

CAPITULO I EL PROYECTO

I.1 PERSONAL VINCULADO AL PROYECTO

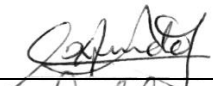
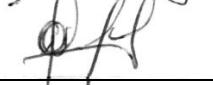
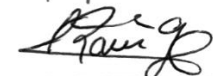
I.1.1 DIRECTOR DE PROYECTO

SANCHEZ Apellido Paterno	GARAY Apellido Materno	ROLANDO ABELINO Nombre	7160684 C.I.
1 Grupo de Taller III	INGENIERÍA INFORMÁTICA Carrera	FACULTAD DE CIENCIAS Y TECNOLOGÍA Facultad:	
----- Teléfono	75149966 Celular	rolando.sanchez.developer@gmail.com Correo electrónico	 Firma

I.1.2 PARTICIPANTES EQUIPO DE TRABAJO

Categoría	Nombres y Apellidos	Profesión	C.I.	Firma
DIRECTOR	Rolando Abelino Sanchez Garay	Estudiante Universitario	7160684	
TUTOR	Lic. Octavio D. Aguilar Mallea	Ing. Informático		

I.1.3 EQUIPO DE TRABAJO DE: EMPRESAS/INSTITUCIONES/ORGANIZACIONES PARTICIPANTES/COOPERANTES

Nombre: Clínica Veterinaria “AMEVET”			
Dirección: Av. Julio Arce Castillo #0955 entre c/ Francisco Uriondo y Hermanos Ruiloba		Teléf. Oficina: 66-3278 - 72947883	
Nombre y Apellidos	Cargo	C.I.	Firma
Dra. Ximena Jurado L.	Administrador	5055431 Tja.	
Dra. Nehyda Ramos	Médico Veterinario	7179292 Tja.	
Nataly Jurado	Auxiliar	7116241 Tja.	

I.1.4 ACTIVIDADES PREVISTAS PARA LOS INTEGRANTES DEL EQUIPO DE INVESTIGACIÓN

Responsable	Actividades
Director	Dirigir el proyecto Análisis y Diseño de sistemas Programación Creación de nuevos modelos
Tutor	Orientación y Seguimiento al proyecto

I.2 DESCRIPCIÓN DEL PROYECTO

I.2.1 RESUMEN EJECUTIVO DEL PROYECTO

Hoy en día, la informática en red se ha convertido en un factor importante en la vida de una empresa la razón principal implica la cantidad de información que actualmente se maneja, hace que el tratamiento automático de la información sea realmente útil y necesario y este es el caso de la clínica veterinaria “AMEVET”.

Las funciones que realiza la clínica veterinaria “AMEVET” son: atender a los pacientes/mascotas para que reciba los servicios de cirugía o tratamiento, control sanitario, peluquería y estética canina. Estos servicios los contrata el dueño de la mascota asistiendo a la clínica personalmente y solicitando una cita si es necesario.

La administración de la información de los pacientes/mascotas, organización de citas y distribución de noticias son actividades primordiales de la clínica.

Tanto la comunicación oportuna con los clientes como la administración de la información de los pacientes/mascotas y la organización de citas, requieren de estrategias en las que los principales criterios sean la rapidez y la completitud, criterios que actualmente no se manifiestan debido principalmente a la falta de recursos y a la falta de un modelo formal.

Las Tecnologías de Información y Comunicación (TIC), y principalmente la tecnología móvil, son servicios que hacen de la comunicación una actividad certera y que cumple con los criterios más importantes a bajo costo, la rapidez y la completitud. La rapidez porque la información que se publica puede ser vista casi al instante, y la completitud, por el bajo costo que implica “subir” mayor información enriquecida con imágenes.

El objetivo principal del presente proyecto, es mejorar la administración de la información de los pacientes de la clínica veterinaria “AMEVET”, asistida por una plataforma tecnológica de comunicación con los clientes, para ello, se desarrollarán dos modelos tecnológicos (sistema

informático, aplicación móvil), además de plantear una socialización de los modelos que permitan crear un sentido de pertenencia para lograr los objetivos propuestos.

I.2.2 DESCRIPCIÓN Y FUNDAMENTACIÓN DEL PROYECTO

El proyecto tiene como propósito mejorar la administración de la información de los pacientes de la clínica veterinaria “AMEVET”, planteando para el efecto, el desarrollo de tres modelos importantes para encarar la inserción de las TIC en los procesos de administración de los pacientes y comunicación con los clientes. El primer modelo se refiere al desarrollo de un sistema computacional que soporte los procesos de administración de pacientes, organización de citas y publicación de noticias, utilizando tecnología de punta. El segundo modelo está referido al desarrollo de una aplicación móvil destinada para el uso de los clientes e integrarla con el sistema del modelo anterior, siendo la plataforma ANDROID la elegida debido a que es la actual tendencia en el mercado de Teléfonos inteligentes. El tercer modelo está referido a la socialización de los dos modelos anteriores.

Los tres modelos planteados son importantes y necesarios, debido principalmente a que la actual forma de administración de los pacientes, organización de citas y distribución de noticias no tiene la eficiencia esperada, haciendo que la imagen institucional se vea disminuida presumiendo que los motivos más importantes son que ha crecido la clientela con respecto a años anteriores y se complica la administración de la información de manera manual (papeles).

I.2.2.1 Análisis de Causas del Problema

El análisis hecho en la clínica veterinaria “AMEVET”, han identificado claramente falencias que afectan el manejo de la información de sus pacientes/mascotas, como la ficha de control sanitario, las atenciones, cirugías, tratamientos, peluquerías y estética canina. Otra falencia importante es la organización de las citas de la agenda de la clínica y alguna manera de promocionar algunas noticias de la clínica como promociones, eventos, anuncios, etc. En el ámbito de los clientes ha habido sugerencias por el motivo que en ciertas ocasiones olvidan llevar a sus mascotas a los controles periódicos determinados, en otras ocasiones pierden o dañan la ficha de control sanitario de su mascota y una sugerencia adicional fue que les ayuden de alguna otra forma en el caso de

cuando extravían a sus mascotas puesto que a veces la distribución de anuncios no es suficiente para que puedan encontrar a su mascota.

Por lo tanto, estos aspectos son importantes para llevar a cabo este proyecto y así satisfacer los requerimientos de los usuarios. Debido a que actualmente la clínica veterinaria “AMEVET” no cuenta con un sistema informático que coadyuvé en estas tareas con una herramienta de control que colabore a superar estas falencias, optimizar la toma de decisiones que está estrechamente relacionado con los procesos de planificación y control, logrando así sus objetivos.

De acuerdo a esto, propongo la elaboración de un sistema de información para la gestión de una clínica veterinaria “Neg-VET” es un sistema que se encarga de la gestión y control de los datos de los pacientes, la organización de las citas de la clínica veterinaria “AMEVET” con el que se manejará el negocio de manera más fácil, rápida y sin pérdida de tiempo.

El sistema automatiza el manejo de la información médica en lo que tiene que ver con la atención que se les da a los pacientes además de manejar cada uno de los parámetros necesarios para llevar a cabo las operaciones de la veterinaria tales como peluquerías, control sanitario, citas y la publicación de noticias.

Así mismo se implementará una aplicación móvil ANDROID destinada a los clientes, con el objetivo de que la clínica pueda disponer de una nueva forma de comunicación con sus clientes.

Con los problemas y objetivos claramente detectados se lo explicara de una manera gráfica mediante el uso de un árbol de problemas y objetivos.

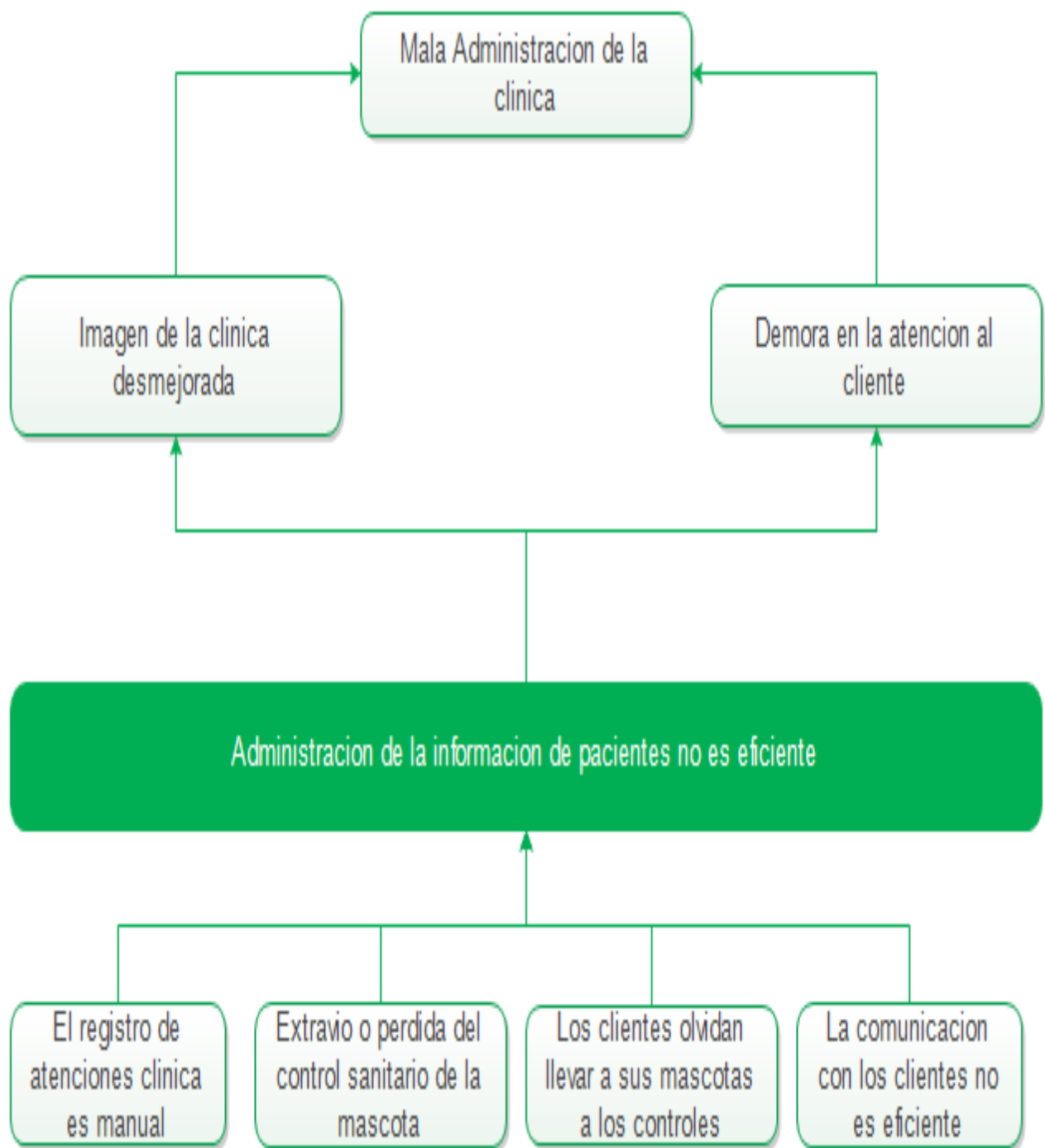


Ilustración I.1 Árbol de Problemas

I.2.2.2 Análisis de Objetivos

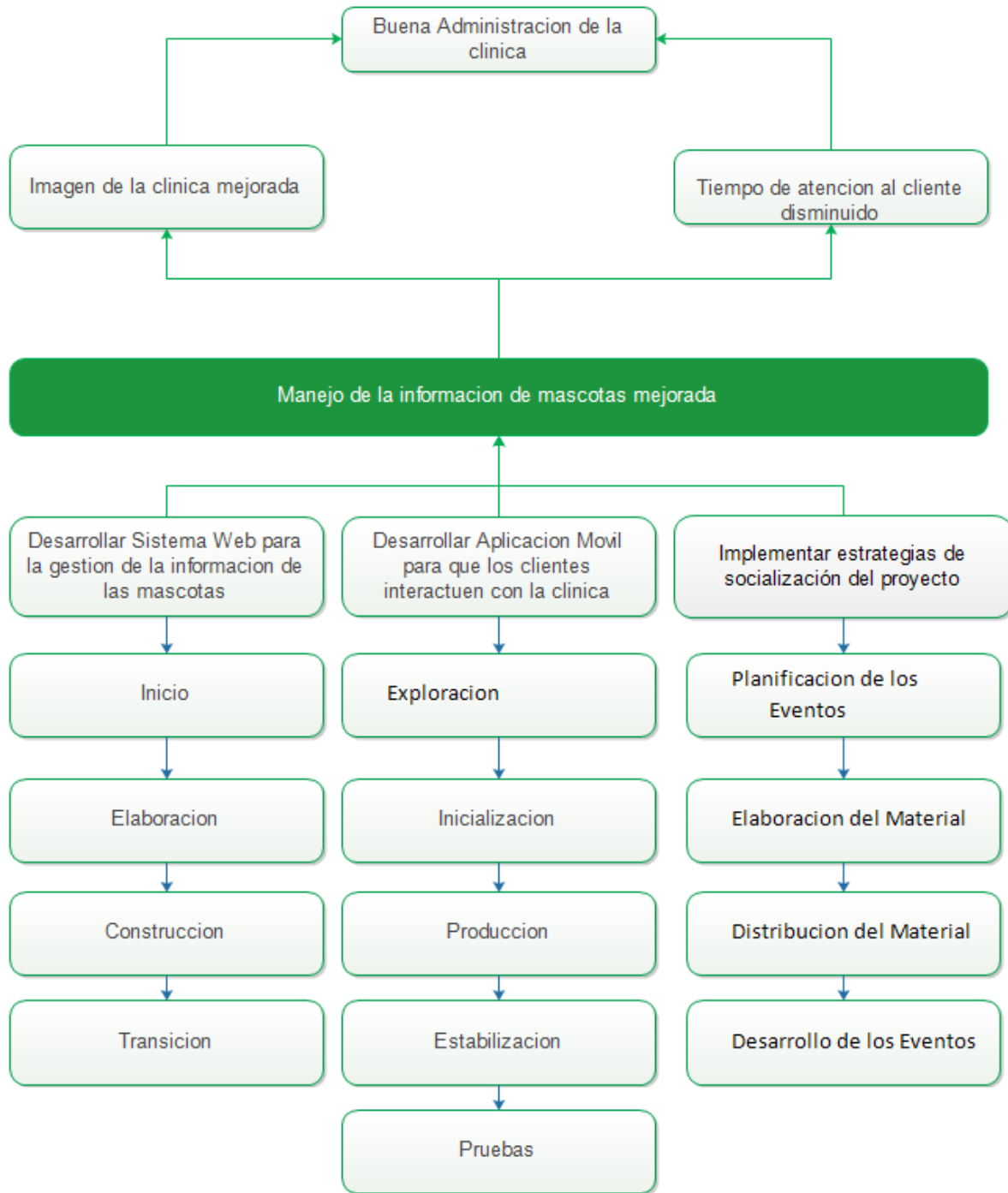


Ilustración I.2 Árbol de Objetivos

I.2.2.3 Situación planteada con y sin proyecto

Situación de la empresa Sin proyecto

- La información de las mascotas no está disponible en cualquier momento ni lugar.
- Frecuentemente olvidan llevar a sus mascotas a los controles periódicos determinados.
- En ocasiones clientes con quejas en la atención.
- Pérdida de Clientes.
- La ayuda que se ofrece a los clientes cuando extravían su mascota no es muy efectiva.
- Sistemas de citas tiene posibilidades de mejora.
- Poca promoción de las noticias y eventos de la clínica.
- Obtención de los datos clínicos de las mascotas es morosa.
- En ocasiones extravío o daño de la ficha de control sanitario.
- La elaboración documentos de control de mascotas es morosa.
- El registro de peluquerías y estética es moroso.
- El reporte de los ingresos de la clínica por atención clínica y peluquería es moroso.

Situación de la empresa con proyecto

- La información de las mascotas disponible desde cualquier lugar y el mayor tiempo posible.
- Los clientes llevan sus mascotas a los controles periódicos regularmente.
- Clientes más satisfechos con la atención.
- Captar más de Clientes.
- Una forma más efectiva para ayuda a los clientes cuando extravían a su mascota.
- Sistema de citas más automatizado.
- Mayor publicidad de las noticias y eventos de la clínica.
- Obtención de los datos clínicos de las mascotas es más rápida.
- La ficha de control sanitario de la mascota está disponible de manera digital y física.
- La elaboración documentos de control de mascotas es más rápida.
- El registro de peluquerías y estética es más rápido.
- Reporte de los ingresos económicos por atención clínica y peluquería es más rápida y eficiente.

I.2.3 OBJETIVOS

I.2.3.1 Objetivo General

Mejorar la administración de la información de los pacientes de la clínica veterinaria AMEVET, incorporando una plataforma de comunicación con los clientes apoyada por las TIC.

I.2.3.2 Objetivos Específicos

- Desarrollar un sistema informático para mejorar la administración de los pacientes, la organización de citas y la distribución de noticias de la clínica veterinaria AMEVET.
- Desarrollar e integrar con el nuevo sistema informático una aplicación móvil ANDROID para asistir la comunicación con los clientes de la clínica veterinaria AMEVET.
- Implementar estrategias de socialización del sistema informático y la aplicación móvil.

I.2.4 MARCO LÓGICO DEL PROYECTO

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Fin Contribuir a mejorar la calidad de atención de la clínica veterinaria “AMEVET”.	Luego de 2 años de finalizado el proyecto, la clínica veterinaria “AMEVET” mejoro la cantidad de clientes satisfechos con la atención brindada en al menos un 40% con respecto al año base (2015).	Cuadro comparativo de encuestas realizadas en el año base y luego de dos años, con referencia a la Información disponible para el cliente, avalado por la clínica veterinaria.	Se mantienen las condiciones tecnológicas y administrativas necesarias para el funcionamiento del Sistema.
Objetivo General La administración de la información de los pacientes de la clínica veterinaria AMEVET ha sido mejorada.	Al finalizar el proyecto, al menos un 60% de los clientes de la clínica veterinaria “AMEVET” han obtenido la información médica de sus mascotas de manera ágil y pertinente.	Reporte del sistema informático, avalado por el gerente de la clínica veterinaria.	La clínica veterinaria presupuesta anualmente la actualización y mantenimiento del sistema Informático.
Objetivos Específicos 1. Sistema informático para la clínica veterinaria desarrollado.	Al finalizar el proyecto, se ha desarrollado un sistema informático para mejorar el control de los pacientes de la clínica veterinaria, basado en los requerimientos expresados bajo la norma IEEE830.	Informe de entrega final y conformidad del Sistema Informático, avalado por el gerente de la clínica Veterinaria.	La clínica veterinaria ha dispuesto el uso formal del sistema informático.

2. Aplicación móvil ANDROID desarrollada e integrada con el nuevo sistema informático.	Al final del proyecto, se ha desarrollado una aplicación móvil para mejorar la comunicación con los clientes de la clínica veterinaria.	Informe de entrega final y conformidad de la aplicación móvil, avalado por el gerente de la clínica Veterinaria.	La mayoría de los clientes tengan acceso a internet y cuenten con Smartphone con Sistema Operativo Android.
3. Estrategias de socialización del sistema informático y de la aplicación móvil implementadas.	Al finalizar el proyecto, se han implementado al menos dos programas de capacitación en el uso del sistema de información. Al finalizar el proyecto, se han distribuido una tarjeta de presentación a cada persona que visitó la clínica, con información adicional sobre la aplicación móvil de la clínica veterinaria.	Informe del gerente de la clínica respecto a la capacitación realizada, adjuntando lista de participantes, algunas fotografías y el modelo del certificado de capacitación. Informe del gerente de la clínica con referencia a la distribución de las tarjetas de presentación, adjuntando modelo de las mismas.	
Actividades Sistema informático para el control de los pacientes de la clínica veterinaria desarrollado. - Inicio - Elaboración - Construcción - Transición	Resumen presupuesto Servicios personales Bs.20.000,00. - Servicios No personales Bs.329,00. - Total: Bs.20.329.-	Informe presupuestario del proyecto, avalado por el director.	Los desembolsos se realizan de acuerdo al cronograma.

Aplicación móvil ANDROID desarrollada e integrada con el nuevo sistema informático. - Exploración - Inicialización - Producción - Estabilización - Pruebas Estrategias de socialización del sistema informático y de la aplicación móvil implementadas. - Planificación de los eventos - Elaboración del material - Distribución del material. - Desarrollo de los eventos			
---	--	--	--

I.2.5 METODOLOGÍAS DE TRABAJO

La metodología propuesta por el trabajo, está dividida en 3 partes, la primera tiene que ver con el desarrollo del sistema web de información, la segunda parte con el desarrollo de la aplicación móvil para los clientes y la tercera con la socialización.

I.2.5.1 Metodología RUP (Rational Unified Process)

La Metodología para el desarrollo del sistema informático es el RUP:

RUP (Proceso Unificado Racional) es un proceso para el desarrollo de un proyecto de un software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Como 3 características esenciales está dirigido por los Casos de Uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la arquitectura: que Relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es iterativo e incremental: divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

Como filosofía RUP maneja 6 principios clave:

1. Adaptación del proceso

El proceso deberá adaptarse a las características propias de la organización. El tamaño del mismo, así como las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

2. Balancear prioridades

Los requerimientos de los diversos clientes a los cuales se les realizara el proyecto (en este caso solo un cliente), pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos (en el caso pertinente que satisfaga los deseos del cliente).

3. Colaboración entre equipos

El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, etc. Para nuestro caso y contradiciendo la colaboración entre equipos que es la que se aplica para un proyecto mayúsculo, este proyecto si se hace con una única persona, ya que el proyecto a realizar no es tan gigante como lo que podría representar un proyecto de una empresa grande.

4. Demostrar valor iterativamente

Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto, así como también los riesgos involucrados.

5. Elevar el nivel de abstracción

Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o esquemas (frameworks) por nombrar algunos. Éstos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo, con UML.

6. Enfocarse en la calidad

El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción.

El ciclo de vida de RUP

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

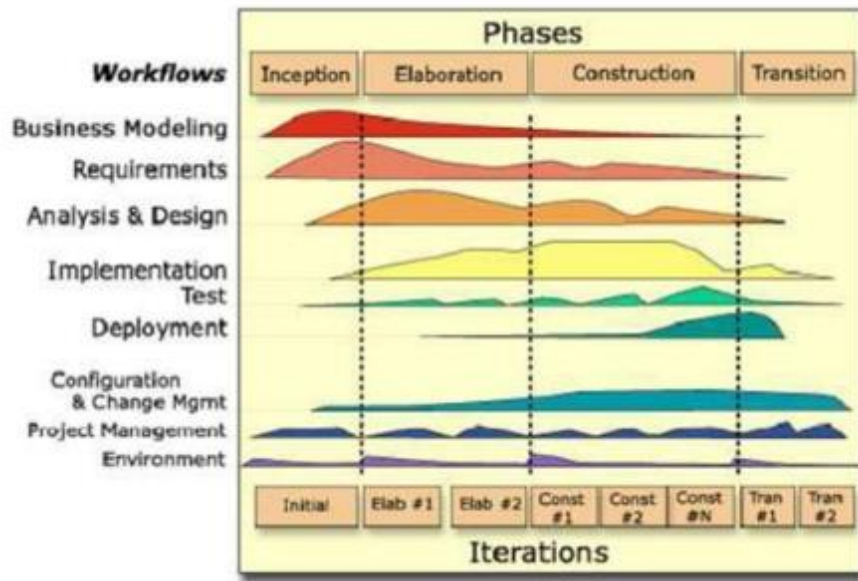


Ilustración I.3 Ciclo de vida de RUP

En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades.

Inicio:

Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. Se define el alcance del proyecto.

Elaboración:

Se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.

Construcción:

Se concentra en la elaboración de un producto totalmente operativo y eficiente junto con el manual de usuario.

Transición:

Se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

DESCRIPCIÓN DE LAS ACTIVIDADES

Dependiendo de las iteraciones del proceso el equipo de desarrollo puede realizar 7 tipos de actividades:

FASE DE INICIO

Durante la fase de inicio las iteraciones hacen poner mayor énfasis en actividades modelado del negocio y de requisitos.

Modelado del negocio

En esta fase el equipo se familiarizará más al funcionamiento de la empresa, sobre conocer sus procesos.

Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado.

Entender el problema actual en la organización objetivo e identificar potenciales mejoras.

Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.

Requisitos

En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos.

Establecer y mantener un acuerdo entre clientes sobre lo que el sistema podría hacer.

Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.

Definir el ámbito del sistema.

Proveer una base para estimar costos y tiempo de desarrollo del sistema.

Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

FASE DE ELABORACIÓN

En la fase de elaboración, las iteraciones se orientan al desarrollo de la base de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (Refinamiento), análisis, diseño y una parte de implementación orientado a la base de la arquitectura.

Análisis y Diseño

En esta actividad se especifican los requerimientos y se describen sobre cómo se van a implementar en el sistema. Transformar los requisitos al diseño del sistema.

Desarrollar una arquitectura para el sistema. Adaptar el diseño para que sea consistente con el entorno de implementación

FASE DE CONSTRUCCIÓN

Implementación

Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. El resultado final es un sistema ejecutable.

Planificar qué subsistemas deben ser implementados y en qué orden deben ser integrados, formando el Plan de Integración. Cada implementador decide en qué orden implementa los elementos del subsistema.

Si encuentra errores de diseño, los notifica.

Se integra el sistema siguiendo el plan.

Pruebas

Este flujo de trabajo es el encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida. Encontrar y documentar defectos en la calidad del software.

Generalmente asesora sobre la calidad del software percibida. Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.

Verificar las funciones del producto de software según lo diseñado.

Verificar que los requisitos tengan su apropiada implementación.

Despliegue

Esta actividad tiene como objetivo producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

Probar el producto en su entorno de ejecución final.

Empaquetar el software para su distribución.

Distribuir el software.

Instalar el software.

Proveer asistencia y ayuda a los usuarios.

Formar a los usuarios y al cuerpo de ventas.

Migrar el software existente o convertir bases de datos.

I.2.5.2 Metodología MOBILE-D

El objetivo de este método es conseguir ciclos de desarrollo muy rápidos en equipos muy pequeños. Fue creado en un proyecto finlandés en 2005, pero sigue estando vigente. Basado en metodologías conocidas pero aplicadas de forma estricta como: extreme programming, Crystal Methodologies y Rational Unified Process.

Se compone de distintas fases: exploración, inicialización, fase de producto, fase de estabilización y la fase de pruebas. Cada una tiene un día de planificación y otro de entrega.

En la fase de **exploración** se centra la atención en la planificación y a los conceptos básicos del proyecto. Aquí es donde hacemos una definición del alcance del proyecto y su establecimiento con las funcionalidades donde queremos llegar.

En la **iniciación** configuramos el proyecto identificando y preparando todos los recursos necesarios como hemos comentado anteriormente en esta fase la dedicaremos un día a la planificación y el resto al trabajo y publicación.

En la **fase de producto** se repiten interactivamente las subfases. Se usa el desarrollo dirigido por pruebas (TDD), antes de iniciar el desarrollo de una funcionalidad debe existir una prueba que verifique su funcionamiento. En esta fase podemos decir que se lleva a acabo toda la implementación.

Después de la fase de producto llega la **fase de estabilización** en la que se realizan las acciones de integración para enganchar los posibles módulos separados en una única aplicación.

Fase de pruebas. Una vez parado totalmente el desarrollo se pasa una fase de testeo hasta llegar a una versión estable según lo establecido en las primeras fases por el cliente. Si es necesario se reparan los errores, pero no se desarrolla nada nuevo.

Una vez acabada todas las fases deberíamos tener una aplicación publicable y entregable al cliente.¹

Ciclo de desarrollo de Mobile-D



Ilustración I.4 Ciclo de desarrollo de Mobile-D

¹ RODRÍGUEZ, T. (s.f.). *Genbetadev*. Obtenido de <http://www.genbetadev.com/desarrollo-aplicaciones-moviles/metodos-aplicables-para-el-desarrollo-de-aplicaciones-moviles>

I.2.5.3 Metodología para la capacitación del personal en el manejo del sistema

La metodología para el desarrollo de este taller de socialización está basada en dos premisas conceptuales importantes:

1. La construcción de conocimientos a partir de una estructura horizontal de transferencia de saberes, donde los participantes de las conferencias adquieren un conocimiento común a sus intereses.
2. La reflexión como punto de partida para el desarrollo de una estrategia de apropiación de conocimientos. En este espacio se generan procesos de reflexión, para que surjan preguntas, inquietudes, dudas, aclaraciones, propuestas de mejora, sugerencias para los ajustes respectivos y otros planteamientos que puedan darse como consecuencia de las tareas realizadas en la etapa de concreción y construcción crítica.

Estas dos premisas enmarcan el desarrollo técnico de las estrategias, pues se pretende acercar de manera más precisa a los actores del proyecto, y así, realizar un diagnóstico que sirva como elemento de consulta para todos los participantes al evento y como punto de partida de reflexiones más detalladas sobre estas temáticas.

La técnica a utilizar para este taller se denomina “Café del Mundo”, debido a que se pudo comprobar que la información discutida o desarrollada en las conferencias, podía ser realmente apropiada en el break o en el momento del café, pues era allí donde se podían reunir las personas más afines y conversar acerca de las diferentes opiniones que tuvieron durante el tiempo de la conferencia. El espacio de la conferencia es cerrado, hablan muy pocas personas, mientras que en el momento del café todos conversan, opinan y discuten de igual a igual, sobre las diferentes apreciaciones que tuvieron de las temáticas abordadas en las capacitaciones.

En el caso de las tarjetas de presentación, se realizará un diseño apropiado, que capte la atención. La forma de distribución, será en la clínica veterinaria, donde cada personal a cargo, hará entrega de una tarjeta a los clientes que visiten las instalaciones.

I.2.6 PLANIFICACIÓN DEL PROYECTO

ID	Nombre de la Tarea	Inicio	Final	Duracion	23/2/2015				1/1/2016
					23/2/2015	1/4/2015	1/7/2015	1/10/2015	1/1/2016
1	☐ Sistema de Informacion	23/2/2015	8/10/2015	164.0 d.					
2	Inicio	23/2/2015	3/4/2015	30.0 d.					
3	Elaboracion	6/3/2015	28/5/2015	60.0 d.					
4	Construccion	27/3/2015	10/9/2015	120.0 d.					
5	Transicion	28/8/2015	8/10/2015	30.0 d.					
6	☐ Aplicacion Movil	22/9/2015	25/12/2015	69.0 d.					
7	Exploracion	22/9/2015	5/10/2015	10.0 d.					
8	Inicializacion	5/10/2015	21/10/2015	13.0 d.					
9	Produccion	21/10/2015	20/11/2015	23.0 d.					
10	Estabilizacion	20/11/2015	4/12/2015	11.0 d.					
11	Pruebas	4/12/2015	25/12/2015	16.0 d.					
12	☐ Socializacion del sistema	4/1/2016	5/2/2016	25.0 d.					
13	Planificacion de los eventos	4/1/2016	8/1/2016	5.0 d.					
14	Elaboracion del Material	8/1/2016	14/1/2016	5.0 d.					
15	Distribucion del material	15/1/2016	5/2/2016	16.0 d.					
16	Desarrollo de los eventos	18/1/2016	5/2/2016	15.0 d.					

I.2.7 RESULTADOS ESPERADOS

I.2.7.1 Sistema informático para la clínica veterinaria desarrollado.

Es un sistema informático en el cual se administrará la información clínica de los pacientes, organización de las citas y publicación de noticias. Se lo realizará basado en software libre, orientado a la web y cumpliendo los criterios de calidad en cuanto a la funcionalidad, usabilidad, entre los más importantes. Solo el personal de la clínica habilitado podrá acceder los datos del sistema mediante vía Web

I.2.7.2 Aplicación móvil ANDROID desarrollada e integrada con el nuevo sistema informático.

Es una aplicación móvil en la que el cliente podrá acceder a la información de sus mascotas, ver las noticias que, publicada la administración de la clínica, solicitar citas y además contara con

notificaciones que recordaran al cliente que debe llevar su mascota a una determinada cita. Se la realizara basada en el software libre, disponible para sistema operativo ANDROID.

I.2.7.3 Estrategias de socialización del sistema informático y de la aplicación móvil implementadas.

Este componente se refiere a la realización de:

Programas de capacitación para el personal de la clínica con referencia al sistema desarrollado.

Creación de un nuevo diseño y distribución de la tarjeta de presentación de la clínica incluyendo la información de la aplicación móvil desarrollada.

Con la implementación de estos componentes, se espera que el manejo de la información de los pacientes de la clínica veterinaria AMEVET se mejore.

I.2.8 TRANSFERENCIA DE RESULTADOS

I.2.8.1 Grupo de beneficiarios de los resultados

Personal de la Clínica veterinaria “AMEVET” y los clientes de la misma.

I.2.9 PRESUPUESTO / JUSTIFICACIÓN

La finalidad de este estudio es determinar qué tan bueno será la implementación de este nuevo sistema en la clínica veterinaria antes mencionada. Además, se verán los costos que tendrá que incurrir la clínica.

Las preguntas que orientan el estudio de factibilidad, entre otras, son: ¿Es realmente necesario? ¿Es beneficioso? ¿Es rentable económico y socialmente? ¿Están las condiciones para emprender el proyecto?

I.2.9.1 TÉCNICA

Se deben considerar aspectos técnicos requeridos para el desarrollo del nuevo sistema, como hardware, software y recursos humanos.

I.2.9.1.1 Hardware

El sistema que se va a realizar debe poseer la capacidad técnica de manejar una cantidad de datos, por lo que se requiere de un computador servidor (arquitectura cliente-servidor) que soporte el almacenamiento de base de datos con todos los registros de la clínica veterinaria.

Para realizar las etapas de análisis de requerimientos, diseño, implementación, pruebas y documentación se cuenta con el siguiente hardware:

Laptop	
Procesador:	Intel Core i7 4ta Gen.
Memoria RAM:	8 Gb
Disco Duro:	1024 Gb
Sistema Operativo:	Windows 10 Home Single Language
Tarjeta de Red:	Realtek PCIe FE Family
Tarjeta de Red Inalámbrica:	Intel® Dual Band Wireless AC-3160

Tabla I.1 Factibilidad de Hardware.

Por lo que la clínica no deberá adquirir ningún sistema computacional nuevo para el desarrollo, ya que existe suficiente hardware para llevar a cabo el proyecto.

I.2.9.1.2 Software

Es necesario utilizar los exploradores de internet antes mencionados que existe en la red que sea compatible sistema operativo que se esté utilizando. Estos se bajan de manera gratuita en sus sitios correspondientes.

I.2.9.1.3 Recurso Humano

Para la realización de este proyecto se cuenta con un estudiante que posee las siguientes competencias académicas:

- Lenguaje de Programación: JAVA, PHP Codeigniter, Ajax, JavaScript, AngularJS, Ionic Framework, HTML5, CSS3.
- Modelamiento de Software: Lenguaje de modelado UML.
- Base de Datos: PostgreSQL, MySQL.

I.2.9.2 ECONÓMICA

Esta factibilidad considera aspectos económicos: gastos e ingresos, para determinar el grado de factibilidad del nuevo sistema al ser incorporado en la clínica.

I.2.9.2.1 Costo de Hardware

Para poder implementar el nuevo sistema en la clínica veterinaria se necesita lo siguiente un Servidor. Y para los clientes la clínica veterinaria posee computadores que son utilizados actualmente por los funcionarios y poseen conexión a internet.

Debido al alto costo de poseer un servidor dedicado e instalado en la misma clínica, se opta por aceptar el sistema de arriendo de hosting boliviano (para evitar inconvenientes).

Nombre	Advance Hosting
Espacio	ilimitado
Transferencia	ilimitado
Base de Datos	1
Precio Anual	Bs 329

Tabla I.2 Hosting.

* No incluyen IVA.

I.2.9.2.2 Costo de Software

Para realizar un estudio sobre el costo del software que se utilizará, es necesario detallar que las herramientas a utilizar en este proyecto, en lo que refiere a programación y elaboración de reportes.

Cabe destacar, que se aprovechará algún software de pago que será asumido por el investigador para realizar los informes como para desarrollo y pruebas. El resto del software que se utilizará es gratuito, como los exploradores, por lo que no se refleja un gasto en software.

I.2.9.2.3 Costo de Personal

En este proyecto trabajará un sólo estudiante, y debemos considerar que el límite para realizar este proyecto es de 10 meses.

Como estudiante de proyecto se estima que se trabajará 5 horas diarias laborales por mes serán 100 horas mensuales.

$$\begin{aligned}
 \text{Costo} &= \text{Salario Mensual} \times \text{Meses Trabajados} \\
 \text{Costo} &= 2000 \times 10 \\
 \text{Costo} &= \mathbf{Bs. 20.000}
 \end{aligned}$$

I.2.9.2.4 Costos Totales

Los costos totales serán calculados por la suma total de los costos de hardware, los costos de software y los costos de personal.

$$\begin{aligned}
 \text{Costos Totales} &= \text{Costo Hardware} + \text{Costo Software} + \text{Costo Personal} \\
 \text{Costos Totales} &= \text{Bs. 329} + 0 + \text{Bs. 20.000} \\
 \text{Costos Totales} &= \mathbf{Bs. 20.329}
 \end{aligned}$$

I.2.9.2.5 Beneficios del Proyecto

Con la implementación y la puesta en marcha de este nuevo sistema, se registrarán los siguientes beneficios:

- Mayor rapidez en las tareas efectuadas.
- El sistema recordará a los clientes sus próximas visitas.
- Eficiencia en consultas a domicilio.
- Más intuitivo y fácil de usar.
-

Costo Inicial Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Bs. 20200	0	Bs. 329	Bs. 329	Bs. 329	Bs. 329

Tabla I.3 Inversión Inicial y Gastos Anuales.

I.2.9.3 OPERACIONAL

Esta factibilidad comprende una determinación de la probabilidad de que un nuevo sistema se use como se supone, y se evalúa el impacto que causara en la clínica.

Los empleados saben utilizar software de oficina como Word, Excel, etc. y conocimientos básicos de computación e internet. Por lo que se requiere de una pequeña capacitación a los usuarios de este nuevo sistema de tal manera que puedan utilizar el sistema en su totalidad.

Algunas características del nuevo sistema cuando esté en funcionamiento:

- ✓ Con el nuevo sistema aumentará la eficiencia en los procesos ya mencionados como la administración de los pacientes, organización de citas y publicación de noticias.
- ✓ El sistema podrá ser visto desde cualquier lugar, lo que otorgará mayor comodidad para los trabajadores.
- ✓ Se mejora la comunicación con los clientes.

Con esto quedan dadas las factibilidades operacionales.

I.2.9.4 LEGAL

El objetivo de la Factibilidad Legal es el poder verificar que, al desarrollar un sistema, éste no incurre, en infracciones, violaciones u otros delitos que podrían implicar en la imposibilidad de poner en práctica o interrumpir el funcionamiento del sistema.

Para el presente proyecto no existen trabas legales que impidan el buen desempeño y funcionamiento del software, puesto que no se incurren en infracciones a las leyes vigentes en la actualidad, de las cuales se especifican:

I.2.9.4.1 LEY N° 1768 en el Capítulo X

La cual Tipifica figuras penales relativas a la informática.

ARTICULO 363 bis. - (MANIPULACIÓN INFORMÁTICA). - El que con la intención de obtener un beneficio indebido para sí o un tercero, manipule un procesamiento o transferencia de datos informáticos que conduzca a un resultado incorrecto o evite un proceso tal cuyo resultado habría sido correcto, ocasionando de esta manera una transferencia patrimonial en perjuicio de tercero, será sancionado con reclusión de uno a cinco años y con multa de sesenta a doscientos días.

ARTICULO 363 ter. - (ALTERACIÓN, ACCESO Y USO INDEBIDO DE DATOS INFORMÁTICOS). - El que sin estar autorizado se apodere, acceda, utilice, modifique, suprima o inutilice, datos almacenados en una computadora o en cualquier soporte informático, ocasionando perjuicio al titular de la información, será sancionado con prestación de trabajo hasta un año o multa hasta doscientos días.

I.2.9.5 CONCLUSIÓN DE FACTIBILIDAD

Dado todas las factibilidades siendo aprobadas, el sistema podrá ser realizado siempre cuando se siga cumpliendo con lo antes mencionado.

CAPITULO II COMPONENTES

I.1 COMPONENTE 1: SISTEMA DE INFORMACIÓN

I.1.1 MARCO TEÓRICO

El sistema denominado “Neg-VET” fue diseñado para satisfacer la demanda y superar los problemas que se presentan en el área mediante la captura de datos generados durante la atención clínica de las mascotas, incluyendo un control eficaz de clientes y brinde consultas sencillas capaces de brindar una mayor calidad de información al médico veterinario a través de una interfaz amigable para brindar mayor información al cliente sobre la salud de su mascota.

También el sistema se encarga del área de citas con soporte para solicitudes online implementando así otra alternativa para solicitar una cita además de las ya existentes como la de vía telefónica o personal.

Otra área en la que va a interactuar el sistema es la socialización de las noticias de la clínica estas podrán ser vistas mediante la aplicación móvil del cliente. El sistema debe estar apto para proporcionar datos a la aplicación móvil del cliente.

I.1.1.1 Antecedentes

La clínica Veterinaria “AMEVET” siempre realizó el control de todas sus actividades de manera manual por considerarlo “más seguro y sencillo” además de la costumbre de llevarlo así hace el proceso habitual, pero a la vez consientes de mejorar sus actividades

Además del incumplimiento de propuestas anteriores que se le hicieron para un sistema integral que nunca se le presentaron.

I.1.1.2 Estudio de la Variable Independiente: Gestión de la Información

I.1.1.2.1 Clínica Veterinaria AMEVET

La clínica veterinaria AMEVET se dedica al cuidado de animales, posee cuidados medicinales (cirugías, vacunas, alimentación, farmacia, etc.), estéticos (baños, peluquería, etc.) y otros servicios.

Fue creada por un grupo de médicos veterinarios que quería ofrecer la mejor medicina posible para perros, gatos y mascotas exóticas.

Actualmente la clínica reside en el barrio SENAC de la ciudad de Tarija, donde además cuenta con dos veterinarios titulados de prestigiosas universidades, los cuales prestan servicios a la clínica y están disponibles para consultas a domicilio.

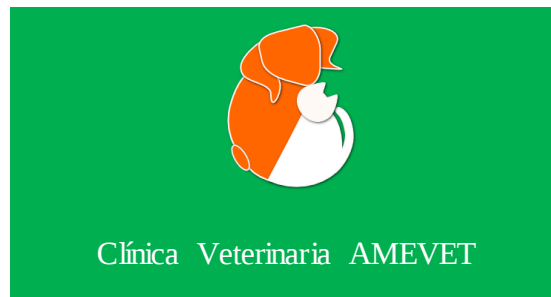


Ilustración 0.1 Logo Clínica Veterinaria AMEVET

I.1.1.2.1.1 Organigrama

A continuación, presentare el organigrama actual de la clínica veterinaria:

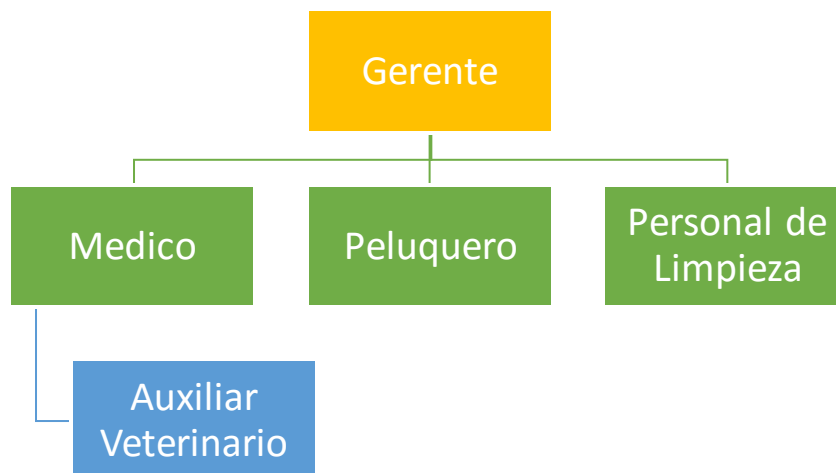


Ilustración 0.2 Organigrama de la clínica Veterinaria

I.1.1.2.2 Misión y Visión

I.1.1.2.2.1 Misión

Ser líderes indiscutibles, en servicios médicos veterinarios y complementarios, para brindar a nuestros clientes calidad y profesionalismo, contando con excelentes instalaciones.

I.1.1.2.2.2 Visión

Ofrecer bienestar tanto animal, como a las familias de nuestros pacientes a través de la prestación de servicios médicos veterinarios y complementarios, entregando calidad y satisfacción, superando las expectativas de nuestros clientes, contribuyendo a la innovación y desarrollo profesional del sector médico veterinario.

I.1.1.3 Estudio de la Variable Dependiente: Sistema de Gestión de la Información

Sistema de información que facilite a la institución a mejorar la eficiencia en sus áreas de operación interna en tareas como gestión de los pacientes, promoción de noticias y control de citas. Para esto debemos tomar en cuenta que metodologías y tecnologías se aplicaran.

I.1.1.4 Relación entre la Variable Independiente y Dependiente

X(i) Desarrollo de sistema Web para una mejor gestión del área administrativa	Y(i) Control del área Administrativa
Analizar la situación actual de la clínica veterinaria “AMEVET”	Factibilidad de los procesos.
Determinar los requerimientos necesarios	Mejorar la administración de la información de los pacientes de la clínica veterinaria.
Elaborar el diseño lógico y físico del sistema Automatizado.	Solucionar necesidades de las áreas clínicas y comunicacionales.

Tabla 0.1 Relación Variable dependiente e independiente

I.1.1.5 Metodología de Desarrollo

I.1.1.5.1 Metodología RUP (Racional Unified Process)

El concepto de esta metodología lo vimos en la sección I.5.2.1 del presente documento.

I.1.1.5.2 UML (Lenguaje Unificado de Modelado)

El Lenguaje de Modelado Unificado (UML: Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90s. UML es llamado un lenguaje de modelado, no un método. Los métodos consisten de ambos de un lenguaje de modelado y de un proceso.

El UML, fusiona los conceptos de la orientación a objetos aportados por Booch, OMT y OOSE (Booch, G. et al., 1999).

UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios.

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño.

La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático.

Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo.

I.1.1.5.2.1 Tipos de Diagramas Utilizados

UML presenta 13 diagramas aquí describiremos solo los 6 utilizados en este proyecto.

I.1.1.5.2.1.1 Diagramas de Clases

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas «estáticos» porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases «conocen» a qué otras clases o qué clases «son parte» de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

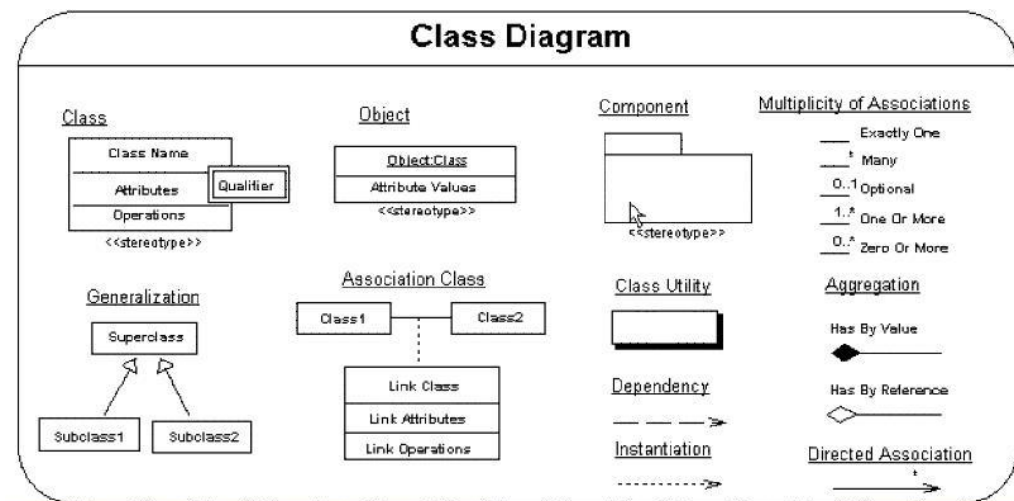


Ilustración 0.3 Simbología Diagrama de Clases

I.1.1.5.2.1.2 Diagramas de Casos de Uso

Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de casos de uso y los actores participantes en el proceso.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan

especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen qué es lo que debe hacer el sistema, pero no cómo.

Simbología de Diagrama de Casos de Usos

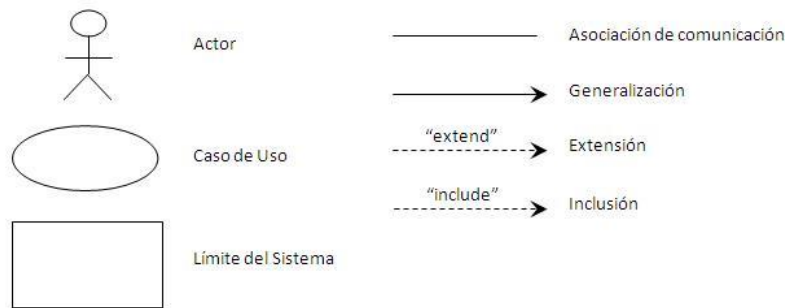


Ilustración 0.4 Simbología de Casos de Uso

I.1.1.5.2.1.3 Diagramas de Actividades

Un diagrama de actividades muestra un proceso de negocio o un proceso de software como un flujo de trabajo a través de una serie de acciones. Las personas, los componentes de software o los equipos pueden realizar estas acciones.

Puede usar un diagrama de actividades para describir procesos de varios tipos, como los ejemplos siguientes:

- Un proceso de negocio o un flujo de trabajo entre los usuarios y el sistema.
- Los pasos que se realizan en un caso de uso. Para más información.
- Un protocolo de software, es decir, las secuencias de interacciones entre componentes permitidas.
- Un algoritmo de software.

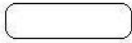
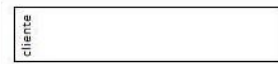
SÍMBOLO	SIGNIFICADO
	Punto de inicio del proceso
	Actividad
	Condicional
	Flujo de secuencia
	Bifurcación o entrada
	Punto final del proceso
	Swinlanes ("Calles")

Ilustración 0.5 Simbología de Diagramas de Casos de Uso

I.1.1.5.2.1.4 Diagramas de Secuencias

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros

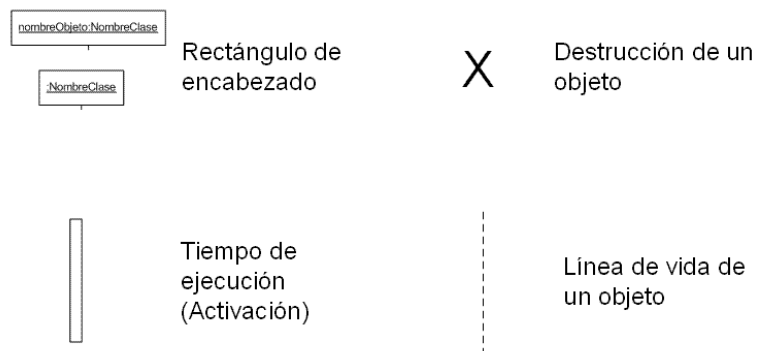


Ilustración 0.6 Simbología del Diagrama de Secuencia

I.1.1.5.2.1.5 Diagrama de Componentes

Los diagramas de componentes muestran los componentes del software (ya sea las tecnologías que lo forman como Kparts, componentes CORBA, Java Beans o simplemente secciones del sistema claramente distintas) y los artilugios de que está compuesto como los archivos de código fuente, las librerías o las tablas de una base de datos.

Los componentes pueden tener interfaces (es decir clases abstractas con operaciones) que permiten asociaciones entre componentes.

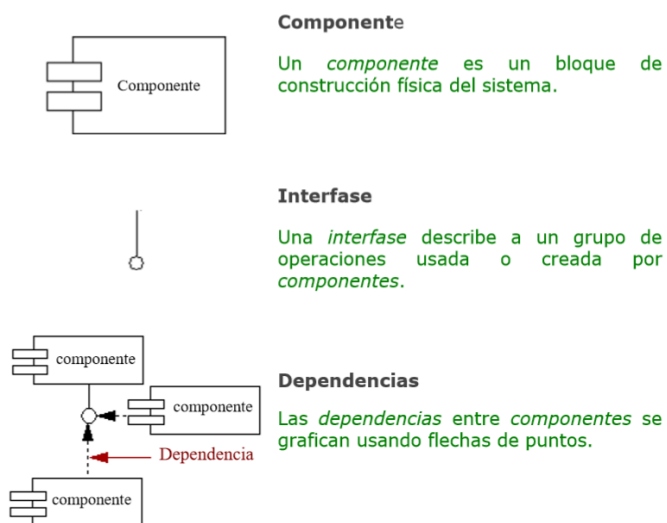


Ilustración 0.7 Simbología de diagrama de componente

I.1.1.5.2.1.6 Diagrama de Despliegue

Los diagramas de Despliegue muestran las instancias existentes al ejecutarse, así como sus relaciones. También se representan los nodos que identifican recursos físicos, típicamente un ordenador, así como interfaces y objetos (instancias de las clases).

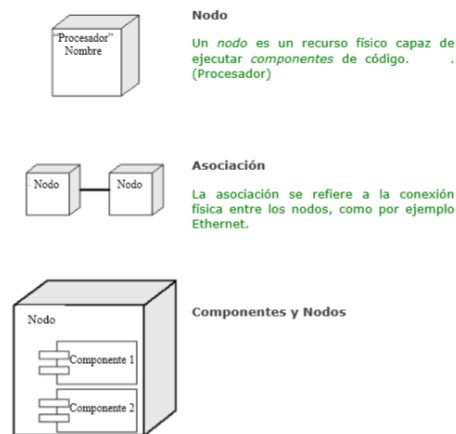


Ilustración 0.8 Simbología Diagrama de Despliegue

I.1.1.6 Herramientas y Tecnologías

En esta sección se detallarán todas las tecnologías utilizadas para el diseño, construcción y visualización de este nuevo sistema.

I.1.1.6.1 Calendarización y Documentación del Proyecto

I.1.1.6.1.1 Edraw Max

Es un software versátil de gráficas, con características que lo hacen perfecto no sólo para diagramas de flujo de aspecto profesional, organigramas, diagramas de red y tablas de negocios, sino también para planos de construcción, mapas mentales, flujo gramas, diseños de moda, diagramas UML, diagramas de ingeniería eléctrica, mapas direccionales, estructuras de programas, diagramas de bases de datos.²



Ilustración 0.9 Logo Edraw Max

² EdrawSoft. (s.f.). *EdrawMax*. Obtenido de <http://www.edrawsoft.com/EDrawMax.php>

I.1.1.6.1.1.2 Microsoft Word 2013

Microsoft Word es el procesador de texto más popular del mercado. Es un potentísimo software desarrollado para el entorno Windows y es la herramienta más intuitiva y profesional para el tratamiento y presentación de la información.

La misión fundamental de Word es ayudarle a escribir, revisar, dar formato e imprimir documentos. A partir de este objetivo, Word ofrece un conjunto de funciones que le permiten crear cualquier documento que pueda imaginar. Puede producir cartas, memorandos, informes, boletines, ensayos, folletos e incluso puede crear correos electrónicos o diseñar páginas Web.

Microsoft lleva varios años actualizando su procesador de texto Word, por lo tanto, existen varias versiones del mismo.³



Ilustración 0.10 Logo Microsoft Word 2013

I.1.1.6.2 Herramientas de Modelado y Diseño del Sistema

I.1.1.6.2.1 Star UML

StarUML es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture), que en un principio era un producto comercial y que hace cerca de un año paso de ser un proyecto comercial (anteriormente llamado plastic) a uno de licencia abierta GNU/GPL.

³ CIBERAULA. (s.f.). Obtenido de http://www.ciberaula.com/curso/word/que_es/

El software heredó todas las características de la versión comercial y poco a poco ha ido mejorando sus características, entre las cuales se encuentran:

- Soporte completo al diseño UML mediante el uso de.
 - Diagrama de casos de uso
 - Diagrama de clase
 - Diagrama de secuencia
 - Diagrama de colaboración.
 - Diagrama de estados
 - Diagrama de actividad.
 - Diagrama de componentes
 - Diagrama de despliegue.
 - Diagrama de composición estructural (UML 2.0)
- Definir elementos propios para los diagramas, que no necesariamente pertenezcan al estándar de UML,
- La capacidad de generar código a partir de los diagramas y viceversa, actualmente funcionando para los lenguajes c++, c# y java.
- Generar documentación en formatos Word, Excel y PowerPoint sobre los diagramas.
- Patrones GoF (Gang of Four), EJB (Enterprise JavaBeans) y personalizados.
- Plantillas de proyectos.
- Posibilidad de crear plugins para el programa.

En definitiva, esta es una de las mejores alternativas gratis que hay en Internet para el modelamiento de software y probablemente una gran ayuda a la hora de programar juegos.⁴



Ilustración 0.11 Logo StarUML

⁴ Black-byte. (s.f.). *black-byte*. Obtenido de <http://black-byte.com/review/staruml/>

I.1.1.6.2.2 Dia

Dia es una aplicación informática de propósito general para la creación de diagramas, desarrollada como parte del proyecto GNOME. Está concebido de forma modular, con diferentes paquetes de formas para diferentes necesidades.

Dia está diseñado como un sustituto de la aplicación comercial Visio de Microsoft. Se puede utilizar para dibujar diferentes tipos de diagramas. Actualmente se incluyen diagramas entidad-relación, diagramas UML, diagramas de flujo, diagramas de redes, diagramas de circuitos eléctricos, etc. Nuevas formas pueden ser fácilmente agregadas, dibujándolas con un subconjunto de SVG e incluyéndolas en un archivo XML.

El formato para leer y almacenar gráficos es XML (comprimido con gzip, para ahorrar espacio). Puede producir salida en los formatos EPS, SVG y PNG.

También conviene recordar que Dia, gracias al paquete dia2code, puede generar el esqueleto del código a escribir, si utilizáramos con tal fin un UML.⁵



Ilustración 0.12 Logo Dia

I.1.1.6.2.3 DbVisualizer

DbVisualizer es una herramienta de base de datos para desarrolladores y administradores de bases de datos, ayudando con el desarrollo y mantenimiento de bases de datos. Cuenta con soporte finalización automática en el editor de SQL, visual Crear / Alter apoyo Tabla, la ejecución de múltiples instrucciones scripts SQL, la cartografía de datos, gestión de datos BLOB / CLOB, exportación de esquema / base de datos, los datos de importación de archivos CSV, Query Builder,

⁵ Wikipedia.(s.f.). Obtenido de [https://es.wikipedia.org/wiki/Dia_\(programa\)](https://es.wikipedia.org/wiki/Dia_(programa))

Foreign Navigator clave de datos, explica el plan de apoyo, e incluye la gestión de puntos de vista, disparadores, cuadros sinópticos, sinónimos, funciones, procedimientos, tablas del sistema, triggers, columnas claves principales, privilegios, restricciones, referencias, DDL, programa, módulos, archivos de datos, y espacios de tabla.⁶



Ilustración 0.13 Logo DbVisualizer

I.1.1.6.3 Plataformas de Desarrollo

I.1.1.6.3.1 Sublime Text 3

Sublime Text es un editor de código multiplataforma, ligero y con pocas concesiones a las florituras. Es una herramienta concebida para programar sin distracciones. Su interfaz de color oscuro y la riqueza de coloreado de la sintaxis, centra nuestra atención completamente.

Sublime Text permite tener varios documentos abiertos mediante pestañas, e incluso emplear varios paneles para aquellos que utilicen más de un monitor. Dispone de modo de pantalla completa, para aprovechar al máximo el espacio visual disponible de la pantalla.

El programa cuenta “de serie” con 22 combinaciones de color posibles, aunque se pueden conseguir más. Para navegar por el código cuenta con Minimap, un panel que permite moverse por el código de forma rápida.

El sistema de resaltado de sintaxis de Sublime Text soporta un gran número de lenguajes (C, C++, C#, CSS, D, Erlang, HTML, Groovy, Haskell, HTML, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, Matlab, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML).

⁶ dbvis. (s.f.). Obtenido de <https://www.dbvis.com>

El programa dispone de auto-guardado, muchas opciones de personalización, cuenta con un buen número de herramientas para la edición del código y automatización de tareas. Soporta macros, Snippets y auto completar, entre otras funcionalidades. Algunas de sus características son ampliables mediante plugins.

Sublime Text es un programa de pago, aunque se puede descargar una versión de prueba, plenamente funcional y sin limitación de tiempo. La licencia individual cuesta 59 dólares. Cada programador es un pequeño maniático con sus credos y sus fobias respecto de las herramientas que emplea, pero si lo que quieres es centrarte únicamente en el código, tal vez deberías probar Sublime Text. La aplicación está disponible para OS X, Linux y Windows.⁷



Ilustración 0.14 Logo Sublime Text

I.1.1.6.3.2 Windows 10

Windows 10 es la nueva versión del sistema operativo de Microsoft, denominado por muchos, como el mejor Sistema Operativo de Windows.⁸

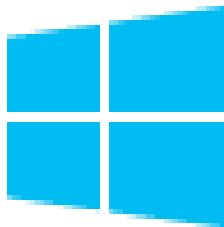


Ilustración 0.15 Logo de Windows 10

⁷ (s.f.). Obtenido de <http://www.genbeta.com/herramientas/sublime-text-un-sofisticado-editor-de-codigo-multiplataforma>

⁸ <https://www.microsoft.com/es-es/software-download/windows10>

I.1.1.6.3.3 Sistema Web

I.1.1.6.3.3.1 Apache

Apache es un servidor web HTTP de código abierto para plataformas Unix-like (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

En sus inicios se basaba en el código de NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Actualmente es el servidor web más usado en todo el mundo, superando en 2009 los 100 millones de sitios web, el 70% del total. Está desarrollado y mantenido por una comunidad de usuarios en torno a la Apache Software Foundation.⁹



Ilustración 0.16 Logo Apache

I.1.1.6.3.3.2 Xamp

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. Desde la versión "5.6.15", XAMPP cambió la base de datos de MySQL A MariaDB. El cual es un fork de MySQL con licencia GPL.

⁹ Hostname. (s.f.). Obtenido de <http://clientes.hostname.cl/knowledgebase.php?action=displayarticle&id=35>

El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

Características y requisitos

XAMPP solamente requiere descargar y ejecutar un archivo ZIP, tar, exe o fkl, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado. Puede encontrarse tanto en versión completa, así como en una versión más ligera que es portátil.

Oficialmente, los diseñadores de XAMPP, fueron los Baiker y Anthony Cooperation los cuales solo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores, testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica, sin embargo, XAMPP es utilizado actualmente como servidor de sitios Web, ya que, con algunas modificaciones, es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes.¹⁰

I.1.1.6.4 Motor de Base de Datos

I.1.1.6.4.1 MySQL

Es un sistema de gestión de bases de datos relacional, multi-hilo y multi-usuario. Es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

¹⁰ Wikipedia. (s.f.). Obtenido de <https://es.wikipedia.org/wiki/XAMPP>,
<https://www.apachefriends.org/es/download.html>

Los objetivos que debe cumplir son:

- Abstracción de la información. Ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- Independencia. Consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- Consistencia. En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. Por otra parte, la base de datos representa una realidad determinada que tiene determinadas condiciones.
- Seguridad. Información almacenada puede llegar a tener un gran valor. Se debe garantizar que esta información se encuentra segura de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- Manejo de transacciones. Una transacción es un programa que se ejecuta como una sola operación.
- Tiempo de respuesta. Lógicamente, es deseable minimizar el tiempo que demora en proporcionar la información solicitada y en almacenar los cambios realizados.¹¹



I.1.1.6.5 Lenguaje de Programación.

I.1.1.6.5.1 PHP

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

¹¹ Corporation, O. (s.f.). MySQL. Obtenido de <http://www.mysql.com>



Ilustración 0.18 Logo PHP

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

¿Cómo trabaja PHP? El lenguaje PHP se procesa en servidores, que son potentes ordenadores con un software y hardware especial. Cuando se escribe una dirección tipo `http://www.aprenderaprogramar.com/index.php` en un navegador web como Internet Explorer, Firefox o Chrome, ¿qué ocurre? Se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso decimos que es un proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática.

El esquema es: Petición de página web al servidor --> El servidor recibe la petición, reúne la información necesaria consultando a bases de datos o a otras páginas webs, otros servidores, etc --> El servidor responde enviando una página web “normal” (estática) pero cuya creación ha sido dinámica (realizando procesos de modo que la página web devuelta no siempre es igual).

En resumen:

Páginas estáticas: Petición --> Respuesta

Páginas dinámicas: Petición --> Procesado y preparación --> Respuesta

En un sitio dinámico, la información generalmente está contenida en una base de datos. Cada vez que mostramos la página, como por ejemplo una página de noticias, buscamos en la base de datos

las últimas noticias que tenemos ingresadas para mostrar en el navegador del visitante. Ahora bien, ¿cómo se consigue que la página vaya mostrando noticias nuevas? Simplemente cargando las mismas en la base de datos, por ejemplo, a través de un formulario que rellena una persona y que una vez relleno cuando pulsa “Enviar” implica que lo que haya escrito se guarde en nuestra base de datos. De esta manera, cuando solicitamos la página web y el servidor consulte la base de datos, encontrará esta nueva información y mostrará nuestra página de forma distinta a como se veía anteriormente.

Por regla general este tipo de lenguaje suele ser utilizado para crear contenido dinámico y poder interactuar con el usuario.

Veamos un ejemplo del contenido de una página web php, que habremos de almacenar en un archivo que termina con la extensión .php y que contiene texto:

```
<!DOCTYPE html>

<html>

<head>

<title>Ejemplo</title>

</head>

<body>

<?php

echo "Bienvenidos a aprenderaprogramar.com";

?>

</body>

</html>
```

Como podrás comprobar el contenido del archivo incluye código HTML y en un momento dado aparece una etiqueta `<?php` que significa que ahí empieza el código PHP. Por eso decimos que el código PHP está embebido (incluido o incrustado) en código HTML.

Por tanto, las páginas PHP contienen HTML con código PHP incluido en ellas para hacer "algo" (en este caso, mostrar Bienvenidos a aprenderaprogramar.com. El código PHP está comprendido entre etiquetas de comienzo y final especiales: `<? php y?>`

Estas etiquetas especiales nos permitirán entrar y salir del "modo PHP". Le sirven al servidor web para saber que tiene que interpretar el código incluido entre esas etiquetas como PHP.

Lo que distingue a PHP de JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente como si fuera una página web estática. El cliente recibirá los resultados que el servidor devuelve después de interpretar el código PHP, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. Es decir, a través de nuestro navegador podríamos ver el código HTML, pero nunca el código PHP que dio lugar al resultado HTML. El servidor web puede ser incluso configurado para que los usuarios no puedan saber si estás o no utilizando PHP.

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales y más avanzados.

Con PHP puedes procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies, entre muchas más cosas. PHP lo utilizan desde pequeñas páginas web hasta grandes empresas. Muchas aplicaciones web están construidas usando PHP. Podemos citar Joomla y Drupal (gestores de contenido de páginas web), osCommerce y Prestashop (tiendas on-line para comercio electrónico), phpBB y SMF (sistemas de foros para páginas web), Moodle (plataforma educativa para educación on-line), etc.

En resumen, PHP es un lenguaje potente, muy usado y de gran interés para los desarrollos web.¹²

¹² Obtenido de <http://aprenderaprogramar.com/php>

I.1.1.6.5.2 SQL

El SQL es el lenguaje estándar ANSI/ISO de definición, manipulación y control de bases de datos relacionales. Es un lenguaje declarativo: sólo hay que indicar qué se quiere hacer. En cambio, en los lenguajes procedimentales es necesario especificar cómo hay que hacer cualquier acción sobre la base de datos. El SQL es un lenguaje muy parecido al lenguaje natural; concretamente, se parece al inglés, y es muy expresivo. Por estas razones, y como lenguaje estándar, el SQL es un lenguaje con el que se puede acceder a todos los sistemas relacionales comerciales.¹³

I.1.1.6.5.3 JavaScript

JavaScript® (a veces abreviado como JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js o Apache CouchDB. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa.

El estándar de JavaScript es ECMAScript. Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en Julio. Se puede seguir el progreso actual de las diferentes, nuevas y mejoradas características en la correspondiente wiki.

Esta sección de la web está dedicada al lenguaje JavaScript, a las partes que no son específicas de las páginas web u otros entornos de servidor. Para información sobre las API específicas de las páginas web, consulte DOM. Para saber más sobre cómo encajan DOM y JavaScript, dispone de la referencia de DOM.

JavaScript no debe ser confundido con el lenguaje de programación Java. Java es una marca registrada de Oracle en Estados Unidos y otros países.¹⁴

¹³ Escofet, C. M. (s.f.). El lenguaje SQL.

¹⁴ Mozilla Foundation. (s.f.). MDN. Obtenido de <https://developer.mozilla.org/es/docs/Web/JavaScript>



Ilustración 0.19 Logo JavaScript

I.1.1.6.5.4 AJAX

Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.¹⁵



Ilustración 0.20 Logo AJAX

I.1.1.6.5.5 JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java,

¹⁵ Wikipedia. (s.f.). *Wikipedia*. Obtenido de <http://es.wikipedia.org/wiki/AJAX>.

JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.¹⁶



Ilustración 0.21 Logo JSON

I.1.1.6.5.6 HTML5

HTML5 es la última versión de HTML. El término representa dos conceptos diferentes:

Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.

Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5 y amigos, a menudo reducido a HTML5.

Diseñado para ser utilizable por todos los desarrolladores de Open Web, esta página referencia a numerosos recursos sobre las tecnologías de HTML5, que se clasifican en varios grupos según su función.

¹⁶ JSON. (s.f.). *JSON*. Obtenido de <http://www.json.org/json-es.html>

Semántica: Permite describir con mayor precisión cuál es su contenido.

Conectividad: Permite comunicarse con el servidor de formas nuevas e innovadoras.

Sin conexión y almacenamiento: Permite a las páginas web almacenar datos localmente en el lado del cliente y operar sin conexión de manera más eficiente.

Multimedia: Nos otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.

Gráficos y efectos 2D/3D: Proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son canvas 2D, WebGL, SVG, etc.

Rendimiento e Integración: Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.

Acceso al dispositivo: Proporciona APIs para el uso de varios componentes internos de entrada y salida de nuestro dispositivo.

CSS3: Nos ofrece una nueva gran variedad de opciones para hacer diseños más sofisticados.¹⁷



Ilustración 0.22 Logo HTML5

I.1.1.6.5.7 CSS3

CSS3 es la última evolución del lenguaje de las Hojas de Estilo en Cascada (Cascading Style Sheets), y pretende ampliar la versión CSS2.1. Trae consigo muchas novedades altamente esperadas, como las esquinas redondeadas, sombras, gradientes, transiciones o animaciones, y

¹⁷ Mozilla Foundation. (s.f.). MDN. Obtenido de <https://developer.mozilla.org/es/docs/HTML/HTML5>

nuevos layouts como multi-columnas, cajas flexibles o maquetas de diseño en cuadrícula (grid layouts).

Las partes experimentales son particulares para cada navegador y deberían ser evitadas en entornos de producción, o usadas con extrema precaución, ya que tanto la sintaxis como la semántica pueden cambiar en el futuro.¹⁸



Ilustración 0.23 Logo CSS3

I.1.1.6.6 Frameworks Utilizados

I.1.1.6.6.1 Codeigniter Framework

Probablemente ya sepamos que un framework es un programa para desarrollar otros programas, CodeIgniter, por tanto, es un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre, libre de uso para cualquier aplicación.

Como cualquier otro framework, Codeigniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación. Esto es, marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. CodeIgniter implementa el proceso de desarrollo llamado Model View Controller (MVC), que es un estándar de programación de aplicaciones, utilizado tanto para

¹⁸ Mozilla Foundation. (s.f.). MDN. Obtenido de <https://developer.mozilla.org/es/docs/Web/CSS/CSS3>

hacer sitios web como programas tradicionales. Este sistema tiene sus características, que veremos en artículos siguientes.

CodeIgniter no es magia, pero contiene muchas ayudas para la creación de aplicaciones PHP avanzadas, que hacen que el proceso de desarrollo más rápido. A la vez, define una arquitectura de desarrollo que hará que programemos de una manera más ordenada y contiene diversas herramientas que ayudan a hacer aplicaciones más versátiles y seguras.

CodeIgniter y otros frameworks PHP pueden ayudarte a dar el salto definitivo como desarrollador PHP, creando aplicaciones web más profesionales y con código más reutilizable, con la diferencia que CodeIgniter está creado para que sea fácil de instalar en cualquier servidor y de empezar a usar que cualquier otro framework. Además muchas de sus utilidades y modos de funcionamiento son opcionales, lo que hace que goces de mayor libertad a la hora de desarrollar sitios web.¹⁹



Ilustración 0.24 Logo Codeigniter

I.1.1.6.6.2 AngularJS 1.4

AngularJS es JavaScript. Es un proyecto de código abierto, realizado en JavaScript que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un framework para el desarrollo, en este caso sobre el lenguaje JavaScript con programación del lado del cliente.

¹⁹ desarrolloweb. (s.f.). *desarrolloweb*. Obtenido de <http://www.desarrolloweb.com/articulos/codeigniter.html>

Al ser un proyecto de código abierto cualquier persona con un poco de curiosidad echar un vistazo con profundidad y ver cómo se ha escrito, incluso admiten colaboraciones de desarrolladores que quiera aportar cosas.²⁰



Ilustración 0.25 Logo AngularJS

I.1.1.6.6.3 Lumx

Lumx es un framework CSS con el cual se construirá el sistema web se integra a la perfección con AngularJS y tiene un sistema responsivo para que se vea igual de genial en cualquier resolución de pantalla.²¹



Ilustración 0.26 Logo Lumx

I.1.1.6.7 Servicios APIS

I.1.1.6.7.1 Punub

Es un servicio que centra exclusivamente en la entrega de datos en tiempo real.

