

Cuidado de Mascotas  
Fin de Grado

Grado en Ingeniería Informática



Trabajo Fin de Grado

Autor:

Wenceslao Diego Pacheco Guevara

Tutor/es:

Estela Saquete

Junio 2025



# **Justificación y Objetivos**

Se pretende diseñar una aplicación web que ofrezca una serie de funciones para optimizar y facilitar el cuidado de la mascota. Uno de los principales motivos que me impulso a desarrollar esta aplicación web fue la de dar conectar dueños con otros dueños y la de alimentar el conocimiento a las personas que no cuentan con experiencia para cuidar una mascota.

Objetivos:

* Estudiar el sistema de información
* Realizar un estudio de la cuestión hasta el momento presente
* Especificación de los requerimientos del sistema
* Diseño de la base de datos
* Implementación de la aplicación
* Evaluación final

# **Agradecimientos**

Me gustaría agradecer a mi tutora del TFG, Estela Saquete, que me ha guiado durante estos meses al desarrollo de mi trabajo final de grado y su esfuerzo por las reuniones mensuales que conllevó.

Por otro lado, me gustaría agradecer a la empresa en la que me encuentro ahora mismo, Idasa Sistemas, ya que durante estos años que he estado trabajando con ellos he adquirido muchos conocimientos que he aplicado a mi trabajo final.

# **Dedicatoria**

Me gustaría dedicar este TFG a todos mis amigos que me han acompañado en la carrera y a todos los profesores con los que he tenido la grata experiencia de aprender un conocimiento con ellos.

# **Citas**

"El software es una gran combinación entre arte e ingeniería."  
— *Bill Gates*

En esta cita Bill Gates está describiendo que desarrollar software no es solo un proceso técnico aburrido. Porque aunque escribir código requiere conocimientos de lógica, algoritmos, estructuras de datos, etc, también hay una parte muy importante de creatividad, diseño y estilo, que se puede considerar como arte y pienso que cada persona tiene su propio estilo de arte.

**Índice de Contenidos**

[**Justificación y Objetivos** 2](#_Toc201491539)

[**Agradecimientos** 3](#_Toc201491540)

[**Dedicatoria** 4](#_Toc201491541)

[**Citas** 5](#_Toc201491542)

[**Índice de Figuras** 8](#_Toc201491543)

[**Introducción** 9](#_Toc201491544)

[Resumen 9](#_Toc201491545)

[Estudio de Mercado 10](#_Toc201491546)

[**Herramientas, tecnologías y lenguajes de programación empleados** 12](#_Toc201491547)

[Herramientas 12](#_Toc201491548)

[Otros recursos utilizados 13](#_Toc201491549)

[Tecnologías y lenguajes de programación 13](#_Toc201491550)

[Planificación del proyecto 13](#_Toc201491551)

[Estimación temporal 13](#_Toc201491552)

[**Requisitos FUNCIONALES Y NO FUNCIONALES** 14](#_Toc201491553)

[**Diseño** 16](#_Toc201491554)

[Bocetos 16](#_Toc201491555)

[Diseño final 30](#_Toc201491556)

[**Arquitectura** 34](#_Toc201491557)

[Modelo Arquitectura en Capas (N-Tier Architecture) 34](#_Toc201491558)

[**Implementación** 36](#_Toc201491559)

[Front End 36](#_Toc201491560)

[CSS 36](#_Toc201491561)

[Bootstrap 36](#_Toc201491562)

[Javascript y Jquery 36](#_Toc201491563)

[Peticiones HTTP 36](#_Toc201491564)

[Back End 36](#_Toc201491565)

[IIS (Internet Information Services) 36](#_Toc201491566)

[Base de Datos 36](#_Toc201491567)

[Esquema Entidad Relación 36](#_Toc201491568)

[Acceso a Datos 36](#_Toc201491569)

[**Seguridad** 37](#_Toc201491570)

[Cifrado de las contraseñas 37](#_Toc201491571)

[**APIS Utilizadas** 37](#_Toc201491572)

[**Pruebas** 37](#_Toc201491573)

[**Trabajos futuros y Posibles mejoras** 37](#_Toc201491574)

[**Conclusiones** 38](#_Toc201491575)

[**Bibliografía y referencias** 39](#_Toc201491576)

# **Índice de Figuras**

[Figura 1 Estudio de mercado - Test Interactivo 10](#_Toc201478726)

[Figura 2 Estudio de mercado - Test interactivo 2 10](#_Toc201478727)

[Figura 3 Mockup Navegador Usuario Normal (Sin hacer Login) 16](#_Toc201478728)

[Figura 4 Mockup Navegador Dueño 16](#_Toc201478729)

[Figura 5 Mockup Navegador Veterinario 16](#_Toc201478730)

[Figura 6 Mockup Navegador Administrador 16](#_Toc201478731)

[Figura 7 Mockup Lista Servicios: Vista Usuario Normal/Dueño 16](#_Toc201478732)

[Figura 8 Mockup Detalle Reserva Cita: Vista Usuario Normal/Dueño 17](#_Toc201478733)

[Figura 9 Mockup Lista Servicios: Vista Veterinario/Administrador 17](#_Toc201478734)

[Figura 10 Mockup Servicios Detalle: Vista Veterinario/Administrador 18](#_Toc201478735)

[Figura 11 Mockup Servicios Eliminar: Vista Veterinario/Administrador 18](#_Toc201478736)

[Figura 12 Mockup Comunidad Lista Publicaciones: Vista Usuario Normal 19](#_Toc201478737)

[Figura 13 Mockup Comunidad Detalle Publicación: Vista Usuario Normal 19](#_Toc201478738)

[Figura 14 Mockup Comunidad Lista Publicaciones: Vista Dueño/Veterinario/Administrador 20](#_Toc201478739)

[Figura 15 Mockup Comunidad Eliminar: Vista Dueño/Veterinario/Administrador 20](#_Toc201478740)

[Figura 16 Mockup Comunidad Editar: Vista Dueño/Veterinario/Administrador 21](#_Toc201478741)

[Figura 17Mockup Comunidad Detalle Publicación: Vista Dueño/Veterinario/Administrador 21](#_Toc201478742)

[Figura 18 Mockup Noticias Lista: Vista Usuario Normal/Dueño 22](#_Toc201478743)

[Figura 19 Mockup Noticias Detalle: Vista Usuario Normal/Dueño 22](#_Toc201478744)

[Figura 20 Mockup Noticias Test: Vista Usuario Normal/Dueño 23](#_Toc201478745)

[Figura 21 Mockup Noticias Lista: Vista Veterinario/Administrador 23](#_Toc201478746)

[Figura 22 Mockup Noticias Eliminar:Vista Veterinario/Administrador 24](#_Toc201478747)

[Figura 23 Mockup Noticias Editar/Añadir Lección: Vista Veterinario/Administrador 24](#_Toc201478748)

[Figura 24 Mockup Noticias Añadir Preguntas: Vista Veterinario/Administrador 25](#_Toc201478749)

[Figura 25 Mockup Mascotas Lista: Vista Dueño 25](#_Toc201478750)

[Figura 26 Mockup Mascotas Lista: Vista Veterinario 26](#_Toc201478751)

[Figura 27 Mockup Mascotas Eliminar: Vista Veterinario/Dueño 26](#_Toc201478752)

[Figura 28 Mockup Mascotas Detalle: Vista Dueño/Veterinario 27](#_Toc201478753)

[Figura 29 Mockup Chat: Vista Dueño/Veterinario/Administrador 27](#_Toc201478754)

[Figura 30 Mockup Nuevo Chat: Vista Dueño/Veterinario/Administrador 28](#_Toc201478755)

[Figura 31 Mockup Usuarios Lista: Vista Administrador 28](#_Toc201478756)

[Figura 32 Mockup Citas: Vista Veterinario 29](#_Toc201478757)

[Figura 33 Servicios: Vista Usuario Normal 30](#_Toc201478758)

[Figura 34 Servicios Solicitar Cita: Vista Usuario Normal 30](#_Toc201478759)

[Figura 35 Servicios Lista: Vista Administrador/Veterinario 31](#_Toc201478760)

[Figura 36 Servicios Editar: Vista Veterinario/Administrador 31](#_Toc201478761)

[Figura 37 Noticias Lista: Vista Veterinario/Administrador 32](#_Toc201478762)

[Figura 38 Noticias Detalle: Vista Administrador/Veterinario 32](#_Toc201478763)

[Figura 39 Noticias Encuesta: Vista Usuario Normal 33](#_Toc201478764)

[Figura 40 Usuarios Lista: Vista Administrador 33](#_Toc201478765)

[Figura 41 Logs Lista: Vista Administrador 33](#_Toc201478766)

# **Introducción**

## Resumen

Se pretende crear un sitio web dirigido tanto a veterinarios como a personas que tengan a su cuidado a alguna mascota. Esta web almacenará la información de la mascota y estará compuesta esencialmente de datos sanitarios relacionados con la mascota

Existirán 4 perfiles:

El usuario normal (el que no está registrado en la web), podrá acceder a información muy interesante de las mascotas (blogs, lecciones interactivas, noticias). Además, podrá concertar nuevas citas sin tener que estar registrado y acceder a los servicios que ofrecen los veterinarios (peluquería, guardería, corte de uñas u otros).

El dueño de la mascota tendrá la capacidad de hacer todo lo que puede hacer un usuario normal y además podrá registrar a sus mascotas en la web. El podrá proveer información diaria y detallada de sus mascotas y mantener contacto con los veterinarios. Además, será capaz de publicar blogs para que otros usuarios puedan responder y mantener una comunidad en relación a la salud de las mascotas.

El veterinario (Este perfil podrá publicar los servicios, noticias, blogs, nuevas lecciones, confirmar las citas de los usuarios) será el encargado de proveer información más técnica de las mascotas para cumplimentar las fichas de las mascotas de los dueños, este perfil está pensado para ser como un “administrador” y puede darse de alta como “clínica” o como “veterinario particular”.

El administrador tiene un rol más de “moderador”, este perfil está pensado para restringir aquel contenido de la web que no es fiable (fraudes) o contienen contenido inapropiado. Además, tendrá la capacidad de bloquear otros perfiles y de visualizar los errores de la web.

Por último, existirá un chat entre el dueño y el veterinario en caso de que el dueño decida hablar directamente con el veterinario.

Como complemento el veterinario podrá crear cuestionarios y enviarlos a los dueños, estos cuestionarios tienen como propósito obtener información más detallada de la mascota en caso de que el dueño no haya cumplimentado la información en la web.

## Estudio de Mercado

URL: https://www.anicura.es/clinicas/san-vicente-hospital-veterinario/

Descripción: Aunque la web tenga el aspecto de un blog, contiene contenido muy interesante para tener una noción de la salud de las mascotas en general. Esta web está destinada principalmente a proporcionar información de la empresa “Anicura” y de sus servicios, esta empresa posee diferentes clínicas en Alicante

Aspectos importantes para tener en cuenta:

* Posibilidad de crear citas online con el veterinario
* Test variados (Existen test para diagnosticar el problema de la mascota y test para identificar que raza de perro es más adecuada para el usuario)
* Noticias
* Tips y Consejos acerca del cuidado de la mascota
* Posibilidad de encontrar otras clínicas veterinarias
* Formación interactiva para que el usuario aprenda las dificultades que tiene la mascota en determinadas ocasiones, ejemplo:



Figura 1 Estudio de mercado - Test Interactivo



Figura 2 Estudio de mercado - Test interactivo 2

Localidad: España, Alicante, San Vicente

URL: https://www.clinicaveterinariasolyluz.es/?subject=Informaci%C3%B3n%20PREVENCI%C3%93N%20LEISHMANIA

Descripción: Esta web sencilla parece estar destinada a informar de los servicios que ofrece la clínica veterinaria “Clínica Veterinaria Sol y Luz”.

Aspectos importantes para tener en cuenta:

* Posibilidad de crear citas online con el veterinario
* Noticias
* Catálogo de servicios (PLAN SALUD PREVENCIÓN LEISHMANIA, PLAN SALUD CACHORROS, PLAN SALUD DESPARASITACIÓN, PLAN SALUD .....)

Lugar: España, Alicante Centro

# **Herramientas, tecnologías y lenguajes de programación empleados**

## Herramientas

Visual Studio 2022: Es un entorno de desarrollo integrado (IDE) creado por Microsoft. Es utilizada para desarrollar aplicaciones en .NET, C#, C++, Python, JavaScript, entre otros lenguajes. Para el desarrollo de este trabajo se ha utilizado la edición Community (Gratuita para estudiantes)

Git: Es un sistema de control de versiones distribuido, fue creado por Linus Torvalds (el mismo creador de Linux) en 2005 para llevar el control de cambios del código fuente a lo largo del tiempo.

GitHub: Es una plataforma web basada en Git que permite almacenar proyectos Git en línea para colaborar con otras personas. Es como una "nube para Git" y fue creado en 2008 y hoy en día es parte de Microsoft. Además para el desarrollo de esta aplicación web, se ha utilizado el complemento que ya ofrece Visual Studio 2022 para subir el código a Github sin salir del IDE.

PgAdmin: Es una herramienta gráfica utilizada para administrar bases de datos PostgreSQL. pgAdmin para visualizar, editar y gestionar los datos sin tener que escribir todo desde la línea de comandos. Esta herramienta se ha utilizado para la creación de la base de datos y la creación de los backup de seguridad para este proyecto.

Dbeaver: Es una aplicación de escritorio que funciona como un cliente universal de bases de datos. Es decir, te permite conectarte y trabajar con muchos tipos diferentes de bases de datos desde una sola interfaz. Esta aplicación ha sido utilizada para realizar las consultas necesarias a la base de datos.

Draw.io: Es una herramienta gratuita para crear diagramas de todo tipo, directamente desde el navegador o como aplicación de escritorio. Se ha utilizado para la creación de Mockups y diseño de la web.

