



Universidad de Valladolid



Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Computación

Aplicación para la gestión de una Clínica Veterinaria

Autor: Paula Higuera Consuegra



Universidad de Valladolid



Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Computación

Aplicación para la gestión de una Clínica Veterinaria

Autor: Paula Higuera Consuegra

Tutor: Valentín Cardeñoso Payo

A mi familia, por su apoyo incondicional. Ha sido mi mayor fortaleza. Gracias

Agradecimientos

Este proyecto no habría sido posible sin el apoyo y la ayuda de muchas personas a lo largo de este proceso.

En primer lugar, me gustaría agradecer a mi tutor, Valentín, por su orientación, paciencia y conocimientos que me han guiado durante todo el desarrollo de este proyecto.

También quiero expresar mi gratitud a mi familia, por su constante apoyo emocional y motivación. Su presencia me ha dado fortaleza en los momentos más difíciles.

Extiendo un agradecimiento especial a los veterinarios que me han brindado su experiencia y conocimiento. Sus aportaciones han sido esenciales para el éxito de este proyecto, y su dedicación a la salud y bienestar de los animales es admirable.

Mi reconocimiento también a mi amigo Daniel Lizano que ha compartido su tiempo y consejos conmigo. Su ayuda ha sido fundamental para mantenerme motivada y enfocada.

Por último, pero no menos importante, quiero agradecer a todas las personas que, de una u otra forma, han contribuido a la realización de este trabajo, ya sea brindando apoyo moral, logístico o académico. Su ayuda ha sido invaluable, y estoy profundamente agradecida por ello.

Gracias.

Resumen

Una parte esencial de una clínica veterinaria es la gestión de la misma, la cual suele consistir en un conjunto complejo de tareas administrativas que requieren una buena organización y eficiencia, las cuales abarcan desde la programación de citas hasta el seguimiento de historiales clínicos, la facturación y la comunicación efectiva con los clientes. La aplicación se concibe como una herramienta centralizada que simplifica estas operaciones, permitiendo un acceso rápido y seguro a la información relevante. Estas tareas pueden ser realizadas por métodos establecidos por los veterinarios o recurriendo a diferentes herramientas especializadas. Para cualquiera de los casos, es necesario que la clínica disponga de una aplicación para facilitar este trabajo a los veterinarios. El objetivo de este proyecto es el desarrollo de una aplicación web para la facilitación de la gestión de citas, historiales, y recursos, de manera integrada, visual y eficiente para las clínicas veterinarias. Esta aplicación se concibe como una herramienta centralizada respondiendo a la necesidad de simplificar las complejas tareas de administración esenciales en la gestión de citas, historiales y recursos en el entorno veterinario.

Ya sea que los veterinarios utilicen métodos tradicionales o herramientas especializadas, la aplicación web se posicionara como una solución esencial para mejorar la fluidez de estas tareas. El principio fundamental es que proporcione un acceso rápido y seguro a la información relevante, permitiendo a los veterinarios optimizar su tiempo y recursos.

Abstract

An essential part of a veterinary clinic is clinic management, which usually consists of a complex set of administrative tasks that require good organization and efficiency, ranging from scheduling appointments to tracking medical records, invoicing and effective communication with clients. The application is conceived as a centralized tool that simplifies these operations, allowing fast and secure access to relevant information. These tasks can be performed by methods established by veterinarians or by resorting to different specialized tools. For either case, it is necessary for the clinic to have an application to facilitate this work for veterinarians. The objective of this project is the development of a web application to facilitate the management of appointments, records, and resources in an integrated, visual and efficient way for veterinary clinics. This application is conceived as a centralized tool responding to the need to simplify the complex administration tasks essential in the management of appointments, records and resources in the veterinary environment.

Whether veterinarians use traditional methods or specialized tools, the web application will position itself as an essential solution to improve the fluidity of these tasks. The fundamental principle is that it provides fast and secure access to the information related to the veterinarian's practice.

Índice general

Índice de cuadros	VI
Índice de figuras	VIII
1. Introducción	1
1.1. Introducción	1
1.2. Motivación	1
1.3. Objetivos	1
2. Metodología Y Planificación	5
2.1. Agile	5
2.2. Etapas metodología Agile	5
2.3. Plan de trabajo	6
2.4. Estimación de costes	7
2.4.1. Costes durante el proyecto	7
2.4.2. Costes globales	8
2.5. Riesgos del proyecto	8
2.5.1. Concepto de riesgo	8
2.5.2. Riesgos identificados	8
2.5.3. Riesgo total	9
2.6. Riesgos identificados	9
3. Marco Conceptual	13
3.1. Servicios Veterinarios	13
3.1.1. Consultas Generales	13
3.1.2. Vacunaciones	13
3.1.3. Servicio de desparasitación	14
3.1.4. Cirugías	14
3.1.5. Servicio de eutanasia	15
3.1.6. Atención de emergencia	15
3.2. Gestión de Historiales	15
3.2.1. Registro detallado	15
3.2.2. Organización y almacenamiento	15
3.2.3. Seguridad y confidencialidad	16
3.3. Gestión Personal	16
3.3.1. Veterinarios	16
3.3.2. Asistentes y Técnicos Veterinarios (ATV)	16
3.3.3. Personal de recepción y administrativo	16
3.3.4. Personal de limpieza y mantenimiento	16
3.3.5. Otros roles especializados	16

3.4.	Gestión de clientes	17
3.4.1.	Atención al cliente	17
3.4.2.	Gestión de citas	17
3.4.3.	Seguimiento post-consulta	17
3.5.	Equipamiento	18
3.6.	Gestión de inventario	19
3.6.1.	Registro y seguimiento	19
3.6.2.	Control de stock	19
3.6.3.	Organización y almacenamiento	19
4.	Soluciones Existentes	21
5.	Análisis	23
5.1.	Requisitos funcionales	23
5.2.	Requisitos no funcionales	26
5.3.	Requisitos funcionales de información	26
5.4.	Casos de uso	27
5.5.	Actores del sistema	50
5.6.	Modelo de clases	51
5.6.1.	Gestión de Usuarios	52
5.6.2.	Gestión de Mascotas	52
5.6.3.	Gestión de Recursos	53
6.	Diseño	55
6.1.	Diseño	55
6.1.1.	Front-end (Angular)	55
6.1.2.	Back-end (Spring Boot)	56
6.1.3.	Comunicación entre Front-end y Back-end	56
6.1.4.	Gestión de Seguridad	56
6.2.	Diseño Interfaz	57
6.2.1.	Interfaz de Portada	57
6.2.2.	Interfaz de acceso al sistema	58
6.2.3.	Interfaz de calendario	60
6.2.4.	Interfaz común	61
7.	Implementación	63
7.1.	Herramientas de Desarrollo	63
8.	Pruebas	65
9.	Conclusiones	67
9.1.	Trabajo futuro	68
	Appendices	69
	Apéndice A. Manual de Instalación	71
	Apéndice B. Manual de Usuario	73
B.1.	Contenido de la página de inicio de la clínica	74
B.2.	Rangos de Permiso y Opciones Disponibles	75
B.2.1.	Rango de Permiso: Cliente	75
B.2.2.	Rango de Permiso: Facultativo	79

B.2.3. Rango de Permiso: Administrador	90
Apéndice C. Manual del Desarrollador	95
C.1. Estructura de carpetas back-end	95
C.2. Estructura de carpetas front-end	97
C.2.1. Instalación de Node.js	97
C.2.2. Instalación de Angular CLI	98
Apéndice D. Enlaces del proyecto	99

Índice de cuadros

2.1. Plan Inicial de Trabajo para Desarrollo de Aplicación Web	6
2.2. Costos Estimados por Personal del Proyecto (en euros)	7
2.3. Costos Estimados de Materiales del Proyecto (en euros)	7
2.4. Riesgo: Desconocimiento de la materia.	9
2.5. Riesgo: Trabajo simultaneo.	10
2.6. Riesgo: Caída de servidor.	10
2.7. Riesgo: Cambio de requisitos.	11
2.8. Riesgo: Falta de implicación de los interesados.	11
2.9. Riesgo: Problemas de gestión de tiempo.	12
2.10. Riesgo: Riesgos externos imprevisibles.	12
5.1. Requisitos funcionales RF-001	23
5.2. Requisitos funcionales RF-002	23
5.3. Requisitos funcionales RF-003	23
5.4. Requisitos funcionales RF-004	24
5.5. Requisitos funcionales RF-005	24
5.6. Requisitos funcionales RF-006	24
5.7. Requisitos funcionales RF-007	24
5.8. Requisitos funcionales RF-008	24
5.9. Requisitos funcionales RF-009	24
5.10. Requisitos funcionales RF-010	24
5.11. Requisitos funcionales RF-011	24
5.12. Requisitos funcionales RF-012	25
5.13. Requisitos funcionales RF-013	25
5.14. Requisitos funcionales RF-014	25
5.15. Requisitos funcionales RF-015	25
5.16. Requisitos funcionales RF-016	25
5.17. Requisitos funcionales RF-017	25
5.18. Requisitos funcionales RF-018	25
5.19. Requisitos funcionales RF-019	25
5.20. Requisitos no funcionales RNF-001	26
5.21. Requisitos no funcionales RNF-002	26
5.22. Requisitos no funcionales RNF-003	26
5.23. Requisitos no funcionales RNF-004	26
5.24. Requisitos no funcionales RNF-005	26
5.25. Requisitos no funcionales RNF-002	26
5.26. Use Case: Adjuntar archivos	28
5.27. Use Case: Anular Cita	29
5.28. Use Case: Añadir Farmaco	29
5.29. Use Case: Añadir Historial	30

5.30. Use Case: Añadir Historial Desparasitación	30
5.31. Use Case: Añadir Intervención	31
5.32. Use Case: Añadir Material	31
5.33. Use Case: Añadir Tratamiento	32
5.34. Use Case: Añadir Vacunación	32
5.35. Use Case: Añadir cita	33
5.36. Use Case: Añadir consulta	33
5.37. Use Case: Añadir nota	34
5.38. Use Case: Cerrar Sesión	34
5.39. Use Case: Consultar Calendario	35
5.40. Use Case: Consultar Facultativos	35
5.41. Use Case: Consultar Farmacos	36
5.42. Use Case: Consultar Informe	36
5.43. Use Case: Consultar Intervención	37
5.44. Use Case: Consultar Mascotas	37
5.45. Use Case: Consultar Material	38
5.46. Use Case: Consultar Notificaciones	38
5.47. Use Case: Consultar Perfil	38
5.48. Use Case: Consultar Tratamiento	39
5.49. Use Case: Consultar historial	39
5.50. Use Case: Consultar las Consultas	40
5.51. Use Case: Consultar Proveedor	40
5.52. Use Case: Dar de Baja Facultativo	41
5.53. Use Case: Dar de Baja Cliente	41
5.54. Use Case: Dar de Baja Mascota	42
5.55. Use Case: Dar de Baja Proveedor	42
5.56. Use Case: Editar Consulta	43
5.57. Use Case: Editar Informe	43
5.58. Use Case: Editar Intervención	44
5.59. Use Case: Editar Perfil	44
5.60. Use Case: Editar Tratamiento	45
5.61. Use Case: Editar historial	46
5.62. Use Case: Eliminar Farmaco	46
5.63. Use Case: Eliminar Material	47
5.64. Use Case: Iniciar Sesión	47
5.65. Use Case: Registrar Facultativo	48
5.66. Use Case: Registrar Mascota	48
5.67. Use Case: Registrarse	49
5.68. AC-001 Usuario	50
5.69. AC-002 Cliente	50
5.70. AC-003 Facultativo	50
5.71. AC-004 Administrador	50

Índice de figuras

2.1. Tabla de riesgo probabilidad e impacto	9
5.1. Diagrama de casos de uso	27
5.2. Diagrama de clases	51
6.1. Interfaz de Portada	57
6.2. Interfaz de acceso	58
6.3. Interfaz de registro	59
6.4. Interfaz de calendario	60
6.5. Interfaz de acceso	61
6.6. Interfaz de registro	61
B.1. Vista inicial	73
B.2. Vista inicial	73
B.3. Vista login	74
B.4. Vista registro	74
B.5. Interfaz de cita cliente	75
B.6. Interfaz de calendario cliente	76
B.7. Interfaz de mascotas cliente	77
B.8. Interfaz de mascotas al detalle	77
B.9. Interfaz de perfil del cliente	78
B.10. Interfaz de edición perfil del cliente	78
B.11. Interfaz de cita facultativo	79
B.12. Interfaz de calendario facultativo	80
B.13. Interfaz de clientes facultativo	81
B.14. Interfaz de cliente en detalle facultativo	81
B.15. Interfaz de estadísticas de las mascotas	82
B.16. Interfaz de búsqueda de mascotas	83
B.17. Interfaz de detalle de mascotas	83
B.18. Interfaz de historial de mascota	84
B.19. Interfaz de consultas	84
B.20. Interfaz de detalles de consulta	85
B.21. Interfaz de material de consulta	85
B.22. Interfaz de informe de consulta	86
B.23. Interfaz de intervención de consulta	86
B.24. Interfaz de tratamiento de consulta	87
B.25. Interfaz de tratamiento	87
B.26. Interfaz de tratamiento medicación	87
B.27. Interfaz de vacunación de mascota	88
B.28. Interfaz de desaparición de mascota	88

B.29. Interfaz de perfil del facultativo	89
B.30. Interfaz de edición perfil del facultativo	89
B.31. Interfaz de consulta de facultativos	90
B.32. Interfaz de datos de facultativos	90
B.33. Interfaz de datos de materiales	91
B.34. Interfaz de datos de materiales (2)	91
B.35. Interfaz de datos de materiales	92
B.36. Interfaz de datos de fármacos	92
B.37. Interfaz de datos de fármacos	92
B.38. Interfaz de datos de fármacos	93
B.39. Interfaz de datos de proveedor	93
B.40. Interfaz de datos de proveedor	93
B.41. Interfaz de datos de proveedor	94

Introducción

1.1 Introducción

En la actualidad la sociedad cada vez esta más comprometida con el bienestar de los animales de compañía, la atención veterinaria se ha convertido en un pilar fundamental para garantizar la salud y el cuidado de las mascotas. El aumento de la demanda de estos servicios ha llevado a una evolución de la forma en la cual se abordan los desafíos de la gestión clínica, el seguimiento de pacientes y la comunicación con los propietarios. Sin embargo, a pesar de los avances en la medicina veterinaria, la gestión de las clínicas no a seguido el mismo ritmo de transformación. Este proyecto se centrara en abordar esta cuestión de manera eficiente y efectiva. La necesidad de una solución integral que contenga la programación de citas, el seguimiento de los historiales veterinarios, la comunicación con los propietarios y otras funciones esenciales, nos ha llevado a la creación de una aplicación de gestión veterinaria. Esta aplicación no solo busca optimizar el trabajo de los profesionales, sino también mejorar la experiencia de los clientes.

1.2 Motivación

Es necesario que la gestión en cualquier ámbito de la salud sea eficiente y efectiva, para proporcionar un servicio adecuado y mejorar la calidad de vida de los pacientes. En el sector veterinario esta premisa cobra especial relevancia, ya que no solo esta en juego la salud de los animales, también la satisfacción de sus dueños. La gestión veterinaria moderna se enfrenta a nuevos desafíos que requieren soluciones y tecnológicas innovadoras, y es precisamente esta necesidad lo que ha impulsado la creación de esta aplicación de gestión veterinaria. La razón fundamental tras este proyecto es la demanda de una herramienta que abarque todas las funciones clave para la gestión de una clínica veterinaria en una sola herramienta, pues en un mundo cada vez más interconectado, resulta esencial que estos profesionales cuenten con herramientas eficientes para facilitarles poder brindar un servicio de excelencia.

1.3 Objetivos

1. Análisis de recursos:

- Medicamentos: Se debe analizar el inventario de medicamentos disponibles en la clínica veterinaria, incluyendo tanto los medicamentos de uso común o específicos para tratamientos especializados.
- Suministros quirúrgicos: Se debe evaluar la disponibilidad de suministros quirúrgicos esenciales, como gasas, suturas, material de curación, instrumental quirúrgico, etc.

- Alimentos y suplementos: Se debe revisar el stock de alimentos para mascotas, así como de suplementos dietéticos recomendados para ciertas condiciones de salud o etapas de la vida de las mascotas.

2. Gestión :

- Gestión de citas: Permitir a los clientes programar fácilmente citas para sus mascotas y a los facultativos gestionar su agenda de manera eficiente.
- Gestión de clientes: Incluir la asignación de cita y la edición de sus datos de contacto.
- Gestión de facultativos: Incluir la asignación de citas y la supervisión de la carga de trabajo de cada profesional.
- Gestión de mascotas: Permitir registrar información básica como especie, raza, edad y nombre, así como detalles más específicos sobre vacunas, alergias y condiciones de salud.
- Gestión de historiales: Permitir el almacenamiento seguro y la gestión eficaz de los historiales de las mascotas. Incluyendo información detallada como diagnósticos previos, tratamientos realizados, vacunas administradas, alergias, y cualquier otra nota relevante para la salud de la mascota
- Gestión de consultas: Facilitar la realización y el registro de consultas veterinarias, permitiendo a los facultativos tomar notas, registrar tratamientos e intervenciones de manera eficiente. Además, ofrece herramientas para el seguimiento del progreso del paciente y la programación de consultas de seguimiento si es necesario
- Gestión del inventario: Ofrecer un sistema completo para la gestión de inventario de la clínica veterinaria, incluyendo medicamentos, suministros quirúrgicos, alimentos, productos de cuidado y otros suministros esenciales. Permite realizar un seguimiento en tiempo real del stock disponible y registrar movimientos de inventario.

3. Centralización de datos:

- Centralización de clientes: Proporcionar una base de datos centralizada de clientes, lo que facilita el acceso rápido a la información relevante, como datos de contacto, historiales médicos de mascotas y preferencias de tratamiento.
- Centralización de historias clínicas: Diseñar y desarrollar un módulo integral para la gestión de historias clínicas digitales, facilitando el acceso, la actualización y el seguimiento de la información médica de los pacientes.
- Centralización de inventario: Establecer un sistema centralizado de gestión de inventario que permita un seguimiento en tiempo real de los productos y suministros de la clínica veterinaria. Esto incluirá la disponibilidad de medicamentos, alimentos para mascotas, productos de higiene y cualquier otro artículo relacionado con el funcionamiento de la clínica.
- Centralización de consultas: Implementar un sistema centralizado de gestión de consultas que permita a los facultativos programar y registrar consultas de manera eficiente. Este sistema incluirá funciones para programar citas, registrar información del paciente durante la consulta, generar notas clínicas, y realizar un seguimiento del progreso del paciente a lo largo del tiempo.

4. Registro:

- Registro de clientes: Establecer mecanismo de registro de clientes en la base de datos centralizada, donde se recopilarán y almacenarán los datos de contacto.
- Registro de facultativos: Establecer un proceso por el cual el administrador de la aplicación pueda registrar a los facultativos en la base de datos centralizada, donde se recopilaran y almacenaran sus datos de contacto, cargo y cualquier otra información relevante.

- Registro de historiales clínicos: Implementar un sistema de registro y almacenamiento de historias clínicas de manera organizada y segura. Esto incluye la creación de registros para cada paciente, donde se registrarán diagnósticos, tratamientos, resultados de exámenes, notas de consulta y cualquier otra información médica relevante.
- Registro de inventario: Implementar de un registro detallado de todo el inventario de la clínica veterinaria, incluyendo medicamentos, suministros quirúrgicos, alimentos para mascotas y otros productos. Se registrarán datos como la cantidad disponible, fecha de caducidad, proveedor y cualquier movimiento de inventario, como compras o ventas.
- Registro de consultas: Implementar un mecanismo para registrar todas las consultas realizadas en la clínica veterinaria, incluyendo información sobre el paciente, el facultativo a cargo, los tratamientos realizados y cualquier seguimiento necesario.

Metodología Y Planificación

Todo proyecto requiere un método el cual seguir, pues sin el podemos divagar a la hora de su realización. Esto se debe a que un proyecto, necesita ser gestionado de manera organizada y estructurada para alcanzar sus objetivos de manera eficiente. Un método proporciona un marco de trabajo que define las etapas, actividades y responsabilidades que se necesitan para llevar a cabo el proyecto con éxito. Esto incluye desde la planificación inicial hasta la ejecución, el seguimiento y el cierre del proyecto. Al seguir un método, se establecen expectativas claras, se minimizan los riesgos y se maximiza la eficiencia.

2.1 Agile

Para el ámbito de desarrollo de una aplicación web la capacidad de adaptación a los cambios y la satisfacción de las necesidades cambiantes de los usuarios es esencial para el éxito del proyecto. En este contexto, la metodología Agile ha surgido como un enfoque fundamental para gestionar proyectos de desarrollo de aplicaciones web de manera eficiente y efectiva. Agile proporciona un marco de trabajo flexible y colaborativo que se adapta perfectamente a la naturaleza dinámica y evolutiva del desarrollo de aplicaciones web [11].

2.2 Etapas metodología Agile

El desarrollo ágil de software se basa en siete pasos fundamentales, que cubren la totalidad del ciclo de vida del programa [15]:

- **Planificación:** En esta etapa inicial, se organiza el material entregado por parte del Cliente, se identifican las necesidades, se asignan los recursos internos de la empresa, se definen las bases del proyecto, se elabora un plan detallado que incluye la definición de alcance, la asignación de recursos, la creación de un cronograma y la identificación de riesgos potenciales.
- **Diseño:** Estudio inicial sobre el proyecto y su entorno. Este estudio inicial incluye un análisis de keywords a tener en cuenta para la estructura. Partiendo de estos estudios, se procede a dar una propuesta de diseño y estructura, y se espera a la aprobación por parte del Cliente.
- **Desarrollo:** Una vez pasada la etapa anterior dará comienzo la maquetación y programación del proyecto donde introduciremos todo el contenido.
- **Pruebas:** Seguiremos con la comprobación de la funcionalidad de lo que se ha desarrollado, y si se encuentra alguna funcionalidad que funciona incorrectamente se anotará para su resolución en la siguiente iteración.

- Despliegue: Esta conlleva al lanzamiento de la versión que se acaba de desarrollar para que pueda ser probada por el público, para poder ejecutar de forma correcta la siguiente etapa.
- Verificación: Esta es la última etapa del ciclo repetitivo, se revisa lo que se realizó durante la iteración y se hacen sugerencias, en el caso de que las hubiera se anotan para la siguiente iteración, en el caso contrario, se pasará a la última etapa.
- Lanzamiento: Se oficializa el proyecto, junto a su documentación para su posterior uso.

