

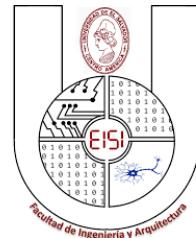
UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERIA Y ARQUITECTURA

ESCUELA DE INGENIERIA Y SISTEMAS INFORMATICOS

EDUCACION A DISTANCIA

PROGRAMACION I – PRN115 | CICLO II – 2025



DESARROLLO DE UNA SOLUCION ORIENTADA A OBJETOS

TUTOR:

ING. JOSE ALBERTO MARTINEZ CAMPOS

GRUPO DE TUTORIA:

03

ESTUDIANTES:

HERNANDEZ LARIN, ERIKSSON JOSUE – HL22009 - <https://github.com/Eriksson03>

ALDANA BELTRAN, JAIME ROBERTO – AB17005 - GitHub: 'ab17005-collab'

SAN SALVADOR, 16 DE NOVIEMBRE DE 2025

## Contenido

<b>Introducción .....</b>	4
<b>Enunciado del ejercicio .....</b>	5
<b>Metodología solucionadora de problemas .....</b>	6
I.    Planteamiento del problema .....	6
II.   Análisis del problema .....	6
a.    Definición de variables de salida.....	6
b.    Definición de datos de entrada .....	7
c.    Restricciones.....	8
d.    Proceso .....	9
e.    Definición de variables de proceso.....	11
III.  Diseño de la Solución .....	13
<b>Diagrama de caso de uso .....</b>	15
<b>Diagrama de clases .....</b>	16
<b>Archivo Script de base de datos en MySQL.....</b>	17
<b>Imagen del modelo Entidad – Relacion de la Base de datos en MySQL .....</b>	26
<b>Manual técnico y de usuario .....</b>	27
<b>Manual Técnico .....</b>	27
<b>Manual de usuario del Sistema de Gestión Veterinaria .....</b>	33
<b>Enlace del videotutorial .....</b>	39

<b>Capturas de uso de GitHub .....</b>	40
<b>Conclusiones.....</b>	43
<b>Recomendaciones.....</b>	44
<b>Fuentes.....</b>	45

## Introducción

Este documento presenta el desarrollo de la Clave 1, correspondiente al sistema de escritorio para la veterinaria “*Patitas y Pelos*”, cuyo objetivo principal es gestionar de forma integrada la información de los dueños, sus mascotas y las citas médicas que se agendan en la clínica. A partir del enunciado se plantea una solución implementada en Windows Forms C#, utilizando los principios de programación orientada a objetos y el concepto de herencia para modelar las entidades principales del sistema, las cuales se almacenan en una base de datos MySQL. Además, el proyecto se desarrolla empleando GitHub para el control de versiones y la colaboración en equipo, e incluye como evidencias los diagramas de caso de uso y de clases, el modelo entidad–relación, el script de la base de datos, el manual técnico y de usuario, y un videotutorial donde se explica el diseño y funcionamiento de la solución propuesta, tal como se solicita en la guía del proyecto.

## Enunciado del ejercicio

### Ejercicio 1: SISTEMA PARA UNA VETERINARIA

La veterinaria “Patitas y pelos” requiere un Sistema Informático que le permite registrar la información general de los clientes (dueños) y sus mascotas. Cada mascota tiene un dueño y un dueño puede tener más de una mascota. El sistema debe permitir al dueño realizar una cita médica, según los horarios disponibles (debe tener una clase/tabla para horario por fecha) por la veterinaria. Los datos mínimos del dueño son sus nombres y apellidos, teléfono móvil, email, dirección y otros que el programador considere necesario.

Adicionalmente el dueño puede cancelar la cita programada o reprogramarla. El usuario administrador de la veterinaria es el encargado de ingresar los datos al sistema y podrá agendar citas, modificarlas o cancelarlas, el sistema permitirá registrar el pago de los servicios prestados, tales como: cita, comida, accesorios o medicamentos que han sido comprados por el dueño. Deberá registrar si el tipo de pago es en efectivo, tarjeta (crédito o débito) o bitcoin. El veterinario (se necesita registrar los veterinarios que trabajan para la empresa) podrá registrar los detalles de la consulta en el expediente de la mascota, con el fin de llevar un cuadro clínico. También podrá registrar el cuadro de vacunas y fechas en las que se le colocó a la mascota. El dueño de una mascota también podrá realizar compras de los productos o accesorios que la clínica veterinaria posee. Por lo tanto, es importante que el Sistema lleve un control de los insumos que oferta la veterinaria. El sistema, es una aplicación de escritorio requerirá de un módulo para el acceso de usuarios al sistema (ingresar nuevo usuario, consultar usuario, modificar usuario, eliminar usuario). La solución debe realizarse en Windows Forms C# y utilizando la base de datos MySQL.

# Metodología solucionadora de problemas

Sistema de gestión de citas veterinaria Patitas y Pelos

## I. Planteamiento del problema

Entrada de datos	Salida de datos
<ul style="list-style-type: none"> <li>• Datos de la mascota</li> <li>• Datos del cliente</li> <li>• Motivo de la cita</li> <li>• Diagnóstico</li> <li>• Fecha de la cita</li> <li>• Veterinario que atenderá</li> <li>• Horario</li> </ul>	<ul style="list-style-type: none"> <li>• Número de cita</li> <li>• Estado de la cita</li> <li>• Datos de la mascota</li> <li>• Datos del cliente</li> <li>• Motivo de la cita</li> <li>• Diagnóstico</li> <li>• Fecha de la cita</li> <li>• Veterinario que atenderá</li> <li>• Horario</li> </ul>



## II. Análisis del problema

### a. Definición de variables de salida

Nombre	Tipo	Descripción
<b>_idCita</b>	Int	Contiene el id de la cita creada.
<b>nombreMascota</b>	String	Contiene el nombre de la mascota que asistirá a la cita.
<b>especie</b>	String	Almacena la especie del paciente.
<b>raza</b>	String	Almacena el tipo de raza de la mascota.
<b>edad</b>	Int	Guarda la edad de la mascota en años.
<b>sexo</b>	String	Almacena el sexo de la mascota.
<b>nombreDueño</b>	String	Contiene el nombre del dueño que llevará al paciente a la cita.
<b>telefono</b>	String	Guarda el número telefónico de contacto del cliente.

## PROYECTO PROGRAMACION

<b>correo</b>	String	Almacena el correo electrónico de contacto del cliente.
<b>direccion</b>	String	Almacena la dirección del cliente.
<b>fechaCita</b>	DateTime	Almacena la fecha para la cual esta agendada la cita.
<b>horaCita</b>	DateTime	Contine la hora para la cual esta agendada la cita.
<b>motivo</b>	String	Contiene el motivo de la cita.
<b>notas</b>	String	Almacena las notas opcionales que se ingresaron al crear la cita.
<b>diagnostico</b>	String	Muestra el diagnostico brindado por el veterinario.
<b>nombreVeterinario</b>	String	Contiene el nombre del veterinario a cargo de la cita.
<b>_estadoActual</b>	String	Contiene el valor del estado del objeto cita, que puede ser próxima, pasada, cancelada.

### b. Definición de datos de entrada

#### *b.1 Definición de variables de entrada*

Nombre	Tipo	Descripción
<b>cbxMascota</b>	String	Contiene el nombre de la mascota.
<b>clvMotivo</b>	String	Contiene el motivo específico de la cita.
<b>idCliente</b>	Int	Contiene el ID en el sistema para el cliente.
<b>idMascota</b>	Int	Almacena el ID de la mascota en el sistema.
<b>notas</b>	String	Contiene las notas opcionales agregadas a la cita.
<b>idVeterinario</b>	Int	Guarda el ID del veterinario en el sistema.

*b.2 Definición de constantes*

Nombre	Tipo	Descripción
<b>HORARIO_INICIO_MATUTINO</b>	TimeSpan	Inicio del horario de mañana (8:00 AM)
<b>HORARIO_FIN_MATUTINO</b>	TimeSpan	Final del horario de mañana (12:00 PM)
<b>HORARIO_INICIO_VESPERTINO</b>	TimeSpan	Inicio del horario de tarde (1:00 PM)
<b>HORARIO_FIN_VESPERTINO</b>	TimeSpan	Final del horario de tarde (5:00 PM)
<b>DURACION_TURNO</b>	Int	Duración de cada cita en minutos (30 min)
<b>ESTADO_PROXIMA</b>	String	Valor del estado “próxima” para una cita activa
<b>ESTADO_CANCELADA</b>	String	Valor del estado “cancelada” cuando se cancela una cita
<b>ESTADO_PASADA</b>	String	Valor del estado “pasada” cuando la fecha ya ocurrió
<b>CADENA_CONEXION</b>	String	Cadena literal para conectarse a MySQL

*c. Restricciones*

*idCita > 0*  
*idCliente > 0*  
*idMascota > 0*  
*idVeterinario > 0*  
*fechaCita >= fecha actual*  
*horaTurno debe existir en el diccionario de horarios*  
*No se puede agendar un horario cuyo valor (true/false) indica ocupado*  
*La mascota seleccionada debe pertenecer al cliente logueado*  
*No se puede cancelar citas que ya están en estado “cancelada”*  
*No se puede reprogramar una cita pasada*  
*El JSON del horario debe contener todas las llaves posibles del día*  
*Los campos obligatorios: mascota, veterinario, fecha, hora, motivo*  
*clbMotivo debe tener al menos 1 ítem marcado*  
*El estado solo puede ser: próxima, pasada, cancelada*

#### d. Proceso

##### 1. Inicio de Sesión del Usuario

```
Si correo y contraseña son válidos en la BD
    obtener datos del cliente
    abrir Form2
sino
    mostrar error "credenciales incorrectas"
fin si
```

##### 2. Búsqueda del Cliente en Form2

```
Si el cliente existe en BD
    mostrar datos para confirmar
    si confirma
        abrir Form3 con los datos del cliente
    fin si
sino
    mostrar "usuario no encontrado"
fin si
```

##### 3. Cargar Datos Iniciales en Form3

```
cargar mascotas del cliente
cargar veterinarios
generar horario por defecto
mostrar en DataGridView el horario
```

#### 4. Selección de Veterinario y Fecha

```
si existe un registro en BD para ese veterinario y esa fecha
    cargar JSON del horario
    llenar DataGridView
sino
    generar horario nuevo
    mostrar horarios por defecto
fin si
```

#### 5. Agendar Cita

```
si faltan campos obligatorios
    mostrar error
    detener proceso
fin si

si horarioTurno[selectedSlot] == false
    mostrar "horario ocupado"
    detener proceso
fin si

marcar horarioTurno[selectedSlot] = false

si existe registro horarioActualId
    actualizar JSON en BD
sino
    insertar nuevo JSON de horarios en BD
fin si

insertar cita en la tabla citas
```

**mostrar** Form4 con el resumen

## 6. Visualizar Cita en Form4

```
obtener idCita  
  
consultar información actualizada en BD  
  
mostrar en labels el detalle completo
```

## 7. Cancelar Cita

```
preguntar "¿seguro que desea cancelar?"  
  
si confirmación == sí  
  
    actualizar estado = "cancelada" en BD  
  
    actualizar DataGridView en Form3  
  
fin si
```

## 8. Modificar o Reagendar Cita

```
llenar controles de Form3 con datos actuales de la cita  
  
si usuario confirma cambios  
  
    actualizar registro de la cita  
  
    actualizar horarios si cambia la hora  
  
fin si
```

