

## Primera parte

Proyecto: **Crunchyroll**.

lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

1 Procedimiento para crear carpeta del Proyecto: **Ulll\_Crunchyroll\_0659**

2 procedimiento para abrir vs code sobre la carpeta

**Ulll\_Crunchyroll\_0659**

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\_Crunchyroll** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el **puerto 8032** 10

procedimiento para copiar y pegar el link en el navegador. 11

procedimiento para crear aplicacion **app\_Crunchyroll** 12 Aquí

el modelo [models.py](#)

**==--==--==--**

```
from django.db import models
from django.core.validators import MinValueValidator, MaxValueValidator
```

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

```
# MODELO: SUSCRIPCIÓN
```

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

```
class Suscripcion(models.Model):
    nombre_plan = models.CharField(max_length=50, unique=True)
    precio = models.DecimalField(max_digits=5, decimal_places=2)
    calidad_video = models.CharField(max_length=20)
    num_dispositivos = models.IntegerField()
    beneficio_extra = models.TextField()
    descarga_offline = models.BooleanField(default=False)
```

```
# Relación muchos a muchos: una suscripción puede tener varios contenidos exclusivos
contenidos_exclusivos = models.ManyToManyField(
    'Contenido',
```

```

        related_name='suscripciones_exclusivas',
        blank=True
    )
    # Relación muchos a muchos: una suscripción puede tener varios usuarios premium
    usuarios_premium = models.ManyToManyField(
        'Usuario',
        related_name='suscripciones_adicionales',
        blank=True
    )

    class Meta:
        db_table = 'suscripcion'

    def __str__(self):
        return f"{self.nombre_plan} - ${self.precio}"

# =====
# MODELO: CONTENIDO
# =====
class Contenido(models.Model):
    TIPO_CHOICES = [
        ('pelicula', 'Película'),
        ('serie', 'Serie'),
        ('documental', 'Documental'),
    ]

    titulo = models.CharField(max_length=100, unique=True)
    tipo = models.CharField(max_length=20, choices=TIPO_CHOICES)
    fecha_lanzamiento = models.DateField()
    categoria = models.IntegerField()
    capitulos = models.CharField(max_length=5)

    # Relación 1 a muchos: una suscripción puede tener varios contenidos principales
    suscripcion = models.ForeignKey(
        Suscripcion,
        on_delete=models.CASCADE,
        related_name='contenidos_principales'
    )

    class Meta:
        db_table = 'contenido'

    def __str__(self):

```

```

        return self.titulo

# =====
# MODELO: USUARIO
# =====
class Usuario(models.Model):
    nombre_usuario = models.CharField(max_length=50, unique=True)
    email = models.EmailField(max_length=254, unique=True)
    password = models.CharField(max_length=128)
    fecha_registro = models.DateField(auto_now_add=True)
    activo = models.BooleanField(default=True)

    # Relación 1 a muchos: una suscripción puede tener varios usuarios principales
    suscripcion = models.ForeignKey(
        Suscripcion,
        on_delete=models.CASCADE,
        related_name='usuarios_principales'
    )

class Meta:
    db_table = 'usuario'

    def __str__(self):
        return self.nombre_usuario

```

=====

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: Suscripcion 14 En view de **app\_Crunchyroll** 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\_Crunchyroll**”. 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 Crunchyroll”, “Inicio”, “Suscripción”, en submenú de categorías (Agregar Suscripción, ver Suscripción, actualizar Suscripción, borrar Suscripción), “Usuarios” en submenú de Usuarios (Agregar Usuarios, ver Usuarios, actualizar Usuarios, borrar Usuarios)

“Contenido” en submenú de Contenido (Agregar Contenido, ver Contenido, actualizar Contenido, borrar Contenido), incluir iconos a las opciones principales, no en los submenús.

19 En el archivo **footer.html** incluir derechos de autor, fecha del sistema y “Creado por Victor Muñiz, 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 **Suscripciones** dentro de **app\_Crunchyroll\templates**.

22 crear los archivos html con su código correspondientes de (agregar\_Suscripciones.html, ver\_Suscripciones.html mostrar en tabla con los botones ver, editar y borrar, actualizar\_Suscripciones.html, Suscripciones.html) dentro de **app\_Crunchyroll\templates\suscripciones**.

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

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

25 procedimiento para agregar **app\_Crunchyroll** en [settings.py](#) de **backend\_Crunchyroll**

**26** realizar las configuraciones correspondiente a [urls.py](#) de **backend\_Crunchyroll** para enlazar con **app\_Crunchyroll**

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

27 por lo pronto solo trabajar con “Suscripciones” dejar pendiente # MODELO: Usuarios y # MODELO: Contenido

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 8032**.