

Primera parte

Proyecto: Spa.

lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

1 Procedimiento para crear carpeta del Proyecto: **Ull_Spa_0606**

2 procedimiento para abrir vs code sobre la carpeta

Ull_Spa_0606

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 8020**

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
    from django.db import models

# =====
# MODELO: USUARIOS
# =====
class Usuarios(models.Model):
    id_usuario = models.AutoField(primary_key=True)
    nombre_completo = models.CharField(max_length=100)
    correo = models.EmailField(unique=True, max_length=250)
    telefono = models.CharField(max_length=15)
    direccion = models.CharField(max_length=200)
    fecha_registro = models.DateTimeField(auto_now_add=True)
    activo = models.BooleanField(default=True)

    def __str__(self):
        return self.nombre_completo
```

```

# =====
# MODELO: PEDIDOS
# =====
class Pedidos(models.Model):
    id_pedido = models.AutoField(primary_key=True, unique=True)
    usuario = models.ForeignKey(
        Usuarios,
        on_delete=models.CASCADE,
        related_name="pedidos"
    )
    fecha_pedido = models.DateTimeField(auto_now_add=True)
    servicio = models.CharField(max_length=100)
    precio_total = models.DecimalField(max_digits=8, decimal_places=2)
    notas_adicionales = models.TextField(blank=True, null=True)
    estado = models.CharField(
        max_length=50,
        default="pendiente",
        choices=[
            ("pendiente", "Pendiente"),
            ("en_proceso", "En Proceso"),
            ("completado", "Completado"),
            ("cancelado", "Cancelado"),
        ]
    )

    def __str__(self):
        return f"Pedido #{self.id_pedido} - {self.servicio}"

# =====
# MODELO: RESEÑAS
# =====
class Reseñas(models.Model):
    id_reseña = models.AutoField(primary_key=True, unique=True)
    usuario = models.ForeignKey(
        Usuarios,
        on_delete=models.CASCADE,
        related_name="reseñas"
    )
    pedido = models.ForeignKey(
        Pedidos,
        on_delete=models.CASCADE,
        related_name="reseñas"
    )

```

```

)
titulo = models.CharField(max_length=100)
comentario = models.TextField()
calificacion = models.IntegerField(default=5)
fecha_creacion = models.DateTimeField(auto_now_add=True)
visible = models.BooleanField(default=True)

def __str__(self):
    return f"{self.titulo} - {self.calificacion}★"

```

==--==--==

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: USUARIOS

14 En view de **app_Spa** crear las funciones con sus códigos correspondientes (inicio_spa, agregar_usuarios, actualizar_usuarios,

realizar_actualizacion_USUARIOS, borrar_usuarios)

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", "spa",en submenu de usuarios(Aregar usuario,ver usuarios, actualizar usuario, borrar usuario), "Pedidos" en submenu de Pedidos(Aregar pedidos,ver pedidos, actualizar pedidos, borrar pedidos)

"Reseñas" en submenu de Reseñas(Aregar reseñas,ver reseñas, actualizar reseñas, borrar reseñas), 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 Valeria Herrera 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 spa.
- 21 Crear la subcarpeta carpeta **usuarios** dentro de **app_Spa\templates**.
- 22 crear los archivos html con su código correspondientes de (agregar_usuario.html, ver_usuarios.html mostrar en tabla con los botones ver, editar y borrar, actualizar_usuario.html, borrar_usuario.html) dentro de **app_Spa\templates\usuarios**.
- 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 usuarios.
- 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 “usuarios” dejar pendiente # MODELO: PEDIDOS y # MODELO: RESEÑAS
- 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 8020**.