

Proyecto:
Sistema de gestión de Veterinaria
“Patitas”

Autor: Romeo Gago Lucia

Fecha: 24/03/2025

Materia: Programación Orientada a Objetos

Final libre



1. Planeamiento del proyecto:

1.1 Alcance. Lo que incluye y no incluye el alcance del sistema de gestión.

Este documento describe el alcance del sistema de gestión veterinaria "Patitas", detallando las funcionalidades incluidas y excluidas. El sistema está diseñado para dos perfiles de usuario principales: veterinarios y clientes (dueños de mascotas).

- Perfil del Veterinario: Destinado a la gestión de la información de clientes y sus mascotas, así como a la actualización y administración de los productos ofrecidos a través del sistema.
- Perfil del Cliente: Orientado a los dueños de mascotas que deseen registrar, modificar o eliminar la información de sus animales, y realizar compras de productos.

Datos de Mascotas

- Cliente: La aplicación permitirá a los clientes registrar, eliminar, visualizar y actualizar los datos identificativos de sus mascotas (nombre, edad, raza). Los clientes podrán almacenar y visualizar una breve descripción del historial de su mascota, pero no podrán modificar el historial médico.
- Veterinario: La aplicación permitirá a los veterinarios registrar, eliminar, visualizar y actualizar los datos identificativos de las mascotas (nombre, edad, raza) e historial médico completo.

Datos de Productos

- Cliente: La aplicación permitirá a los clientes visualizar los productos disponibles, su stock y precio, comprar productos y gestionar un inventario personal de los productos adquiridos.
- Veterinario: La aplicación permitirá a los veterinarios registrar, eliminar, visualizar y actualizar los datos de los productos ofrecidos.

Datos de Clientes

- Veterinario: La aplicación permitirá a los veterinarios registrar, eliminar, visualizar y actualizar los datos identificativos de los clientes y sus mascotas.



1.2 Actores.

El sistema requiere de la presencia de dos actores principales:

Nombre	Descripción
Cliente	Usuario que utiliza el sistema para registrar y gestionar una o varias mascotas. Usuario que utiliza el sistema para visualizar y comprar productos de veterinaria.
Profesionales(Veterinario)	Usuarios que utilizan el sistema para registrar y gestionar sus clientes (clientes y sus mascotas). Usuarios que utilizan el sistema para registrar y gestionar los productos ofrecidos.

1.3 Casos de uso.

Caso de Uso 1:

Código	CU1
Nombre	Gestionar datos de las mascotas
Actores	Usuario cliente
Funcion	Registrar los datos de su/s mascota/s
Descripción	El cliente puede agregar, eliminar, modificar (excepto el historial clínico) y visualizar la información de sus mascotas a través del sistema.

Caso de Uso 2:

Código	CU2
Nombre	Dar del alta un cliente (cliente y/o mascota)
Actores	Usuario profesional (veterinario)
Funcion	Registrar los datos de un cliente



Descripción	El veterinario puede registrar un nuevo cliente y/o su mascota, introduciendo los datos correspondientes (Cliente: nombre, ID, DNI; Mascota: nombre, raza, edad, historial médico).
-------------	---

Caso de Uso 3:

Código	CU3
Nombre	Gestionar productos
Actores	Usuario profesional (veterinario)
Funcion	Agregar, eliminar, modificar y visualizar productos en el inventario.
Descripción	El usuario profesional/veterinario puede gestionar los productos disponibles en el inventario, incluyendo agregar nuevos productos, eliminarlos, modificar sus detalles y visualizar la información de los productos.

Caso de Uso 4:

Código	CU4
Nombre	Interactuar con el sistema a través de menús
Actores	Usuarios del sistema (clientes y veterinarios)
Funcion	Navegar y utilizar las funcionalidades del sistema mediante menús interactivos.
Descripción	Los usuarios pueden interactuar con el sistema utilizando menús que les permiten acceder a diferentes funcionalidades, como la gestión de clientes, veterinarios y productos.

2. Diseño

El diseño consta, por la definición de la arquitectura a utilizar, junto con la definición de la base de datos.



2.1 Descripción del Sistema.

El sistema implementado utiliza un patrón de diseño basado en entidades, modelos y gestores. A continuación, se describe cómo se relacionan estos componentes:

- **Entidades y Modelos:**

- Un cliente puede tener múltiples mascotas mediante la referencia a MascotaModelo.
- Un cliente utiliza un inventario.
- Un inventario contiene múltiples productos.

- **Gestores:**

- ClienteManager gestiona a los clientes.
- VeterinarioManager gestiona a los veterinarios.
- MascotaManager gestiona a las mascotas.
- ProductoManager gestiona los productos.
- InventarioManager gestiona los inventarios.

- **Herencia y Implementación:**

Todos los gestores (ClienteManager, VeterinarioManager, MascotaManager, ProductoManager, InventarioManager) heredan funcionalidad de ArrayIdManager, que es la clase base más general.

Además, estos gestores implementan la interfaz ABMinterface, comprometiéndose a proporcionar métodos para operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

- **Interacción con la Vista:**

La clase MenuAdministrador, junto con Menu, actúa como vista. Interactúa con el usuario, presenta la información organizada y permite la entrada de datos.

MenuAdministrador utiliza los gestores para obtener datos del modelo y mostrarlos al usuario, así como para recibir instrucciones sobre acciones a realizar.

2.2 Arquitectura.

En el ámbito de las aplicaciones una de las arquitecturas de software más extendidas es el modelo MVC (Modelo Vista Controlador). Dicho esto, fue el que se trabajó en el plan de estudio de Programación Orientada a Objetos. Es por esto que ha sido adoptado en este sistema.