Loop11: Es una herramienta especializada en pruebas de usabilidad remotas. Se usa para evaluar qué tan fácil de usar y eficaz es un sitio web o una interfaz digital, observando cómo los usuarios reales interactúan con ella.

Selenium IDE: Es una herramienta de automatización de pruebas para aplicaciones web, que funciona como una extensión del navegador (disponible para Chrome y Firefox).Se usa para grabar, reproducir y automatizar interacciones con sitios web sin necesidad de escribir código.

## Otros recursos utilizados

Shadow: Es un servicio de computación en la nube principalmente orientado a videojuegos y otras tareas intensivas, que te permite usar un PC gaming de alto rendimiento alojado en la nube, desde cualquier dispositivo con conexión a internet. Actualmente tengo contratado un plan con este servicio ya que no dispongo de un ordenador lo suficientemente potente para realizar este proyecto.

Las características que me ofrece Shadow actualmente son las siguientes:

* Procesador AMD EPYC 7513 32-Core Processor 2.60 GHz
* RAM instalada 12,0 GB
* Almacenamiento 512 GB SSD QEMU QEMU HARDDISK
* Tarjeta gráfica NVIDIA Quadro RTX 4000 (8 GB)
* Tipo de sistema Sistema operativo de 64 bits, procesador basado en x64

IIS (Internet Information Services): Es un servidor web desarrollado por Microsoft. Su función principal es alojar sitios web, aplicaciones web y servicios en sistemas operativos Windows. Se podría considerar que es un servidor web como Apache lo es para Linux.

## Tecnologías y lenguajes de programación

ASP.NET: Es un framework de desarrollo web creado por Microsoft que permite construir aplicaciones web dinámicas, sitios web, y servicios web. Está basado en el lenguaje .NET y permite usar varios lenguajes de programación como C# o VB.NET.

Visual Basic: Es un lenguaje de programación desarrollado por Microsoft. Es un lenguaje basado en el lenguaje BASIC (Beginner’s All-purpose Symbolic Instruction Code), diseñado para ser accesible a principiantes. Es el lenguaje con el cual se ha programado el backend para este proyecto.

PostgreSql: Es un sistema de gestión de bases de datos relacional y objeto (RDBMS) de código abierto, muy potente y avanzado. Es una herramienta que permite almacenar, organizar y consultar datos de forma eficiente. Es el único gestor de bases de datos para este proyecto.

## Planificación del proyecto

## Estimación temporal

# **Requisitos FUNCIONALES Y NO FUNCIONALES**

|  |  |  |
| --- | --- | --- |
| REQUISITOS FUNCIONALES | | |
| REQ. | DESCRIPCIÓN | ESTADO |
| REQ. 1 | El dueño tiene que ser capaz de registrarse y darse de baja en la web. | REALIZADO |
| REQ. 2 | El veterinario tiene que ser capaz de registrarse y darse de baja en la web. | REALIZADO |
| REQ. 3 | El dueño y usuario normal tiene que ser capaz de encontrar un veterinario y poder concertar una cita con él. | REALIZADO |
| REQ. 4 | Todos los perfiles podrán cambiar la contraseña. | REALIZADO |
| REQ. 5 | El veterinario será capaz de registrar los datos de la mascota:  - Temperatura  - Frecuencia Cardiaca  - Peso  - Alergias  - Historial clínico (si ha sido operado o si ha recibido medicamentos previamente)  - Diagnóstico  - Medicamentos  - Otros | REALIZADO |
| REQ. 6 | El dueño será capaz de registrar la información diaria de la mascota:   - Hábitos alimenticios  - Nivel de actividad  - Patrones de sueño  - Estado del tratamiento o medicamento (ej: segunda pastilla del día, estiramiento de patas antes de dormir, etc)  - Malestares. | REALIZADO |
| REQ. 7 | El veterinario será capaz de establecer un tratamiento o un horario de medicación a las mascotas. | REALIZADO |
| REQ. 8 | El veterinario podrá publicar sus servicios. | REALIZADO |
| REQ. 9 | El veterinario podrá crear lecciones interactivas sobre las enfermades. | REALIZADO |
| REQ. 10 | El dueño/usuario normal/veterinario podrá crear crear/eliminar/editar sus blogs y comentar sobre los ya existentes. | REALIZADO |
| REQ. 11 | El usuario normal/dueño podrá concertar citas con los veterinarios. | REALIZADO |
| REQ. 12 | El usuario normal/dueño podrá acceder a los servicios de los veterinarios. | REALIZADO |
| REQ. 13 | El dueño/veterinario/administrador podrán mantener un chat con otros dueños, veterinarios o administradores. | REALIZADO |
| REQ. 14 | El veterinario podrá crear una pauta de medicamentos y/o tratamientos sobre cada mascota. | REALIZADO |
| REQ. 15 | El dueño recibirá alertas si la mascota está recibiendo una pauta de medicamentos y/o tratamientos. | PENDIENTE |
| REQ. 16 | El administrador podrá eliminar servicios de veterinarios . | REALIZADO |
| REQ. 17 | El administrador podrá bloquear/desbloquear tanto dueños como veterinarios . | REALIZADO |
| REQ. 18 | El administrador podrá visualizar los logs de la web en caso de que ocurran errores. | REALIZADO |
| REQ. 19 | El administrador podrá comunicarse con otros usuarios a través de un chat. | REALIZADO |

|  |  |  |
| --- | --- | --- |
| REQUISITOS NO FUNCIONALES | | |
| REQ. | DESCRIPCIÓN | ESTADO |
| REQ. 20 | La contraseña del usuario debe ser encriptados utilizando algún algoritmo de seguridad. | REALIZADO |
| REQ. 21 | La web tiene que ser responsive | REALIZADO |
| REQ. 22 | El sistema debe ser capaz de integrarse con el sistema de gestión de base de datos PostgreSQL.. | REALIZADO |
| REQ. 23 | La aplicación debe cargar en menos de 5 segundos. | REALIZADO |
| REQ. 24 | Los fallos en la web tienen que estar centralizados en un sistema de logs . | REALIZADO |
| REQ. 25 | El tiempo promedio de aprendizaje para un usuario normal debe ser inferior a 10 minutos. | PENDIENTE |
| REQ. 26 | El tiempo promedio de aprendizaje para un dueño debe ser inferior a 15 minutos. | PENDIENTE |
| REQ. 27 | El tiempo promedio de aprendizaje para un veterinario debe ser inferior a 30 minutos. | PENDIENTE |
| REQ. 28 | El sistema debe poder soportar un aumento de usuarios hasta 100 simultáneos sin perder rendimiento. | PENDIENTE |

# **Diseño**

## Bocetos

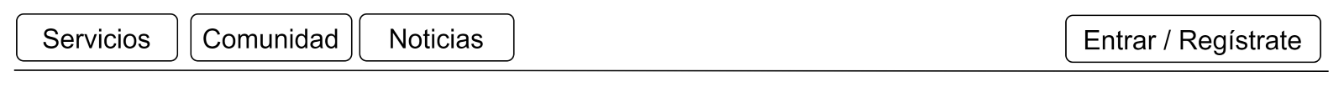


Figura 3 Mockup Navegador Usuario Normal (Sin hacer Login)

