

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN SISTEMAS DE INFORMACIÓN

TEMA:

DESARROLLO DE APLICACIÓN WEB PARA LA GESTIÓN DE PROCESOS DENTRO

DE VETERINARIAS

CASO DE ESTUDIO: CLÍNICA VETERINARIA VETERPET

AUTOR:

ANDRÉS LOZADA C.

QUITO, 05 – 2022

## DEDICATORIA

---

A mi familia, novia y amigos, quienes me han acompañado a lo largo de mi carrera y me han motivado a superarme.

## AGRADECIMIENTO

---

Agradecimiento a la comunidad de desarrolladores, quienes comparten sus conocimientos para una mejor comunidad que integra cada vez más personas.

## RESUMEN

---

El proyecto presente consiste en el desarrollo de una aplicación web que busca mejorar la organización dentro de veterinarias, así como la automatización de varios procesos que se manejan dentro de las mismas como lo es la gestión de sus clientes, gestión de historiales de las mascotas que acuden a este tipo de centros, gestión de citas para sus clientes donde se puedan asignar los empleados a cargo, etc.

Cabe aclarar, que esta aplicación no busca únicamente satisfacer las necesidades de una veterinaria específica, pues, si bien el caso de estudio es la Clínica Veterinaria Veterpet, esta sirve como base para la obtención de requerimientos para la aplicación, esta permite conocer los procesos esenciales dentro de las veterinarias, de esta manera se pueden establecer los módulos necesarios dentro de una aplicación web que será usada por este tipo de centros clínicos a nivel general.

Además, en el proyecto se usará la metodología ágil basada en el framework Scrum con el fin de realizar un desarrollo eficaz de la aplicación, consiguiendo una mayor eficiencia para la creación de nuevos módulos y la capacidad de obtener mejores resultados en la variable del tiempo.

# ÍNDICE

---

## Contents

ÍNDICE DE GRÁFICOS .....	IV
ÍNDICE DE TABLAS .....	VI
CAPÍTULO I: INTRODUCCIÓN.....	1
1. MARCO DE REFERENCIA .....	1
1.1. JUSTIFICACIÓN.....	1
1.2. Planteamiento del problema .....	2
1.3. Objetivo General .....	3
1.4. Objetivos Específicos .....	3
1.5. Antecedentes .....	3
1.5.1. Actividades Clínica Veterinaria Veterpet.....	4
1.5.2. Situación actual de la Clínica Veterinaria Veterpet .....	5
1.6. Alcance .....	5
CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA .....	6
2. Marco Teórico .....	6
2.1. Metodologías de desarrollo de software.....	6
2.1.1 Metodologías Tradicionales.....	6
2.1.1.1 Modelo en Cascada.....	6

2.1.1.2	Modelo en V .....	7
2.1.2	Metodologías Ágiles .....	8
2.1.2.1	SCRUM .....	8
2.2.	Lenguajes de programación.....	10
2.3.	Frameworks de desarrollo web .....	11
2.4.	Sistema Gestor de Contenidos (CMS) .....	13
2.5.	Bases de Datos .....	13
CAPÍTULO III: METODOLOGÍA .....		15
3.	Análisis, diseño de la aplicación web basado en Scrum .....	15
3.1.	Arquitectura de Software .....	15
3.2.	Modelo Entidad Relación .....	16
3.3.	Resumen de Historias de Usuario.....	17
3.4.	Casos de uso de la aplicación .....	19
CAPÍTULO IV: DESARROLLO DE LA APLICACIÓN .....		42
4.1.	Estándares en el desarrollo de la aplicación .....	42
4.1.1.	Estándares de codificación.....	42
4.1.2.	Organización de componentes, directivos y módulos .....	43
4.2.	Sprints.....	43
4.2.1.	Sprint 1.....	43
4.2.2.	Sprint 2.....	45
4.2.3.	Sprint 3.....	46

4.2.4. Sprint 4.....	47
4.3. Muestras de la Aplicación.....	48
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES .....	52
5.1. Conclusiones.....	52
5.2. Recomendaciones.....	52
ANEXOS .....	54
Acta de Aceptación del Software .....	54
BIBLIOGRFÍA.....	55

## ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS

### ÍNDICE DE GRÁFICOS

Gráfico 1. Fases del Modelo en Cascada. Autor: Andrés Lozada, 2022 .....	7
Gráfico 2. Rodriguez, E. (2017). Modelo en V. Artículos Virtuales. <a href="https://articulosvirtuales.com/articles/educacion/modelos-de-ciclo-de-vida-de-desarrollo-de-software">https://articulosvirtuales.com/articles/educacion/modelos-de-ciclo-de-vida-de-desarrollo-de-software</a> .....	8
Gráfico 3. Sydle. (2022). Framework Scrum. Sydle. <a href="https://www.sydle.com/es/blog/pmbok-61e80383f41fbf069eb3ef2b/">https://www.sydle.com/es/blog/pmbok-61e80383f41fbf069eb3ef2b/</a> .....	10
Gráfico 4. Stack Overflow. (2022). Most Popular Technologies. Stack Overflow. <a href="https://survey.stackoverflow.co/2022/#most-loved-dreaded-and-wanted-webframe-love-dread">https://survey.stackoverflow.co/2022/#most-loved-dreaded-and-wanted-webframe-love-dread</a>	12
Gráfico 5. Arquitectura de Software. Autor: Andrés Lozada, 2022 .....	15
Gráfico 6. Modelo Entidad Relación. Autor: Andrés Lozada, 2022.....	17
Gráfico 7. Caso de uso de Administrador. Autor: Andrés Lozada, 2022 .....	19
Gráfico 8. Caso de uso Empleado Autor: Andrés Lozada, 2022 .....	20
Gráfico 9. Caso de uso: Administración de veterinaria. Autor: Andrés Lozada, 2022.....	21
Gráfico 10. Caso de uso: Administración de Empleados. Autor: Andrés Lozada, 2022 .....	22
Gráfico 11. Caso de uso: Ingreso al sistema. Autor: Andrés Lozada, 2022 .....	25
Gráfico 12. Caso de uso: Administración de Clientes. Autor: Andrés Lozada, 2022 .....	26
Gráfico 13. Caso de uso: Administración de Mascotas. Autor: Andrés Lozada, 2022.....	28
Gráfico 14. Caso de uso: Administración de Historiales de Mascotas. Autor: Andrés Lozada, 2022.....	31
Gráfico 15. Caso de Uso: Administración de Citas. Autor: Andrés Lozada, 2022 .....	34
Gráfico 16. Casos de Uso para Clientes. Autor: Andrés Lozada, 2022.....	37
Gráfico 17. Caso de Uso: Inicio de Sesión. Autor: Andrés Lozada, 2022.....	38
Gráfico 18. Caso de Uso: Administración de Cliente. Autor: Andrés Lozada, 2022 .....	38

Gráfico 19. Caso de Uso: Visualización de sus citas. Autor: Andrés Lozada, 2022 .....	40
Gráfico 20. Caso de Uso: Visualización de sus mascotas. Autor: Andrés Lozada, 2022 .....	40
Gráfico 21. Caso de Uso: Visualización del historial médico de sus mascotas. Autor: Andrés Lozada, 2022 .....	41
Gráfico 22. Código de ejemplificación camelCase. Autor: Andrés Lozada, 2022 .....	42
Gráfico 23. Página de Inicio. Autor: Andrés Lozada, 2022.....	48
Gráfico 24. Página de inicio, parte inferior. Autor: Andrés Lozada, 2022 .....	48
Gráfico 25. Inicio de Sesión en la aplicación web. Autor: Andrés Lozada, 2022 .....	49
Gráfico 26. Administración de clientes en página web. Autor: Andrés Lozada, 2022 .....	49
Gráfico 27. Administración de mascotas en página web. Autor: Andrés Lozada, 2022.....	50
Gráfico 28. Administración del historial de mascota en página web. Autor: Andrés Lozada, 2022.....	51

## ÍNDICE DE TABLAS

Tabla 1. Sprint Backlog 1. Autor: Andrés Lozada, 2022.....	44
Tabla 2. Sprint Backlog 2. Autor: Andrés Lozada, 2022.....	45
Tabla 3. Sprint Backlog 3. Autor: Andrés Lozada, 2022.....	46
Tabla 4. Sprint Backlog 4. Autor: Andrés Lozada, 2022.....	47

## **CAPÍTULO I: INTRODUCCIÓN**

---

### **1. MARCO DE REFERENCIA**

El presente trabajo de titulación propone y proyecta mediante el desarrollo de una aplicación web, la automatización de procesos dentro de centros veterinarios, tomando como base a la Clínica Veterinaria Veterpet para su análisis, delimitación de requerimientos, etc. Esta veterinaria mantenía procesos de manera manual y el flujo de datos se lo tenía bajo una modalidad física, donde los historiales médicos de mascotas, así como información de sus dueños o clientes de la veterinaria se los almacenaba en carpetas dentro de estanterías; además, la modalidad para asignar citas a sus clientes era de igual forma manual, y se comunicaba a estos mediante redes sociales.

A través de esta aplicación web, se busca optimizar todos estos procesos de gestión de historiales médicos de mascotas, administración de citas, asignación de empleados a estas, entre otras; de manera que se obtengan facilidades en la gestión de información, así como eficacia y mejoras en la consistencia de los datos de las veterinarias.

#### **1.1. JUSTIFICACIÓN**

El desarrollo tecnológico ha permitido una mejora en el manejo de procesos, datos e información, lo cual ha obligado a empresas e instituciones a implementar sistemas de información de manera que se economicen tiempos, aumente eficacia y productividad, y se permita escalar y mantenerse en el mercado.

Si bien la implementación de tecnología en todo tipo de áreas es una necesidad actual, instituciones dentro del área veterinaria mantienen la gestión de sus procesos e información de manera tradicional, es decir, a base de información almacenada físicamente en carpetas y carnés

que son otorgados a los dueños de mascotas, lo cual significa vulnerabilidad de los datos, falta de control en los procesos, pérdidas económicas y de tiempo, entre otros.

Con el fin de solventar estas problemáticas en centros veterinarios, surge la oportunidad de desarrollar este proyecto, el cual busca implementar una aplicación web que permita la gestión de datos y procesos dentro de los mismos, así como una comunicación con los dueños de mascotas que acuden a estos centros. Esto implica la capacidad de gestionar clientes, mascotas con sus historiales médicos respectivos; el manejo citas para mantener organización y distribución del personal médico de manera eficaz; además de proveer a sus clientes el acceso al carné digital de sus mascotas, así como notificaciones sobre sus citas próximas.

## **1.2. Planteamiento del problema**

A nivel de centros veterinarios, la tecnología implementada no permite el crecimiento de estos, muchos menos la seguridad de la información manejada dentro de los mismos, lo cual supone un gran riesgo en varios ámbitos, ya que supone debilidades ante la competencia, posibilidad de pérdida de datos lo cual implica perjuicios en la salud de las mascotas, entre otros.

Es claro que el uso de sistemas de información en toda institución permite un mejor desenvolvimiento de esta gracias a la automatización de procesos y optimización de recursos y tiempo, por lo cual la ausencia de los mismos en veterinarias significa grandes pérdidas.

En función de la problemática planteada se propone la siguiente pregunta principal:

- ¿Cómo el manejo de historiales médicos en carpetas físicas perjudica el cumplimiento de citas y consultas médicas?

Y las siguientes preguntas secundarias:

- ¿Las veterinarias llevan historial de sus citas médicas realizadas, así como la categoría de las mismas?

- ¿Cómo afecta el uso de carnés físicos a los dueños de mascotas en el cumplimiento de citas a tiempo?
- ¿Cuál es la prioridad y criticidad de cada proceso que necesita automatizar la veterinaria?

### **1.3. Objetivo General**

Desarrollar una aplicación web que automatice la gestión de clientes, mascotas, historiales médicos y citas dentro de centros veterinarios.

### **1.4. Objetivos Específicos**

- Determinar los procesos críticos a automatizar dentro de centros veterinarios.
- Organizar la distribución de médicos a través del sistema de información.
- Organizar la gestión de citas médicas mediante el uso de medios digitales, de manera que se tenga control sobre calendarios, tiempo, etc.

### **1.5. Antecedentes**

La Clínica Veterinaria Veterpet es un centro de salud destinado a la atención de mascotas de diferentes tipos: perros, gatos, conejos, etc.; el cual ofrece una serie de servicios que pasan por la oncología veterinaria, medicina preventiva, emergencias, cirugías, hospitalización, laboratorio clínico e incluso estética canina y felina.

La cantidad de información que se usa a través de sus servicios requiere de una automatización, pues es insostenible mantener tal cantidad de procesos manejados de manera manual y con datos e información física, lo cual dificulta el cumplimiento de determinadas tareas, procesos, etc.

### **1.5.1. Actividades Clínica Veterinaria Veterpet**

Las actividades dentro de esta clínica parten de los servicios respecto a la salud de las mascotas, pero con el paso del tiempo y las necesidades de sus clientes, se han implementado algunos otros servicios y actividades relacionados a la estética tanto canina como felina, así como hospedaje para mascotas.

Cabe mencionar que, debido a la pandemia, Veterpet ha incluido servicios de atención a domicilio para consultas generales realizadas a mascotas, las cuales no dependan de equipos muy complejos.

De manera detallada las actividades que se realizan en la clínica son:

- Oncología veterinaria.
- Vacunación y desparasitación.
- Ecografías.
- Rayos X.
- Actividades realizadas en laboratorio clínico.
- Estética canina y felina.
- Hospedaje.

Esta lista de actividades implica un desglose desde los diferentes actores que la conforman, es decir los pacientes o mascotas, los clientes, y los empleados dentro de la clínica. De una manera detallada, los empleados que se manejan dentro de la veterinaria son doctores en diferentes especialidades puesto que también se consta con área de hospitalización y cirugía; o personal no necesariamente del área medicinal como empleados encargados de servicios de peluquería y estética, empleados a cargo del área de hospitalización, personal para atención al cliente, personal de limpieza, etc.

