

Proyecto: **Hotel**
lenguaje: **Python**
Framework: **Django**
Editor: **VS code**

1 Procedimiento para crear carpeta del Proyecto: **UIII_Hotel_0743**

2 procedimiento para abrir vs code sobre la carpeta
UIII_Hotel_0743

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_Hotel** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto **8036**

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

11 procedimiento para crear aplicacion **app_Hotel**

12 Aqui el modelo models.py
=====

```
from django.db import models

# =====
# MODELO: HUESPED
# =====
class Huesped(models.Model):
    nombre = models.CharField(max_length=100)
    apellido = models.CharField(max_length=100)
    email = models.EmailField(unique=True)
    telefono = models.CharField(max_length=20)
    fecha_nacimiento = models.DateField()
    direccion = models.CharField(max_length=200)
    def __str__(self):
        return self.nombre

# =====
```

```

# MODELO: HABITACION
# =====
class Habitacion(models.Model):
    numero = models.IntegerField(unique=True)
    tipo = models.CharField(max_length=50)
    precio = models.DecimalField(max_digits=8, decimal_places=2)
    descripcion = models.TextField()
    estado = models.CharField(max_length=50)
    piso = models.IntegerField()
    def __str__(self):
        return f"Habitación {self.numero}"

# =====
# MODELO: RESERVA
# =====
class Reserva(models.Model):
    fecha_inicio = models.DateField()
    fecha_fin = models.DateField()
    fecha_reserva = models.DateField(auto_now_add=True)
    estado = models.CharField(max_length=50)
    huesped = models.ForeignKey(Huesped, on_delete=models.CASCADE)
    habitaciones = models.ManyToManyField(Habitacion,
through='ReservaHabitacion')
    def __str__(self):
        return f"Reserva {self.id}"

# =====
# TABLA INTERMEDIA (7 CAMPOS)
# =====
class ReservaHabitacion(models.Model):
    reserva = models.ForeignKey(Reserva, on_delete=models.CASCADE)
    habitacion = models.ForeignKey(Habitacion,
on_delete=models.CASCADE)
    precio = models.DecimalField(max_digits=8, decimal_places=2)
    fecha_checkin = models.DateField()
    fecha_checkout = models.DateField()
    estatus = models.CharField(max_length=50)
    def __str__(self):
        return f"Reserva {self.reserva.id} - Habitación
{self.habitacion.numero}"

```

=====

12.5 Procedimiento para realizar las migraciones (makemigrations y migrate)

13 primero trabajamos con el MODELO: **HUESPED**

14 En view de app_Hotel crear las funciones con sus códigos correspondientes (inicio_hotel, agregar_huesped, actualizar_huesped, realizar_actualizacion_huesped, borrar_huesped)

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

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 Hotel”, “Inicio”, “Huésped”, en submenu de Huésped(Agregar huésped, ver huéspedes, actualizar huésped, borrar huésped), “Habitación” en submenu de Habitación(Agregar habitación, ver habitaciones, actualizar habitación, borrar habitación) “Reservas” en submenu de Reservas(Agregar reserva, ver reservas, actualizar reserva, borrar reserva), 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 hoteles.

21 Crear la subcarpeta carpeta **huesped** dentro de app_Hotel\templates.

22 crear los archivos html con su código correspondientes de (agregar_huesped.html, ver_huesped.html mostrar en tabla con los botones ver, editar y borrar, actualizar_huesped.html, borrar_huesped.html) dentro de app_Hotel\templates\huesped.

23 No utilizar forms.py.

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

25 procedimiento para agregar app_Hotel en settings.py de backend_Hotel

26 realizar las configuraciones correspondiente a urls.py de backend_Hotel para enlazar con app_Hotel

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

27 por lo pronto solo trabajar con “HUESPED” dejar pendiente MODELO: HABITACION y MODELO: RESERVA

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 8036.