Tiene versiones de pago como una gratuita con ciertas limitaciones. Este servicio se lo utilizará para mostrar notificaciones en tiempo real en el sistema web.²²

²⁰ Angular JS. (s.f.). *Angular JS*. Obtenido de <https://angularjs.org/>

²¹ <http://ui.lumapps.com/>

²² <http://www.pubnub.com>

I.1.1.6.7.2 Ionic.io

Ionic.io es un servicio que facilita a los desarrolladores la integración de GCM en aplicaciones híbridas, por tal motivo es que lo utilizaremos en la aplicación móvil.²³

I.1.1.6.7.3 GCM

El envío de mensajes a través de la nube de Google (Google Cloud Messaging, GCM) es un servicio gratuito que permite a los programadores enviar mensajes en varias plataformas, como Android, iOS y Chrome. Por ejemplo, un servidor puede enviar mensajes directamente a dispositivos individuales, grupos de dispositivos o dispositivos suscritos a temas. Además, la aplicación en un dispositivo puede enviar mensajes directamente a un servidor y a los dispositivos que pertenecen al mismo grupo.²⁴

I.1.1.6.8 Herramientas de Presentación y Visualización

I.1.1.6.8.1 Google Chrome 18

Es un navegador web desarrollado por Google y compilado con base en componentes de código abierto como el motor de renderizado Web Kit y su estructura de desarrollo de aplicaciones. Google Chrome es actualmente el navegador más utilizado en Internet.²⁵



Ilustración 0.27 Logo Google Chrome

²³ <http://ionic.io>

²⁴ Obtenido de <https://support.google.com/googleplay/android-developer/answer/2663268?hl=es-419>

²⁵ <https://www.google.es/chrome/browser/desktop/>

I.1.1.6.8.2 Mozilla Firefox 12

Es un navegador web libre y de código abierto, en cuyo desarrollo puede colaborar cualquier usuario que lo desee. Es descendiente de Mozilla Application Suite y es desarrollado por la Fundación Mozilla.²⁶ Mozilla Firefox es el segundo navegador más utilizado de Internet.²⁶



Ilustración 0.28 Logo Mozilla Firefox

I.1.1.6.8.3 Internet Explorer 9

Conocido comúnmente como IE, es un navegador web desarrollado por Microsoft para el sistema operativo Microsoft Windows desde 1995, la versión compatible es la 9.0.²⁷



Ilustración 0.29 Logo Internet Explorer

I.1.1.6.8.4 Microsoft Edge

Microsoft Edge es un navegador web desarrollado por Microsoft, que se encuentra incluido en Windows 10, reemplazando a Internet Explorer como navegador web por defecto.

Microsoft Edge está diseñado para ser un navegador web ligero con un motor de renderizado de código abierto¹ construido en torno a los estándares web, y Microsoft se ha comprometido a actualizar Edge para integrarlo con los nuevos y existentes estándares que aún no soporta. Edge no cuenta con soporte de las actuales tecnologías existentes, como ActiveX, en favor de las extensiones y la integración con otros servicios de Microsoft, como el asistente digital Cortana y el servicio de almacenamiento en línea OneDrive. También incluye herramientas

²⁶ <https://www.mozilla.org/es-ES/firefox/new/>

²⁷ <http://windows.microsoft.com/es-xl/windows7/getting-started-with-internet-explorer-9>

de anotación y modo de lectura. Fue desarrollado inicialmente bajo el nombre en código (codename) “Project Spartan”.



Ilustración 0.30 Logo Microsoft Edge

I.1.1.6.9 Arquitectura y Diseño: Modelo Vista Controlador (MVC)

I.1.1.6.9.1 Concepto: Patrón MVC

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

I.1.1.6.9.1.1 Comunicación

El modelo, la vista y el controlador deben comunicarse de una manera estable los unos con los otros, de manera que sea coherente con las iteraciones que el usuario realizara. Como es lógico la comunicación entre la vista y el controlador es bastante básica pues están diseñados para operar juntos, pero los modelos se comunican de una manera diferente, un poco más sutil

I.1.1.6.9.1.2 Modelo pasivo

No es necesario para el modelo hacer ninguna tener alguna disposición a él, simplemente basta con tener en cuenta su existencia. El modelo no tiene ninguna responsabilidad para comunicar los cambios a la vista porque ocurren solo por orden del usuario, por lo que esta función la llevara a cabo el controlador porque será el que interprete las ordenes de este usuario debido a que solo debe comunicar que algo ha cambiado. Por esto, el modelo es se encuentra en modo inconsciente y su participación en este caso es irrisoria.

Unión del modelo con la vista y el controlador

Como no todos los modelos pueden ser pasivos, necesitamos algo que comunique al controlador y a la vista, por lo que, en este caso, sí que necesitamos el modelo, ya que solo este puede llevar a cabo los cambios necesarios al estado actual en el que estos se encuentran.

Al contrario que el modelo, que puede ser asociado a múltiples asociaciones con otras vistas y controladores, cada vista solo puede ser asociada a un único controlador, por lo que han de tener una variable de tipo controler que notificara a la vista cuál es su controlador o modelo asignado. De igual manera, el controlador tiene una variable llamada View que apunta a la vista. De esta manera, pueden enviarse mensajes directos el uno al otro y al mismo tiempo, a su modelo.

Al final, la vista es quien lleva la responsabilidad de establecer la comunicación entre los elementos de nuestro patrón MVC. Cuando la vista recibe un mensaje que concierne al modelo o al controlador, lo deja registrado como el modelo con el cual se comunicara y apunta con la variable controller al controlador asignado, enviándole al mismo su identificación para que el controlador establezca en su variable view el identificador de la vista y así puedan operar conjuntamente. El responsable de deshacer estas conexiones, seguirá siendo la vista, quitándose a sí misma como dependiente del modelo y liberando al controlado

I.1.1.6.9.2 Ventajas de Usar MVC

1. Podrás dividir la lógica de negocio del diseño, haciendo tu proyecto más escalable.
2. Te facilitará el uso de URL amigables, importantes para el SEO (Posicionamiento web), la mayoría de frameworks MVC lo controlan.
3. Muchos frameworks MVC ya incluyen librerías de JavaScript como JQuery, lo que te facilitará validar formularios (Ej. JQuery.Validate) en el cliente y en el servidor.
4. Puedes utilizar abstracción de datos, como lo hace Ruby on Rails o con frameworks como Hibernate para Java o NHibernate para ASP .NET MVC, facilitando la realización de consultas a la base de datos.
5. La mayoría de frameworks controlan el uso de la memoria Caché, hoy en día muy importante para el posicionamiento web, ya que buscadores como google dan prioridad a las webs que tengan menor tiempo de descarga.
6. En el caso de proyectos donde hay varios desarrolladores, el seguir métodos comunes de programación, hace que el código sea más entendible entre estos, pudiendo uno continuar el trabajo de otro. En estos casos es conveniente utilizar herramientas de control de versiones como Subversion.

7. Los frameworks están creados para facilitar el trabajo de los desarrolladores, encontrarás clases para controlar fechas, URL's, Webservices, etc. lo que tiene una gran ventaja en cuanto a productividad. Inicialmente como es lógico habrá una curva de aprendizaje, pero luego tendrás muchos beneficios.
8. Poco a poco el desarrollo web se orienta a lo que se denomina "Agile Web Development" (Desarrollo ágil de aplicaciones web), con frameworks como Ruby on Rails que ayudan a crear proyectos de calidad y en corto tiempo. Existen varios frameworks en PHP e incluso ASP .NET que en su nueva versión ya contempla el MVC con Visual C#.
9. Utilizar herramientas con tecnología escalable hace más atractivo tu proyecto en caso de buscar inversión externa, muchas veces para hacer crecer un proyecto, es necesario buscar socios o Bussines Angels que te ayuden a impulsarlo.
10. Un Framework MVC te ayuda a controlar los recursos del servidor, evitando Bugs que puedan repercutir en el rendimiento, por ejemplo, muchas veces olvidamos cerrar conexiones a la base de datos, sobrecargando el servidor.

I.1.1.6.9.3 Componentes Patrón MVC

De esta forma, dividimos el sistema en tres capas donde, como explicaremos más adelante, tenemos la encapsulación de los datos, la interfaz o vista por otro y por último la lógica interna o controlador.

El patrón de arquitectura "modelo vista controlador", es una filosofía de diseño de aplicaciones, compuesta por:

I.1.1.6.9.3.1 Modelo

Contiene el núcleo de la funcionalidad (dominio) de la aplicación.

Encapsula el estado de la aplicación.

No sabe nada / independiente del Controlador y la Vista.

I.1.1.6.9.3.2 Vista

Es la presentación del Modelo.

Puede acceder al Modelo, pero nunca cambiar su estado.

Puede ser notificada cuando hay un cambio de estado en el Modelo.

I.1.1.6.9.3.3 Controlador

Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente

Para entender cómo funciona nuestro patrón Modelo vista controlador, se debe entender la división a través del conjunto de estos tres elementos y como estos componentes se comunican unos con los otros y con otras vistas y controladores externos a el modelo principal. Para ello, es importante saber que el controlador interpreta las entradas del usuario (tanto teclado como el ratón), enviado el mensaje de acción al modelo y a la vista para que se proceda con los cambios que se consideren adecuados

I.1.1.7 Base de Datos

I.1.1.7.1 Definición de Base de Datos

Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada o estructurada.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

I.1.1.7.2 Herramienta phpMyAdmin para el Manejo de la Base de Datos

phpMyAdmin es un programa de libre distribución en PHP, creado por una comunidad sin ánimo de lucro, que sólo trabaja en el proyecto por amor al arte. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva.

La aplicación en si no es más que un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en

nuestro servidor de bases de datos y todas sus tablas. La herramienta nos permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editarlos y borrarlos, borrar tablas y un largo etcétera, incluso ejecutar sentencias SQL y hacer un backup de la base de datos.²⁸



Ilustración 0.31 Logo phpMyAdmin

I.1.1.8 Sistema de Información Automatizado

I.1.1.8.1 El Internet

Internet se podría definir como una red global de redes de ordenadores cuya finalidad es permitir el intercambio libre de información entre todos sus usuarios. Pero sería un error considerar Internet únicamente como una red de computadoras. Podemos considerar las computadoras simplemente como el medio que transporta la información. En este caso Internet sería una gran fuente de información práctica y divertida. Con Internet podemos enviar mensajes, programas ejecutables, ficheros de texto, consultar catálogos de bibliotecas, pedir libros, hacer compras, ... Hay que tener en cuenta que todos los recursos que se pueden encontrar en Internet existen porque alguna persona de forma voluntaria ha dedicado su tiempo en generarlos.

Internet se basa básicamente en cuatro servicios:

- el servicio de correo electrónico (e-mail) que transmite y recibe mensajes: nos podemos poner en contacto con cualquier otro usuario mediante el intercambio de mensajes. Utilizando listas de correo una misma información es enviada a varios destinos al mismo tiempo.
- servicio de noticias (news): te suscribes a un grupo de noticias y recibirás información sobre ese tema. Las noticias son almacenadas en un servidor de noticias al que acceden los usuarios interesados.

²⁸ Desarrolloweb. (s.f.). Obtenido de <http://www.desarrolloweb.com/articulos/844.php>

- acceso remoto (telnet): potencialmente puedes conectarte como terminal y establecer una sesión de trabajo en cualquier ordenador (ordenador remoto) de la red si dispones de los permisos de acceso necesarios para acceder a él.
- transferencia de ficheros (ftp , File Transfer Protocol o Protocolo de Transferencia de Archivos) que permite transferir archivos de una computadora a otra.

Además, como usuario puedes generar noticias e información. Internet dispone de herramientas que facilitan el trabajo: Whois, Archie, Wais, Gopher, WWW ...

Los clientes de la red pueden ser: usuarios y ordenadores. Los usuarios disponen de un identificador de usuario. Cada ordenador puede darle acceso a la red a uno o varios usuarios, por lo que el identificador completo de un usuario dentro de Internet se compone del identificador de usuario, seguido de y del identificador del ordenador.

Dentro de Internet es muy importante el tema de la seguridad que debe ser controlado por cada uno de los usuarios. Hay que tener en cuenta que un ordenador puede ser utilizado para a través de él introducirse en otros ordenadores. El mantener la seguridad de nuestra máquina disponemos de las claves de acceso. Elige una clave que pueda recordar pero que no sea obvia. Para que las claves de acceso sean una buena protección, sólo hay que tener en cuenta las siguientes reglas:

longitud mínima de 6 caracteres.

- que no sea una palabra con significado.
- que no corresponda con iniciales del usuario, empresa, apellidos, datos personales, numero de teléfono, DNI, nombre de familiares...
- que no sea una letra repetida. que no esté formada por letras contiguas del alfabeto.
- que no esté formada por letras contiguas del teclado.
- que mezcle mayúsculas, minúsculas y números.
- que se cambie periódicamente.

I.1.1.8.2 La World Wide Web o la Web

La World Wide Web, abreviado “www” es una red mundial de datos a la que se puede acceder a través de Internet. Esos documentos están unidos entre sí con enlaces. A la World Wide Web también se le conoce como Internet. Sin embargo, este término no es correcto ya que se trata de dos sistemas totalmente diferentes. Internet sólo tiene acceso a determinados datos de la World

Wide Web. Así pues, hay por ejemplo funciones de Internet que no tienen nada que ver con la WWW, por ejemplo, el email.

I.1.1.9 Sistema de información gerencial vía Web

Es un sistema de información basado en computadora, que presenta una colección de personas, procedimientos, bases. Los sistemas de información para la administración tienen como objetivo principal mostrar una visión general de la situación de la empresa. Consecuentemente, estos muestran la situación de las operaciones regulares de la empresa para que los directivos puedan controlar, organizar, planear y dirigir.²⁹

Los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las ventajas de un sistema web:

- **Ahorra tiempo:** Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- **No hay problemas de compatibilidad:** Basta tener un navegador actualizado para poder utilizarlas.
- **No ocupan espacio** en nuestro disco duro.
- **Actualizaciones inmediatas:** Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- **Consumo de recursos bajo:** Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
- **Multiplataforma:** Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- **Portables:** Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
- **La disponibilidad suele ser alta** porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.

²⁹ Wikipedia.(s.f.). Obtenido de
https://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n_para_la_administraci%C3%B3n

- **Se reduce el daño por virus** debido a que los datos porque éstos están guardados en el servidor de la aplicación.
- **Colaboración:** Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.

I.1.2 PLAN DE DESARROLLO DE SOFTWARE

I.1.2.1 Introducción

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto de prácticas de la asignatura de Taller III de la Carrera de Ing. Informática de la Facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho. Este documento provee una visión global del enfoque de desarrollo propuesto.

El proyecto ha sido presentado por el estudiante Rolando Abelino Sanchez Garay; basado en una metodología de Rational Unified Process en la que únicamente se procederá a cumplir con las tres primeras fases que marca la metodología, constando únicamente en la tercera fase de dos iteraciones. Es importante destacar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio y Elaboración y adicionalmente se esbozarán las fases posteriores de Construcción y Transición para dar una visión global de todo proceso.

Como se ha observado que la administración la información de los pacientes de la clínica veterinaria todavía se la realiza de forma manual provocando en ocasiones pérdida de datos como historiales clínicos por tal motivo se desarrollara un sistema web que corrija estas falencias así mismo el sistema debe proveer datos a la aplicación móvil que se detallara en el componente posterior.

I.1.2.1.1 Propósito

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

- El jefe del proyecto lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.
- Los miembros del equipo de desarrollo lo usan para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.

I.1.2.1.2 Alcance

El Plan de Desarrollo del Software describe los procesos que se automatizarán en la clínica veterinaria. Durante el proceso de desarrollo en el artefacto “Visión” se definen las características del producto a desarrollar, lo cual constituye la base para la planificación de las iteraciones. Para la versión 2.0 del Plan de Desarrollo del Software, nos hemos basado en la captura de requisitos mediante el uso de cuestionarios y entrevistas para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto “Visión”, el cual se utilizará para refinar este documento. Posteriormente, el avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

I.1.2.1.3 Resumen

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

Vista General del Proyecto — proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

Organización del Proyecto — describe la estructura organizacional del equipo de desarrollo.

Gestión del Proceso — explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.

Planes y Guías de aplicación — proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

I.1.2.2 Vista General del Proyecto

I.1.2.2.1 Propósito, Alcance y Objetivos

La información que a continuación se incluye ha sido extraída de las diferentes reuniones que se han celebrado con la gerente propietaria de la clínica veterinaria desde el inicio del proyecto, Dra. Ximena Jurado.

I.1.2.2.1.1 Propósito

La clínica veterinaria “AMEVET” se encarga de ofrecer servicios médicos a mascotas en la ciudad de Tarija Capital, así como también dispone de PetShop y alimentos para canes y felinos. La entrada en un mercado competitivo como en el que encuentra inmersa la clínica conllevará una previsible adaptación a los nuevos sistemas de información y a la evolución tecnológica. Por ello, la clínica veterinaria “AMEVET” considera necesario el desarrollo de un nuevo sistema para la administración de la información de los pacientes de la clínica veterinaria.

I.1.2.2.1.2 Alcance

El proyecto debe proporcionar una propuesta para el desarrollo de todos los subsistemas implicados en la administración de la información de los pacientes. Estos procesos se pueden diferenciar en siete grandes bloques:

- a) Gestión de Mascotas, incluyendo:
 - a. Ficha de Control Sanitario
 - b. Ficha de Atención clínica
- b) Gestión de Clientes:
- c) Gestión de Peluquerías:
- d) Gestión de Citas.
- e) Gestión de Medicamentos.
- f) Elaboración de Reportes, incluyendo.
 - a. Ingresos por Peluquerías y Atenciones Clínicas

- b. Clientes de la clínica
- c. Pacientes de la clínica

I.1.2.2.1.3 Objetivos

I.1.2.2.1.3.1 Objetivo General

Desarrollar un sistema de información para facilitar el manejo de la información de los pacientes y extender la socialización de la clínica veterinaria, mediante el uso de tecnología WEB.

I.1.2.2.1.3.2 Objetivos Específicos

- Implementar la metodología RUP para realizar el sistema.
- Incorporar notificaciones de citas en tiempo real.
- Configurar el sistema para que se comuniquen con la aplicación móvil.

I.1.2.2.2 Suposiciones y Restricciones

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de las entrevistas con la gerente propietaria de la clínica son:

Debe contemplarse las implicaciones de los siguientes puntos críticos:

- a) El sistema de momento no implementará el módulo Ventas ni stock de inventarios de productos.
- b) El módulo medicamento no es un inventario este módulo solo se lo realiza para facilitar el registro de nuevos datos.
- c) El sistema de citas debe ser muy flexible puesto que de vez en cuando se cuenta con un personal que realiza atención clínica y peluquería; y la mayoría de las ocasiones se cuenta con más personal por lo que en un determinado tiempo se puede atender varias mascotas.

Como es natural, la lista de suposiciones y restricciones se incrementará durante el desarrollo del proyecto, particularmente una vez establecido el artefacto “Visión”.

I.1.2.2.3 Entregables del proyecto

A continuación, se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

I.1.2.2.3.1 Plan de Desarrollo del Software

Corresponde a la sección II.1.2 de este documento.

I.1.2.2.3.2 Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.). permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

I.1.2.2.3.2.1 Actores

I.1.2.2.3.2.1.1 Proveedor

Es la empresa encargada de proveer productos para la clínica veterinaria

I.1.2.2.3.2.1.2 Cliente

Es la persona propietaria de la mascota y tiene la responsabilidad de la crianza de la mascota realiza actividades como registrar sus datos y los de su mascota, comprar productos, contratar servicios y solicitar una cita.

I.1.2.2.3.2.2 Diagramas de Casos de Uso del Negocio

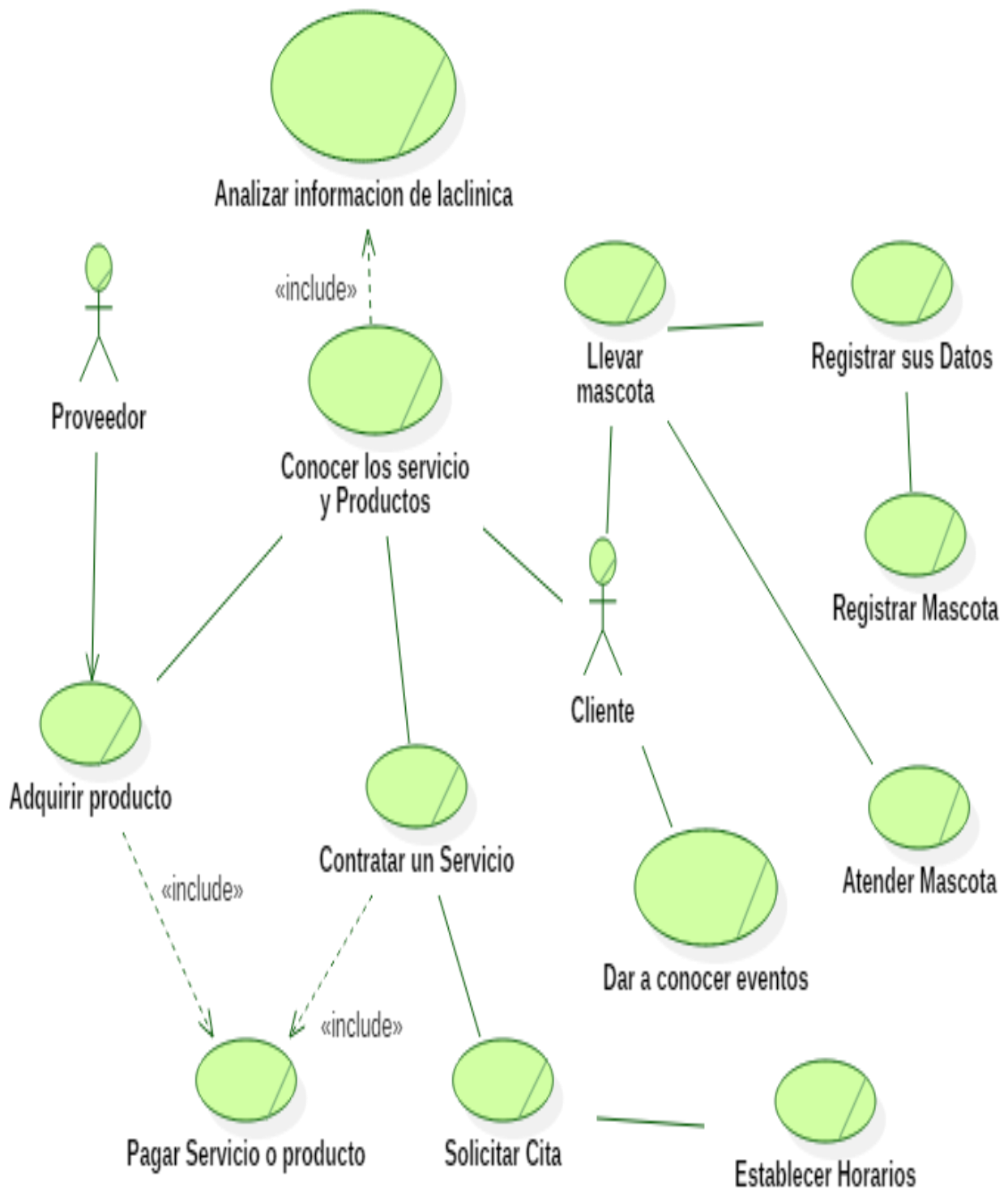


Ilustración 0.32 Modelo de Negocio

I.1.2.2.3.3 Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.

I.1.2.2.3.3.1 Introducción

La terminología que mayormente se utilizara en el proyecto está relacionado con tecnología web y la tecnología móvil como también algunos términos utilizados en los procesos de la clínica veterinaria.

I.1.2.2.3.3.1.1 Propósito

El propósito de este entregable es facilitar a los lectores la comprensión de términos técnicos incluidos en este documento para que el documento sea comprendido a su máximo nivel.

I.1.2.2.3.3.1.2 Alcance

En este entregable se utilizarán los términos utilizados en los tres componentes del proyecto como lo son el sistema informático, la aplicación móvil y el modelo de socialización de los componentes

I.1.2.2.3.3.1.3 Referencias

El presente glosario hace referencia a los siguientes entregables:

- Plan de desarrollo de software de proyecto
- Visión del proyecto
- Especificación de casos de uso del proyecto
- Especificación e casos de prueba del proyecto.

I.1.2.2.3.3.1.4 Organización del Glosario

El presente entregable está organizado por definiciones de términos ordenados de forma ascendente según la ordenación alfabética tradicional del español.

I.1.2.2.3.3.2 Definiciones

- **Ancho de banda:** Es la cantidad de datos (medida en Mega o Gigabytes (MB, GB)) que puedo transmitir y/o recibir.
- **API APPLICATION PROGRAM INTERFACE,** y representa una interfaz de comunicación entre componentes software
- **Archivos:** sección del weblog en la que se accede a todos los contenidos publicados, normalmente organizada por años y meses.
- **Categorías:** conjunto de secciones temáticas de un weblog, sirven como forma alternativa de navegación. Suelen estar relacionadas con la temática del blog.
- **Código Fuente:** En inglés Source Code. Conjunto de instrucciones que componen el núcleo de un programa informático
- **Comentarios:** en cada anotación, designa las respuestas, sugerencias, opiniones y críticas de los lectores de un weblog. Constituyen el espacio donde tiene lugar la conversación pública entre el blogger y sus lectores.
- **CSS:** Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.
- **Dominio:** Dirección de Internet.
- **Hipervínculo:** (También llamado hiperenlace, link o liga) es un elemento de un que hace referencia a otro recurso, por ejemplo, otro documento o un punto específico del mismo o de otro documento.
- **Hosting:** Espacio en un servidor informático de almacenamiento que se compra para alojar las páginas web y se puedan ver en Internet.
- **Interfaz:** Es la representación gráfica de una aplicación interactiva, es lo que media y facilita la comunicación e interacción entre el individuo y la máquina, ella es lo que el usuario ve en la pantalla y con la que interactúa finalmente.
- **Login** Clave de acceso que se le asigna a un usuario para acceder de forma segura a una web.
- **Microsoft:** es una empresa multinacional estadounidense (la mayor del mundo), fundada en 1975 por Bill Gates y Paul Allen. Dedicada al sector de la informática, con sede en Redmond,

Washington, Estados Unidos. Microsoft desarrolla, fabrica, licencia y produce software para equipos electrónicos. Siendo sus productos más usados el Sistema operativo Microsoft Windows y la suite Microsoft Office, estos productos tienen una importante posición entre los ordenadores personales. Con una cuota de mercado cercana al 90% para Office en 2003 y para Windows en el 2006. Siguiendo la estrategia de Bill Gates de "tener una estación de trabajo que funcione con nuestro software en cada escritorio y en cada hogar".

- **Office:** Microsoft Office (MSO) es un conjunto de programas informáticos desarrollados por la empresa Microsoft.
- **Office** hizo su primera aparición en 1989 en un Mac, y más adelante en Windows en 1990. El término fue inicialmente usado de marketing para vender un set de aplicaciones, que previamente se vendían separadamente. El principal argumento de venta era que comprar el paquete completo resultaba más barato que comprar cada aplicación por separado. La primera versión de Office contenía las aplicaciones Microsoft Word, Microsoft Excel y Microsoft PowerPoint. Adicionalmente, una llamada "versión profesional" de Office incluía también Microsoft Access y Schedule Plus.
- **Off Line** Opuesto a on line, fuera de conexión o de línea.
- **On Line** En línea o en tiempo real.
- **Página Web:** Es una unidad de información individual a la que se accede a través de la WWW, en ella se presentan texto, imágenes, sonidos, vídeos y enlaces, y su extensión puede ser más larga que una pantalla de computador.
- **Servidor:** Es el computador en el que se ejecutan los programas que realizan alguna tarea en beneficio de otras computadoras. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un sitio web, así como ejecutar sus aplicaciones asociadas, en beneficio directo del usuario final.
- **Sistemas Operativos:** Constituyen un conjunto de programas básicos y utilidades, compilados y distribuidos en formato sencillo para su fácil instalación, y de los cuales depende el funcionamiento de la computadora.
- **Sitio Web:** Varias páginas web pueden estar agrupadas conformando un sitio Web, se trata de productos comunicacionales cuya característica básica es que potencian una “deestructuración comunicativa”, es decir, que el sitio no hace explícito todas sus posibilidades de una sola vez, sino que para conocerlas se incita al usuario a explorar y a interactuar con los distintos elementos que aparecen en pantalla.

- **URL:** acrónimo de Uniform Resource Locator, dirección de una página web.
- **World Wide Web:** Es un sistema hipermedial que permite el acceso a una diversidad de contenidos a través del seguimiento de enlaces desde un elemento a otro, colocando a disposición de los usuarios referencias o vínculos a otros documentos ya sean en formato de sonidos, gráficas o vídeo.

I.1.2.2.3.3.3 Estereotipos UML

No existen estereotipos adicionales que no estén predefinidos por UML

I.1.2.2.3.4 Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso. Estas funciones fueron obtenidas del Documento de ERS Neg-VET más detalles sobre el Documento de ERS en el ANEXO 1.

I.1.2.2.3.4.1 Introducción

I.1.2.2.3.4.1.1 Propósito

Representación de Modelo de Casos de Uso del Sistema Neg-VET

I.1.2.2.3.4.1.2 Alcance

Modelo de Casos de Uso del sistema explosionados a segundo nivel.

I.1.2.2.3.4.2 Diagrama de Casos de Uso General



Ilustración 0.33 Diagrama de Casos de Uso

I.1.2.2.3.4.3 Explosión de Casos de Uso General

I.1.2.2.3.4.3.1 Explosión del Caso de Uso Ingresar al Sistema

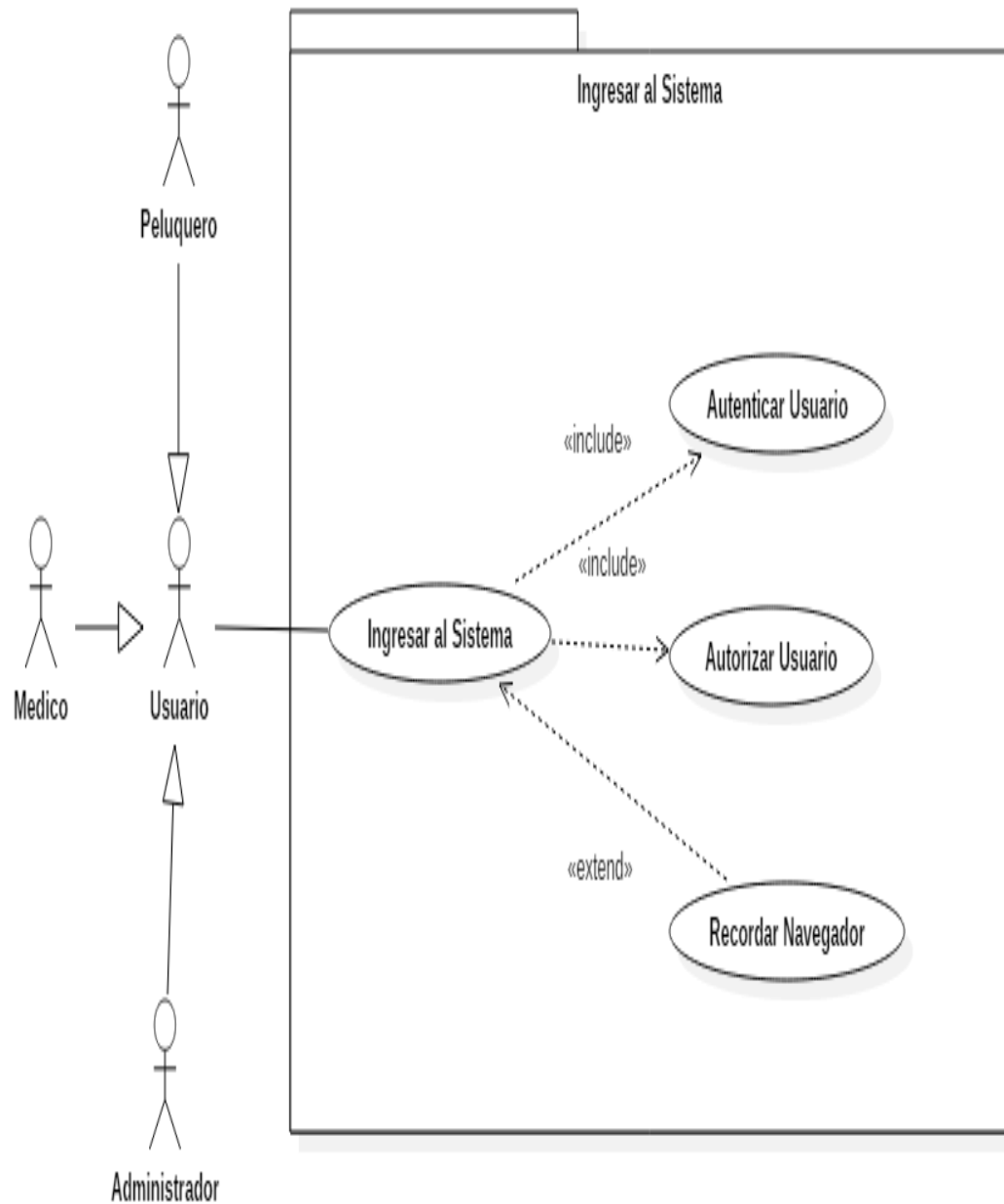


Ilustración 0.34 Diagrama de CU Ingresar al Sistema

I.1.2.2.3.4.3.2 Explosión del Caso de Uso Gestionar Usuarios



Ilustración 0.35 Diagrama CU Gestionar Usuarios

I.1.2.2.3.4.3.3 Explosión del Caso de Uso Gestionar Roles

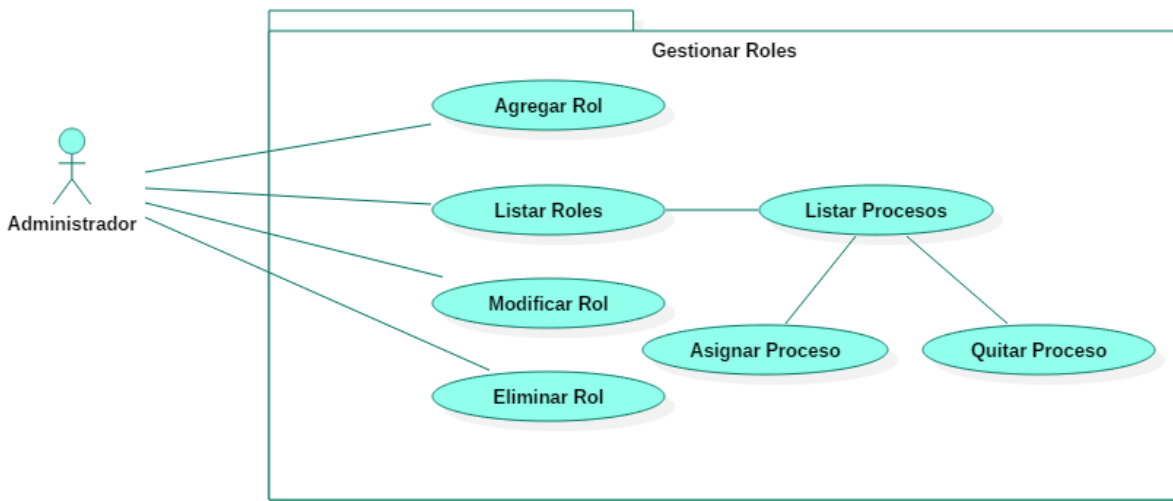


Ilustración 0.36 Diagrama CU Gestionar Roles

I.1.2.2.3.4.3.4 Explosión del Caso de Uso Generar Reportes

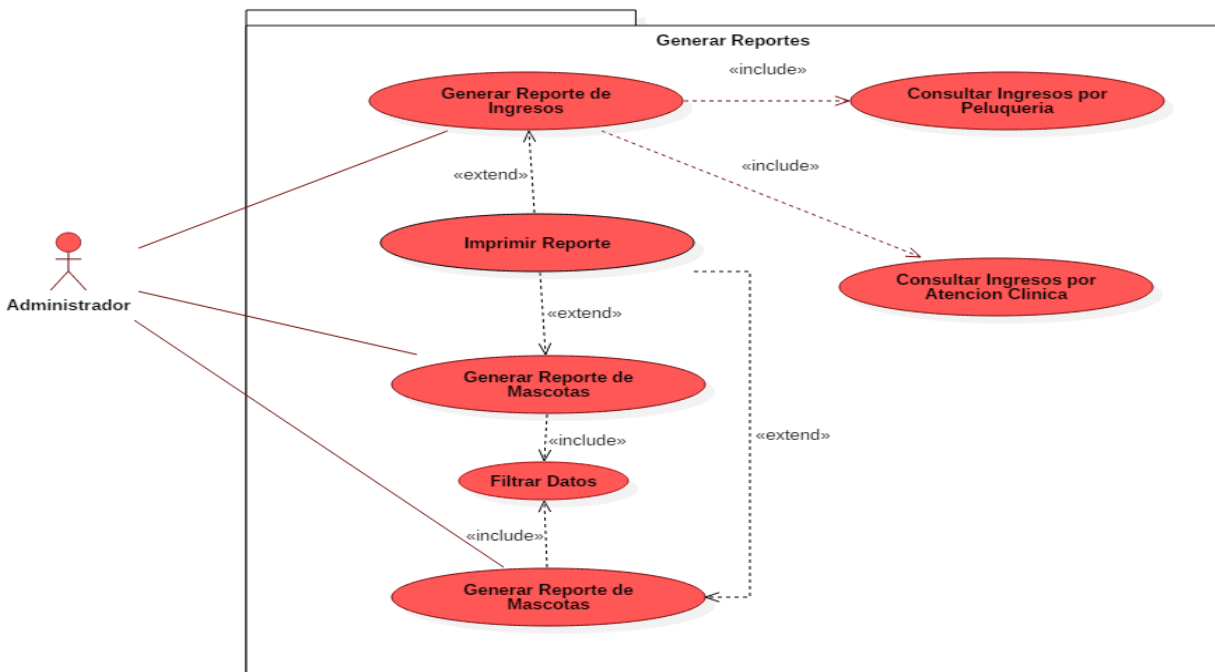


Ilustración 0.37 Diagrama CU Generar Reportes

I.1.2.2.3.4.3.5 Explosión del Caso de Uso Gestionar Noticias

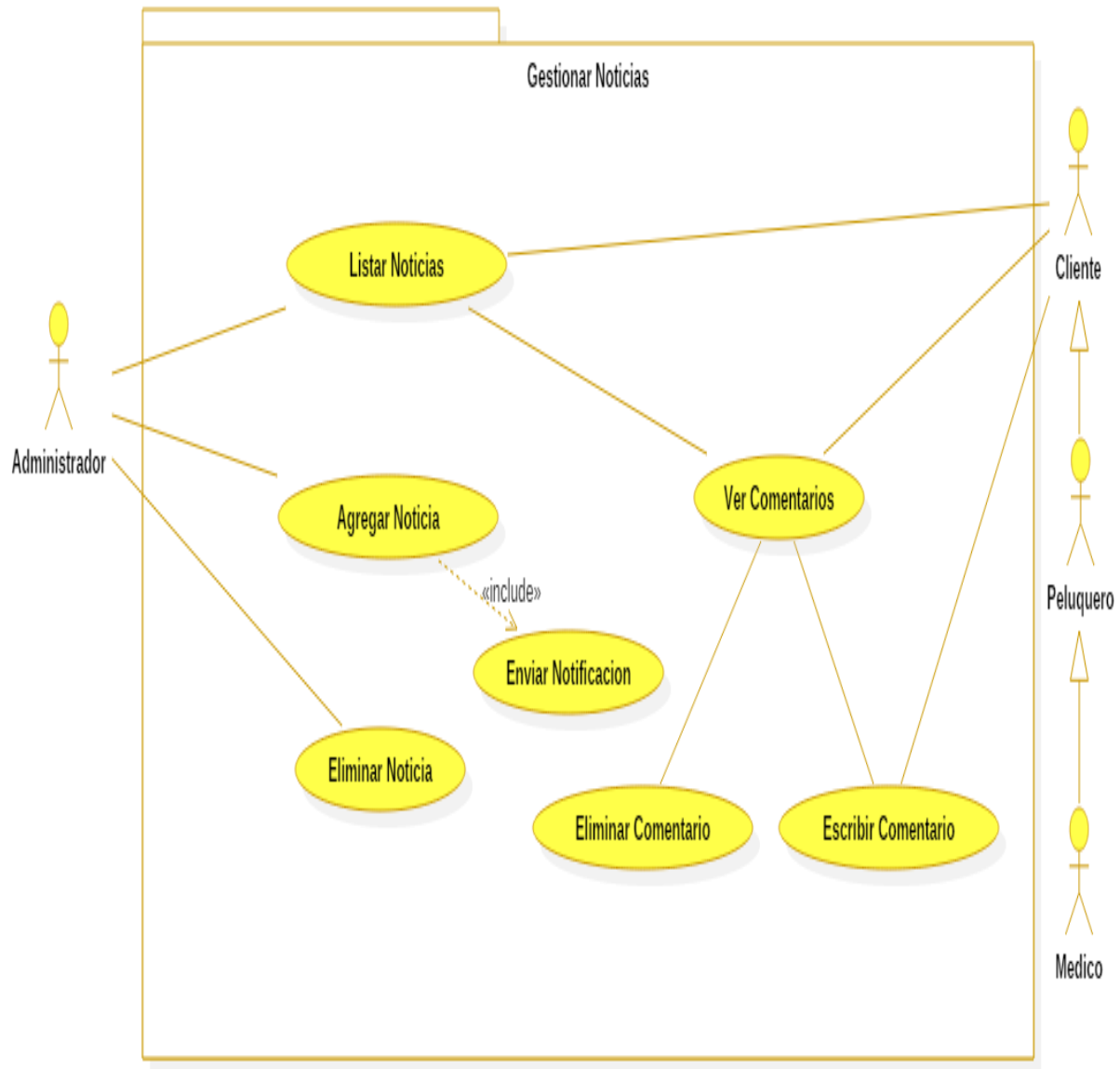


Ilustración 0.38 Diagrama CU Gestionar Noticias

I.1.2.2.3.4.3.6 Explosión del Caso de Uso Gestionar Medicamentos

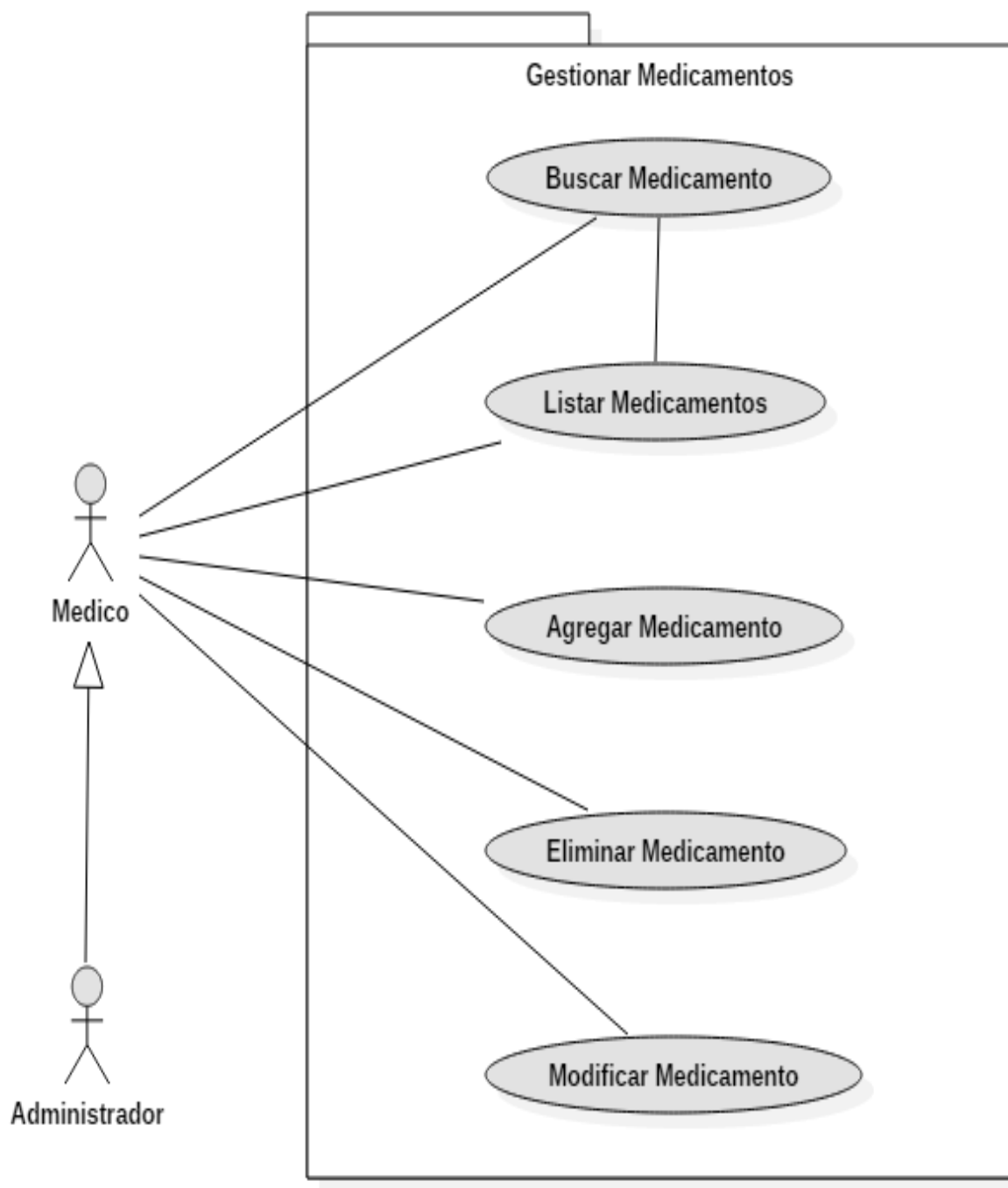


Ilustración 0.39 Diagrama CU Gestionar Medicamentos

I.1.2.2.3.4.3.7 Explosión del Caso de Uso Gestionar Clientes

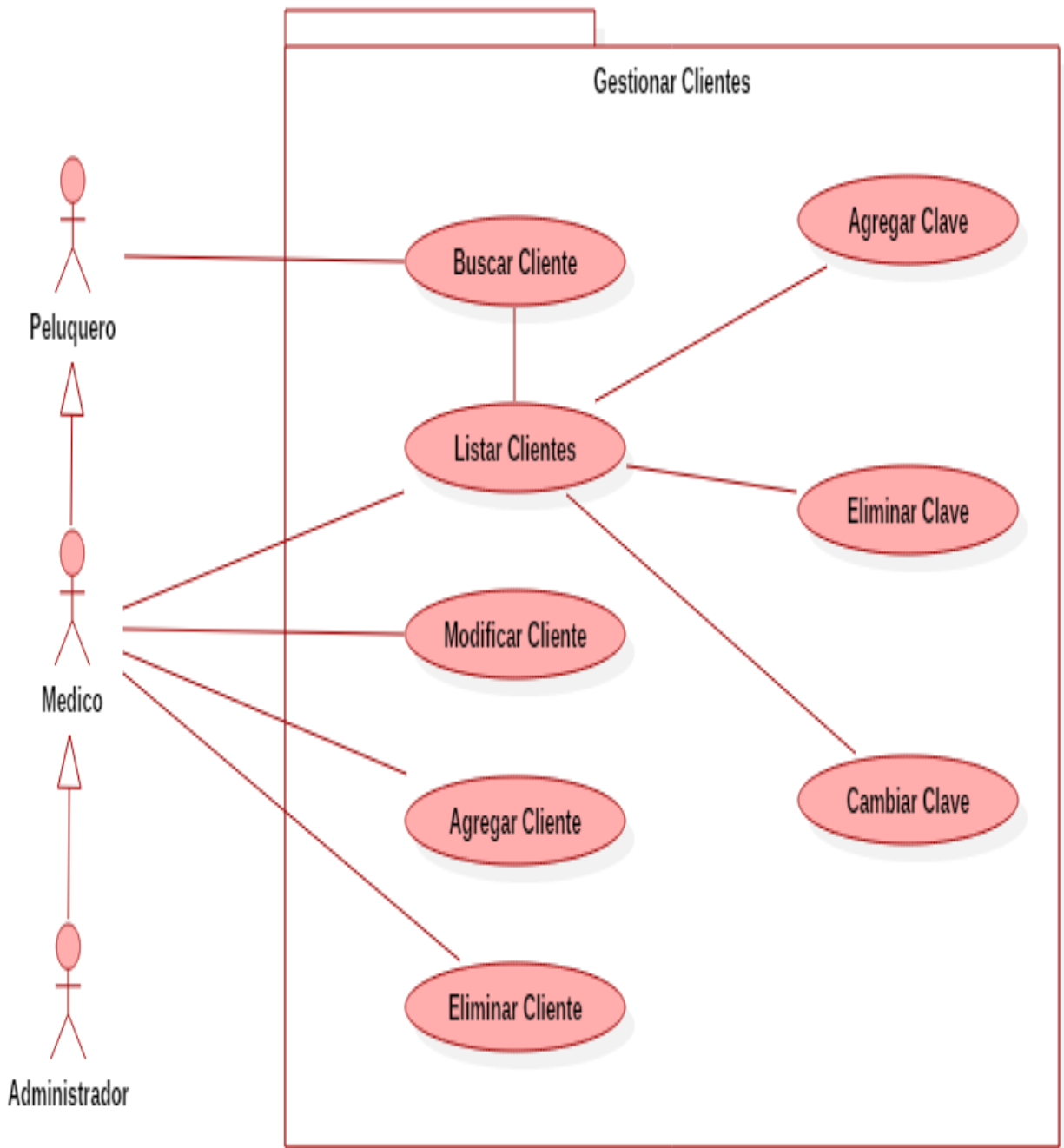


Ilustración 0.40 Diagrama CU Gestionar Clientes

I.1.2.2.3.4.3.8 Explosión del Caso de Uso Gestionar Mascotas

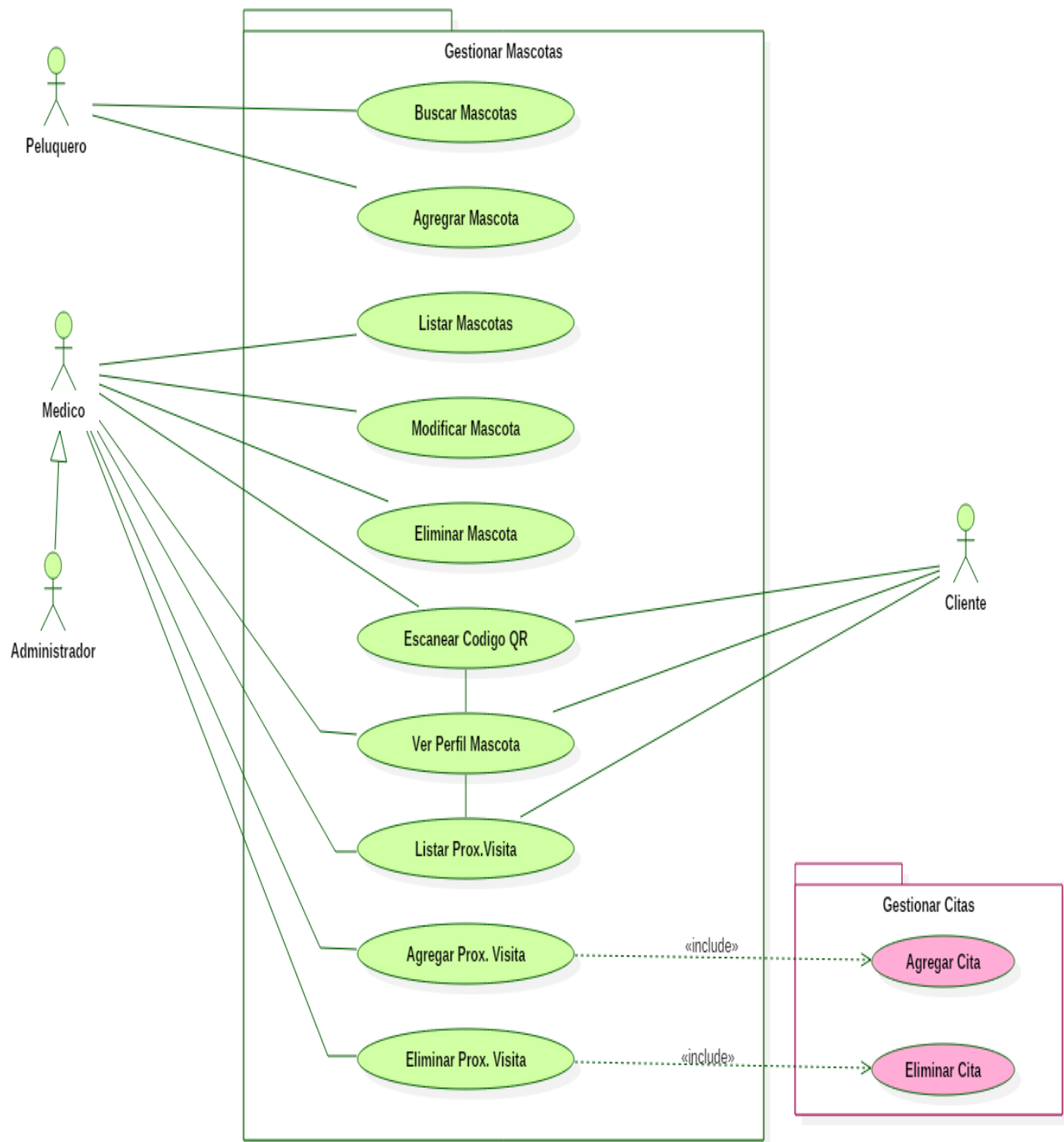


Ilustración 0.41 Diagrama CU Gestionar Mascotas

I.1.2.2.3.4.3.9 Explosión del Caso de Uso Gestionar Peluquerías

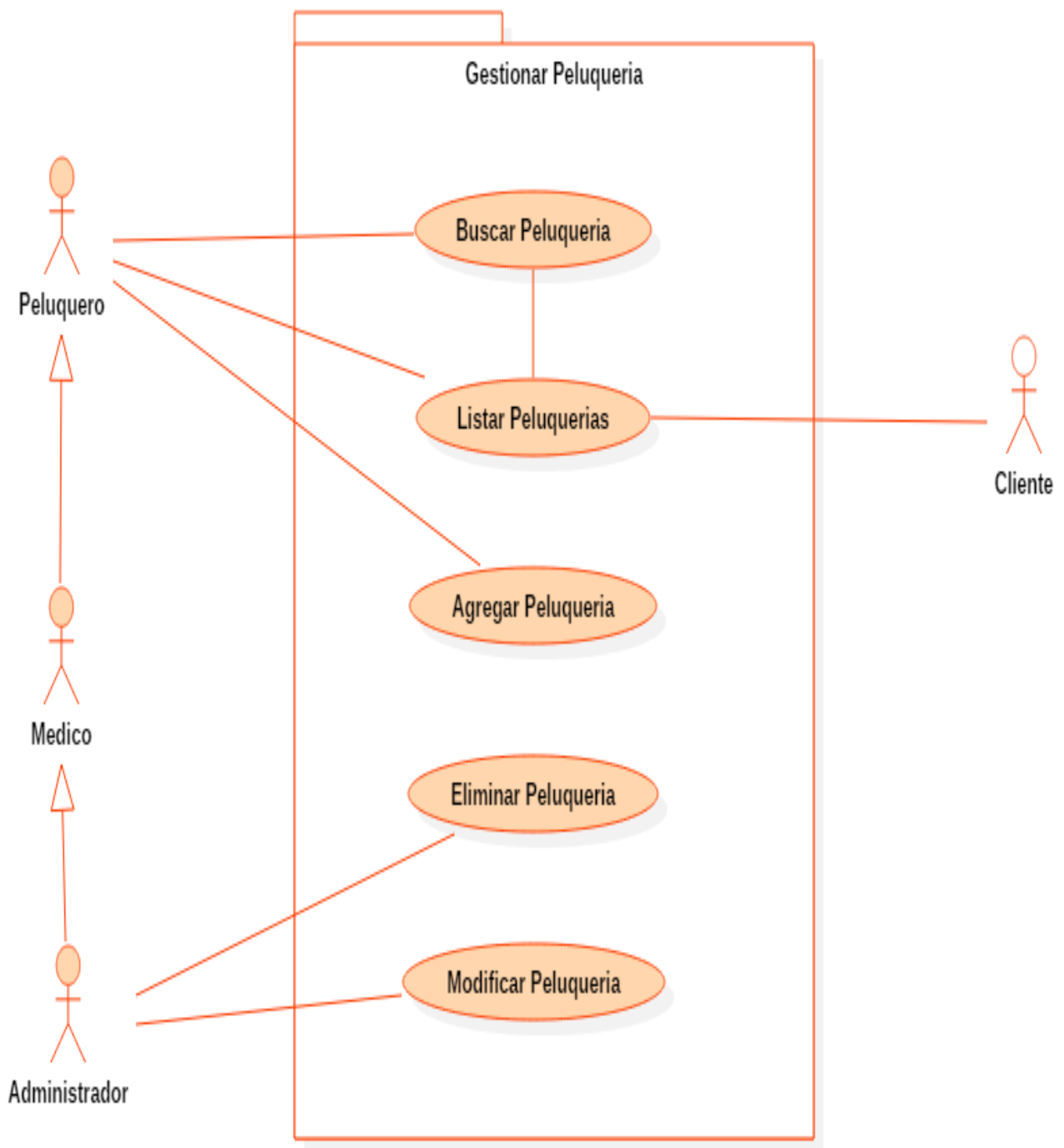


Ilustración 0.42 Diagrama CU Gestionar Peluquerías

I.1.2.2.3.4.3.10

Explosión del Caso de Uso Gestionar Control Sanitario

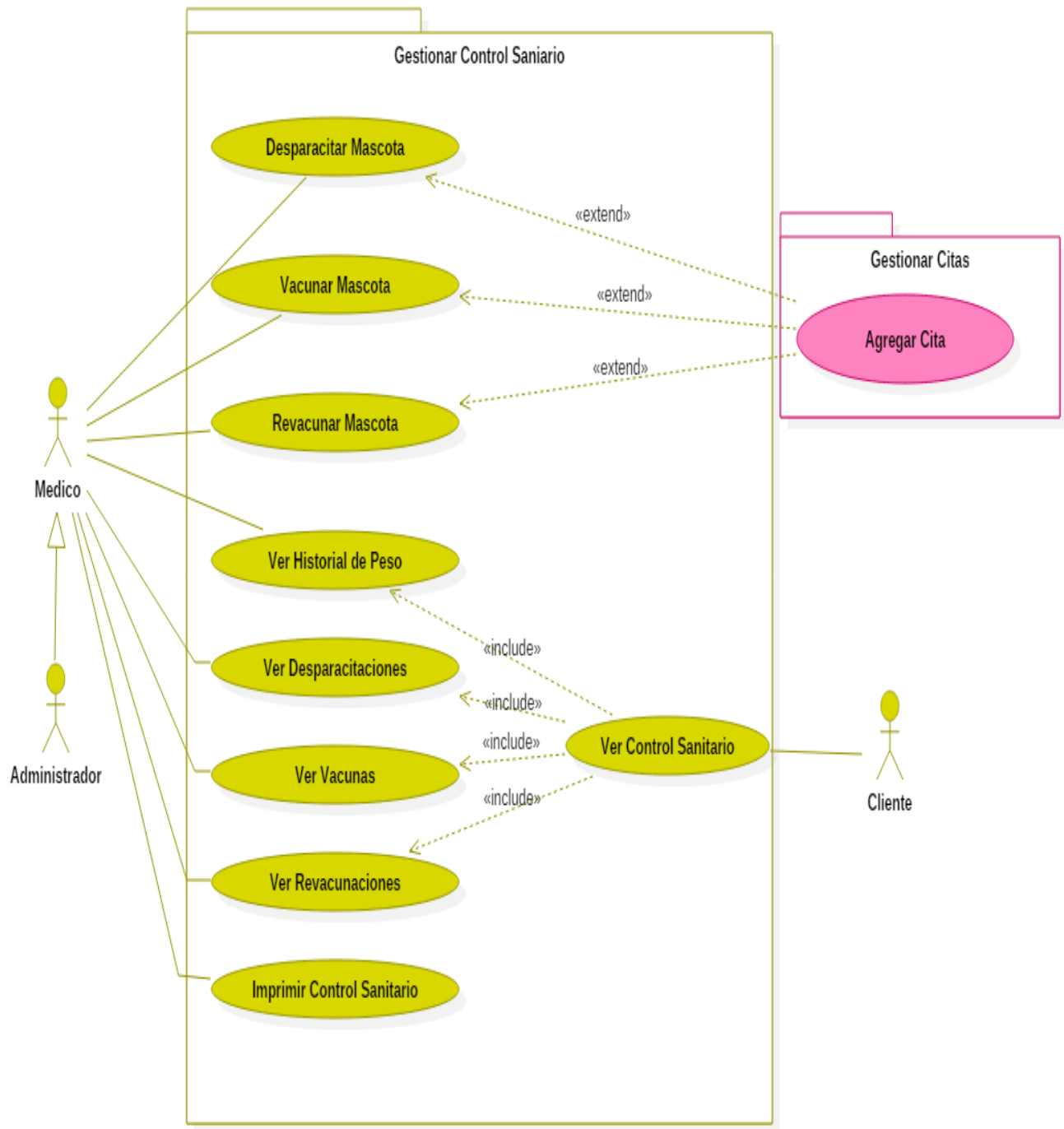


Ilustración 0.43 Diagrama CU Gestionar Control Sanitario

I.1.2.2.3.4.3.11

Explosión del Caso de Uso Gestionar Ficha Clínica

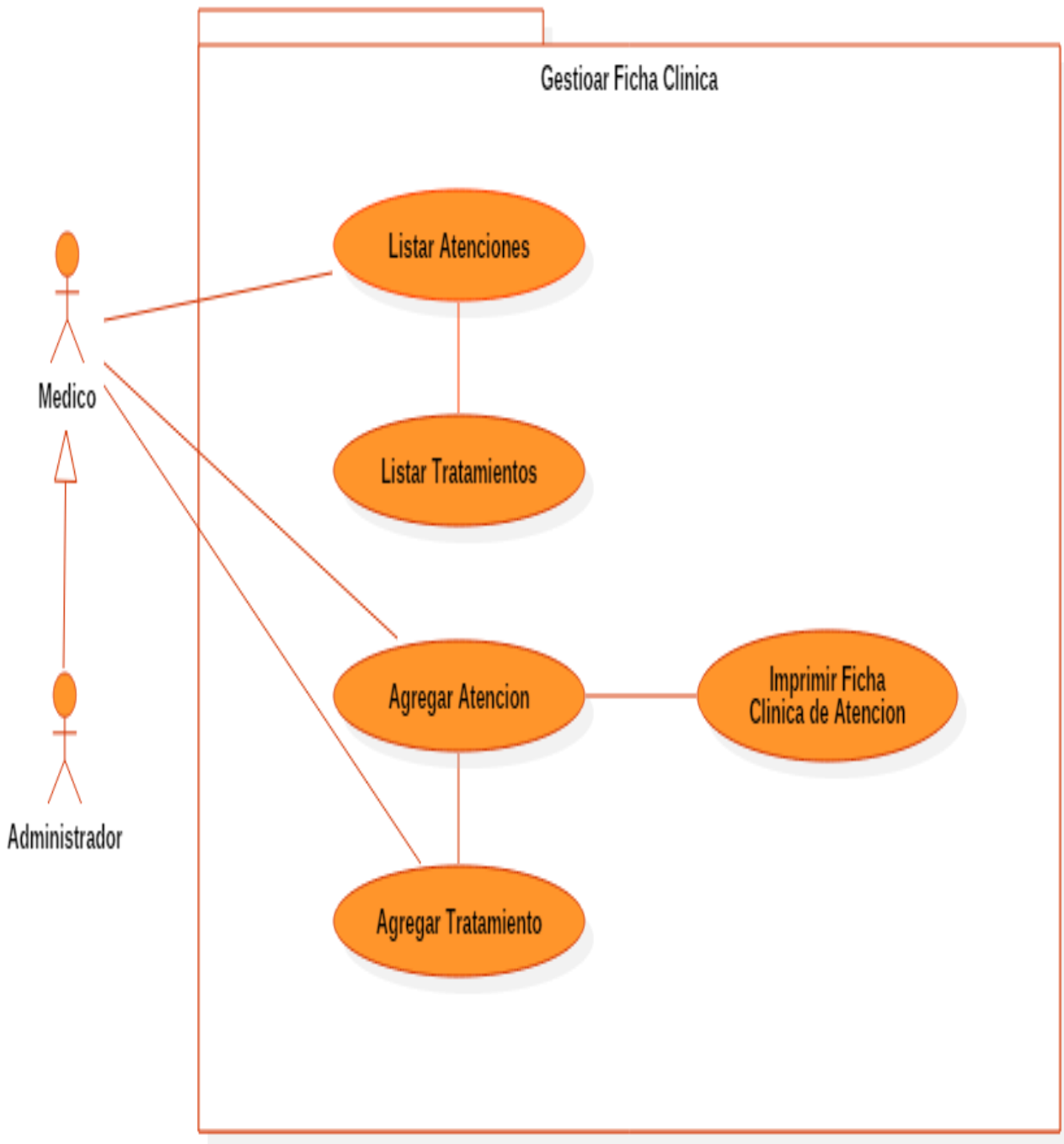


Ilustración 0.44 Diagrama CU Gestionar Ficha Clínica

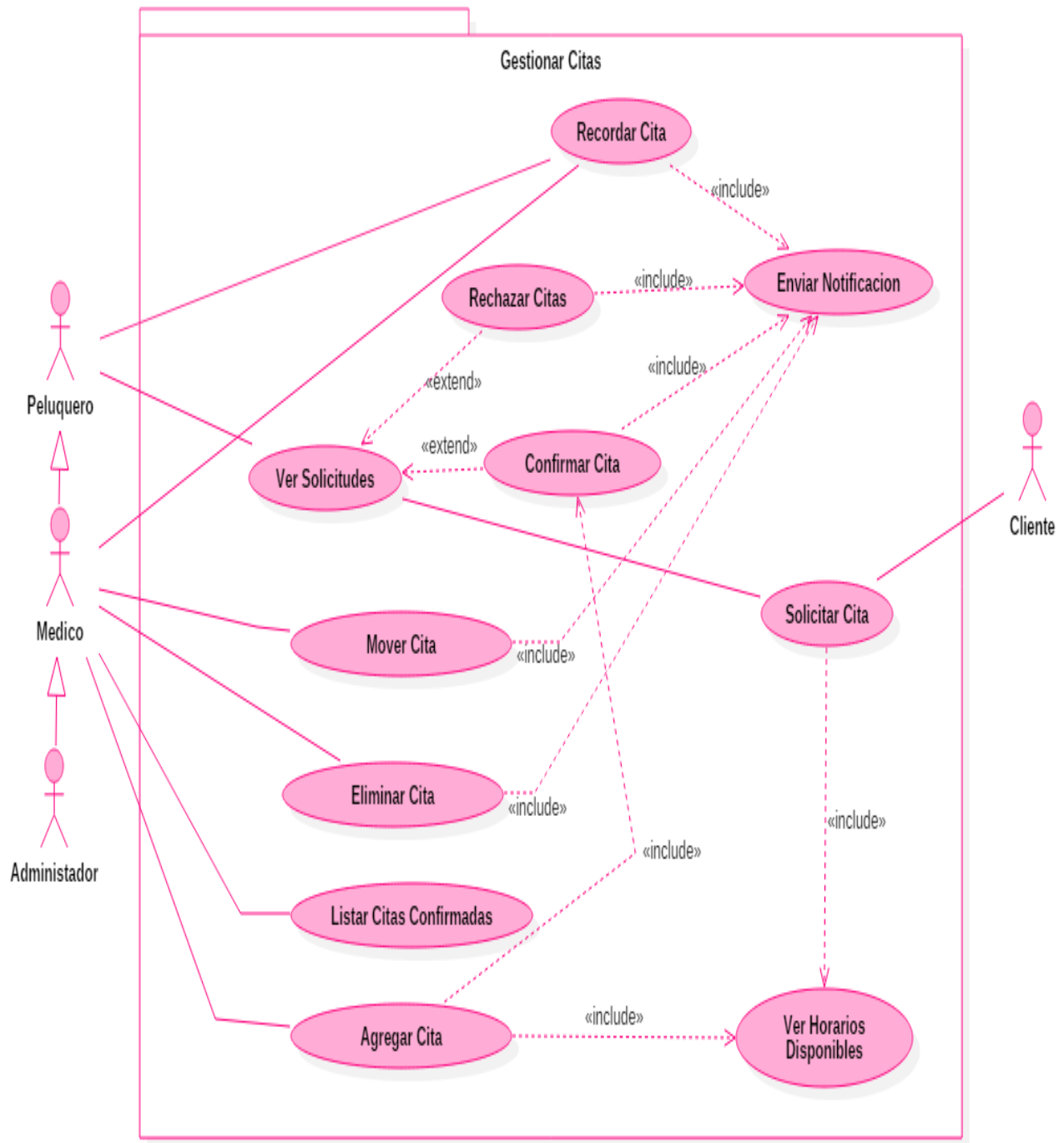


Ilustración 0.45 Diagrama CU Gestionar Citas

I.1.2.2.3.4.3.13 Explosión del Caso de Uso Perfil

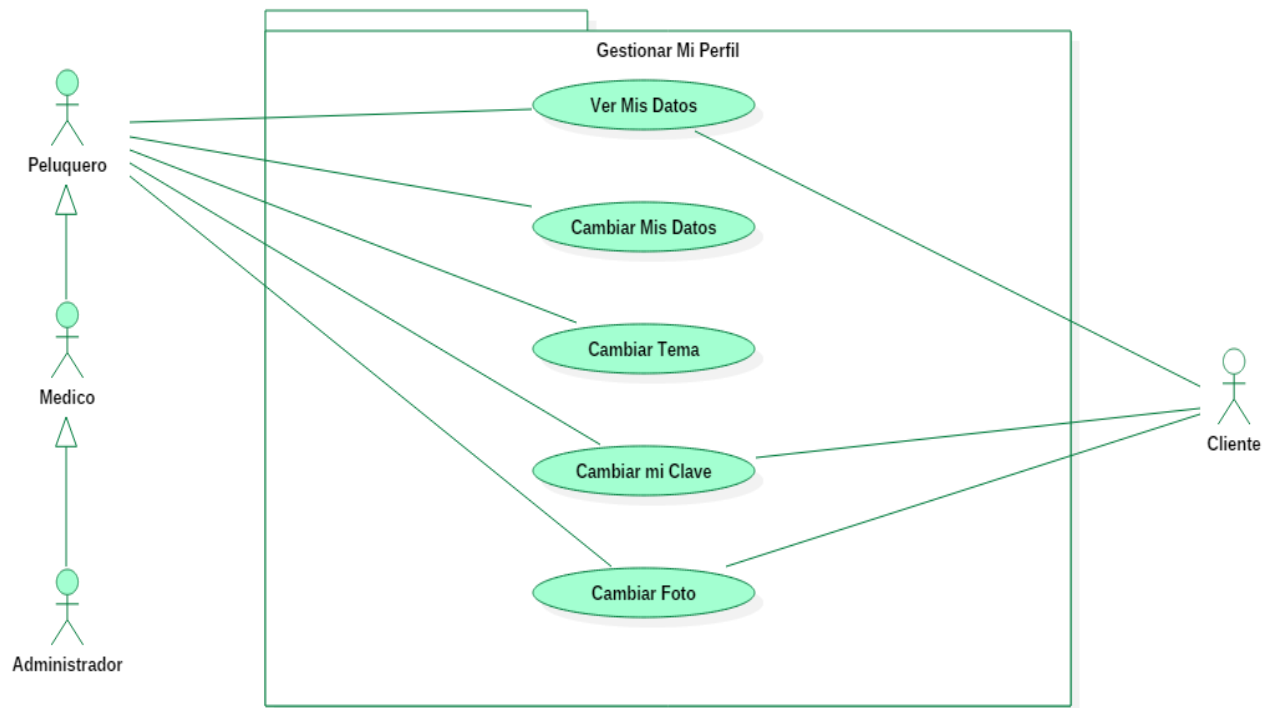


Ilustración 0.46 Diagrama CU Gestionar Mi Perfil

I.1.2.2.3.5 Visión

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.

I.1.2.2.3.5.1 Introducción

I.1.2.2.3.5.1.1 Propósito

El propósito de éste documento es recoger, analizar y definir las necesidades de alto nivel y las características del sistema de administración de pacientes de la clínica veterinaria. El documento se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales.

Esta funcionalidad se basa principalmente en la gestión de los almacenes que la empresa tiene repartidos por las distintas zonas en las que actúa, de forma que dichos almacenes sean capaces de atender los distintos pedidos que les son realizados.

Los detalles de cómo el sistema cubre los requerimientos se pueden observar en la especificación de los casos de uso y otros documentos adicionales.

I.1.2.2.3.5.1.2 Alcance

El documento Visión se ocupa, como ya se ha apuntado, del sistema de administración de pacientes de la clínica veterinaria. Dicho sistema será desarrollado por el estudiante Rolando Abelino Sanchez Garay.

El sistema permitirá a los usuarios de la clínica gestionar la información de sus pacientes, Los propietarios de las mascotas, los controles sanitarios, las atenciones clínicas, peluquerías y gestionar Citas. De igual forma servirá para que la clínica socialice sus noticias mediante una aplicación móvil ANDROID que estará integrada con el sistema. Además, también permitirá a los clientes realizar citas online, mediante una aplicación móvil, y enviar recordatorios de sus citas etc.

I.1.2.2.3.5.1.3 Definiciones Acrónicas

RUP: Son las siglas de Rational Unified Process. Se trata de una metodología para describir el proceso de desarrollo de software.

I.1.2.2.3.5.1.4 Referencias

- Glosario.
- Plan de desarrollo de software.
- RUP (Rational Unified Process).
- Diagrama de casos de uso.

