Primera parte
Proyecto: **BYD**.
lenguaje: Python.
Framework: Django.
Editor: VS code.
1 Procedimiento para crear carpeta del Proyecto: UIII_BYD_0478
2 procedimiento para abrir vs code sobre la carpeta
UIII_BYD_0478
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_Byd sin duplicar
carpeta.
9 procedimiento para ejecutar servidor en el puerto 8047
10 procedimiento para copiar y pegar el link en el navegador.
11 procedimiento para crear aplicacion app_Proveedor
12 Aqui el modelo models.py
=-=-=-=-=-

```
from django.db import models

# --- 1. Modelo Proveedor (7 campos) ---
class Proveedor(models.Model):
    # Campo 1 (PK ID) no se cuenta, pero existe.
    nombre_empresa = models.CharField(max_length=100, unique=True,
verbose_name="Nombre de la Empresa") # 1
    contacto_principal = models.CharField(max_length=100, verbose_name="Contacto
Principal") # 2
    telefono = models.CharField(max_length=15, blank=True, null=True) # 3
    email = models.EmailField(unique=True) # 4
    direccion = models.CharField(max_length=200, verbose_name="Dirección") # 5
    fecha_registro = models.DateTimeField(auto_now_add=True, verbose_name="Fecha de
Registro") # 6
    activo = models.BooleanField(default=True) # 7

    class Meta:
        verbose_name = "Proveedor"
        verbose_name_plural = "Proveedores"
        ordering = ['nombre_empresa']

    def __str__(self):
        return self.nombre_empresa

# --- 2. Modelo Distribuidor (7 campos) ---
class Distribuidor(models.Model):
    # Campo 1 (PK ID) no se cuenta, pero existe.
```

```
    nombre_distribuidor = models.CharField(max_length=100, unique=True,
verbose_name="Nombre del Distribuidor") # 1
    tipo_servicio = models.CharField(max_length=50, choices=[('LOCAL', 'Local'),
('NACIONAL', 'Nacional'), ('INTERNACIONAL', 'Internacional')]) # 2
    ciudad = models.CharField(max_length=50) # 3
    pais = models.CharField(max_length=50) # 4
    tiempo_entrega_dias = models.IntegerField(verbose_name="Tiempo de Entrega (días)")
# 5
    comision = models.DecimalField(max_digits=5, decimal_places=2, help_text="Comisión
en porcentaje") # 6
    ultima_revision = models.DateField(auto_now=True, verbose_name="Última Revisión") #
7

    class Meta:
        verbose_name = "Distribuidor"
        verbose_name_plural = "Distribuidores"
        ordering = ['nombre_distribuidor']

    def __str__(self):
        return self.nombre_distribuidor

# --- 3. Modelo Producto (7 campos, incluyendo las relaciones) ---
class Producto(models.Model):
    # 1. Relación Uno a Muchos (FK) - Campo 1
    # Se traduce a la columna id_proveedor en la BD, pero en Django se accede como
'producto.proveedor'
    proveedor = models.ForeignKey(
        Proveedor,
        on_delete=models.CASCADE,
        related_name='productos_suministrados',
        verbose_name="ID Proveedor" # Para referencia en la Interfaz/Admin
    )

    # 2. Relación Muchos a Muchos - Campo 2
    # No genera una columna simple, usa una tabla intermedia
    distribuidores = models.ManyToManyField(
        Distribuidor,
        related_name='productos_distribuidos',
        verbose_name="Distribuidores"
    )

    # 3. Campos de Atributo restantes (5 campos)
    nombre = models.CharField(max_length=150, verbose_name="Nombre del Producto") #
3
    sku = models.CharField(max_length=50, unique=True, verbose_name="SKU/Código") # 4
    precio = models.DecimalField(max_digits=10, decimal_places=2) # 5
    stock_actual = models.IntegerField(verbose_name="Stock Actual") # 6
    descripcion = models.TextField(blank=True, verbose_name="Descripción") # 7
```

```
    class Meta:
        verbose_name = "Producto"
        verbose_name_plural = "Productos"
        ordering = ['nombre']

    def __str__(self):
        return self.nombre
```

=-=-=-=-=-

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: PROVEEDOR

14 En view de app_Proveedor crear las funciones con sus códigos correspondientes (inicio_proveedor, agregar_proveedor, actualizar_proveedor, realizar_actualizacion_proveedor, borrar_proveedor)

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

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 BYD”, “Inicio”, “Proveedor”,en submenu de proveedor (Agregar Proveedor, ver Proveedor, actualizar Proveedor, borrar proveedor), “Distribuidor” en submenu de Distribuidor (Aregar Distribuidor, ver Distribuidor, actualizar Distribuidor, borrar Distribuidor)
“Producto” en submenu de Producto(Agregar Producto,ver Producto, actualizar Producto, borrar Producto), 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 Arroyo Carlos, 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 byd.

21 Crear la subcarpeta carpeta proveedor dentro de app_Proveedor\templates.

22 crear los archivos html con su codigo correspondientes de (agregar_proveedor.html, ver_proveedores.html mostrar en tabla con los botones ver, editar y borrar, actualizar_proveedor.html, borrar_proveedor.html)
dentro de app_Proveedor\templates\proveedor.

23 No utilizar forms.py.

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

25 procedimiento para agregar app_Proveedor en settings.py de backend_Byd

26 realizar las configuraciones correspondiente a urls.py de

backend_Byd para enlazar con app_Proveedor

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: Distribuidor y # MODELO: Producto
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 8047.