

Primera parte

Proyecto: Pokemon.

lenguaje: Python.

Framework: Django.

Editor: VS code.

1 Procedimiento para crear carpeta del Proyecto: UIII_Pokemon_0227

2 procedimiento para abrir vs code sobre la carpeta

UIII_Pokemon_0227

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_Pokemon sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto 8027

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

11 procedimiento para crear aplicacion app_Pokemon

12 Aqui el modelo models.py

==--==--==--

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

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

```
# MODELO: CLIENTE (7 campos sin ID)
```

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

```
class Cliente(models.Model):
```

```
    nombre_cliente = models.CharField(max_length=100, unique=True)
```

```
    email = models.EmailField(max_length=254, unique=True)
```

```
    direccion = models.CharField(max_length=255)
```

```
    telefono = models.CharField(max_length=15, blank=True)
```

```
    fecha_registro = models.DateField(auto_now_add=True)
```

```
    puntos_lealtad = models.IntegerField(default=0)
```

```
    tipo_entrenador = models.CharField(max_length=50)
```

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

```
        return self.nombre_cliente
```

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

```
# MODELO: PEDIDO (7 campos + FK a Cliente)
```

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

```
class Pedido(models.Model):
```

```
    cliente = models.ForeignKey(
```

```
        Cliente,
```

```
        on_delete=models.CASCADE,
```

```
        related_name='pedidos'
```

```
)
```

```
    fecha_pedido = models.DateTimeField(auto_now_add=True)
```

```

estado_pedido = models.CharField(
    max_length=50,
    choices=[('P', 'Pendiente'), ('E', 'Enviado'), ('T', 'Entregado')],
    default='P'
)
total_monto = models.DecimalField(max_digits=10, decimal_places=2, default=0.00)
metodo_pago = models.CharField(max_length=50)
codigo_rastreo = models.CharField(max_length=100, blank=True)
fecha_entrega_estimada = models.DateField(null=True, blank=True)
es_pago_verificado = models.BooleanField(default=False)
observaciones = models.TextField(blank=True) # Campo para completar los 7

def __str__(self):
    return f"Pedido N°{self.id} de {self.cliente.nombre_cliente}"

# =====
# MODELO: PRODUCTO (7 campos + FK a Pedido)
# =====
class Producto(models.Model):
    # FK a Pedido (1:N)
    pedido = models.ForeignKey(
        Pedido,
        on_delete=models.SET_NULL,
        related_name='productos_en_pedido',
        null=True,
        blank=True
    )

    # 7 campos propios
    nombre_producto = models.CharField(max_length=100)
    descripcion = models.TextField()
    precio = models.DecimalField(max_digits=10, decimal_places=2)
    cantidad_comprada = models.IntegerField(default=1)
    categoria = models.CharField(max_length=50)
    generacion = models.IntegerField(default=1)
    es_legendario = models.BooleanField(default=False)

    def __str__(self):
        return self.nombre_producto

```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: CLIENTE

14 En view de app_Pokemon crear las funciones con sus códigos correspondientes (inicio_Pokemon, agregar_cliente, actualizar_cliente, realizar_actualizacion_cliente, borrar_cliente)

- 15 Crear la carpeta "templates" dentro de "app_Clientes".
- 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 Pokemon", "Inicio", en submenu de clientes(Agregar Clientes, ver cliente, actualizar cliente, borrar cliente), "Pedidos" en submenu de Pedidos(Agregar Pedidos, ver Pedidos, actualizar Pedidos, borrar pedidos) "Productos" en submenu de Productos(Agregar Productos, ver Productos, actualizar Productos, borrar Productos), 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 Ing. Yahel Villa, 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 cinepolis.
- 21 Crear la subcarpeta carpeta categoria dentro de app_Pokemon/templates.
- 22 crear los archivos html con su código correspondientes de (agregar_cliente.html, ver_cliente.html mostrar en tabla con los botones ver, editar y borrar, actualizar_cliente.html, borrar_cliente.html) dentro de app_Pokemon/templates/clientes.
- 23 No utilizar forms.py.
- 24 procedimiento para crear el archivo urls.py en app_Pokemon con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en clientes.
- 25 procedimiento para agregar app_Pokemon en settings.py de backend_Pokemon
- 26 realizar las configuraciones correspondiente a urls.py de backend_Pokemon para enlazar con app_Pokemon
- 27 procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.
- 27 por lo pronto solo trabajar con "cliente" dejar pendiente # MODELO: PRODUCTO y # MODELO: PEDIDOS
- 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 8027.