

## Primera parte

Proyecto: **Granja Vacas**

Lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

1 Procedimiento para crear carpeta del Proyecto: **UIII\_Granja\_Vacas\_0627**

2 procedimiento para abrir vs code sobre la carpeta

**UIII\_Granja\_Vacas\_0627**

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\_Granja\_Vacas** sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el **puerto 8023**

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

11 procedimiento para crear aplicación **app\_Granja\_Vacas**

12 Aquí el modelo models.py

# models.py

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

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

```
# MODELO: VACA
```

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

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

```
    numero_identificacion = models.CharField(max_length=20, unique=True,  
verbose_name="Número de Identificación")
```

```
    nombre = models.CharField(max_length=100, blank=True, null=True,  
verbose_name="Nombre de la Vaca")
```

```
    fecha_nacimiento = models.DateField(verbose_name="Fecha de Nacimiento")
```

```
    raza = models.CharField(max_length=50, verbose_name="Raza")
```

```
    estado_reproductivo = models.CharField(max_length=50, default="No gestante",  
verbose_name="Estado Reproductivo")
```

```
    peso_kg = models.DecimalField(max_digits=6, decimal_places=2, verbose_name="Peso  
(kg)")
```

```
    corral_actual = models.CharField(max_length=30, verbose_name="Corral Actual")
```

```
    notas = models.TextField(blank=True, verbose_name="Notas de la Vaca")
```

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

```
        return f'{self.nombre or "Vaca"} {self.numero_identificacion}'
```

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

```
# MODELO: PRODUCCION (Relación 1:N con Vaca)
```

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

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

```

vaca = models.ForeignKey(Vaca, on_delete=models.CASCADE,
related_name='producciones', verbose_name="Vaca asociada")
fecha_registro = models.DateField(verbose_name="Fecha de Registro")
cantidad_litros = models.DecimalField(max_digits=5, decimal_places=2,
verbose_name="Cantidad (Litros)")
turno = models.CharField(max_length=20, verbose_name="Turno")
calidad = models.CharField(max_length=100, verbose_name="Detalles de Calidad")
empleado_registro = models.CharField(max_length=100, verbose_name="Empleado que
Registra")
equipo_usado = models.CharField(max_length=100, blank=True, verbose_name="Equipo
Usado")

def __str__(self):
    return f"Prod. {self.cantidad_litros}L de {self.vaca.numero_identificacion}"

# =====
# MODELO: EVENTOSANITARIO (Relación N:M con Vaca)
# =====
class EventoSanitario(models.Model):
    vacas_afectadas = models.ManyToManyField(Vaca, related_name='eventos_sanitarios',
verbose_name="Vacunas Afectadas")
    tipo_evento = models.CharField(max_length=100, verbose_name="Tipo de Evento")
    fecha_evento = models.DateField(verbose_name="Fecha del Evento")
    tratamiento = models.TextField(verbose_name="Tratamiento Aplicado")
    veterinario = models.CharField(max_length=100, verbose_name="Veterinario
Responsable")
    costo = models.DecimalField(max_digits=7, decimal_places=2, default=0.00,
verbose_name="Costo Total (€/$)")
    dias_retiro = models.IntegerField(default=0, verbose_name="Días de Retiro")

    def __str__(self):
        return f"{self.tipo_evento} el {self.fecha_evento}"

# =====

```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: Vaca

14 En view de **app\_Granja\_Vacas** crear las funciones con sus códigos correspondientes (inicio\_Granja\_Vacas, agregar\_vaca,

actualizar\_vaca, realizar\_actualizacion\_vaca,  
borrar\_vaca)

15 Crear la carpeta “**templates**” dentro de “**app\_Granja\_Vacas**”.

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 Granja\_Vacas”, “Inicio”, “vaca”, en submenu de vacas(Aregar Vaca,ver vaca, actualizar vaca, borrar vaca), “Produccion” en submenu de Produccion(Aregar Produccion,ver Produccion, actualizar Produccion, borrar Produccion) “Evento\_Sanitario” en submenu de Evento\_Sanitario(Aregar Evento\_Sanitario,ver Evento\_Sanitario, actualizar Evento\_Sanitario, borrar Evento\_Sanitario), 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 Maximiliano Lira, 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\_Granja\_Vacas\templates**.

22 crear los archivos html con su codigo correspondientes de (agregar\_categoria.html, ver\_categorias.html mostrar en tabla con los botones ver, editar y borrar, actualizar\_categoria.html, borrar\_categoria.html)

dentro de **app\_Granja\_Vacas\templates\Vacas**.

23 No utilizar forms.py.

24 procedimiento para crear el archivo urls.py en **app\_Cinepolis** con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en categorias.

25 procedimiento para agregar **app\_Cinepolis** en settings.py de **backend\_Granja\_Vacas**

26 realizar las configuraciones correspondiente a urls.py de **backend\_Granja\_Vacas** para enlazar con **app\_Granja\_Vacas**

**27 procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.**

27 por lo pronto solo trabajar con “categoría” dejar pendiente # MODELO: SALA y # MODELO: PELÍCULA

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 8023**.