I.1.2.2.3.5.2 Posicionamiento

I.1.2.2.3.5.2.1 Oportunidad del negocio

Este sistema permitirá a la empresa informatizar el control de algunos de sus procesos (gestión de clientes, gestión de mascotas, gestión de citas y Noticias), lo cual supondrá un acceso rápido y sencillo a los datos, gracias a interfaces gráficas sencillas y amigables. Además, los datos accedidos estarán siempre actualizados, lo cual es un factor muy importante para poder llevar un control centralizado de los distintos pacientes.

El sistema también permite a los clientes acceder a los servicios de la empresa a través de una aplicación móvil ANDROID, de forma rápida y sencilla.

I.1.2.2.3.5.2.2 Sentencia que define el Problema

El problema de	Controlar la información de las mascotas como el control sanitario, sus atenciones clínicas y peluquerías. Gestionar las noticias para distribuir a los clientes. Gestionar las citas de la clínica.
afecta a	Médico Veterinario, Peluquero, Administrador y Clientes.
El impacto asociado es	Almacenar toda la información referente a los pacientes, clientes, peluquerías, controles sanitarios, atenciones clínicas, citas y noticias, y que esta información esté al instante accesible y actualizada en lugares físicamente muy distantes es un proceso prácticamente imposible de realizar en el caso de que no esté informatizado.
Una solución adecuada sería	Informatizar el proceso, usando internet y generar interfaces amigables y sencillas con las que acceder a dicha base de datos.

Tabla 0.2 Sentencia que define el problema

I.1.2.2.3.5.2.3 Sentencia que define la Posición del Producto

para	Médico Veterinario, Peluquero, Administrador y Clientes.
quienes	Controlan los pacientes, clientes, los controles sanitarios, las atenciones clínicas, las peluquerías, las noticias y las citas los clientes solo pueden solicitar citas.
El nombre del producto	Es una herramienta software.
que	Almacena la información necesaria para gestionar una clínica veterinaria.
no como	El sistema manual actualmente utilizado.
Nuestro producto	Permite gestionar las distintas actividades de la clínica mediante una interfaz gráfica sencilla y amigable. Además proporciona un acceso rápido y actualizado a la información desde cualquier punto que tenga acceso a la base de datos.

Tabla 0.3 Sentencia que define la posición del producto

I.1.2.2.3.5.3 Descripción de Stakeholders (Participantes en el Proyecto) y Usuarios

Para proveer de una forma efectiva productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los usuarios del sistema y asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que éstos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que éstos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de por qué estos requisitos son necesarios.

I.1.2.2.3.5.3.1 Resumen de Stakeholders

Nombre	Descripción	Responsabilidades
Dra. Ximena Jurado.	Representante Global de la clínica veterinaria AMEVET	El stakeholder realiza: Representa a todos los usuarios posibles del sistema. Seguimiento del desarrollo del

		proyecto. Aprueba requisitos y funcionalidades
--	--	---

Tabla 0.4 Resumen de Stakeholders

I.1.2.2.3.5.3.2 Resumen de Usuarios

Nombre	Descripción	Stakeholder
Dra. Ximena Jurado	Gerente propietaria , encargado de administrar el sistema por completo y la posibilidad de promocionar las noticias de la clínica.	Medico Veterinaria
Dra. Nehyda Ramos	Responsable médico veterinario , encargado de la gestión de la información de los pacientes, la gestión de citas.,	Medico Veterinaria
Nataly Jurado	Apoyar al médico veterinario responsable así como también se encarga de la estética canina de la veterinaria.	Auxiliar

Tabla 0.5 Resumen de usuarios

I.1.2.2.3.5.3.3 Entorno de usuario

Los usuarios entrarán al sistema identificándose ingresando a la página de acceso de sistema mediante un dispositivo con conexión a internet y tras este paso entrarán a la parte de aplicación diseñada para cada uno según su papel en la empresa. Este sistema es similar a cualquier aplicación Web y por tanto los usuarios estarán familiarizados con su entorno.

Los informes serán generados con Microsoft Word versión 2016, lo cual también resultará familiar

I.1.2.2.3.5.4 Descripción Global del Producto

I.1.2.2.3.5.4.1 Perspectiva del Producto

El producto a desarrollar es un sistema global para la clínica veterinaria “AMEVET”, con la intención de agilizar su funcionamiento.

I.1.2.2.3.5.4.2 Características del Producto

A continuación, se mostrará un listado con los beneficios que obtendrá el cliente a partir del producto:

Beneficio del cliente	Características que lo apoyan
Mayor agilidad en las citas dando la posibilidad de hacerlo vía móvil.	Aplicación móvil desde la cual poder realizar las solicitudes.
Gestión automatizada de la información de los pacientes.	Sistema de optimización de del stock en el almacén y previsión de pedidos
Mayor facilidad para la gestión de los pacientes.	Base de datos centralizada con la información de todos los pacientes.
Posibilidad de mantener al tanto a los clientes sobre las noticias que comunica la clínica.	Aplicación móvil desde la que va a poder ver las noticias.
Envío de notificaciones a los clientes.	Aplicación móvil que comunica a los clientes las notificaciones en tiempo real.

Tabla 0.6 Características del Producto

I.1.2.2.3.6 Especificaciones de Casos de Uso

I.1.2.2.3.6.1 Introducción

I.1.2.2.3.6.1.1 Propósito

El propósito de éste documento es ofrecer más detalles sobre los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. También,

para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

I.1.2.2.3.6.1.2 Alcance

El documento Especificaciones de caso de uso se ocupa, como ya se ha apuntado, de detallar la funcionalidad del sistema de administración de pacientes de la clínica veterinaria.

La especificación se la realizara a los caos de usos que se consideren que su descripción narrativa no es suficiente para entender el proceso.

I.1.2.2.3.6.2 Especificaciones de Casos de Uso

I.1.2.2.3.6.2.1 Especificación de Caso de Uso Ingresar al Sistema

Caso de Uso	Ingresar al Sistema
Descripción	En este caso de uso el actor puede agregar un nuevo rol al sistema.
Actores	Administrador, Medico, Peluquero
Precondiciones	El actor debe ingresar a la página de acceso al sistema.
Flujo Normal	1. El sistema muestra un formulario para ingresar el nombre de usuario y la clave. 2. El actor ingresa los datos 3. El actor envía los datos 4. El sistema autentica los datos. 5. El sistema autoriza al usuario.
Flujo Alternativo	2.A El actor selecciona la opción recordar navegador. 4.A El sistema comprueba la validez de los datos ingresado, si no pertenecen a un usuario dado de alta en el sistema este muestra un mensaje error de Usuario no Encontrado.
Pos-condiciones	El usuario ha ingresado al sistema.

Tabla 0.7 Especificación CU Ingresar al sistema

I.1.2.2.3.6.2.2 Especificación de Caso de Uso Gestionar Usuarios

Caso de Uso	Agregar Usuario
--------------------	------------------------

Descripción	En este caso de uso el actor puede agregar datos de un nuevo usuario.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Usuarios.
Flujo Normal	1. El actor debe presionar en el botón de agregar usuario. 2. El sistema muestra una pantalla con un formulario para el llenado de datos. 3. El actor ingresa los datos. 4. El sistema comprueba la validez de los datos ingresados. 5. El actor envía los datos. 6. El Sistema almacena los datos. 7. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos ingresado, si los datos ingresado están correctos el sistema habilita el botón de guardar. 4.B Si ya existe un login como el ingresado el sistema muestra un mensaje de error. 6.C Si ocurre un error al almacenar los datos el sistema informa con un mensaje al actor.
Pos-condiciones	El usuario ha sido almacenado en el sistema.

Tabla 0.8 Especificación CU Agregar Usuario

Caso de Uso	Buscar Usuarios
Descripción	En este caso de uso el actor puede buscar un usuario determinado.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Usuarios.
Flujo Normal	1. El actor se sitúa dentro de la caja de texto de búsqueda. 2. EL actor debe escribir los términos de búsqueda (Nombre o Apellido Paterno o Apellido Materno). 3. El sistema devuelve una lista de usuarios que coincide con los términos de búsqueda.
Flujo Alternativo	3.A Si el sistema no encuentra resultados muestra un mensaje indicando que no encontró resultados.
Pos-condiciones	ninguna

Tabla 0.9 Especificaciones CU Buscar Usuarios

Caso de Uso	Listar Usuarios
--------------------	------------------------

Descripción	En este caso de uso el actor puede ver una lista de los usuarios registrados en el sistema.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Usuarios.
Flujo Normal	1. El Sistema devuelve una lista de usuarios dados de alta en el sistema.
Flujo Alternativo	1.A Si no existe usuario registrados muestra un mensaje indicando que no encontró resultados.
Pos-condiciones	ninguna

Tabla 0.10 Especificaciones CU Listar Usuarios

Caso de Uso	Eliminar Usuario
Descripción	En este caso de uso el actor puede dar de baja al usuario del sistema.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Usuarios.
Flujo Normal	1. El actor debe presionar en el botón de eliminar usuario. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta la baja del usuario. 4. El sistema da de baja al usuario. 5. El sistema muestra un mensaje de éxito.
Flujo Alternativo	2.A El sistema muestra un dialogo de confirmación, si el actor presiona CANCELAR, la operación se cancela. 4.B Si ocurre un error al dar de baja al usuario el sistema muestra un mensaje de error.
Pos-condiciones	El usuario ha sido dado de baja del sistema.

Tabla 0.11 Especificación CU Eliminar Usuario

Caso de Uso	Modificar Usuario
Descripción	En este caso de uso el actor puede modificar los datos de un usuario registrado en el sistema
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema.

	El actor debe ingresar a la sección de Usuarios.
Flujo Normal	1. El actor debe presionar el botón de modificar. 2. El sistema muestra un formulario con los datos actuales. 3. El actor ingresa los nuevos datos del usuario. 4. El sistema comprueba la validez de los datos. 5. El actor envía los datos. 6. El sistema actualiza los nuevos datos. 7. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos ingresados, si los datos ingresado están correctos el sistema habilita el botón de guardar.
Pos-condiciones	Los datos del usuario han sido actualizados.

Tabla 0.12 Especificación CU Modificar Usuario

Caso de Uso	Cambiar Clave
Descripción	En este caso de uso el actor puede cambiar la clave de acceso al sistema del usuario.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Usuarios.
Flujo Normal	1. El actor debe presionar el botón de cambiar clave. 2. El sistema muestra un campo de texto para ingresa la nueva clave. 3. El actor ingresa la nueva clave. 4. El sistema comprueba la validez de la clave. 5. El actor envía la nueva clave. 6. El sistema actualiza la clave. 7. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos ingresados, si los datos ingresado están correctos el sistema habilita el botón de guardar.
Pos-condiciones	La clave del usuario ha sido actualizada.

Tabla 0.13 Especificación CU Cambiar Clave

I.1.2.2.3.6.2.3 Especificación de Caso de Uso Gestionar Roles

Caso de Uso	Agregar Rol
Descripción	En este caso de uso el actor puede agregar un nuevo rol al sistema.

Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Roles.
Flujo Normal	6. El sistema muestra un campo de texto para ingresar el nombre de rol. 7. El actor ingresa el nombre del rol. 8. El sistema comprueba la validez del dato ingresado. 9. El actor envía el nuevo dato. 10. El sistema almacena el rol. 11. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez del dato ingresado, si el dato ingresado está correcto el sistema habilita el botón de guardar.
Pos-condiciones	El nuevo rol ha sido almacenado en el sistema.

Tabla 0.14 Especificación CU Agregar Rol

Caso de Uso	Listar Rol
Descripción	En este caso de uso el actor puede ver una lista de roles dados de alta en el sistema.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Roles.
Flujo Normal	1. El sistema muestra una lista con los roles dados de alta en el sistema.
Flujo Alternativo	1.A Si no existe roles registrados muestra un mensaje indicando que no encontró resultados.
Pos-condiciones	Lista con los roles dados de alta.

Tabla 0.15 Especificación CU Listar Rol

Caso de Uso	Modificar Rol
Descripción	En este caso de uso el actor puede cambiar el nombre de un rol.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Roles.
Flujo Normal	1. El actor presiona en el botón de modificar rol.

	2. El sistema muestra un campo de texto para ingresar el nuevo nombre de rol. 3. El actor ingresa el nuevo nombre del rol. 4. El sistema comprueba la validez del dato ingresado. 5. El actor envía el nuevo dato. 6. El sistema actualiza el nombre del rol. 7. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez del dato ingresado, si el dato ingresado está correcto el sistema habilita el botón de guardar.
Pos-condiciones	El nombre del rol ha sido cambiado.

Tabla 0.16 Especificación CU Modificar Rol

Caso de Uso	Eliminar Rol
Descripción	En este caso de uso el actor puede dar de baja un rol del sistema.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Roles.
Flujo Normal	1. El actor presiona en el botón de eliminar rol. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta la baja del rol. 4. El sistema da de baja el rol. 5. El sistema muestra un mensaje de éxito.
Flujo Alternativo	2.A El sistema muestra un dialogo de confirmación, si el actor presiona CANCELAR, la operación se cancela. 4.B Si ocurre un error al dar de baja al rol el sistema muestra un mensaje de error.
Pos-condiciones	El rol ha sido dado de baja del sistema.

Tabla 0.17 Especificación CU Eliminar Rol

Caso de Uso	Listar Procesos
Descripción	En este caso de uso el actor puede listar todos los procesos del sistema identificando a que procesos tiene permiso y a cuales no un determinado rol.
Actores	Administrador

Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de procesos de un rol.
Flujo Normal	1. El sistema muestra los procesos asignados y los no asignados.
Flujo Alternativo	
Pos-condiciones	Ninguna.

Tabla 0.18 Especificación CU Listar Procesos

Caso de Uso	Asignar Proceso
Descripción	En este caso de uso el actor puede asignar un proceso a un determinado rol.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de procesos de un rol.
Flujo Normal	1. El sistema muestra los procesos asignados y no asignados. 2. El actor marca un checkbox desmarcado. 3. El sistema asigna el proceso al rol. 4. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A Si ocurre un error al asignar el proceso, se desmarca el checkbox determinado y muestra un mensaje de error.
Pos-condiciones	Se asigna un proceso al rol.

Tabla 0.19 Especificación CU Asignar Proceso

Caso de Uso	Quitar Proceso
Descripción	En este caso de uso el actor puede quitar un proceso a un determinado rol.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de procesos de un rol.
Flujo Normal	1. El sistema muestra los procesos asignados y no asignados. 2. El actor desmarca un checkbox marcado. 3. El sistema quita el proceso del rol. 4. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A Si ocurre un error al asignar el proceso, se marca el checkbox determinado y muestra un mensaje de error.
Pos-condiciones	Se quita un proceso de un rol.

Tabla 0.20 Especificación CU Quitar Proceso

I.1.2.2.3.6.2.4 Especificación de Caso de Uso Generar Reportes

Caso de Uso	Generar Estadísticas de Ingreso
Descripción	En este caso de uso el actor puede visualizar de manera gráfica una estadística anual del ingreso de peluquerías y atenciones clínicas expresado en bolivianos.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Estadísticas de ingreso.
Flujo Normal	1. El usuario ingresa el año deseado en una caja de texto. 2. El sistema muestra una gráfica con los datos de peluquerías y atenciones clínicas.
Flujo Alternativo	
Pos-condiciones	Datos Listo para imprimir

Tabla 0.21 Especificación CU Generar Estadísticas de Ingreso

Caso de Uso	Generar Estadísticas de aplicación móvil
Descripción	En este caso de uso el actor puede ver de manera gráfica un reporte anual de los clientes que usan la aplicación móvil en dichos meses.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de reportes de visitas.
Flujo Normal	1. El usuario ingresa el año deseado en una caja de texto. 2. El sistema muestra una gráfica con los datos de visitas.
Flujo Alternativo	
Pos-condiciones	Datos Listos para imprimir

Tabla 0.22 Generar Estadísticas de aplicación móvil

Caso de Uso	Imprimir Datos
Descripción	En este caso de uso el actor puede ver de manera más detallada en forma de tablas los datos de la estadística deseada.
Actores	Administrador

Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección del reporte deseado
Flujo Normal	1. El actor presiona el botón vista de impresión. 2. El sistema muestra una pantalla lista para impresión. 3. El actor presiona el botón imprimir. 4. El navegador muestra una vista previa de impresión. 5. El actor presiona el botón de confirmar impresión. 6. El navegador imprime el documento.
Flujo Alternativo	6.A Si el navegador no encuentra una impresora le informa al usuario.
Pos-condiciones	Los datos han sido impresos.

Tabla 0.23 Especificación CU Imprimir Datos

I.1.2.2.3.6.2.5 Especificación de Caso de Uso Gestionar Noticias

Caso de Uso	Listar Noticias
Descripción	En este caso de uso el actor puede ver las noticias que están registradas en el sistema. Esta puede ser vista en la
Actores	Administrador, Cliente, Medico, Peluquero
Precondiciones	Solo el administrador es necesario que este autenticado en el sistema. El actor debe ingresar a la sección de noticias.
Flujo Normal	1. El sistema muestra las noticias.
Flujo Alternativo	
Pos-condiciones	ninguna

Tabla 0.24 Especificación CU Listar Noticias

Caso de Uso	Agregar Noticia
Descripción	En este caso de uso el actor puede agregar una nueva noticia al sistema.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de noticias.
Flujo Normal	1. El actor presiona el botón de agregar noticia. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado

	6. El actor envía los datos. 7. El sistema almacena la noticia en el sistema. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar.
Pos-condiciones	La nueva noticia se ha almacenado en el sistema. Se ha enviado una notificación PUSH a todos los clientes que tengan instalada la aplicación móvil.

Tabla 0.25 Especificación CU Agregar Noticias

Caso de Uso	Eliminar Noticia
Descripción	En este caso de uso el actor puede dar de baja una noticia del sistema.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de noticias.
Flujo Normal	1. El actor presiona el botón para eliminar noticia. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta. 4. El sistema da de baja a la noticia. 5. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A El actor no acepta, el sistema cancela a operación y no realiza ningún cambio.
Pos-condiciones	La noticia ha sido dada de baja en el sistema.

Tabla 0.26 Especificación CU Eliminar Noticia

Caso de Uso	Ver Comentarios
Descripción	En este caso de uso el actor puede ver los comentarios que hicieron a una Noticia.
Actores	Administrador, Cliente, Medico, Peluquero
Precondiciones	El actor debe estar autenticado en el sistema a excepción del cliente. El actor debe ingresar a la sección de noticias.
Flujo Normal	1. El sistema muestra una lista de comentarios.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.27 Especificación CU Ver comentarios

Caso de Uso	Escribir Comentario
Descripción	En este caso de uso el actor puede escribir un comentario para un determinado contenido publicado.
Actores	Administrador, Cliente, Medico, Peluquero
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de noticias.
Flujo Normal	1. El sistema muestra una caja de texto para ingresar el comentario. 2. El actor escribe el comentario. 3. El sistema comprueba la validez de los datos. 4. El sistema habilita el botón de envió. 5. El actor envía los datos. 6. El sistema agrega el comentario. 7. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar.
Pos-condiciones	El comentario se ha almacenado en el sistema.

Tabla 0.28 Especificación CU Escribir comentario

Caso de Uso	Eliminar Comentario
Descripción	En este caso de uso el actor puede eliminar un comentario de un usuario determinado.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de novedades e ingresar dentro de una categoría y ver los comentarios.
Flujo Normal	1. El actor presiona el botón para eliminar comentario. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta. 4. El sistema borra el comentario. 5. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A El actor no acepta, el sistema cancela a operación y no realiza ningún cambio.
Pos-condiciones	El comentario ha sido borrado del sistema.

Tabla 0.29 Especificación CU Eliminar Comentario

I.1.2.2.3.6.2.6 Especificación de Caso de Uso Gestionar Medicamentos

Caso de Uso	Listar Medicamentos
Descripción	En este caso de uso el actor puede ver los medicamentos que reciben las mascotas.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de medicamentos.
Flujo Normal	1. El sistema muestra una lista de medicamentos.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.30 Especificación CU Listar Medicamentos

Caso de Uso	Agregar Medicamento
Descripción	En este caso de uso el actor puede agregar un nuevo medicamento al sistema.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de medicamentos
Flujo Normal	1. El actor presiona el botón que es para agregar medicamento. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado 6. El actor envía los datos. 7. El sistema almacena el producto. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar.
Pos-condiciones	El medicamento se ha almacenado en el sistema.

Tabla 0.31 Especificación CU Agregar Medicamento

Caso de Uso	Modificar Medicamento
-------------	-----------------------

Descripción	En este caso de uso el actor puede modificar los datos de un medicamento.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de medicamentos.
Flujo Normal	1. El actor presiona sobre el botón que es para modificar medicamento. 2. El sistema muestra un formulario con los datos actuales. 3. El actor ingresa los nuevos datos. 4. El sistema comprueba la validez de los datos. 5. El sistema habilita el botón de guardado. 6. El actor envía los datos. 7. El sistema almacena los nuevos datos. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar.
Pos-condiciones	El medicamento se ha modificado.

Tabla 0.32 Especificación CU Modificar Medicamento

Caso de Uso	Eliminar Medicamento
Descripción	En este caso de uso el actor puede dar de baja a un medicamento del sistema.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de medicamentos.
Flujo Normal	1. El actor presiona el botón para eliminar medicamento. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta. 4. El sistema da de baja al medicamento. 5. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A El actor no acepta, el sistema cancela a operación y no realiza ningún cambio.
Pos-condiciones	El medicamento ha sido dado de baja del sistema.

Tabla 0.33 Especificación CU Eliminar Medicamento

Caso de Uso	Buscar Medicamentos
--------------------	----------------------------

Descripción	En este caso de uso el actor puede buscar un medicamento en el sistema.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de medicamento
Flujo Normal	1. El sistema muestra un campo de texto para la búsqueda. 2. El actor ingresa los términos de búsqueda del medicamento. (Nombre) 3. El sistema devuelve una lista con los resultados que corresponden al término de búsqueda.
Flujo Alternativo	3.A Si el sistema no encuentra resultados, devuelve un mensaje al usuario indicando que no encontró resultados.
Pos-condiciones	

Tabla 0.34 Especificación CU Buscar Medicamento

I.1.2.2.3.6.2.7 Especificación de Caso de Uso Gestionar Clientes

Caso de Uso	Listar Clientes
Descripción	En este caso de uso el actor puede ver los clientes registrados en la clínica veterinaria.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de clientes.
Flujo Normal	1. El sistema muestra una lista de clientes.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.35 Especificación CU Listar Clientes

Caso de Uso	Agregar Cliente
Descripción	En este caso de uso el actor puede agregar un nuevo cliente al sistema.
Actores	Administrador, Medico, Peluquero
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de productos

Flujo Normal	1. El actor presiona el botón que es para agregar cliente. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado 6. El actor envía los datos. 7. El sistema almacena el cliente. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar.
Pos-condiciones	El producto se ha almacenado en el sistema.

Tabla 0.36 Especificación CU Agregar cliente

Caso de Uso	Modificar Cliente
Descripción	En este caso de uso el actor puede modificar los datos de un cliente.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de clientes
Flujo Normal	1. El actor presiona sobre el botón que es para modificar cliente. 2. El sistema muestra un formulario con los datos actuales. 3. El actor ingresa los nuevos datos. 4. El sistema comprueba la validez de los datos. 5. El sistema habilita el botón de guardado. 6. El actor envía los datos. 7. El sistema almacena los nuevos datos. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar.
Pos-condiciones	El cliente se ha modificado.

Tabla 0.37 Especificación CU Modificar Cliente

Caso de Uso	Eliminar Cliente
Descripción	En este caso de uso el actor puede dar de baja a un cliente del sistema.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de clientes
Flujo Normal	1. El actor presiona el botón para eliminar cliente. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta.

	4. El sistema da de baja al cliente. 5. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A El actor no acepta, el sistema cancela a operación y no realiza ningún cambio.
Pos-condiciones	El cliente ha sido dado de baja del sistema.

Tabla 0.38 Especificación CU Eliminar Cliente

Caso de Uso	Buscar Cliente
Descripción	En este caso de uso el actor puede buscar un cliente en el sistema.
Actores	Administrador, Medico, Peluquero
Precondiciones	El actor debe estar autenticado en el sistema.
Flujo Normal	1. El sistema muestra un campo de texto para la búsqueda. 2. El actor ingresa los términos de búsqueda del cliente. (CI, Nombre, Apellido paterno o materno) 3. El sistema devuelve una lista con los resultados que corresponden al término de búsqueda.
Flujo Alternativo	3.A Si el sistema no encuentra resultados, devuelve un mensaje al usuario indicando que no encontró resultados.
Pos-condiciones	

Tabla 0.39 Especificación CU Buscar cliente

Caso de Uso	Agregar Clave
Descripción	En este caso de uso el actor puede agregar una clave al cliente para que pueda acceder al sistema mediante la aplicación móvil.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de clientes. El cliente no debe contar ya con una clave de acceso y debe tener registrado un número de celular.
Flujo Normal	1. El actor presiona el botón que es para agregar una clave al cliente. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado 6. El actor envía los datos. 7. El sistema almacena la nueva clave. 8. El sistema muestra un mensaje de éxito.

Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar.
Pos-condiciones	La clave de cliente se ha almacenado en el sistema.

Tabla 0.40 Especificación CU Agregar Clave

Caso de Uso	Cambiar Clave
Descripción	En este caso de uso el actor puede modificar la clave de acceso de un cliente.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de clientes. El cliente debe contar con una clave de acceso.
Flujo Normal	1. El actor presiona sobre el botón que es para modificar la clave del cliente. 2. El sistema muestra un formulario para que ingrese la nueva clave. 3. El actor ingresa la nueva clave. 4. El sistema comprueba la validez de la nueva clave. 5. El sistema habilita el botón de guardado. 6. El actor envía la nueva clave. 7. El sistema almacena la nueva clave. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de la clave, si no es correcta el sistema no habilita el botón de guardar.
Pos-condiciones	La clave de acceso del cliente se ha modificado.

Tabla 0.41 Especificación CU Cambiar Clave

Caso de Uso	Eliminar Clave
Descripción	En este caso de uso el actor puede dar de baja el permiso que tiene un cliente para acceder al sistema.
Actores	Administrador, Medico, Secretaria
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de clientes. El cliente debe contar con una clave de acceso.
Flujo Normal	1. El actor presiona el botón para eliminar la clave del cliente. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta. 4. El sistema da de baja el permiso de acceso al cliente.

	5. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A El actor no acepta, el sistema cancela a operación y no realiza ningún cambio.
Pos-condiciones	Se ha dado de baja el permiso para que el cliente acceda al sistema desde la aplicación móvil.

Tabla 0.42 Especificación CU Eliminar Clave

I.1.2.2.3.6.2.8 Especificación de Caso de Uso Gestionar Mascotas

Caso de Uso	Buscar Mascota
Descripción	En este caso de uso el actor puede buscar una mascota en el sistema.
Actores	Administrador, Medico, Peluquero
Precondiciones	El actor debe estar autenticado en el sistema.
Flujo Normal	1. El sistema muestra un campo de texto para la búsqueda. 2. El actor ingresa los términos de búsqueda de la mascota. (Código, Nombre) 3. El sistema devuelve una lista con los resultados que corresponden al término de búsqueda.
Flujo Alternativo	3.A Si el sistema no encuentra resultados, devuelve un mensaje al usuario indicando que no encontró resultados.
Pos-condiciones	

Tabla 0.43 Especificación CU Buscar Mascotas

Caso de Uso	Listar Mascotas
Descripción	En este caso de uso el actor puede ver las mascotas registradas en la clínica veterinaria.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de mascotas.
Flujo Normal	1. El sistema muestra una lista de mascotas.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.44 Especificación CU Listar Mascotas

Caso de Uso	Ver perfil Mascota
Descripción	En este caso de uso el actor puede ver los datos de una mascota registrada en el sistema.
Actores	Administrador, Medico, Cliente
Precondiciones	El actor debe estar autenticado en el sistema a excepción del cliente no es necesario. El actor debe ingresar a la sección de mascotas. El actor debe seleccionar una mascota de la lista o utilizar el escáner de Código QR.
Flujo Normal	1. El sistema muestra los datos básicos de la mascota y una lista de las próximas visitas.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.45 Especificación CU Ver Perfil

Caso de Uso	Agregar Mascota
Descripción	En este caso de uso el actor puede agregar una mascota al sistema.
Actores	Administrador, Medico, Peluquero
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de mascotas
Flujo Normal	1. El actor presiona el botón que es para agregar mascota. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado 6. El actor envía los datos. 7. El sistema almacena la mascota. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar.
Pos-condiciones	La mascota se ha almacenado en el sistema.

Tabla 0.46 Especificación CU Agregar Mascota

Caso de Uso	Modificar Mascota
--------------------	--------------------------

Descripción	En este caso de uso el actor puede modificar los datos de un cliente.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de mascotas. El actor debe seleccionar una mascota de la lista o utilizar el escáner de Código QR.
Flujo Normal	1. El actor presiona sobre una mascota de la lista o mediante el escáner de código QR 2. El sistema muestra un formulario con los datos actuales. 3. El actor ingresa los nuevos datos. 4. El sistema comprueba la validez de los datos. 5. El sistema habilita el botón de guardado. 6. El actor envía los datos. 7. El sistema almacena los nuevos datos. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar.
Pos-condiciones	La mascota se ha modificado.

Tabla 0.47 Especificación CU Modificar Mascota

Caso de Uso	Eliminar Mascota
Descripción	En este caso de uso el actor puede dar de baja a la mascota del sistema.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de mascotas.
Flujo Normal	1. El actor presiona el botón para eliminar mascota. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta. 4. El sistema da de baja a la mascota. 5. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A El actor no acepta, el sistema cancela a operación y no realiza ningún cambio.
Pos-condiciones	Se ha dado de baja a la mascota del sistema.

Tabla 0.48 Especificación CU Eliminar Mascota

Caso de Uso	Escanear Código QR
--------------------	---------------------------

Descripción	En este caso de uso el actor puede ver el perfil mediante un dispositivo lector de código QR o la aplicación móvil lectora de código QR
Actores	Administrador, Medico, Peluquero, Cliente
Precondiciones	El actor debe estar autenticado en el sistema a excepción del cliente.
Flujo Normal – Aplicación Móvil (Medico, Administrador, Peluquero)	1. El actor abre el lector QR 2. El sistema identifica el código. 3. El sistema muestra una notificación en el computador. 4. El Actor abre la notificación. 5. El sistema muestra el perfil deseado de la mascota.
Flujo Normal – Aplicación Móvil (Cliente)	6. El actor abre el lector QR de la aplicación. 7. El sistema identifica el código. 8. El sistema muestra el perfil clínico de la mascota.
Flujo Normal – Dispositivo lector de Código (Medico, Administrador)	9. El Actor ingresa a la sección escáner QR. 10. El sistema lee el código. 11. El sistema ingresa al perfil de la mascota.
Flujo Alternativo	2.A Si el sistema no encuentra el código de la mascota, indica que no encontró resultados
Pos-condiciones	El actor ve la información de la mascota.

Tabla 0.49 Especificación CU Escanear Código QR

Caso de Uso	Listar Calendario
Descripción	En este caso de uso el actor puede ver las próximas visitas de la mascota.
Actores	Administrador, Medico, Cliente
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ver el perfil de la mascota.
Flujo Normal	1. El sistema muestra una lista de todas las visitas de la mascota.
Flujo Alternativo	
Pues-condiciones	

Tabla 0.50 Especificación CU Listar Calendario

Caso de Uso	Agregar Calendario
Descripción	En este caso de uso el actor puede agregar una próxima visita de la mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de mascotas
Flujo Normal	1. El actor presiona el botón que es para agregar una próxima visita. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado 6. El actor envía los datos. 7. El sistema almacena la próxima visita de la mascota y programa una cita. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar.
Pos-condiciones	La mascota se ha almacenado en el sistema. Se ha agregado una cita.

Tabla 0.51 Especificación CU Agregar Calendario

Caso de Uso	Eliminar Calendario
Descripción	En este caso de uso el actor puede eliminar la próxima vista a la clínica de una mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de mascotas. La mascota aun no debe haber asistido a la visita y debe ser antes del día programado.
Flujo Normal	1. El actor presiona el botón para eliminar próxima visita. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta. 4. El sistema da de baja la próxima visita y también da de baja la cita programada. 5. El sistema muestra un mensaje de éxito.

Flujo Alternativo	3.A El actor no acepta, el sistema cancela a operación y no realiza ningún cambio.
Pos-condiciones	Se ha dado de baja la próxima visita. Se ha eliminado la cita.

Tabla 0.52 Especificación CU Eliminar Calendario

I.1.2.2.3.6.2.9 Especificación de Caso de Uso Gestionar Peluquerías

Descripción de Casos de Uso

Caso de Uso	Listar Peluquerías
Descripción	En este caso de uso el actor puede ver las peluquerías que están registradas en el sistema.
Actores	Administrador, Medico, Cliente
Precondiciones	El actor debe estar autenticado en el sistema. El actor (Administrador, Medico, Secretaria) debe entrar a la sección de peluquerías. El cliente debe ver el perfil de la mascota.
Flujo Normal	1. El sistema muestra una lista de las peluquerías.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.53 Especificación CU Listar Peluquerías

Caso de Uso	Buscar Peluquerías
Descripción	En este caso de uso el actor puede ver las peluquerías que se realizó a una mascota.
Actores	Administrador, Medico, Peluquero
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de peluquerías.
Flujo Normal	1. El sistema muestra un campo de texto para la búsqueda. 2. El actor ingresa el nombre o el código de la mascota. 3. El sistema devuelve una lista con las peluquerías de la mascota.
Flujo Alternativo	

Pos-condiciones	
------------------------	--

Tabla 0.54 Especificación CU Buscar Peluquerías

Caso de Uso	Agregar Peluquería
Descripción	En este caso de uso el actor puede agregar una peluquería.
Actores	Administrador, Medico, Peluquero
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de peluquerías.
Flujo Normal	1. El actor debe buscar y seleccionar una macota. 2. El actor presiona el botón que es para agregar una nueva peluquería. 3. El sistema muestra un formulario. 4. El actor ingresa los datos en el formulario 5. El sistema comprueba la validez de los datos 6. El sistema habilita el botón de guardado 7. El actor envía los datos. 8. El sistema almacena la peluquería. 9. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar. 8.B Si no se puede almacenar el sistema muestra un mensaje de error.
Pos-condiciones	La peluquería se ha almacenado en el sistema.

Tabla 0.55 Especificación CU Agregar Peluquería

Caso de Uso	Modificar Peluquería
Descripción	En este caso de uso el actor puede modificar los datos de una peluquería.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Peluquerías.
Flujo Normal	1. El sistema muestra un formulario con los datos actuales. 2. El actor cambia los datos. 3. El sistema comprueba la validez de los datos. 4. El sistema habilita el botón de guardado. 5. El actor envía los datos. 6. El sistema cambia los datos de la peluquería. 7. El sistema muestra un mensaje de éxito.

Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar. 6.B Si no se puede modificar el sistema muestra un mensaje de error.
Pos-condiciones	La peluquería se ha modificado.

Tabla 0.56 Especificación CU Modificar Peluquería

Caso de Uso	Eliminar Peluquería
Descripción	En este caso de uso el actor puede dar de baja a una peluquería registrada en el sistema.
Actores	Administrador
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de peluquerías.
Flujo Normal	1. El actor presiona el botón para eliminar peluquería. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta. 4. El sistema da de baja la peluquería. 5. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A El actor no acepta, el sistema cancela a operación y no realiza ningún cambio. 6.B Si no se puede eliminar el sistema muestra un mensaje de error.
Pos-condiciones	Se ha dado de baja la peluquería.

Tabla 0.57 Especificación CU Eliminar Peluquería

I.1.2.2.3.6.2.10 Especificación de Caso de Uso Gestionar Control Sanitario

Caso de Uso	Ver Control Sanitario
Descripción	En este caso de uso el actor puede ver el control sanitario de manera general de la mascota.
Actores	Cliente
Precondiciones	El actor debe ver el perfil de la mascota.
Flujo Normal	1. El sistema muestra una lista de las desparasitaciones, vacunas, revacunaciones, historial de pesos.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.58 Especificación CU Ver control Sanitario

Caso de Uso	Desparasitar Mascota
Descripción	En este caso de uso el actor puede agregar una desparasitación a la mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de control sanitario de una mascota y presionar en la pestaña de desparasitaciones.
Flujo Normal	1. El actor presiona el botón que es para agregar una nueva desparasitación. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario y si desea programa una próxima visita. 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado 6. El actor envía los datos. 7. El sistema almacena la desparasitación y si existe una próxima visita crea una cita. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar. 8.B Si no se puede almacenar el sistema muestra un mensaje de error.
Pos-condiciones	La desparasitación se ha almacenado en el sistema.

Tabla 0.59 Especificación CU Desparasitar Mascota

Caso de Uso	Ver Desparasitaciones
Descripción	En este caso de uso el actor puede ver las desparasitaciones que recibió la mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de control sanitario de una mascota y presionar en la pestaña de desparasitaciones.
Flujo Normal	1. El sistema muestra una lista de las desparasitaciones, que recibió la mascota.
Flujo Alternativo	

Pos-condiciones	
------------------------	--

Tabla 0.60 Especificación CU Ver Desparasitaciones

Caso de Uso	Vacunar Mascota
Descripción	En este caso de uso el actor puede agregar una de las primeras vacunas a la mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de control sanitario de una mascota y presionar en la pestaña de vacunaciones.
Flujo Normal	1. El actor presiona el botón que es para agregar una nueva vacunación. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario y si desea programa una próxima visita. 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado 6. El actor envía los datos. 7. El sistema almacena la vacunación y si existe una próxima visita crea una cita. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar. 8.B Si no se puede almacenar el sistema muestra un mensaje de error.
Pos-condiciones	La vacunación se ha almacenado en el sistema.

Tabla 0.61 Especificación CU Vacunar Mascota

Caso de Uso	Ver Vacunaciones
Descripción	En este caso de uso el actor puede ver las primeras vacunas que recibió la mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de control sanitario de una mascota y presionar en la pestaña de vacunaciones.
Flujo Normal	1. El sistema muestra una lista de las vacunas, que recibió la mascota.

Flujo Alternativo	
Pos-condiciones	

Tabla 0.62 Ver Vacunaciones

Caso de Uso	Injectar Anticonceptivo
Descripción	En este caso de uso el actor puede agregar una vacuna como anticonceptivos u otras que no pertenezcan a las primeras vacunas de la mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de control sanitario de una mascota y presionar en la pestaña de revacunaciones.
Flujo Normal	1. El actor presiona el botón que es para agregar una nueva revacunación. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario y si desea programa una próxima visita. 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado 6. El actor envía los datos. 7. El sistema almacena la revacunación y si existe una próxima visita crea una cita. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar. 8.B Si no se puede almacenar el sistema muestra un mensaje de error.
Pos-condiciones	La revacunación se ha almacenado en el sistema.

Tabla 0.63 Especificación CU Injectar anticonceptivo

Caso de Uso	Ver Revacunaciones
Descripción	En este caso de uso el actor puede ver las revacunaciones que recibió la mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema.

	El actor debe ingresar a la sección de control sanitario de una mascota y presionar en la pestaña de revacunaciones.
Flujo Normal	1. El sistema muestra una lista de las revacunaciones que recibió la mascota.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.64 Especificación CU Ver Revacunaciones

Caso de Uso	Imprimir Control Sanitario
Descripción	En este caso de uso el actor puede imprimir el control sanitario de la mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar al control sanitario de la mascota.
Flujo Normal	1. El actor presiona el botón para vista de impresión. 2. El sistema muestra una pantalla lista para impresión. 3. El actor presiona el botón imprimir. 4. El navegador muestra una vista previa de impresión. 5. El actor presiona el botón de confirmar impresión. 6. El navegador imprime el documento.
Flujo Alternativo	6.A Si el navegador no encuentra una impresora le informa al usuario.
Pos-condiciones	El Control Sanitario ha sido impreso.

Tabla 0.65 Especificación CU Imprimir Control Sanitario

I.1.2.2.3.6.2.11 Especificación de Caso de Uso Gestionar Ficha Clínica

Caso de Uso	Listar Atenciones
Descripción	En este caso de uso el actor puede ver las atenciones que recibió la mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de ficha clínica de atención de una mascota.
Flujo Normal	1. El sistema muestra una lista de las fichas clínicas de atenciones que recibió la mascota.

Flujo Alternativo	
Pos-condiciones	

Tabla 0.66 Especificación CU Listar Atenciones

Caso de Uso	Listar Tratamientos
Descripción	En este caso de uso el actor puede ver los tratamientos que recibió la mascota para una determinada ficha clínica de atención.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe seleccionar una determinada ficha clínica de atención.
Flujo Normal	1. El sistema muestra una lista de los tratamientos que recibió la mascota.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.67 Especificación CU Listar Tratamientos

Caso de Uso	Agregar Atención
Descripción	En este caso de uso el actor puede agregar una nueva atención clínica de la mascota ya sea por alguna enfermedad o cirugía.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la Ficha Clínica de la mascota.
Flujo Normal	1. El actor presiona el botón que es para agregar una nueva atención Clínica. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario. 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado 6. El actor envía los datos. 7. El sistema almacena la atención clínica. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar. 8.B Si no se puede almacenar el sistema muestra un mensaje de error.
Pos-condiciones	La atención se ha almacenado en el sistema.

Tabla 0.68 Especificación CU Agregar Atención

Caso de Uso	Agregar Tratamiento
Descripción	En este caso de uso el actor puede agregar un día de tratamiento a una determinada atención clínica.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la Ficha Clínica y Seleccionar una atención clínica.
Flujo Normal	1. El actor presiona el botón que es para agregar un tratamiento. 2. El sistema muestra un formulario. 3. El actor ingresa los datos en el formulario. 4. El sistema comprueba la validez de los datos 5. El sistema habilita el botón de guardado 6. El actor envía los datos. 7. El sistema almacena el tratamiento. 8. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar. 8.B Si no se puede almacenar el sistema muestra un mensaje de error.
Pos-condiciones	El tratamiento se ha almacenado en el sistema.

Tabla 0.69 Especificación CU Agregar Atención

Caso de Uso	Imprimir Ficha Clínica
Descripción	En este caso de uso el actor puede una atención clínica de la mascota.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la ficha clínica de la mascota.
Flujo Normal	1. El actor presiona el botón para vista de impresión. 2. El sistema muestra una pantalla lista para impresión. 3. El actor presiona el botón imprimir. 4. El navegador muestra una vista previa de impresión. 5. El actor presiona el botón de confirmar impresión. 6. El navegador imprime el documento.
Flujo Alternativo	6.A Si el navegador no encuentra una impresora le informa al usuario.
Pos-condiciones	La atención clínica ha sido impresa.

Tabla 0.70 Especificación CU Imprimir Ficha Clínica

I.1.2.2.3.6.2.12

Especificación de Caso de Uso Gestionar Citas

Caso de Uso	Ver Citas Confirmadas
Descripción	En este caso de uso el actor puede ver las citas que han sido confirmadas.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de citas.
Flujo Normal	1. El sistema muestra una lista de las citas correspondientes al día actual.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.71 Especificación CU Ver Citas confirmadas

Caso de Uso	Ver Solicitudes
Descripción	En este caso de uso el actor puede las solicitudes de citas que hizo un cliente por la aplicación móvil.
Actores	Administrador, Medico, Peluquero
Precondiciones	El actor debe estar autenticado en el sistema.
Flujo Normal	1. El sistema muestra un contador con la cantidad de citas pendientes para solicitud. 2. El actor presiona sobre el contador. 3. El sistema muestra un lista con las solicitudes del día actual y posteriores.
Flujo Alternativo	1.A Si no hay citas pendientes el sistema no muestra el contador.
Pos-condiciones	

Tabla 0.72 Especificación CU Ver solicitudes de Citas

Caso de Uso	Agregar Cita
Descripción	En este caso de uso el actor puede agregar una cita.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de Citas.
Flujo Normal	1. El sistema muestra los horarios que están disponibles. 2. El actor presiona el botón que es para agregar una cita. 3. El sistema muestra un formulario.

	4. El actor ingresa los datos en el formulario. 5. El sistema comprueba la validez de los datos 6. El sistema habilita el botón de guardado 7. El actor envía los datos. 8. El sistema almacena la cita y la almacena como una cita confirmada. 9. El sistema muestra un mensaje de éxito.
Flujo Alternativo	4.A El sistema comprueba la validez de los datos, si no son correctos el sistema no habilita el botón de guardar. 8.B Si no se puede almacenar el sistema muestra un mensaje de error.
Pos-condiciones	Se ha almacenado una cita y esta fue confirmada automáticamente en el sistema.

Tabla 0.73 Especificación CU Agregar Cita

Caso de Uso	Eliminar Cita
Descripción	En este caso de uso el actor puede dar de baja a una cita registrada en el sistema.
Actores	Administrador, Medico.
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de citas.
Flujo Normal	1. El actor presiona el botón para eliminar cita. 2. El sistema muestra un dialogo de confirmación. 3. El actor acepta. 4. El sistema da de baja la cita. 5. El sistema muestra un mensaje de éxito.
Flujo Alternativo	3.A El actor no acepta, el sistema cancela a operación y no realiza ningún cambio. 6.B Si no se puede eliminar el sistema muestra un mensaje de error.
Pos-condiciones	Se ha dado de baja la cita y se envía una notificación al cliente.

Tabla 0.74 Especificación CU Eliminar Cita

Caso de Uso	Recordar Citas
Descripción	En este caso de uso el actor puede hacer recuerdo a un cliente que tiene una cita.
Actores	Administrador, Medico, Secretaria, Sistema

Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de citas.
Flujo Normal	1. El sistema muestra una lista de las citas correspondientes al día actual. 2. El actor selecciona una cita que quiere hacer recuerdo y las envía. 3. El sistema envía una notificación al cliente.
Flujo Alternativo	3.A Si la aplicación móvil no tiene acceso a internet no se muestra la notificación.
Pos-condiciones	La aplicación móvil muestra una notificación al cliente.

Tabla 0.75 Especificación CU Recordar Citas

I.1.2.2.3.6.2.13 Especificación de Caso de Uso Gestionar Perfil

Caso de Uso	Ver Mis Datos
Descripción	En este caso de uso el actor puede ver los datos que le pertenecen.
Actores	Administrador, Medico
Precondiciones	El actor debe estar autenticado en el sistema. El actor debe ingresar a la sección de perfil.
Flujo Normal	2. El sistema muestra un formulario con los datos actual.
Flujo Alternativo	
Pos-condiciones	

Tabla 0.76 Especificación CU Ver Mis Datos

I.1.2.2.3.7 Especificaciones Adicionales

Este documento capturará todos los requisitos que no han sido incluidos como parte de los casos de uso y se refieren requisitos no-funcionales globales. Dichos requisitos incluyen: requisitos legales o normas, aplicación de estándares, requisitos de calidad del producto, tales como: confiabilidad, desempeño, etc., u otros requisitos de ambiente, tales como: sistema operativo, requisitos de compatibilidad, etc. Ver los requisitos funcionales del documento de ERS ANEXO 1.

I.1.2.2.3.8 Prototipos de Interfaces de Usuario

I.1.2.2.3.8.1 Introducción

La interfaz de un sistema web es el conjunto gráfico que permite la presentación y la navegación del sistema. Esto se consigue con la inclusión de elementos gráficos comunes a todo el sistema que son estándares, haciendo que los usuarios tengan completo control sobre las funcionalidades desde el momento mismo de entrar a él sin que para ello deba tener amplios conocimientos ni preparación anterior alguna. Para lograr que la interacción con el usuario sea lo más intuitiva posible, se deben utilizar recursos como la gráfica, pictogramas, estereotipos, y símbolos, todo sin afectar el funcionamiento.

I.1.2.2.3.8.1.1 Propósito

Dar a conocer a los interesados a plantilla la plantilla que registrará a la aplicación, así como también los archivos de configuración, los mismos que servirán de base para las aplicaciones futuras.

I.1.2.2.3.8.1.2 Alcance

Pantallas del sistema Neg-VET, junto con una de descripción de la funcionalidad de las mismas.

I.1.2.2.3.8.2 Prototipos de Pantalla

I.1.2.2.3.8.2.1 Ingresar al Sistema

Para entrar al sistema, todo usuario debe ser autenticado mediante el sistema de log-in.

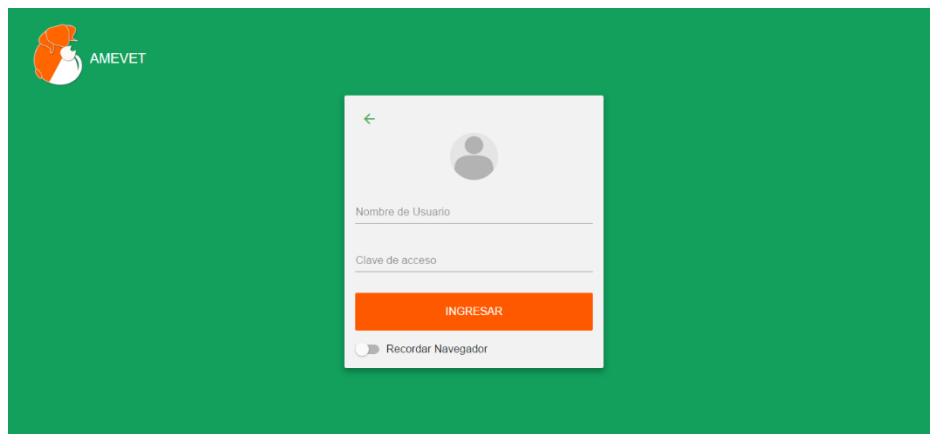


Ilustración 0.47 Pantalla Ingresar al sistema

I.1.2.2.3.8.2.2 Pantalla principal

Cuando ya está iniciada la sesión, la primera pantalla que se muestra es la de los procesos a los que puede acceder el usuario.

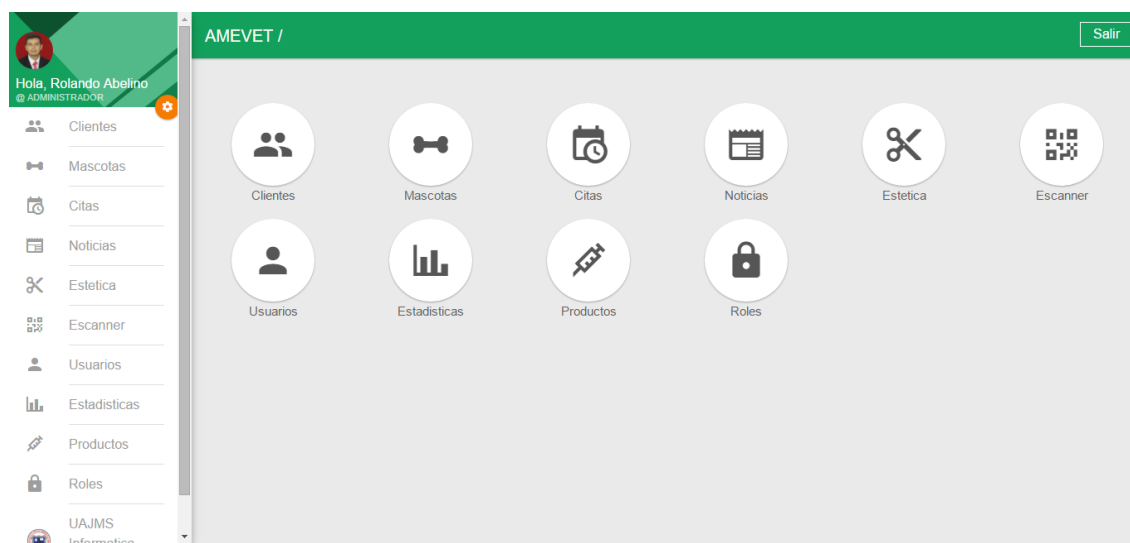


Ilustración 0.48 Pantalla Principal

I.1.2.2.3.8.2.3 Módulos del Sistema

A continuación, se detallarán los distintos módulos del sistema, vistos anteriormente en el menú lateral del sistema.

I.1.2.2.3.8.2.3.1 Pantallas Gestionar Perfil

Se muestra los datos de cada usuario con la posibilidad de cambiar sus datos personales, su foto, su clave de acceso y el color de apariencia del sistema.

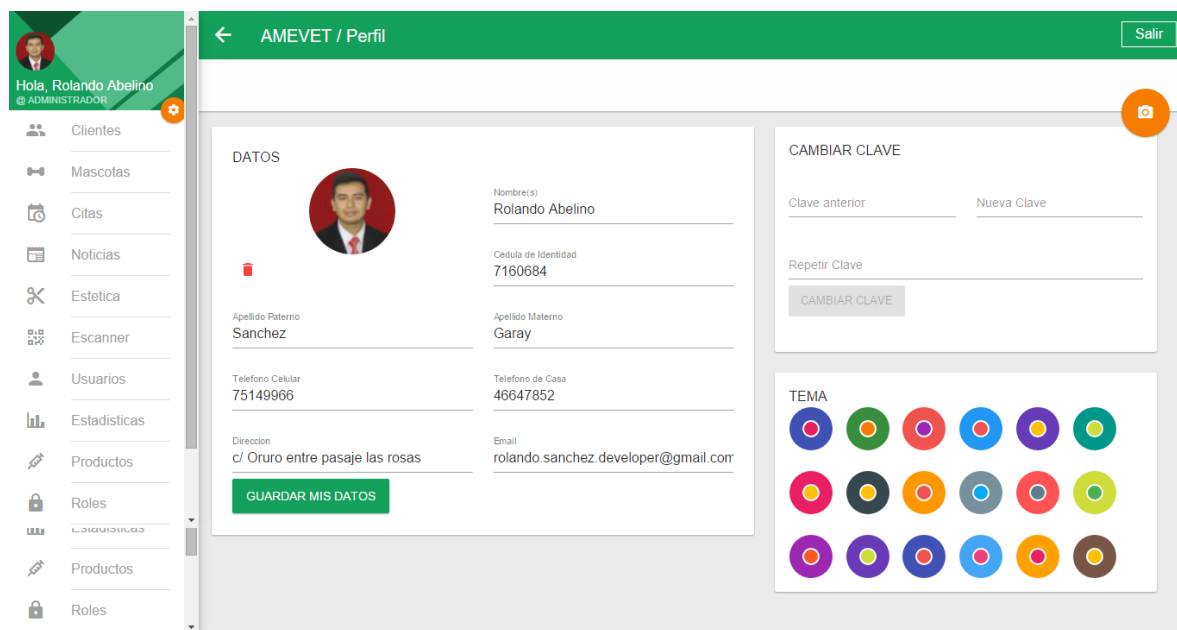


Ilustración 0.49 Pantalla Gestionar Perfil

I.1.2.2.3.8.2.3.2 Pantallas Gestionar Usuarios

Estas pantallas describen un listado de los usuarios dados de alta en el sistema, agregar un usuario, dar de baja a un usuario modificar un usuario y cambiar la clave de un usuario.

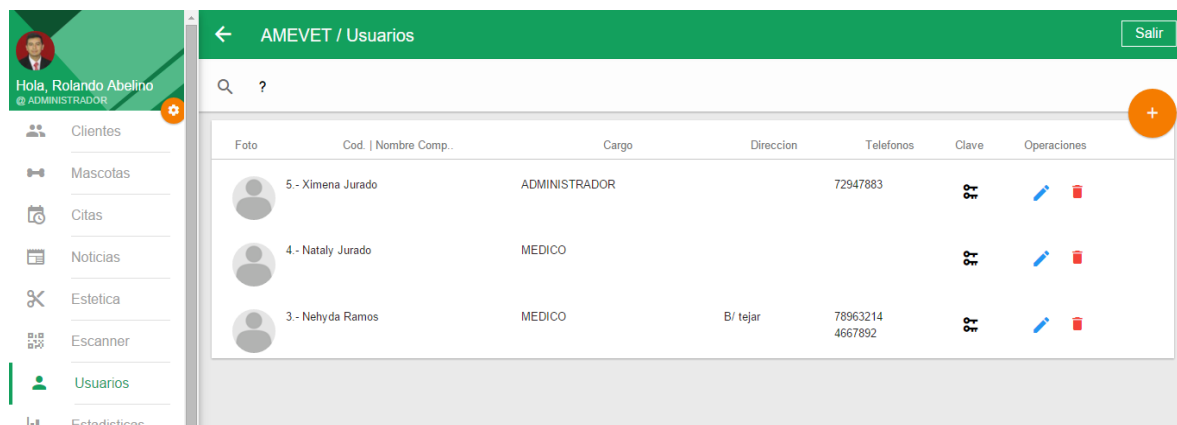


Ilustración 0.50 Pantalla Gestionar Usuarios

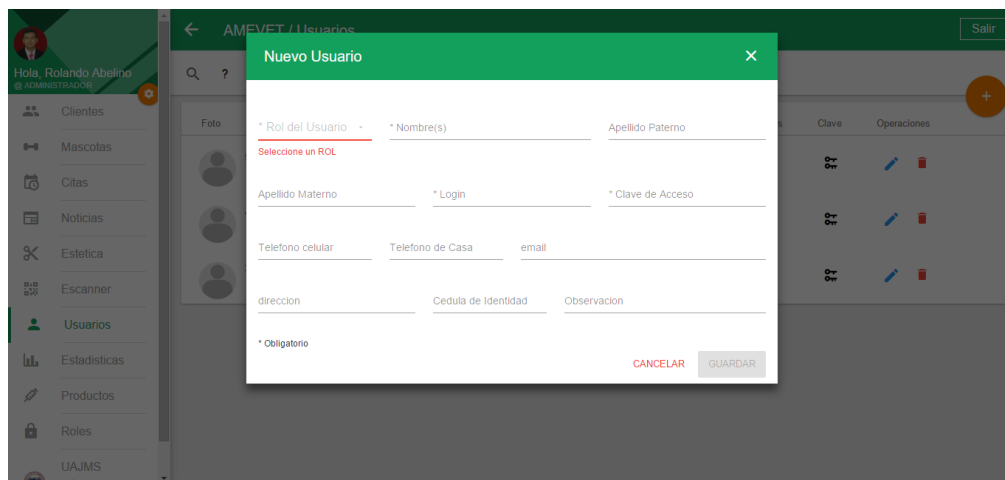


Ilustración 0.51 Pantalla Agregar Usuario

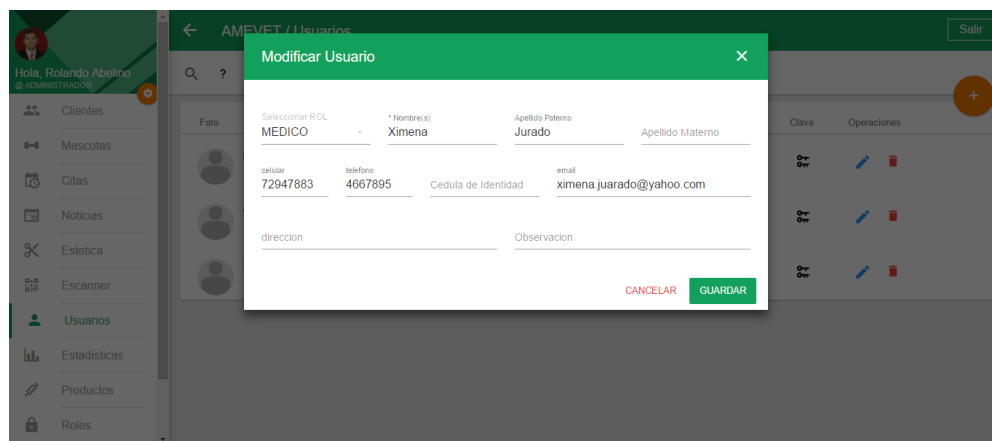


Ilustración 0.52 Pantalla Modificar Usuario

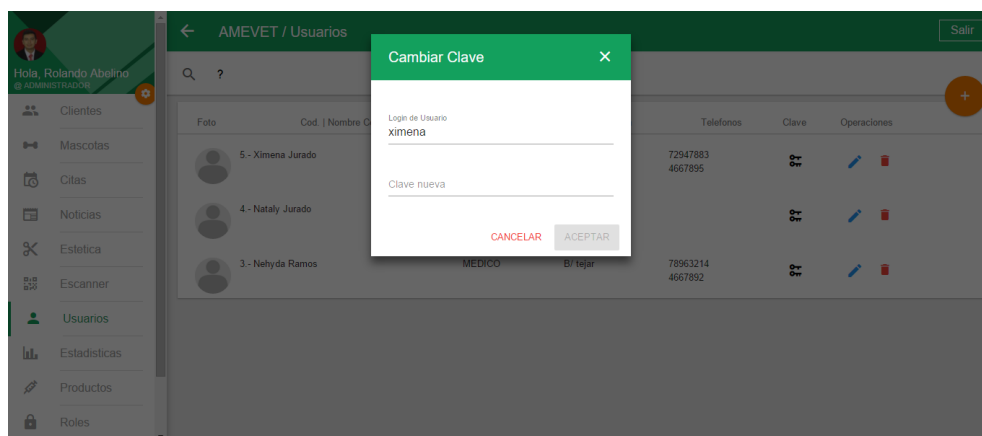


Ilustración 0.53 Pantalla Cambiar Clave de usuario

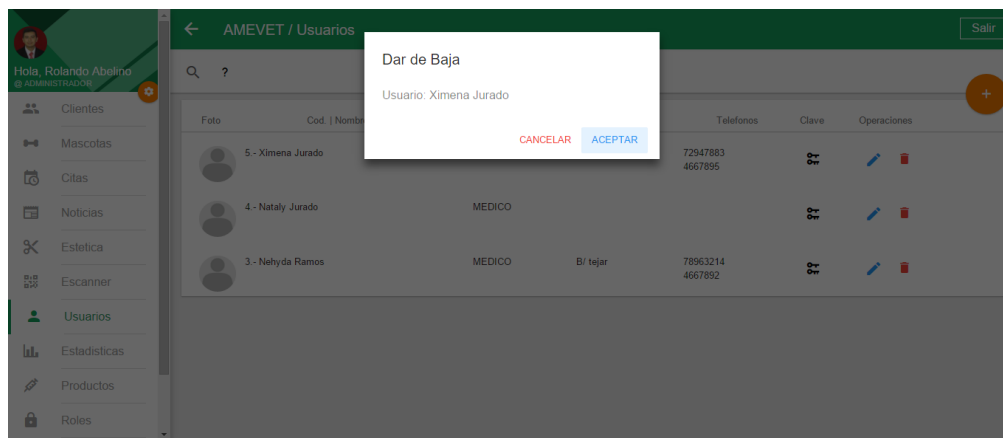


Ilustración 0.54 Pantalla Dar de Baja al Usuario

I.1.2.2.3.8.2.3.3 Pantallas Gestionar Roles

En estas pantallas se describe los procesos de los roles como un listado de los roles con sus respectivos permisos, agregar un rol, modificar un rol y eliminar un rol.

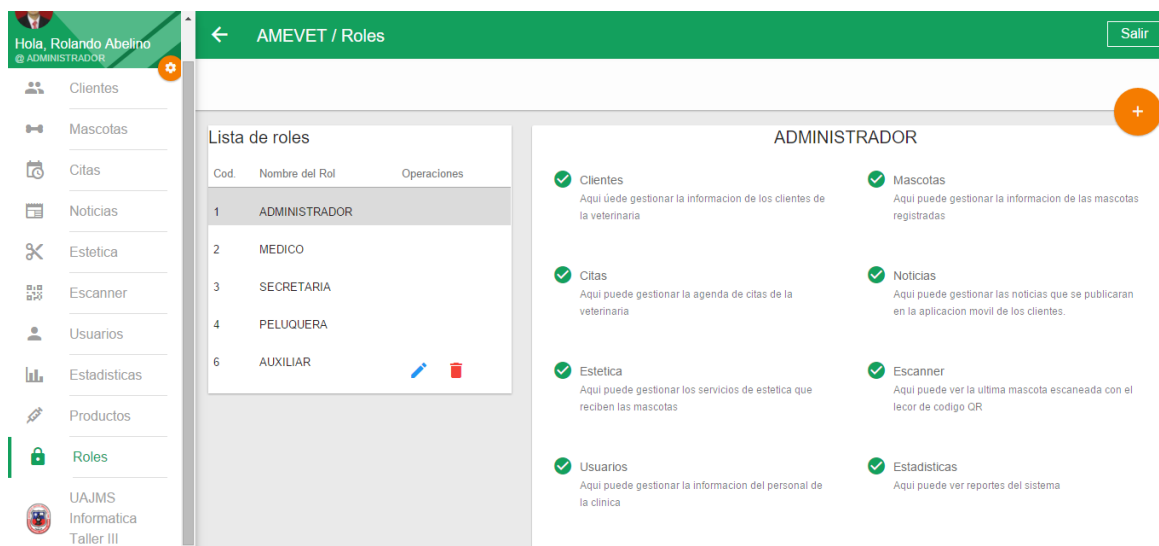


Ilustración 0.55 Pantalla Listar Roles y Procesos

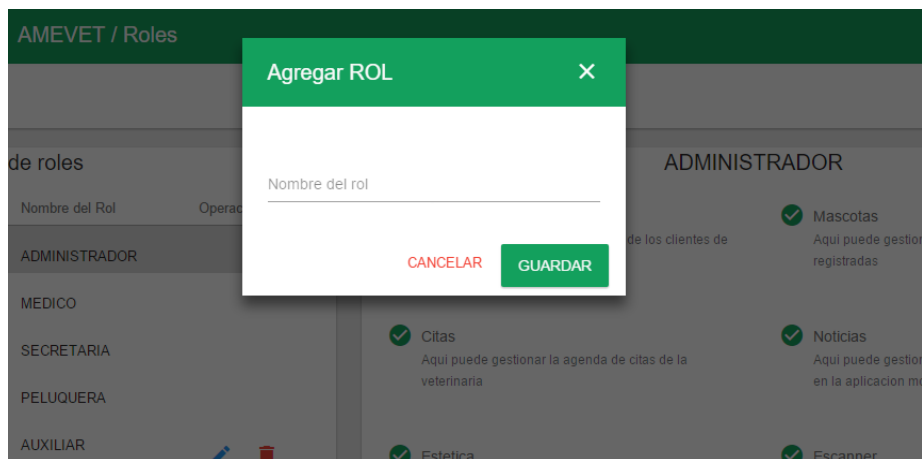


Ilustración 0.56 Pantalla Agregar Rol

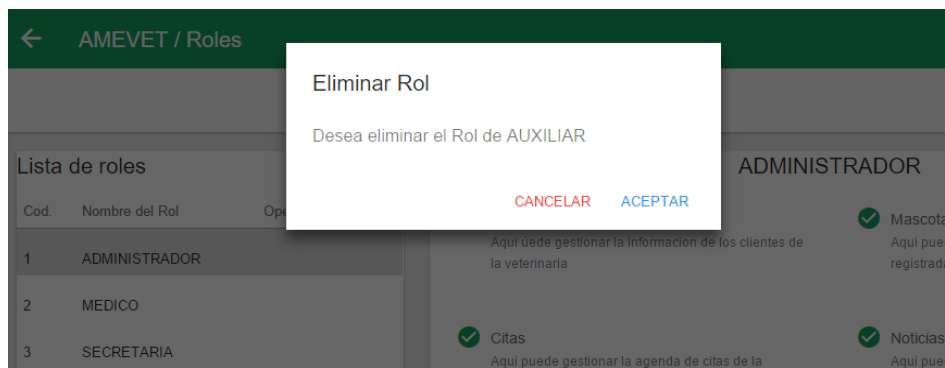


Ilustración 0.57 Pantalla Eliminar Rol

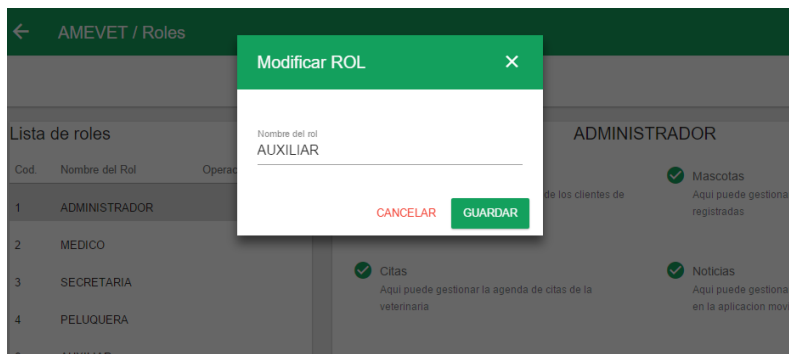


Ilustración 0.58 Pantalla Modificar Rol

I.1.2.2.3.8.2.3.4 Pantallas Gestionar Productos

Estas pantallas describen un listado de productos, agregar producto, eliminar producto y modificar un producto.

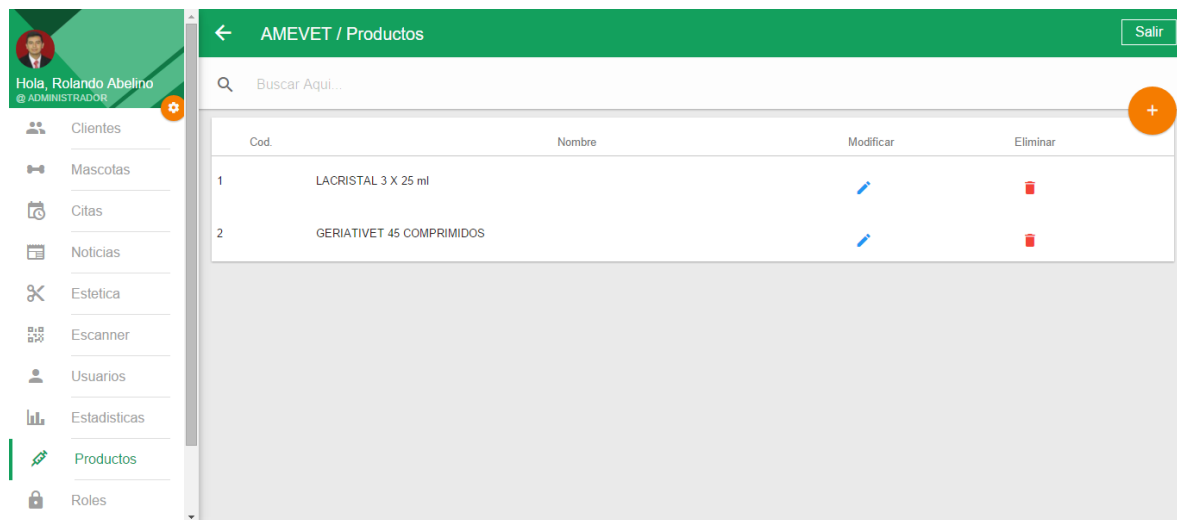


Ilustración 0.59 Pantalla Listar Productos

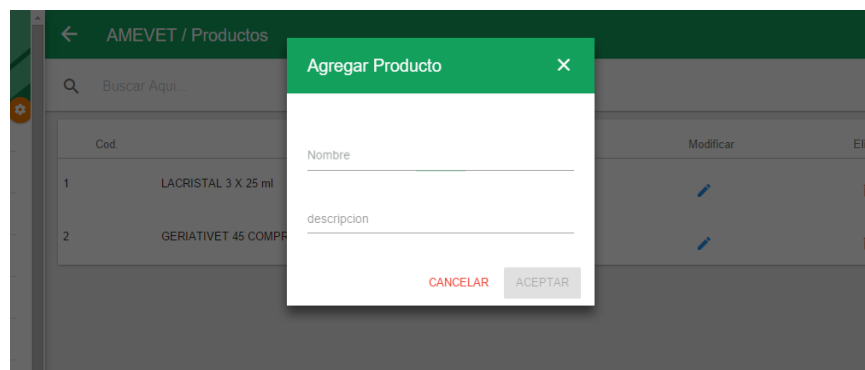


Ilustración 0.60 Pantalla Agregar Producto

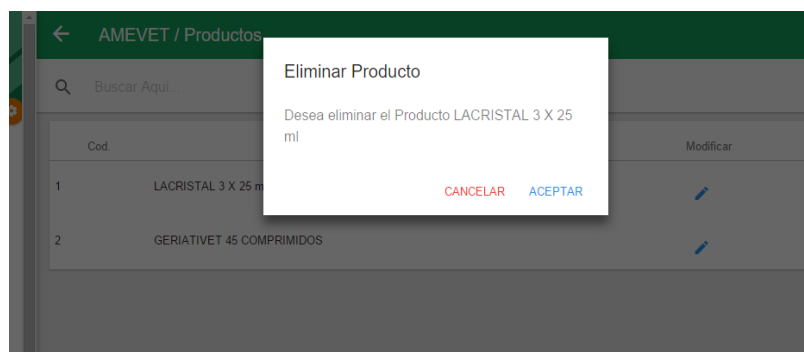


Ilustración 0.61 Pantalla Eliminar Producto

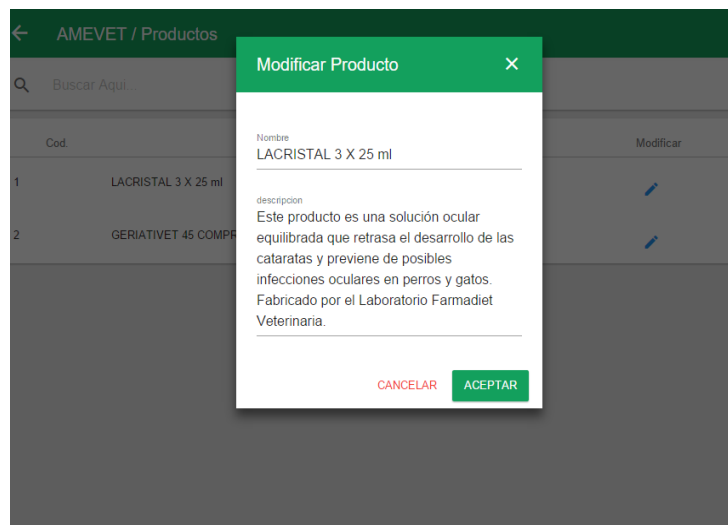


Ilustración 0.62 Pantalla Modificar Producto

I.1.2.2.3.8.2.3.5 Pantallas Gestionar Clientes

Estas pantallas describen los procesos de listado de clientes, agregar un cliente, modificar cliente eliminar cliente y también la gestión de las claves de clientes.

