

Primera parte

- **Proyecto:** Parisina.
- **Lenguaje:** Python.
- **Framework:** Django.
- **Editor:** VS code.

1 Procedimiento para crear carpeta del Proyecto: UIII_Parisina_0559

2 procedimiento para abrir vs code sobre la carpeta

UIII_Parisina_0559

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_Parisina sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto 0559

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

11 procedimiento para crear aplicacion app_Parisina

12 Aquí el modelo models.py

=====

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

```
# =====
# MODELO: CLIENTE
# =====
class Cliente(models.Model):
    nombre = models.CharField(max_length=100)
    apellido = models.CharField(max_length=100)
    correo_electronico = models.EmailField(unique=True)
    telefono = models.CharField(max_length=15)
    direccion = models.CharField(max_length=200)
    ciudad = models.CharField(max_length=100)
    fecha_registro = models.DateField(auto_now_add=True)

    def __str__(self):
        return f'{self.nombre} {self.apellido}'
```

```

# =====

# MODELO: PROVEEDOR
# =====

class Proveedor(models.Model):
    nombre_empresa = models.CharField(max_length=150)
    contacto = models.CharField(max_length=100)
    correo = models.EmailField()
    telefono = models.CharField(max_length=15)
    direccion = models.CharField(max_length=200)
    pais = models.CharField(max_length=100)
    fecha_registro = models.DateField(auto_now_add=True)

    def __str__(self):
        return self.nombre_empresa

# =====

# MODELO: PRODUCTO
# =====

class Producto(models.Model):
    nombre = models.CharField(max_length=150)
    descripcion = models.TextField()
    categoria = models.CharField(max_length=100)
    precio = models.DecimalField(max_digits=10, decimal_places=2)
    stock = models.PositiveIntegerField()
        cliente = models.ForeignKey(Cliente, on_delete=models.CASCADE,
related_name='productos') # Relación 1 a muchos
    proveedores = models.ManyToManyField(Proveedor, related_name='productos')
# Relación muchos a muchos

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

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate).

13 primero trabajamos con el MODELO: CLIENTE

14 En view de app_Parisina crear las funciones con sus códigos correspondientes (inicio_parisina, agregar_cliente, actualizar_cliente, realizar_actualizacion_cliente, borrar_cliente)

15 Crear la carpeta “templates” dentro de “app_Parisina”.

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 Parisina”, “Inicio”, “Cliente”, en submenu de Cliente(Aregar Cliente,ver Cliente, actualizar Cliente, borrar Cliente), “Proveedor” en submenu de Proveedor(Agregar Proveedor,ver Proveedor, actualizar Proveedor, borrar Proveedor)

“Producto” en submenu de Producto(Agregar Producto,ver Producto, actualizar Producto, borrar Producto), 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 De La Rosa Abril, 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 Parisina.

21 Crear la subcarpeta carpeta Cliente dentro de app_Parisina\templates.

22 crear los archivos html con su codigo correspondientes de (agregar_Cliente.html, ver_Cliente.html mostrar en tabla con los botones ver, editar y borrar, actualizar_Cliente.html, borrar_Cliente.html)

dentro de app_Parisina\templates\Cliente.

23 No utilizar forms.py.

24 procedimiento para crear el archivo urls.py en app_Parisina con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en Cliente.

25 procedimiento para agregar app_Parisina en settings.py de backend_Parisina

26 realizar las configuraciones correspondiente a urls.py de backend_Parisina para enlazar con app_Parisina

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

27 por lo pronto solo trabajar con “Cliente” dejar pendiente #

MODELO: PROVEEDOR y # MODELO: PRODUCTO

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 0559.