



# Manual de Despliegue - Sistema de Gestión Veterinaria LaMascotApp

## Requisitos Previos

### Software Necesario:

#### 1. XAMPP (versión 8.2 o superior)

- [Descargar XAMPP](#)

- Incluye: Apache, MySQL, PHP, phpMyAdmin

#### 2. Node.js (versión 18 o superior) – Versión usada para el proyecto: v24.11.1

- [Descargar Node.js](#)

- Incluye npm (Node Package Manager)

#### 3. Git (para clonar el repositorio)

- [Descargar Git](#)

### Sistemas Operativos Compatibles:

- Windows 10/11 (64-bit)

- macOS 10.15 o superior

- Linux Ubuntu 20.04 o superior

## Paso 1: Instalación de XAMPP

### Windows:

1. Descargar el instalador de XAMPP

2. Ejecutar como administrador

3. Seleccionar componentes: **Apache, MySQL, phpMyAdmin**

4. Instalar en **C:\xampp** (ruta recomendada)

5. Finalizar instalación

### **Verificación de instalación:**

1. Abrir Panel de Control de XAMPP
2. Iniciar los servicios:
  - **Apache** (puerto 80)
  - **MySQL** (puerto 3306)
3. Verificar en navegador: `http://localhost`
  - Debería aparecer el dashboard de XAMPP

### **Paso 2: Configuración de la Base de Datos**

#### **Importar la base de datos:**

1. Abrir phpMyAdmin: <http://localhost/phpmyadmin>  
(o dar click en **Admin**, al lado del botón **Stop** de **MySQL** en XAMPP)
2. Hacer clic en "**Nueva**" en el menú lateral
3. Crear base de datos:
  - Nombre: `veterinaria_db`
  - Cotejamiento: `utf8mb4_general_ci`
  - Crear
4. Seleccionar la base de datos `veterinaria_db`
5. Ir a pestaña "**Importar**"
6. Copiar y pegar el siguiente código SQL:

```
-- Crear base de datos si no existe
CREATE DATABASE IF NOT EXISTS veterinaria_db;
USE veterinaria_db;

-- Tabla USUARIOS
CREATE TABLE IF NOT EXISTS usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
```

```

usuario VARCHAR(50) UNIQUE,
email VARCHAR(100) UNIQUE NOT NULL,
password VARCHAR(255) NOT NULL,
telefono VARCHAR(15),
rol ENUM('propietario', 'veterinario', 'repcionista') DEFAULT
'propietario'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Tabla MASCOTAS
CREATE TABLE IF NOT EXISTS mascotas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    especie VARCHAR(50) NOT NULL,
    raza VARCHAR(50),
    edad INT,
    peso DECIMAL(5,2),
    id_propietario INT NOT NULL,
    FOREIGN KEY (id_propietario) REFERENCES usuarios(id) ON DELETE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Tabla CITAS
CREATE TABLE IF NOT EXISTS citas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    fecha DATE NOT NULL,
    hora TIME NOT NULL,
    motivo TEXT NOT NULL,
    nombre_mascota VARCHAR(100) NOT NULL,
    id_veterinario INT NOT NULL,
    estado ENUM('activa', 'cancelada') DEFAULT 'activa',
    FOREIGN KEY (id_veterinario) REFERENCES usuarios(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Insertar datos de prueba (CONTRASEÑAS EN TEXTO PLANO: 123456)
INSERT INTO usuarios (nombre, usuario, email, password, telefono, rol)
VALUES
('Juan Pérez', 'juan', 'juan@email.com', '123456', '555-1234',
'propietario'),
('María García', 'maria', 'maria@email.com', '123456', '555-5678',
'propietario'),
('Dr. Carlos López', 'carlos', 'carlos@vet.com', '123456', '555-9012',
'veterinario'),
('Dra. Ana Martínez', 'ana', 'ana@vet.com', '123456', '555-3456',
'veterinario'),
('Repcionista Uno', 'repcion', 'repcion@vet.com', '123456', '555-
7890', 'repcionista');

-- Mascotas
INSERT INTO mascotas (nombre, especie, raza, edad, peso, id_propietario)
VALUES
('Firulais', 'Perro', 'Labrador', 3, 25.50, 1),
('Mishi', 'Gato', 'Siamés', 2, 4.20, 1),
('Rex', 'Perro', 'Pastor Alemán', 5, 30.00, 2);