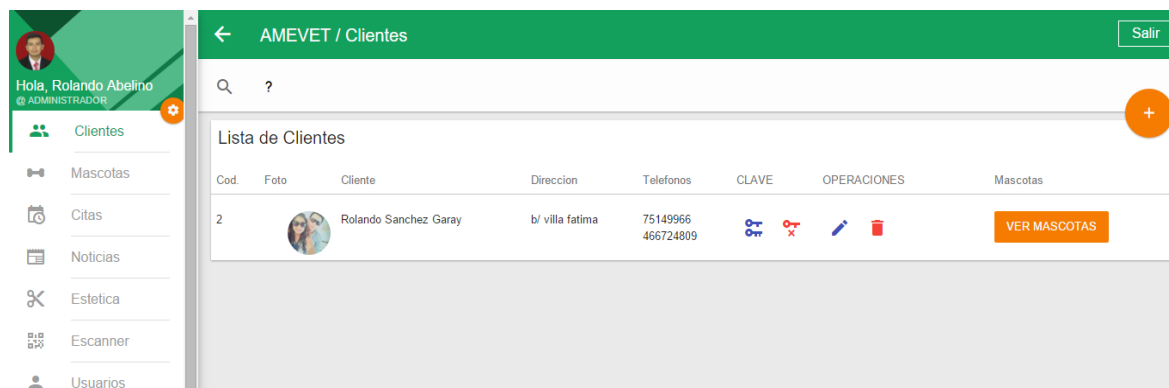


Ilustración 0.63 Pantalla Listar Clientes

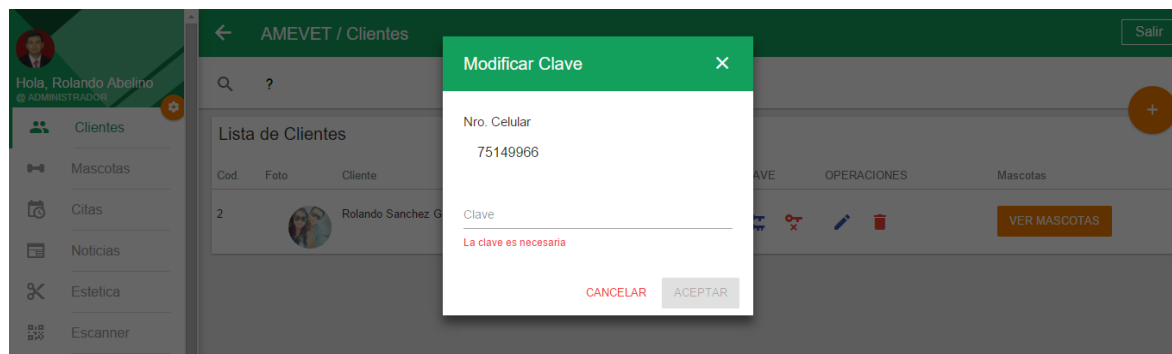


Ilustración 0.64 Pantalla modificar Clave del cliente

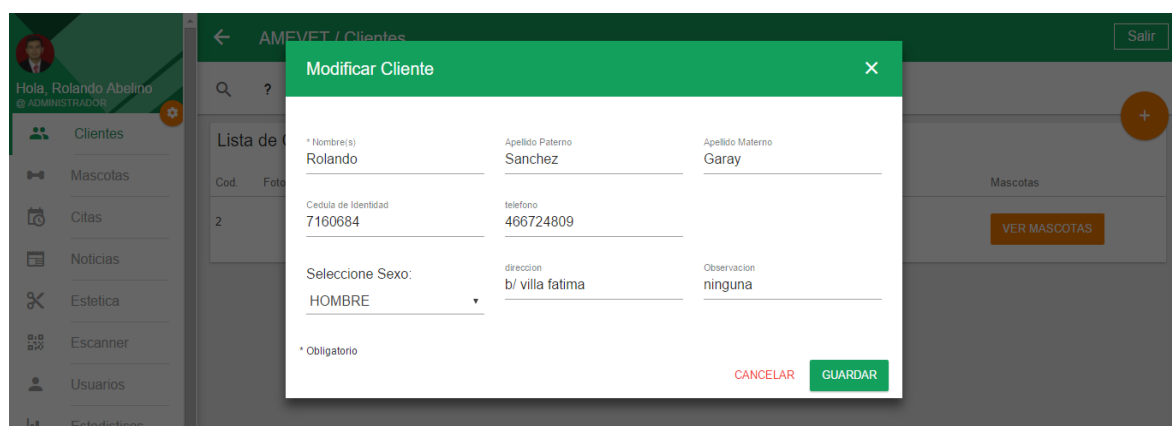


Ilustración 0.65 Pantalla modificar Cliente

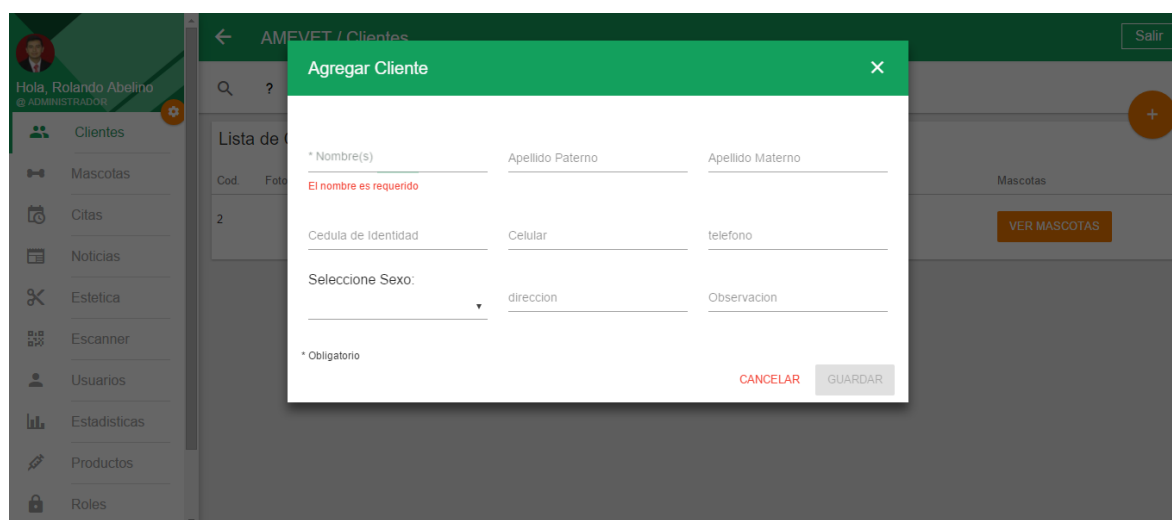


Ilustración 0.66 Pantalla Agregar Cliente

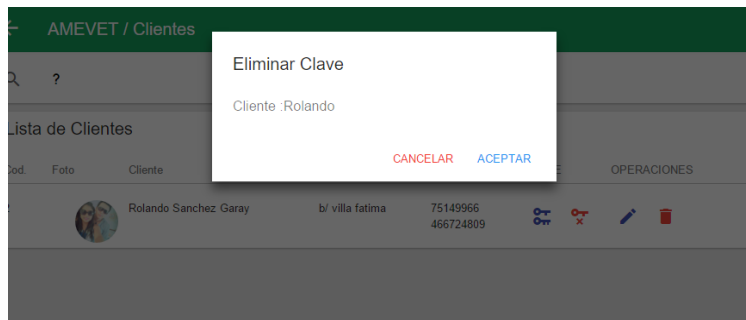


Ilustración 0.67 Pantalla Eliminar Clave

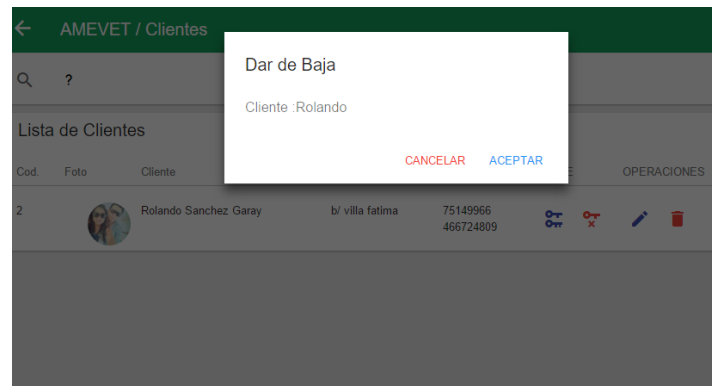


Ilustración 0.68 Pantalla Dar de Baja al Cliente

I.1.2.2.3.8.2.3.6 Pantallas Gestionar Mascotas

Estas pantallas describen El proceso de listar las mascotas, agregar una nueva mascota, ver perfil de la mascota, agregar calendario de vacunación, reportar extravió de la mascota.

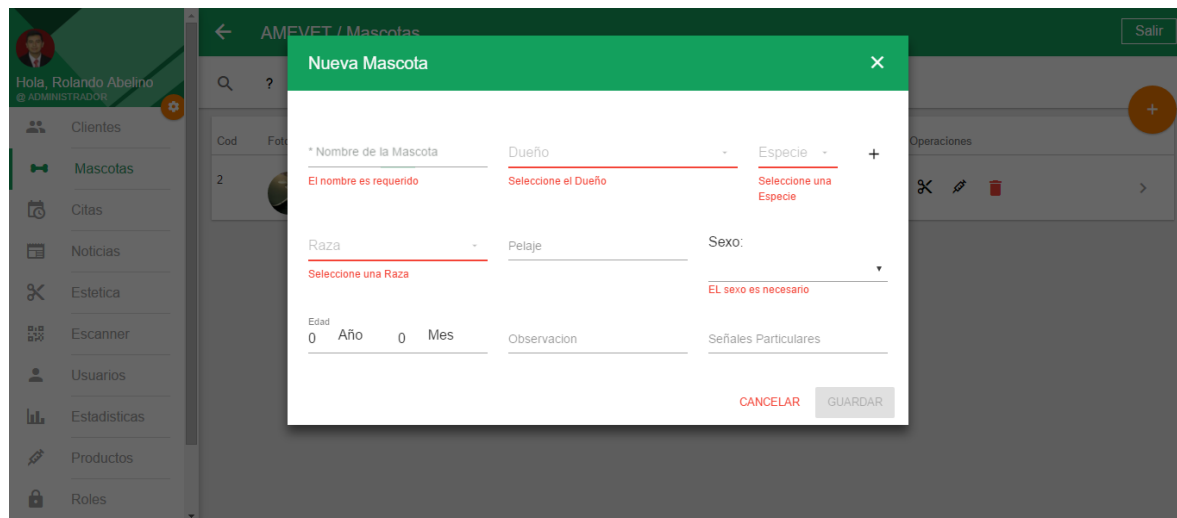


Ilustración 0.69 Pantalla Agregar Mascota

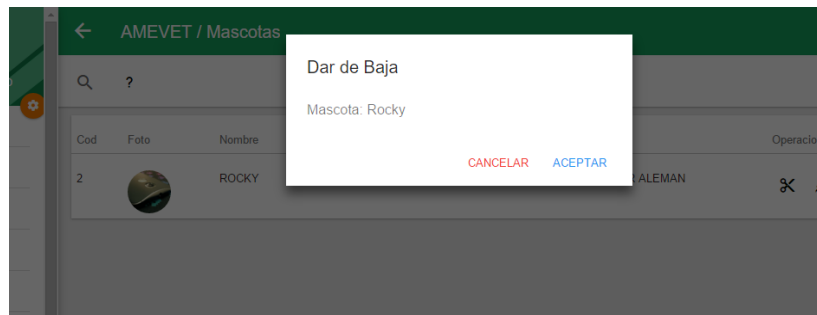


Ilustración 0.70 Pantalla Dar de Baja Mascota

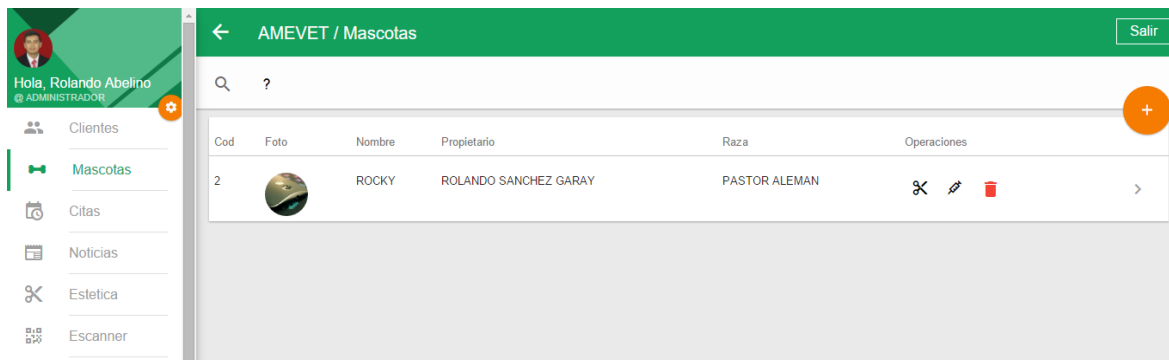


Ilustración 0.71 Pantalla Listar Mascotas

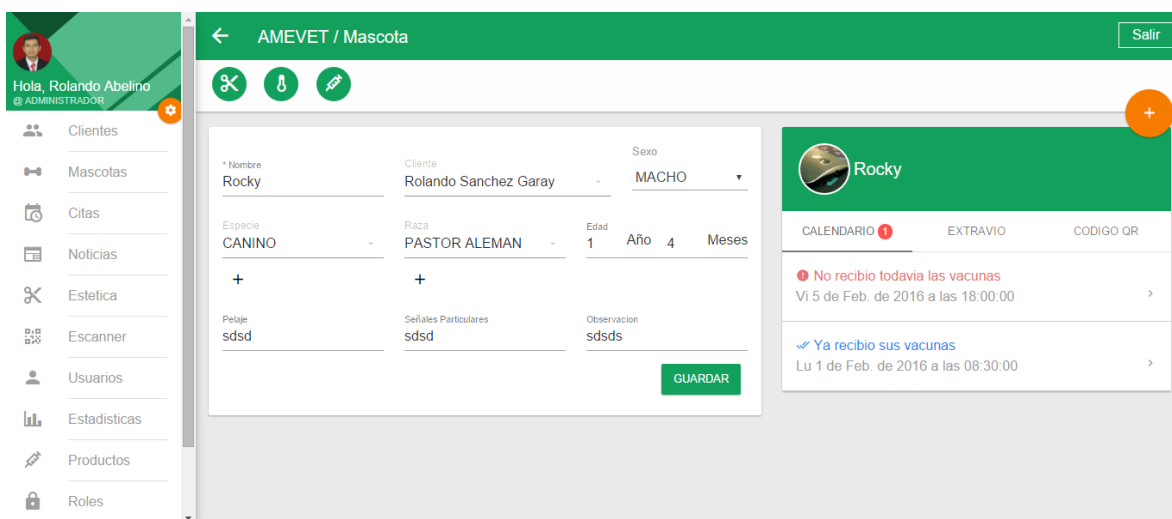


Ilustración 0.72 Pantalla Perfil de Mascota

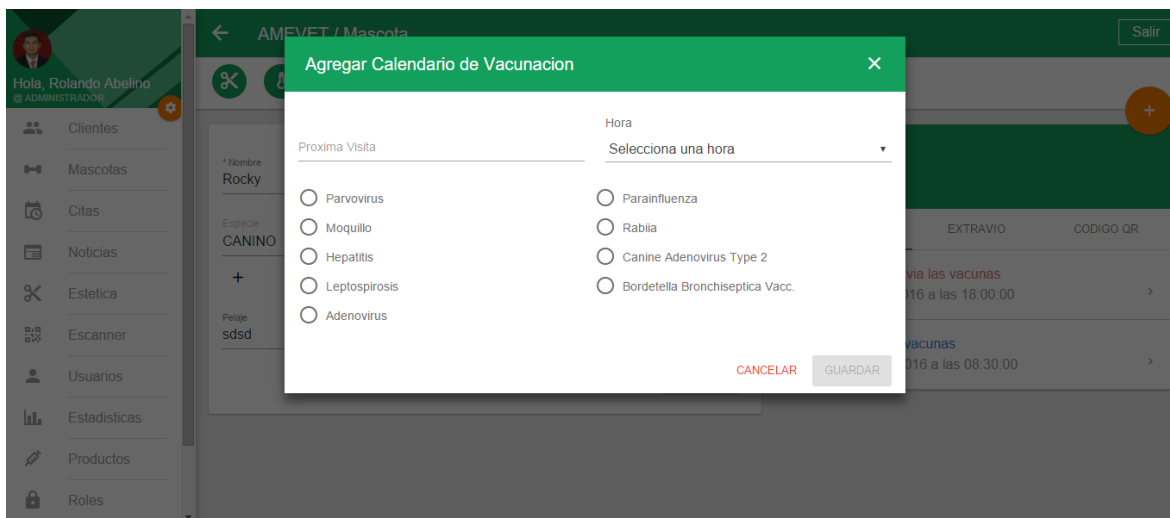


Ilustración 0.73 Pantalla Agregar Calendario

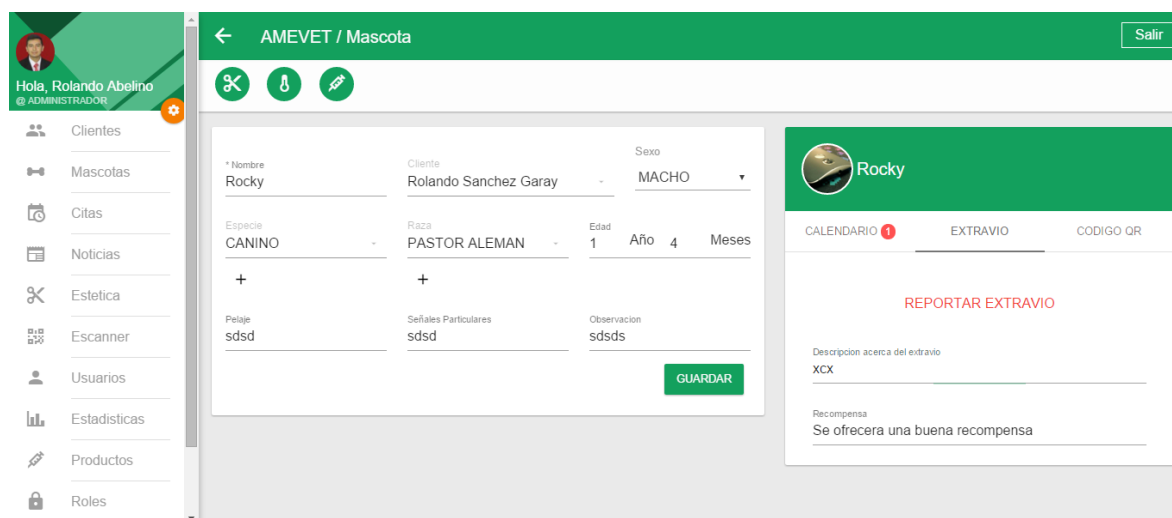


Ilustración 0.74 Pantalla Reportar Extravío

I.1.2.2.3.8.2.3.7 Pantallas Gestionar Noticias

Estas pantallas describen una lista de las noticias y agregar una noticia

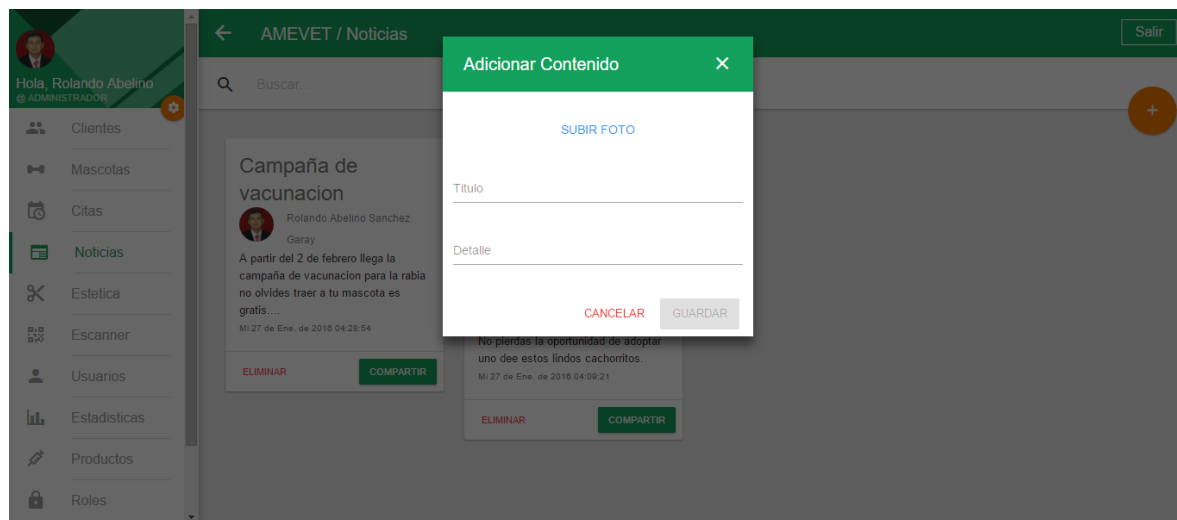


Ilustración 0.75 Pantalla Agregar Noticia

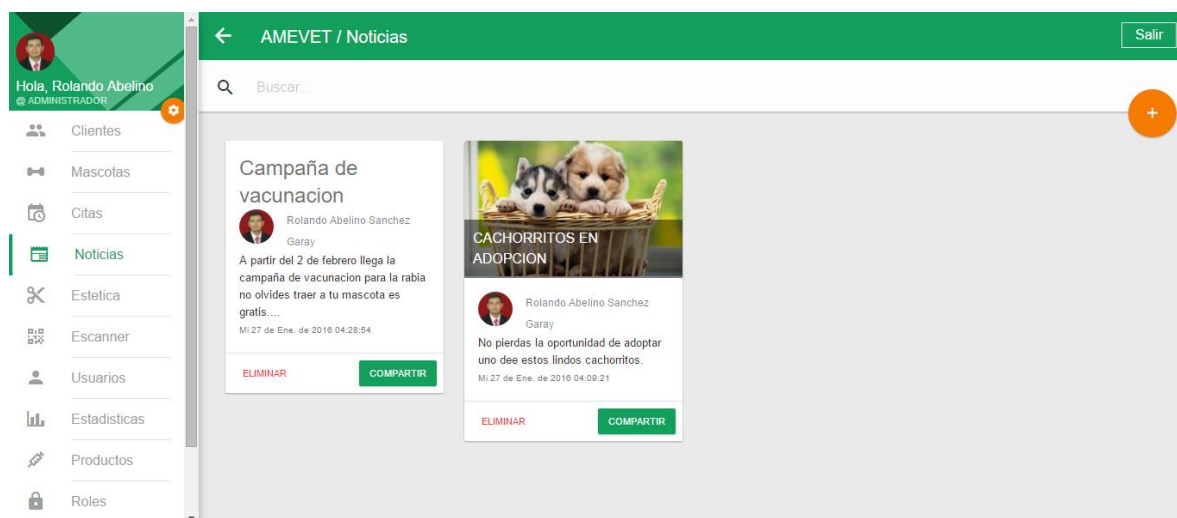


Ilustración 0.76 Pantalla Listar Noticias

I.1.2.2.3.8.2.3.8 Pantallas Gestionar Citas

Estas pantallas describen los procesos de listado de citas, agregar, modificar eliminar citas y las solicitudes de nuevas citas.

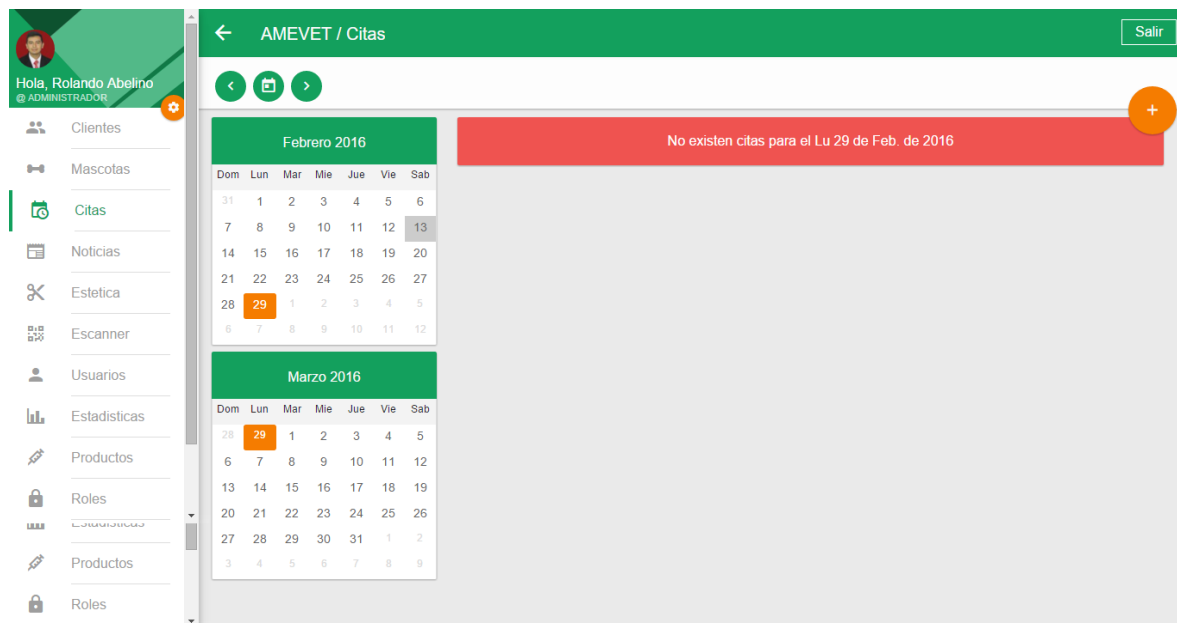


Ilustración 0.77 Pantalla Listar Citas

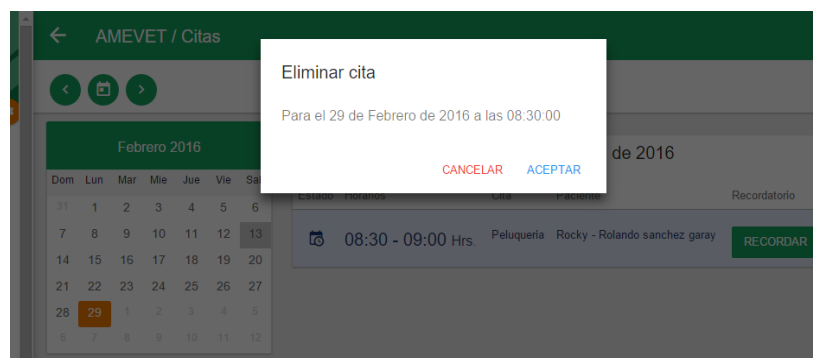


Ilustración 0.78 Pantalla Eliminar Cita

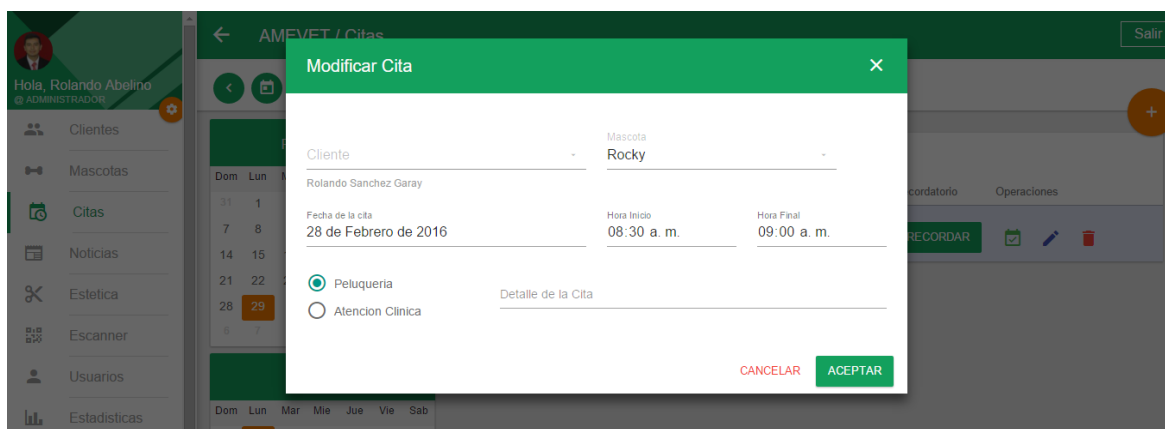


Ilustración 0.79 Pantalla Modificar Citas

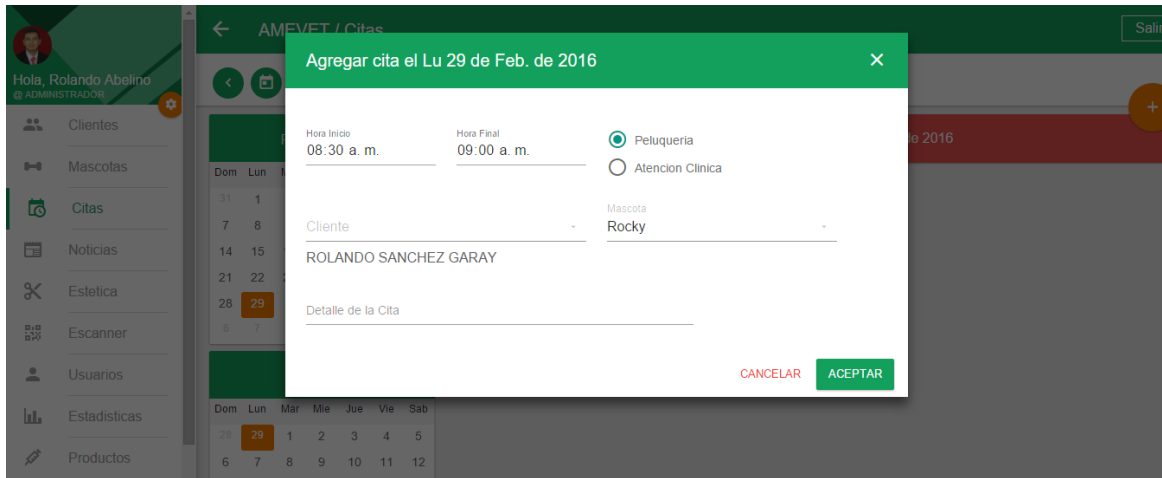


Ilustración 0.80 Pantalla Agregar Cita

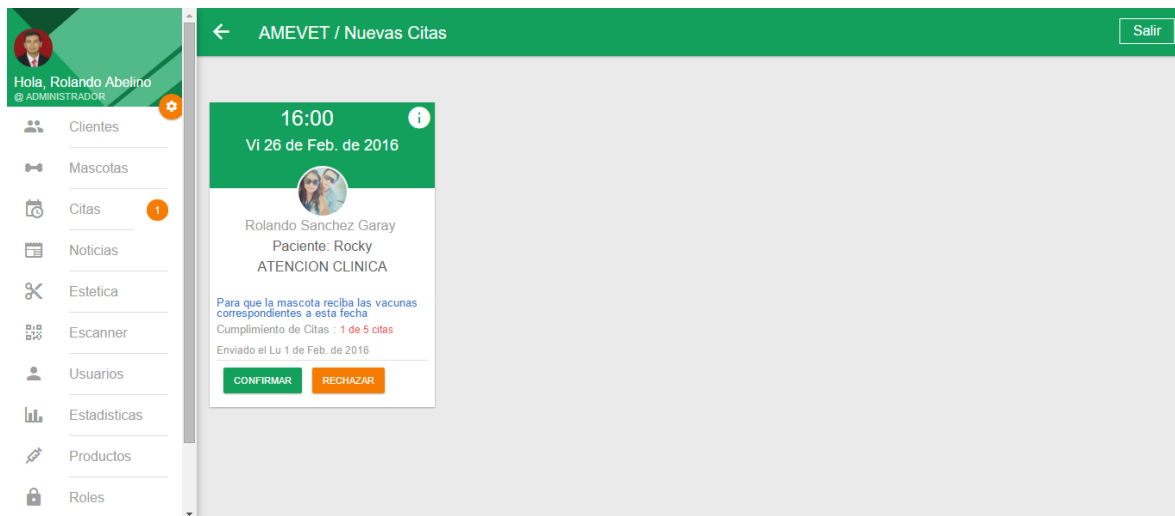


Ilustración 0.81 Pantalla Ver Solicitudes de Citas

I.1.2.2.3.8.2.3.9 Pantallas Gestionar Peluquerías

Estas pantallas Describen Listar Peluquerías y Agregar una nueva peluquería.

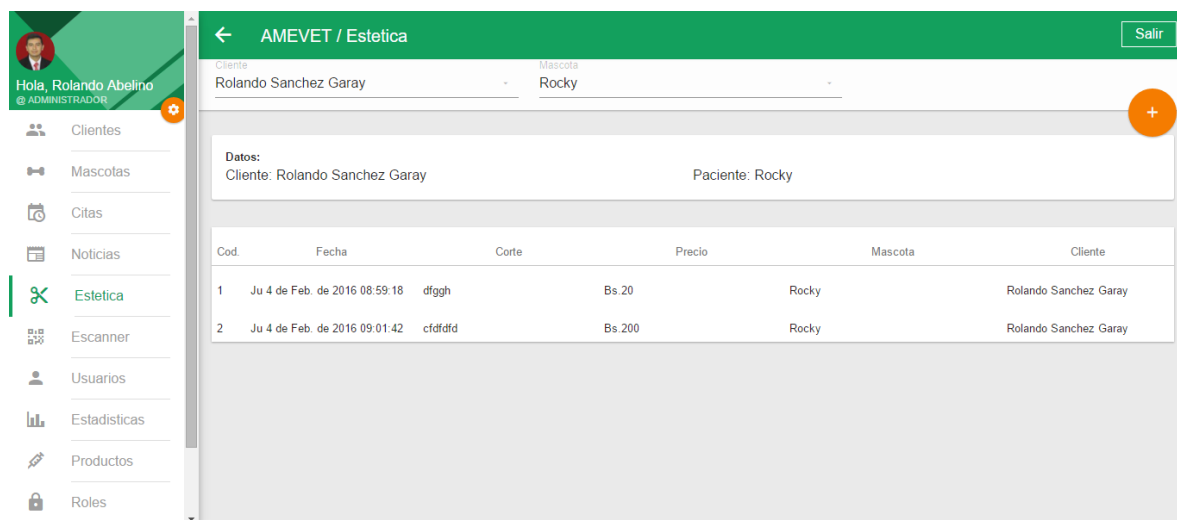


Ilustración 0.82 Pantalla Listar Peluquerías

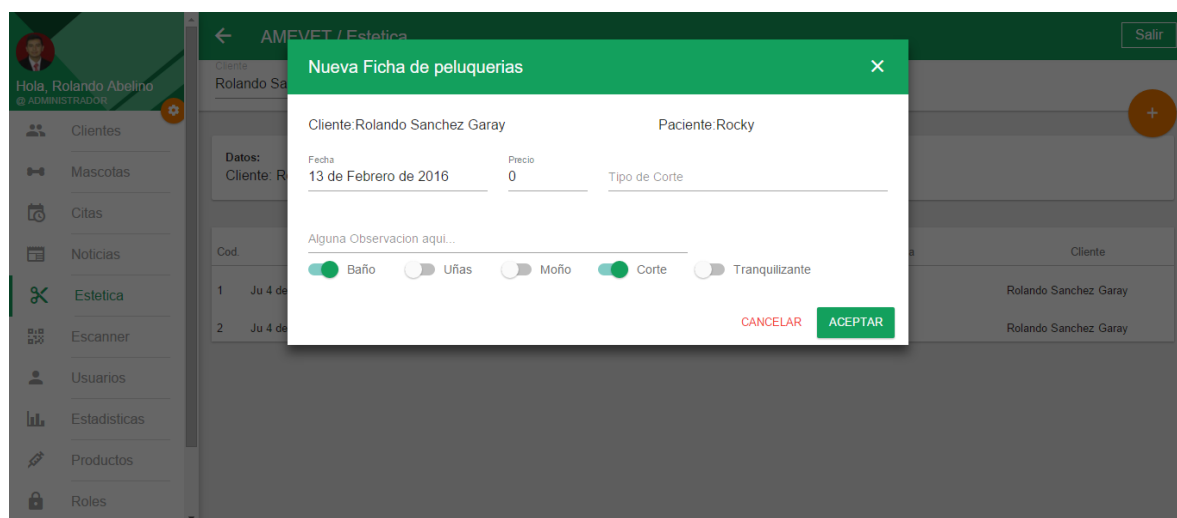


Ilustración 0.83 Pantalla Agregar Peluquería

I.1.2.2.3.8.2.3.10 Pantallas Gestionar Control sanitario

Estas pantallas describen todo el proceso de un control sanitario de una mascota, como el listado de desparasitaciones, vacunaciones, revacunaciones y ver peso de manera gráfica y analítica.



Hola, Rolando Abelino
@ ADMINISTRADOR

- Clientes
- Mascotas
- Citas 1
- Noticias
- Estetica
- Escanner
- Usuarios
- Estadísticas
- Productos
- Roles

← AMEVET / Control Sanitario Salir

Datos del Propietario

Nombre.....Rolando Sanchez Garay

Direccion.....b/ villa fatima

Tel..... 466724809 Cel.....

Datos del Paciente

Nombre..... ROCKY

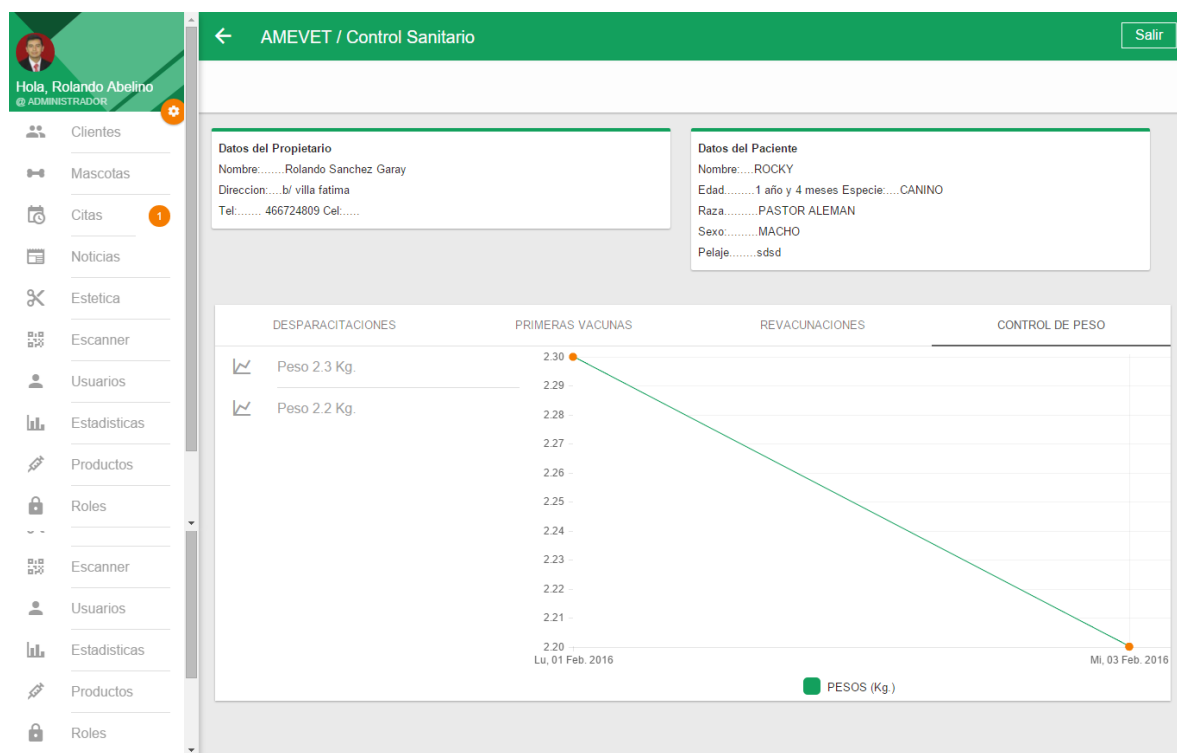
Edad.....1 año y 4 meses Especie.....CANINO

Raza.....PASTOR ALEMAN

Sexo.....MACHO

Pelaje.....sdsd

DESPARACITACIONES		PRIMERAS VACUNAS		REVACUNACIONES		CONTROL DE PESO	
Fecha	Peso	Producto	Medico	Prox. Visita	Obs.		
Lu 1 de Feb. de 2016 16:00:00	2.3 Kg.	LACRISTAL 3 X 25 ml	Rolando Abelino Sanchez Garay	Vi 26 de Feb. de 2016 16:00:00			



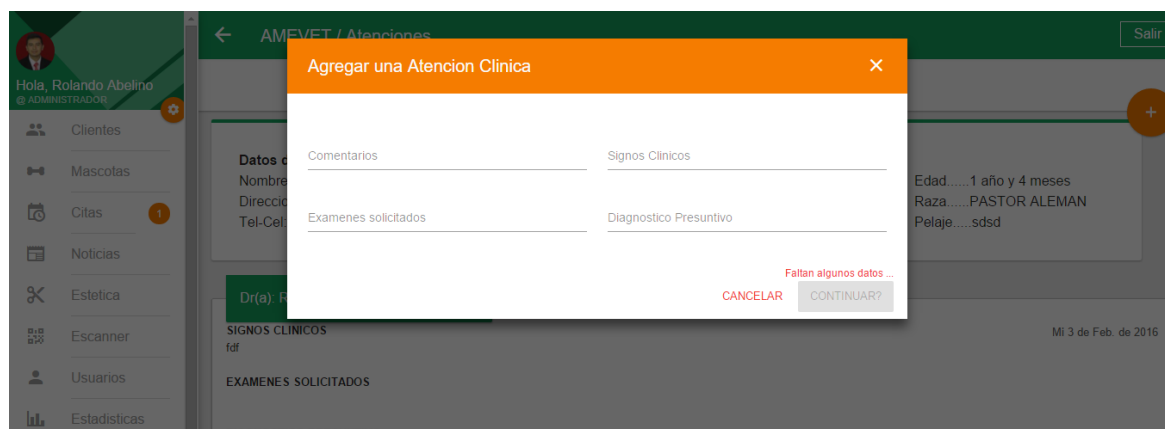


Ilustración 0.86 Pantalla agregar atención clínica

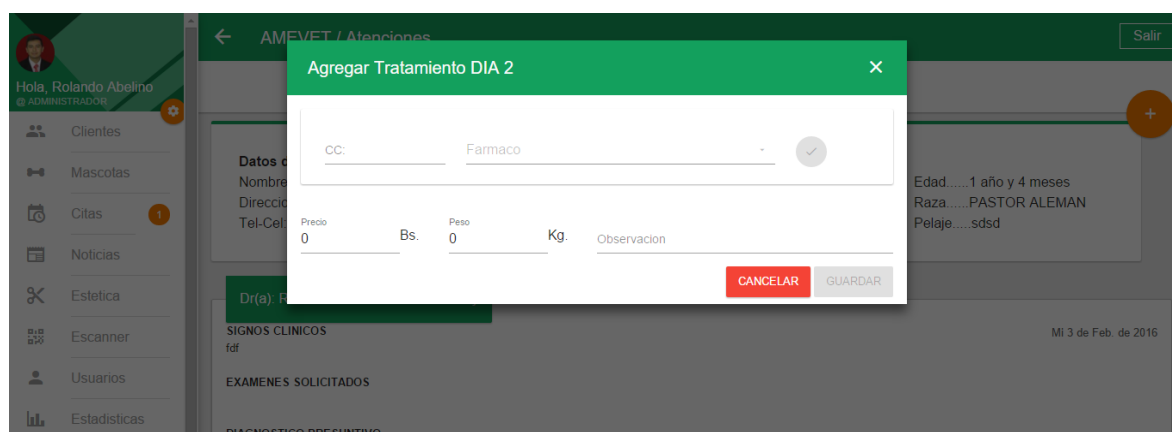


Ilustración 0.87 Pantalla agregar tratamiento

1.1.2.2.3.8.2.3.12 Pantallas Generar Reportes

Estas pantallas muestran los datos de ingreso por peluquería y atención clínica, así como datos de la aplicación móvil.

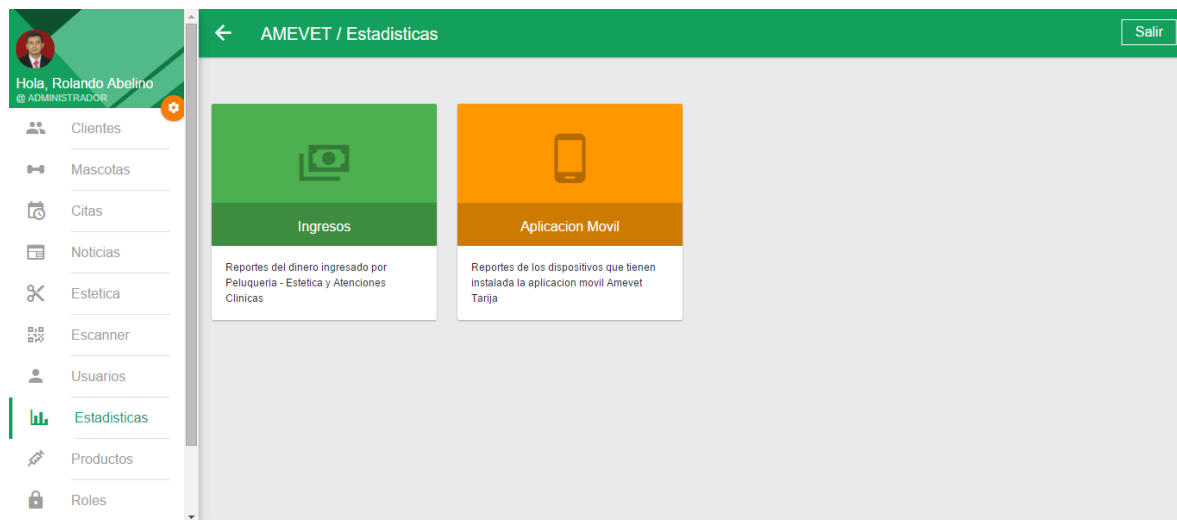


Ilustración 0.88 Pantalla principal de estadísticas

I.1.2.2.3.9 Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

I.1.2.2.3.9.1 Modelado de Diagrama de Actividades

I.1.2.2.3.9.1.1 Introducción

En UML un diagrama de actividades se usa para mostrar la secuencia de actividades. Los diagramas de actividades muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad.

I.1.2.2.3.9.1.1.1 Propósito

El propósito de un diagrama de actividad es modelar un proceso de flujo de trabajo y/o modelar las operaciones. Una operación es un servicio proporcionado por un objeto, que está disponible a través de una interfaz.

I.1.2.2.3.9.1.1.2 Alcance

Estos diagramas de actividades se utilizarán para describir el flujo de casos de usos que sean más complejos o sea necesaria una descripción.

I.1.2.2.3.9.1.2 Diagrama de Actividades

A continuación, en esta sección se mostrará algunas de las actividades del sistema en las que el flujo difiere de las demás actividades utilizadas en el sistema.