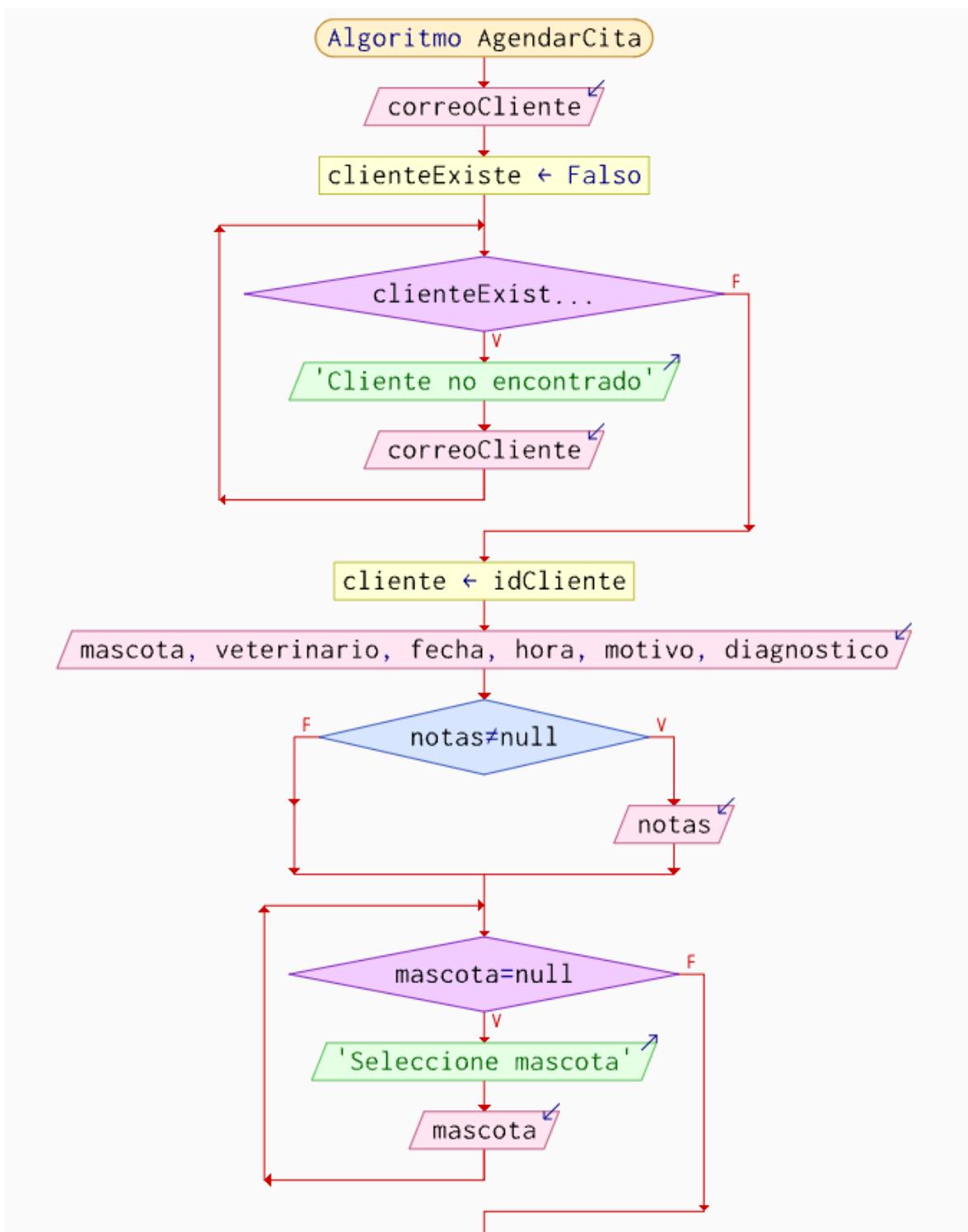
### e. Definición de variables de proceso

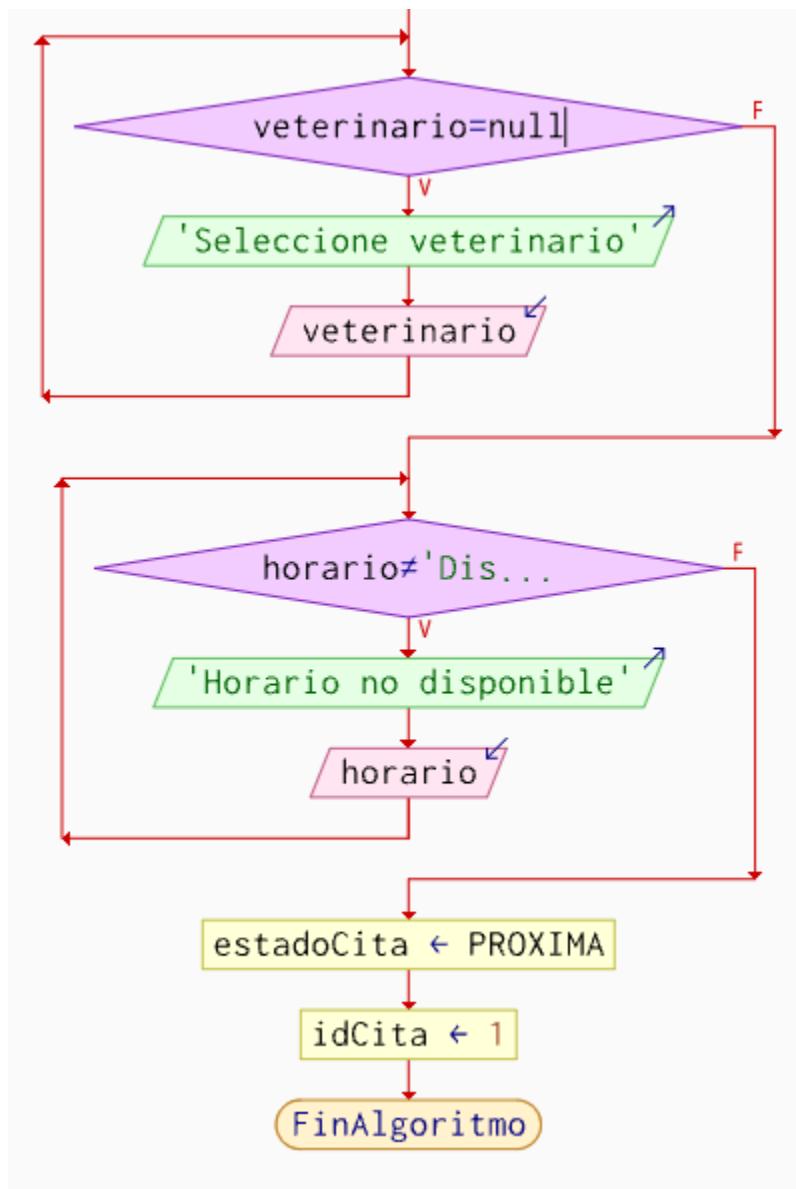
Nombre	Tipo	Descripción
<b>selectedSlot</b>	String	Slot de horario seleccionado en el DGV
<b>fechaSeleccionada</b>	DateTime	Fecha del calendario
<b>horaCita</b>	TimeSpan	Hora obtenida del slot seleccionado
<b>horarioActualId</b>	Int?	Id del registro de horarios del veterinario o null

## PROYECTO PROGRAMACION

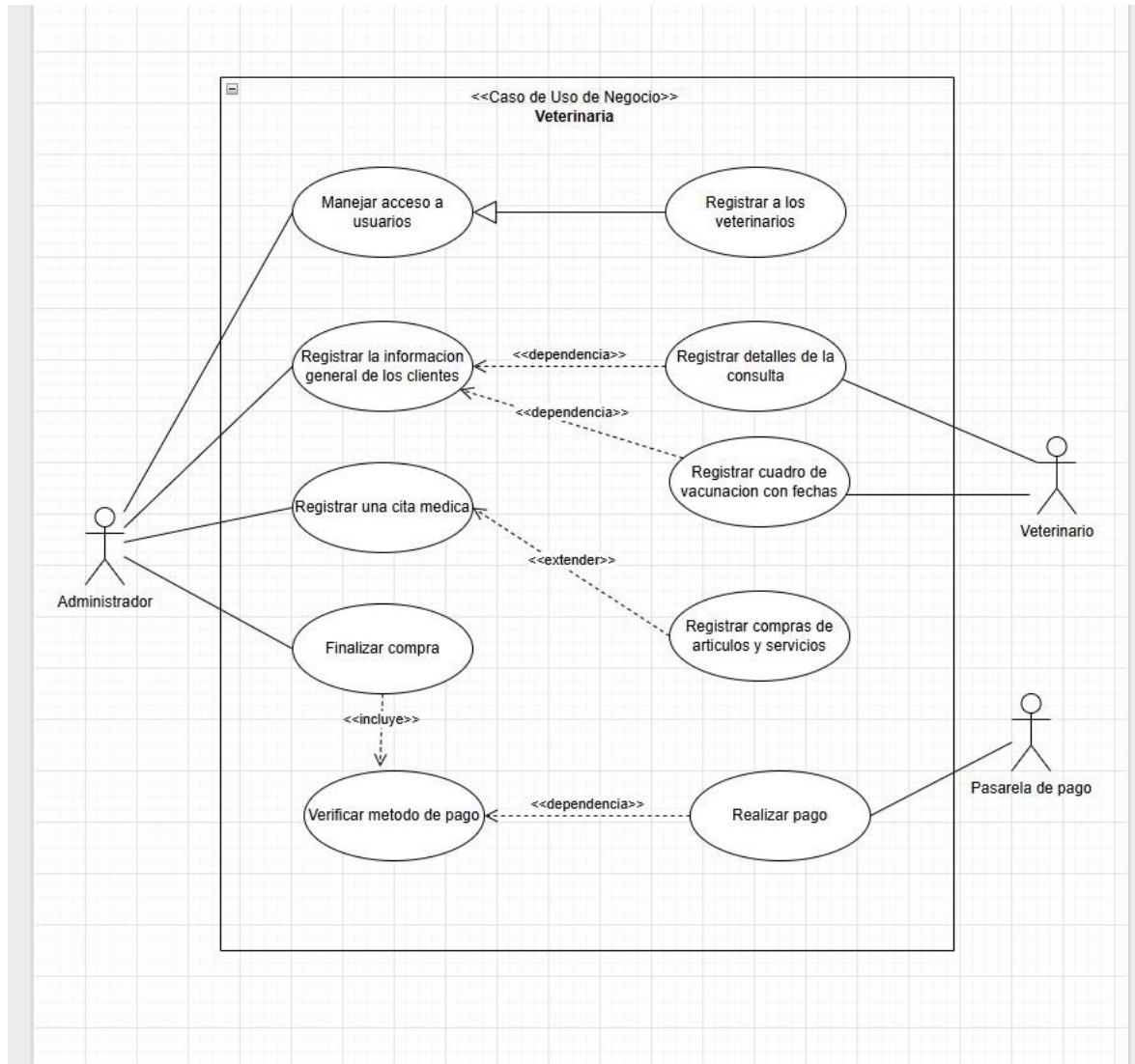
<b>jsonHorario</b>	String	Cadena JSON que representa el diccionario de horarios
<b>idCita</b>	Int	Identificador único de la cita creada o modificada
<b>motivosSeleccionados</b>	String	Motivos concatenados desde el CheckedListBox
<b>diagnostico</b>	String	Diagnóstico ingresado en el formulario
<b>estadoActual</b>	String	Estado dinámico de la cita (próxima/cancelada/pasada)
<b>mascotaSeleccionada</b>	Object	Instancia de la mascota seleccionada
<b>veterinarioSeleccionado</b>	Object	Instancia del veterinario seleccionado