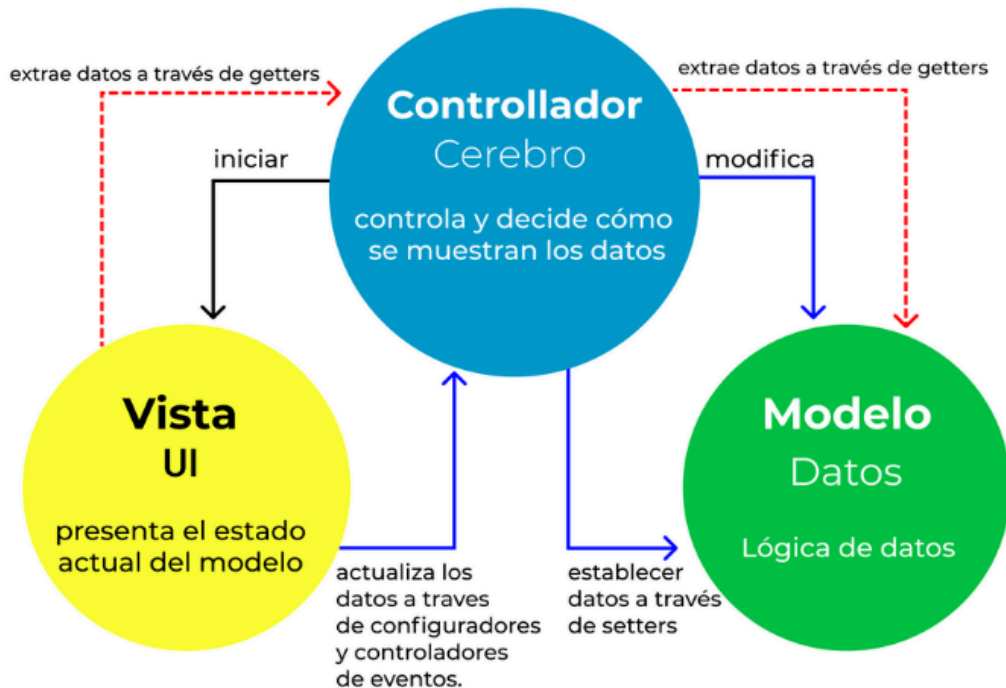
Esta estructura está formada por tres componentes:

- **Modelo:** contiene la representación de los datos de la aplicación y la lógica de negocio relacionada.
- **Vista:** Proporciona la interfaz de usuario para el uso de la aplicación.



- **Controlador:** Es la conexión, intermediario, entre el modelo y la vista. Gestionando las interacciones del usuario y actualizando el modelo según sea necesario. Contiene la lógica de negocio de la aplicación.

Patrones de Arquitectura MVC



Gráfica de patrones de arquitectura MVC

2.3 Análisis de la arquitectura

Teniendo en cuenta la implementación de la arquitectura MVC, es necesario el desarrollar la explicación de la presentación de cada una de las capas:

- **El modelo:** Representa los datos de la aplicación y la lógica de negocio asociada. En este sistema, las clases Cliente, Mascota, Veterinario, Producto, e Inventario son las entidades centrales. Sin embargo, son los modelos (ClienteModelo, MascotaModelo, VeterinarioModelo, ProductoModelo, InventarioModelo) los que interactúan directamente con la base de datos a través de Conexion, encapsulando la lógica de acceso a datos.
- **La vista:** La clase MenuAdmin, en conjunto con la clase Menu, cumple el rol de vista. Es responsable de interactuar con el usuario, presentando la información de manera organizada y permitiendo la entrada de datos. MenuAdmin utiliza los Managers para obtener los datos del modelo y mostrarlos al usuario, así como para recibir las instrucciones del usuario sobre las acciones a realizar.



- El controlador: Los Managers, como ClienteManager, MascotaManager, VeterinarioManager, y ProductoManager, funcionan como controladores. Gestionan las interacciones entre el modelo y la vista, implementando la lógica de negocio del sistema. Reciben las solicitudes de la vista, acceden al modelo para realizar las operaciones necesarias (como crear, leer, actualizar, y eliminar datos), y actualiza la vista con los resultados.

El lenguaje de programación utilizado para el desarrollo del proyecto es PHP, debido al conocimiento y experiencia del desarrollador en este lenguaje, adquirido durante el plan de estudios de la materia.

2.3.1 Acceso a datos.

En este proyecto, se ha optado por utilizar MySQL 9.2 como sistema de gestión de bases de datos, administrado a través de la herramienta DBeaver. DBeaver ofrece una interfaz gráfica intuitiva para gestionar y administrar bases de datos, lo que facilita la creación, modificación y consulta de estructuras de datos.

2.4 Interfaz de comunicación.

Se utiliza PDO (PHP Data Objects) como interfaz de comunicación con la base de datos MySQL. PDO proporciona una interfaz unificada para acceder a diferentes sistemas de gestión de bases de datos, mejorando la seguridad y facilitando el manejo de errores.

Lo utilizamos creando una instancia con la cadena de conexión adecuada, para métodos, tal como prepare() y execute(), ejecutando consultas preparadas, mejorando la seguridad al evitar la inyección de SQL, y para el manejo de errores mediante excepciones, permitiendo capturar y tratar errores.

2.5 Base de datos.

Para la realización de este proyecto se han definido un conjunto de tablas, las cuales almacenarán los datos necesarios para solventar todos los casos de uso planteados.

Para la definición del Cliente, se generará la tabla Cliente que contiene información de identificación del usuario, como su nombre, DNI, ID, así como dos arrays importantes: referenciaAnimal e inventario. El array referenciaAnimal almacena las mascotas asociadas a cada cliente, mientras que el array inventario contiene los productos que el cliente puede tener. Otra de las entidades definidas es la tabla Mascota que almacena detalles sobre las mascotas, como su nombre, edad, raza, ID, historial médico y el ID del cliente al que pertenecen. Esta tabla está



estrechamente relacionada con la tabla Cliente, ya que cada mascota está asociada a un cliente específico.

Existe una relación de uno a muchos(one-to-many) entre Cliente y Macota. Esto significa que un cliente puede tener varias mascotas, pero cada mascota solo pertenece a un cliente. Por ejemplo, si Juan es un cliente, puede tener varias mascotas como un perro y un gato, pero cada una de esas mascotas solo está asociada a Juan.

Por otro lado, existen otras entidades, tales como Inventario, Producto, Veterinario, etc.

La gestión de estas entidades es llevada a cabo por los Managers, como ClienteManager, VeterinarioManager, MascotaManager, y ProductoManager. Estos Managers gestionan sus respectivas entidades y utilizan la conexión a la base de datos a través de los Modelos (ClienteModelo, MascotaModelo, etc.). Los Managers, a su vez, aprovechan la funcionalidad proporcionada por la clase base ArrayIdManager para la gestión de colecciones de objetos y la interfaz ABMInterface para asegurar la implementación de las operaciones básicas de gestión (CRUD).

3. Implementación.

Para cumplimentar los requisitos definidos para este proyecto, la implementación se ha basado en los estándares del MVC. Los estándares de este modelo definen una serie de buenas prácticas de estructuración y codificación de clases, para facilitar el mantenimiento, la escalabilidad, y la reutilización del código.

Se divide el sistema en varias capas o niveles, donde cada capa tiene una responsabilidad específica y se comunica con las capas adyacentes de manera bien definida. Cada capa actúa como un componente independiente que realiza una parte del procesamiento total.

Usuario → MenuAdmin (Capa de Presentación)



Managers (Capa de Negocio)

↓ (mediante los modelos)

Conexión → (Capa de Datos)

Foto 1. diagrama que describe una arquitectura de sistema en capas.



