



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN

**Desarrollo de un sistema informático para el control de
historiales clínicos para veterinarias.**

TRABAJO PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADA EN INFORMÁTICA

PRESENTA:

FANNY ITZEL VALDIVIESO SANDOVAL

ASESOR:

L.I. MAURICIO JAQUES SOTO



CUAUTITLÁN IZCALLI, ESTADO DE MÉXICO, 2022



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
SECRETARÍA GENERAL
DEPARTAMENTO DE TITULACIÓN

ASUNTO: VOTO APROBATORIO

DR. DAVID QUINTANAR GUERRERO
DIRECTOR DE LA FES CUAUTITLÁN
PRESENTE

ATN: DRA. MARÍA DEL CARMEN VALDERRAMA BRAVO
Jefa del Departamento de Titulación
de la FES Cuautitlán.

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: **Trabajo Profesional**.

Desarrollo de un sistema informático para el control de historiales clínicos para veterinarias.

Que presenta la pasante: **Fanny Itzel Valdivieso Sandoval**
Con número de cuenta: **416117305** para obtener el título de: **Licenciada en Informática**

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el **EXAMEN PROFESIONAL** correspondiente, otorgamos nuestro **VOTO APROBATORIO**.

ATENTAMENTE
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Cuautitlán Izcalli, Méx. a 05 de Agosto de 2022.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	Mtro. Ing. Elect. Gerardo Vigil Sanabria	
VOCAL	Dra. Claudia Cruz Sánchez	
SECRETARIO	L.I. Mauricio Jaques Soto	
1er. SUPLENTE	Mtro. Domingo Márquez Ortega	
2do. SUPLENTE	L.I. Carla Pamela Pérez Solano	

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional.

MCVB/ntm*

DEDICATORIAS:

Primeramente, quiero agradecerle a Dios por haberme permitido llegar hasta donde estoy ahora, haber culminado una etapa para iniciar otra que es difícil pero no imposible de superar día tras día sin la ayuda de él.

A mi mamá que siempre estuvo apoyándome en todos los ámbitos de principio a fin y que sin la ayuda de ella no hubiera logrado terminar la licenciatura. Sé que en donde esté siempre estará cuidándome, apoyándome y que siempre estará orgullosa de mí como lo estuvo en vida y es por ella a quien le dedico principalmente este informe porque ella era quien anhelaba verme titulada, era quien estaba tan emocionada por ver este informe, pero por las circunstancias de esta pandemia me la arrebató de la noche a la mañana, a mi ser querido máspreciado y que para alguien puede ser lo máspreciado en esta vida, porque sin ella la vida no es la misma: mi madre Sandra. Me hace mucha falta, pero sé que ella intercede por mí para darme muchas fuerzas y es por eso por lo que hoy estoy escribiendo este informe. Donde quiera que estés es: ¡por ti y para ti mamá! Sin ti la vida ya no es igual, pero hago el mejor esfuerzo.

A mi padre que también me apoyó en varios momentos escolares que, sea como sea es un buen padre responsable, inteligente, entre muchas otras cosas más y es por eso por lo que lo admiro.

A mis hermanos que, aunque no lo demuestre, pero los quiero mucho también y que daría lo que fuera por siempre tenerlos a mi lado y que siempre tendrán mi apoyo.

A la universidad en general por haberme brindado buenos profesores, vivencias y experiencias únicas e irrepetibles y por cada una de las personas que me fui encontrando a lo largo de este camino que gracias a ellos soy la persona que soy hoy en día, así como las amistades que hice y que pude contar con su apoyo en el peor momento que una persona puede pasar.

De todo corazón... ¡Gracias a cada uno de ustedes!

ATENTAMENTE: Fanny Itzel Valdivieso Sandoval.

Índice

OBJETIVOS.....	6
INTRODUCCIÓN.....	6
CAPÍTULO 1. EL DESARROLLADOR WEB Y SUS HERRAMIENTAS PARA EL DESARROLLO.....	8
1.1 Diagramas UML.....	9
1.1.1 Diagrama de casos de uso.....	10
1.1.2 Diagrama de clases.....	13
1.1.3 Diagrama de objetos.....	14
1.1.4 Diagrama de paquetes.....	15
1.1.5 Diagrama de Estados.....	16
1.1.6 Diagrama de actividades.....	18
1.1.7 Diagrama de secuencia.....	20
1.1.8 Diagrama de comunicación.....	22
1.1.9 Diagrama de componentes.....	23
1.1.10 Diagrama de despliegue.....	24
1.2 Diagramas para base de datos.....	25
1.2.1 ¿Qué es una base de datos?.....	25
1.2.2 Diagrama Entidad – Relación.....	26
1.2.3 ¿Qué es una relación en las bases de datos?.....	28
1.3 Conceptos.....	30
1.3.1 Sistema de información.....	30
1.3.2 HTML.....	31
1.3.3 Bootstrap.....	31
1.3.4 PHP.....	32
1.3.5 JavaScript.....	33
1.3.6 MySQL.....	33
1.3.7 Navegador web.....	35
1.3.8 Protocolos HTTP y HTTPS.....	35
1.3.9 Cliente / Servidor.....	36
1.4 SOFTWARE QUE REQUIERE EL DESARROLLADOR.....	39
1.4.1 Visual Studio Code.....	39
1.4.2 Apache.....	40

1.4.3 Xampp.....	41
1.4.4 HeidiSQL.....	42
CAPÍTULO 2. Empresa Lupware.....	45
2.1 Acerca de la empresa.....	45
2.2 Organigrama.....	47
CAPÍTULO 3. Experiencia en Lupware.....	48
3.1 Demostración del proceso de desarrollo del sistema	49
3.1.1 Realización de diagramas UML.	50
3.1.2 Realización de diagrama entidad – relación.	65
3.1.3 Diseño de base de datos en MySQL.....	66
3.2 Realización del sistema (codificación).....	72
3.3 Realización del sistema web (interfaces).....	125
CONCLUSIONES.....	164
BIBLIOGRAFÍA.....	166

OBJETIVOS.

El presente informe tiene como finalidad de llegar al lector describiéndole la experiencia laboral que he adquirido después de haber egresado de la licenciatura en informática en donde he cumplido mis funciones como desarrolladora de software.

De manera general se explicará la relación que tengo dentro de la empresa Lupware desde poder ingresar al campo laboral a partir de que concluí mis estudios de la licenciatura. En él, se mencionará la experiencia que he adquirido, así como una breve descripción del desempeño dentro de la empresa para cumplir los objetivos que se han puesto a prueba.

INTRODUCCIÓN.

El alumno en licenciatura en informática de la Facultad de Estudios Superiores Cuautitlán en cuanto concluye satisfactoriamente el plan de estudios, este se dispone a buscar ofertas laborales para poder integrarse a la actividad profesional en la cual empieza a buscar varias ramas de la carrera para ir desarrollando sus capacidades y habilidades que adquirió durante su estancia.

Para esto, con toda la información que recabó durante sus años de estudios está preparado para resolver las problemáticas que se van presentando en el ámbito laboral, específicamente si el egresado decide por el área de sistemas de información por lo cual, debe de contar con los conocimientos teóricos y prácticos para laborar en la empresa si es que lo que está buscando es para desarrollador de software. Los desarrolladores son los que buscan una solución en base a un problema, es decir, actualizar sistemas obsoletos por la tecnología más nueva o la más reciente o, en su caso contrario, crear nuevos sistemas desde cero o retomar algo ya existente, pero modificándolo o adaptándolo a las necesidades del cliente.

La finalidad de este informe es dar a conocer de manera más amplia y detallada en que consiste en que el egresado en informática tenga ciertas aptitudes por si desea involucrarse en el mundo del desarrollo y que probablemente haya algunos cambios a futuro porque la tecnología evoluciona rápidamente cada día y por ende también nosotros tenemos que seguir estudiando y aprendiendo.

También se realiza una descripción de como he ido involucrándome dentro de la empresa después de haber egresado de la licenciatura ya que la empresa no tiene mucho de su creación y aquí, tanto el creador como los que estamos laborando son igualmente egresados de la Facultad.

Para poder entender o tener noción básica del tema a tratar se desarrollará una introducción de los conceptos básicos pero necesarios que nos llevarán a comprender este informe de una manera clara y concisa.

Con los temas seleccionados, estos nos engloban a los fundamentos del desarrollo de software, los diagramas que se necesitan para entender claramente los requerimientos del cliente y la importancia del desarrollo de las actividades dentro de la empresa para lograr los objetivos de dichos sistemas y páginas web, de tal manera para poder realizar este informe.

La empresa para la cual se desarrolla este informe es para Lupware, que brinda servicios de consultoría en computación que brinda soluciones de software a la medida y que se ajusta al presupuesto con la intención de entregar los proyectos adecuados y cubriendo las necesidades de cada uno de sus clientes en tiempo y forma.

En esta empresa se ha tenido la función de consultor PHP desde octubre del 2020 hasta hoy en día en que estoy elaborando este informe por lo que, cuento con la información justa y necesaria para abordar los temas que se explican en este trabajo.

Como primer capítulo de este informe se desarrollará la teoría que sustenta la parte práctica, es decir, lo que se hace en el trabajo abordando el tema del uso de las herramientas que se necesitan para desarrollar los sistemas.

El segundo capítulo hablaremos acerca de la empresa y que requerimientos solicita para poder integrarlos al equipo de trabajo.

Y finalmente, se describe la experiencia que se ha obtenido en este tiempo desde el ingreso, así como las funciones y cómo han impactado los proyectos en los que se hizo participación en todo este tiempo, pero algo más importante que recalcar es en cuanto el crecimiento profesional.

CAPÍTULO 1. EL DESARROLLADOR WEB Y SUS HERRAMIENTAS PARA EL DESARROLLO.

Hoy en día la tecnología ha avanzado de forma rápida y se ha vuelto una necesidad para el ser humano, con ella hemos logrado comunicarnos de un lugar a otro, tener la información almacenada en internet sin tener la necesidad de utilizar métodos antiguos como el envío de postales, almacenar la información en archiveros o peor aún, traspapelar o perder los documentos físicos, pero la duda es: ¿Qué tan seguro es guardar mi información ante los ataques de los hackers? ¿Cómo le hacen para salvaguardar mi información en caso de que se elimine accidentalmente? ¿Cómo protegen mi información de virus, malware, entre otros?

Bueno, aunque actualmente muchos aún le temen a estos cambios (especialmente las personas mayores) a actualizarse y utilizar la tecnología como el internet; ya que actualmente se empieza a trabajar en la seguridad informática (aunque son pocos los que se dedican a esta rama) creando tipos de protección como copias de seguridad de almacenamiento que se configuran para que automáticamente guarde los cambios nuevos para respaldar toda la información, software que ayuda a limpiar o detectar virus maliciosos antes o después de causar algún daño así como informar los usuarios de donde pueden surgir los ataques y como prevenir severos problemas tanto a las computadoras como a la empresa exponiendo lo más valioso: su información confidencial.

Entonces, la importancia o las ventajas de usar sistemas informáticos es para ahorrar papel, tiempo, espacio y con ello consultar la información en cualquier momento con una conexión a internet y en los diferentes dispositivos electrónicos (celular, Tablet, computadora, etcétera).

Para el desarrollo de este tipo de sistemas el desarrollador de software necesita tener los conocimientos previos, especialmente los prácticos aunque también la teoría es importante para el desarrollo de estos, principalmente para la elaboración de los diagramas correspondientes para plasmar los requerimientos de los clientes y tener una idea más clara de lo que el usuario necesita y que al final no le sea un problema, porque en caso de que el desarrollador carezca de estos puede influir en que los proyectos salgan con errores de lógica o que por estudiar ciertos temas se presenten atrasos con

las entregas.

Para poder realizar un proyecto de software el desarrollador no debe de carecer en conocimientos y debe de dominar o conocer, aunque sea lo básico de los diferentes lenguajes de programación que existen. Aunque también el desarrollador debe de tomar en cuenta con que lenguaje va a trabajar la mayor parte de su vida ya que se clasifican para los desarrolladores de acuerdo con su experiencia profesional como son: junior, semi senior y senior.

El desarrollador junior significa que es un pasante o practicante, sino que están en un punto donde ya se sabe programar, aunque no con las mejores prácticas y es por ende que aún se supervisa la realización del proyecto. Se conoce por lo menos un lenguaje de programación y se dan tareas específicas pues aún, se requiere de conocimientos y habilidades para dimensionar los proyectos y el desarrollo.

El desarrollador semi senior observa el panorama que va cambiando porque ya tiene los conocimientos técnicos más amplios (desarrollo, pruebas, documentación, implementación, análisis) y se tiene ya una buena práctica de desarrollo por lo que puede manejar distintos lenguajes y no se necesita de tanta supervisión.

El desarrollador senior no solamente le basta las capacidades que se obtuvieron en junior y semi senior sino también de habilidades enfocadas a la gestión donde se debe ser capaz de manejar equipos de trabajo, saber integrar los objetivos de desarrollo con los de la empresa y cómo se debe ser apto para asesorar a desarrolladores junior y semi junior, así como revisiones periódicas del código para mejorar su calidad y estructura.

Hoy en día, el desarrollador de software ha tomado un papel importante en la mayor parte de las ramas desde la física, química, medicina, astronomía, etcétera y es por eso por lo que se ha vuelto una profesión muy demandada.

A continuación, se desglosará el capítulo explicando los conceptos que se utilizaron para el desarrollo del sistema, así como su definición y que por defecto debe de tener un desarrollador de software.

1.1 Diagramas UML.

El UML (Unified Modeling Language) está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje

unificado de modelado¹, cuenta con reglas para combinar tales elementos.

La finalidad de los diagramas es preservar diversas perspectivas de un sistema a las cuales se les conoce como modelo.

El modelo UML de un sistema es similar a un modelo a escala de un edificio junto con la interpretación del artista del edificio. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice como implementar dicho sistema.

1.1.1 Diagrama de casos de uso.

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Para los desarrolladores del sistema, esta es una herramienta valiosa ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde el punto de vista del usuario. Esto es importante si la finalidad es crear un sistema que pueda ser utilizado por la gente en general. (Kimmel, 2008)

El símbolo del caso de uso se utiliza para representar capacidades. Al caso de uso se le da un nombre y una descripción mediante un texto. Este último debe describir como inicia y finaliza el caso de uso e incluye una descripción de la capacidad descrita por el nombre de esta, así como escenarios de apoyo y requisitos no funcionales.

El uso de símbolos de los casos de uso.

Los símbolos básicos son: actor, conector, óvalo del caso de uso y el rectángulo. La figura del actor representa a los participantes de los casos de uso (pueden ser personas o cosas).

La figura del caso de uso se utiliza para representar capacidades. Al caso de uso se le da un nombre y una descripción mediante un texto, este debe describir como inicia y finaliza el caso de uso e incluye una descripción de la capacidad descrita por el nombre de esta, así como escenarios de apoyo y requisitos no funcionales.

Los casos de uso pueden estar asociados con los actores en donde se utilizan los conectores para indicar la manera en que ambos están asociados. Además de que los estilos de los conectores pueden cambiar para transmitir más información acerca de la relación entre los actores y los casos de uso. El rectángulo nos ejemplifica el sistema, es

¹ En lugar de indicarle a usted cuáles son los elementos y las reglas, veamos directamente los diagramas ya que los utilizará para hacer el análisis del sistema.

decir, estamos modelando el sistema como se observa en la figura 1. Símbolos básicos para el diagrama de caso de uso.

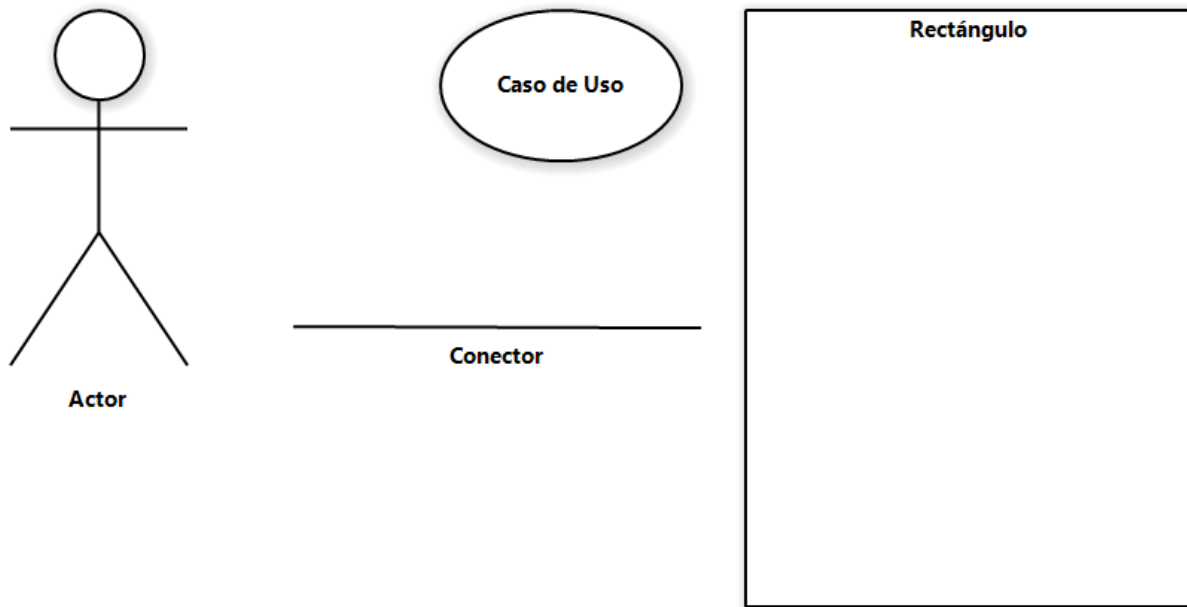


Figura 1. Símbolos básicos para el diagrama de caso de uso.

Los estilos de líneas para los conectores son:

1. *Asociación*: es una línea simple y se usa para mostrar cuales actores están relacionados con cuales casos de uso.
2. *Dependencia*: es una línea punteada con la flecha direccional. La flecha apunta hacia el caso de uso que depende, como se observa en la figura 2 el caso de uso validar usuario y contraseña depende de que el dentista inicie sesión.
3. *Generalización*: es una línea dirigida con un triángulo hueco, la palabra “generalización” que en UML significa “herencia”. Cuando mostramos una relación de generalización entre dos actores o dos casos de uso, estamos indicando que el actor o el caso de uso “hijos” son un caso del actor y uso básico, como se observa en la figura 3 diagrama de casos de uso en el que se muestra una relación de generalización entre tres casos de uso.



Figura 2. El caso de uso validar usuario y contraseña depende de que el dentista inicie sesión.

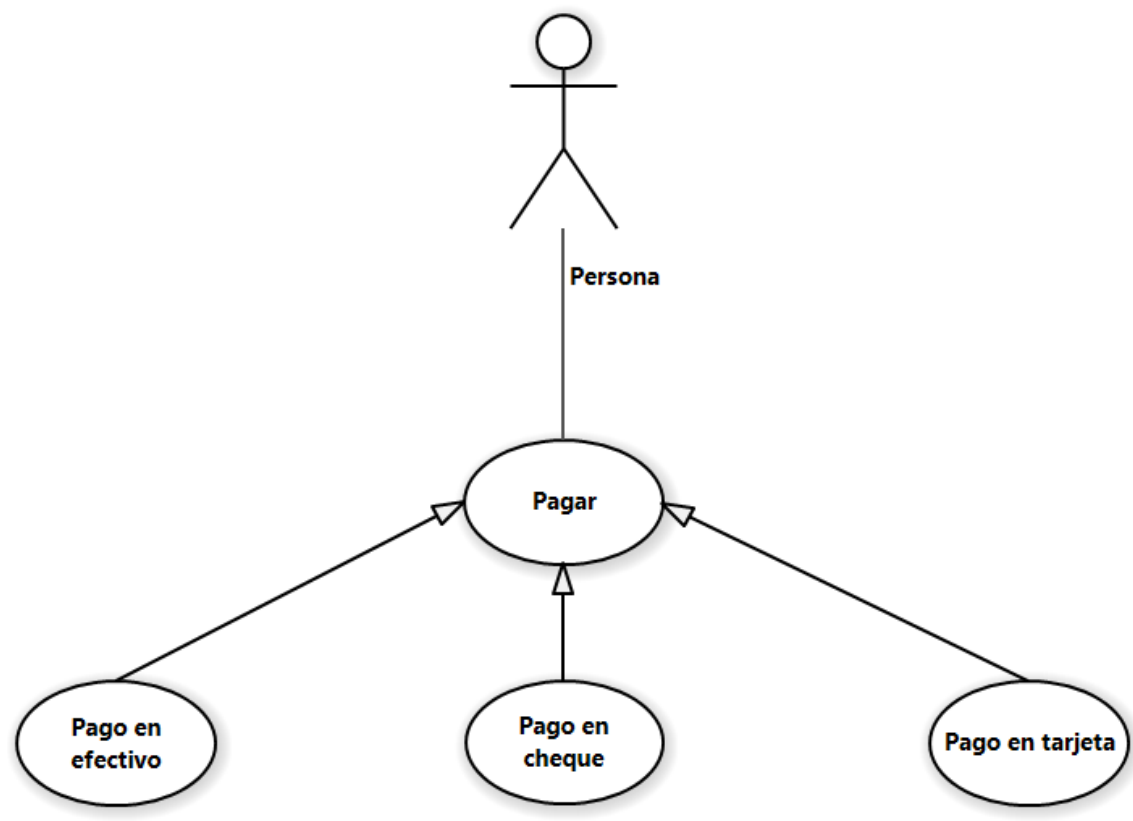


Figura 3. Diagrama de casos de uso en el que se muestra una relación de generalización entre tres casos de uso.

Test WAVE

Este test nos sirve para saber si estamos haciendo correctamente los casos de uso en el diagrama.

Para esto recorreremos cada una de estas preguntas:

W El caso de uso describe QUÉ (what) HACER y NO como hacerlo.

- A El caso de uso esta descrito desde la perspectiva del actor (sí o no).
- V El caso de uso incluye valor para el actor.
- E Es el flujo de eventos un escenario completo.

1.1.2 Diagrama de clases.

Los diagramas de clases se usan para mostrar las clases de un sistema y las relaciones entre ellas. Una sola clase puede mostrarse en más de un diagrama de clases y no es necesario mostrar todas las clases en un solo diagrama monolítico de clases. Es muy popular con los ingenieros de software para documentar la arquitectura de un sistema. (Lucidchart, s.f.)

Es un diagrama estático porque nos describen acción, es decir, la interacción entre clases y objetos. Los dos elementos más importantes para este diagrama son: el rectángulo y una línea. Los rectángulos son clases y las líneas son conectores que muestran la relación entre esas clases. Las clases incluirán comportamientos y atributos llamados también características. Los atributos pueden ser campos, propiedades o ambos. Los comportamientos son los métodos que realiza la clase.

Para esto usamos un clasificador (rectángulo) y en el colocamos el nombre, los atributos y las operaciones, como se aprecia en la figura 4.

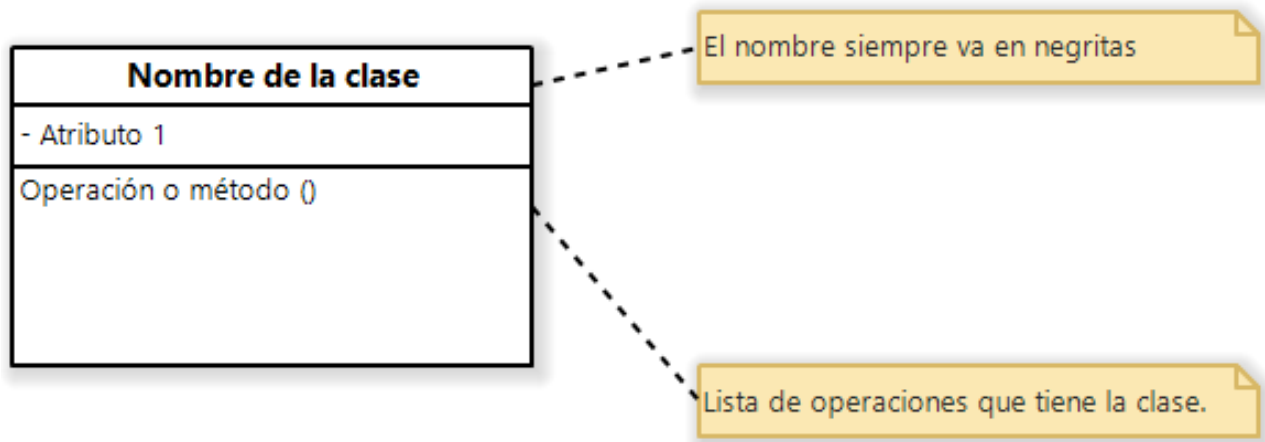


Figura 4. Ejemplo del clasificador del diagrama de clases.

Los beneficios que se obtienen al realizar los diagramas de clases son:

- Ilustrar modelos de datos para sistemas de información sin importar que tan simples o complejos sean.

- Expresar visualmente las necesidades específicas de un sistema.
- Comprender la visión general de los esquemas de una aplicación.

Todas las clases poseen diferentes niveles de acceso en función de su visibilidad de acceso. A continuación, se muestran los niveles de acceso y sus símbolos correspondientes.

- Público (+)
- Privado (-)
- Protegido (#)
- Paquete (~)
- Derivado (/)
- Estático (subrayado)

1.1.3 Diagrama de objetos.

Un objeto es una instancia de clase “una entidad que tiene valores específicos de los atributos y acciones” (Schmuller, 1999, p. 9) cuando este se representa gráficamente se observan muchas similitudes con el diagrama de clases.

Los diagramas de objetos son sencillos de crear, se componen de objetos representados por rectángulos conectados entre líneas como se observa en la figura 5 ejemplificando los componentes de los diagramas de clases.

Objeto:

Es posible describir un objeto, aunque no es frecuente, así como también colocar objetos anónimos.

