

**Proyecto:** Videojuegos.

**lenguaje:** Python.

**Framework:** Django.

**Editor:** VS code.

- 1 Procedimiento para crear carpeta del Proyecto: UIII\_Videojuegos\_0358
- 2 procedimiento para abrir vs code sobre la carpeta UIII\_Videojuegos\_0358
- 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\_Videojuegos sin duplicar carpeta.
- 9 procedimiento para ejecutar servidor en el puerto 8035
- 10 procedimiento para copiar y pegar el link en el navegador.
- 11 procedimiento para crear aplicacion app\_Videojuegos
- 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)
```

```
    telefono = models.CharField(max_length=20)
```

```
    email = models.EmailField()
```

```
    direccion = models.TextField()
```

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

```
    empleado_asignado = models.ForeignKey('Empleado', on_delete=models.SET_NULL,  
null=True, related_name='clientes_asignados')
```

```
    producto_interes = models.CharField(max_length=200)
```

```
    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)
```

```
    puesto = models.CharField(max_length=100)
```

```
    telefono = models.CharField(max_length=20)
```

```
    email = models.EmailField()
```

```
    fecha_contratacion = models.DateField()
```

```

    cliente_referido = models.ForeignKey('Cliente', on_delete=models.SET_NULL, null=True,
blank=True, related_name='empleados_referidos')
    producto_asignado = models.CharField(max_length=200)

```

```

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

```

```

# =====
# MODELO: VENTA
# =====

```

```

class Venta(models.Model):

```

```

    METODOS_PAGO = [
        ('EFFECTIVO', 'Efectivo'),
        ('TARJETA', 'Tarjeta'),
        ('TRANSFERENCIA', 'Transferencia'),
        ('CHEQUE', 'Cheque'),
    ]

```

```

    cliente = models.ForeignKey(Cliente, on_delete=models.CASCADE,
related_name='ventas')
    empleado = models.ForeignKey(Empleado, on_delete=models.CASCADE,
related_name='ventas')
    producto = models.CharField(max_length=200)
    fecha_venta = models.DateField(auto_now_add=True)
    monto_total = models.DecimalField(max_digits=10, decimal_places=2)
    metodo_pago = models.CharField(max_length=20, choices=METODOS_PAGO)
    folio = models.CharField(max_length=50, unique=True)

```

```

def __str__(self):
    return f"Venta {self.folio} - {self.cliente.nombre}"

```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: GÉNERO

14 En view de app\_Videojuegos crear las funciones con sus códigos correspondientes (inicio\_videojuegos, agregar\_genero, actualizar\_genero, realizar\_actualizacion\_genero, borrar\_genero)

15 Crear la carpeta "templates" dentro de "app\_Videojuegos".

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 Videojuegos", "Inicio", "Géneros", en submenu de generos(Agregar Genero, ver genero, actualizar genero, borrar genero), "Plataformas" en submenu de Plataformas(Agregar plataformas, ver plataformas, actualizar plataformas, borrar plataformas) "Videojuegos" en submenu de Videojuegos(Agregar videojuegos, ver videojuegos, actualizar videojuegos, borrar videojuegos), 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 Andrés Sandova, 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 videojuegos.
- 21 Crear la subcarpeta carpeta genero dentro de app\_Videojuegos\templates.
- 22 crear los archivos html con su codigo correspondientes de (agregar\_genero.html, ver\_generos.html mostrar en tabla con los botones ver, editar y borrar, actualizar\_genero.html, borrar\_genero.html) dentro de app\_Videojuegos\templates\genero.
- 23 No utilizar forms.py.
- 24 procedimiento para crear el archivo urls.py en app\_Videojuegos con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en generos.
- 25 procedimiento para agregar app\_Videojuegos en settings.py de backend\_Videojuegos
- 26 realizar las configuraciones correspondiente a urls.py de backend\_Videojuegos para enlazar con app\_Videojuegos
- 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: VENTA
- 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 8035.