3.1 Capa de acceso a base de datos.

Esta capa maneja la recuperación y el almacenamiento de la información en la base de datos. Cuando la capa de lógica de negocios necesita acceder a datos específicos, se comunica con esta capa. Esta es responsable de garantizar que los datos se almacenen de manera segura y se recuperen eficientemente cuando sea necesario.

En este proyecto, los componentes principales de esta capa son:

- La clase **Conexion**: Gestiona la comunicación con la base de datos mediante la ejecución de consultas SQL.
- Los **Modelos** (**ClienteModelo**, **MascotaModelo**, **ProductoModelo**, **VeterinarioModelo**): Proporcionan una interfaz para interactuar con las tablas de la base de datos, realizando operaciones CRUD sobre las entidades. Además, se incluye **InventarioModelo** para gestionar la persistencia de datos relacionados con el inventario de los clientes.
- Los **Managers** utilizan los **Modelos** y la clase **Conexion** para interactuar con la base de datos. Es importante destacar que la clase **Conexion** implementa el patrón **Singleton**, garantizando que solo exista una instancia de conexión a la base de datos. Esto optimiza el uso de recursos y centraliza la gestión de la conexión.

```
class Conexion
{
// Prepara una sentencia SQL para su ejecución
static function prepare($sql)
{
return self::getConexion()->prepare($sql);
}
}

class MascotaModelo {
// Guarda la información de una mascota en la base de datos
public function guardar(Mascota $mascota)
{
try {
// Consulta SQL para insertar una nueva mascota
$sql = "INSERT INTO Mascota (nombre, edad, raza, historialMedico, clienteld)
VALUES (:nombre, :edad, :raza, :historialMedico, :clienteld)";
$stmt = Conexion::prepare($sql);
// ...
}
}
```

3.2 Capa de servicios.

Es recomendable seguir un diseño de desacople para una de las capas de la aplicación, con el fin de facilitar cambios y actualizaciones futuras. Es por ello, que a la hora de consumir datos y métodos ofrecidos por la capa de acceso a base de datos, es recomendable implementar una capa de servicios.



Para desarrollar una capa de servicios es imprescindible comenzar implementando una interfaz. Esta se puede entender como la definición de métodos que estarán disponibles, definiendo a su vez parámetros y valores devueltos.

La interfaz es simplemente la definición del “esqueleto” de los métodos a utilizar, definiendo: nombre, valores y parámetros de entrada.

Una vez definida, es necesario implementarla, para ello se crea la implementación que debe incluir todos los métodos definidos en esta.

Es este proyecto, los componentes para la capa de servicio son:

- La interfaz `ABMInterface`, que define los métodos obligatorios (alta, baja, modificar, levantar), asegurando consistencia en todos los managers.
- Los managers: (`ClienteManager`, `MascotaManager`, `VeterinarioManager`, `ProductoManager`): Implementan las operaciones CRUD definidas por `ABMInterface` y las validaciones específicas de cada entidad.

La clase base `ArrayIdManager`: Proporciona funcionalidad genérica para la gestión de colecciones de objetos, así como la gestión de IDs.

- Adicionalmente, `InventarioManager` también forma parte de esta capa, gestionando la relación de los clientes con sus inventarios de productos. Implementa `ABMInterface` para asegurar la consistencia en la gestión del inventario y extiende `ArrayIdManager` para la gestión de colecciones de objetos de inventario.

- Ejemplo en código:

```
interface ABMInterface {  
    //Da de alta un elemento en la base de datos y en la estructura  
    public function alta();  
}
```

```
abstract class ArrayIdManager {  
    // Agregar un objeto nuevo en la posición id del elemento  
    public function agregar($elemento)  
    {  
        $id = $elemento->getId();  
        $this->arreglo[$id] = $elemento;  
    }  
}
```

```
class ClienteModelo {  
    // Guarda Cliente  
    public function guardar(Cliente $cliente) {  
        try {  
            $sql = "INSERT INTO Cliente (nombre, dni)  
                VALUES (:nombre, :dni)";  
  
            $stmt = Conexion::prepare($sql);  
            $nombre = $cliente->getNombre();  
            // ...  
        }  
    }  
}
```



```
}  
  
class ClienteManager extends arrayIdManager implements ABMInterface {  
    // Crea un Cliente  
    public function alta()  
    {  
    // ...  
        $modelo = new ClienteModelo();  
    // ...  
        $cliente = new Cliente($nombre, $dni);  
        if ($modelo->guardar($cliente)) {  
            $this->agregar($cliente);  
        }  
    // ...  
    }  
}
```

3.3 Capa de presentación.

En el año 2024, cuando curse la materia, utilizamos como la cara visible del sistema, donde los usuarios podrán interactuar directamente, la consola de visual studio code o mismo la de cualquier navegador de internet en el año 2023 en mi primer año de carrera, en programación I.

Esta capa se centra en la presentación y la interacción, transmitiendo los datos ingresados por el usuario a la capa de lógica de negocios para su procesamiento.

Sus características claves son que está desacoplada de la lógica de negocio, solo se comunica con la capa de negocio (en este caso, los managers), y la herencia, ya que MenuAdministrador hereda de Menu, reutilizando funcionalidades base y extendiéndose para roles administrativos.

Los componentes de esta capa en el sistema son:

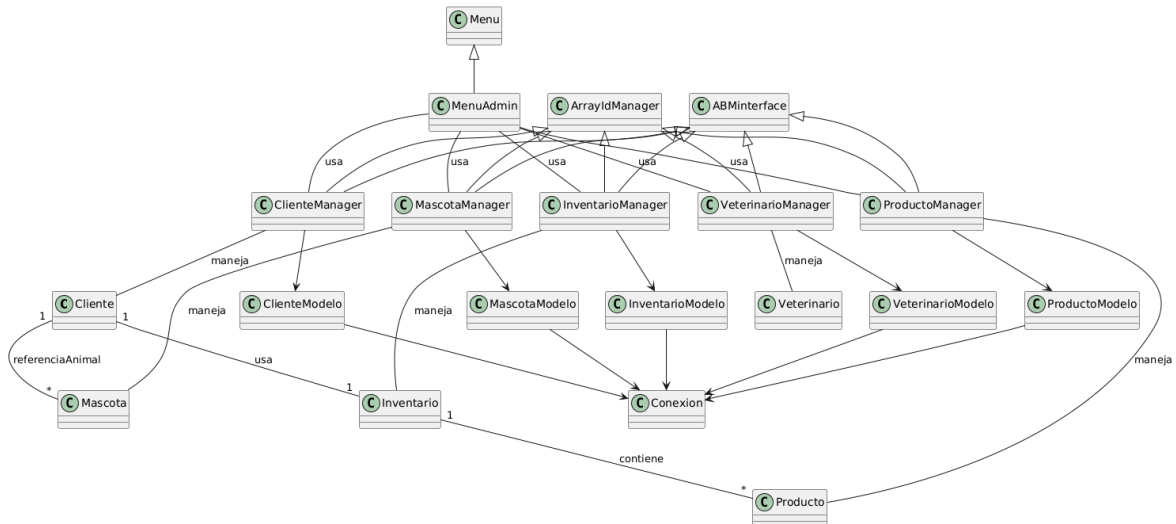
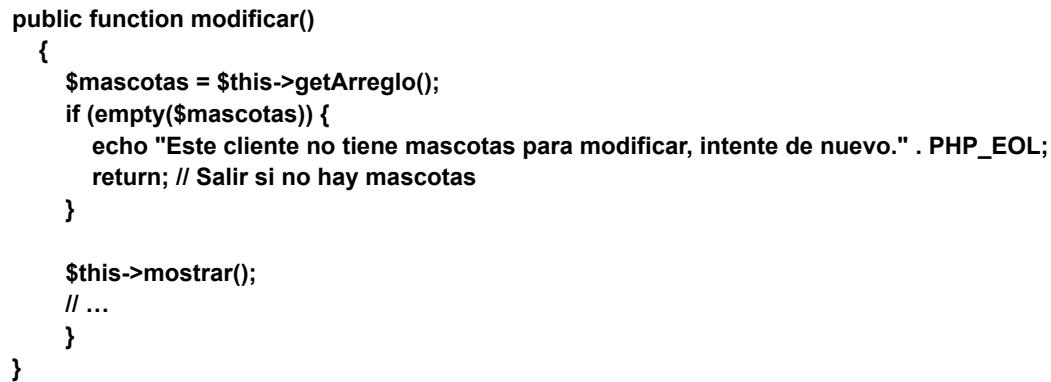
- Menu y MenuAdmin, que actúan como la interfaz de usuario (UI), mostrando opciones y recibiendo las entradas del usuario.

Ejemplo. Cuando el Usuario selecciona “Modificar Mascota” en el MenuAdmin, este componente envía la solicitud al MascotaManager (capa de negocio).

- Código de Ejemplo:

```
class MenuAdmin extends Menu {  
    // Submenú para gestionar mascotas  
    protected function menuGestionMascotas(MascotaManager $mascotaManager)  
    {  
    // ...  
        $opciones[3][0] = 3;  
        $opciones[3][1] = "Modificar mascota";  
        $opciones[3][2] = array($mascotaManager, "modificar");  
  
        self::menu($titulo, $opciones);  
    }  
}
```

```
class MascotaManager extends arrayIdManager {
```



Como resultado de la implementación del proyecto, se ha obtenido un sistema de gestión de una Veterinaria, en este caso llamada Patitas, que implementa todos los casos de uso definidos. Ofreciendo las funcionalidades requeridas.

Se puede ver la pantalla de acceso al sistema, permitiendo ingresar como veterinario o cliente.

```
-----  
Bienvenido a la Veterinaria 'Patitas'  
-----  
0 - Salir del sistema  
1 - Ingresar como Cliente  
2 - Ingresar como Veterinario  
  
Elija una opción: █
```

12 Romeo Gago Lucia 2025 Final Libre



En cualquiera de los dos casos se le va a pedir que ingrese su id, que ya existe en la base de datos.

4.2 Menú principal del usuario Cliente.

Una vez que el cliente con su perfil entra al sistema, se le mostrará el menú principal del perfil Cliente.

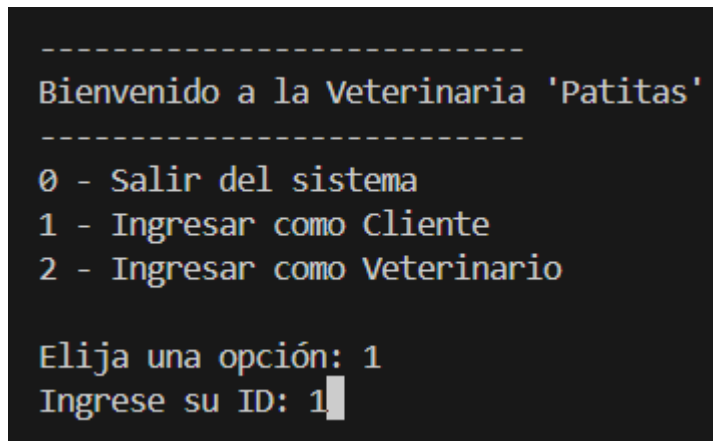


Foto 3. ingreso de clientes.

Esta ventana permite a un usuario Cliente acceder a las distintas funcionalidades de las que se dispone en el sistema: administrar a sus mascotas, visualizarlas, comprar productos o visualizar su inventario.

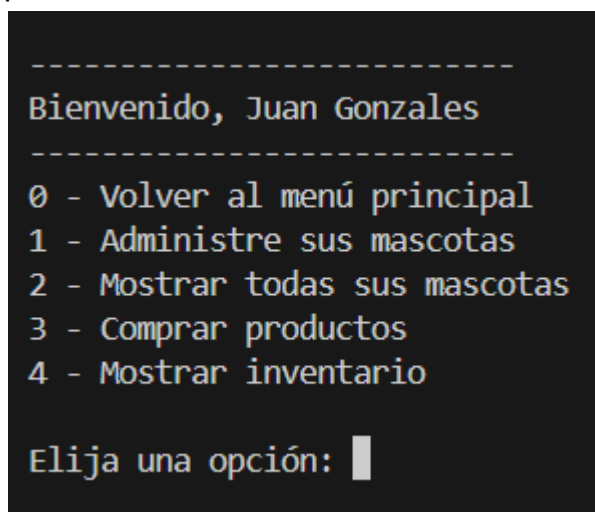


Foto 4. menú de entrada del cliente.

Cada una de las funcionalidades descritas será accesible mediante el ingreso del número del enunciado.

4.2.1 Menú de administración de mascotas.



Tras pulsar el número “1”, se ingresa al menú de gestión de mascotas para el cliente, el cual está compuesto por opciones para poder agregar, eliminar y modificar a su/su mascota/s. Al igual que la opción de volver al menú principal.

```
-----  
Gestión de Mascotas  
-----  
0 - Volver al menú del cliente  
1 - Agregar mascota  
2 - Eliminar mascota  
3 - Modificar mascota  
  
Elija una opción: █
```

Foto 5. Menú del cliente de gestión de mascotas.