I.1.2.2.3.9.1.2.1 Diagrama de Actividad Ingresar al Sistema

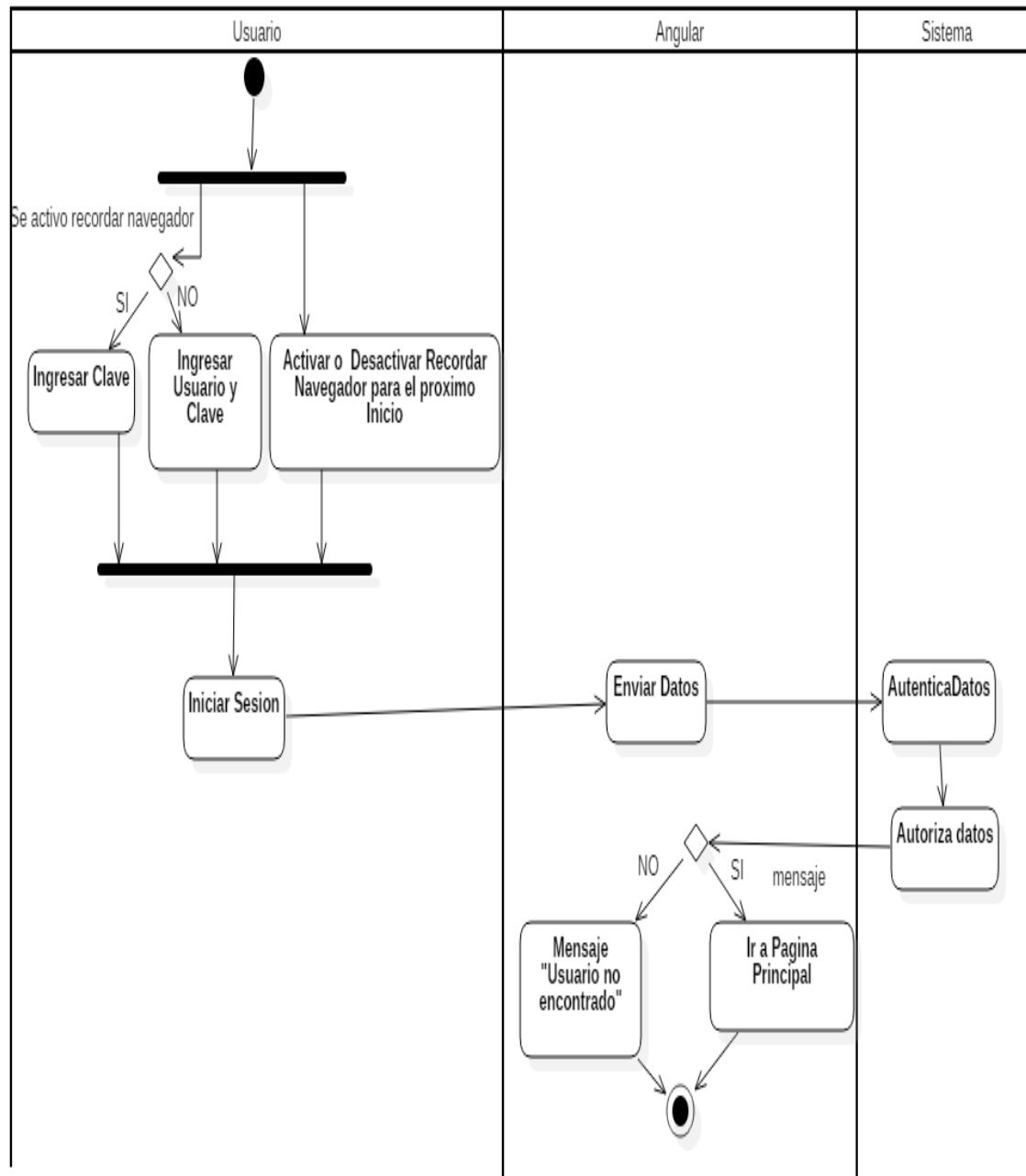


Ilustración 0.89 Diagrama de Actividad Ingresar al Sistema

I.1.2.2.3.9.1.2.2 Diagramas de Actividad Gestionar Usuarios

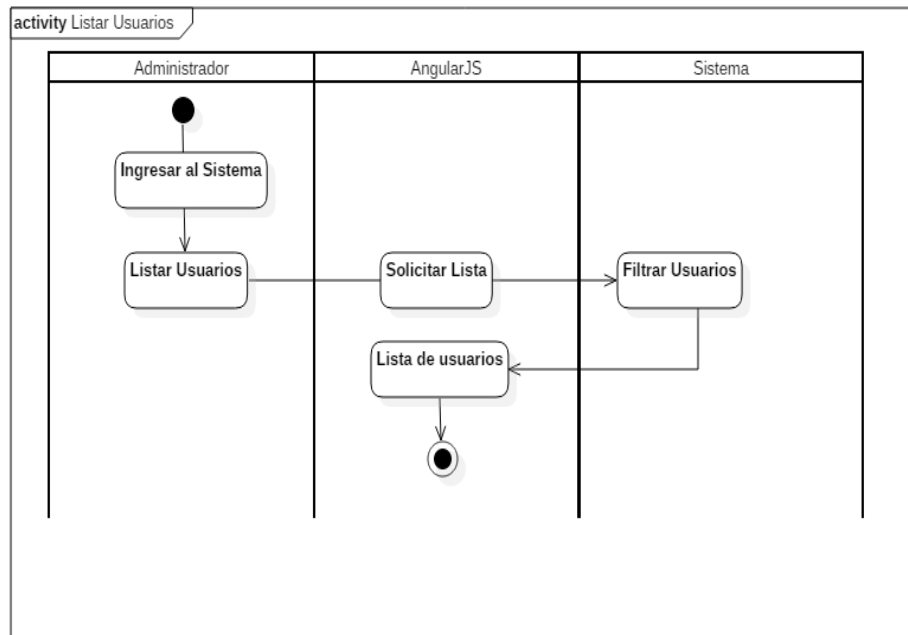


Ilustración 0.90 Diagrama de Actividad Listar Usuarios

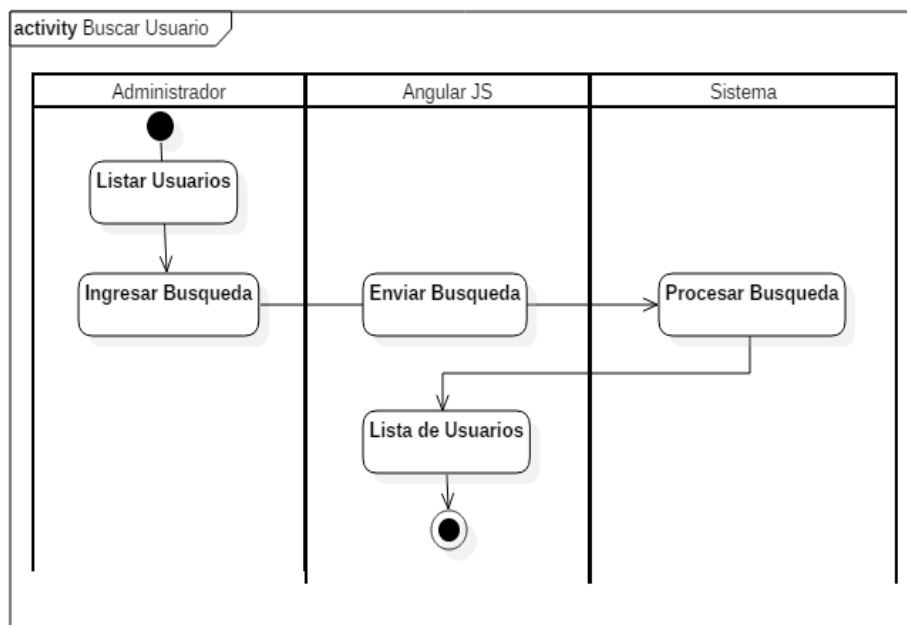


Ilustración 0.91 Diagrama de actividad Buscar usuarios

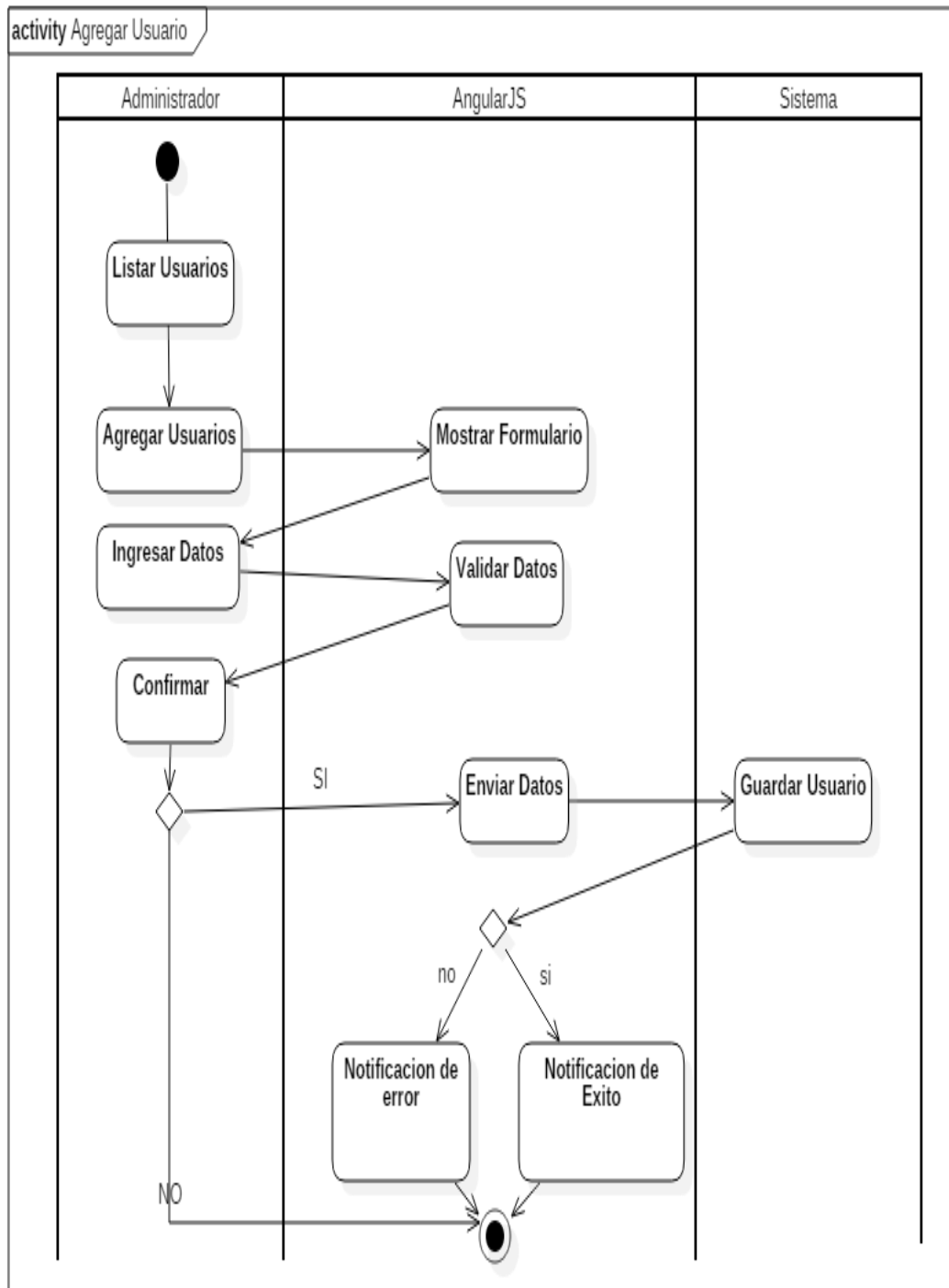


Ilustración 0.92 Diagrama de actividad Agregar Usuario

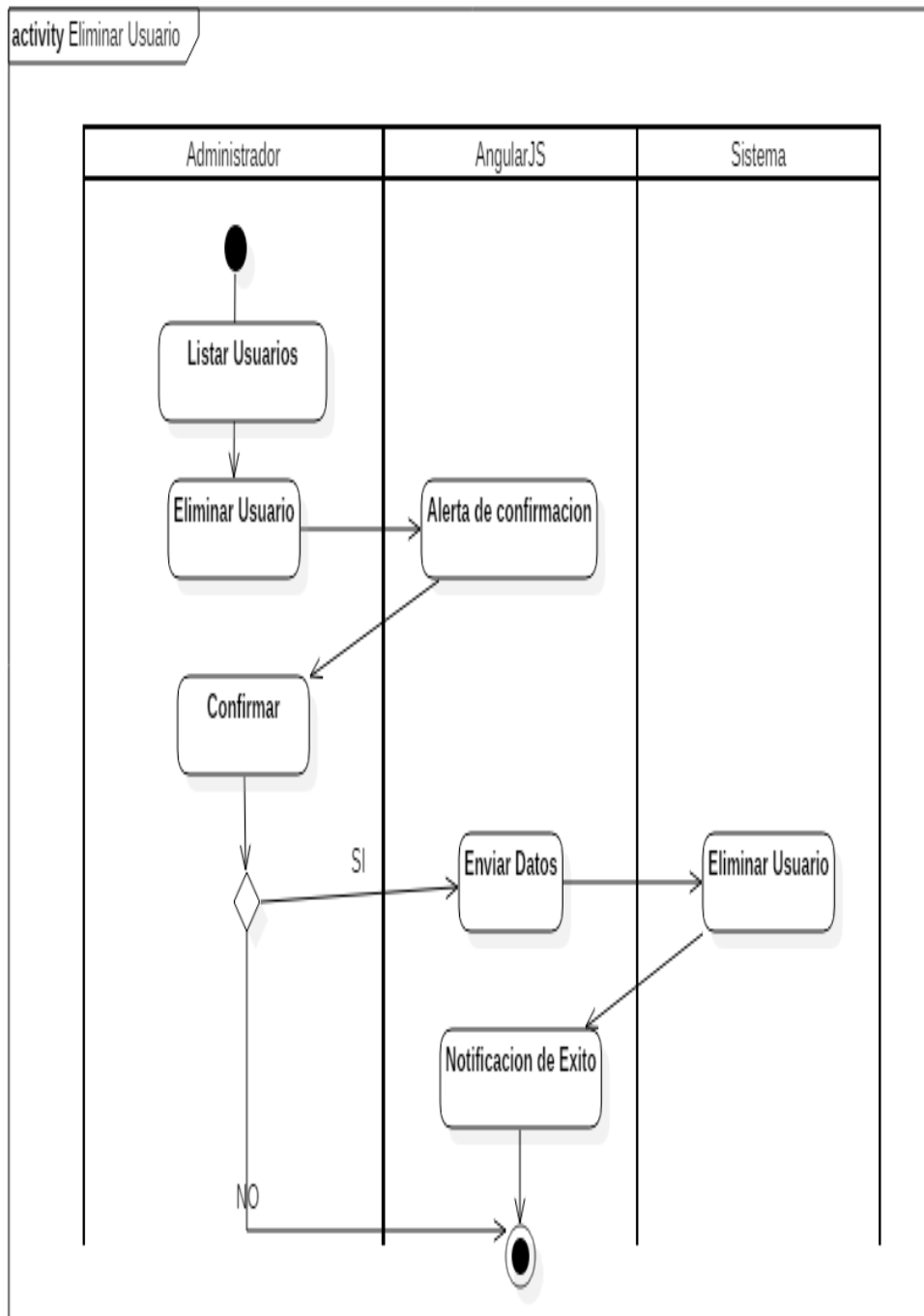


Ilustración 0.93 Diagrama de Actividad Eliminar Usuario

activity Modificar Usuario

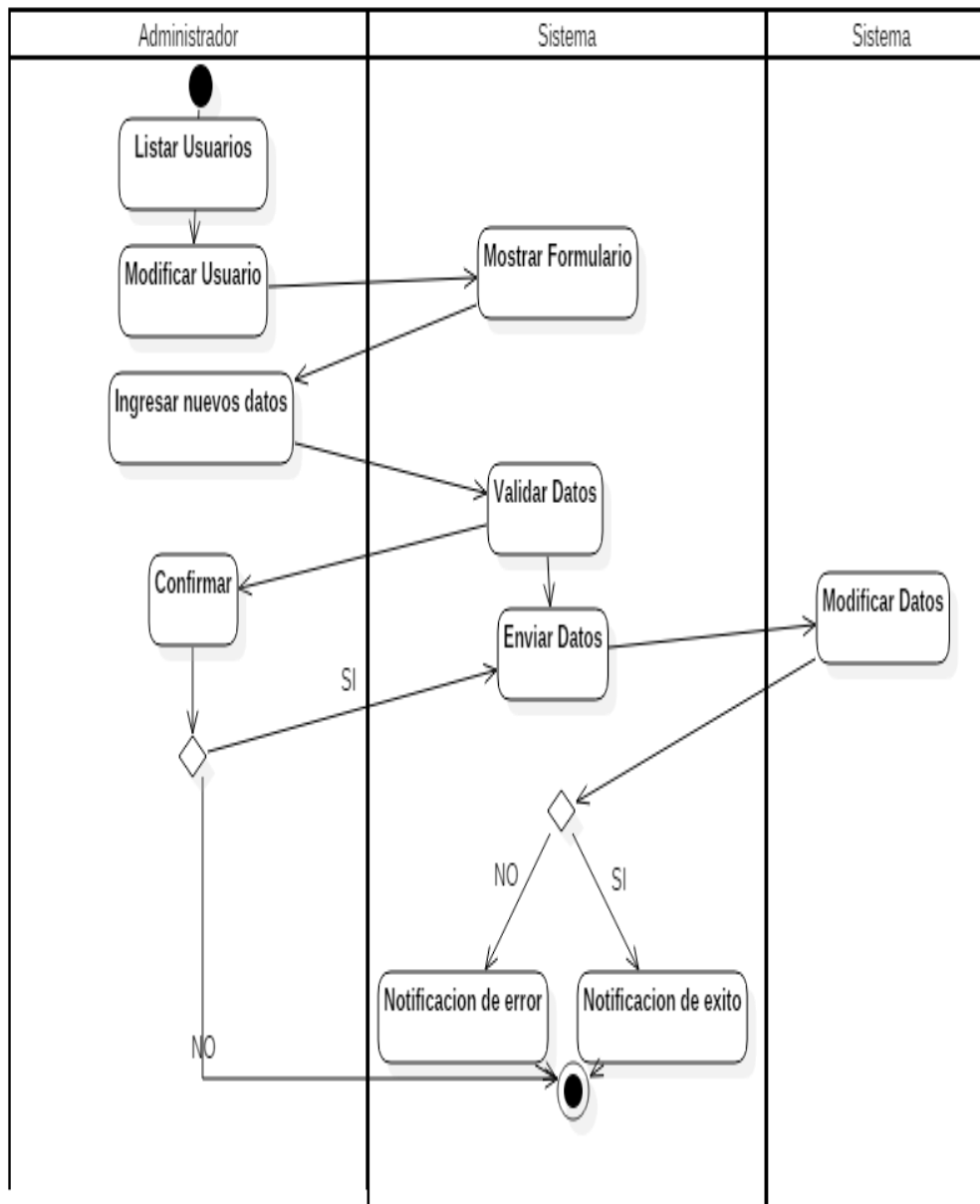


Ilustración 0.94 Diagrama de Actividad Modificar Usuario

I.1.2.2.3.9.1.2.3 Diagramas de Actividad Gestionar Citas

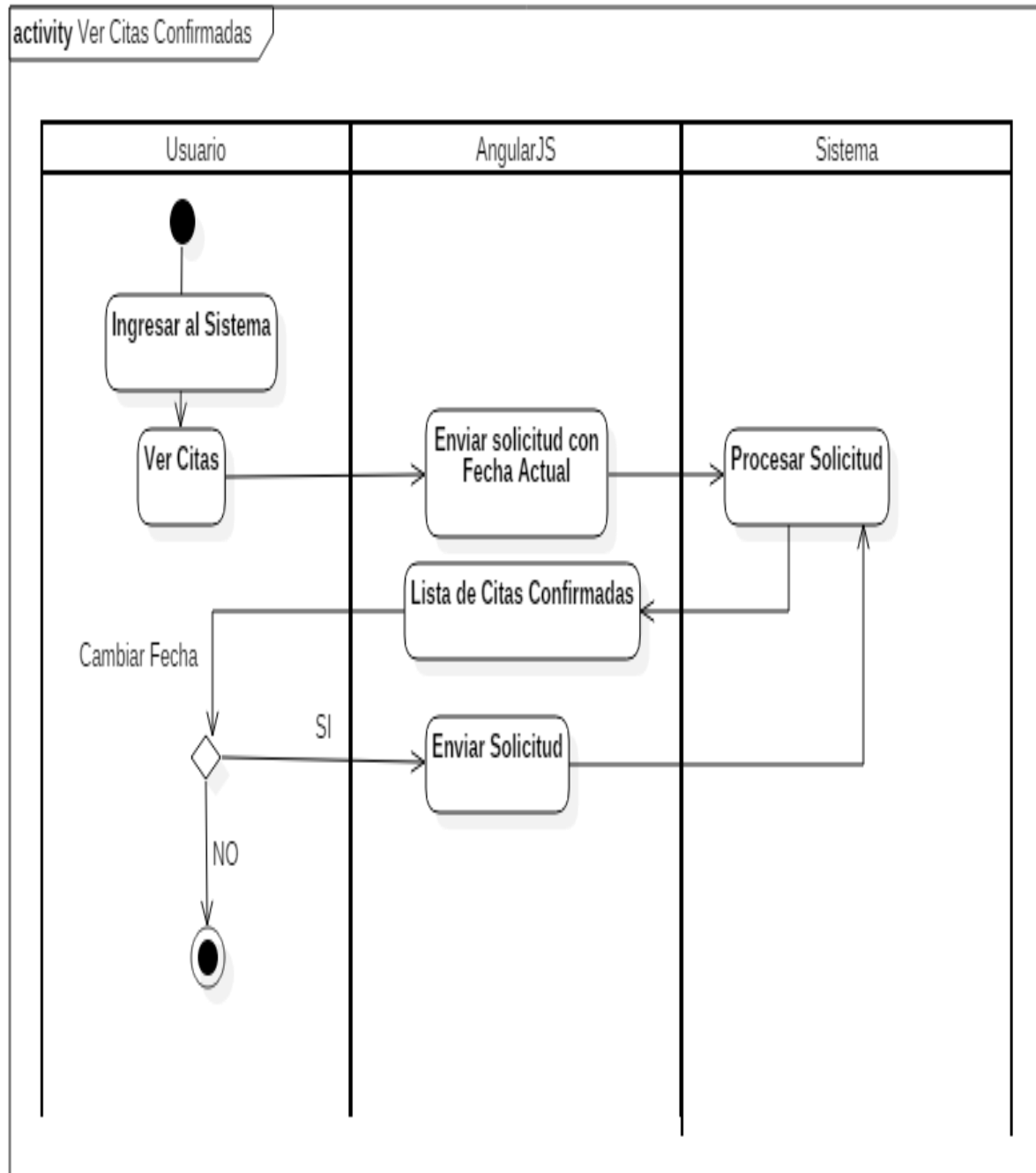


Ilustración 0.95 Diagrama de Actividad Listar Cita

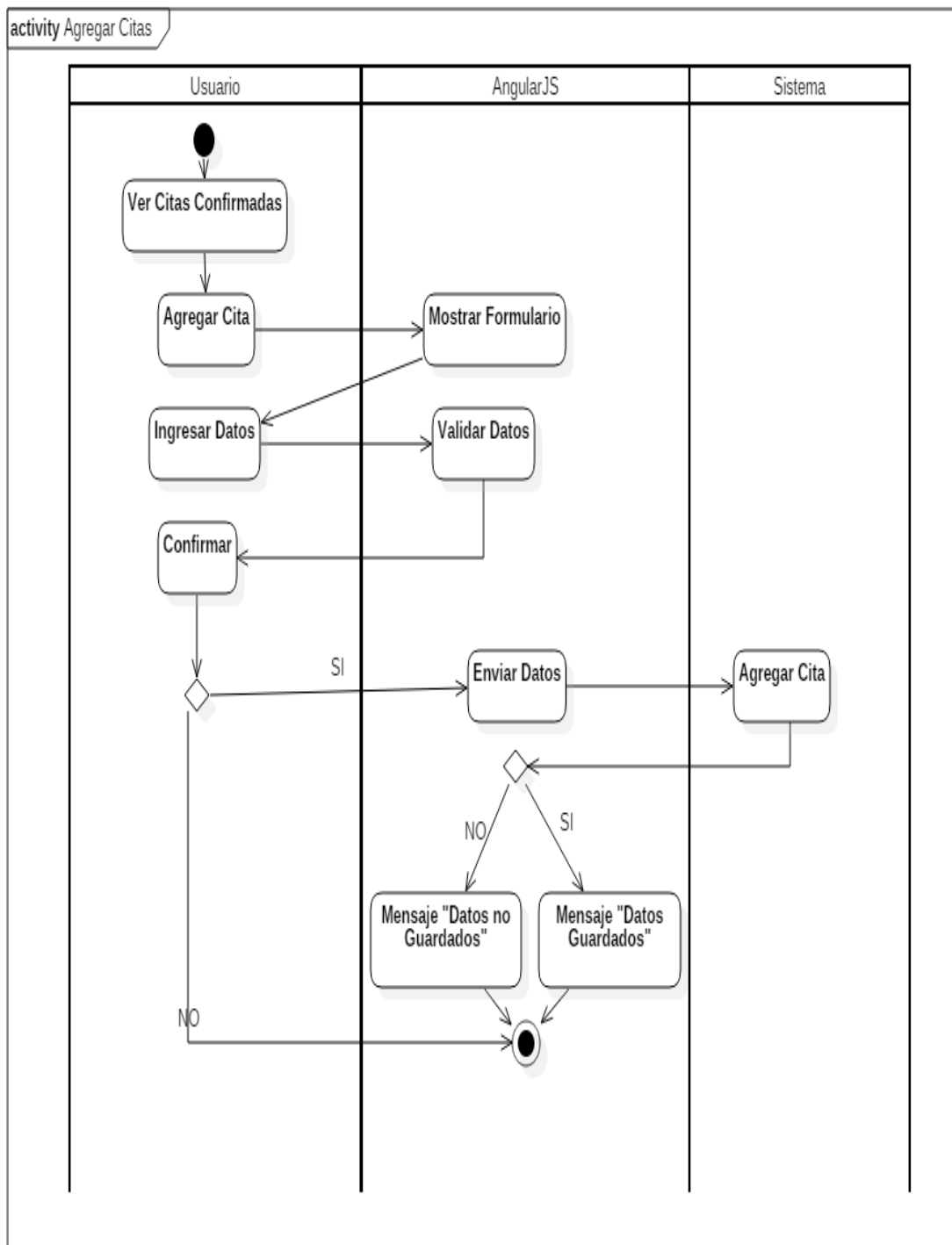


Ilustración 0.96 Diagrama de actividad Agregar Cita

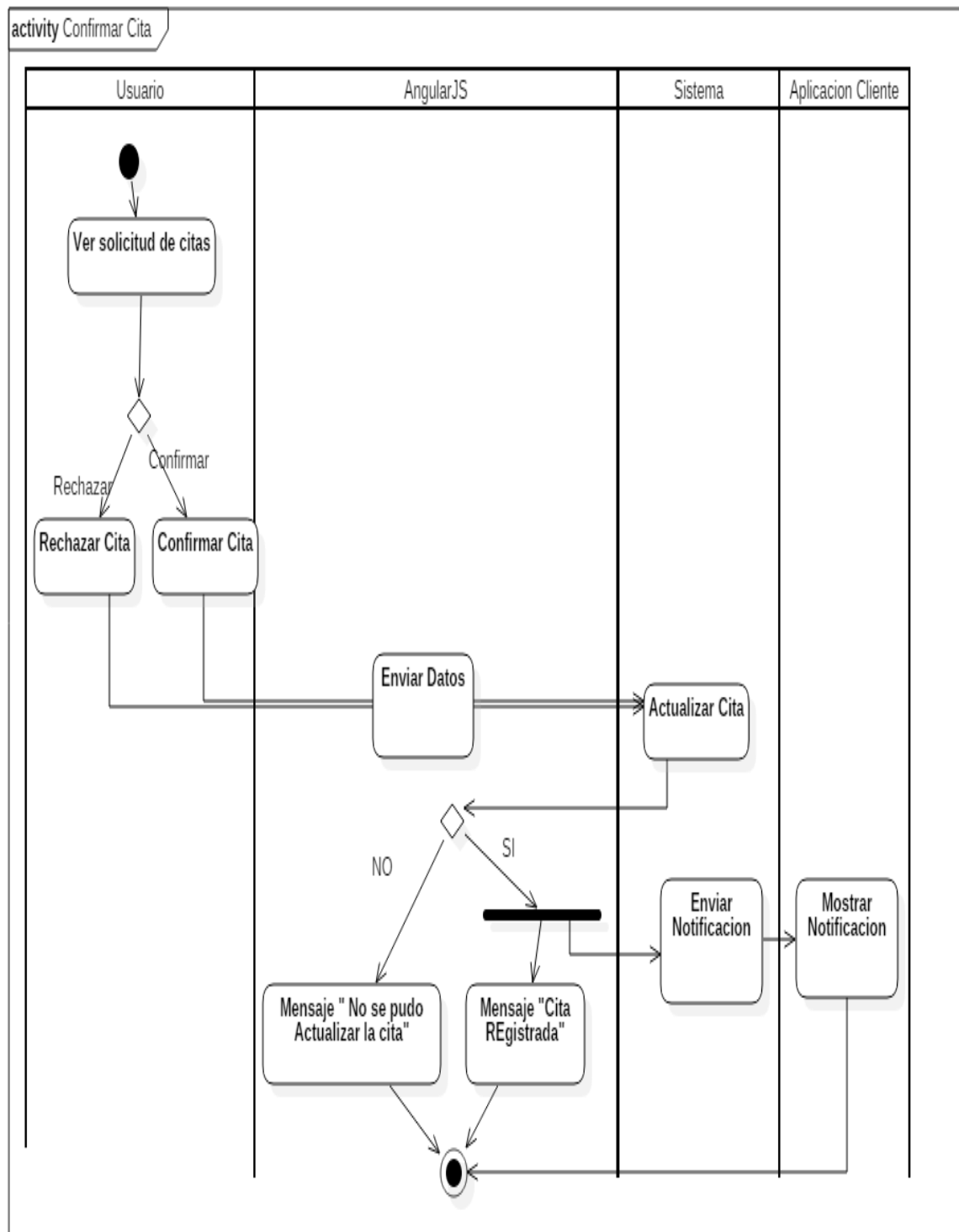


Ilustración 0.97 Diagrama de actividad confirmar Cita

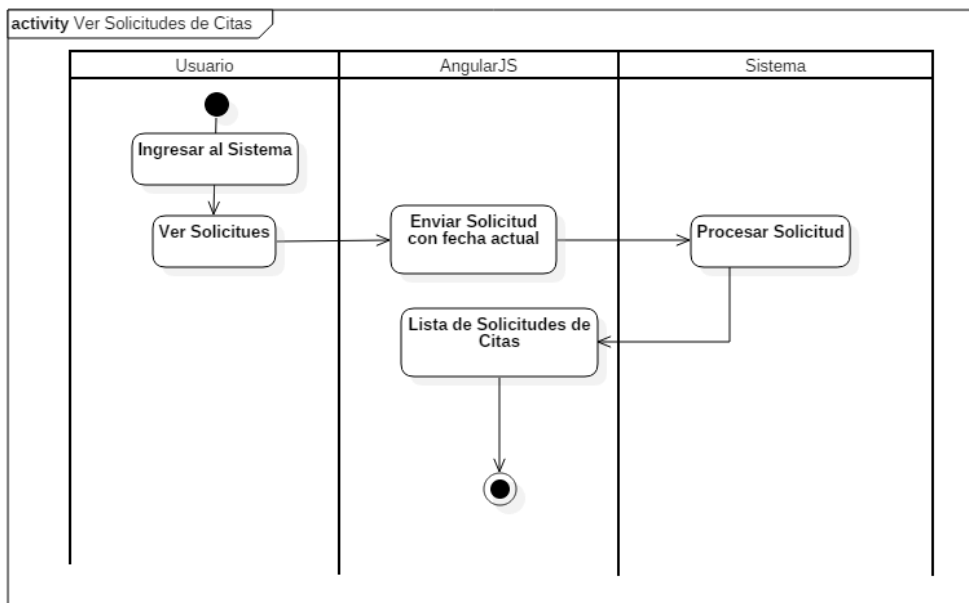


Ilustración 0.98 Diagrama de Actividad Ver solicitudes de Cita

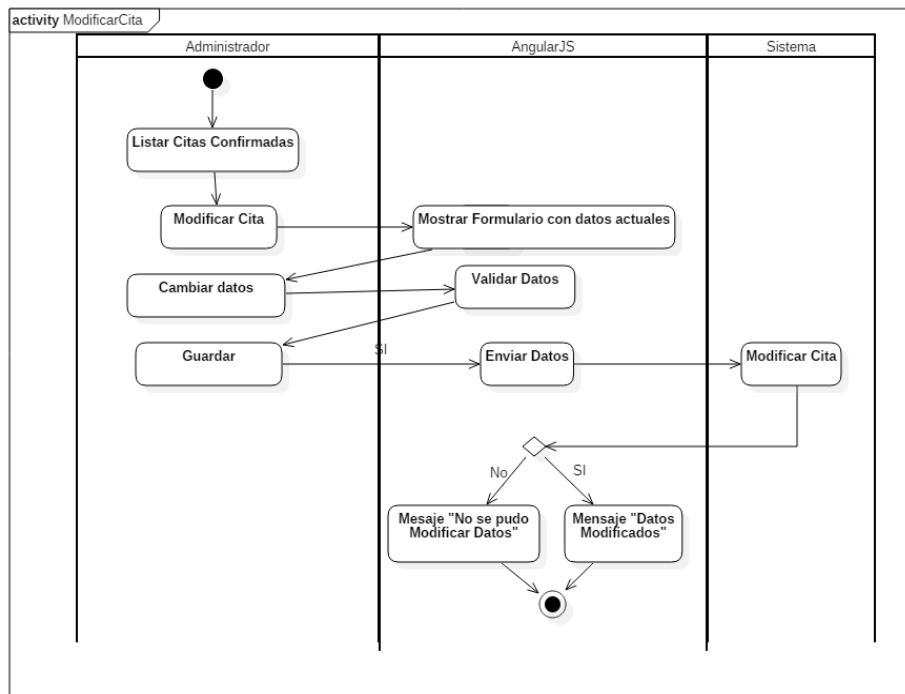


Ilustración 0.99 Diagrama de Actividad Modificar Cita

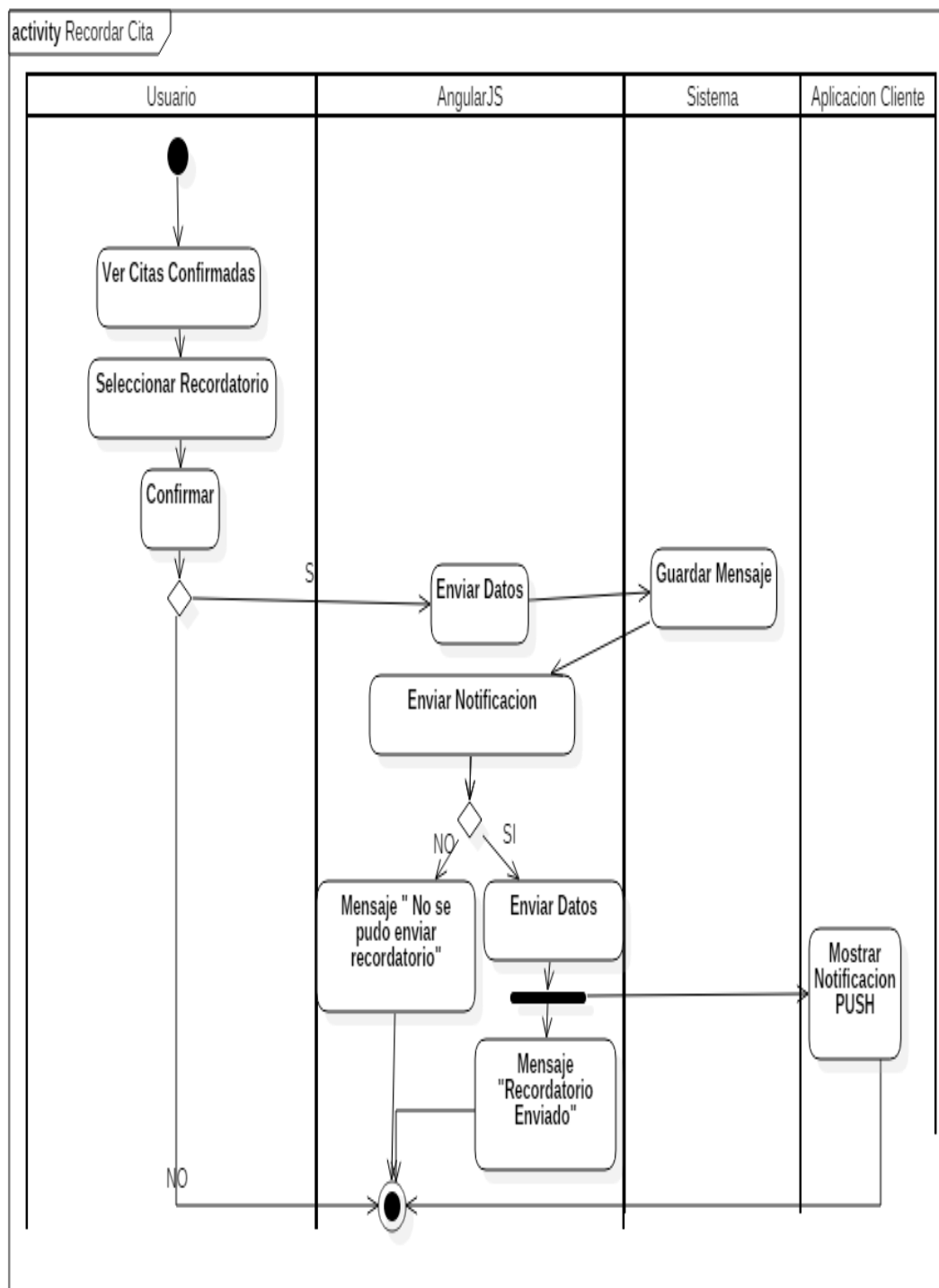


Ilustración 0.100 Diagrama de Actividad Enviar Recordatorio

I.1.2.2.3.9.1.2.4 Diagramas de Actividad Gestionar Roles

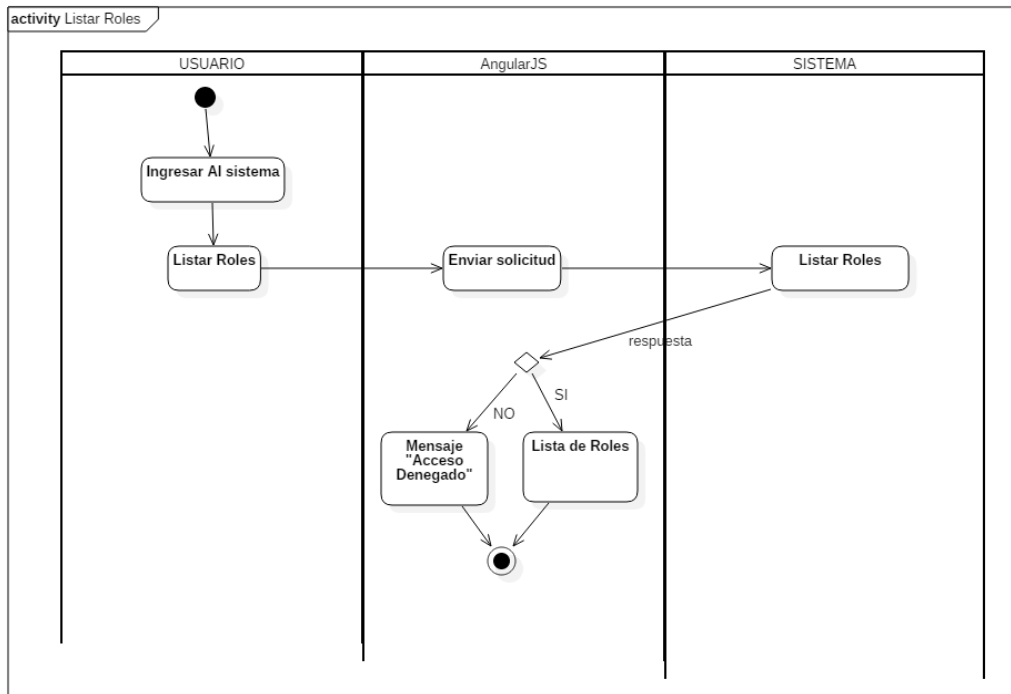


Ilustración 0.101 Diagrama de Actividad Listar Roles

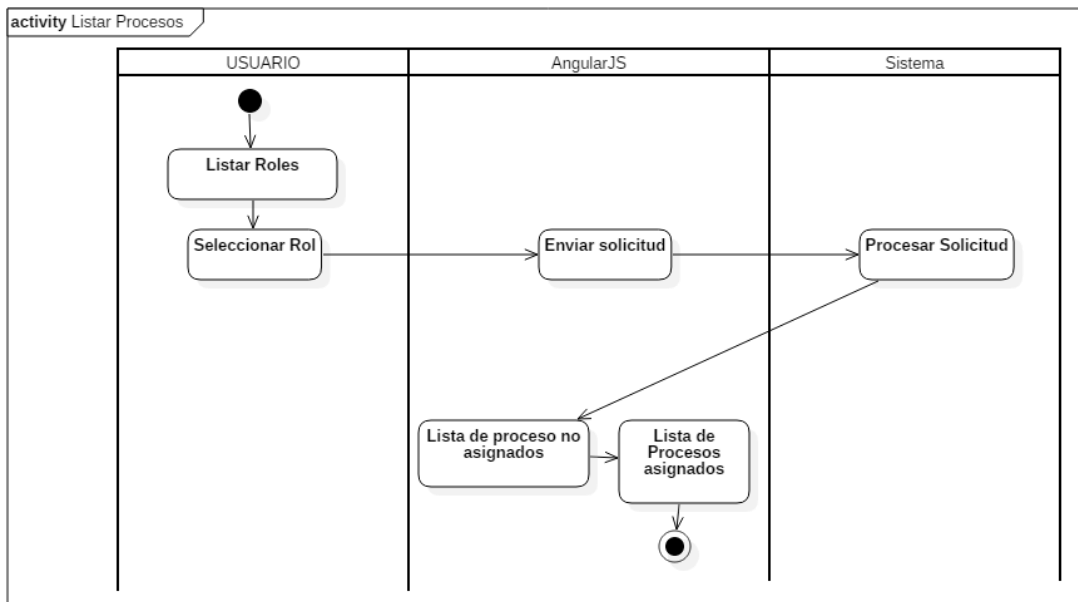


Ilustración 0.102 Diagrama de Actividad Listar Procesos

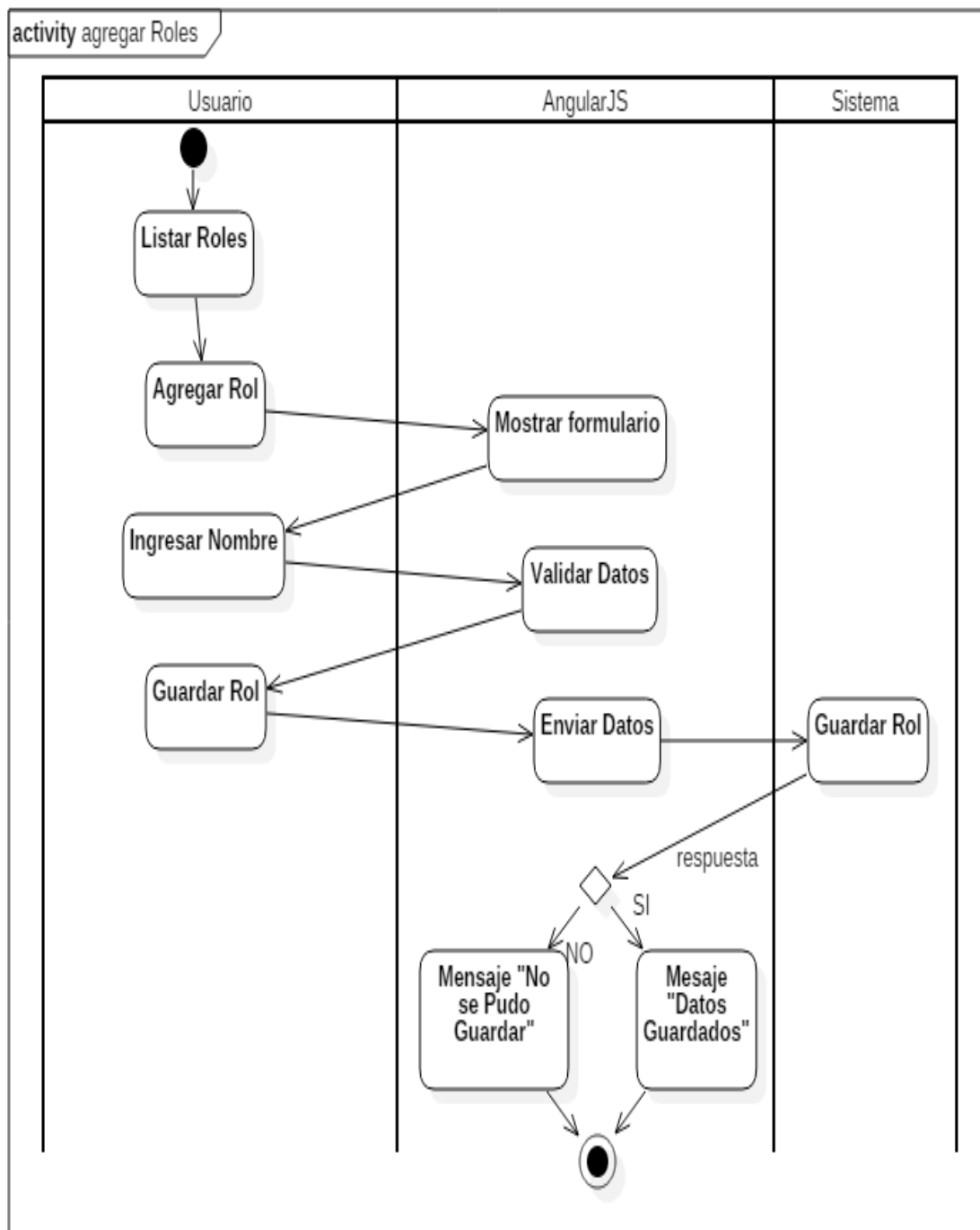


Ilustración 0.103 Diagrama de Actividad Agregar Rol

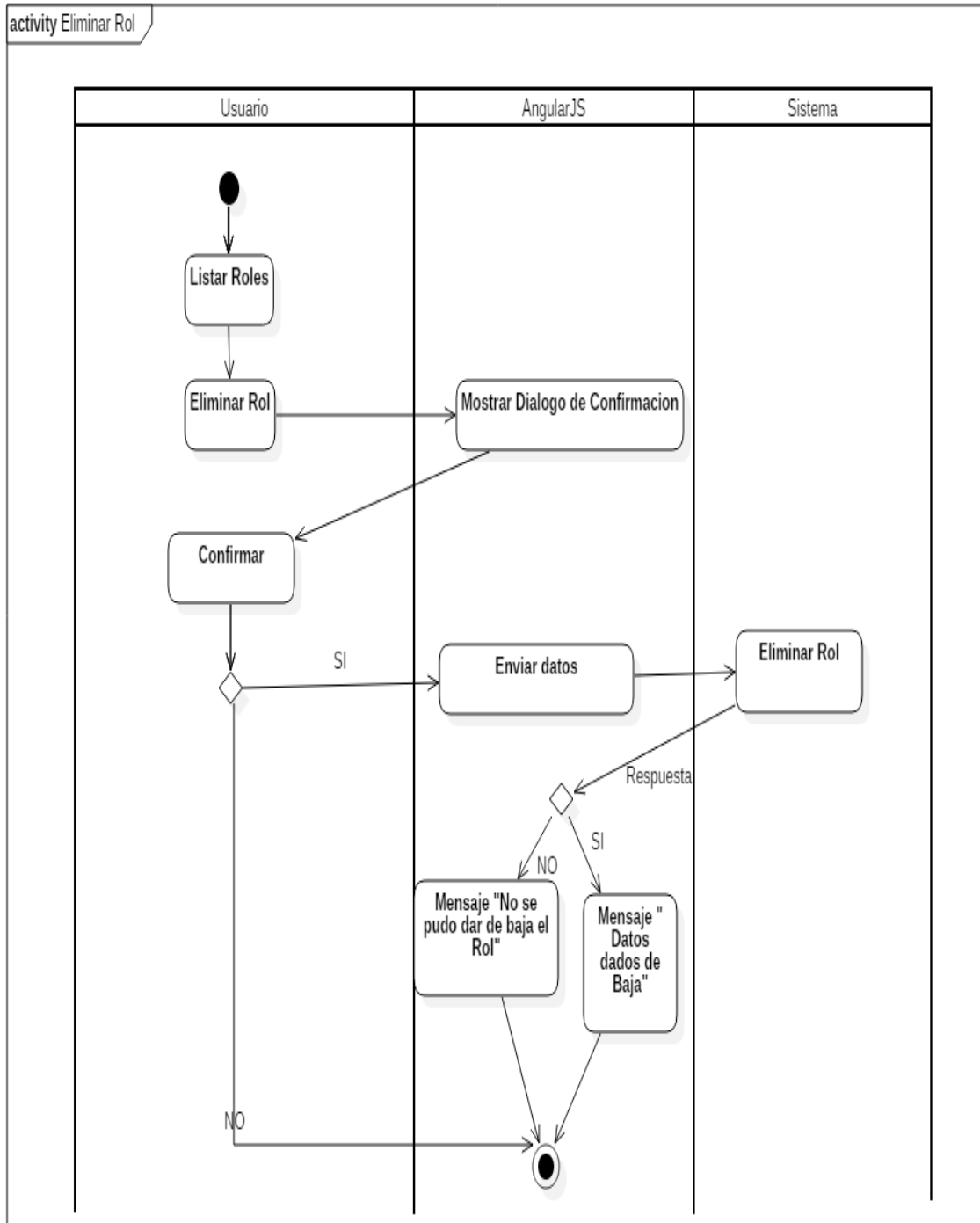


Ilustración 0.104 Diagrama de Actividad Eliminar Rol

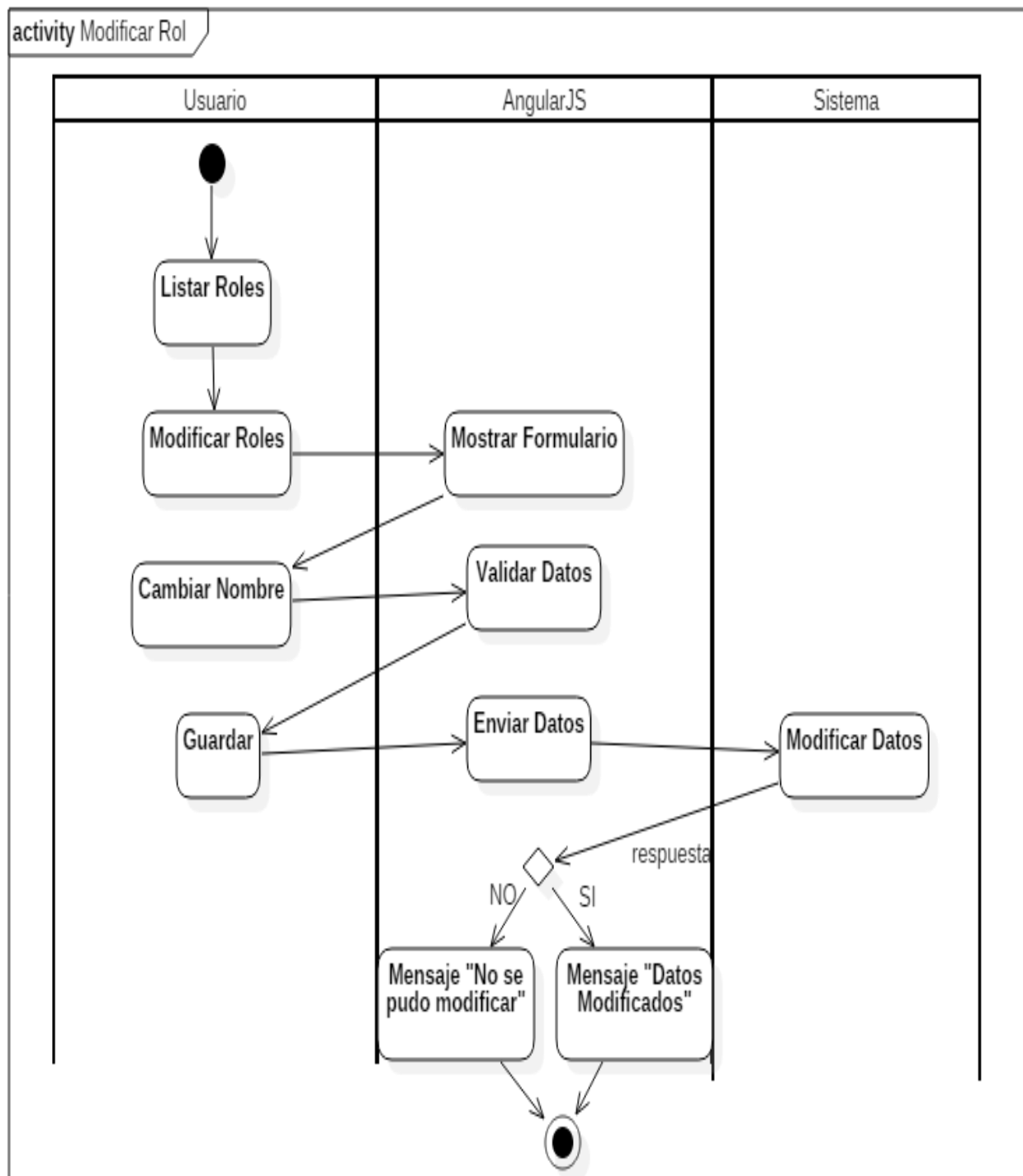


Ilustración 0.105 Diagrama de Actividad Modificar Rol

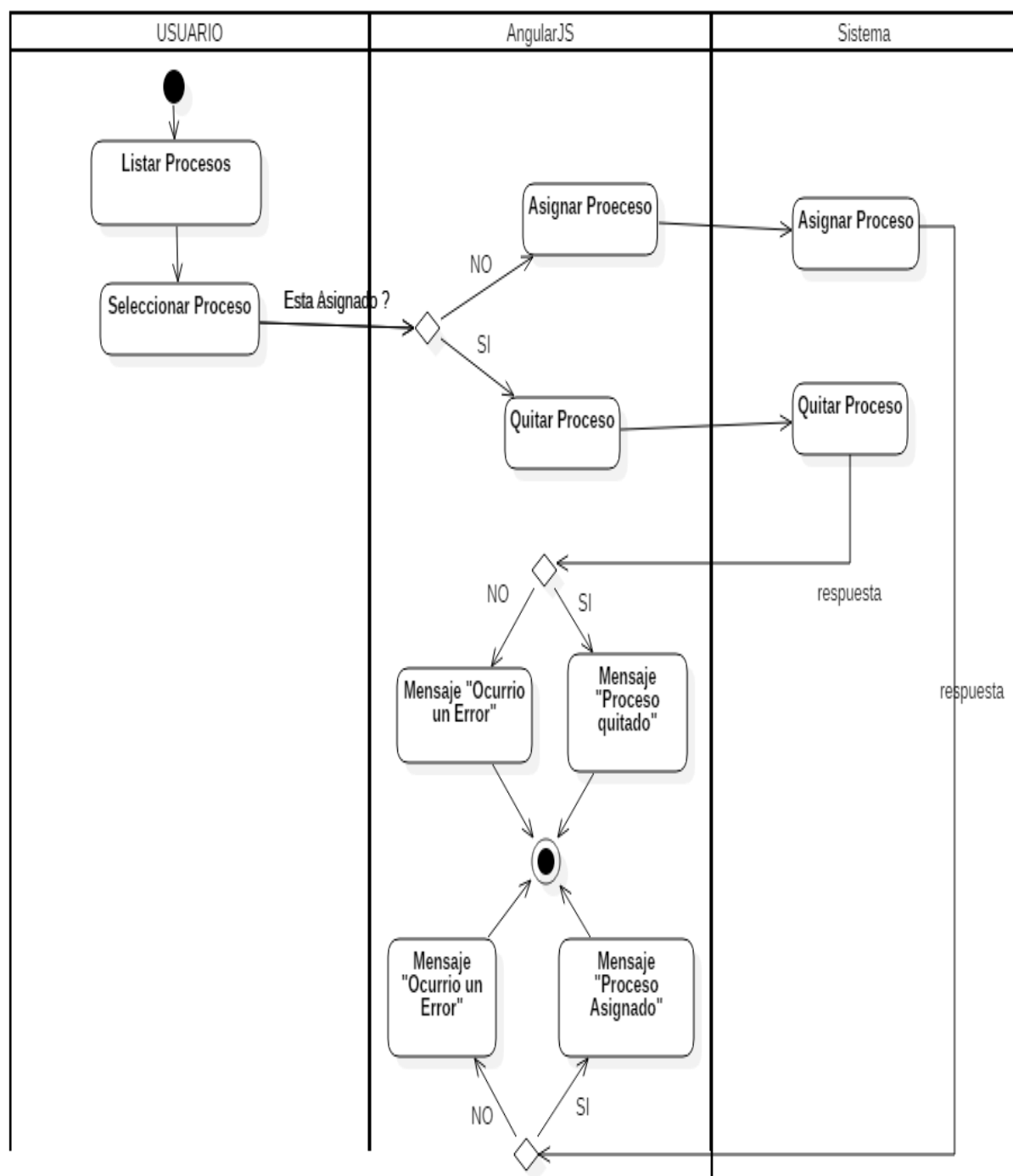


Ilustración 0.106 Diagrama de Actividad Asignar y Quitar Procesos

I.1.2.2.3.9.1.2.5 Diagramas de Actividad Gestionar Noticias

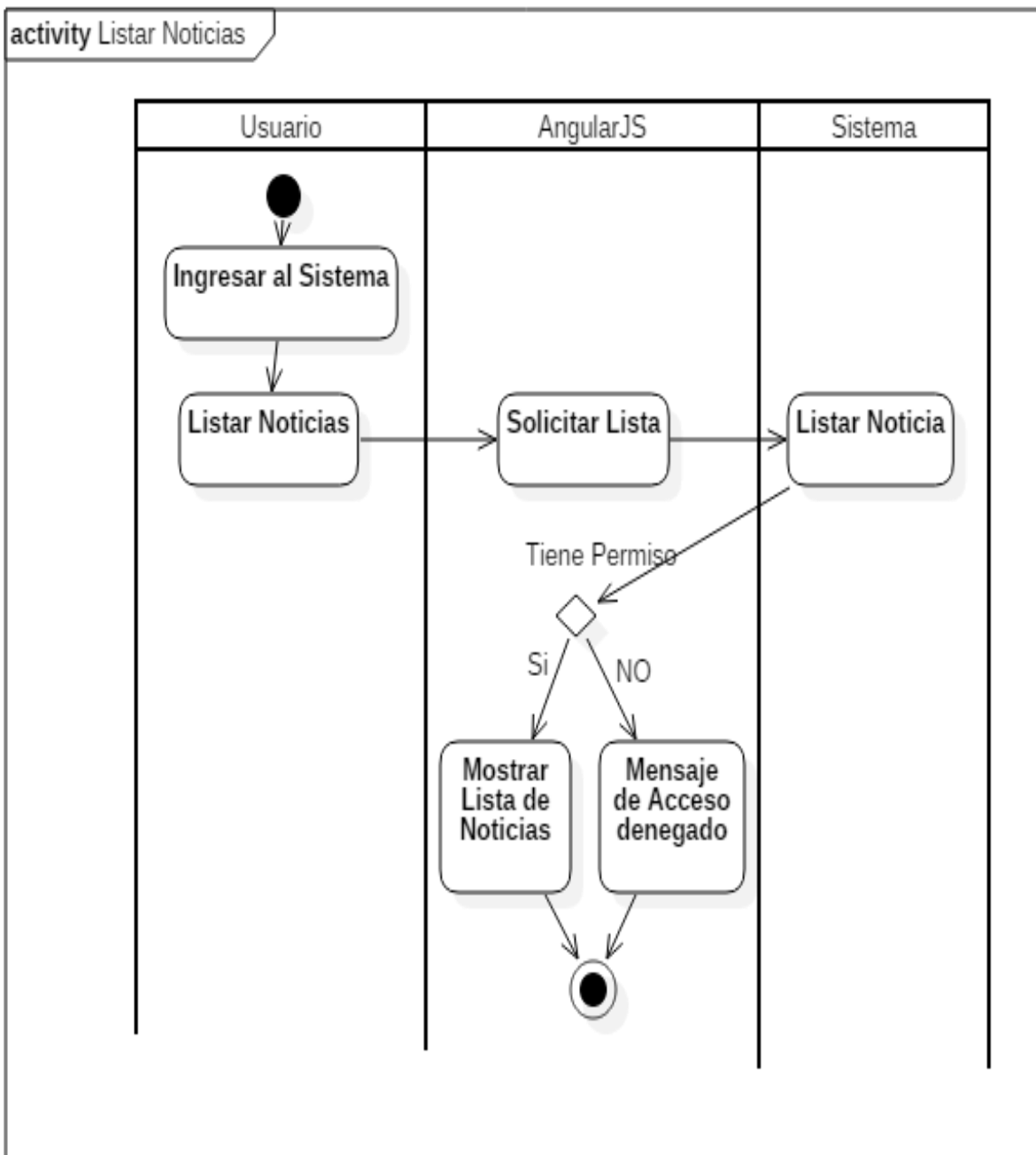


Ilustración 0.107 Diagrama de Actividad Listar noticias

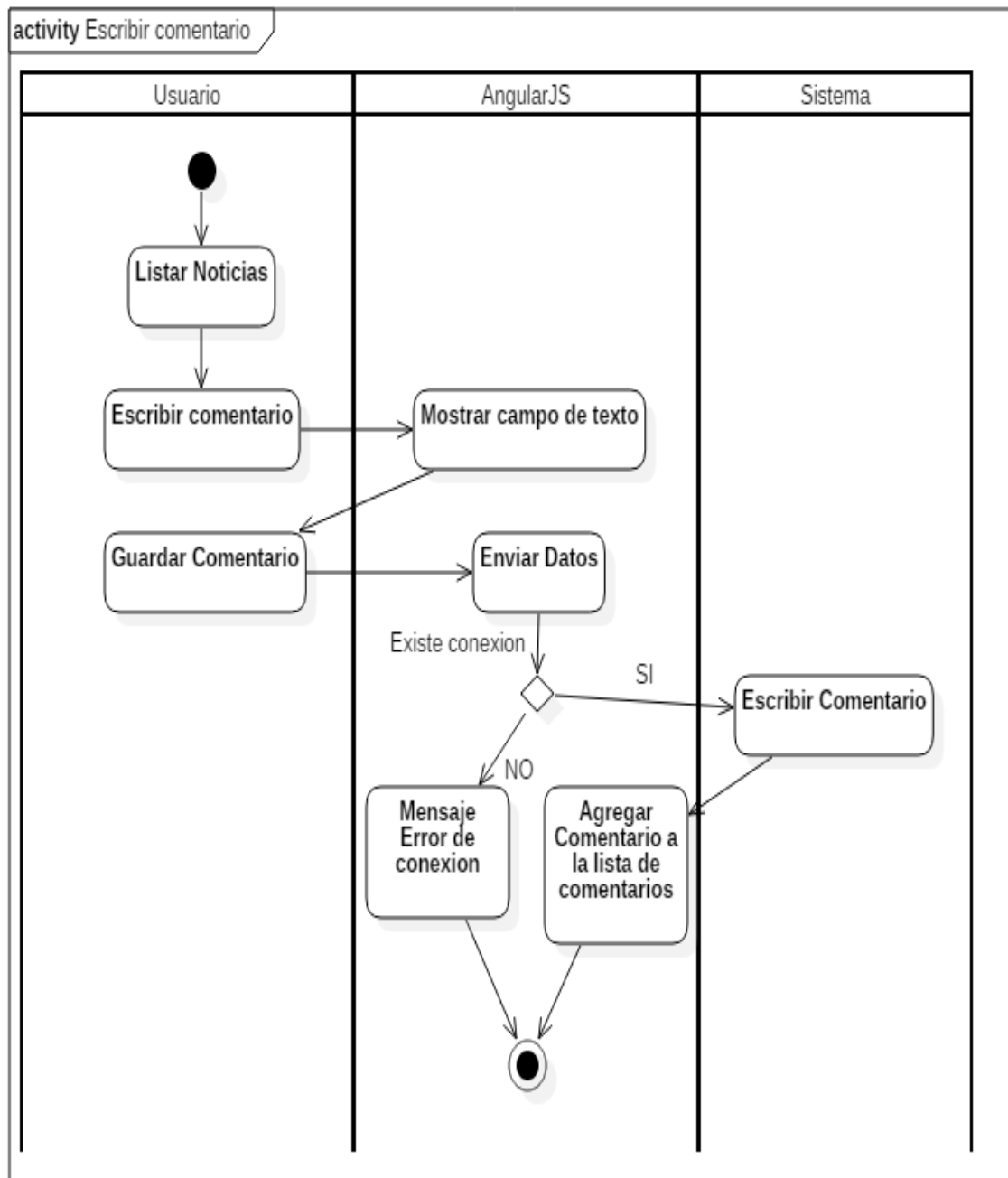


Ilustración 0.108 Diagrama de Actividad Escribir Comentario

activity Eliminar Noticia

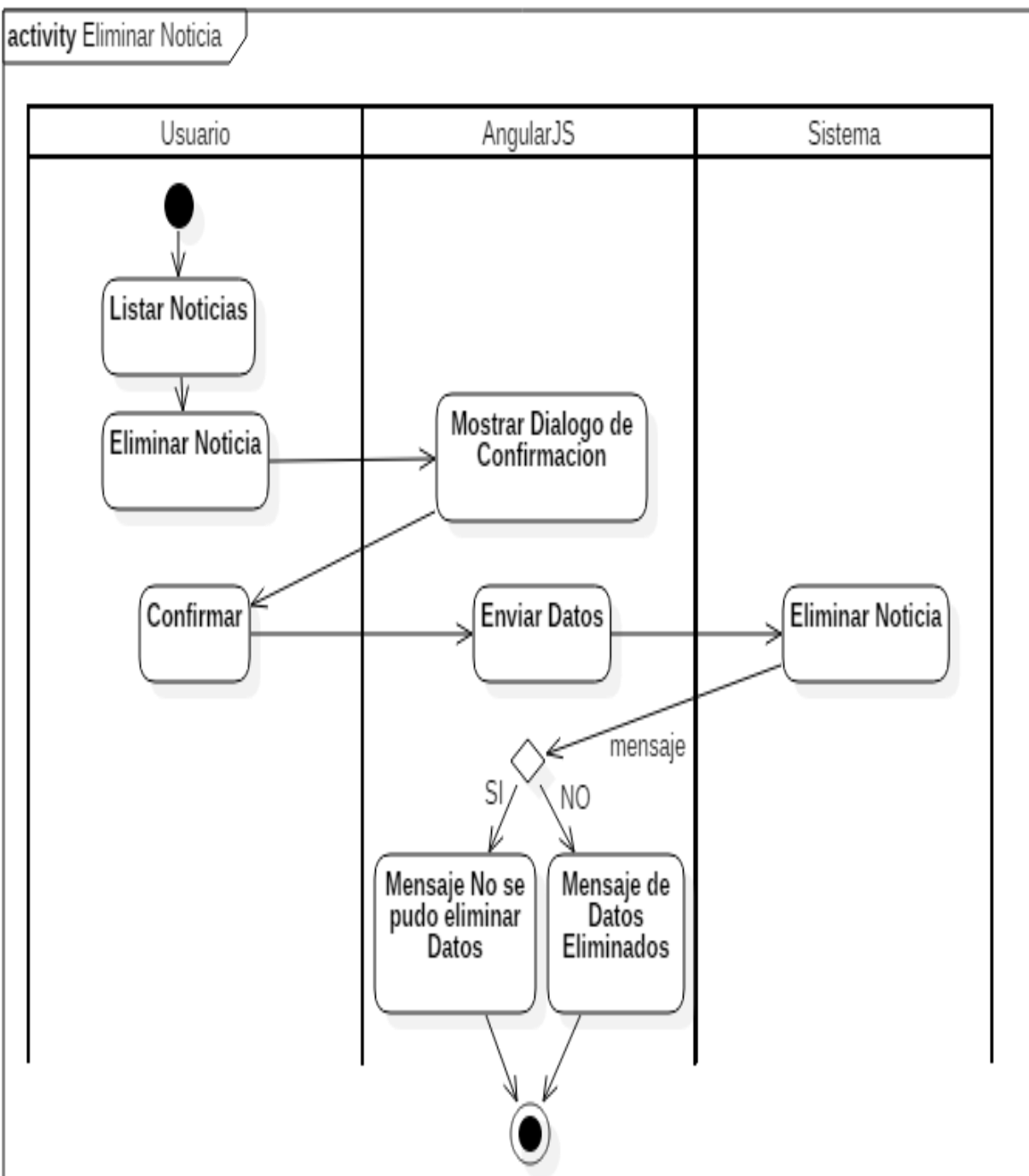


Ilustración 0.109 Diagrama de Actividad Eliminar Noticia

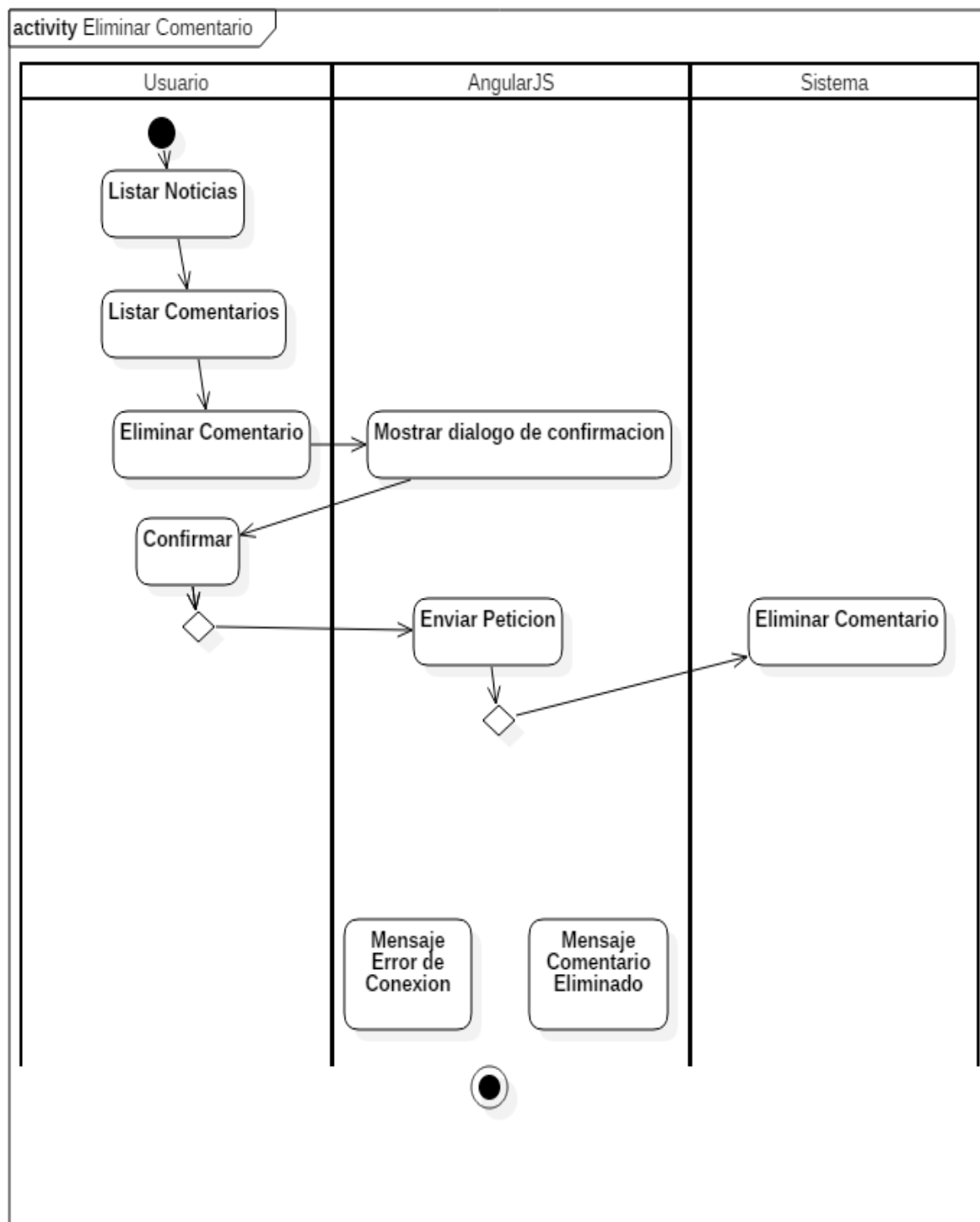


Ilustración 0.110 Diagrama de Actividad Eliminar Comentario

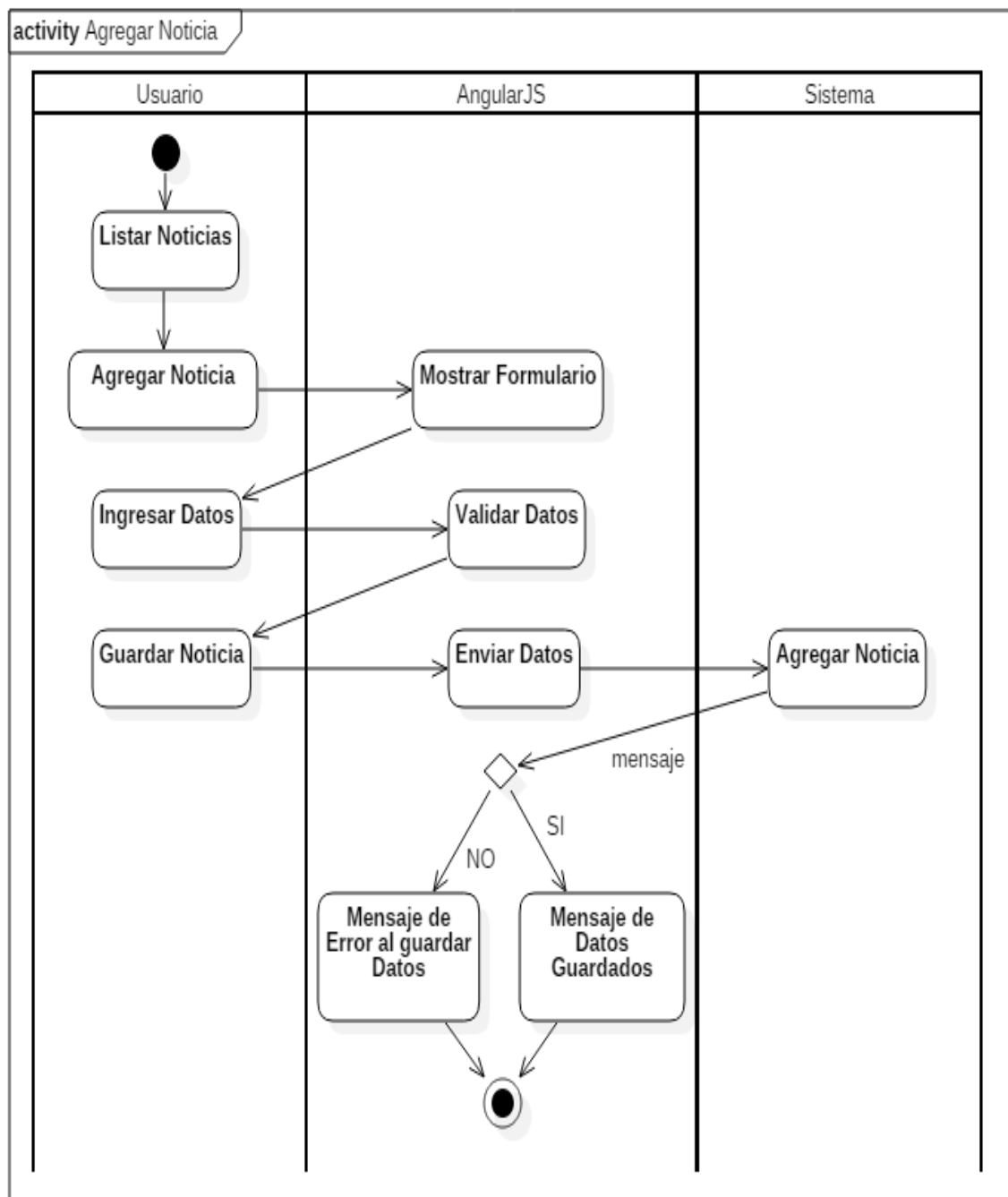


Ilustración 0.111 Diagrama de Actividad Agregar Comentario

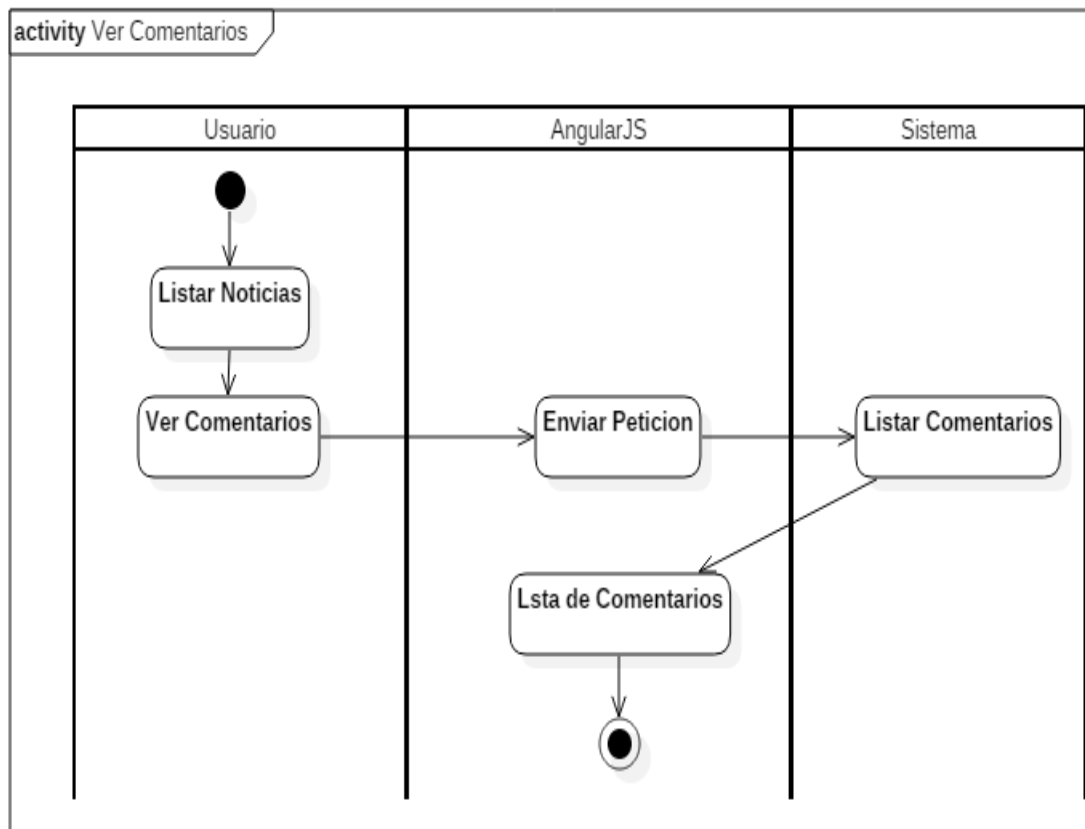


Ilustración 0.112 Diagrama de Actividad Ver Comentarios

I.1.2.2.3.9.2 Modelado de Diagrama de Secuencias

I.1.2.2.3.9.2.1 Introducción

Los diagramas de clases y los de objetos representan información estática. No obstante, en un sistema funcional, los objetos interactúan entre sí, y tales interacciones suceden con el tiempo. El diagrama de secuencias UML muestra la mecánica de la interacción con base en tiempos.

I.1.2.2.3.9.2.1.1 Propósito

- Poner énfasis en el orden y momento en que se envían los mensajes a los objetos.
- Proporcionar un camino a partir de los escenarios para describir las operaciones en una forma más detallada.
- Mostrar la secuencia de comportamiento de un caso de uso.

I.1.2.2.3.9.2.1.2 Alcance

- Mostrar la secuencia de mensajes entre objetos durante un escenario concreto.
- Cada objeto viene dado por una barra vertical.
- El tiempo transcurre de arriba abajo.
- Cuando existe demora entre el envío y la atención se puede indicar usando una línea oblicua.
- Solo se tomaron en cuenta los diagramas donde las secuencias difieren mas de los otros diagramas por que si no se mostrarían secuencias casi similares.