### **1.5.2. Situación actual de la Clínica Veterinaria Veterpet**

Actualmente, la veterinaria maneja varios procesos de manera manual. No maneja ningún sistema automatizado que permita un mejor manejo por parte de la gestión de historiales y citas médicas, por lo que esta se encuentra limitada a la cantidad de datos que almacena, a un posible crecimiento de la empresa por falta de automatización en procesos; además de falencias críticas en la asignación y notificación de citas a sus clientes, pues muchas veces estos no acuden a las citas gracias a la mala gestión de estas.

### **1.6. Alcance**

El proyecto actual está dirigido a centros veterinarios y dueños de mascotas que acudan a estos, de manera que, a nivel de las veterinarias se pueda tener control sobre procesos como: Gestión de sus clientes, gestión de mascotas, así como sus historiales médicos respectivos, control de citas; y a nivel de clientes tener acceso digital a los carnés o historiales de sus mascotas además de sus citas. Cabe mencionar que el alcance para el presente proyecto se limita al desarrollo de la aplicación, y, si bien se desarrollarán módulos que impliquen la gestión de citas, no se llegará a realizar para todos los tipos de citas, por ejemplo, módulos de gestión de citas para hospitalización, peluquería u otros no serán incluidos en el desarrollo.

## CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

---

### **2. Marco Teórico**

#### **2.1. Metodologías de desarrollo de software**

Las metodologías de desarrollo de software surgen a partir de cubrir necesidades que radican en la mejora de la calidad de sistemas y programas. Estas metodologías han atravesado una serie de cambios, una evolución constante que ha permitido mejorar cada vez más la calidad de los productos y servicios entregados al cliente; metodologías que trabajan en base a conjuntos de procesos, técnicas, documentaciones, etc., para poder estructurar, planificar y controlar el proceso de desarrollo de software.

Dentro de las metodologías que se han creado, se han categorizado principalmente en tradicionales y ágiles, estas últimas con distintos marcos de trabajo.

#### **2.1.1 Metodologías Tradicionales**

##### **2.1.1.1 Modelo en Cascada**

Esta metodología fue construida en 1970 y consiste en un modelo con estructura lineal, donde se trabaja en varias fases estableciendo documentación por cada una de estas donde se verifique el cumplimiento de los objetivos planteados con anterioridad. En el gráfico 1 se observa el modelo planteado.

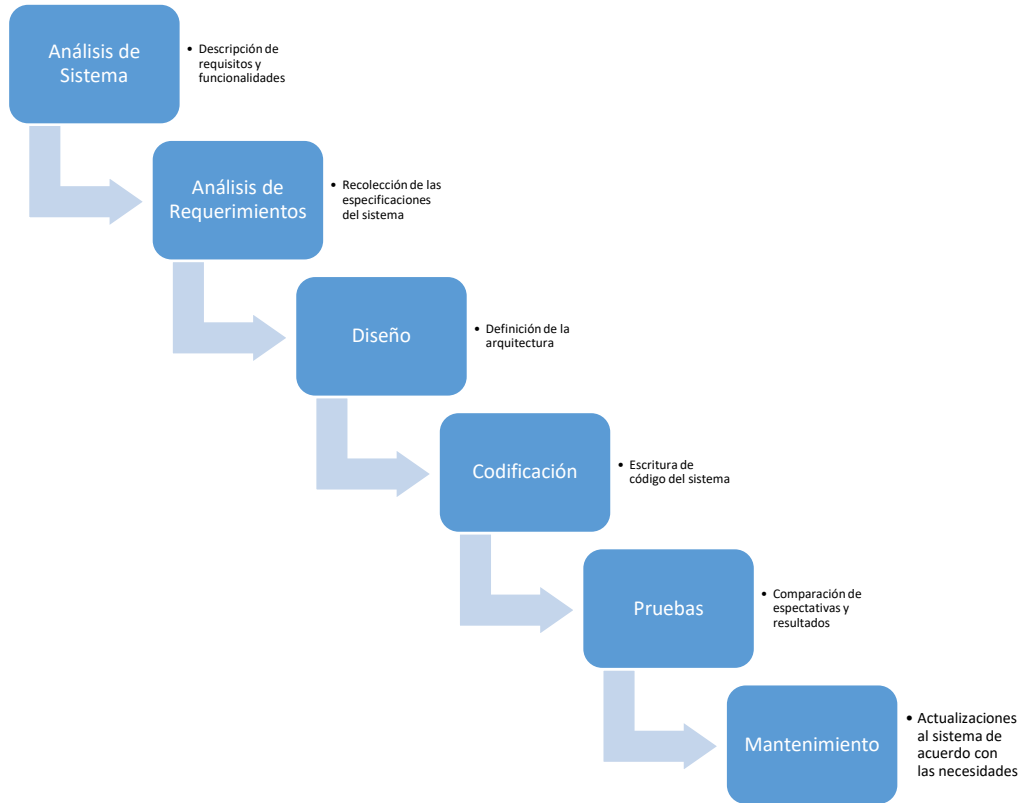


Gráfico 1. Fases del Modelo en Cascada. Autor: Andrés Lozada, 2022

### 2.1.1.2 Modelo en V

El modelo en V consiste una metodología que se basa en la realización de pruebas conforme avanza el desarrollo e integración del sistema, por lo cual se va realizando pruebas por cada fase del modelo, esto se lo observa en el gráfico 2.

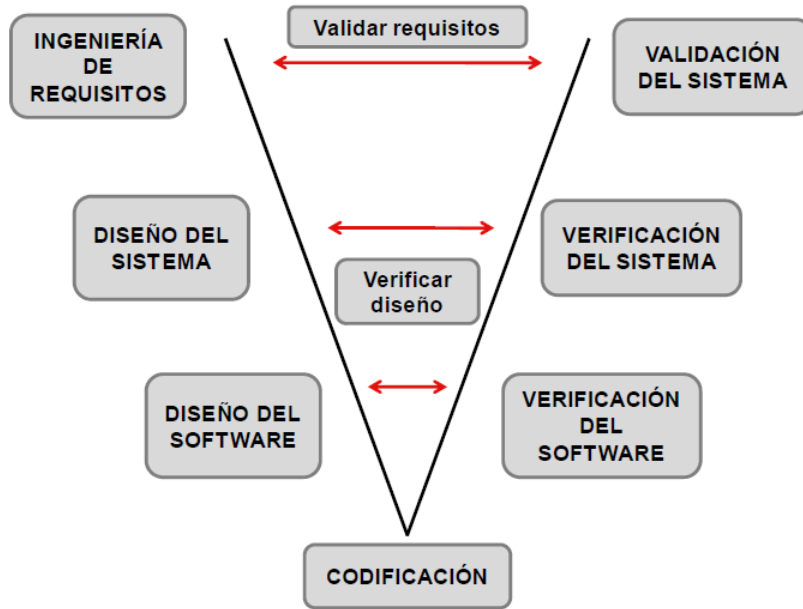


Gráfico 2. Rodríguez, E. (2017). Modelo en V. *Artículos Virtuales*.  
<https://articulosvirtuales.com/articulos/educacion/modelos-de-ciclo-de-vida-de-desarrollo-de-software>

En el Gráfico 2 se observan 2 ramas que forman parte de este modelo, la de desarrollo y la de pruebas, en las cuales las fases corresponden una a una horizontalmente, es decir, la validación del sistema corresponde a pruebas realizadas a la Ingeniería de Requisitos del lado del desarrollo, la Verificación del Sistema al Diseño del Sistema, La Verificación del Software al Diseño del Software y únicamente la codificación se mantiene en la punta de la V.

### 2.1.2 Metodologías Ágiles

Las metodologías ágiles trabajan de acuerdo con diferentes frameworks establecidos, entre ellos existen marcos de trabajo como eXtreme Programing, SCRUM, entre otros. Estas metodologías, permiten mayor agilidad en los procesos de desarrollo de software de manera que se optimice tiempo en los procesos e implica mucha menos documentación a comparación de los métodos tradicionales.

#### 2.1.2.1 SCRUM

Scrum es un tipo de marco de trabajo para la metodología ágil que se basa en tres principios: transparencia, pues debe ser accesible y visible para todos los involucrados en el proyecto;

inspección, haciendo referencia al control periódico sobre problemas, cambios y otros; y adaptación, ya que ante posibles cambios se debe estar listo para adaptarse y evitar complicaciones.

Además, esta metodología trabaja en base a roles, estos son:

- Scrum Master: Líder de equipo que guía a este a través de reuniones y artefactos.
- Product Owner: Supervisa al equipo y facilita la comunicación entre los desarrolladores. También genera el Product Backlog.
- Developers: También llamados desarrolladores son quienes se encargan del desarrollo del software.

Además, el marco SCRUM, presenta una serie de eventos mediante los cuales se va direccionando el proyecto, estos giran en torno a los Sprints, los cuales consisten en etapas donde se desarrolla el software y concluye con la entrega de software funcional implementando nuevos módulos o características. Estos Sprints pasan por una fase de planificación, donde se definen las tareas para este; también la revisión donde se compara respecto a los objetivos planteados y una retrospectiva donde se buscan maneras de mejorar la eficiencia en el proyecto.

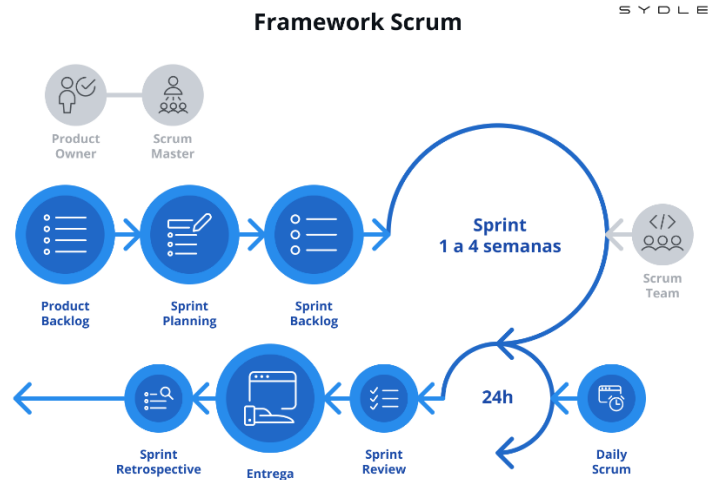


Gráfico 3. Sydle. (2022). Framework Scrum. Sydle. <https://www.sydle.com/es/blog/pmbok-61e80383f41fbf069eb3ef2b/>

Para el proyecto presente, se ha escogido la metodología ágil con el marco de trabajo SCRUM, gracias a todos los puntos mencionados con anterioridad, es decir, gracias a que esta metodología permite una mayor flexibilidad a la hora de desarrollo de aplicaciones, así como la revisión y retrospectión de estas.

## 2.2. Lenguajes de programación

A Un lenguaje de programación consiste en un conjunto de símbolos y códigos usados para escribir un conjunto de órdenes, acciones consecutivas, datos y algoritmos, de manera que se pueda desarrollar programas completos que permitan controlar el comportamiento físico y lógico del computador.

Este sistema estructurado de comunicación hace uso de reglas tanto semánticas y sintácticas para lograr un entendimiento entre los programadores y máquinas.

Para el presente proyecto se ha escogido el lenguaje de programación Typescript, pues este lenguaje se encuentra vinculado al desarrollo en el framework Angular. Typescript es lo que se conoce como un superset de Javascript. Esto significa que Typescript es una tecnología que

puede ejecutar programas propios de este, así como del lenguaje del cual es un superset, para el caso Javascript. Typescript surge a partir de inconvenientes presentados en Javascript para bases de código grandes, aplicaciones de gran escala y proyectos que han perdurado en el tiempo.

### **2.3. Frameworks de desarrollo web**

Un framework o marco de trabajo, dentro del ámbito del desarrollo de sistemas, consiste en una estructura software compuesta de componentes, los cuales pueden ser personalizables e intercambiables para el desarrollo del software. Por lo tanto, un framework consiste en una plataforma que provee los cimientos para el desarrollo software, el cual permite añadir o reemplazar funcionalidades para otorgar nuevas funcionalidades a la aplicación.

Dentro de los objetivos que presentan los frameworks se encuentra el reducir el tiempo de desarrollo, la reutilización de código o componentes y también el fomentar y facilitar el uso de patrones y buenas prácticas de desarrollo.

Ahora bien, un web framework es todo lo mencionado anteriormente con la característica que el conjunto de componentes de este, sus objetivos y funcionalidades están destinados al desarrollo de sistemas web, y así mismo buscará la agilidad, reducción de tiempos, reutilización de elementos y facilidad de desarrollo para este tipo de sistemas.

Es necesario tomar en cuenta que para el desarrollo web se cuentan con frameworks para Back-End y para Front-End. Dentro de los frameworks para Back-End se puede incluir a Django, un web framework que utiliza Python y permite gran velocidad, seguridad y escalabilidad; mientras que para Front-End se tiene frameworks como: Angular, React, Vue, entre otros.

Para escoger el framework para el presente proyecto, se consultaron las tecnologías más populares en este ámbito, obteniendo los siguientes resultados.