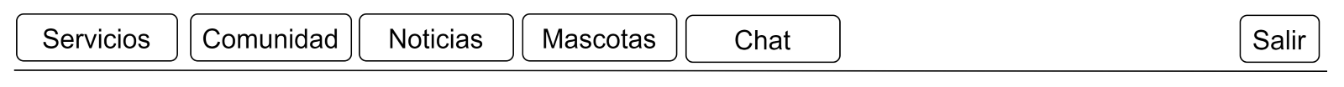


Figura 4 Mockup Navegador Dueño

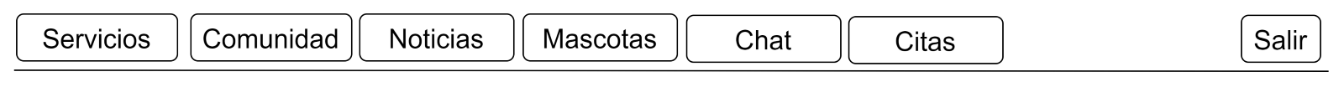


Figura 5 Mockup Navegador Veterinario



Figura 6 Mockup Navegador Administrador

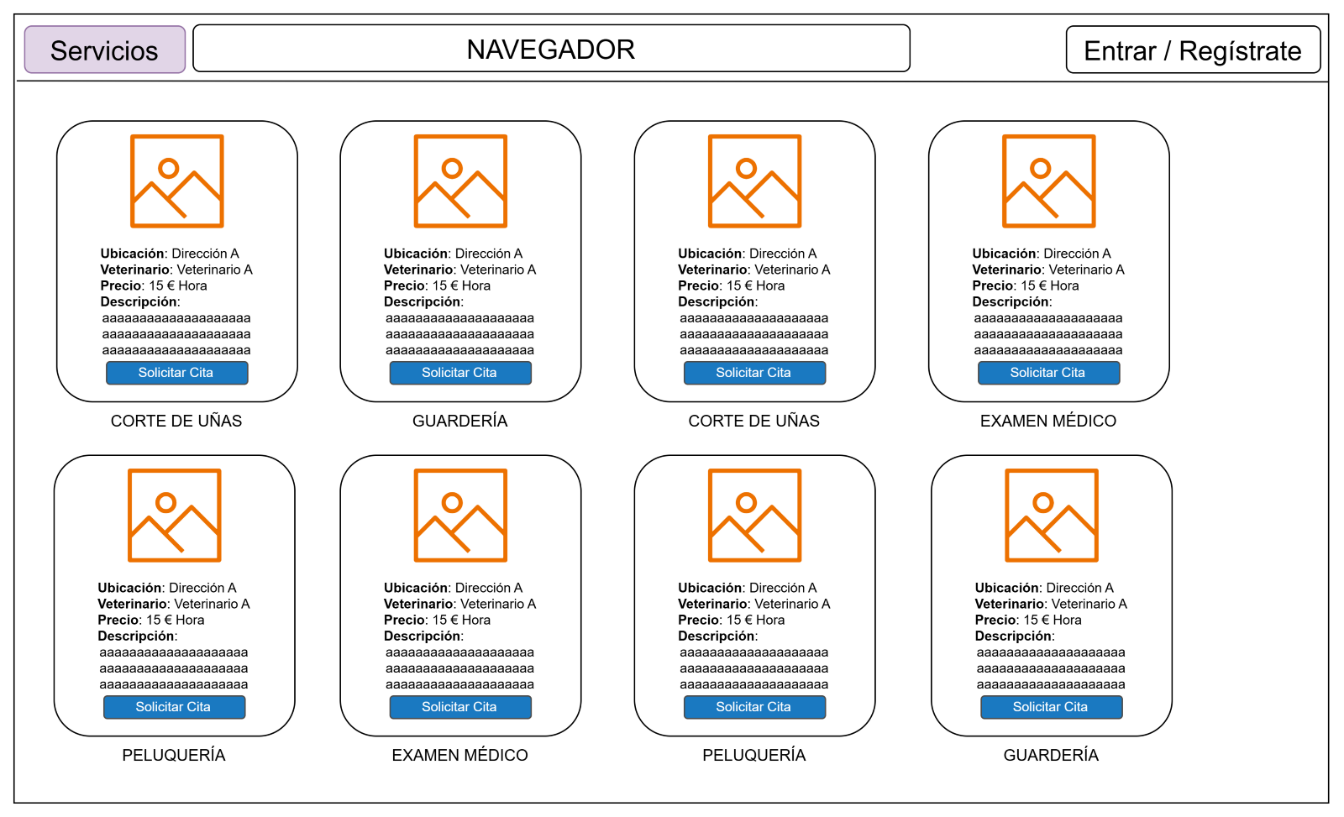


Figura 7 Mockup Lista Servicios: Vista Usuario Normal/Dueño



Figura 8 Mockup Detalle Reserva Cita: Vista Usuario Normal/Dueño

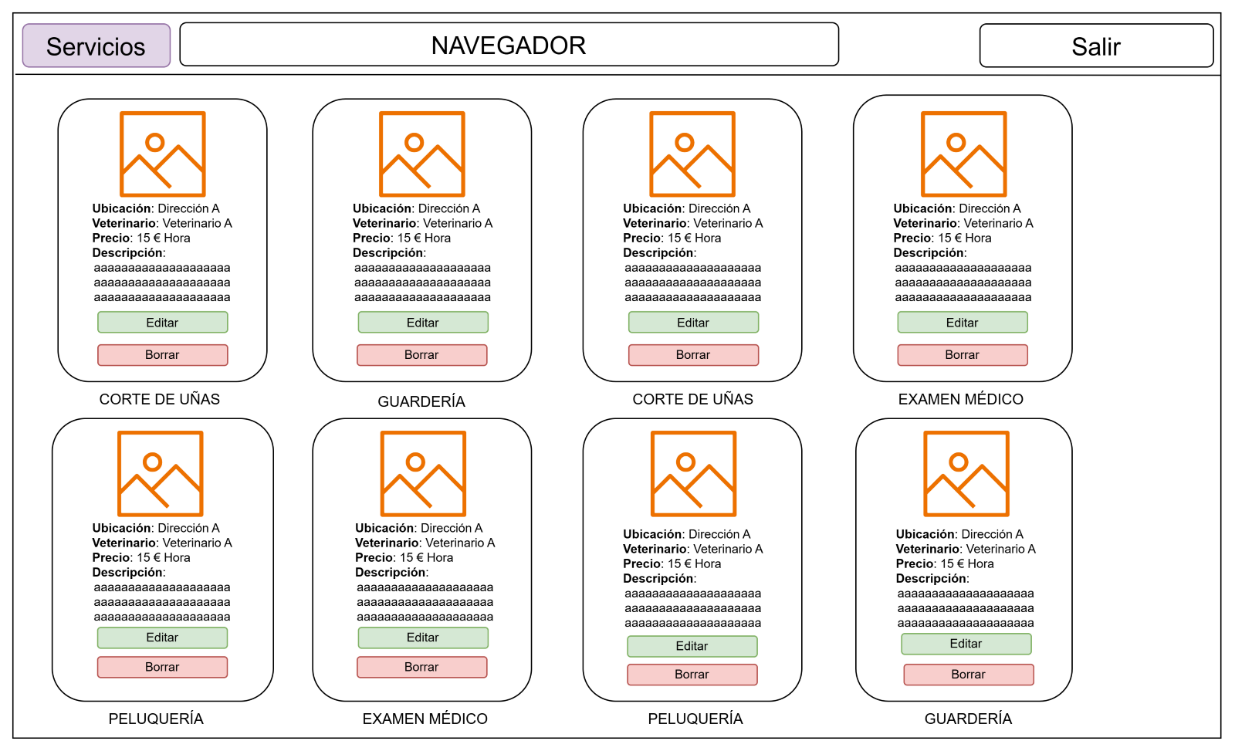


Figura 9 Mockup Lista Servicios: Vista Veterinario/Administrador

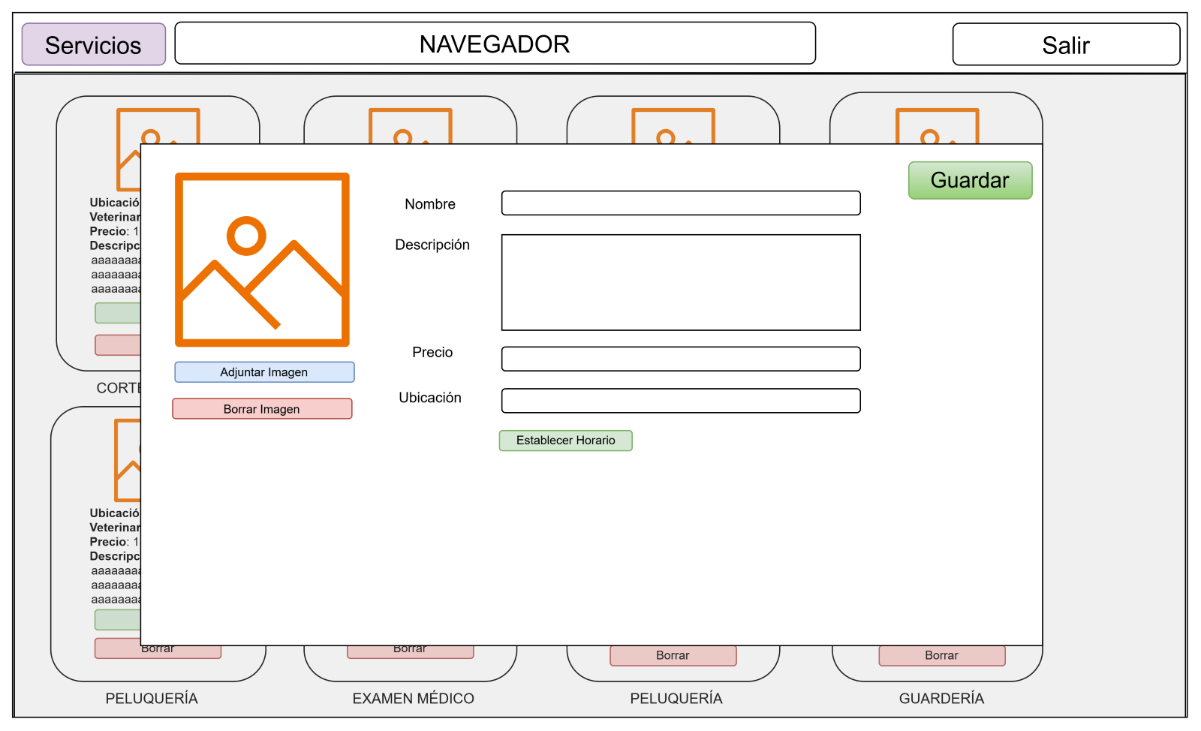


Figura 10 Mockup Servicios Detalle: Vista Veterinario/Administrador

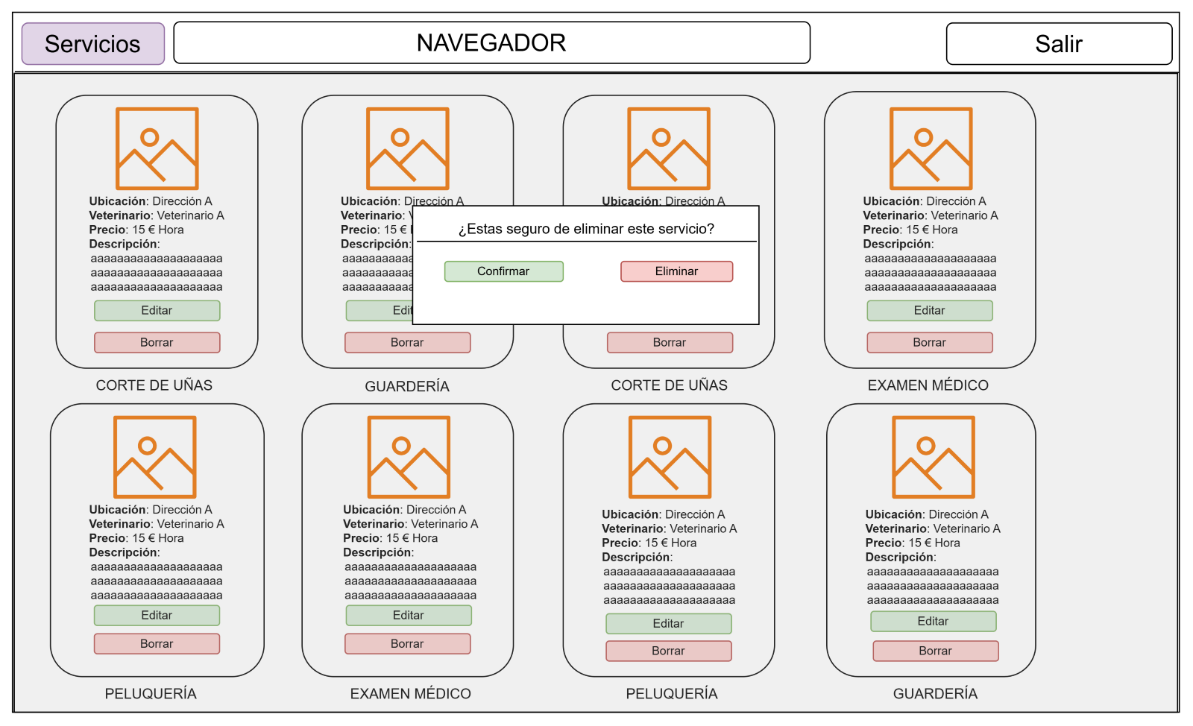


Figura 11 Mockup Servicios Eliminar: Vista Veterinario/Administrador

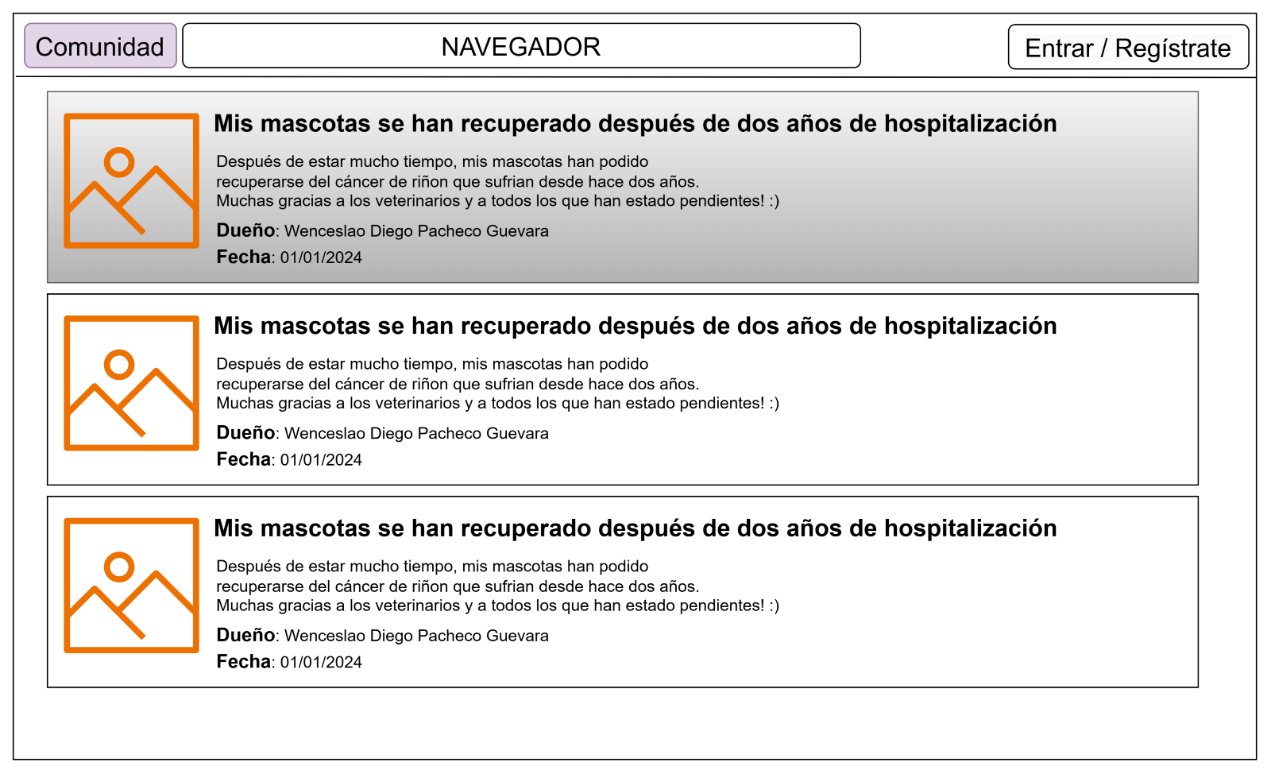


Figura 12 Mockup Comunidad Lista Publicaciones: Vista Usuario Normal

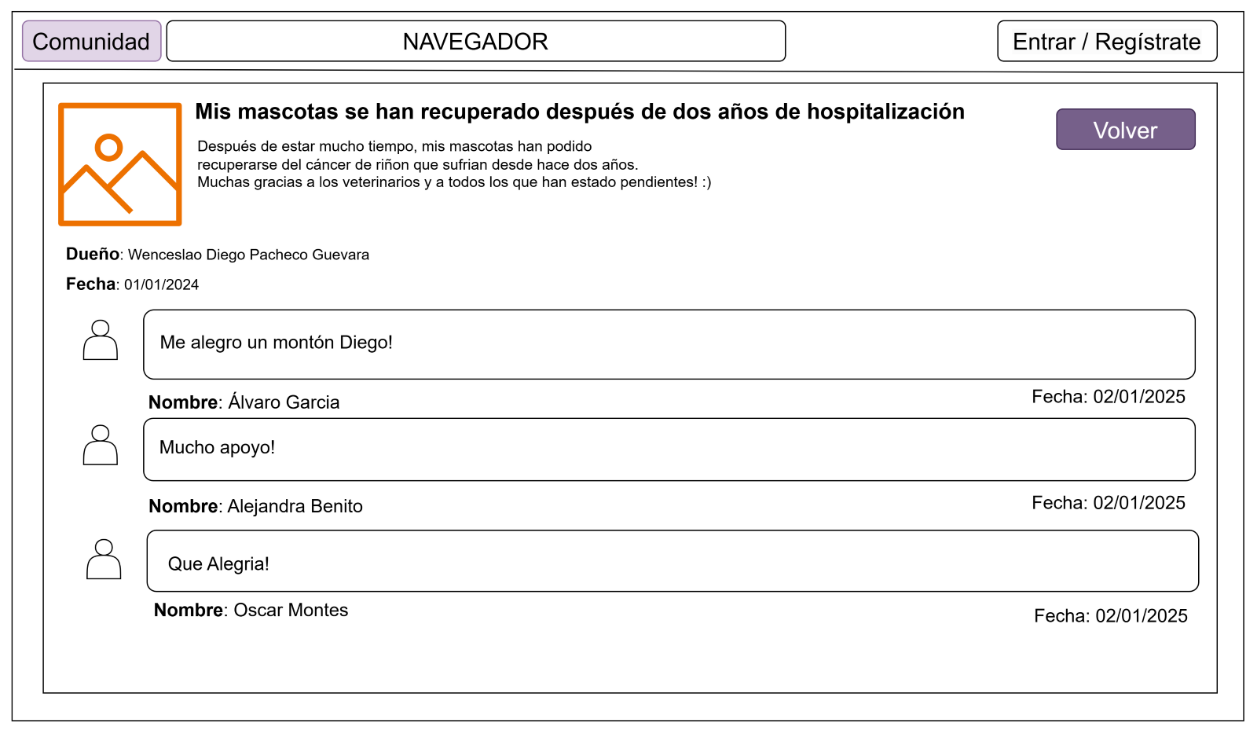


Figura 13 Mockup Comunidad Detalle Publicación: Vista Usuario Normal



Figura 14 Mockup Comunidad Lista Publicaciones: Vista Dueño/Veterinario/Administrador

