

Primera parte

Proyecto: Elektra.

lenguaje: Python.

Framework: Django.

Editor: VS code.

1 Procedimiento para crear carpeta del Proyecto: UIII_Elektra_0644

2 procedimiento para abrir vs code sobre la carpeta

UIII_Elektra_0644

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_Elektra sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto 8029

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

11 procedimiento para crear aplicacion app_Elektra

12 Aqui el modelo models.py

==--==--==--

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

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

```
# MODELO: PRODUCTOS
```

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

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

```
    nombre = models.CharField(max_length=50, unique=True)
```

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

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

```
    stock = models.IntegerField()
```

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

```
    proveedor = models.ForeignKey(Proveedor, on_delete=models.CASCADE)
```

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

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

```
        return self.nombre
```

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

```
# MODELO: PROVEEDORES
```

```
# =====
class Proveedor(models.Model):
    nombre = models.CharField(max_length=100)
    pais = models.CharField(max_length=50)
    direccion = models.TextField()
    telefono = models.CharField(max_length=20)
    email = models.EmailField(max_length=100, unique=True)
    fecha_registro = models.DateField(auto_now_add=True)
    activo = models.BooleanField(default=True)

    def __str__(self):
        return self.nombre

# =====
# MODELO: VENTAS
# =====
class Venta(models.Model):
    folio = models.CharField(max_length=50, unique=True)
    fecha_venta = models.DateTimeField(default=timezone.now)
    total = models.DecimalField(max_digits=10, decimal_places=2)
    metodo_pago = models.CharField(max_length=50)
    estado = models.CharField(max_length=50)
    vendedor_id = models.IntegerField()
    id_producto = models.ForeignKey(Producto, on_delete=models.CASCADE)

    def __str__(self):
        return f"Venta {self.folio}"
```

=====

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: PRODUCTOS

14 En view de app_Elektra crear las funciones con sus códigos correspondientes (inicio_elektra, agregar_producto,

actualizar_producto, realizar_actualizacion_producto, borrar_producto)

15 Crear la carpeta "templates" dentro de "app_Elektra".

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 Elektra”, “Inicio”, “producto”, en submenú de categorías (Agregar producto, ver producto, actualizar producto, borrar producto), “Proveedores” en submenú de Salas (Agregar proveedores, ver proveedores, actualizar proveedores, borrar proveedores), “Ventas” en submenú de Ventas (Agregar ventas, ver ventas, actualizar ventas, borrar ventas), 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. Alexis Mireles, 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 elektra.

21 Crear la subcarpeta carpeta productos dentro de app_Elektra\templates.

22 crear los archivos html con su código correspondientes de (agregar_producto.html, ver_producto.html mostrar en tabla con los botones ver, editar y borrar, actualizar_producto.html, borrar_producto.html) dentro de app_Elektra\templates\productos.

23 No utilizar forms.py.

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

25 procedimiento para agregar app_Elektra en settings.py de backend_Elektra

26 realizar las configuraciones correspondiente a urls.py de backend_Elektra para enlazar con app_Elektra

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

27 por lo pronto solo trabajar con “productos” dejar pendiente # MODELO: PROVEEDORES y # MODELO: VENTAS

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 8029.