

Primera parte

Proyecto: Hospital de la mujer.

lenguaje: Python.

Framework: Django.

Editor: VS code.

1 Procedimiento para crear carpeta del Proyecto: UIII\_Hospitaldelamujer\_0290

2 procedimiento para abrir vs code sobre la carpeta

UIII\_Hospitaldelamujer\_0290

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\_Hospitalmujer sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto 8023

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

11 procedimiento para crear aplicacion app\_Hospital

12 Aqui el modelo models.py

==--==--==--

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

=====

=====

## MODELO: Beneficiaria

=====

=====

```
class Beneficiaria(models.Model):
```

```
id = models.AutoField(primary_key=True)
```

```
nombre = models.CharField(max_length=100)
```

```
apellido = models.CharField(max_length=100)
```

```
fecha_nacimiento = models.DateField()
```

```
direccion = models.CharField(max_length=255)
```

```
telefono = models.CharField(max_length=20, blank=True, null=True)
```

```
correo = models.EmailField(unique=True, blank=True, null=True)
numero_expediente = models.CharField(max_length=50, unique=True)
```

codeCode

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

```
=====
=====
```

## MODELO: Personal

```
=====
=====
```

```
class Personal(models.Model):
    id = models.AutoField(primary_key=True)
    nombre = models.CharField(max_length=100)
    apellido = models.CharField(max_length=100)
    puesto = models.CharField(max_length=100)
    especialidad = models.CharField(max_length=100, blank=True, null=True)
    telefono = models.CharField(max_length=20, blank=True, null=True)
    correo = models.EmailField(unique=True, blank=True, null=True)
    fecha_contratacion = models.DateField()
```

codeCode

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

```
=====
=====
```

## MODELO: Consulta

```
=====
=====
```

```

class Consulta(models.Model):
    id = models.AutoField(primary_key=True)
    beneficiaria = models.ForeignKey(Beneficiaria, on_delete=models.CASCADE,
    related_name='consultas')
    personal = models.ForeignKey(Personal, on_delete=models.CASCADE,
    related_name='consultas')
    fecha_consulta = models.DateTimeField(auto_now_add=True)
    motivo = models.TextField()
    diagnostico = models.TextField(blank=True, null=True)
    tratamiento = models.TextField(blank=True, null=True)
    costo = models.DecimalField(max_digits=10, decimal_places=2, default=0.00)

```

codeCode

```

def __str__(self):
    return f"Consulta de {self.beneficiaria.nombre} el {self.fecha_consulta.strftime('%Y-%m-%d %H:%M')}"

```

==--==--==

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: BENEFICIARIA

14 En view de app\_Hospital crear las funciones con sus códigos correspondientes (inicio\_hospital, agregar\_beneficiaria, actualizar\_beneficiaria, realizar\_actualizacion\_beneficiaria, borrar\_beneficiaria)

15 Crear la carpeta “templates” dentro de “app\_Hospital”.

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 Hospital de la Mujer”, “Inicio”, “Beneficiarias”, en submenu de Beneficiarias (Agregar Beneficiaria, ver Beneficiarias, actualizar Beneficiaria, borrar Beneficiaria), “Personal” en submenu de Personal (Agregar Personal, ver Personal, actualizar Personal, borrar Personal), “Consultas” en submenu de Consultas (Agregar Consulta, ver Consultas, actualizar Consulta, borrar Consulta), 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 un hospital de la mujer.

21 Crear la subcarpeta 'beneficiaria' dentro de app\_Hospital\templates.

22 crear los archivos html con su código correspondientes de (agregar\_beneficiaria.html, ver\_beneficiarias.html mostrar en tabla con los botones ver, editar y borrar, actualizar\_beneficiaria.html, borrar\_beneficiaria.html) dentro de app\_Hospital\templates\beneficiaria.

23 No utilizar forms.py.

24 procedimiento para crear el archivo urls.py en app\_Hospital con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en beneficiarias.

25 procedimiento para agregar app\_Hospital en settings.py de backend\_Hospitalmujer

26 realizar las configuraciones correspondiente a urls.py de backend\_Hospitalmujer para enlazar con app\_Hospital

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

27 por lo pronto solo trabajar con “Beneficiaria” dejar pendiente # MODELO: PERSONAL y # MODELO: CONSULTA

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 8023.