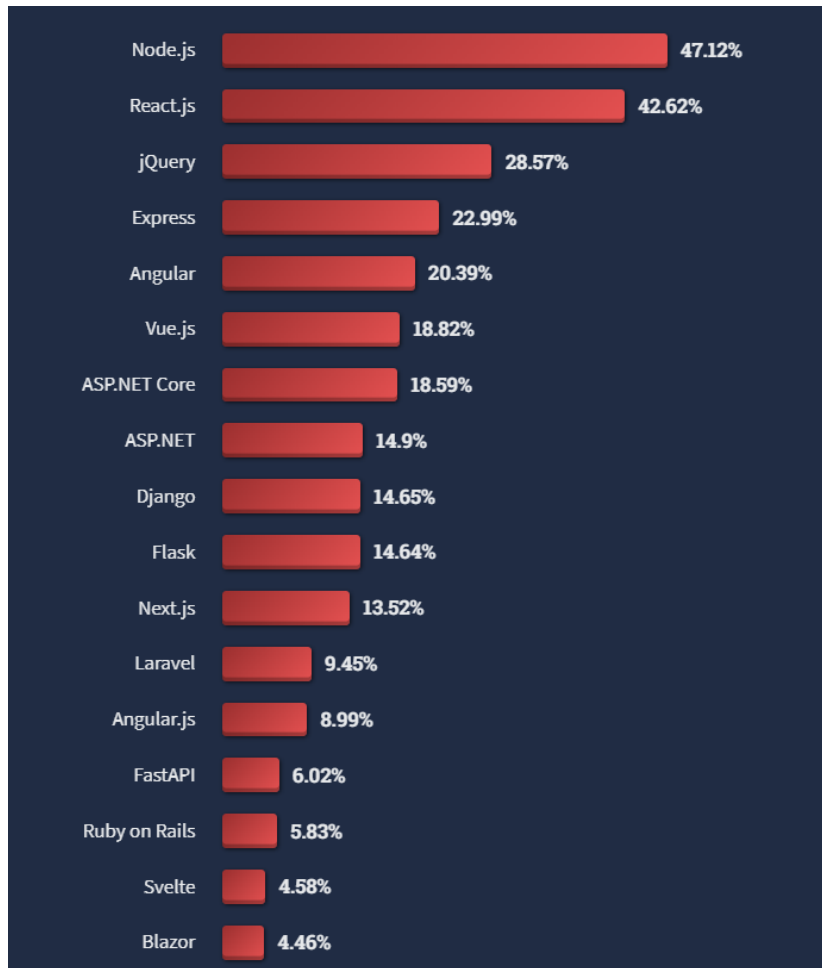


Gráfico 4. Stack Overflow. (2022). Most Popular Technologies. Stack Overflow.  
<https://survey.stackoverflow.co/2022/#most-loved-dreaded-and-wanted-webframe-love-dread>

Como se observa en el gráfico 4 existe una gran variedad de frameworks para desarrollo web populares, lo cual implica que resulta factible el uso de cualquiera de estos; por lo tanto, la decisión para escoger alguno de estos frameworks para el proyecto presente recae en otros aspectos como conocimientos previos, facilidad de acoplamiento, entre otros.

Para este proyecto se ha escogido desarrollar con Angular, el cual se encuentra entre los 5 frameworks más populares dentro de la lista del gráfico 4.

Detallando con mayor profundidad, Angular está escrito en Typescript, combina el uso del lenguaje de marcado HTML y su propio lenguaje (Typescript), en conjunto con CSS, LESS o SASS para la construcción de módulos, componentes, proporcionar estilos a los mismos, etc.

La arquitectura de Angular se encuentra cimentado en algunos conceptos fundamentales. En primera instancia, como bloques base de su arquitectura, se tienen a los NgModules, los cuales proveen un contexto de compilación a los componentes. Estos NgModules recopilan código relacionado en conjuntos funcionales y definen, en conjunto, la aplicación de Angular. Cabe mencionar que toda aplicación con este framework contiene al menos un módulo raíz que permite el inicio de la aplicación, y contiene más módulos de funcionalidad.

Por otra parte, se tienen los componentes, los cuales definen vistas, es decir, conjuntos de elementos en pantalla que Angular tiene la capacidad de manejar, alterar y escoger de acuerdo con la lógica del programa desarrollado. Estos componentes consumen servicios que proporcionan funcionalidades específicas no directamente relacionadas con las vistas mencionadas y que también pueden inyectarse en estos componentes como dependencias, permitiendo la reutilización, eficiencia y modularidad.

#### **2.4. Sistema Gestor de Contenidos (CMS)**

Un CMS es un software que facilita la gestión de contenidos para el desarrollo de una aplicación web. Este permite la administración de la estructura del contenido y facilita la subida de datos al mismo. Uno de los sistemas gestores de contenido más reconocidos y dirigidos a desarrolladores es Strapi, el cual facilita la administración del backend al permitir el guardado de en bases de datos como PostgreSQL, MySQL, etc., además de proporcionar el acceso a API's para la gestión de su contenido, las cuales pueden ser administradas por ejemplo para restricción de permisos para diferentes tipos de usuario. Para el presente proyecto se usará este CMS mencionado, Strapi, debido a los beneficios que presenta para la creación de API's, autenticación y restricción de usuarios y flexibilidad que posee para la gestión de contenido en general.

#### **2.5. Bases de Datos**

Una base de datos es un conjunto de datos que pertenecen a un mismo contexto y almacenados de manera sistemática para su posterior uso. Aquí también surge el término

Sistemas Gestores de Bases de Datos que permiten almacenar los datos y acceder a ellos de forma rápida y estructurada.

Las bases de datos pueden ser definidas como sistemas que almacenan, organizan y administran datos pertenecientes a un mismo contexto, a la vez que controlan redundancia, consistencia, seguridad y accesibilidad a estos (Vacacela, Tenecota, Torres & Celleri, 2018, 35-44).

Además, es necesario mencionar que existen bases de tipo relacional y no relacional, lo cual influye en si los datos almacenados contienen accesos a puntos de datos relacionados entre sí, para el caso de bases de datos relacionales, o si no incluyen estos para bases de tipo no relacional. Hoy en día las bases relacionales conforman el modelo más utilizado.

Para el proyecto presente se ha decidido usar base de datos de tipo relacional, específicamente PostgreSQL. PostgreSQL es una base de datos relacional, de código abierto, que usa y extiende el lenguaje SQL y lo combina con más funcionalidades que permiten un almacenamiento de datos seguro y escalable. PostgreSQL es reconocido por su arquitectura, confiabilidad, mantenimiento de la integridad de los datos, funcionalidades robustas, etc.

Esta base de datos puede correr en cualquier sistema operativo de mayor nivel y cumple con el estándar ACID, es decir: Atomicidad, Consistencia, Aislamiento y Durabilidad.

### 3. Análisis, diseño de la aplicación web basado en Scrum

#### 3.1. Arquitectura de Software

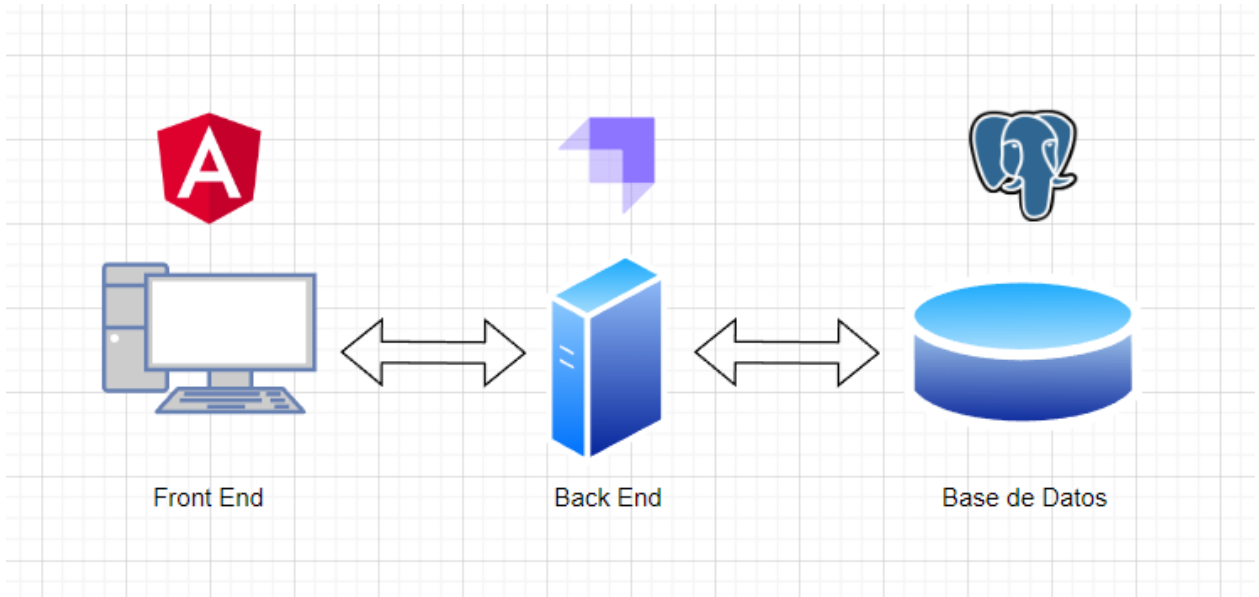
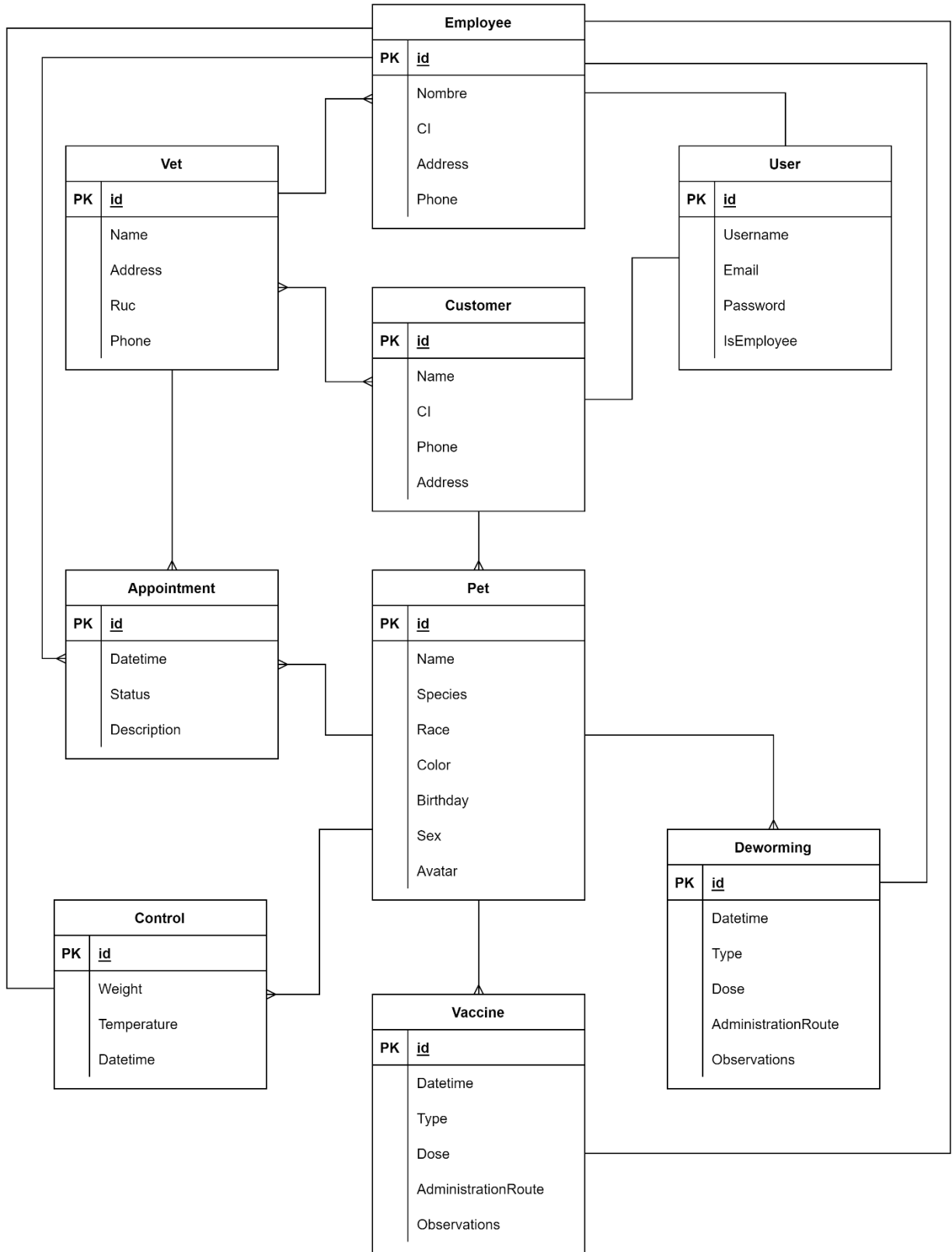


Gráfico 5. Arquitectura de Software. Autor: Andrés Lozada, 2022

### 3.2. Modelo Entidad Relación



### 3.3. Resumen de Historias de Usuario

Las historias de usuario se han obtenido a través de una reunión realizada con el administrador principal de la veterinaria que también forma parte de los doctores encargados de atender a pacientes o mascotas. Dentro de estas historias de usuario se han declarado ciertos actores y procesos los cuales también han sido afectados por el tiempo en el cual se realiza el proyecto.

Las historias de usuario pueden resumirse de la siguiente manera:

- Administrador de la veterinaria
  - Inicio de sesión mediante 2 parámetros: usuario y contraseña.
  - Administración total de la veterinaria y su información.
  - Capacidad de crear, visualizar, editar y eliminar registros de sus empleados de todo tipo.
  - Capacidad de crear, visualizar, editar y eliminar registros de sus clientes o dueños de mascotas que acuden al centro.
  - Capacidad de crear, visualizar, editar y eliminar registros de sus pacientes o mascotas de la veterinaria.
  - Capacidad de crear, visualizar, editar y eliminar registros de los historiales médicos de las mascotas.
  - Gestión de citas gestionadas a través de un calendario que permite la visualización y creación de estas con un apartado para la completa administración de registros.
  - Consulta de citas filtradas por la fecha y tipo.
- Empleado de la veterinaria
  - Inicio de sesión mediante 2 parámetros: usuario y contraseña.

