

Primera parte

Proyecto: **Cinépolis.**

Lenguaje: **Python.**

Framework: **Django.**

Editor: **VS code.**

1 Procedimiento para crear carpeta del Proyecto: UIII_Lavanderia_0452

2 procedimiento para abrir vs code sobre la carpeta

UIII_Lavanderia_0452

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_Lavanderia sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto 8052

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

11 procedimiento para crear aplicación app_Lavanderia

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)
    telefono = models.CharField(max_length=15)
    correo = models.EmailField()
    direccion = models.CharField(max_length=200)
    fecha_registro = models.DateField(auto_now=True)
    notas = models.TextField(blank=True, null=True)

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


# =====
# MODELO EMPLEADO
# =====
class Empleado(models.Model):
    nombre = models.CharField(max_length=100)
    apellido = models.CharField(max_length=100)
```

```

puesto = models.CharField(max_length=100)
telefono = models.CharField(max_length=15)
correo = models.EmailField()
fecha_contratacion = models.DateField()
salario = models.DecimalField(max_digits=8, decimal_places=2)

def __str__(self):
    return f"{self.nombre} ({self.puesto})"

# =====
# MODELO SERVICIO
# =====
class Servicio(models.Model):
    cliente = models.ForeignKey(Cliente, on_delete=models.CASCADE,
related_name='servicios') # Relación 1:N
    empleados = models.ManyToManyField(Empleado,
related_name='servicios') # Relación N:M

    tipo_servicio = models.CharField(max_length=100) # Ej: Lavado,
Planchado, Tintorería
    fecha_servicio = models.DateField()
    hora_entrega = models.TimeField() # Nuevo campo: hora estimada de
entrega
    descripcion = models.TextField()
    precio = models.DecimalField(max_digits=8, decimal_places=2)
    estado = models.CharField(
        max_length=50,
        choices=[
            ('Pendiente', 'Pendiente'),
            ('En proceso', 'En proceso'),
            ('Completado', 'Completado')
        ],
        default='Pendiente'
    )
    forma_pago = models.CharField(max_length=50, default='Efectivo') # Nuevo campo: método de pago

def __str__(self):
    return f"{self.tipo_servicio} - {self.cliente.nombre}"

```

=====

12.5 Procedimiento para realizar las migraciones(makemigrations y

migrate.

13 primero trabajamos con el MODELO: CLIENTE

14 En view de app_Lavanderia crear las funciones con sus códigos correspondientes (inicio_lavanderia, agregar_cliente,

actualizar_cliente, realizar_actualizacion_cliente,
borrar_cliente)

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

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 Lavanderia”, “Inicio”, “cliente”, en submenu de categorias(Aregar Cliente,ver cliente, actualizar cliente, borrar cliente), “Salas” en submenu de Empleado(Aregar empleado,ver empleado, actualizar empleado, borrar empleado)

“Peliculas” en submenu de Servicio(Aregar servicio,ver servicio, actualizar servicio, borrar servicio), 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 Jesus Campoya, 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 lavanderia.

21 Crear la subcarpeta carpeta categoria dentro de app_Lavanderia\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_Lavanderia\templates\clientes.

23 No utilizar forms.py.

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

25 procedimiento para agregar app_Lavanderia en settings.py de backend_Lavanderia

26 realizar las configuraciones correspondiente a urls.py de backend_Lavanderia para enlazar con app_Lavanderia

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: EMPLEADO y # MODELO: SERVICIO

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 8052.