2.3 Plan de trabajo

A continuación se detallarán las horas trabajadas por fases que ha llevado realizar el proyecto, llegando a un total de 580 horas durante estos meses:

Fase del Proyecto	Descripción	Horas
Investigación preliminar	Revisión de requisitos del cliente, análisis de la competencia, estudio de mercado	25
Planificación	Planificación del proyecto	25
Diseño de la interfaz de usuario	Creación de bocetos y prototipos de la interfaz de usuario, definición de la arquitectura de navegación	50
Desarrollo de JWT y roles		100
Desarrollo del facultativo	Desarrollo de las funcionalidades, interfaces y tablas de BBDD del rol facultativo, con sus respectivas pruebas	100
Desarrollo del cliente	Desarrollo de las funcionalidades, interfaces y tablas de BBDD del rol cliente, con sus respectivas pruebas	100
Desarrollo del administrador	Desarrollo de las funcionalidades, interfaces y tablas de BBDD del rol administrador, con sus respectivas pruebas	100
Pruebas de rendimiento y seguridad	Evaluación del rendimiento y seguridad de la aplicación, optimización del rendimiento si es necesario	25
Pruebas generales	Pruebas de el funcionamiento global de la aplicación	25
Documentación y despliegue	Preparación de documentación para usuarios y desarrolladores, despliegue de la aplicación en el entorno de producción	25
Pruebas realizadas por usuarios	Pruebas de usabilidad y aceptación realizadas por usuarios finales	5

Cuadro 2.1: Plan Inicial de Trabajo para Desarrollo de Aplicación Web

2.4 Estimación de costes

A pesar de que este proyecto se ha llevado a cabo en un contexto no comercial, debemos estimar su valor, según el personal y materiales empleados para su desarrollo. A continuación, se detalla una estimación de los costos asociados con cada aspecto del proyecto:

2.4.1 Costes durante el proyecto

Estos son los costos estimados durante la planificación y desarrollo del proyecto:

Costes del personal [20]

Titulo	Costo por Hora (€)	Horas Estimadas	Costo Total Estimado (€)
Desarrollador Senior	19,56	325	6.357
Desarrollador Junior	17,05	300	5.115
Diseñador UX/UI	17,50	50	875
Gestor de Proyecto	23,79	350	8.326,5
Costo Total del Personal			14.322,857

Cuadro 2.2: Costos Estimados por Personal del Proyecto (en euros)

Costes del material

Material	Obsevación	Costo Estimado (€)
Ordenador (Intel Core i3, 12GB RAM)	Como mínimo se necesitaría un ordenar con estas cualidades, el precio variaría según el hardware de este	466,44
Servidor	El servidor fue ofrecido por la Uva, pero en un caso promedio se podría alquilar uno por mes con un predio promedio de 20 por mes	160
Software		
Astah	Se puedo usar gratis gracias a la licencia de estudiante, pero para usuarios normales sería 11,99€ al mes	95,92
Balsamiq	Se puedo usar gratis gracias a la licencia de estudiante,pero para usuarios normales sería 9€ al mes	72
GitHub	Software gratuito	
Visual Studio	Software gratuito	
MySQL	Software gratuito	
Java	Software gratuito	
Spring Boot	Software gratuito	
Hibernate	Software gratuito	
Node	Software gratuito	
Microsoft Teams	Software gratuito	
Costo Total de Materiales		794,36

Cuadro 2.3: Costos Estimados de Materiales del Proyecto (en euros)

2.4.2 Costes globales

Para este proyecto se optado por optimizar los costes de recursos, seleccionando software de uso gratuito, y minimizando la cantidad de recursos hardware posibles. Dejando unos costes globales:

- Costes Materiales y Software:626,44 €
- Costes personal:14.322,85€

En total, el coste del proyecto es de 15117,21€

2.5 Riesgos del proyecto

En cualquier proyecto, surgen circunstancias inesperadas que pueden obstaculizar su desarrollo inicialmente. Por eso, es crucial contar con un plan de acción para hacer frente a estos imprevistos y reducir al mínimo su impacto. Identificamos estos riesgos y definimos las medidas necesarias para prevenir su ocurrencia o gestionarlos eficazmente en caso de que ya hayan surgido.

2.5.1 Concepto de riesgo

Un riesgo se puede definir como una situación inesperada que conlleva un costo, ya sea en términos de tiempo, dinero y recursos, entre otros. Estas situaciones suelen tener un impacto negativo, aunque ocasionalmente pueden presentarse situaciones positivas, denominadas oportunidades.

Es crucial contar con un plan de acción para mitigar estos riesgos y garantizar que el proyecto avance según lo previsto. Sin embargo, la anticipación de todos estos posibles riesgos puede ser complicada. Por esta razón, es fundamental elaborar un plan de gestión de riesgos que nos permita identificar, abordar y minimizar estos riesgos antes de que se conviertan en una amenaza. En el caso de que ya hayan ocurrido, este plan nos ayudará a corregirlos.

2.5.2 Riesgos identificados

En un proyecto como este, los riesgos pueden clasificarse en diferentes categorías según su naturaleza y origen. Aquí hay una lista de tipos de riesgos comunes:

- Tecnológicos: Relacionados con la tecnología utilizada en el proyecto, como la compatibilidad de plataformas, la capacidad de escalabilidad, el rendimiento del software, la obsolescencia tecnológica, etc.
- Requerimientos: Incluye riesgos asociados con la comprensión y gestión de los requisitos del cliente, como la falta de claridad en los requisitos, cambios frecuentes en los requisitos, requisitos contradictorios, etc
- Planificación y Estimación: Se refiere a riesgos relacionados con la planificación y estimación del proyecto, como la subestimación del tiempo o del esfuerzo requerido, la asignación inadecuada de recursos, la planificación deficiente, etc.
- Gestión del Proyecto: Incluye riesgos relacionados con la gestión del proyecto, como la falta de liderazgo o comunicación, la mala gestión del alcance, presupuesto o cronograma, problemas de coordinación del equipo, etc.
- Calidad del Software: Relacionados con la calidad del producto entregado, como errores de diseño, codificación defectuosa, problemas de pruebas, deficiencias en la seguridad del software, etc.
- Recursos Humanos: Se refiere a riesgos asociados con el equipo de desarrollo, como la falta de habilidades o experiencia necesarias, la rotación del personal, la resistencia al cambio, etc.

- Externos: Incluye riesgos externos al proyecto, como cambios en la legislación o regulaciones, fluctuaciones en el mercado, problemas de terceros (proveedores, clientes, socios), desastres naturales, etc.
- Seguridad: Relacionados con la seguridad de la información y la protección de datos sensibles, como la vulnerabilidad del software a ataques cibernéticos, el robo de datos, la falta de medidas de seguridad adecuadas, etc.

2.5.3 Riesgo total

Otro de los pasos a realizar en la gestión de riesgos, es la identificación de los riesgos con mayor probabilidad e impacto, pues como podemos observar en la figura 2.1, dependiendo del grado en cada una, sera mas alto o no el riesgo. Esto nos sirve a la hora de priorizar las actuaciones para resolverlos [18].

		Probabilidad		
		Baja	Media	Alta
Impacto	Bajo	Bajo	Bajo	Medio
	Medio	Bajo	Medio	Alto
	Alto	Medio	Alto	Alto

Figura 2.1: Tabla de riesgo probabilidad e impacto

2.6 Riesgos identificados

En este apartado mostraremos los riesgos encontrados para este proyecto:

RSK-001	Desconocimiento de la materia
Descripción	El poco conocimiento de la materia nos hace recurrir a un experto
Categoría	Recursos Humanos
Vulnerabilidad	El desconocimiento para resolver funcionalidades
Amenaza	Al no saber el funcionamiento de algunas funcionalidades que hay que realizar, hay que recurrir a los expertos. Esto puede retrasar el proyecto, puesto que los expertos pueden tardar en estar localizables
Probabilidad	Alta
Impacto	Alto
Riesgo Total	Alto
Actuaciones	Encontrar a un experto antes de empezar el proyecto
Soluciones	Encontrar a un experto que este localizable cuando tengamos alguna duda.

Cuadro 2.4: Riesgo: Desconocimiento de la materia.

RSK-002	Trabajo simultaneo
Descripción	Mientras se ha realizado el proyecto, se ha estado trabajando a jornada completa simultáneamente.
Categoría	Recursos Humanos
Vulnerabilidad	El tiempo de trabajo se ve reducido gravemente
Amenaza	El tiempo que se tardara en el desarrollo del proyecto sera mayor
Probabilidad	Alta
Impacto	Alto
Riesgo Total	Alto
Actuaciones	Alargar el tiempo de desarrollo del proyecto
Soluciones	Una buena planificación para la realización de ambas cosas.

Cuadro 2.5: Riesgo: Trabajo simultaneo.

RSK-003	Caída de servidor
Descripción	La caída del servidor puede provocar la interrupción del servicio y la pérdida de datos
Categoría	Tecnológicos
Vulnerabilidad	Falta de redundancia en el hardware o en la infraestructura
Amenaza	Ataques, fallos de hardware, errores de configuración
Probabilidad	Baja
Impacto	Alto
Riesgo Total	Medio
Actuaciones	Implementar medidas de redundancia y respaldo, establecer procedimientos de recuperación ante fallos, monitorear activamente la salud del servidor
Soluciones	Utilizar sistemas de alta disponibilidad, balanceadores de carga, sistemas de respaldo automáticos, realizar auditorías de seguridad y mantenimiento periódicas

Cuadro 2.6: Riesgo: Caída de servidor.

RSK-004	Cambio de requisitos
Descripción	Los cambios en los requisitos pueden impactar en el alcance y los plazos del proyecto
Categoría	Tecnológicos
Vulnerabilidad	Falta de comunicación efectiva con el cliente o stakeholders
Amenaza	Cambios de prioridades, falta de comprensión inicial de los requisitos, cambios regulados
Probabilidad	Alta
Impacto	Alto
Riesgo Total	Alto
Actuaciones	Establecer un proceso claro para la gestión de cambios, comunicación efectiva con el cliente y partes interesadas, documentar y analizar exhaustivamente los nuevos requisitos antes de su implementación
Soluciones	Realizar revisiones periódicas del alcance del proyecto, establecer acuerdos formales para la gestión de cambios, asignar recursos adicionales si es necesario

Cuadro 2.7: Riesgo: Cambio de requisitos.

RSK-005	Falta de implicación de los interesados
Descripción	La falta de implicación de los interesados puede afectar la toma de decisiones y el avance del proyecto
Categoría	Requerimientos
Vulnerabilidad	Falta de comunicación efectiva con los interesados, desinterés o falta de compromiso por parte de los interesados
Amenaza	Cambios de prioridades, falta de claridad en los objetivos del proyecto, resistencia al cambio
Probabilidad	Medio
Impacto	Alto
Riesgo Total	Alto
Actuaciones	Identificar y comprometer a los interesados clave desde el inicio del proyecto, establecer canales de comunicación efectiva, realizar reuniones periódicas para mantener a los interesados informados y comprometidos
Soluciones	Establecer expectativas claras y responsabilidades para cada parte interesada, asignar recursos para la gestión de la participación de los interesados

Cuadro 2.8: Riesgo: Falta de implicación de los interesados.

RSK-006	Problemas de gestión de tiempo
Descripción	Los problemas de gestión de tiempo pueden retrasar la entrega del proyecto
Categoría	Gestión del Proyecto
Vulnerabilidad	Falta de planificación adecuada, subestimación de la complejidad de las tareas, cambios frecuentes en el alcance del proyecto
Amenaza	Falta de seguimiento y control del progreso, retrasos en la obtención de recursos necesarios, imprevistos y emergencias
Probabilidad	Medio
Impacto	Alto
Riesgo Total	Alto
Actuaciones	Establecer un plan de proyecto detallado con hitos y plazos claros, asignar recursos adecuadamente, realizar seguimiento regular del progreso, identificar y abordar los problemas de manera oportuna
Soluciones	Utilizar herramientas de gestión del tiempo y seguimiento del progreso, realizar análisis de riesgos y establecer planes de contingencia

Cuadro 2.9: Riesgo: Problemas de gestión de tiempo.

RSK-007	Riesgos externos imprevisibles
Descripción	Eventos externos, como desastres naturales, crisis económicas o pandemias, pueden impactar en la disponibilidad de recursos, la demanda del mercado y la viabilidad del proyecto
Categoría	Externos
Vulnerabilidad	Dependencia de factores fuera del control del equipo del proyecto
Amenaza	Desastres naturales, pandemias
Probabilidad	Baja
Impacto	Alto
Riesgo Total	Medio
Actuaciones	Realizar un análisis de riesgos externos, desarrollar planes de contingencia para diferentes escenarios.
Soluciones	Implementar tecnologías y procesos que permitan la continuidad del proyecto en situaciones adversas

Cuadro 2.10: Riesgo: Riesgos externos imprevisibles.

Marco Conceptual

Para proceder ha desarrollar este proyecto es necesario conocer una serie de conceptos básicos con relación al campo de la medicina veterinaria y su gestión en clínica, para ello se realizaran búsquedas en diversas fuentes y se consultara a los expertos pertinentes, esto nos permitirá crear un mejor modelo y nos ayuda a obtener los diversos requisitos que va a necesitar el proyecto.

3.1 Servicios Veterinarios

Un clínica veterinaria debe ofrecer una amplia gama de servicios médicos diseñados para el mantenimiento de la salud y el bienestar de las mascotas, así como para abordar cualquier problema que puedan enfrentar. Estos servicios incluyen:

3.1.1 Consultas Generales

Este tipo de consultas veterinarias son la base en la atención veterinaria, ya que permiten a los propietarios abordar cualquier preocupación relacionada con la salud y el bienestar de sus mascotas. Durante estas consultas, los veterinarios realizan exámenes físicos para evaluar la salud general del animal, así como para identificar posibles problemas de salud o enfermedades en etapas tempranas.

Además de los exámenes físicos, durante las consultas se brinda asesoramiento personalizado a los propietarios sobre el cuidado adecuado de sus mascotas, incluyendo la nutrición, el ejercicio, la higiene y otras necesidades específicas de cada animal. Los veterinarios también ofrecen recomendaciones para la prevención de enfermedades, como la vacunación o la desparasitación.

Estas consultas no solo son importantes para diagnosticar y tratar enfermedades existentes, sino también para prevenir problemas de salud futuros y promover un estilo de vida saludable para las mascotas. Además, proporcionan la oportunidad a los propietarios para recibir orientación y apoyo en la toma de decisiones relacionadas con la salud y el bienestar de sus animales.

3.1.2 Vacunaciones

La vacunación es una medida fundamental para la protección de las mascotas contra enfermedades infecciosas. Las clínicas ofrecen programas de vacunación personalizados para cada animal.

Estos programas de vacunación se basan en las pautas y recomendaciones establecidas por las asociaciones veterinarias y las organizaciones de salud animal reconocidas. Estas pautas varían según la región y pueden incluir vacunas obligatorias y opcionales, dependiendo de la prevalencia de ciertas enfermedades en el área geográfica.

Los veterinarios evalúan cuidadosamente el historial médico de la mascota y discuten con los propietarios los riesgos y beneficios de cada vacuna antes de administrarlas. También proporcionan recomendaciones sobre el calendario de vacunación adecuado, incluyendo la frecuencia de refuerzos y actualizaciones de las vacunas a lo largo de la vida del animal [16].

3.1.3 Servicio de desparasitación

La desparasitación es crucial para prevenir y tratar infestaciones de parásitos internos y externos en las mascotas. Los parásitos internos comunes incluyen lombrices intestinales, ácaros del oído, pulgas y garrapatas. Los parásitos pueden causar una variedad de problemas de salud, como anemia, pérdida de peso, irritación de la piel e incluso enfermedades transmitidas por vectores. Los veterinarios pueden recomendar programas de desparasitación regulares, que pueden incluir medicamentos orales, tópicos o inyectables, dependiendo del tipo de parásito y las necesidades específicas de la mascota. Además, la prevención de parásitos externos puede implicar el uso de collares antipulgas, pipetas, champú y pulverizadores especiales. desparasitación [14]

3.1.4 Cirugías

Se realizan cirugías diversas en una clínica, tanto de rutina como de emergencia para abordar las diversas necesidades de salud de las mascotas. Estas cirugías pueden ser [4]:

Esterilizaciones

La esterilización es una cirugía rutinaria realizada sobre todo en perros y gatos para prevenir la reproducción no deseada y proporcionar beneficios de salud a largo plazo, como la reducción del riesgo de ciertos cánceres y enfermedades uterinas.

Extracción de tumores

Los tumores pueden desarrollarse en diferentes partes del cuerpo del animal y pueden ser benignos o malignos. La cirugía para la extracción de tumores es crucial para eliminar el tejido afectado y prevenir la propagación del cáncer.

Reconstrucciones

Después de una lesión traumática, como mordeduras, accidentes automovilísticos o heridas por quemaduras, pueden ser necesaria una cirugías de reconstrucción para restaurar la función y la apariencia normales del área o áreas afectadas.

Cirugías Ortopédicas

Las cirugías ortopédicas se realizan para tratar fracturas, luxaciones, displasia de cadera, ruptura de ligamentos cruzados y otras afecciones musculoesqueléticas que afectan a la movilidad y el bienestar del animal.

Cirugías Dentales

Las cirugías dentales son comunes para tratar problemas como la periodontitis, fracturas dentales, abscesos, etc. Estas cirugías pueden implicar extracciones dentales, tratamientos de conducto radicular o procedimientos de limpieza dental profunda.

Cirugías de tejidos blandos

Estas incluyen una amplia gama de procedimientos que van desde la corrección de paladar hendido hasta la eliminación de cuerpos extraños en el tracto gastrointestinal o la reparación de hernias.

Cirugías de emergencia

Estas cirugías pueden ser necesarias para tratar condiciones urgentes y potencialmente mortales, como la torsión gástrica en perros o la obstrucción intestinal.

3.1.5 Servicio de eutanasia

El servicio de eutanasia es un aspecto importante y sensible de la atención veterinaria, que permite a los propietarios despedirse de sus mascotas de manera digna y compasiva cuando su calidad de vida se ve comprometida irremediablemente por enfermedad, dolor o sufrimiento. Las clínicas veterinarias suelen ofrecer este servicio como una opción humanitaria para aliviar el sufrimiento de los animales y brindar consuelo a sus dueños en momentos difíciles.

Es importante que los propietarios reciban información clara y concisa sobre el proceso de eutanasia, incluidos los aspectos médicos y emocionales involucrados, para que puedan tomar decisiones informadas y sentirse cómodos con el curso de acción elegido para su mascota.

3.1.6 Atención de emergencia

Las clínicas suelen contar con personal y equipo especializado para brindar esta atención. El personal de emergencia debe estar capacitado para evaluar rápidamente la situación, estabilizar al paciente y proporcionar el tratamiento necesario para salvar vidas. Las clínicas que brindan este servicio suelen estar equipadas con instalaciones y equipos avanzados, como unidades de cuidados intensivos, equipos de monitorización, equipos de diagnóstico por imágenes y quirófanos, para brindar una atención de alta calidad en momentos críticos.

3.2 Gestión de Historiales

Las clínicas deben llevar una adecuada gestión de historiales veterinarios de cada mascota, además de realizar las prácticas recomendadas para mantener registros precisos y actualizados de la información médica de cada paciente. La gestión de historiales es esencial para garantizar una atención veterinaria de calidad y proporcionar un seguimiento adecuado del progreso de los pacientes a lo largo del tiempo. A continuación, se describen algunas prácticas clave en la gestión de historiales:

3.2.1 Registro detallado

- **Historial completo:** Registrar de manera detallada y precisa la información médica de cada paciente, incluyendo datos como la historia de vacunación, tratamientos anteriores, resultados de exámenes, diagnósticos, procedimientos quirúrgicos y cualquier otra información relevante para la salud del animal.
- **Notas de consulta exhaustivas:** Tomar notas detalladas durante cada consulta, incluyendo la razón de la visita, los síntomas presentados, el examen físico realizado, las pruebas diagnósticas realizadas y cualquier recomendación de tratamiento o seguimiento.

3.2.2 Organización y almacenamiento

- **Sistema de archivos eficiente:** Mantener un sistema de archivos organizado y fácil de usar para almacenar y recuperar historiales médicos de manera rápida y precisa.
- **Registro electrónico:** Utilizar sistemas de gestión de registros electrónicos (EHR) para facilitar la entrada de datos, la búsqueda de información y la generación de informes personalizados.

3.2.3 Seguridad y confidencialidad

- **Protección de datos:** Implementar medidas de seguridad para proteger la información médica de los pacientes, como el uso de contraseñas seguras, cifrado de datos y acceso restringido a los registros.
- **Cumplimiento normativo:** Asegurarse de cumplir con las regulaciones locales y nacionales sobre privacidad y protección de datos, como la Ley de Portabilidad y Responsabilidad del Seguro de Salud.

3.3 Gestión Personal

En esta sección, se abordará la importancia del personal en una clínica veterinaria y los roles que desempeñan para brindar una atención integral y de calidad a las mascotas. El personal es uno de los pilares fundamentales de cualquier clínica veterinaria, y su experiencia, capacitación y compromiso son cruciales para el éxito y la reputación del establecimiento. A continuación, se describen algunos de los roles clave en una clínica veterinaria:

3.3.1 Veterinarios

- **Veterinario general:** Responsable de realizar exámenes físicos, diagnosticar enfermedades, prescribir tratamientos, realizar cirugías y brindar atención médica general a los pacientes.
- **Especialistas:** Veterinarios con formación avanzada en áreas específicas como cirugía, medicina interna, dermatología, oftalmología, oncología, entre otras, que brindan atención especializada a los pacientes con necesidades particulares.

3.3.2 Asistentes y Técnicos Veterinarios (ATV)

- **Asistente veterinario:** Ayuda en la atención general de los pacientes, realiza tareas de limpieza, prepara equipos, ayuda en procedimientos quirúrgicos y proporciona apoyo al veterinario durante consultas y cirugías.
- **Técnico veterinario:** Tiene una formación más avanzada y puede realizar tareas más especializadas, como la toma de radiografías, la administración de medicamentos y tratamientos, la toma de muestras para análisis de laboratorio, entre otras funciones.

3.3.3 Personal de recepción y administrativo

- **Recepcionista:** Responsable de recibir a los propietarios y pacientes, programar citas, gestionar registros médicos, gestionar pagos y proporcionar información sobre servicios veterinarios.
- **Personal administrativo:** Encargado de tareas administrativas como contabilidad, facturación, gestión de seguros, compras de suministros y coordinación de programas de marketing y publicidad.

