

Primera parte

Proyecto: **Escuela_de_danza**

Lenguaje: **Python**.

Framework: **Django**.

Editor: VS code.

1 Procedimiento para crear carpeta del Proyecto: **UIII_escueladanza_0310**

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_escueladanza** 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 aplicación **app_escueladanza**

12 Aquí el modelo models.py

=====

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

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

```
# MODELO: PROFESOR
```

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

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

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

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

```
    telefono = models.CharField(max_length=15)
```

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

```
    especialidad = models.CharField(max_length=50, help_text="Tipo de danza que enseña")
```

```
    fecha_contratacion = models.DateField()
```

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

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

```
    return f"{self.nombre} {self.apellido}"
```

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

```
# MODELO: ALUMNO
```

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

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

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

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

```
    fecha_nacimiento = models.DateField()
```

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

```

telefono = models.CharField(max_length=15)
nivel = models.CharField(max_length=30, choices=[
    ('Principiante', 'Principiante'),
    ('Intermedio', 'Intermedio'),
    ('Avanzado', 'Avanzado')
])
fecha_incripcion = models.DateField()

def __str__(self):
    return f'{self.nombre} {self.apellido}'


# =====
# MODELO: CLASE
# =====
class Clase(models.Model):
    nombre = models.CharField(max_length=100)
    descripcion = models.TextField(blank=True, null=True)
    duracion = models.DurationField(help_text="Duración de la clase (hh:mm:ss)")
    horario = models.CharField(max_length=50)
    cupo_maximo = models.PositiveIntegerField()
    profesor = models.ForeignKey(Profesor, on_delete=models.CASCADE,
                                 related_name="clases")
    # Relación 1 a muchos: un profesor puede impartir varias clases
    alumnos = models.ManyToManyField(Alumno, related_name="clases")
    # Relación muchos a muchos: una clase puede tener varios alumnos y un alumno puede
    estar en varias clases

    def __str__(self):
        return self.nombre

```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el **MODELO: PROFESOR**

14 En view de app_escueladanza crear las funciones con sus códigos correspondientes (inicio_danza, agregar_profesor, actualizar_profesor, realizar_actualizacion_profesor, borrar_profesor)

15 Crear la carpeta “templates” dentro de “app_escueladanza”.

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 Escuela de Danza”, “Inicio”, “categoria”,en submenu de categorias(Aregar Categoria,ver categoria, actualizar categoria, borrar categoria),“**Profesores**” con submenu: (Aregar Profesor, Ver Profesores, Actualizar Profesor, Borrar Profesor)“**Alumnos**” con submenu: (Aregar Alumno, Ver Alumnos, Actualizar Alumno, Borrar Alumno)“**Clases**” con submenu: (Aregar Clase, Ver Clases, Actualizar Clase, Borrar Clase) 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. Alan Pacheco, 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 escuela de danza
- 21 Crear la subcarpeta carpeta profesor dentro de app_EscuelaDanza\templates.
- 22 crear los archivos html con su código correspondientes de (agregar_profesor.html, ver_profesores.html mostrar en tabla con los botones ver, editar y borrar, actualizar_profesor.html, borrar_profesor.html) dentro de app_EscuelaDanza\templates\profesor
- 23 No utilizar forms.py.
- 24 procedimiento para crear el archivo urls.py en app_EscuelaDanza con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en profesores
- 25 procedimiento para agregar app_EscuelaDanza en settings.py de backend_EscuelaDanza
- 26 realizar las configuraciones correspondiente a urls.py de backend_EscuelaDanza para enlazar con app_EscuelaDanza
- 27 procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.
- 27 por lo pronto solo trabajar con “PROFESOR” dejar pendiente # MODELO: CLASE y # MODELO:ALUMNO
- 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.