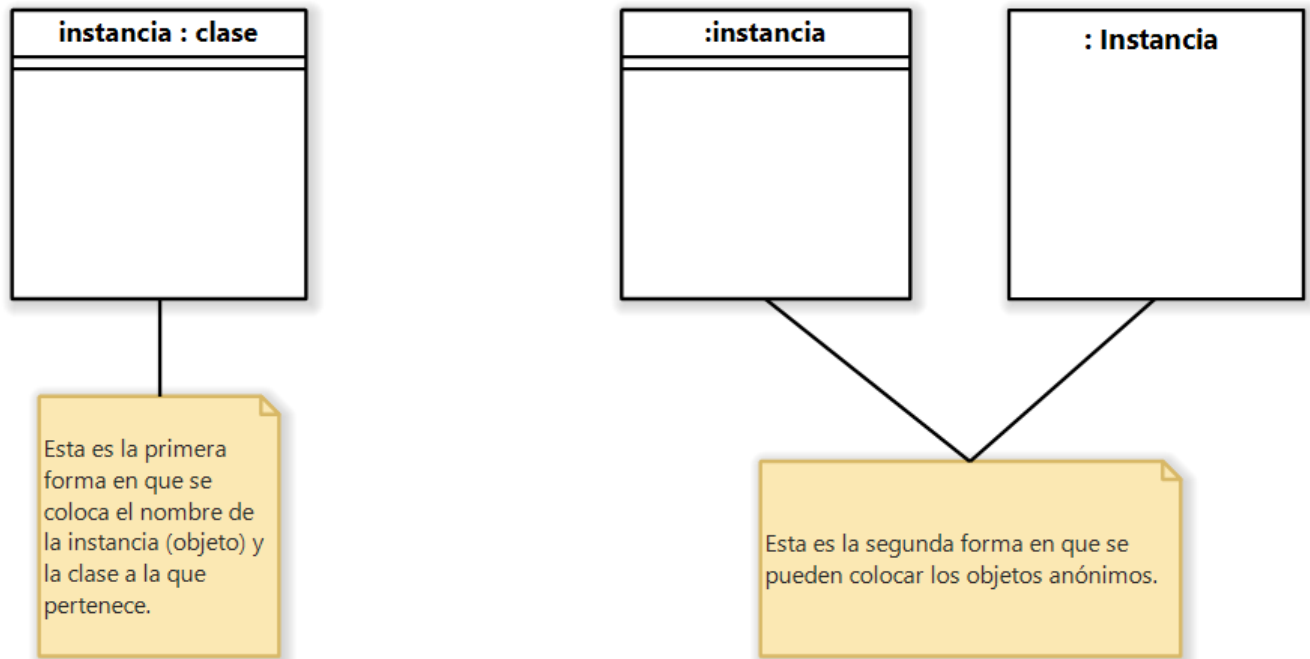


Figura 5. Ejemplo de los componentes de los diagramas de clases.

1.1.4 Diagrama de paquetes.

El diagrama de paquetes se utiliza para organizar y para ayudar a crear una estructura al agrupar los elementos relacionados entre sí o que sean similares.

Este diagrama nos permite tener una vista de alto nivel del sistema, los cuales se podrán observar los componentes principales y como es que se relacionan para un sistema complejo. También ayuda a observar fácilmente las dependencias y se pueden empaquetar gran variedad de elementos, así como tener paquetes adentro de paquetes. Las clases que se encuentran en la misma cadena de herencia deben de estar en el mismo paquete. Su símbolo se denota por una carpeta de archivos. Este símbolo se usa en forma genérica para representar un nivel más elevado de abstracción que el clasificador.

Un paquete se puede implementar como un espacio de nombre o un subsistema, con un estereotipo, también se puede usar para la organización en general y sencillamente para representar una carpeta de archivos como se observa en la figura 6.

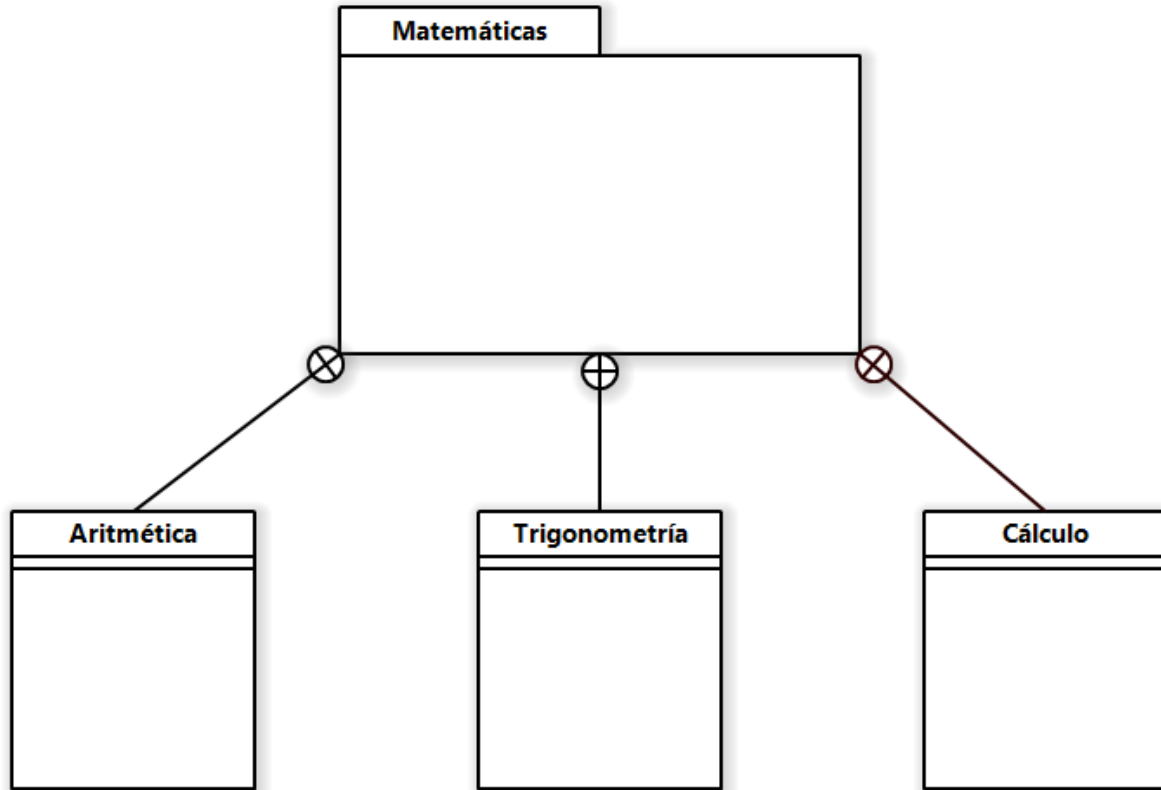


Figura 6. Ejemplo donde el diagrama muestra la clase “aritmética” que está siendo contenida en el paquete “matemáticas”.

1.1.5 Diagrama de Estados.

Los diagramas de estados (también conocidos como máquinas de estado) son buenos para mostrar el estado de un objeto sobre muchos casos de uso y para definir protocolos que definen una orquestación correcta de los mensajes. Los diagramas de estados se componen de manera significativa de símbolos llamados estados y líneas llamadas transiciones.

La sencillez de los símbolos es la parte más fácil del modelado son:

- La identificación de los problemas.
- La captación de las soluciones.
- La captura de esta comprensión.

Tres cosas importantes que hay que recordar para la realización de los diagramas son:

- Conocer todos los símbolos y la gramática no implica que deba usarlos todos.

- Es esencial modelar los aspectos importantes del sistema, así como modelar aquellos que no son obvios.
- No necesita toda suerte de diagrama para toda suerte de problema, o sea, ser selectivo.

Los símbolos del diagrama de estado son el símbolo, que se encuentra en la parte superior (circulo relleno) de la figura representa su estado inicial y el de la parte inferior (la diana) su estado final. El estado es representado por un rectángulo con vértices redondeados y una línea continua, así como una punta de flecha que apunta hacia el estado donde se hará la transición como se observa en la figura 7.

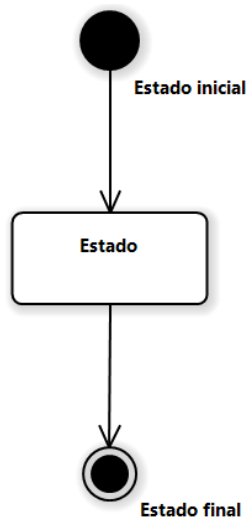


Figura 7. Ejemplo donde se muestran los elementos del diagrama de estados.

Los elementos de un estado son:

- Nombre
- Acciones de entrada / salida
- Transiciones internas
- Subestados
- Eventos diferidos

Las transiciones es el elemento más importante en el diagrama por lo que de un estado A a un estado B se produce cuando se origina el evento asociado y satisface la condición especificada.

Los elementos de una transición son:

- Estados origen y destino
- Evento de disparo
- Condición de guardar
- Acción

1.1.6 Diagrama de actividades

El diagrama de actividades es muy parecido a los diagramas de flujo ya que muestra los pasos, así como puntos de decisión y bifurcaciones como se aprecia en la figura 8.

Este ha sido diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o proceso. Este precisamente resalta las actividades.

Se representa:

- Un rectángulo con las esquinas redondeadas (más angosto y ovalado). El procesamiento dentro de una actividad se lleva a cabo y al realizarse se continúa con la actividad.
- Una flecha representa la transición de una a otra actividad.
- Cuenta con un punto inicial (círculo relleno) y uno al final (diana).

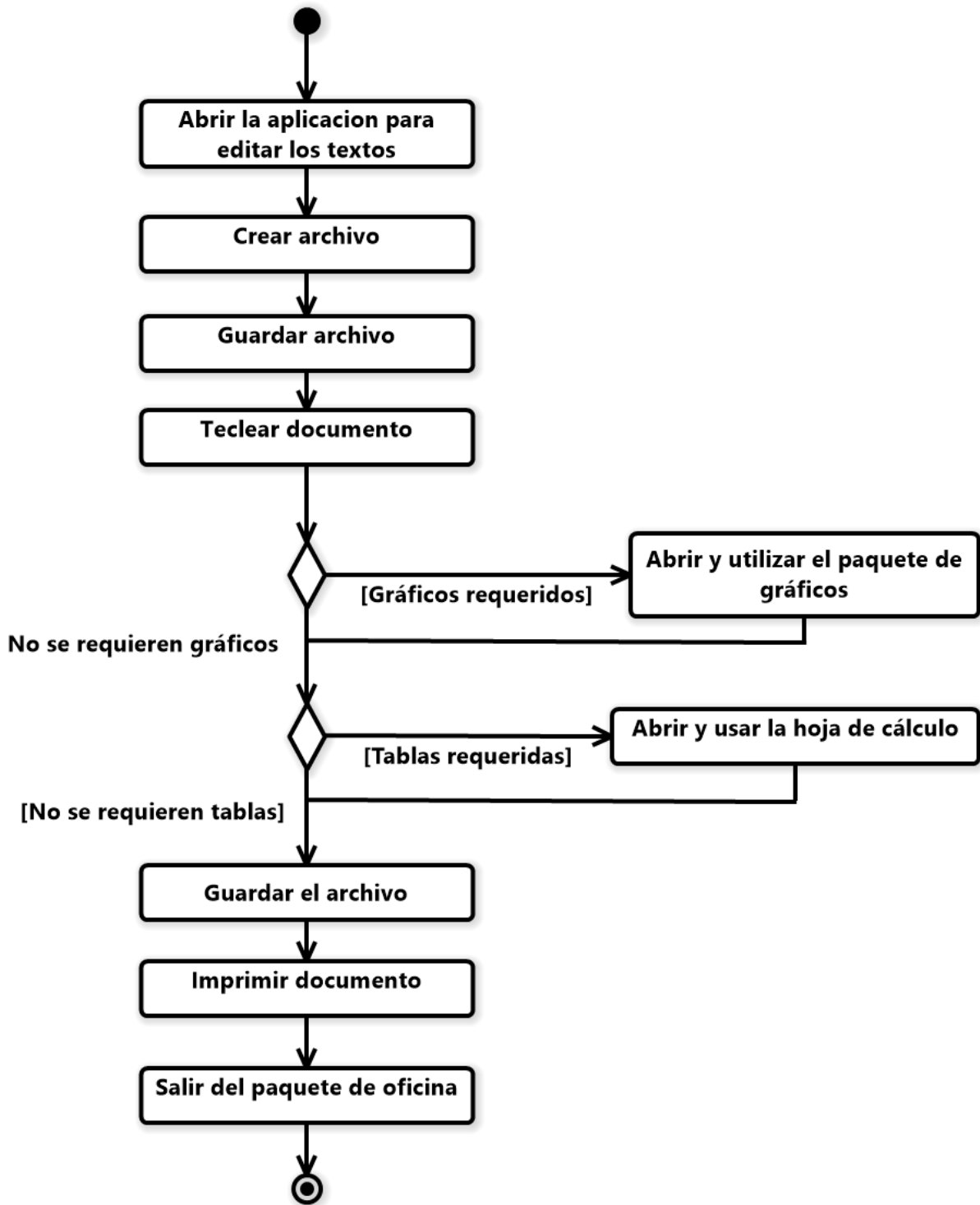


Figura 8. Ejemplo de diagrama de actividades para el proceso de creación de un documento.

1.1.7 Diagrama de secuencia.

El diagrama de secuencias consta de objetos que se representan de forma usual y establece y muestra la forma en que los objetos se comunican entre si al transcurrir el tiempo.

Los elementos del diagrama son:

- Rectángulos con nombre (subrayado).
- Líneas continuas con una punta de flecha que son para los mensajes.
- Una progresión vertical representa el tiempo.

Una línea de vida es un rectángulo con una recta vertical que desciende del rectángulo y este es el lugar conveniente para sujetar mensajes entrantes y salientes.

Objetos:

Se colocan cerca de la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La extensión que esta debajo (y en forma descendente) de cada objeto será una línea discontinua conocida como la línea de vida de un objeto. En la línea de vida de un objeto se encuentra un pequeño rectángulo conocido como activación.

Mensaje:

Un mensaje que va de un objeto a otro pasa de la línea de vida de un objeto a la de otro, es decir, que son líneas dirigidas que conectan líneas de vida.

Un objeto puede enviarse un mensaje a sí mismo, es decir, desde su línea de vida a su misma línea de vida.

Un mensaje puede ser simple, sincrónico o asincrónico como se observa en la figura 9, la cual muestra los elementos de las líneas de vida.

- Simple: es la transferencia del control de un objeto a otro.
- Sincrónico: esperará la respuesta a tal mensaje antes de continuar con su trabajo.
- Asincrónico: no esperará una respuesta antes de continuar.

Sus símbolos de cada uno son:

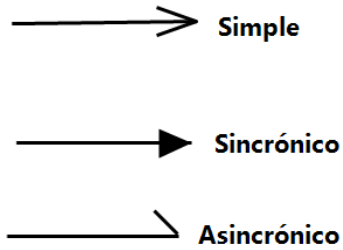


Figura 9. De los elementos de las líneas de vida.

El tiempo se representa de forma vertical en el diagrama, se inicia en la parte superior y avanza hacia la parte inferior.

Como se aprecia en la figura 10, el diagrama tiene dos dimensiones:

- La dimensión horizontal es la disposición de los objetos.
- La dimensión vertical muestra el paso del tiempo.

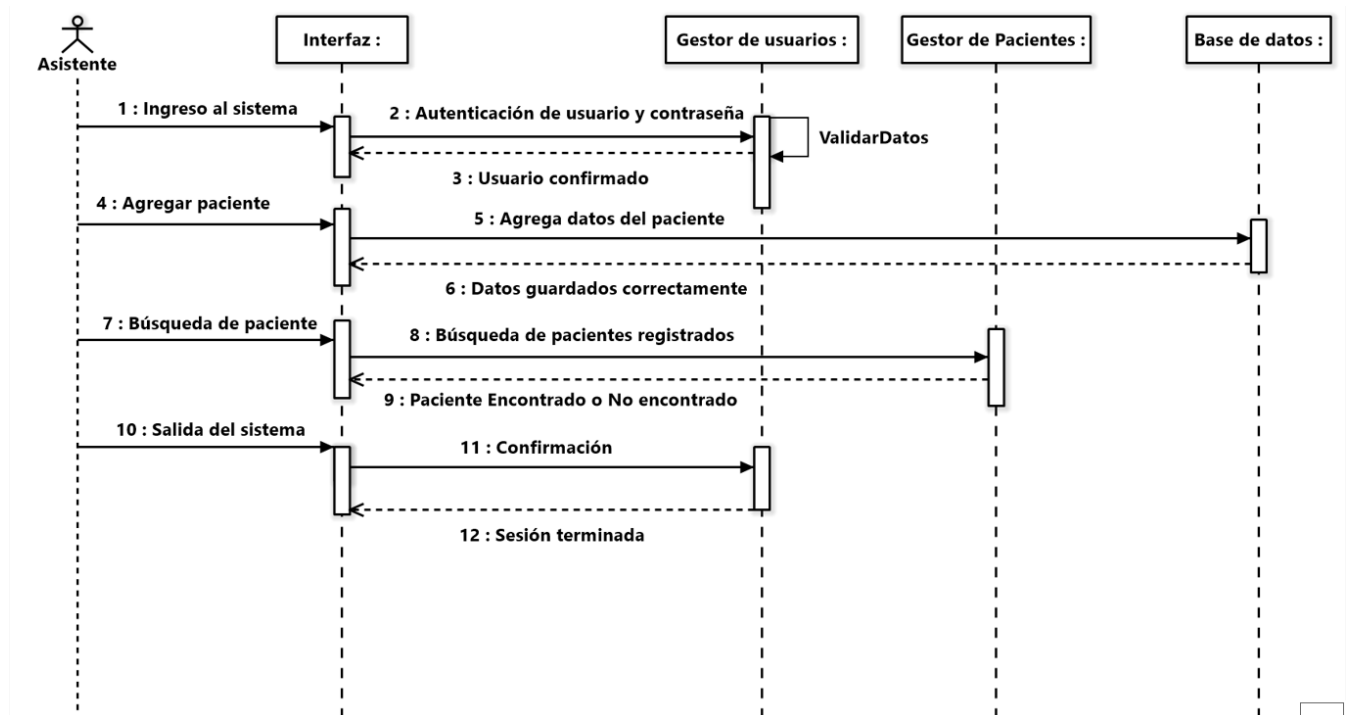


Figura 10. Ejemplo de un diagrama de secuencia donde los objetos se colocan a la izquierda a derecha.

1.1.8 Diagrama de comunicación.

Este diagrama modela el comportamiento dinámico de un sistema haciendo énfasis en la secuencia de los mensajes intercambiados por un conjunto de objetos para un caso de uso en particular como se observa en la figura 11 donde se realizan peticiones entre el usuario, el sistema y la base de datos.

Muestra el orden en el que suceden los mensajes, hay énfasis en las ligas entre los objetos.

Consta de 3 elementos:

- Objetos
- Ligas
- Mensajes

Objeto:

- Estos interactúan entre ellos.
- Se muestra de la forma usual por medio de un rectángulo.

Liga:

- También se conocen como ligas de comunicación.
- Se muestran por medio de una línea, es decir, si la línea conecta dos objetos eso significa que los objetos pueden comunicarse o interactuar entre ellos.

Mensajes:

- Se colocan a lo largo de las ligas.
- Muestra la comunicación entre los objetos.
- Se usa una flecha y los mensajes se nombran.
- Tienen un número que nos indica el orden de los mensajes.

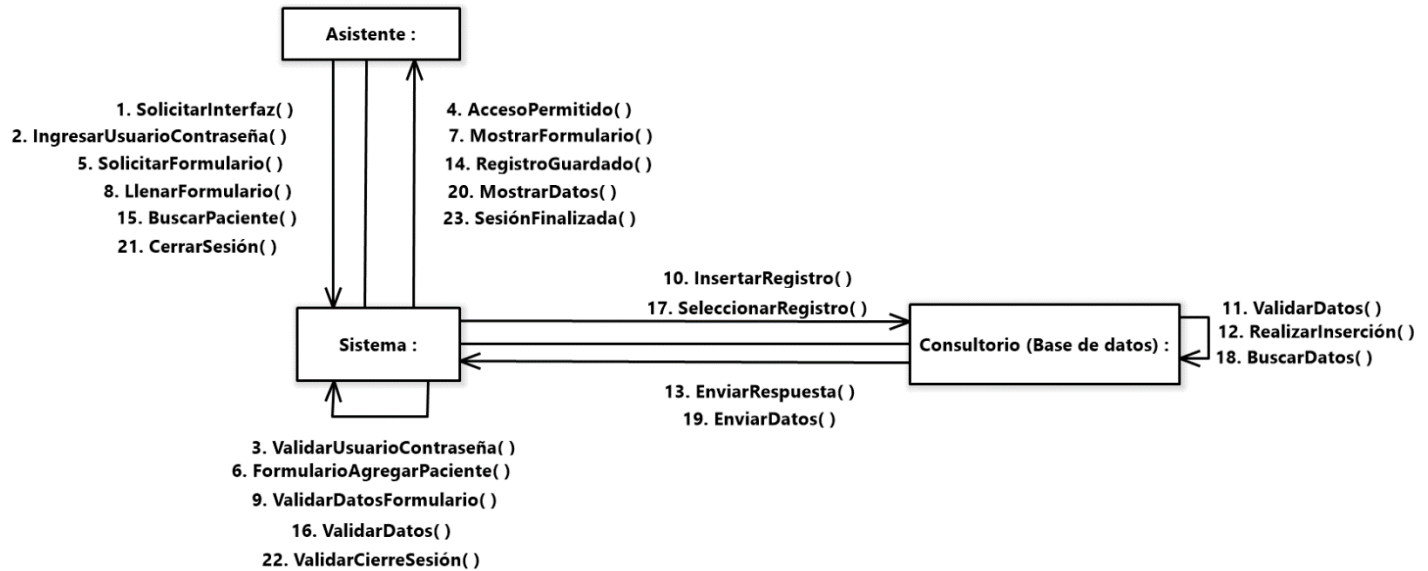


Figura 11. Ejemplo de un diagrama de comunicación donde realiza las peticiones entre el usuario, el sistema y la base de datos.

1.1.9 Diagrama de componentes.

Un diagrama de componentes contiene sus respectivos componentes, interfaces y relaciones como se aprecia en la figura 13, donde se observa todo lo que se usa para su creación.

Un componente de software es una parte física de un sistema. Un componente es la interfaz que utiliza una clase es la misma que la que utiliza su implementación de software.

Un componente puede hacer disponible su interfaz para que otros componentes puedan utilizar las operaciones que contiene, es decir, este puede acceder a los servicios de otro componente. El componente que proporciona los servicios se dice que provee una interfaz de exportación.

Tipos de componentes:

Existen 3 tipos:

1. Componentes de distribución que conforman el fundamento de los sistemas ejecutables (DLL, ejecutables, Java Beans y controles ActiveX).
2. Componentes para trabajar en el producto a partir de los cuales se han creado los componentes de distribución (Archivos de base de datos y de código).
3. Componentes de ejecución creados como resultado de un sistema de ejecución.

Se representan mediante un rectángulo que tiene otros dos sobrepuestos en su lado izquierdo. El nombre se debe colocar el nombre del componente dentro del símbolo y este a su vez es una cadena como se aprecia en la figura 12.



Figura 12. Símbolo que representa a un componente.

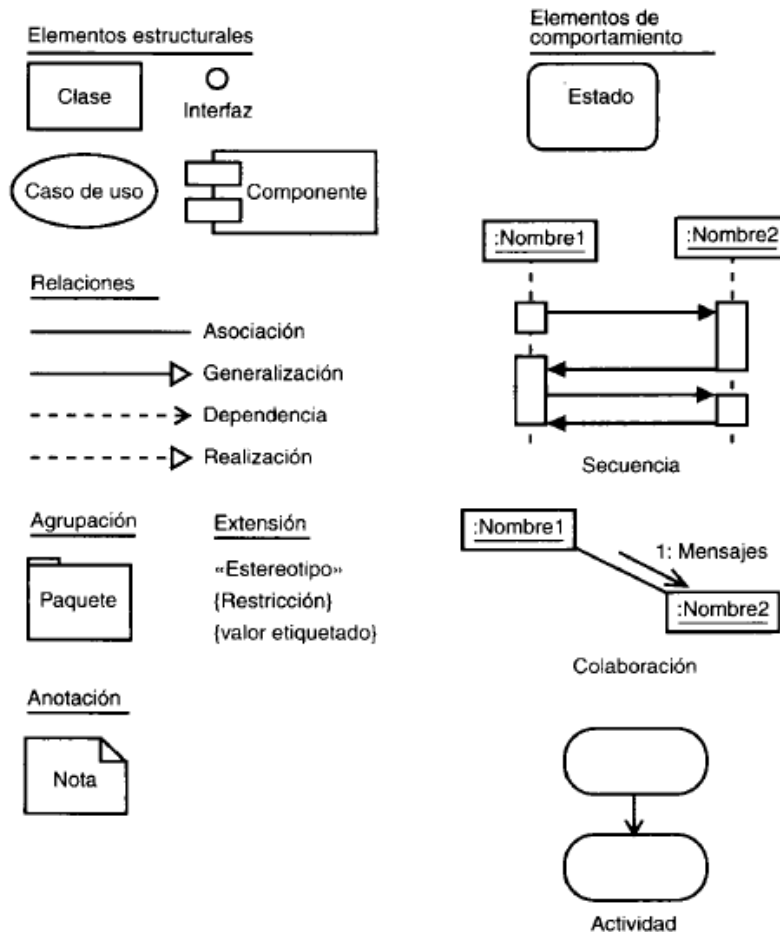


Figura 13. Panorama de lo que se usa en el diagrama de componentes.

1.1.10 Diagrama de despliegue.

La topología de despliegue significa la forma en que el sistema lucirá cuando se ponga en uso.

Este diagrama muestra al lector los elementos lógicos, sus ubicaciones físicas y como se

comunican estos elementos, así como el número y variedad de elementos físicos y lógicos.

Sirve para mostrar en donde está el servidor web y si se tiene más de uno. Se usan para mostrar donde está el servidor de la base de datos o si se tiene más de uno, así como cuales son las relaciones de los servidores con otros elementos, es decir, como están conectados estos elementos, cuales protocolos se están usando para comunicarse y cuales sistemas operativos o dispositivos físicos (incluyendo computadoras y otros dispositivos) están presentes.

Modelado de nodos:

Los nodos son cajas tridimensionales, como se observa en la figura 14 que representan dispositivos físicos o entornos de ejecución que pueden ser computadoras, sistemas operativos o entornos de autocontención.

