

**Primera parte Proyecto:** In-N-Out. **Lenguaje:** Python. **Framework:** Django. **Editor:** VS code.

1. Procedimiento para crear carpeta del Proyecto: **Ulll\_in\_n\_out\_0421**
2. Procedimiento para abrir vs code sobre la carpeta **Ulll\_in\_n\_out\_0421**
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\_in\_n\_out** sin duplicar carpeta.
9. Procedimiento para ejecutar servidor en el puerto **8421**
10. Procedimiento para copiar y pegar el link en el navegador.
11. Procedimiento para crear aplicación **app\_in\_n\_out**

Aqui el modelo models.py

Python

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

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

```
# MODELO: TRABAJADOR
```

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

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

```
    # 7 campos (ID es automático)
```

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

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

```
    puesto = models.CharField(max_length=50, help_text="Ej: Cajero, Cocinero")
```

```
    fecha_contratacion = models.DateField()
```

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

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

```
    activo = models.BooleanField(default=True)
```

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

```
        return f"{self.nombre} {self.apellido}"
```

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

```
# MODELO: PRODUCTO
```

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

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

```
    # 7 campos (ID es automático)
```

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

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

```
    precio = models.DecimalField(max_digits=6, decimal_places=2)
```

```
    categoria = models.CharField(max_length=50, help_text="Ej: Hamburguesa, Bebida")
```

```
    disponible = models.BooleanField(default=True)
```

```
    calorias = models.IntegerField(null=True, blank=True)
```

```
    imagen_url = models.URLField(max_length=255, blank=True, null=True)
```

```

def __str__(self):
    return self.nombre_producto

# =====
# MODELO: CLIENTE
# =====
class Cliente(models.Model):
    # 7 campos (ID automático + 5 campos de datos + 1 FK + 1 M2M)
    nombre = models.CharField(max_length=100)
    apellido = models.CharField(max_length=100)
    email = models.EmailField(unique=True)
    fecha_registro = models.DateField(auto_now_add=True)
    telefono = models.CharField(max_length=20, blank=True, null=True)

    # --- Relación 1:M ---
    # Un Trabajador puede atender/registrarse a muchos Clientes
    trabajador_asignado = models.ForeignKey(
        Trabajador,
        on_delete=models.SET_NULL, # Si se borra el trabajador, el cliente queda 'sin asignar'
        related_name="clientes_atendidos",
        null=True,
        blank=True
    )

    # --- Relación M:M ---
    # Un Cliente puede tener muchos Productos favoritos
    productos_favoritos = models.ManyToManyField(
        Producto,
        related_name="clientes_que_lo_prefieren",
        blank=True # Un cliente puede no tener favoritos
    )

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

```

- 12.
13. Procedimiento para realizar las migraciones (makemigrations y migrate).
14. Primero trabajamos con el **MODELO: PRODUCTO**
15. En **views.py** de **app\_in\_n\_out** crear las funciones con sus códigos correspondientes (**inicio\_in\_n\_out**, **agregar\_producto**, **actualizar\_producto**, **realizar\_actualizacion\_producto**, **borrar\_producto**)
16. Crear la carpeta "templates" dentro de "app\_in\_n\_out".
17. En la carpeta templates crear los archivos html (base.html, header.html, navbar.html, footer.html, inicio.html).
18. En el archivo base.html agregar bootstrap para css y js.

19. En el archivo navbar.html incluir las opciones ( “Sistema de Administración **In-N-Out**”, “Inicio”, “**Producto**”, en submenu de **Producto**(Agregar Producto, Ver Producto, Actualizar Producto, Borrar Producto), “**Cientes**” en submenu de **Cientes**(Agregar Cliente, Ver Clientes, Actualizar Cliente, Borrar Cliente), “**Trabajadores**” en submenu de **Trabajadores**(Agregar Trabajador, Ver Trabajador, Actualizar Trabajador, Borrar Trabajador), incluir iconos a las opciones principales, no en los submenu.
20. En el archivo footer.html incluir derechos de autor, fecha del sistema y “Creado por Ing. Eliseo Nava, Cbtis 128” y mantenerla fija al final de la página.
21. En el archivo inicio.html se usa para colocar información del sistema más una imagen tomada desde la red sobre **In-N-Out**.
22. Crear la subcarpeta **producto** dentro de **app\_in\_n\_out\templates**.
23. Crear los archivos html con su código correspondientes de (**agregar\_producto.html**, **ver\_productos.html** mostrar en tabla con los botones ver, editar y borrar, **actualizar\_producto.html**, **borrar\_producto.html**) dentro de **app\_in\_n\_out\templates\producto**.
24. No utilizar forms.py.
25. Procedimiento para crear el archivo **urls.py** en **app\_in\_n\_out** con el código correspondiente para acceder a las funciones de **views.py** para operaciones de crud en **productos**.
26. Procedimiento para agregar **app\_in\_n\_out** en **settings.py** de **backend\_in\_n\_out**
27. Realizar las configuraciones correspondiente a **urls.py** de **backend\_in\_n\_out** para enlazar con **app\_in\_n\_out**
28. Procedimiento para registrar los modelos en **admin.py** y volver a realizar las migraciones.
29. Por lo pronto solo trabajar con “**Producto**” dejar pendiente # MODELO: CLIENTE y # MODELO: TRABAJADOR
30. Utilizar colores suaves, atractivos y modernos, el código de las páginas web sencillas.
31. No validar entrada de datos.
32. Al inicio crear la estructura completa de carpetas y archivos.
33. Proyecto totalmente funcional.
34. Finalmente ejecutar servidor en el puerto **8421**.