

Primera parte

Proyecto: Ford
Lenguaje: Python
Framework: Django
Editor: VS Code

Procedimiento para crear carpeta del Proyecto: UIII_Ford_0493

Procedimiento para abrir VS Code sobre la carpeta UIII_Ford_0493

Procedimiento para abrir terminal en VS Code

Procedimiento para crear carpeta entorno virtual “.venv” desde terminal de VS Code

Procedimiento para activar el entorno virtual

Procedimiento para activar intérprete de Python

Procedimiento para instalar Django

Procedimiento para crear proyecto backend_Ford sin duplicar carpeta

Procedimiento para ejecutar servidor en el puerto 8093

Procedimiento para copiar y pegar el link en el navegador

Procedimiento para crear aplicación app_Ford

12. Aquí el modelo models.py
from django.db import models

=====

=====

MODELO: EMPLEADO

=====

=====

```

class Empleado(models.Model):
    nombre = models.CharField(max_length=100)
    apellido = models.CharField(max_length=100)
    puesto = models.CharField(max_length=100)
    telefono = models.CharField(max_length=30, blank=True, null=True)
    email = models.EmailField(blank=True, null=True)
    fecha_contratacion = models.DateField(blank=True, null=True)
    salario = models.DecimalField(max_digits=10, decimal_places=2, blank=True, null=True)

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

```

=====

=====

MODELO: VEHÍCULO

=====

=====

```

class Vehiculo(models.Model):
    marca = models.CharField(max_length=100)
    modelo = models.CharField(max_length=100)
    anio = models.PositiveIntegerField()
    color = models.CharField(max_length=50, blank=True, null=True)
    numero_serie = models.CharField(max_length=100, unique=True)
    precio = models.DecimalField(max_digits=12, decimal_places=2)
    cantidad_disponible = models.PositiveIntegerField(default=1)

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

```

=====

=====

MODELO: VENTA

=====

=====

```
class Venta(models.Model):
    vehiculo = models.ForeignKey(Vehiculo, on_delete=models.CASCADE,
    related_name='ventas')
    empleado = models.ForeignKey(Empleado, on_delete=models.SET_NULL, null=True,
    blank=True, related_name='ventas')
    cliente_nombre = models.CharField(max_length=150)
    cliente_telefono = models.CharField(max_length=50, blank=True, null=True)
    fecha_venta = models.DateField(auto_now_add=True)
    total = models.DecimalField(max_digits=12, decimal_places=2)
    metodo_pago = models.CharField(max_length=50, blank=True, null=True)
    folio = models.CharField(max_length=100, blank=True, null=True)

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

12.5 Procedimiento para realizar las migraciones (makemigrations y migrate)

13 Primero trabajamos con el MODELO: VEHÍCULO

14 En views de app_Ford crear las funciones:

inicio_ford, agregar_vehiculo, ver_vehiculos, actualizar_vehiculo,
realizar_actualizacion_vehiculo, borrar_vehiculo

15 Crear la carpeta “templates” dentro de “app_Ford”

16 En la carpeta templates crear los archivos html:

base.html, header.html, navbar.html, footer.html, inicio.html

17 En base.html agregar bootstrap para css y js

18 En navbar.html incluir las opciones:

“Sistema de Administración Ford”

“Inicio”

“Vehículos” (Agregar vehículo, Ver vehículos, Actualizar vehículo, Borrar vehículo)

“Empleados” (solo menú)

“Ventas” (solo menú)

Incluir iconos únicamente en las opciones principales

19 En footer.html incluir:

Derechos de autor

Fecha del sistema

“Creado por Ing. Emiliano Avilez, Cbtis 128”

Mantener fijo al final de la página

20 En inicio.html colocar información del sistema más una imagen desde internet sobre Ford

21 Crear la subcarpeta “vehiculos” dentro de app_Ford/templates

22 Crear los archivos html en app_Ford/templates/vehiculos:

agregar_vehiculo.html

ver_vehiculos.html (tabla con ver, editar, borrar)

actualizar_vehiculo.html

borrar_vehiculo.html

23 No utilizar forms.py

24 Crear archivo urls.py en app_Ford para enlazar funciones de views.py para CRUD de vehículos

25 Agregar app_Ford en settings.py de backend_Ford

26 Configurar urls.py de backend_Ford para enlazar con app_Ford

27 Registrar los modelos en admin.py y volver a realizar las migraciones

Trabajar solo con “vehículo”, dejar pendiente empleado y venta

28 Utilizar colores suaves, modernos y código sencillo

28 No validar entrada de datos

29 Crear la estructura completa de carpetas y archivos desde el inicio

30 Proyecto totalmente funcional

31 Finalmente ejecutar servidor en el puerto 8093