Primera parte
Proyecto: **mascotas.**
lenguaje: **Python**.
Framework: **Django.**
Editor: **VS code.**
1 Procedimiento para crear carpeta del Proyecto: **UIII_mascotas_0237**
2 procedimiento para abrir vs code sobre la carpeta
UIII_mascotas_0237
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_mascotas** sin duplicar carpeta.
9 procedimiento para ejecutar servidor en el **puerto 8037**
10 procedimiento para copiar y pegar el link en el navegador.
11 procedimiento para crear aplicacion **app_tienda**
12 Aqui el modelo models.py
=-=-=-=-=-

```
from django.db import models

# ========================================
# MODELO: ANIMALES
# La tabla central que es referenciada por Clientes y Empleados
# ========================================
class Animal(models.Model):
    id_animales = models.AutoField(primary_key=True)
    nombre = models.CharField(max_length=100)
    edad = models.CharField(max_length=20, blank=True, null=True)
    peso = models.DecimalField(max_digits=6, decimal_places=2)
    cuidados = models.TextField(blank=True, null=True)
    enfermedades = models.TextField(blank=True, null=True)
    especie = models.CharField(max_length=50)
    alimentacion = models.TextField(blank=True, null=True)

    def __str__(self):
        return f"{self.nombre} ({self.especie}) - ID: {self.id_animales}"

# ========================================
# MODELO: CLIENTES
# Relacion 1 a muchos: Un Cliente puede tener muchos Animales.
# ========================================
class Cliente(models.Model):
    # Clave primaria
    id_cliente = models.AutoField(primary_key=True)
```

```
    apepaterno = models.CharField(max_length=100)
    apematerno = models.CharField(max_length=100)
    nombre = models.CharField(max_length=100)
    domicilio = models.CharField(max_length=255, blank=True, null=True)
    correo = models.CharField(max_length=100, unique=True)
    telefono = models.CharField(max_length=20)
    alergias = models.TextField(blank=True, null=True)
    def __str__(self):
        return f"{self.nombre} {self.apepaterno}"
Animal.add_to_class('dueno', models.ForeignKey(
    Cliente,
    on_delete=models.CASCADE,
    related_name="animales_poseidos",
    blank=True,
    null=True # Permite animales sin dueño registrado si es necesario
))

# ========================================
# MODELO: EMPLEADOS #
# ========================================
Relación Muchos a Muchos: Un Empleado atiende muchos Animales, # y un Animal puede ser atendido por muchos Empleados. #
======================================== class Empleado(models.Model): # Clave primaria id_empleado = models.AutoField(primary_key=True) # Atributos personales apepaterno = models.CharField(max_length=100) apematerno = models.CharField(max_length=100) nombre = models.CharField(max_length=100) telefono = models.CharField(max_length=20) domicilio = models.TextField(blank=True, null=True) ocupacion = models.CharField(max_length=100) alergias = models.TextField(blank=True, null=True) # RELACIÓN MUCHOS A MUCHOS (M:M) # Crea una tabla intermedia automáticamente (ej: Empleado_animales_a_cargo). animales_a_cargo = models.ManyToManyField( 'Animal', related_name="empleados_responsables", blank=True ) def __str__(self): return f"Empleado: {self.nombre} {self.apepaterno} ({self.ocupacion})"
```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: animales

14 En view de **app_mascotas** crear las funciones con sus códigos correspondientes (inicio_mascotas, agregar_mascota, actualizar_mascota, realizar_actualizacion_mascota, borrar_mascota)

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

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 mascotas”, “Inicio”, “mascotas”,en submenu de mascotas(Aregar mascotas,ver mascotas, actualizar mascotas,

borrar mascota), “clientes” en submenú de clientes(Agregar clientes,ver clientes, actualizar clientes, borrar clientes)
“empleados” en sub menu de empleados(Agregar empleados,ver empleados, actualizar empleados, borrar empleados), incluir iconos a las opciones principales, no en los submenús.
19 En el archivo **footer.html** incluir derechos de autor,fecha del sistema y “Creado por Sofia Granados Sanchez, 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 cinepolis.
21 Crear la subcarpeta carpeta **mascotas** dentro de **app_mascotas\templates.**
22 crear los archivos html con su codigo correspondientes de (agregar_mascotas.html, ver_mascotas.html mostrar en tabla con los botones ver, editar y borrar, actualizar_mascotas.html, borrar_mascotas.html)
dentro de **app_mascotas\templates\mascotas.**
23 No utilizar forms.py.
24 procedimiento para crear el archivo urls.py en **app_**mascotas con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en mascotas.
25 procedimiento para agregar **app_mascotas** en settings.py de **backend_mascotas**
26 realizar las configuraciones correspondiente a urls.py de **backend_mascotas** para enlazar con **app_mascotas**

27 **procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.**
27 por lo pronto solo trabajar con “mascotas” dejar pendiente # MODELO: clientes y # MODELO: empleados
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 **8037**.