I.1.2.2.3.9.2.2 Diagramas de Secuencia

I.1.2.2.3.9.2.2.1 Diagrama de secuencia iniciar Sesión

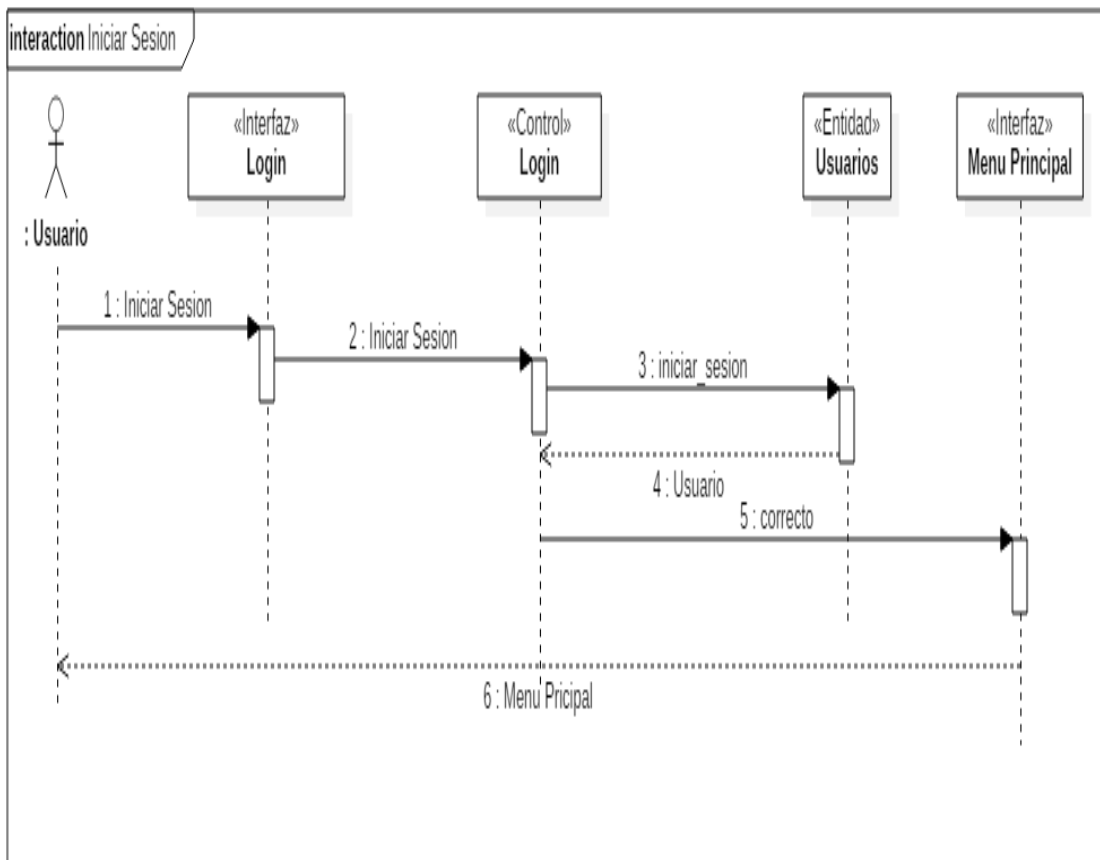


Ilustración 0.113 Diagrama de Secuencia iniciar Sesión

I.1.2.2.3.9.2.2.2 Diagrama de Secuencia Gestionar roles

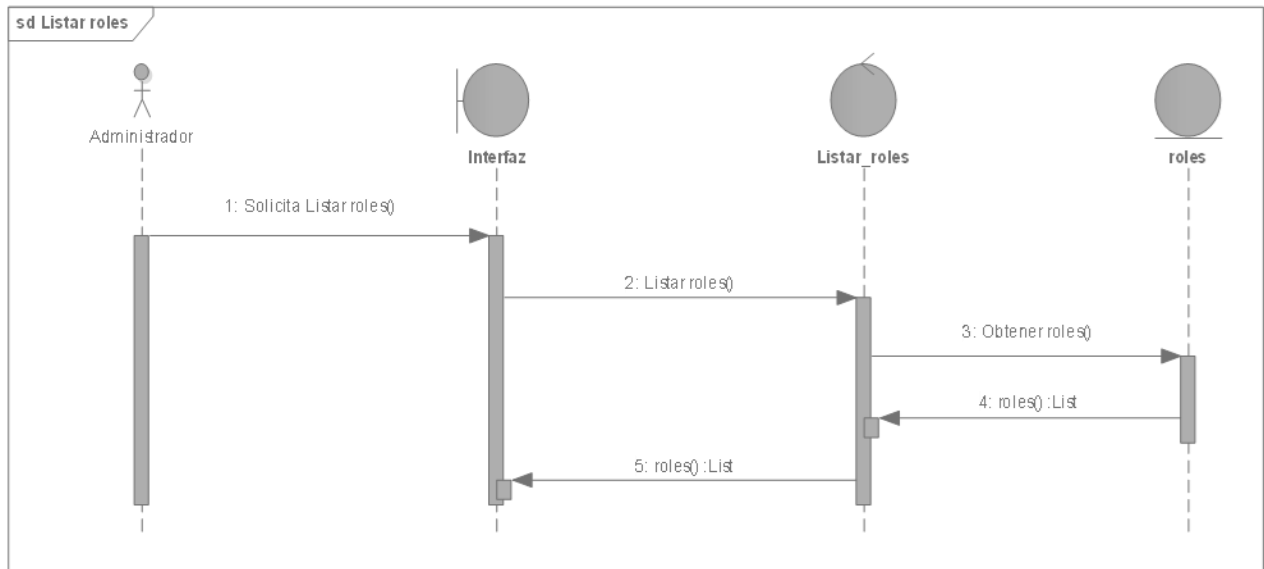


Ilustración 0.114 Secuencia listar roles

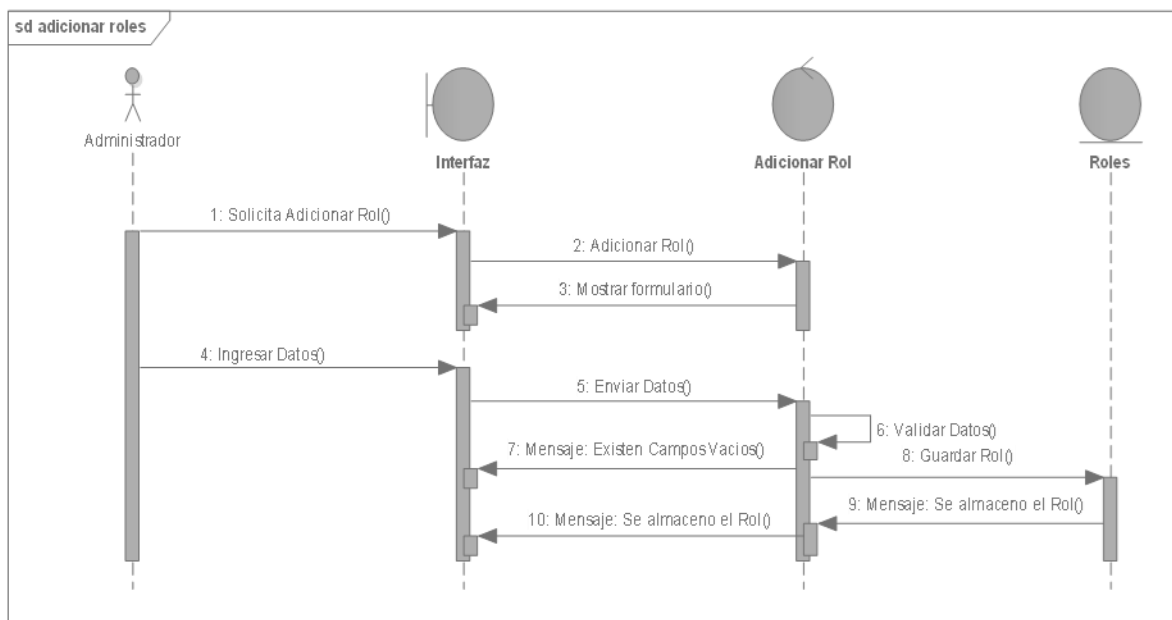


Ilustración 0.115 Secuencia agregar rol

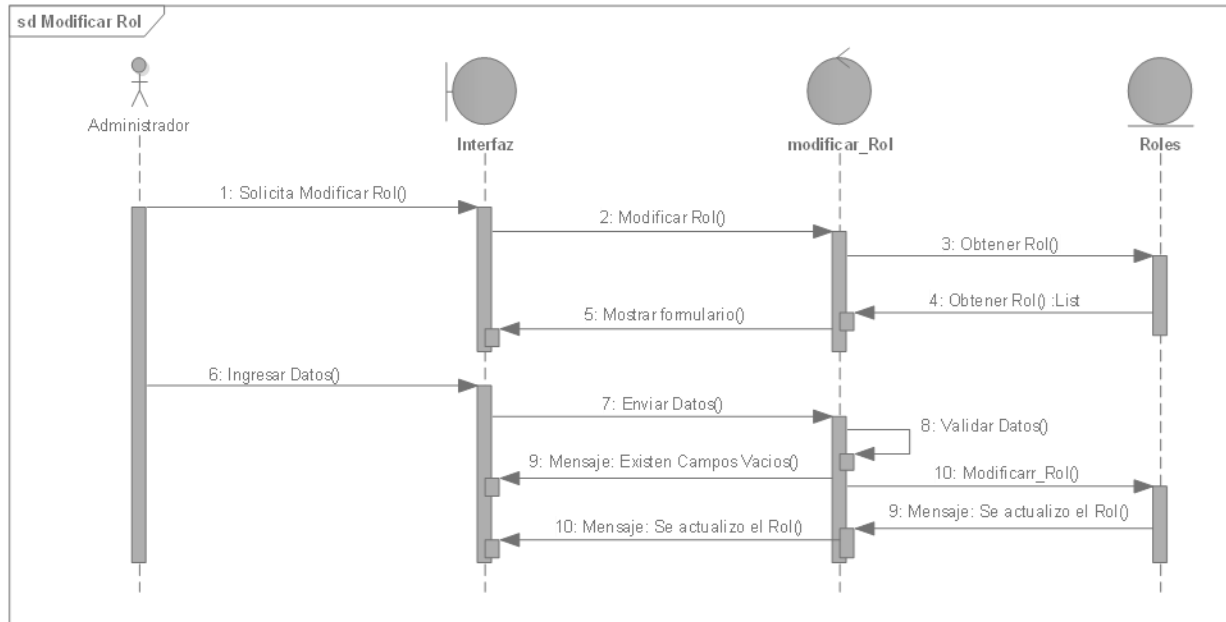


Ilustración 0.116 Secuencia modificar Rol

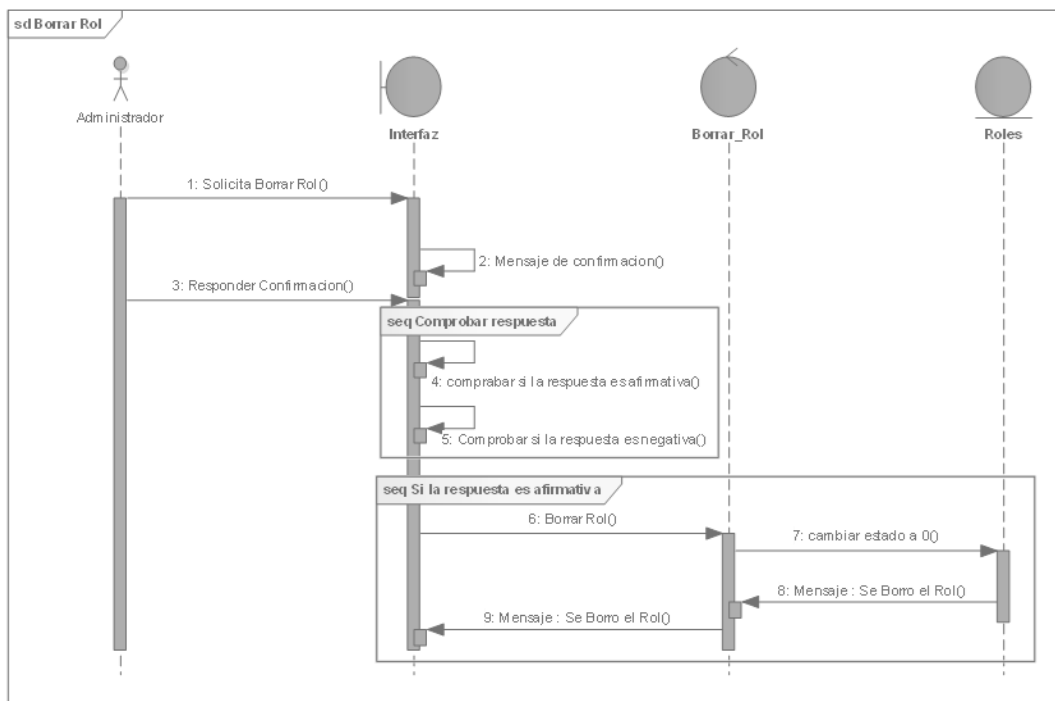


Ilustración 0.117 Secuencia eliminar rol

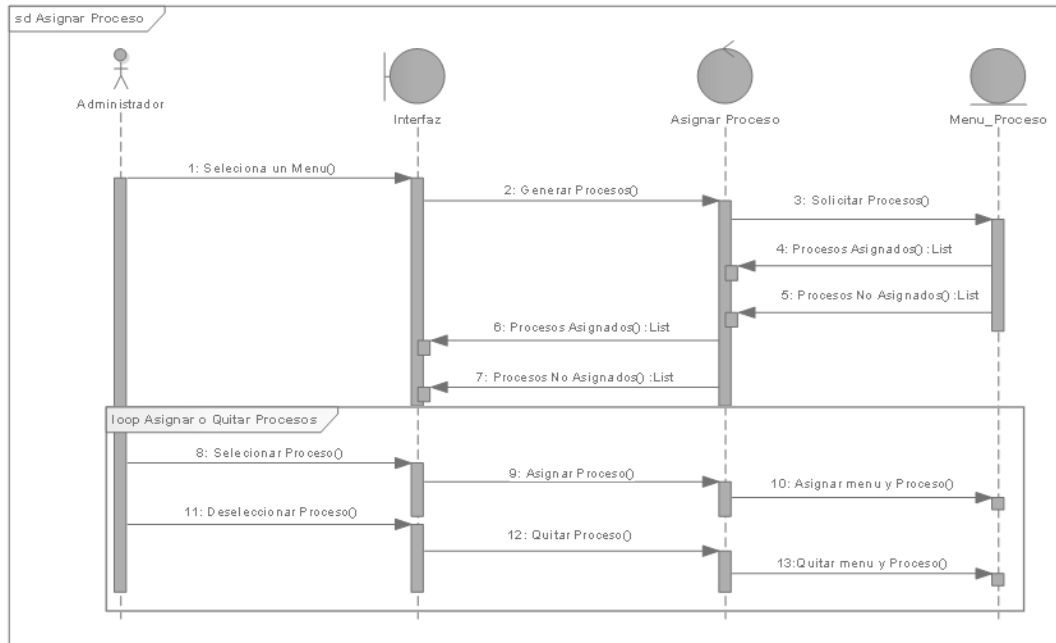


Ilustración 0.118 Secuencia Asignar y quitar un proceso

I.1.2.2.3.9.2.2.3 Diagrama de secuencia Gestionar usuarios

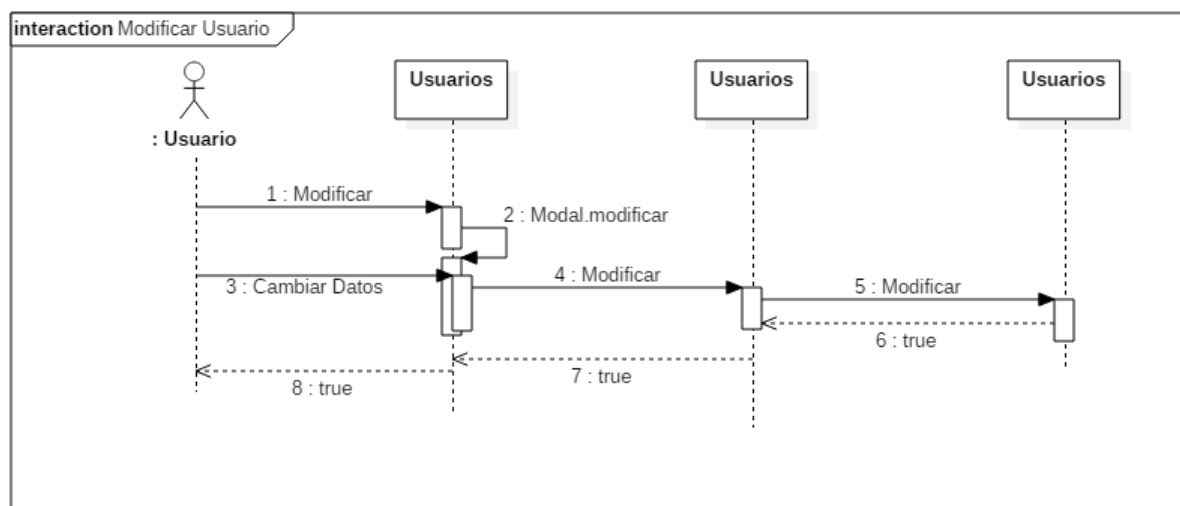


Ilustración 0.119 Diagrama de Secuencia Modificar Usuario

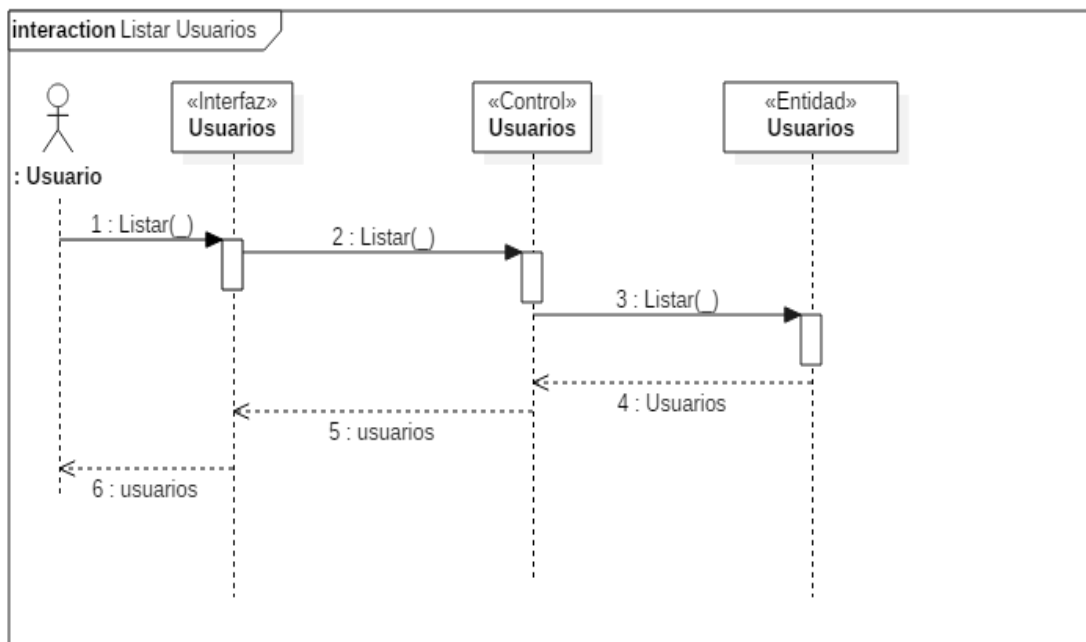


Ilustración 0.120 Diagrama de Secuencia Listar Usuarios

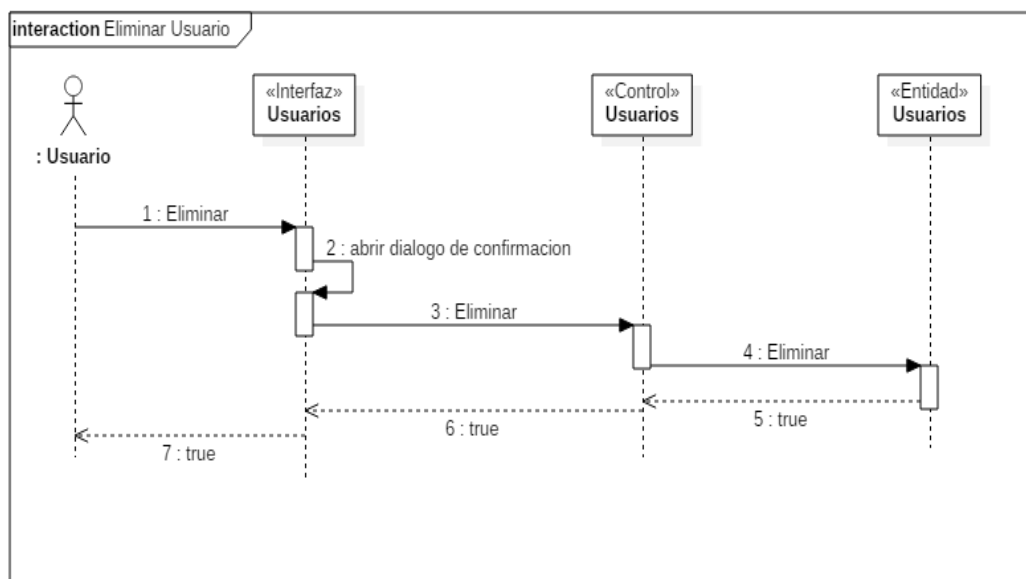


Ilustración 0.121 Diagrama de Secuencia Eliminar Usuario

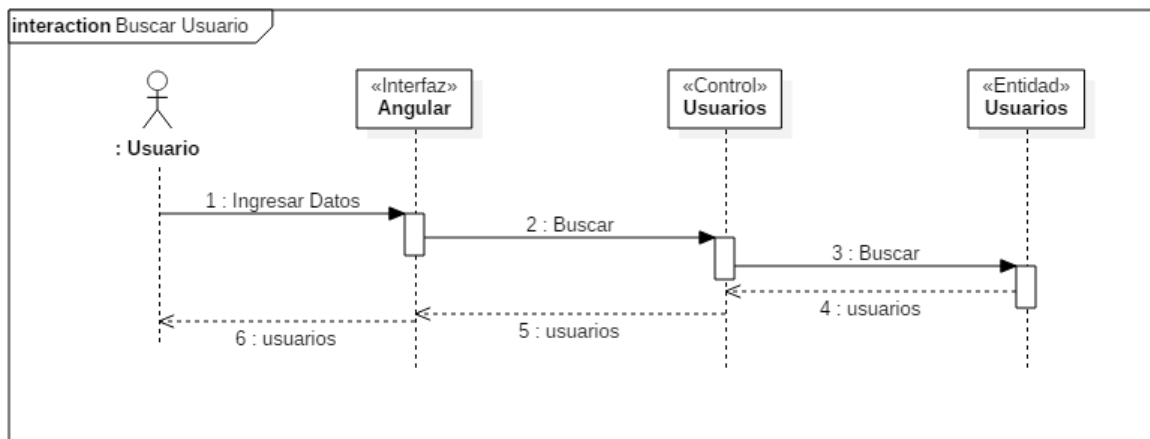


Ilustración 0.122 Diagrama de Secuencia Buscar Usuario

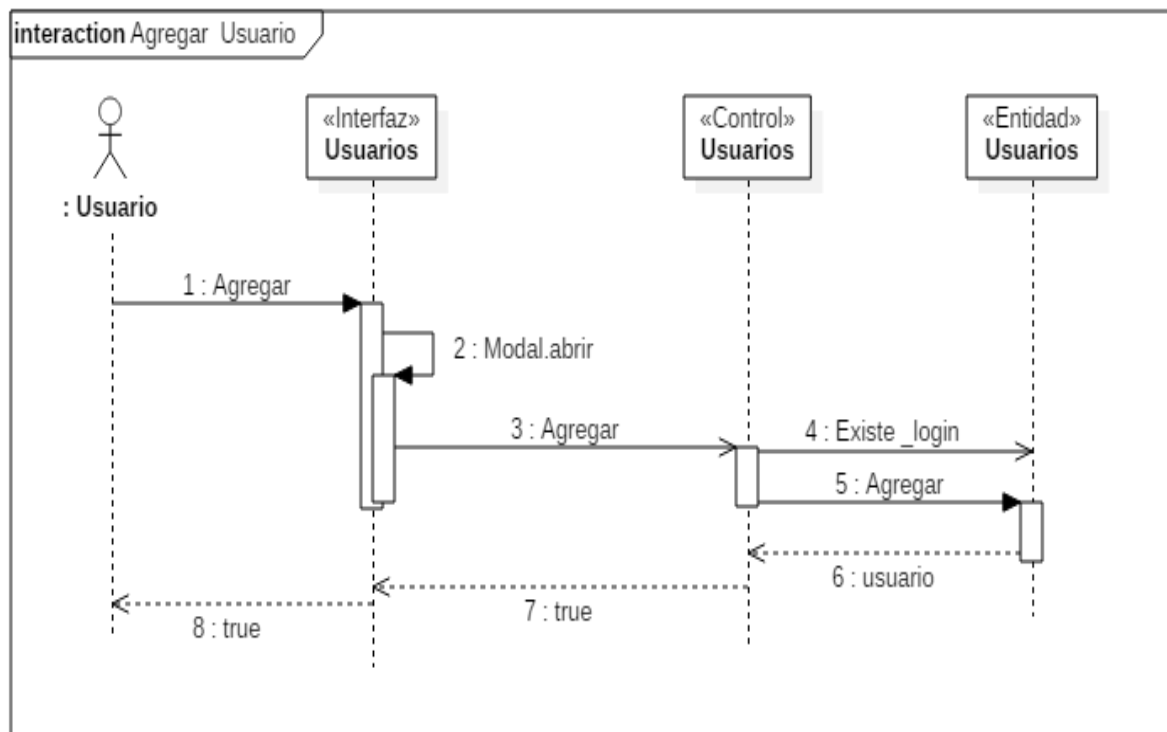


Ilustración 0.123 Diagrama de Secuencia Agregar Usuario

I.1.2.2.3.9.2.2.4 Diagrama de secuencia Gestionar Clientes

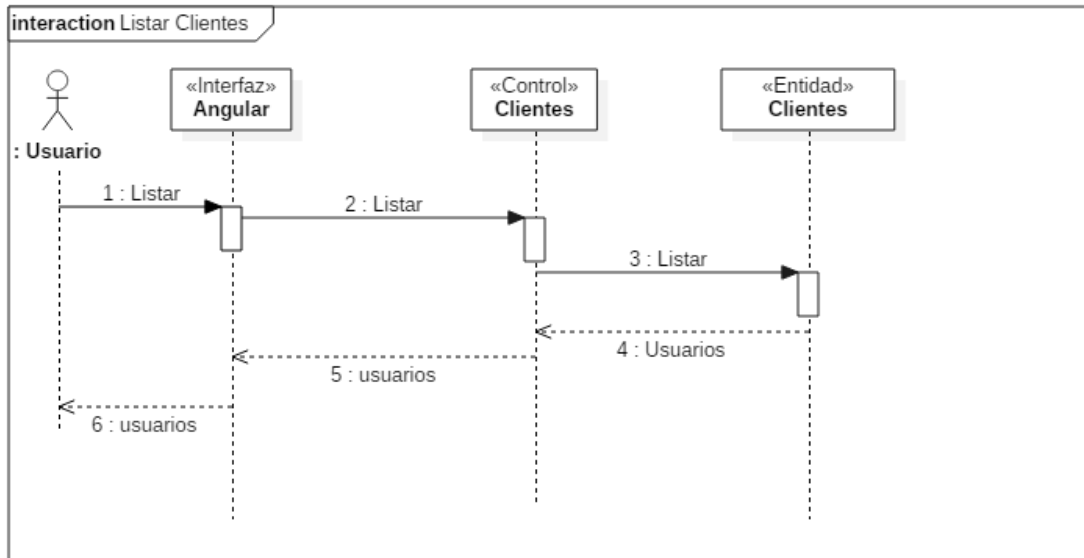


Ilustración 0.124 Diagrama de Secuencia Listar Clientes

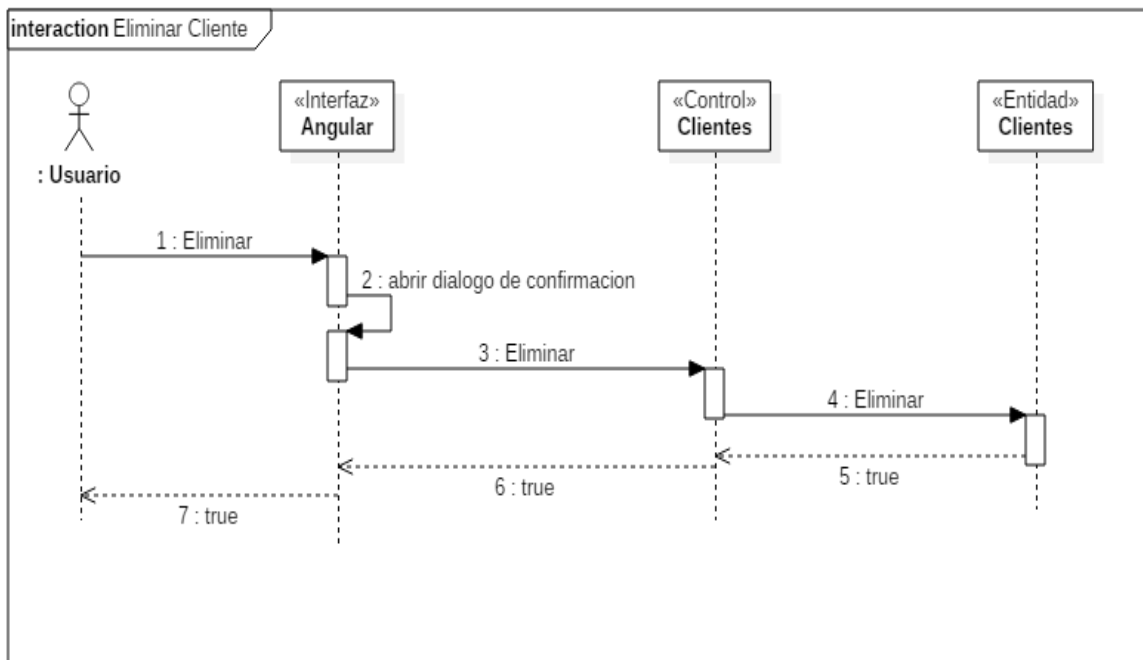


Ilustración 0.125 Diagrama de Secuencia Eliminar Cliente

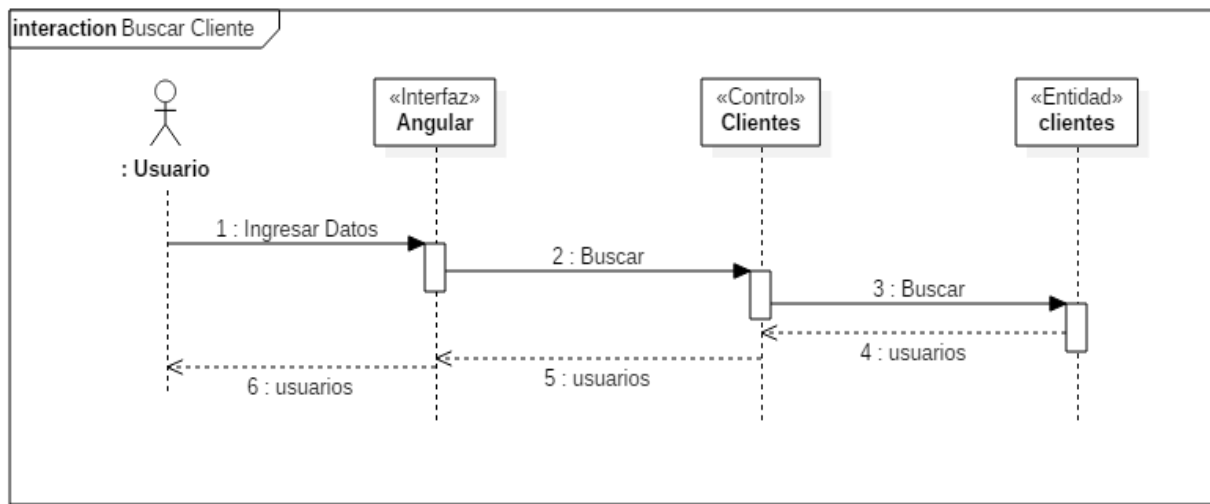


Ilustración 0.126 Diagrama de Secuencia Buscar Cliente

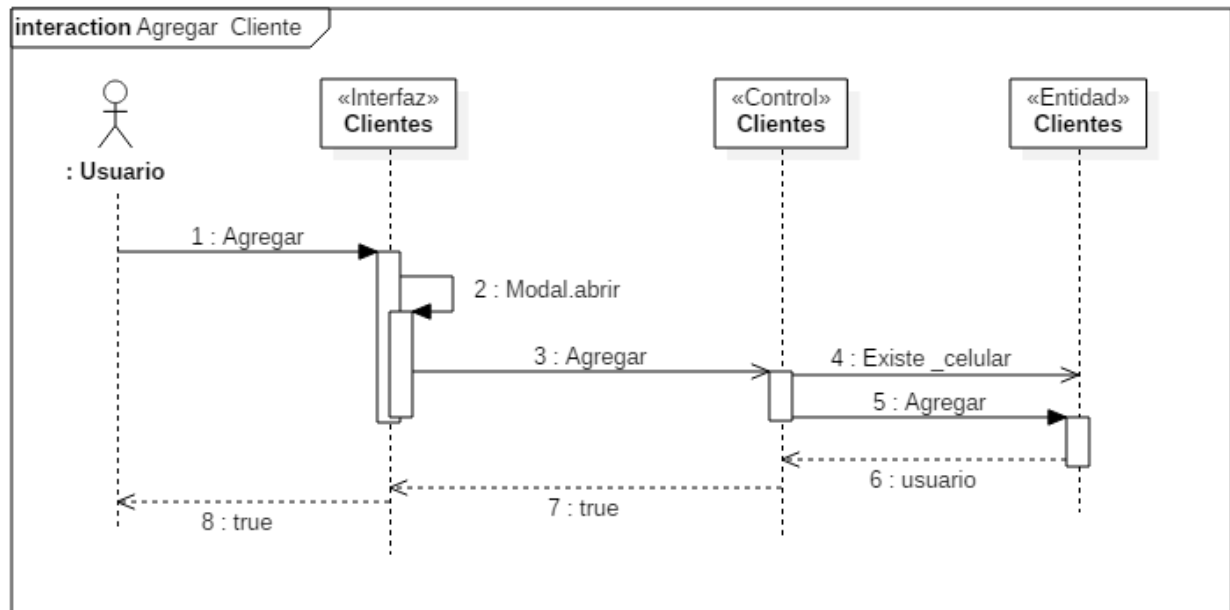


Ilustración 0.127 Diagrama de Secuencia Agregar Cliente

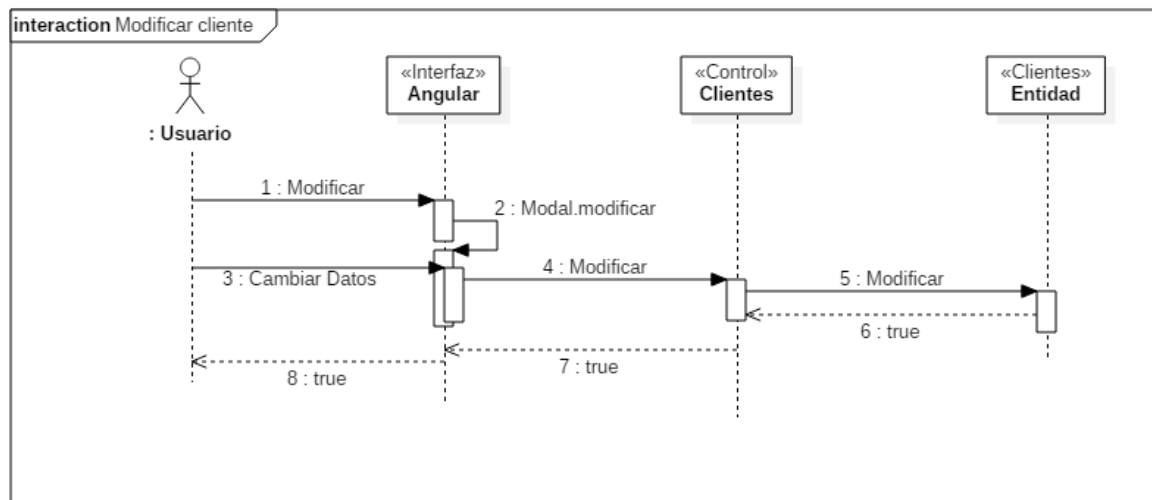


Ilustración 0.128 Diagrama de Secuencia Modificar Cliente

I.1.2.2.3.10 Modelo de Datos

I.1.2.2.3.10.1 Introducción

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

I.1.2.2.3.10.1.1 Propósito

El propósito de esto es dar a conocer la estructura lógica del sistema

I.1.2.2.3.10.1.2 Alcance

Estructura lógica del sistema

Diagrama de Clases

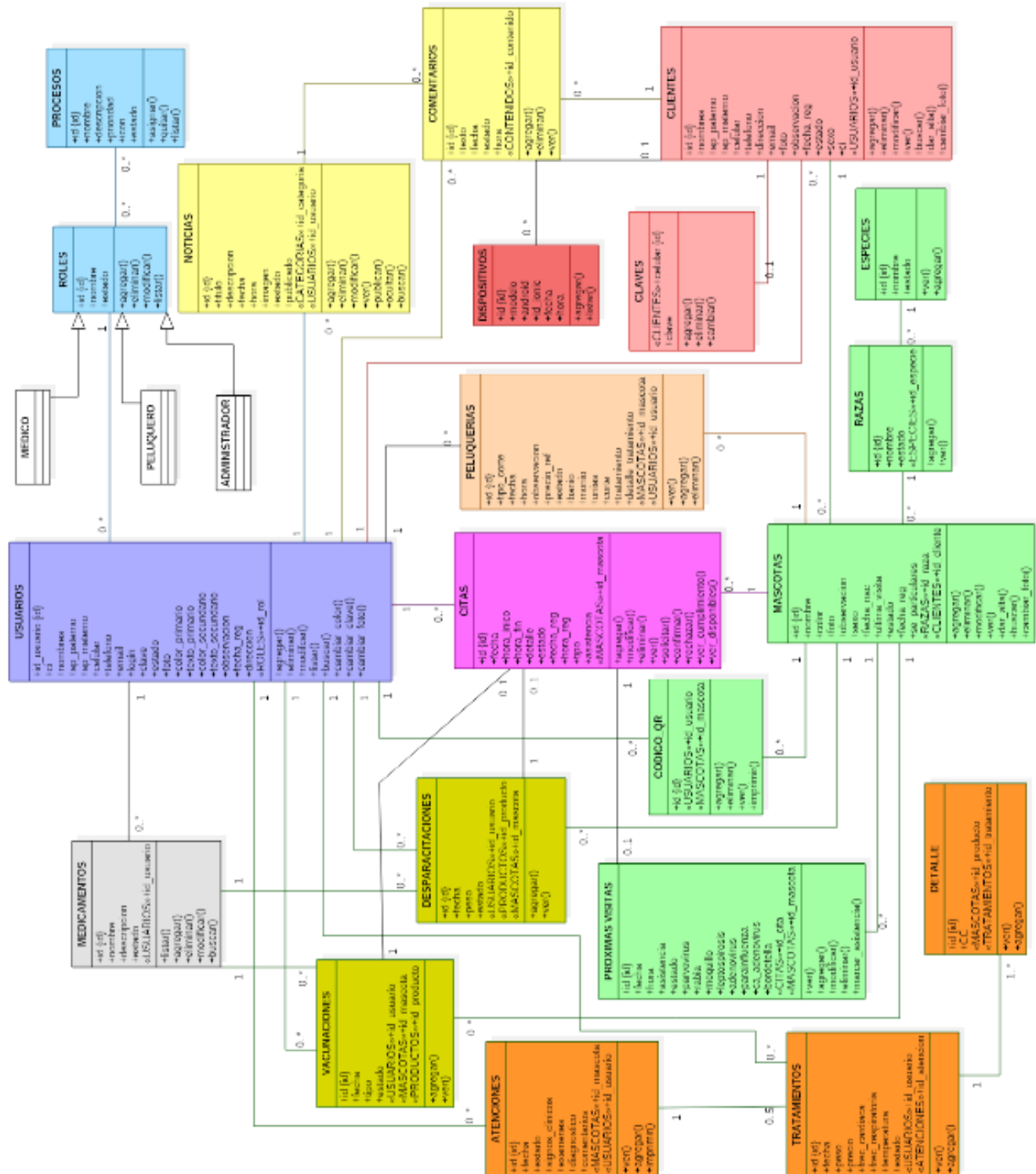


Ilustración 0.129 Diagrama de Clases

I.1.2.2.3.10.2 Modelo de la Base de Datos

I.1.2.2.3.10.3 Modelo Entidad Relación de la Base de Datos

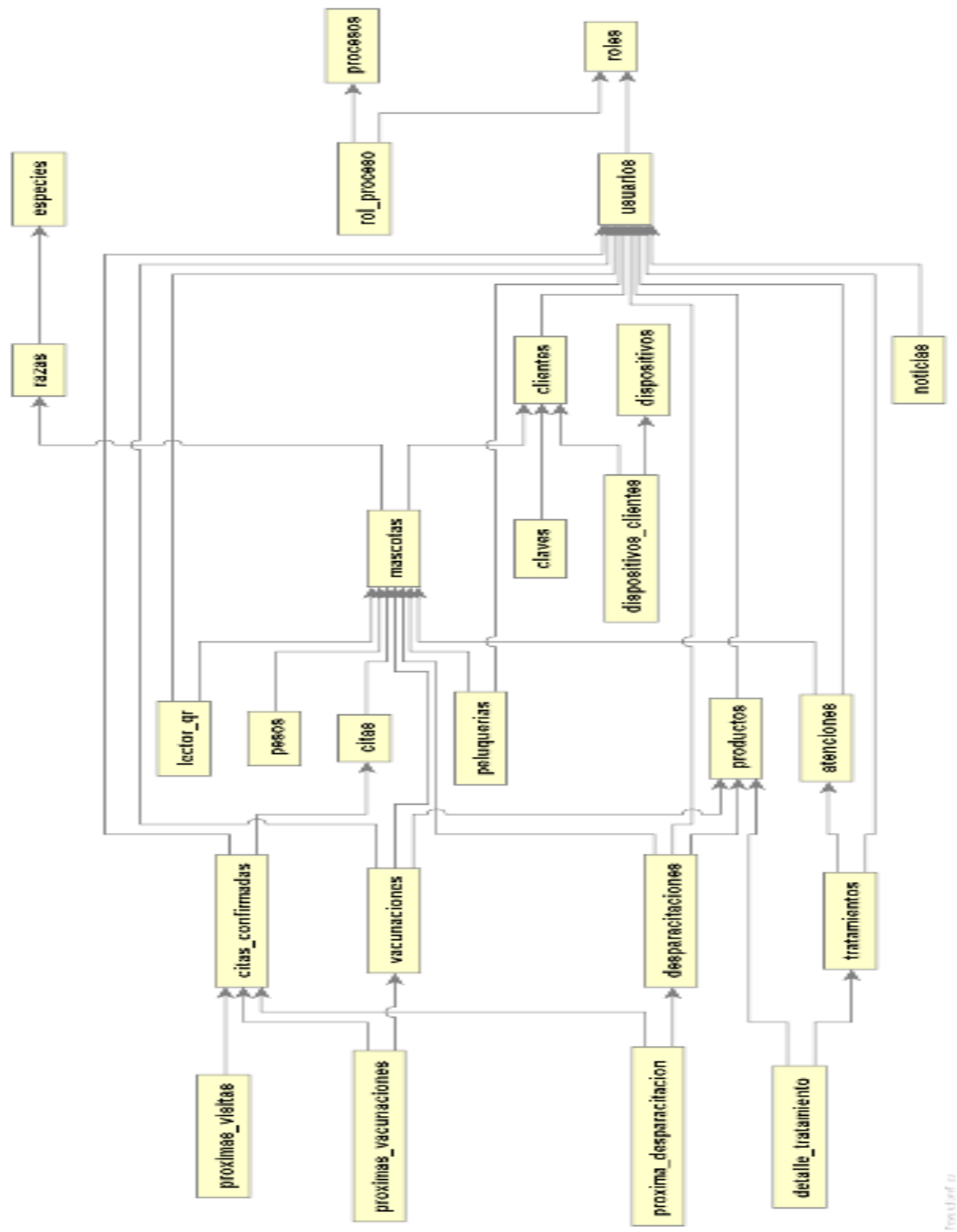


Ilustración 0.130 Diagrama E-R de la BD

I.1.2.2.3.10.4 Modelo Relacional de la base de datos

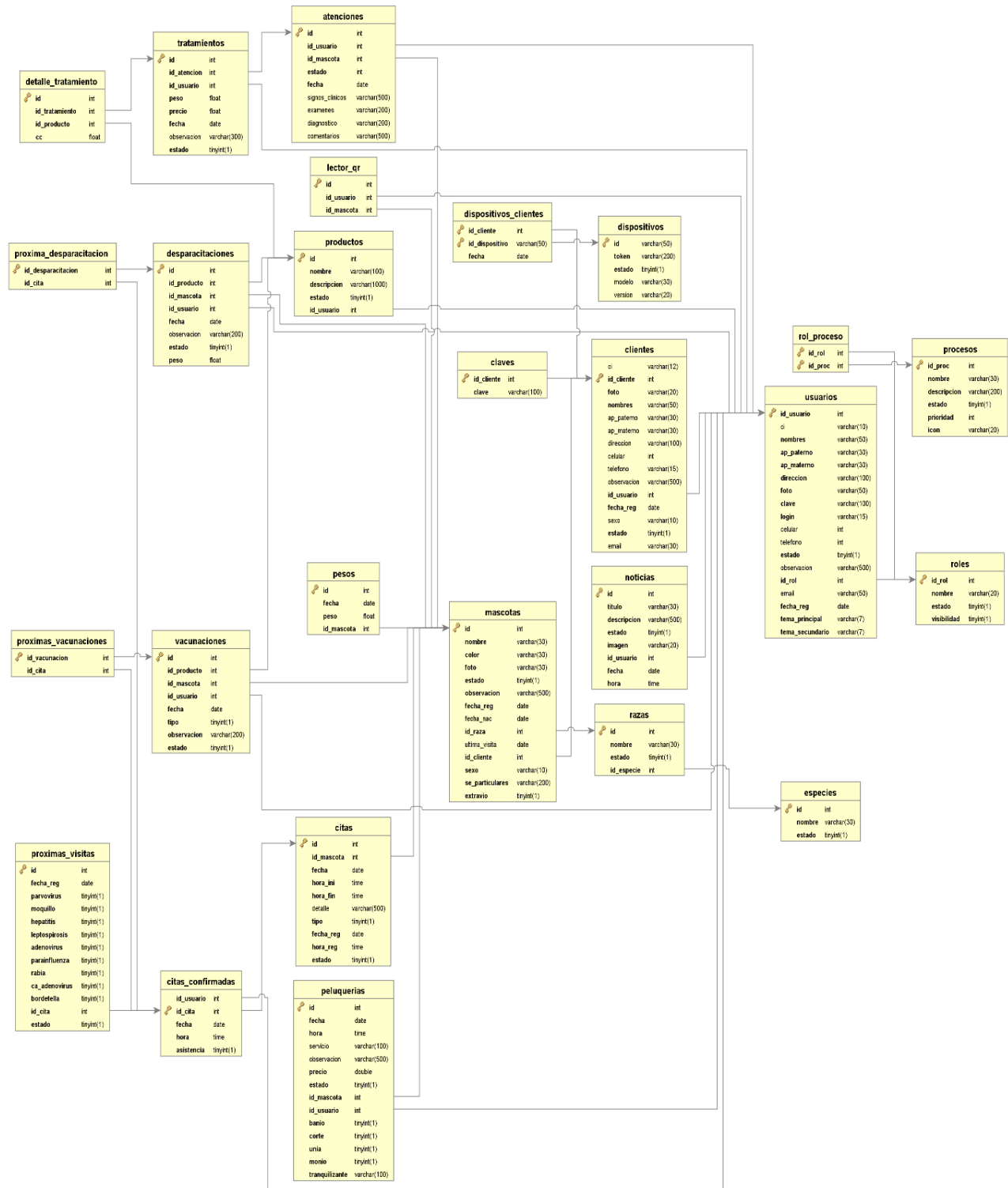


Ilustración 0.131 Modelo Relacional de la BD

1.1.2.2.3.10.5 Creación de la Tablas, Procedimientos, Funciones y Vistas

```
DELIMITER $$
```

```
CREATE PROCEDURE `actualizar_dispositivo`(IN  
`xid` VARCHAR(50), IN `xtoken` VARCHAR(200), IN  
`xmodelo` VARCHAR(20), IN `xversion`  
VARCHAR(10))
```

```
NO SQL
```

```
BEGIN
```

```
declare existe_id INT;
```

```
SELECT COUNT(*) into existe_id FROM dispositivos  
WHERE id=xid;
```

```
IF (existe_id=0) THEN
```

```
INSERT INTO `dispositivos`(`id`, `token`, `modelo`,  
`version`) VALUES (xid,xtoken,xmodelo,xversion);
```

```
ELSE
```

```
UPDATE `dispositivos` SET  
`token`=xtoken,`modelo`=xmodelo,`version`=xversi  
on WHERE id=xid;
```

```
END IF;
```

```
END$$
```

```
CREATE PROCEDURE `agregar_cita`(IN  
`xid_mascota` INT, IN `xtipo` INT, IN `xfecha` DATE,  
IN `xhora_ini` TIME, IN `xhora_fin` TIME, IN  
`xfecha_reg` DATE, IN `xhora_reg` TIME, IN  
`xdetalle` VARCHAR(500), IN `xid_usuario` INT)
```

```
NO SQL
```

```
BEGIN
```

```
declare xid_cita INT;
```

```
INSERT INTO `citas`(`id_mascota`, `fecha`,  
`hora_ini`,`hora_fin`, `detalle`, `tipo`, `fecha_reg`,  
`hora_reg`) VALUES  
(xid_mascota,xfecha,xhora_ini,xhora_fin,xdetalle,xti  
po,xfecha_reg,xhora_reg);
```

```
SELECT MAX(id) into xid_cita FROM citas;
```

```
INSERT INTO `citas_confirmadas`(`id_usuario`,  
`id_cita`, `fecha`, `hora`, `asistencia`) VALUES  
(xid_usuario,xid_cita,xfecha_reg,xhora_reg,0);
```

```
SELECT xid_cita;
```

```
END$$
```

```
CREATE FUNCTION `agregar_cita`(`xid_mascota`  
INT, `xtipo` INT, `xfecha` INT, `xhora_ini` INT,  
`xhora_fin` INT, `xfecha_reg` INT, `xhora_reg` INT,  
`xdetalle` INT, `xid_usuario` INT) RETURNS int(11)
```

```
NO SQL
```

```
BEGIN
```

```
declare xid_cita INT;
```

```
INSERT INTO `citas`(`id_mascota`, `fecha`,  
`hora_ini`,`hora_fin`, `detalle`, `tipo`, `fecha_reg`,  
`hora_reg`) VALUES  
(xid_mascota,xfecha,xhora_ini,xhora_fin,xdetalle,xti  
po,xfecha_reg,xhora_reg);
```

```
SELECT MAX(id) into xid_cita FROM citas;
```

```
INSERT INTO `citas_confirmadas`(`id_usuario`,
`id_cita`, `fecha`, `hora`, `asistencia`) VALUES
(xid_usuario,xid_cita,xfecha_reg,xhora_reg,0);
```

```
RETURN xid_cita;
```

```
END$$
```

```
DELIMITER ;
```

```
CREATE TABLE IF NOT EXISTS `atenciones` (
```

```
`id` int(11) NOT NULL,
```

```
`id_usuario` int(11) NOT NULL,
```

```
`id_mascota` int(11) NOT NULL,
```

```
`estado` int(11) NOT NULL DEFAULT '1',
```

```
`fecha` date NOT NULL,
```

```
`signos_clinicos` varchar(500) DEFAULT NULL,
```

```
`examenenes` varchar(200) DEFAULT NULL,
```

```
`diagnostico` varchar(200) DEFAULT NULL,
```

```
`comentarios` varchar(500) DEFAULT NULL
```

```
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT
CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `atenciones_view` (
```

```
`id` int(11)
```

```
,`id_usuario` int(11)
```

```
,`id_mascota` int(11)
```

```
,`estado` int(11)
```

```
,`fecha` date
```

```
,`signos_clinicos` varchar(500)
```

```
,`examenenes` varchar(200)
```

```
,`diagnostico` varchar(200)
```

```
,`comentarios` varchar(500)
```

```
,`usuario` varchar(112)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS
```

```
`calendario_vacunacion` (
```

```
`id` int(11) NOT NULL,
```

```
`parvovirus` tinyint(1) NOT NULL DEFAULT '0',
```

```
`moquillo` tinyint(1) NOT NULL DEFAULT '0',
```

```
`hepatitis` tinyint(1) NOT NULL DEFAULT '0',
```

```
`leptospirosis` tinyint(1) NOT NULL DEFAULT '0',
```

```
`adenovirus` tinyint(1) NOT NULL DEFAULT '0',
```

```
`parainfluenza` tinyint(1) NOT NULL DEFAULT '0',
```

```
`rabia` tinyint(1) NOT NULL DEFAULT '0',
```

```
`ca_adenovirus` tinyint(1) NOT NULL DEFAULT '0',
```

```
`bordetella` tinyint(1) NOT NULL DEFAULT '0',
```

```
`estado` tinyint(1) NOT NULL DEFAULT '1',
```

```
`asistencia` tinyint(1) NOT NULL DEFAULT '0',
```

```
`id_mascota` int(11) NOT NULL,
```

```
`id_cita` int(11) NOT NULL
```

```
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT
CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `calendario_view` (
```

```
`fecha` date
```

```
,`hora_ini` time
```

```
,`id` int(11)
```

```
,`parvovirus` tinyint(1)
```

```
,`moquillo` tinyint(1)
```

```
,`hepatitis` tinyint(1)
```

```
,`leptospirosis` tinyint(1)
```

```
,`adenovirus` tinyint(1)
```

```
,`parainfluenza` tinyint(1)
```

```
,`rabia` tinyint(1)
```

```
,`ca_adenovirus` tinyint(1)
```

```
,`bordetella` tinyint(1)
```

```
,`estado` tinyint(1)
```

```
,`asistencia` tinyint(1)
```

```
,`id_mascota` int(11)
```

```
,`id_cita` int(11)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS `celulares` (
```

```
`id_cliente` int(11) NOT NULL,
```

```
`numero` int(11) NOT NULL,
```

```
`estado` tinyint(1) NOT NULL DEFAULT '1'
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `citas` (
```

```
`id` int(11) NOT NULL,
```

```
`id_mascota` int(11) NOT NULL,
```

```
`fecha` date NOT NULL,
```

```
`hora_ini` time NOT NULL,
```

```
`hora_fin` time NOT NULL,
```

```
`detalle` varchar(500) DEFAULT NULL,
```

```
`tipo` tinyint(1) NOT NULL,
```

```
`fecha_reg` date NOT NULL,
```

```
`hora_reg` time NOT NULL,
```

```
`estado` tinyint(1) NOT NULL DEFAULT '1'
```

```
) ENGINE=InnoDB AUTO_INCREMENT=49
```

```
DEFAULT CHARSET=latin1 COMMENT='citas  
realizadas a la clinica';
```

```
CREATE TABLE IF NOT EXISTS `citas_confirmadas` (
```

```
`id_usuario` int(11) NOT NULL,
```

```
`id_cita` int(11) NOT NULL,
```

```
`fecha` date NOT NULL,
```

```
`hora` time NOT NULL,
```

```
`asistencia` tinyint(1) NOT NULL DEFAULT '0'
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```

CREATE TABLE IF NOT EXISTS `citas_solicitudes` (
  `id_cliente` int(11)
, `foto` varchar(20)
, `cliente` varchar(112)
, `id` int(11)
, `id_mascota` int(11)
, `fecha` date
, `hora_ini` time
, `hora_fin` time
, `detalle` varchar(500)
, `tipo` tinyint(1)
, `fecha_reg` date
, `hora_reg` time
, `estado` tinyint(1)
, `mascota` varchar(30)
, `raza` varchar(30)
, `especie` varchar(30)
);

```

```

CREATE TABLE IF NOT EXISTS `citas_view` (
  `raza` varchar(30)
, `especie` varchar(30)
, `foto` varchar(20)

```

```

, `usuario` varchar(112)
, `mascota` varchar(30)
, `id_cliente` int(11)
, `cliente` varchar(112)
, `id` int(11)
, `id_mascota` int(11)
, `fecha` date
, `hora_ini` time
, `hora_fin` time
, `detalle` varchar(500)
, `tipo` tinyint(1)
, `fecha_reg` date
, `hora_reg` time
, `estado` tinyint(1)
, `asistencia` tinyint(1)
, `hora_confirmacion` time
, `fecha_confirmacion` date
);

```

```

CREATE TABLE IF NOT EXISTS `claves` (
  `id_cliente` int(11) NOT NULL,
  `clave` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE IF NOT EXISTS `clientes` (
  `ci` varchar(12) DEFAULT NULL,
  `id_cliente` int(11) NOT NULL,
  `foto` varchar(20) NOT NULL DEFAULT
'defecto.png',
  `nombres` varchar(50) NOT NULL DEFAULT "",
  `ap_paterno` varchar(30) DEFAULT "",
  `ap_materno` varchar(30) DEFAULT "",
  `direccion` varchar(100) DEFAULT NULL,
  `telefono` varchar(15) DEFAULT NULL,
  `observacion` varchar(500) DEFAULT NULL,
  `id_usuario` int(11) NOT NULL,
  `fecha_reg` date NOT NULL,
  `sexo` varchar(10) DEFAULT NULL,
  `estado` tinyint(1) NOT NULL DEFAULT '1',
  `email` varchar(30) DEFAULT NULL
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT
CHARSET=latin1;

```

```

CREATE TABLE IF NOT EXISTS `clientes_view` (
  `ci` varchar(12)
, `id_cliente` int(11)
, `foto` varchar(20)
, `nombres` varchar(50)

```

```

, `ap_paterno` varchar(30)
, `ap_materno` varchar(30)
, `direccion` varchar(100)
, `telefono` varchar(15)
, `observacion` varchar(500)
, `id_usuario` int(11)
, `fecha_reg` date
, `sexo` varchar(10)
, `estado` tinyint(1)
, `email` varchar(30)
, `datos` bigint(21)
);

```

```

CREATE TABLE IF NOT EXISTS `desparasitaciones` (
  `id` int(11) NOT NULL,
  `id_producto` int(11) NOT NULL,
  `id_mascota` int(11) NOT NULL,
  `id_usuario` int(11) NOT NULL,
  `fecha` date NOT NULL,
  `observacion` varchar(200) DEFAULT NULL,
  `estado` tinyint(1) NOT NULL DEFAULT '1',
  `peso` float NOT NULL DEFAULT '0',
  `id_cita` int(11) NOT NULL

```

```
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT
CHARSET=latin1 COMMENT='Las desparasitaciones
que recibe una mascota';
```

```
CREATE TABLE IF NOT EXISTS
`desparasitaciones_view` (
`usuario` varchar(112)
,`producto` varchar(100)
,`proxima_fecha` date
,`hora_ini` time
,`id` int(11)
,`id_producto` int(11)
,`id_mascota` int(11)
,`id_usuario` int(11)
,`observacion` varchar(200)
,`estado` tinyint(1)
,`fecha` date
,`peso` float
,`id_cita` int(11)
);
```

```
CREATE TABLE IF NOT EXISTS
`detalle_tratamiento` (
`id` int(11) NOT NULL,
`id_tratamiento` int(11) NOT NULL,
```

```
`id_producto` int(11) NOT NULL,
`cc` float NOT NULL
```

```
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT
CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS
`detalle_tratamiento_view` (
`id` int(11)
,`id_tratamiento` int(11)
,`id_producto` int(11)
,`cc` float
,`producto` varchar(100)
);
```

```
CREATE TABLE IF NOT EXISTS `dispositivos` (
`id` varchar(50) NOT NULL,
`token` varchar(200) NOT NULL,
`estado` tinyint(1) NOT NULL DEFAULT '1',
`modelo` varchar(30) DEFAULT 'desconocido',
`version` varchar(20) DEFAULT 'desconocido'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS
`dispositivos_clientes` (
`id_cliente` int(11) NOT NULL,
```

```

`id_dispositivo` varchar(50) NOT NULL,

`fecha` date NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE IF NOT EXISTS `especies` (

`id` int(11) NOT NULL,

`nombre` varchar(30) NOT NULL,

`estado` tinyint(1) NOT NULL DEFAULT '1'

) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT
CHARSET=latin1;

```

```

INSERT INTO `especies` (`id`, `nombre`, `estado`)
VALUES

(1, 'CANINO', 1),

(2, 'FELINO', 1),

(3, 'AVE', 1),

(4, 'PORCINO', 1),

(5, 'VACUNO', 1);

```

```

CREATE TABLE IF NOT EXISTS `extravios` (

`id` int(11) NOT NULL,

`id_mascota` int(11) NOT NULL,

`fecha_reg` date NOT NULL,

`detalle` varchar(500) NOT NULL,

```

```

`recompensa` varchar(50) NOT NULL DEFAULT 'Se
ofrecera una recompensa'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE IF NOT EXISTS `extravios_view` (

`id` int(11)

,`id_mascota` int(11)

,`fecha_reg` date

,`detalle` varchar(500)

,`recompensa` varchar(50)

,`id_cliente` int(11)

,`mascota` varchar(30)

,`color` varchar(30)

,`foto` varchar(30)

,`fecha_nac` date

,`se_particulares` varchar(200)

,`raza` varchar(30)

,`especie` varchar(30)

,`sexo` varchar(10)

,`cliente` varchar(112)

,`foto_cliente` varchar(20)

,`telefono` varchar(15)

,`direccion` varchar(100)

,`email` varchar(30)

```

```
);
```

```
CREATE TABLE IF NOT EXISTS `lector_qr` (
```

```
  `id` int(11) NOT NULL,
```

```
  `id_usuario` int(11) NOT NULL,
```

```
  `id_mascota` int(11) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `lector_qr_view` (
```

```
  `id` int(11)
```

```
, `id_usuario` int(11)
```

```
, `id_mascota` int(11)
```

```
, `nombre` varchar(30)
```

```
, `cliente` varchar(112)
```

```
, `foto` varchar(30)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS `mascotas` (
```

```
  `id` int(11) NOT NULL,
```

```
  `nombre` varchar(30) NOT NULL,
```

```
  `color` varchar(30) NOT NULL,
```

```
  `foto` varchar(30) NOT NULL DEFAULT  
'defecto.png',
```

```
  `estado` tinyint(1) NOT NULL DEFAULT '1',
```

```
  `observacion` varchar(500) NOT NULL,
```

```
  `fecha_reg` date NOT NULL,
```

```
  `fecha_nac` date DEFAULT NULL,
```

```
  `id_raza` int(11) NOT NULL,
```

```
  `ultima_visita` date DEFAULT NULL,
```

```
  `id_cliente` int(11) NOT NULL,
```

```
  `sexo` varchar(10) NOT NULL,
```

```
  `se_particulares` varchar(200) NOT NULL
```

```
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT  
CHARSET=latin1 COMMENT='tabla donde se  
almacenan todas las mascotas de los clientes';
```

```
CREATE TABLE IF NOT EXISTS `mascotas_view` (
```

```
  `foto_c` varchar(20)
```

```
, `id` int(11)
```

```
, `id_especie` int(11)
```

```
, `nombre` varchar(30)
```

```
, `color` varchar(30)
```

```
, `foto` varchar(30)
```

```
, `estado` tinyint(1)
```

```
, `observacion` varchar(500)
```

```
, `fecha_reg` date
```

```
, `fecha_nac` date
```

```
, `id_raza` int(11)
```

```
,`ultima_visita` date
```

```
,`se_particulares` varchar(200)
```

```
,`id_cliente` int(11)
```

```
,`sexo` varchar(10)
```

```
,`cliente` varchar(112)
```

```
,`telefono` varchar(15)
```

```
,`direccion` varchar(100)
```

```
,`especie` varchar(30)
```

```
,`raza` varchar(30)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS `noticias` (
```

```
  `id` int(11) NOT NULL,
```

```
  `titulo` varchar(30) NOT NULL,
```

```
  `descripcion` varchar(500) NOT NULL,
```

```
  `estado` tinyint(1) NOT NULL DEFAULT '1',
```

```
  `imagen` varchar(20) NOT NULL DEFAULT  
'defecto.png',
```

```
  `id_usuario` int(11) NOT NULL,
```

```
  `fecha` date NOT NULL,
```

```
  `hora` time NOT NULL
```

```
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT  
CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `noticias_view` (
```

```
  `foto` varchar(50)
```

```
, `usuario` varchar(112)
```

```
, `id` int(11)
```

```
, `titulo` varchar(30)
```

```
, `descripcion` varchar(500)
```

```
, `estado` tinyint(1)
```

```
, `imagen` varchar(20)
```

```
, `id_usuario` int(11)
```

```
, `fecha` date
```

```
, `hora` time
```

```
);
```

```
CREATE TABLE IF NOT EXISTS `peluqueras` (
```

```
  `id` int(11) NOT NULL,
```

```
  `fecha` date NOT NULL,
```

```
  `hora` time NOT NULL,
```

```
  `servicio` varchar(100) DEFAULT NULL,
```

```
  `observacion` varchar(500) DEFAULT NULL,
```

```
  `precio` double NOT NULL DEFAULT '0',
```

```
  `estado` tinyint(1) NOT NULL DEFAULT '1',
```

```
  `id_mascota` int(11) NOT NULL,
```

```
  `id_usuario` int(11) NOT NULL,
```

```
  `banio` tinyint(1) NOT NULL DEFAULT '0',
```

```

`corte` tinyint(1) NOT NULL DEFAULT '0',
`unia` tinyint(1) NOT NULL DEFAULT '0',
`monio` tinyint(1) NOT NULL DEFAULT '0',
`tranquilizante` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT
CHARSET=latin1;

```

```

CREATE TABLE IF NOT EXISTS `peluqueras_view` (
`id` int(11)
,`fecha` date
,`hora` time
,`servicio` varchar(100)
,`observacion` varchar(500)
,`precio` double
,`estado` tinyint(1)
,`id_mascota` int(11)
,`id_usuario` int(11)
,`banio` tinyint(1)
,`corte` tinyint(1)
,`unia` tinyint(1)
,`monio` tinyint(1)
,`tranquilizante` tinyint(1)
,`mascota` varchar(30)
,`cliente` varchar(112)

```

```

,`usuario` varchar(112)
);

```

```

CREATE TABLE IF NOT EXISTS `pesos` (
`id` int(11) NOT NULL,
`fecha` date NOT NULL,
`peso` float NOT NULL DEFAULT '0',
`id_mascota` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE IF NOT EXISTS `procesos` (
`id_proc` int(11) NOT NULL,
`nombre` varchar(30) NOT NULL,
`descripcion` varchar(200) NOT NULL,
`estado` tinyint(1) NOT NULL DEFAULT '1',
`prioridad` int(11) NOT NULL DEFAULT '0',
`icon` varchar(20) NOT NULL DEFAULT 'mdi-cat'
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT
CHARSET=latin1;

```

```

INSERT INTO `procesos` (`id_proc`, `nombre`,
`descripcion`, `estado`, `prioridad`, `icon`) VALUES
(1, 'Noticias', 'Aqui puede gestionar las noticias que se
publicaran en la aplicacion movil de los clientes.', 1, 4,
'mdi-newspaper'),

```

(2, 'Roles', 'Aqui puede gestionar los roles y permisos de los usuarios', 1, 10, 'mdi-lock'),

(3, 'Clientes', 'Aqui úede gestionar la informacion de los clientes de la veterinaria', 1, 1, 'mdi-account-multiple'),

(4, 'Mascotas', 'Aqui puede gestionar la informacion de las mascotas registradas', 1, 2, 'mdi-bone'),

(5, 'Citas', 'Aqui puede gestionar la agenda de citas de la veterinaria', 1, 3, 'mdi-calendar-clock'),

(6, 'Estadísticas', 'Aqui puede ver reportes del sistema', 1, 8, 'mdi-chart-bar'),

(7, 'Usuarios', 'Aqui puede gestionar la informacion del personal de la clinica', 1, 7, 'mdi-account'),

(8, 'Estetica', 'Aqui puede gestionar los servicios de estetica que reciben las mascotas', 1, 5, 'mdi-content-cut'),

(9, 'Escanner', 'Aqui puede ver la ultima mascota escaneada con el lecor de codigo QR', 1, 6, 'mdi-qrcode'),

(10, 'Productos', 'Aqui puede gestionar los productos con los que dispone la clinica', 1, 9, 'mdi-needle');

```
CREATE TABLE IF NOT EXISTS `productos` (  
  `id` int(11) NOT NULL,  
  `nombre` varchar(100) NOT NULL,  
  `descripcion` varchar(1000) NOT NULL,  
  `estado` tinyint(1) NOT NULL DEFAULT '1',
```

```
`id_usuario` int(11) NOT NULL
```

```
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT  
CHARSET=latin1;
```

```
INSERT INTO `productos` (`id`, `nombre`,  
`descripcion`, `estado`, `id_usuario`) VALUES
```

(1, 'LACRISTAL 3 X 25 ml', 'Este producto es una solución ocular equilibrada que retrasa el desarrollo de las cataratas y previene de posibles infecciones oculares en perros y gatos. Fabricado por el Laboratorio Farmadiet Veterinaria.', 1, 1),

(2, 'GERIATIVET 45 COMPRIMIDOS', 'GERIATIVET Es el primer suplemento que combina todos los nutrientes necesarios para mantener la salud y vitalidad de los perros con edad avanzada y que debido a su edad tienen incrementadas sus necesidades nutricionales en estas sustancias. GeriativVet esta especialmente formulado para reducir la perdida en la función mental y mantener los niveles de actividad.\r\nReduce la degeneración articular, activa el sistema inmune, normaliza el funcionamiento del corazón, protege la retina de la degeneración, mantiene la masa muscular, aumenta la protección renal, protege las neuronas retrasando el envejecimiento cerebral.', 1, 1);

```
CREATE TABLE IF NOT EXISTS `razas` (
```

```

`id` int(11) NOT NULL,

`nombre` varchar(30) NOT NULL,

`estado` tinyint(1) NOT NULL DEFAULT '1',

`id_especie` int(11) NOT NULL

) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT
CHARSET=latin1;

```

```

INSERT INTO `razas` (`id`, `nombre`, `estado`,
`id_especie`) VALUES

(1, 'CANICHE', 1, 1),

(2, 'CHAPI', 1, 1),

(3, 'ANGORA', 1, 2),

(4, 'MONTES', 1, 2),

(5, 'GRANDANES', 1, 1),

(6, 'PIT BULL', 1, 1),

(7, 'PASTOR ALEMAN', 1, 1),

(8, 'DOGO ALEMAN', 1, 1),

(9, 'CHIHUAHUA', 1, 1),

(12, 'MINI PIG', 1, 4),

(13, 'DOGO ALEMAN', 1, 1);

```

```

CREATE TABLE IF NOT EXISTS `roles` (

`id_rol` int(11) NOT NULL,

`nombre` varchar(20) NOT NULL,

```

```

`estado` tinyint(1) NOT NULL DEFAULT '1',

`visibilidad` tinyint(1) NOT NULL DEFAULT '1'

) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT
CHARSET=latin1;

```

```

INSERT INTO `roles` (`id_rol`, `nombre`, `estado`,
`visibilidad`) VALUES

(1, 'ADMINISTRADOR', 1, 0),

(2, 'MEDICO', 1, 0),

(3, 'AUXILIAR', 1, 0);

```

```

CREATE TABLE IF NOT EXISTS `rol_proceso` (

`id_rol` int(11) NOT NULL,

`id_proc` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

INSERT INTO `rol_proceso` (`id_rol`, `id_proc`)
VALUES

(1, 1),

(1, 2),

(1, 3),

(1, 4),

(1, 5),

(1, 6),

(1, 7),

```

(1, 8),

(1, 9),

(1, 10);

CREATE TABLE IF NOT EXISTS `tratamientos` (

`id` int(11) NOT NULL,

`id\_atencion` int(11) NOT NULL,

`id\_usuario` int(11) NOT NULL,

`peso` float NOT NULL,

`precio` float NOT NULL,

`fecha` date NOT NULL,

`observacion` varchar(300) DEFAULT NULL,

`estado` tinyint(1) NOT NULL DEFAULT '1'

) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT

CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `tratamientos\_view`

(

`id` int(11)

, `id\_atencion` int(11)

, `id\_usuario` int(11)

, `peso` float

, `precio` float

, `fecha` date

, `observacion` varchar(300)

, `estado` tinyint(1)

, `usuario` varchar(112)

);

CREATE TABLE IF NOT EXISTS `usuarios` (

`id\_usuario` int(11) NOT NULL,

`ci` varchar(10) DEFAULT NULL,

`nombres` varchar(50) NOT NULL DEFAULT "",

`ap\_paterno` varchar(30) NOT NULL DEFAULT "",

`ap\_materno` varchar(30) NOT NULL DEFAULT "",

`direccion` varchar(100) NOT NULL DEFAULT 'S/N',

`foto` varchar(50) NOT NULL DEFAULT
'defecto.png',

`clave` varchar(100) NOT NULL,

`login` varchar(15) NOT NULL,

`celular` int(11) DEFAULT NULL,

`telefono` int(11) DEFAULT NULL,

`estado` tinyint(1) NOT NULL DEFAULT '1',

`observacion` varchar(500) DEFAULT NULL,

`id\_rol` int(11) NOT NULL,

`email` varchar(50) DEFAULT NULL,

`fecha\_reg` date NOT NULL,

```

`tema_principal` varchar(7) NOT NULL DEFAULT
'#13A05D',

`tema_secundario` varchar(7) NOT NULL DEFAULT
'#f57c00'

) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT
CHARSET=latin1;

```

```

INSERT INTO `usuarios` (`id_usuario`, `ci`,
`nombres`, `ap_paterno`, `ap_materno`, `direccion`,
`foto`, `clave`, `login`, `celular`, `telefono`, `estado`,
`observacion`, `id_rol`, `email`, `fecha_reg`,
`tema_principal`, `tema_secundario`) VALUES

```

```

(1, '7160684', 'Rolando Abelino', 'Sanchez', 'Garay', 'c/
Oruro entre pasaje las rosas',
'f_1453857848875_thumb.png',
'$2a$08$qe5k0ukzlasL.Fp1ZoMnjOjLeTsyOcxCroXv
HwMZwPxylXSkeccV.', 'rolando', 75149966,
46647852, 1, 'ninguna', 1,
'rolando.sanchez.developer@gmail.com', '2015-06-
03', '#13A05D', '#F57C00'),

```

```

(3, '7145512', 'Nehyda', 'Ramos', '', 'B/ tejara',
'defecto.png',
'$2a$08$C2dfsvOlubElSHEmRtGNuu5szlhJr41b.jako
EiSjGfUgA/nJQmmu', 'jose', 78963214, 4667892, 1,
'ninguna', 2, 'nadia.ramos@gmail.com', '2016-01-26',
'#13A05D', '#f57c00'),

```

```

(4, '7145541', 'Nataly', 'Jurado', '', '', 'defecto.png',
'$2a$08$tDdPga4afXnbaCBtKop2o.dBN0x8OTPwQ
KeANmn.tCzW2ljiiqkni', 'naty', 0, 0, 1, '', 4, '', '2016-
02-13', '#13A05D', '#f57c00'),

```

```

(5, '', 'Ximena', 'Jurado', '', '', 'defecto.png',
'$2a$08$8eaWY/GX/COligfs7qTua.sGbA9gO68s.bVr
FDWEsh7eK3cdotice', 'ximena', 72947883, 4667895,
1, '', 1, 'ximena.juarado@yahoo.com', '2016-02-13',
'#13A05D', '#f57c00');

```

```

CREATE TABLE IF NOT EXISTS `vacunaciones` (

```

```

`id` int(11) NOT NULL,

`id_producto` int(11) NOT NULL,

`id_mascota` int(11) NOT NULL,

`id_usuario` int(11) NOT NULL,

`id_cita` int(11) NOT NULL,

`fecha` date NOT NULL,

`tipo` tinyint(1) NOT NULL,

`observacion` varchar(200) NOT NULL,

`estado` tinyint(1) NOT NULL DEFAULT '1'

```

```

) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT
CHARSET=latin1 COMMENT='Las vacunaciones que
recibe una mascota';

```

```

CREATE TABLE IF NOT EXISTS `vacunaciones_view`
(

```

```

`usuario` varchar(112)

, `producto` varchar(100)

, `proxima_fecha` date

, `hora_ini` time

```

```

'id` int(11)
'id_producto` int(11)
'id_mascota` int(11)
'id_usuario` int(11)
'id_cita` int(11)
'fecha` date
'tipo` tinyint(1)
'observacion` varchar(200)
'estado` tinyint(1)
);

CREATE TABLE IF NOT EXISTS `visitas` (
  `id` int(11) NOT NULL,
  `fecha` date NOT NULL,
  `hora` time NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `atenciones_view`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `atenciones_view` AS (select `a`.`id`
AS `id`,`a`.`id_usuario` AS
`id_usuario`,`a`.`id_mascota` AS
`id_mascota`,`a`.`estado` AS `estado`,`a`.`fecha` AS
`fecha`,`a`.`signos_clinicos` AS
`signos_clinicos`,`a`.`examenenes` AS
`examenenes`,`a`.`diagnostico` AS
`diagnostico`,`a`.`comentarios` AS
`comentarios`,concat(`u`.`nombres`,`
'u`.`ap_paterno`,` '`,`u`.`ap_materno`) AS `usuario`
from (`atenciones` `a` join `usuarios` `u`) where
(`a`.`id_usuario` = `u`.`id_usuario`));

DROP TABLE IF EXISTS `calendario_view`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `calendario_view` AS (select
`c`.`fecha` AS `fecha`,`c`.`hora_ini` AS
`hora_ini`,`cv`.`id` AS `id`,`cv`.`parvovirus` AS
`parvovirus`,`cv`.`moquillo` AS
`moquillo`,`cv`.`hepatitis` AS
`hepatitis`,`cv`.`leptospirosis` AS
`leptospirosis`,`cv`.`adenovirus` AS
`adenovirus`,`cv`.`parainfluenza` AS
`parainfluenza`,`cv`.`rabia` AS
`rabia`,`cv`.`ca_adenovirus` AS
`ca_adenovirus`,`cv`.`bordetella` AS
`bordetella`,`cv`.`estado` AS
`estado`,`cv`.`asistencia` AS
`asistencia`,`cv`.`id_mascota` AS
`id_mascota`,`cv`.`id_cita` AS `id_cita` from
(`calendario_vacunacion` `cv` join `citas` `c`) where
(`cv`.`id_cita` = `c`.`id`));

DROP TABLE IF EXISTS `citas_solicitudes`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `citas_solicitudes` AS select

```

```

`cl`.`id_cliente` AS `id_cliente`,`cl`.`foto` AS
`foto`,concat(`cl`.`nombres`,`cl`.`ap_paterno`,`cl`.`ap_materno`) AS `cliente`,`c`.`id` AS
`id`,`c`.`id_mascota` AS `id_mascota`,`c`.`fecha` AS
`fecha`,`c`.`hora_ini` AS `hora_ini`,`c`.`hora_fin` AS
`hora_fin`,`c`.`detalle` AS `detalle`,`c`.`tipo` AS
`tipo`,`c`.`fecha_reg` AS `fecha_reg`,`c`.`hora_reg`
AS `hora_reg`,`c`.`estado` AS
`estado`,`m`.`nombre` AS `mascota`,`m`.`raza` AS
`raza`,`m`.`especie` AS `especie` from (((`citas` `c`
join `mascotas_view` `m`) join `clientes` `cl`) where
((`c`.`estado` = 1) and (not(`c`.`id` in (select
`citas_confirmadas`.`id_cita` from
`citas_confirmadas`)))) and (`m`.`id_cliente` =
`cl`.`id_cliente`) and (`c`.`id_mascota` = `m`.`id`));

```

```

DROP TABLE IF EXISTS `citas_view`;

```

```

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `citas_view` AS select `m`.`raza` AS
`raza`,`m`.`especie` AS `especie`,`cl`.`foto` AS
`foto`,concat(`u`.`nombres`,`u`.`ap_paterno`,`u`.`ap_materno`) AS `usuario`,`m`.`nombre` AS
`mascota`,`cl`.`id_cliente` AS
`id_cliente`,concat(`cl`.`nombres`,`cl`.`ap_paterno`,`cl`.`ap_materno`) AS
`cliente`,`c`.`id` AS `id`,`c`.`id_mascota` AS
`id_mascota`,`c`.`fecha` AS `fecha`,`c`.`hora_ini` AS
`hora_ini`,`c`.`hora_fin` AS `hora_fin`,`c`.`detalle`
AS `detalle`,`c`.`tipo` AS `tipo`,`c`.`fecha_reg` AS
`fecha_reg`,`c`.`hora_reg` AS
`hora_reg`,`c`.`estado` AS `estado`,`cc`.`asistencia`
AS `asistencia`,`cc`.`hora` AS

```

```

`hora_confirmacion`,`cc`.`fecha` AS
`fecha_confirmacion` from ((((`citas` `c` join
`citas_confirmadas` `cc`) join `clientes` `cl`) join
`mascotas_view` `m`) join `usuarios` `u`) where
((`c`.`id` = `cc`.`id_cita`) and (`c`.`id_mascota` =
`m`.`id`) and (`cl`.`id_cliente` = `m`.`id_cliente`) and
(`u`.`id_usuario` = `cc`.`id_usuario`));

```

```

DROP TABLE IF EXISTS `clientes_view`;

```

```

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `clientes_view` AS (select `c`.`ci` AS
`ci`,`c`.`id_cliente` AS `id_cliente`,`c`.`foto` AS
`foto`,`c`.`nombres` AS `nombres`,`c`.`ap_paterno`
AS `ap_paterno`,`c`.`ap_materno` AS
`ap_materno`,`c`.`direccion` AS
`direccion`,`c`.`telefono` AS
`telefono`,`c`.`observacion` AS
`observacion`,`c`.`id_usuario` AS
`id_usuario`,`c`.`fecha_reg` AS
`fecha_reg`,`c`.`sexo` AS `sexo`,`c`.`estado` AS
`estado`,`c`.`email` AS `email`,(select count(o) from
`claves` `d` where (`d`.`id_cliente` = `c`.`id_cliente`))
AS `datos` from (`clientes` `c` join `usuarios` `u`)
where (`c`.`id_usuario` = `u`.`id_usuario`));

```

```

DROP TABLE IF EXISTS
`comentarios_clientes_view`;

```

```

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `comentarios_clientes_view` AS
(select `c`.`id` AS `id`,`c`.`texto` AS
`texto`,`c`.`estado` AS `estado`,`c`.`id_contenido`

```

```

AS      `id_contenido`,`cc`.`id_cliente`      AS
`id_cliente`,`cc`.`id_comentario`      AS
`id_comentario`,`cc`.`fecha` AS `fecha`,`cc`.`hora`
AS      `hora`,`concat(`cl`.`nombres`,`
`,`cl`.`ap_paterno`,``,`cl`.`ap_materno`) AS
`cliente`,`cl`.`foto` AS `foto` from ((`comentarios` `c`
join `comentarios_clientes` `cc`) join `clientes` `cl`)
where ((`c`.`id` = `cc`.`id_comentario`) and
(`cc`.`id_cliente` = `cl`.`id_cliente`));

```

```

DROP TABLE IF EXISTS `comentarios_u_view`;

```

```

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `comentarios_u_view` AS (select
`c`.`id` AS `id`,`c`.`texto` AS `texto`,`c`.`estado` AS
`estado`,`c`.`id_contenido` AS
`id_contenido`,`cc`.`id_usuario` AS
`id_usuario`,`cc`.`id_comentario` AS
`id_comentario`,`cc`.`fecha` AS `fecha`,`cc`.`hora`
AS `hora`,`concat(`u`.`nombres`,``,`u`.`ap_paterno`,``,`u`.`ap_materno`) AS `cliente`,`u`.`foto` AS `foto`
from ((`comentarios` `c` join `comentarios_usuarios`
`cc`) join `usuarios` `u`) where ((`c`.`id` =
`cc`.`id_comentario`) and (`cc`.`id_usuario` =
`u`.`id_usuario`));

```

```

DROP TABLE IF EXISTS `contenidos_view`;

```

```

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `contenidos_view` AS (select `c`.`id`
AS      `id`,`c`.`id_tipo_contenido` AS
`id_tipo_contenido`,`c`.`titulo` AS
`titulo`,`c`.`descripcion` AS

```

```

`descripcion`,`c`.`imagen` AS `imagen`,`c`.`fecha`
AS `fecha`,`c`.`hora` AS `hora`,`c`.`publicado` AS
`publicado`,`c`.`estado` AS `estado`,`c`.`id_usuario`
AS      `id_usuario`,`concat(`u`.`nombres`,`
`,`u`.`ap_paterno`,``,`u`.`ap_materno`) AS
`usuario`,`u`.`foto` AS `foto_u` from (`contenidos`
`c` join `usuarios` `u`) where (`u`.`id_usuario` =
`c`.`id_usuario`));

```

```

DROP TABLE IF EXISTS `desparacitaciones_view`;

```

```

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `desparacitaciones_view` AS select
concat(`u`.`nombres`,``,`u`.`ap_paterno`,``,`u`.`ap_materno`) AS `usuario`,`p`.`nombre` AS
`producto`,`c`.`fecha` AS
`proxima_fecha`,`c`.`hora_ini` AS `hora_ini`,`d`.`id`
AS      `id`,`d`.`id_producto` AS
`id_producto`,`d`.`id_mascota` AS
`id_mascota`,`d`.`id_usuario` AS
`id_usuario`,`d`.`observacion` AS
`observacion`,`d`.`estado` AS `estado`,`d`.`fecha`
AS `fecha`,`d`.`peso` AS `peso`,`d`.`id_cita` AS
`id_cita` from (((`desparacitaciones` `d` join `citas`
`c`) join `productos` `p`) join `usuarios` `u`) where
((`d`.`id_cita` = `c`.`id`) and (`d`.`id_producto` =
`p`.`id`) and (`d`.`id_usuario` = `u`.`id_usuario`));

```

```

DROP TABLE IF EXISTS `detalle_tratamiento_view`;

```

```

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `detalle_tratamiento_view` AS
(select `d`.`id` AS `id`,`d`.`id_tratamiento` AS

```

```

'id_tratamiento','d'.id_producto' AS
'id_producto','d'.cc' AS 'cc','p'.nombre' AS
'producto' from ('productos' 'p' join
'detalle_tratamiento' 'd') where ('p'.id' =
'd'.id_producto));

```

```
DROP TABLE IF EXISTS `extravios_view`;
```

```

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `extravios_view` AS (select `e`.id`
AS 'id','e'.id_mascota' AS
'id_mascota','e'.fecha_reg' AS
'fecha_reg','e'.detalle' AS
'detalle','e'.recompensa' AS
'recompensa','m'.id_cliente' AS
'id_cliente','m'.nombre' AS 'mascota','m'.color'
AS 'color','m'.foto' AS 'foto','m'.fecha_nac' AS
'fecha_nac','m'.se_particulares' AS
'se_particulares','m'.raza' AS 'raza','m'.especie'
AS 'especie','m'.sexo' AS
'sexo',concat('c'.nombres','c'.ap_paterno','
c'.ap_materno') AS 'cliente','c'.foto' AS
'foto_cliente','c'.telefono' AS
'telefono','c'.direccion' AS 'direccion','c'.email'
AS 'email' from (('clientes' `c` join `mascotas_view`
`m`) join `extravios` `e`) where (('e'.id_mascota' =
`m`.id) and (`c`.id_cliente' = `m`.id_cliente') and
(`m`.estado' = 1));

```

```
DROP TABLE IF EXISTS `lector_qr_view`;
```

```

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `lector_qr_view` AS (select `l`.id` AS

```

```

'id`,`l`.id_usuario' AS 'id_usuario`,`l`.id_mascota'
AS 'id_mascota','m'.nombre' AS
'nombre','m'.cliente' AS 'cliente','m'.foto' AS
'foto' from ('lector_qr' `l` join `mascotas_view` `m`)
where (`l`.id_mascota' = `m`.id));

```

```
DROP TABLE IF EXISTS `mascotas_view`;
```

```

CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `mascotas_view` AS (select `c`.foto`
AS 'foto_c','m'.id' AS 'id','e'.id' AS
'id_especie','m'.nombre' AS 'nombre','m'.color'
AS 'color','m'.foto' AS 'foto','m'.estado' AS
'estado','m'.observacion' AS
'observacion','m'.fecha_reg' AS
'fecha_reg','m'.fecha_nac' AS
'fecha_nac','m'.id_raza' AS
'id_raza','m'.ultima_visita' AS
'ultima_visita','m'.se_particulares' AS
'se_particulares','m'.id_cliente' AS
'id_cliente','m'.sexo' AS
'sexo',concat('c'.nombres','c'.ap_paterno','
c'.ap_materno') AS 'cliente','c'.telefono' AS
'telefono','c'.direccion' AS
'direccion','e'.nombre' AS 'especie','r'.nombre'
AS 'raza' from (((`mascotas` `m` join `clientes` `c`)
join `razas` `r`) join `especies` `e`) where
((`m`.id_cliente' = `c`.id_cliente') and
(`m`.id_raza' = `r`.id) and (`r`.id_especie' =
`e`.id));