Cada función que quiera realizar, “abrirá” un nuevo espacio donde deberá ingresar los datos que se le pidan.

```
-----  
Gestión de Mascotas  
-----  
0 - Volver al menú del cliente  
1 - Agregar mascota  
2 - Eliminar mascota  
3 - Modificar mascota  
  
Elija una opción: 1  
Ingrese el nombre de la mascota: Ejemplo  
  
Ingrese la edad de la mascota: 0  
  
Ingrese la raza de la mascota: Ejemplo  
  
Ingrese el historial médico de la mascota: ejemplo █
```

Foto 6. función agregar mascota

```
Lista de mascotas del cliente para eliminar: Juan Gonzales  
-----  
Nombre: Piponsio, Edad: 8, Raza: Pitbull, Id: 1, Historial Médico: Esta enfermito  
Nombre: Ola, Edad: 3, Raza: Perro callejero, Id: 2, Historial Médico: Vacunas al dia  
Presiona Enter para continuar...  
Ingrese el ID de la mascota a eliminar: █
```

Foto 7. función eliminar mascota



```
Lista de mascotas del cliente: Juan Gonzales
-----
Nombre: Piponsio, Edad: 8, Raza: Pitbull, Id: 1, Historial Médico: Esta enfermito
Nombre: Ola, Edad: 3, Raza: Perro callejero, Id: 2, Historial Médico: Vacunas al dia
Presiona Enter para continuar...
Ingrese ID de la mascota a modificar: 1

Está por modificar la siguiente mascota del sistema:

Nombre: Piponsio, Edad: 8, Raza: Pitbull, Id: 1, Historial Médico: Esta enfermito
¿Está seguro? S/N: S

Ingrese el nuevo nombre de la mascota (deje en blanco para no modificar):

Ingrese la nueva edad de la mascota (deje en blanco para no modificar):

Ingrese la nueva raza de la mascota (deje en blanco para no modificar):
```

Foto 8. modificar mascota

4.2.2 Visualización de las mascotas del cliente.

Tras pulsar el número “2”, el sistema mostrará las mascotas que el cliente ingresado contiene en la base de datos, con la información (nombre, raza, id, historial médico) de cada una.

```
Lista de mascotas del cliente: Juan Gonzales
-----
Nombre: Piponsio, Edad: 8, Raza: Pitbull, Id: 1, Historial Médico: Esta enfermito
Nombre: Ola, Edad: 3, Raza: Perro callejero, Id: 2, Historial Médico: Vacunas al dia
Presiona Enter para continuar...█
```

Foto 9. Visualización de todas las mascotas asociadas al cliente.

En caso de no estar asociado a ninguna mascota, mostrará un cartel explicativo.



```
-----  
Bienvenido, Roberta Martinez  
-----  
0 - Volver al menú principal  
1 - Administre sus mascotas  
2 - Mostrar todas sus mascotas  
3 - Comprar productos  
4 - Mostrar inventario  
  
Elija una opción: 2  
Este cliente no tiene mascotas.
```

Foto 10. El cliente no tiene mascotas asociadas.

4.2.3 Menú compra de productos.

Tras pulsar el número “3” el cliente accede a un menú para gestionar la compra de un producto.

```
Elija una opción: 3  
  
-----  
Gestión de Productos  
-----  
0 - Volver al menú del cliente  
1 - Mostrar todos los productos disponibles  
2 - Comprar producto  
  
Elija una opción: █
```

Foto 11. menú de gestión de productos

Tendrá las opciones de visualizar productos con su correspondiente información (nombre, id, precio y stock).

```
Lista de productos disponibles:  
-----  
Id: 11, Nombre: Shampoo Pulguicida, Precio: 16140, Stock: 10  
Id: 12, Nombre: Spray eliminar de olor en seco, Precio: 28000, Stock: 10  
Id: 13, Nombre: Crema Desenredante perro, Precio: 17200, Stock: 10  
Presiona Enter para continuar...█
```

Foto 12. lista de productos disponibles para el cliente.

Y de ingresar el producto que desea comprar.



Lista de productos disponibles:

Id: 11, Nombre: Shampoo Pulguicida, Precio: 16140, Stock: 10
Id: 12, Nombre: Spray eliminar de olor en seco, Precio: 28000, Stock: 10
Id: 13, Nombre: Crema Desenredante perro, Precio: 17200, Stock: 10
Presiona Enter para continuar...

Ingrese el ID del producto que desea comprar: 11
Ingrese la cantidad que desea comprar: 10

Foto 13. menú para comprar un producto.

4.2.4 Visualización del inventario del cliente.

Tras pulsar el número “4”, el sistema mostrará el inventario del cliente ingresado en caso de contener productos asociados a su id.

Bienvenido, Juan Gonzales

0 - Volver al menú principal
1 - Administre sus mascotas
2 - Mostrar todas sus mascotas
3 - Comprar productos
4 - Mostrar inventario

Elija una opción: 4
Inventario de Juan Gonzales:
- Shampoo Pulguicida (Cantidad: 10)

Foto 14. inventario con productos.

En caso de no tener ningún producto asociado, mostrará un cartel explicativo.

Bienvenido, Roberta Martinez

0 - Volver al menú principal
1 - Administre sus mascotas
2 - Mostrar todas sus mascotas
3 - Comprar productos
4 - Mostrar inventario

Elija una opción: 4
El cliente no tiene productos en su inventario.



Foto 15. el cartel explicativo sobre la falta de productos asociados.

4.3 Menu principal del usuario Veterinario.

Una vez un usuario con el perfil de Veterinario ingrese, se le mostrará el menú principal del perfil Veterinario.

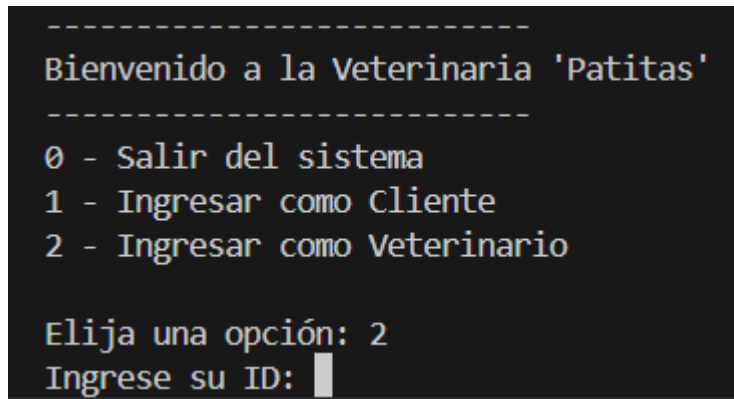


Foto 16. ingreso de veterinario.

Este menú permite a un usuario con el perfil de veterinario acceder a distintas funcionalidades de las que dispone el sistema: administrar clientes, mascotas y productos.

Cada una de estas funcionalidades descritas serán accesibles mediante el ingreso del número que ingrese de acuerdo a las opciones.

