Primera parte
Proyecto: Construccion.
lenguaje: Python.
Framework: Django.
Editor: VS code.
1 Procedimiento para crear carpeta del Proyecto: UIII_Construccion_0335
2 procedimiento para abrir vs code sobre la carpeta
UIII_Consruccion_0335
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_Construccion sin duplicar
carpeta.
9 procedimiento para ejecutar servidor en el puerto 8032
10 procedimiento para copiar y pegar el link en el navegador.
11 procedimiento para crear aplicacion app_Construccion
12 Aqui el modelo models.py
=-=-=-=-=-

```
from django.db import models

# ========================================
# MODELO: Cliente
# ========================================
class Cliente(models.Model)
nombre = models.CharField(max_length=100)
    apellido = models.CharField(max_length=100)
    tipo_cliente = models.CharField(
        max_length=50,
        choices=[
            ("Residencial", "Residencial"),
            ("Comercial", "Comercial"),
            ("Gubernamental", "Gubernamental"),
        ],
        default="Residencial"
    )
    telefono = models.CharField(max_length=20)
    email = models.EmailField(unique=True)
    direccion = models.CharField(max_length=200)
    fecha_registro = models.DateField(auto_now_add=True)

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

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

# MODELO: Empleado
# ========================================
class Empleado(models.Model):
 nombre = models.CharField(max_length=100)
    apellido = models.CharField(max_length=100)
    cargo = models.CharField(max_length=100)
    telefono = models.CharField(max_length=20)
    email = models.EmailField(unique=True)
    fecha_contratacion = models.DateField()
    salario = models.DecimalField(max_digits=10, decimal_places=2)

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


# ========================================
# MODELO: Proyecto
# ========================================
class Proyecto(models.Model):
nombre = models.CharField(max_length=150)
    descripcion = models.TextField()
    fecha_inicio = models.DateField()
    fecha_fin = models.DateField()
    presupuesto = models.DecimalField(max_digits=12, decimal_places=2)
    cliente = models.ForeignKey(Cliente, on_delete=models.CASCADE,
related_name="proyectos")
    empleados = models.ManyToManyField(Empleado, related_name="proyectos")
    estado = models.CharField(
        max_length=50,
        choices=[
            ("En planificación", "En planificación"),
            ("En curso", "En curso"),
            ("Finalizado", "Finalizado"),
        ],
        default="En planificación",
    )

    def __str__(self):
        return self.nombre
```

=-=-=-=-=-

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: Empleado

14 En view de app_Construccion crear las funciones con sus códigos correspondientes (inicio_construccion, agregar_empleado,

actualizar_empleado, realizar_actualizacion_empleado,
borrar_empleado)
15 Crear la carpeta “templates” dentro de “app_Construccion”.
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 Construccion”, “Inicio”, “categoria”,en submenu de
empleados(Aregar empleado,ver empleado, actualizar empleado,
borrar empleado), “Proyectos” en submenu de proyectos(Aregar proyectos,ver
proyectos, actualizar proyectos, borrar proyectos)
“Clientes” en submenu de Clientes(Aregar clientes,ver clientes,
actualizar clientes, borrar clientes), 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 Tec. Angel Dario Rojas, 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 construccion.
21 Crear la subcarpeta carpeta empleado dentro de
app_Construccion\templates.
22 crear los archivos html con su codigo correspondientes de
(agregar_empleado.html, ver_empleado.html mostrar en tabla con
los botones ver, editar y borrar, actualizar_empleado.html,
borrar_empleado.html)
dentro de app_Construccion\templates\empleado.
23 No utilizar forms.py.
24 procedimiento para crear el archivo urls.py en app_Construccion con
el código correspondiente para acceder a las funciones de views.py
para operaciones de crud en empleados.
25 procedimiento para agregar app_Construccion en settings.py de
backend_Construccion
26 realizar las configuraciones correspondiente a urls.py de
backend_Construccion para enlazar con app_Construccion

27 procedimiento para registrar los modelos en admin.py y volver a
realizar las migraciones.
27 por lo pronto solo trabajar con “empleado” dejar pendiente #
MODELO: PROYECTO y # MODELO: CLIENTE
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 8032.