Primera parte
Proyecto: Sistema de Autobuses
Lenguaje: Python.
Framework: Django.
Editor: VS code.

1 Procedimiento para crear carpeta del Proyecto: UIII_Autobuses_0777

2 procedimiento para abrir vs code sobre la carpeta UIII_Autobuses_0777

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_Autobuses sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto 8036

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

11 procedimiento para crear aplicacion app_Autobuses

12 Aqui el modelo models.py
=-=-=-=-=-

```
python
from django.db import models

# ========================================
# MODELO: AUTOBUS
# ========================================
class Autobus(models.Model):
    TIPOS_SERVICIO = [
        ('Ejecutivo', 'Ejecutivo'),
        ('Económico', 'Económico'),
        ('Lujo', 'Lujo'),
        ('Primera Clase', 'Primera Clase'),
    ]
```

```
    ESTADOS = [
        ('Activo', 'Activo'),
        ('Mantenimiento', 'Mantenimiento'),
        ('Inactivo', 'Inactivo'),
    ]

    placa = models.CharField(max_length=10, unique=True)
    marca = models.CharField(max_length=50)
    modelo = models.CharField(max_length=50)
    capacidad_pasajeros = models.IntegerField()
    tipo_servicio = models.CharField(max_length=20, choices=TIPOS_SERVICIO)
    año_fabricacion = models.IntegerField()
    estado = models.CharField(max_length=20, choices=ESTADOS,
default='Activo')

    def __str__(self):
        return f"{self.placa} - {self.marca} {self.modelo}"

# ========================================
# MODELO: RUTA
# ========================================
class Ruta(models.Model):
    clave_ruta = models.CharField(max_length=10, unique=True)
    nombre_ruta = models.CharField(max_length=100)
    terminal_origen = models.CharField(max_length=100)
    terminal_destino = models.CharField(max_length=100)
    distancia_km = models.DecimalField(max_digits=8, decimal_places=2)
    duracion_estimada_min = models.IntegerField()
    precio_base = models.DecimalField(max_digits=8, decimal_places=2)
    autobus = models.ForeignKey(Autobus, on_delete=models.CASCADE,
related_name="rutas")
    # Relación 1 a muchos: un autobús puede tener varias rutas asignadas

    def __str__(self):
        return f"{self.clave_ruta} - {self.terminal_origen} a
{self.terminal_destino}"

# ========================================
# MODELO: EMPLEADO
# ========================================
class Empleado(models.Model):
    PUESTOS = [
        ('Conductor', 'Conductor'),
        ('Ayudante', 'Ayudante'),
        ('Jefe de Terminal', 'Jefe de Terminal'),
```

```
        ('Vendedor de Boletos', 'Vendedor de Boletos'),
        ('Administrativo', 'Administrativo'),
        ('Mantenimiento', 'Mantenimiento'),
    ]

    nombre = models.CharField(max_length=50)
    apellido_paterno = models.CharField(max_length=50)
    apellido_materno = models.CharField(max_length=50)
    rfc = models.CharField(max_length=13, unique=True)
    puesto = models.CharField(max_length=30, choices=PUESTOS)
    fecha_contratacion = models.DateField()
    telefono = models.CharField(max_length=15)

    # Relación muchos a muchos: un empleado puede trabajar en varias rutas
    rutas = models.ManyToManyField(Ruta, related_name="empleados")

    def __str__(self):
        return f"{self.nombre} {self.apellido_paterno}
{self.apellido_materno}"
```

=-=-=-=-=-

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: AUTOBUS

14 En view de app_Autobuses crear las funciones con sus códigos correspondientes (inicio_autobuses, agregar_autobus, actualizar_autobus, realizar_actualizacion_autobus, borrar_autobus)

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

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.

**