3.3.4 Personal de limpieza y mantenimiento

- **Personal de limpieza:** Encargado de mantener las instalaciones limpias y ordenadas, desinfectar áreas de trabajo y hospitalización, y garantizar un entorno higiénico y seguro para los pacientes y el personal.

3.3.5 Otros roles especializados

- **Groomers (peluqueros):** Especialistas en cuidado y arreglo personal de las mascotas, que ofrecen servicios como baño, corte de pelo, limpieza de uñas y cuidado de la piel y el pelaje.

- **Entrenadores de comportamiento:** Profesionales especializados en modificar comportamientos no deseados en mascotas, proporcionando orientación y entrenamiento a propietarios sobre técnicas de manejo y modificación de conducta.

3.4 Gestión de clientes

En una clínica veterinaria es de vital importancia la gestión de clientes y las estrategias utilizadas para proporcionar un servicio eficiente y satisfactorio a los propietarios de mascotas. La gestión de clientes es fundamental para mantener relaciones sólidas con los propietarios y garantizar su satisfacción con los servicios veterinarios ofrecidos.

3.4.1 Atención al cliente

- **Recepción amigable:** Proporcionar una recepción cordial y acogedora a los propietarios y sus mascotas al llegar a la clínica.
- **Comunicación clara:** Mantener una comunicación clara y efectiva con los propietarios sobre los servicios ofrecidos, los costos asociados y los procedimientos médicos.
- **Escucha activa:** Escuchar atentamente las preocupaciones y necesidades de los propietarios, mostrando empatía y comprensión.
- **Respuesta rápida:** Responder rápidamente a las consultas y solicitudes de los propietarios, ya sea por teléfono, correo electrónico o en persona.

3.4.2 Gestión de citas

- **Programación eficiente:** Gestionar las citas de manera eficiente para minimizar el tiempo de espera de los propietarios y optimizar la agenda de los veterinarios.
- **Recordatorios de citas:** Enviar recordatorios de citas por correo electrónico, mensajes de texto o llamadas telefónicas para ayudar a los propietarios a recordar sus citas programadas.
- **Flexibilidad:** Ofrecer opciones de programación flexibles para adaptarse a las necesidades y horarios de los propietarios.

3.4.3 Seguimiento post-consulta

- **Seguimiento médico:** Realizar un seguimiento con los propietarios después de las consultas para verificar el progreso de la mascota y responder cualquier pregunta adicional.
- **Encuestas de satisfacción:** Solicitar retroalimentación a los propietarios mediante encuestas de satisfacción para evaluar la calidad de los servicios y identificar áreas de mejora.
- **Fidelización de clientes:** Implementar programas de fidelización de clientes, ofreciendo descuentos especiales, programas de recompensas o beneficios exclusivos para clientes recurrentes.

3.5 Equipamiento

En esta sección, se abordará los distintos tipos de equipamiento necesario en una clínica veterinaria para brindar una atención de alta calidad a las mascotas. Estos elementos son fundamentales para garantizar un entorno seguro, cómodo y eficiente tanto para los pacientes como para el personal veterinario.

Equipos de diagnóstico

Incluyen equipos como estetoscopios, otoscopios, oftalmoscopios, termómetros, tensiómetros y glucómetros para evaluar la salud de los pacientes.

Equipos quirúrgicos

Instrumentos quirúrgicos, lámparas de quirófano, mesas de cirugía, monitores de anestesia y equipos de esterilización necesarios para realizar procedimientos quirúrgicos.

Equipos de anestesia

Incluyen máquinas de anestesia, ventiladores, monitores de gases anestésicos y suministros para administrar anestesia de manera segura durante los procedimientos quirúrgicos.

Equipos de imagenología

Equipos de rayos X, ecógrafos y otros equipos de diagnóstico por imagen necesarios para evaluar problemas de salud internos y lesiones.

Equipos de laboratorio

Incluyen centrífugas, microscopios, analizadores bioquímicos y hematológicos, y otros equipos necesarios para realizar pruebas de laboratorio en muestras de sangre, orina y tejidos.

Equipos de hospitalización

Jaulas, camas, mantas térmicas y equipos de infusión necesarios para brindar cuidados intensivos a los pacientes hospitalizados.

Equipos de seguridad y control de infecciones

Incluyen autoclaves, equipos de esterilización, desinfectantes, equipos de protección personal y sistemas de gestión de residuos médicos para garantizar un entorno seguro y limpio.

3.6 Gestión de inventario

Una adecuada gestión de inventario en una clínica y las prácticas recomendadas para garantizar un control eficiente y efectivo de los suministros y equipos médicos, es una gran ayuda para la eficiencia de la clínica. La gestión de inventario es esencial para asegurar que la clínica disponga de los productos y materiales necesarios para brindar una atención veterinaria de calidad y evitar la escasez o exceso de stock.

3.6.1 Registro y seguimiento

- **Inventario completo:** Realizar un inventario inicial de todos los suministros y equipos médicos de la clínica, incluyendo medicamentos, dispositivos médicos, materiales de curación, alimentos, entre otros.
- **Seguimiento de entradas y salidas:** Registrar todas las entradas y salidas de inventario de manera precisa y actualizada, utilizando sistemas de registro electrónico o manual.

3.6.2 Control de stock

- **Niveles de stock adecuados:** Establecer niveles mínimos y máximos de stock para cada producto, con el fin de evitar la escasez o exceso de inventario.
- **Reabastecimiento oportuno:** Programar pedidos de reabastecimiento de inventario de manera regular y oportuna, con base en la demanda y el uso estimado de los productos.

3.6.3 Organización y almacenamiento

- **Almacenamiento adecuado:** Organizar los suministros y equipos médicos de manera ordenada y segura, utilizando estanterías, armarios o contenedores adecuados para cada tipo de producto.
- **Rotación de stock:** Implementar prácticas de rotación de stock para garantizar que los productos más antiguos se utilicen primero y evitar la caducidad de medicamentos u otros productos perecederos.

Soluciones Existentes

En el ámbito de la medicina veterinaria, una gestión eficiente de la información es esencial para brindar un servicio de calidad tanto para los profesionales en el campo como para los propietarios de mascotas. En este contexto, han sido desarrolladas diversas soluciones de software para abordar las necesidades específicas de las clínicas y hospitales veterinarios. Sin embargo, es importante destacar que muchas de estas soluciones aún presentan limitaciones en cuanto a la comunicación con los propietarios de las mascotas y la accesibilidad a la información.

A continuación, se presenta un análisis de algunas de las soluciones existentes en el mercado, destacando sus características principales y señalando las áreas en las que podrían mejorar para satisfacer mejor las necesidades tanto de los profesionales como de los propietarios de mascotas.

- **Covetrus EMEA Software:** Covetrus EMEA Software es una solución de software de gestión veterinaria ampliamente utilizada que ofrece una variedad de funciones, como la gestión de registros médicos, programación de citas, facturación y seguimiento de inventario. Sin embargo, carece de capacidades de comunicación con los propietarios de las mascotas y no ofrece una interfaz que permita a los propietarios acceder a información sobre su animal, consultar citas o verificar tratamientos y medicamentos necesarios [6].
- **VetIT:** VetIT es una solución de software versátil que se adapta tanto a clínicas veterinarias como a hospitales de animales más grandes. Ofrece una amplia gama de funciones, desde registros médicos hasta programación de citas y análisis de datos. A pesar de ello, no incluye la capacidad de comunicación con los dueños o responsables de los animales, ni un calendario con las citas y los detalles del animal para una gestión más eficiente por parte del veterinario [23].
- **VETport:** VETport es un sistema de gestión veterinaria basado en la nube que abarca todas las necesidades de una clínica veterinaria, desde la gestión de pacientes hasta la facturación, la analítica de datos y tutoriales en video. Aunque ofrece una amplia gama de funciones, tampoco proporciona la funcionalidad de comunicación con el paciente ni una interfaz para que este consulte datos sobre su mascota, ver sus citas o los tratamientos y medicamentos necesarios [24].

Análisis

El análisis es una etapa crucial en el desarrollo del proyecto, donde se desglosan los elementos clave y se establecen las bases para su diseño y desarrollo. En este capítulo, se examinarán detalladamente los requisitos proporcionados por el usuario, que son la base de cualquier proyecto, además también se abordarán otros aspectos importantes del análisis, como el modelo de clases y los casos de uso. Estos son elementos fundamentales en la ingeniería de software que contribuyen considerablemente a la comprensión y representación del sistema en desarrollo.

5.1 Requisitos funcionales

Los requisitos funcionales son requisitos detallados de funciones específicas que el sistema debe ser capaz de realizar, es decir, definen como el sistema debe comportarse bajo ciertas condiciones o en respuesta a determinadas acciones por parte de los actores.

RF-001	Inicio de sesión
Descripción	El sistema deberá permitir el inicio de sesión a los clientes

Cuadro 5.1: Requisitos funcionales RF-001

RF-002	Inicio de sesión
Descripción	El sistema deberá permitir el inicio de sesión del administrador.

Cuadro 5.2: Requisitos funcionales RF-002

RF-003	Inicio de sesión
Descripción	EL sistema deberá permitir el inicio de sesión los facultativos.

Cuadro 5.3: Requisitos funcionales RF-003

RF-004	Consultar mascotas
Descripción	El sistema deberá permitir al cliente la visualización de los datos de sus mascotas.

Cuadro 5.4: Requisitos funcionales RF-004

RF-005	Consultar mascotas
Descripción	El sistema deberá permitir al facultativo la visualización de todas las mascotas.

Cuadro 5.5: Requisitos funcionales RF-005

RF-006	Consultar calendario
Descripción	El sistema deberá permitir al usuario la visualización de su calendario de citas y eventos.

Cuadro 5.6: Requisitos funcionales RF-006

RF-007	Consultar historiales
Descripción	El sistema deberá permitir al facultativos visualizar los historiales de cada mascota.

Cuadro 5.7: Requisitos funcionales RF-007

RF-008	Consultar clientes
Descripción	El sistema deberá permitir al facultativos visualizar los datos de cada cliente.

Cuadro 5.8: Requisitos funcionales RF-008

RF-009	Consulta de inventario
Descripción	El sistema deberá permitir al facultativos y administrador la visualización de los recursos de la clínica.

Cuadro 5.9: Requisitos funcionales RF-009

RF-010	Gestión de inventario
Descripción	El sistema deberá permitir al facultativos y administrador incorporar nuevos recursos al sistema.

Cuadro 5.10: Requisitos funcionales RF-010

RF-011	Gestión de inventario
Descripción	El sistema deberá permitir al facultativos y administrador eliminar recursos al sistema.

Cuadro 5.11: Requisitos funcionales RF-011

RF-012	Gestión de notificaciones
Descripción	El sistema deberá informar al cliente mediante notificaciones de las citas y tratamientos.

Cuadro 5.12: Requisitos funcionales RF-012

RF-013	Envío de citas
Descripción	El sistema deberá permitir mandar citas a los usuarios no registrados a los facultativos.

Cuadro 5.13: Requisitos funcionales RF-013

RF-014	Registro de historiales
Descripción	El sistema deberá permitir al facultativos registrar nuevos historiales.

Cuadro 5.14: Requisitos funcionales RF-014

RF-015	Registro de consultas
Descripción	El sistema deberá permitir al facultativos registrar nuevas consultas.

Cuadro 5.15: Requisitos funcionales RF-015

RF-016	Gestión de historiales
Descripción	El sistema deberá permitir editar datos de los historiales de cada mascota.

Cuadro 5.16: Requisitos funcionales RF-016

RF-017	Gestión de calendario
Descripción	El sistema deberá permitir al facultativo adjuntar historiales, documentos o imágenes a eventos del calendario.

Cuadro 5.17: Requisitos funcionales RF-017

RF-018	Registro
Descripción	El sistema deberá permitir el registro a los visitantes.

Cuadro 5.18: Requisitos funcionales RF-018

RF-019	Registro
Descripción	El sistema deberá permitir que el administrador registre nuevos facultativos.

Cuadro 5.19: Requisitos funcionales RF-019

5.2 Requisitos no funcionales

Este tipo de requisito describe las características y restricciones que no están directamente relacionadas con las funciones específicas del sistema. Suelen ser restricciones u obligaciones. En este proyecto, los requisitos no funcionales extraídos son:

RNF-001	Encriptación de datos
Descripción	Todos los datos sensibles, como la información del paciente, deben estar encriptados en tránsito y en reposo.

Cuadro 5.20: Requisitos no funcionales RNF-001

RNF-002	Tiempo
Descripción	El sistema debe estar disponible para los usuarios al menos el 99 por ciento del tiempo, excluyendo ventanas de mantenimiento programadas.

Cuadro 5.21: Requisitos no funcionales RNF-002

RNF-003	Documentación
Descripción	Debe haber documentación completa y actualizada para el código, arquitectura y procedimientos.

Cuadro 5.22: Requisitos no funcionales RNF-003

RNF-004	Localización
Descripción	El sistema debe admitir diferentes idiomas y regiones, si es aplicable.

Cuadro 5.23: Requisitos no funcionales RNF-004

RNF-005	Autorización
Descripción	El sistema deberá permitir el acceso a las partes del sistema para las que tienen autorización.

Cuadro 5.24: Requisitos no funcionales RNF-005

5.3 Requisitos funcionales de información

Los requisitos de información o de datos, como dice su nombre se centran en los datos que el sistema debe de manejar, almacenar y procesar. En este proyecto, los requisitos de información extraídos son:

RNF-001	Formato de fechas y monedas
Descripción	El sistema debe ser configurable para mostrar fechas y monedas según las preferencias locales.

Cuadro 5.25: Requisitos no funcionales RNF-002

5.4 Casos de uso

Los casos de uso desglosan las interacciones de los usuarios con el sistema, además de definir los accesos a este. En esta sección se describen los casos de uso encontrados durante el análisis.

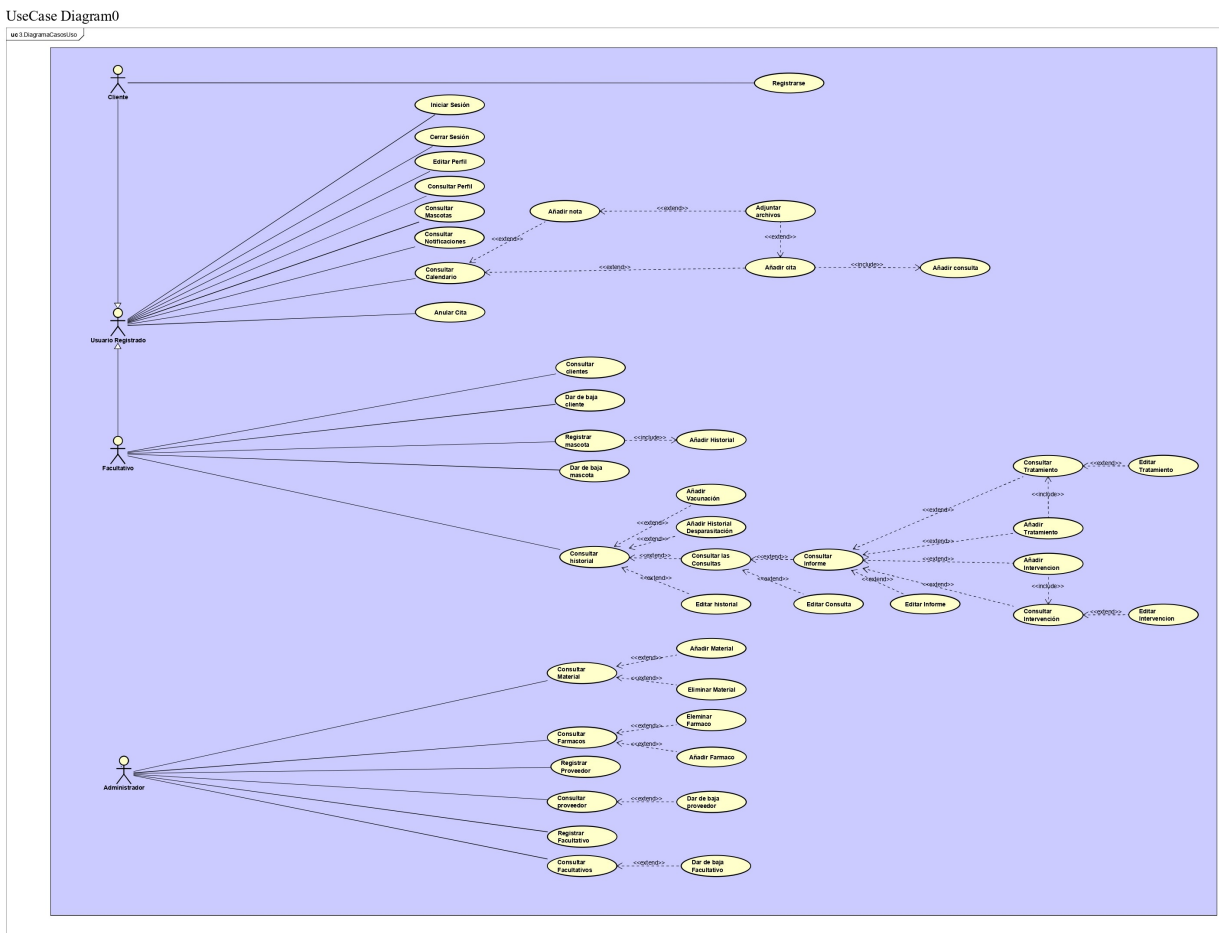


Figura 5.1: Diagrama de casos de uso

Use Case	Adjuntar archivos
Summary	
Actor	
Precondition	<p>Usuario registrado con sesión iniciada</p> <p>Usuario registrado con recordatorios en su calendario</p> <p>Usuario registrado con citas en su calendario</p>
Postcondition	
Base Sequence	<ol style="list-style-type: none"> 1. Usuario registrado pide consultar el calendario 2. Sistema muestra el calendario 3. Usuario registrado pide añadir archivo a un recordatorio o cita 4. Sistema pide archivo al usuario registrado 5. Usuario sube archivo al sistema. 6. Sistema pide confirmación 7. Usuario registrado confirma 8. Sistema guarda archivo, y fin del caso de uso
Branch Sequence	<p>5a. Usuario no introduce fecha, hora y descripción</p> <ol style="list-style-type: none"> 1. Sistema no guarda archivo, ir al paso uno <p>7a. Usuario no confirma la subida</p> <ol style="list-style-type: none"> 1. El sistema pide confirmación de cancelación 2. Usuario cancela, ir al paso uno
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.26: Use Case: Adjuntar archivos

Use Case	Anular Cita
Summary	Un usuario registrado solicita anular una cita previamente agendada en su calendario.
Actor	Usuario Registrado
Precondition	Usuario registrado con sesión iniciada.
Postcondition	La cita se anula correctamente en el calendario.
Base Sequence	<ol style="list-style-type: none"> 1. Usuario Registrado solicita anular cita. 2. Sistema solicita confirmación de anulación y envía la petición de anulación al otro usuario registrado. 3. Usuario Registrado confirma la anulación. 4. Sistema anula la cita y finaliza el caso de uso.
Branch Sequence	3a. Usuario no da confirmación de anulación <ol style="list-style-type: none"> 1. El sistema devuelve la solicitud de anulación al otro usuario registrado, regresar al paso uno.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.27: Use Case: Anular Cita

Use Case	Añadir Farmaco
Summary	Un administrador registrado añade un nuevo fármaco al sistema.
Actor	Administrador
Precondition	Administrador registrado con sesión iniciada.
Postcondition	El fármaco se añade correctamente al sistema.
Base Sequence	<ol style="list-style-type: none"> 1. El administrador registrado solicita consultar los fármacos. 2. El sistema muestra los fármacos. 3. El administrador registrado solicita añadir un nuevo fármaco. 4. El sistema solicita información del fármaco al administrador registrado. 5. El administrador introduce la información del fármaco en el sistema. 6. El sistema guarda el fármaco y finaliza el caso de uso.
Branch Sequence	5a. El administrador no introduce información del fármaco <ol style="list-style-type: none"> 1. El sistema no guarda el fármaco, regresar al paso uno.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.28: Use Case: Añadir Farmaco

Use Case	Añadir Historial
Summary	Un facultativo registrado con sesión iniciada registra una mascota en el sistema, creando su historial.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada y mascota registrada.
Postcondition	Se registra el historial de la mascota en el sistema.
Base Sequence	1. El facultativo registra una mascota. 2. El sistema crea un historial para la mascota y finaliza el caso de uso.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.29: Use Case: Añadir Historial

Use Case	Añadir Historial Desparasitación
Summary	Un facultativo registrado con sesión iniciada añade una nueva fecha de desparasitación al historial de una mascota registrada en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada y mascota registrada.
Postcondition	Se añade una nueva fecha de desparasitación al historial de la mascota en el sistema.
Base Sequence	1. El facultativo solicita consultar el historial de desparasitación de una mascota. 2. El sistema muestra el historial de desparasitación. 3. El facultativo solicita añadir una nueva fecha de desparasitación. 4. El sistema solicita la fecha de desparasitación. 5. El facultativo introduce la fecha. 6. El sistema registra la nueva fecha de desparasitación y vuelve a mostrar el historial de desparasitación. 7. Fin del caso de uso.
Branch Sequence	5a. El facultativo no introduce la fecha 1. El sistema no registra la nueva fecha de desparasitación y vuelve a mostrar el historial de desparasitación, y regresar al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.30: Use Case: Añadir Historial Desparasitación

Use Case	Añadir Intervención
Summary	Un facultativo registrado con sesión iniciada añade una nueva intervención al historial de una mascota registrada en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada y mascota registrada.
Postcondition	Se añade una nueva intervención al historial de la mascota en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. El facultativo solicita consultar el informe con las intervenciones. 2. El sistema muestra el informe con las intervenciones. 3. El facultativo solicita añadir una nueva intervención. 4. El sistema solicita información sobre la nueva intervención. 5. El facultativo introduce la información de la intervención. 6. El sistema registra la intervención.
Branch Sequence	5a.El facultativo no introduce información sobre la nueva intervención <ol style="list-style-type: none"> 1. El sistema no registra la intervención,y regresar al paso 1.
Exception Sequence	
Sub UseCase	Consultar Intervención
Note	

Cuadro 5.31: Use Case: Añadir Intervención

Use Case	Añadir Material
Summary	Un administrador registrado con sesión iniciada añade un nuevo material al sistema.
Actor	Administrador
Precondition	Administrador registrado con sesión iniciada.
Postcondition	Se añade un nuevo material al sistema.
Base Sequence	<ol style="list-style-type: none"> 1. El administrador solicita consultar los materiales. 2. El sistema muestra los materiales. 3. El administrador solicita añadir un nuevo material. 4. El sistema solicita información sobre el nuevo material. 5. El administrador introduce la información del material. 6. El sistema guarda el material.
Branch Sequence	5a. El administrador no introduce la información del material <ol style="list-style-type: none"> 1. El sistema no guarda el material,y regresar al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.32: Use Case: Añadir Material

