

Primera parte

Proyecto: **YouTube**.

lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

1 Procedimiento para crear carpeta del Proyecto: **Ulll_YouTube_0517**

2 procedimiento para abrir vs code sobre la carpeta

Ulll_YouTube_0517

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_YouTube** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el **puerto 8157**

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

11 procedimiento para crear aplicacion **app_YouTube**

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

==--==--==--

```
from django.db import models
from django.contrib.postgres.fields import JSONField
# Para usar JSONB en PostgreSQL
```

```
class Canal(models.Model):
    id = models.AutoField(primary_key=True)
    nombre = models.CharField(max_length=100)
    descripcion = models.TextField()
    fecha_creacion = models.DateField()
    suscriptores = models.IntegerField(default=0)
    url_personalizada = models.CharField(max_length=100, unique=True)
    pais = models.CharField(max_length=50)
    categoria_principal = models.CharField(max_length=50)
```

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

```
return self.nombre
```

```
class Video(models.Model):
    id = models.AutoField(primary_key=True)
    canal = models.ForeignKey(Canal, on_delete=models.CASCADE, related_name='videos')
    titulo = models.CharField(max_length=200)
    descripcion = models.TextField()
    duracion = models.DurationField() # Equivalente a INTERVAL
    vistas = models.IntegerField(default=0)
    publico = models.BooleanField(default=True)
    url_miniatra = models.CharField(max_length=200)
    likes = models.CharField(max_length=200)

    def __str__(self):
        return f"{self.titulo} ({self.canal.nombre})"
```

```
class Playlist(models.Model):
    id = models.AutoField(primary_key=True)
    nombre = models.CharField(max_length=150)
    descripcion = models.TextField()
    fecha_creacion = models.DateField()
    publica = models.BooleanField(default=True)
    canal_creador = models.ForeignKey(Canal, on_delete=models.CASCADE,
related_name='playlists')
    videos = models.ManyToManyField(Video, related_name='playlists') # 👉 Relación
muchos a muchos
    orden_videos = JSONField() # Guarda el orden de los videos o metadatos extra
    n_videos = models.DecimalField(max_digits=10, decimal_places=0)

    def __str__(self):
        return f"{self.nombre} - {self.canal_creador.nombre}"
```

==--==--==--

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO:CANAL

14 En view de **app_YouTube** crear las funciones con sus códigos correspondientes (inicio_YouTube, agregar_canal, actualizar_canal, realizar_actualizacion_canal,

borrar_canal)

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

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 YouTube", "Inicio", "Canal", en submenú de Canal (Agregar Canal, Ver Canal, Actualizar Canal, Borrar Canal), "Video" en submenú de Video (Agregar Video, Ver Video, Actualizar Video, Borrar Video), "Playlist" en submenú de Playlist (Agregar Playlist, Ver Playlist, Actualizar Playlist, Borrar Playlist), incluir íconos a las opciones principales, no en los submenús.

19 En el archivo **footer.html** incluir derechos de autor, fecha del sistema y "Creado por Sr.Alexander Cancino, 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 YouTube.

21 Crear la subcarpeta carpeta **canal** dentro de **app_YouTube\templates**.

22 crear los archivos html con su código correspondientes de (agregar_canal.html, ver_canal.html mostrar en tabla con los botones ver, editar y borrar, actualizar_canal.html, borrar_canal.html)

dentro de **app_YouTube\templates\canal**.

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

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

25 procedimiento para agregar **app_YouTube** en [settings.py](#) de **backend_YouTube**

- 26 realizar las configuraciones correspondiente a [urls.py](#) de **backend_YouTube** para enlazar con **app_YouTube**
- 27 procedimiento para registrar los modelos en [admin.py](#) y volver a realizar las migraciones.
- 27 por lo pronto solo trabajar con “canal” dejar pendiente #
MODELO: VIDEO y # MODELO: PLAYLIST
- 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 8517**.