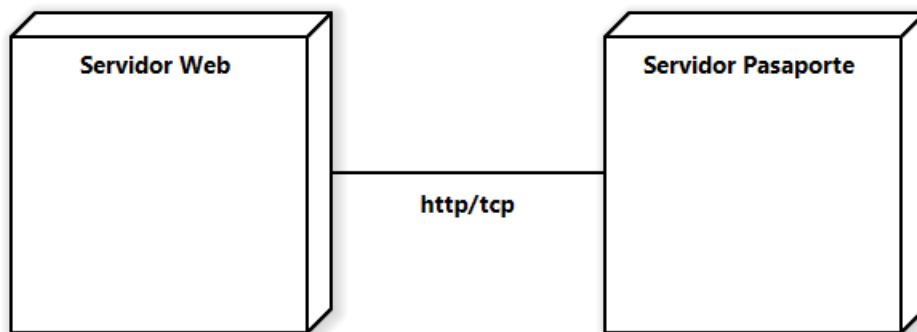


Figura 14. Ejemplo de nodos conectando del servidor de la oficina de pasaporte al servidor principal.

1.2 Diagramas para base de datos.

En este capítulo se expondrá todo lo relacionado a las bases de datos, por lo cual se inicia con la pregunta que se presenta en el capítulo 1.2.1 hasta tener la idea de lo que son, para que sirven, así como su uso.

1.2.1 ¿Qué es una base de datos?

Es una serie de datos relacionados que forman una estructura lógica, es decir una estructura reconocible desde un programa informático.

Según Ramírez (2004), están compuestas de datos y de metadatos. Los metadatos son datos que sirven para especificar la estructura de las bases de datos por ejemplo que

tipos de datos se almacenan, que nombre se le da a cada dato, como están agrupados, cómo se relacionan, etcétera.

Se dividen en dos grupos:

- Estructura lógica

Indica la composición y distribución teórica de la base de datos. La estructura lógica sirve para que las aplicaciones puedan utilizar los elementos de la base de datos sin saber realmente cómo se están almacenando.

Es una estructura que permite idealizar a la base de datos. Sus elementos son:

- Objetos
- Entidades
- Nodos
- Relaciones
- Enlaces

- Estructura física

Es la estructura de los datos tan cual se almacenan en las unidades de disco. La correspondencia entre la estructura lógica y la física se almacena en la base de datos (metadatos) (p.8).

Según Kendall (2011) los metadatos son datos sobre los datos del archivo o base de datos; describen el nombre proporcionado y la longitud asignada a cada elemento de datos, al igual que describen la longitud y la composición de cada uno de los registros (p. 410).

1.2.2 Diagrama Entidad – Relación.

El modelo de datos entidad relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades y de relaciones entre objetos.

Una entidad es una cosa u objeto en el mundo real que es distinguible de otros objetos.

Las entidades se describen en una base de datos mediante un conjunto de atributos.

En el modelo entidad relación los conjuntos de entidades como se observa en la figura 15, se representan con un rectángulo dentro del cual se escribe el nombre de la entidad.



Figura 15. Representación de una entidad del diagrama entidad relación.

Tipos de entidades.

Existen 2 tipos de entidades:

- **Regulares:**
Son las entidades normales que tienen existencia por sí mismas sin depender de otras. Su representación gráfica es como se observa en la figura 15.
- **Débiles:**

Su existencia depende de otras.

Las entidades débiles se representan como se muestra en la figura 16:

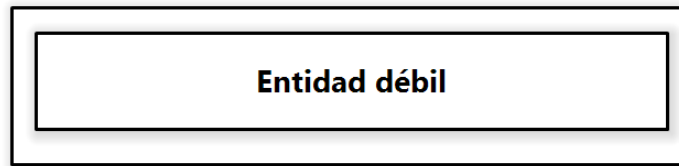


Figura 16. Representación de una entidad débil del diagrama entidad relación.

El diagrama consta de los siguientes elementos principales:

- *Rectángulos*: representan conjuntos de entidades.
- *Elipses*: representan atributos.
- *Rombos*: representan relaciones.
- *Líneas*: unen atributos a conjuntos de entidades y conjuntos de entidades a conjuntos de relaciones.
- *Elipses dobles*: representan atributos multivalorados.
- *Elipses discontinuas*: denotan atributos derivados.
- *Líneas dobles*: indican participación total de una entidad en un conjunto de relaciones.
- *Rectángulos dobles*: representan conjuntos de entidades débiles.

1.2.3 ¿Qué es una relación en las bases de datos?

Representan las asociaciones entre entidades, es decir, es el elemento del modelo que permite relacionar en sí los datos, como se aprecia en la figura 17 en donde el doctor trabaja con los pacientes.

Las entidades se pueden relacionar de diferentes formas tal y como se observa en la figura 18.

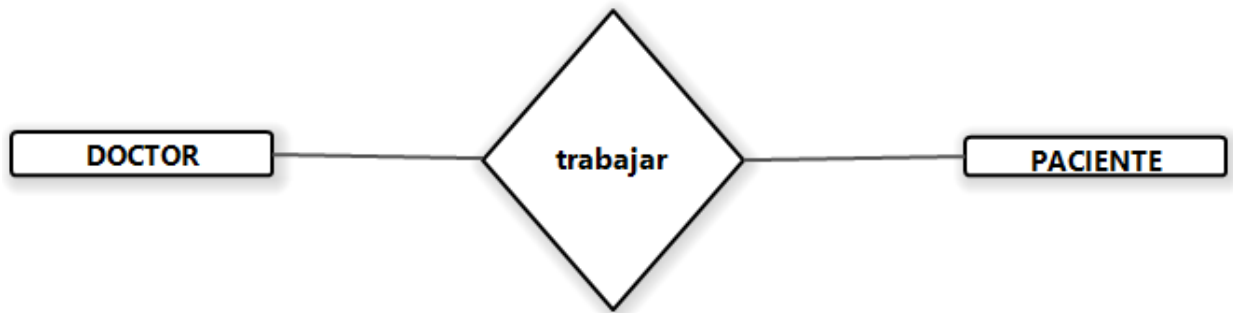


Figura 17. Representación gráfica de la relación de las entidades de doctor y paciente.

Ejemplos de las relaciones que pueden existir en las bases de datos.

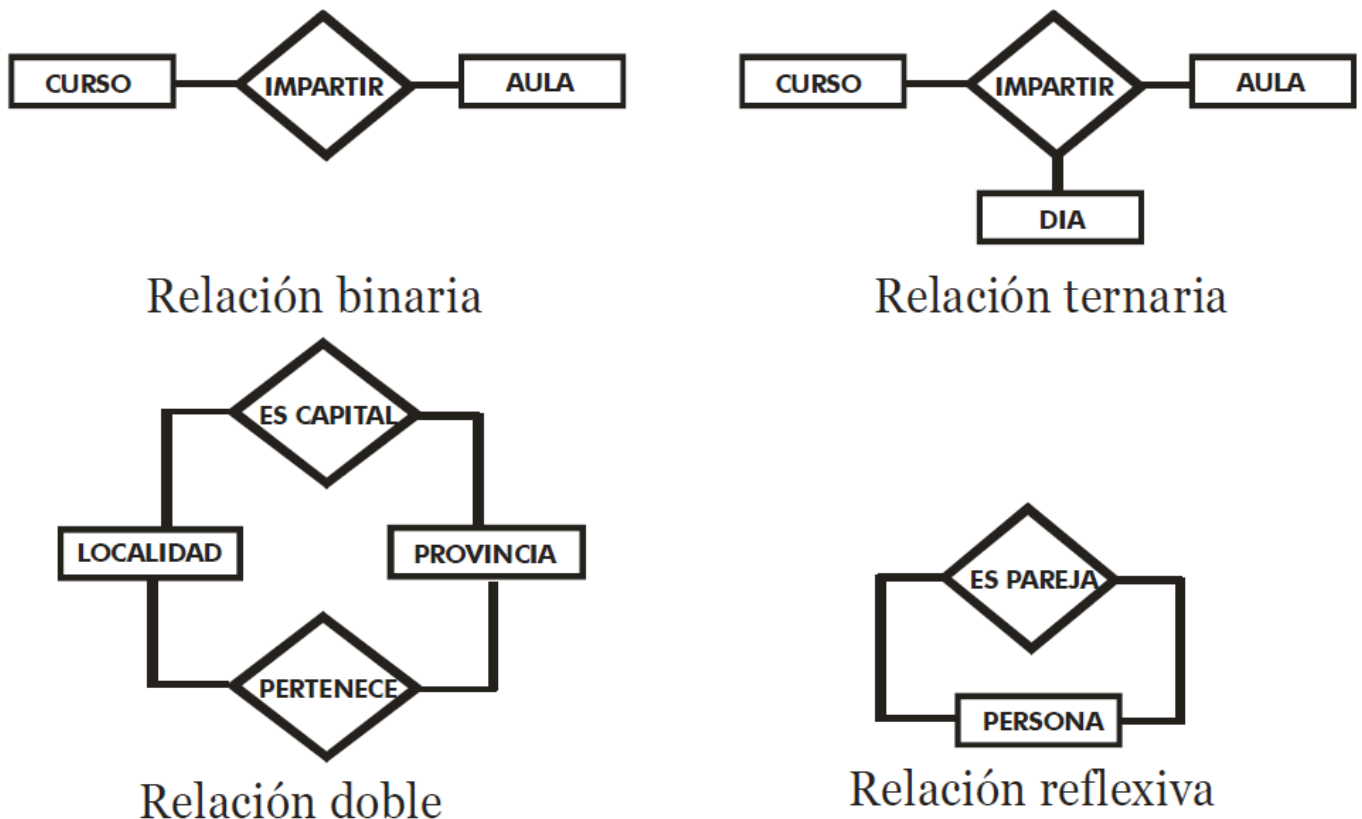


Figura 18. Ejemplos de relaciones.

Cardinalidad.

De acuerdo con Sánchez (2004), “indica el número de relaciones en las que una entidad puede aparecer” (p. 20).

- Cardinalidad mínima.
Indica el número mínimo de asociaciones en las que aparecerá cada ejemplar de la entidad (el valor que se anota es de cero a uno).
- Cardinalidad máxima.
Indica el número máximo de relaciones en las que se puede aparecer cada ejemplar de la entidad (puede ser uno a muchos).

En los esquemas entidad / relación la cardinalidad, como se aprecia en la figura 19 se puede indicar de muchas formas.

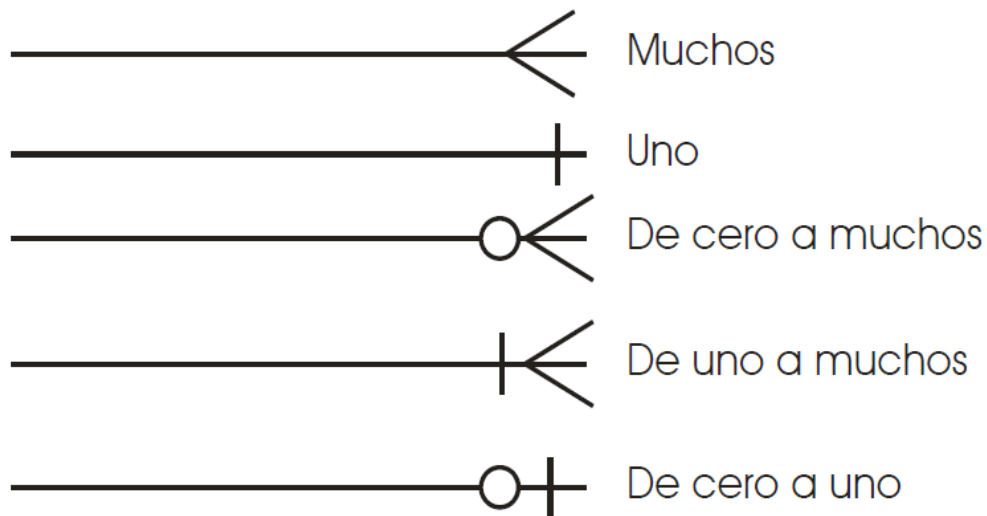


Figura 19. Figura donde muestra las diferentes cardinalidades en un diagrama entidad relación.

Atributos.

Describen propiedades de las entidades y relaciones.

Los tipos de atributos pueden ser:

- Compuestos
- Múltiples

Pasos para el diseño del diagrama entidad relación:

1. Encontrar entidades (conjunto de entidades).
2. Identificar atributos de las entidades.

3. Buscar identificadores.
4. Especificar las relaciones y cardinalidades.
5. Identificar entidades débiles.
6. Especificar y generalizar entidades en dónde sea posible.

Claves en los diagramas entidad relación.

Los valores de los atributos de una entidad deben ser tales que permitan identificar de manera unívocamente a la entidad, es decir, no se permite que ningún par de entidades tengan exactamente los mismos valores de sus atributos.

Una clave permite identificar un conjunto de atributos suficiente para distinguir las entidades entre sí.

Los tipos de claves son:

- Superclave: es un conjunto de uno o más atributos que permiten identificar de forma única una entidad en el conjunto de entidades.
- Llave primaria: es el elemento principal para identificar las entidades dentro de un conjunto de entidades. Se debe de elegir de una manera en que sus atributos nunca o muy raramente cambien.
- Llave foránea: atributo simple o compuesto en una relación que está definido en el mismo dominio que la llave primaria en la relación.

1.3 Conceptos.

En este capítulo se abordarán los temas con su respectiva definición que el licenciado en informática, principalmente el desarrollador web domina, así como las herramientas que usa para llevar a cabo su profesión. A continuación, se explican los conceptos que se utilizan para el desarrollo del sistema que se menciona en el capítulo 3.

1.3.1 Sistema de información.

Los sistemas de información como cualquier otro sistema cuya finalidad dentro de una organización son procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas.

Los sistemas de información están formados por subsistemas que incluyen hardware, software, medios de almacenamiento de datos para archivos y bases de datos. Los sistemas que poseen datos, en forma tal que puedan ser utilizados por quien los recibe

para fines de tomas de decisiones. (Senn, 2005, p. 23).

1.3.2 HTML.

El lenguaje HTML (HyperText Markup Language) se utiliza para crear documentos que muestren una estructura de hipertexto. Un documento de hipertexto es aquel que contiene información cruzada con otros documentos lo cual nos permite pasar de un documento al referenciado desde la primera aplicación con la que estamos visualizando. Los documentos HTML básico se conforman como documentos de texto plano (sin ningún tipo de formateo especial) en la que todo el formato del texto se especifica mediante marcas de texto (llamados tags) que delimitan los contenidos a los que afecta la etiqueta.

HTML usa un lenguaje de etiquetas para construir páginas web. Estas etiquetas HTML son palabras clave y atributos rodeados de los signos mayor y menor, por ejemplo: <html>.

La mayoría de las etiquetas se utilizan en pares una etiqueta de apertura y una etiqueta de cierre y el contenido se declara entre ellas.

<html lang="es"> Indica el comienzo del código HTML

</html> Indica el final.

La diferencia de las etiquetas entre una y otra es que en la de cierre se distingue por una barra invertida antes de la palabra clave. (Mateu, 2004, pp. 51-52).

HTML es el lenguaje usado por los navegadores para mostrar las páginas web al usuario, siendo hoy en día la interface más extendida en la red. Este lenguaje nos permite aglutinar textos, sonidos e imágenes y combinarlos a nuestro gusto. (Cornejo, 2012, p. 7)

1.3.3 Bootstrap.

Es un framework de interfaz de usuario, de código abierto, creado para un desarrollo web más rápido y sencillo. Mark Otto y Jacob Thornton fueron de los creadores iniciales.

Contiene todo tipo de plantillas de diseño basadas en HTML y CSS para diversas funciones y componentes como navegación, sistema de cuadrícula, carruseles de imágenes y botones.

Ahorra tiempo al desarrollador de tener que administrar las plantillas repetidamente, su objetivo principal es crear sitios responsive. Permite que la interfaz de usuario de un sitio web funcione de manera óptima en todos los tamaños de pantalla ya sea en teléfonos de

pantalla pequeña o en dispositivos de escritorio de pantalla grande.

Hay 3 archivos primarios de Bootstrap que administran esta interfaz de usuario y la funcionalidad.

- *Bootstrap.css*: Es un framework CSS que organiza y gestiona el diseño de un sitio web. Mientras que HTML gestiona el contenido y la estructura de una página web y el CSS se ocupa del diseño del sitio. Debido a sus funciones, CSS te permite crear una apariencia uniforme en tantas páginas web como se desee. Todo lo que se necesita hacer es referir las páginas web al archivo CSS. Cualquier alteración necesaria se puede hacer en ese único archivo.
- *Bootstrap.js*: Este es el archivo es la parte principal de Bootstrap. Consiste en archivos Javascript que son responsables de la interactividad del sitio web.
- *Glyphicons*: los iconos son una parte integral de la interfaz de un sitio web por lo que Bootstrap incluye un conjunto de ellos para uso gratuito.

El framework es muy flexible y puede adaptarse a casi todas las necesidades de desarrollo web front-end. Su mejor característica es las plantillas de diseño que hacen que las páginas web funcionen de forma óptima en todos los tamaños de pantalla. (Deyimar, 2022).

1.3.4 PHP.

Sus siglas recurren a un acrónimo recursivo (PHP: Hypertext preprocessor) es un lenguaje sencillo de sintaxis cómoda y similar a la de otros lenguajes como Perl, C y C++. Es un lenguaje rápido, interpretado, orientado a objetos y multiplataforma. Existen una multitud de librerías son de código libre con las que el programador dispone un impresionante arsenal de herramientas libres para desarrollar aplicaciones.

Es un lenguaje ideal para aprender a desarrollar aplicaciones web como para desarrollar aplicaciones web complejas.

Suele ser utilizado juntamente con Perl, Apache, MySQL o PostgreSQL (estos componentes son de código libre).

¿Cómo funciona?

Si solicitamos a nuestro servidor una página PHP esta envía dicha página al intérprete

de PHP que la ejecuta y devuelve el resultado en un archivo HTML al servidor web el cual se lo enviará al cliente. (Mateu, 2004, pp. 186-187)

1.3.5 JavaScript.

Según Mateu (2004), “es un lenguaje de programación interpretado un lenguaje de tipo script” (p. 93). Es ligero, interpretado o compilado justo a tiempo (just-in-time) basado en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa.

Este lenguaje se usa para añadir características interactivas a tu sitio web como pueden ser:

- Juegos
- Eventos que ocurren al presionar los botones
- Datos que son introducidos en los formularios
- Efectos de estilo dinámicos
- Animación
- Etcétera

No existen intérpretes no dependientes de ningún navegador en un lenguaje de script que suele encontrarse vinculado a páginas web. Javascript y Java son dos lenguajes de programación distintos con filosofías muy diferentes. El único punto en común es la sintaxis ya que cuando Netscape diseñó Javascript se inspiró en la sintaxis de Java.

Puedes empezar con pequeñas cosas como carruseles, galerías de imágenes, diseños fluctuantes y respuestas a las pulsaciones de botones. También es capaz de crear juegos, animaciones 2D y gráficos 3D, aplicaciones integradas basadas en bases de datos. (MDN Contributors, 2022)

1.3.6 MySQL.

MySQL es un sistema gestor de base de datos (SGBD, DBMS) es muy conocido y ampliamente usado por su simplicidad y notable rendimiento.

Es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto.

Es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento

precisamente por su facilidad de uso y tiempo reducido de puesta en marcha.

Está disponible para múltiples plataformas, pero sus diferencias con cualquier otra plataforma son prácticamente nulas ya que la herramienta utilizada en este caso es el cliente mysql-client que permite interactuar con un servidor MySQL (local o remoto) en modo texto. De este modo es posible realizar todos los ejercicios sobre un servidor instalado localmente o a través de internet sobre un servidor remoto.

Características de MySQL.

Trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente.

MySQL está basado en código abierto y es de fácil acceso ya que la mayoría de los programadores que trabajan en desarrollo web han pasado usar MySQL en sus proyectos.

Algunas otras características son:

- **Arquitectura cliente y servidor:** MySQL basa su funcionamiento en un modelo cliente y servidor. Clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios o establecer nuevas tablas de registros.
- **Compatibilidad con SQL:** es un lenguaje generalizado dentro de la industria. Ofrece plena compatibilidad por lo que si has trabajado en otro motor de bases de datos no tendrás problemas en migrar a MySQL.
- **Vistas:** ofrece compatibilidad para poder configurar vistas personalizadas del mismo que podemos hacerlo en otras bases de datos SQL. En bases de datos de gran tamaño las vistas se hacen un recurso imprescindible.
- **Procedimientos almacenados:** MySQL posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.
- **Desencadenantes:** permite además poder automatizar ciertas tareas dentro de la base de datos.
- **Transacciones:** el sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas. (Robledano, 2019)

1.3.7 Navegador web.

Es una aplicación de software que permite acceder a la World Wide Web. Con un solo clic, abre una ventana a todo el conocimiento humano: puede buscar respuestas a todas las preguntas que tenga. Con un navegador web, puede navegar en cualquier página web y visitar fácilmente otros sitios. (Bodnar, 2021)

Es un software o aplicación que permite al usuario acceder a sitios en Internet alrededor del mundo para conseguir información. Con el fin de brindar una respuesta confiable, el navegador obtiene datos de un servidor web y luego se los enseña a un usuario a través de una pantalla.

Ejemplos de navegadores web.

Estas plataformas se encuentran disponibles para todo tipo de dispositivos como computadores, teléfonos, tablets e incluso televisores inteligentes. Algunos ejemplos de navegadores web son:

- Google Chrome
- Safari
- Firefox
- Opera
- Eminent
- Internet Explorer (Vega, 2017.)

1.3.8 Protocolos HTTP y HTTPS.

HTTP por sus siglas en inglés HyperText Transfer Protocol es el nombre del protocolo del cual nos permite realizar una petición de datos y recursos como son los archivos de HTML.

Es la base de cualquier intercambio de datos en la web y un protocolo de estructura cliente – servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador web.

Clientes y servidores se comunican intercambiando mensajes individuales (en contraposición a las comunicaciones que utilizan flujos continuos de datos). Los mensajes que envía el cliente a través del navegador web se le llaman peticiones y los mensajes enviados por el servidor se llaman respuestas.

Características clave del protocolo HTTP.

HTTP sencillo:

Está pensado y desarrollado para ser leído y fácilmente interpretado por las personas haciendo de esta manera más fácil la depuración de errores y reduciendo la curva de aprendizaje para las personas que empiezan a trabajar con él.

HTTP extensible:

Presentadas en la versión HTTP/1.0 las cabeceras de HTTP han hecho que este protocolo sea fácil de ampliar y de experimentar con él. Funcionalidades nuevas pueden desarrollarse sin más que un cliente y su servidor comprendan la misma semántica sobre las cabeceras de HTTP.

HTTP es un protocolo con sesiones, pero sin estados:

Es un protocolo, pero sin estado, es decir, no guarda ningún dato entre dos peticiones en la misma sesión. Esto crea problemáticas en caso de que los usuarios requieran interactuar con determinadas páginas web de forma ordenada y coherente.

El uso de HTTP cookies, si permite guardar datos con respecto a la sesión de comunicación. Usando la capacidad de ampliación del protocolo HTTP, las cookies permiten crear un contexto común para cada sesión de comunicación.

HTTP y conexiones:

Una conexión se gestiona al nivel de la capa de transporte y por lo tanto queda fuera del alcance del protocolo HTTP, no necesita que el protocolo que lo sustenta mantenga una conexión continua entre los participantes en la comunicación, solamente necesita que sea un protocolo fiable o que no pierda mensajes como mínimo un protocolo que sea capaz de detectar que se ha pedido un mensaje y reporte un error. (MDN, 2022)

1.3.9 Cliente / Servidor.

Es un sistema distribuido entre múltiples procesadores donde hay clientes que solicitan servicios y servidores que los proporcionan. Es un modelo que implica productos y servicios enmarcados en el uso de la tecnología de punta y permite la distribución de la información en forma ágil y eficaz a las diversas áreas de una organización (empresa u organización privada o pública).

En la arquitectura cliente/Servidor:

- Los clientes (o programas que representan entidades que necesitan servicios) y los servidores (o programas que proporcionan servicios) son objetos separados desde un punto de vista lógico y que se comunican a través de una red de comunicaciones para realizar una o varias tareas de forma conjunta.

Un cliente hace una petición de un servicio y recibe la respuesta a dicha petición; un servidor recibe y procesa la petición y devuelve la respuesta solicitada.

Hay diferentes tipos de tecnología del cliente/servidor, se explicarán algunos de ellos:

- Servidores de ficheros:

Los clientes son quienes hacen solicitudes de ficheros al servidor. Las formas de compartir ficheros en una red pueden ser: repositorios de documentos, imágenes, programas, etcétera.

- Servidores de bases de datos:

Aplicaciones del cliente mandan solicitudes SQL al servidor. El servidor devuelve el resultado de la consulta.

- Servidores de transacciones:

El cliente invoca procedimientos remotos o transacciones (conjunto de instrucciones SQL) sobre la base de datos.

Los datos intercambiados son:

- Cliente -> servidor: solicitud
- Servidor -> cliente: mensaje de resultado

- Servidores groupware:

Intercambio de información semiestructurada como son el de texto, imágenes, etcétera.

En estos tiempos cada vez se usa más el E-mail.

- Servidores de aplicaciones de objetos:

Los objetos del cliente usan un Object Request Broker (ORB). El cliente invoca un método remoto, así como localizar una instancia de la clase del objeto en el servidor, invoca el método y devuelve el resultado al objeto del cliente.

- Servidores de aplicaciones web:

La arquitectura cliente/servidor necesita de la world wide web. La solicitud es por nombre y el protocolo es HTTP.

Funciones del servidor:

- Esperar peticiones de clientes (mensajes). A veces sesión por cliente y otro conjunto dinámico de sesiones.
- Atender soluciones simultaneas. Sin riesgo para la integridad de los recursos compartidos.
- Prioridades en la atención de las solicitudes.
- Capacidad de lanzar tareas en segundo plano no relacionadas con el servicio.
- Robustez: critica en servidores.
- Escalabilidad y extensibilidad.

Características del cliente:

Se dividen en tres tipos.

- Sin GUI (Interfaz gráfico de usuario).

Lectores de códigos de barras.

- Con GUI.

Sustituyeron a los terminales sin gráficos.

Normalmente usan el modelo objeto/acción: selección de objetos y acciones para realizar estos.

