

Primera parte

Proyecto: Clinicanutricional

Lenguaje: Python.

Framework: Django.

Editor: VS code.

1 Procedimiento para crear carpeta del Proyecto: UIII_Clinicanutricional_8313

2 procedimiento para abrir vs code sobre la carpeta UIII_Clinicanutricional_8313

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_clinicanutricional sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto 8313

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

11 procedimiento para crear aplicación app_clinicanutricional

12 Aquí el modelo models.py

=====

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

MODELO: NUTRÍÓLOGOS

```
class Nutriologo(models.Model):
    nombre = models.CharField(max_length=50)
    apellido = models.CharField(max_length=50)
    correo = models.EmailField(unique=True)
    direccion = models.CharField(max_length=100)
    telefono = models.CharField(max_length=10)
    experiencia = models.PositiveIntegerField(help_text="Años de experiencia")
    especialidad = models.CharField(max_length=100)
```

```
def __str__(self):
```

```
    return f"{self.nombre} {self.apellido}"\n\n\nclass Meta:\n    verbose_name = "Nutriólogo"\n    verbose_name_plural = "Nutriólogos"
```

MODELO: PACIENTES

```
class Paciente(models.Model):
    nombre = models.CharField(max_length=50)
    apellido = models.CharField(max_length=50)
    telefono = models.CharField(max_length=10)
    fechnacimiento = models.DateField()
    correo = models.EmailField(unique=True)
    direccion = models.CharField(max_length=100)
    sexo = models.CharField(max_length=10, choices=[('Femenino', 'Femenino'), ('Masculino', 'Masculino')])
    id_nut = models.ForeignKey(Nutriologo, on_delete=models.CASCADE,
    related_name='pacientes')

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


class Meta:
    verbose_name = "Paciente"
    verbose_name_plural = "Pacientes"
```

MODELO: CITAS

```
class Cita(models.Model):
    fecha = models.DateField()
    horario = models.TimeField()
    motivo = models.CharField(max_length=200)
    estado = models.CharField(max_length=50, choices=[
        ('pendiente', 'Pendiente'),
        ('confirmada', 'Confirmada'),
        ('cancelada', 'Cancelada'),
        ('finalizada', 'Finalizada'),
    ])
    observaciones = models.TextField(blank=True, null=True)
    duracion = models.PositiveIntegerField(help_text="Duración en minutos")
    id_nut = models.ForeignKey(Nutriologo, on_delete=models.CASCADE,
                               related_name='citas')

    def __str__(self):
        return f"Cita con {self.id_nut.nombre} el {self.fecha} a las
{self.horario}"

class Meta:
    verbose_name = "Cita"
    verbose_name_plural = "Citas"
```

- 12.5 Procedimiento para realizar las migraciones (makemigrations y migrate).
- 13 primero trabajamos con el MODELO: NUTRÍÓLOGOS
- 14 En view de app_clinicanutricional crear las funciones con sus códigos correspondientes (inicio_clinicanutricional, agregar_nutriologo, actualizar_nutriologo, realizar_actualizacion_nutriologo, borrar_nutriologo)
- 15 Crear la carpeta “templates” dentro de “app_clinicanutricional”.
- 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 Clínica Nutricional”, “Inicio”, “Nutriólogos”, en submenu de Nutriólogos (Agregar Nutriólogo, Ver Nutriólogos, Actualizar Nutriólogo, Borrar Nutriólogo), “Pacientes” en submenu de Pacientes (Agregar Paciente, Ver Pacientes, Actualizar Paciente, Borrar Paciente), “Citas” en submenu de Citas (Agregar Cita, Ver Citas, Actualizar Cita, Borrar Cita), 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 Ing. Eliseo Nava, 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 la Clínica Nutricional.

21 Crear la subcarpeta nutriologos dentro de app_clinicanutricional\templates.

22 crear los archivos html con su código correspondiente de (agregar_nutriologo.html, ver_nutriologos.html mostrar en tabla con los botones ver, editar y borrar, actualizar_nutriologo.html, borrar_nutriologo.html) dentro de app_clinicanutricional\templates\nutriologos.

23 No utilizar forms.py.

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

25 procedimiento para agregar app_clinicanutricional en settings.py de backend_clinicanutricional

26 realizar las configuraciones correspondiente a urls.py de backend_clinicanutricional para enlazar con app_clinicanutricional

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

27 por lo pronto solo trabajar con “Nutriólogos” dejar pendiente # MODELO: PACIENTES y # MODELO: CITAS

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 8313.