

Proyecto: spa  
lenguaje: Python.  
Framework: Django.  
Editor: VS code.

- 1 Procedimiento para crear carpeta del Proyecto: UIII\_Spa\_0262
- 2 procedimiento para abrir vs code sobre la carpeta UIII\_spa\_0262
- 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\_spa sin duplicar carpeta.
- 9 procedimiento para ejecutar servidor en el puerto 0262
- 10 procedimiento para copiar y pegar el link en el navegador.
- 11 procedimiento para crear aplicacion app\_spa
- 12 Aqui el modelo models.py

=====

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

=====

MODELO: Empleados

=====

```
class Empleados(models.Model):  
    id_empleados = models.AutoField(primary_key=True)  
    nombre = models.CharField(max_length=100)  
    apellido = models.CharField(max_length=100)  
    especialidad = models.CharField(max_length=100)  
    telefono = models.IntegerField()  
    cargo = models.CharField(max_length=100)  
    id_servicios = models.IntegerField()
```

text

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

=====

MODELO: Clientes

=====

```
class Clientes(models.Model):  
    id_clientes = models.AutoField(primary_key=True)  
    nombre = models.CharField(max_length=100)  
    apellido = models.CharField(max_length=100)  
    email = models.EmailField(max_length=100)  
    telefono = models.IntegerField()  
    fecha_registro = models.DateField(auto_now_add=True)
```

```
alergias = models.TextField(blank=True, null=True)
preferencias = models.TextField(blank=True, null=True)

text
def __str__(self):
    return f'{self.nombre} {self.apellido}'
=====
MODELO: Servicios
=====
class Servicios(models.Model):
    id_servicios = models.AutoField(primary_key=True)
    nombre = models.CharField(max_length=100)
    descripcion = models.TextField(blank=True, null=True)
    precio = models.DecimalField(max_digits=10, decimal_places=2)
    duracion = models.IntegerField(help_text="Duración en minutos")
    tipo_servicio = models.CharField(max_length=100)
    id_clientes = models.IntegerField()

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

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate).
13 primero trabajamos con el MODELO: EMPLEADOS
14 En view de app_spa crear las funciones con sus códigos correspondientes (inicio_spa,
agregar_empleado, actualizar_empleado, realizar_actualizacion_empleado, borrar_empleado)
15 Crear la carpeta "templates" dentro de "app_spa".
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 spa", "Inicio",
"Empleados",en submenu de Empleados(Agregar Empleado,ver Empleado, actualizar
Empleado, borrar Empleado), "Clientes" en submenu de Clientes(Agregar Cliente,ver Cliente,
actualizar Cliente, borrar Cliente) "Servicios" en submenu de Servicios(Agregar 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 Martinez
Gomez Oscar, 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 spa.
21 Crear la subcarpeta carpeta empleados dentro de app_spa\templates.
22 crear los archivos html con su codigo correspondientes de (agregar_empleado.html,
ver_empleados.html mostrar en tabla con los botones ver, editar y borrar,
actualizar_empleado.html, borrar_empleado.html) dentro de app_spa\templates\empleados.
```

- 23 No utilizar forms.py.
- 24 procedimiento para crear el archivo urls.py en app\_spa con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en empleados.
- 25 procedimiento para agregar app\_spa en settings.py de backend\_spa
- 26 realizar las configuraciones correspondiente a urls.py de backend\_spa para enlazar con app\_spa
- 27 procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.
- 27 por lo pronto solo trabajar con "EMPLEADOS" dejar pendiente # MODELO: CLIENTES y # MODELO: SERVICIOS
- 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 0262.