

Primera parte

Proyecto: **Parque de atracciones**

Lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

1 Procedimiento para crear carpeta del Proyecto: **UILL_parquedeatracciones_0248**

2 procedimiento para abrir vs code sobre la carpeta

UILL_parquedeatracciones_0248

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_parque** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el **puerto 8016**

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

11 procedimiento para crear aplicación **app_parque**

12 Aquí el modelo [**models.py**](#)

=====

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

```
# =====
# MODELO: EMPLEADO
# =====
class Empleado(models.Model):
    id_emp = models.AutoField(primary_key=True, unique=True)
    nombre = models.CharField(max_length=50)
    apellido = models.CharField(max_length=50)
    puesto = models.CharField(max_length=40)
    telefono = models.CharField(max_length=15)
    salario = models.DecimalField(max_digits=8, decimal_places=2)
    id_atr = models.IntegerField()

    def __str__(self):
        return self.nombre

# =====
# MODELO: CLIENTE
# =====
class Cliente(models.Model):
    id_cli = models.AutoField(primary_key=True, unique=True)
    nombre = models.CharField(max_length=50)
    apellido = models.CharField(max_length=50)
    telefono = models.CharField(max_length=15)
    correo = models.CharField(max_length=50)
    fecha_reg = models.DateField(auto_now_add=True)
```

```

id_atr = models.IntegerField()

def __str__(self):
    return self.nombre

# =====
# MODELO: ATRACCIÓN
# =====

class Atraccion(models.Model):
    id_atr = models.AutoField(primary_key=True, unique=True)
    nombre = models.CharField(max_length=50)
    tipo = models.CharField(max_length=50)
    capacidad = models.IntegerField()
    estado = models.CharField(max_length=100)
    altura_min = models.DecimalField(max_digits=3, decimal_places=2)
    id_emp = models.IntegerField()

    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_parque** crear las funciones con sus códigos correspondientes (inicio_parque, agregar_empleado, actualizar_empleado, realizar_actualizacion_empleado, borrar_empleado)

15 Crear la carpeta “**templates**” dentro de “**app_parque**”.

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 parque de atracciones”, “Inicio”, “empleado”, en submenu de empleados(Agregar empleado,ver empleado, actualizar empleado, borrar empleado), “Clientes” en submenu de Clientes(Aregar cliente,ver cliente, actualizar cliente, borrar cliente)

“Atracciones” en submenu de Atracciones(Agregar atracciones,ver atracciones, actualizar atracciones, borrar atracciones), 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 Tecnico en programacion Lidia Hinojos, 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 parque de atracciones.

21 Crear la subcarpeta carpeta **empleado** dentro de **app_parque\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_parque\templates\empleado**.

23 No utilizar [forms.py](#).

24 procedimiento para crear el archivo [urls.py](#) en **app_parque** con el código correspondiente para acceder a las funciones de [views.py](#) para operaciones de crud en empleado

25 procedimiento para agregar **app_parque** en [settings.py](#) de **backend_parque**

26 realizar las configuraciones correspondiente a [urls.py](#) de **backend_parque** para enlazar con **app_parque**

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: CLIENTE y # MODELO: ATRACCION

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 8016**.