### III. Diseño de la Solución

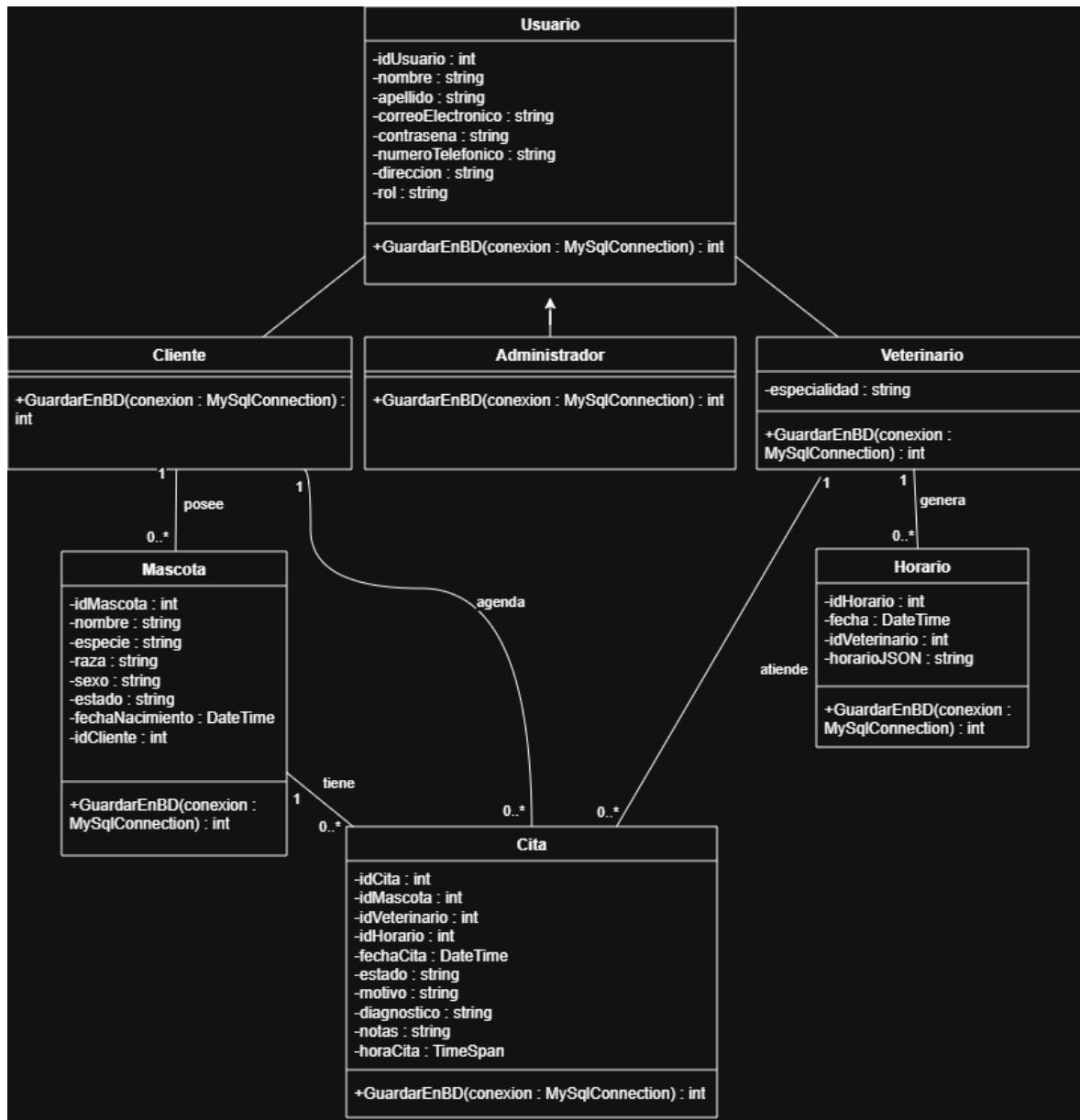




## Diagrama de caso de uso



## Diagrama de clases



## Archivo Script de base de datos en MySQL

```
-- MySQL dump 10.13 Distrib 8.0.44, for Win64 (x86_64)

--
-- Host: localhost Database: veterinariapatitasypelos
-----
-- Server version      8.0.26

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;

/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS *;;

/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION *;;

/*!50503 SET NAMES utf8 *;;

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE *;;

/*!40103 SET TIME_ZONE='+00:00' *;;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 *;;

/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 *;;

/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' *;;

/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 *;;

--

-- Table structure for table `administradores`


DROP TABLE IF EXISTS `administradores`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;
```

```

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `administradores` (
  `id_administrador` int NOT NULL AUTO_INCREMENT,
  `id_usuario` int NOT NULL,
  PRIMARY KEY (`id_administrador`),
  KEY `id_usuario` (`id_usuario`),
  CONSTRAINT `administradores_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES `usuarios` (`id_usuario`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `administradores`
-- 

LOCK TABLES `administradores` WRITE;
/*!40000 ALTER TABLE `administradores` DISABLE KEYS */;
INSERT INTO `administradores` VALUES (1,1);
/*!40000 ALTER TABLE `administradores` ENABLE KEYS */;
UNLOCK TABLES;

-- 
-- Table structure for table `citas`
-- 

DROP TABLE IF EXISTS `citas`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8mb4 */;
```

```

CREATE TABLE `citas` (
  `id_cita` int NOT NULL AUTO_INCREMENT,
  `fecha_cita` datetime NOT NULL,
  `motivo` varchar(255) DEFAULT NULL,
  `diagnostico` text,
  `tratamiento` text,
  `id_mascota` int NOT NULL,
  `id_cliente` int NOT NULL,
  PRIMARY KEY (`id_cita`),
  KEY `id_mascota` (`id_mascota`),
  KEY `id_cliente` (`id_cliente`),
  CONSTRAINT `citas_ibfk_1` FOREIGN KEY (`id_mascota`) REFERENCES `mascotas`(`id_mascota`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `citas_ibfk_2` FOREIGN KEY (`id_cliente`) REFERENCES `clientes`(`id_cliente`)
    ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `citas`
-- 

LOCK TABLES `citas` WRITE;
/*!40000 ALTER TABLE `citas` DISABLE KEYS */;
/*!40000 ALTER TABLE `citas` ENABLE KEYS */;
UNLOCK TABLES;

-- 
-- Table structure for table `clientes`

```

```
--
```

```
DROP TABLE IF EXISTS `clientes`;

/*!40101 SET @saved_cs_client    = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `clientes` (
  `id_cliente` int NOT NULL AUTO_INCREMENT,
  `id_usuario` int NOT NULL,
  PRIMARY KEY (`id_cliente`),
  KEY `id_usuario` (`id_usuario`),
  CONSTRAINT `clientes_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES `usuarios` (`id_usuario`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--
```

```
-- Dumping data for table `clientes`
```

```
--
```

```
LOCK TABLES `clientes` WRITE;
/*!40000 ALTER TABLE `clientes` DISABLE KEYS */;
INSERT INTO `clientes` VALUES (1,3),(2,4);
/*!40000 ALTER TABLE `clientes` ENABLE KEYS */;
UNLOCK TABLES;
```

```
--
```

```
-- Table structure for table `horarios`
```

```
--
```

```
DROP TABLE IF EXISTS `horarios`;

/*!40101 SET @saved_cs_client    =@@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `horarios` (
  `id_horario` int NOT NULL AUTO_INCREMENT,
  `fecha` date NOT NULL,
  `hora_inicio` time NOT NULL,
  `hora_fin` time NOT NULL,
  `disponible` tinyint(1) DEFAULT '1',
  PRIMARY KEY (`id_horario`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `horarios`
-- 

LOCK TABLES `horarios` WRITE;

/*!40000 ALTER TABLE `horarios` DISABLE KEYS */;

/*!40000 ALTER TABLE `horarios` ENABLE KEYS */;

UNLOCK TABLES;

-- 
-- Table structure for table `mascotas`
-- 

DROP TABLE IF EXISTS `mascotas`;
```

```

/*!40101 SET @saved_cs_client    = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `mascotas` (
  `id_mascota` int NOT NULL AUTO_INCREMENT,
  `nombre` varchar(100) NOT NULL,
  `especie` varchar(50) DEFAULT NULL,
  `raza` varchar(50) DEFAULT NULL,
  `sexo` enum('Macho','Hembra') DEFAULT NULL,
  `fecha_nacimiento` date DEFAULT NULL,
  `id_cliente` int NOT NULL,
  PRIMARY KEY (`id_mascota`),
  KEY `id_cliente` (`id_cliente`),
  CONSTRAINT `mascotas_ibfk_1` FOREIGN KEY (`id_cliente`) REFERENCES
  `clientes` (`id_cliente`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `mascotas`
-- 

LOCK TABLES `mascotas` WRITE;
/*!40000 ALTER TABLE `mascotas` DISABLE KEYS */;
/*!40000 ALTER TABLE `mascotas` ENABLE KEYS */;
UNLOCK TABLES;

-- 
-- Table structure for table `usuarios`
-- 


```

```

DROP TABLE IF EXISTS `usuarios`;

/*!40101 SET @saved_cs_client    = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `usuarios` (
  `id_usuario` int NOT NULL AUTO_INCREMENT,
  `nombre` varchar(100) NOT NULL,
  `apellido` varchar(100) NOT NULL,
  `correo_electronico` varchar(150) NOT NULL,
  `contrasena` varchar(100) NOT NULL,
  `numero_telefonico` varchar(20) DEFAULT NULL,
  `direccion` varchar(200) DEFAULT NULL,
  `rol` enum('Cliente','Veterinario','Administrador') NOT NULL,
  PRIMARY KEY (`id_usuario`),
  UNIQUE KEY `correo_electronico` (`correo_electronico`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `usuarios`
-- 

LOCK TABLES `usuarios` WRITE;
/*!40000 ALTER TABLE `usuarios` DISABLE KEYS */;

INSERT INTO `usuarios` VALUES
(1,'Jaime','Aldana','jaime@aldana.com','1234','45784578','San
Salvador','Administrador'),
(2,'Alicia','Rivas','alicia@rivas.com','4321','457845784578','San
Miguel','Veterinario'),
(3,'Marcos','Cisneros','marcos@cisneros.com','9876','12345678','La

