

Primera parte

Proyecto: **Restaurante**.

lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

1 Procedimiento para crear carpeta del Proyecto: **UIII_Restaurante_0676**

2 procedimiento para abrir vs code sobre la carpeta

UIII_Restaurante_0676

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_Restaurante** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el **puerto 8076**

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

11 procedimiento para crear aplicacion **app_Restaurante**

12 Aqui el modelo [models.py](#)

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

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

```
# =====
```

```
# MODELO: MENU
```

```
# =====
```

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

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

```
    descripcion = models.TextField()
```

```
    precio = models.DecimalField(max_digits=6, decimal_places=2)
```

```
    categoria = models.CharField(max_length=50)
```

```
    disponible = models.BooleanField(default=True)
```

```
    fecha_creacion = models.DateTimeField(auto_now_add=True)
```

```
    imagen = models.URLField(blank=True, null=True)
```

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

```
        return self.nombre
```

```
# =====
```

```
# MODELO: PEDIDO
```

```
# =====
```

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

```
    # Relación 1 a muchos: un menú puede tener muchos pedidos
```

```
    menu = models.ForeignKey(Menu, on_delete=models.CASCADE,  
related_name='pedidos')
```

```

nombre_cliente = models.CharField(max_length=100)
cantidad = models.PositiveIntegerField()
total = models.DecimalField(max_digits=8, decimal_places=2)
metodo_pago = models.CharField(max_length=50)
fecha_pedido = models.DateTimeField(auto_now_add=True)
entregado = models.BooleanField(default=False)

def __str__(self):
    return f"Pedido #{self.id} - {self.nombre_cliente}"
# =====
# MODELO: RESERVACION
# =====

class Reservacion(models.Model):
    nombre_cliente = models.CharField(max_length=100)
    correo = models.EmailField()
    telefono = models.CharField(max_length=15)
    fecha_reservacion = models.DateField()
    hora = models.TimeField()
    numero_personas = models.PositiveIntegerField()

    # Relación muchos a muchos: una reservación puede incluir varios menús
    menus = models.ManyToManyField(Menu, related_name='reservaciones')

    notas = models.TextField(blank=True, null=True)

    def __str__(self):
        return f"Reservación de {self.nombre_cliente} ({self.fecha_reservacion})"
=====

```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: MENU

14 En view de **app_Restaurante** crear las funciones con sus códigos correspondientes (inicio_Restaurante, agregar_Menu, actualizar_Menu, realizar_actualizacion_Menu, borrar_Menu)

15 Crear la carpeta “**templates**” dentro de “**app_Restaurante**”.

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 Restaurante”, “Inicio”, “Menu”, en submenu de Menu(Aregar Menu, ver Menu, actualizar Menu, borrar Menu), “Pedidos” en submenu de Pedidos(Aregar Pedidos, ver Pedidos, actualizar Pedidos, borrar Pedidos)

“Reservaciones” en submenu de Reservaciones(Aregar Reservaciones, ver Reservaciones,

actualizar Reservas, borrar Reservas), incluir iconos a las opciones principales, no en los submenú.

19 En el archivo **footer.html** incluir derechos de autor, fecha del sistema y "Creado por Nicolas Rios, 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 cinopolis.

21 Crear la subcarpeta carpeta **Menu** dentro de **app_Restaurante\templates**.

22 crear los archivos html con su código correspondientes de (agregar_Menu.html, ver_Menu.html mostrar en tabla con los botones ver, editar y borrar, actualizar_Menu.html, borrar_Menu.html)

dentro de **app_Restaurante\templates\Menu**.

23 No utilizar forms.py.

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

25 procedimiento para agregar **app_Restaurante** en settings.py de **backend_Restaurante**

26 realizar las configuraciones correspondiente a urls.py de **backend_Restaurante** para enlazar con **app_Restaurante**

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

27 por lo pronto solo trabajar con "Menu" dejar pendiente # MODELO: PEDIDO y # MODELO: RESERVACION

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 8076**