Normalmente los diálogos de naturaleza secuencial.

Algunos ejemplos son los sistemas operativos antiguos o las páginas web con formularios.

- Con OOUI (Interfaz de usuario orientada a objetos).

El usuario manipula de forma objetos en pantalla lo que conocemos como drag and drop. (E.U. Informática en Segovia, Departamento de informática [Universidad de Valladolid], 2006)

1.4 SOFTWARE QUE REQUIERE EL DESARROLLADOR.

El desarrollador de software requiere de herramientas para realizar su trabajo con los programas que se han creado para ese fin, es por lo que en este capítulo se abordarán algunos de ellos, así como una breve explicación y en donde los podemos descargar.

1.4.1 Visual Studio Code.

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Incluye soporte incorporado para JavaScript, TypeScript y Node.js y tiene un amplio ecosistema de extensiones para diferentes lenguajes de programación como C++, C#, Java, Python, PHP y Go, así como para tiempos de ejecución como .NET y Unity.

La interfaz gráfica del software se observa en la figura 20.

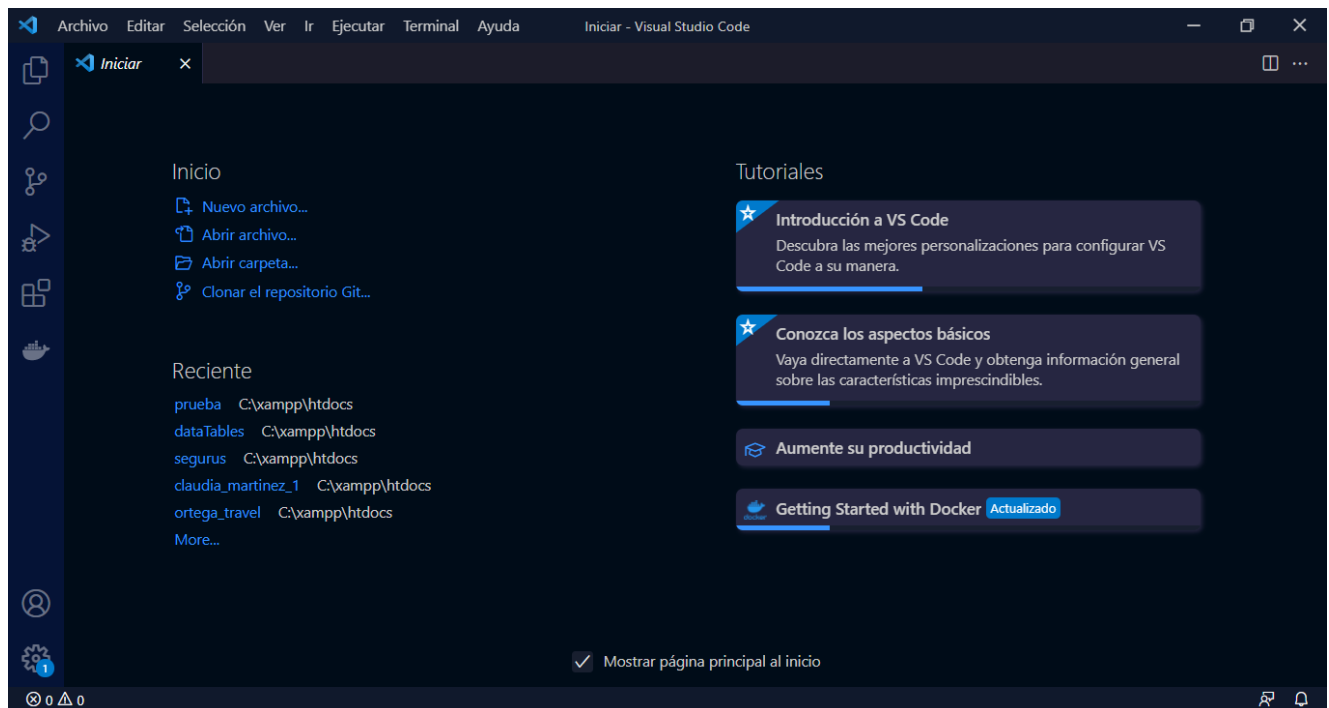


Figura 20. Pantalla principal de Visual Studio Code.

Para descargar el software es en la siguiente página:

<https://code.visualstudio.com/>

1.4.2 Apache.

Es un proyecto de servidor apache HTTP de internet el cual está basado en un esfuerzo por desarrollar y mantener un servidor HTTP de código abierto para los sistemas operativos modernos incluidos UNIX y Windows.

Su objetivo principal es proporcionar un servidor seguro, eficiente y extensible que brinde servicios HTTP en sincronía con los estándares HTTP actuales.

El servidor Apache HTTP (“httpd”) ha sido el servidor web más popular en internet desde el mes de abril de 1995.

Lo que define a Apache es:

- Abierto:

Apache Software Foundation brinda soporte para más de 350 proyectos apache y sus comunidades lo que promueve su misión de proporcionar software de código abierto para el bien público.

- Innovación:

Los proyectos de apache se definen por procesos colaborativos basados en el consenso, una licencia de software abierta y pragmática con el deseo de crear software de alta calidad que lidere el camino en su campo.

- Comunidad:

Es una comunidad de desarrolladores y usuarios de proyectos apache de código abierto de nivel empresarial que se utilizan en todos los países del planeta conectados a internet. Apache tiene una gran lista de proyectos que es por eso por lo que es utilizado por todo el mundo.

Se puede descargar en su página oficial:

<https://httpd.apache.org/download.cgi>

o bien, utilizar algunas opciones populares para implementarlo.

En el siguiente capítulo se explicará uno de los que comúnmente se utiliza y que en nuestro caso también se utilizó en el ámbito profesional. (The Apache Software Foundation,2022)

1.4.3 Xampp.

Es el entorno más popular de desarrollo PHP. Es una distribución de Apache completamente gratuita y fácil de instalar que contiene MaríaDB, PHP y Perl.

El paquete de instalación de Xampp ha sido diseñado para ser increíblemente fácil de instalar y usar como podemos observar en la figura 21 su interfaz de panel de control para poder iniciar cada servicio. En la figura 22 se puede apreciar la pantalla para el ingreso de la consola y trabajar con comandos.

Para descargarlo desde su página oficial es:

<https://www.apachefriends.org/es/download.html>

Aquí se encuentra para diferentes sistemas operativos como son Windows, Linux y OS X.

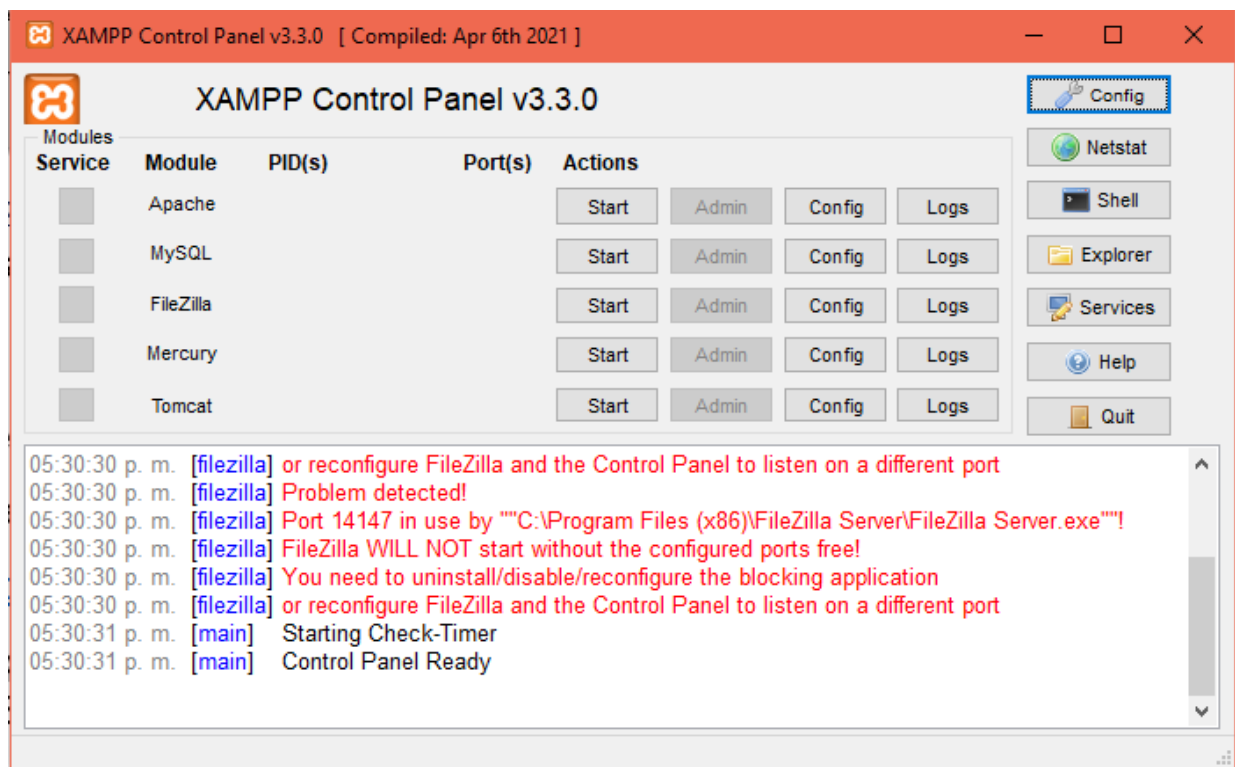


Figura 21. Interfaz gráfica de Xampp.



```
Setting environment for using XAMPP for Windows.
loshe@FANNY c:\xampp
# _
```

Figura 22. Interfaz de comandos para acceder a la base de datos desde consola.

1.4.4 HeidiSQL.

HeidiSQL es un software libre y tiene como objetivo que sea fácil de aprender. Permite ver, editar datos y estructuras desde computadoras que ejecutan uno de los sistemas de base de datos como pueden ser: MariaDB, MySQL, Microsoft SQL, PostgreSQL y SQLite. Inventado en 2002 por Ansgar, HeidiSQL pertenece a las herramientas más populares para MariaDB y MySQL en todo el mundo. La pantalla principal de esta herramienta se aprecia en la figura 23. Una vez ingresado el usuario, se procede a la siguiente pantalla donde se tienen todas las bases de datos almacenadas como se aprecia en la figura 24. Sus características son:

- Gratis para todos, OpenSource.
- Conéctese a múltiples servidores en una ventana.
- Sistemas de bases de datos compatibles: MariaDB, MySQL, MS SQL, PostgreSQL y SQLite.
- Conéctese a los servidores a través de la línea de comandos.
- Conéctese a través del túnel SSH o pase la configuración SSL.
- Cree y edite tablas, vistas, rutinas almacenadas, disparadores y eventos programados.

- Genere buenas exportaciones de SQL, comprímalas después o colóquelas en el portapapeles.
- Exporte desde un servidor/base de datos directamente a otro servidor/base de datos.
- Administrar privilegios de usuario.
- Importar archivos de texto.
- Exporte filas de tablas como CSV, HTML, XML, SQL, LaTeX, Wiki Markup y PHP Array.
- Explore y edite los datos de la tabla usando una cuadrícula cómoda.
- Tablas de edición masiva (mover a db, cambiar motor, intercalación, etc.).
- Inserción por lotes de archivos ASCII o binarios en tablas.
- Escriba consultas con resaltado de sintaxis personalizable y finalización de código.
- Bastante reformateado SQL desordenado.
- Supervise y elimine los procesos del cliente
- Encuentre texto específico en todas las tablas de todas las bases de datos de un servidor.
- Optimizar y reparar tablas de forma por lotes.
- Inicie una ventana de línea de comando `mysql.exe` paralela usando su configuración de conexión actual.

Para descargar el software visitar la siguiente página web:

<https://www.heidisql.com/download.php>

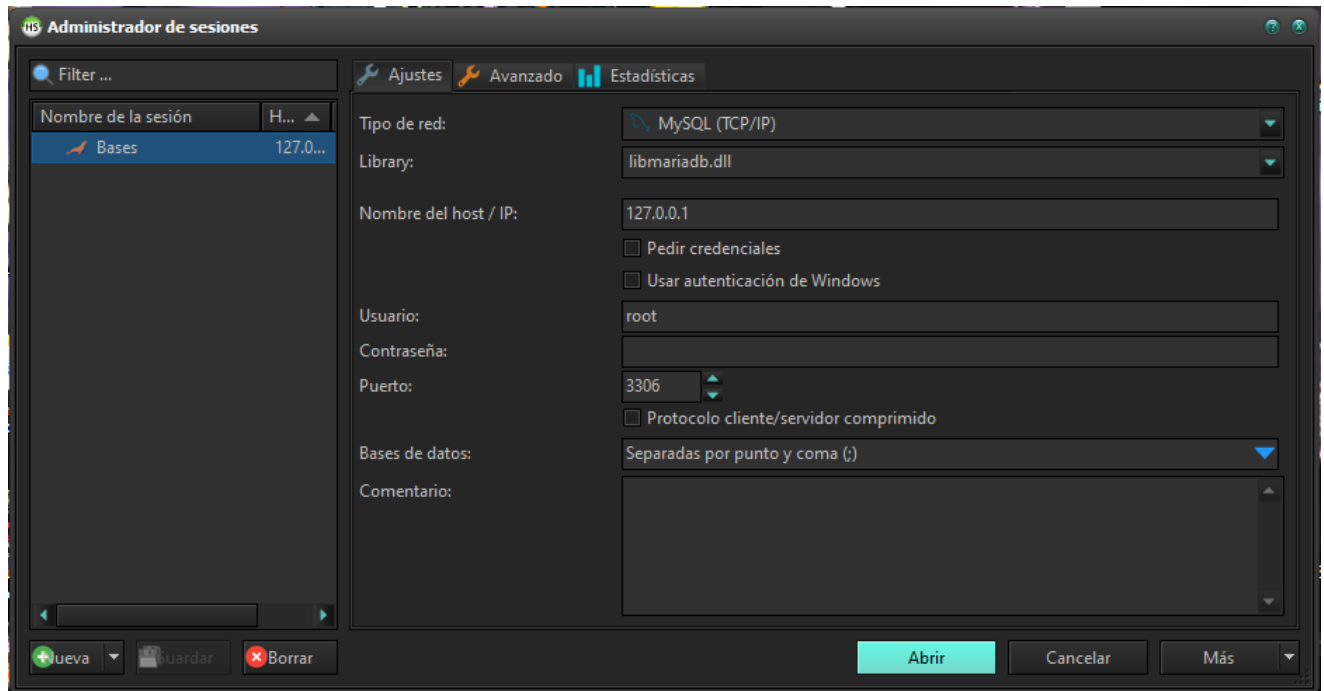


Figura 23. Interfaz principal de HeidiSQL.

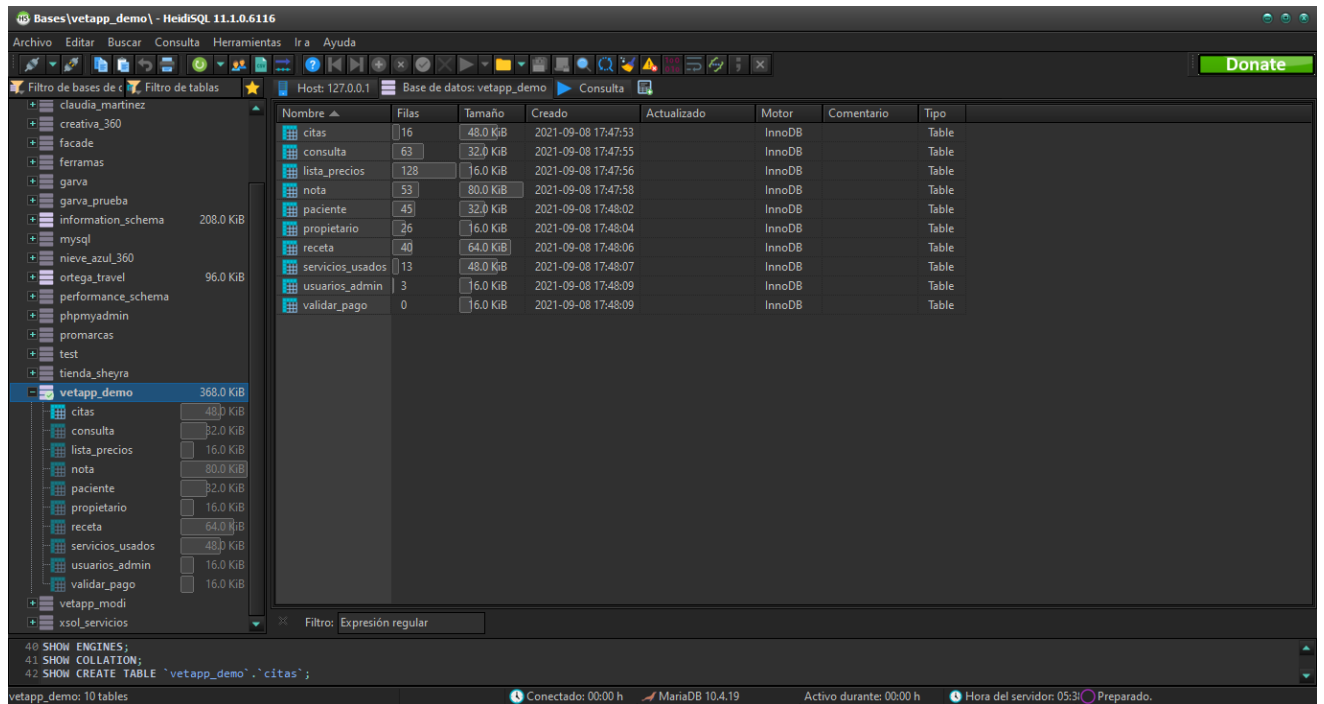


Figura 24. Interfaz donde se muestran todas las bases de datos guardadas.

CAPÍTULO 2. Empresa Lupware.

2.1 Acerca de la empresa.



Figura 25. Logo de Lupware

La empresa Lupware nace en el año 2019 como un proyecto individual del ex alumno de la Facultad de Estudios Superiores Cuautitlán, Javier Alejandro Olguín Mendoza, Licenciado en Informática (gen. 2012 - 2017), con el objetivo de acercar la tecnología a personas que buscan automatizar y mejorar los procesos administrativos en sus negocios o empresas.

El mundo de la informática es extenso, y con el paso del tiempo, cada vez es más claro que las organizaciones requieren de tecnología actualizada y certera para mantenerse vigentes en el mercado, es ahí donde Lupware busca convertirse en el mejor aliado tecnológico de sus clientes.

De hecho, la frase que más se comparte de manera comercial es: “Céntrate en tus negocios, nosotros en la tecnología para ti”; dado que la meta siempre es que los consumidores se sientan tranquilos de que, en Lupware, pueden delegar la responsabilidad de preparar tanto el software como el hardware adecuado para las más altas demandas de trabajo en su organización.

Así pues, con el paso del tiempo, Lupware se ha convertido en un “hub” de soluciones informáticas que a lo largo de 3 años ha desarrollado diversos sistemas informáticos presentes en más de 10 empresas.

Lupware actualmente trabaja con empresas de telecomunicaciones, importadoras de cafeteras italianas, aseguradoras, ferreteras y veterinarias, siendo este último grupo en el que la plataforma “Vet-app” ha logrado estar presente en 5 negocios simultáneos, siendo un éxito comercial donde ha sido presentado.

Sin embargo, el éxito que Lupware ha tenido a lo largo de su corta trayectoria no sería posible sin las extraordinarias personas que han trabajado durante estos 3 años en la

organización y, si bien es cierto que los clientes son muy importantes, lo son aún más los colaboradores, aspecto que está presente en la misión de la compañía:

“Brindar un ambiente acogedor donde nuestros colaboradores se conviertan en nuestra familia y nuestros clientes se conviertan en nuestros invitados de honor”.

Para lograr dicha misión, en Lupware nos regimos bajo los siguientes principios que impulsamos en cada proyecto que desarrollamos:

- Crear un lugar donde todos sean bienvenidos.

Lupware será siempre un lugar donde cualquier persona será bienvenida sin importar su género, ideología, raza o identidad sexual. Ya sea un colaborador o un cliente, siempre encontrarán un lugar seguro para desenvolverse libremente bajo el principio de respetar a los demás.

- Siempre dar lo mejor de nosotros y responsabilizarnos de los resultados.

Un equipo debe mantenerse unido tanto en “las buenas y en las malas”. Desarrollar proyectos muchas veces puede significar un reto importante para todos los integrantes por lo que buscamos crear las condiciones ideales para desempeñar las actividades, así como tomar acción cuando algo no está funcionando adecuadamente. El aprendizaje en el mundo del desarrollo de software es continuo, por lo que

- Tratar a los demás con respeto y amabilidad.

Derivado del punto anterior, por más dificultades que puedan presentarse, siempre el respeto hacia los demás debe ser pilar fundamental de nuestra compañía. Pueden existir diferentes puntos de vista a lo largo del desarrollo de proyectos y siempre es sano presentarlos el uno con el otro, pero nunca olvidar que la amabilidad y el respeto son la llave que nos permitirá crecer como personas y como empresa.

2.2 Organigrama.

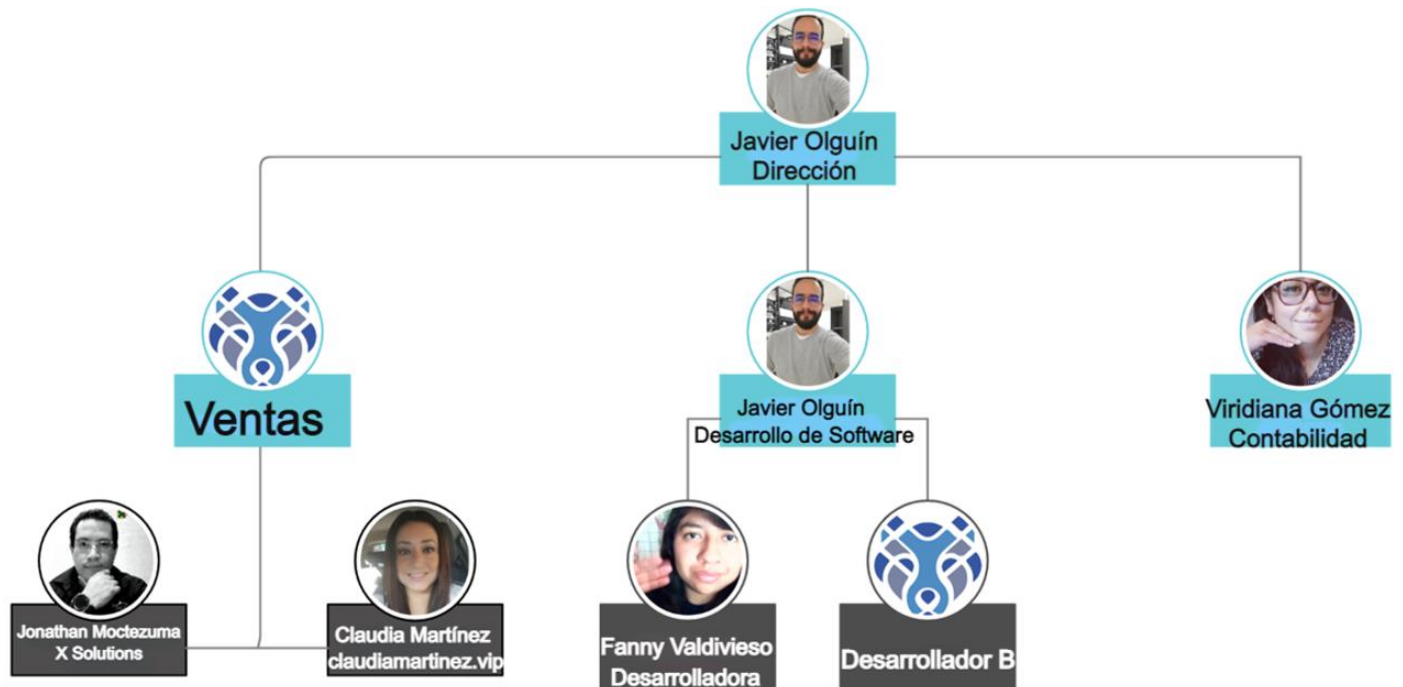


Figura 26. Organigrama de la empresa.

A continuación, se explicará las actividades que realiza cada una de las áreas que conforman al organigrama que se aprecia en la figura 26.

- **Dirección**

Es la encargada de asegurar que todos los proyectos cuenten con todas las validaciones para garantizar que se entregará software de calidad a cada cliente.

Si la Dirección no valida que un proyecto es adecuado para presentarlo, el equipo de Desarrollo atenderá las incidencias presentadas.

Además, la Dirección se encarga de gestionar las buenas relaciones con sus clientes y proveedores para permitir que Lupware se continúe expandiendo en el mercado.

- **Ventas**

Se encarga de gestionar todo lo relacionado con la búsqueda de posibles clientes y realizar el análisis correspondiente para generar propuestas que los prospectos consideren atractivas para elegir a Lupware como la responsable del desarrollo de su solicitud.

Una vez se ha cerrado el proceso de venta, su participación es de vital importancia en el desarrollo para que el equipo tenga todos los elementos necesarios para poder empezar a realizar el proyecto.

- Desarrollo de Software

Es la encargada de llevar a cabo el desarrollo de todos los proyectos que se encuentren activos dentro de la empresa, debido a esto son una de las piezas más importantes para garantizar la entrega de un producto de calidad.

Su trabajo dentro de la organización es de los más sensibles debido a que sus miembros se deben de encontrar totalmente capacitados para detectar problemas dentro de los proyectos que se realizan.

- Contabilidad

Se encarga de gestionar todo lo relacionado al personal y las asignaciones que se tienen dentro de los proyectos en los que se trabaja. Aquí es donde se realizan los análisis necesarios para las decisiones que se tomarán a futuro en la empresa y que afectarán a todos los miembros de la organización.

Constantemente busca y aplica nuevas formas de manejar el negocio para que la empresa se encuentre preparada para saber cómo enfrentar nuevos retos dentro y fuera de la organización.

CAPÍTULO 3. Experiencia en Lupware.

En el momento en el que me integré a esta empresa tuve la oportunidad para ir formándome en el ámbito laboral con los proyectos que me fueron asignados y complementar los que ya se tenían con algunos requerimientos que necesitaban los clientes.