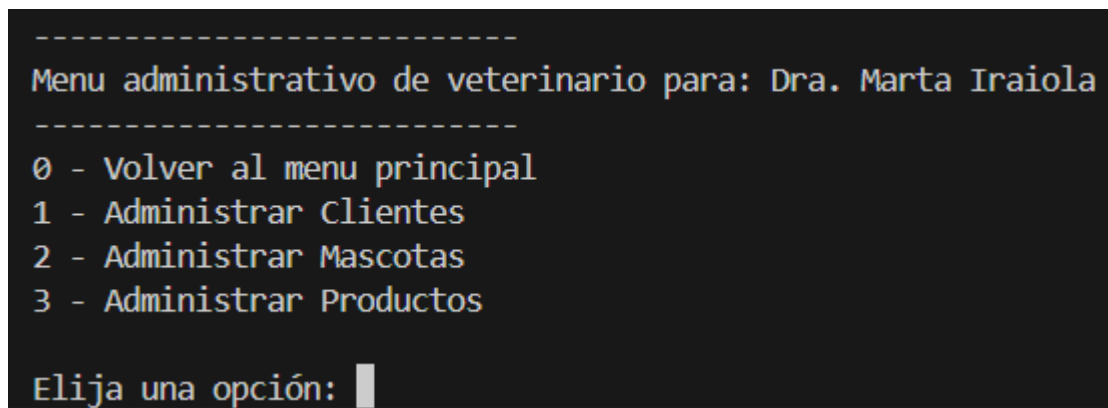


Foto 17. Menú principal del veterinario.

4.3.1 Menú veterinario para administrar clientes.

Tras pulsar “1”, se ingresa al menú administrativo de Clientes para veterinarios, donde tendrá opciones para gestionarlos, opciones como agregar, eliminar, modificar o visualizar todos los clientes. Al igual que la opción de volver al menú principal.

Cada función que quiera realizar, “abrirá” un nuevo espacio donde deberá ingresar los datos que se le pidan.



```
-----  
Menu administrativo de Clientes  
-----  
0 - Volver al menu anterior  
1 - Agregar cliente  
2 - Eliminar cliente  
3 - Modificar cliente  
4 - Mostrar todos los clientes  
  
Elija una opción: █
```

Foto 18. menú veterinario administrativo de clientes.

```
-----  
Menu administrativo de Clientes  
-----  
0 - Volver al menu anterior  
1 - Agregar cliente  
2 - Eliminar cliente  
3 - Modificar cliente  
4 - Mostrar todos los clientes  
  
Elija una opción: 1  
Ingrese el nombre y apellido del cliente: Natalia Perez  
  
Ingrese el dni del cliente: 31258741  
  
Ingrese el id del cliente: 9  
  
¿Desea agregar una mascota? (si/no): si  
  
Ingrese el nombre de la mascota: Uriel  
  
Ingrese la edad de la mascota: 8  
  
Ingrese la raza de la mascota: Doberman perro  
  
Ingrese el historial médico de la mascota: Lesion en pata trasera derecha █
```

Foto 19. agregar un nuevo cliente y su mascota asociada.



```
Lista de los clientes existentes en nuestro sistema
-----
Id: 1, Nombre: Liam Cuthill, Dni: 43263811
Mascotas del cliente Liam Cuthill:
- Nombre: Fito, Edad: 2, Raza: Perro Jack Russell Terrier, Id: 2, Historial Médico: Sano, falta vacuna antirrabica

Id: 3, Nombre: Lucia Romeo Gago, Dni: 42773590
Mascotas del cliente Lucia Romeo Gago:
- Nombre: Hipatia, Edad: 3, Raza: Gato Bombay, Id: 5, Historial Médico: Sana, falta revisar la libreta de vacunacion

Id: 4, Nombre: Martina Bandi, Dni: 44236987
Mascotas del cliente Martina Bandi:
- Nombre: Pascual, Edad: 3, Raza: Perro Salchicha, Id: 1, Historial Médico: Sano, falta revision de vacunas
- Nombre: Coco, Edad: 16, Raza: Gato blanco, Id: 12, Historial Médico: Problemas urinarios

Id: 11, Nombre: Mariana Gomez, Dni: 11222547
Mascotas del cliente Mariana Gomez:
- Nombre: Luna Alejandra, Edad: 11, Raza: Pez Koi, Id: 17, Historial Médico: Le falta la aleta izquierda
- Nombre: Osvaldo , Edad: 17, Raza: Lagarto Overo, Id: 18, Historial Médico: Se le perdio la mitad de la cola

Presiona Enter para continuar...
Ingrese el id cliente a eliminar: |
```

Foto 20. visualizar a todos los clientes y sus mascotas asociadas.

```
Lista de los clientes existentes en nuestro sistema
-----
Id: 1, Nombre: Liam Cuthill, Dni: 43263811
Mascotas del cliente Liam Cuthill:
- Nombre: Fito, Edad: 2, Raza: Perro Jack Russell Terrier, Id: 2, Historial Médico: Sano, falta vacuna antirrabica

Id: 3, Nombre: Lucia Romeo Gago, Dni: 42773590
Mascotas del cliente Lucia Romeo Gago:
- Nombre: Hipatia, Edad: 3, Raza: Gato Bombay, Id: 5, Historial Médico: Sana, falta revisar la libreta de vacunacion

Id: 4, Nombre: Martina Bandi, Dni: 44236987
Mascotas del cliente Martina Bandi:
- Nombre: Pascual, Edad: 3, Raza: Perro Salchicha, Id: 1, Historial Médico: Sano, falta revision de vacunas
- Nombre: Coco, Edad: 16, Raza: Gato blanco, Id: 12, Historial Médico: Problemas urinarios

Id: 11, Nombre: Mariana Gomez, Dni: 11222547
Mascotas del cliente Mariana Gomez:
- Nombre: Luna Alejandra, Edad: 11, Raza: Pez Koi, Id: 17, Historial Médico: Le falta la aleta izquierda
- Nombre: Osvaldo , Edad: 17, Raza: Lagarto Overo, Id: 18, Historial Médico: Se le perdio la mitad de la cola

Presiona Enter para continuar...
Ingrese Id de cliente a modificar: |
```

Foto 21. modificar un cliente.

```
Lista de los clientes existentes en nuestro sistema
-----
Id: 1, Nombre: Liam Cuthill, Dni: 43263811
Mascotas del cliente Liam Cuthill:
- Nombre: Fito, Edad: 2, Raza: Perro Jack Russell Terrier, Id: 2, Historial Médico: Sano, falta vacuna antirrabica

Id: 3, Nombre: Lucia Romeo Gago, Dni: 42773590
Mascotas del cliente Lucia Romeo Gago:
- Nombre: Hipatia, Edad: 3, Raza: Gato Bombay, Id: 5, Historial Médico: Sana, falta revisar la libreta de vacunacion

Id: 4, Nombre: Martina Bandi, Dni: 44236987
Mascotas del cliente Martina Bandi:
- Nombre: Pascual, Edad: 3, Raza: Perro Salchicha, Id: 1, Historial Médico: Sano, falta revision de vacunas
- Nombre: Coco, Edad: 16, Raza: Gato blanco, Id: 12, Historial Médico: Problemas urinarios

Id: 11, Nombre: Mariana Gomez, Dni: 11222547
Mascotas del cliente Mariana Gomez:
- Nombre: Luna Alejandra, Edad: 11, Raza: Pez Koi, Id: 17, Historial Médico: Le falta la aleta izquierda
- Nombre: Osvaldo , Edad: 17, Raza: Lagarto Overo, Id: 18, Historial Médico: Se le perdio la mitad de la cola

Presiona Enter para continuar...|
```



Foto 22. visualizar a todos los clientes y sus mascotas.

