- 26 realizar las configuraciones correspondiente a urls.py de backend_Cinopolis para enlazar con app_Cinopolis
- 27 procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.
- 27 por lo pronto solo trabajar con “categoría” 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 8036.