Al asignármelos, fui adquiriendo habilidades nuevas para obtener un mejor control en el orden de la codificación y realización de bases de datos. También se aprendió a utilizar mejor las herramientas de desarrollo, realizar sus respectivos diagramas haciendo un análisis para el correcto funcionamiento hacia el cliente final para que este sea claro, entendible y conciso.

El proyecto que se realizó fue un sistema para médicos veterinarios el cual sirve para llevar el control de historiales clínicos de las mascotas en donde se les brinda un seguimiento desde su primera consulta, que enfermedad le detectaron, recetas, notas, datos personales de su propietario, entre otros. También cuenta con un calendario del mes actual en donde el veterinario se le muestra todas las consultas que tiene agendadas esto con el fin de que no se le olvide al veterinario y al propietario de la mascota se le envía un recordatorio por medio de mensaje de whatsapp o correo electrónico. Para el uso adecuado del sistema o si a los veterinarios les surgen dudas para crear algo en específico pueden revisarlo con los videotutoriales o el manual de usuario que esta anexado en la sección de ayuda.

Algunos veterinarios que adquirieron el software han quedado satisfechos ya que es fácil de usar, entendible y a un costo considerable. También nos apoyaron para dar mejoras al sistema por lo que se tuvo una buena comunicación para un software de calidad.

Para entender más acerca del trabajo que realicé dentro de mi estancia dentro de la empresa lo describiré a continuación en el siguiente capítulo.

3.1 Demostración del proceso de desarrollo del sistema

El haber estado en esta compañía y adquirir experiencia fue una gran oportunidad muy importante ya que apenas empezó en este ámbito del desarrollo, empezar desde cero involucrándose en todo el proceso de una página o sistema, ya que éramos pocos los integrantes en el equipo de desarrollo y por ende se nos pasaba un proyecto completo.

El jefe encargado y dueño de la empresa cuenta con la experiencia profesional y es por ello por lo que es el que se encarga del equipo, realizar las reuniones con los clientes y este una vez ya concretado el negocio es quien nos da las indicaciones para realizar el proyecto.

El esquema con el que se trabaja es bajo FreeLancer, no se tiene un horario fijo, pero si tener el trabajo en tiempo y forma para que se presente al cliente. Siempre se contó con la calidad de entregar el trabajo en tiempo y forma, pero ya si no les gustaba aún se pueden hacer algunos cambios.

De acuerdo con las indicaciones del jefe de desarrollo, se va realizando el proyecto con los requerimientos que el cliente ya solicitó y con ellos ya se va realizando los respectivos

diagramas. Solo por nuestro lado del equipo de desarrollo ya se va diseñando la base de datos, buscar la plantilla que se adecúe a lo que el cliente requiere e ir modificando al igual que ir agregando estilos de CSS.

A continuación, se hará la presentación de uno de los proyectos de todos los que se realizaron que es el de un sistema para llevar el control de veterinarias y que se irá explicando detalladamente.

3.1.1 Realización de diagramas UML.

Para tener un mejor análisis para el desarrollo del sistema web se realizaron sus respectivos diagramas los cuales fueron los que se describieron en el capítulo 1.

Diagrama de casos de uso.

Este fue el primer diagrama que realicé para empezar el sistema donde se muestra al usuario médico veterinario zootecnista (MVZ) que interactúa con el sistema como se observa en la figura 27.

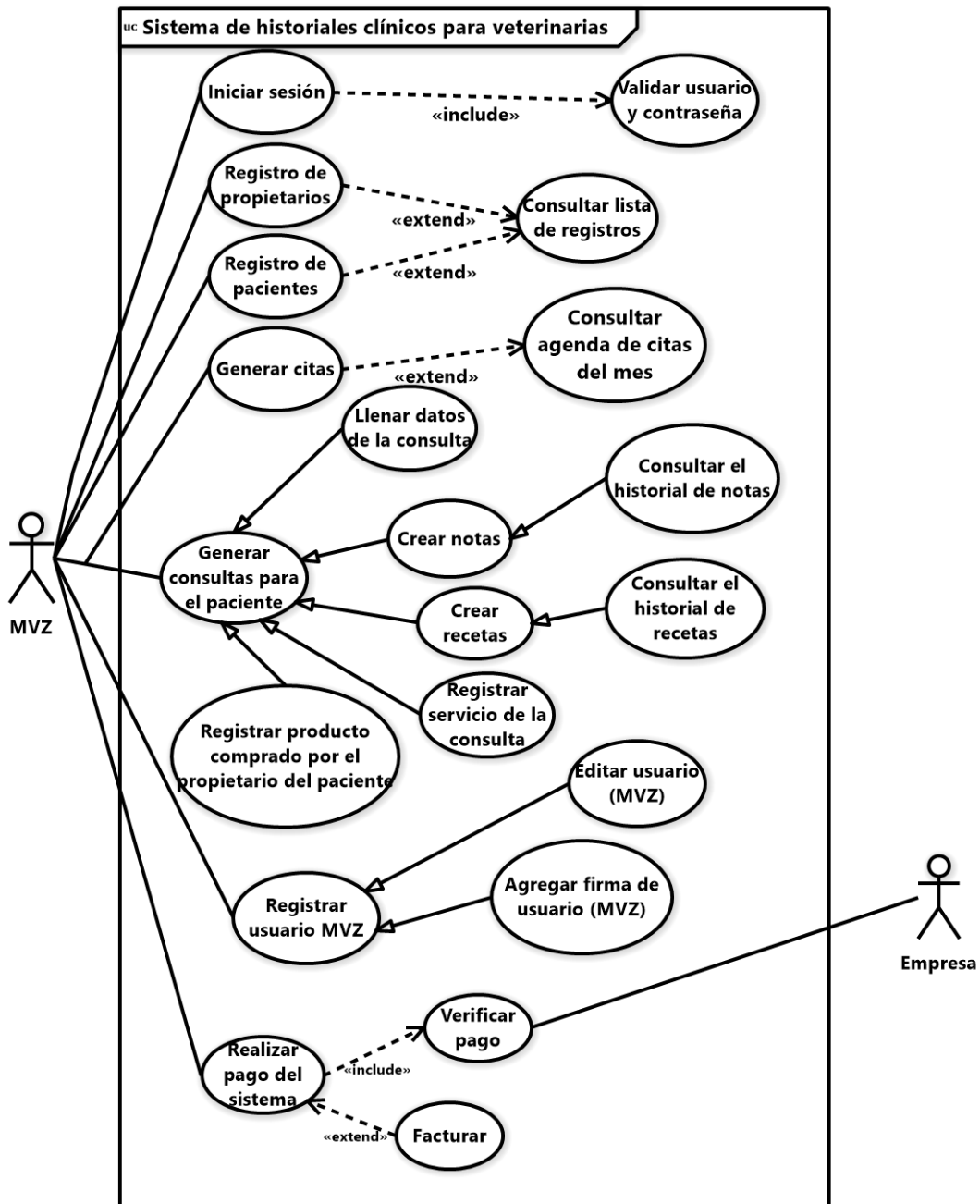


Figura 27. Diagrama de casos de uso de VetApp.

Diagrama de clases.

Diagrama donde indica que tablas, datos, tipos de datos y métodos tendrá el sistema como se observa en la figura 28.

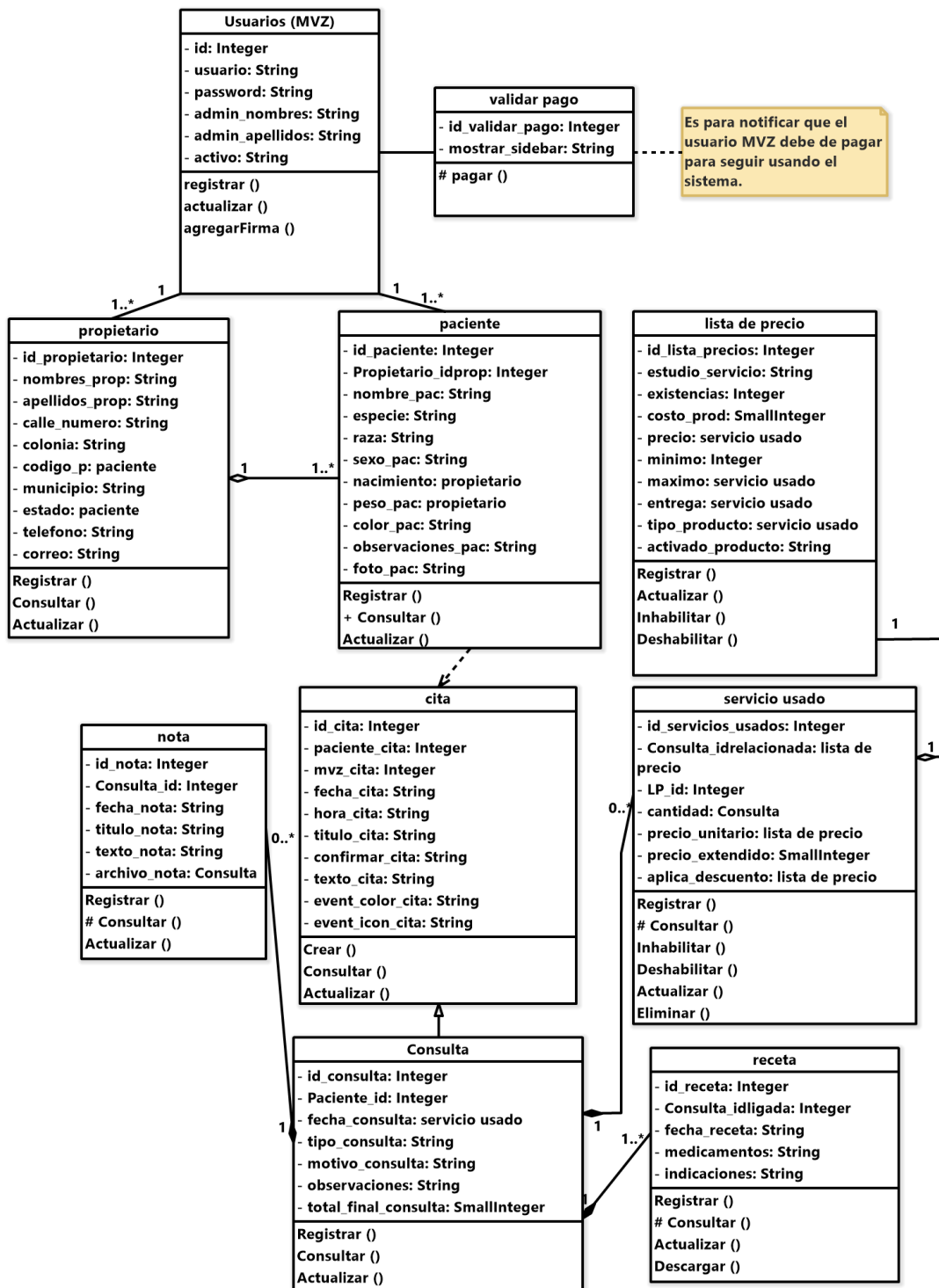


Figura 28. Diagrama de clases de VetApp.

Diagrama de objetos.

Diagrama que indica que tipo de datos se pueden ingresar en el sistema cuando ya se estén levantando registros como se aprecia en la figura 29.

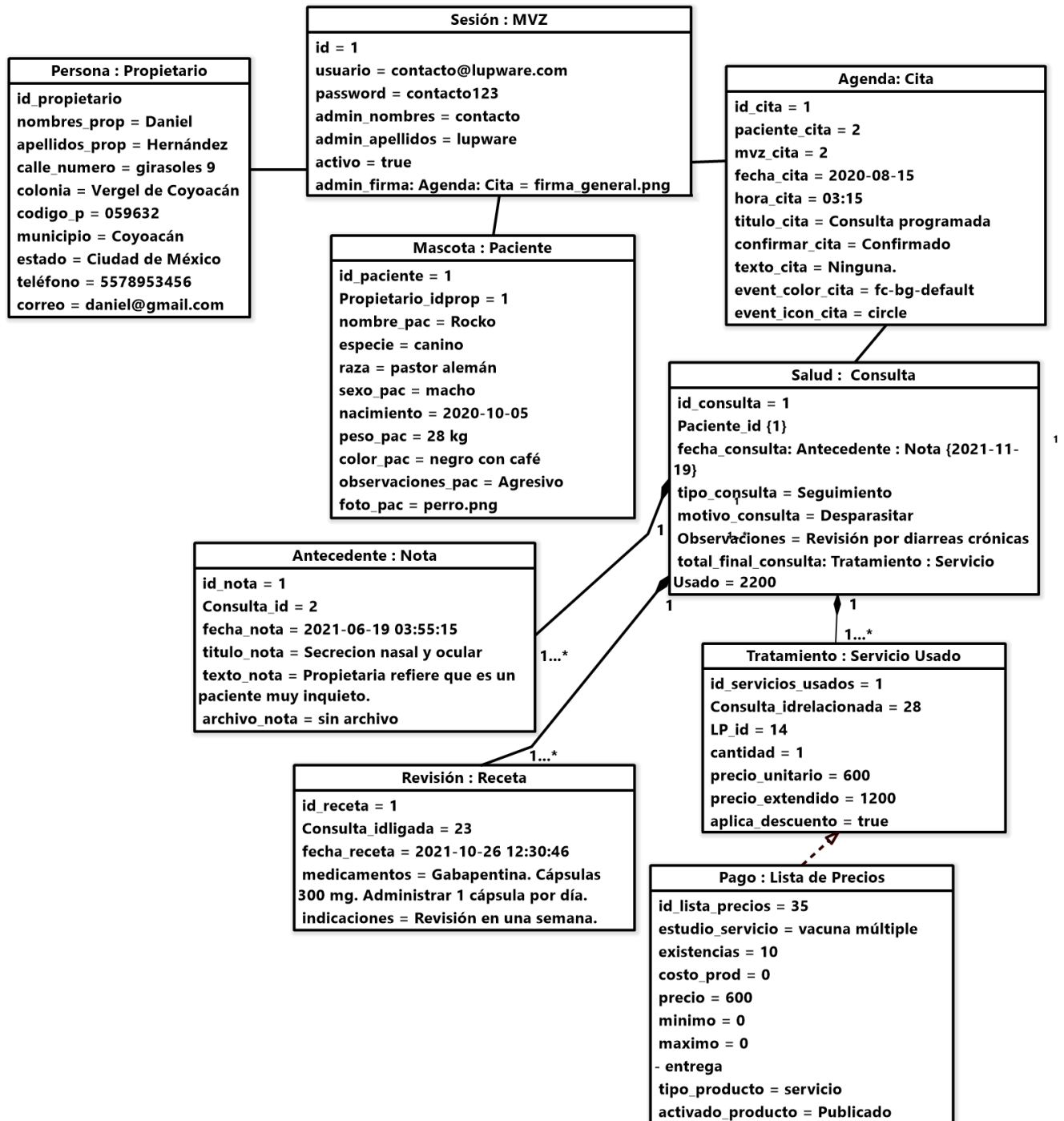


Figura 29. Diagrama de objetos de VetApp.

Diagrama de paquetes.

Diagrama que indica como está dividido el sistema, así como los procesos que realizará cada clase y los tipos de datos que se almacenarán, como se observa en la figura 30.

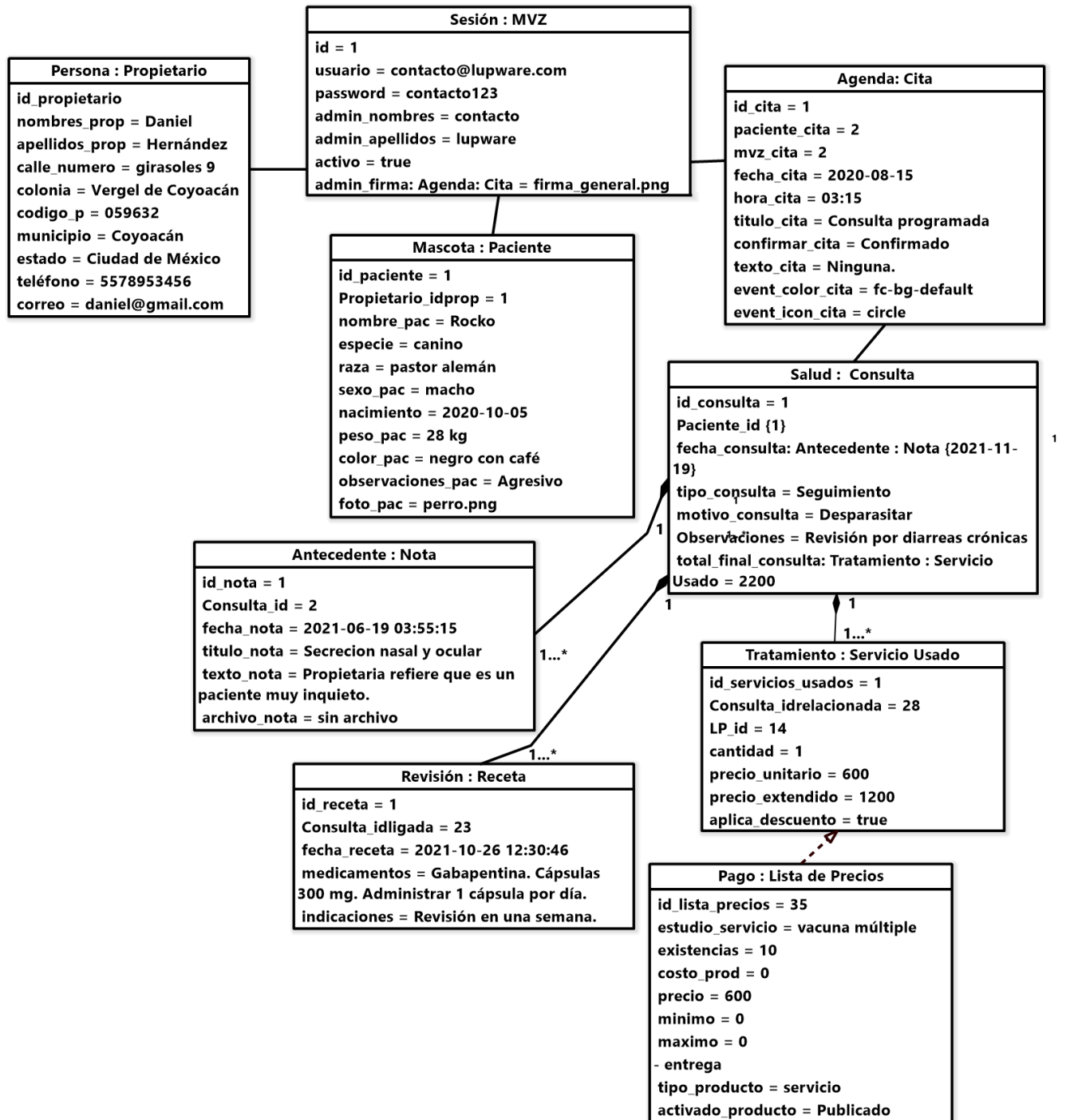
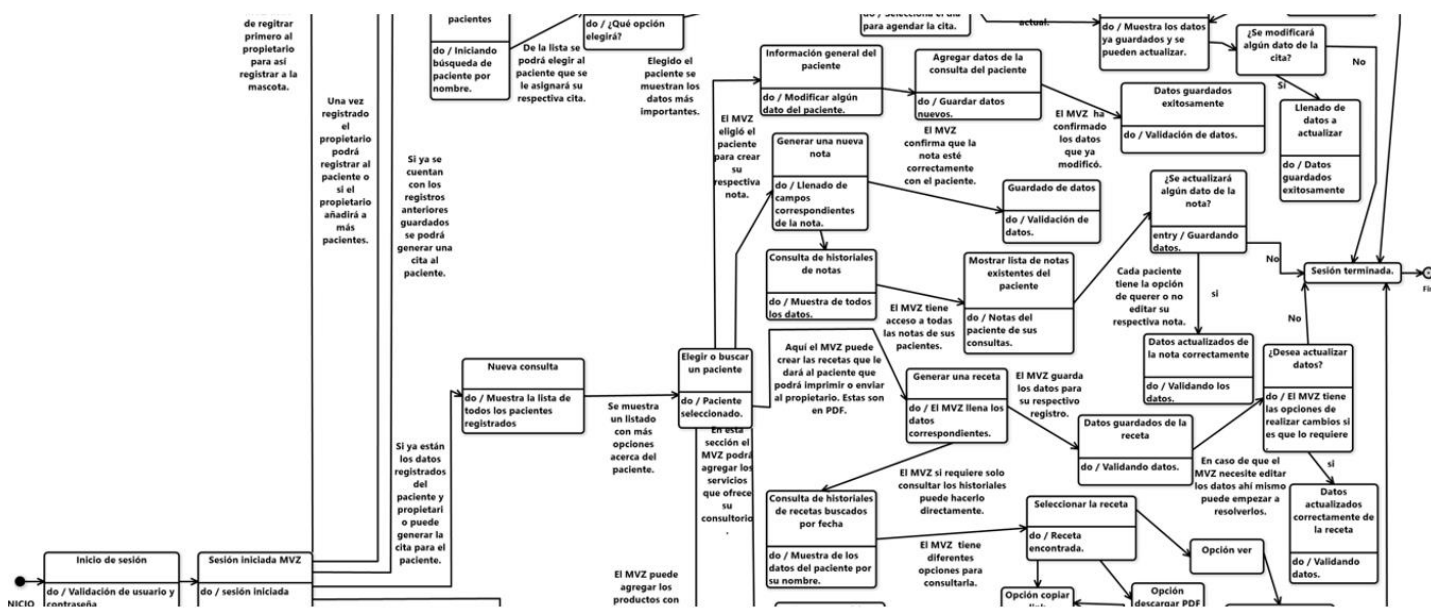
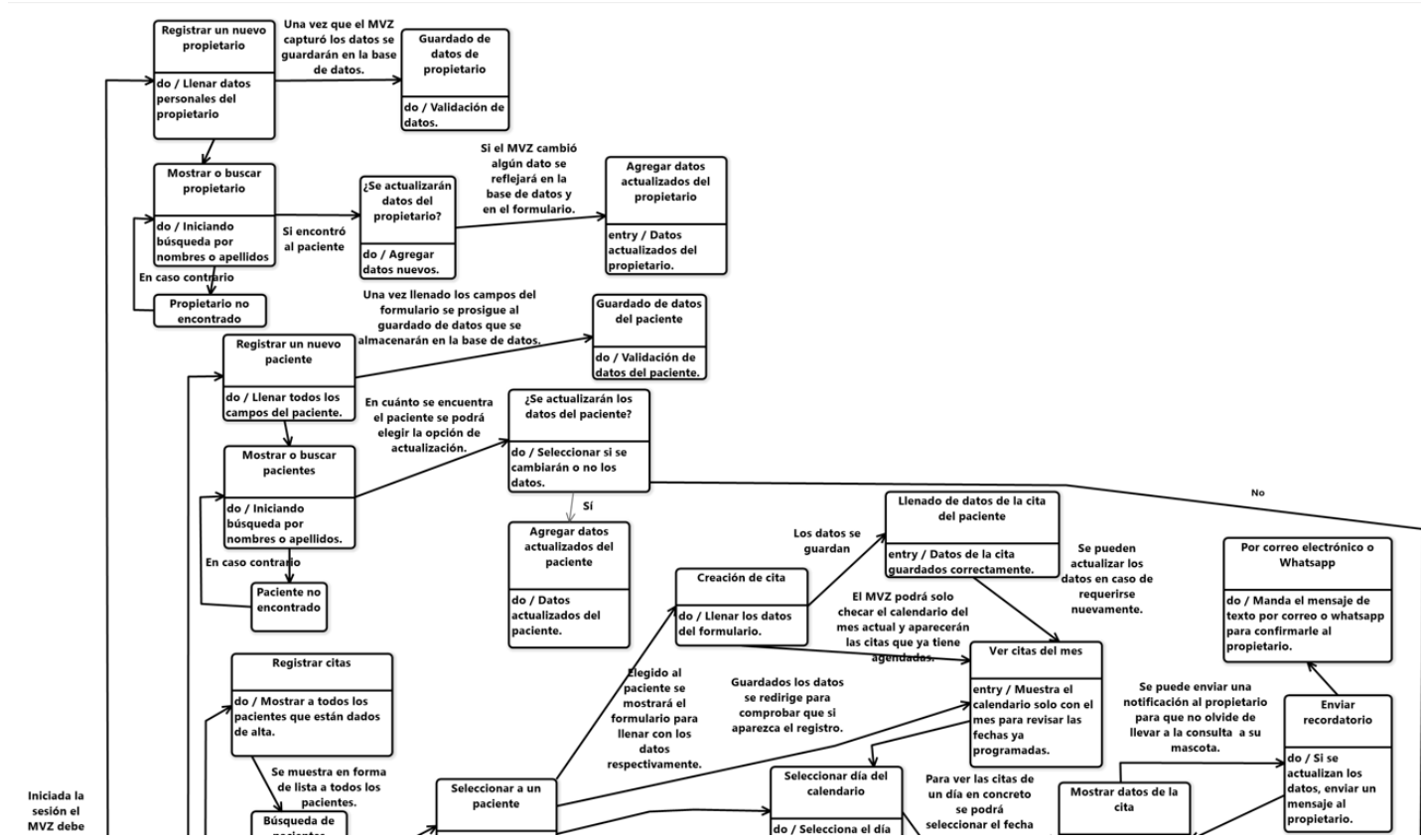
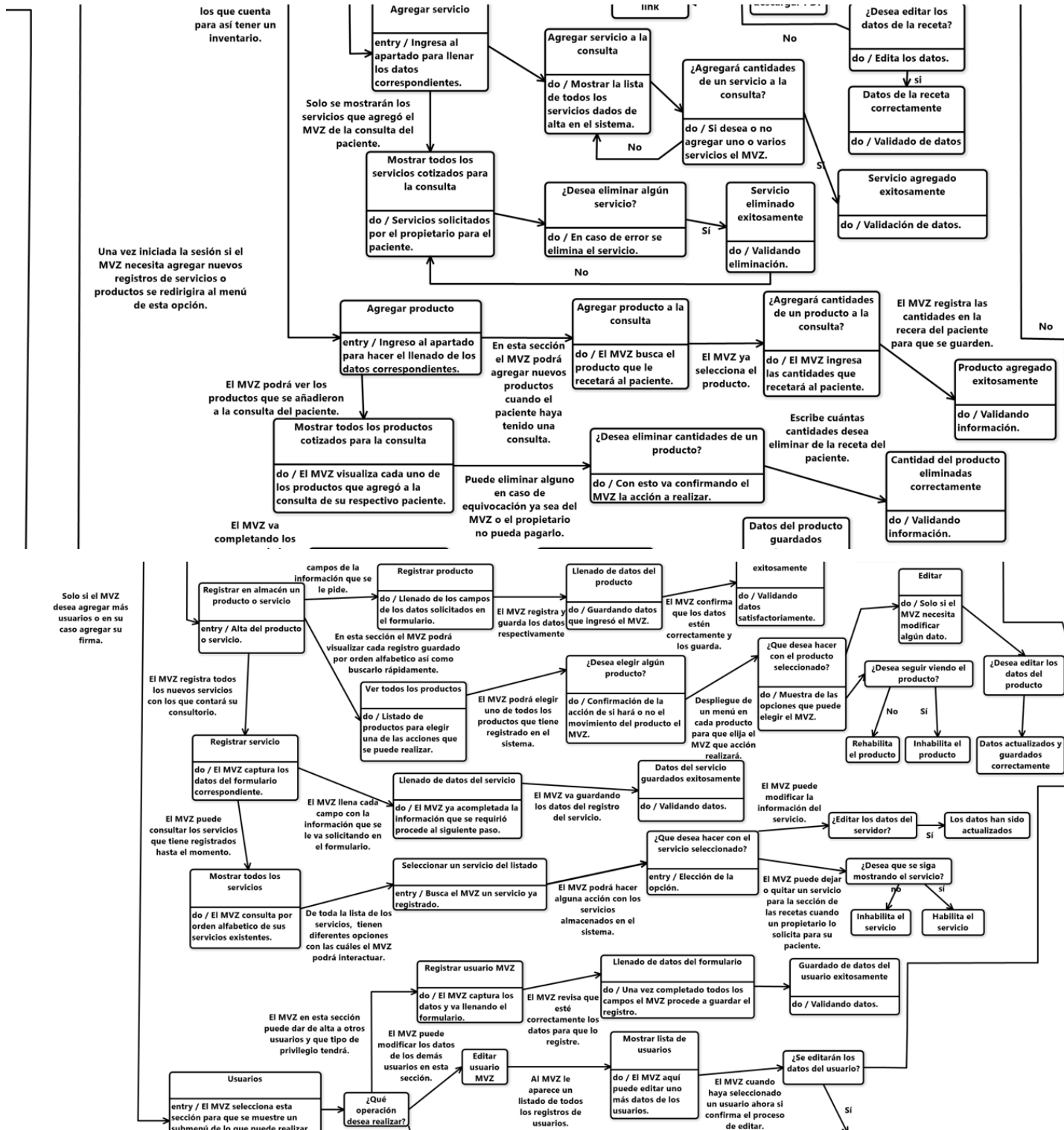


Figura 30. Diagrama de paquetes de VetApp.

