

# Primera parte

Proyecto: **Restaurante.**

Lenguaje: **Python.**

Framework: **Django.**

Editor: **VS Code.**

- 1 Procedimiento para crear carpeta del Proyecto: **Ulll\_Restaurante\_0344**
- 2 procedimiento para abrir **vs code** sobre la carpeta **Ulll\_Restaurante\_0344**
- 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 8044**
- 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: PLATILLO
```

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

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

```
    nombre = models.CharField(max_length=100, unique=True)
```

```
    descripcion = models.TextField(blank=True, null=True)
```

```
    precio = models.DecimalField(max_digits=10, decimal_places=2)
```

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

```
    tiempo_preparacion = models.IntegerField(verbose_name="Tiempo (min)")
```

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

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

```
    popularidad = models.PositiveIntegerField(default=0)
```

```

    icono = models.CharField(max_length=100, blank=True, null=True)

    def __str__(self):
        return self.nombre

# =====

# MODELO: PEDIDO

# =====

class Pedido(models.Model):
    cliente = models.CharField(max_length=100)
    fechahora = models.DateTimeField(auto_now_add=True)
    estado = models.CharField(max_length=30, default='Pendiente')
    direccionentrega = models.CharField(max_length=200)
    total = models.DecimalField(max_digits=10, decimal_places=2,
                                default=0.00)
    notas = models.TextField(blank=True, null=True)

    class Meta:
        ordering = ['-fechahora']

    def __str__(self):
        return f"Pedido #{self.id} de {self.cliente}"

# =====

# MODELO: DETALLEPEDIDO

# =====

class DetallePedido(models.Model):

```

```

    pedido = models.ForeignKey(Pedido, on_delete=models.CASCADE,
related_name='detalles')

    platillo = models.ForeignKey(Platillo, on_delete=models.PROTECT,
related_name='detalles_pedido')

    cantidad = models.IntegerField()

    preciounitario = models.DecimalField(max_digits=10, decimal_places=2)

    subtotal = models.DecimalField(max_digits=10, decimal_places=2)

    notas = models.CharField(max_length=255, blank=True, null=True)

class Meta:

    unique_together = ('pedido', 'platillo')

def __str__(self):

    return f"Detalle {self.id} - Platillo: {self.platillo.nombre}"

```

==--==--==

**12.5** Procedimiento para realizar las migraciones (**makemigrations** y **migrate**).

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

**14** En **view** de **app\_Restaurante** crear las funciones con sus códigos correspondientes (**inicio\_restaurante**, **agregar\_platillo**, **actualizar\_platillo**, **realizar\_actualizacion\_platillo**, **borrar\_platillo**)

**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”**, **“Platillos”** en submenu de Platillos (**Agregar Platillo**, **ver platillos**, **actualizar platillo**, **borrar platillo**), **“Pedidos”** en submenu de Pedidos (**Agregar pedidos**, **ver pedidos**, **actualizar pedidos**, **borrar pedidos**), **“Detalle Pedido”** en submenu de Detalle Pedido (**Agregar detalles**, **ver detalles**, **actualizar detalles**, **borrar detalles**), 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 restaurante**.

**21** Crear la subcarpeta carpeta **platillo** dentro de **app\_Restaurante/templates**.

**22** crear los archivos **html** con su código correspondientes de (**agregar\_platillo.html**, **ver\_platillos.html** —mostrar en tabla con los botones ver,

editar y borrar—, **actualizar\_platillo.html**, **borrar\_platillo.html**) dentro de **app\_Restaurante\templates\platillo**.

**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 platillos.

**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**.

**27.5** por lo pronto solo trabajar con “**platillo**” dejar pendiente **# MODELO: PEDIDO** y **# MODELO: DETALLEPEDIDO**

**28** Utilizar **colores suaves, atractivos y modernos**, el código de las páginas web sencillas.

**28.5** 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 8044**.