**tercera parte**
Proyecto: **Caffenio**.
lenguaje: **Python**.
Framework: **Django**.
Editor: **VS code**.
Carpeta del Proyecto: **UIII_Caffenio_0647**
Proyecto:**backend_Caffenio**
aplicación: **app_Caffenio**
1 Aqui el modelo models.py que ya esta creado

```
# ========================================
# TABLA: VENTAS  (7 CAMPOS)
# ========================================
class Venta(models.Model):
    fecha = models.DateTimeField(auto_now_add=True)
    cliente = models.CharField(max_length=100)
    total = models.DecimalField(max_digits=10, decimal_places=2)
    metodo_pago = models.CharField(
        max_length=50,
        choices=[
            ('efectivo', 'Efectivo'),
            ('tarjeta', 'Tarjeta'),
            ('transferencia', 'Transferencia')
        ]
    )
    sucursal = models.CharField(
        max_length=50,
        choices=[
            ('Sucursal 1', 'Sucursal 1'),
            ('Sucursal 2', 'Sucursal 2')
        ]
    )
    empleado = models.CharField(max_length=100)
    productos = models.ManyToManyField(
        Producto,
        related_name='ventas'
    )

    def __str__(self):
        return f"Venta {self.id} - {self.cliente}"
```

2 Procedimiento para realizar las migraciones(makemigrations y migrate).
3 Ahora trabajamos con el MODELO: Venta
4 En view de app_Caffenio crear las funciones con sus códigos correspondientes ( agregar_venta, actualizar_venta, realizar_actualizacion_venta borrar_venta)
**5. En la opción de agregar_venta, la tabla Venta está relacionada con Producto y Proveedor.**

**Primero se capturan los datos no relacionados de la venta (como cliente, total, método de pago, sucursal y empleado).**
**Después, en el campo productos debe aparecer un combobox con los productos disponibles por su nombre, para poder seleccionar uno o varios de ellos.**
**Asimismo, en el campo proveedor debe mostrarse un combobox con los proveedores registrados, mostrando su nombre para poder seleccionar el correspondiente.**

**6.** Modificar el archivo **navbar.html** que está dentro de **templates**, para actualizar la opción "**ventas**" en el submenú de **ventas** (*Agregar venta, ver ventas en forma de tabla, actualizar venta, borrar ventas*).

**7.** Crear la subcarpeta **venta** dentro de **app_Caffenio\templates**.

**8.** Crear los archivos HTML con su código correspondiente:
(**agregar_venta.html**, **ver_venta.html** —mostrar en tabla con los botones *ver*, *editar* y *borrar—**, **actualizar_venta.html**, **borrar_venta.html**)
dentro de **app_Caffenio\templates\venta**.

**9.** No utilizar **forms.py**.

**10.** Procedimiento para agregar en **urls.py** de **app_Caffenio** el código correspondiente para acceder a las funciones de **views.py** para las operaciones **CRUD** en **Venta**.

**11.** Procedimiento para registrar los modelos en **admin.py** y volver a realizar las **migraciones**.

**12.** Trabajando con **todos los modelos** (**Proveedores**, **Productos** y **Ventas**).

**13.** Utilizar **colores suaves, atractivos y modernos**, con páginas web sencillas para **Venta**.

**14.** No validar la entrada de datos.

**15.** Al inicio, crear la **estructura completa de carpetas y archivos actualizados**, donde se muestra la carpeta **venta** dentro de **templates** con los archivos HTML correspondientes a las operaciones **CRUD** de **Venta**.

**16.** Proyecto totalmente funcional.

**17.** Finalmente, ejecutar el servidor en el **puerto 8030**.