Diagrama de estados.

Diagrama que indica todo el proceso en el sistema, así como el orden en el que funciona como se observa en la figura 31.





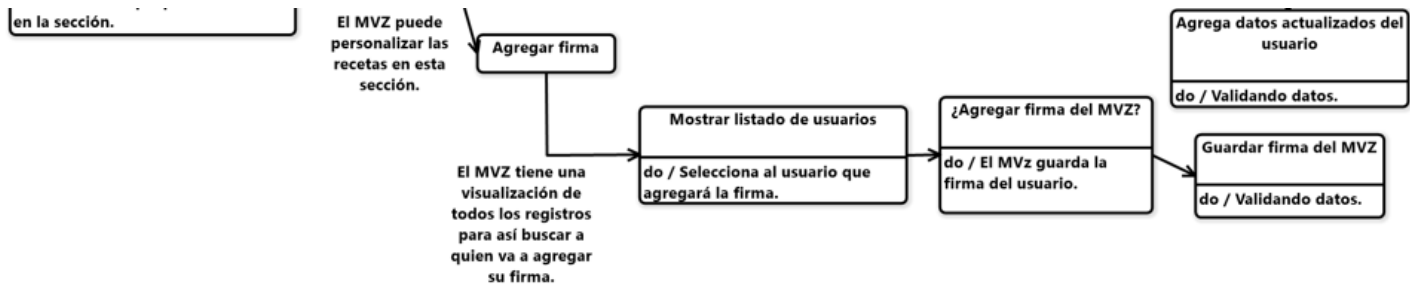
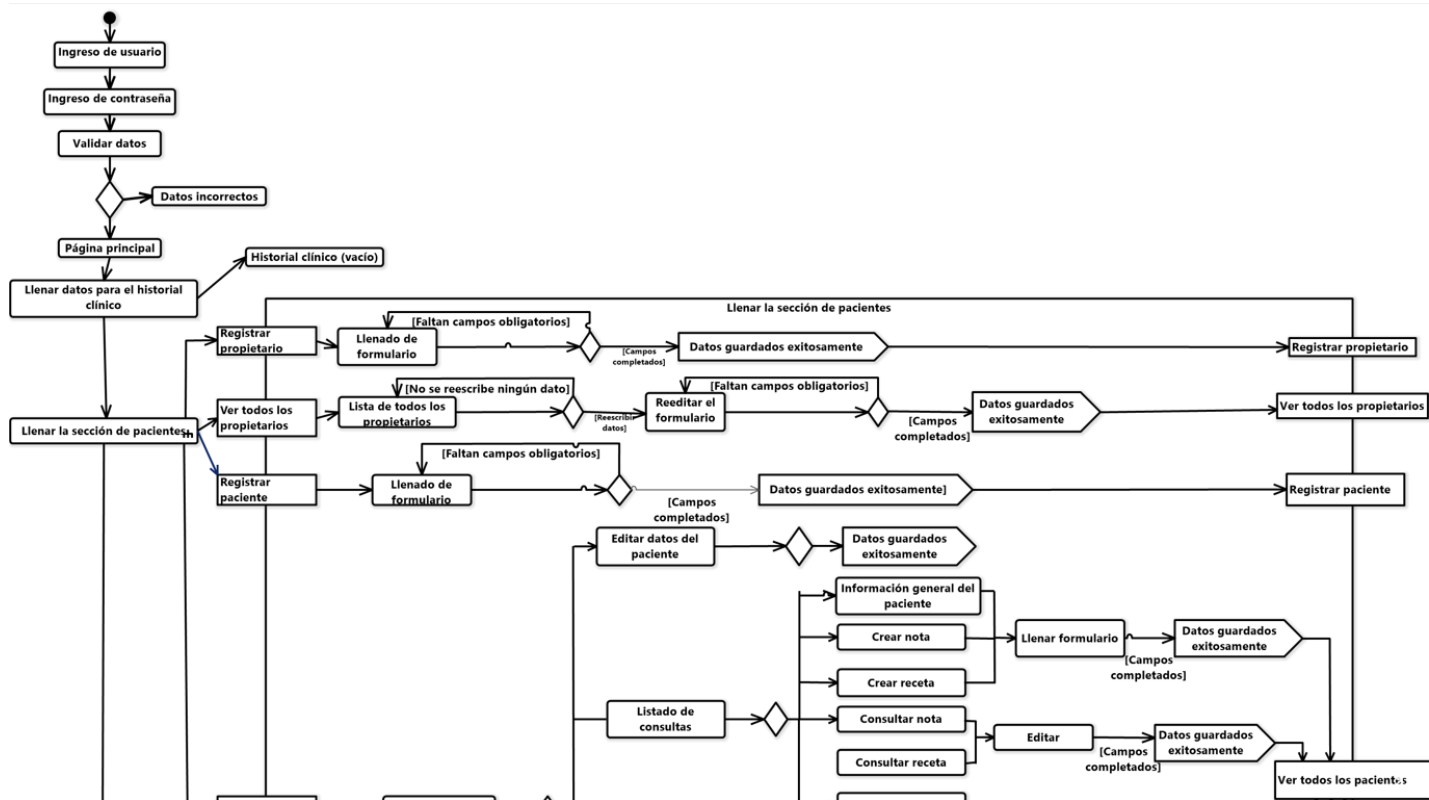


Figura 31. Diagrama de estados de Vetapp.

Diagrama de actividades.

Diagrama que indica el funcionamiento de cada sección del sistema como se observa en la figura 32.



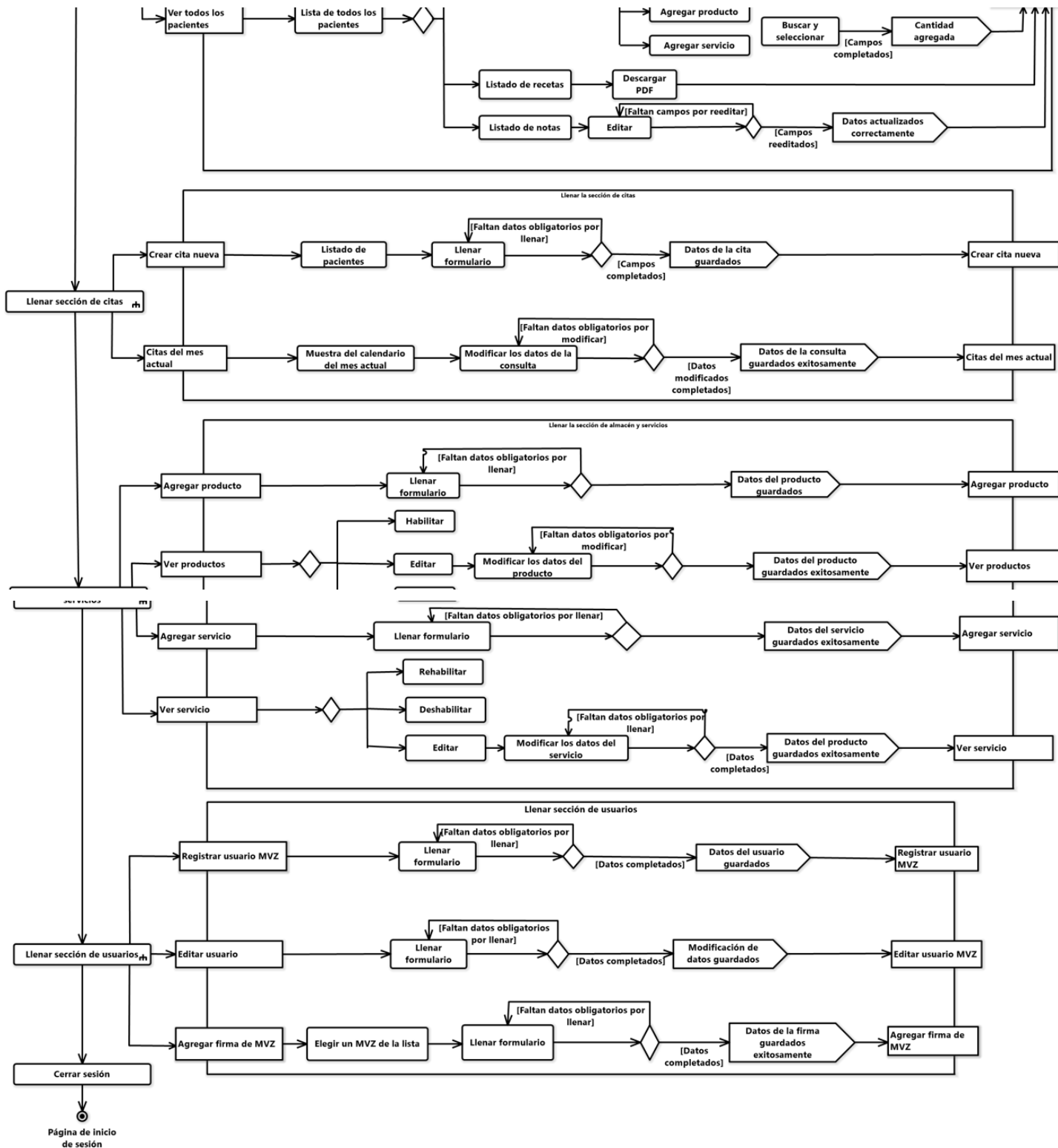
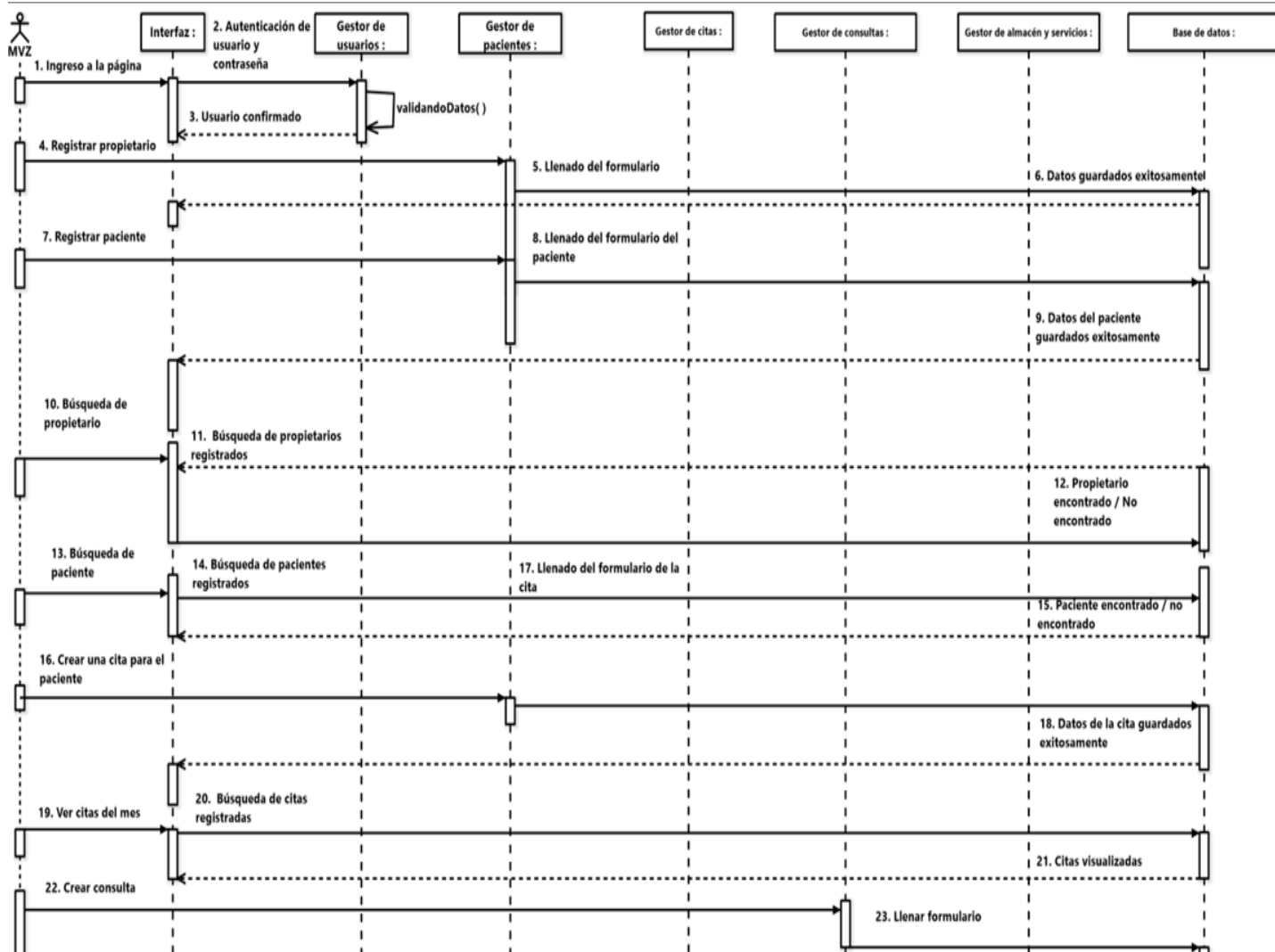


Figura 32. Diagrama de actividades de VetApp.

Diagrama de secuencia.

Diagrama que indica las transacciones del sistema entre sus mensajes y el cual representa su comportamiento como se observa en la figura 33.



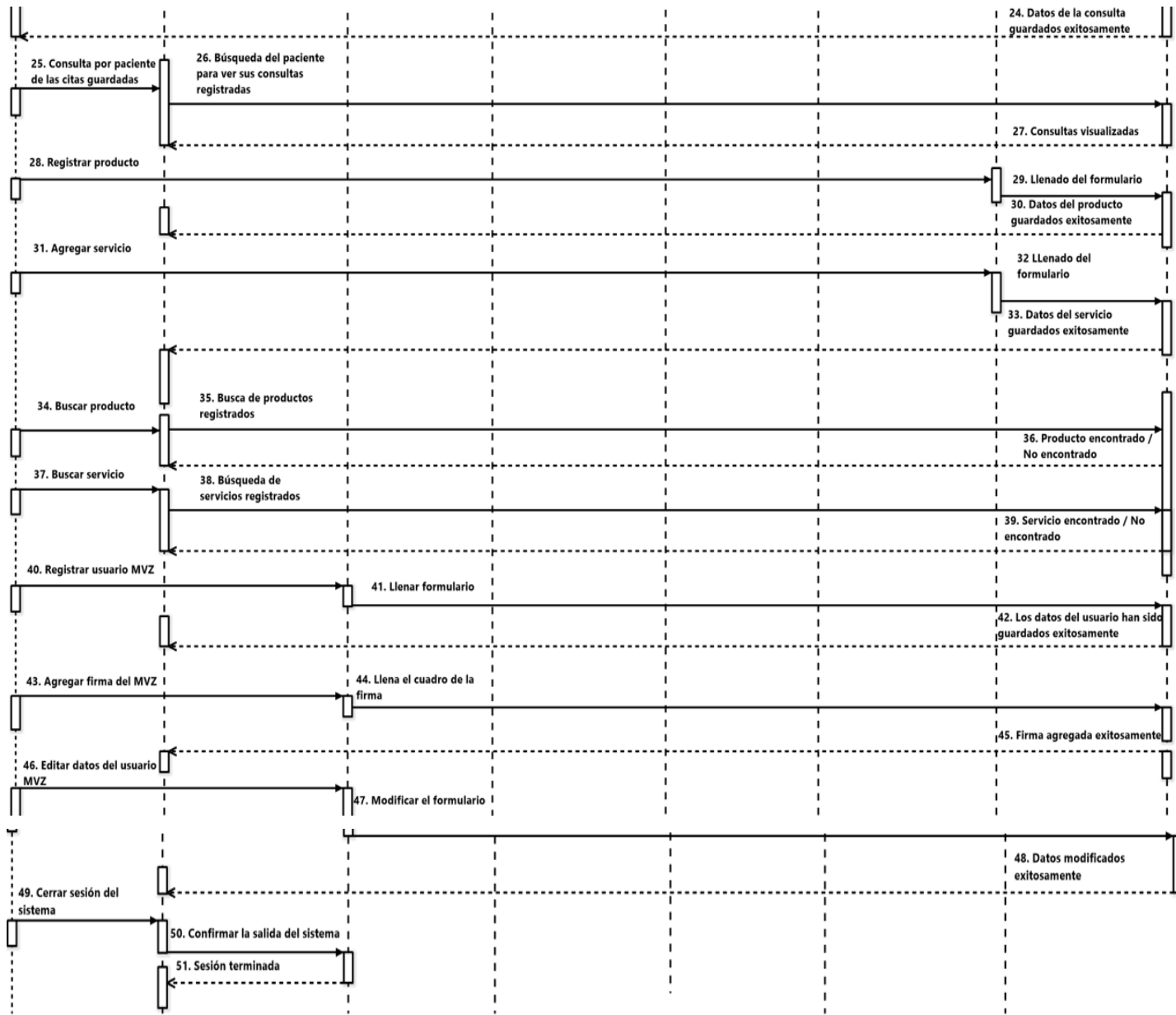
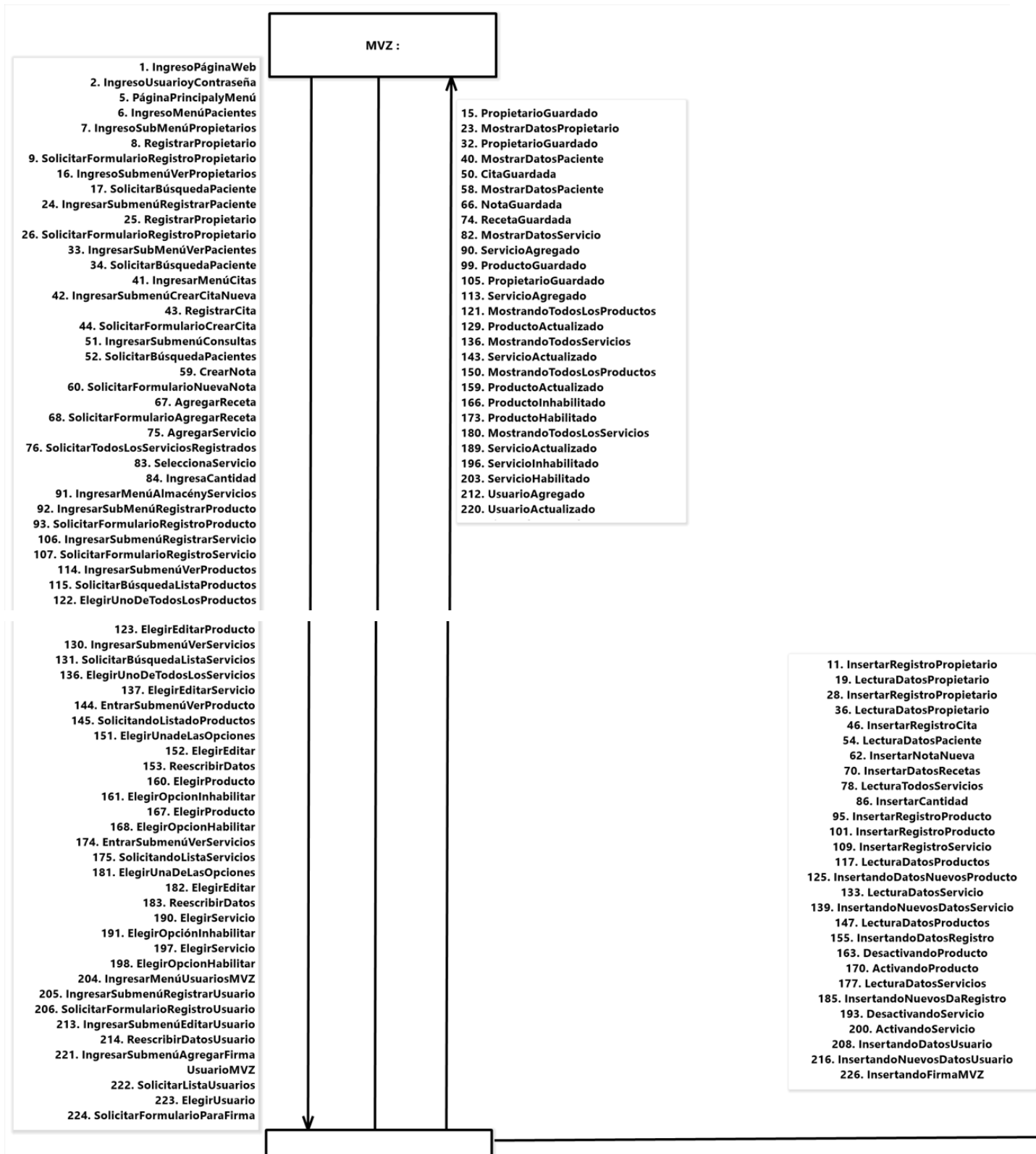
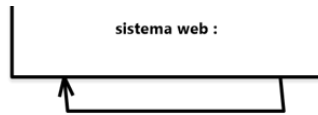


Figura 33. Diagrama de secuencia de VetApp.

Diagrama de comunicación.

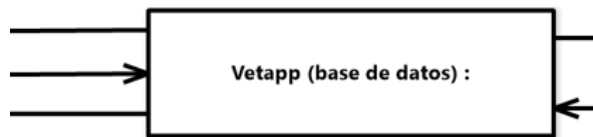
Este diagrama modela las interacciones que existe entre los objetos como se observa en la figura 34.





