segunda 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: PRODUCTO  (7 CAMPOS)
# ========================================
class Producto(models.Model):
    nombre = models.CharField(max_length=100)
    descripcion = models.TextField()
    precio = models.DecimalField(max_digits=10, decimal_places=2)
    stock = models.IntegerField()
    categoria = models.CharField(
        max_length=50,
        choices=[
            ('bebidas', 'Bebidas'),
            ('alimentos', 'Alimentos'),
            ('complementos', 'Complementos'),
            ('otro', 'Otro')
        ]
    )
    proveedor = models.ForeignKey(
        Proveedor,
        on_delete=models.CASCADE,
        related_name='productos'
    )
    sucursal = models.CharField(
        max_length=50,
        choices=[
            ('Sucursal 1', 'Sucursal 1'),
            ('Sucursal 2', 'Sucursal 2')
        ]
    )

    def __str__(self):
        return self.nombre
# ========================================
```

2 Procedimiento para realizar las migraciones(makemigrations y migrate.
3 Ahora trabajamos con el **MODELO: PRODUCTO**
4 En view de app_Caffenio crear las funciones con sus códigos

correspondientes (agregar_producto, actualizar_producto,
realizar_actualizacion_producto
borrar_producto)
5 Modificar el archivo **navbar.html** que está dentro de **templates**,para actualizar la opcion
"**Productos**" en submenu de Productos(Agregar producto,ver producto, actualizar producto, borrar producto)

6 Crear la subcarpeta **producto** dentro de **app_Caffenio\templates**.
7 crear los archivos html con su codigo correspondientes de
(agregar_producto.html, ver_producto.html mostrar en tabla con
los botones ver, editar y borrar, actualizar_producto.html,
borrar_tabla.html) dentro de **app_Caffenio\templates\producto**.
8 No utilizar forms.py.
9 procedimiento para agregar en urls.py en **app_Caffenio** con
el código correspondiente para acceder a las funciones de views.py
para operaciones de crud en **productos**.
10 procedimiento para registrar los modelos en admin.py y volver a
realizar las migraciones.
11 por lo pronto solo trabajar con "producto", que es la segunda y seguir dejando pendiente # MODELO: VENTAS
12 Utilizar colores suaves, atractivos y modernos, el código de las
páginas web sencillas para productos.
13 No validar entrada de datos.
14 Al inicio crear la estructura completa de carpetas y archivos actualizados, donde se muestra la carpeta productos dentro de templates con los archivos html correspondientes a las operaciones crud productos.
15 proyecto totalmente funcional.
16 finalmente ejecutar servidor en el puerto puerto 8030.