```

```

Libertad','Cliente'),(4,'Maria','Gimenez','maria@gimenez.com','9878','72399079','San
Salvador','Cliente');

/*!40000 ALTER TABLE `usuarios` ENABLE KEYS */;

UNLOCK TABLES;

--


-- Table structure for table `veterinarios`


--


DROP TABLE IF EXISTS `veterinarios`;

/*!40101 SET @saved_cs_client      = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `veterinarios` (
  `id_veterinario` int NOT NULL AUTO_INCREMENT,
  `id_usuario` int NOT NULL,
  PRIMARY KEY (`id_veterinario`),
  KEY `id_usuario` (`id_usuario`),
  CONSTRAINT `veterinarios_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES
  `usuarios` (`id_usuario`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;

--


-- Dumping data for table `veterinarios`


--


LOCK TABLES `veterinarios` WRITE;

/*!40000 ALTER TABLE `veterinarios` DISABLE KEYS */;

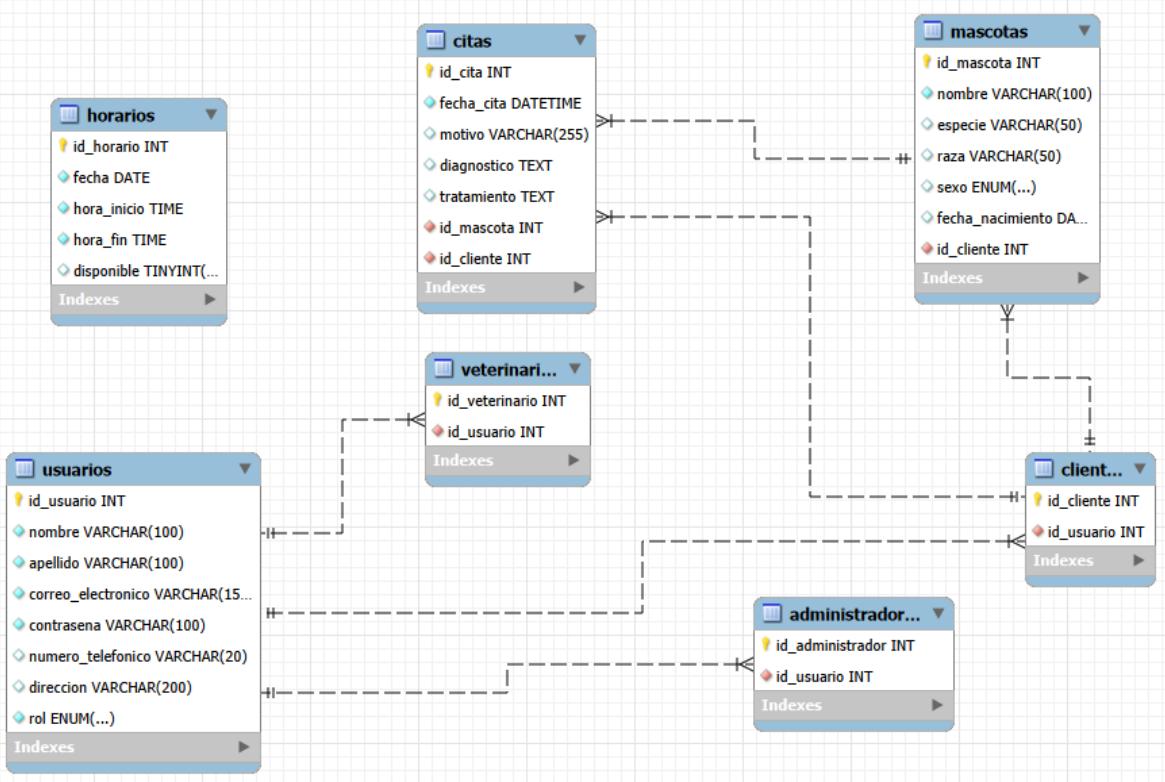
```

```
INSERT INTO `veterinarios` VALUES (1,2);
/*!40000 ALTER TABLE `veterinarios` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2025-11-13 23:09:42
```

# Imagen del modelo Entidad – Relacion de la Base de datos en MySQL



# Manual técnico y de usuario

## Manual Técnico

### Requerimientos técnicos

#### Software

- Navegador web: Para poder tener acceso al proyecto almacenado en GitHub.
- Gestor de Base de datos: (MySQL) para la administración de los registros almacenados.
- Entorno de desarrollo: (Visual Studio Community) para clonar el proyecto y tener acceso al formulario.

#### Hardware

- Una computadora completa: esto incluye mouse, teclado, CPU, monitor. En el caso de una laptop funcionamiento de la pantalla, el teclado y el mouse táctil.
- Procesador: CPU **64-bit x86 (Intel/AMD)**, al menos **1.8 GHz**, de 2 núcleos (mejor 4 núcleos).
- Arquitectura: Sistema **64 bits** (no sirve Windows de 32 bits).
- RAM: 4 GB.

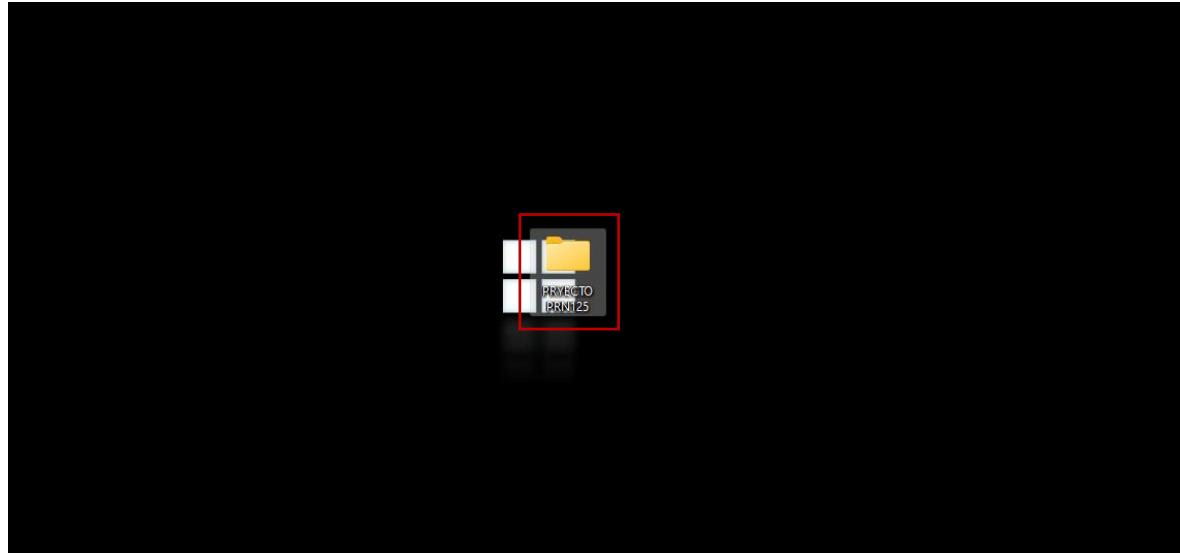
#### Herramientas utilizadas para el desarrollo

- Navegador Web: Para tener acceso a GitHub y poder almacenar el repositorio.
- Git: para el manejo de versiones, crear ramas, hacer commits, hacer push para subir los cambios realizados y poder trabajar en equipo.
- Visual Studio Community: para la creación de los formularios y crear código para los mismo y conectar la base de datos.

- MySQL: como base de datos para la administración de datos registrados en los formularios.

### Clonación del repositorio.

Creamos una nueva carpeta en nuestro escritorio donde guardaremos todas las carpetas del repositorio.



Entramos al link del repositorio en GitHub <https://github.com/ab17005-collab/TareaDeDesarrollo-Proyecto>

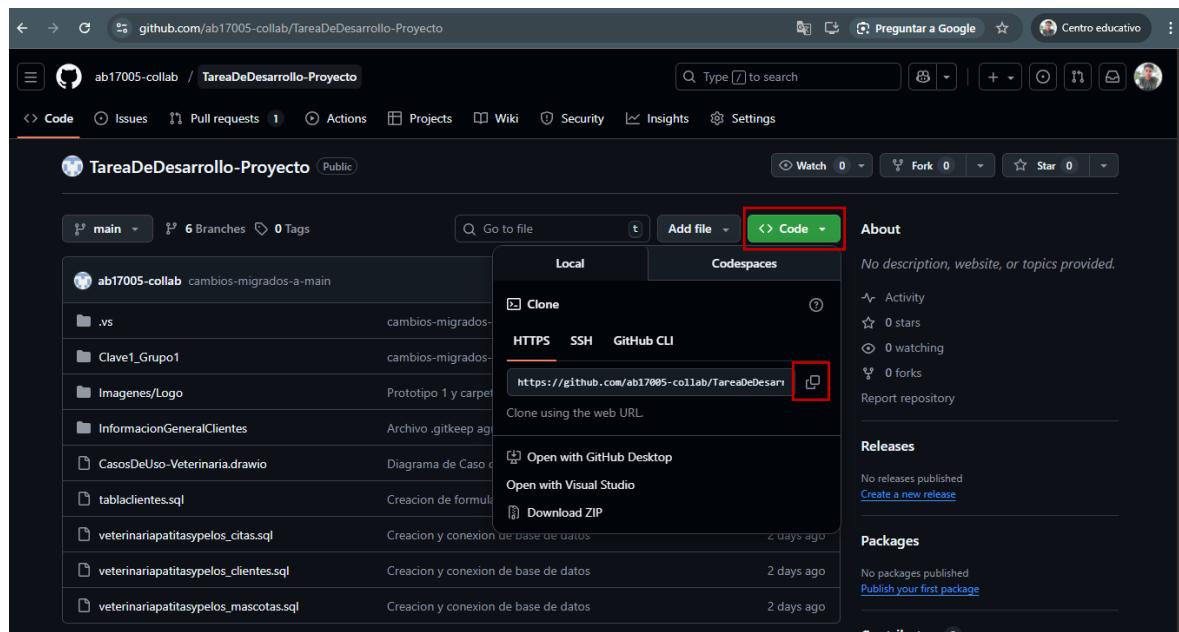
The screenshot shows a GitHub repository page for 'TareaDeDesarrollo-Proyecto'. The repository is public and has 6 branches and 0 tags. The main branch is selected. The commit history shows 17 commits, with the most recent being 'cambios-migrados-a-main' by user 'ab17005-collab' 4 hours ago. The repository contains several files and folders:

- .vs
- Clave1\_Grupo1
- Imagenes/Logo
- InformacionGeneralCientes
- CasosDeUso-Veterinaria.drawio
- tablaclientes.sql
- veterinariapatitasypelos\_citas.sql
- veterinariapatitasypelos\_clientes.sql
- veterinariapatitasypelos\_mascotas.sql

The repository has 0 stars, 0 forks, and 0 releases. There is no description or website provided.

