

## Primera parte

Proyecto: **Taller de Arte.**

lenguaje: **Python.**

Framework: **Django.**

Editor: **VS code.**

1 Procedimiento para crear carpeta del Proyecto: **UIII\_TallerdeArte\_0272**

2 procedimiento para abrir vs code sobre la carpeta

**UIII\_TallerdeArte\_0272**

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\_TallerdeArte** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el **puerto 8272**

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

11 procedimiento para crear aplicacion **app\_TallerdeArte**

12 Aqui el modelo models.py

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

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

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

```
# MODELO: Instructor
```

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

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

```
    instructor_id = models.AutoField(primary_key=True)
```

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

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

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

```
    telefono = models.CharField(max_length=20)
```

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

```
    experiencia_anios = models.PositiveIntegerField()
```

```
    sueldo = models.DecimalField(max_digits=8, decimal_places=2)
```

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

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

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

```
# MODELO: Material
```

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

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

```
    material_id = models.AutoField(primary_key=True)
```

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

```

tipo = models.CharField(max_length=50)
marca = models.CharField(max_length=50)
cantidad_stock = models.PositiveIntegerField()
costo_unitario = models.DecimalField(max_digits=8, decimal_places=2)
nombre_proveedor = models.CharField(max_length=100)
descripcion = models.TextField()

# Relación 1:M (un curso puede tener muchos materiales)
curso = models.ForeignKey(Curso, on_delete=models.CASCADE,
related_name='materiales')

def __str__(self):
    return self.nombre_material

# =====
# MODELO: CURSO
# =====
class Curso(models.Model):
    curso_id = models.AutoField(primary_key=True)
    nombre_curso = models.CharField(max_length=100)
    descripcion = models.TextField()
    salon_num = models.IntegerField()
    duracion_semanas = models.PositiveIntegerField()
    costo = models.DecimalField(max_digits=8, decimal_places=2)
    fecha_inicio = models.DateField()
    cupo_maximo = models.PositiveIntegerField()

    # Relación 1:M (un instructor puede tener muchos cursos)
    instructor = models.ForeignKey(Instructor, on_delete=models.CASCADE,
related_name='cursos')

    def __str__(self):
        return self.nombre_curso

```

## 12.5 Procedimiento para realizar las migraciones(makemigrations y migrate).

13 primero trabajamos con el MODELO: INSTRUCTOR

14 En view de **app\_TallerdeArte** crear las funciones con sus códigos correspondientes (inicio\_TallerdeArte, agregar\_instructor,

actualizar\_instructor, realizar\_actualizacion\_instructor, borrar\_instructor)

15 Crear la carpeta “**templates**” dentro de “**app\_TallerdeArte**”.

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 TallerdeArte”, “Inicio”, “instructor”, en submenu de instructor(Agregar instructor, ver instructor, actualizar instructor, borrar instructor), “Curso” en submenu de curso(Agregar curso, ver curso, actualizar curso, borrar curso)  
“Material” en submenu de material(Agregar material, ver material, actualizar material, borrar material), 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 Victor Martinez, 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 taller de arte.
- 21 Crear la subcarpeta carpeta **instructor** dentro de **app\_TallerdeArte\templates**.
- 22 crear los archivos html con su código correspondientes de (agregar\_instructor.html, ver\_instructor.html mostrar en tabla con los botones ver, editar y borrar, actualizar\_instructor.html, borrar\_instructor.html)  
dentro de **app\_TallerdeArte\templates\instructor**.
- 23 No utilizar forms.py.
- 24 procedimiento para crear el archivo urls.py en app\_TallerdeArte con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en instructor.
- 25 procedimiento para agregar app\_TallerdeArte en settings.py de **backend\_TallerdeArte**
- 26 realizar las configuraciones correspondiente a urls.py de **backend\_TallerdeArte** para enlazar con **app\_TallerdeArte**
- 27 **procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.**
- 27 por lo pronto solo trabajar con “instructor” dejar pendiente # MODELO: CURSO y # MODELO: MATERIAL
- 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 8272.