- Capacidad de crear, visualizar, editar y eliminar registros de sus clientes o dueños de mascotas que acuden al centro.
- Capacidad de crear, visualizar, editar y eliminar registros de sus pacientes o mascotas de la veterinaria.
- Capacidad de crear, visualizar, editar y eliminar registros de los historiales médicos de las mascotas.
- Gestión de citas gestionadas a través de un calendario que permite la visualización y creación de estas con un apartado para la completa administración de registros.
- Consulta de citas filtradas por la fecha y tipo.
- Clientes
  - Registro de su identidad a través de usuario, contraseña y más datos personales.
  - Inicio de sesión mediante 2 parámetros: usuario y contraseña.
  - Visualización de sus citas.
  - Visualización del historial médico de su mascota.

### 3.4. Casos de uso de la aplicación

#### 3.4.1. Casos de uso para Administrador y Empleados de la veterinaria

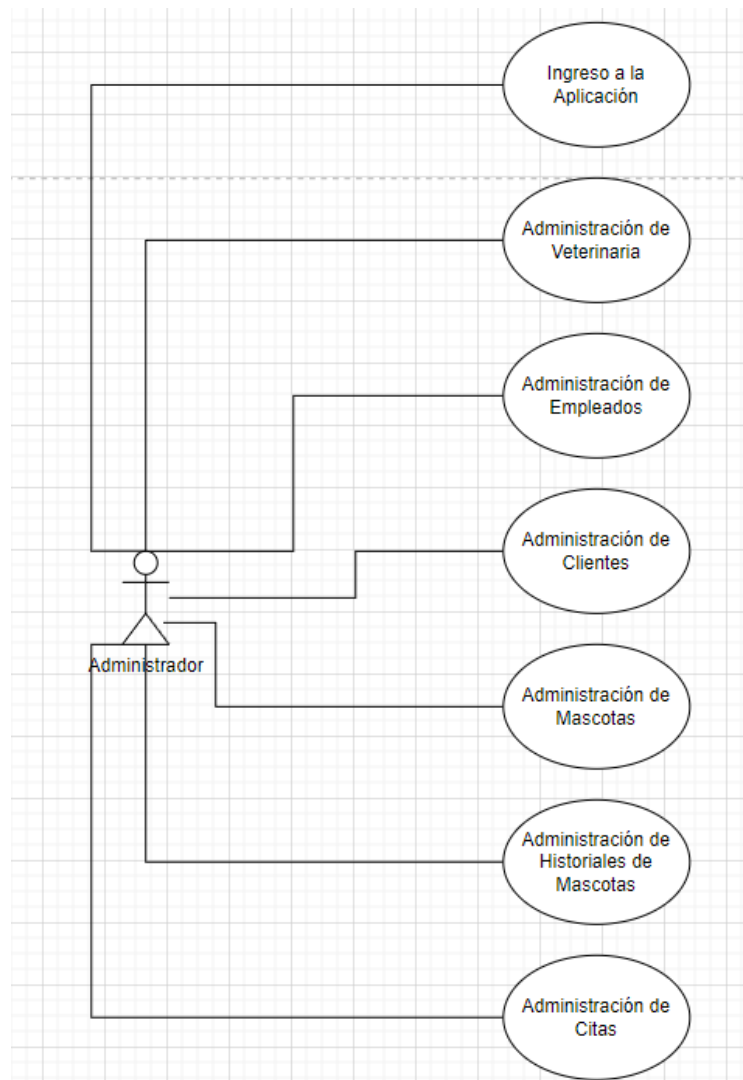


Gráfico 7. Caso de uso de Administrador. Autor: Andrés Lozada, 2022

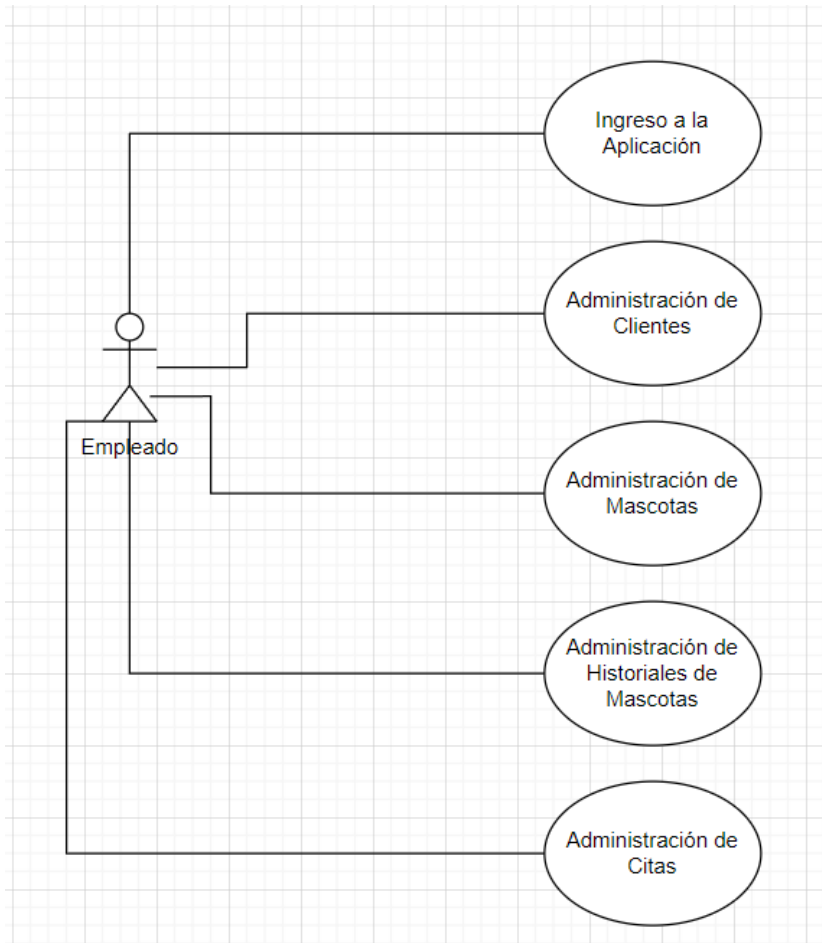


Gráfico 8. Caso de uso Empleado Autor: Andrés Lozada, 2022

3.4.1.1. Caso de Uso: Administración de Veterinaria

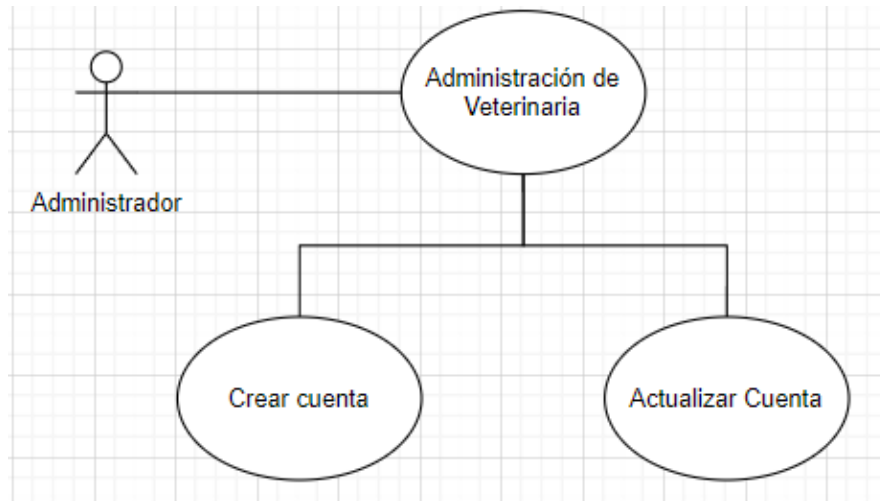


Gráfico 9. Caso de uso: Administración de veterinaria. Autor: Andrés Lozada, 2022

#### 3.4.1.1.1. Caso de uso: Crear cuenta

**Descripción:** El actor podrá crear una nueva cuenta de veterinaria.

**Actores:** Administrador.

#### **Flujo Principal:**

1. El actor selecciona el apartado de Crear cuenta.
2. El sistema presenta 2 opciones: registro como veterinaria y como cliente.
3. El actor escoge Veterinaria.
4. El sistema carga el formulario respectivo para veterinaria.
5. El actor completa el formulario.
6. El actor presiona el botón Crear cuenta.
7. El sistema crea una nueva veterinaria.
8. El sistema inicia la sesión del administrador.

#### 3.4.1.1.2. Caso de uso: Actualizar cuenta

**Descripción:** El actor podrá actualizar su cuenta de veterinaria.

**Actores:** Administrador.

**Flujo Principal:**

1. El actor selecciona el apartado Perfil.
2. El actor selecciona Actualizar Datos.
3. El sistema carga el formulario con los datos existentes de la veterinaria.
4. El actor modifica los datos.
5. El actor presiona el botón Guardar.
6. El sistema actualiza el registro.
7. El sistema muestra el mensaje de éxito o error respectivamente.

3.4.1.2. Caso de uso: Administración de Empleados

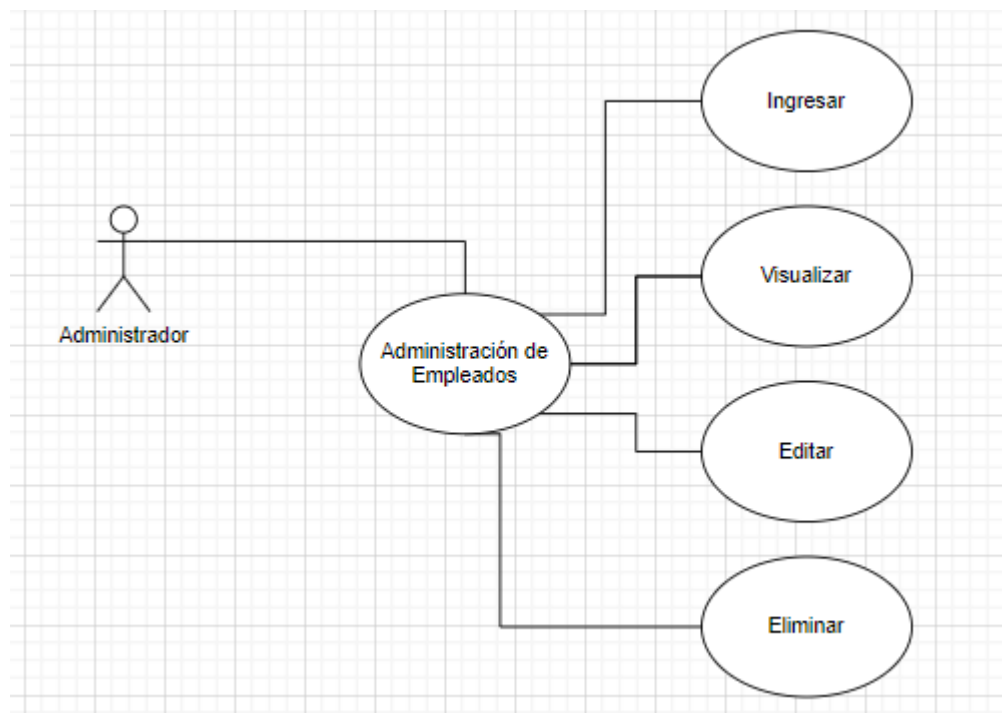


Gráfico 10. Caso de uso: Administración de Empleados. Autor: Andrés Lozada, 2022

3.4.1.2.1. Caso de uso: Ingresar

**Descripción:** El actor podrá ingresar empleados a la veterinaria.

**Actores:** Administrador

### **Flujo Principal**

1. El actor va al apartado Mi Veterinaria.
2. EL sistema carga la página de Mi Veterinaria.
3. El actor va a la pestaña Empleados.
4. El sistema carga este apartado.
5. El actor presiona el botón Nuevo Empleado.
6. El sistema muestra el formulario respectivo.
7. El actor llena los datos.
8. El actor presiona Guardar.
9. El sistema guarda el registro.
10. El sistema notifica el éxito o error.
11. El sistema carga al nuevo empleado en el listado en caso de resultar exitoso.

3.4.1.2.2. Caso de uso: Actualizar

**Descripción:** El actor podrá actualizar empleados a la veterinaria.

**Actores:** Administrador

### **Flujo Principal**

1. El actor va al apartado Mi Veterinaria.
2. EL sistema carga la página de Mi Veterinaria.
3. El actor va a la pestaña Empleados.
4. El sistema carga este apartado.
5. El actor presiona el botón de editar en el empleado deseado.
6. El sistema muestra el formulario respectivo.
7. El actor cambia los datos.

8. El actor presiona Guardar.
9. El sistema guarda el registro.
10. El sistema notifica el éxito o error.
11. El sistema carga al empleado actualizado en el listado en caso de resultar exitoso.

#### 3.4.1.2.3. Caso de uso: Eliminar

**Descripción:** El actor podrá eliminar empleados a la veterinaria.

**Actores:** Administrador

#### **Flujo Principal**

1. El actor va al apartado Mi Veterinaria.
2. EL sistema carga la página de Mi Veterinaria.
3. El actor va a la pestaña Empleados.
4. El sistema carga este apartado.
5. El actor presiona el botón de eliminar en el empleado deseado.
6. El sistema muestra un mensaje de confirmación.
7. El actor presiona Confirmar.
8. El sistema elimina el registro.
9. El sistema notifica el éxito o error.
10. El sistema carga la lista sin el empleado en caso de resultar exitoso.

#### 3.4.1.2.4. Caso de uso: Visualizar

**Descripción:** El actor podrá visualizar empleados a la veterinaria.

**Actores:** Administrador

#### **Flujo Principal**

1. El actor va al apartado Mi Veterinaria.

2. EL sistema carga la página de Mi Veterinaria.
3. El actor va a la pestaña Empleados.
4. El sistema carga el listado de empleados.