```

```
-- Citas con fechas específicas
INSERT INTO citas (fecha, hora, motivo, nombre_mascota, id_veterinario)
VALUES
('2025-12-05', '10:00:00', 'Vacunación anual', 'Firulais', 3),
('2025-12-06', '14:30:00', 'Control general', 'Mishi', 4),
('2025-12-07', '11:00:00', 'Chequeo dental', 'Rex', 3);
```

7. Hacer clic en "**Continuar**" para ejecutar

#### Verificar tablas creadas:

SQL:

```
SHOW TABLES;
```

Debería ver: usuarios, mascotas, citas

### **Paso 3: Clonar el Proyecto**

#### Opción A: desde la línea de comandos

- Abrir terminal/CMD en la carpeta donde quieras el proyecto

#### # Windows:

```
cd C:\xampp\htdocs
```

#### # macOS:

```
cd /Applications/XAMPP/htdocs
```

#### # Linux:

```
cd /opt/lampp/htdocs
```

#### # Clonar repositorio

```
git clone https://github.com/licenciadodev/veterinaria-citas.git
```

#### # Acceder al directorio del proyecto

```
cd veterinaria-citas
```

#### Opción B: Si no tienes Git:

1. Visitar: <https://github.com/licenciadodev/veterinaria-citas>
2. Hacer clic en "**Code**" → "**Download ZIP**"
3. Extraer en: C:\xampp\htdocs\veterinaria-citas

### Paso 4: Configuración del Backend (Node.js)

#### Instalación de Dependencias

# En la **carpeta del proyecto** y luego ir a la carpeta **backend**:

```
cd veterinaria-citas
```

```
cd backend
```

```
npm install
```

**Verificar dependencias instaladas (dentro de la carpeta backend):**

```
```bash
```

```
npm list
```

```
...
```

Debería ver: express, mysql2, bcryptjs, entre otras.

#### Nota sobre bcryptjs:

-  **bcryptjs está instalado** pero no se está usando actualmente
-  **Propósito futuro:** Para implementar hash de contraseñas cuando sea necesario
-  **Actual:** Las contraseñas están en texto plano para facilitar pruebas académicas

### Paso 5: Configuración del Proyecto

**Verificar estructura de archivos:**

```
veterinaria-citas/
├── backend
│   ├── db.js
│   ├── package-lock.json
│   ├── package.json
│   ├── prueba-final.js
│   └── routes
│       ├── appointments.js
│       ├── auth.js
│       ├── medicalHistory.js
│       ├── pets.js
│       └── users.js
└── server.js
└── test-db.js
├── database
│   └── veterinaria_db.sql
└── frontend
    ├── css
    │   ├── css-acceso
    │   │   ├── citas-medicas.css
    │   │   ├── historial-medico.css
    │   │   ├── inicio-sesion.css
    │   │   └── registro.css
    │   ├── css-perfiles
    │   │   ├── dashboard-propietario.css
    │   │   ├── dashboard-repcionista.css
    │   │   └── dashboard-veterinario.css
    │   └── css-registros
```

```
|   |   └── registro-propietario.css
|   |   └── registro-repcionista.css
|   └── └── registro-veterinario.css
|   └── global.css
|   └── index.css
└── html
    ├── html-acceso
    |   ├── citas-medicas.html
    |   ├── historial-medico.html
    |   ├── inicio-sesion.html
    |   └── registro.html
    ├── html-perfiles
    |   ├── dashboard-propietario.html
    |   ├── dashboard-repcionista.html
    |   └── dashboard-veterinario.html
    ├── html-registros
    |   ├── registro-propietario.html
    |   ├── registro-repcionista.html
    |   └── registro-veterinario.html
    └── index.html
└── images
    ├── equipo.png
    └── veterinaria.png
└── js
    ├── js-acceso
    |   ├── citas-medicas.js
    |   ├── historial-medico.js
    |   └── inicio-sesion.js
```