- 3. ValidarUsuarioContrasena
- 10. ConfirmandoDatosPropietario
- 18. ValidandoNombreApellido
- 27. ConfirmandoDatosPropietario
- 35. ValidandoNombreApellido
- 45. ConfirmandoDatosCita
- 53. ValidandoNombreApellido
- 61. ConfirmandoDatosNota
- 69. ConfirmandoDatosReceta
- 77. ConfirmandoListaDeServicios
- 85. ValidandoCantidad
- 94. ConfirmandoDatosProducto
- 100. ConfirmandoDatosProducto
- 108. ConfirmandoDatosServicio
- 116. ValidandoDatosBúsquedaProducto
- 124. ValidandoNuevosDatosProducto
- 132. ValidandoBúsquedaServicio
- 138. ValidandoNuevosDatosServicio
- 146. ConfirmandoListaProductos
- 154. ConfirmandoNuevosDatos
- 162. ConfirmandoValidacióndeInhabilitaciónProducto
- 169. ConfirmandoValidacióndeRehabilitarProducto
- 176. ConfirmandoListaServicios
- 184. ConfirmandoNuevosDatos
- 192. ConfirmandoValidacióndeInhabilitaciónServicio
- 199. ConfirmandoValidacióndeRehabilitarServicio
- 207. IngresandoConfirmandoDatosUsuario
- 215. ConfirmandoNuevosDatosUsuario
- 225. ConfirmandoFirma

- 14. EnviarMensajeConfirmacionPropietario
- 22. EnviandoDatosPaciente
- 31. EnviarMensajeConfirmacionPropietario
- 39. EnviandoDatosPropietario
- 49. EnviarMensajeConfirmacionCita
- 57. EnviandoDatosPaciente
- 65. EnviarMensajeConfirmacionNota
- 73. EnviarMensajeConfirmaciónReceta
- 81. EnviandoDatosServicios
- 89. EnviarMensajeConfirmaciónServicio
- 98. EnviarMensajeConfirmacionProducto
- 104. EnviarMensajeConfirmacionProducto
- 112. EnviandoMensajeConfirmaciónServicio
- 120. EnviandoDatosMVZ
- 128. EnviandoMensajeConfirmaciónProducto
- 135. EnviandoConfirmaciónMVZ
- 142. EnviandoConfirmaciónServicioMVZ
- 149. EnviandoListaDeDatosProductos
- 158. EnviandoMensajeProducto
- 165. EnviandoMensajeDesactivación
- 172. EnviandoMensajeReactivación
- 179. EnviandoListaDeDatosServicios
- 188. EnviandoMensajeServicio
- 195. EnviandoMensajeDesactivación
- 202. EnviandoMensajeReactivación
- 211. EnviandoMensajeConfirmaciónAgregarUsuario
- 219. EnviandoMensajeConfirmaciónUsuario
- 229. EnviandoMensajeConfirmaciónFirma



- 12. ValidandoDatosPropietario
 - 13. PropietarioAgregado
 - 20. ValidandoDatosExistentesPropietario
 - 21. BúsquedaPacienteSeleccionado
 - 29. ValidandoDatosPropietario
 - 30. PropietarioAgregado
 - 37. ValidandoDatosExistentesPropietario
 - 38. BúsquedaPacienteSeleccionado
 - 47. ValidandoDatosCita
 - 48. CitaAgregada
 - 55. ValidandoDatosExistentesPaciente
 - 56. BúsquedaPacienteSeleccionado
 - 63. ValidandoDatosNota
 - 64. NotaAgregada
 - 71. ValidandoDatosReceta
 - 72. RecetaAgregada
 - 79. EnviandoDatosServicios
 - 80. ValidandoDatosExistentesServicios
 - 87. ValidandoCantidadServicios
 - 88. ServicioSolicitado
 - 96. ValidandoDatosProducto
 - 97. ProductoAgregado
 - 102. ValidandoDatosPropietario
 - 103. ProductoAgregado
 - 110. ValidandoDatosServicio
 - 111. ServicioAgregado
 - 118. ValidandoDatosExistentesProductos
 - 119. BúsquedaProductoSeleccionado
 - 126. ValidandoNuevosDatosProducto
 - 127. ActualizandoDatosProducto
 - 134. BúsquedaServicioSeleccionado
 - 140. ValidandoNuevosDatosServicio
 - 141. ActualizandoDatosServicio
 - 148. ValidandoProductosRegistrados
 - 156. ValidandoDatosProducto
 - 157. ProductoActualizado
 - 164. ProductoDesactivandose
 - 171. ProductoReactivándose
-
- 178. ValidandoServiciosRegistrados
 - 186. ValidandoDatosServicio
 - 187. ServicioActualizado
 - 194. ProductoDesactivandose
 - 201. ServicioReactivándose
 - 209. ValidandoDatosUsuario
 - 210. UsuarioAgregado
 - 217. ValidandoNuevosDatosUsuario
 - 218. ActualizandoDatosUsuario
 - 227. ValidandoFirma
 - 228. FirmaAgregada

Figura 34. Diagrama de comunicación de VetApp.

Diagrama de componentes.

Diagrama que representa el sistema dividido en componentes y muestra sus dependencias como se observa en la figura 35.

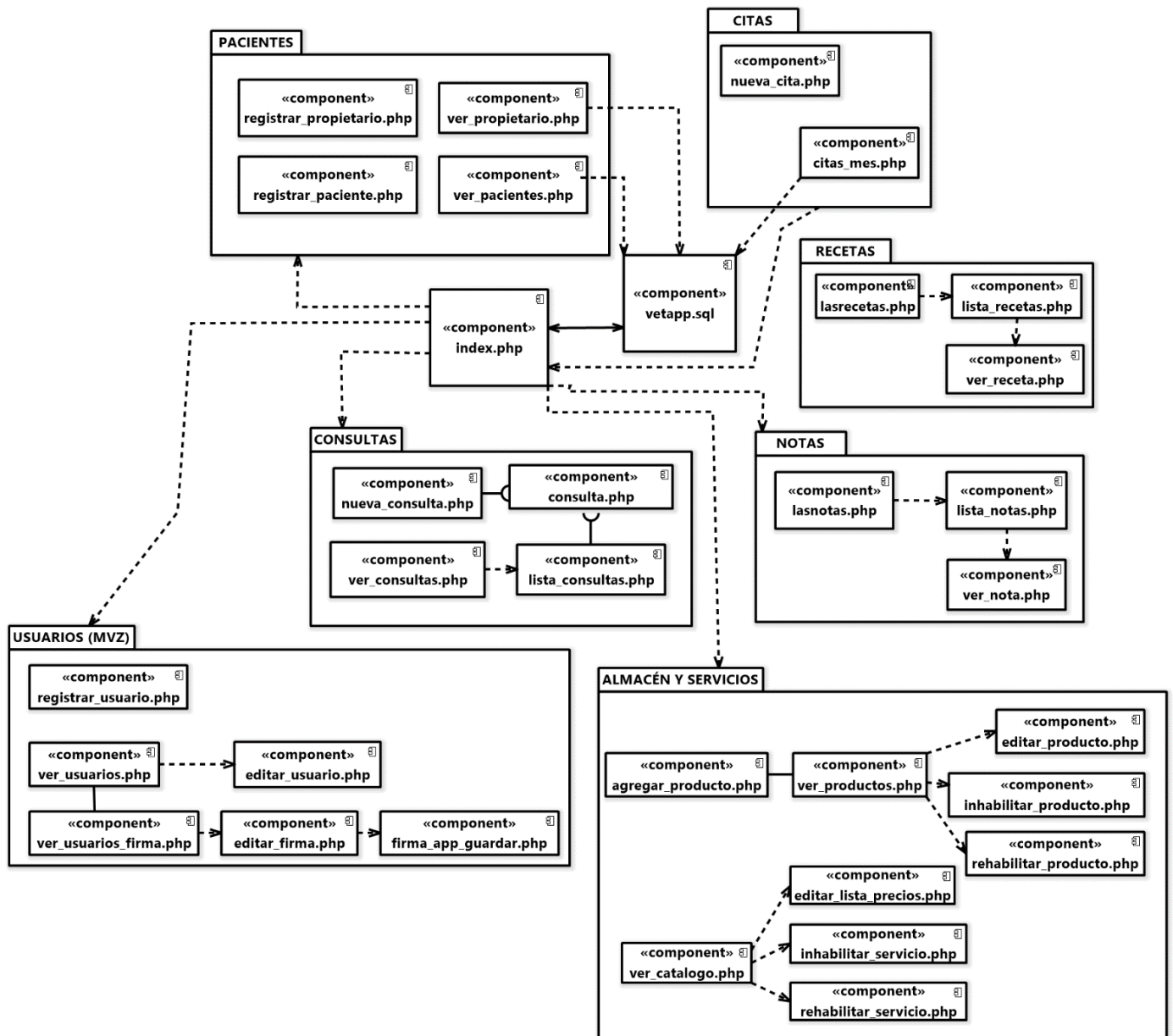


Figura 35. Diagrama de componentes de VetApp.

Diagrama de despliegue.

Diagrama que visualiza los nodos, los enlaces de comunicación y su colocación de archivos de software como se observa en la figura 36.

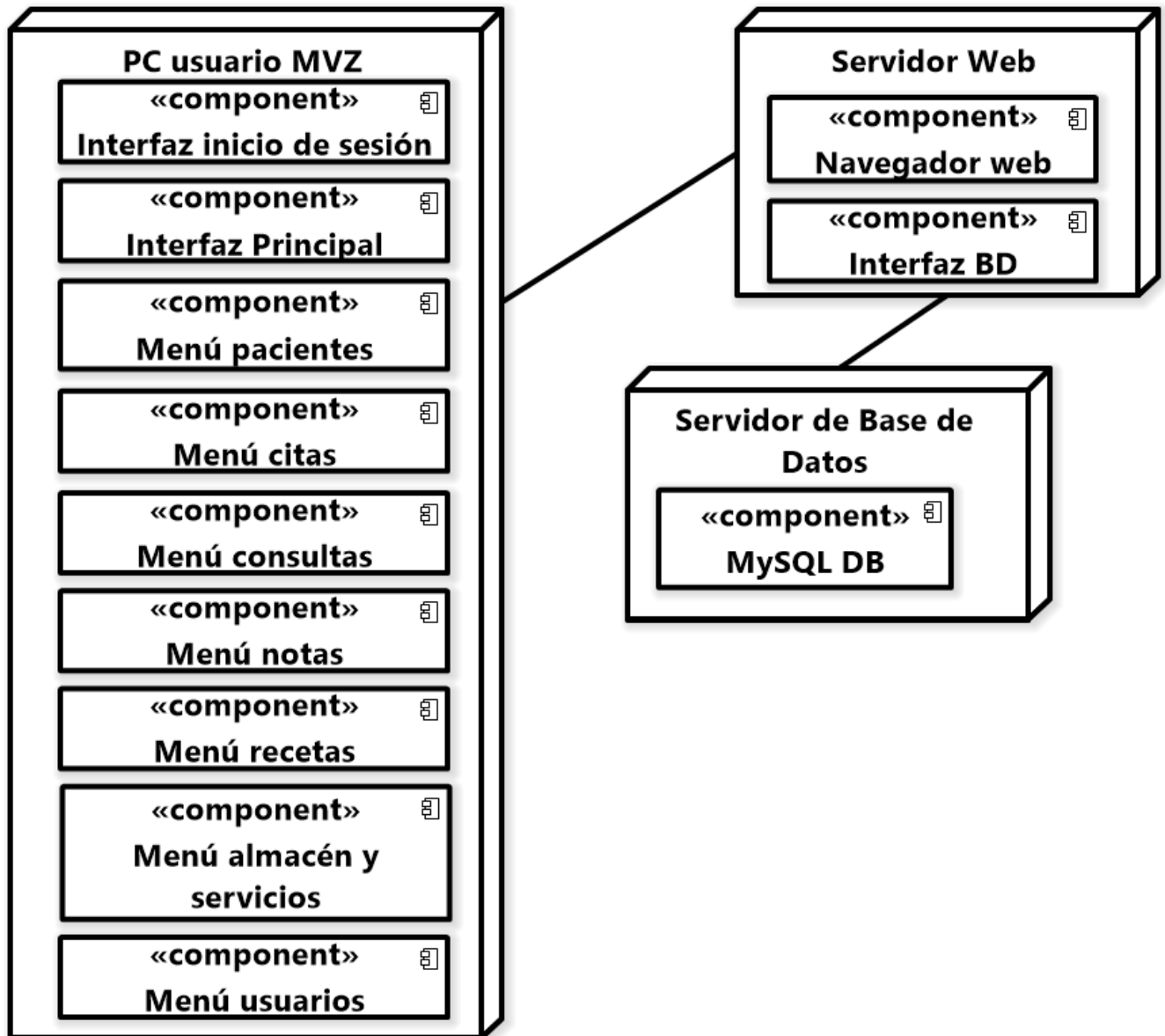


Figura 36. Diagrama de despliegue de VetApp.

3.1.2 Realización de diagrama entidad – relación.

Para realizar la base de datos nos enfocamos primero a obtener las entidades, tipo de datos serán, sus campos, cómo estarán relacionadas con sus respectivas claves primarias y foráneas como se observa en la figura 37.

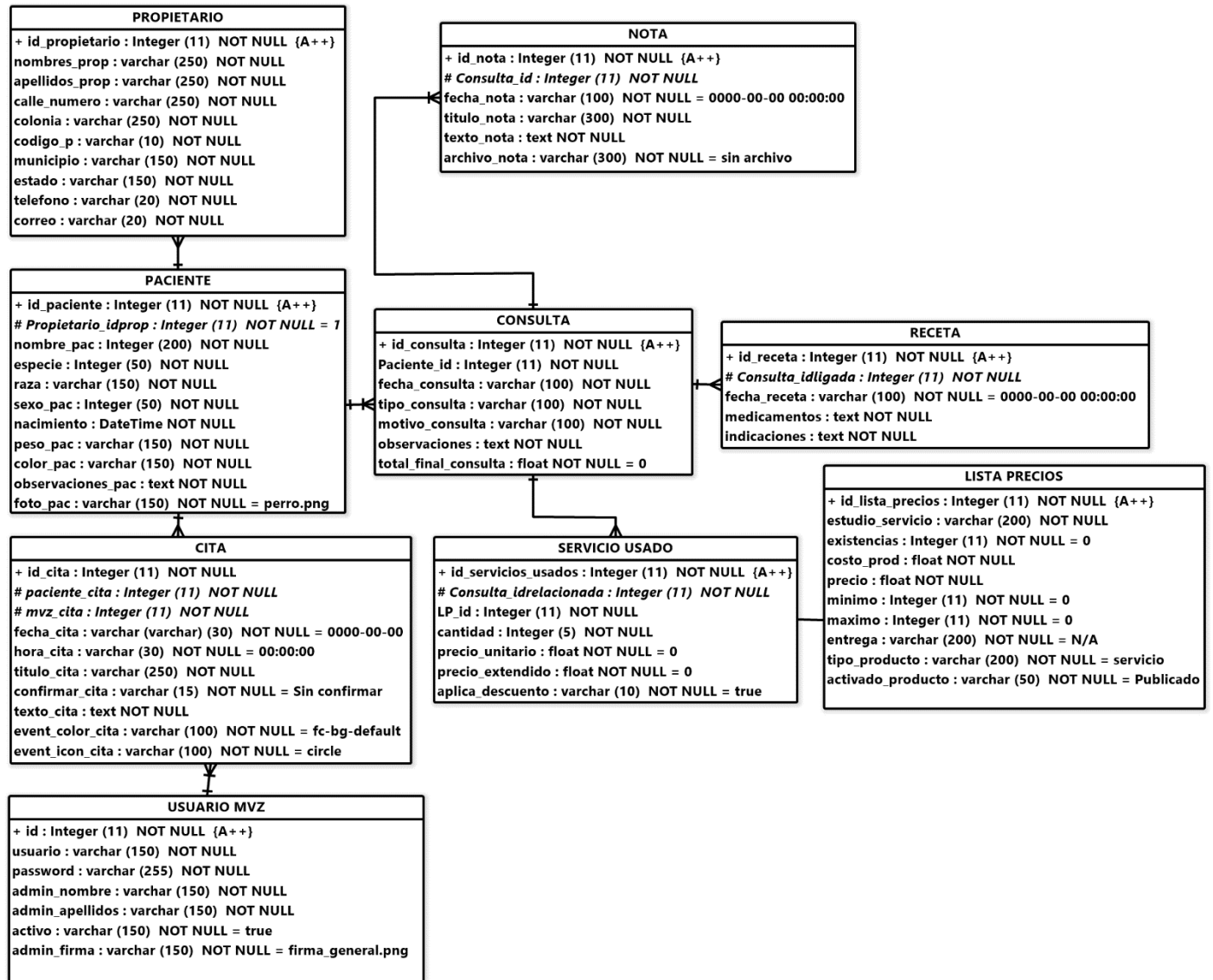


Figura 37. Diagrama de entidad relación de VetApp.

3.1.3 Diseño de base de datos en MySQL.

A continuación, en este capítulo se abordará acerca de las tablas que se crearon en la base de datos MySQL en el software de HeidiSQL.

Como se puede observar en la figura 38 todas las tablas que conforman a la base de datos de vetapp.

Tabla	Tamaño
vetapp_demo	368.0 KiB
citas	48.0 KiB
consulta	32.0 KiB
lista_precios	16.0 KiB
nota	80.0 KiB
paciente	32.0 KiB
propietario	16.0 KiB
receta	64.0 KiB
servicios_usados	48.0 KiB
usuarios_admin	16.0 KiB
validar_pago	16.0 KiB

Figura 38. Tablas que conforman a la base de datos Vetapp.

La tabla para almacenar los datos para las citas está estructurada como se muestra en la figura 39.

Host: 127.0.0.1 Base de datos: vetapp_demo Tabla: citas Datos Consulta

Básico Opciones Índices Llaves foráneas Particiones Código CREATE Código ALTER

Nombre: citas

Comentario:

Columnas: + Agregar × Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id_cita	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	paciente_cita	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
3	mvz_cita	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
4	fecha_cita	VARCHAR	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0000-00-00'
5	hora_cita	VARCHAR	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'00:00:00'
6	titulo_cita	VARCHAR	250	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
7	confirmar_cita	VARCHAR	15	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'Sin confirmar'
8	texto_cita	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
9	event_color_cita	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'fc-bg-default'
10	event_icon_cita	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'circle'

Figura 39. Tabla citas.

Para el guardado de datos de las consultas creadas por cada paciente serán almacenadas como se muestra en la figura 40.

Host: 127.0.0.1 Base de datos: vetapp_demo Tabla: consulta Datos Consulta

Básico Opciones Índices Llaves foráneas Particiones Código CREATE Código ALTER

Nombre: consulta

Comentario:

Columnas: + Agregar x Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id_consulta	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	Paciente_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
3	fecha_consulta	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
4	tipo_consulta	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
5	motivo_consulta	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
6	observaciones	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
7	total_final_consulta	FLOAT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

Figura 40. Tabla consulta.

La tabla para guardar los datos de los precios de productos y servicios está estructurada como se muestra en la figura 41.

Host: 127.0.0.1 Base de datos: vetapp_demo Tabla: lista_precios Datos Consulta

Básico Opciones Índices Llaves foráneas Particiones Código CREATE Código ALTER

Nombre: lista_precios

Comentario:

Columnas: + Agregar x Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id_lista_precios	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	estudio_servicio	VARCHAR	200	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
3	existencias	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
4	costo_prod	FLOAT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
5	precio	FLOAT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
6	minimo	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
7	maximo	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
8	entrega	VARCHAR	200	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'N/A'
9	tipo_producto	VARCHAR	200	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'servicio'
10	activado_producto	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'Publicado'

Figura 41. Tabla lista_precios.

Para las notas que cree el veterinario serán guardadas en la siguiente tabla como se puede observar en la figura 42.

Host: 127.0.0.1 Base de datos: vetapp_demo Tabla: nota Datos Consulta

Básico Opciones Índices Llaves foráneas Particiones Código CREATE Código ALTER

Nombre: nota

Comentario:

Columnas: + Agregar x Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id_nota	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	Consulta_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
3	fecha_nota	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0000-00-00 00:00:00'
4	titulo_nota	VARCHAR	300	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
5	texto_nota	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
6	archivo_nota	VARCHAR	300	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'Sin archivo'

Figura 42. Tabla nota.

El veterinario cuando registre un nuevo paciente, sus datos serán guardados en la tabla que se puede observar en la figura 43.

Host: 127.0.0.1 Base de datos: vetapp_demo Tabla: paciente Datos Consulta

Básico Opciones Índices Llaves foráneas Particiones Código CREATE Código ALTER

Nombre: paciente

Comentario:

Columnas: + Agregar x Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id_paciente	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	Propietario_idprop	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'
3	nombre_pac	VARCHAR	200	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
4	especie	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
5	raza	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
6	sexo_pac	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
7	nacimiento	DATE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
8	peso_pac	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
9	color_pac	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
10	observaciones_pac	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
11	foto_pac	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'perro.png'

Figura 43. Tabla paciente.

El veterinario cuando registre al propietario de su paciente, sus datos serán guardados en la tabla que se puede observar en la figura 44.

Host: 127.0.0.1 Base de datos: vetapp_demo Tabla: propietario Datos Consulta

Básico Opciones Índices Llaves foráneas Particiones Código CREATE Código ALTER

Nombre: propietario

Comentario:

Columnas: + Agregar x Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id_propietario	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	nombres_prop	VARCHAR	250	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
3	apellidos_prop	VARCHAR	250	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
4	calle_numero	VARCHAR	250	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
5	colonia	VARCHAR	250	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
6	codigo_p	VARCHAR	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
7	municipio	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
8	estado	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
9	telefono	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
10	correo	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado

Figura 44. Tabla propietario.

El médico veterinario cuando registre una nueva receta para el paciente esta será guardada en la tabla que se puede observar en la figura 45.

Host: 127.0.0.1 Base de datos: vetapp_demo Tabla: receta Datos Consulta

Básico Opciones Índices Llaves foráneas Particiones Código CREATE Código ALTER

Nombre: receta

Comentario:

Columnas: + Agregar x Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id_receta	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	Consulta_idligada	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
3	fecha_receta	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0000-00-00 00:00:00'
4	medicamentos	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
5	indicaciones	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado

Figura 45. Tabla receta.

El médico veterinario cuando registre un o más servicios que haya solicitado el propietario para el paciente se registrará en la tabla que se observa en la figura 46.

Host: 127.0.0.1 Base de datos: vetapp_demo Tabla: servicios_usados Datos Consulta

Básico Opciones Índices Llaves foráneas Particiones Código CREATE Código ALTER

Nombre: servicios_usados

Comentario:

Columnas: + Agregar x Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id_servicios_usados	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	Consulta_idrelacionada	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
3	LP_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
4	cantidad	INT	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
5	precio_unitario	FLOAT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
6	precio_extendido	FLOAT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
7	aplica_descuento	VARCHAR	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'true'

Figura 46. Tabla servicios_usados.

Para los usuarios que estén dados de alta para ingresar al sistema, así como registrar a los nuevos se almacenarán sus datos en la tabla que se muestra en la figura 47.

Host: 127.0.0.1 Base de datos: vetapp_demo Tabla: usuarios_admin Datos Consulta

Básico Opciones Índices Llaves foráneas Particiones Código CREATE Código ALTER

Nombre: usuarios_admin

Comentario:

Columnas: + Agregar x Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	usuario	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
3	password	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
4	admin_nombre	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
5	admin_apellidos	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
6	activo	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'true'
7	admin_firma	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'firma_general.png'

Figura 47. Tabla usuarios_admin.

La tabla que se observa en la figura 48, es para validar que el medico veterinario haya realizado el pago correspondiente para poder seguir usando el sistema, es decir, que se les notificará en el sistema para que no se les suspenda el servicio.

Host: 127.0.0.1 Base de datos: vetapp_demo Tabla: validar_pago Datos Consulta

Básico Opciones Índices Llaves foráneas Particiones Código CREATE Código ALTER

Nombre: validar_pago

Comentario:

Columnas: + Agregar x Borrar ▲ Subir ▼ Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id_validar_pago	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	mostrar_sidebar	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'false'

Figura 48. Tabla validar_pago.

3.2 Realización del sistema (codificación).

En este capítulo se abordará acerca de algunos códigos que se realizaron en el sistema web y que a continuación se describe cada uno para saber su función. Para conectarse a la base de datos se utilizó la codificación que se observa en la figura 49.

```
admin_config.php > ...
1 <?php
2 define('RUTA', 'inicio.php');
3
4 date_default_timezone_set('America/Mexico_City');
5
6 $bd_config = array(
7     'basedatos' => 'vetapp_demo',
8     'usuario' => 'root',
9     'pass' => ''
10 );
11 ?>
```

Figura 49. Conexión con la base de datos.