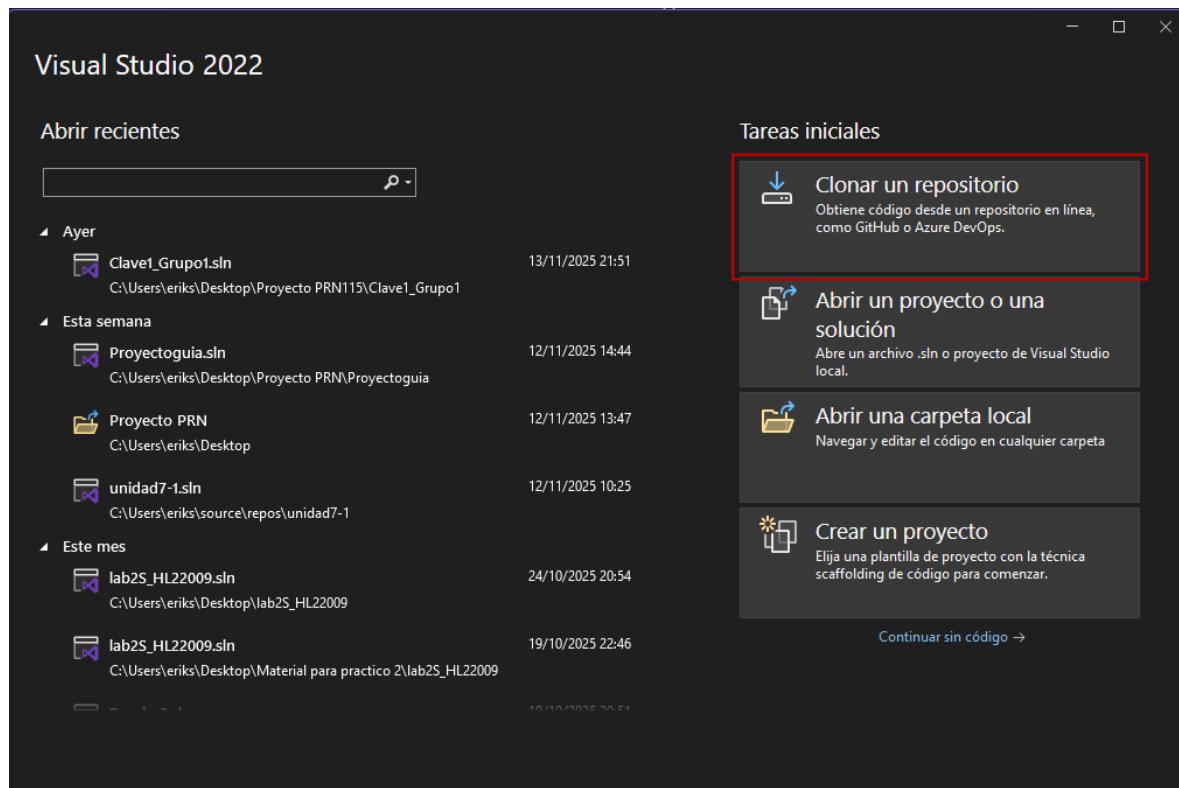
## PROYECTO PROGRAMACION

Copiamos el link del repositorio para poder clonarlo



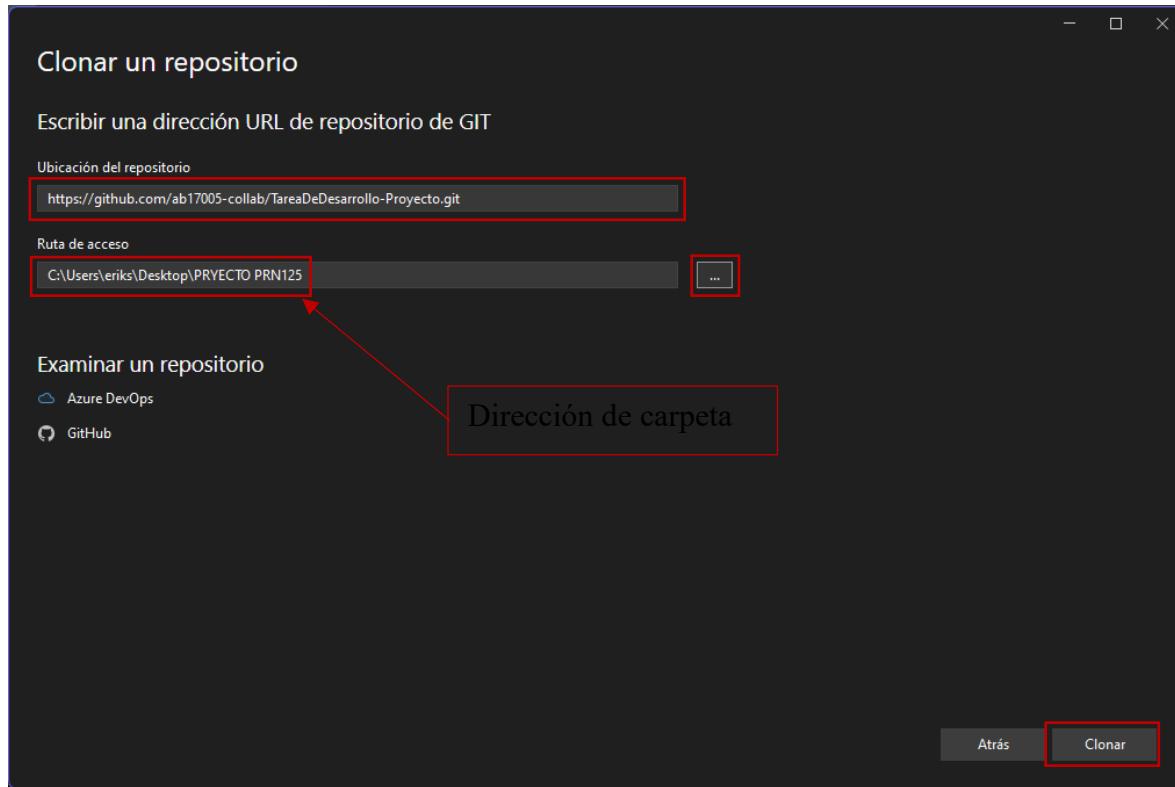
The screenshot shows a GitHub repository page for 'TareaDeDesarrollo-Proyecto'. The 'Code' dropdown menu is open, highlighting the 'Clone' option. A red box highlights the 'HTTPS' link under the Clone section, which contains the URL <https://github.com/ab17005-collab/TareaDeDesarrollo-Proyecto>.

Abrimos Visual Studio Community en nuestra computadora y seleccionamos la opción de clonar repositorio.

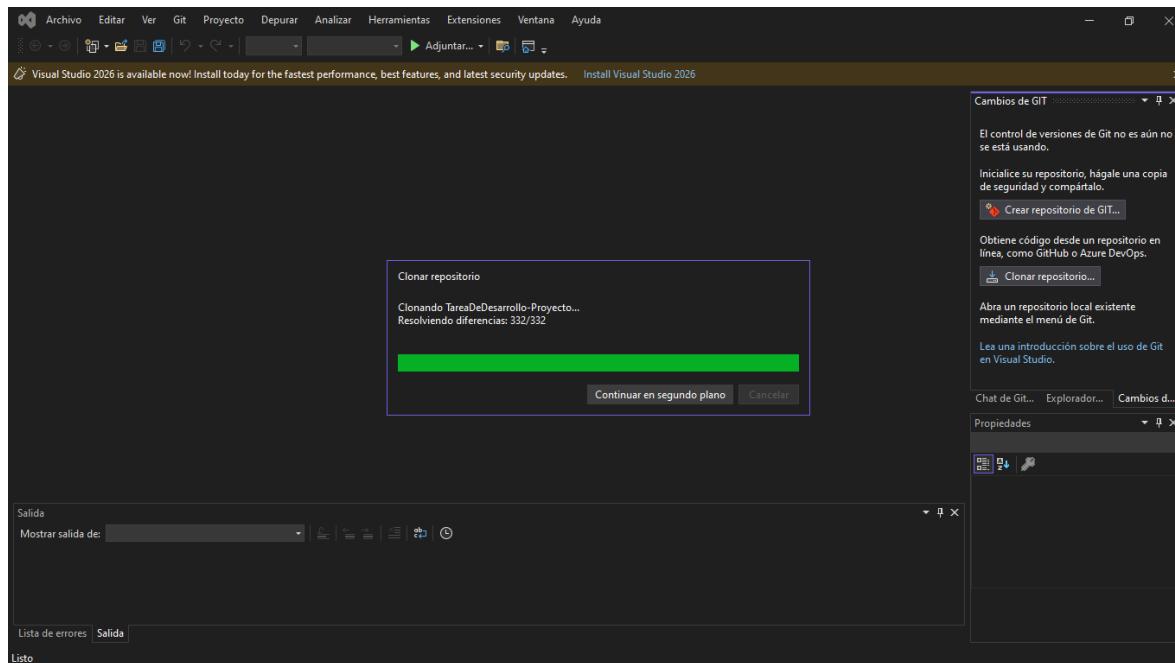


The screenshot shows the Visual Studio 2022 start screen. The 'Tareas iniciales' (Initial tasks) pane is open, with the 'Clonar un repositorio' (Clone a repository) task highlighted by a red box. This task is described as obtaining code from an online repository like GitHub or Azure DevOps. Other tasks shown include 'Abrir un proyecto o una solución' (Open a project or solution) and 'Crear un proyecto' (Create a project).

Pegamos el link que copiamos del repositorio en GitHub, elegimos la carpeta que creamos al principio para que todo se guarde en la misma carpeta y hacemos click en la opción clonar

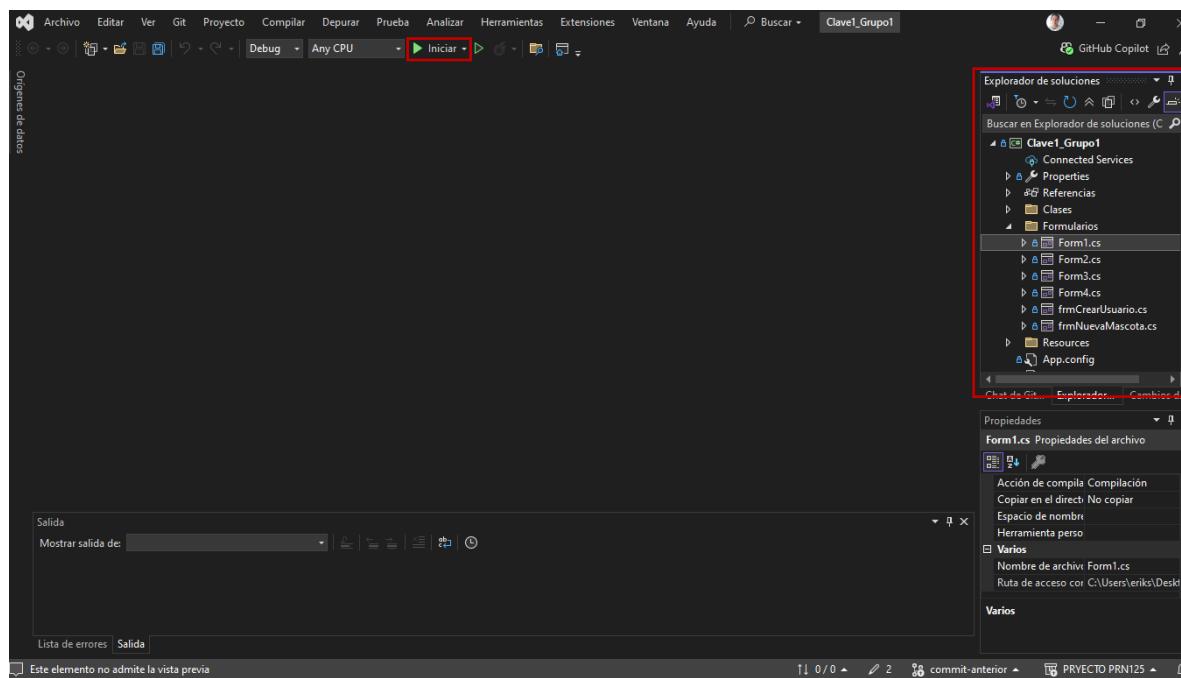


Automáticamente empezara a clonar el repositorio.

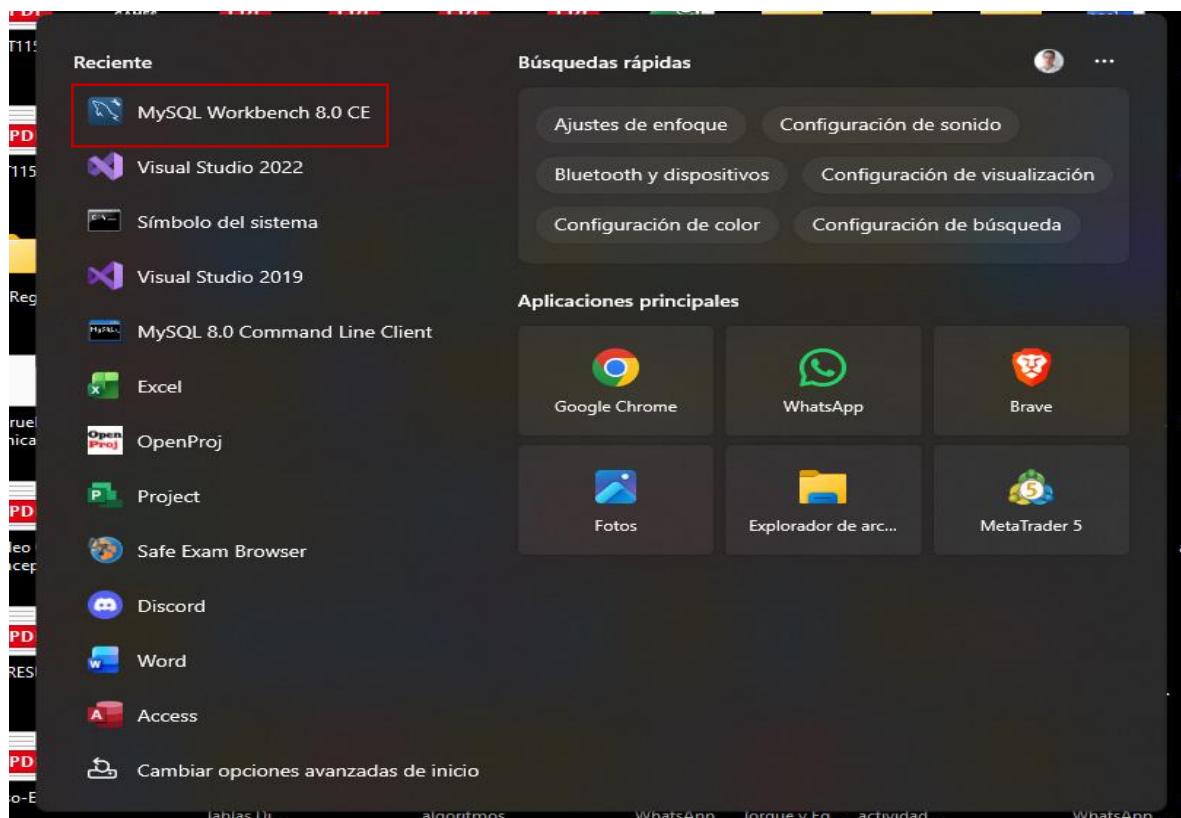


## PROYECTO PROGRAMACION

En el explorador de archivos tendremos los diferentes formularios para poder ver una vista previa antes de ejecutar nuestro Window Form. Damos click en iniciar para para ejecutar el programa.



