

# Primera parte

Proyecto: **Venta\_Teléfonos.**

Lenguaje: **Python.**

Framework: **Django.**

Editor: **VS code.**

1 Procedimiento para crear carpeta del Proyecto: **UIII\_Venta\_Teléfonos\_0619**

2 procedimiento para abrir vs code sobre la carpeta

**UIII\_Venta\_Telefonos\_0619**

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\_Venta\_Telefonos** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el **puerto 8017**

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

11 procedimiento para crear aplicación app\_Venta\_Telefonos

12 Aquí el modelo models.py

=====

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

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

```
# class Proveedor(models.Model):
```

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

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

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

```
    telefono = models.CharField(max_length=20)
```

```
    email = models.EmailField()
```

```
    direccion = models.CharField(max_length=200)
```

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

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

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

```
        return f'{self.nombre} ({self.empresa})'
```

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

```
# class Producto(models.Model):
```

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

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

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

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

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

```

stock = models.PositiveIntegerField()
descripcion = models.TextField(blank=True, null=True)
fecha_ingreso = models.DateField(auto_now_add=True)

# Relación muchos a muchos con Producto_Proveedor
productos_proveedor = models.ManyToManyField('Producto_Proveedor',
related_name='productos')

def __str__(self):
    return f'{self.nombre} - {self.marca} ({self.modelo})'

# =====

# class Producto_Proveedor(models.Model):
# =====
nombre_producto = models.CharField(max_length=100)
marca = models.CharField(max_length=50)
modelo = models.CharField(max_length=50)
precio_compra = models.DecimalField(max_digits=10, decimal_places=2)
cantidad_disponible = models.PositiveIntegerField()
garantia_meses = models.PositiveIntegerField(default=12)
fecha_entrega = models.DateField()

# Relación 1 a muchos con Proveedor
proveedor = models.ForeignKey(Proveedor, on_delete=models.CASCADE,
related_name='productos_proveedor')

def __str__(self):
    return f'{self.marca} {self.modelo} ({self.proveedor.empresa})'

# =====

# Relación 1 a muchos: una sala puede tener varias películas proyectadas
categorias = models.ManyToManyField(Categoría, related_name="películas")
# Relación muchos a muchos: una película puede pertenecer a varias categorías
def __str__(self):
    return self.título
=====
```

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

13 Primero trabajamos con el MODELO: **Proveedor**.

14 En **views.py** de **app\_Venta\_Teléfonos** crear las funciones con sus códigos correspondientes: (inicio\_ventas, agregar\_proveedor, actualizar\_proveedor, realizar\_actualizacion\_proveedor, borrar\_proveedor).

15 Crear la carpeta “templates” dentro de “**app\_Venta\_Teléfonos**”..

- 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 Venta de Teléfonos”, “Inicio”, “Proveedores” (en submenú de Proveedores: Agregar Proveedor, Ver Proveedores, Actualizar Proveedor, Borrar Proveedor), “Productos” (en submenú de Productos: Agregar Producto, Ver Productos, Actualizar Productos, Borrar Productos), “Relación Producto-Proveedor” (en submenú: Agregar Relación, Ver Relaciones, Actualizar Relación, Borrar Relación). Incluir iconos a las opciones principales, no en los submenús.
- 19 En el archivo **footer.html** incluir derechos de autor, fecha del sistema y “Creado por Ing. Sebastián Lara, 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 **venta de teléfonos**
- 21 Crear las subcarpetas **proveedor**, **producto** y **producto\_proveedor** dentro de **app\_Venta\_Teléfonos\templates**.
- 22 Crear los archivos html con su código 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\_Venta\_Teléfonos\templates\proveedor**.
- 23 No utilizar **forms.py**.
- 24 Procedimiento para crear el **archivo urls.py** en **app\_Venta\_Teléfonos** con el código correspondiente para acceder a las funciones de **views.py** para operaciones de CRUD en **proveedores, productos y producto\_proveedor**.
- 25 Procedimiento para agregar **app\_Venta\_Teléfonos** en **settings.py** de **backend\_Venta\_Teléfonos**.
- 26 Realizar las configuraciones correspondiente a **urls.py** de **backend\_Venta\_Teléfonos** para enlazar con **app\_Venta\_Teléfonos**.
- 27 procedimiento para registrar los modelos en **admin.py** y volver a realizar las migraciones.
- 27 Por lo pronto solo trabajar con **Proveedor, Producto y Producto\_Proveedor**.
- 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 **8017**.