

Primera parte

Proyecto: **Webtoon**.

Lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

1 Procedimiento para crear carpeta del Proyecto: **UIII_Webtoon_0456**

2 procedimiento para abrir vs code sobre la carpeta

UIII_Webtoon_0456

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_Webtoon** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el **puerto 8001**

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

11 procedimiento para crear aplicación **app_Webtoon**

12 Aquí el modelo models.py

=====

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

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

```
# MODELO: USUARIO
```

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

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

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

```
    email = models.EmailField(unique=True)
```

```
    contraseña = models.CharField(max_length=128)
```

```
    fecha_registro = models.DateTimeField(auto_now_add=True)
```

```
    avatar = models.CharField(max_length=255, blank=True, null=True)
```

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

```
    tipo_usuario = models.CharField(max_length=20, choices=[
```

```
        ('Lector', 'Lector'),
```

```
        ('Autor', 'Autor'),
```

```
        ('Admin', 'Admin')
```

```
    ], default='Lector')
```

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

```
    return self.nombre_usuario
```

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

```
# MODELO: WEBTOON
```

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

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

```
    titulo = models.CharField(max_length=100)
```

```
    descripcion = models.TextField()
```

```

genero = models.CharField(max_length=50)
fecha_publicacion = models.DateField()
portada = models.CharField(max_length=255, blank=True, null=True)
estado = models.CharField(max_length=20, choices=[
    ('En curso', 'En curso'),
    ('Finalizado', 'Finalizado'),
    ('Hiato', 'Hiato')
], default='En curso')
autor = models.ForeignKey(Usuario, on_delete=models.CASCADE,
related_name='webtoons')

def __str__(self):
    return self.titulo

# =====
# MODELO: MEMBRESIA
# =====

class Membresia(models.Model):
    nombre = models.CharField(max_length=50)
    descripcion = models.TextField()
    precio = models.DecimalField(max_digits=6, decimal_places=2)
    duracion_dias = models.IntegerField()
    beneficios = models.TextField()
    nivel = models.CharField(max_length=20, choices=[
        ('Basico', 'Basico'),
        ('Premium', 'Premium'),
        ('VIP', 'VIP')
    ], default='Basico')
    usuarios_asociados = models.TextField(
        help_text="Lista de IDs de usuarios asociados, separada por comas (simulación de
relación muchos a muchos)"
    )

    def __str__(self):
        return self.nombre
=====

12.5 Procedimiento para realizar las migraciones(makemigrations y
migrate.
13 primero trabajamos con el MODELO: USUARIO
14 En view de app_Webtoon crear las funciones con sus códigos
correspondientes (inicio_webtoon, agregar_usuario,
actualizar_usuario, realizar_actualizacion_usuario,
borrar_usuario)
15 Crear la carpeta “templates” dentro de “app_Webtoon”.
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 Webtoon”, “Inicio”, “Usuario”, en submenu de usuarios(Aregar Usuario,ver usuario, actualizar usuario, borrar usuario), “Webtoons” en submenu de Webtoons(Aregar webtoons,ver webtoons, actualizarwebtoons, borrar webtoons) “Membresias” en submenu de Membresias(Aregar membresia,vermembresia, actualizar membresia, borrar membresia), 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 Tec. Ingrid Alvarado, 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 webtoon.

21 Crear la subcarpeta carpeta **usuario** dentro de **app_Webtoon\templates**.

22 crear los archivos html con su codigo correspondientes de (agregar_usuario.html, ver_usuario.html mostrar en tabla con los botones ver, editar y borrar, actualizar_usuario.html, borrar_usuario.html)

dentro de **app_Webtoon\templates\usuario**.

23 No utilizar forms.py.

24 procedimiento para crear el archivo urls.py en **app_Webtoon** con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en usuarios.

25 procedimiento para agregar **app_Webtoon** en settings.py de **backend_Webtoon**

26 realizar las configuraciones correspondiente a urls.py de **backend_Webtoon** para enlazar **con app_Webtoon**

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: WEBTOON y # MODELO: MEMBRESIA

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 8001**.