Para ver la base de datos abrimos la aplicación MySQL Workbech 8.0 CE



Del lado izquierdo podremos visualizar la base de datos llamada **veterinariapatitasypelos**, con el siguiente código podemos ver los datos que están en la tabla **usuario**. Luego hacemos clic en el símbolo de rayo.

The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays the database schema with a red box highlighting the 'veterinariapatitasypelos' database. The 'Tables' section under it shows tables like 'administradores', 'ctas', 'clientes', 'horarios', 'mascotas', 'usuarios', and 'veterinarios'. The 'Information' pane below shows the schema is 'veterinariapatitasypelos'. In the center, the 'Result Grid' pane displays the data from the 'usuarios' table:

	<b>id_usuario</b>	<b>nombre</b>	<b>apellido</b>	<b>correo_electronico</b>	<b>contraseña</b>	<b>numero_telefonico</b>	<b>direccion</b>	<b>rol</b>
▶	1	Jaime	Aldana	jaime@aldana.com	1234	45784578	San Salvador	Administrador
▶	2	Alicia	Rivas	alicia@rivas.com	4321	457845784578	San Miguel	Veterinario
▶	3	Marcos	Csneros	marcos@csneros.com	9876	12345678	La Libertad	Cliente
▶	4	Maria	Gmenez	maria@gmenez.com	9878	72399079	San Salvador	Cliente
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Manual de usuario del Sistema de Gestión Veterinaria

Sistema de gestión de citas para la clínica veterinaria “Patitas y Pelos”

### Introducción

A continuación, se brindan los detalles de usuario de la solución, basado en los requerimientos solicitados para este caso específico, en el que se trabajó en un sistema de manejo de citas veterinarias, para lo cual se utilizó la tecnología de WinForms en lenguaje C#.

### Objetivos de la solución

Los objetivos de la solución es registrar a los usuarios que estarán involucrados en el funcionamiento, que pueden ser clientes, administradores y veterinarios. El sistema clasifica cada tipo de usuario para manejar sus actividades basado en clases y tablas de la base de datos. El sistema permite a clientes y agentes de la veterinaria gestionar las citas.

### Alcances

- Registrar usuarios por tipo ya sea cliente, veterinario o administrador.
- Gestiona las mascotas de los clientes permitiendo enlazar la información de dicha mascota con el perfil del cliente disponibles en cualquier cita a agendar.
- Se lleva un control de horarios disponibles por segmentos de hora, por fecha y por veterinario, para agendar citas de manera eficiente y automatizada.

- Cancelación y modificación de citas, son gestionadas en la solución ya que se lleva un control de registros en la base de datos para cada cita y se define el estado de la misma.

## Arquitectura del sistema

Sistema cliente–servidor. - Cliente: WinForms .NET Framework - Servidor: MySQL -

Comunicación: conector MySql.Data

Se gestiona el sistema Cliente-Servidor mediante el consumo de la base de datos a través de la aplicación y el seguimiento externo de la base utilizando MySQL Workbench 8.0 CE.

El sistema se comunica con la base de datos a través de la solución construida en WinForms .NET utilizando el sistema de comunicación MySql.Data

## Requerimientos

- Windows 10+
- MySQL Server 8.0+
- .NET Framework 4.8
- Visual Studio 2019 o superior

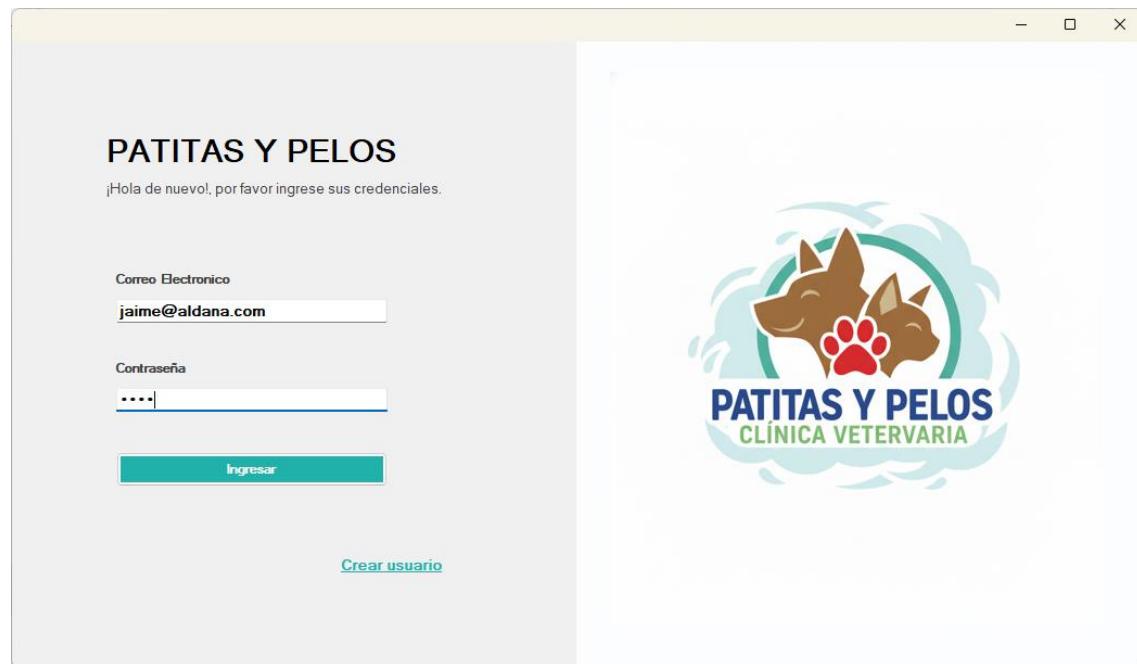
Dependencia NuGet: MySql.Data, Newtonsoft.Json

## Arquitectura del software

### Modelo de capas de formularios

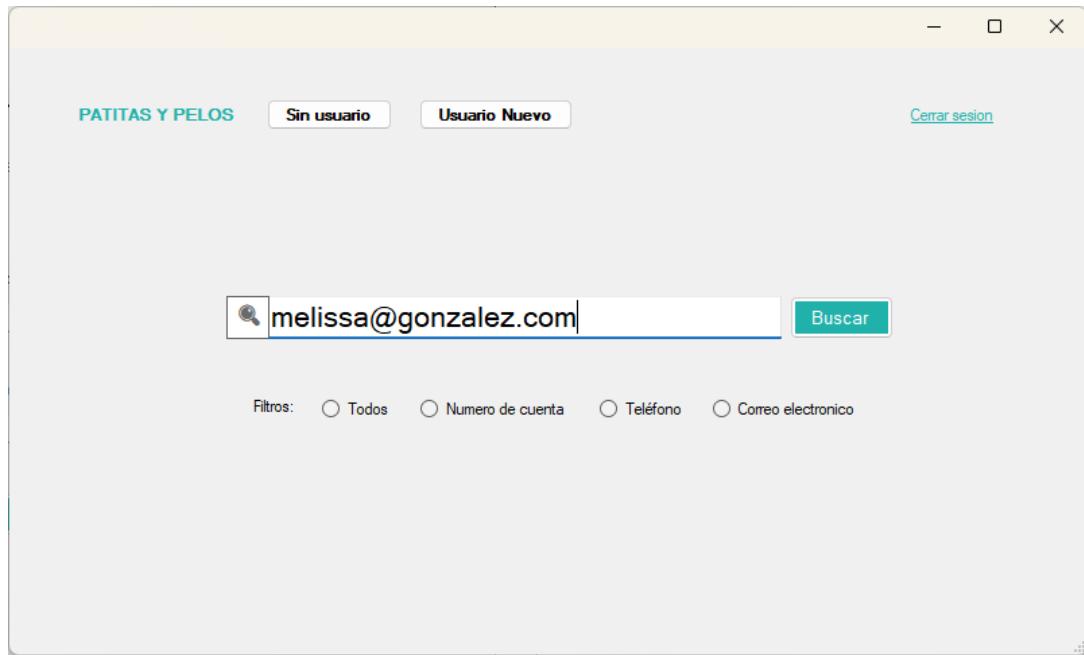
Formularios:

**Form1 → Log In**



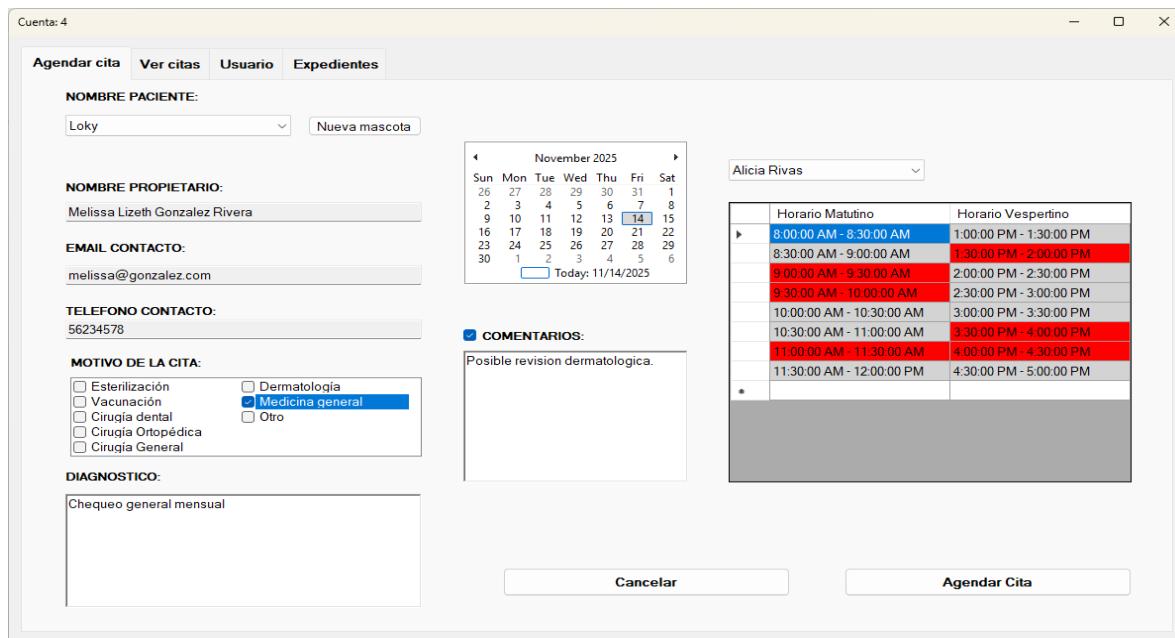
Se debe ingresar con un correo de usuario administrador o veterinario.