Use Case	Añadir Tratamiento
Summary	Un facultativo registrado con sesión iniciada añade un nuevo tratamiento para una mascota en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada. Mascota registrada en el sistema.
Postcondition	Se añade un nuevo tratamiento para la mascota en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. El facultativo solicita consultar el informe. 2. El sistema muestra el informe con los tratamientos. 3. El facultativo solicita añadir un nuevo tratamiento. 4. El sistema solicita información sobre el nuevo tratamiento. 5. El facultativo introduce la información del tratamiento. 6. El sistema registra el tratamiento.
Branch Sequence	5a. El facultativo no introduce la información del tratamiento <ol style="list-style-type: none"> 1. El sistema no registra el tratamiento, y regresar al paso 1.
Exception Sequence	
Sub UseCase	Consultar Tratamiento
Note	

Cuadro 5.33: Use Case: Añadir Tratamiento

Use Case	Añadir Vacunación
Summary	Un facultativo registrado con sesión iniciada añade una nueva vacunación para una mascota en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada. Mascota registrada en el sistema.
Postcondition	Se añade una nueva vacunación para la mascota en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. El facultativo solicita consultar las vacunaciones. 2. El sistema muestra las vacunaciones. 3. El facultativo solicita añadir una nueva vacunación. 4. El sistema solicita fecha y duración. 5. El facultativo introduce la fecha y duración. 6. El sistema registra la nueva vacunación.
Branch Sequence	5a. El facultativo no introduce la fecha y duración <ol style="list-style-type: none"> 1. El sistema no registra la nueva vacunación, y regresar al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.34: Use Case: Añadir Vacunación

Use Case	Añadir cita
Summary	Un usuario registrado con sesión iniciada añade una nueva cita en el calendario.
Actor	Usuario registrado
Precondition	Usuario registrado con sesión iniciada.
Postcondition	Se añade una nueva cita en el calendario del usuario.
Base Sequence	<ol style="list-style-type: none"> 1. El usuario registrado solicita consultar el calendario. 2. El sistema muestra el calendario. 3. El usuario registrado solicita añadir una cita. 4. El sistema solicita fecha, hora, paciente y descripción. 5. El usuario introduce fecha, hora, paciente y descripción. 6. El sistema envía una notificación al receptor para que acepte la cita. 7. El usuario registrado confirma la cita. 8. El sistema guarda la cita.
Branch Sequence	<ol style="list-style-type: none"> 5a. Usuario no introduce fecha, hora y descripción <ol style="list-style-type: none"> 1 El sistema no guarda la cita y vuelve al paso 1. 7a. Usuario no confirma la cita <ol style="list-style-type: none"> 1 El sistema solicita una nueva fecha al usuario. 2 El usuario introduce una nueva fecha. 3 El sistema envía la nueva cita al receptor y vuelve al paso 6.
Exception Sequence	
Sub UseCase	Añadir consulta
Note	

Cuadro 5.35: Use Case: Añadir cita

Use Case	Añadir consulta
Summary	Un facultativo registrado con sesión iniciada acepta una cita y crea una consulta y un informe.
Actor	Facultativo registrado
Precondition	Facultativo registrado con sesión iniciada y mascota registrada con historial.
Postcondition	Se crea una consulta y un informe asociados a la cita aceptada.
Base Sequence	<ol style="list-style-type: none"> 1. El facultativo registrado acepta una cita. 2. El sistema crea una consulta y un informe.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.36: Use Case: Añadir consulta

Use Case	Añadir nota
Summary	Un usuario registrado con sesión iniciada agrega una nota a su calendario.
Actor	Usuario registrado
Precondition	Usuario registrado con sesión iniciada.
Postcondition	Se agrega la nota al calendario del usuario.
Base Sequence	<ol style="list-style-type: none"> 1. El usuario registrado pide consultar su calendario. 2. El sistema muestra el calendario. 3. El usuario registrado pide añadir una nota. 4. El sistema solicita fecha, hora y descripción al usuario registrado. 5. El usuario introduce fecha, hora y descripción en el sistema. 6. El sistema guarda la nota.
Branch Sequence	<ol style="list-style-type: none"> 5a. El usuario no introduce fecha, hora y descripción en el sistema 1. El sistema no guarda la nota, y regresar al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.37: Use Case: Añadir nota

Use Case	Cerrar Sesión
Summary	Un usuario registrado solicita cerrar su sesión en el sistema.
Actor	Usuario Registrado
Precondition	El usuario tiene una sesión iniciada en el sistema.
Postcondition	La sesión del usuario se cierra correctamente.
Base Sequence	<ol style="list-style-type: none"> 1. El usuario registrado solicita cerrar sesión. 2. El sistema solicita confirmación al usuario. 3. El usuario confirma el cierre de sesión. 4. El sistema cierra la sesión del usuario.
Branch Sequence	<ol style="list-style-type: none"> 3a. El usuario no confirma el cierre de sesión. 1. El sistema no cierra la sesión y termina el caso de uso.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.38: Use Case: Cerrar Sesión

Use Case	Consultar Calendario
Summary	Un usuario registrado solicita consultar su calendario en el sistema.
Actor	Usuario Registrado
Precondition	El usuario tiene una sesión iniciada en el sistema.
Postcondition	El usuario puede ver su calendario correctamente.
Base Sequence	1. El usuario registrado solicita consultar su calendario. 2. El sistema muestra el calendario al usuario.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.39: Use Case: Consultar Calendario

Use Case	Consultar Facultativos
Summary	Un administrador solicita consultar la lista de facultativos registrados en el sistema.
Actor	Administrador
Precondition	El administrador tiene una sesión iniciada en el sistema.
Postcondition	El administrador puede ver la lista de facultativos correctamente.
Base Sequence	1. El administrador solicita consultar la lista de facultativos. 2. El sistema muestra la lista de facultativos al administrador.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.40: Use Case: Consultar Facultativos

Use Case	Consultar Farmacos
Summary	Un administrador solicita consultar la lista de fármacos registrados en el sistema.
Actor	Administrador
Precondition	El administrador tiene una sesión iniciada en el sistema.
Postcondition	El administrador puede ver la lista de fármacos correctamente.
Base Sequence	1. El administrador solicita consultar la lista de fármacos. 2. El sistema muestra la lista de fármacos al administrador.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.41: Use Case: Consultar Farmacos

Use Case	Consultar Informe
Summary	Un facultativo solicita consultar el informe médico de una mascota registrada en el sistema.
Actor	Facultativo
Precondition	El facultativo tiene una sesión iniciada en el sistema y la mascota está registrada.
Postcondition	El facultativo puede ver el informe médico de la mascota correctamente.
Base Sequence	1. El facultativo solicita consultar el informe médico. 2. El sistema muestra el informe médico al facultativo.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.42: Use Case: Consultar Informe

Use Case	Consultar Intervención
Summary	Un facultativo solicita consultar una intervención médica realizada en una mascota registrada en el sistema.
Actor	Facultativo
Precondition	El facultativo tiene una sesión iniciada en el sistema y la mascota está registrada.
Postcondition	El facultativo puede ver la intervención médica correctamente.
Base Sequence	<ol style="list-style-type: none"> 1. El facultativo solicita consultar el informe médico. 2. El sistema muestra el informe médico con las intervenciones al facultativo. 3. El facultativo selecciona una intervención para ver los detalles. 4. El sistema muestra los detalles de la intervención seleccionada.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.43: Use Case: Consultar Intervención

Use Case	Consultar Mascotas
Summary	Un usuario registrado solicita consultar las mascotas registradas en el sistema.
Actor	Usuario Registrado
Precondition	El usuario tiene una sesión iniciada en el sistema y al menos una mascota está registrada.
Postcondition	El usuario puede ver la lista de sus mascotas correctamente.
Base Sequence	<ol style="list-style-type: none"> 1. El usuario registrado solicita consultar sus mascotas. 2. El sistema muestra la lista de mascotas del usuario.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.44: Use Case: Consultar Mascotas

Use Case	Consultar Material
Summary	Un administrador registrado solicita consultar el material disponible en el sistema.
Actor	Administrador
Precondition	El administrador tiene una sesión iniciada en el sistema.
Postcondition	El administrador puede ver la lista de material correctamente.
Base Sequence	1. El administrador solicita consultar el material. 2. El sistema muestra la lista de material disponible.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.45: Use Case: Consultar Material

Use Case	Consultar Notificaciones
Summary	Un usuario registrado solicita consultar sus notificaciones en el sistema.
Actor	Usuario Registrado
Precondition	El usuario registrado tiene una sesión iniciada en el sistema.
Postcondition	El usuario puede ver sus notificaciones correctamente.
Base Sequence	1. El usuario registrado solicita consultar sus notificaciones. 2. El sistema muestra las notificaciones del usuario.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.46: Use Case: Consultar Notificaciones

Use Case	Consultar Perfil
Summary	Un usuario registrado solicita consultar su perfil en el sistema.
Actor	Usuario Registrado
Precondition	El usuario registrado tiene una sesión iniciada en el sistema.
Postcondition	El usuario puede ver la información de su perfil correctamente.
Base Sequence	1. El usuario registrado solicita consultar su perfil. 2. El sistema muestra la información del perfil del usuario.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.47: Use Case: Consultar Perfil

Use Case	Consultar Tratamiento
Summary	Un facultativo con sesión iniciada solicita consultar el informe con los tratamientos de una mascota registrada en el sistema.
Actor	Facultativo
Precondition	El facultativo tiene una sesión iniciada en el sistema. La mascota está registrada en el sistema.
Postcondition	El facultativo puede ver los tratamientos de la mascota correctamente.
Base Sequence	1. El facultativo solicita consultar el informe. 2. El sistema muestra el informe con los tratamientos. 3. El facultativo selecciona un tratamiento para consultar. 4. El sistema muestra el tratamiento seleccionado.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.48: Use Case: Consultar Tratamiento

Use Case	Consultar historial
Summary	Un facultativo registrado con sesión iniciada y una mascota registrada solicita consultar el historial en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada, mascota registrada.
Postcondition	El historial se muestra en el sistema.
Base Sequence	1. Facultativo pide consultar historial. 2. Sistema muestra el historial, y fin del caso de uso.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.49: Use Case: Consultar historial

Use Case	Consultar las Consultas
Summary	Un facultativo registrado con sesión iniciada y una mascota registrada solicita consultar las consultas en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada, mascota registrada.
Postcondition	El sistema muestra las consultas.
Base Sequence	1. Facultativo pide consultar consultas. 2. Sistema muestra las consultas, y fin del caso de uso.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.50: Use Case: Consultar las Consultas

Use Case	Consultar Proveedor
Summary	Un administrador registrado con sesión iniciada solicita consultar los proveedores en el sistema.
Actor	Administrador
Precondition	Administrador registrado con sesión iniciada.
Postcondition	El sistema muestra los proveedores.
Base Sequence	1. Administrador pide consultar los proveedores. 2. Sistema muestra los proveedores, y fin de caso de uso.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.51: Use Case: Consultar Proveedor

Use Case	Dar de Baja Facultativo
Summary	Un administrador registrado con sesión iniciada solicita dar de baja a un facultativo en el sistema.
Actor	Administrador
Precondition	Administrador registrado con sesión iniciada.
Postcondition	El facultativo es dado de baja en el sistema.
Base Sequence	1. Administrador pide dar de baja al facultativo. 2. Sistema pide confirmación de baja. 3. Administrador da confirmación de la baja. 4. Sistema da de baja al facultativo, fin del caso de uso.
Branch Sequence	3a. Administrador no da confirmación de baja: 1. El sistema pide confirmación de cancelación. 2. Administrador cancela, ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.52: Use Case: Dar de Baja Facultativo

Use Case	Dar de Baja Cliente
Summary	Un facultativo registrado con sesión iniciada solicita dar de baja a un cliente en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada.
Postcondition	El cliente es dado de baja en el sistema.
Base Sequence	1. Facultativo pide dar de baja cliente. 2. Sistema pide confirmación de baja. 3. Facultativo da confirmación de la baja. 4. Sistema da de baja al cliente, fin del caso de uso.
Branch Sequence	3a. Facultativo no da confirmación de baja: 1. El sistema vuelve a mostrar el cliente, ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.53: Use Case: Dar de Baja Cliente

Use Case	Dar de Baja Mascota
Summary	Un facultativo registrado con sesión iniciada solicita dar de baja a una mascota en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada, mascota registrada.
Postcondition	La mascota es dada de baja en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Facultativo pide dar de baja mascota. 2. Sistema pide confirmación de baja. 3. Facultativo da confirmación de la baja. 4. Sistema da de baja a la mascota, fin del caso de uso.
Branch Sequence	3a. Facultativo no da confirmación de baja: <ol style="list-style-type: none"> 1. El sistema vuelve a mostrar la información de la mascota, ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.54: Use Case: Dar de Baja Mascota

Use Case	Dar de Baja Proveedor
Summary	Un administrador registrado con sesión iniciada solicita dar de baja a un proveedor en el sistema.
Actor	Administrador
Precondition	Administrador registrado con sesión iniciada.
Postcondition	El proveedor es dado de baja en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Administrador pide dar de baja proveedor. 2. Sistema pide confirmación de baja. 3. Administrador da confirmación de la baja. 4. Sistema da de baja al proveedor, fin del caso de uso.
Branch Sequence	3a. Administrador no da confirmación de baja: <ol style="list-style-type: none"> 1. El sistema pide confirmación de cancelación. 2. Administrador cancela, ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.55: Use Case: Dar de Baja Proveedor

Use Case	Editar Consulta
Summary	Un facultativo registrado con sesión iniciada solicita editar una consulta de una mascota en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada, mascota registrada.
Postcondition	La consulta de la mascota es modificada en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Facultativo pide consultar consulta. 2. Sistema muestra las consultas. 3. Facultativo selecciona una consulta. 4. Sistema muestra la información de esa consulta. 5. Facultativo pide modificar consulta. 6. Sistema pide la información nueva para la consulta. 7. Facultativo introduce la información nueva. 8. Sistema guarda la consulta modificada, fin del caso de uso.
Branch Sequence	7a. Facultativo no introduce información de la consulta: <ol style="list-style-type: none"> 1. Sistema no guarda consulta modificada, ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.56: Use Case: Editar Consulta

Use Case	Editar Informe
Summary	Un facultativo registrado con sesión iniciada solicita editar un informe de una mascota en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada, mascota registrada.
Postcondition	El informe de la mascota es modificado en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Facultativo pide consultar informe. 2. Sistema muestra el informe. 3. Facultativo selecciona el informe a editar. 4. Sistema muestra la información del informe. 5. Facultativo modifica la información del informe. 6. Sistema guarda la modificación, fin del caso de uso.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.57: Use Case: Editar Informe

Use Case	Editar Intervención
Summary	Un facultativo registrado con sesión iniciada solicita editar una intervención de una mascota en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada, mascota registrada.
Postcondition	La intervención de la mascota es modificada en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Facultativo pide consultar informe. 2. Sistema muestra el informe con las intervenciones. 3. Facultativo pide consultar una intervención. 4. Sistema muestra información de la intervención. 5. Facultativo pide modificar la intervención. 6. Sistema pide información a modificar de la intervención. 7. Facultativo introduce la información modificada. 8. Sistema guarda la modificación, fin del caso de uso.
Branch Sequence	7a. Facultativo no introduce la información modificada. <ol style="list-style-type: none"> 1. El sistema no guarda la modificación , se vuelve al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.58: Use Case: Editar Intervención

Use Case	Editar Perfil
Summary	Un usuario registrado con sesión iniciada solicita editar su perfil en el sistema.
Actor	Usuario Registrado
Precondition	Usuario registrado con sesión iniciada.
Postcondition	Los cambios en el perfil del usuario son guardados en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Usuario pide editar perfil. 2. Sistema permite al usuario editar perfil. 3. Usuario guarda cambios. 4. Sistema pide confirmación de guardado. 5. Usuario confirma el guardado. 6. Sistema guarda cambios, fin del caso de uso.
Branch Sequence	4a. El usuario no confirma el guardado. <ol style="list-style-type: none"> 1. Sistema no cierra sesión, ir al paso 2.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.59: Use Case: Editar Perfil

Use Case	Editar Tratamiento
Summary	Un facultativo registrado con sesión iniciada solicita editar un tratamiento de una mascota en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada, mascota registrada.
Postcondition	El tratamiento de la mascota es modificado en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Facultativo pide consultar informe. 2. Sistema muestra el informe con los tratamientos. 3. Facultativo pide consultar un tratamiento. 4. Sistema muestra información del tratamiento. 5. Facultativo pide modificar el tratamiento. 6. Sistema pide información a modificar del tratamiento. 7. Facultativo introduce la información modificada. 8. Sistema guarda la modificación, fin del caso de uso.
Branch Sequence	<ol style="list-style-type: none"> 7a. El facultativo no introduce la información modificada. 1. Sistema no guarda la modificación, ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.60: Use Case: Editar Tratamiento

Use Case	Editar historial
Summary	Un facultativo registrado con sesión iniciada solicita editar el historial de una mascota en el sistema.
Actor	Facultativo
Precondition	Facultativo registrado con sesión iniciada, mascota registrada.
Postcondition	El historial de la mascota es modificado en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Facultativo pide consultar historial. 2. Sistema muestra el historial. 3. Facultativo pide editar historial. 4. Sistema pide información nueva del historial al facultativo. 5. Facultativo introduce la información nueva del historial en el sistema. 6. Sistema pide confirmación de la modificación. 7. Facultativo confirma. 8. Sistema guarda historial modificado, fin del caso de uso.
Branch Sequence	5a. Facultativo no introduce información del historial: <ol style="list-style-type: none"> 1. Sistema no guarda historial, ir al paso 1. 7a. Facultativo no confirma la modificación: <ol style="list-style-type: none"> 1. Ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.61: Use Case: Editar historial

Use Case	Eliminar Farmaco
Summary	Un administrador registrado con sesión iniciada solicita dar de baja un fármaco del sistema.
Actor	Administrador
Precondition	Administrador registrado con sesión iniciada.
Postcondition	El fármaco es eliminado del sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Administrador pide dar de baja fármaco. 2. Sistema pide confirmación de baja. 3. Administrador da confirmación de la baja. 4. Sistema da de baja al fármaco, fin del caso de uso.
Branch Sequence	3a. Administrador no da confirmación de baja: <ol style="list-style-type: none"> 1. El sistema pide confirmación de cancelación. 2. Administrador cancela, ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.62: Use Case: Eliminar Farmaco

Use Case	Eliminar Material
Summary	Un administrador registrado con sesión iniciada solicita dar de baja un material del sistema.
Actor	Administrador
Precondition	Administrador registrado con sesión iniciada.
Postcondition	El material es eliminado del sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Administrador pide dar de baja material. 2. Sistema pide confirmación de baja. 3. Administrador da confirmación de la baja. 4. Sistema da de baja al material, fin del caso de uso.
Branch Sequence	3a. Administrador no da confirmación de baja: <ol style="list-style-type: none"> 1. El sistema pide confirmación de cancelación. 2. Administrador cancela, ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.63: Use Case: Eliminar Material

Use Case	Iniciar Sesión
Summary	Un usuario registrado en el sistema intenta iniciar sesión proporcionando su correo y contraseña.
Actor	Usuario Registrado
Precondition	El usuario debe estar registrado previamente en el sistema.
Postcondition	El usuario inicia sesión correctamente en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Usuario registrado quiere iniciar sesión. 2. Sistema solicita correo y contraseña. 3. Usuario registrado introduce correo y contraseña. 4. Sistema verifica credenciales y finaliza el caso de uso.
Branch Sequence	3a. El usuario registrado cancela el inicio de sesión. <ol style="list-style-type: none"> 1. El sistema interrumpe el inicio de sesión, ir al paso 2. 4a. El sistema no verifica las credenciales: <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de error, ir al paso 2.
Exception Sequence	
Sub UseCase	Consultar Calendario, Dar de baja mascota, Registrar mascota, Dar de baja cliente, Consultar clientes, Consultar Mascotas
Note	

Cuadro 5.64: Use Case: Iniciar Sesión

Use Case	Registrar Facultativo
Summary	Un administrador registrado en el sistema solicita registrar un nuevo facultativo, proporcionando la información requerida.
Actor	Administrador
Precondition	El administrador debe tener una sesión iniciada en el sistema.
Postcondition	El nuevo facultativo queda registrado en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Administrador solicita registrar facultativo. 2. Sistema solicita información del facultativo. 3. Administrador proporciona la información requerida. 4. Sistema registra al facultativo y finaliza el caso de uso.
Branch Sequence	<ol style="list-style-type: none"> 3a. El administrador no proporciona la información requerida. 1. El sistema no registra al facultativo, ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.65: Use Case: Registrar Facultativo

Use Case	Registrar Mascota
Summary	Un facultativo registrado en el sistema solicita registrar una nueva mascota, proporcionando la información necesaria.
Actor	Facultativo
Precondition	El facultativo debe tener una sesión iniciada en el sistema.
Postcondition	La nueva mascota queda registrada en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Facultativo solicita registrar mascota. 2. Sistema solicita información de la mascota. 3. Facultativo proporciona la información requerida. 4. Sistema registra la mascota y finaliza el caso de uso.
Branch Sequence	<ol style="list-style-type: none"> 3a. El facultativo no proporciona la información requerida. 1. El sistema no registra la mascota, ir al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.66: Use Case: Registrar Mascota

Use Case	Registrarse
Summary	Un cliente solicita registrarse en el sistema.
Actor	Cliente
Precondition	No se requiere ninguna condición previa.
Postcondition	El cliente queda registrado en el sistema.
Base Sequence	<ol style="list-style-type: none"> 1. Cliente solicita registrarse. 2. Sistema solicita información al cliente. 3. Cliente proporciona la información requerida. 4. Sistema solicita confirmación de registro. 5. Cliente confirma el registro. 6. Sistema registra al cliente y finaliza el caso de uso.
Branch Sequence	<ol style="list-style-type: none"> 2a. El cliente no proporciona la información requerida <ol style="list-style-type: none"> 1. El sistema no interrumpirá el proceso y volverá al paso 1. 5a. El cliente no confirma el registro <ol style="list-style-type: none"> 1. El sistema solicita confirmación de cancelación. 2. Cliente cancela la operación, regresando al paso 1.
Exception Sequence	
Sub UseCase	
Note	

Cuadro 5.67: Use Case: Registrarse

5.5 Actores del sistema

AC-001	Usuario
Descripción	Actor abstracto que sirve como punto de entrada al sistema para todos los actores autenticados.
Comentarios	

Cuadro 5.68: AC-001 Usuario

AC-002	Cliente
Descripción	Usuario que accede al sistema para gestionar y consultar los datos de su mascota.
Comentarios	Contiene todas las funcionalidades del actor Usuario.