```

```
DROP TABLE IF EXISTS `noticias_view`;
```

```
CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `noticias_view` AS (select `u`.`foto`
AS `foto`,concat(`u`.`nombres`,` `,`u`.`ap_paterno`,`
`,`u`.`ap_materno`) AS `usuario`,`n`.`id` AS
`id`,`n`.`titulo` AS `titulo`,`n`.`descripcion` AS
`descripcion`,`n`.`estado` AS `estado`,`n`.`imagen`
AS `imagen`,`n`.`id_usuario` AS
`id_usuario`,`n`.`fecha` AS `fecha`,`n`.`hora` AS
`hora` from (`noticias` `n` join `usuarios` `u`) where
(`u`.`id_usuario` = `n`.`id_usuario`));
```

```
DROP TABLE IF EXISTS `peluqueras_view`;
```

```
CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `peluqueras_view` AS (select `p`.`id`
AS `id`,`p`.`fecha` AS `fecha`,`p`.`hora` AS
`hora`,`p`.`servicio` AS `servicio`,`p`.`observacion`
AS `observacion`,`p`.`precio` AS
`precio`,`p`.`estado` AS `estado`,`p`.`id_mascota`
AS `id_mascota`,`p`.`id_usuario` AS
`id_usuario`,`p`.`banio` AS `banio`,`p`.`corte` AS
`corte`,`p`.`unia` AS `unia`,`p`.`monio` AS
`monio`,`p`.`tranquilizante` AS
`tranquilizante`,`m`.`nombre` AS
`mascota`,`m`.`cliente` AS
`cliente`,concat(`u`.`nombres`,` `,`u`.`ap_paterno`,`
`,`u`.`ap_materno`) AS `usuario` from ((`peluqueras`
`p` join `mascotas_view` `m`) join `usuarios` `u`)
where ((`p`.`id_mascota` = `m`.`id`) and
(`p`.`id_usuario` = `u`.`id_usuario`));
```

```
DROP TABLE IF EXISTS
`proximas_vacunaciones_view`;
```

```
CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `proximas_vacunaciones_view` AS
(select `pv`.`id_vacunacion` AS
`id_vacunacion`,`c`.`fecha` AS `fecha`,`c`.`hora` AS
`hora`,`cc`.`asistencia` AS `asistencia` from
((`proximas_vacunaciones` `pv` join `citas` `c`) join
`citas_confirmadas` `cc`) where ((`pv`.`id_cita` =
`cc`.`id_cita`) and (`cc`.`id_cita` = `c`.`id`));
```

```
DROP TABLE IF EXISTS `proximas_visitas_view`;
```

```
CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `proximas_visitas_view` AS (select
`p`.`id` AS `id`,`p`.`fecha_reg` AS
`fecha_reg`,`p`.`parvovirus` AS
`parvovirus`,`p`.`moquillo` AS
`moquillo`,`p`.`hepatitis` AS
`hepatitis`,`p`.`leptospirosis` AS
`leptospirosis`,`p`.`adenovirus` AS
`adenovirus`,`p`.`parainfluenza` AS
`parainfluenza`,`p`.`rabia` AS
`rabia`,`p`.`ca_adenovirus` AS
`ca_adenovirus`,`p`.`bordetella` AS
`bordetella`,`p`.`id_cita` AS `id_cita`,`p`.`estado` AS
`estado`,`c`.`id_mascota` AS
`id_mascota`,`c`.`hora` AS `hora`,`c`.`fecha` AS
`fecha`,`cc`.`asistencia` AS `asistencia` from
((`proximas_visitas` `p` join `citas` `c`) join
`citas_confirmadas` `cc`) where ((`p`.`id_cita` =
`cc`.`id_cita`) and (`c`.`id` = `cc`.`id_cita`));
```

```
DROP TABLE IF EXISTS
`proxima_desparacion_view`;
```

```
CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `proxima_desparacitacion_view` AS
(select      `pd`.`id_desparacitacion`      AS
`id_desparacitacion`,`c`.`fecha`      AS
`fecha`,`c`.`hora`      AS `hora`,`cc`.`asistencia`      AS
`asistencia` from ((`proxima_desparacitacion` `pd`
join `citas` `c`) join `citas_confirmadas` `cc`) where
((`pd`.`id_cita` = `cc`.`id_cita`) and (`cc`.`id_cita` =
`c`.`id`));
```

```
DROP TABLE IF EXISTS `tratamientos_view`;
```

```
CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `tratamientos_view` AS (select
`t`.`id`      AS      `id`,`t`.`id_atencion`      AS
`id_atencion`,`t`.`id_usuario`      AS
`id_usuario`,`t`.`peso`      AS `peso`,`t`.`precio`      AS
`precio`,`t`.`fecha`      AS `fecha`,`t`.`observacion`      AS
`observacion`,`t`.`estado`      AS
`estado`,concat(`u`.`nombres`,`u`.`ap_paterno`,`u`.`ap_materno`)
      AS      `usuario` from
(`tratamientos` `t` join `usuarios` `u`) where
(`t`.`id_usuario` = `u`.`id_usuario`));
```

```
DROP TABLE IF EXISTS `vacunaciones_view`;
```

```
CREATE ALGORITHM=UNDEFINED SQL SECURITY
DEFINER VIEW `vacunaciones_view` AS (select
concat(`u`.`nombres`,`u`.`ap_paterno`,`u`.`ap_materno`)
      AS      `usuario`,`p`.`nombre`      AS
`producto`,`c`.`fecha`      AS
`proxima_fecha`,`c`.`hora_ini`      AS `hora_ini`,`v`.`id`
```

```
      AS      `id`,`v`.`id_producto`      AS
`id_producto`,`v`.`id_mascota`      AS
`id_mascota`,`v`.`id_usuario`      AS
`id_usuario`,`v`.`id_cita`      AS `id_cita`,`v`.`fecha`      AS
`fecha`,`v`.`tipo`      AS `tipo`,`v`.`observacion`      AS
`observacion`,`v`.`estado`      AS `estado` from
(((`vacunaciones` `v` join `citas` `c`) join `productos`
`p`) join `usuarios` `u`) where ((`v`.`id_cita` = `c`.`id`)
and (`v`.`id_producto` = `p`.`id`) and (`v`.`id_usuario`
= `u`.`id_usuario`));
```

```
ALTER TABLE `atenciones`
```

```
ADD PRIMARY KEY (`id`), ADD KEY `id_usuario`
(`id_usuario`), ADD KEY `id_mascota`
(`id_mascota`);
```

```
ALTER TABLE `calendario_vacunacion`
```

```
ADD PRIMARY KEY (`id`), ADD KEY `id_mascota`
(`id_mascota`), ADD KEY `id_cita` (`id_cita`);
```

```
ALTER TABLE `celulares`
```

```
ADD PRIMARY KEY (`numero`), ADD KEY
`id_cliente` (`id_cliente`);
```

```
ALTER TABLE `citas`
```

```
ADD PRIMARY KEY (`id`), ADD KEY `id_mascota`
(`id_mascota`);
```

```
ALTER TABLE `citas_confirmadas`

ADD PRIMARY KEY (`id_cita`), ADD KEY
`id_usuario` (`id_usuario`);
```

```
ALTER TABLE `claves`

ADD PRIMARY KEY (`id_cliente`);
```

```
ALTER TABLE `clientes`

ADD PRIMARY KEY (`id_cliente`), ADD KEY
`id_usuario` (`id_usuario`);
```

```
ALTER TABLE `desparasitaciones`

ADD PRIMARY KEY (`id`), ADD KEY `id_producto`
(`id_producto`), ADD KEY `id_mascota`
(`id_mascota`), ADD KEY `id_usuario` (`id_usuario`),
ADD KEY `id_cita` (`id_cita`);
```

```
ALTER TABLE `detalle_tratamiento`

ADD PRIMARY KEY (`id`), ADD KEY
`id_tratamiento` (`id_tratamiento`), ADD KEY
`id_producto` (`id_producto`);
```

```
ALTER TABLE `dispositivos`

ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `dispositivos_clientes`
```

```
ADD PRIMARY KEY (`id_cliente`,`id_dispositivo`),
ADD KEY `id_dispositivo` (`id_dispositivo`);
```

```
ALTER TABLE `especies`

ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `extravios`

ADD PRIMARY KEY (`id`), ADD KEY `id_mascota`
(`id_mascota`);
```

```
ALTER TABLE `lector_qr`

ADD PRIMARY KEY (`id`), ADD KEY `id_usuario`
(`id_usuario`), ADD KEY `id_mascota`
(`id_mascota`);
```

```
ALTER TABLE `mascotas`

ADD PRIMARY KEY (`id`), ADD KEY `id_raza`
(`id_raza`), ADD KEY `id_cliente` (`id_cliente`);
```

```
ALTER TABLE `noticias`

ADD PRIMARY KEY (`id`), ADD KEY `id_usuario`
(`id_usuario`), ADD FULLTEXT KEY `titulo`
(`titulo`,`descripcion`);
```

```
ALTER TABLE `peluquerias`
```

```
ADD PRIMARY KEY (`id`), ADD KEY `id_mascota`  
(`id_mascota`), ADD KEY `id_usuario` (`id_usuario`);
```

```
ALTER TABLE `pesos`
```

```
ADD PRIMARY KEY (`id`), ADD KEY `id_mascota`  
(`id_mascota`);
```

```
ALTER TABLE `procesos`
```

```
ADD PRIMARY KEY (`id_proc`);
```

```
ALTER TABLE `productos`
```

```
ADD PRIMARY KEY (`id`), ADD KEY `id_usuario`  
(`id_usuario`);
```

```
ALTER TABLE `razas`
```

```
ADD PRIMARY KEY (`id`), ADD KEY `id_especie`  
(`id_especie`);
```

```
ALTER TABLE `roles`
```

```
ADD PRIMARY KEY (`id_rol`);
```

```
ALTER TABLE `rol_proceso`
```

```
ADD PRIMARY KEY (`id_rol`,`id_proc`), ADD KEY  
`id_proc` (`id_proc`);
```

```
ALTER TABLE `tratamientos`
```

```
ADD PRIMARY KEY (`id`), ADD KEY `id_atencion`  
(`id_atencion`), ADD KEY `id_usuario` (`id_usuario`);
```

```
ALTER TABLE `usuarios`
```

```
ADD PRIMARY KEY (`id_usuario`), ADD UNIQUE  
KEY `login` (`login`), ADD KEY `id_rol` (`id_rol`);
```

```
ALTER TABLE `vacunaciones`
```

```
ADD PRIMARY KEY (`id`), ADD KEY `id_mascota`  
(`id_mascota`), ADD KEY `id_usuario` (`id_usuario`),  
ADD KEY `id_producto` (`id_producto`), ADD KEY  
`id_cita` (`id_cita`);
```

```
ALTER TABLE `visitas`
```

```
ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `atenciones`
```

```
MODIFY `id` int(11) NOT NULL  
AUTO_INCREMENT,AUTO_INCREMENT=2;
```

```
ALTER TABLE `calendario_vacunacion`
```

```
MODIFY `id` int(11) NOT NULL  
AUTO_INCREMENT,AUTO_INCREMENT=15;
```

```
ALTER TABLE `citas`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=49;
```

```
ALTER TABLE `clientes`
```

```
MODIFY    `id_cliente`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=3;
```

```
ALTER TABLE `desparasitaciones`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=2;
```

```
ALTER TABLE `detalle_tratamiento`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=2;
```

```
ALTER TABLE `especies`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=6;
```

```
ALTER TABLE `extravios`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT;
```

```
ALTER TABLE `lector_qr`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT;
```

```
ALTER TABLE `mascotas`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=3;
```

```
ALTER TABLE `noticias`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=6;
```

```
ALTER TABLE `peluqueras`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=3;
```

```
ALTER TABLE `pesos`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT;
```

```
ALTER TABLE `procesos`
```

```
MODIFY    `id_proc`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=11;
```

```
ALTER TABLE `productos`
```

```
MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=3;
```

```
ALTER TABLE `razas`

MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=14;
```

```
ALTER TABLE `roles`

MODIFY    `id_rol`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=8;
```

```
ALTER TABLE `tratamientos`

MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=2;
```

```
ALTER TABLE `usuarios`

MODIFY    `id_usuario`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=6;
```

```
ALTER TABLE `vacunaciones`

MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT,AUTO_INCREMENT=3;
```

```
ALTER TABLE `visitas`

MODIFY    `id`    int(11)    NOT    NULL
AUTO_INCREMENT;
```

```
ALTER TABLE `atenciones`
```

```
ADD CONSTRAINT `atenciones_ibfk_1` FOREIGN
KEY (`id_usuario`) REFERENCES `usuarios`
(`id_usuario`),
```

```
ADD CONSTRAINT `atenciones_ibfk_2` FOREIGN
KEY (`id_mascota`) REFERENCES `mascotas` (`id`);
```

```
ALTER TABLE `calendario_vacunacion`
```

```
ADD CONSTRAINT `calendario_vacunacion_ibfk_1`
FOREIGN KEY (`id_mascota`) REFERENCES
`mascotas` (`id`),
```

```
ADD CONSTRAINT `calendario_vacunacion_ibfk_2`
FOREIGN KEY (`id_cita`) REFERENCES `citas` (`id`);
```

```
ALTER TABLE `celulares`
```

```
ADD CONSTRAINT `celulares_ibfk_1` FOREIGN KEY
(`id_cliente`) REFERENCES `clientes` (`id_cliente`);
```

```
ALTER TABLE `citas`
```

```
ADD CONSTRAINT `citas_ibfk_1` FOREIGN KEY
(`id_mascota`) REFERENCES `mascotas` (`id`);
```

```
ALTER TABLE `citas_confirmadas`
```

```
ADD CONSTRAINT `citas_confirmadas_ibfk_1`
FOREIGN KEY (`id_cita`) REFERENCES `citas` (`id`),
```

```
ADD CONSTRAINT `citas_confirmadas_ibfk_2`
FOREIGN KEY (`id_usuario`) REFERENCES
`usuarios` (`id_usuario`);
```

```
ALTER TABLE `claves`
```

```
ADD CONSTRAINT `claves_ibfk_1` FOREIGN KEY
(`id_cliente`) REFERENCES `clientes` (`id_cliente`);
```

```
ALTER TABLE `clientes`
```

```
ADD CONSTRAINT `clientes_ibfk_1` FOREIGN KEY
(`id_usuario`) REFERENCES `usuarios` (`id_usuario`);
```

```
ALTER TABLE `desparacitaciones`
```

```
ADD CONSTRAINT `desparacitaciones_ibfk_1`
FOREIGN KEY (`id_producto`) REFERENCES
`productos` (`id`),
```

```
ADD CONSTRAINT `desparacitaciones_ibfk_2`
FOREIGN KEY (`id_mascota`) REFERENCES
`mascotas` (`id`),
```

```
ADD CONSTRAINT `desparacitaciones_ibfk_3`
FOREIGN KEY (`id_usuario`) REFERENCES
`usuarios` (`id_usuario`),
```

```
ADD CONSTRAINT `desparacitaciones_ibfk_4`
FOREIGN KEY (`id_cita`) REFERENCES `citas` (`id`);
```

```
ALTER TABLE `detalle_tratamiento`
```

```
ADD CONSTRAINT `detalle_tratamiento_ibfk_1`
FOREIGN KEY (`id_tratamiento`) REFERENCES
`tratamientos` (`id`),
```

```
ADD CONSTRAINT `detalle_tratamiento_ibfk_2`
FOREIGN KEY (`id_producto`) REFERENCES
`productos` (`id`);
```

```
ALTER TABLE `dispositivos_clientes`
```

```
ADD CONSTRAINT `dispositivos_clientes_ibfk_1`
FOREIGN KEY (`id_cliente`) REFERENCES `clientes`
(`id_cliente`),
```

```
ADD CONSTRAINT `dispositivos_clientes_ibfk_2`
FOREIGN KEY (`id_dispositivo`) REFERENCES
`dispositivos` (`id`);
```

```
ALTER TABLE `extravios`
```

```
ADD CONSTRAINT `extravios_ibfk_1` FOREIGN KEY
(`id_mascota`) REFERENCES `mascotas` (`id`);
```

```
ALTER TABLE `lector_qr`
```

```
ADD CONSTRAINT `lector_qr_ibfk_1` FOREIGN KEY
(`id_usuario`) REFERENCES `usuarios` (`id_usuario`),
```

```
ADD CONSTRAINT `lector_qr_ibfk_2` FOREIGN KEY
(`id_mascota`) REFERENCES `mascotas` (`id`);
```

```
ALTER TABLE `mascotas`
```

```
ADD CONSTRAINT `mascotas_ibfk_1` FOREIGN  
KEY (`id_cliente`) REFERENCES `clientes`  
(`id_cliente`),
```

```
ADD CONSTRAINT `mascotas_ibfk_2` FOREIGN  
KEY (`id_raza`) REFERENCES `razas` (`id`);
```

```
ALTER TABLE `noticias`
```

```
ADD CONSTRAINT `noticias_ibfk_1` FOREIGN KEY  
(`id_usuario`) REFERENCES `usuarios` (`id_usuario`);
```

```
ALTER TABLE `peluquerias`
```

```
ADD CONSTRAINT `peluquerias_ibfk_1` FOREIGN  
KEY (`id_mascota`) REFERENCES `mascotas` (`id`),
```

```
ADD CONSTRAINT `peluquerias_ibfk_2` FOREIGN  
KEY (`id_usuario`) REFERENCES `usuarios`  
(`id_usuario`);
```

```
ALTER TABLE `pesos`
```

```
ADD CONSTRAINT `pesos_ibfk_1` FOREIGN KEY  
(`id_mascota`) REFERENCES `mascotas` (`id`);
```

```
ALTER TABLE `productos`
```

```
ADD CONSTRAINT `productos_ibfk_1` FOREIGN  
KEY (`id_usuario`) REFERENCES `usuarios`  
(`id_usuario`);
```

```
ALTER TABLE `razas`
```

```
ADD CONSTRAINT `razas_ibfk_1` FOREIGN KEY  
(`id_especie`) REFERENCES `especies` (`id`);
```

```
ALTER TABLE `rol_proceso`
```

```
ADD CONSTRAINT `rol_proceso_ibfk_1` FOREIGN  
KEY (`id_rol`) REFERENCES `roles` (`id_rol`),
```

```
ADD CONSTRAINT `rol_proceso_ibfk_2` FOREIGN  
KEY (`id_proc`) REFERENCES `procesos` (`id_proc`);
```

```
ALTER TABLE `tratamientos`
```

```
ADD CONSTRAINT `tratamientos_ibfk_1` FOREIGN  
KEY (`id_atencion`) REFERENCES `atenciones` (`id`),
```

```
ADD CONSTRAINT `tratamientos_ibfk_2` FOREIGN  
KEY (`id_usuario`) REFERENCES `usuarios`  
(`id_usuario`);
```

```
ALTER TABLE `usuarios`
```

```
ADD CONSTRAINT `usuarios_ibfk_1` FOREIGN KEY  
(`id_rol`) REFERENCES `roles` (`id_rol`);
```

```
ALTER TABLE `vacunaciones`
```

```
ADD CONSTRAINT `vacunaciones_ibfk_1` FOREIGN  
KEY (`id_mascota`) REFERENCES `mascotas` (`id`),
```

```
ADD CONSTRAINT `vacunaciones_ibfk_2` FOREIGN  
KEY (`id_usuario`) REFERENCES `usuarios`  
(`id_usuario`),
```

```
ADD CONSTRAINT `vacunaciones_ibfk_3` FOREIGN  
KEY (`id_producto`) REFERENCES `productos` (`id`),  
  
ADD CONSTRAINT `vacunaciones_ibfk_4` FOREIGN  
KEY (`id_cita`) REFERENCES `citas` (`id`);
```

1.1.2.2.3.10.6 Diccionario de Datos

1. ESTRUCTURA DE TABLA PARA LA TABLA ATENCIONES

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
id_usuario	int(11)	No	
id_mascota	int(11)	No	
estado	int(11)	No	1
fecha	date	No	
signos_clinicos	varchar(500)	Sí	NULL
exámenes	varchar(200)	Sí	NULL
diagnostico	varchar(200)	Sí	NULL
comentarios	varchar(500)	Sí	NULL

1.1. Estructura stand-in para la vista atenciones_view

Columna	Tipo	Nulo	Predeterminado
id	int(11)	No	0
id_usuario	int(11)	No	
id_mascota	int(11)	No	
estado	int(11)	No	1
fecha	date	No	
signos_clinicos	varchar(500)	Sí	NULL
exámenes	varchar(200)	Sí	NULL
diagnostico	varchar(200)	Sí	NULL
comentarios	varchar(500)	Sí	NULL
usuario	varchar(112)	No	

2. ESTRUCTURA DE TABLA PARA LA TABLA CITAS

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
id_mascota	int(11)	No	

fecha	date	No	
hora_ini	time	No	
hora_fin	time	No	
detalle	varchar(500)	Sí	NULL
tipo	tinyint(1)	No	
fecha_reg	date	No	
hora_reg	time	No	
estado	tinyint(1)	No	1

3. ESTRUCTURA DE TABLA PARA LA TABLA CITAS_CONFIRMADAS

Columna	Tipo	Nulo	Predeterminado
id_usuario	int(11)	No	
<i>id_cita</i>	int(11)	No	
fecha	date	No	
hora	time	No	
asistencia	tinyint(1)	No	0

3.1.Estructura stand-in para la vista citas_solicitudes

Columna	Tipo	Nulo	Predeterminado
id_cliente	int(11)	No	0
foto	varchar(20)	No	defecto.png
cliente	varchar(112)	Sí	NULL
id	int(11)	No	0
id_mascota	int(11)	No	
fecha	date	No	
hora_ini	time	No	
hora_fin	time	No	
detalle	varchar(500)	Sí	NULL
tipo	tinyint(1)	No	
fecha_reg	date	No	
hora_reg	time	No	

estado	tinyint(1)	No	1
masкота	varchar(30)	No	
raza	varchar(30)	No	
especie	varchar(30)	No	

3.2.Estructura Stand-in para la vista citas_view

Columna	Tipo	Nulo	Predeterminado
raza	varchar(30)	No	
especie	varchar(30)	No	
foto	varchar(20)	No	defecto.png
usuario	varchar(112)	No	
masкота	varchar(30)	No	
id_cliente	int(11)	No	0
cliente	varchar(112)	Sí	NULL
id	int(11)	No	0
id_mascota	int(11)	No	
fecha	date	No	
hora_ini	time	No	
hora_fin	time	No	
detalle	varchar(500)	Sí	NULL
tipo	tinyint(1)	No	
fecha_reg	date	No	
hora_reg	time	No	
estado	tinyint(1)	No	1
asistencia	tinyint(1)	No	0
hora_confirmacion	time	No	
fecha_confirmacion	date	No	

4. ESTRUCTURA DE TABLA PARA LA TABLA CLAVES

Columna	Tipo	Nulo	Predeterminado
<i>id_cliente</i>	int(11)	No	

clave	varchar(100)	No	
--------------	--------------	----	--

5. ESTRUCTURA DE TABLA PARA LA TABLA CLIENTES

Columna	Tipo	Nulo	Predeterminado
ci	varchar(12)	Sí	NULL
<i>id_cliente</i>	int(11)	No	
foto	varchar(20)	No	defecto.png
nombres	varchar(50)	No	
ap_paterno	varchar(30)	Sí	
ap_materno	varchar(30)	Sí	
direccion	varchar(100)	Sí	NULL
celular	int(11)	Sí	NULL
telefono	varchar(15)	Sí	NULL
observacion	varchar(500)	Sí	NULL
id_usuario	int(11)	No	
fecha_reg	date	No	
sexo	varchar(10)	Sí	NULL
estado	tinyint(1)	No	1
email	varchar(30)	Sí	NULL

5.1.Estructura Stand-in para la vista clientes_view

Columna	Tipo	Nulo	Predeterminado
ci	varchar(12)	Sí	NULL
id_cliente	int(11)	No	0
foto	varchar(20)	No	defecto.png
nombres	varchar(50)	No	
ap_paterno	varchar(30)	Sí	
ap_materno	varchar(30)	Sí	
direccion	varchar(100)	Sí	NULL
celular	int(11)	Sí	NULL
telefono	varchar(15)	Sí	NULL

observacion	varchar(500)	Sí	NULL
id_usuario	int(11)	No	
fecha_reg	date	No	
sexo	varchar(10)	Sí	NULL
estado	tinyint(1)	No	1
email	varchar(30)	Sí	NULL
datos	bigint(21)	Sí	NULL

6. ESTRUCTURA DE TABLA PARA LA TABLA DESPARACITACIONES

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
id_producto	int(11)	No	
id_mascota	int(11)	No	
id_usuario	int(11)	No	
fecha	date	No	
observacion	varchar(200)	Sí	NULL
estado	tinyint(1)	No	1
peso	float	No	0

6.1.Estructura Stand-in para la vista desparacitaciones_view

Columna	Tipo	Nulo	Predeterminado
id	int(11)	No	0
id_producto	int(11)	No	
id_mascota	int(11)	No	
id_usuario	int(11)	No	
fecha	date	No	
observacion	varchar(200)	Sí	NULL
estado	tinyint(1)	No	1
peso	float	No	0
producto	varchar(100)	No	
usuario	varchar(112)	No	

proxima	bigint(21)	Sí	NULL
----------------	------------	----	------

7. ESTRUCTURA DE TABLA PARA LA TABLA DETALLE_TRATAMIENTO

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
id_tratamiento	int(11)	No	
id_producto	int(11)	No	
cc	float	No	

7.1 Estructura Stand-in para la vista detalle_tratamiento_view

Columna	Tipo	Nulo	Predeterminado
id	int(11)	No	0
id_tratamiento	int(11)	No	
id_producto	int(11)	No	
cc	float	No	
producto	varchar(100)	No	

8. ESTRUCTURA DE TABLA PARA LA TABLA DISPOSITIVOS

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	varchar(50)	No	
token	varchar(200)	No	
estado	tinyint(1)	No	1
modelo	varchar(30)	Sí	desconocido
version	varchar(20)	Sí	desconocido

9. ESTRUCTURA DE TABLA PARA LA TABLA DISPOSITIVOS_CLIENTES

Columna	Tipo	Nulo	Predeterminado
<i>id_cliente</i>	int(11)	No	
<i>id_dispositivo</i>	varchar(50)	No	
fecha	date	No	

10. ESTRUCTURA DE TABLA PARA LA TABLA ESPECIES

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
nombre	varchar(30)	No	
estado	tinyint(1)	No	1

11. ESTRUCTURA DE TABLA PARA LA TABLA LECTOR_QR

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
id_usuario	int(11)	No	
id_mascota	int(11)	No	

11.1. ESTRUCTURA STAND-IN PARA LA VISTA LECTOR_QR_VIEW

Columna	Tipo	Nulo	Predeterminado
id	int(11)	No	0
id_usuario	int(11)	No	
id_mascota	int(11)	No	
nombre	varchar(30)	No	
cliente	varchar(112)	Sí	NULL
foto	varchar(30)	No	defecto.png

12. ESTRUCTURA DE TABLA PARA LA TABLA MASCOTAS

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
nombre	varchar(30)	No	
color	varchar(30)	No	
foto	varchar(30)	No	defecto.png
estado	tinyint(1)	No	1
observacion	varchar(500)	No	

fecha_reg	date	No	
fecha_nac	date	Sí	NULL
id_raza	int(11)	No	
ultima_visita	date	Sí	NULL
id_cliente	int(11)	No	
sexo	varchar(10)	No	
se_particulares	varchar(200)	No	
extravio	tinyint(1)	No	0

12.1. Estructura Stand-in para la vista mascotas_view

Columna	Tipo	Nulo	Predeterminado
foto_c	varchar(20)	No	defecto.png
id	int(11)	No	0
id_especie	int(11)	No	0
nombre	varchar(30)	No	
color	varchar(30)	No	
foto	varchar(30)	No	defecto.png
estado	tinyint(1)	No	1
observacion	varchar(500)	No	
fecha_reg	date	No	
fecha_nac	date	Sí	NULL
id_raza	int(11)	No	
ultima_visita	date	Sí	NULL
se_particulares	varchar(200)	No	
id_cliente	int(11)	No	
sexo	varchar(10)	No	
cliente	varchar(112)	Sí	NULL
telefono	varchar(15)	Sí	NULL
celular	int(11)	Sí	NULL
direccion	varchar(100)	Sí	NULL
especie	varchar(30)	No	

raza	varchar(30)	No	
-------------	-------------	----	--

13. ESTRUCTURA DE TABLA PARA LA TABLA NOTICIAS

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
titulo	varchar(30)	No	
descripcion	varchar(500)	No	
estado	tinyint(1)	No	1
imagen	varchar(20)	No	defecto.png
id_usuario	int(11)	No	
fecha	date	No	
hora	time	No	

13.1 ESTRUCTURA STAND-IN PARA LA VISTA NOTICIAS_VIEW

Columna	Tipo	Nulo	Predeterminado
foto	varchar(50)	No	defecto.png
usuario	varchar(112)	No	
id	int(11)	No	0
titulo	varchar(30)	No	
descripcion	varchar(500)	No	
estado	tinyint(1)	No	1
imagen	varchar(20)	No	defecto.png
id_usuario	int(11)	No	
fecha	date	No	
hora	time	No	

14. ESTRUCTURA DE TABLA PARA LA TABLA PELUQUERIAS

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	

fecha	date	No	
hora	time	No	
servicio	varchar(100)	Sí	NULL
observacion	varchar(500)	Sí	NULL
precio	double	No	0
estado	tinyint(1)	No	1
id_mascota	int(11)	No	
id_usuario	int(11)	No	
banio	tinyint(1)	No	0
corte	tinyint(1)	No	0
unia	tinyint(1)	No	0
monio	tinyint(1)	No	0
tranquilizante	varchar(100)	No	

14.1. Estructura Stand-in para la vista peluqueras_view

Columna	Tipo	Nulo	Predeterminado
id	int(11)	No	0
fecha	date	No	
hora	time	No	
servicio	varchar(100)	Sí	NULL
observacion	varchar(500)	Sí	NULL
precio	double	No	0
estado	tinyint(1)	No	1
id_mascota	int(11)	No	
id_usuario	int(11)	No	
banio	tinyint(1)	No	0
corte	tinyint(1)	No	0
unia	tinyint(1)	No	0
monio	tinyint(1)	No	0
tranquilizante	varchar(100)	No	
mascota	varchar(30)	No	

cliente	varchar(112)	Sí	NULL
usuario	varchar(112)	No	

15. ESTRUCTURA DE TABLA PARA LA TABLA PESOS

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
fecha	date	No	
peso	float	No	0
id_mascota	int(11)	No	

16. ESTRUCTURA DE TABLA PARA LA TABLA PROCESOS

Columna	Tipo	Nulo	Predeterminado
<i>id_proc</i>	int(11)	No	
nombre	varchar(30)	No	
descripcion	varchar(200)	No	
estado	tinyint(1)	No	1
prioridad	int(11)	No	0
icon	varchar(20)	No	mdi-cat

16.1. Volcado de datos para la tabla procesos

2	Noticias	Aqui puede gestionar las noticias que se publicaran en la aplicacion movil de los clientes.	1	4	mdi-newspaper
3	Roles	Aqui puede gestionar los roles y permisos de los usuarios	1	10	mdi-lock
4	Clientes	Aqui úede gestionar la informacion de los clientes de la veterinaria	1	1	mdi-account-multiple
5	Mascotas	Aqui puede gestionar la informacion de las mascotas registradas	1	2	mdi-bone
6	Citas	Aqui puede gestionar la agenda de citas de la veterinaria	1	3	mdi-calendar-clock
7	Reportes	Aqui puede ver reportes del sistema	1	8	mdi-chart-bar

9	Usuarios	Aqui puede gestionar la informacion del personal de la clinica	1	7	mdi-account
10	Estetica	Aqui puede gestionar los servicios de estetica que reciben las mascotas	1	5	mdi-content-cut
11	Escanner	Aqui puede ver la ultima mascota escaneada con el lector de codigo QR	1	6	mdi-qrcode
12	Productos	Aqui puede gestionar los productos con los que dispone la clinica	1	9	mdi-needle

17. ESTRUCTURA DE TABLA PARA LA TABLA PRODUCTOS

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
nombre	varchar(100)	No	
descripcion	varchar(1000)	No	
estado	tinyint(1)	No	1
id_usuario	int(11)	No	

18. ESTRUCTURA DE TABLA PARA LA TABLA PROXIMAS_VACUNACIONES

Columna	Tipo	Nulo	Predeterminado
<i>id_vacunacion</i>	int(11)	No	
id_cita	int(11)	No	

19. ESTRUCTURA DE TABLA PARA LA TABLA PROXIMAS_VISITAS

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
fecha_reg	date	No	
parvovirus	tinyint(1)	No	0
moquillo	tinyint(1)	No	0
hepatitis	tinyint(1)	No	0
leptospirosis	tinyint(1)	No	0
adenovirus	tinyint(1)	No	0

parainfluenza	tinyint(1)	No	0
rabia	tinyint(1)	No	0
ca_adenovirus	tinyint(1)	No	0
bordetella	tinyint(1)	No	0
id_cita	int(11)	No	
estado	tinyint(1)	No	1

20. ESTRUCTURA DE TABLA PARA LA TABLA PROXIMA_DESPARACITACION

Columna	Tipo	Nulo	Predeterminado
<i>id_desparacitacion</i>	int(11)	No	
id_cita	int(11)	No	

21. ESTRUCTURA DE TABLA PARA LA TABLA RAZAS

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
nombre	varchar(30)	No	
estado	tinyint(1)	No	1
id_especie	int(11)	No	

21.1 Volcado de datos para la tabla razas

1	CANICHE	1	1
2	CHAPI	1	1
3	ANGORA	1	2
4	MONTES	1	2
5	GRAN DANES	1	1
6	PIT BULL	1	1
7	PASTOR ALEMAN	1	1
8	DOGO ALEMAN	1	1
9	CHIHUAHUA	1	1
12	MINI PIG	1	4

22. ESTRUCTURA DE TABLA PARA LA TABLA ROLES

Columna	Tipo	Nulo	Predeterminado
<i>id_rol</i>	int(11)	No	
nombre	varchar(20)	No	
estado	tinyint(1)	No	1
visibilidad	tinyint(1)	No	1

22.1. Volcado de datos para la tabla roles

1	ADMINISTRADOR	1	0
2	MEDICO	1	0
4	PELUQUERO	1	0

23. ESTRUCTURA DE TABLA PARA LA TABLA ROL_PROCESO

Columna	Tipo	Nulo	Predeterminado
<i>id_rol</i>	int(11)	No	
<i>id_proc</i>	int(11)	No	

24. ESTRUCTURA DE TABLA PARA LA TABLA TRATAMIENTOS

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
id_atencion	int(11)	No	
id_usuario	int(11)	No	
peso	float	No	
precio	float	No	
fecha	date	No	
observacion	varchar(300)	Sí	NULL
estado	tinyint(1)	No	1

24.1. Estructura Stand-in para la vista tratamientos_view

Columna	Tipo	Nulo	Predeterminado
id	int(11)	No	0
id_atencion	int(11)	No	
id_usuario	int(11)	No	
peso	float	No	
precio	float	No	
fecha	date	No	
observacion	varchar(300)	Sí	NULL
estado	tinyint(1)	No	1
usuario	varchar(112)	No	

25. ESTRUCTURA DE TABLA PARA LA TABLA USUARIOS

Columna	Tipo	Nulo	Predeterminado
<i>id_usuario</i>	int(11)	No	
ci	varchar(10)	Sí	NULL
nombres	varchar(50)	No	
ap_paterno	varchar(30)	No	
ap_materno	varchar(30)	No	
direccion	varchar(100)	No	S/N
foto	varchar(50)	No	defecto.png
clave	varchar(100)	No	
login	varchar(15)	No	
celular	int(11)	Sí	NULL
telefono	int(11)	Sí	NULL
estado	tinyint(1)	No	1
observacion	varchar(500)	Sí	NULL
id_rol	int(11)	No	
email	varchar(50)	Sí	NULL
fecha_reg	date	No	
tema_principal	varchar(7)	No	#13A05D
tema_secundario	varchar(7)	No	#f57c00

26. ESTRUCTURA DE TABLA PARA LA TABLA VACUNACIONES

Columna	Tipo	Nulo	Predeterminado
<i>id</i>	int(11)	No	
id_producto	int(11)	No	
id_mascota	int(11)	No	
id_usuario	int(11)	No	
fecha	date	No	
tipo	tinyint(1)	No	
observacion	varchar(200)	No	
estado	tinyint(1)	No	1

26.1 Estructura Stand-in para la vista vacunaciones_view

Columna	Tipo	Nulo	Predeterminado
proxima	bigint(21)	Sí	NULL
id	int(11)	No	0
observacion	varchar(200)	No	
id_producto	int(11)	No	
id_mascota	int(11)	No	
id_usuario	int(11)	No	
fecha	date	No	
tipo	tinyint(1)	No	
estado	tinyint(1)	No	1
nombre	varchar(100)	No	
usuario	varchar(112)	No	

I.1.2.2.3.11 Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema.

Los diagramas de Componentes se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

Uno de los usos principales es que puede servir para ver que componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

I.1.2.2.3.11.1 Diagramas de Componente

I.1.2.2.3.11.1.1 Diagrama de Componente Gestionar Usuarios

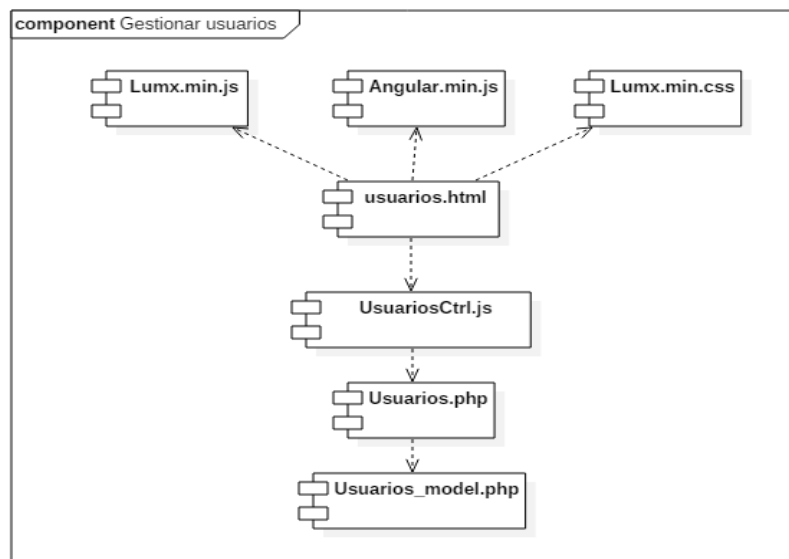


Ilustración 0.132 Diagrama de Componente Gestionar Usuarios

I.1.2.2.3.11.1.2 Diagrama de Componente Gestionar Roles

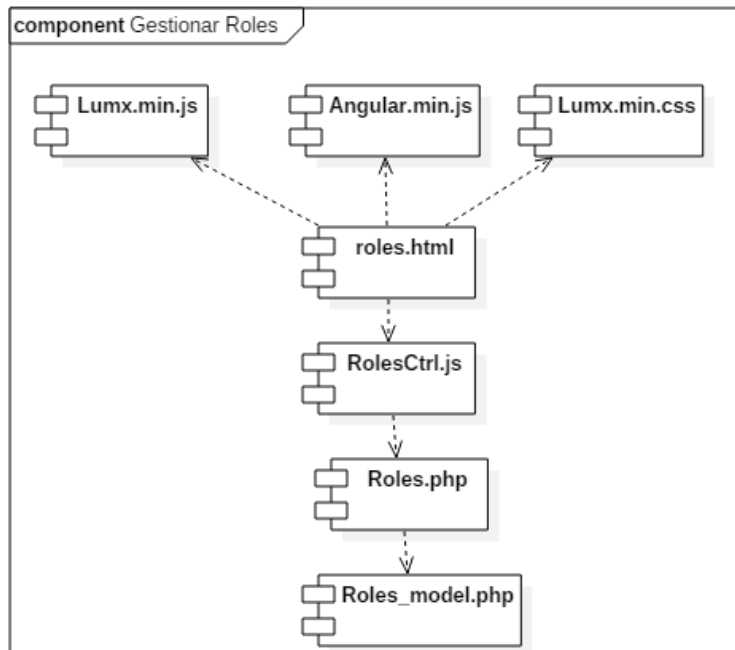


Ilustración 0.133 Diagrama de componente Gestionar Roles

I.1.2.2.3.11.1.3 Diagrama de componente Gestionar Noticias

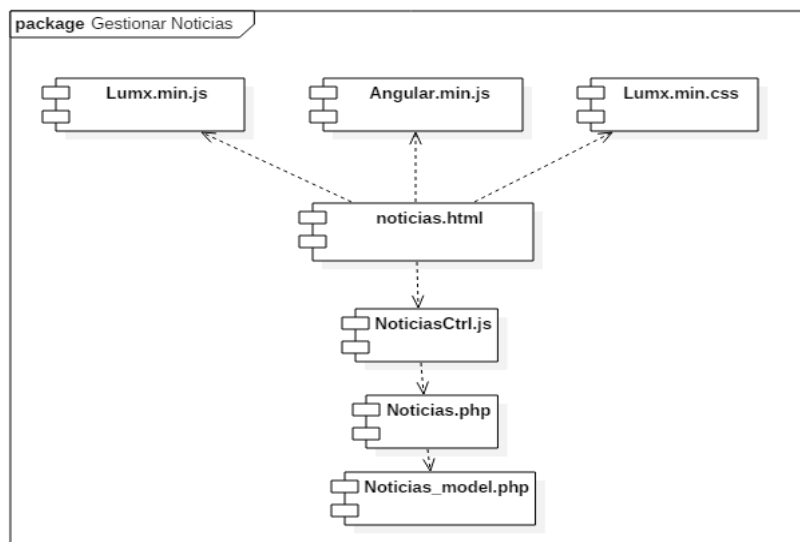


Ilustración 0.134 Diagrama de Componente Gestionar Noticias

I.1.2.2.3.11.1.4 Diagrama de Componentes Gestionar Clientes

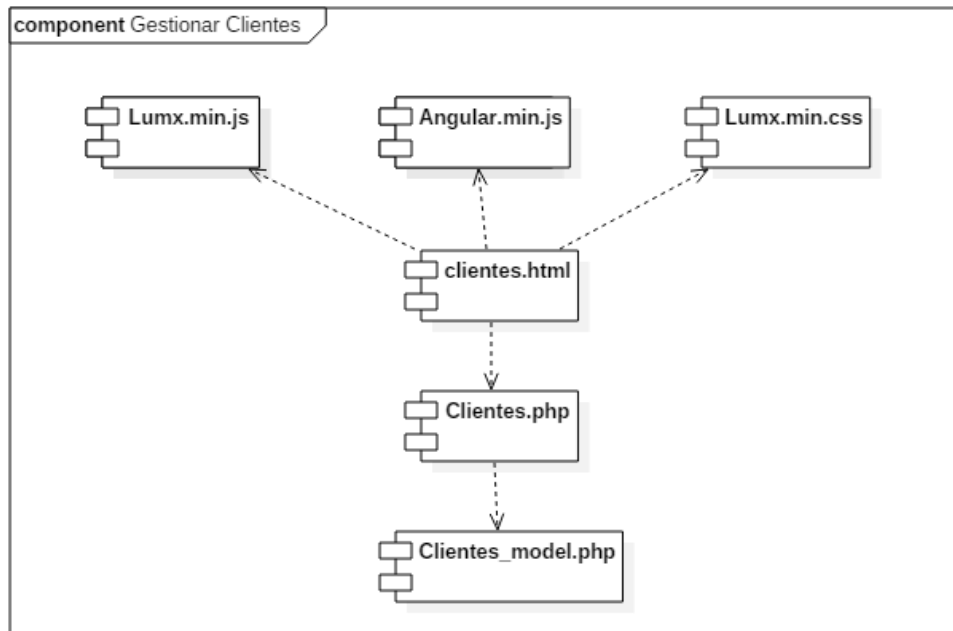


Ilustración 0.135 Diagrama de Componentes Gestionar Clientes

I.1.2.2.3.11.1.5 Diagrama de Componentes Gestionar Medicamentos

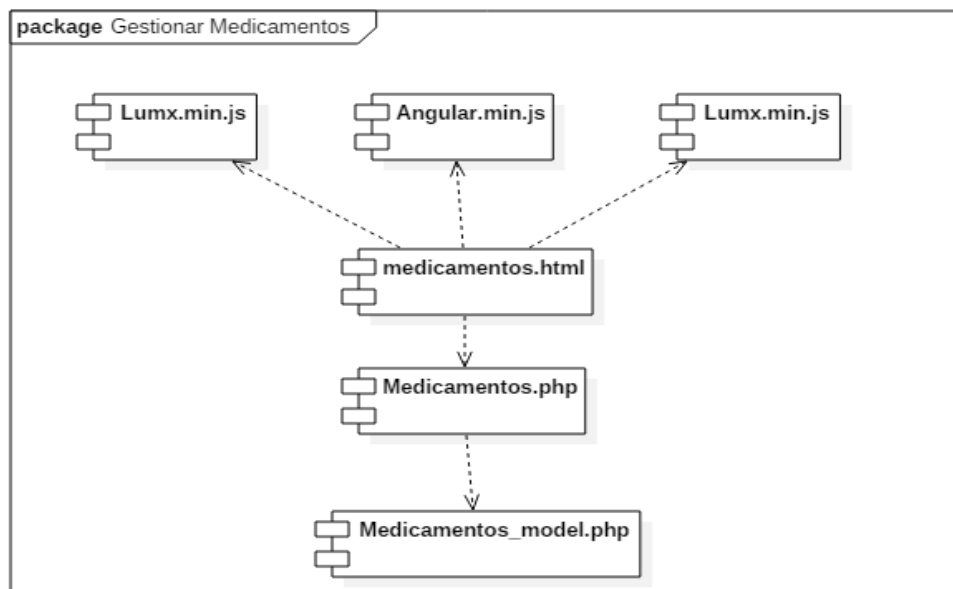


Ilustración 0.136 Diagrama de componente Gestionar Medicamentos

I.1.2.2.3.11.1.6 Diagrama de Componente Gestionar Citas

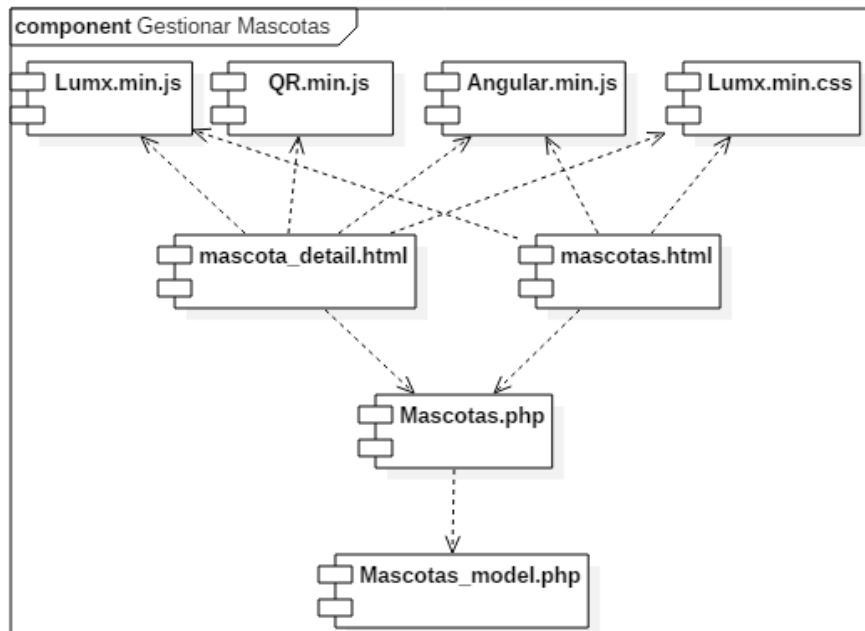


Ilustración 0.137 Diagrama de Componente Gestionar Citas

I.1.2.2.3.11.1.7 Diagrama de Componente Gestionar Mascotas

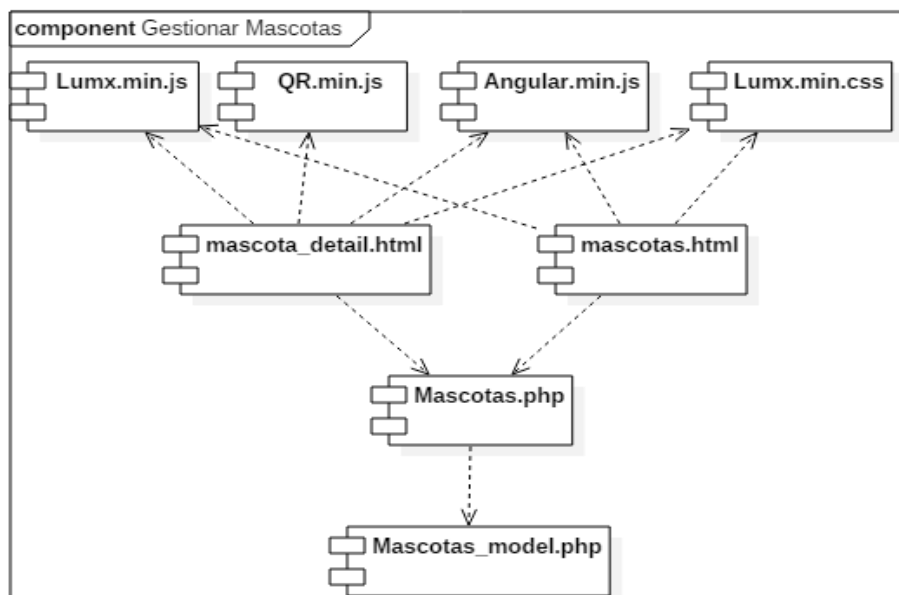


Ilustración 0.138 Diagrama de componente Gestionar Mascotas

I.1.2.2.3.11.1.8 Diagrama de Componente Gestionar Peluquerías

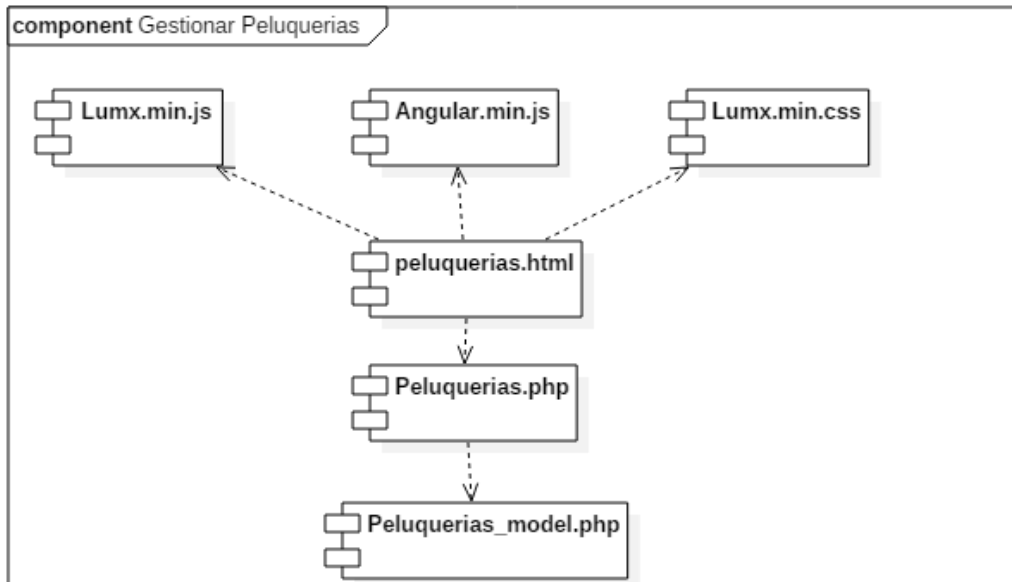


Ilustración 0.139 Diagrama de componente Gestionar Peluquerías

I.1.2.2.3.11.1.9 Diagrama de Componente Gestionar Control Sanitario

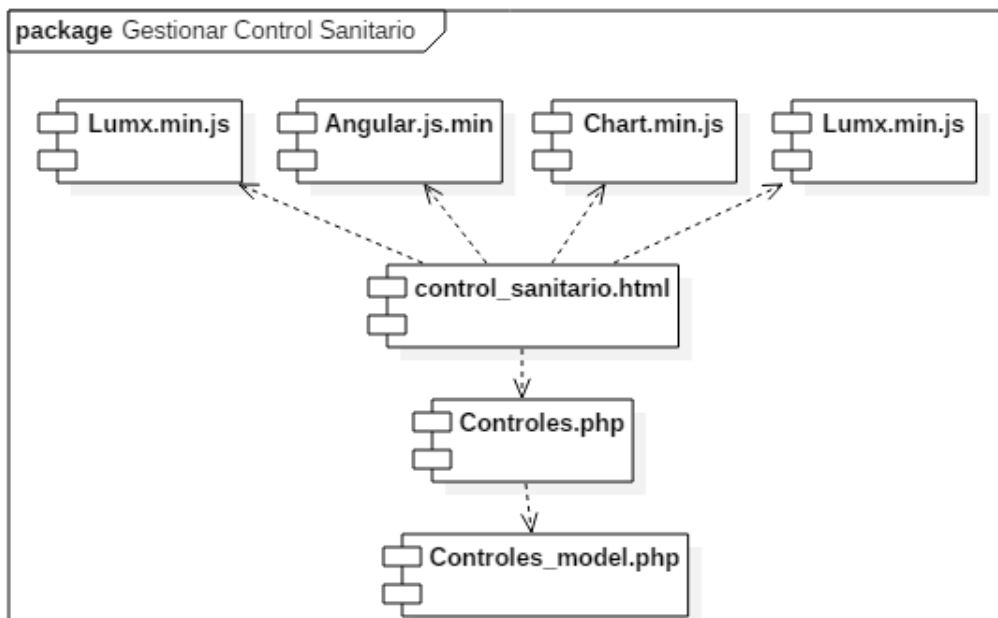


Ilustración 0.140 Diagrama de Despliegue Gestionar Control Sanitario

I.1.2.2.3.11.1.10 Diagrama de Componentes Gestionar Atención Clínica

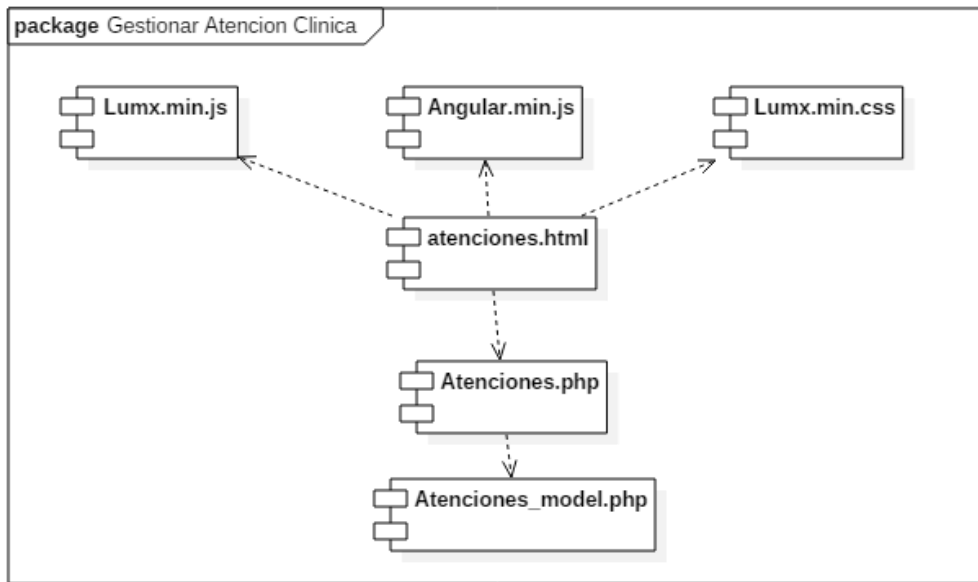


Ilustración 0.141 Diagrama de Componentes Gestionar Atención clínica

I.1.2.2.3.11.1.11 Diagrama de Componentes Generar Estadística

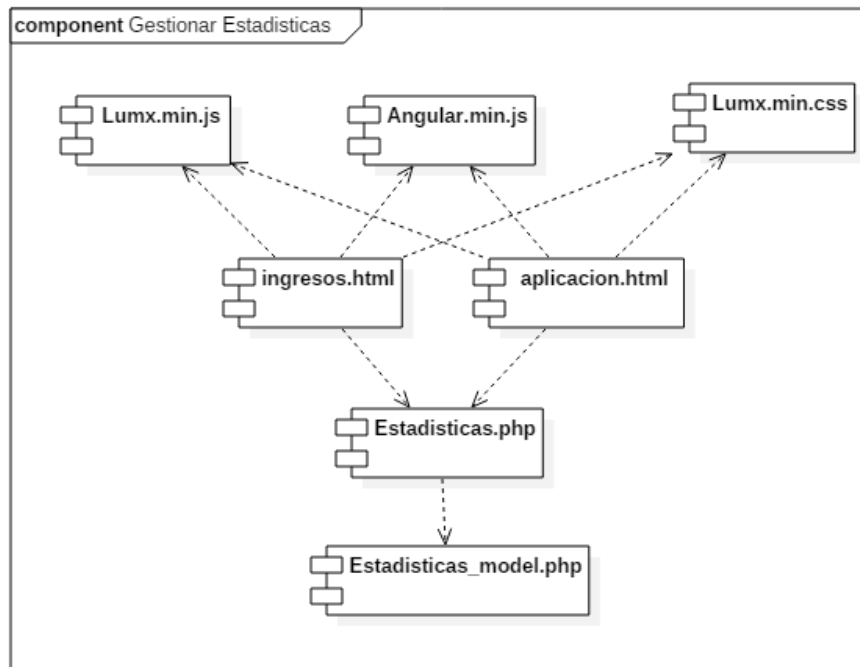


Ilustración 0.142 Diagrama de Componente generar estadísticas

I.1.2.2.3.12 Modelo de Despliegue

Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

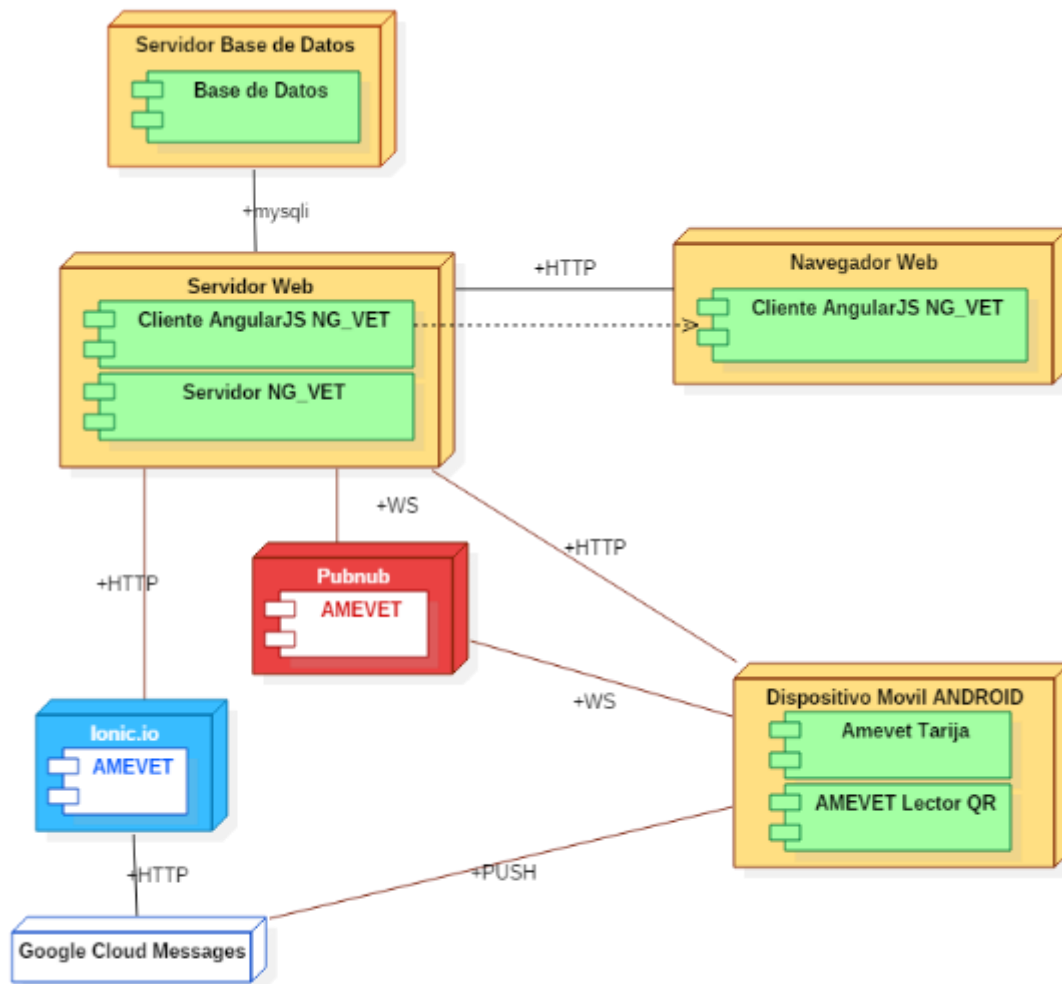


Ilustración 0.143 Diagrama de Despliegue

I.1.2.2.3.13 Casos de Prueba

El objetivo último del proceso de verificación y validación es establecer la seguridad de que el sistema software está “hecho para un propósito”. Esto significa que el sistema debe ser lo suficientemente bueno para su uso pretendido. El nivel de confianza requerido depende del

propósito del sistema, las expectativas de los usuarios del sistema y el entorno de mercado actual del sistema.³⁰

I.1.2.2.3.13.1OBJETIVOS

Las pruebas en un software son aplicadas como una etapa más del proceso de desarrollo del software y su objetivo es asegurar que el software cumpla con las especificaciones requeridas y eliminar los posibles defectos que éste pudiera tener.³¹

I.1.2.2.3.13.1.1 Objetivos de las Pruebas

- Encontrar defectos en el software.
- Una prueba es considerada exitosa si se descubre un defecto.
- Una prueba fracasa si hay un defecto y no las descubre.
- Asegurar la calidad del software.

I.1.2.2.3.13.2TIPOS DE PRUEBAS

I.1.2.2.3.13.2.1 Pruebas Unitarias

Las pruebas de unidades se centran en un módulo. Para ello, se usa la descripción del diseño detallado como guía, se prueban los caminos importantes de control, con la finalidad de descubrir errores dentro del ámbito del módulo. La prueba de unidad hace un uso intensivo de las técnicas de prueba de caja blanca.

I.1.2.2.3.13.2.2 Pruebas de Integración

El objetivo es tomar los módulos ya probados en las pruebas de unidad, y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. Existen dos formas de integración:

- **Integración No Incremental:** Se combinan todos los módulos por anticipado y se prueba todo el programa en conjunto.

³⁰ I. Sommerville, Ingeniería Del Software, Madrid: Pearson Educación S.A.

³¹CalidadSoftware.(s.f.).*CalidadSoftware*.Obtenido de http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.

- **Integración Incremental:** El programa se construye y se prueba en pequeños segmentos.

En la prueba de integración, el foco de atención es el diseño y la construcción de la arquitectura del software. Las técnicas que más prevalecen son las de diseño de casos de prueba de caja negra, aunque se pueden llevar a cabo unas pocas pruebas de caja blanca.

I.1.2.2.3.13.2.3 Pruebas de Sistema

Verifica que cada elemento se ajusta de forma adecuada, además de que se alcanza la funcionalidad y el rendimiento del sistema como un todo. La prueba del sistema está constituida por una serie de pruebas diferentes, cuyo propósito primordial es ejercitar profundamente el sistema.

I.1.2.2.3.13.3 ENFOQUES DE PRUEBAS

I.1.2.2.3.13.3.1 Prueba de Caja Blanca

Permiten examinar la estructura interna del programa. Se diseñan casos de prueba para examinar la lógica del programa. Este es un método de diseño de casos de prueba, en el que usa la estructura de control del diseño procedimental, para derivar casos de prueba que garanticen que:

- Se recorren todos los caminos independientes de cada módulo.
- Se ejecutan todas las decisiones lógicas.
- Se ejecutan todos los bucles.
- Se ejecutan las estructuras de datos internas.

Aun así, puede ser imposible cubrir el cien por ciento de los procesos o estados que pueda tener el sistema, ya que pueden existir fragmentos de código que nunca se llegará a ejecutar.

I.1.2.2.3.13.3.2 Pruebas de Caja Negra

Las pruebas se llevan a cabo sobre la interfaz del software, y es completamente indiferente el comportamiento interno y la estructura del programa. Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma adecuada.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.



Ilustración 11.1. Representación de una Caja Negra.

I.1.2.2.3.13.4 CASOS DE PRUEBAS

A continuación, se detallarán algunas de las pruebas realizadas con el enfoque de caja negra. Éstas tendrán un formato basados en una plantilla con los siguientes campos:

- **Módulo:** Indica el módulo que se le ejecuta al caso de prueba.
- **Descripción:** Una breve descripción del caso de prueba a tratar.
- **Resultado Esperado:** Es lo que se espera al concluir la prueba, éste puede fracasar o terminar con éxito.
- **Resultado Obtenido:** Es el resultado real obtenido al finalizar la prueba.

I.1.2.2.3.13.4.1 Pruebas Unitarias

Las pruebas unitarias persiguen comprobar el funcionamiento de alguna característica, para posteriormente realizar su integración, en donde lo determine el diseño de la aplicación.

Módulo	Descripción	R. Esperado	R. Obtenido
Clientes	Crear un cliente con celular repetido	Mensaje de error	Mensaje de error
	Buscar un cliente con un nombre	Mostrar todos los clientes con alcance	Mostrar todos los clientes con
Pacientes	Crear paciente sin completar los campos.	Muestra mensaje	Muestra mensaje indicando
Usuarios	Eliminar todos los usuarios incluyendo el	No debería permitir eliminar a todos	Se eliminan todos.
Log-in	Acceder al sitio copiando la URL	Mensaje de error indicando acceso restringido	Re direcciona a la página de

I.1.2.2.3.13.4.2 Pruebas de Integración

Estas pruebas persiguen principalmente la integración de todos los módulos del sistema.

Descripción	Resultado Esperado	Resultado Obtenido
Crear nuevo cliente y luego crearle un paciente.	Mensaje satisfactorio al crear cliente y paciente.	Crea cliente exitosamente, Luego crea paciente exitosamente.
Iniciar sesión con usuario de menor rango.	Mostrar solo los módulos disponibles para el rango.	Mostrar solo los módulos disponibles para el rango.
Acceder a Registros con Perfil	Ver solo los Registros referentes a peluquería	Muestra Registros de peluquería
Iniciar sesión utilizando datos erróneos.	Mensaje advirtiendo el Error.	Mensaje de error, y permanece en el log-in.

Tabla 0.77 Pruebas de integración

I.1.2.2.3.13.4.3 Pruebas de Sistema

Se centran principalmente en que se cumplan las funcionalidades de los requerimientos del sistema en conjunto con los módulos.

Descripción	Resultado Esperado	Resultado Obtenido
Cambio de nombre de mi usuario.	Mensaje de bienvenida Actualiza el nombre.	No se actualiza, hasta que inicie sesión nuevamente.
Copiar URL del sistema y pegarla cuando ya se ha iniciado la sesión	Muestra la pantalla principal del sistema	Muestra la pantalla principal del sistema.

Tabla 0.78 Pruebas de sistema

I.1.2.2.3.14 Solicitud de Cambio

Los cambios propuestos para los artefactos se formalizan mediante este documento. Mediante este documento se hace un seguimiento de los defectos detectados, solicitud de mejoras o cambios en los requisitos del producto. Así se provee un registro de decisiones de cambios, de su evaluación e impacto, y se asegura que éstos sean conocidos por el equipo de desarrollo. Los cambios se establecen respecto de la última baseline (el estado del conjunto de los artefactos en un momento determinado del proyecto) establecida. En nuestro caso no se presentaron solicitudes de cambio.

I.1.2.2.3.15 Plan de Iteración

Es un conjunto de actividades y tareas ordenadas temporalmente, con recursos asignados, dependencias entre ellas. Se realiza para cada iteración, y para todas las fases.

I.1.2.2.3.16 Lista de Riesgos

Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.

El análisis de riesgo está enfocado para revisar de forma global los riesgos que pueden influir a lo largo de todo el desarrollo del proyecto. Es por esto, que se identificará las amenazas y vulnerabilidades del sistema, con el fin de minimizar sus efectos. El propósito es integrar, el manejo de estos, para así mejorar el desarrollo y funcionamiento del sistema.

Se evaluarán los riesgos según su grado de probabilidad, como se detalla en la tabla a continuación:

Probabilidad
Muy Alto
Alto
Moderado
Bajo
Muy Bajo

Tabla 0.79 Rango de Probabilidad.

Además, se evaluarán los riesgos según su grado de impacto en el proyecto, según el siguiente criterio:

Impacto
Catastrófico
Crítico
Moderado
Tolerable
Despreciable

Tabla 0.80 Rango de Impacto.

I.1.2.2.3.16.1 Identificación

A continuación, se presentan los riesgos identificados, donde además se evalúa la probabilidad de que puedan ocurrir éstos, y el impacto con las notas antes mencionadas.

En la siguiente tabla se analizarán los riesgos que afectan al proyecto que se está desarrollando:

Riesgo	Probabilidad	Impacto
Mala estimación del tiempo de	Moderado	Catastrófico
Pérdida de tiempo en el inicio	Alto	Crítico
Comunicación con el cliente no es óptima	Moderado	Catastrófico
Cambio en los requerimientos	Moderado	Crítico
Falta de experiencia en tecnología	Bajo	Crítico
Demora en la toma de decisiones	Bajo	Crítico
Incumplimiento de compromisos	Moderado	Catastrófico
Los usuarios finales se resisten al	Muy Alto	Moderado
Planificación excesivamente	Bajo	Crítico
Cambiar herramientas a mitad de	Bajo	Crítico
Falta de compromiso	Bajo	Catastrófico
Desarrollador desmotivado	Moderado	Crítico
Sobrecarga de trabajo, largos horarios	Muy Bajo	Crítico

Diseño construido no sea el más	Moderado	Catastrófico
Realizar un proyecto que nadie lo	Muy Bajo	Tolerable
El servidor se cae	Bajo	Catastrófico
Los clientes de la clínica no saben descargar la aplicación móvil	Bajo	Moderado
Los clientes de la clínica no saben utilizar la aplicación móvil	Bajo	Moderado
No se coloca el Código QR en el collar de la mascota	Alto	Tolerable
El cliente no cuenta con una buena conexión a internet.	Moderado	Moderado

Tabla 0.81 Riesgos del Proyecto.

I.1.2.2.3.16.2 Mitigación y Contingencia

En esta sección, se analizarán de los riesgos anteriormente presentados aquellos que posean una Probabilidad: Muy Alto, Alto, o Moderado; y un Impacto: Catastrófico, Crítico, o Moderado. Todos éstos serán divididos en dos grupos: Mitigación (prevenir el problema) y Contingencia (resolverlo).

La siguiente tabla se resolverá los problemas que afectan al desarrollo del proyecto.

Riesgo	Mitigación	Contingencia
Mala estimación del tiempo de desarrollo	Se deberá evaluar el tiempo de desarrollo con claridad	Se deberá estimar el tiempo posterior con mayor claridad, y el tiempo perdido deberá ser recuperado
Pérdida de tiempo en el inicio	Se deberá tomar mayor importancia al proyecto.	Se deberá recuperar lo antes posible el tiempo perdido.
Comunicación con el cliente no es óptima	Comunicarse continuamente con el cliente.	Establecer compromisos de reuniones con anticipación
Cambio en los requerimientos	Se tendrán conversaciones constantes con el cliente.	Se deberá minimizar la carga del nuevo requerimiento

Incumplimiento de compromisos	de	Se deberán cumplir al día con todas las tareas acordadas.	Recalendarizar compromisos de manera ordenada y responsable.
Los usuarios finales se resisten al sistema	se	Realizar pruebas de usabilidad.	Impartir cursos de aprendizaje para uso el software.
Desarrollador desmotivado		Organizar el tiempo de trabajo y de descanso.	Realizar tareas que sean del agrado del desarrollador.
Diseño construido no sea el más adecuado		Se debe buscar ayuda del profesor guía.	Dedicar más tiempo para rediseñar la solución.
El cliente no cuenta con una buena conexión a internet		Realizar pruebas con conexiones lentas de internet	Programar la aplicación para que consuma los mínimos recursos del sistema.

Tabla 0.82 Mitigación y Contingencia del Proyecto.

I.1.2.2.3.17 Manual de Instalación

Este documento incluye las instrucciones para realizar la instalación del producto.

Se lo puede ver en el ANEXO 4 del documento Final.

I.1.2.2.3.18 Material de Apoyo al Usuario Final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación, Guías de Mantenimiento y Sistema de Ayuda en Línea.

El manual de Usuario se lo presenta en un documento separado a este.

I.1.2.2.4 Producto

Los ficheros del producto empaquetados y almacenadas en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción

es desarrollado incremental e iterativamente, obteniéndose una nueva release al final de cada iteración.

I.1.2.2.5 Evolución del Plan de Desarrollo del Software

El Plan de Desarrollo del Software se revisará semanalmente y se refinará antes del comienzo de cada iteración.

I.1.2.3 Organización del Proyecto

I.1.2.3.1 Participantes en el Proyecto

El personal del proyecto considerando las fases de Inicio, Elaboración y dos iteraciones de la fase de Construcción, estará formado por:

Jefe de Proyecto. Labor de Rolando Abelino Sanchez Garay (que también debe cumplir con otros cargos como de Analista de Sistema, Ingeniero de Software y Programador) alumno del último curso de la carrera de Ingeniería Informática en la Facultad de Ciencias y Tecnología de la Universidad Autónoma “Juan Misael Saracho”. Con una experiencia modesta en metodologías de desarrollo, herramientas CASE y notaciones, en particular la notación UML y el proceso de desarrollo RUP.

I.1.2.3.2 Interfaces Externas

No se definieron Interfaces externas.

I.1.2.3.3 Roles y Responsabilidades

A continuación, se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Puesto	Responsabilidad
--------	-----------------

Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.

I.1.2.4 Gestión del Proceso

I.1.2.4.1 Estimaciones del Proyecto

El presupuesto del proyecto y los recursos involucrados se adjuntan en un documento separado.

I.1.2.4.2 Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

I.1.2.4.2.1 Plan de las Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es sólo una aproximación muy preliminar)

Fase	Nro. Iteraciones	Duración
Fase de Inicio	1	4 semanas
Fase de Elaboración	1	8 semanas
Fase de Construcción	2	12 semanas
Fase de Transición	1	4 Semanas

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
Fase de Inicio	En esta fase desarrollará los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente / usuario

	del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera release de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Ambas iteraciones tendrán una duración de una semana.

Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis / Diseño. El producto se construye en base a 2 iteraciones, cada una produciendo una release a la cual se le aplican las pruebas y se valida con el cliente / usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la release 3.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios para pruebas beta.
Fase de Transición	En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

I.1.2.4.2.2 Calendario del Proyecto

A continuación, se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto, pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto.

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

Disciplinas / Artefactos generados o modificados durante la Fase de Inicio	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 1 23/2/15 – 30/2/15	Semana 4 27/3/15 – 3/4/15
Requisitos		
Glosario	Semana 1 23/2/15 – 30/2/15	Semana 4 27/3/15 – 3/4/15
Visión	Semana 2 1/3/15 – 8/3/15	Semana 4 27/3/15 – 3/4/15
Modelo de Casos de Uso	Semana 3 9/3/15 – 16/3/15	siguiente fase
Especificación de Casos de Uso	Semana 3 9/3/15 – 16/3/15	siguiente fase
Especificaciones Adicionales	Semana 3 9/3/15 – 16/3/15	siguiente fase
Análisis / Diseño		
Modelo de Análisis / Diseño	Semana 2 1/3/15 – 8/3/15	siguiente fase
Modelo de Datos	Semana 2 1/3/15 – 8/3/15	siguiente fase
Implementación		
Prototipos de Interfaces de Usuario	Semana 3 9/3/15 – 16/3/15	siguiente fase

Modelo de Implementación	Semana 3 9/3/15 – 16/3/15	siguiente fase
Pruebas		
Casos de Pruebas Funcionales	Semana 3 9/3/15 – 16/3/15	siguiente fase
Despliegue		
Modelo de Despliegue	Semana 3 9/3/15 – 16/3/15	siguiente fase
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 1.0 y planes de las Iteraciones	Semana 1 23/2/15 – 30/2/15	Semana 3 9/3/15 – 16/3/15
Ambiente	Durante todo el proyecto	

Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 2 20/3/15 – 30/3/15	aprobado
Requisitos		
Glosario	Semana 2 20/3/15 – 30/3/15	aprobado
Visión	Semana 3 1/4/15 – 10/4/15	aprobado

Modelo de Casos de Uso	Semana 5 30/4/15 – 10/5/15	Semana 8 22/8/15 – 28/5/15
Especificación de Casos de Uso	Semana 5 30/4/15 – 10/5/15	Semana 8 22/8/15 – 28/5/15
Especificaciones Adicionales	Semana 5 30/4/15 – 10/5/15	Semana 8 22/8/15 – 28/5/15
Análisis / Diseño		
Modelo de Análisis / Diseño	Semana 3 1/4/15 – 10/4/15	Revisar en cada iteración
Modelo de Datos	Semana 3 1/4/15 – 10/4/15	Revisar en cada iteración
Implementación		
Prototipos de Interfaces de Usuario	Semana 5 30/4/15 – 10/5/15	Revisar en cada iteración
Modelo de Implementación	Semana 5 30/4/15 – 10/5/15	Revisar en cada iteración
Pruebas		
Casos de Pruebas Funcionales	Semana 5 30/4/15 – 10/5/15	Revisar en cada iteración
Despliegue		
Modelo de Despliegue	Semana 5 30/4/15 – 10/5/15	Revisar en cada iteración
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 2.0 y planes de las Iteraciones	Semana 7 15/8/15 – 22/5/15	Revisar en cada iteración

Ambiente	Durante todo el proyecto
-----------------	--------------------------

Disciplinas / Artefactos generados o modificados durante la Fase de Construcción (Iteración 1)	Comienzo	Aprobación
Casos de Uso negociados para la Primera Release		
Caso de uso escogido	Semana 13	Semana 14

Disciplinas / Artefactos generados o modificados durante la Fase de Construcción (Iteración 2)	Comienzo	Aprobación
Casos de Uso negociados para la Primera Release		
Caso de uso escogido	Semana 15	Semana 16

I.1.2.4.3 Seguimiento y Control del Proyecto

Gestión de Requisitos

Los requisitos del sistema son especificados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto y por el Comité de Seguimiento y Control.