3.4.1.3. Caso de uso: Ingresar al sistema

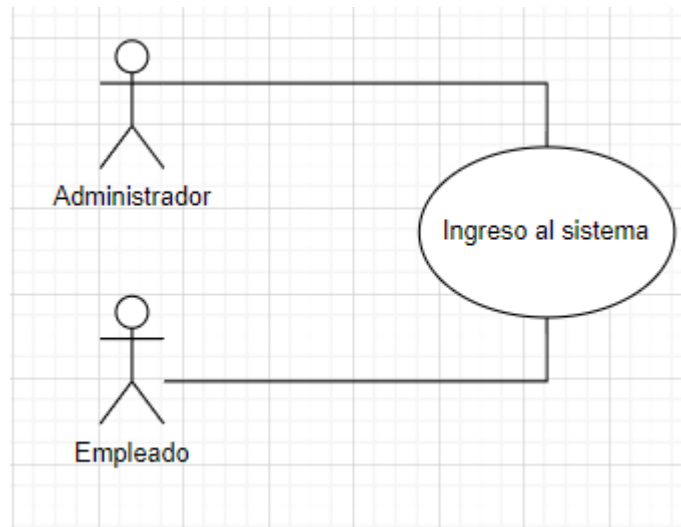


Gráfico 11. Caso de uso: Ingreso al sistema. Autor: Andrés Lozada, 2022

**Descripción:** El actor puede ingresar al sistema como miembro de veterinaria.

**Actores:** Administrador, Empleado

**Flujo Principal:**

1. El actor ingresa al apartado Iniciar Sesión.
2. El sistema carga la pantalla respectiva.
3. El actor ingresa su usuario y contraseña.
4. El sistema valida sus credenciales.
5. El sistema notifica el resultado.
6. El sistema navega al Inicio en caso de ser exitoso el proceso.

#### 3.4.1.4. Caso de uso: Administración de Clientes

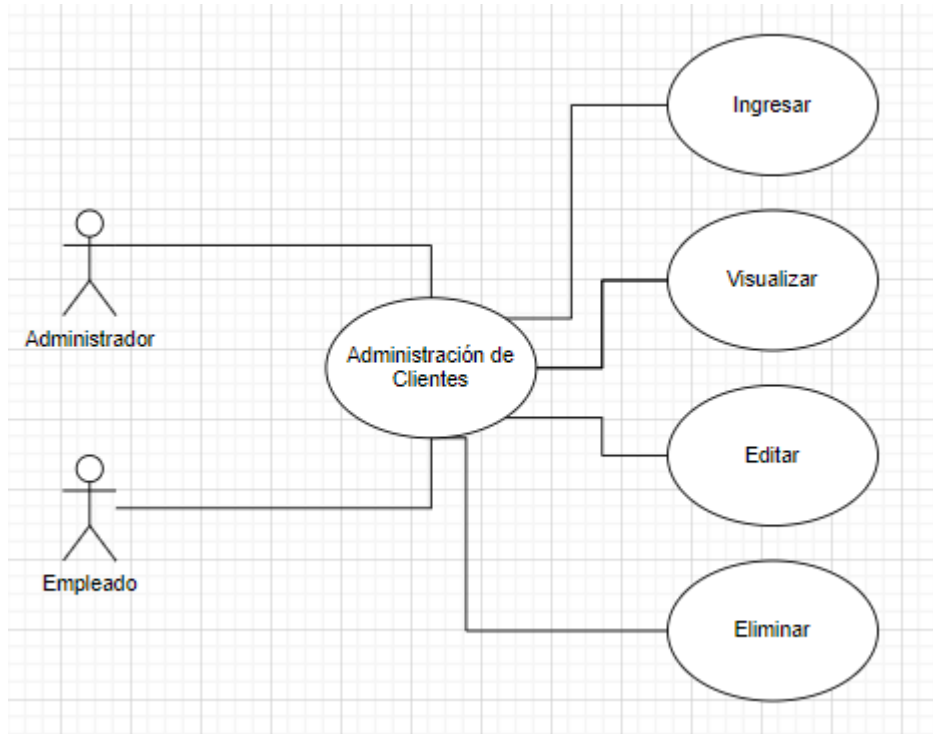


Gráfico 12. Caso de uso: Administración de Clientes. Autor: Andrés Lozada, 2022

##### 3.4.1.4.1. Caso de uso: Ingresar

**Descripción:** El actor podrá ingresar clientes a la veterinaria.

**Actores:** Administrador, Empleado

#### Flujo Principal

1. El actor va al apartado Clientes.
2. El sistema carga la página de Clientes.
3. El actor presiona el botón Nuevo Cliente.
4. El sistema muestra el formulario respectivo.
5. El actor llena los datos.
6. El actor presiona Guardar.
7. El sistema guarda el registro.

8. El sistema notifica el éxito o error.
9. El sistema carga al nuevo empleado en el listado en caso de resultar exitoso.

#### 3.4.1.4.2. Caso de uso: Actualizar

**Descripción:** El actor podrá actualizar clientes de la veterinaria.

**Actores:** Administrador, Empleado

#### **Flujo Principal**

1. El actor va al apartado Clientes.
2. El sistema carga la página de Clientes.
3. El actor presiona el botón de editar en el cliente deseado.
4. El sistema muestra el formulario respectivo.
5. El actor cambia los datos.
6. El actor presiona Guardar.
7. El sistema guarda el registro.
8. El sistema notifica el éxito o error.
9. El sistema carga al empleado actualizado en el listado en caso de resultar exitoso.

#### 3.4.1.4.3. Caso de uso: Eliminar

**Descripción:** El actor podrá eliminar clientes de la veterinaria.

**Actores:** Administrador, Empleado

#### **Flujo Principal**

1. El actor va al apartado Clientes.
2. El sistema carga la página de Clientes.
3. El actor presiona el botón de eliminar en el cliente deseado.
4. El sistema muestra un mensaje de confirmación.

5. El actor presiona Confirmar.
6. El sistema elimina el registro.
7. El sistema notifica el éxito o error.
8. El sistema carga la lista sin el cliente en caso de resultar exitoso.

#### 3.4.1.4.4. Caso de uso: Visualizar

**Descripción:** El actor podrá visualizar clientes de la veterinaria.

**Actores:** Administrador, Empleado

#### **Flujo Principal**

1. El actor va al apartado Clientes.
2. El sistema carga el listado de clientes.

#### 3.4.1.5. Caso de uso: Administración de Mascotas

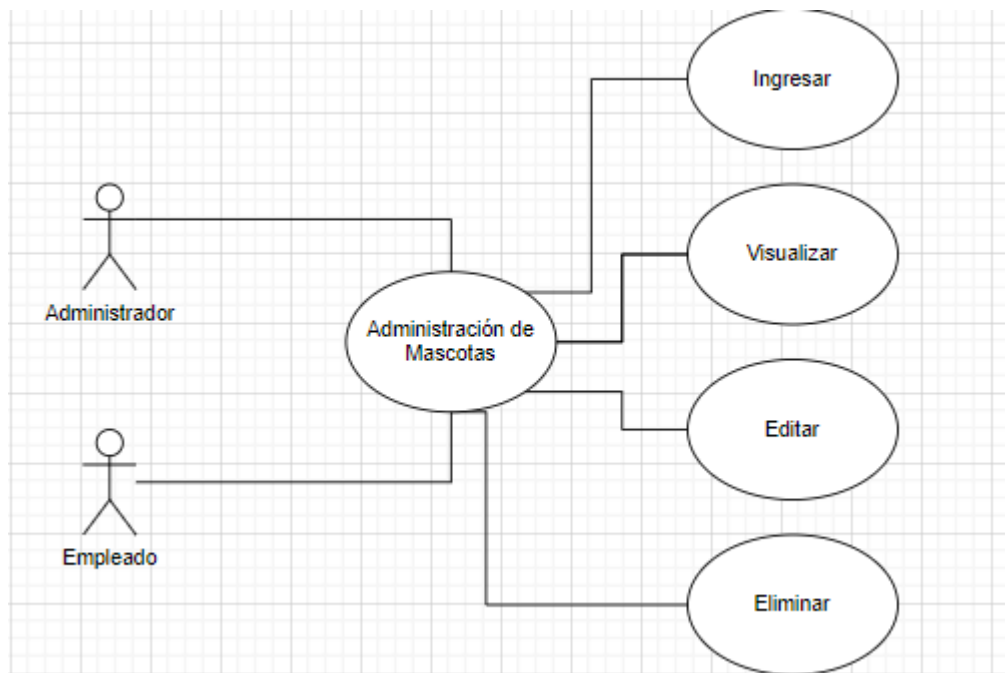


Gráfico 13. Caso de uso: Administración de Mascotas. Autor: Andrés Lozada, 2022

#### 3.4.1.5.1. Caso de uso: Ingresar

**Descripción:** El actor podrá ingresar mascotas a la veterinaria.

**Actores:** Administrador, Empleado

##### **Flujo Principal**

1. El actor va al apartado Clientes.
2. El sistema carga la página de Clientes.
3. El actor selecciona un cliente.
4. El sistema carga el listado de sus mascotas.
5. El actor presiona el botón Nueva Mascota.
6. El sistema muestra el formulario respectivo.
7. El actor llena los datos.
8. El actor presiona Guardar.
9. El sistema guarda el registro.
10. El sistema notifica el éxito o error.
11. El sistema carga a la nueva mascota en el listado en caso de resultar exitoso.

#### 3.4.1.5.2. Caso de uso: Actualizar

**Descripción:** El actor podrá actualizar mascotas de la veterinaria.

**Actores:** Administrador, Empleado

##### **Flujo Principal**

1. El actor va al apartado Clientes.
2. El sistema carga la página de Clientes.
3. El actor selecciona un cliente.
4. El sistema carga el listado de sus mascotas.

5. El actor presiona el botón de editar en la mascota deseada.
6. El sistema muestra el formulario respectivo.
7. El actor cambia los datos.
8. El actor presiona Guardar.
9. El sistema guarda el registro.
10. El sistema notifica el éxito o error.
11. El sistema carga a la mascota actualizada en el listado en caso de resultar exitoso.

#### 3.4.1.5.3. Caso de uso: Eliminar

**Descripción:** El actor podrá eliminar mascotas de la veterinaria.

**Actores:** Administrador, Empleado

#### **Flujo Principal**

1. El actor va al apartado Clientes.
2. El sistema carga la página de Clientes.
3. El actor selecciona un cliente.
4. El sistema carga el listado de sus mascotas.
5. El actor presiona el botón de eliminar en la mascota deseada.
6. El sistema muestra un mensaje de confirmación.
7. El actor presiona Confirmar.
8. El sistema elimina el registro.
9. El sistema notifica el éxito o error.
10. El sistema carga la lista sin la mascota en caso de resultar exitoso.

#### 3.4.1.5.4. Caso de uso: Visualizar

**Descripción:** El actor podrá visualizar mascotas de la veterinaria.

**Actores:** Administrador, Empleado

## Flujo Principal

1. El actor va al apartado Clientes.
2. El sistema carga el listado de clientes.
3. El actor selecciona un cliente.
4. El sistema carga el listado de sus mascotas.

### 3.4.1.6. Caso de uso: Administración de Historiales de Mascotas

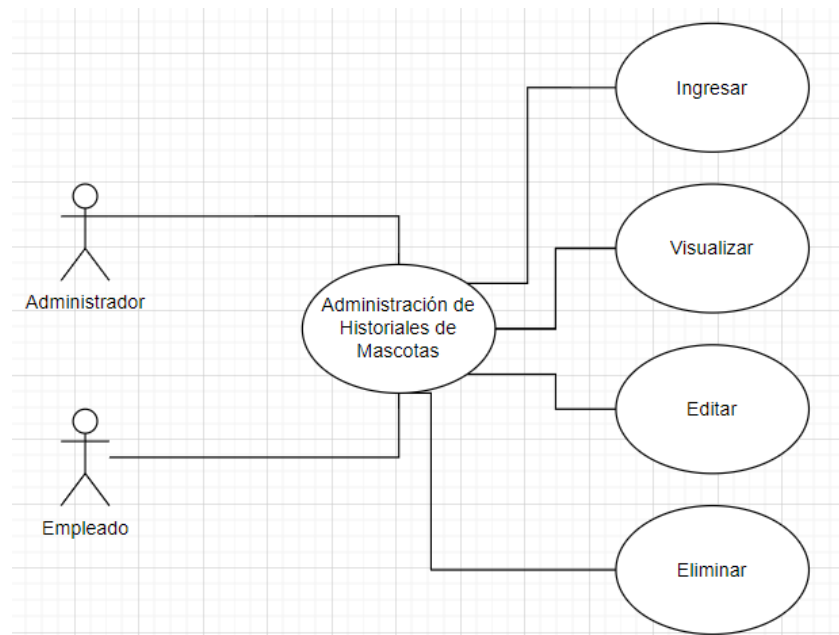


Gráfico 14. Caso de uso: Administración de Historiales de Mascotas. Autor: Andrés Lozada, 2022

#### 3.4.1.6.1. Caso de uso: Ingresar

**Descripción:** El actor podrá ingresar historiales de mascotas a la veterinaria.

**Actores:** Administrador, Empleado

## Flujo Principal

1. El actor va al apartado Clientes.
2. El sistema carga la página de Clientes.
3. El actor selecciona un cliente.

4. El sistema carga el listado de sus mascotas.
5. El actor selecciona una Mascota.
6. El sistema muestra todo el historial de la mascota.
7. El actor escoge para ingresar un nuevo registro al historial.
8. El sistema despliega el formulario.
9. El actor llena los datos.
10. El actor presiona Guardar.
11. El sistema guarda el registro.
12. El sistema notifica el éxito o error.
13. El sistema carga al nuevo registro en el listado en caso de resultar exitoso.