Cuadro 5.69: AC-002 Cliente

AC-003	Facultativo
Descripción	Usuario profesional o experto del sector veterinario que utiliza el sistema para la gestión de la clínica.
Comentarios	Contiene todas las funcionalidades del actor Usuario.

Cuadro 5.70: AC-003 Facultativo

AC-004	Administrador
Descripción	Usuario con privilegios especiales para gestionar a los facultativos y los recursos de la clínica.
Comentarios	

Cuadro 5.71: AC-004 Administrador

5.6 Modelo de clases

Todo proyecto, independiente de su naturaleza, requiere una estructura de datos que defina las entidades principales y sus relaciones. Para este proyecto nuestro diagrama de clases sera el siguiente. 5.2. Obsérvese que el diagrama puede ser dividido en tres grandes partes,que están relacionados entre si, mediante las citas.

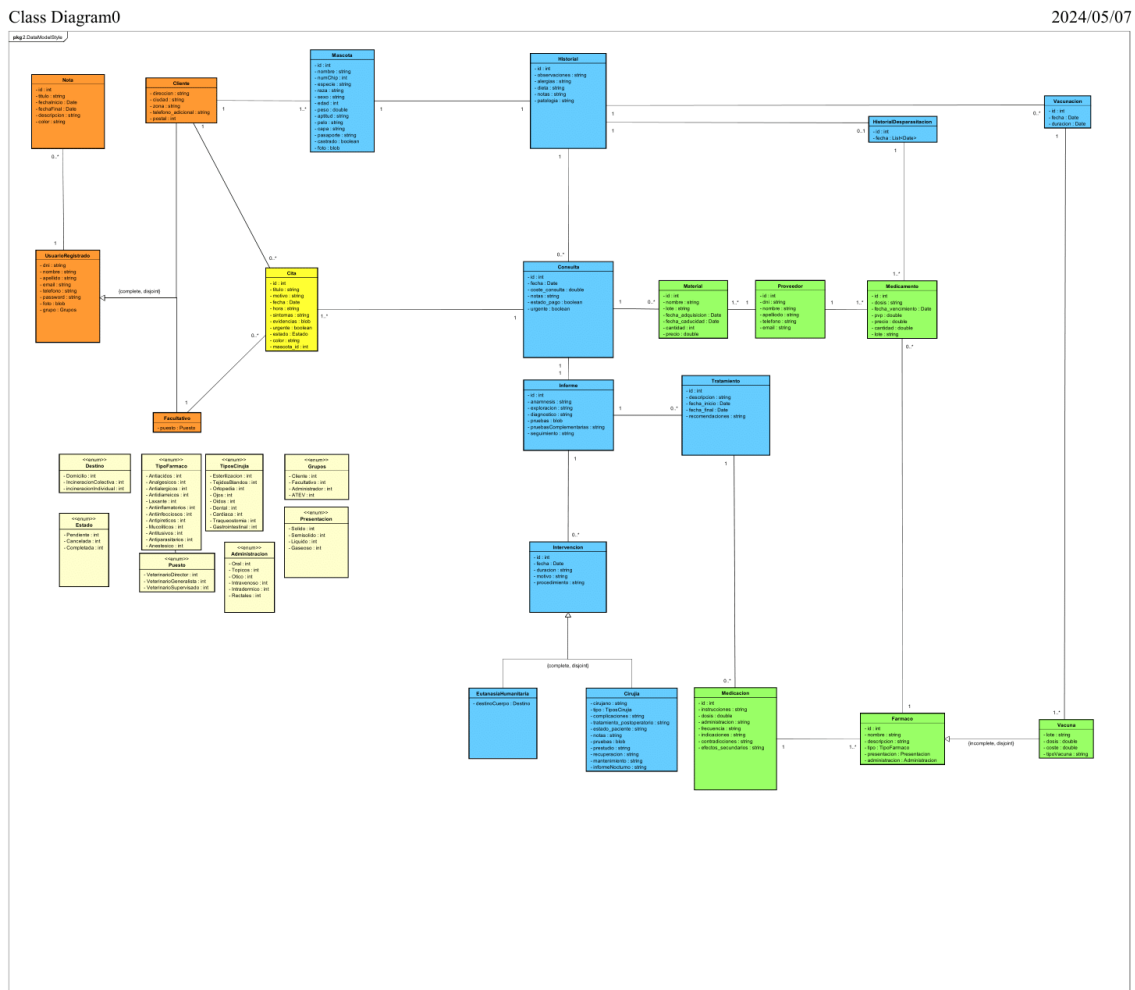


Figura 5.2: Diagrama de clases

5.6.1 Gestión de Usuarios

Para la gestión de usuarios, contamos con una clase general que se subdivide en dos categorías distintas. Cada una de estas categorías posee permisos diferentes dentro del sistema. Además esta sección contiene la clase **Cita**, que nos ayuda a enlazar la parte de gestión de los Usuarios, con la gestión de las mascotas en el sistema.

- **Usuario** Esta clase representa la abstracción general de cualquier individuo registrado en nuestro sistema. Contiene atributos y métodos comunes a todos los usuarios, como nombre de usuario, dirección de correo electrónico y contraseña.
- **Facultativo** Esta clase es una especialización de la clase **Usuario** y representa a profesionales veterinarios y expertos que pueden proporcionar servicios dentro de nuestra plataforma. Además de heredar los atributos y métodos de la clase **Usuario**, puede contener características adicionales específicas de los facultativos, como el puesto que ejercen dentro de la clínica.
- **Cliente** Esta clase también es una especialización de la clase **Usuario** y representa a los usuarios que buscan acceder a los servicios proporcionados por los facultativos en nuestra plataforma. Al igual que la clase **Facultativo**, hereda los atributos y métodos de la clase **Usuario**, y puede contener atributos adicionales como su dirección, código postal, teléfono adicional, etc.
- **Cita:** Esta clase representa las citas programadas entre los clientes y los facultativos. Actúa como un puente entre la gestión de usuarios y la gestión de mascotas, ya que las citas suelen estar relacionadas con la atención de una mascota en particular. Puede contener atributos como fecha, hora, motivo de la cita, referencias a los usuarios involucrados y las mascotas relacionadas.

5.6.2 Gestión de Mascotas

En esta sección del modelo, nos enfocamos en la gestión integral de las mascotas que están registradas en nuestro sistema. La clase **Cita**, mencionada anteriormente, también forma parte de esta sección, ya que sirve como el puente entre la gestión de usuarios y la gestión de mascotas al programar citas para dar consultas. Además de la clase **Cita**, esta sección incluye varias clases que abarcan aspectos específicos del cuidado y seguimiento de las mascotas, incluyendo:

- **Historial:** Esta clase registra el historial médico de cada mascota nada mas ser registrada en el sistema, esta clase es la base de la información clínica relacionada con la mascota, pues en ella están enlazadas todas las demás clases importantes de esta gestión.
- **HistorialDesparasitación:** Esta clase registra la información relacionada con el proceso de desparasitación de una mascota, para llevar un registro de las fechas en las que se ha desparasitado una mascota.
- **Vacunación:** Esta clase registra la información relacionada con el proceso de vacunación de una mascota, incluyendo la fecha y la duración de esta.
- **Consulta:** Representa las consultas médicas realizadas a cada mascota. Estas consultas están asignadas a citas específicas, programadas para una mascota en particular. Durante una consulta, se pueden utilizar diversos materiales y recursos médicos, lo que la relaciona con la sección de recursos mediante la clase **Material**.
- **Mascota:** Esta clase representa a cada una de las mascotas registradas en el sistema, incluyendo información como nombre, especie, raza, etc.
- **Informe:** Derivado de las consultas, este informe registra los detalles de cada visita médica de la mascotas.
- **Intervención:** Esta clase actúa como la generalización de las intervenciones médicas realizadas en una mascota, tales como cirugías o eutanasia. Proporciona una abstracción común para gestionar estas acciones médicas específicas.

- **Cirugía:** Esta clase representa la intervención quirúrgica realizada en una mascota, en ella se especifican todos los detalles de la cirugía realizada.
- **Eutanasia:** Esta clase representa el procedimiento de eutanasia aplicado a una mascota, en esta clase se encuentra el destino de la mascota después de su fallecimiento.
- **Tratamiento:** Representa los tratamientos médicos prescritos a las mascotas. Puede contener una clase **Medicacion**, que actúa como puente hacia la sección de recursos, facilitando el registro y acceso a los medicamentos utilizados.

5.6.3 Gestión de Recursos

Esta sección se centra en la administración y seguimiento de los recursos necesarios para llevar a cabo las diferentes actividades médicas y de atención a las mascotas en nuestro sistema. Incluye tanto los materiales utilizados durante las consultas y tratamientos, como los proveedores que suministran dichos materiales, así como los medicamentos y vacunas necesarios para el tratamiento.

- **Material:** Esta clase actúa como puente entre la gestión de mascotas y la gestión de recursos, representando los materiales y equipos utilizados durante las consultas médicas. Se relaciona con la clase **Consulta** de la sección anterior.
- **Medicación:** Similar a la clase **Material**, esta clase sirve como puente entre la gestión de mascotas y la gestión de recursos, representando los medicamentos prescritos a las mascotas durante su tratamiento. Se relaciona con la clase **Tratamiento** de la sección anterior.
- **Vacuna:** Esta clase es una especificación de la clase **Farmaco** que representa las vacunas utilizadas para prevenir enfermedades en las mascotas. Esta clase actúa como puente entre la gestión de mascotas y la gestión de recursos, relacionándose con la clase **Vacunación** de la sección anterior.
- **Proveedor:** Esta clase representa a los proveedores de materiales, medicamentos y vacunas utilizados en el cuidado de las mascotas. Contiene información como el nombre del proveedor, la dirección y los detalles de contacto.
- **Medicamento:** Representa los medicamentos disponibles para su uso en el tratamiento de las mascotas. Cada medicamento tiene una cantidad, un precio tanto de venta al público, como el precio que paga la clínica por el, la fecha de vencimiento de este, la dosis a suministrar y la cantidad de este en el inventario.
- **Fármaco:** Esta clase detalla la información básica y las propiedades de cada medicamento, proporcionando información específica sobre su administración, su tipo y su presentación.

Diseño

El desarrollo de cualquier aplicación comienza con la implementación de unas bases sólidas, similares a un conjunto de planos que guíen el proceso. En este proyecto, abarcaremos tanto el diseño de la interfaz como de la estructura de los datos. Por tanto, es crucial establecer un esquema detallado de la aplicación y definir los patrones que seguiremos durante su desarrollo.

6.1 Diseño

El patrón Model-View-Viewmodel de diseño se compone de tres componentes principales [17]:

- **Modelo:** Es responsable de la lógica de datos y la representación de los datos. Define la estructura de los datos y cómo interactuar con ellos, como las bases de datos.
- **Vista:** Se encarga de la presentación de los datos y la interfaz de usuario. Las vistas muestran la información al usuario final, utilizando los datos del modelo.
- **Controlador:** Actúa como un intermediario entre el modelo y la vista. Maneja la lógica de negocio y las interacciones del usuario, recibiendo entradas de la vista y actualizando el modelo en consecuencia.

En un proyecto que utiliza Angular en el front-end y Spring Boot en el back-end, se puede adoptar por esta arquitectura MVC que divide el proyecto en capas para una mejor organización, mantenimiento y escalabilidad.

6.1.1 Front-end (Angular)

En el front-end con Angular, la arquitectura MVC se refleja en la forma en que se estructura la aplicación:

- **Modelo:** Los modelos se pueden representar mediante interfaces o clases que definen la estructura de los datos con los que trabaja la aplicación. Estos modelos mantienen la consistencia de los datos y realizan validaciones.
- **Vista:** Las vistas son los componentes que muestran la interfaz de usuario. Los componentes manejan la lógica específica de cada vista, incluyendo la gestión de los datos del modelo para presentarlos correctamente al usuario.
- **Controlador:** El controlador se refleja en los componentes y sus controladores asociados. Los controladores de los componentes manejan la lógica de negocio, como responder a eventos del usuario, interactuar con servicios, o gestionar el estado de la aplicación.

6.1.2 Back-end (Spring Boot)

En el back-end con Spring Boot, se puede adoptar una arquitectura similar, siguiendo el patrón MVC:

- **Modelo:** Los modelos son clases Java que representan las entidades del sistema, usualmente con anotaciones de JPA para la persistencia de datos en bases de datos.
- **Vista:** La vista se define mediante los ficheros DTO, pues esta se centra en la presentación de los datos al usuario final, pero no esta involucrada en las actualizaciones dinámicas como lo esta en el front-end. Su responsabilidad principal es representar la información de manera adecuada para su consumo por parte del cliente.
- **Controlador:** Los controladores en Spring Boot son las clases que gestionan las solicitudes HTTP entrantes. Se encargan de manejar las peticiones de los clientes, procesar datos y devolver las respuestas apropiadas.

6.1.3 Comunicación entre Front-end y Back-end

La comunicación entre el front-end (Angular) y el back-end (Spring Boot) se realiza a través de una API REST. Spring Boot puede exponer sus servicios mediante controladores REST, mientras que Angular consume esos servicios mediante HTTPClient

6.1.4 Gestión de Seguridad

- **Autenticación:**
 - En el back-end, se ofrece el mecanismo para la autenticación, JWT (JSON Web Tokens). Spring Security se utiliza para gestionar la autenticación en el back-end, configurando un filtro de autenticación JWT para proteger los endpoints REST [13].
 - En el front-end, se almacena el token de autenticación recibido desde el back-end en el almacenamiento local. Al hacer peticiones al back-end, el front-end envía el token como parte de los encabezados de las solicitudes HTTP para autenticarse.
- **Autorización:**
 - En el back-end, Spring Security también maneja la autorización, permitiendo el acceso a qué recursos o endpoints a determinados usuarios según su rol [2].
 - En el front-end, se gestiona la autorización verificando el rol del usuario autenticado y mostrando u ocultando elementos de la interfaz de usuario según corresponda.

6.2 Diseño Interfaz

El diseño de la interfaz es un aspecto crítico en el desarrollo de cualquier aplicación. Es la cara visible con la que los usuarios interactúan y, por lo tanto, influye en gran medida en su experiencia y satisfacción. En esta sección, expondremos el diseño mediante mock-ups.

6.2.1 Interfaz de Portada

En esta pantalla, buscamos un diseño que nos ayude a exponer toda la información de presentación de la clínica y, al mismo tiempo, llame la atención de los usuarios a través de un diseño sencillo e intuitivo. Por ello, hemos optado por un enfoque visual más destacado y atractivo, como podemos observar 6.1.

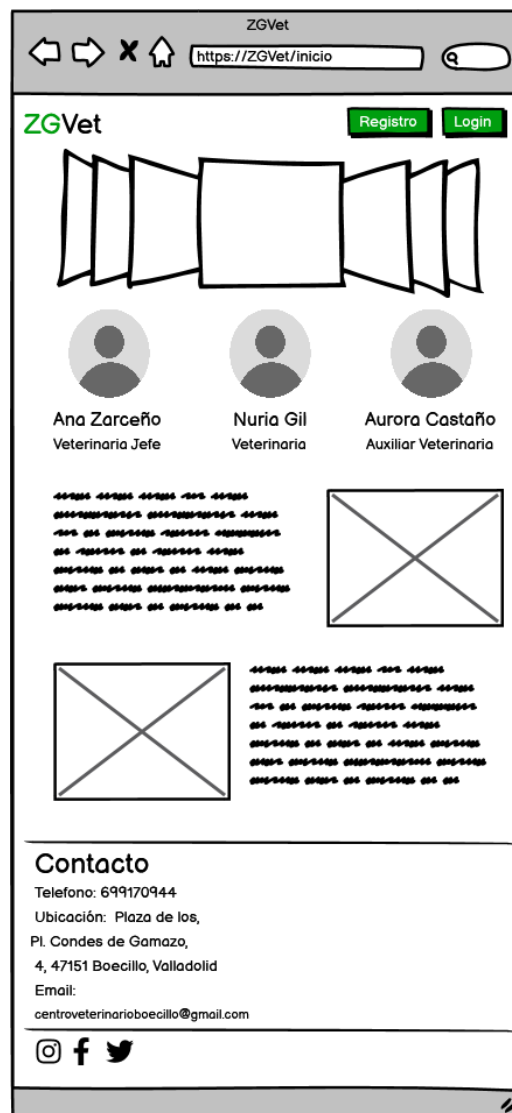


Figura 6.1: Interfaz de Portada

6.2.2 Interfaz de acceso al sistema

La interfaz de acceso al sistema es la pantalla con la que se encuentra el usuario para iniciar sesión en la aplicación. Su diseño debe ser claro, intuitivo y seguro. A continuación, visualizamos la interfaz de acceso en la Figura 6.2 y describimos los elementos principales de esta interfaz:



Figura 6.2: Interfaz de acceso

- **Campo de DNI Usuario:** Este campo permite al usuario ingresar su DNI asociado a su cuenta.
- **Campo de Contraseña:** Aquí el usuario puede ingresar su contraseña segura para acceder al sistema. Este campo oculta los caracteres ingresados para mayor seguridad. Además, advierte al usuario si la contraseña introducida es correcta o no.
- **Enlace de Recuperación de Contraseña:** Incluye un enlace para la recuperación de contraseña, proporcionando ayuda adicional al usuario.

Por otro lado, la interfaz de registro es la pantalla que permite a los clientes crear nuevas cuentas en el sistema. A continuación, visualizamos la interfaz de registro en la Figura 6.3. En esta etapa inicial, se han incluido los componentes de información básicos necesarios para el registro en el sistema. Una vez registrado, los usuarios tendrán la capacidad de editar estos atributos y agregar información adicional según sus necesidades y preferencias.

La interfaz de registro ha sido diseñada para proporcionar una experiencia intuitiva y fácil de usar para los usuarios que desean crear una cuenta en el sistema.



The image shows a web browser window titled "ZGVet" with the URL "https://ZGVet/registro". The page features the ZGVet logo and a "Login" button in the top right. Below the logo is a circular icon with the text "Creando Veterinarios Dedicados" and the word "Registro" in a large font. The registration form consists of several input fields: "Nombre" and "Apellido" (first and last name), "Telefono" and "Dni" (phone number and national ID), "Direccion" and "Codigo postal" (address and postal code), "Email", and "Foto" (photo). The "Foto" field has a placeholder image of a dog. At the bottom of the form are two buttons: "Registrar" (Register) and "Volver" (Back).

Figura 6.3: Interfaz de registro

6.2.3 Interfaz de calendario

Nada más iniciar sesión se mostrara la interfaz común de calendario donde se proporciona tanto al facultativo como al cliente sus citas programadas, y la opción de impartir o pedir citas, esta interfaz tiene un diseño sencillo, como se puede observar en la imagen 6.4, se visualiza el calendario y el nabvar de navegación, el cual según el rol que tenga en el sistema mostrara unos accesos u otros.



Figura 6.4: Interfaz de calendario

En esta interfaz los usuarios pueden añadir citas haciendo clic en el día deseado. Esto mostrara una pagina donde podrán introducir los detalles de la cita, como la fecha, la hora, el motivo y cualquier otra información relevante. Una vez completados los datos, tendrán la opción de enviar la cita.

Dependiendo de quién cree la cita, ésta se enviará al destinatario correspondiente para su revisión y confirmación. Si el usuario que crea la cita es un profesional (por ejemplo, un facultativo), la cita será enviada al cliente para que la acepte o la rechace. Por otro lado, si la cita es creada por un cliente, será enviada al profesional correspondiente para su aceptación.

Las notificaciones sobre nuevas citas se mostrarán a los usuarios a través de un icono de campana en el navbar. Cuando se reciba una nueva solicitud de cita, se generará automáticamente una notificación que aparecerá en la campana. Al hacer clic en la campana, los usuarios podrán ver todas las notificaciones pendientes y tomar las acciones necesarias, como aceptar o rechazar la cita, directamente desde allí.

6.2.4 Interfaz común

La interfaz común constituye el elemento central en la experiencia de usuario de la aplicación, proporcionando un entorno cohesivo y consistente para interactuar con las diversas funcionalidades. En esta sección, presentaré la interfaz para la pestaña de "Mascotas" de nuestra aplicación. Es importante destacar que esta interfaz servirá como plantilla para el diseño de todas las pestañas dentro de la aplicación, manteniendo una estructura coherente y consistente en todo el sistema.

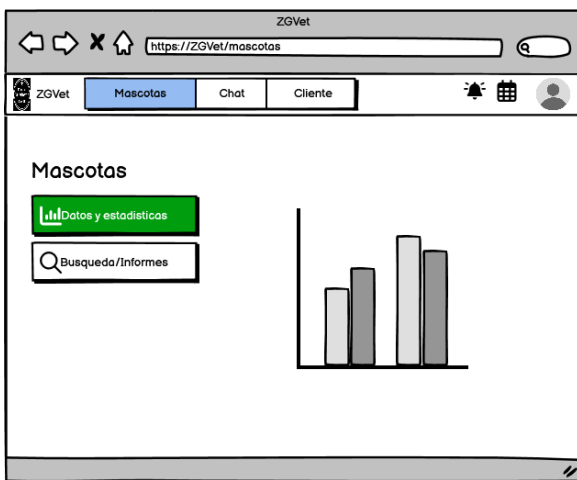


Figura 6.5: Interfaz de acceso

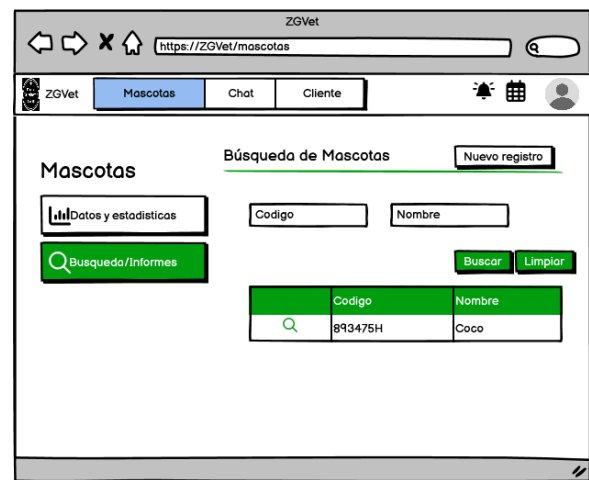


Figura 6.6: Interfaz de registro

La primera vista 6.5 se compone de un conjunto de gráficas que muestran las estadísticas más relevantes del apartado en el que nos encontremos. Estas visualizaciones ofrecen una visión rápida y concisa de los datos clave, permitiendo a los usuarios comprender de manera intuitiva el estado y el rendimiento de la sección en cuestión.

