

Primera parte

Proyecto: **Costco.**

lenguaje: **Python.**

Framework: **Django.**

Editor: **VS code.**

1 Procedimiento para crear carpeta del Proyecto: **Ull_Costco_0599**

2 procedimiento para abrir vs code sobre la carpeta

Ull_Costco_0599

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_Costco** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el **puerto 8018**

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

11 procedimiento para crear aplicacion **app_Costco**

12 Aqui el modelo [models.py](#)

=====

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

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

```
    # Campos de Usuario
```

```
    nombre_usuario = models.CharField(max_length=100, unique=True)
```

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

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

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

```
    direccion = models.TextField()
```

```
    telefono = models.CharField(max_length=20, blank=True, null=True)
```

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

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

```
        return self.nombre_usuario
```

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

```
    # Campos de Producto
```

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

```
    descripcion = models.TextField()
```

```

precio = models.DecimalField(max_digits=10, decimal_places=2)
stock = models.PositiveIntegerField(default=0)
categoria = models.CharField(max_length=100, blank=True, null=True)
codigo_barras = models.CharField(max_length=50, unique=True, blank=True, null=True)
fecha_creacion = models.DateTimeField(auto_now_add=True)

def __str__(self):
    return self.nombre

class Pedido(models.Model):
    # Relación uno a muchos: Un Usuario puede tener muchos Pedidos
    usuario = models.ForeignKey(Usuario, on_delete=models.CASCADE,
related_name='pedidos')

    # Campos de Pedido
    fecha_pedido = models.DateTimeField(auto_now_add=True)
    estado_pedido = models.CharField(max_length=50, default='Pendiente')
    direccion_envio = models.TextField()
    total_pedido = models.DecimalField(max_digits=10, decimal_places=2, default=0.00)
    metodo_pago = models.CharField(max_length=50)
    fecha_entrega_estimada = models.DateField(blank=True, null=True)
    numero_seguimiento = models.CharField(max_length=100, blank=True, null=True)

    # Relación muchos a muchos: Un Pedido puede tener muchos Productos y un Producto
puede estar en muchos Pedidos
    productos = models.ManyToManyField(Producto, related_name='pedidos')

    def __str__(self):
        return f"Pedido #{self.id} de {self.usuario.nombre_usuario}"

```

==--==--==

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

13 primero trabajamos con el MODELO: USUARIO

14 En view de **app_Costco** crear las funciones con sus códigos correspondientes (inicio_costco, agregar_usuario, actualizar_usuario, realizar_actualizacion_categoria, borrar_usuario)

15 Crear la carpeta "**templates**" dentro de "**app_Costco**".

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 Costco”, “Inicio”, “Usuarios”, en submenu de usuarios (Aregar Usuario, ver Usuario, actualizar Usuario, borrar Usuario), “Productos” en submenu de productos(Agregar producto, ver producto, actualizar producto, borrar producto) “Pedidos” en submenu de Pedidos(Aregar pedido,ver pedido, actualizar pedido, borrar pedido), 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. Hernandez Roman, 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 Costco.
- 21 Crear la subcarpeta carpeta **usuario** dentro de **app_Costco\templates**.
- 22 crear los archivos html con su código correspondientes de (agregar_usuario.html, ver_usuario.html mostrar en tabla con los botones ver, editar y borrar, actualizar_usuario.html, borrar_usuario.html) dentro de **app_Costco\templates\usuario**.
- 23 No utilizar [forms.py](#).
- 24 procedimiento para crear el archivo [urls.py](#) en **app_Costco** con el código correspondiente para acceder a las funciones de [views.py](#) para operaciones de crud en usuario
- 25 procedimiento para agregar **app_Costco** en [settings.py](#) de **backend_Costco**
- 26** realizar las configuraciones correspondiente a [urls.py](#) de **backend_Costco** para enlazar con **app_Costco**
- 27 procedimiento para registrar los modelos en [admin.py](#) y volver a realizar las migraciones.**
- 27 por lo pronto solo trabajar con “usuario” dejar pendiente #
MODELO: PRODUCTO y # MODELO: PEDIDO
- 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 8018**.