El primer usuario administrador es [jaime@aldana.com](mailto:jaime@aldana.com) y contraseña: 1234, pero se pueden crear más usuarios administradores en la aplicación.

**Form2 → Búsqueda de cliente**

En el cuadro de búsqueda se debe ingresar el correo electrónico del cliente a buscar.

El correo electrónico debe ser de un usuario tipo cliente.

**Form3 → Gestión de citas.**

1. Se debe seleccionar la mascota o registrar una mascota nueva y seleccionarla. Se puede registrar una mascota nueva del cliente mediante el botón Nueva Mascota y posteriormente seleccionarla.
2. Marcar uno o varios motivos de la cita veterinaria. Agregar el comentario de diagnóstico para el cual se agenda la cita.
3. Seleccionar una fecha en el control de calendario, seleccionar un veterinario de la lista, los cuales tienen cada uno un seguimiento de que días y horas tienen espacios para agendarse la cita.
4. Agregar un comentario opcional y dar clic al botón Agendar Cita.
5. En caso que no se dese crear la cita se puede cancelar el proceso mediante el botón Cancelar.

#### Form4 → Detalles de cita.

**Veterinaria Patitas y Pelos**

**Detalles de la cita**

22/11/2025 14:30 Cita #: 9

Información de la mascota		Información del dueño	
	Nombre Loky	Especie Perro	Nombre Melissa Lizeth Gonzalez Rivera
Raza Pastor Aleman	Edad 1 años	E-mail melissa@gonzalez.com	Teléfono 56234578
Sexo Macho		Lugar La Union	

**Detalles de la cita**

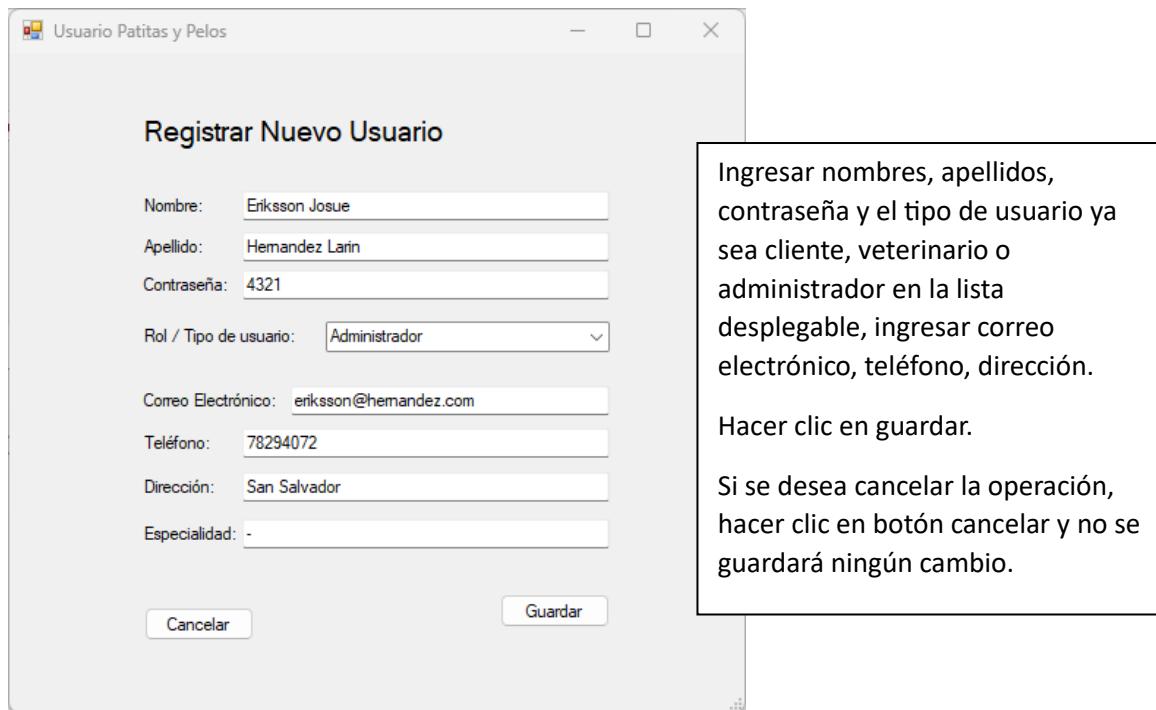
Veterinario Alicia Rivas	Duración 30 minutos	Diagnóstico Diagnóstico
Motivo de la cita Dermatología	Notas adicionales Posible revisión ortopédica.	

**Botones:**

- Cancelar
- Modificar / Reagendar
- Listo

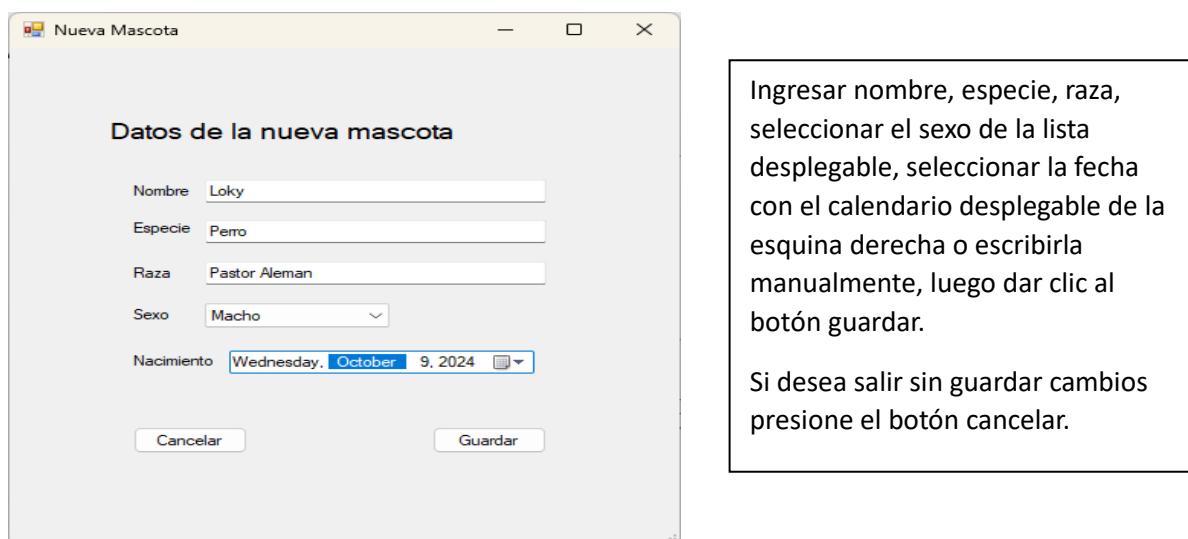
Se despliega una ventana con los detalles de la cita que ya ha sido registrada en la base de datos, ofreciendo las opciones al administrador de cancelar la cita o modificarla.

### **frmCrearUsuario → Crea usuario por tipo.**



Ingresar los datos correspondientes del nuevo usuario y hacer clic en guardar.