En la segunda vista 6.6, se presenta la funcionalidad de búsqueda de entidades, siguiendo una estructura de entrada de información, en cada sección, los parámetros de búsqueda pueden variar según los requerimientos específicos, proporcionando flexibilidad y adaptabilidad a las necesidades de los usuarios. Una vez que se realiza la búsqueda, los resultados se muestran en una tabla organizada, que presenta de manera ordenada y legible la información recuperada. Cada fila de la tabla representa una entidad encontrada que coincide con los criterios de búsqueda especificados. Además, se incluye un icono de lupa que permite a los usuarios acceder a información más detallada sobre cada entidad, permitiéndoles explorar más a fondo los detalles y atributos asociados. Desde una visión general de las estadísticas hasta la capacidad de buscar y exploración de entidades con más detalle, esta estructura ofrece una interfaz intuitiva y funcional para satisfacer las necesidades de los usuarios en cada paso del camino.

Implementación

7.1 Herramientas de Desarrollo

Para la implementación de este proyecto, se han utilizado diversas herramientas y tecnologías que han facilitado el desarrollo tanto del frontend como del backend. A continuación, se detallan las principales herramientas utilizadas y las razones por las que fueron elegidas:

En el desarrollo del frontend de la aplicación, se ha optado por el framework Angular [1] junto con TypeScript [22] como lenguaje de programación. Angular [1] ofrece una estructura robusta y modular para la construcción de aplicaciones web, permitiendo una organización clara del código y facilitando el desarrollo de interfaces de usuario interactivas y dinámicas. Además, TypeScript [22] añade un sistema de tipos estático que ayuda a detectar y prevenir errores durante el desarrollo, lo que resulta en un código más robusto y mantenible.

Para la implementación de la interfaz del calendario, se utilizó la librería FullCalendar [19]. Esta elección se basó en la amplia gama de funcionalidades que ofrece FullCalendar [19], que permite una visualización clara y eficiente de las citas programadas. Además, su integración con Angular [1] es sencilla y ofrece una gran flexibilidad para personalizar la apariencia y el comportamiento del calendario según las necesidades del proyecto.

En cuanto al backend de la aplicación, se decidió utilizar el framework Spring Boot [21] en conjunto con Hibernate como ORM [12] y MySQL [5] como sistema de gestión de bases de datos. Spring Boot [21] ofrece un entorno de desarrollo ágil y eficiente para la creación de servicios REST en Java, permitiendo una configuración mínima y simplificando el proceso de desarrollo y despliegue. Hibernate [12], por su parte, simplifica el acceso y la manipulación de los datos en la base de datos, proporcionando una capa de abstracción sobre las consultas SQL y facilitando la integración con el framework Spring. MySQL [5] fue elegido como sistema de gestión de bases de datos debido a su estabilidad y su fiabilidad en entornos de producción.

Además, se integró Docker [8] y Docker Compose [7] en el proceso de desarrollo para facilitar la gestión y el despliegue de la aplicación en diferentes entornos y máquinas. Docker [8] permite empaquetar la aplicación y sus dependencias en contenedores ligeros y portátiles, lo que simplifica la configuración del entorno de desarrollo y garantiza la consistencia entre diferentes máquinas. Docker Compose [7], por otro lado, facilita la gestión de aplicaciones multi-contenedor, permitiendo definir y ejecutar entornos de desarrollo complejos con múltiples servicios de manera sencilla y reproducible.

Estas herramientas han sido elegidas por su capacidad para proporcionar un entorno de desarrollo robusto, escalable y fácil de mantener, lo que permitió desarrollar una aplicación sólida y eficiente para la gestión de una clínica veterinaria.

Además de las herramientas mencionadas anteriormente, se utilizó GitHub [3] para el control de versiones y el almacenamiento del código fuente del proyecto. GitHub [3] ofrece una plataforma sólida y confiable para el almacenamiento y la colaboración en proyectos de desarrollo de software.

La integración con GitHub [3] también nos facilita el proceso de actualización de la máquina virtual (MV) donde está desplegada la aplicación. Pudiendo realizar cambios en el código, crear nuevas funcionalidades o corregir errores, y luego subir estos cambios a GitHub [3]. Desde allí, estos cambios pueden ser fácilmente descargados y desplegados en la MV mediante el uso de herramientas de integración continua o simplemente mediante la clonación del repositorio y la actualización manual del código en la MV.

Pruebas

El proceso de pruebas de la aplicación ha sido llevado a cabo de manera continua. Con el objetivo de garantizar la calidad del producto final, se ha establecido un calendario regular para la realización de pruebas. Cada semana, se ha realizado un despliegue a un entorno de preproducción específico para facilitar la evaluación exhaustiva de la aplicación.

Durante estas sesiones de prueba, tanto el experto que ha brindado orientación y asesoramiento durante todo el ciclo de desarrollo de la aplicación como a individuos externos al campo de veterinaria, han ido probando meticulosamente la aplicación. Esta inclusión de diferentes perspectivas ha mejorado el proceso de prueba al proporcionar una variedad de enfoques y opiniones.

Estos participantes han dedicado tiempo y esfuerzo a explorar las diversas funcionalidades de la aplicación, poniendo a prueba su usabilidad, rendimiento y robustez. Se les ha alentado a compartir sus comentarios, sugerencias y observaciones sobre la experiencia de uso, así como a identificar cualquier defecto o problema encontrado durante la interacción con la aplicación.

La recopilación y análisis de estos comentarios han sido esenciales para alimentar el ciclo de mejora continua del desarrollo. Cada observación se ha evaluado e incorporado al proceso de desarrollo de manera oportuna, con el fin de optimizar la calidad y la funcionalidad general de la aplicación.

Conclusiones

Durante la realización de este proyecto, hubiese deseado disponer de más tiempo para refinar y optimizar el código. Aunque lo desarrollado hasta ahora abarca los requisitos básicos solicitados por el veterinario, me hubiera gustado añadir más detalles y funcionalidades que habría facilitado la experiencia del usuario y mejorado la movilidad dentro de la aplicación. Sin embargo, estoy satisfecha con el logro de los objetivos básicos establecidos inicialmente.

El proyecto ha abarcado una amplia gama de funciones y secciones, todas ellas se han completado con éxito. Esto incluye la creación de una API REST en Spring, el desarrollo de la aplicación web utilizando Angular y Bootstrap, la conexión entre ambas instancias, la conversión de ambas en microservicios utilizando contenedores Docker y la implementación y despliegue continuo en la máquina virtual.

Además de cumplir con los requisitos, he logrado implementar funcionalidades adicionales que enriquecen la experiencia del usuario y agregan valor a la aplicación. Estas funcionalidades van más allá de lo básico solicitado por el veterinario, lo que demuestra mi compromiso con la mejora continua del proyecto.

Aunque enfrenté diversos desafíos durante el desarrollo, incluyendo limitaciones de tiempo y recursos, he aprendido a adaptarme y a superar obstáculos de manera efectiva. Estos desafíos han fortalecido mi habilidad para resolver problemas y tomar decisiones bajo presión.

Mirando hacia el futuro, tengo la intención de continuar mejorando y ampliando la aplicación. Planeo implementar nuevas características y funcionalidades que mejoren aún más la experiencia del usuario y la utilidad de la aplicación en el campo de la gestión veterinaria.

En conclusión, este proyecto no solo ha cumplido con los objetivos establecidos, sino que también ha sentado las bases para futuras mejoras y desarrollos. Espero que esta aplicación contribuya de manera significativa al campo de la gestión veterinaria.

9.1 Trabajo futuro

Este proyecto ofrece una base sólida sobre la cual se pueden implementar diversas mejoras y expansiones para enriquecer aún más la experiencia de los usuarios y maximizar el potencial de la aplicación. A continuación, se mencionan algunas áreas en las que se puede trabajar en el futuro:

- **Chat entre facultativos y clientes:** Incorporar un sistema de mensajería instantánea que permita la comunicación directa entre los profesionales veterinarios y los propietarios de mascotas. Esto facilitaría la resolución de dudas, la coordinación de citas y el seguimiento de tratamientos de manera más eficiente.
- **Foro para cada tipo de mascota:** Implementar foros de discusión dedicados a cada tipo de mascota, donde los usuarios puedan compartir experiencias, consejos y preguntas relacionadas con el cuidado y la salud de sus animales. Esto fomentaría la comunidad y el intercambio de conocimientos entre los propietarios y los profesionales.
- **Exportación de informes:** Agregar la funcionalidad de exportar informes y datos relevantes en diferentes formatos, como PDF o Excel. Esto permitiría a los usuarios generar informes personalizados para compartir con otros profesionales o para mantener un registro detallado de la salud y el bienestar de sus mascotas.
- **Añadir notas:** Agregar la funcionalidad de poder añadir notas al calendario, tanto individuales como compartidas.
- **Diseño para dispositivos móviles:** Adaptar la aplicación para que sea completamente compatible y funcional en dispositivos móviles, como teléfonos inteligentes y tabletas. Esto brindaría a los usuarios la flexibilidad de acceder y gestionar la información de sus mascotas desde cualquier lugar y en cualquier momento.

Estas son solo algunas de las posibles áreas de mejora y expansión que podrían explorarse en el futuro.

Apéndices

Apéndice A

Manual de Instalación

Gracias a la arquitectura dividida en microservicios y a la herramienta de Docker, el proceso de instalación de nuestro sistema se simplifica enormemente. Para comenzar, es necesario descargar e instalar Docker y Docker Compose siguiendo la documentación oficial de Docker [8].

Una vez instalados Docker y Docker Compose, encontraremos en el repositorio del proyecto el archivo **docker-compose.yml**, que contiene toda la definición de la infraestructura que Docker ejecutará.

En este directorio, abrimos una consola de terminal y ejecutamos el comando: **docker-compose build**

Este comando generará los contenedores Docker necesarios para el proyecto, utilizando la configuración definida en el archivo docker-compose.yml.

Una vez completada la construcción de los contenedores, simplemente iniciamos estos contenedores ejecutando el siguiente comando: **docker-compose up** Ahora, con los contenedores en ejecución, podemos acceder a la aplicación a través del navegador web. Basta con dirigirse a la dirección **http://localhost**, que nos llevará a la página principal de la aplicación web.

Para acceder a la API, simplemente visitamos la dirección **http://localhost/api**, puesto que Caddy, se encargará de encaminar el tráfico a la ubicación correspondiente dentro de nuestra aplicación.

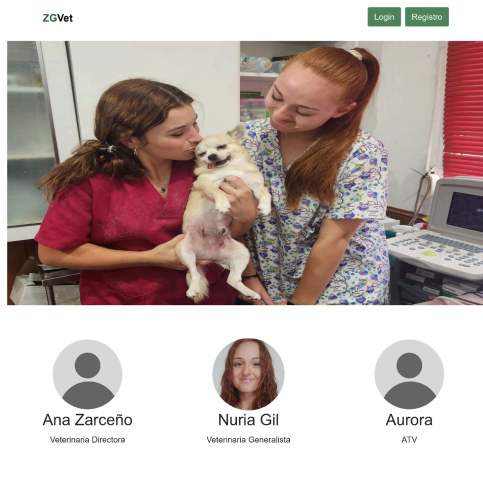
Este proceso de instalación proporciona una forma rápida y sencilla de poner en funcionamiento nuestro sistema, gracias a la potencia y flexibilidad de Docker y Docker Compose.

Apéndice B

Manual de Usuario

En esta sección se encuentran instrucciones claras y directas, así como consejos útiles, que te permitirán sacar el máximo provecho de proyecto. Este manual está pensado para que puedas guiarte en cualquier momento, para resolver dudas o encontrar soluciones rápidas a posibles problemas.

Una vez que accedes a la aplicación web, la primera interfaz que verás es común para todos los usuarios, es la página de información relacionada con la clínica. En esta página B.1, B.2, puedes encontrar información esencial sobre la clínica y sus servicios. Aquí tienes un resumen de lo que puedes ver y hacer en esta página:



En Boecillo, comprendemos que tus mascotas son más que solo animales, son parte de tu familia. Por eso, ofrecemos una amplia gama de servicios médicos, desde exámenes de rutina y vacunaciones hasta cirugía avanzada y cuidados intensivos



Figura B.1: Vista inicial



Contacto
Teléfono: 699170944
Ubicación: Plaza de los Pi, Condes de Gamazo, 4, 47151 Boecillo, Valladolid [📍](#)
Email: centroveterinarioboecillo@gmail.com
[f](#) [t](#) [@](#)

Figura B.2: Vista inicial

B.1 Contenido de la página de inicio de la clínica

- **Información sobre la Clínica:** La página presenta detalles sobre quiénes son y otra información relevante acerca de la clínica.
- **Datos de Contacto:** Puedes encontrar información de contacto esencial al final de de esta, como números de teléfono, direcciones de correo electrónico y cualquier otra forma de contacto directo con la clínica.
- **Ubicación:** Se muestra la dirección física de la clínica, incluyendo un mapa mediante un icono de ubicación para llegar a la clínica desde tu ubicación actual.
- **Opciones de Inicio de Sesión o Registro:** En la esquina superior derecha de la página, encontrarás opciones para iniciar sesión o registrarte. Estas opciones te permitirán acceder a las funciones de la aplicación una vez que hayas ingresado tus credenciales o creado una nueva cuenta.
 - **Inicio de Sesión:** Si ya tienes una cuenta, haz clic en el botón de inicio de sesión e ingresa tus credenciales (DNI de usuario y contraseña) para acceder a tu cuenta B.3.
 - **Registro:** Si no tienes una cuenta y eres un cliente, puedes hacer clic en el botón de registro para crear una nueva cuenta. Se te solicitará que proporciones información básica como tu nombre, dirección de correo electrónico y una contraseña segura. Si no tienes cuenta y te vas a inscribir como facultativo, tendrás que pedirle al administrador que te de de alta en la aplicación. B.4.

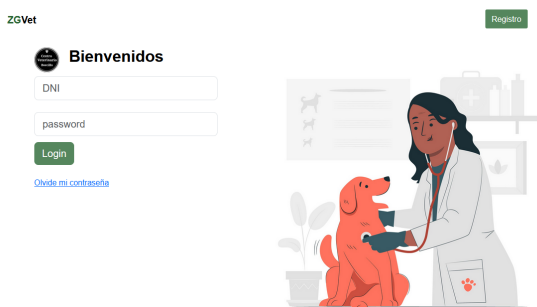


Figura B.3: Vista login

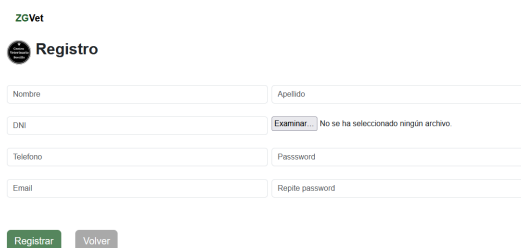


Figura B.4: Vista registro

Una vez que hayas sido registrado en la aplicación web y te hayas autenticado con éxito, dependiendo de tu rango de autorización, accederás a una página u otra.

B.2 Rangos de Permiso y Opciones Disponibles

B.2.1 Rango de Permiso: Cliente

La primera vista para un cliente es la de calendario, que muestra eventos y actividades relacionados con la clínica y con tu perfil , tendrá las opciones de viajar a las paginas de mascotas, a su perfil y a la portada clicando en el icono ZGVet , además de tener el acceso a las siguientes opciones en la página de calendario B.6:

Contenido de la página de calendario de la clínica

- **Ver Citas Programadas:** Puedes visualizar las citas programadas en el calendario, incluyendo fecha, hora, y médico asignado.
- **Solicitar Nuevas Citas:** Puedes solicitar nuevas citas médicas clicando en el día que quieras la cita,te aparecerá un formulario B.5 donde seleccionando un médico y una fecha y hora disponibles podrás enviar tu propuesta de cita al facultativo seleccionado.

The screenshot shows the ZGVet user interface for creating an appointment. At the top, there are navigation links for 'ZGVet', 'Mascotas', and 'Chat'. On the right side, there are icons for a notification bell, a calendar showing the date '19', and a user profile icon. The main form area includes a dark square icon, a text input field for 'Hora Inicio', and a checkbox labeled 'Urgente'. Below these are four stacked text input fields for 'Titulo', 'Motivo', and 'Síntomas'. A dropdown menu labeled 'Selecciona Receptor' is positioned below the 'Síntomas' field. At the bottom of the form, there is a section for 'Evidencias' with a button labeled 'Examinar...' and a message stating 'No se ha seleccionado ningún archivo.'. Finally, there are two buttons at the bottom: a green 'Guardar' button and a grey 'Cancelar' button.

Figura B.5: Interfaz de cita cliente

- **Ver tus notificaciones:** Puedes ver las notificaciones de citas clicando en el icono de la campana, y aceptar la cita o denegarla, al seleccionar esta última opción la cita se volverá a enviar al facultativo.

ZGVet Mascotas

Calendario
abril de 2024

Mes Semana Día
Hoy < >

lun	mar	mié	jue	vie
1	2	3	4	5
8	9	10	11	12
15	16	17	18	19
22	23	24	25	26
29	30	1	2	3
6	7	8	9	10

Vacunacion

Figura B.6: Interfaz de calendario cliente

Contenido de la página de mascotas

Cuando selecciones la opción **Mascotas** en la barra de navegación superior, serás dirigido a una página con un buscador que te permitirá buscar las mascotas asociadas a tu perfil. Después de realizar una búsqueda, los datos de las mascotas aparecerán en una tabla B.7. Al hacer clic en el icono de lupa junto a cada mascota en la tabla, podrás ver más detalles sobre esa mascota B.8.

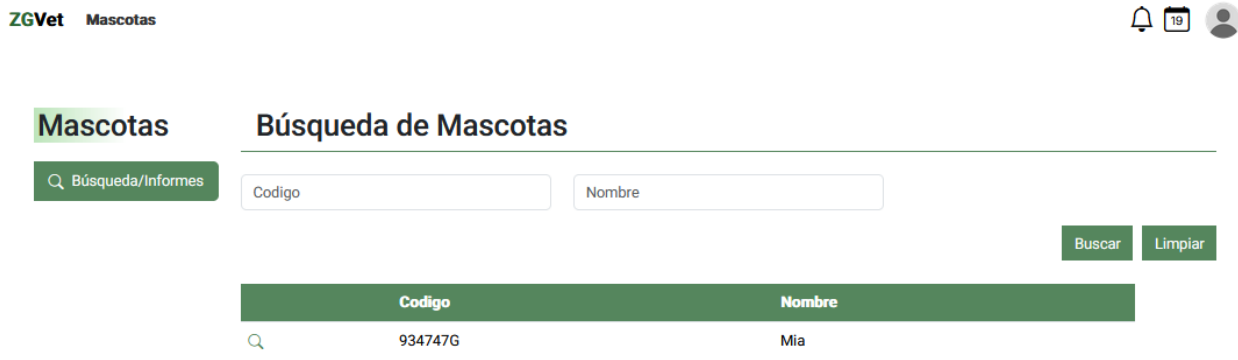


Figura B.7: Interfaz de mascotas cliente

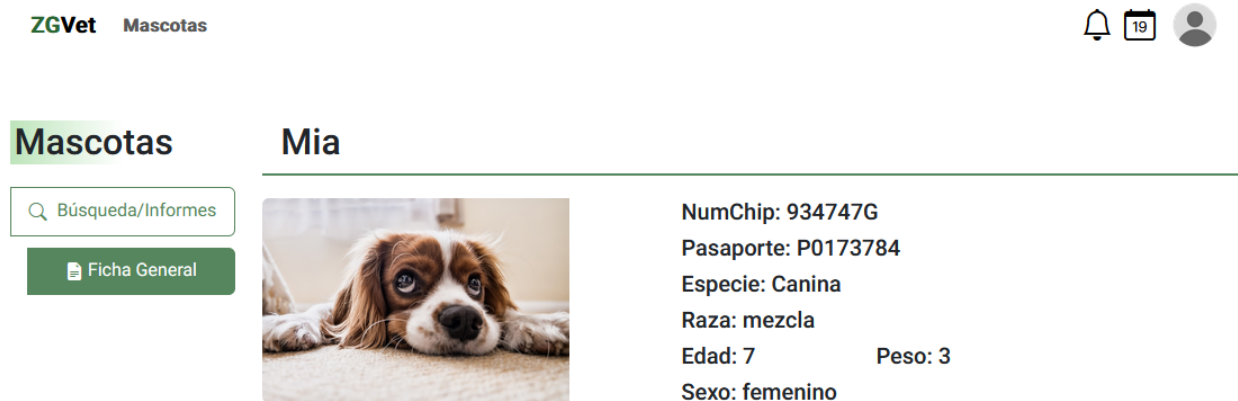


Figura B.8: Interfaz de mascotas al detalle

Contenido de la página de perfil

Para acceder a esta pagina debes hacer clic en la imagen del usuario en la parte superior de la página, al hacer resto aparecerá un menú desplegable con las siguientes opciones:

- **Perfil:** Al seleccionar esta opción, serás redirigido a la página de perfil B.9. En esta página, podrás ver tus datos personales. Además, tendrás la opción de editar tus datos haciendo clic en el icono de lápiz que aparecerá en la esquina superior derecha B.10.



Valentin Cardeñoso

valentinCarde@gmail.com

Telefono: 653494736

Figura B.9: Interfaz de perfil del cliente

Editar Perfil

 No se ha seleccionado ningún archivo.