#### 3.4.1.6.2. Caso de uso: Actualizar

**Descripción:** El actor podrá actualizar historiales de mascotas de la veterinaria.

**Actores:** Administrador, Empleado

#### **Flujo Principal**

1. El actor va al apartado Clientes.
2. El sistema carga la página de Clientes.
3. El actor selecciona un cliente.
4. El sistema carga el listado de sus mascotas.
5. El actor selecciona una Mascota.
6. El sistema muestra todo el historial de la mascota.
7. El actor escoge el registro a modificar.
8. El sistema despliega el formulario.
9. El actor llena los datos.
10. El actor presiona Guardar.

11. El sistema actualiza el registro.
12. El sistema notifica el éxito o error.
13. El sistema carga al registro con los nuevos datos en el listado en caso de resultar exitoso.

13.1.1.1.1. Caso de uso: Eliminar

**Descripción:** El actor podrá eliminar historiales de mascotas de la veterinaria.

**Actores:** Administrador, Empleado

**Flujo Principal**

1. El actor va al apartado Clientes.
2. El sistema carga la página de Clientes.
3. El actor selecciona un cliente.
4. El sistema carga el listado de sus mascotas.
5. El actor selecciona una Mascota.
6. El sistema muestra todo el historial de la mascota.
7. El actor escoge la opción para eliminar el registro deseado.
8. El sistema muestra un mensaje de confirmación.
9. El actor presiona Confirmar.
10. El sistema elimina el registro.
11. El sistema notifica el éxito o error.
12. El sistema carga la lista sin el registro en caso de resultar exitoso.

13.1.1.1.2. Caso de uso: Visualizar

**Descripción:** El actor podrá visualizar historiales de mascotas de la veterinaria.

**Actores:** Administrador, Empleado

## Flujo Principal

1. El actor va al apartado Clientes.
2. El sistema carga la página de Clientes.
3. El actor selecciona un cliente.
4. El sistema carga el listado de sus mascotas.
5. El actor selecciona una Mascota.
6. El sistema muestra todo el historial de la mascota.

### 3.4.1.7. Administración de Citas

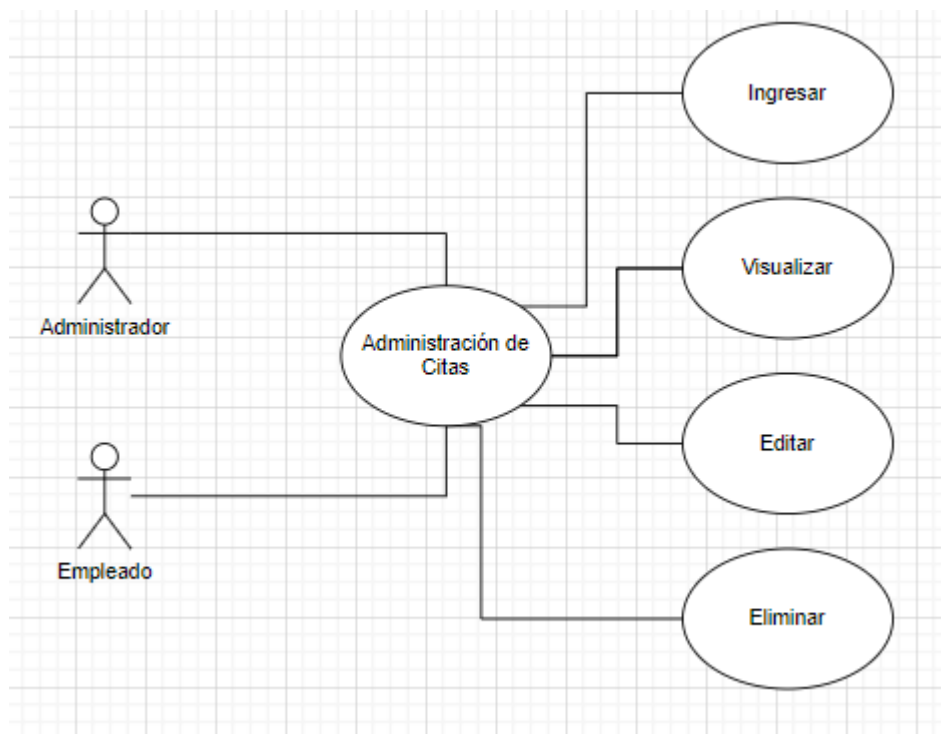


Gráfico 15. Caso de Uso: Administración de Citas. Autor: Andrés Lozada, 2022

#### 3.4.1.7.1. Caso de uso: Ingresar

**Descripción:** El actor podrá ingresar citas de la veterinaria.

**Actores:** Administrador, Empleado

## Flujo Principal

1. El actor va al apartado Calendario.
2. El sistema carga la página de Calendario.
3. El actor presiona la casilla con la fecha en la que quiera la cita.
4. El sistema despliega el horario del día con un botón para agregar la cita.
5. El actor presiona el botón
6. El sistema despliega un formulario.
7. El actor llena el formulario.
8. El actor presiona Guardar.
9. El sistema guarda el registro.
10. El sistema notifica el éxito o error.
11. El sistema carga la nueva cita en el calendario en caso de resultar exitoso.

#### 3.4.1.7.2. Caso de uso: Actualizar

**Descripción:** El actor podrá actualizar clientes de la veterinaria.

**Actores:** Administrador, Empleado

#### **Flujo Principal**

1. El actor va al apartado Calendario.
2. El sistema carga la página de Calendario.
3. El actor presiona en la casilla con la cita a modificar deseada.
4. El sistema despliega las citas del día con más detalle.
5. El actor presiona el enlace con la cita a modificar deseada.
6. El sistema navega a la página del historial de la mascota con la cita respectiva.
7. El actor presiona el botón Editar en el registro correspondiente.
8. El sistema despliega el formulario de la cita actual.
9. El actor edita los datos de la cita.

10. El actor presiona Guardar.
11. El sistema guarda el registro.
12. El sistema notifica el éxito o error.
13. El sistema carga la cita actualizada en el listado en caso de resultar exitoso.

#### 3.4.1.7.3. Caso de uso: Eliminar

**Descripción:** El actor podrá eliminar citas de la veterinaria.

**Actores:** Administrador, Empleado

#### **Flujo Principal**

1. El actor va al apartado Calendario.
2. El sistema carga la página de Calendario.
3. El actor presiona en la casilla con la cita a modificar deseada.
4. El sistema despliega las citas del día con más detalle.
5. El actor presiona el enlace con la cita a modificar deseada.
6. El sistema navega a la página del historial de la mascota con la cita respectiva.
7. El actor presiona Eliminar en el registro respectivo.
8. El sistema muestra un mensaje de confirmación.
9. El actor presiona Confirmar.
10. El sistema elimina el registro.
11. El sistema notifica el éxito o error.
12. El sistema carga la lista sin la cita en caso de resultar exitoso.

#### 3.4.1.7.4. Caso de uso: Visualizar

**Descripción:** El actor podrá visualizar citas de la veterinaria.

**Actores:** Administrador, Empleado

## Flujo Principal

1. El actor va al apartado Calendario.
2. El sistema carga el listado de citas en un calendario para mejor organización.

### 3.4.2. Casos de uso para Clientes

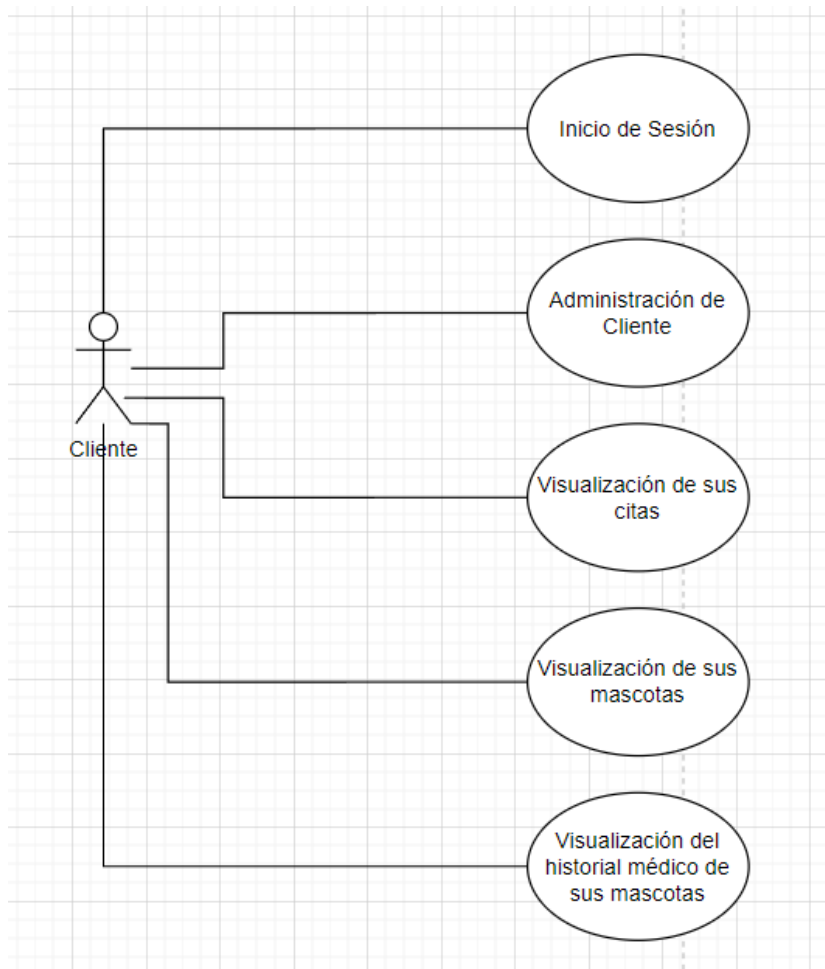


Gráfico 16. Casos de Uso para Clientes. Autor: Andrés Lozada, 2022

### 3.4.2.1. Caso de Uso: Inicio de Sesión

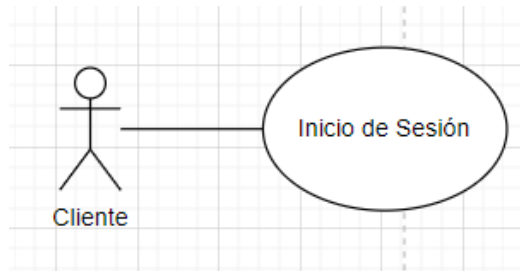


Gráfico 17. Caso de Uso: Inicio de Sesión. Autor: Andrés Lozada, 2022

**Descripción:** El actor puede ingresar al sistema como cliente.

**Actores:** Cliente

**Flujo Principal:**

1. El actor ingresa al apartado Iniciar Sesión.
2. El sistema carga la pantalla respectiva.
3. El actor ingresa su usuario y contraseña.
4. El sistema valida sus credenciales.
5. El sistema notifica el resultado.
6. El sistema navega al Inicio en caso de ser exitoso el proceso.

### 3.4.2.2. Administración de Cliente

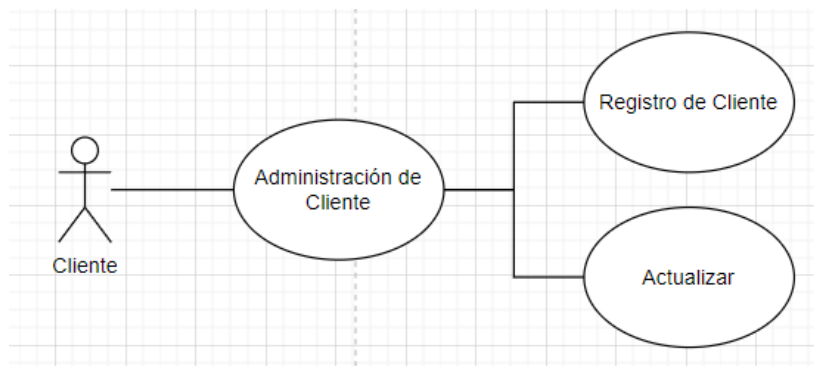


Gráfico 18. Caso de Uso: Administración de Cliente. Autor: Andrés Lozada, 2022

#### 3.4.2.2.1. Caso de uso: Registro de Cliente

**Descripción:** El actor podrá crear una nueva cuenta de cliente.

**Actores:** Cliente.

**Flujo Principal:**

1. El actor selecciona el apartado de Crear cuenta.
2. El sistema presenta 2 opciones: registro como veterinaria y como cliente.
3. El actor escoge Cliente.
4. El sistema carga el formulario respectivo para veterinaria.
5. El actor completa el formulario.
6. El actor presiona el botón Crear cuenta.
7. El sistema crea un nuevo Cliente.
8. El sistema inicia la sesión del cliente.

#### 3.4.2.2.2. Caso de uso: Actualizar cuenta

**Descripción:** El actor podrá actualizar su cuenta de cliente.

**Actores:** Cliente.

**Flujo Principal:**

1. El actor selecciona el apartado Perfil.
2. El actor selecciona Actualizar Datos.
3. El sistema carga el formulario con los datos existentes del cliente.
4. El actor modifica los datos.
5. El actor presiona el botón Guardar.
6. El sistema actualiza el registro.
7. El sistema muestra el mensaje de éxito o error respectivamente.