```
|── js-perfiles
|   ├── dashboard-propietario.js
|   ├── dashboard-repcionista.js
|   └── dashboard-veterinario.js
├── js-registros
|   ├── registro-propietario.js
|   ├── registro-repcionista.js
|   └── registro-veterinario.js
└── script.js
```

#### Configurar variables de entorno (**opcional**) o si prefiere, saltar al punto 6:

1. Si existe `.env.example`, renombrarlo a `.env`:

##### Windows (PowerShell):

```
```powershell
```

```
Rename-Item .env.example .env
```

```
```
```

##### Linux/macOS:

```
```bash
```

```
mv .env.example .env
```

2. Editar el archivo `.env` (dentro de la carpeta **backend**) si es necesario:

```
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=
DB_NAME=veterinaria_db
PORT=3000
```

## Paso 6: Iniciar el Servidor

Método recomendado:

``` bash o cmd:

-Ir a la **carpeta del proyecto** y luego a la carpeta **backend**:

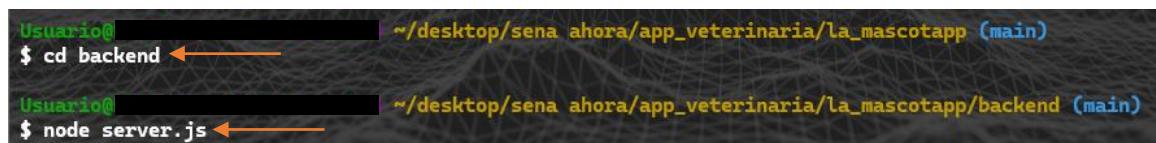
```
cd veterinaria-citas  
cd backend
```

**Ejecutar desde la carpeta backend:**

```
node server.js
```

```

**Ejemplo visual desde el git bash:**



```
Usuario@... ~ /desktop/sena ahora/app_veterinaria/la_mascotapp (main)  
$ cd backend ←  
  
Usuario@... ~ /desktop/sena ahora/app_veterinaria/la_mascotapp/backend (main)  
$ node server.js ←
```

**¿Qué sucede al ejecutar este comando?**

1.  Se inicia el servidor Node.js
2.  Se establece conexión con MySQL
3.  Se configuran las rutas de la aplicación
4.  El servidor queda escuchando en el puerto 3000

**Verificar que el servidor está funcionando:**

1. En la terminal deberías ver algo así (o similar):

```

SERVIDOR LISTO PARA PRUEBAS

Conectado a MySQL – veterinaria\_db

Estructura de tabla “usuarios”:

- id (int)
- nombre (varchar)
- usuario (varchar)
- email (varchar)
- password (varchar)
- telefono (varchar)
- rol (enum) ` ``

2. Abrir navegador web
3. Visitar: <http://localhost:3000>
4. Debería aparecer la página principal de la aplicación veterinaria

### **Paso 7: Acceso a la Aplicación**

**URL principal:**

<http://localhost:3000>

**Acceso a phpMyAdmin (gestión de BD):**

<http://localhost/phpmyadmin>

### **Paso 8: Credenciales de Prueba**

**Usuarios disponibles (todas las contraseñas son: 123456):**

| Nombre            | Usuario | Email           | Rol         | Contraseña |
|-------------------|---------|-----------------|-------------|------------|
| Juan Pérez        | juan    | juan@email.com  | Propietario | 123456     |
| María García      | maria   | maria@email.com | Propietario | 123456     |
| Dr. Carlos López  | carlos  | carlos@vet.com  | Veterinario | 123456     |
| Dra. Ana Martínez | ana     | ana@vet.com     | Veterinario | 123456     |
| Repcionista Uno   | repcion | repcion@vet.com | Repcionista | 123456     |

## Paso 9: Solución de Problemas

**Error: "Cannot find module"**

```bash o cmd:

**Reinstalar dependencias**

`npm install`

```

**Error: "Port 3000 is already in use"**

```bash

*# Opción 1: Cambiar puerto en server.js*

# Buscar: app.listen(3000, ...) y cambiar a otro puerto (ej: 3001)

*# Opción 2: Liberar el puerto (Windows)*

`netstat -ano | findstr :3000`

`taskkill /PID [NUMERO_PID] /F`

*# Opción 3: Usar otro puerto temporalmente*

# Modificar temporalmente server.js y usar node server.js

### Error de conexión a MySQL

1. Verificar que MySQL esté corriendo en XAMPP

2. Verificar credenciales en el código:

- Abrir `server.js`

- Buscar conexión a base de datos

3. Probar conexión manual:

```bash

`mysql -u root -p -h localhost`

# Presionar Enter si no hay contraseña

**Error: “Access denied for user 'root'@'localhost'”**

1. Abrir phpMyAdmin: <http://localhost/phpmyadmin>
2. Verificar que el usuario root no tenga contraseña
3. O configurar la contraseña en el archivo de conexión de Node.js

**Flujo de trabajo típico:**

1. Iniciar XAMPP (Apache y MySQL)
2. Abrir terminal en la **carpeta del proyecto** y luego ir a la carpeta **backend**
3. Ejecutar: `node server.js`
4. Abrir navegador en: <http://localhost:3000>
5. Hacer click en **iniciar sesión** (parte superior derecha de la aplicación).
6. Detener servidor (Ctrl+C)
7. Volver a ejecutar: `node server.js`

**Continúa en la siguiente página:**

localhost:3000

**LaMascotApp**

Iniciar sesión 

# Cuidado médico profesional para tus mejores amigos

Atención veterinaria especializada con tecnología avanzada y un trato humano para el bienestar de tus mascotas.

[INICIAR SESIÓN](#) [REGISTRARSE](#)



localhost:3000/login

**LaMascotApp**

Iniciar sesión Iniciar sesión

## Iniciar sesión en LaMascotApp

Seleccione su perfil para acceder al sistema

 **Propietario de mascota**  
Gestione las citas, historial médico y datos de sus mascotas.  
[ACCEDER COMO PROPIETARIO](#)

 **Recepcionista**  
Administre citas, pacientes y comunicación con usuarios.  
[ACCEDER COMO RECEPCIONISTA](#)

 **Veterinario**  
Consulte historiales, registre diagnósticos y tratamientos.  
[ACCEDER COMO VETERINARIO](#)  
Activar Windows  
Ve a Configuración para activar Windows.

**Acceder como propietario:**

Usuario: juan

Contraseña: 123456

**Acceder como recepcionista:**

Usuario: recepcion

Contraseña: 123456

**Acceder como veterinario:**

Usuario: ana

Contraseña: 123456

### Reiniciar servicios:

1. **Servidor Node.js:** Ctrl+C en la terminal, luego node server.js
2. **XAMPP:** Desde el panel de control, reiniciar Apache y MySQL

### Enlaces importantes:

- Repositorio: <https://github.com/licenciadodev/veterinaria-citas>
- XAMPP: <https://www.apachefriends.org/es/index.html>
- Node.js: <https://nodejs.org/es/>
- MySQL Documentation: <https://dev.mysql.com/doc/>

### Comandos rápidos de referencia:

```
```bash
```

- *Iniciar servidor*

```
node server.js
```

- *Instalar dependencias*

```
npm install
```

- *Clonar proyecto*

```
git clone https://github.com/licenciadodev/veterinaria-citas.git
```

- *Verificar puertos en uso (Windows)*

```
netstat -ano | findstr :3000
```

- Verificar puertos en uso (Linux/macOS)

```
sudo lsof -i :3000
```

**⚠ Nota de Seguridad:** Esta aplicación utiliza contraseñas en texto plano **solo con fines educativos y de prueba**. El paquete bcryptjs está incluido para una posible implementación de seguridad en el futuro, pero no está activado en la versión actual.

 Fecha de elaboración: Diciembre 2025

 Repositorio: <https://github.com/licenciadodev/veterinaria-citas>

 Comando de inicio: `node server.js`

 URL de acceso: `http://localhost:3000`