Figura B.10: Interfaz de edición perfil del cliente

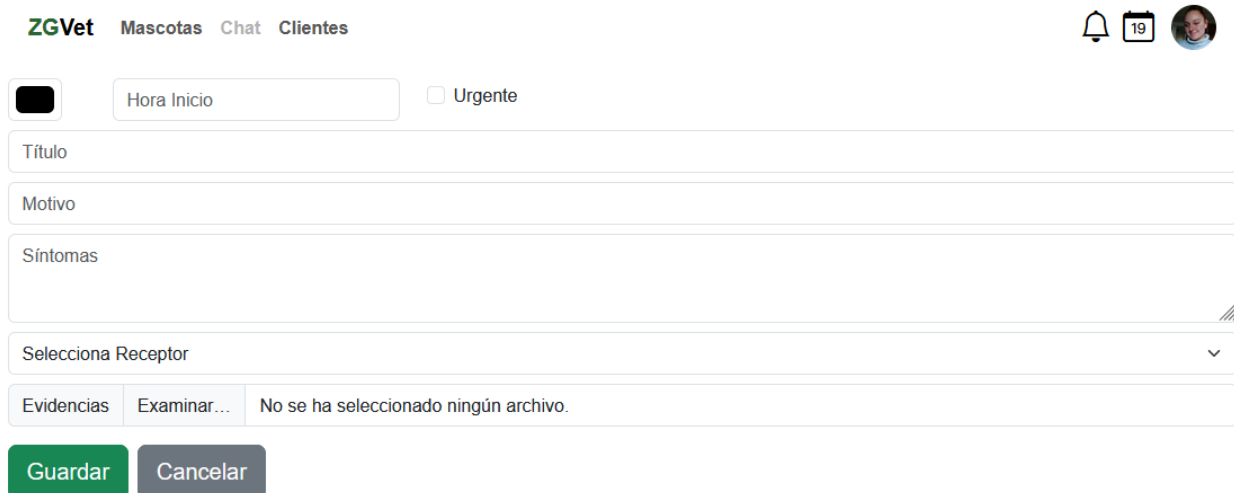
- **Logout:** Al seleccionar esta opción, cerrarás sesión en la aplicación. Se eliminarán tus credenciales de acceso y serás redirigido a la página de inicio.

B.2.2 Rango de Permiso: Facultativo

La primera vista para un facultativo es la de calendario, que muestra eventos y actividades relacionados con la clínica y con tu perfil, además tendrá las opciones de viajar a las paginas de mascotas , a la de clientes, a su perfil y a la portada clicando en el icono ZGVet además de tener el acceso a las siguientes opciones en la página de calendario B.12:

Contenido de la página de calendario de la clínica

- **Ver Citas Programadas:** Puedes visualizar las citas programadas en el calendario, incluyendo fecha, hora, y médico asignado.
- **Solicitar Nuevas Citas:** Puedes solicitar nuevas citas médicas clicando en el día que quieras la cita, te aparecerá un formulario B.11 donde seleccionando a un cliente y una fecha y hora disponibles podrás enviar tu propuesta de cita al cliente seleccionado.



The screenshot shows the ZGVet user interface for creating a new appointment. At the top, there are navigation links for 'Mascotas', 'Chat', and 'Clientes', along with a notification bell icon, a calendar icon showing '19', and a user profile picture. The form includes a 'Hora Inicio' input field with a time selection icon, an 'Urgente' checkbox, and text input fields for 'Título', 'Motivo', and 'Síntomas'. Below these is a 'Selecciona Receptor' dropdown menu. At the bottom, there is an 'Evidencias' section with an 'Examinar...' button and a message 'No se ha seleccionado ningún archivo.' Two buttons, 'Guardar' (green) and 'Cancelar' (grey), are positioned at the bottom left of the form.

Figura B.11: Interfaz de cita facultativo

- **Ver tus notificaciones:** Puedes ver las notificaciones de citas clicando en el icono de la campana, y aceptar la cita o denegarla.

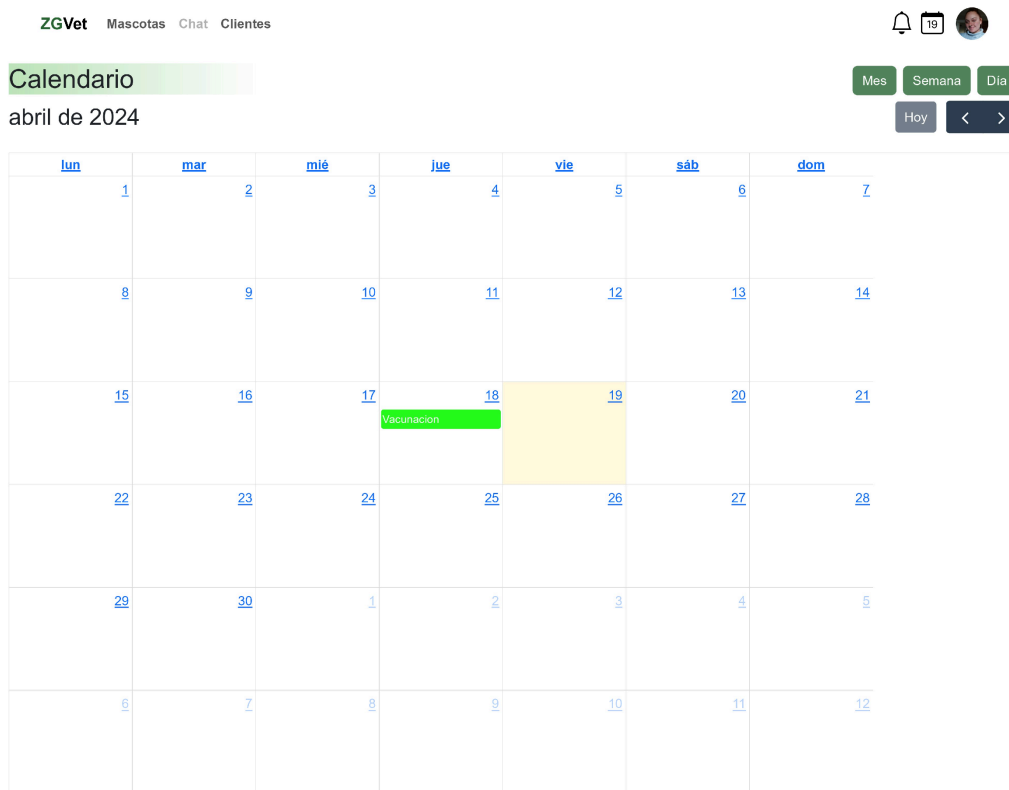


Figura B.12: Interfaz de calendario facultativo

Contenido de la página de clientes

Cuando selecciones la opción **Cientes** en la barra de navegación superior, serás dirigido a una página en la que se muestra un buscador B.13. En esta página, podrás realizar búsquedas de los clientes registrados en la app, y al pulsar en el icono de la lupa podrás ver en detalle sus datos B.14, además de poseer dar de baja al cliente, como se puede observar en la imagen:

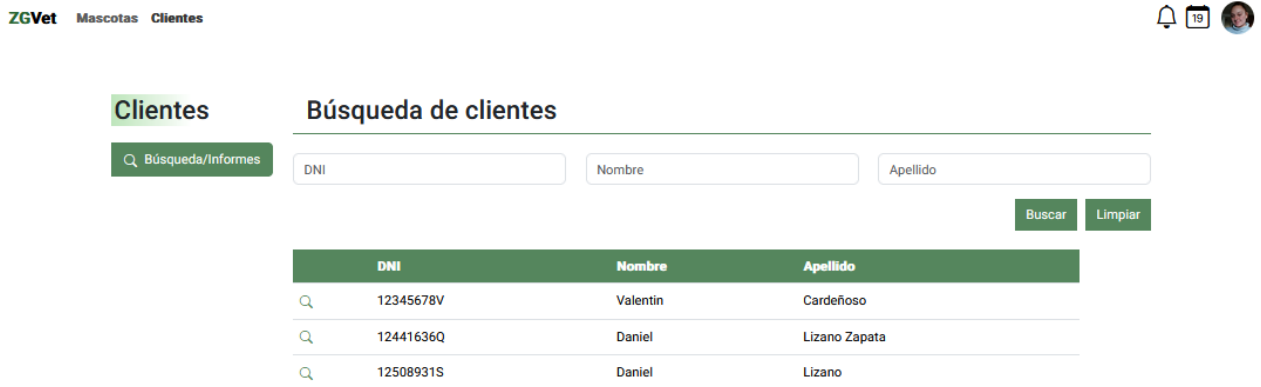


Figura B.13: Interfaz de clientes facultativo



Figura B.14: Interfaz de cliente en detalle facultativo

Contenido de la página de mascotas

Cuando selecciones la opción **Mascotas** en la barra de navegación superior, serás dirigido a una página con una serie de gráficos que muestran estadísticas sobre los datos de las mascotas. En esta página, podrás observar información relevante sobre las mascotas, como se puede observar en la imagen:



Figura B.15: Interfaz de estadísticas de las mascotas

Además de visualizar las estadísticas de mascotas, tienes la opción de buscar mascotas utilizando una función de búsqueda similar a la que tienen los clientes, con la diferencia de que podrás buscar entre todas las mascotas registradas en el sistema B.16.

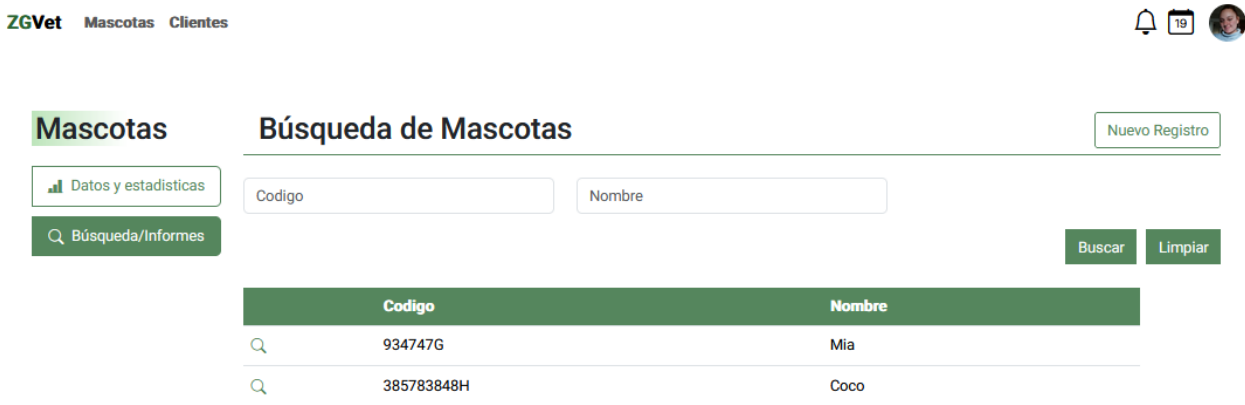


Figura B.16: Interfaz de búsqueda de mascotas

Cuando realizas una búsqueda de mascotas, puedes ver más detalles sobre una mascota específica haciendo clic en el icono de lupa que aparece junto a ella en los resultados de la búsqueda.

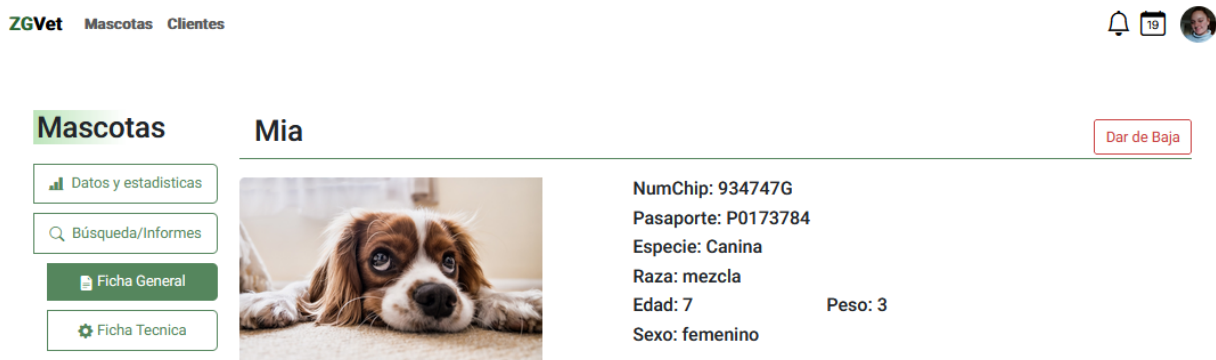


Figura B.17: Interfaz de detalle de mascotas

Una vez que accedas a los detalles de una mascota, tendrás varias opciones adicionales disponibles para gestionar su información, al hacer clic en ficha técnica. Al acceder a la ficha técnica de una mascota, la primera página que se visualizará será la del historial de la mascota. Esta página muestra información detallada sobre el historial médico, que podrás editar clicando en los pinceles, y las pestañas para dirigirse a las actividades anteriores de la mascota.

ZGVet Mascotas Clientes

Mascotas

Historial Consulta Vacunación Desparasitación

Datos y estadísticas

Búsqueda/Informes

Ficha General

Ficha Técnica

Observaciones

Presenta un pelaje brillante y sin signos de pérdida excesiva de pelo. Se ha observado una leve cojera en la pata trasera derecha durante los últimos días, por lo que se recomienda una evaluación adicional en la próxima visita médica

Alergias

No presenta alergias conocidas a alimentos, plantas o medicamentos. No se han observado signos de reacciones alérgicas durante las visitas médicas anteriores.

Dieta

Sin dietas

Patología

Notas

Figura B.18: Interfaz de historial de mascota

Sección de consultas Al dirigirse a la sección de consultas, se mostrará un buscador de consultas que te permite filtrar y buscar consultas específicas.

ZGVet Mascotas Clientes

Mascotas

Historial Consulta Vacunación Desparasitación

Datos y estadísticas

Búsqueda/Informes

Ficha General

Ficha Técnica

dd / mm / aaaa

Buscar Limpiar

Fecha
2024-04-18

Figura B.19: Interfaz de consultas

Ver detalles de una consulta

- Una vez que hayas buscado y encontrado la consulta que deseas ver, haz clic en el icono de lupa junto a la consulta para ver más detalles.
- Al seleccionar una consulta específica, se mostrará toda la información relacionada con esa consulta, incluyendo datos como la fecha, el estado de la consulta, el coste y notas.
- Además, tendrás la posibilidad de editar la información de la consulta utilizando el icono de lápiz de edición. Esto te permite realizar modificaciones en los datos de la consulta según sea necesario.

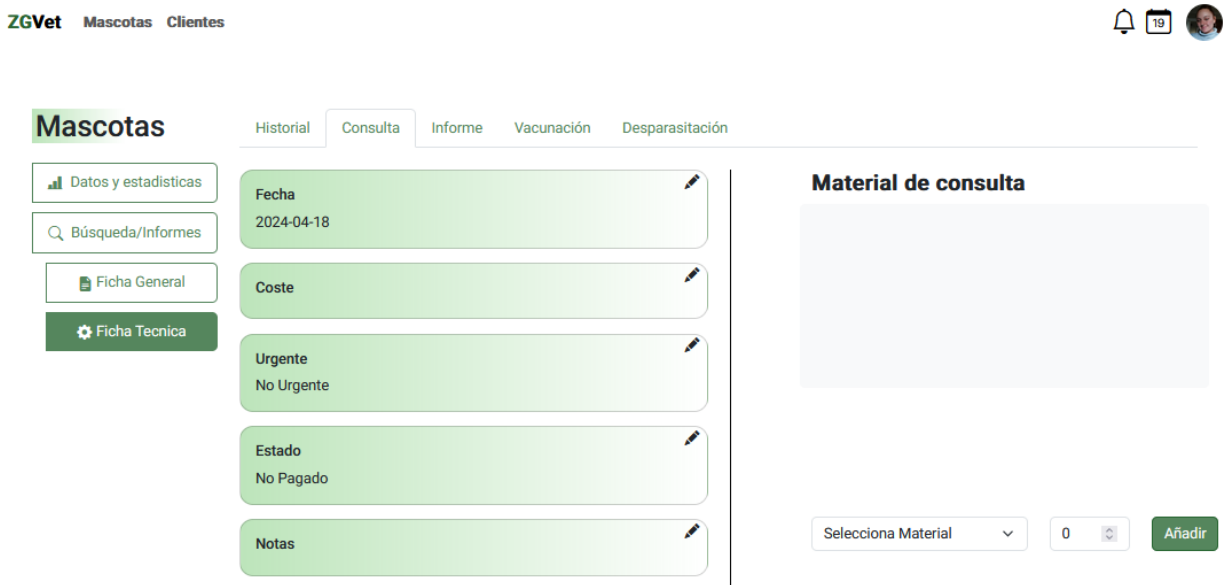


Figura B.20: Interfaz de detalles de consulta

Gestión de materiales utilizados en la consulta

- Junto con los detalles de la consulta, se mostrará una lista con el material utilizado durante la consulta. Puedes revisar esta lista para obtener información sobre los recursos empleados.
- También tendrás la opción de añadir más material a la consulta si es necesario, utilizando las funciones de edición disponibles.



Figura B.21: Interfaz de material de consulta

Informe de la consulta

- En la sección de consulta, se presenta una pestaña oculta que contiene el informe de la consulta. Puedes acceder a esta pestaña para ver el informe detallado de la consulta seleccionada.

-Podrás editar estos campos mediante el icono del lápiz, cuando sea necesario, además de poder añadir intervenciones o tratamientos.

Figura B.22: Interfaz de informe de consulta

Intervención de la consulta Al seleccionar la opción para añadir una intervención, serás dirigido a la vista correspondiente donde podrás ingresar la información sobre la intervención que deseas añadir.

Figura B.23: Interfaz de intervención de consulta

Tratamiento de la consulta Al seleccionar la opción para añadir un tratamiento, serás dirigido a la vista correspondiente donde podrás ingresar la información sobre la intervención que deseas añadir.

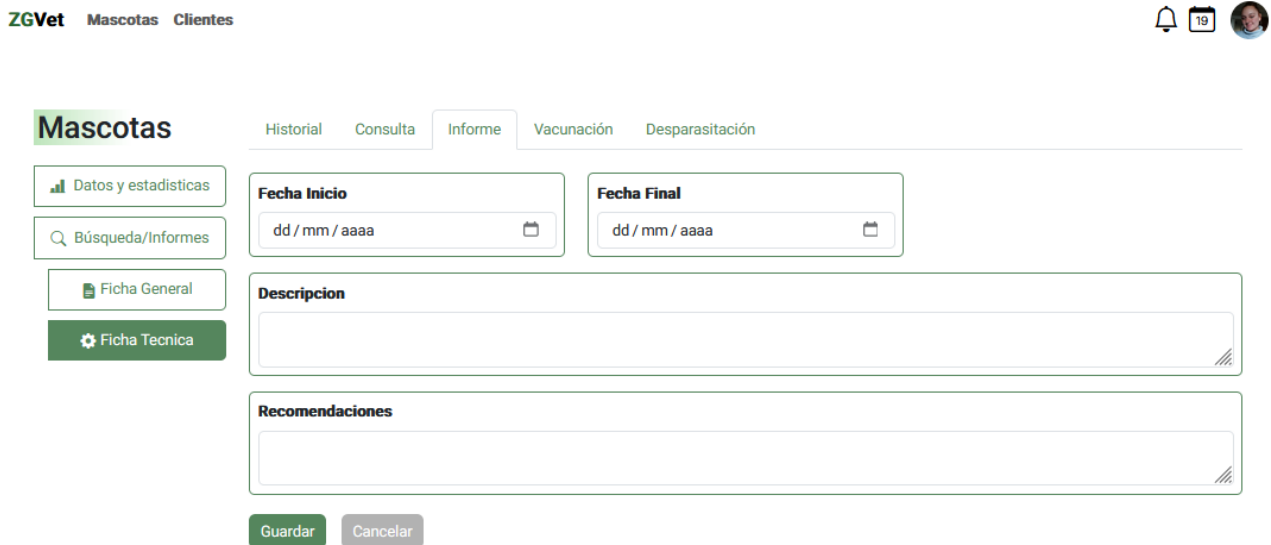


Figura B.24: Interfaz de tratamiento de consulta

Una vez que hayas guardado un tratamiento, puedes añadir la medicación necesaria para dicho tratamiento, pulsando en el botón añadir medicación. Para añadir datos de la medicación a un tratamiento, sigue los pasos que se indican a continuación: 1. En la vista de tratamiento, busca la sección para añadir medicación.

2. Utiliza el menú desplegable (**select**) para elegir el medicamento que deseas añadir a la medicación. El menú mostrará una lista de medicamentos disponibles.

3. Selecciona el medicamento deseado de la lista.

4. Pulsar botón añadir, y el medicamento se añadirá a la lista.

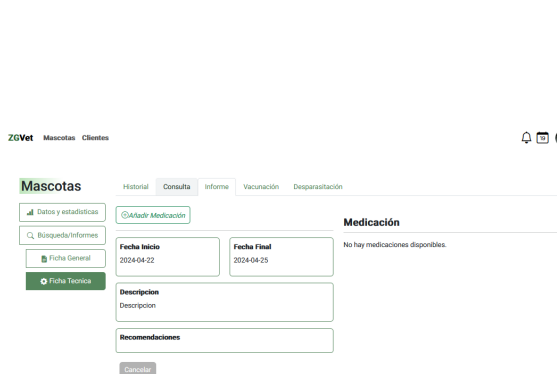


Figura B.25: Interfaz de tratamiento

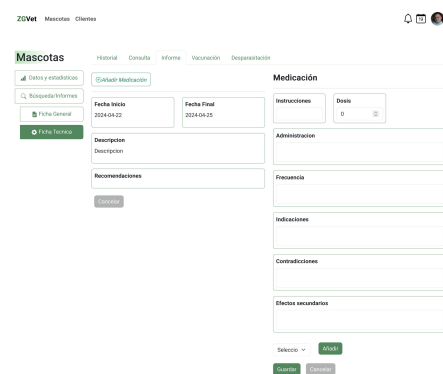


Figura B.26: Interfaz de tratamiento medicación

Sección de vacunación En la sección de vacunación, puedes añadir la información sobre las vacunaciones de la mascota. Cuando registres una vacunación para la mascota, la información se añadirá a una tabla que muestra un registro visual de todas las vacunaciones anteriores de la mascota. Esta tabla facilita el seguimiento de las vacunas administradas y programadas.



Figura B.27: Interfaz de vacunación de mascota

Sección de desparasitación Cuando registres una desparasitación para la mascota, la información se añadirá a una tabla que muestra un registro visual de todas las fechas de las desparasitaciones anteriores de la mascota. Esta tabla facilita el seguimiento de los tratamientos de desparasitación administrados y programados.



Figura B.28: Interfaz de desparasitación de mascota

Contenido de la página de perfil

Para acceder a esta página debes hacer clic en la imagen del usuario en la parte superior de la página, al hacer esto aparecerá un menú desplegable con las siguientes opciones:

- **Perfil:** Al seleccionar esta opción, serás redirigido a la página de perfil B.29. En esta página, podrás ver tus datos personales. Además, tendrás la opción de editar tus datos haciendo clic en el icono de lápiz que aparecerá en la esquina superior derecha B.30.



Nuria Gil
 nuriagr@gmail.com
 Telefono: 654392846
 Cargo: VeterinarioGeneralista

Figura B.29: Interfaz de perfil del facultativo

Editar Perfil

No se ha seleccionado ningún archivo.

Figura B.30: Interfaz de edición perfil del facultativo

- **Logout:** Al seleccionar esta opción, cerrarás sesión en la aplicación. Se eliminarán tus credenciales de acceso y serás redirigido a la página de inicio.