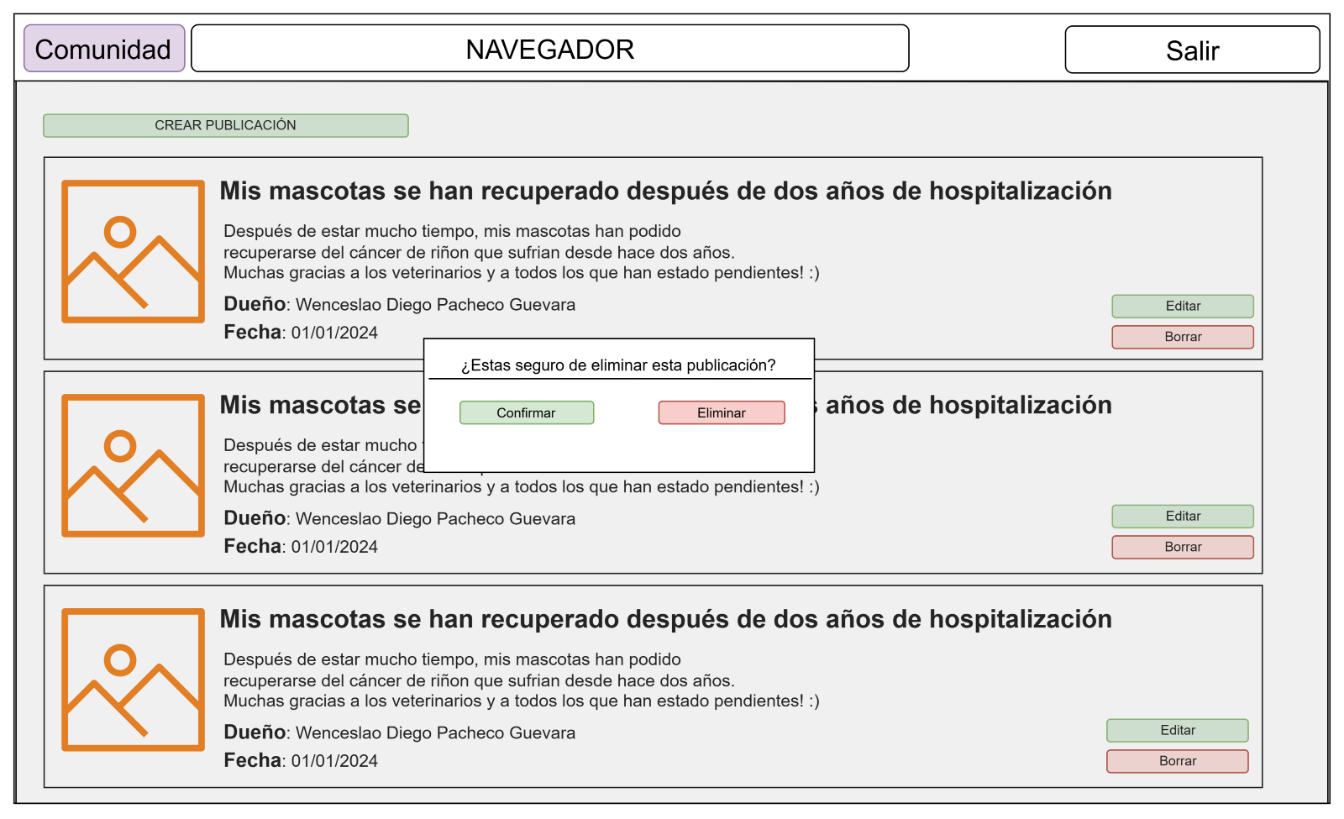


Figura 15 Mockup Comunidad Eliminar: Vista Dueño/Veterinario/Administrador

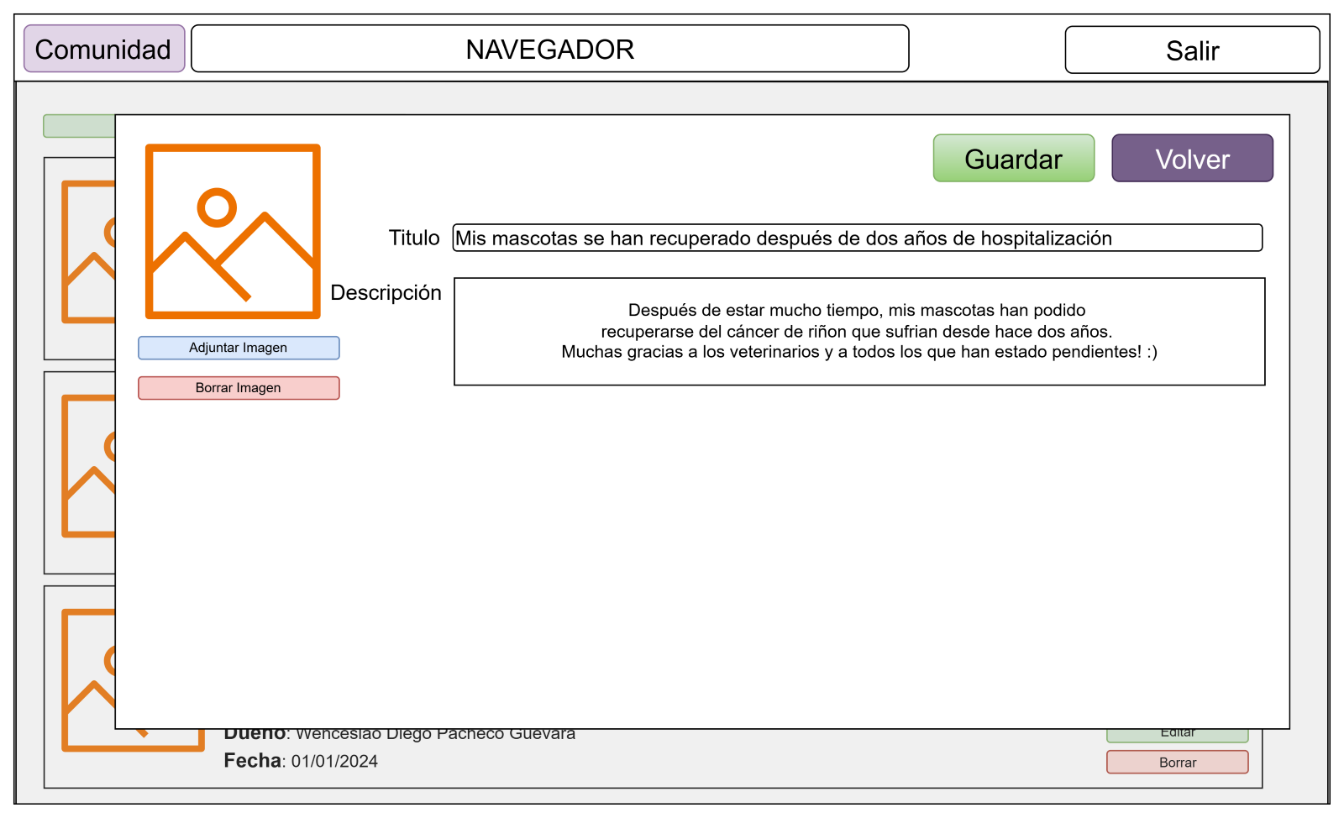


Figura 16 Mockup Comunidad Editar: Vista Dueño/Veterinario/Administrador

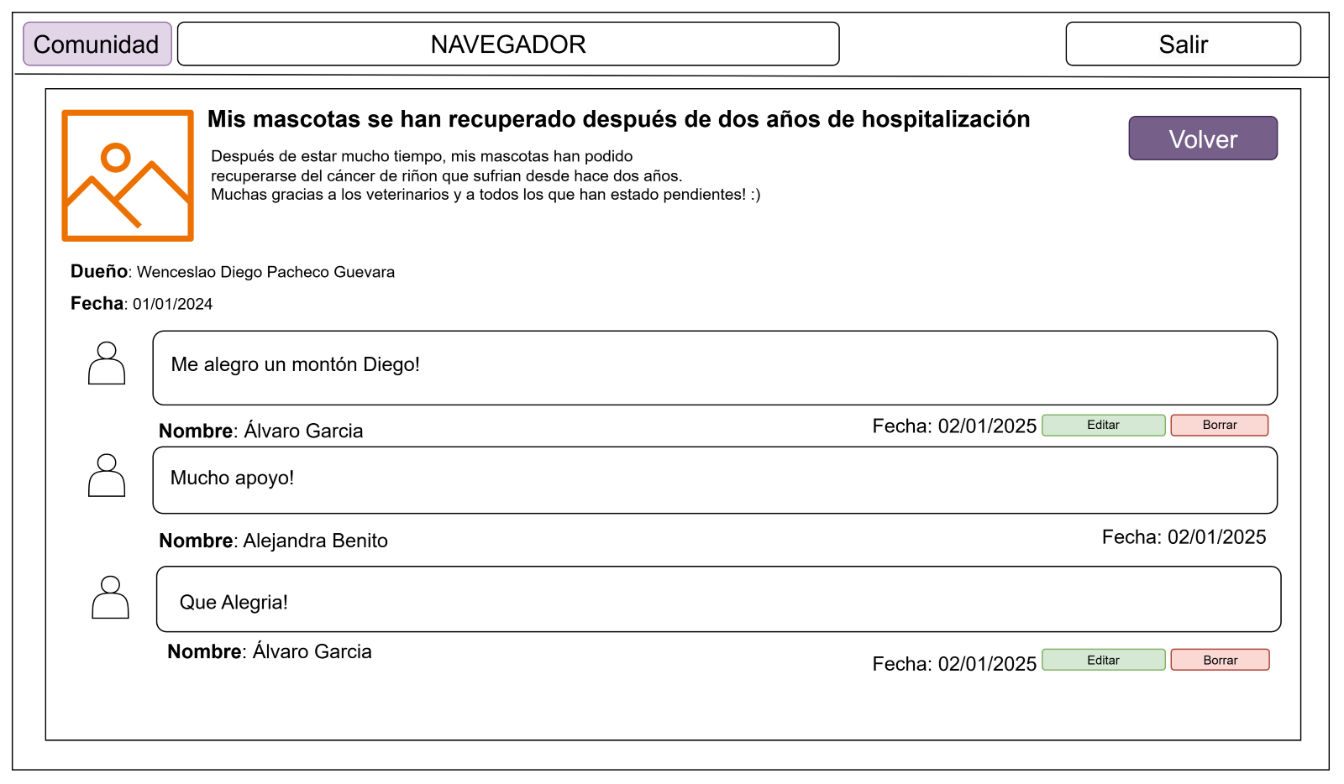


Figura 17Mockup Comunidad Detalle Publicación: Vista Dueño/Veterinario/Administrador

