

Manual Técnico y de Usuario - Sistema Veterinario "VetSystem"

1. Tecnologías de Desarrollo

El sistema ha sido desarrollado utilizando un stack tecnológico moderno, robusto y escalable, ideal para aplicaciones web empresariales:

- **Lenguaje de Programación (Backend):** Python 3
 - **Framework Web:** Django (MVT - Model View Template)
 - **Base de Datos:** MySQL / MariaDB
 - **Frontend (Interfaz de Usuario):**
 - HTML5 / CSS3
 - Bootstrap 5 (Framework CSS)
 - JavaScript Vanilla (Para interactividad y librerías)
 - Chart.js (Para gráficos estadísticos interactivos)
 - FontAwesome (Iconografía)
 - **Arquitectura:** Monolítica con renderizado del lado del servidor (SSR).
-

2. Funcionalidades del Sistema

El sistema VetSystem está compuesto por módulos integrados que cubren todas las necesidades operativas de una clínica veterinaria:

2.1. Panel de Control (Dashboard)

- Visualización en tiempo real de métricas clave (KPIs): Total de mascotas, clientes, citas de hoy e ingresos del día.
- Gráficos estadísticos dinámicos (Pacientes por especie, Citas por estado, Rendimiento mensual).
- Tablas resumen interactivas de los próximos eventos.

2.2. Gestión de Clientes y Mascotas

- Registro completo de propietarios (Datos personales, contacto, ubicación).
- Registro detallado de mascotas asociadas a cada cliente (Nombre, especie, raza, peso, fecha de nacimiento, foto).

2.3. Agenda Médica

- Calendario interactivo para programar, reprogramar y cancelar turnos.
- Manejo de distintos tipos de citas (Consulta general, vacunación, peluquería, cirugía).
- Control de estados (Pendiente, Confirmada, Cancelada, Atendida).

2.4. Área Clínica (Médico)

- **Historia Clínica:** Registro evolutivo de cada paciente. Incluye motivo de consulta, examen físico detallado (peso, temperatura, frecuencia cardíaca/respiratoria), diagnóstico y tratamiento.
- **Vacunaciones:** Control del esquema de vacunación, lote de vacunas y fechas de próximas dosis recomendadas.
- **Telemedicina:** Plataforma integrada para registrar consultas virtuales a distancia y adjuntar enlaces de videollamada.

2.5. Peluquería (Grooming)

- Gestión de órdenes de servicio estético para mascotas.
- Control de fecha y hora de inicio/fin del servicio, observaciones y estado del progreso (En curso, Terminado).

2.6. Gestión Comercial e Inventario

- **Inventario:** Catálogo completo de productos, medicamentos y accesorios.
- Control riguroso de stock (Stock actual vs. Stock mínimo).
- Kárdex de movimientos: Registro automático de entradas (compras) y salidas (ventas/uso interno).
- **Facturación:** Emisión de comprobantes de pago detallados, selección de métodos de pago (Efectivo, Tarjeta, Transferencia) y cálculo automático de IGV/IVA.
- **Control de Caja:** Apertura y cierre diario de caja registradora de la clínica, calculando el dinero inicial, los ingresos del día y el arqueo final.

2.7. Reportes y Business Intelligence (BI)

- Generación de reportes tabulares y gráficos financieros.
- Filtros dinámicos por rango de fechas para analizar el rendimiento económico mensual o diario de la veterinaria.

2.8. Sistema y Seguridad

- **Configuración Global:** Personalización de los datos de la clínica (Nombre, Logo corporativo, Moneda local, Contacto, Colores de la interfaz).
 - **Usuarios:** Creación de perfiles de acceso (Administradores, Cajeros, Veterinarios).
 - **Mantenimiento:** Herramienta interna para realizar Backups (Copias de seguridad) directos de la base de datos SQL, Restauración de backups y Reseteo del sistema (limpieza de datos transaccionales para iniciar en limpio).
-

3. Instalación y Configuración Local

Para ejecutar el sistema en una computadora nueva (con Windows), siga estos pasos:

Prerrequisitos

1. Instalar **Python 3.10 o superior** (asegurarse de marcar la casilla "Add Python to PATH" durante la instalación).
2. Instalar un servidor local de bases de datos como **XAMPP, WAMP** o instalar **MySQL Server** de forma independiente.

Paso a paso

1. Preparar la Base de Datos:

- Inicie su servidor MySQL (ej. abrir panel de XAMPP e iniciar "MySQL").
- Abra phpMyAdmin o la consola MySQL y cree una base de datos vacía llamada: `sistema_veterinaria_db`.

2. Configurar el Proyecto:

- Copie la carpeta del código fuente (`sistema-veterinaria`) en su computadora (ej. en `C:\Webs\Python\sistema-veterinaria`).
- Abra la terminal (Símbolo del sistema o PowerShell) y navegue hasta la carpeta del proyecto: `cd C:\Webs\Python\sistema-veterinaria`

3. Entorno Virtual y Dependencias (Recomendado):

- Cree un entorno virtual (solo la primera vez): `python -m venv venv`
- Active el entorno virtual: `venv\Scripts\activate`
- Instale las librerías necesarias: `pip install -r requirements.txt` (o instale `django` , `mysqlclient` , etc., si no hay requirements).

