

segunda parte

Proyecto: **muebleria**.

Lenguaje: **Python**.

Framework: **Django**.

Editor: VS code.

carpeta del Proyecto: **UIII_muebleria_0387**

proyecto: **backend_muebleria**

aplicacion **app_muebleria**

Aqui el modelo models.py ya esta creado

=====

```
from django.db import models

# =====
# MODELO: EMPLEADO
# =====
class Empleado(models.Model):
    fecha_contratacion = models.DateField()
    id_empleado = models.AutoField(primary_key=True)
    nombre = models.CharField(max_length=100)
    edad = models.IntegerField()
    cargo = models.CharField(max_length=20, choices=CARGO_CHOICES)
    telefono = models.CharField(max_length=15)
    sueldo = models.DecimalField(max_digits=10, decimal_places=2)

    def __str__(self):
        return f'{self.nombre} - {self.cargo}'


# =====
# MODELO: SUCURSAL
# =====
class Sucursal(models.Model):
    id_sucursal = models.AutoField(primary_key=True)
    telefono = models.CharField(max_length=15)
    direccion = models.CharField(max_length=200)
    num_sucursal = models.IntegerField()
    encargado = models.CharField(max_length=100)
    codigo_postal = models.CharField(max_length=10)
    ciudad = models.CharField(max_length=50)
    id_empleado = models.ForeignKey(Empleado, on_delete=models.CASCADE,
                                    related_name="sucursales")

    def __str__(self):
        return f'Sucursal {self.num_sucursal} - {self.ciudad}'


# =====
# MODELO: PRODUCTO
# =====
```

```

# =====
class Producto(models.Model):
    nombre = models.CharField(max_length=100)
    categoria = models.CharField(max_length=20, choices=CATEGORIA_CHOICES)
    material = models.CharField(max_length=20, choices=MATERIAL_CHOICES)
    precio = models.FloatField()
    stock = models.IntegerField()
    color = models.CharField(max_length=30)
    id_producto = models.AutoField(primary_key=True)
    sucursales = models.ManyToManyField(
        Sucursal,
        related_name="productos"
    )

    def __str__(self):
        return f"{self.nombre} - ${self.precio}"

```

1. Procedimiento para realizar las migraciones(makemigrations y migrate.

2. ahora trabajamos con el MODELO: sucursal

3. En view de **app_muebleria** crear las funciones con sus códigos correspondientes (inicio_sucursal, agregar_sucursal, actualizar_sucursal, realizar_actualizacion_sucursal, borrar_sucursal)

4. Crear la carpeta “**templates**” dentro de “**app_muebleria** ”.

5 modificar el archivo navbar.html que esta dentro de templates para actualizar la opcion sucursal con la carpeta “sucursal” en submenu de sucursal (inicio_sucursal, agregar_sucursal, actualizar_sucursal, realizar_actualizacion_sucursal, borrar_sucursal)

6 Crear la subcarpeta carpeta sucursal dentro de app_muebleria\templates.

7 crear los archivos html con su codigo correspondientes de (agregar_sucursal.html, ver_sucursal.html mostrar en tabla con los botones ver, editar y borrar, actualizar_sucursal.html, borrar_sucursal.html) dentro de app_muebleria\templates\sucursal

8 No utilizar forms.py

9 procedimiento para agregar en urls.py en app_muebleria con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en sucursal.

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

11 por lo pronto solo trabajar con “sucursal” dejar pendiente # MODELO: producto

12 Utilizar colores suaves, atractivos y modernos, el código de las páginas web sencillas para clientes.

13 No validar entrada de datos.

14 Al inicio crear la estructura completa de carpetas y archivos actualizados donde se muestra la carpeta sucursal dentro de templates con los archivos correspondientes a la operación crud sucursal.

15 proyecto totalmente funcional.

16 finalmente ejecutar servidor en el puerto puerto 8087.