4.3.2 Menú veterinario de administración de mascotas de un cliente

Tras pulsar el número “2”, se mostrará en pantalla los clientes y sus mascotas asociadas, allí se deberá ingresar el id del cliente al que se desee gestionar sus mascotas.

```
Lista de los clientes existentes en nuestro sistema
-----
Id: 1, Nombre: Liam Cuthill, Dni: 43263811
Mascotas del cliente Liam Cuthill:
- Nombre: Fito, Edad: 2, Raza: Perro Jack Russell Terrier, Id: 2, Historial Médico: Sano, falta vacuna antirrabica

Id: 3, Nombre: Lucia Romeo Gago, Dni: 42773590
Mascotas del cliente Lucia Romeo Gago:
- Nombre: Hipatia, Edad: 3, Raza: Gato Bombay, Id: 5, Historial Médico: Sana, falta revisar la libreta de vacunacion

Id: 4, Nombre: Martina Bandi, Dni: 44236987
Mascotas del cliente Martina Bandi:
- Nombre: Pascual, Edad: 3, Raza: Perro Salchicha, Id: 1, Historial Médico: Sano, falta revision de vacunas
- Nombre: Coco, Edad: 16, Raza: Gato blanco, Id: 12, Historial Médico: Problemas urinarios

Id: 6, Nombre: Lola Lopez Rivero, Dni: 29357412
Mascotas del cliente Lola Lopez Rivero:
- Nombre: Limon Helado, Edad: 9, Raza: Gato Naranja, Id: 7, Historial Médico: Proximo a renovacion de vacunas anuales
- Nombre: Paloma, Edad: 5, Raza: Pez, Id: 11, Historial Médico: Enfermo

Id: 11, Nombre: Mariana Gomez, Dni: 11222547
Mascotas del cliente Mariana Gomez:
- Nombre: Luna Alejandra, Edad: 11, Raza: Pez Koi, Id: 17, Historial Médico: Le falta la aleta izquierda
- Nombre: Osvaldo , Edad: 17, Raza: Lagarto Overo, Id: 18, Historial Médico: Se le perdio la mitad de la cola

Presiona Enter para continuar...
Ingrese el ID del cliente para gestionar sus mascotas: █
```

Foto 23. lista de clientes para gestionar sus mascotas.

Al ingresar, se mostrará un menú administrativo de las mascotas de ese cliente en particular, siendo este compuesto por opciones para poder agregar, eliminar, modificar(teniendo en cuenta a diferencia del cliente, que el veterinario puede modificar el historial médico de la mascota) y mostrar las mascotas.

```
-----
Menu veterinario administrativo de Mascotas de: Graciela Ugarte
-----
0 - Volver al menu anterior
1 - Agregar mascota
2 - Eliminar mascota
3 - Modificar mascota
4 - Mostrar todas las mascotas

Elija una opción: █
```

Foto 24. menú administrativo para mascotas de un cliente específico.

```
Lista de mascotas del cliente: Juan Gonzales
-----
Nombre: Piponsio, Edad: 8, Raza: Pitbull, Id: 1, Historial Médico: Padece de asma
Nombre: Ola, Edad: 3, Raza: Perro callejero, Id: 2, Historial Médico: Vacunas al dia
Presiona Enter para continuar...█
```

Foto 25. lista de las mascotas de un cliente específico



Lista de mascotas del cliente: Juan Gonzales

Nombre: Piponsio, Edad: 8, Raza: Pitbull, Id: 1, Historial Médico: Esta enfermito
Nombre: Ola, Edad: 3, Raza: Perro callejero, Id: 2, Historial Médico: Vacunas al dia
Presiona Enter para continuar...
Ingrese ID de la mascota a modificar: 1

Está por modificar la siguiente mascota del sistema:

Nombre: Piponsio, Edad: 8, Raza: Pitbull, Id: 1, Historial Médico: Esta enfermito

¿Está seguro? S/N: S

Ingrese el nuevo nombre de la mascota (deje en blanco para no modificar):

Ingrese la nueva edad de la mascota (deje en blanco para no modificar):

Ingrese la nueva raza de la mascota (deje en blanco para no modificar):

Ingrese el nuevo historial médico de la mascota (deje en blanco para no modificar): Padece de asma

Foto 26. modificar una mascota.

Menu veterinario administrativo de Mascotas de: Natalia Perez

- 0 - Volver al menu anterior
1 - Agregar mascota
2 - Eliminar mascota
3 - Modificar mascota
4 - Mostrar todas las mascotas

Elija una opción: 1

Ingrese el nombre de la mascota: Amadeo

Ingrese la edad de la mascota: 7

Ingrese la raza de la mascota: Jack Russell perro

Ingrese el historial médico de la mascota: Sano

La mascota se ha creado con éxito.

Foto 27. agregar mascota a un cliente.



```
Lista de mascotas del cliente para eliminar: Martina Bandi
-----
Nombre: Pascual, Edad: 3, Raza: Perro Salchicha, Id: 3, Historial Médico: Sano
Nombre: Coco, Edad: 20, Raza: Gato Blanco, Id: 5, Historial Médico: En tratamiento por problemas urinarios
Presiona Enter para continuar...
Ingrese el ID de la mascota a eliminar: █
```

Foto 28. eliminar mascota a un cliente.

4.3.3 Menú veterinario de administración de productos.

Tras pulsar el número “3”, se ingresa al menú de gestión veterinario de productos, el cual está compuesto por opciones para poder agregar, eliminar, modificar productos y visualizar. Al igual que la opción de volver al menú principal.