4. Migrar la Base de Datos:

- Construya las tablas en MySQL ejecutando: `python manage.py makemigrations` `python manage.py migrate`

4. Cómo Ejecutar la Aplicación (Modo Desarrollo)

Cada vez que desee encender el sistema para trabajar localmente, siga estos dos sencillos pasos desde el Símbolo del Sistema (CMD) o PowerShell:

1. Navegue a la carpeta y active el entorno virtual: `cd C:\Webs\Python\sistema-veterinaria\venv\Scripts\activate`
 2. Encienda el servidor local de Django: `python manage.py runserver`
 3. Abra su navegador web preferido y visite la siguiente dirección: **http://127.0.0.1:8000/**
-

5. Credenciales de Acceso (Usuario Administrador)

Si es la primera vez que instala el sistema, debe crear un Superusuario maestro. Con el entorno virtual activado, ejecute:

```
python manage.py createsuperuser
```

Siga las instrucciones en la pantalla de la consola: * **Username (Usuario):** admin * **Email:** (puede dejarlo en blanco presionando Enter) * **Password (Contraseña):** admin123 *(Nota: al escribir la contraseña en la consola, no se verán los caracteres por seguridad, pero sí se están registrando).* * **Password (Confirmación):** admin123

Utilice estas credenciales en la pantalla de inicio de sesión de la aplicación web.

6. Recomendación de Hosting y Despliegue a Producción

Para llevar esta aplicación a internet para que funcione 24/7, la recomendación ideal, buscando la mejor relación calidad/precio para aplicaciones en Django, es utilizar un plan **VPS (Servidor Privado Virtual)** o un servicio PaaS.

Las mejores opciones: 1. **DigitalOcean (Droplet) / Linode / Vultr** (Recomendado - Mayor control y economía, ~\$5 - \$10 USD/mes). 2. **PythonAnywhere** (Excelente para principiantes en Python, muy fácil despliegue, plan básico ~\$5 USD/mes). 3. **Railway.app / Render** (Modernos, sincronización directa con GitHub, cobran por uso).

Guía Paso a Paso para Despliegue (Ejemplo General para VPS con Ubuntu Server)

Para un VPS (como DigitalOcean), la arquitectura estándar de producción es: **Gunicorn (Servidor de aplicaciones) + Nginx (Proxy Inverso) + MySQL/PostgreSQL.**

Paso 1: Preparar el Servidor 1. Acceda a su VPS mediante SSH: `ssh root@ip_del_servidor` 2. Actualice los paquetes del servidor: `apt update && apt upgrade -y` 3. Instale dependencias de sistema: `apt install python3-pip python3-venv nginx mysql-server libmysqlclient-dev -y`

Paso 2: Configurar la Base de Datos en el Servidor 1. Entre a MySQL: `mysql -u root` 2. Cree la base de datos y un usuario seguro: `CREATE DATABASE sistema_veterinaria_db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci; CREATE USER 'vet_user'@'localhost' IDENTIFIED BY 'TuContraSegura123!'; GRANT ALL PRIVILEGES ON sistema_veterinaria_db.* TO 'vet_user'@'localhost'; FLUSH PRIVILEGES; EXIT;`

Paso 3: Subir el Código y Preparar Entorno 1. Transfiera los archivos de su proyecto al servidor (usando FTP/FileZilla, SCP, o clonando desde GitHub). Colóquelo en `/var/www/sistema-veterinaria/` 2. Navegue al directorio: `cd /var/www/sistema-veterinaria` 3. Cree y active un entorno virtual: `python3 -m venv venv` y `source venv/bin/activate` 4. Instale las librerías de Python e incluya el servidor de producción: `pip install -r requirements.txt gunicorn` 5. Edite su archivo `settings.py` para producción: * Cambie `DEBUG = False` * Añada la IP o Dominio del servidor: `ALLOWED_HOSTS = ['midominio.com', 'IP_DEL_SERVIDOR']` * Actualice las credenciales en `DATABASES` para que coincidan con el Paso 2. 6. Ejecute migraciones y recolecte archivos estáticos: `python manage.py migrate` `python manage.py collectstatic`

Paso 4: Configurar Gunicorn como Servicio del Sistema (Para que nunca se apague) 1. Cree un archivo de servicio: `nano /etc/systemd/system/gunicorn.service` 2. Pegue esta configuración básica: `[[Unit] Description=gunicorn daemon After=network.target`

`[Service] User=root Group=www-data WorkingDirectory=/var/www/sistema-veterinaria Environment="PATH=/var/www/sistema-veterinaria/venv/bin" ExecStart=/var/www/sistema-veterinaria/venv/bin/gunicorn --access-logfile - --workers 3 --bind unix:/var/www/sistema-veterinaria/sistema_veterinaria.sock veterinaria.wsgi:application`

`[Install] WantedBy=multi-user.target` 3. Inicie el servicio: `systemctl start gunicorn` y `systemctl enable gunicorn`

Paso 5: Configurar Nginx (Para recibir a los usuarios por el puerto 80) 1. Cree un archivo de configuración Nginx: `nano /etc/nginx/sites-available/sistema_veterinaria` 2. Agregue el siguiente bloque (cambie su_dominio.com): `nginx server { listen 80; server_name su_dominio.com IP_DEL_SERVIDOR;`

```
location = /favicon.ico { access_log off; log_not_found off; }

location /static/ {
```

```
    root /var/www/sistema-veterinaria;
}

location /media/ {
    root /var/www/sistema-veterinaria;
}

location / {
    include proxy_params;
    proxy_pass http://unix:/var/www/sistema-veterinaria/sistema_veterinaria.sock;
}
```

} `` 3. **Active el sitio:** ln -s /etc/nginx/sites-available/sistema_veterinaria /etc/nginx/sites-enabled 4. **Reinicie Nginx:** systemctl restart nginx`

Paso final: Si va a su navegador y digita la IP de su servidor o su dominio, el sistema VetSystem estará disponible públicamente, de manera robusta y segura.