

Primera parte

Proyecto: **Coca Cola**

Lenguaje: **Python**

Framework: **Django**

Editor: **VS Code**

- 1 Procedimiento para crear carpeta del Proyecto: **UIII_Cocacola_1299**
- 2 Procedimiento para abrir VS Code sobre la carpeta **UIII_Cocacola_1299**
- 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_cocacola 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_cocacola**
- 12 Aquí el modelo models.py

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

```
# =====
# MODELO: CLIENTE
# =====
class Cliente(models.Model):
    nombre = models.CharField(max_length=50)
    apellido = models.CharField(max_length=50)
    direccion = models.CharField(max_length=255)
    telefono = models.CharField(max_length=20)
    email = models.EmailField(max_length=100)
    fecha_registro = models.DateField(auto_now_add=True)
    preferencia = models.CharField(max_length=100, blank=True, null=True)

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

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

```
# MODELO: PROVEEDOR
# =====
class Proveedor(models.Model):
    nombre_empresa = models.CharField(max_length=100)
    contacto = models.CharField(max_length=100)
    telefono = models.CharField(max_length=20)
    correo = models.EmailField(max_length=100)
    direccion = models.CharField(max_length=200)
```

```

pais = models.CharField(max_length=50)
tipo_producto_suministro = models.CharField(max_length=100)

def __str__(self):
    return self.nombre_empresa

=====
# MODELO: PRODUCTO
=====
class Producto(models.Model):
    nombre_producto = models.CharField(max_length=100)
    categoria = models.CharField(max_length=100)
    precio_unitario = models.DecimalField(max_digits=10, decimal_places=2)
    stock_actual = models.IntegerField()
    fecha_fabricacion = models.DateField()
    fecha_vencimiento = models.DateField()
    cliente = models.ForeignKey(Cliente, on_delete=models.CASCADE)
    proveedor = models.ForeignKey(Proveedor, on_delete=models.CASCADE)

    def __str__(self):
        return self.nombre_producto
=====
```

- 12.5 Procedimiento para realizar las migraciones (makemigrations y migrate)
- 13 Primero trabajamos con el MODELO: CLIENTE
- 14 En views.py de **app_cocacola** crear las funciones con sus códigos correspondientes (inicio_cocacola, agregar_cliente, actualizar_cliente, realizar_actualizacion_cliente, borrar_cliente)
- 15 Crear la carpeta **templates** dentro de **app_cocacola**
- 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 Coca Cola, Inicio, Cliente, en submenú de Cliente (Agregar Cliente, Ver Cliente, Actualizar Cliente, Borrar Cliente), Proveedor en submenú de Proveedor (Agregar Proveedor, Ver Proveedor, Actualizar Proveedor, Borrar Proveedor), Producto en submenú de Producto (Agregar Producto, Ver Producto, Actualizar Producto, Borrar Producto), incluir íconos a las opciones principales, no en los submenús)
- 19 En el archivo **footer.html** incluir derechos de autor, fecha del sistema y "Creado por Alumna Ximena Reyes, CBTis 128" y mantenerla fija al final de la página
- 20 En el archivo inicio.html colocar información del sistema y una imagen tomada desde la red sobre Coca Cola
- 21 Crear la subcarpeta **cliente** dentro de **app_cocacola/templates**
- 22 Crear los archivos html con su código correspondiente (agregar_cliente.html, ver_cliente.html mostrando tabla con botones Ver, Editar y Borrar, actualizar_cliente.html, borrar_cliente.html) dentro de **app_cocacola/templates/cliente**

23 No utilizar forms.py

24 Procedimiento para crear el archivo urls.py en **app_cocacola** con el código correspondiente para acceder a las funciones de views.py para operaciones CRUD en clientes

25 Procedimiento para agregar **app_cocacola** en settings.py de **backend_cocacola**

26 Realizar las configuraciones correspondientes a urls.py de **backend_cocacola** para enlazar con **app_cocacola**

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

28 Por lo pronto solo trabajar con Cliente, dejar pendiente Proveedor y Producto

29 Utilizar colores suaves, atractivos y modernos, con código HTML sencillo

30 No validar entrada de datos

31 Al inicio **crear la estructura completa** de carpetas y archivos

32 Proyecto totalmente funcional

33 Finalmente ejecutar servidor en el **puerto 8036**