

Primera parte

Proyecto: **Taller de Costura.**

Lenguaje: **Python.**

Framework: **Django.**

Editor: **VS Code.**

1 Procedimiento para crear carpeta del Proyecto: **UIII_taller_costura_0306**

2 procedimientos para abrir VS Code sobre la carpeta

UIII_Cinepolis_0777

3 procedimientos 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 procedimientos para activar intérprete de Python.

7 Procedimiento para instalar Django

8 procedimientos para crear proyecto **backend_tallercostura** sin duplicar carpeta.

9 procedimientos para ejecutar servidor en el **puerto 8306**

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

11 procedimientos para crear aplicación **app_alumnos**

12 Aquí el modelo [**models.py**](#)

=====

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

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

```
# MODELO: ALUMNO
```

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

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

```
    # --- 7 campos ---
```

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

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

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

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

```
    direccion = models.CharField(max_length=200)
```

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

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

```
# Relación muchos a muchos (Alumno ↔ Clase)
```

```
    clases = models.ManyToManyField('Clase', through='Inscripcion',
related_name='alumnos')
```

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

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

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

```

# MODELO: CLASE
# =====
# --- 7 campos ---
nombre = models.CharField(max_length=100)
descripcion = models.TextField()
nivel = models.CharField(max_length=50, choices=[
    ('Básico', 'Básico'),
    ('Intermedio', 'Intermedio'),
    ('Avanzado', 'Avanzado'),
])
fecha_inicio = models.DateField()
fecha_fin = models.DateField()
duracion_horas = models.PositiveIntegerField()
cupo_maximo = models.PositiveIntegerField()

def __str__(self):
    return f"{self.nombre} ({self.nivel})"

# =====
# MODELO: INSCRIPCION
# =====
class Inscripcion(models.Model):
    # --- 7 campos ---
    alumno = models.ForeignKey(Alumno, on_delete=models.CASCADE)
    clase = models.ForeignKey(Clase, on_delete=models.CASCADE)
    fecha_inscripcion = models.DateField(auto_now_add=True)
    metodo_pago = models.CharField(max_length=50, choices=[
        ('Efectivo', 'Efectivo'),
        ('Tarjeta', 'Tarjeta'),
        ('Transferencia', 'Transferencia'),
    ])
    monto_pagado = models.DecimalField(max_digits=8, decimal_places=2)
    estado = models.CharField(max_length=50, choices=[
        ('Activa', 'Activa'),
        ('Cancelada', 'Cancelada'),
        ('Finalizada', 'Finalizada'),
    ])
    observaciones = models.TextField(blank=True, null=True)

    def __str__(self):
        return f"{self.alumno} → {self.clase}"

=====

```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

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

14 En view de **app_alumnos** crear las funciones con sus códigos correspondientes (inicio_tallercostura, agregar_alumno, actualizar_alumno, realizar_actualizacion_alumno, borrar_alumno)

15 Crear la carpeta “**templates**” dentro de “**app_taller**”.

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 Taller de Costura”, “Inicio”, “alumnos”,

En submenú de alumnos (Aregar alumno,ver alumno, actualizar alumno, borrar alumno),

“Clases” en submenu de clases (Aregar clase,ver clase, actualizar clase, borrar clase)

“inscripciones” en submenu de inscripciones (Aregar inscripciones,ver inscripciones, actualizar inscripciones, borrar inscripciones),

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 tecnico .Emma Ortiz, 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 costura.

21 Crear la subcarpeta carpeta alumno dentro de app_taller\templates.

22 crear los archivos html con su codigo correspondientes de (**agregar_alumno .html, ver_alumno.html** mostrar en tabla con los botones ver, editar y borrar, **actualizar_alumno.html, borrar_alumno.html**) dentro de **app_taller\templates\alumno**.

23 No utilizar forms.py.

24 procedimiento para crear el archivo urls.py en **app_taller** con

el código correspondiente para acceder a las funciones de views.py para operaciones de crud en categorías.

25 procedimiento para agregar **app_taller** en settings.py de **backend_tallercostura**

26 realizar las configuraciones correspondiente a urls.py de **backend_tallercostura** para enlazar con **app_taller**

27 procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.

27 por lo pronto solo trabajar con “**alumnos**” dejar pendiente # MODELO: inscripciones y # MODELO: clases

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 8306**.