Control de Calidad

Los defectos detectados en las revisiones y formalizados también en una Solicitud de Cambio tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias. Para la revisión de cada artefacto y su correspondiente garantía de calidad se utilizarán las guías de revisión y checklist (listas de verificación) incluidas en RUP.

Gestión de Riesgos

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Esta lista será evaluada al menos una vez en cada iteración.

Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean accesibles a todo el participante en el proyecto. Al final de cada iteración se establecerá una baseline (un registro del estado de cada artefacto, estableciendo una versión), la cual podrá ser modificada sólo por una Solicitud de Cambio aprobada.

I.1.3 MEDIOS DE VERIFICACIÓN DEL COMPONENTE SISTEMA DE INFORMACIÓN

I.1.3.1 Carta de Conformidad



UNIVERSIDAD AUTÓNOMA
"JUAN MISAEL SARACHO"
CLÍNICA VETERINARIA "AMEVET"

Tarija, 12 febrero de 2016

A QUIEN CORRESPONDA

PRESENTE. -

De mi mayor consideración:

A través de la presente carta deseo saludar muy cordialmente y a la vez expresar mi conformidad con el sistema realizado por el Univ. **Rolando Abelino Sanchez Garay, C.I. 7160684** estudiante de último grado de la carrera de Ingeniería Informática para realizar el sistema web "**Sistema de Mejoramiento en la administración de la información de los pacientes de la clínica veterinaria "AMEVET" (Neg-VET)**".

Es cuando expreso mi conformidad con el sistema realizado para los fines que le convenga al interesado.

Sin otro particular me despido de usted muy atentamente.

Atentamente.


Dra. Ximena Jurado



Ilustración 144. Carta de Conformidad Sistema Web

I.1.4 ANEXOS DEL COMPONENTE

I.1.4.1 ANEXO 1: Documento Especificación de Requisitos de Software

Mejorar la administración de la información de los pacientes de la clínica veterinaria AMEVET, asistida por una plataforma tecnológica de comunicación con los clientes.

Especificación de Requerimientos de Software

Versión 1.1

Rolando Abelino Sanchez Garay

rolando.sanchez.developer@gmail.com

Historia de revisiones

Fecha	Versión	Descripción	Autor
20/05/2015	1.0	Versión Inicial, previa aprobación del docente.	Rolando Abelino Sanchez Garay
27/10/2015	1.1	Corrección de las observaciones realizadas por el docente de taller III.	Rolando Abelino Sánchez Garay

I.1.4.1.1 Introducción

Este documento es una especificación de requisitos software para la Clínica Veterinaria “AMEVET”. Todo su contenido ha sido elaborado en colaboración con el docente de taller III de informática de la Universidad Autónoma Juan Misael Saracho. Esta especificación se ha estructurado inspirándose en las directrices dadas por el estándar “IEEE Recommended Practice for Software Requirement Specifications ANSI/IEEE 830 1998”.

I.1.4.1.1.1 Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales y del sistema para la implementación de una aplicación WEB que permitirá administrar y consultar la información médica de las mascotas de la clínica veterinaria “AMEVET”.

El documento va dirigido al docente de Taller III. Este documento será el canal de comunicación entre las partes implicadas tomando parte en su confección miembros de cada parte. Esta especificación está sujeta a revisiones por el encargado del proyecto que se recogerán por medio de sucesivas versiones del documento, hasta su aprobación por la totalidad del docente de taller III. Una vez aprobado servirá de base para la construcción del nuevo sistema.

I.1.4.1.1.2 Alcance

La finalidad por la cual se realiza el desarrollo del sistema es de mejorar la interacción entre el cliente y la clínica veterinaria “AMEVET”. Definiremos al sistema como Neg-VET.

Es un sistema no es muy extenso para su desarrollo y de dificultad media en cuanto a su implementación: Hay que tener la información clínica de la mascota disponible en el mayor tiempo posible para que el usuario ya sea la clínica o el cliente acceda desde su Teléfono Inteligente

mediante una aplicación móvil o desde una computadora vía web (estos dispositivos deben estar conectados a internet).

El sistema no cubrirá en su totalidad la parte de la administración de la clínica como ventas, control de inventario, pero si contribuirá en el control de citas solicitadas a la clínica por parte del cliente, con la finalidad de optimizar la distribución horaria de la agenda y así brindar la disponibilidad de reservas necesarias.

Se tendrá que pasar por todas las etapas del Proceso Unificado de Desarrollo Software como así lo requiere la asignatura de taller III.

I.1.4.1.1.3 Definiciones, acrónimos y abreviaturas.

- **Sistema:** conformado por la suma de hardware, software y los usuarios.
- **Notificaciones.** - Son alertas del sistema que son notificadas a un usuario.
- **Personal de la clínica.** - Se refiere al Administrador, Medico, Peluquero.
- **ERS.** - Especificación de requisitos Software.
- **Android.** - Nombre de un sistema operativo de teléfonos inteligentes
- **Base de Datos.** - Cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar. La información se organiza en campos y registros. Un campo se refiere a un tipo o atributo de información, y un registro, a toda la información sobre un individuo.
- **Conexión.** - Comunicación entre dos entes que tienen características similares de comunicación.
- **Interfaz.** - Medio que permite la comunicación entre el usuario y el sistema.
- **Internet.** - interconexión de redes informáticas que permite a los ordenadores o computadoras conectadas comunicarse directamente, es decir, cada ordenador de la red

puede conectarse a cualquier otro ordenador de la red. El término suele referirse a una interconexión en particular, de carácter planetario y abierto al público, que conecta redes informáticas de organismos oficiales, educativos y empresariales.

- **Sistema Operativo.** - Software básico que controla una computadora o teléfono inteligente. El sistema operativo tiene tres grandes funciones: coordina y manipula el hardware del ordenador o teléfono inteligente y gestiona los errores de hardware y la pérdida de datos.
- **Hosting Compartido.** - Es un servidor de aplicaciones web que aloja y procesa la información de estas.
- **TCP/IP.** - Acrónimo de Transmisión Control Protocol/Internet Protocol (protocolo de control de transmisiones/protocolo de Internet), protocolos usados para el control de la transmisión en Internet. Permite que diferentes tipos de ordenadores o computadoras se comuniquen a través de redes heterogéneas.

I.1.4.1.1.4 Referencias

- IEEE Recomendad Practices for Software Requierements especification ANSI/IEEE 830 1998.
- Apuntes de clase de la asignatura “Ingeniería de Software I”.
- Apuntes de clase de la asignatura “Taller III”.

I.1.4.1.1.5 Visión general

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requerimientos que debe satisfacer el sistema.

I.1.4.1.2 Descripción general

En esta sección se presenta una descripción a alto nivel del sistema. Se presentarán las principales áreas de negocio a las cuales el sistema debe dar soporte, las funciones que el sistema debe realizar, la información utilizada, las restricciones y otros factores que afecten al desarrollo del mismo.

I.1.4.1.2.1 Perspectiva del producto

El sistema Neg-VET será un producto diseñado para trabajar en entornos WEB, lo que permitirá su utilización de forma descentralizada, además trabajará de manera independiente por lo tanto no interactuará con otros sistemas.

I.1.4.1.2.2 Funciones del producto

En términos generales, el sistema Neg-VET deberá proporcionar soporte a las siguientes tareas de gestión:

- **Gestión de Usuarios**
- **Gestión de Clientes**
- **Gestión de Mascotas**
- **Gestión de Citas**
- **Gestión de Peluquerías**
- **Gestión de Medicamentos**
- **Gestión de Noticias**
- **Gestión de Roles**
- **Generación de Reportes**

A continuación, se describirán con más detalle estas tareas y como serán soportadas por el sistema.

Gestión de Usuarios: Este proceso se encarga de gestionar la información de los usuarios que harán uso del sistema.

Gestión de Mascotas: Este proceso se encarga de gestionar la información de las mascotas como las próximas visitas, la ficha de atención clínica (Enfermedades, cirugías y Tratamientos) y la ficha de control sanitario (Desparasitaciones, Vacunaciones y controles de Pesos).

Gestión de Citas: Este proceso abarca desde que se solicita una reserva hasta que se otorga la misma.

- Se puede generar una cita por 3 vías: Se recepta la solicitud de atención del cliente en forma telefónica, mediante la aplicación móvil o personalmente.
- Las notificaciones de citas usando la aplicación móvil deben ser en tiempo real.

Gestión de Peluquerías: Este proceso comprende la gestión de peluquerías y estética canina que realiza la clínica veterinaria.

Gestión de Productos: Este proceso se encarga de la gestión de los medicamentos que se suministraran a las mascotas.

Gestión de Noticias: Este proceso conlleva gestionar las noticias de la clínica (eventos, anuncios, consejos, mascotas en adopción, etc.) que se mostrara en la sección de noticias de la aplicación móvil.

Gestión de Roles: Este proceso se encarga de la gestionar los roles y permiso que tendrá un usuario.

Generación de Reportes: Este proceso se encargará de generar reportes de los ingresos por peluquería y atenciones clínicas, así mismo otros reportes acerca de clientes que hacen uso de la aplicación móvil.

I.1.4.1.2.3 Características de los usuarios

Médico: Cuenta con un perfil en uso de internet y llenado de formularios web. Podrá encargarse de la gestión de Mascotas, Clientes, Productos, Novedades, Peluquerías y Citas.

Administrador de la Clínica: Es el Actor que maneja completamente el Sistema. Tendrá los privilegios de acceder a todas las tareas del sistema debe tener conocimiento en computación web y tecnología móvil.

Cliente: Debe tener conocimiento en el uso de un teléfono inteligente ANDROID. Puede acceder al sistema mediante la aplicación móvil y solicitar una cita, ver los datos de sus mascotas y de otras si posee el código para escanear mediante la aplicación, también podrá ver las novedades que publica la clínica y subir una foto de perfil de él y sus mascotas.

Peluquero: Cuenta con un perfil en uso de internet y llenado de formularios web. Podrá encargarse de la gestión de peluquerías.

I.1.4.1.2.4 Restricciones

I.1.4.1.2.4.1 Políticas reguladoras

La aplicación se desarrollará mediante software de licencia abierta por lo tanto no se deberá pagar por el uso de: servidor WEB (Apache), Sistema de Gestión de base de datos (MySQL) y el lenguaje de programación (PHP), por lo tanto, la utilización de estos programas se hará mediante las políticas establecidas por este tipo de licenciamiento.

I.1.4.1.2.4.2 Limitaciones de hardware

Para esta aplicación será necesario un hosting compartido puesto que la veterinaria no cuenta con un computador servidor y le resulta más económico contratar uno.

I.1.4.1.2.4.3 Interfaces con otras aplicaciones

El sistema Neg-VET no extrae ni manipula información de sistemas externos. Por tal motivo no se desarrollarán interfaces con otras aplicaciones.

Cabe recalcar que las aplicaciones móviles que recopilaran información del sistema son partes del mismo proyecto.

I.1.4.1.2.4.4 Funciones de control

El sistema Neg-VET debe controlar los permisos que tiene cada usuario para su accesibilidad de una manera correcta, de tal forma que pueda acceder la información que le corresponde de acuerdo a su rol.

I.1.4.1.2.4.5 Requisitos del lenguaje

Todo el material que se realice para el usuario y la aplicación debe de estar en lenguaje español.

I.1.4.1.2.4.6 Protocolos señalados

Se usará protocolos de comunicación TCP/IP, HTTP.

I.1.4.1.2.5 Supuestos y dependencias

- Se ejecutará en un navegador con soporte de HTML5, JavaScript y CSS3.
- La aplicación móvil Se ejecutará desde cualquier Smartphone igual o superior a ANDROID 4.1
- Los datos se almacenarán en una base de datos MySQL.
- El lenguaje de servidor elegido es PHP con el framework Codeigniter
- La aplicación web se alojará en un hosting compartido.
- Se necesita una conexión a internet para interactuar con el sistema.

I.1.4.1.2.6 Requisitos futuros

- Es viable una ampliación a otros módulos.

I.1.4.1.3 Requerimientos específicos

I.1.4.1.4 Interfaces Externas.

El sistema Neg-VET no tendrá interconexión con otros sistemas de información, por lo tanto, no es necesario la utilización de interfaz alguna.

La relación con un servidor WEB, DNS y Gestor de Base de Datos se hará a través de los archivos de configuración de éstos.

I.1.4.1.5 Requisitos Funcionales.

En esta sección se detallará las funcionalidades del nuevo sistema, y se clasificarán según las tareas de gestión del mismo.

I.1.4.1.5.1 Usuarios

- **RF001:** El sistema debe permitir listar, agregar, modificar y dar de baja los datos de un usuario.
- **RF002:** El sistema debe permitir buscar a un usuario según su Cedula de identidad, nombre, o apellido paterno o materno. Si el sistema encuentra resultados deberá mostrar a los que coinciden con la búsqueda.

I.1.4.1.5.2 Clientes

- **RF003:** El sistema debe listar, agregar, modificar y dar de baja los datos de un cliente.
- **RF004:** Si se da de baja a un cliente el sistema debe automáticamente dar de baja a las mascotas del mismo y eliminar la clave de acceso.
- **RF005:** El sistema debe permitir buscar a un cliente según su Cedula de identidad, nombre, o apellido paterno o materno. Si el sistema no encuentra al cliente, deberá mostrar un

mensaje indicando que no encontró resultados y si encuentra deberá mostrar los que coinciden con los criterios búsqueda.

- **RF006:** El sistema debe permitir agregar una clave de acceso siempre y cuando el cliente tenga registrado un número de teléfono celular.
- **RF007:** El sistema solo debe admitir eliminar y modificar una clave de acceso registrada.

I.1.4.1.5.3 Mascotas

- **RF008:** El sistema debe permitir listar, agregar, modificar y dar de baja los datos de una mascota.
- **RF009:** El sistema debe permitir buscar a una mascota según su nombre o código. Si el sistema no encuentra a la mascota, deberá mostrar un mensaje indicando que no encontró resultados y si encuentra deberá mostrar los que coinciden con la búsqueda.
- **RF010:** El sistema debe permitir generar e imprimir el código de la mascota en forma de código QR.
- **RF011:** El sistema debe permitir listar, agregar y eliminar las próximas visitas.
- **RF012:** El sistema debe permitir listar y agregar razas
- **RF013:** El sistema debe permitir listar y agregar especies

I.1.4.1.5.4 Ficha de Control Sanitario

- **RF014:** El sistema debe permitir imprimir la ficha de control sanitario de la mascota.
- **RF015:** El sistema debe permitir agregar desparasitaciones.
- **RF016:** El sistema debe permitir agregar primeras vacunas.
- **RF017:** El sistema debe permitir agregar revacunaciones.
- **RF018:** El sistema debe permitir ver los pesos de una mascota de manera gráfica para poder apreciar mejor su evolución en ciertas fechas.

I.1.4.1.5.5 Ficha Clínica de Atención

- **RF019:** El sistema debe permitir ver, agregar e imprimir una ficha de atención clínica.
- **RF020:** El sistema debe permitir listar los tratamientos.
- **RF021:** El sistema debe permitir agregar máximo 5 tratamientos.

I.1.4.1.5.6 Citas

- **RF022:** El sistema debe permitir agregar, eliminar, modificar, solicitar y confirmar una cita desde la fecha actual en adelante
- **RF023:** El sistema debe permitir notificar y listar las solicitudes de citas.
- **RF024:** El sistema debe permitir listar las citas confirmadas.
- **RF025:** El sistema debe recordar automáticamente al cliente que tiene una cita una hora y media antes de lo programado mediante una notificación por la aplicación móvil.
- **RF026:** El sistema debe informar al cliente que su cita sido confirmada o rechazada mediante una notificación por la aplicación móvil.
- **RF027:** El sistema debe permitir marcar asistencia a una cita solo si la fecha programada coincide con la fecha actual.

I.1.4.1.5.7 Peluquerías

- **RF028:** El sistema debe permitir listar, y agregar datos de una peluquería.
- **RF029:** El sistema debe permitir buscar peluquerías por clientes o por mascotas.

I.1.4.1.5.8 Medicamentos

- **RF030:** El sistema debe permitir listar, agregar, modificar y dar de baja los datos de un medicamento.
- **RF031:** El sistema debe permitir buscar un medicamento según su nombre. Si el sistema no encuentra el producto, deberá mostrar un mensaje indicando que no encontró resultados y si encuentra deberá mostrar los que coinciden con los criterios búsqueda.

I.1.4.1.5.9 Noticias

- **RF032:** El sistema debe permitir ver, agregar, eliminar, buscar, publicar y ocultar una noticia.
- **RF033:** Al momento de publicar una noticia el sistema debe enviar una notificación a todos los clientes que tienen instalada la aplicación móvil.
- **RF034:** El sistema debe permitir que los usuarios validados publiquen comentarios sobre una noticia.

I.1.4.1.5.10 Roles

- **RF035:** El sistema debe permitir ver y agregar roles de usuario.
- **RF036:** El sistema debe permitir eliminar y modificar los roles que no sean Administrador, Medico y Peluquero.
- **RF037:** El sistema debe permitir asignar y quitar procesos a los roles.

I.1.4.1.5.11 Estadísticas

- **RF038:** El sistema debe generar unas estadísticas graficas de forma anual de los ingresos por peluquería y atenciones clínicas mostrando solo el ingreso en Bs por Mes.
- **RF039:** El sistema debe generar una estadística para impresión con más detalle de todo el año de los ingresos por peluquería y atenciones clínicas.
- **RF040:** El sistema debe generar una estadística detallada de todos los dispositivos que tienen instalada la aplicación móvil.

I.1.4.1.6 Requisitos no funcionales

- El sistema debe permitir inicio de sesión segura, de tal manera que exista una correcta autenticación.
- El manejo de las notificaciones de citas debe ser en tiempo real.
- Debe indicar el formato correcto de la información que debe ingresar el usuario al sistema en caso de que no corresponda a lo solicitado.
- Cuando exista problema de conectividad el sistema deberá indicarlo mediante un mensaje.

I.1.4.1.6.1 Requisitos de Disponibilidad

El sistema deberá estar disponible 24(horas)/7(días) siempre y cuando los servidores donde se alberga estén funcionando correctamente.

I.1.4.1.6.2 Requisitos de Mantenimiento

El mantenimiento del sistema queda en manos del administrador del sistema, para lo cual se capacitará a esta persona. Si el mantenimiento va más allá de las funcionalidades del administrador la empresa deberá contactar al equipo de desarrollo del sistema.

I.1.4.1.6.3 Requisitos de Portabilidad

El portal web deberá funcionar en los sistemas operativos: Windows, Linux y Mac OS. Siempre y cuando la máquina cuente con un explorador (Google Chrome 30+, Mozilla Firefox 28+, Safari 12+ e Internet Explorer 8+).

La aplicación móvil debe funcionar correctamente en un Smartphone con SO ANDROID 4.1+.

I.1.4.1.6.4 Requisitos de Rendimiento

El tiempo de respuesta al usuario final va a depender mucho de la velocidad de conexión a internet con una velocidad de 1Mbps el tiempo de respuesta debe ser menor a 10seg con 200 usuarios simultáneos.

I.1.4.1.6.5 Requisitos de Desarrollo

El ciclo de vida elegido para desarrollar el producto será el de prototipo evolutivo orientado a objetos, de manera que se puedan incorporar fácilmente cambios y nuevas funciones, así como aprovechar las ventajas de reusabilidad proporcionada por el paradigma de orientación a objetos.

La metodología de desarrollo a utilizar será el proceso unificado de desarrollo y el lenguaje notacional UML.

I.1.4.1.6.6 Requisitos tecnológicos

	Requisitos de Windows	Requisitos de Mac	Requisitos de Linux	Requisitos de Android
Sistema operativo	Windows XP Service Pack 2 y versiones posteriores Windows Vista Windows 7 Windows 8 Windows 10	Mac OS X 10.6 y versiones posteriores	Ubuntu 12.04 y versiones posteriores Debian 7 y versiones posteriores OpenSuSE 12.2 y versiones posteriores Fedora Linux 17	Android 4.1 y versiones posteriores
Procesador	Intel Pentium 4 y posterior	Intel	Intel Pentium 4 o versiones posteriores	1Ghz

Espacio libre en disco	350 MB	20MB
RAM	512 MB	512MB

I.1.4.1.7 Atributos del sistema

I.1.4.1.7.1 Seguridad

Los usuarios deben autenticarse en el sistema para que se carguen los procesos correspondientes al rol de:

- **Administrador**
- **Medico**
- **Peluquero**
- **Cliente**

I.2 COMPONENTE 2: APLICACIÓN MÓVIL

I.2.1 INTRODUCCIÓN

Para hablar de lo que son los diferentes tipos de aplicaciones móviles y sus características, hay que dejar claro algo, que es una aplicación móvil. Entonces se puede decir que, una aplicación móvil, es una aplicación informática desarrollada para ser ejecutada a través de un dispositivo móvil inteligente, Tablet u otro para el cual se desee implementar. Estas se encuentran en tiendas, por medio de las cuales son accedidas por el público que desee usarlas.

Dentro de estas plataformas de distribución de las aplicaciones móviles, se podrá encontrar de dos tipos, gratis y pagas, pero eso es algo lo cual no es de interés en este momento. Pero es importante aclarar, que las aplicaciones móviles no todas, no tienen las mismas características, ni son del mismo tipo.

I.2.1.1 Aplicaciones Nativas

Las aplicaciones nativas son aquellas desarrolladas bajo un lenguaje y entorno de desarrollo específico, lo cual permite, que su funcionamiento sea muy fluido y estable para el sistema operativo que fue creada. Pero también es importante recordar, que todo en esta vida tiene su ventajas y desventajas, y que las aplicaciones nativas no son la excepción. Las ventajas y desventajas de estas son:

Ventajas

- Utilización de los recursos tanto del sistema como del hardware.
- Permite ser publicada en tiendas para su distribución.
- En su mayoría, no necesitan estar conectadas a Internet para su funcionamiento.

Desventajas

- Solo pueden ser utilizadas por un dispositivo que cuente con el sistema para el cual fue desarrollada.
- Requiere de un costo para distribuirla en una tienda, y dependiendo el sistema, para el uso del entorno de desarrollo.
- Necesitan aprobación para ser publicadas en la plataforma.

I.2.1.2 Aplicaciones Web

A continuación, se encuentran, las aplicaciones web o web application. Son aquellas desarrolladas usando lenguajes para el desarrollo web como lo son HTML, CSS y JavaScript y un framework para el desarrollo de aplicaciones web, como por ejemplo jQuery Mobile, Sencha, Kendo UI, entre otros. Se podría decir que este tipo de aplicaciones es muy usado para brindar accesibilidad a la información desde cualquier dispositivo, sin importar el sistema operativo, ya que solo se necesita contar con un navegador para acceder a esta. Las ventajas y desventajas de estas son:

Ventajas

- Pueden ser utilizadas desde cualquier dispositivo sin importar el sistema operativo.
- Puede que requiera un coste para su desarrollo, pero este puede ser mínimo en comparación con las nativas.
- No requieren de ninguna aprobación para su publicación.

Desventajas

- No pueden ser publicadas en plataformas para su distribución
- No utilizan los recursos del sistema ni del dispositivo de manera óptima.

I.2.1.3 Aplicaciones Híbridas

Por último están, las aplicaciones híbridas, como su nombre lo indica tienen un poco de cada tipo de las aplicaciones ya nombradas. Este tipo de aplicaciones se desarrolla utilizando lenguajes de desarrollo web y un framework dedicado para la creación de aplicaciones híbridas, como por ejemplo PhoneGap, Titanium Appcelerator, Sencha Touch, entre otros. La facilidad que brinda este tipo de desarrollo es que no hay un entorno específico el cual hay que utilizar para su desarrollo y la mayoría de las herramientas son de uso gratuito, también pudiendo integrarlo con las herramientas de aplicaciones nativas. Las ventajas y desventajas de este tipo de desarrollo de aplicaciones son:

Ventajas

- Uso de los recursos del dispositivo y del sistema operativo
- El costo de desarrollo puede ser menor que el de una nativa

- Son multiplataforma
- Permite distribución a través de las tiendas de su respectiva plataforma.

Desventaja

- La documentación puede ser un poco escasa y desordenada.
- +

Habiendo analizado cual es la mejor opción para implementar y considerando las características de la aplicación se ha optado por la tecnología de aplicaciones Híbridas.

I.2.2 EXPLORACIÓN E INICIALIZACIÓN

En esta sección se define el planteamiento y establecimiento del proyecto para sentar las bases para la implementación controlada el producto en relación con el desarrollo de software y lograr el éxito en las siguientes fases del proyecto, preparando y verificando todos los posibles problemas críticos de desarrollo y sean mitigados a tiempo.

Los objetivos en esta etapa son:

- Establecer los grupos de interés
- Definir los objetivos y el alcance para el proyecto de desarrollo de software
- Ganar un buen entendimiento global del producto.
- Prepara los recursos físicos, técnicos, humanos, y planes del proyecto.

Salida:

- Requerimientos iniciales.
- Plan de interacciones.
- Descripción de la línea de arquitectura.
- Diseño del sistema.
- Funcionalidad implementada (prototipo).
- Ilustraciones y notas sobre cada requerimiento.
- Historial de usuario.
- Pruebas de aceptación para cada requerimiento.

I.2.2.1 Requerimientos

I.2.2.1.1 Definición de los grupos de interés.

- Clientes: Son los clientes que se encuentran registrados en la clínica y cuentan con su clave de acceso.
- No clientes: Son personas que instalan la aplicación móvil solo con el fin de obtener más información de la clínica veterinaria.

I.2.2.1.2 Colección de requerimientos

Es una tarea en la cual los requerimientos para el producto son establecidos en nivel apropiado, los cuales pueden ser funcionales y no funcionales. El objetivo es producir la definición general inicial del producto, propósito y funcionalidad.

Los requerimientos iniciales fueron obtenidos de los grupos de interés, a través de una tormenta de ideas y encuestas; en bases a los resultados de la encuesta se identificó la importancia de cada requerimiento.

I.2.2.1.3 Requerimientos iniciales

Ver en el ANEXO 2 del componente II (Especificación de requisitos de software).

I.2.2.2 Establecimiento del proyecto

Esta etapa consiste en establecer los recursos físicos y técnicos para el proyecto, así como también el ambiente para monitorearlo. También se establecen los planes de entrenamiento del equipo de ser necesarios y se establecen los medios de comunicación con los grupos de interés.

I.2.2.2.1 Entrenamiento

Es una tarea en la cual el equipo de desarrollo se entrena de acuerdo a las necesidades específicas. Incluye procesos desarrollo, metodología, colección de datos, problemas técnicos relacionados como nuevas herramientas y métodos.

I.2.2.2.2 Planificación inicial

I.2.2.2.2.1 Análisis de requerimientos y pre-requisitos

Con el fin de realizar los requerimientos funcionales para el usuario es necesario cumplir con pre-requisitos a nivel técnico que permitan implementar el requerimiento funcional.

Pre-requisitos:

- Compartir noticia
 - Ver noticia
 - Listado de noticias
- Ver perfil de mascota
 - Listado de mascota
 - Iniciar Sesión
- Solicitar cita
 - Seleccionar horarios disponibles
 - Iniciar Sesión
- Ver Mensajes
 - Iniciar Sesión

I.2.2.3 Diseño del sistema

El diseño del sistema está formado por tres componentes principales: un web service y una aplicación móvil la cual acceden al web service a través de internet. El servidor que aloja el web service también contiene la base de datos (Componente 1).

I.2.2.3.1 Descripción de la interfaz de usuario

La aplicación móvil tiene el siguiente esquema de navegabilidad también conocido como “Storyboard”.

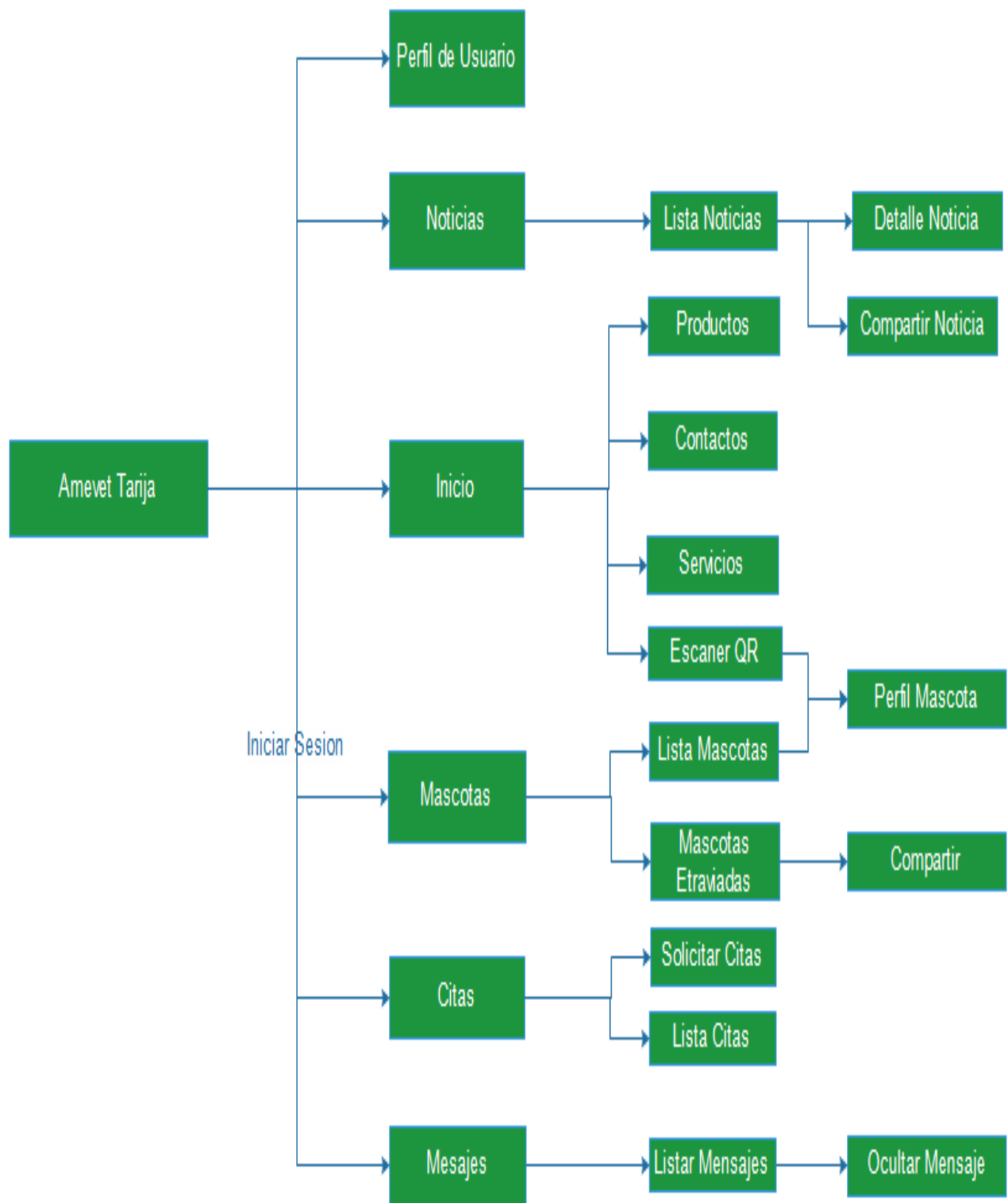


Ilustración 0.145 Storyboard Aplicación móvil

Pantalla principal

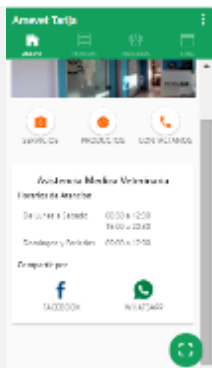


Ilustración 0.146 aplicación
pantalla principal

Es la primera pantalla a la cual accede el usuario y presenta las pestañas de inicio, noticias, mascotas, citas.

Como acción alterna en la parte superior derecha se encuentra un menú donde puede iniciar sesión o ver el perfil.

Contactos



Ilustración 0.147 aplicación
contactos

En esta pantalla se muestra la información para que el cliente pueda contactarse con la clínica.

Lista de Noticias



En esta pantalla se puede observar un listado de las noticias que publica la clínica.

Ilustración 0.148 Aplicación lista de noticias

Detalle de la Noticia



En esta sección se puede observar de manera más detallada una noticia con la opción de compartir esta mediante WhatsApp o Facebook.

Ilustración 0.149 Aplicación lista de noticias

Inicio de Sesión



En esta pantalla el cliente puede ingresar su número de celular y su clave de acceso para poder ver la información de sus mascotas y las citas.

Ilustración 0.150 Aplicación Inicio de Sesión

Perfil de Mascotas



En esta pantalla se observa la información de las próximas visitas, tratamientos, desparasitaciones, vacunaciones, revacunaciones y un control de peso gráfico.

Ilustración 0.151 de una mascota

Lista de citas realizadas



En esta pantalla se observa las citas que un cliente realizó.

Ilustración 0.152 aplicación lista citas

Calendario de Citas



Ilustración 0.153 Aplicación
calendario

Muestra un calendario de tres meses para que el cliente seleccione un día deseado.

Solicitar una cita



Ilustración 0.154 aplicación
solicitar una cita

En esta pantalla el cliente ingresa algunos datos para solicitar una cita a la clínica veterinaria.

Seleccionar Horario



Ilustración 0.155 aplicación
solicitar una cita

En esta pantalla el cliente puede seleccionar el horario de una cita que esté disponible.

Perfil del Cliente



Esta pantalla se puede observar el perfil del cliente como su nombre una foto de perfil y la opción de cambiar la clave de acceso y cerrar sesión

**Ilustración 0.156 aplicación perfil
cliente**

I.2.3 PRODUCCIÓN, ESTABILIZACIÓN Y PRUEBAS

En esta sección se implementa la funcionalidad e el producto aplicando un ciclo de desarrollo interactivo incremental; además se asegura la calidad del producto, haciendo uso de pruebas para comprobar que el sistema implementa las funcionalidades de forma correcta.

Los objetivos en esta etapa son:

- Implementar las funcionales órdenes de importancia para el usuario.
- Finalizar la implementación del producto.
- Mejorar y asegurar la calidad del producto.
- Finalizar la documentación de la aplicación móvil.
- Probar la aplicación basada en la documentación.
- Corregir los defectos.
- Producir un sistema libre de errores dentro de lo posible.

Salidas:

- Resultados de las pruebas de aceptación
- Documento final de la aplicación
- Una aplicación probada y con correcciones.

I.2.3.1 Implementación del sistema

En la codificación se seguirán los siguientes principios generales.

Claridad: se debe ser claro y corto en lo posible pero la claridad no debe sufrir por la brevedad. En lo general no se abrevian los nombres, inclusive si trata de nombres largos. Se debe evitar la ambigüedad en los nombres.

Consistencia: Se debe usar los nombres consecuentemente durante toda la codificación.

No auto referencia: los nombres no deben ser auto referentes.

I.2.3.2 Pruebas del sistema

- Para cada pantalla se prueba lo siguiente:
- Datos validos
- Valores límite
- Datos inválidos
- El diseño debe ser como esta en la documentación.
- Enlaces a otras pantallas funcionan como se describe en la documentación.

I.2.3.3 Análisis de resultados

I.2.3.3.1 Resultado de interacción 1

	Número de pruebas	porcentaje
Pruebas aceptadas	1	100%
Pruebas reprobadas	0	0
Total	1	100%
Pruebas corregidas	0	100%

I.2.3.3.2 Resultado de interacción 2

	Número de pruebas	porcentaje
Pruebas aceptadas	14	54%
Pruebas reprobadas	12	46%
Total	26	100%

Pruebas corregidas	12	100%
--------------------	----	------

I.2.3.3.3 Resumen de resultados de interacciones

	Número de pruebas	porcentaje
Pruebas aceptadas	15	41%
Pruebas reprobadas	12	59%
Total	27	100%
Pruebas corregidas	12	100%

I.2.3.3.4 Conclusiones de las pruebas

El 59% de las pruebas reprobadas en promedio se deben a textos diferentes al esperado en los mensajes de advertencia o de confirmación mostrados al usuario.

I.2.4 MEDIOS DE VERIFICACIÓN DEL COMPONENTE APLICACIÓN MÓVIL

En este apartado se implicarán los medios de verificación que fueron planteados en la matriz de marco lógico para el componente aplicación móvil.

I.2.4.1 Carta de Conformidad con la aplicación móvil



UNIVERSIDAD AUTÓNOMA
"JUAN MISAEL SARACHO"
CLÍNICA VETERINARIA "AMEVET"

Tarija, 12 febrero de 2016

A QUIEN CORRESPONDA

PRESENTE. -

De mi mayor consideración:

A través de la presente carta deseo saludar muy cordialmente y a la vez expresar mi conformidad con la aplicación móvil ANDROID realizada por el Univ. **Rolando Abelino Sanchez Garay, C.I. 7160654** estudiante de último grado de la carrera de Ingeniería Informática para realizar la aplicación móvil "Amevet Tarija".

Es cuando expreso mi conformidad con la aplicación realizada para los fines que le convenga al interesado.

Sin otro particular me despido de usted muy atentamente.

Atentamente.

Dra. Ximena Jurado



Ilustración 157. Carta de conformidad Aplicación Móvil

I.2.5 ANEXOS

I.2.5.1 ANEXO 1: Herramientas utilizadas para el desarrollo de la aplicación móvil.

Java JDK

Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en JAVA. Puede instalarse en una computadora local o en una unidad de red.

En la unidad de red se pueden tener las herramientas distribuidas en varias computadoras y trabajar como una sola aplicación.

JDK en sistemas Windows

En los sistemas Windows sus variables de entorno son

JAVAPATH: Es un path completo del directorio donde está instalado el JDK.

CLASSPATH: Son las librerías o clases de usuario.

PATH: variable donde se agrega la ubicación de JDK.

El JDK es el entorno de desarrollo oficial de Sun Microsystem

Componentes del JDK:

- Compilador java (javac.exe)
- Interprete java (java.exe)
- Interprete de applets (appletviewer.exe)
- Depurador java (jdb.exe)
- Generador de Documentación (javadoc.exe)
- Desensamblador (javap.exe)

- Integrador de c, c++ (JNI) (javah.exe).³²



Ilustración 0.158 Logo Java 8

Android SDK

Android SDK son las siglas de Android Software Development Kit. Es un kit de desarrollo con el que podremos desde desarrollar aplicaciones hasta ejecutar un emulador del sistema Android en la versión que queramos. Cabe decir que es imprescindible para efectuar acciones como el desbloqueo del arranque o utilizar el controlador adb.³³



Ilustración 0.159 Logo ANDROID SDK

NodeJS

Node.js es un entorno JavaScript de lado de servidor que utiliza un modelo asíncrono y dirigido por eventos.

Node usa el motor de JavaScript V8 de Google: una VM tremendamente rápida y de gran calidad escrita por gente como Lars Bak, uno de los mejores ingenieros del mundo especializados en VMs. No olvidemos que V8 es actualizado constantemente y es uno de los intérpretes más rápidos que

³² Salazar, G. M. (10 de Marzo de 2011). *TIC TAC*. Obtenido de Tecnologías Información y Comunicaciones Tamén Amigos y Colegas): <http://tic-tac.teleco.uvigo.es/profiles/blogs/java-development-kit-jdk>

³³ Obtenido de <http://android.scenebeta.com/noticia/android-sdk>
<http://developer.android.com/sdk/index.html>

puedan existir en la actualidad para cualquier lenguaje dinámico. Además, las capacidades de Node para I/O (Entrada/Salida) son realmente ligeras y potentes, dando al desarrollador la posibilidad de utilizar a tope la I/O del sistema.

Node soporta protocolos TCP, DNS y HTTP.³⁴



Ilustración 0.160 Logo Node JS

Git

Git (pronunciado "guit") es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

El mantenimiento del software Git está actualmente (2009) supervisado por Junio Hamano, quien recibe contribuciones al código de alrededor de 280 programadores.³⁵



³⁴ Muños, R. (1 de Mayo de 2011). Obtenido de <http://www.rmuno.net/introduccion-a-node-js.html>
<https://nodejs.org/en/download/>

³⁵ Wikipedia. (s.f.). Obtenido de <https://es.wikipedia.org/wiki/Git>
<http://git-scm.com/>

Genymotion

Genymotion es un emulador de Android que aprovecha la arquitectura x86 para ejecutar de forma fluida y rápida distintos dispositivos Android. Olvidando la lentitud del emulador nativo de Android podemos ejecutar todo tipo de aplicaciones y juegos en nuestro Windows, Mac o Linux.

Uno de los principales usos de Genymotion es facilitar el desarrollo de aplicaciones Android. Casi más de 900.000 usuarios registrados usan sus máquinas virtuales para crear aplicaciones usando Eclipse o IntelliJ. Se integran perfectamente con el adb, línea de comandos y los diferentes entornos de desarrollo. Podemos lanzar en apenas un par de segundos las aplicaciones conectadas al ADT desde el IDE para testearla. Mucho tiempo ahorrado.

Genymotion está basado en el uso de máquinas virtuales x86 optimizadas para correr sobre Virtualbox. A muchos les sonarán proyectos similares, pero la gente de Genymotion ha conseguido crear una interfaz simple capaz de soportar distintas funcionalidades accesibles a cualquier usuario, sin olvidar a los desarrolladores (su principal target y modelo de negocio).³⁶



Ilustración 0.162 Logo Genymotion

Apache Ant

Apache Ant es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción (build).

Esta herramienta, hecha en el lenguaje de programación Java, tiene la ventaja de no depender de las órdenes del shell de cada sistema operativo, sino que se basa en archivos de configuración XML

³⁶ Obtenido de <http://www.xatakandroid.com/roms-android/genymotion-el-emulador-mas-rapido-de-android>
<https://www.genymotion.com/#!/download>

y clases Java para la realización de las distintas tareas, siendo idónea como solución multi-plataforma.

La diferencia más notable entre Ant y Make es que Ant utiliza XML para describir el proceso de generación y sus dependencias, mientras que Make utiliza formato makefile. Por defecto, el archivo XML se denomina build.xml.

Ant es un proyecto de la Apache Software Foundation. Es software open source, y se lanza bajo la licencia Apache Software.³⁷



Ilustración 0.163 Logo Apache Ant

GapDebug

Es una herramienta para depurar errores en aplicaciones móviles híbridas realizadas con el framework Apache cordova.³⁸



Ilustración 0.164 Logo GapDebug

³⁷ Obtenido de https://es.wikipedia.org/wiki/Apache_Ant

³⁸ <https://www.genuitec.com/products/gapdebug/download/>

Aplicación Movil Android Amevet Tarija.

Especificación de Requerimientos de Software

Versión 1.1

Rolando Abelino Sanchez Garay

rolando.sanchez.developer@gmail.com

Historia de revisiones

Fecha	Versión	Descripción	Autor
20/05/2015	1.0	Versión Inicial, previa aprobación del docente.	Rolando Abelino Sanchez Garay
27/10/2015	1.1	Corrección de las observaciones realizadas por el docente de taller III.	Rolando Abelino Sánchez Garay

I.2.5.2.1 Introducción

Este documento es una especificación de requisitos software para la Clínica Veterinaria “AMEVET”. Todo su contenido ha sido elaborado en colaboración con el docente de taller III de informática de la Universidad Autónoma Juan Misael Saracho. Esta especificación se ha estructurado inspirándose en las directrices dadas por el estándar “IEEE Recommended Practice for Software Requirement Specifications ANSI/IEEE 830 1998”.

I.2.5.2.1.1 Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales y del sistema para la implementación de una aplicación móvil que permitirá consultar la información médica de las mascotas de la clínica veterinaria “AMEVET” por parte de los clientes, así como poder solicitar citas y estar al tanto con las noticias y eventos de la clínica.

El documento va dirigido al docente de Taller III. Este documento será el canal de comunicación entre las partes implicadas tomando parte en su confección miembros de cada parte. Esta especificación está sujeta a revisiones por el encargado del proyecto que se recogerán por medio de sucesivas versiones del documento, hasta su aprobación por la totalidad del docente de taller III. Una vez aprobado servirá de base para la construcción del nuevo sistema.

I.2.5.2.1.2 Alcance

La finalidad por la cual se realiza el desarrollo de la aplicación móvil es de mejorar la comunicación entre el cliente y la clínica veterinaria “AMEVET”. Definiremos al sistema como Neg-VET.

Es un sistema no es muy extenso para su desarrollo y de dificultad media en cuanto a su implementación: Hay que tener la información clínica de la mascota disponible en el mayor tiempo

posible para que el cliente acceda desde su Teléfono Inteligente mediante una aplicación móvil (este dispositivo debe estar conectado a internet).

El sistema no cubrirá en su totalidad la parte de la administración de la clínica como ventas, control de inventario, pero si contribuirá en el control de citas solicitadas a la clínica por parte del cliente, con la finalidad de optimizar la distribución horaria de la agenda y así brindar la disponibilidad de reservas necesarias.

I.2.5.2.1.3 Definiciones, acrónimos y abreviaturas.

- **Sistema:** conformado por la suma de hardware, software y los usuarios.
- **Notificaciones Push.** - Son mensajes en tiempo real que se muestran en el teléfono inteligente ANDROID.
- **ERS.** - Especificación de requisitos Software.
- **Android.** - Nombre de un sistema operativo de teléfonos inteligentes
- **Base de Datos.** - Cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar. La información se organiza en campos y registros. Un campo se refiere a un tipo o atributo de información, y un registro, a toda la información sobre un individuo.
- **Conexión.** - Comunicación entre dos entes que tienen características similares de comunicación.
- **Interfaz.** - Medio que permite la comunicación entre el usuario y el sistema.
- **Internet.** - interconexión de redes informáticas que permite a los ordenadores o computadoras conectadas comunicarse directamente, es decir, cada ordenador de la red puede conectarse a cualquier otro ordenador de la red. El término suele referirse a una interconexión en particular, de carácter planetario y abierto al público, que conecta redes informáticas de organismos oficiales, educativos y empresariales.

- **Sistema Operativo.** - Software básico que controla una computadora o teléfono inteligente. El sistema operativo tiene tres grandes funciones: coordina y manipula el hardware del ordenador o teléfono inteligente y gestiona los errores de hardware y la pérdida de datos.

I.2.5.2.1.4 Referencias

- IEEE Recomendad Practices for Software Requierements especification ANSI/IEEE 830 1998.
- Apuntes de clase de la asignatura “Ingeniería de Software I”.
- Apuntes de clase de la asignatura “Taller III”.

I.2.5.2.1.5 Visión general

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requerimientos que debe satisfacer el sistema.

I.2.5.2.2 Descripción general

En esta sección se presenta una descripción a alto nivel del sistema. Se presentarán las principales áreas de negocio a las cuales el sistema debe dar soporte, las funciones que el sistema debe realizar, la información utilizada, las restricciones y otros factores que afecten al desarrollo del mismo.

I.2.5.2.2.1 Perspectiva del producto

La aplicación AMEVET Tarija será un producto diseñado para trabajar en entornos móviles ANDROID, esta aplicación debe interactuar con el sistema mencionado en el componente uno.

I.2.5.2.2.2 Funciones del producto

En términos generales, el sistema Neg-VET deberá proporcionar soporte a las siguientes tareas de gestión:

- **Citas**
- **Noticias**
- **Mascota**
- **Perfil del Cliente**

A continuación, se describirán con más detalle estas tareas y como serán soportadas por el sistema.

Citas: Este proceso se encarga de listar y solicitar citas del cliente.

Mascotas: Este proceso se encarga de listar la información de las mascotas que pertenecen a un cliente determinado.

Noticias: Este proceso conlleva listar las noticias de la clínica (eventos, anuncios, consejos, mascotas en adopción, etc.) con la posibilidad de compartir el contenido por WhatsApp o Facebook.

Perfil del cliente: Este proceso es de editar datos personales del cliente como la clave, foto o iniciar o cerrar sesión de su cuenta.

I.2.5.2.2.3 Características de los usuarios

Cliente: Debe tener conocimiento en el uso de un teléfono inteligente ANDROID. Puede acceder al sistema mediante la aplicación móvil y solicitar una cita, ver los datos de sus mascotas y de otras si posee el código para escanear mediante la aplicación, también podrá ver las novedades que publica la clínica y subir una foto de perfil de él y sus mascotas.

I.2.5.2.2.4 Restricciones

I.2.5.2.2.4.1 Políticas reguladoras

La aplicación se desarrollará mediante software de licencia abierta por lo tanto no se deberá pagar por el uso de: servidor WEB (Apache), Sistema de Gestión de base de datos (MySQL) y el lenguaje de programación (PHP), por lo tanto, la utilización de estos programas se hará mediante las políticas establecidas por este tipo de licenciamiento.

I.2.5.2.2.4.2 Limitaciones de hardware

Para esta aplicación será necesario un dispositivo con conexión a internet.

I.2.5.2.2.4.3 Interfaces con otras aplicaciones

Cabe recalcar que la aplicación móvil que recopilaran información del sistema son partes del mismo proyecto.

I.2.5.2.2.4.4 Funciones de control

El sistema Neg-VET debe controlar los permisos que tiene cada usuario para su accesibilidad de una manera correcta, de tal forma que pueda acceder la información que le corresponde como cliente.

I.2.5.2.2.4.5 Requisitos del lenguaje

Todo el material que se realice para el usuario y la aplicación debe de estar en lenguaje español.

I.2.5.2.2.4.6 Protocolos señalados

Se usará protocolos de comunicación TCP/IP, HTTP.

I.2.5.2.2.5 Supuestos y dependencias

- La aplicación móvil Se ejecutará desde cualquier Smartphone igual o superior a ANDROID 4.1
- Los datos se almacenarán en una base de datos MySQL.
- El lenguaje de servidor elegido es PHP con el framework Codeigniter
- Se necesita una conexión a internet para interactuar con el sistema.

I.2.5.2.2.6 Requisitos futuros

- Es viable una integración con registro automático por Facebook.

I.2.5.2.3 Requerimientos específicos

I.2.5.2.4 Interfaces Externas.

La relación con un servidor WEB, DNS y Gestor de Base de Datos se hará a través de los archivos de configuración de éstos.

I.2.5.2.5 Requisitos Funcionales.

En esta sección se detallará las funcionalidades de la aplicación móvil, y se clasificarán según las tareas de gestión de la misma.

Los requerimientos iniciales identificados son los siguientes:

- Listado de noticias.
- Listado de mascotas.
- Perfil de Mascota.
- Ver Mascotas Extraviadas.
- Ver los Mensaje que tiene cada cliente.
- Reportar extravió de una mascota.
- Reportar hallazgo de una mascota extraviada.
- Listado de Citas realizadas.
- Solicitar una cita.
- Cambiar clave de acceso.
- Compartir noticia.

En base a los requerimientos iniciales se ha podido identificar los diferentes módulos de acuerdo a cada aplicación del sistema, listados a continuación.

- 1) Modulo Noticias

- a. Listado de noticias
- b. Compartir noticia
- 2) Modulo Mascotas
 - a. Listado de Mascotas Extraviadas
 - b. Listado de mascotas
 - c. Ver perfil de mascota
 - d. Reportar extravío
 - e. Reportar hallazgo
 - f. Escanear código QR
 - g. Subir foto
- 3) Modulo Citas
 - a. Listado de citas realizadas.
 - b. Solicitar cita.
- 4) Modulo Perfil
 - a. Subir foto
 - b. Cambiar clave de acceso
- 5) Modulo Mensajes
 - a. Listar los mensajes
 - b. Ocultar Mensajes

I.2.5.2.6 Supuestos y dependencias

- Para el correcto funcionamiento es importante que los usuarios tengan los conocimientos básicos de uso de un teléfono inteligente (ANDROID).
- Los datos se toman del sistema web de la clínica veterinaria AMEVET (Componente 1).
- Los usuarios manejan como idioma principal el español, siendo este el lenguaje que se opera en la interfaz de usuario.

I.2.5.2.7 Requisitos no funcionales

- La aplicación solo se puede utilizar en dispositivos ANDROID versión 4.1.1
- Es necesaria una conexión a internet.

I.2.5.2.7.1 Requisitos de Disponibilidad

El sistema deberá estar disponible 24(horas)/7(días) siempre y cuando los servidores donde se alberga la información de la clínica estén funcionando correctamente.

I.2.5.2.7.2 Requisitos de Mantenimiento

El mantenimiento del sistema queda en manos del desarrollador puesto que se debe de informar al cliente si existen actualizaciones de la aplicación móvil.

I.2.5.2.7.3 Requisitos de Portabilidad

La aplicación móvil deberá funcionar en el Sistema Operativo ANDROID 4.1 en adelante.

I.2.5.2.7.4 Requisitos de Rendimiento

El tiempo de respuesta al usuario final va a depender mucho de la gama de dispositivo móvil donde se instale la aplicación móvil.

I.2.5.2.7.5 Requisitos de Desarrollo

El ciclo de vida elegido para desarrollar el producto será el de prototipo evolutivo orientado a objetos, de manera que se puedan incorporar fácilmente cambios y nuevas funciones, así como aprovechar las ventajas de reusabilidad proporcionada por el paradigma de orientación a objetos.

La metodología de desarrollo a utilizar será MOBILE-D.

I.2.5.2.7.6 Requisitos tecnológicos

Requisitos de Android	
Sistema operativo	Android 4.1 y versiones posteriores
Procesador	1Ghz
Espacio libre en disco	5MB
RAM	512MB

I.2.5.2.8 Atributos del sistema

I.2.5.2.8.1 Seguridad

Los clientes deben autenticarse en el sistema para que se habiliten los procesos citas y mascotas.

I.3 COMPONENTE 3: SOCIALIZACIÓN DE LOS COMPONENTES ANTERIORES

I.3.1 CAPACITACIÓN EN EL USO DEL SISTEMA INFORMÁTICO NEG-VET

I.3.1.1 Introducción

El objetivo de este componente es capacitar a los usuarios en el uso del sistema NEG-VET según el nivel de los mismos empleando métodos y medios de enseñanza–aprendizaje adecuados.

El propósito del proyecto es: mejorar la administración de la información de los pacientes de la clínica veterinaria AMEVET, apoyada por una plataforma de comunicación con los clientes apoyada por las TIC; la capacitación en el uso del sistema informático al personal afectado por el proyecto se convierte en un componente fundamental para el logro del mismo.

El componente capacitación, se encamina hacia el siguiente objetivo: usar adecuadamente el sistema informático NEG-VET por el personal de la clínica veterinaria AMEVET explotando las fortalezas del mismo.

La Capacitación será presencial dada la corta duración de la misma, la disponibilidad de ambientes, de materiales didácticos y la importancia de posibilitar que el alumno (usuario) reciba asesoramiento oportuno ante cualquier consulta.

I.3.1.2 Contexto

La Capacitación se desarrollará en dos partes: la primera parte tiene como objetivo que el personal a capacitar conozca en forma global los alcances y beneficios que el sistema informático NEG-VET aporta a la clínica veterinaria AMEVET, así como los cambios positivos y responsabilidades que esto implica para la empresa.

Se realizarán actividades de capacitación personalizadas de acuerdo al rol que a cada uno le compete.

En este contexto el Capacitador confeccionó la Guía para Capacitación tomando en cuenta los diferentes niveles de preparación del usuario final.

El rol del capacitador estará en función a las categorías de los usuarios según el siguiente detalle:

Nivel ejecutivo:

- Se mostrará la importancia de la capacitación, objetivos y participación del personal seleccionado.

Usuarios Finales

I.3.1.3 Propuesta pedagógica

La propuesta pedagógica a utilizar dada las características de los usuarios del sistema NEG-VET tendrá en cuenta sus particularidades, el rol que juega dentro de la organización y niveles de conocimiento.

Los métodos de enseñanza a utilizar pondrán su énfasis principalmente en tres teorías de aprendizajes: la cognitiva, con su máximo exponente en el constructivismo, la colaborativa, fundamentalmente para ser explotada con intensidad en la formación del personal técnico y finalmente la significativa, aunque también estará presente en la formación del personal de las categorías de nivel ejecutivo y de usuarios finales.

Finalmente se pone de manifiesto el aprendizaje significativo porque el alumno tiene que incorporar los nuevos conocimientos en forma sustantiva en su estructura cognitiva. Esto se logra cuando el alumno relaciona los nuevos conocimientos con los anteriormente adquiridos; pero también es necesario que el alumno se interese por aprender lo que se le está mostrando. De esta forma el alumno no solo obtendrá resultados satisfactorios en un trabajo final, sino que será capaz de enfrentarse a diversas situaciones donde podrá aplicar los conocimientos adquiridos.

I.3.1.4 Contenidos de la capacitación

I.3.1.4.1 Unidad I: Introducción al Sistema

Lección 1: Ingresar al Sistema.

En esta lección el usuario aprenderá a ingresar al sistema autenticándose con su nombre de usuario y clave de acceso.

Lección 2: Familiarización con la navegación.

En esta lección el usuario aprenderá a navegar por las diferentes páginas del sistema utilizando el menú lateral y el botón de retroceso.

Lección 3: Administrar el perfil

En esta lección el usuario aprenderá a cambiar sus datos personales, subir una fotografía de perfil, cambiar su clave de acceso al sistema y la apariencia de la interfaz gráfica del sistema.

Lección 4: Utilizar el Sistema de Ayuda

En esta sección el usuario aprenderá a utilizar el sistema de ayuda que se encuentra disponible en cada pantalla del sistema.

I.3.1.4.2 Unidad II: Tareas del Sistema

Lección 5: Roles y Procesos

En esta lección el usuario aprenderá a manejar las operaciones que se pueden realizar en la sección de roles como la creación, modificación y eliminación de estos, así mismo asignar y remover los procesos a cada uno.

Lección 6: Usuarios

En esta lección el usuario aprenderá a manejar las operaciones que se pueden realizar en la sección de usuarios como la creación, modificación, eliminación, búsqueda y cambiar la clave de acceso de estos.

Lección 7: Medicamentos

En esta lección el usuario aprenderá a manejar las operaciones que se pueden realizar en la sección de medicamentos como la creación, modificación, eliminación y búsqueda de estos.

Lección 8: Noticias

En esta lección el usuario aprenderá a manejar las operaciones que se pueden realizar en la sección de noticias como la creación, dada de baja y compartir una noticia

Lección 9: Clientes

En esta lección el usuario aprenderá a manejar las operaciones que se pueden realizar en la sección de clientes como la creación, modificación, eliminación, búsqueda agregar clave, eliminar clave y modificar clave

Lección 10: Mascotas

En esta lección el usuario aprenderá a manejar las operaciones que se pueden realizar en la sección de mascotas como la creación, eliminación, modificación y búsqueda.

Lección 11: Citas

En esta lección el usuario aprenderá a manejar las operaciones que se pueden realizar en la sección de Citas como la creación, eliminación, mover cita, confirmar una cita o rechazar una cita.

Lección 12: Peluquerías y Estéticas

En esta lección el usu

ario aprenderá a manejar las operaciones que se pueden realizar en la sección de peluquerías como lo creación y eliminación.

Lección 13: Escáner QR

En esta lección el usuario aprenderá a manejar las operaciones que se pueden realizar en la sección de Escáner Qr donde puede conectar un dispositivo y escanear el código QR.

I.3.1.4.3 Unidad III: Reportes del Sistema

Lección 14: ingresos

En esta lección el usuario aprenderá a manejar las operaciones que se pueden realizar en la sección de reportes de ingreso tanto como la generación grafica para vista de impresión.

Lección 15: aplicación móvil

En esta sección el cliente aprenderá a manejar operaciones que se pueden realizar es esta sección con operaciones como la de los dispositivos que tienen instalada la aplicación móvil.

Lección 16: Imprimir Datos

En esta lección el usuario aprenderá a imprimir los datos de las estadísticas generadas por el sistema.

I.3.1.5 Plan de Clases

Nro.	CONTENIDO	OBJETIVO	Fecha	DURACIÓN (horas)	MATERIAL DIDÁCTICO	MEDIOS DE ENSEÑANZA - APRENDIZAJE	DESTINATARIO
1	Lección 1: Ingresar al sistema	Que el usuario se identifique y pueda entrar al sistema de acuerdo a su rol.		1		computadora	Administrador de la clínica, Medico, Peluquero
2	Lección 2: Familiarización con la navegación	Que el usuario se navegue con facilidad por las diferentes secciones del sistema.		1		computadora	Administrador de la clínica, Medico, Peluquero
3	Lección 3: Administrar el perfil	Que el usuario se pueda administrar sus datos personales, clave de acceso, foto y el tema		1		computadora	Administrador de la clínica, Medico, Peluquero
4	Lección 4: Utilizar el Sistema de Ayuda	Que el usuario pueda hacer un buen uso del sistema		1		computadora	Administrador de la clínica, Medico, Peluquero

5	Lección 5: Roles y Procesos	Que el usuario pueda gestionar estos datos		1		computadora	Administrador de la clínica
6	Lección 6: Usuarios	Que el usuario pueda gestionar estos datos		1		computadora	Administrador de la clínica, Medico, Peluquero
7	Lección 7: Medicamentos	Que el usuario pueda gestionar estos datos		1		computadora	Administrador de la clínica
8	Lección 8: Noticias	Que el usuario pueda gestionar estos datos		1		computadora	Administrador de la clínica, Medico, Peluquero
9	Lección 9: Clientes	Que el usuario pueda gestionar estos datos		1		computadora	Administrador de la clínica, Medico, Peluquero
10	Lección 10: Mascotas	Que el usuario pueda gestionar estos datos		1		computadora	Administrador de la clínica, Medico, Peluquero

11	Lección 11: Citas	Que el usuario pueda gestionar estos datos		2		computadora	Administrador de la clínica, Medico, Peluquero
12	Lección 12: Peluquerías y Estéticas	Que el usuario pueda gestionar estos datos		1		computadora	Administrador de la clínica, Medico, Peluquero
13	Escáner QR	El usuario aprende a escanear desde un dispositivo lector o la aplicación móvil		1		computadora	Administrador de la clínica, Medico, Peluquero
14	Lección 14: Ingresos	El usuario puede generar un informe cuando lo desee		1		computadora	Administrador de la clínica, Medico, Peluquero
15	Lección 15: Aplicación Móvil	El usuario puede generar un informe cuando lo desee		1		computadora	Administrador de la clínica, Medico, Peluquero
16	Lección 16: Imprimir Datos	El usuario puede generar un informe cuando lo desee		1		computadora	Administrador de la clínica, Medico, Peluquero

I.3.1.6 Cronograma

ID	Actividad	Duracion	2016/1/18					2016/1/24					2016/1/31					
			18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2
1	☐Primera Capacitacion	5.0 d.																
2	Leccion 1	1.0 d.																
3	Leccion 2	1.0 d.																
4	Leccion 3	1.0 d.																
5	Leccion 4	1.0 d.																
6	Leccion 5	1.0 d.																
7	Leccion 6	1.0 d.																
8	Leccion 7	1.0 d.																
9	Leccion 8	1.0 d.																
10	☐Segunda Capacitacion	5.0 d.																
11	Leccion 9	1.0 d.																
12	Leccion 10	1.0 d.																
13	Leccion 11	1.0 d.																
14	Leccion 12	1.0 d.																
15	Leccion 13	1.0 d.																
16	Leccion 14	1.0 d.																
17	Leccion 15	1.0 d.																
18	Leccion 16	1.0 d.																

Ilustración 0.165 Cronograma de plan de capacitación

I.3.1.7 Resultados esperados

Al finalizar el proyecto, se han implementado al menos dos programas de capacitación en el uso del sistema de información.

I.3.1.8 Medios de Verificación del Componente

Informe del gerente de la clínica respecto a la capacitación realizada, adjuntando lista de participantes.

I.3.1.9 Conclusiones

Una vez realizada la capacitación los usuarios deben de estar listos para poder usar el sistema sin ningún problema.

El diseño de la nueva tarjeta de presentación promocionando la nueva aplicación móvil se ha diseñado con el fin de llamar la atención de los lectores haciendo que no pasen desapercibidas.

I.3.2 ANEXOS

I.3.2.1 Lista del Personal Capacitado

LISTA DEL PERSONAL CAPACITADO DE LA CLÍNICA VETERINARIA "AMEVET"

N°	NOMBRE	CI	CARGO	FIRMA
1	Ximena Jurado	5055432 Tja	Metenarrio Responsable	<i>Ximena Jurado</i>
2	Nataly Jurado León	71772572 Tja	Auxiliar de Rehabilitación	<i>Nataly Jurado León</i>
3	Nehyda L. Ramos G.	72162411 Tja	Técnico Veterinario	<i>Nehyda L. Ramos G.</i>
4				
5				



Ilustración 166. Lista del personal capacitado

I.3.2.2 Fotos del Personal en Capacitación

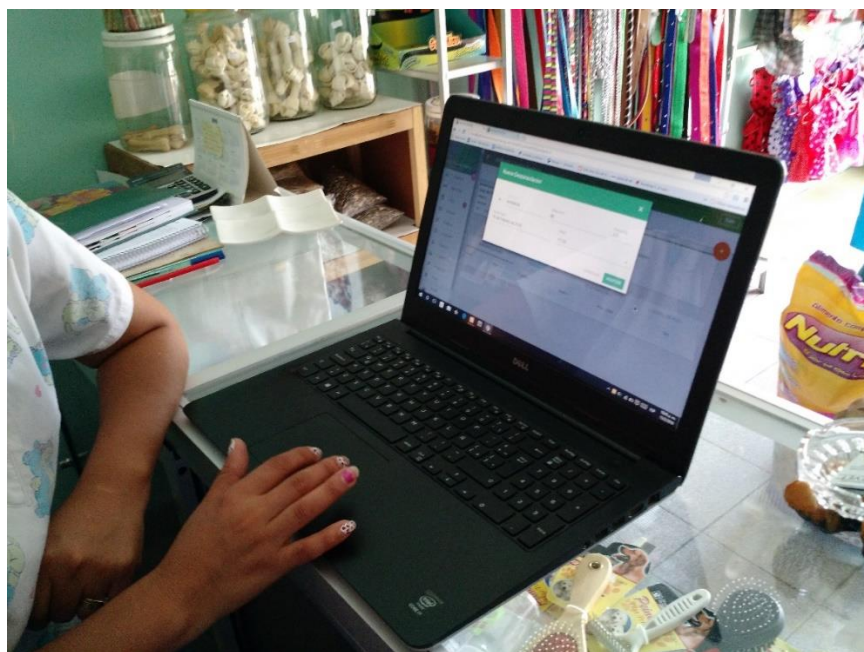


Ilustración 167. Foto1 Capacitación

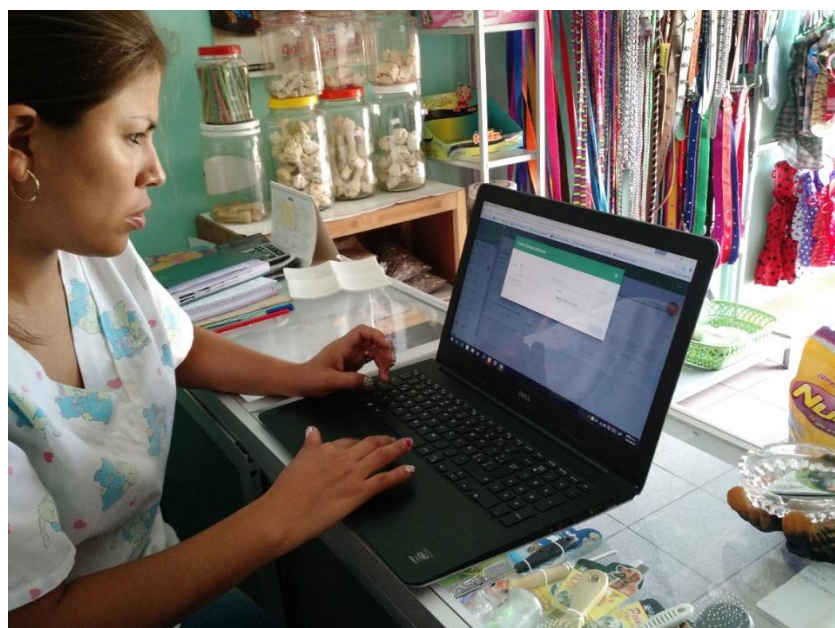


Ilustración 168. Foto 2 Capacitación

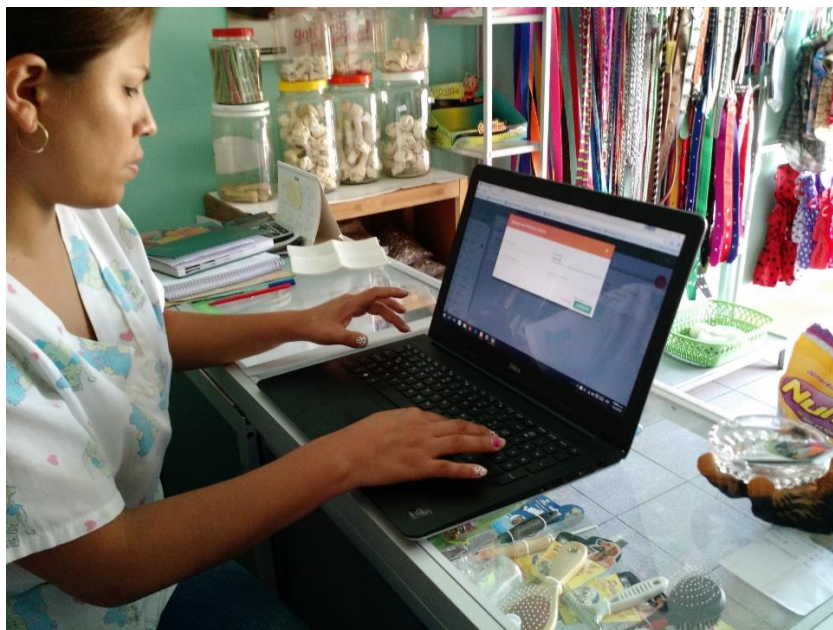


Ilustración 169. Foto 3 Capacitación

I.3.2.3 Modelo Certificado de capitación



Certificado de Capacitación

Se certifica que:

Ha cumplido satisfactoriamente con el curso de uso del
Sistema Web Neg-Vet

Firmado: _____ Fecha: _____

CAPACITADOR

The image shows a certificate template with a decorative green and yellow border. The title 'Certificado de Capacitación' is at the top. Below it, 'Se certifica que:' is followed by a blank line for a name. The main text states 'Ha cumplido satisfactoriamente con el curso de uso del Sistema Web Neg-Vet'. At the bottom, there are fields for 'Firmado:' and 'Fecha:' with blank lines, and the word 'CAPACITADOR' centered below the signature line.

Ilustración 170. Certificado de Capacitacion

I.3.2.4 Modelo de la tarjeta de presentación



Ilustración 171. Tarjeta de Presentacion Lado frontal



Ilustración 172. Tarjeta de presentación

CAPITULO III CONCLUSIONES Y RECOMENDACIONES

I.4 CONCLUSIONES

Ya terminado este documento se ha concluido la realización tanto del software, como la documentación asociada mediante las etapas de inicio, elaboración, construcción y transición del modelo de desarrollo Proceso Unificado.

Mediante el estudio de factibilidad se ha determinado, cuán factible es el proyecto si lo llevamos a cabo, donde se evaluó en el ámbito técnico, operacional, económico, y legal. Se estudió sobre posibles riesgos que puedan afectar el desarrollo del proyecto, y se completó un plan de mitigación y contingencia para los más riesgosos.

De esta misma forma, han sido estudiados los distintos usuarios que compondrán el sistema, indicando las características y las tareas que cada uno de ellos realiza.

Se ha documentado minuciosamente todos los procesos para la elaboración de este producto software que ha sido enfocado como un sistema web para la ayuda a los actuales sistemas que posee la clínica veterinaria.

La construcción de la aplicación móvil utilizando estas nuevas tecnologías ha sido gratificante porque me ha facilitado la reutilización de código puesto que trabaje con las mismas tecnologías ahorrándome así un poco más de tiempo en el desarrollo.

No he podido subir la aplicación a la tienda oficial de aplicaciones Android puesto que es necesario que cuente con una tarjeta de crédito que aún no dispongo por el momento.

La capacitación se llevo a cabo correctamente por que los usuarios finales ya tenían conocimientos en este tipo de sistemas.

Para finalizar, los requerimientos, el análisis, el diseño, la implementación, y las pruebas, no han terminado para este sistema, es más, aún seguirán teniendo estas instancias, siempre cuando se continúe sacando actualizaciones. Como trabajo futuro este software espera implementar el sistema de fotografías de los pacientes, así como también el poder administrar los perfiles detalladamente, indicando a que módulos se les permite acceder, entre otras actualizaciones que vayan saliendo en el camino.

I.5 RECOMENDACIONES

Se recomienda el uso de la metodología Mobile-D especialmente para el desarrollo de aplicaciones móviles, debido a que esta especializa en este tipo de desarrollo.

Recomiendo el desarrollo de una aplicación híbrida para aplicaciones donde no sea excesivo el consumo de recursos de hardware, porque el desarrollo de una aplicación híbrida se lo realiza con tecnología web y no se hace necesario aprender java para programar en Android.

Para la recolección de información y requisitos de una o varias dependencias, utilizar diferentes técnicas de investigación, que capturen lo que realmente necesita el sistema.

Evitar el descuido del seguimiento en las fases del modelo RUP, con la finalidad de cumplir con los plazos y tiempos establecidos.

Mantener seguida reuniones y entrevistas con los stakeholders y directivos del proyecto.

Realizar constantes pruebas y con diferentes casos, una vez que se encuentre desarrollado el aplicativo para así estar seguros de su correcto funcionamiento.

El manual de funciones es un documento muy necesario en la veterinaria, por lo que se recomienda elaborarlo en su momento para conocer con claridad el área de cada uno de los usuarios.