

## Primera parte

Proyecto: **Tienda de Instrumentos**

Lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

1 Procedimiento para crear carpeta del Proyecto: **UIII\_Instrumental\_0777**

2 procedimiento para abrir vs code sobre la carpeta

**UIII\_Instrumental\_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\_instrumental** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto **8429**

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

11 procedimiento para crear aplicación **app\_instrumental**

12 Aquí el modelo models.py

=====

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

```
# =====
# MODELO: INSTRUMENTO
# =====
class Instrumento(models.Model):
    id_instrumento = models.AutoField(primary_key=True)
    nombre = models.CharField(max_length=255) marca = models.CharField(max_length=255)
    modelo = models.CharField(max_length=255)
    precio_venta = models.DecimalField(max_digits=10, decimal_places=2)
    cantidad_stock = models.IntegerField()
    categoria = models.CharField(max_length=255)
    fecha_ingreso = models.DateField()
    def __str__(self): return self.nombre

# =====
# MODELO: VENTA
# =====
class Venta(models.Model):
    id_venta = models.AutoField(primary_key=True)
    fecha_venta = models.DateField()
    total_venta = models.DecimalField(max_digits=10, decimal_places=2)
    iva_aplicado = models.DecimalField(max_digits=5, decimal_places=2)
    no_transaccion = models.CharField(max_length=255)
    nombre_vendedor = models.CharField(max_length=255)
    registro_caja = models.CharField(max_length=255)
```

```

def __str__(self): return f"Venta {self.id_venta} - {self.fecha_venta}"

# =====
# MODELO: DETALLE_VENTA
# =====
class DetalleVenta(models.Model):
    id_venta = models.ForeignKey(Venta, on_delete=models.CASCADE)
    id_instrumento = models.ForeignKey(Instrumento, on_delete=models.CASCADE)
    cantidad = models.IntegerField()
    precio_unitario = models.DecimalField(max_digits=10, decimal_places=2)
    subtotal = models.DecimalField(max_digits=10, decimal_places=2)
    costo_instrumento = models.DecimalField(max_digits=10, decimal_places=2)
    no_linea = models.IntegerField()
    nota_detalle = models.CharField(max_length=255, blank=True, null=True)
    porcentaje_descuento_item = models.DecimalField(max_digits=5, decimal_places=2,
    default=0.00)
    class Meta:
        unique_together = ('id_venta', 'no_linea') # Asegura que no haya dos líneas con el mismo
        número en la misma venta
    def __str__(self): return f"Detalle Venta {self.id_venta.id_venta} - Línea {self.no_linea}"
=====
```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO:INSTRUMENTOS

14 En view de app\_instrumental crear las funciones con sus códigos correspondientes (inicio\_instrumental, agregar\_instrumento,

actualizar\_instrumento, realizar\_actualizacion\_instrumento, borrar\_instrumento)

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

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 Instrumental”, “Inicio”, “instrumento”,en submenu de instrumentos(Agregar instrumento,ver instrumento, actualizar instrumento, borrar instrumento), “ventas” en submenu de ventas(Aregar venta,ver venta, actualizar venta, borrar venta)

“Detalle\_venta” en submenu de detalle\_venta(Agregar detalle\_venta,ver detalle\_venta, actualizar detalle\_venta, borrar detalle\_venta), 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. Alejandro Aceves, 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 instrumental.

21 Crear la subcarpeta carpeta instrumentos dentro de app\_instrumental\templates.

22 crear los archivos html con su código correspondientes de (agregar\_instrumento.html, ver\_instrumento.html mostrar en tabla con los botones ver, editar y borrar, actualizar\_instrumento.html, borrar\_instrumento.html)

dentro de app\_instrumental\templates\instrumentos.

23 No utilizar forms.py.

24 procedimiento para crear el archivo urls.py en app\_instrumental con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en instrumentos.

25 procedimiento para agregar app\_instrumental en settings.py de backend\_instrumental

26 realizar las configuraciones correspondiente a urls.py de backend\_instrumental para enlazar con app\_instrumental

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

27 por lo pronto solo trabajar con “INSTRUMENTOS” dejar pendiente # MODELO: VENTAS y # MODELO: DETALLE\_VENTA

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.