B.2.3 Rango de Permiso: Administrador

La primera página para un administrador es la de consulta de facultativos, que muestra los facultativos y además tendrá las opciones de viajar a las páginas de inventario y a la portada clicando en el icono ZGVet.

Contenido de la página consulta de facultativos

En esta pagina se muestran los facultativos también se puede buscarlos por nombre,DNI o apellido,además de tener el acceso a las siguientes opciones de la página B.31:

ZGVet Facultativos Inventario

Facultativos Búsqueda de Facultativos Nuevo Registro

Búsqueda/Informes

DNI Nombre Apellido

Buscar Limpiar

	DNI	Nombre	Apellido
Q	14508930J	Valentin	Cardeñoso
Q	71232946Z	Nuria	Gil

Figura B.31: Interfaz de consulta de facultativos

- **Nuevo registro:** Puede dar de alta a los nuevos facultativos que va a formar parte de la clínica mediante un formulario, mediante el botón "Nuevo registro".
- **Ver detalles del facultativo:** Puede ver los datos de un facultativo gracias al icono de la lupa que le mostrara los datos de ese facultativo B.32.
- **Dar de baja al facultativo:** Puede dar de baja a un facultativo, seleccionado a un facultativo, mediante el botón "Dar de baja" B.32

ZGVet Facultativos Inventario

Facultativos Valentin Cardeñoso Dar de Baja

Búsqueda/Informes

Ficha General

DNI: 14508930J
Email: valentin@gmail.com
Telefono: 653474923
Cargo: VeterinarioDirector

Figura B.32: Interfaz de datos de facultativos

Contenido de la página inventario

La sección de inventario proporciona una visión general y detallada de los recursos disponibles en el sistema, incluyendo materiales, fármacos y proveedores.

Al acceder a la sección de inventario, la primera vista que se presenta es una serie de gráficos que muestran datos relevantes sobre el inventario. Estos gráficos pueden incluir información sobre el stock de materiales, fármacos y proveedores, ayudando a tener una comprensión visual de la situación actual del inventario.

Búsqueda de materiales

1. Al hacer clic en la opción de búsqueda, primero se mostrará la sección de materiales.
2. En esta sección, encontrarás un filtro de búsqueda que te permite buscar materiales específicos según diversos criterios.
3. Los resultados de la búsqueda mostrarán una lista de materiales registrados.

The screenshot shows the ZGVet 'Inventario' page. At the top left, the navigation bar includes 'ZGVet', 'Facultativos', and 'Inventario'. On the right, there is a notification bell icon and a green circle with the letter 'A'. Below the navigation, the 'Inventario' section has three tabs: 'Materiales' (selected), 'Farmacos', and 'Proveedores'. On the left side of the search area, there are two buttons: 'Datos y estadísticas' (with a bar chart icon) and 'Búsqueda/Informes' (with a magnifying glass icon). On the right side, there is a 'Nuevo Registro' button. The search area contains two input fields: 'Nombre' and 'Lote'. At the bottom right of the search area, there are two buttons: 'Buscar' and 'Limpiar'.

Figura B.33: Interfaz de datos de materiales

The screenshot shows the ZGVet 'Inventario' page for data entry. The navigation bar is the same as in Figure B.33. The 'Inventario' section has the same three tabs: 'Materiales', 'Farmacos', and 'Proveedores'. On the left side, there are two buttons: 'Datos y estadísticas' and 'Búsqueda/Informes'. The main area contains several input fields: 'Nombre' (with a magnifying glass icon), 'Numero de lote' (with a magnifying glass icon), 'Fecha adquisicion' (with a date format 'dd / mm / aaaa' and a calendar icon), 'Fecha caducidad' (with a date format 'dd / mm / aaaa' and a calendar icon), 'Cantidad' (with a magnifying glass icon), 'Precio' (with a magnifying glass icon), and a dropdown menu labeled 'Selecciona Proveedor'. At the bottom, there are two buttons: 'Guardar' and 'Volver'.

Figura B.34: Interfaz de datos de materiales (2)

4. Puedes ver más detalles sobre un material específico haciendo clic en el icono de lupa junto a él.
5. Además, tienes la opción de dar de baja un material si ya no es necesario.

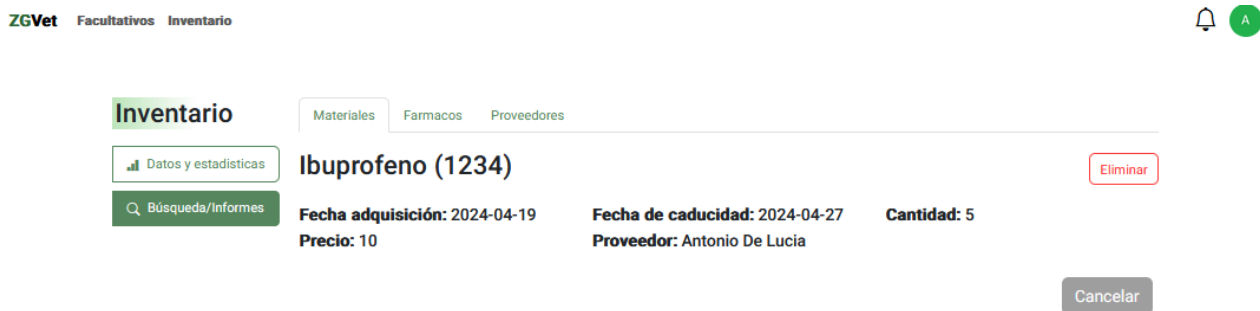


Figura B.35: Interfaz de datos de materiales

Búsqueda de fármacos

1. En la pestaña de fármacos, la funcionalidad es similar a la de materiales: puedes buscar fármacos registrados utilizando un filtro de búsqueda.
2. Los resultados mostrarán una lista de fármacos registrados.



Figura B.36: Interfaz de datos de fármacos

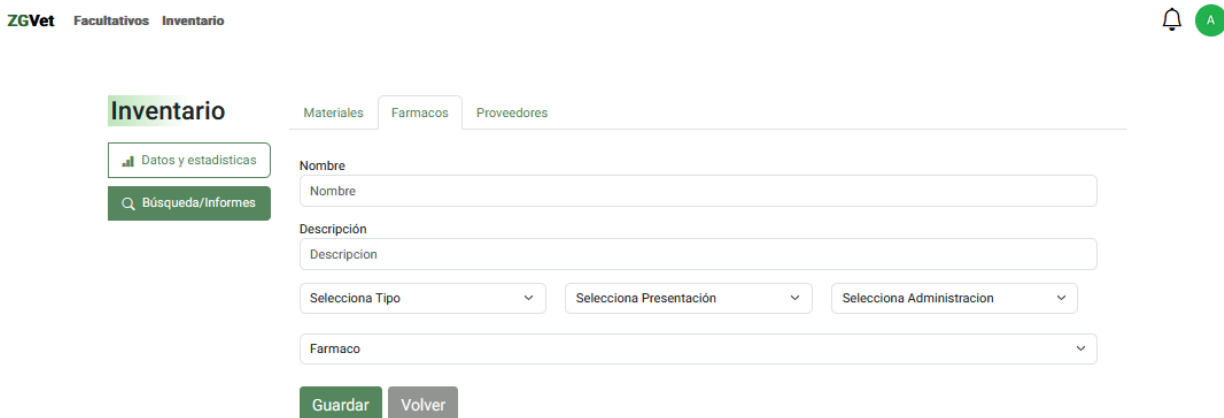


Figura B.37: Interfaz de datos de fármacos

3. Puedes ver más detalles sobre un fármaco específico haciendo clic en el icono de lupa junto a él.
4. Al guardar un nuevo fármaco, tienes la posibilidad de añadir medicamentos relacionados con ese fármaco.



Figura B.38: Interfaz de datos de fármacos

Búsqueda de proveedores

1. En la pestaña de proveedores, la funcionalidad es similar a la de materiales: puedes buscar proveedores registrados utilizando un filtro de búsqueda.
2. Los resultados mostrarán una lista de proveedores registrados.



Figura B.39: Interfaz de datos de proveedor

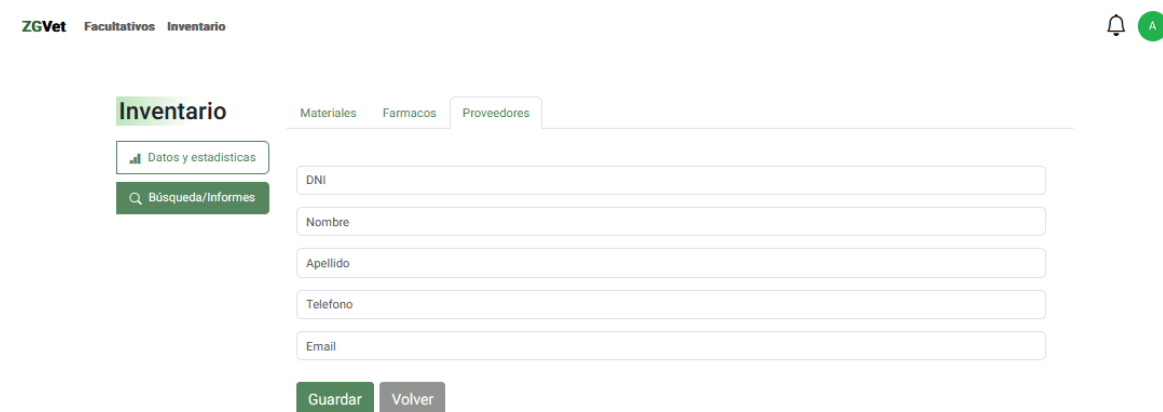


Figura B.40: Interfaz de datos de proveedor

3. Puedes ver más detalles sobre un proveedor específico haciendo clic en el icono de lupa junto a él.

The screenshot shows the 'Inventario' (Inventory) section of the ZGVet application. The breadcrumb trail is 'ZGVet > Facultativos > Inventario'. The main navigation tabs are 'Materiales', 'Farmacos', and 'Proveedores', with 'Proveedores' selected. On the left sidebar, there are buttons for 'Datos y estadísticas' and 'Búsqueda/Informes'. The main content area displays the details for a provider named 'Antonio De Lucia'. Below the name, there are three buttons: 'Material', 'Medicamento', and 'Dar de Baja'. The provider's contact information is listed: 'DNI: 12345678A', 'Email: unemail@ejemplo.com', and 'Telefono: 89 616 4355'. A section titled 'Materiales' contains a table with the following data:

Nombre	Lote	Fecha Adquisición	Fecha Caducidad	Cantidad	Precio
Ibuprofeno	1234	2024-04-19	2024-04-27	5	10

At the bottom right of the table area, there is a 'Cancelar' button.

Figura B.41: Interfaz de datos de proveedor

Además en esta sección también tienes la opción de añadir directamente a un proveedor un material o un medicamento, y la opción de dar de baja al proveedor.

Este manual de usuario está creado para ayudarte a familiarizarte con las distintas secciones de la aplicación y cómo utilizarlas.

Manual del Desarrollador

Antes de realizar modificaciones en la aplicación, es esencial poseer una base sólida de conocimientos sobre ella. Por lo tanto, se recomienda leer detenidamente los capítulos 5 y 6 antes de proceder con cualquier cambio. Esto asegurará que cuentes con la información necesaria para entender la estructura, las funcionalidades y los posibles riesgos de la aplicación, permitiéndote realizar cambios de manera segura y efectiva.

C.1 Estructura de carpetas back-end

El back-end de la aplicación está organizado en una serie de carpetas, cada una de ellas con un propósito específico. A continuación se describe cada una de ellas:

- **Carpeta controller** Esta carpeta contiene los controladores, que son responsables de manejar las solicitudes HTTP y coordinar la comunicación entre los modelos y las vistas.
- **Carpeta model** En esta carpeta se encuentran los modelos, que representan la lógica de negocio y los datos de la aplicación
 - **Carpeta dto** Los Data Transfer Objects (DTO) son objetos que se utilizan para transferir datos entre diferentes partes de la aplicación, como entre los controladores y los servicios, o entre los servicios y las capas de datos. Los DTO ayudan a desacoplar las diferentes capas de la aplicación y a simplificar la transmisión de datos. En esta carpeta se encuentran todos los DTO de todas las clases del proyecto.
 - **Carpeta commons** En esta carpeta se encuentran elementos comunes y reutilizables en toda la aplicación, como enumeraciones (enums) que definen conjuntos de valores constantes, y otras utilidades compartidas.
 - **Carpeta filters** Los filtros de búsqueda de las entidades nos permiten aplicar condiciones o criterios específicos al buscar datos en la base de datos. Los filtros pueden incluir parámetros de búsqueda, rangos de fechas, estados, etc., y se utilizan para refinar los resultados obtenidos.
 - **Carpeta entity** Aquí se encuentran entidades que representan las tablas de la base de datos y sus relaciones. Cada entidad es una clase que define los atributos de una tabla y cómo se mapean a la base de datos mediante JPA (Java Persistence API).
 - **Carpeta repository** Aquí se encuentran los repositorios que son interfaces que definen operaciones de acceso a datos para las entidades. Utilizando JPA, los repositorios permiten realizar operaciones de CRUD (crear, leer, actualizar, eliminar) en la base de datos de manera eficiente y segura.

- **Carpeta request** Los objetos de solicitud (request) se utilizan para recibir datos de los clientes en las peticiones HTTP. Estos objetos permiten mapear los datos recibidos en una estructura comprensible por la aplicación, facilitando la validación y el manejo de los datos.

Estos subdirectorios dentro de model ayudan a organizar los elementos relacionados con los datos y la lógica de negocio de la aplicación, permitiendo una separación clara de responsabilidades y facilitando el desarrollo y mantenimiento del código.

- **Carpeta services** Esta carpeta contiene los servicios de la aplicación, que actúan como puente entre el modelo y el controlador. Los servicios son responsables de gestionar las operaciones entre ambos, proporcionando una capa intermedia para la lógica de negocio y la manipulación de datos.
- **Carpeta config** La carpeta config contiene la configuración de la aplicación y archivos relacionados con la seguridad. Estos archivos son fundamentales para definir el comportamiento de la aplicación y gestionar aspectos clave de su seguridad.

Los elementos clave de esta carpeta incluyen:

Configuración de la aplicación: Aquí se expone la configuración que define los parámetros y opciones de la aplicación. Esto puede incluir configuraciones de autenticación, encriptación de contraseñas, etc.

Archivo de seguridad: El archivo de seguridad es un componente crítico que define las políticas y reglas de seguridad de la aplicación. Este archivo puede contener configuraciones para controlar el acceso a los recursos, gestionar la autenticación y autorización de los usuarios, y configurar otras medidas de seguridad, como control de acceso.

Estos elementos en la carpeta config son esenciales para personalizar y gestionar la aplicación de acuerdo con los requisitos específicos del proyecto y las mejores prácticas de seguridad. Mantener estos archivos organizados y actualizados es importante para garantizar el correcto funcionamiento y la seguridad de la aplicación.

- **Carpeta jwt** Dentro de la carpeta jwt se encuentran los archivos relacionados con la autenticación utilizando JSON Web Tokens (JWT). Estos archivos son fundamentales para la autenticación y autorización de los usuarios en la aplicación, permitiendo verificar y validar los tokens JWT en las solicitudes HTTP.

Los componentes clave de esta carpeta son:

JwtAuthenticationFilter: Esta clase es un filtro de autenticación que extiende OncePerRequestFilter y se encarga de verificar los tokens JWT en las solicitudes HTTP entrantes. Si se encuentra un token válido, el filtro autentica al usuario correspondiente estableciendo un objeto UsernamePasswordAuthenticationToken en el contexto de seguridad de Spring.

JwtService: Esta clase es un servicio que gestiona la creación y verificación de tokens JWT. Proporciona métodos para generar tokens a partir de un usuario, así como para verificar la validez de los tokens.

Estos componentes dentro de la carpeta jwt permiten gestionar y asegurar la autenticación de los usuarios mediante JSON Web Tokens, lo que es esencial para garantizar el acceso seguro y autorizado a los recursos de la aplicación. Mantener estos archivos actualizados y seguros es fundamental para la integridad y seguridad de la aplicación.

- **Carpeta exceptions** Esta carpeta alberga las clases de excepción personalizadas utilizadas en la aplicación para manejar errores de manera controlada y proporcionar mensajes de error claros y específicos.
- **Paquete de resources** En esta carpeta encontrarás el archivo `application.properties`, que contiene la configuración de la aplicación. Este archivo es esencial para definir las propiedades que afectan el comportamiento y las características de la aplicación. Algunas de las configuraciones que puedes encontrar en este archivo incluyen:

- **Configuración de base de datos:** Los parámetros para establecer la conexión con la base de datos, como la URL, el nombre de usuario, la contraseña y el dialecto SQL.

C.2 Estructura de carpetas front-end

- **Carpeta component** Contiene los componentes de la aplicación, que son las unidades fundamentales de la interfaz de usuario. Los componentes definen la lógica y la presentación de secciones específicas de la aplicación y se componen de archivos TypeScript, HTML y CSS.
- **Carpeta core** Esta carpeta contiene los elementos esenciales de la aplicación, como los modelos, las solicitudes y las interfaces de usuario. Es una sección clave para mantener el núcleo del proyecto bien organizado.
 - **Carpeta model** Incluye los modelos de datos que se utilizan para representar las estructuras y objetos de datos en la aplicación. Estos modelos se definen mediante interfaces o clases y son esenciales para manejar datos en el front-end.
 - **Carpeta request** Contiene los objetos de solicitud que se utilizan para enviar datos desde el front-end a los servicios del back-end.
 - **Carpeta ui** Almacena los componentes y estilos de la interfaz de usuario que son reutilizables en toda la aplicación. Incluye elementos visuales como las gráficas.
- **Carpeta services** Alberga los servicios que se encargan de la lógica de negocio en el front-end y de la comunicación con el back-end. Estos servicios realizan tareas como solicitudes HTTP, procesamiento de datos y manejo de estados compartidos.

Además para desarrollar el front-end de la aplicación, el desarrollador debe instalar Node.js y Angular CLI. A continuación, se describe cómo hacerlo, junto con los comandos relevantes:

C.2.1 Instalación de Node.js

Para instalar Node.js [10], sigue estos pasos:

1. Ve a la página oficial de [Node.js](<https://nodejs.org/>) y descarga la versión 20.8.0.
2. Una vez descargado el instalador, ejecútalo y sigue las instrucciones para completar la instalación.
3. Para verificar que Node.js se ha instalado correctamente, ejecuta el siguiente comando en el terminal, esto mostrará la versión de Node.js instalada.:

```
node --version
```

C.2.2 Instalación de Angular CLI

Para instalar Angular CLI [9], sigue estos pasos:

1. Abre una línea de comandos o terminal, con dirección en la carpeta del front-end.
2. Ejecuta el siguiente comando para instalar Angular CLI globalmente,este descargará e instalará Angular CLI a nivel global:

```
npm install -g @angular/cli@16.2.12
```

3. Para verificar que Angular CLI se ha instalado correctamente, ejecuta el siguiente comando, este mostrará la versión de Angular CLI instalad:

```
ng version
```

Apéndice D

Enlaces del proyecto

A continuación dispondremos de los enlaces del proyecto:

- **Enlace al repositorio GitHub del proyecto:**
<https://github.com/PaulaHC/ZGVet>
- **Enlace a la página web donde está desplegado el proyecto:**
<http://virtual.lab.inf.uva.es:20062/inicio>

Bibliografía

- [1] Angular. *Angular*. 29-03-2024. URL: <https://angular.io/doc>.
- [2] Spring Boot. *Spring Security*. 23-02-2024. URL: <https://spring.io/projects/spring-security>.
- [3] Tom Preston-Werner y Scott Chacon Chris Wanstrath P. J. Hyett. *Github*. 16-11-2023. URL: <https://github.com/>.
- [4] *Cirugía Veterinaria*. 04-04-2024. URL: <https://clinicaveterinariamadrid.es/cirugia/>.
- [5] Oracle Corporation. *Mysql*. 16-11-2023. URL: <https://www.mysql.com/>.
- [6] Covetrus. *Covetrus*. 15-11-2023. URL: <https://software.covetrus.com/emea/>.
- [7] docker. *Docker Compose*. 16-11-2023. URL: <https://docs.docker.com/compose/>.
- [8] Inc. Docker. *Docker*. 16-11-2023. URL: <https://www.docker.com/>.
- [9] *Download Angular*. 10-10-2023. URL: <https://docs.angular.lat/cli>.
- [10] *Download Node*. 10-10-2023. URL: <https://nodejs.org/en/download>.
- [11] Red Hat. *¿Qué es la metodología ágil?* 15-01-2024. URL: <https://www.redhat.com/es/topics/devops/what-is-agile-methodology>.
- [12] Red Hat. *Hibernate*. 16-11-2023. URL: <https://hibernate.org/>.
- [13] jwt. *JSON Web Tokens*. 23-02-2024. URL: <https://jwt.io/>.
- [14] *La desparasitación interna y externa en perros y gatos*. 04-04-2024. URL: <https://www.esveterinaria.com/la-desparasitacion-en-perros-y-gatos/>.
- [15] Roberto López. *Desarrollo de software: metodologías en cascada (Waterfall) y ágil (Agile)*. 15-01-2024. URL: <https://www.yunbitsoftware.com/blog/2016/05/20/desarrollo-de-software-metodologias-waterfall-agile/>.
- [16] R. D. Schultz M. J. Day M. C. Horzinek y R. A. Squires. *DIRECTRICES PARA LA VACUNACIÓN DE PERROS Y GATOS*. 04-04-2024. URL: <https://wsava.org/wp-content/uploads/2020/01/WSAVA-vaccination-guidelines-2015-Spanish.pdf>.
- [17] Juan Pavón Mestras. *El patrón Modelo-Vista-Controlador (MVC)*. 20-02-2024. URL: <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.mvc.pdf>.
- [18] *Metodología evaluación del riesgo de incendio Meseri*. 23-03-2024. URL: https://www.urbicad.com/mico/metodos_riesgos.htm.
- [19] npm. *Fullcalendar*. 16-11-2023. URL: <https://www.npmjs.com/package/@fullcalendar/angular>.
- [20] *Salarios medios EU*. 15-05-2024. URL: <https://es.indeed.com/career/desarrollador-junior/salaries>.
- [21] Spring. *Spring*. 4-12-2023. URL: <https://spring.io/projects>.
- [22] TypeScript. *TypeScript*. 16-11-2023. URL: <https://www.typescriptlang.org/>.
- [23] Vetit. *Vetit*. 15-11-2023. URL: <https://www.vetit.co.uk/>.
- [24] Vetport. *Vetport*. 15-11-2023. URL: <https://www.vetport.com/>.