

Primera parte

Proyecto: In-N-Out

Lenguaje: Python

Framework: Django

Editor: VS Code

- 1 Procedimiento para crear carpeta del Proyecto: UIII_InNOut_0421
- 2 Procedimiento para abrir VS Code sobre la carpeta UIII_InNOut_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_InNOut 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_InNOut

12 Aquí el modelo models.py

=====

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

MODELO: SUCURSAL

```
class Sucursal(models.Model):  
    nombre_tienda = models.CharField(max_length=100)  
    direccion = models.CharField(max_length=255, blank=True)  
    ciudad = models.CharField(max_length=100)  
    codigo_postal = models.CharField(max_length=10, blank=True)  
    telefono_tienda = models.CharField(max_length=20, blank=True)  
    fecha_apertura = models.DateField(null=True, blank=True)  
    es_drive_thru = models.BooleanField(default=True)
```

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

```
return self.nombre_tienda
```



```
=====
```



```
=====
```

MODELO: TRABAJADOR


```
=====
```



```
=====
```

```
class Trabajador(models.Model):  
    nombre = models.CharField(max_length=100)  
    apellido = models.CharField(max_length=100)  
    puesto = models.CharField(max_length=50)  
    fecha_contratacion = models.DateField()  
    email = models.EmailField(unique=True)  
    telefono_personal = models.CharField(max_length=20, blank=True)  
    sucursal = models.ForeignKey(Sucursal, on_delete=models.CASCADE,  
        related_name="trabajadores")  
  
    def __str__(self):  
        return f'{self.nombre} {self.apellido}'
```



```
=====
```



```
=====
```

MODELO: PEDIDO


```
=====
```



```
=====
```

```

class Pedido(models.Model):
    fecha_pedido = models.DateTimeField(auto_now_add=True)
    numero_orden = models.PositiveIntegerField()
    total_pedido = models.DecimalField(max_digits=7, decimal_places=2)
    tipo_orden = models.CharField(max_length=50, help_text="Comer ahí o Para llevar")
    nombre_cliente_temporal = models.CharField(max_length=100, blank=True)
    pagado = models.BooleanField(default=False)
    trabajador = models.ForeignKey(Trabajador, on_delete=models.SET_NULL, null=True,
                                   blank=True, related_name="pedidos")

def __str__(self):
    return f"Pedido #{self.numero_orden}"

```

=====

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

13 Primero trabajamos con el MODELO: SUCURSAL.

14 En views.py de app_InNOut crear las funciones con sus códigos correspondientes (inicio_in_n_out, agregar_sucursal, actualizar_sucursal, realizar_actualizacion_sucursal, borrar_sucursal).

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

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 In-N-Out”, “Inicio”, “Sucursal”, en submenu de Sucursal (Aregar Sucursal, Ver Sucursales, Actualizar Sucursal, Borrar Sucursal), “Trabajadores” en submenu de Trabajadores (Aregar Trabajador, Ver Trabajadores, Actualizar Trabajador, Borrar Trabajador), “Pedidos” en submenu de Pedidos (Aregar Pedido, Ver Pedidos, Actualizar Pedido, Borrar Pedido), incluir íconos 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. Santiago Carrasco, 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 In-N-Out.

21 Crear la subcarpeta sucursal dentro de app_InNOut\templates.

22 Crear los archivos html con su código correspondiente de (agregar_sucursal.html, ver_sucursales.html mostrando en tabla con los botones ver, editar y borrar, actualizar_sucursal.html, borrar_sucursal.html) dentro de app_InNOut\templates\sucursal.

23 No utilizar forms.py.

24 Procedimiento para crear el archivo urls.py en app_InNOut con el código correspondiente para acceder a las funciones de views.py para operaciones CRUD en sucursales.

25 Procedimiento para agregar app_InNOut en settings.py de backend_InNOut.

26 Realizar las configuraciones correspondientes a urls.py de backend_InNOut para enlazar con app_InNOut.

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

27 Por lo pronto solo trabajar con “Sucursal”, dejar pendiente # MODELO: TRABAJADOR 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 8421.