### **frmNuevaMascota → Agrega una nueva mascota al perfil del cliente.**



## Enlace del videotutorial

<https://youtu.be/JnPSZHL6Q5o>

## Capturas de uso de GitHub

```
Command Prompt
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jaime\cd "C:\Users\jaime\OneDrive\Documentos\UES\2025\Ciclo II\Programacion I\Evaluaciones\Proyecto"
C:\Users\jaime\OneDrive\Documentos\UES\2025\Ciclo II\Programacion I\Evaluaciones\Proyecto>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified: Clavel_Grupol/.vs\Clavel_Grupol\CapilotIndices\17_14_878_3237\CodeChunks.db
    modified: Clavel_Grupol/.vs\Clavel_Grupol\CapilotIndices\17_14_878_3237\SemanticSymbols.db
  deleted: Clavel_Grupol/.vs\Clavel_Grupol\FileContentIndex\137787c-f-87f6-42c4-94ff-38a1dd4a3fe4.vsidx
  deleted: Clavel_Grupol/.vs\Clavel_Grupol\FileContentIndex\e0fadfa-41b9-0420-ab91-e06751e0f5dd.vsidx
  deleted: Clavel_Grupol/.vs\Clavel_Grupol\FileContentIndex\60791821-1959-4df2-a86e-c91145457567.vsidx
  deleted: Clavel_Grupol/.vs\Clavel_Grupol\FileContentIndex\fc630f7-932d-40e5-9aaa-31fb67b203f8.vsidx
  modified: Clavel_Grupol/.vs\Clavel_Grupol\1v17.suo
  modified: Clavel_Grupol/.vs\Clavel_Grupol\1v17\DocumentLayout.backup.json
  modified: Clavel_Grupol/.vs\Clavel_Grupol\1v17\DocumentLayout.json
  modified: Clavel_Grupol/Form2.Designer.cs
  modified: Clavel_Grupol/Form2.cs
  modified: Clavel_Grupol/Form2.resx
  modified: Clavel_Grupol/bin/Debug/Clavel_Grupol.exe
  modified: Clavel_Grupol/bin/Debug/Clavel_Grupol.pdb
  modified: Clavel_Grupol/obj/Debug/Clavel_Grupol.Form2.resources
  Clavel_Grupol/obj/Debug/Clavel_Grupol.csproj.GenerateResource.cache
  modified: Clavel_Grupol/obj/Debug/Clavel_Grupol.suo
  modified: Clavel_Grupol/obj/Debug/Clavel_Grupol.vbproj
  modified: Clavel_Grupol/obj/Debug/DesignTimeResolveAssemblyReferences.cache

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Clavel_Grupol/.vs\Clavel_Grupol\FileContentIndex\0ef12010-0d91-0461-8798-2079dc4ff0ba.vsidx
    Clavel_Grupol/.vs\Clavel_Grupol\FileContentIndex\51971e20-2290-4ef3-bde7-38a5bb1d29.vsidx
    Clavel_Grupol/.vs\Clavel_Grupol\FileContentIndex\607953ab-e724f-4efc-b82b-135f6bb5d584.vsidx
    Clavel_Grupol/.vs\Clavel_Grupol\FileContentIndex\679191821-1959-4df2-a86e-c91145457567.vsidx

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\jaime\OneDrive\Documentos\UES\2025\Ciclo II\Programacion I\Evaluaciones\Proyecto>git add .

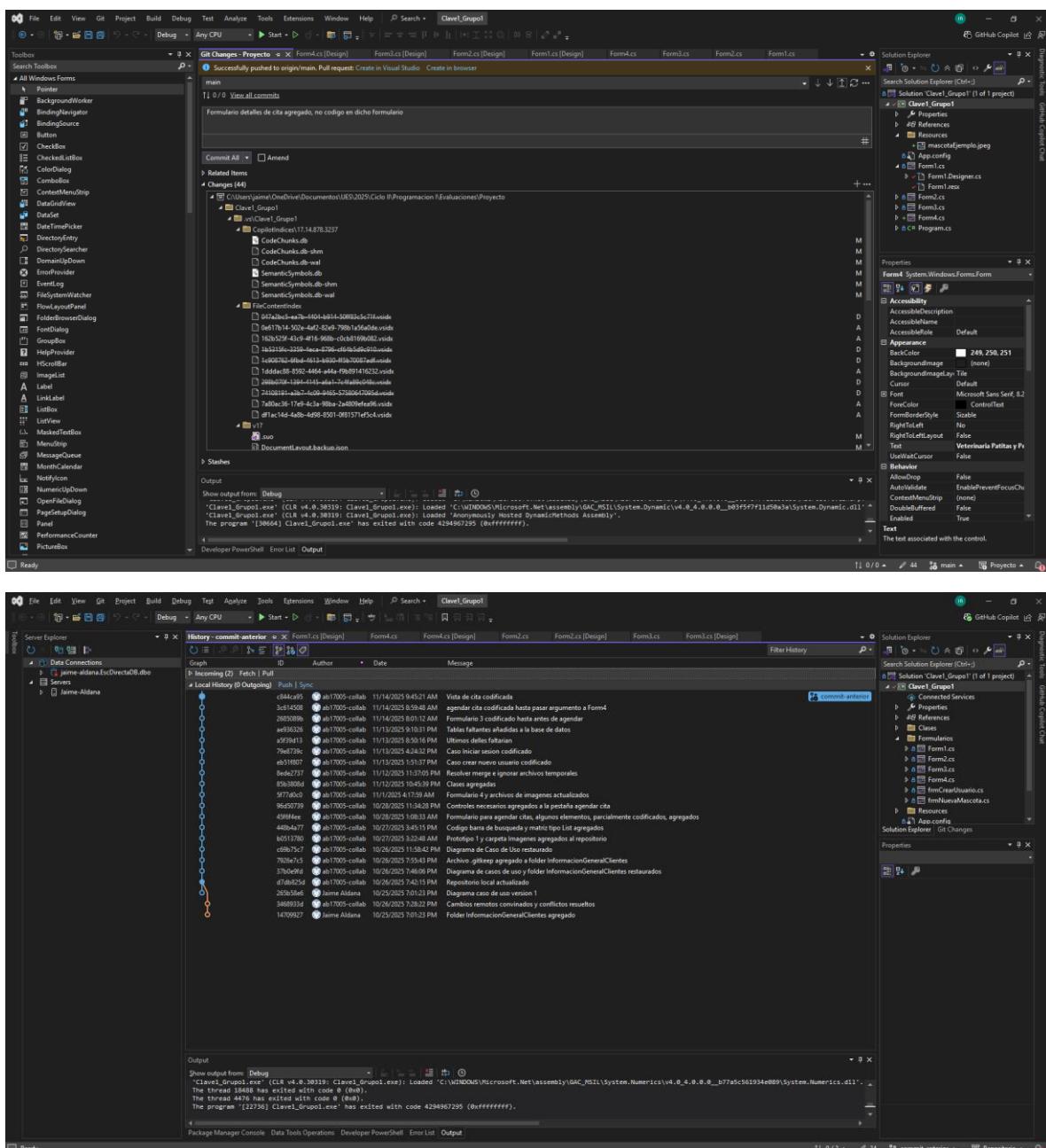
C:\Users\jaime\OneDrive\Documentos\UES\2025\Ciclo II\Programacion I\Evaluaciones\Proyecto>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: Clavel_Grupol/.vs\Clavel_Grupol\CapilotIndices\17_14_878_3237\CodeChunks.db
    modified: Clavel_Grupol/.vs\Clavel_Grupol\CapilotIndices\17_14_878_3237\SemanticSymbols.db

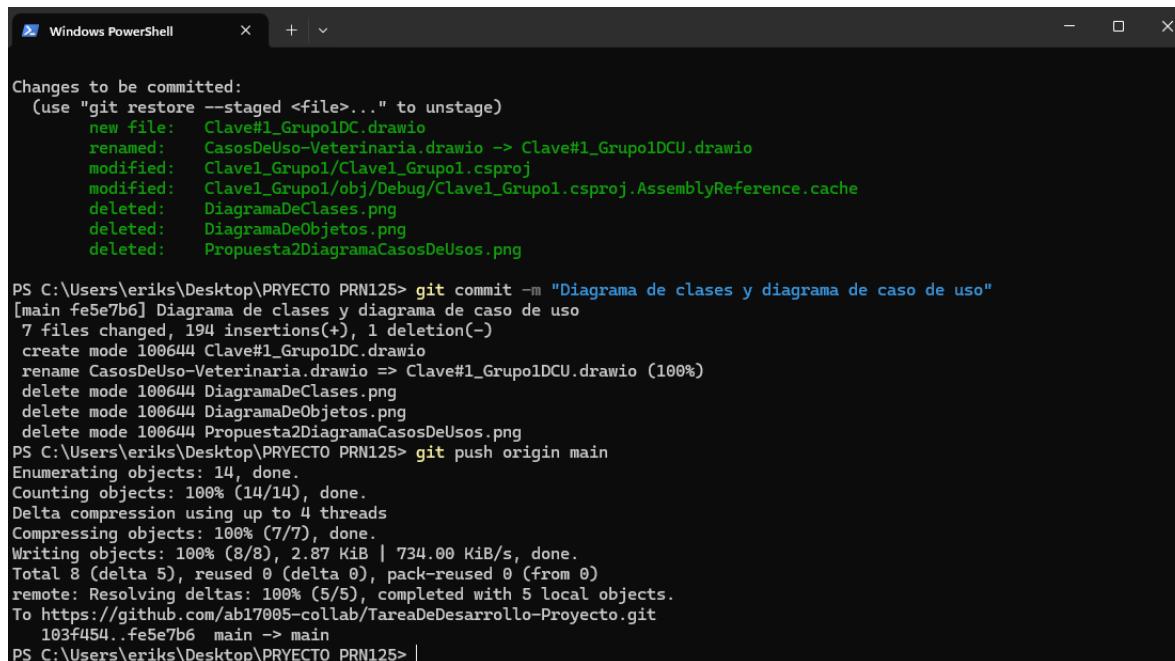
Command Prompt
(use "git restore --staged <file>..." to unstage)
  modified: Clavel_Grupol/.vs\Clavel_Grupol\CapilotIndices\17_14_878_3237\CodeChunks.db
  modified: Clavel_Grupol/.vs\Clavel_Grupol\CapilotIndices\17_14_878_3237\SemanticSymbols.db

(Clave...)
```

# PROYECTO PROGRAMACION



## PROYECTO PROGRAMACION



```
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
  new file:  Clave#1_Grupo1DC.drawio
  renamed:   CasosDeUso-Veterinaria.drawio -> Clave#1_Grupo1DCU.drawio
  modified:  Clave1_Grupo1/Clave1_Grupo1.csproj
  modified:  Clave1_Grupo1/obj/Debug/Clave1_Grupo1.csproj.AssemblyReference.cache
  deleted:   DiagramaDeClases.png
  deleted:   DiagramaDeObjetos.png
  deleted:   Propuesta2DiagramaCasosDeUsos.png

PS C:\Users\eriks\Desktop\PRYECTO PRN125> git commit -m "Diagrama de clases y diagrama de caso de uso"
[main fe5e7b6] Diagrama de clases y diagrama de caso de uso
 7 files changed, 194 insertions(+), 1 deletion(-)
 create mode 100644 Clave#1_Grupo1DC.drawio
 rename CasosDeUso-Veterinaria.drawio => Clave#1_Grupo1DCU.drawio (100%)
 delete mode 100644 DiagramaDeClases.png
 delete mode 100644 DiagramaDeObjetos.png
 delete mode 100644 Propuesta2DiagramaCasosDeUsos.png
PS C:\Users\eriks\Desktop\PRYECTO PRN125> git push origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 2.87 KiB | 734.00 KiB/s, done.
Total 8 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/ab17005-collab/TareaDeDesarrollo-Proyecto.git
  103f454..fe5e7b6  main -> main
PS C:\Users\eriks\Desktop\PRYECTO PRN125> |
```

## Conclusiones

El desarrollo del proyecto nos permitió aplicar de forma práctica muchos de los contenidos vistos en Programación I, pasando de la teoría de POO a una solución real para la veterinaria “*Patitas y Pelos*”. A lo largo del proyecto diseñamos y programamos un sistema de escritorio en **Windows Forms C#** conectado a una base de datos **MySQL**, lo que nos ayudó a comprender mejor el manejo de clases, herencia, validaciones y operaciones básicas con datos persistentes. Además, el uso de **GitHub** y el trabajo por ramas fortaleció nuestra organización como grupo y nos dio una idea más clara de cómo se colabora en proyectos de software reales. Finalmente, consideramos que esta experiencia no solo cumplió con los requisitos académicos, sino que también nos dejó habilidades técnicas y de trabajo en equipo que serán útiles en cursos posteriores y en el ámbito profesional.

## Recomendaciones

- Fortalecer las validaciones en los formularios, asegurando que los campos obligatorios estén completos y que datos como correo electrónico, teléfono y nombres tengan el formato correcto, evitando así registros incompletos o erróneos en la base de datos.
- Mejorar el manejo de errores y mensajes al usuario, de manera que cuando falle la conexión a MySQL u ocurra algún problema con el sistema, se muestren mensajes claros y entendibles, en lugar de errores técnicos difíciles de interpretar.
- Organizar el proyecto en capas lógicas (presentación, lógica de negocio y acceso a datos), lo que facilitaría el mantenimiento, la lectura del código y posibles mejoras futuras sin tener que modificar todo el programa.
- Mejorar el uso de GitHub en el trabajo en equipo, utilizando ramas por funcionalidad, mensajes de commit más descriptivos y revisiones entre compañeros, para tener un control de versiones más profesional y un historial de cambios más claro.
- Pulir la interfaz gráfica en Windows Forms, haciendo el diseño más intuitivo y amigable para el usuario, y acompañar el sistema con un manual de usuario actualizado y pruebas más sistemáticas, para asegurar que la aplicación sea fácil de usar y funcione de forma estable.

## Fuentes

Universidad de El Salvador. (2025). *Guía de laboratorio N° 15. Unidad 7: Manejo de datos.* [https://campus.ues.edu.sv/pluginfile.php/6216915/mod\\_label/intro/Guia15-PRN115-Unidad%207%20v3.pdf](https://campus.ues.edu.sv/pluginfile.php/6216915/mod_label/intro/Guia15-PRN115-Unidad%207%20v3.pdf)

Universidad de El Salvador. (2025). *Guía de práctica de Windows forms y estructuras selectivas.* [https://campus.ues.edu.sv/pluginfile.php/6043549/mod\\_label/intro/GuiaPracticaU2.pdf](https://campus.ues.edu.sv/pluginfile.php/6043549/mod_label/intro/GuiaPracticaU2.pdf)

Universidad de El Salvador. (2025). *TÉCNICAS DE DOCUMENTACIÓN.* [https://campus.ues.edu.sv/pluginfile.php/5989716/mod\\_label/intro/VIRFIA\\_PRN115\\_U1\\_CT\\_1.2-1.2\\_ME%202023.pdf?time=1689455692321](https://campus.ues.edu.sv/pluginfile.php/5989716/mod_label/intro/VIRFIA_PRN115_U1_CT_1.2-1.2_ME%202023.pdf?time=1689455692321)

Ing. William Vides. (2017, marzo 28). *Introducción a UML* [Video]. YouTube. <https://www.youtube.com/watch?v=uuGD9rWzUxQ&t=903s>

Legibilidad del código. (2025). learn.microsoft.com. <https://learn.microsoft.com/es-es/power-apps/guidance/coding-guidelines/code-readability>

*UML Class Diagram Tutorial.* (2025). Visual Paradigm. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

Denis, P. (2025). *Section 5: Object Oriented Programming (OOP).* Udemy.com. <https://www.udemy.com/course/complete-csharp-masterclass/learn/lecture/43720936#overview>

Yogesh, D. (2025). *The Complete UML Course: Learn to Design UML Diagrams.* Udemy.com. <https://www.udemy.com/course/unified-modeling-language-uml-course-uml-diagram-software-enginnering/learn/lecture/37224902#overview>