

## **Primera parte**

Proyecto: **Cinépolis.**

Lenguaje: **Python.**

Framework: **Django.**

Editor: **VS code.**

- 1 Procedimiento para crear carpeta del Proyecto: **UIII\_Paleteria\_0588**
- 2 procedimiento para abrir vs code sobre la carpeta **UIII\_Paleteria\_0588**
- 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\_paleteria** 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 aplicación **app\_Paleteria**
- 12 Aquí el modelo [\*\*models.py\*\*](#)

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

# =====
# MODELO: PROVEEDOR
# =====
class Proveedor(models.Model):
    id_proveedor = models.AutoField(primary_key=True)
    nombre_empresa = models.CharField(max_length=100, unique=True)
    contacto = models.CharField(max_length=100, unique=True)
    telefono = models.CharField(max_length=50, unique=True)
    email = models.CharField(max_length=100, unique=True)
    direccion = models.CharField(max_length=100, unique=True)
    fecha_registro = models.DateField()
    ciudad = models.CharField(max_length=100)

    def __str__(self):
        return self.nombre_empresa

# =====
# MODELO: PRODUCTO
# =====
class Producto(models.Model):
    id_producto = models.AutoField(primary_key=True)
    nombre = models.CharField(max_length=100, unique=True)
    sabor = models.CharField(max_length=100, unique=True)
    precio = models.CharField(max_length=100)
    stock = models.CharField(max_length=100)
    tipo = models.CharField(max_length=100)
    fecha_elaboracion = models.CharField(max_length=100)
    id_proveedor = models.ForeignKey(Proveedor, on_delete=models.CASCADE,
db_column='id_proveedor', related_name='productos')

    def __str__(self):
        return f"{self.nombre} ({self.sabor})"
```

```

# =====
# MODELO: CLIENTE
# =====
class Cliente(models.Model):
    id_cliente = models.AutoField(primary_key=True)
    nombre = models.CharField(max_length=100, unique=True)
    apellido = models.CharField(max_length=100, unique=True)
    telefono = models.CharField(max_length=50, unique=True)
    email = models.CharField(max_length=100, unique=True)
    direccion = models.CharField(max_length=100, unique=True)
    fecha_registro = models.DateField()
    productos = models.ManyToManyField(Producto, related_name='clientes')

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

=====

- 12.5 Procedimiento para realizar las migraciones(makemigrations y migrate).
- 13 primero trabajamos con el MODELO: PROVEEDOR
- 14 En view de **app\_Paleteria** crear las funciones con sus códigos correspondientes (inicio\_paleteria, agregar\_proveedor, actualizar\_proveedor, realizar\_actualizacion\_proveedor, borrar\_proveedor)
- 15 Crear la carpeta “**templates**” dentro de “**app\_Paleteria**”.
- 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 Paleteria”, “Inicio”, “proveedor”,en submenu de proveedor(Agregar Proveedor, ver Proveedor, actualizar Proveedor, borrar Proveedor), “Producto” en submenu de Producto(Agregar producto,ver producto, actualizar producto, borrar producto) “Cliente” en submenu de Cliente(Agregar cliente,ver cliente, actualizar cliente, borrar cliente), 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 Alumno Estefania Gonzalez, 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 paletteria.
- 21 Crear la subcarpeta carpeta **proveedor** dentro de **app\_Paleteria\templates**.
- 22 crear los archivos html con su codigo correspondientes de (agregar\_proveedor.html, ver\_proveedor.html mostrar en tabla con los botones ver, editar y borrar, actualizar\_proveedor.html, borrar\_proveedor.html) dentro de **app\_Paleteria\templates\proveedor**.
- 23 No utilizar forms.py.
- 24 procedimiento para crear el archivo urls.py en **app\_Paleteria** con el código correspondiente para acceder a las funciones de [views.py](#) para operaciones de crud en proveedor.
- 25 procedimiento para agregar **app\_Paleteria** en settings.py de **backend\_Paleteria**
- 26 realizar las configuraciones correspondiente a urls.py de **backend\_Paleteria** para enlazar con **app\_Paleteria**
- **27 procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.**
- 27 por lo pronto solo trabajar con “proveedor” dejar pendiente # MODELO: PRODUCTO y # MODELO: CLIENTE
- 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**.