### 3.4.2.3. Visualización de sus citas

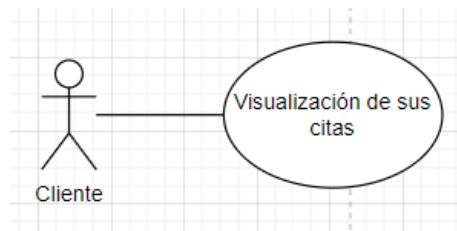


Gráfico 19. Caso de Uso: Visualización de sus citas. Autor: Andrés Lozada, 2022

**Descripción:** El actor puede visualizar sus citas.

**Actores:** Cliente

**Flujo Principal:**

1. El actor navega a Mis Citas
2. El sistema carga el listado de citas a manera de calendario.

### 3.4.2.4. Visualización de sus Mascotas

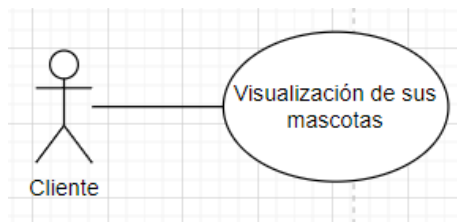


Gráfico 20. Caso de Uso: Visualización de sus mascotas. Autor: Andrés Lozada, 2022

**Descripción:** El actor puede visualizar sus mascotas.

**Actores:** Cliente

**Flujo Principal:**

1. El actor navega a Mis Mascotas
2. El sistema carga el listado de mascotas.

#### 3.4.2.5. Visualización del historial médico de sus mascotas

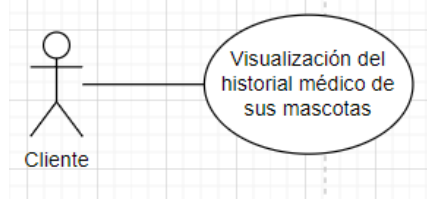


Gráfico 21. Caso de Uso: Visualización del historial médico de sus mascotas. Autor: Andrés Lozada, 2022

**Descripción:** El actor puede visualizar el historial médico de sus mascotas.

**Actores:** Cliente

**Flujo Principal:**

1. El actor navega a Mis Mascotas
2. El sistema carga el listado de mascotas.
3. El actor escoge la mascota deseada.
4. El sistema carga la página del historial médico de la mascota.
5. El sistema carga el listado de registros.

## CAPÍTULO IV: DESARROLLO DE LA APLICACIÓN

---

### 4.1. Estándares en el desarrollo de la aplicación

Para un correcto y organizado desarrollo de la aplicación, se han planteado una serie de estándares para la codificación, así como para la organización de componentes, módulos, directivos, etc.

#### 4.1.1. Estándares de codificación

Para la declaración de variables, métodos, clases e interfaces se mantiene como estándar el uso de camelCase, es decir, el inicio de cada palabra dentro de una variable o método se establece con mayúscula a excepción de la primera palabra únicamente para variables y métodos. En el siguiente ejemplo se evidencia su uso tanto para clases, variables y métodos.

```
export class EmployeesTableComponent implements OnInit {
  @Input() employeesDS: MatTableDataSource<Employee> = new MatTableDataSource()
  displayedColumns = ['name', 'ci', 'phone', 'address']

  @ViewChild(MatPaginator) paginator!: MatPaginator;
  @ViewChild(MatSort) sort!: MatSort;

  constructor(
    private dialog: MatDialog,
    private signupService: SignupService,
    private uiService: UiService,
    private storageService: StorageService
  ) {}

  ngOnInit(): void {
  }

  ngAfterViewInit(){
    this.employeesDS.paginator = this.paginator
    this.employeesDS.sort = this.sort
  }

  applyFilter(event: Event) {
    const filterValue = (event.target as HTMLInputElement).value;
    this.employeesDS.filter = filterValue.trim().toLowerCase();

    if (this.employeesDS.paginator) {
      this.employeesDS.paginator.firstPage();
    }
  }
}
```

Gráfico 22. Código de ejemplificación camelCase. Autor: Andrés Lozada, 2022

En el gráfico presentado se puede observar la clase “EmployeesTableComponent”, variables como “displayedColumns” y métodos como applyFilter que mantienen el estándar de uso de camelCase.

Cabe mencionar que para la codificación se utilizará como idioma el inglés, debido a que de esta forma se mantiene simple el nombre de todo tipo de estructuras, por lo tanto, variables, métodos, clases, interfaces e incluso entidades serán nombradas en inglés.

#### **4.1.2. Organización de componentes, directivos y módulos**

Para la organización de componentes se diferenciarán los mismos en dos tipos, componentes que sean reutilizables y componentes que sean páginas dentro de la aplicación web. Una vez realizada esta distinción, aquellos componentes que sean páginas del proyecto serán ubicados en la carpeta “pages” mientras que aquellos que sean reutilizables se ubicarán en la carpeta “components”. Cabe mencionar que cada uno de los componentes generados se mantendrán en su propia carpeta con su nombre y constarán con su plantilla HTML, su código fuente en Typescript y su hoja de estilos en SCSS.

Por otra parte, los directivos se mantendrán en su propia carpeta “directives” y se mantiene el módulo general generado por Angular donde se agregarán las declaraciones e importaciones respectivas para el correcto funcionamiento de la aplicación.

### **4.2. Sprints**

#### **4.2.1. Sprint 1**

##### *4.2.1.1. Sprint Planning*

La planificación para el primer sprint consiste en el desarrollo de módulos base relacionados con el establecimiento de conexiones a la base de datos, conexión entre front y backend y autenticación y registro de los usuarios, tanto veterinarias y sus empleados como dueños de mascotas.

#### 4.2.1.2. *Sprint Backlog*

<b>Backlog Item</b>	<b>Descripción</b>	<b>Usuario</b>	<b>Criterio de Aceptación</b>
Conexión con base de datos	Conexión del backend con la base de datos	-	Se establece una conexión del backend con la base de datos
Conexión entre frontend y backend	Conexión entre frontend y backend	-	Se establece una conexión entre frontend y backend a través de endpoints
Autenticación	Los usuarios podrán autenticarse en la aplicación	Todos	Los usuarios podrán iniciar sesión en la aplicación
Registro de usuarios	Los usuarios podrán crear su cuenta ya sea como veterinaria o como cliente	Todos	Los usuarios pueden crear su cuenta de veterinaria o de dueño de mascota

*Tabla 1. Sprint Backlog 1. Autor: Andrés Lozada, 2022*

#### 4.2.1.3. *Sprint Review*

Una vez terminado el desarrollo de los módulos establecidos, se realizó una revisión y pruebas de estos, en las cuales se verificó su correcto funcionamiento y se comparó entre lo planeado y el resultado. El resultado de esta revisión fue satisfactorio puesto que se cumplió con lo planificado.

#### 4.2.1.4. *Retroalimentación*

Al ser un proyecto desarrollado por una sola persona y no un equipo, es difícil obtener una retroalimentación que permita mejorar ciertos aspectos del desarrollo a nivel individual, por esta razón, la retroalimentación del sprint será incluida únicamente en sprints donde, al realizar la revisión respectiva, se distingan fallos o posibles mejoras en diseño de interfaces, escritura de código u otros.

## 4.2.2. Sprint 2

### 4.2.2.1. Sprint Planning

Para el segundo Sprint se planifica completar con el desarrollo de módulos para la administración de Veterinaria, donde se pueda visualizar la información de la veterinaria a la que un empleado pertenece y se pueda actualizar esta información. Además, se concluirá con el desarrollo del módulo de administración de empleados, el cual será vinculado con la administración de la veterinaria a nivel de la estructura de las páginas de la aplicación.

### 4.2.2.2. Sprint Backlog

Backlog Item	Descripción	Usuario	Criterio de Aceptación
Visualizar y modificar veterinaria	Visualizar y modificar datos de la veterinaria a la que pertenece el usuario	Administrador	Se pueden visualizar y actualizar los datos de la veterinaria a la que pertenece el usuario
Crear, visualizar, actualizar y eliminar empleados	CRUD de empleados dentro de la veterinaria	Administrador	Se puede realizar el CRUD completo de empleados dentro de la veterinaria

Tabla 2. Sprint Backlog 2. Autor: Andrés Lozada, 2022

### 4.2.2.3. Sprint Review

Al realizar la revisión y pruebas respectivas del sprint se corroboró el correcto funcionamiento de cada módulo; en estas pruebas se ejecutaron todas las tareas posibles de manera manual, verificando que se complete correctamente el flujo.

### 4.2.2.4. Retroalimentación

Mediante la revisión del sprint se determinaron ciertas falencias en la estructura de la página para la administración de la veterinaria, estas falencias serán incluidas para su corrección en el próximo sprint.

### 4.2.3. Sprint 3

#### 4.2.3.1. Sprint Planning

Para este sprint se planifica cumplir con el desarrollo de los módulos para la Administración de Clientes y la Administración de Mascotas, además, se ha incluido un item que consiste en la corrección de la pantalla de Administrar Veterinaria, falencia que se encontró en el sprint anterior.

#### 4.2.3.2. Sprint Backlog

Backlog Item	Descripción	Usuario	Criterio de Aceptación
Administración de Clientes	Crear, visualizar, editar y eliminar clientes	Administrador, Empleado	Un usuario Administrador o Empleado puede crear, actualizar, eliminar y consultar Clientes vinculados a su veterinaria
Administración de Mascotas	Crear, visualizar, editar y eliminar mascotas	Administrador, Empleado, Cliente	Se puede realizar el CRUD completo de la entidad Mascota
Corrección disposición en pantalla Administración de Veterinaria	Mejor organización de los elementos en la pantalla	-	Los elementos de la pantalla se encuentran más ordenados y existe mayor facilidad en la usabilidad

Tabla 3. Sprint Backlog 3. Autor: Andrés Lozada, 2022

#### 4.2.3.3. Sprint Review

Al culminar con el desarrollo se realizaron las pruebas respectivas en cada uno de los módulos y no se encontraron errores. Se cumplió con las expectativas del sprint, pues todo lo planificado se terminó a tiempo.

#### 4.2.3.4. Retroalimentación

No existe una retroalimentación específica para este sprint puesto que se cumplió con lo planificado correctamente.

#### 4.2.4. Sprint 4

##### 4.2.4.1. Sprint Planning

Para el cuarto sprint se desarrollarán módulos que implican la gestión de historiales de mascotas, y la visualización de citas. Esto implica diferentes módulos para Administradores y Empleados, y para Clientes, puesto que los usuarios de tipo cliente solo pueden visualizar la información en ambos módulos, a diferencia de los otros 2 usuarios que pueden realizar la administración completa.

##### 4.2.4.2. Sprint Backlog

Backlog Item	Descripción	Usuario	Criterio de Aceptación
Administración de historiales de mascotas	Crear, visualizar y modificar datos de historiales de mascotas	Administrador, Empleado	Se pueden crear, visualizar y actualizar los datos de historiales médicos de las mascotas
Visualizar historiales de mascotas	Visualizar historiales de mascotas	Cliente	Se pueden visualizar los historiales de mascotas
Administración de citas	Crear, visualizar, actualizar y eliminar citas	Administrador, Empleado	Se puede realizar el CRUD completo de citas dentro de la veterinaria
Visualizar citas	Visualizar citas	Cliente	Se puede visualizar citas

Tabla 4. Sprint Backlog 4. Autor: Andrés Lozada, 2022

##### 4.2.4.3. Sprint Review

Mediante la revisión respectiva del sprint se comprobó que los módulos funcionen correctamente a través de las pruebas manuales con diferentes datos.

##### 4.2.4.4. Retroalimentación

Para este sprint no existieron problemas, todo se desarrolló acorde a lo planificado.

### 4.3. Muestras de la Aplicación

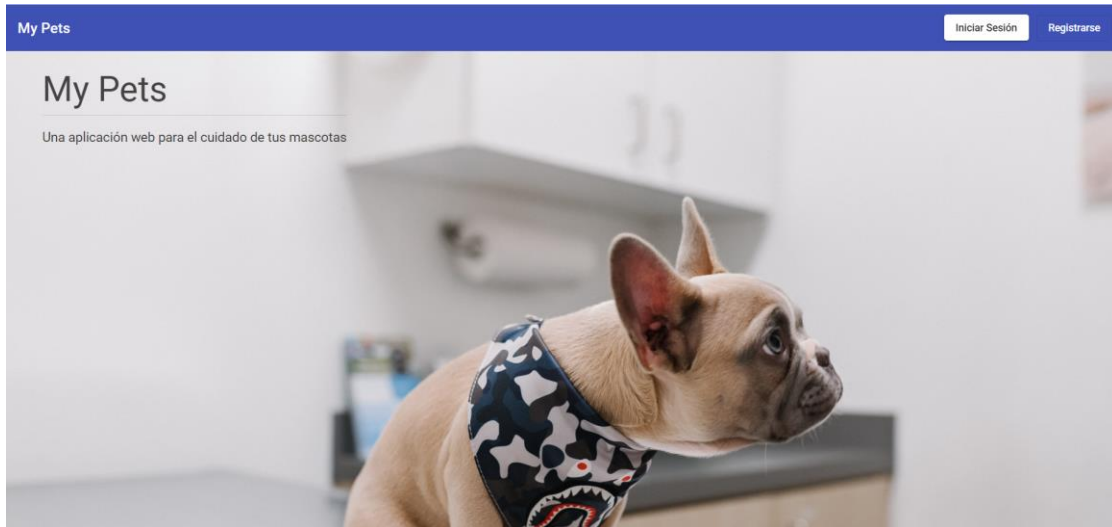


Gráfico 23. Página de Inicio. Autor: Andrés Lozada, 2022

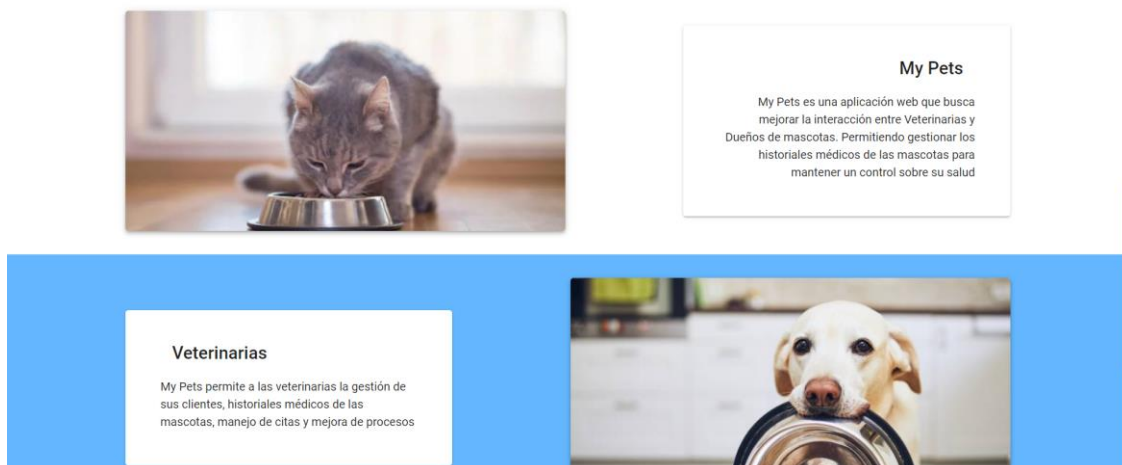


Gráfico 24. Página de inicio, parte inferior. Autor: Andrés Lozada, 2022

En los gráficos 20 y 21, se puede observar la pantalla de inicio, esta contiene una barra de navegación para poder dirigirse a Iniciar Sesión o a Registrarse. Además, el contenido de la página incluye breves detalles sobre la aplicación web del proyecto presente.