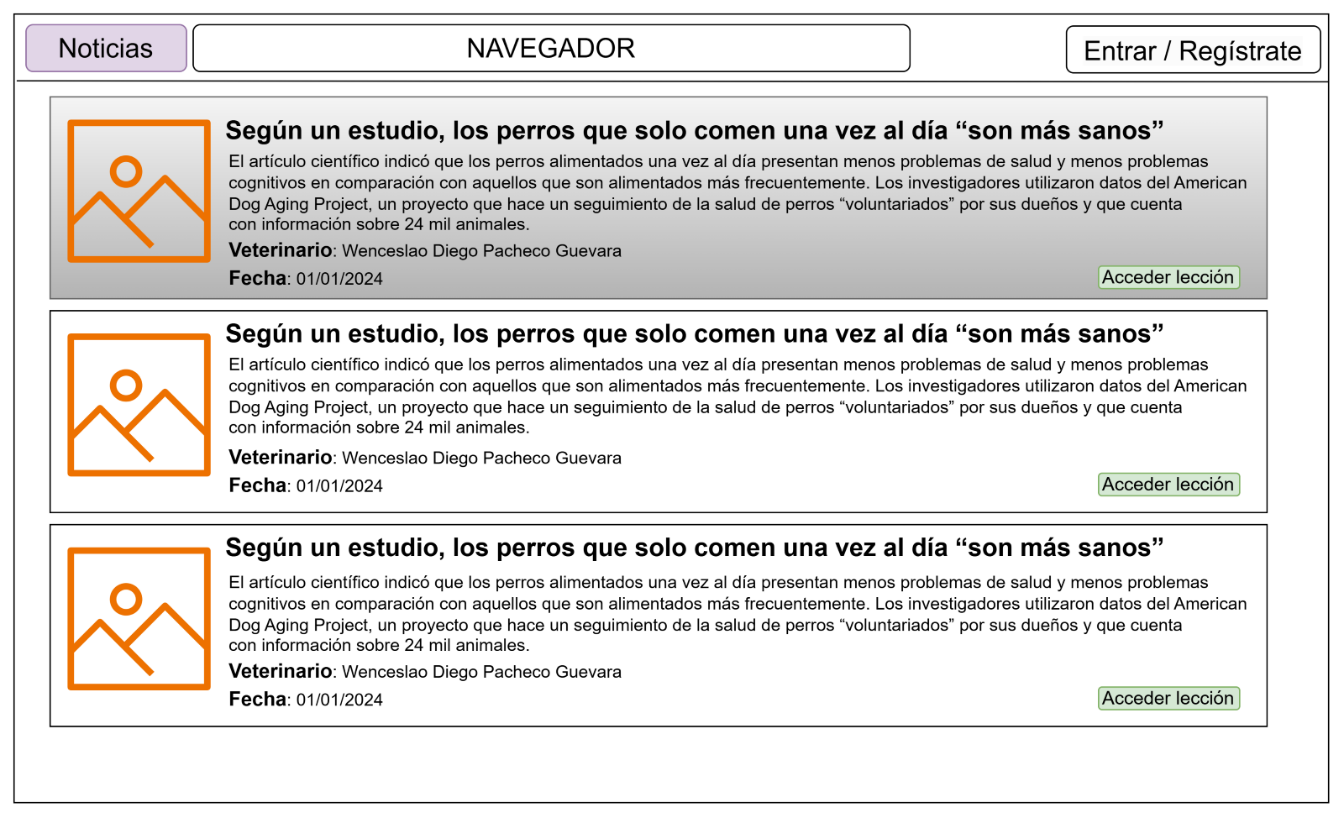


Figura 18 Mockup Noticias Lista: Vista Usuario Normal/Dueño

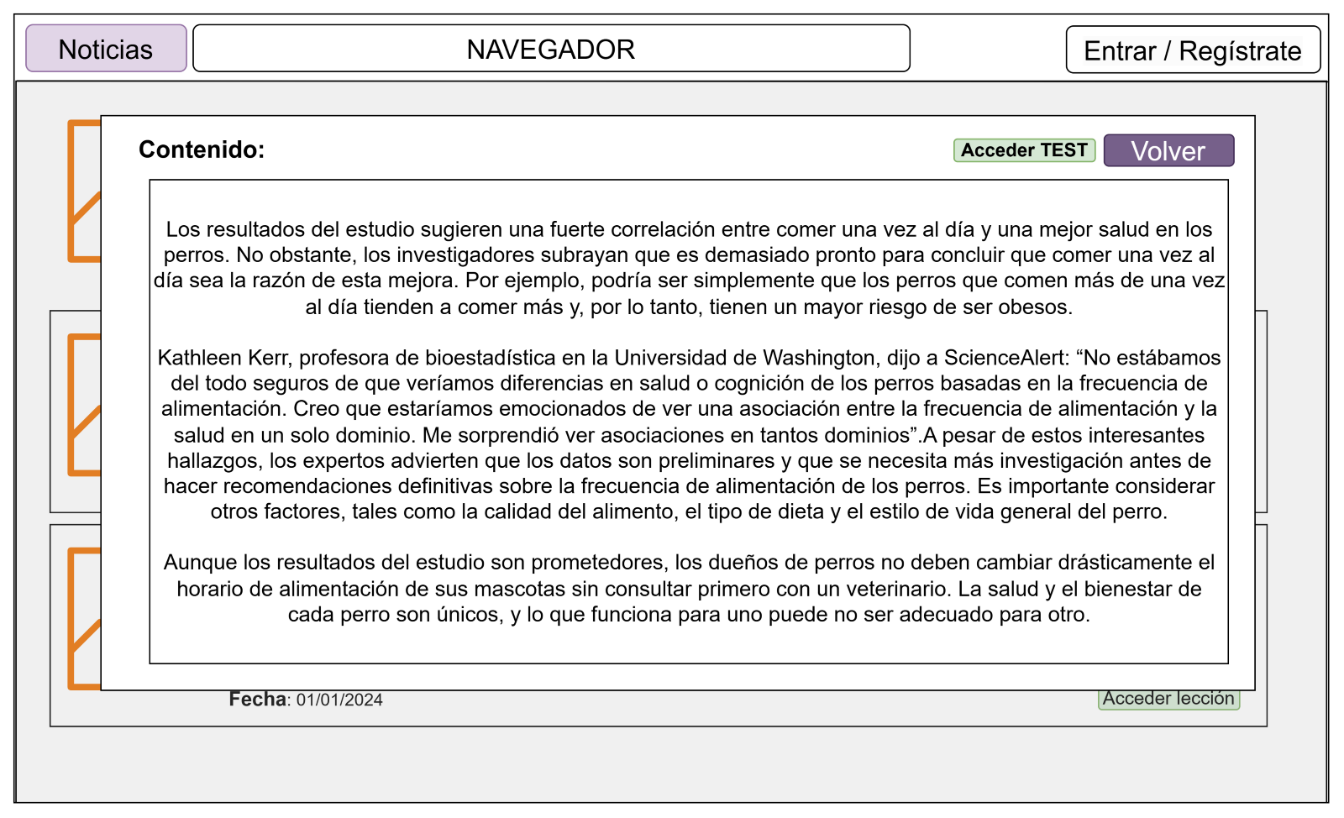


Figura 19 Mockup Noticias Detalle: Vista Usuario Normal/Dueño

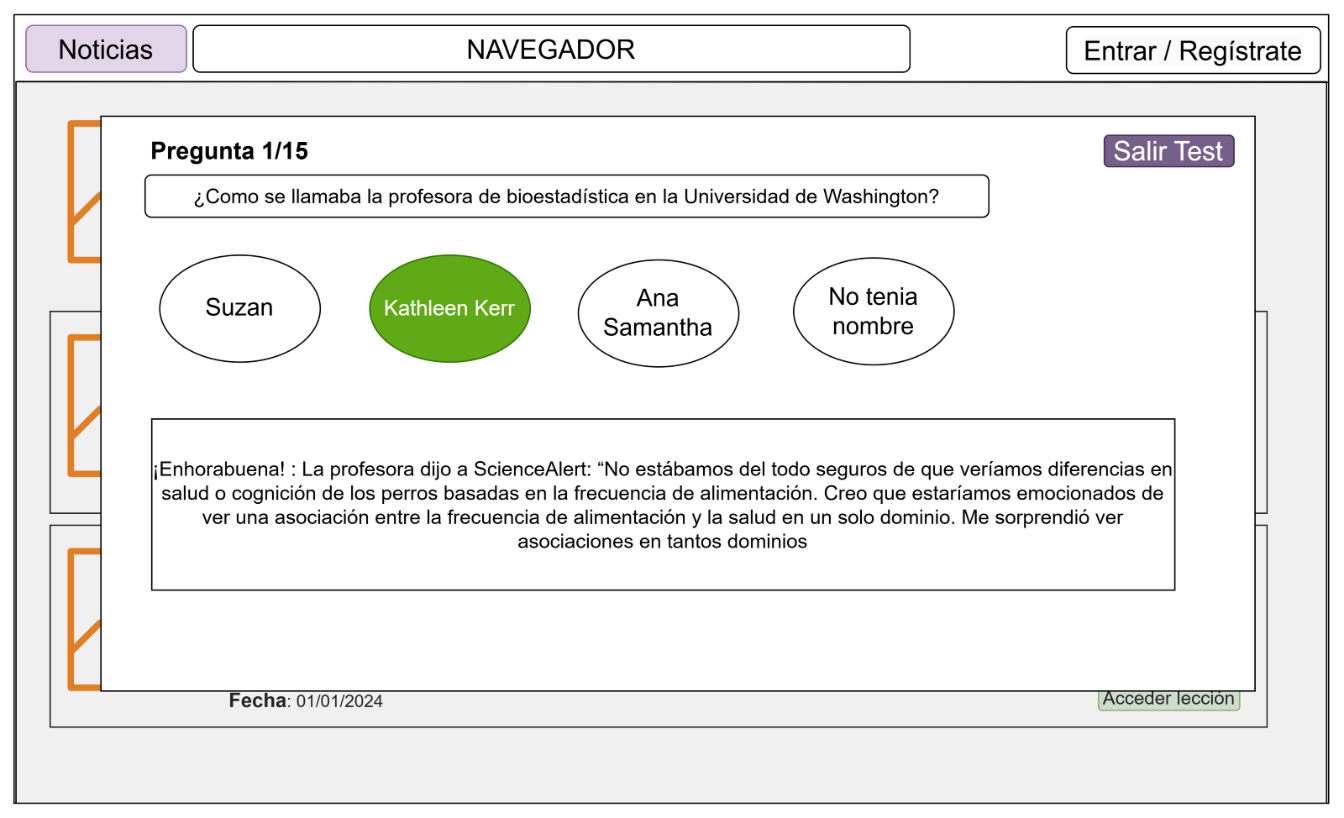


Figura 20 Mockup Noticias Test: Vista Usuario Normal/Dueño

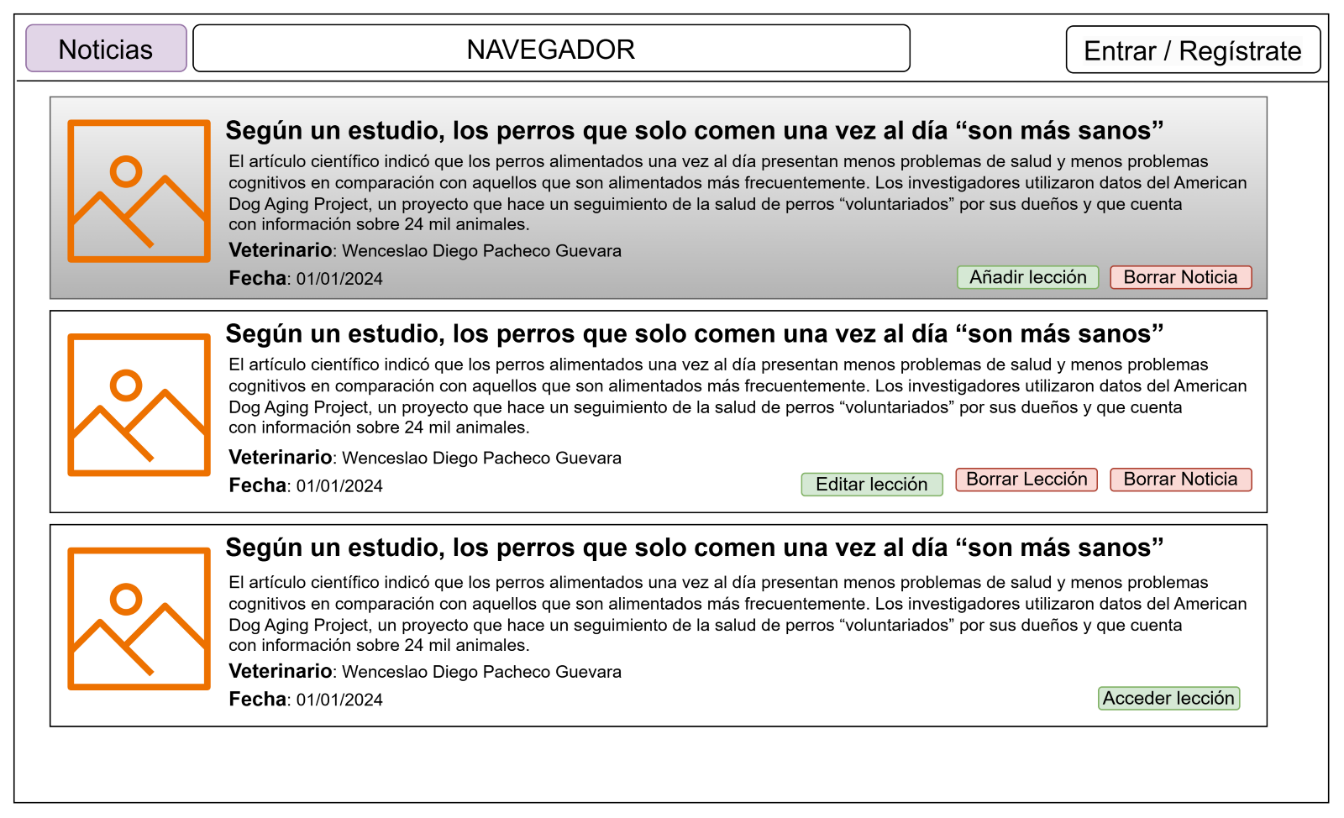


Figura 21 Mockup Noticias Lista: Vista Veterinario/Administrador

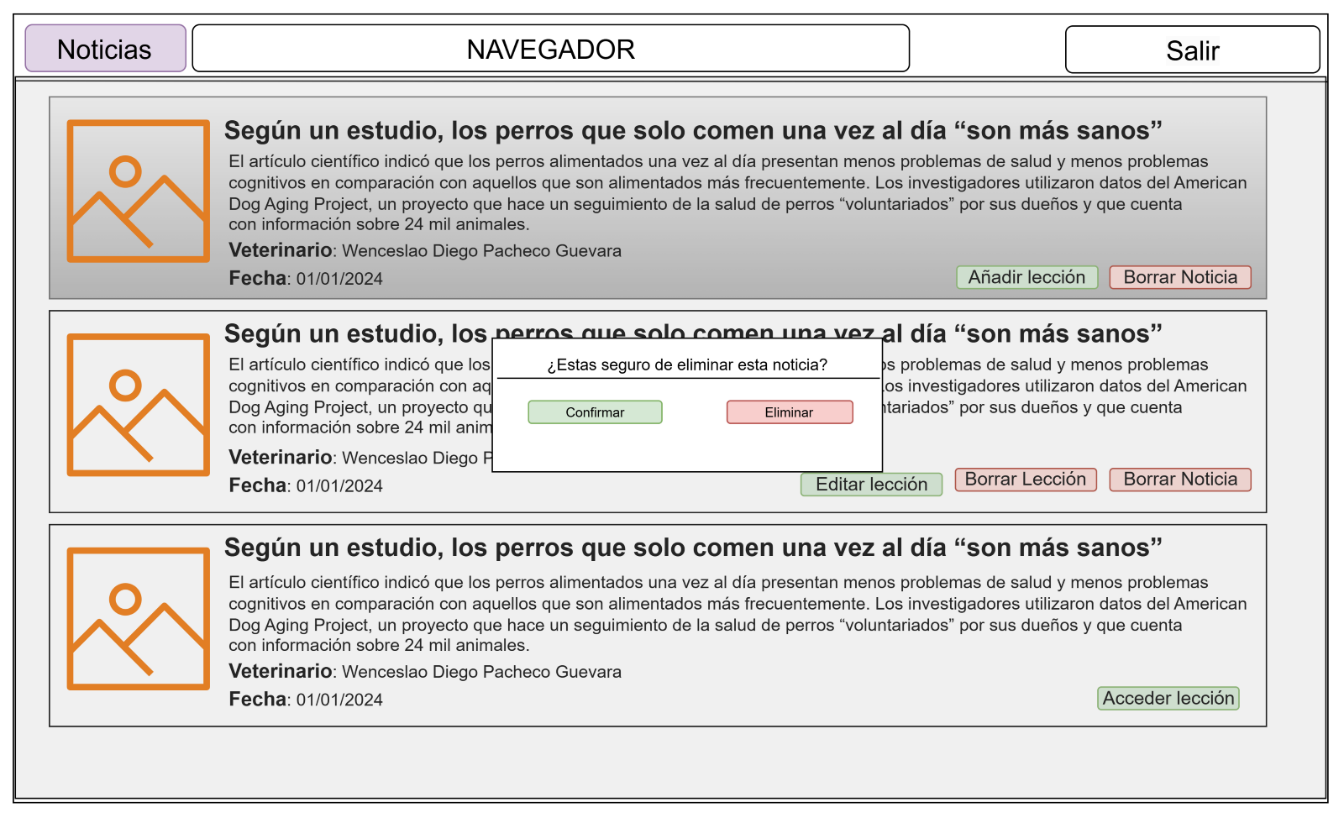


Figura 22 Mockup Noticias Eliminar:Vista Veterinario/Administrador



Figura 23 Mockup Noticias Editar/Añadir Lección: Vista Veterinario/Administrador

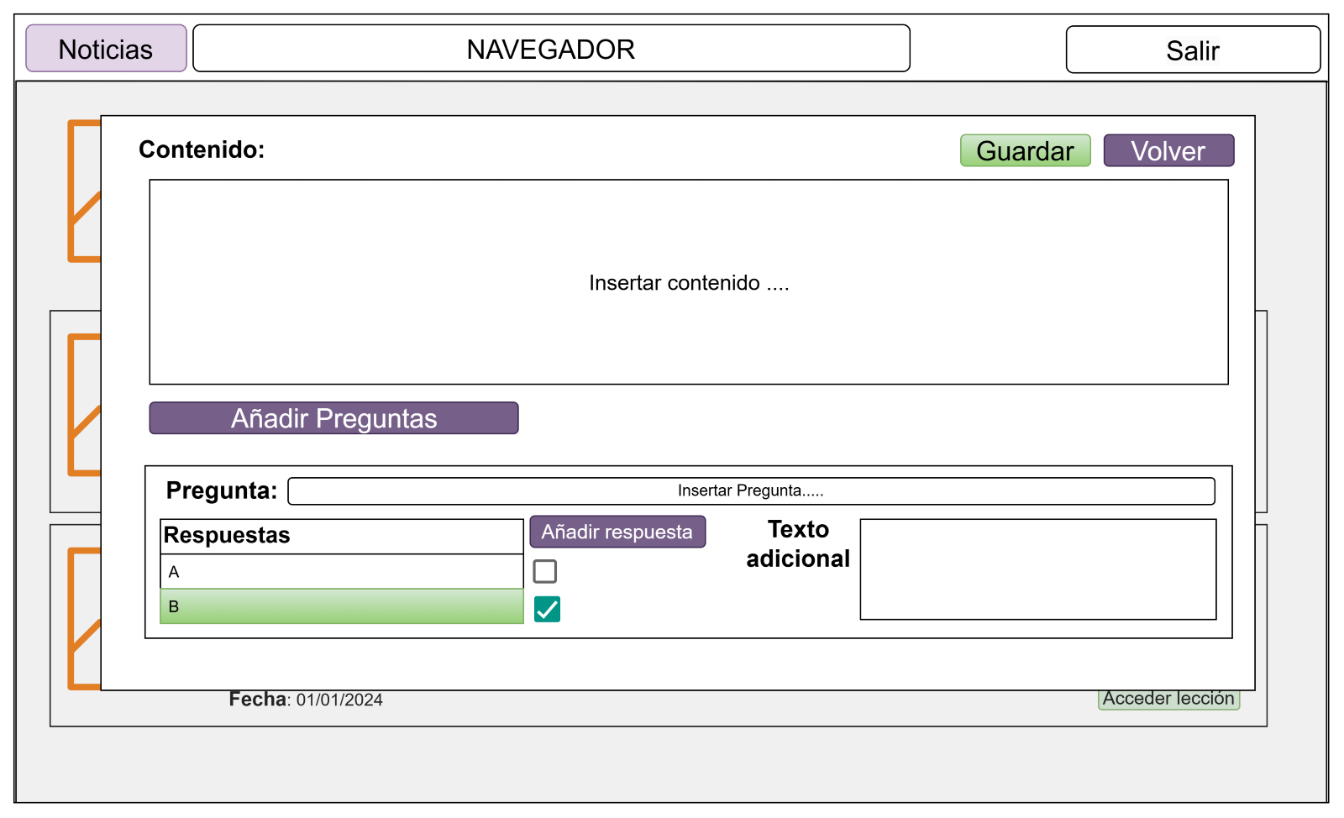


Figura 24 Mockup Noticias Añadir Preguntas: Vista Veterinario/Administrador

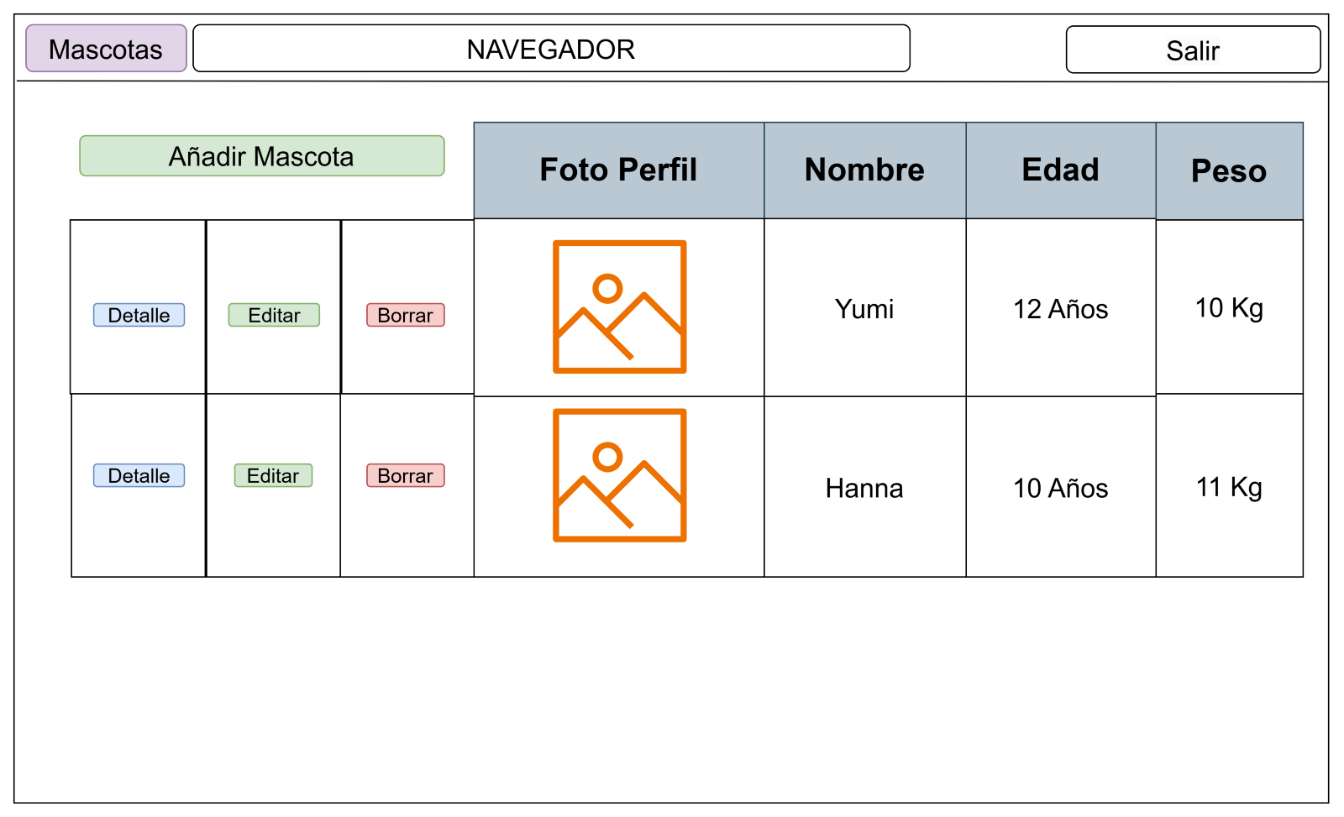


Figura 25 Mockup Mascotas Lista: Vista Dueño

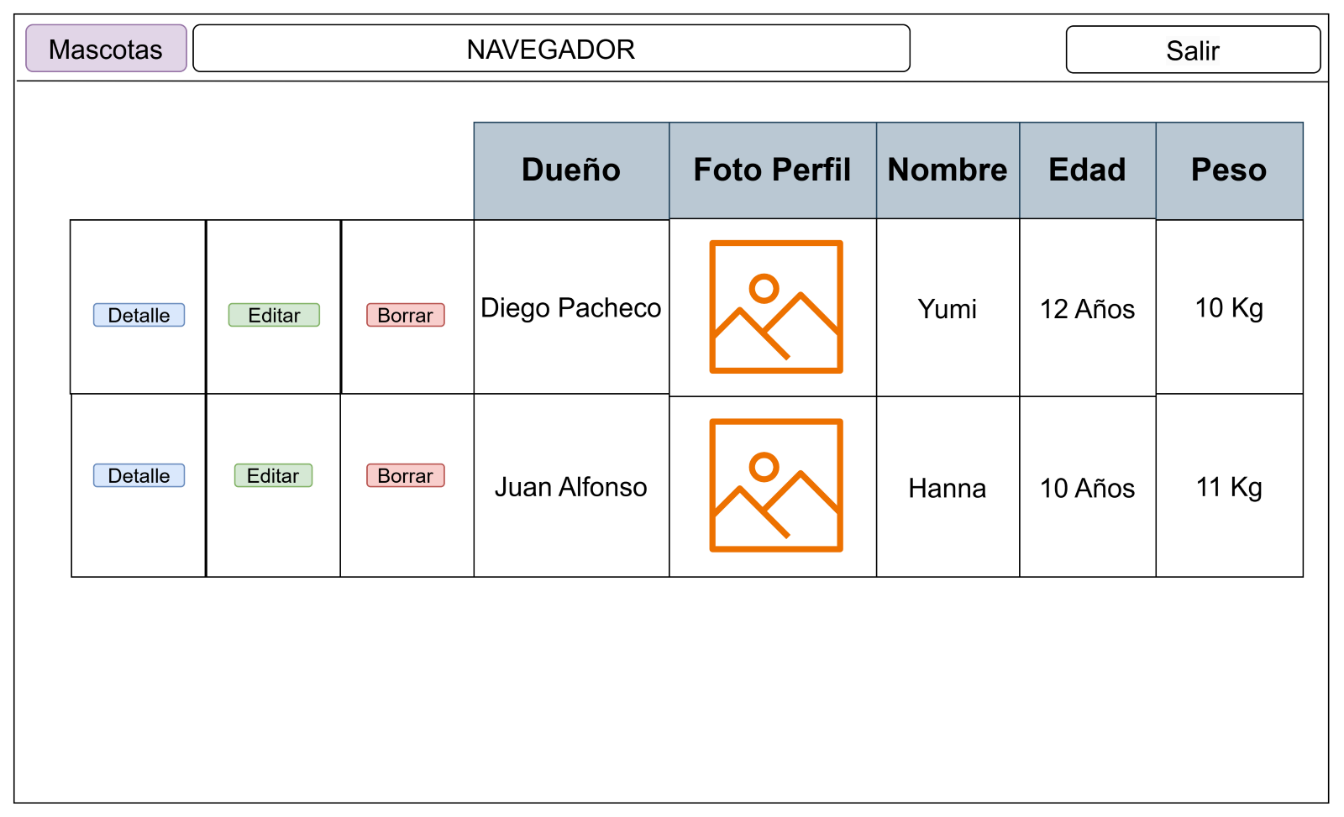


Figura 26 Mockup Mascotas Lista: Vista Veterinario

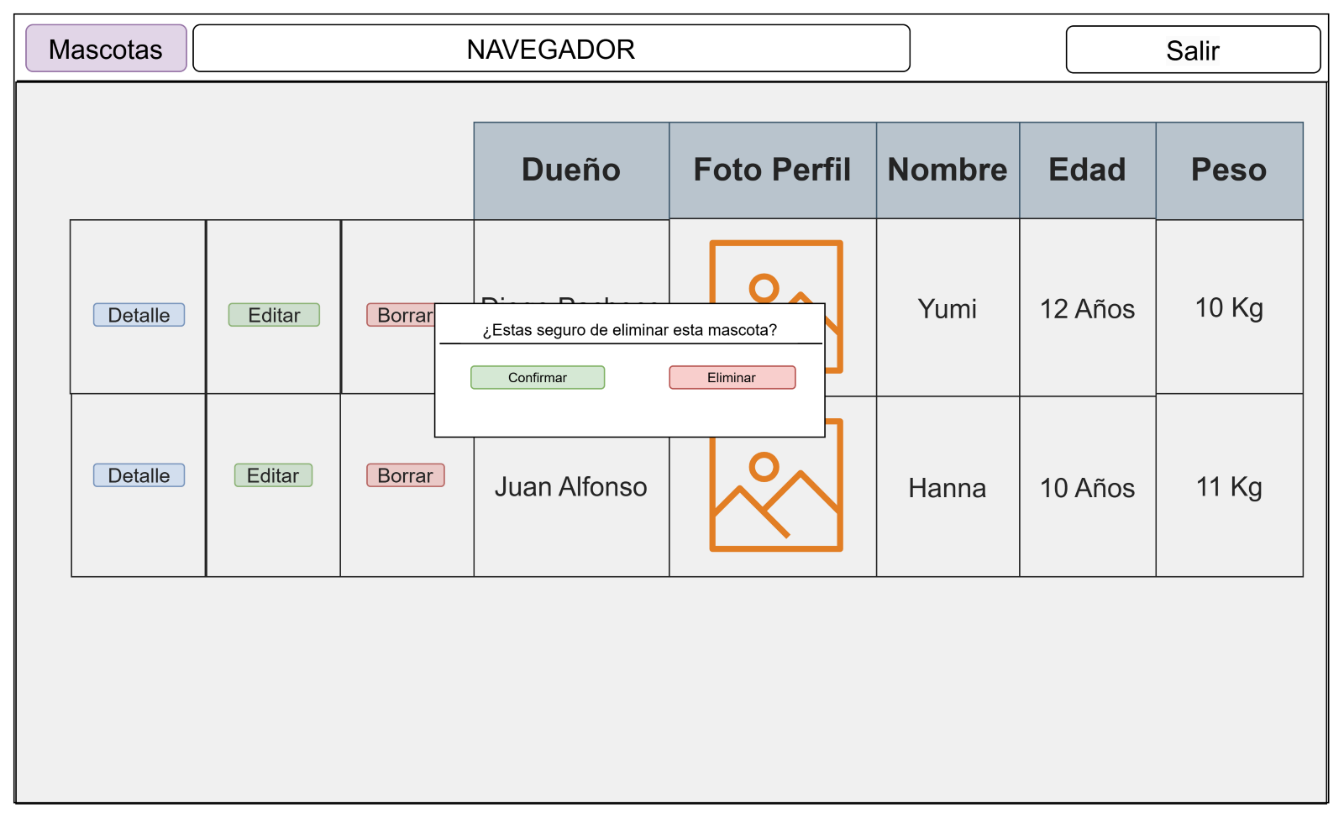