Cada función que quiera realizar, “abrirá” un nuevo espacio donde deberá ingresar los datos que se le pidan.

```
-----
Menu administrativo de Productos
-----
0 - Volver al menu anterior
1 - Agregar producto
2 - Eliminar producto
3 - Actualizar producto
4 - Mostrar todos los productos

Elija una opción: █
```

Foto 29. menú veterinario administrativo de productos.



```
-----  
Menu administrativo de Productos  
-----
```

- ```
0 - Volver al menu anterior
1 - Agregar producto
2 - Eliminar producto
3 - Actualizar producto
4 - Mostrar todos los productos
```

```
Elija una opción: 1
```

```
Ingrese el id del producto: 16
```

```
Ingrese el nombre del producto: Cepillo de vapor para gatos
```

```
Ingrese el precio del producto: 4100
```

```
Ingrese el stock del producto: 10
```

```
Se ha guardado exitosamente el producto.
```

Foto 30. agregar un producto.

```
Lista de productos disponibles:

```

```
Id: 11, Nombre: Shampoo Pulguicida, Precio: 16140, Stock: 0
```

```
Id: 12, Nombre: Spray eliminar de olor en seco, Precio: 28000, Stock: 10
```

```
Id: 13, Nombre: Crema Desenredante perro, Precio: 17200, Stock: 10
```

```
Id: 16, Nombre: Cepillo de vapor para gatos, Precio: 4100, Stock: 10
```

```
Presiona Enter para continuar...
```

```
Ingrese el ID del producto a eliminar: █
```

Foto 31. lista de productos





Lista de productos disponibles:

-----  
Id: 11, Nombre: Shampoo Pulguicida, Precio: 16140, Stock: 0  
Id: 12, Nombre: Spray eliminar de olor en seco, Precio: 28000, Stock: 10  
Id: 13, Nombre: Crema Desenredante perro, Precio: 17200, Stock: 10  
Id: 16, Nombre: Cepillo de vapor para gatos, Precio: 4100, Stock: 10  
Presiona Enter para continuar...  
Ingrese el ID del producto a eliminar: 11

Está por eliminar el siguiente producto:

Id: 11, Nombre: Shampoo Pulguicida, Precio: 16140, Stock: 0  
¿Está seguro que desea eliminar este producto? (S/N): S

Foto 32. modificar un producto.

Lista de productos disponibles:

-----  
Id: 12, Nombre: Spray eliminar de olor en seco, Precio: 28000, Stock: 10  
Id: 13, Nombre: Crema Desenredante perro, Precio: 17200, Stock: 10  
Id: 16, Nombre: Cepillo de vapor para gatos, Precio: 4100, Stock: 10  
Presiona Enter para continuar...  
Ingrese el ID del producto a modificar: 12

Está por modificar el siguiente producto:

Id: 12, Nombre: Spray eliminar de olor en seco, Precio: 28000, Stock: 10  
¿Está seguro que desea modificar este producto? (S/N): S

A continuación ingrese los nuevos datos, ENTER para dejarlos sin modificar  
Ingrese el nuevo nombre del producto: Spray eliminador de olor en seco

Ingrese el nuevo precio del producto:

Ingrese el nuevo stock del producto:

Foto 33. modificar un producto.

#### 4.4 Mensaje de despedida del sistema de gestión de la veterinaria "Patitas".

Tras pulsar el número "0", se imprimirá un mensaje y se saldrá del sistema.

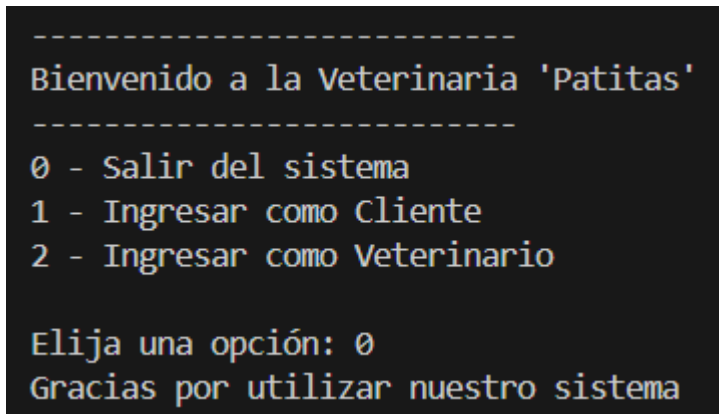


Foto 34. despedida del sistema de gestión de la veterinaria "Patitas"

## 5. Conclusiones

Tal como se expuso en el objetivo principal del proyecto, el propósito era desarrollar un sistema de gestión para una veterinaria que permitiera centralizar y facilitar las tareas relacionadas con la administración de clientes, mascotas y productos. Este objetivo se ha cumplido satisfactoriamente, logrando una solución funcional y eficiente.

### Logros del Proyecto

1. Centralización de Datos: Se logró centralizar la información de las mascotas junto con la de sus dueños (clientes), permitiendo una gestión más organizada y accesible.
2. Gestión Integral: El sistema permite administrar clientes, mascotas y productos de manera efectiva, cumpliendo con los requisitos planteados.
3. Interacción Cliente-Producto: Se implementó una funcionalidad que facilita la interacción entre los clientes y los productos ofrecidos por la veterinaria, mejorando la experiencia del usuario.

### Análisis de Resultados

Al analizar los resultados expuestos en el apartado 4, se observa que el sistema cumple con los requisitos definidos. Por un lado, el sistema es capaz de gestionar los datos de las mascotas en términos de identificación, salud y asociación con sus dueños. Esto proporciona un registro claro y accesible tanto para veterinarios como para clientes.

Por otro lado, el sistema satisface las necesidades del veterinario al permitir la visualización y asociación de clientes con sus mascotas, así como la gestión eficiente de productos (actualización, eliminación y creación). Estas funcionalidades aseguran que el sistema sea útil tanto para la administración interna como para la interacción con los clientes.



## **Lecciones Aprendidas**

Durante la implementación del proyecto se identificaron malas prácticas en el código utilizado. Esto destaca la importancia de dedicar tiempo suficiente al diseño del sistema y al estudio teórico previo a la implementación. Algunos errores podrían haberse evitado simplemente planificando con mayor detalle desde el inicio.

La falta de planificación adecuada generó decisiones tomadas sobre la marcha durante la implementación, lo que resultó en fallos como recursos repetidos o mal aprovechados. Estos problemas impactan negativamente en la eficiencia del desarrollo y subrayan la relevancia de un enfoque estructurado desde el comienzo.

## **Posibles Mejoras Futuras**

Con base en el conocimiento adquirido durante el desarrollo del proyecto, se identificaron áreas para posibles mejoras:

1. Optimización del Código: Implementar mejores prácticas para evitar redundancias y mejorar la eficiencia general.
2. Ampliación de Funcionalidades: Incluir nuevas características.
3. Mejor Diseño Arquitectónico: Dedicar más tiempo a planificar una arquitectura escalable que facilita futuras actualizaciones y expansiones.

En conclusión, este proyecto proporcionó lecciones sobre diseño, planificación e implementación en proyectos de desarrollo.