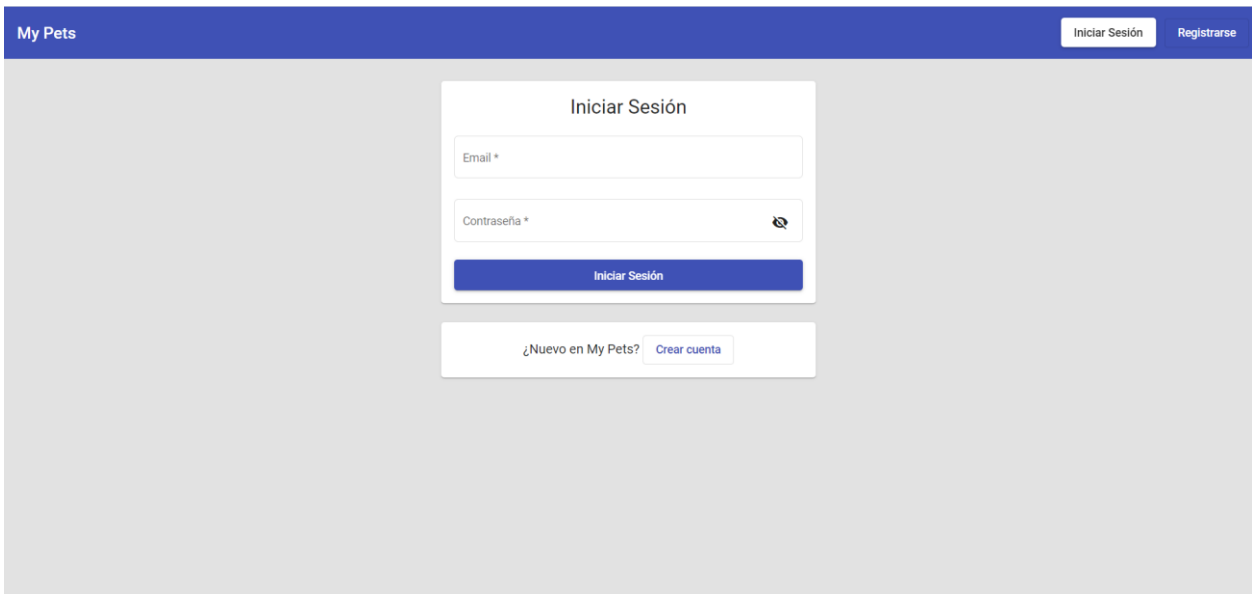


Gráfico 25. Inicio de Sesión en la aplicación web. Autor: Andrés Lozada, 2022

En el gráfico 22 se observa la pantalla de Iniciar Sesión, donde se accede a la aplicación web a través de las credenciales, es decir el email y contraseña del usuario. Además, existe una indicación para redirigir a Registrarse en caso de que el usuario no tenga una cuenta.

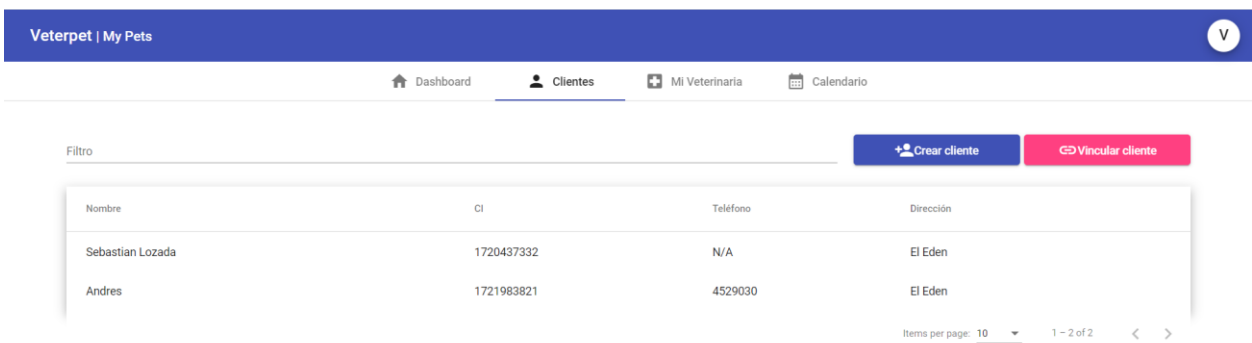


Gráfico 26. Administración de clientes en página web. Autor: Andrés Lozada, 2022

En el gráfico 23 se presenta la pantalla de Clientes de la aplicación. Para acceder a esta página es necesario que se autentique el usuario y que este sea o bien un administrador de la

veterinaria o un empleado. En esta página se cuenta con las acciones para crear un nuevo cliente, vincular uno a la veterinaria (funcionalidad aún no desarrollada) y se muestra la lista completa de clientes en una tabla con su respectiva información.

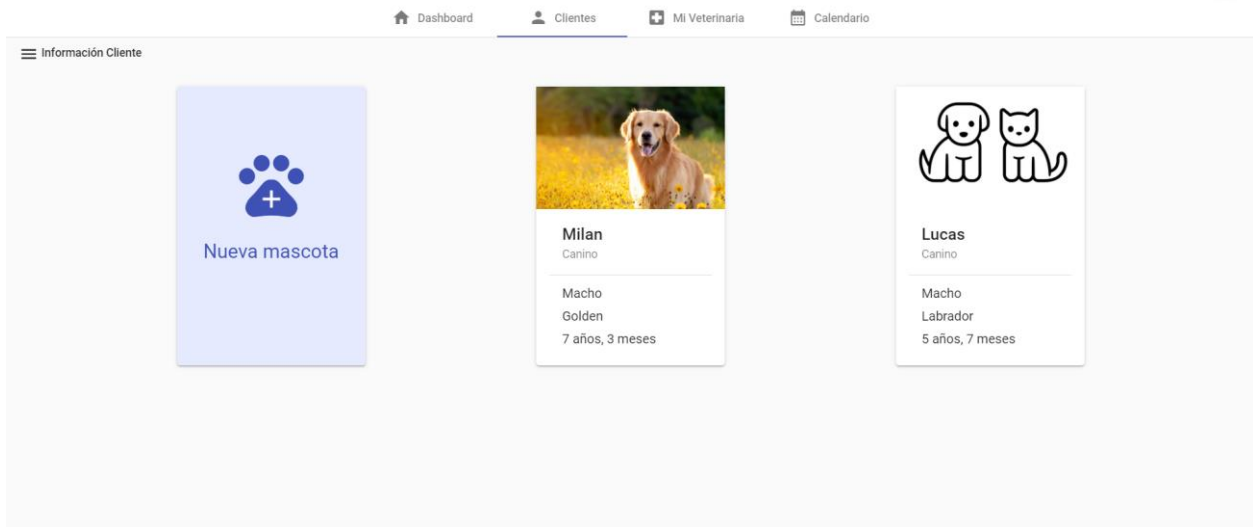


Gráfico 27. Administración de mascotas en página web. Autor: Andrés Lozada, 2022

En el gráfico 24 se tiene la pantalla de mascotas, en este se presentan mediante tarjetas información esencial de las mascotas. Además, se incluyen las acciones para crear una mascota, editar y eliminar las existentes. También se presenta un menú desplegable con la información del cliente.

The screenshot shows a web application interface for managing a pet's history. On the left, a dark sidebar displays the pet's name 'Milan' and details: Especie: Canino, Raza: Golden, Sexo: Macho, Color: Amarillo, Edad: 7 años, 3 meses, and the owner's name 'Sebastian Lozada' with contact information. The top navigation bar includes 'Dashboard', 'Clientes', 'Mi Veterinaria', and 'Calendario'. Below this is a dashboard with seven action buttons: 'Nueva Cita', 'Nuevo Diagnóstico', 'Nuevo Control', 'Nueva Consulta', 'Nueva Vacuna', 'Nueva Desparasitación', and 'Historial'. The main content area features a section titled 'Próximas Citas' with a table of upcoming appointments.









Fecha	Hora	Descripción	Doctores / Empleados	Acciones
17-02-2022	06:30 PM	Vacuna tos		 
24-02-2022	03:30 PM	Vacuna rabia		 
25-02-2022	03:20 PM	control de peso		 
26-02-2022	12:00 PM	test	Veterpet	 

Gráfico 28. Administración del historial de mascota en página web. Autor: Andrés Lozada, 2022

En el gráfico 25 se tiene la pantalla de gestión de historial de una mascota. En esta pantalla se presenta un menú desplegable con la información de la mascota y del dueño, además, se tiene acceso directo a la tabla con la información sobre próximas citas y se pueden realizar las siguientes acciones: Crear nueva cita, nuevo diagnóstico, nuevo control, nueva consulta, nueva vacuna, nueva desparasitación y acceder al historial de la mascota.

## CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

---

### 5.1. Conclusiones

El uso de la metodología ágil, implantada a través del marco de trabajo Scrum, permite un rápido desarrollo de la aplicación web, puesto que establece metas en el lapso de un sprint, el cual puede ser de 1 o 2 semanas, de manera que se completen tareas o módulos dentro de este tiempo, se realizan las evaluaciones respectivas de estas y se avanza rápidamente al próximo sprint, reduciendo pérdidas de tiempo en la creación de mucha documentación.

La aplicación desarrollada contribuye para la ejecución de procesos dentro de las veterinarias, puesto que automatiza la gestión de su información, la de sus clientes y mascotas; también facilita la comunicación con los clientes sobre sus próximas citas y a los dueños de mascotas les permite mantener un control sobre la salud de su mascota sin necesidad de un carné físico.

Trabajar con una arquitectura cliente - servidor a través del uso de API's facilita la comunicación entre ambas partes e incluso permite que un Frontend o cliente se comunique con cualquier Backend o servidor, esto implica una gran flexibilidad en el desarrollo puesto que se pueden incluir fácilmente nuevos módulos que impliquen el uso de recursos externos, los cuales son consumidos a través de dichas API's.

### 5.2. Recomendaciones

Se recomienda el uso de sistemas gestores de contenido o CMS para el desarrollo de proyectos pequeños y medianos, aunque estos puedan ser implementados para cualquier tipo de proyecto, puesto que ofrecen herramientas que facilitan un inicio rápido pero pueden llegar a tener limitaciones en comparación a otros métodos para la gestión de contenido y datos.

Para el desarrollo de aplicaciones web se recomienda el uso de API's para poder mantener una arquitectura cliente – servidor y que incluso se pueda consumir recursos externos que sean de gran utilidad acorde a las necesidades del proyecto.

Para el desarrollo de software se recomienda mantener estándares en la codificación y organización de archivos, puesto que de esta manera se facilita el entendimiento para todos los miembros del equipo o incluso otra persona puede continuar con el desarrollo evitando confusiones que pueden conllevar errores en la programación.

## ANEXOS

---

### Acta de Aceptación del Software

Quito, 20 de septiembre de 2022

Yo, Dr. Carlos Martínez con C.C. 1718721341, Director Médico y Gerente de Hospital Veterinario VETERPET, doy la aceptación de la aplicación web para veterinarias desarrollada por Andrés Nicolás Lozada Carrera, C.C. 1721983821, y confirmo que cumple con los requerimientos planteados para el trabajo de titulación.

Atentamente,

  
  
**Dr. Carlos Martínez**  
Reg. Senecyt  
1005-13-1189684

Dr. Carlos Martínez

Director Médico - Gerente

## BIBLIOGRFÍA

---

- Zuñiga, L. (2020). Desarrollo de aplicaciones web utilizando Angular como framework. Mazatlán, México: Universidad Politécnica De Sinaloa.
- Bierman, G., Abadi, M., Torgersen, M. (2014). Understanding TypeScript. In: Jones, R. (eds) ECOOP 2014 – Object-Oriented Programming. ECOOP 2014. Lecture Notes in Computer Science, vol 8586. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-662-44202-9\\_11](https://doi.org/10.1007/978-3-662-44202-9_11)
- Angular. (2022). Introduction to Angular concepts. Angular. <https://angular.io/guide/architecture>
- The PostgreSQL Global Development Group. (2022). What is PostgreSQL?. PostgreSQL. <https://www.postgresql.org/about/>
- Subra, J. P., & Vannieuwenhuysse, A. (2018). Scrum: un método ágil para sus proyectos. Ediciones ENI.
- Vacacela, E., Tenecota, J., Torres, J., & Celleri, J. (2018). Automatización de procesos de Investigación, Vinculación, Prácticas/Pasantías Preprofesionales para universidades ecuatorianas. Dialnet, 35-44.
- Vicuña, A., & Caranqui, Á. (2018). Automatización del proceso de salud ocupacional del distrito 17D04" Puengasí a Itchimbía". Quito, Ecuador: Universidad Politécnica Salesiana.
- Vaca, D. Villacreces, B. (2021). Análisis, diseño, desarrollo e implementación de un sistema de automatización. Quito, Ecuador: Pontificia Universidad Católica del Ecuador.