Figura 27 Mockup Mascotas Eliminar: Vista Veterinario/Dueño



Figura 28 Mockup Mascotas Detalle: Vista Dueño/Veterinario

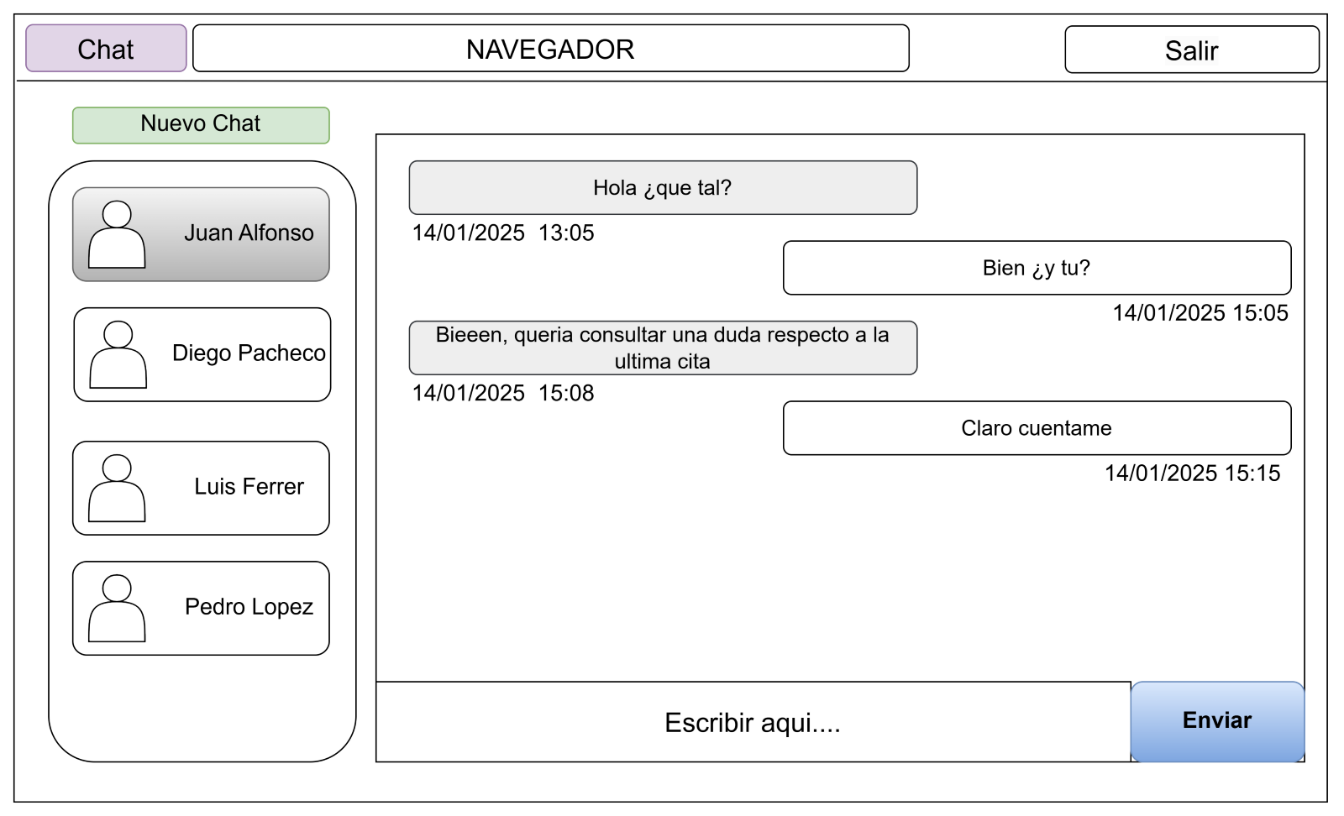


Figura 29 Mockup Chat: Vista Dueño/Veterinario/Administrador

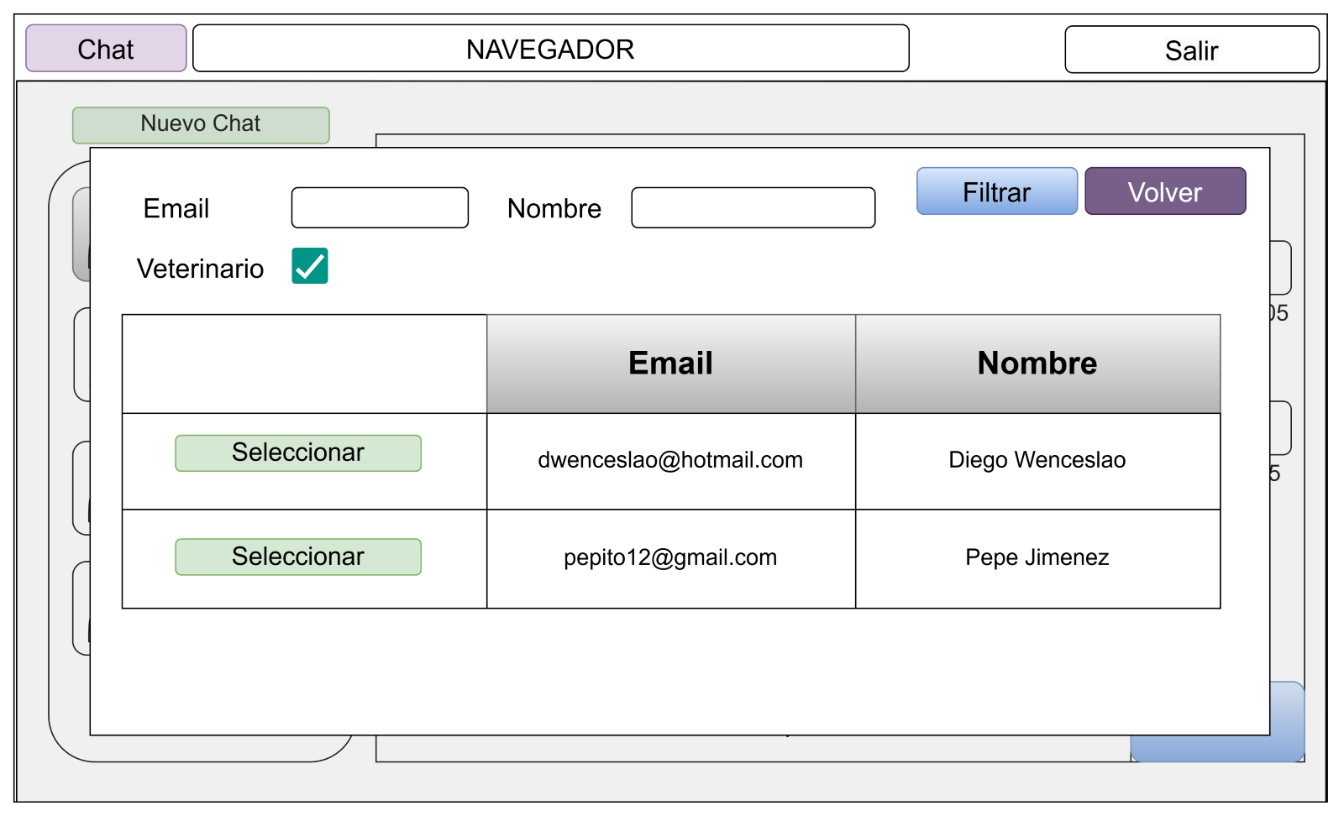


Figura 30 Mockup Nuevo Chat: Vista Dueño/Veterinario/Administrador

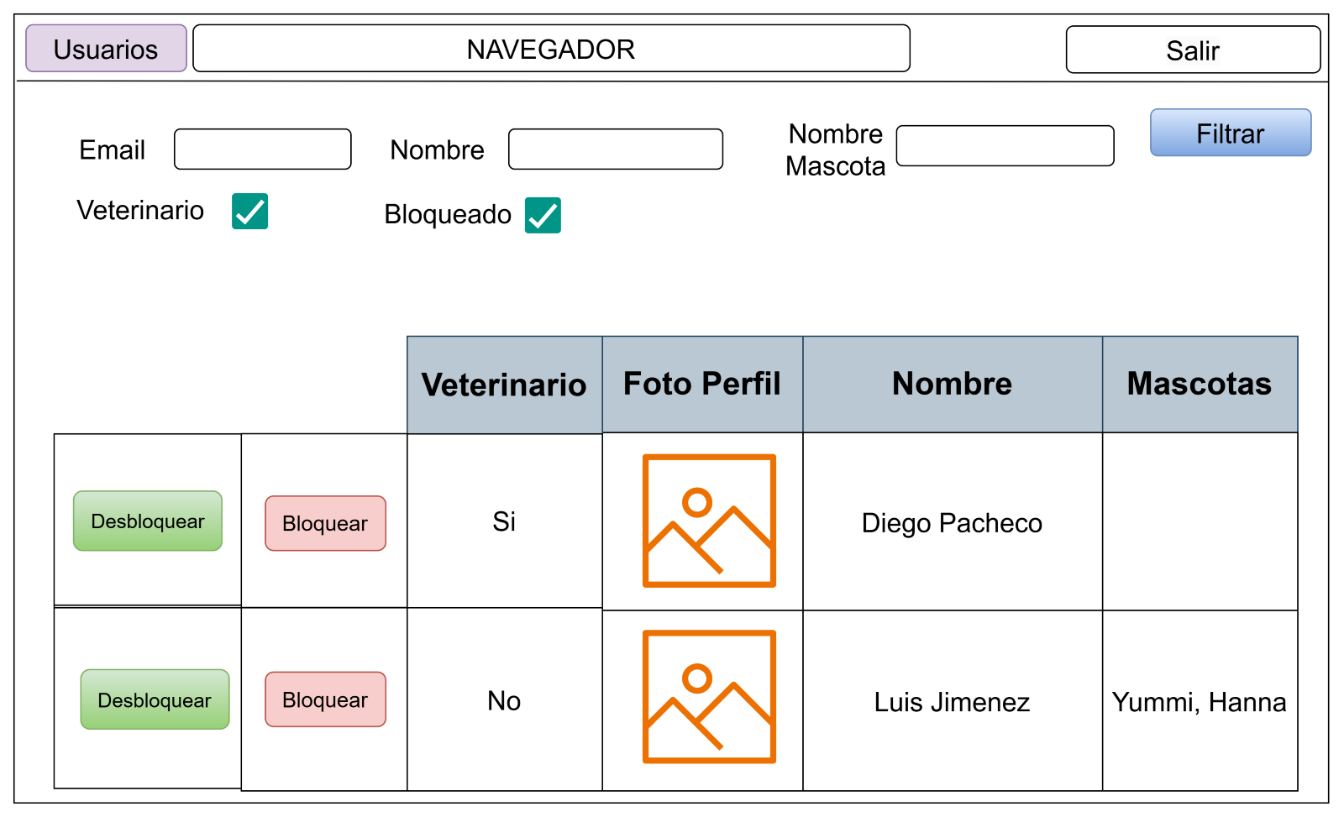


Figura 31 Mockup Usuarios Lista: Vista Administrador



Figura 32 Mockup Citas: Vista Veterinario

## Diseño final

A continuación, se mostrará el diseño final: (Para no saturar la memoria de imágenes se mostrará solo un fragmento de la web).

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Figura 33 Servicios: Vista Usuario Normal

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Figura 34 Servicios Solicitar Cita: Vista Usuario Normal

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

Figura 35 Servicios Lista: Vista Administrador/Veterinario

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Figura 36 Servicios Editar: Vista Veterinario/Administrador

Texto

El contenido generado por IA puede ser incorrecto.

Figura 37 Noticias Lista: Vista Veterinario/Administrador

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura 38 Noticias Detalle: Vista Administrador/Veterinario

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura 39 Noticias Encuesta: Vista Usuario Normal

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Figura 40 Usuarios Lista: Vista Administrador

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Figura 41 Logs Lista: Vista Administrador

# **Arquitectura**

## Modelo Arquitectura en Capas (N-Tier Architecture)

La arquitectura en capas es un patrón de diseño que separa la aplicación en distintas partes (capas), cada una con una responsabilidad específica. Esto ayuda a organizar el código, facilitar su mantenimiento y reutilización.

Ventajas de esta arquitectura:

* Separación de responsabilidades.
* Mayor mantenibilidad.
* Posibilidad de reutilizar la capa de datos.
* Escalable a futuro para migrar a MVC o Web API.

Para este proyecto el número de capas se ha reducido a dos, la primera capa “WebForms” muestra la interfaz al usuario y maneja los eventos del navegador. La segunda capa “Clases” contiene reglas del negocio, validaciones, cálculos, control de flujo y la comunicación con la base de datos.

TFGVeterinaria

|

|\_\_Clases

|

|\_\_WebForms

Texto

El contenido generado por IA puede ser incorrecto.

Figura 42 Arquitectura Proyecto: Nivel Lógica de negocio (BLL) - Acceso a datos (DAL)

Texto

El contenido generado por IA puede ser incorrecto.

Figura 43 Arquitectura Proyecto: Nivel Presentación UI

WebForms - Capa de Presentación (UI):

Es la parte que interactúa con el usuario. En ASP.NET WebForms, esta capa está compuesta por:

* .aspx: HTML + Controles de servidor (<asp:Button>, <asp:GridView>, etc.)
* .aspx.vb: Código que maneja eventos como Button\_Click, Page\_Load, etc.

Clases - Capa de Lógica de Negocio (BLL) y Capa de Acceso a Datos (DAL):

Aquí se encuentra la lógica central de la aplicación:

* Validaciones (¿el usuario ingresó un dato correcto?)
* Reglas del negocio (¿el usuario puede acceder a esta sección de la web?)
* Procesamiento de datos (¿el horario del veterinario es el correcto?)
* Ejecutar SELECT, INSERT, UPDATE, DELETE
* Obtener y guardar datos en objetos

# **Implementación**

## Front End

Como ya he mencionado anteriormente, para este proyecto estoy haciendo uso de archivos con extensión .aspx, este tipo de archivos contiene código HTML mezclado con código del lado del servidor. Para identificar que código NO es HTML, hacemos uso de la palabra ‘asp’ y el atributo ‘runat=”server”’antes de la etiqueta.

Por ejemplo, no es lo mismo “<button> </button>” que “<asp:Button runat=”server”> </Button>”.

Con este tipo de etiquetas podemos obtener resultados como el siguiente:

Texto

El contenido generado por IA puede ser incorrecto.

Figura 44 Fragmento de código de Mascotas\_Detalle.aspx

Con este tipo de etiquetas podemos obtener dos tipos de etiquetas muy interesantes para mostrar listas de datos.

GridView: Es un control de servidor que se utiliza para mostrar, editar, eliminar y ordenar datos, muestra los datos en filas y columnas, similar a una tabla HTML.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Figura 45 Fragmento de Código de LogSistema.aspx, Ejemplo de GridView

Para enlazar los datos con el componente, tenemos que hacer uso de un tipo de dato llamado “DataTable”, este tipo de dato es una estructura en memoria que representa una tabla de datos (similar a una tabla de una base de datos), y forma parte del espacio de nombres System.Data.

Es similar a rellenar una matriz (una lista de una lista), pero más adelante se explicará más a detalle el uso interno de este tipo de dato. Consultar la Figura 41 Logs Lista: Vista Administrador.

A continuación, se muestra un ejemplo sencillo de como rellenar este tipo de dato:

Texto

El contenido generado por IA puede ser incorrecto.

Figura Fragmento de Código de LogSistema.aspx.vb, Ejemplo de enlace de datos con GridView

ListView: Es un control de servidor web para mostrar datos de forma **altamente personalizable**, es mucho más flexible que el GridView.

Un ejemplo sería el siguiente:

Texto

El contenido generado por IA puede ser incorrecto.

Figura Fragmento de Código de Servicios.aspx

En el ejemplo ya podemos observar que en el propio componente podemos utilizar libremente el uso de diferentes etiquetas html para mostrar un registro.

Consultar Figura 35 Servicios Lista: Vista Administrador/Veterinario.

Al igual que en el GridView, el enlace de datos se realiza a través de un DataTable:

Texto

El contenido generado por IA puede ser incorrecto.

Figura 48 Fragmento de Código de Servicios.aspx.vb: Ejemplo de enlace de datos con ListView

Por último, cabe destacar el uso de herencia de páginas aspx, todas las páginas heredan el contenido de una “Pagina Master” el cual contiene el código HTML del menú de Navegación y cada aspx inserta su código HTML en el fragmento “ContentPlaceHolder”.

Texto

El contenido generado por IA puede ser incorrecto.

Figura 49 Fragmento de Código de Site.Master

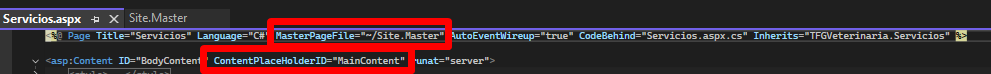


Figura 50 Fragmento de Código de Servicios.aspx: Herencia de código HTML

### CSS

### CSS significa Cascading Style Sheets (en español: Hojas de Estilo en Cascada). Es un lenguaje utilizado para dar estilo y presentación visual a las páginas web escritas en HTML.

Este proyecto hace uso de una librería externa de css que ya ofrece los estilos de una web moderna. Sin embargo, en algunos casos se ha necesitado ajustar el estilo para obtener un resultado más apropiado para la vista (ya que la web se ha diseñado de momento para contemplar el modo oscuro y en algunos casos la librería no mostraba el resultado deseado)

Texto

El contenido generado por IA puede ser incorrecto.

Figura Fragmento de Código de ServiciosCita.aspx: Ejemplo de uso de CSS

### Bootstrap

Bootstrap es, en gran parte, una colección muy grande de estilos CSS ya preparados que se pueden aplicar directamente a la web web para que se vea moderno y profesional, sin tener que escribir manualmente todo el código CSS.

En este proyecto se han descargado todas las librerías necesarias para contemplar la web en modo oscuro. Estas librerías se encuentran en la carpeta “Scripts”.

Para hacer uso de estas librerías, se han importado en el Site.Master de la siguiente manera:

<asp:PlaceHolder runat="server">

<%: Scripts.Render("~/Scripts/bootstrap.js") %>

</asp:PlaceHolder>

De forma que todas las páginas ASPX contengan la hoja de estilos de Bootstrap.

Con la ayuda de Bootstrap podemos abreviar muchos estilos como clases, por ejemplo “d-flex” es lo mismo que “display: flex” o “mb-2” es los mismo que “margin-bottom: 2px”.

A continuación, se muestra un pequeño fragmento de como se consume esta librería en los archivos .aspx.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Figura Fragmento de Código Mascotas\_Detalle.aspx

### Javascript y Jquery

JavaScript es un lenguaje de programación que se usa principalmente para hacer que las páginas web sean interactivas y dinámicas. Funciona en el navegador del usuario, junto con HTML y CSS.

jQuery es una librería de JavaScript. Es decir, es un conjunto de funciones ya escritas que simplifican el uso de JavaScript. Las librerías de jquery se han guardado al igual que las de boostrap en la carpeta “Scripts” del proyecto y se han importado de la siguiente manera:

public static void RegisterJQueryScriptManager() {

ScriptManager.ScriptResourceMapping.AddDefinition("jquery",

new ScriptResourceDefinition

{

Path = "~/scripts/jquery-3.7.0.min.js",

DebugPath = "~/scripts/jquery-3.7.0.js",

CdnPath = "http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.7.0.min.js",

CdnDebugPath = "http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.7.0.js"

});

}

Esta función es llamada por su constructor de clase que a su vez es llamada por el archivo global de configuración Global.asax (es el archivo global de configuración y contiene eventos como Application\_start (Se ejecuta una sola vez, justo cuando la aplicación se inicia por primera vez en el servido)), esta función registra la palabra “jquery” para indicar que hace referencia a esas librerías. Luego en el Site.master del cual heredan todas las páginas pueden hacer uso de la siguiente línea para importar la librería: <asp:ScriptReference Name="jquery" />

En este proyecto jquery y javascript se ha utilizado para la lógica de confirmación para eliminar un registro (cuando elimines un registro que abra un popup personalizado pidiendo la confirmación del usuario).

Texto

El contenido generado por IA puede ser incorrecto.

Figura Fragmento de código de Noticias.aspx

### Peticiones HTTP

Normalmente para navegar en páginas ASPX en la url se suele escribir el dominio + la ruta del archivo aspx correspondiente, como por ejemplo:

<https://TFGVeterinaria/WebForms/Noticias/Noticias.aspx>

o bien

<https://TFGVeterinaria/WebForms/Noticias/Noticias_Detalle.aspx>

para simplicar las URL se ha utilizado un enrutador de .NET. RouteTable es una clase estática que pertenece al espacio de nombres System.Web.Routing. Su función principal es almacenar la colección de rutas (RouteCollection) que utiliza la aplicación web para decidir cómo se deben enrutar las solicitudes entrantes

Texto

El contenido generado por IA puede ser incorrecto.

Figura 54 Fragmento de Código de RouteConfig.cs

Si tenemos definido el enrutador, podemos utilizar url más sencillas como por ejemplo:

<https://TFGVeterinaria/Noticias>

o bien

<https://TFGVeterinaria/Noticias_Detalle/25>

Y la forma de utilizar este tipo de redireccionamiento internamente en el código es a través de la función que ya ofrece .NET “RedirectToRoute”.

protected void gvLecciones\_RowCommand(object sender, GridViewCommandEventArgs e) {

int index = Convert.ToInt32(e.CommandArgument);

if (e.CommandName == "Acceder") {

Response.RedirectToRoute("noticiasDetalleRouteParam", new { ID = index });

}

}

Otro punto importante para mencionar son los validadores http que ofrece .NET, como <asp: RegularExpressionValidator> para validar el formato de un email a través de expresiones regulares o bien <asp:RequiredFieldValidator> para indicar si un campo es obligatorio, además podemos utilizar estos validadores para realizar únicamente un postback cuando se hayan cumplido todos los requisitos de los validadores.

Estos validadores funcionan como etiqueta más de ASP.NET, en el cual a través de los atributos reciben los parámetros que tienen que validar o sobre que campo tienen que validar.

Podemos encontrar diferentes ejemplos de validadores en Registro.aspx.

Texto

El contenido generado por IA puede ser incorrecto.

Figura 55 Fragmento de código de Registro.aspx

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura 56 Registro: Validadores .NET

## Back End

Como ya se ha mencionado anteriormente, en este proyecto existen dos capas que manejan lógicas diferentes, en la Figura 46 Fragmento de Código de LogSistema.aspx.vb ya vimos como se enlazaban los datos con las etiquetas de .NET. Es importante mencionar que la centralización de Logs se realiza principalmente en la primera capa UI (WebForms) y es que todas las funciones que contengan una llamada a la segunda capa (Clases), están rodeadas de un try-catch para poder controlar todas las excepciones que ocurren inesperadamente.

Texto

El contenido generado por IA puede ser incorrecto.

Figura 57 Fragmento de Código de LogSistema.aspx

En cuanto a la estructura de la segunda capa (Clases) se rige por la funcionalidad que ofrece y por la página a la que afecta, es decir, existe una clase por cada opción de menú que tiene la web (Noticias\_Class, Citas\_Class, Comunidad\_Class, Mascotas\_Class, Logs\_Class, Usuarios\_Class, Chat\_Class, Servicios\_Class, Login\_Class) y existe una clase genérica (Librería) el cual contiene funciones que sirven para cualquier página.

Un ejemplo de clase sería el siguiente:

Texto

El contenido generado por IA puede ser incorrecto.

Figura 58 Fragmento de Código de Login\_Class.cs: Función ExisteUsuarioBloqueado

Las funciones de las clases tienen como propósito recoger funcionalidades básicas concretas, como comprobaciones y CRUD, es decir, las clases cuentan con una alta granularidad.

## IIS (Internet Information Services)

Aquí hablar de como funciona el servidor que lanza el proyecto de desarrollo y como hacemos una publicación para que corra en una red lan

## Base de Datos

Aquí hablar un poco de como se ha creado la base de datos y como hemos creado un backup de nuestra base de datos

## Esquema Entidad Relación

Aquí explicar un poco por encima las relaciones más notables y mostrar el esquema del dbeaver

## Acceso a Datos

Aquí hablar de como esta hecha la conexión con la base de datos dentro del código y como se accede

# **Seguridad**

## Cifrado de las contraseñas

Aquí hablar de como están hechas las contraseñas

# **APIS Utilizadas**

Aquí hablar de como esta hecha la api que utilizamos para actualizar las noticias

# **Pruebas**

Aquí hablar de como tenemos configurado el selenium

# **Trabajos futuros y Posibles mejoras**

Aquí hablar de futuros requisitos que podríamos tener en cuenta y mejoras respecto al front end y la posibilidad de dividir la ultima capa en dos capas diferentes

# **Conclusiones**

obligatoriamente se incluirá una sección de conclusiones donde se

realizará un resumen de los objetivos conseguidos así como de los resultados

obtenidos si proceden

# **Bibliografía y referencias**

se incluirá también la relación de obras y materiales

consultados y empleados en la elaboración de la memoria del TFG/TFM. La

bibliografía y las referencias serán indexadas en orden alfabético (sistema nombre

y fecha) o se numerará correlativamente según aparezca (sistema numérico). Se

empleará la familia 1 como tipo de letra. Podrá utilizarse cualquier sistema

bibliográfico normalizado predominante en la rama de conocimiento,

estableciéndose como prioritarios el sistema ISO 690, sistema APA (American

Psychological Association) o Harvard (no necesariamente en ese orden de

preferencia).

1. Diagrams.net. (s.f.). *diagrams.net*. <https://www.draw.io/>
2. PostgreSQL Global Development Group. (s.f.). PostgreSQL: *The world's most advanced open source relational database*. <https://www.postgresql.org/>