Se utilizan funciones para invocar al momento de consultar la base de datos, que están en un archivo de php.

```
<?php
////////////////////// GENERALES ////////////////////////
function conexion($bd_config){
    try {
        $conexion = new
            PDO('mysql:host=localhost;dbname='.$bd_config['basedatos'],
                $bd_config['usuario'], $bd_config['pass']);
        return $conexion;
    } catch (PDOException $e) {
        return false;
    }
}

function limpiarDatos($datos){
    $datos = trim($datos);
    $datos = stripslashes($datos);
    $datos = htmlspecialchars($datos);
    return $datos;
}

function id_recibido($id){
    return (int)limpiarDatos($id);
}

////////////////////// PAGOS ////////////////////////
function infoPago($conexion){
    $statement = $conexion->prepare("SELECT * FROM validar_pago WHERE
    id_validar_pago = 1 LIMIT 1;");
    $statement->execute(array());
    $resultado = $statement->fetchAll();
    return $resultado;
}
```

```

}
////////////////////// QUITAR $ y , ////////////////////////
function limpiarPrecio($str) {
return str_replace(array("$", ",", "MXN", "pesos", "PESOS", "Pesos", "mxn", " "), "",
$str);
}
function limpiarPorcentajes($str) {
return str_replace(array("%", "desc", "desc.", "descuento", "de descuento", "DESC", "
"), "", $str);
}
////////////////////// SALTO DE LINEA ////////////////////////
function saltoLinea($str) {
return str_replace(array("\r\n", "\r", "\n"), "<br />", $str);
}
function saltoLineaRev($str) {
return str_replace(array("<br />"), "\n", $str);
}
////////////////////// MES en letra ////////////////////////
function mesEquivalente($a_cambiar) {
switch ($a_cambiar) {
case '01':
$a_cambiar = 'Enero';
break;
case '02':
$a_cambiar = 'Febrero';
break;
case '03':
$a_cambiar = 'Marzo';
break;
case '04':
$a_cambiar = 'Abril';
}
}

```

```
        break;
    case '05':
        $a_cambiar = 'Mayo';
        break;
    case '06':
        $a_cambiar = 'Junio';
        break;
    case '07':
        $a_cambiar = 'Julio';
        break;
    case '08':
        $a_cambiar = 'Agosto';
        break;
    case '09':
        $a_cambiar = 'Septiembre';
        break;
    case '10':
        $a_cambiar = 'Octubre';
        break;
    case '11':
        $a_cambiar = 'Noviembre';
        break;
    case '12':
        $a_cambiar = 'Diciembre';
        break;
    default:
        $a_cambiar = $a_cambiar;
        break;
}
return $a_cambiar;
}
```

```

////////////////////////////////// CITAS ////////////////////////////////////
function  citasDelMes($conexion, $ano_mes){
    $statement = $conexion->prepare("SELECT * FROM citas cit INNER JOIN paciente pac
    ON cit.paciente_cita = pac.id_paciente WHERE fecha_cita LIKE :ano_mes ORDER BY
    cit.fecha_cita ASC;");
    $statement->execute(array(':ano_mes' => $ano_mes . '%'));
    $resultado = $statement->fetchAll();
    return $resultado;
}

function  infoCita($conexion, $id_cita){
    $statement = $conexion->prepare("SELECT * FROM citas WHERE id_cita = :id_cita;");
    $statement->execute(array(':id_cita' => $id_cita));
    $resultado = $statement->fetchAll();
    return $resultado;
}

////////////////////////////////// LISTA DE PRECIOS ////////////////////////////////////
function  listaPrecios($conexion){
    $statement = $conexion->prepare("SELECT * FROM lista_precios ORDER BY
    id_lista_precios ASC;");
    $statement->execute(array());
    $resultado = $statement->fetchAll();
    return $resultado;
}

function  listaProductos($conexion){
    $statement = $conexion->prepare("SELECT * FROM lista_precios WHERE
    tipo_producto = 'producto' ORDER BY id_lista_precios ASC;");
    $statement->execute(array());
    $resultado = $statement->fetchAll();
    return $resultado;
}

function  listaServicios($conexion){

```

```

$statement = $conexion->prepare("SELECT * FROM lista_precios WHERE
tipo_producto = 'servicio' ORDER BY id_lista_precios ASC;");
$statement->execute(array());
$resultado = $statement->fetchAll();
return $resultado;
}

function verListaEdicion($conexion, $id_lp){
$statement = $conexion->prepare("SELECT * FROM lista_precios WHERE
id_lista_precios = :id_lp;");
$statement->execute(array(':id_lp' => $id_lp));
$resultado = $statement->fetchAll();
return $resultado;
}

////////// ESTUDIOS / SERVICIOS //////////

function cotizadosConsulta($conexion, $id_consulta){
$statement = $conexion->prepare("SELECT * FROM servicios_usados servi INNER
JOIN lista_precios lip ON servi.LP_id = lip.id_lista_precios AND
servi.Consulta_idrelacionada = :id_consulta;");
$statement->execute(array(':id_consulta' => $id_consulta));
$resultado = $statement->fetchAll();
return $resultado;
}

////////// PROPIETARIOS //////////

function listaPropietarios($conexion){
$statement = $conexion->prepare("SELECT * FROM propietario ORDER BY
nombres_prop ASC;");
$statement->execute(array());
$resultado = $statement->fetchAll();
return $resultado;
}

function editarPropietario($conexion, $id_propietario){

```

```

$statement = $conexion->prepare("SELECT * FROM propietario WHERE id_propietario
= :id_propietario;");
$statement->execute(array(':id_propietario' => $id_propietario));
$resultado = $statement->fetchAll();
return $resultado;
}

////////////////////// PACIENTES ////////////////////////

function listaPacientes($conexion){
$statement = $conexion->prepare("SELECT * FROM paciente pac INNER JOIN
propietario prop ON pac.Propietario_idprop = prop.id_propietario");
$statement->execute(array());
$resultado = $statement->fetchAll();
return $resultado;
}

function editarPaciente($conexion, $id_paciente){
$statement = $conexion->prepare("SELECT * FROM paciente pace INNER JOIN
propietario propi ON pace.Propietario_idprop = propi.id_propietario AND
pace.id_paciente = :id_paciente;");
$statement->execute(array(':id_paciente' => $id_paciente));
$resultado = $statement->fetchAll();
return $resultado;
}

function VerUnPaciente($conexion, $id_paciente){
$statement = $conexion->prepare("SELECT * FROM paciente pac INNER JOIN
propietario prop ON pac.Propietario_idprop = prop.id_propietario AND pac.id_paciente =
:id_paciente;");
$statement->execute(array(':id_paciente' => $id_paciente));
$resultado = $statement->fetchAll();
return $resultado;
}

function totalMachos($conexion){

```

```

$statement = $conexion->prepare("SELECT COUNT(*) FROM paciente WHERE
sexo_pac = 'Macho;");
$statement->execute(array());
$resultado = $statement->fetchColumn();
return $resultado;
}

function totalMachosCas($conexion){
$statement = $conexion->prepare("SELECT COUNT(*) FROM paciente WHERE
sexo_pac = 'Macho Castrado;");
$statement->execute(array());
$resultado = $statement->fetchColumn();
return $resultado;
}

function totalHembras($conexion){
$statement = $conexion->prepare("SELECT COUNT(*) FROM paciente WHERE
sexo_pac = 'Hembra;");
$statement->execute(array());
$resultado = $statement->fetchColumn();
return $resultado;
}

function totalHembrasCas($conexion){
$statement = $conexion->prepare("SELECT COUNT(*) FROM paciente WHERE
sexo_pac = 'Hembra Castrada;");
$statement->execute(array());
$resultado = $statement->fetchColumn();
return $resultado;
}

function totalCaninos($conexion){
$statement = $conexion->prepare("SELECT COUNT(*) FROM paciente WHERE
especie = 'Canino;");
$statement->execute(array());

```



```

$resultado = $statement->fetchColumn();
return $resultado;
}

function totalFelinos($conexion){
    $statement = $conexion->prepare("SELECT COUNT(*) FROM paciente WHERE
    especie = 'Felino;");
    $statement->execute(array());
    $resultado = $statement->fetchColumn();
    return $resultado;
}

//////////////////// CONSULTAS //////////////////////

function traerConsulta($conexion, $paciente, $fecha){//ID de la consulta
    $statement = $conexion->prepare("SELECT id_consulta FROM consulta WHERE
    Paciente_id = :paciente AND fecha_consulta = :fecha;");
    $statement->execute(array(':paciente' => $paciente, ':fecha' => $fecha));
    $resultado = $statement->fetchColumn();
    return $resultado;
}

function infoConsulta($conexion, $id_consulta){
    $statement = $conexion->prepare("SELECT * FROM consulta con INNER JOIN paciente
    pac INNER JOIN propietario prop ON con.Paciente_id = pac.id_paciente AND
    pac.Propietario_idprop = prop.id_propietario AND con.id_consulta = :id_consulta;");
    $statement->execute(array(':id_consulta' => $id_consulta));
    $resultado = $statement->fetchAll();
    return $resultado;
}

function notasDeConsulta($conexion, $id_consulta){
    $statement = $conexion->prepare("SELECT * FROM nota WHERE Consulta_id =
    :id_consulta ORDER BY fecha_nota ASC;");
    $statement->execute(array(':id_consulta' => $id_consulta));
    $resultado = $statement->fetchAll();
}

```

```

return $resultado;
}
function recetasDeConsulta($conexion, $id_consulta){
    $statement = $conexion->prepare("SELECT * FROM receta WHERE Consulta_idligada
    = :id_consulta ORDER BY fecha_receta ASC;");
    $statement->execute(array(':id_consulta' => $id_consulta));
    $resultado = $statement->fetchAll();
    return $resultado;
}
function listadoConsulta($conexion, $id_paciente){
    $statement = $conexion->prepare("SELECT * FROM consulta WHERE Paciente_id =
    :id_paciente ORDER BY fecha_consulta ASC;");
    $statement->execute(array(':id_paciente' => $id_paciente));
    $resultado = $statement->fetchAll();
    return $resultado;
}
////////////////////////////////// NOTAS ////////////////////////////////////
function consultasConNotas($conexion, $id_paciente){
    $statement = $conexion->prepare("SELECT id_consulta FROM consulta WHERE
    Paciente_id = :id_paciente ORDER BY fecha_consulta ASC;");
    $statement->execute(array(':id_paciente' => $id_paciente));
    $resultado = $statement->fetchAll();
    return $resultado;
}
function listadoNotas($conexion, $id_consulta){
    $statement = $conexion->prepare("SELECT * FROM nota WHERE Consulta_id =
    :id_consulta ORDER BY fecha_nota ASC;");
    $statement->execute(array(':id_consulta' => $id_consulta));
    $resultado = $statement->fetchAll();
    return $resultado;
}

```

```

function infoNota($conexion, $id_nota){
    $statement = $conexion->prepare("SELECT * FROM nota notita INNER JOIN consulta
con INNER JOIN paciente pac INNER JOIN propietario prop ON notita.Consulta_id =
con.id_consulta AND con.Paciente_id = pac.id_paciente AND pac.Propietario_idprop =
prop.id_propietario AND notita.id_nota = :id_nota;");
    $statement->execute(array(':id_nota' => $id_nota));
    $resultado = $statement->fetchAll();
    return $resultado;
}

////////////////////// RECETAS ////////////////////////

function consultasConRecetas($conexion, $id_paciente){
    $statement = $conexion->prepare("SELECT id_consulta FROM consulta WHERE
Paciente_id = :id_paciente ORDER BY fecha_consulta ASC;");
    $statement->execute(array(':id_paciente' => $id_paciente));
    $resultado = $statement->fetchAll();
    return $resultado;
}

function listadoRecetas($conexion, $id_consulta){
    $statement = $conexion->prepare("SELECT * FROM receta WHERE Consulta_idligada
= :id_consulta ORDER BY fecha_receta ASC;");
    $statement->execute(array(':id_consulta' => $id_consulta));
    $resultado = $statement->fetchAll();
    return $resultado;
}

function infoReceta($conexion, $id_receta){
    $statement = $conexion->prepare("SELECT * FROM receta recetita INNER JOIN
consulta con INNER JOIN paciente pac INNER JOIN propietario prop ON
recetita.Consulta_idligada = con.id_consulta AND con.Paciente_id = pac.id_paciente
AND pac.Propietario_idprop = prop.id_propietario AND recetita.id_receta = :id_receta;");
    $statement->execute(array(':id_receta' => $id_receta));
    $resultado = $statement->fetchAll();
}

```

```

return $resultado;
}

////////// USUARIOS //////////

function listaUsuarios($conexion){
    $statement = $conexion->prepare("SELECT * FROM usuarios_admin WHERE id != 1;");
    $statement->execute(array());
    $resultado = $statement->fetchAll();
    return $resultado;
}

function editarUsuario($conexion, $id){
    $statement = $conexion->prepare("SELECT * FROM usuarios_admin WHERE id = :id;");
    $statement->execute(array(':id' => $id));
    $resultado = $statement->fetchAll();
    return $resultado;
}

////////// COMPROBAR SESIÓN //////////

function comprobarSession(){

    if (!isset($_SESSION['usuario'])) {

        header('Location: login.php');

    }

}

?>

```

Código para validar el usuario y contraseña del login.

```

<?php session_start();
if (isset($_SESSION['usuario'])) {
    header('Location: inicio.php');
}

```

```

require 'admin_config.php';
require 'admin_funciones.php';
$conexion = conexion($bd_config);
if (!$conexion) {
    header('Location: error.php');
}
//$fecha_sistema = date("Y-m-d H:i:s");
$errores = "";
//Comprobar si han enviado los datos (vía POST)
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    //Tomar las variables
    $usuario = strtolower($_POST['usuario']);
    $password = $_POST['password'];
    $activo = 'true';
    $errores = "";
    //Ahora hashear
    $password = hash('sha512', $password);
    $statement = $conexion->prepare('SELECT * FROM usuarios_admin WHERE
    usuario = :usuario AND password = :password AND activo = :activo');
    $statement->execute(array(':usuario' => $usuario, ':password' => $password,
    ':activo' => $activo));
    $resultado = $statement->fetchAll();//Con JS, no olvidar fetchALL
    if ($resultado != false) {//Significa que los datos fueron correctos. IMPORTANTE:
    en JS sólo un =, no dos ==
    $enviarLocalS = $resultado[0];
    //Guardar LocalStorage
    echo "<script type='text/javascript'>localStorage.setItem('Usuario', "
    .$enviarLocalS['usuario'] . "");
    localStorage.setItem('Fecha', new Date());</script>";
    $_SESSION['usuario']= $usuario;//Inicia la sesión con ese usuario y ya lo envía al
    inicio

```

```
//Redireccion
    echo "<script type='text/javascript'>window.location='inicio.php';</script>";
} else {
    $errores .= '<strong>Datos incorrectos</strong>';
}
}
require 'views/login.view.php';
?>
```

Código para cerrar las sesiones en caso de que el usuario MVZ no lo haya hecho correctamente.

```
<?php session_start();
//Para cerrar la sesión si o si
session_destroy();
$_SESSION = array(); //Para dejar la sesión en 0
header('Location: login.php')
?>
```

Código para cuando el usuario MVZ inicia sesión y es donde se muestran las gráficas, pero por parte del servidor y la base de datos.

```
<?php session_start();
require 'admin_config.php';
require 'admin_funciones.php';
$conexion = conexion($bd_config);
if(!$conexion){
    header('Location: admin_error.php');
}
$fecha_sistema = date('Y-m-d H:i:s');
//validar si hay existencias 0
$stmt = $conexion->prepare("SELECT COUNT(*) FROM lista_precios WHERE
existencias <=9 AND tipo_producto ='producto';");
```

```

$statement->execute(array());
$existencias = $statement->fetchColumn();
$machos = totalMachos($conexion);
$machos_castrados = totalMachosCas($conexion);
$hembras = totalHembras($conexion);
$hembras_castradas = totalHembrasCas($conexion);
$caninos = totalCaninos($conexion);
$felinos = totalFelinos($conexion);
//Revisar si se tiene una sesión abierta si no lo regresa al login/registro
if(isset($_SESSION['usuario'])){
    require 'views/inicio.view.php';
}
else{
    header('Location: index.php');
}
?>

```

Para no repetir códigos en todos los archivos se separaron los encabezados y pie de página y solo se mandaron a llamar con una instrucción llamada include.

Este es el código que se utiliza para el head.

```

<!DOCTYPE html>
<html>
<head>
    <!-- Basic Page Info -->
    <meta charset="utf-8">
    <title>Vetapp de Lupware &#174;</title>
    <!-- Site favicon -->
    <link rel="apple-touch-icon" sizes="180x180" href="images/loginvet.png">
    <link rel="icon" type="image/png" sizes="32x32" href="images/login-vet
sidebar.png">

```

```

<link rel="icon" type="image/png" sizes="16x16" href="images/login-vet-sidebar.png">
<!-- Mobile Specific Metas -->
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
<!-- Información general-->
<meta content="Vetapp Lupware " name="author">
<meta content="Software veterinario, perros, gatos, veterinario, programa, vet"
name="keywords">
<meta content="Software de expedientes veterinarios 'Vetapp de Lupware'"
name="description">
<!-- Google Font -->
<link
href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700;800&
display=swap" rel="stylesheet">
<!-- CSS -->
<link rel="stylesheet" type="text/css" href="vendors/styles/core.css">
<link rel="stylesheet" type="text/css" href="vendors/styles/icon-font.min.css">
<link rel="stylesheet" type="text/css"
href="src/plugins/datatables/css/dataTables.bootstrap4.min.css">
<link rel="stylesheet" type="text/css"
href="src/plugins/datatables/css/responsive.bootstrap4.min.css">
<link rel="stylesheet" type="text/css" href="vendors/styles/style.css">
</head>

```

El archivo footer contiene el siguiente código:

```

<div class="footer-wrap pd-20 mb-20 card-box">
  Desarrollo por <a href="https://lupware.com" target="_blank">Lupware</a>
</div>

```

Y el código que se presenta a continuación es para parte del menú con el que interactúa el usuario MVZ, es el archivo al que se le denomina sidebar.


```

<?php
    $cobros = infoPago($conexion);
    $cobros = $cobros[0];
?>
<div class="left-side-bar">
    <div class="brand-logo">
        <a href="inicio.php">
            
            
        </a>
        <div class="close-sidebar" data-toggle="left-sidebar-close">
            <i class="ion-close-round"></i>
        </div>
    </div>
    <div class="menu-block customscroll">
        <div class="sidebar-menu">
            <ul id="accordion-menu">
                <?php if ($cobros['mostrar_sidebar'] != 'false') { ?>
                <li class="dropdown">
                    <a href="pago.php" class="dropdown-toggle no-arrow">
                        <span class="micon fa fa-exclamation-circle"></span>
                        <span class="mtext">Realizar su pago <span class="fi-burst-new text-
                            danger new"></span>
                    </a>
                </li>
                <?php } ?>
                <li class="dropdown">
                    <a href="inicio.php" class="dropdown-toggle no-arrow">
                        <span class="micon dw dw-house-1"></span>
                        <span class="mtext">Inicio</span>
                    </a>

```

```

</li>
<li class="dropdown">
  <a href="javascript:;" class="dropdown-toggle">
    <span class="micon ion-person-stalker"></span>
    <span class="mtext">Pacientes</span>
  </a>
  <ul class="submenu">
    <li><a href="registrar_propietario.php">Registrar propietario</a></li>
    <li><a href="ver_propietarios.php">Ver todos los propietarios</a></li>
    <li><a href="registrar_paciente.php">Registrar paciente</a></li>
    <li><a href="ver_pacientes.php">Ver todos los pacientes</a></li>
  </ul>
</li>
<li class="dropdown">
  <a href="javascript:;" class="dropdown-toggle">
    <span class="micon ti-calendar"></span>
    <span class="mtext">Citas <span class="fi-burst-new text-danger new"></span></span>
  </a>
  <ul class="submenu">
    <li><a href="nueva_cita.php">Crear cita nueva</a></li>
    <li><a href="citas_mes.php">Citas del mes actual</a></li>
  </ul>
</li>
<li class="dropdown">
  <a href="javascript:;" class="dropdown-toggle">
    <span class="micon ti-folder"></span>
    <span class="mtext">Consultas</span>
  </a>
  <ul class="submenu">
    <li><a href="nueva_consulta.php">Nueva consulta</a></li>

```

```

        <li><a href="ver_consultas.php">Ver consultas</a></li>
    </ul>
</li>
<li class="dropdown">
    <a href="javascript:;" class="dropdown-toggle">
        <span class="micon fa fa-clone"></span>
        <span class="mtext">Notas</span>
    </a>
    <ul class="submenu">
        <li><a href="lasnotas.php">Ver todas las notas</a></li>
    </ul>
</li>
<li class="dropdown">
    <a href="javascript:;" class="dropdown-toggle">
        <span class="micon fa fa-briefcase"></span>
        <span class="mtext">Recetas</span>
    </a>
    <ul class="submenu">
        <li><a href="lasrecetas.php">Ver todas las recetas</a></li>
    </ul>
</li>
<li class="dropdown">
    <a href="javascript:;" class="dropdown-toggle">
        <span class="micon fa fa-wrench"></span>
        <span class="mtext">Almacén y servicios</span>
    </a>
    <ul class="submenu">
        <li><a href="agregar_producto.php">Registrar producto</a></li>
        <li><a href="ver_productos.php">Ver productos</a></li>
        <li><a href="registrar_servicio.php">Registrar servicio</a></li>
        <li><a href="ver_catalogo.php">Ver servicios</a></li>
    </ul>

```

```

    </ul>
  </li>
  <li class="dropdown">
    <a href="javascript:;" class="dropdown-toggle">
      <span class="micon fa fa-user-circle-o"></span>
      <span class="mtext">Usuarios (MVZ)</span>
    </a>
    <ul class="submenu">
      <li><a href="registrar_usuario.php">Registrar Usuario (MVZ)</a></li>
      <li><a href="ver_usuarios.php">Editar Usuario (MVZ)</a></li>
      <li><a href="ver_usuarios_firma.php">Agregar Firma de
        Usuario (MVZ)</a></li>
    </ul>
  </li>
  <li class="dropdown">
    <a href="tutoriales.php" class="dropdown-toggle no-arrow">
      <span class="micon fa fa-question-circle"></span>
      <span class="mtext">Respondemos tus dudas aquí</span>
    </a>
  </li>
  <?php if ($_SESSION['usuario'] == 'javier@lupware.com') { ?>
  <li class="dropdown">
    <a href="mostrar_pagar.php" class="dropdown-toggle no-arrow">
      <span class="micon fa fa-dollar"></span>
      <span class="mtext">Pagos</span>
    </a>
  </li>
  <?php } ?>

  <li class="dropdown">
    <a href="cerrar.php" class="dropdown-toggle no-arrow">

```

```

        <span class="micon dw dw-exit"></span>
        <span class="mtext">Salir</span>
    </a>
</li>
</ul>
</div>
</div>
</div>

```

El archivo header tiene el código para cuando se valida el usuario y contraseña para ingresar al sistema.

```

<div class="pre-loader">
    <div class="pre-loader-box">
        <div class="loader-logo"></div>
        <div class='loader-progress' id="progress_div">
            <div class='bar' id='bar1'></div>
        </div>
        <div class='percent' id='percent1'>0%</div>
        <div class="loading-text">
            Cargando...
        </div>
    </div>
</div>
<div class="header">
    <div class="header-left">
        <div class="menu-icon dw dw-menu"></div>
        <!-- <div class="search-toggle-icon dw dw-search2"
datatoggle="header_search"></div>
        <div class="header-search">
            </div> -->
    </div>
    <div class="header-right">

```

```

<div class="dashboard-setting user-notification">
  <div class="dropdown">
    <a class="dropdown-toggle no-arrow" href="javascript:;" data-toggle="right-
      sidebar">
      <i class="dw dw-settings2"></i>
    </a>
  </div>
</div>
<div class="user-notification">
  <div class="dropdown">
  </div>
</div>
<div class="user-info-dropdown">
  <div class="dropdown">
    <a class="dropdown-toggle" href="#" role="button" data-
      toggle="dropdown">
      <span class="user-icon dw dw-user-12">
      </span>
      <span class="user-name">
        <?php echo " ".$_SESSION['usuario']." "; ?></span>
      </a>
      <div class="dropdown-menu dropdown-menu-right dropdown-menu-icon-
        list">
        <a class="dropdown-item" href="cerrar.php"><i class="dw dw-
          logout"></i> Salir</a>
      </div>
    </div>
  </div>
</div>
</div>-->
</div>
</div>

```

Se muestran algunos códigos que fueron similares en varios archivos.
Código para registros.

Para el guardado en la base de datos y que tiene comunicación con el servidor:

```
<?php session_start();  
require 'sesion.php';  
require 'admin_config.php';  
require 'admin_funciones.php';  
  
$conexion = conexion($bd_config);  
if (!$conexion) {  
    header('Location: error.php');  
}  
  
//Para que las pueda usar si le dice que debe re-escribir datos  
$nombres_prop = "";  
$apellidos_prop = "";  
$calle_numero = "";  
$colonia = "";  
$codigo_p = "";  
$municipio = "";  
$estado = "";  
$telefono = "";  
$correo = "";  
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
  
//Tomar las variables  
$nombres_prop = ucwords(strtolower($_POST['nombres_prop']));  
$apellidos_prop = ucwords(strtolower($_POST['apellidos_prop']));  
$calle_numero = $_POST['calle_numero'];  
$colonia = $_POST['colonia'];  
$codigo_p = $_POST['codigo_p'];
```

```

$municipio = $_POST['municipio'];
$estado = $_POST['estado'];
$telefono = $_POST['telefono'];
$correo = $_POST['correo'];
$errores = "";

//Comprobar si ya existe el usuario
$statement = $conexion -> prepare('SELECT * FROM propietario WHERE
nombres_prop = :nombres_prop AND apellidos_prop = :apellidos_prop AND telefono =
:telefono LIMIT 1');
$statement->execute(array(':nombres_prop' => $nombres_prop, ':apellidos_prop' =>
$apellidos_prop, ':telefono' => $telefono));
$resultado = $statement ->fetch();
if ($resultado != false) {
    //Alert
    echo '<script type="text/javascript">
        alert("ERROR: Ya existe el propietario.");
    </script>';
    $errores = 'Ya existe el propietario.';
}
if ($errores == "") {
    $statement = $conexion->prepare('INSERT INTO propietario (id_propietario,
nombres_prop, apellidos_prop, calle_numero, colonia, codigo_p, municipio,
estado, telefono, correo) VALUES (null, :nombres_prop, :apellidos_prop,
:calle_numero, :colonia, :codigo_p, :municipio, :estado, :telefono, :correo)');

    $statement->execute(array(':nombres_prop' => $nombres_prop, ':apellidos_prop'
=> $apellidos_prop, ':calle_numero' => $calle_numero, ':colonia' => $colonia,
':codigo_p' => $codigo_p, ':municipio' => $municipio, ':estado' => $estado,
':telefono' => $telefono, ':correo' => $correo));

    //Alert para avisar que ya quedó
    echo '<script type="text/javascript"> alert("Propietario almacenado");</script>';

```



```
//Redirección
echo "<script type='text/javascript'> window.location='registrar_propietario.php';
</script>";
}
}
require 'views/registrar_propietario.view.php';
?>
```

Código que se muestra en pantalla al cliente:

```
<?php include('views/include/head.php'); ?>
<body>
<?php include('include/header.php'); ?>
<?php include('include/sidebar.php'); ?>
<?php include ('right-sidebar.php'); ?>
<!-- <div class="mobile-menu-overlay"></div -->
<div class="main-container">
  <div class="pd-ltr-20 xs-pd-20-10">
    <div class="min-height-200px">
      <div class="page-header">
        <div class="row">
          <div class="col-md-6 col-sm-12">
            <div class="title">
              <h4>Pacientes</h4>
            </div>
            <nav aria-label="breadcrumb" role="navigation">
              <ol class="breadcrumb">
                <li class="breadcrumb-item">
                  <a href="inicio.php">Inicio</a></li>
                <li class="breadcrumb-item active" aria-current="page">Registrar
propietario</li>
              </ol>
```

```

        </nav>
    </div>
    <div class="col-md-6 col-sm-12 text-right">
        <div class="dropdown">
        </div>
    </div>
</div>
<div class="pd-20 card-box mb-30">
    <div class="clearfix">
        <div class="pull-left">
            <h4 class="text-blue h4">Registrar propietario</h4>
            <p class="mb-30">* Los campos en asterísco son obligatorios. </p>
        </div>
    </div>
    <form enctype="multipart/form-data" id="registrar_propietario" action="<?php
echo htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="POST"
name="registrar_propietario" >
        <?php if(!empty($errores)): ?>
            <div class="error">
                <ul>
                    <?php echo $errores; ?>
                </ul>
            </div>
        <?php endif; ?>
        <div class="form-group row">
            <label class="col-sm-3 col-md-2 col-form-label">
                Nombre(s) *</label>
            <div class="col-sm-9 col-md-4">
                <input class="form-control" name="nombres_prop" value="<?php
echo $nombres_prop ?>" placeholder="Escribe el nombre(s) del

```

```

        propietario" type="text" required="">
    </div>
    <label class="col-sm-3 col-md-2 col-form-label">Apellidos *</label>
    <div class="col-sm-9 col-md-4">
        <input class="form-control" name="apellidos_prop" value="<?php
        echo $apellidos_prop ?>" placeholder="Escribe el apellido paterno"
        type="text" required="">
    </div>
</div>
<div class="form-group row">
    <label class="col-sm-3 col-md-2 col-form-label">Calle y
    número</label>
    <div class="col-sm-9 col-md-4">
        <input class="form-control" name="calle_numero" value="<?php
        echo $calle_numero ?>" placeholder="Escribe la calle y número"
        type="text">
    </div>
    <label class="col-sm-3 col-md-2 col-form-label">Colonia</label>
    <div class="col-sm-9 col-md-4">
        <input class="form-control" name="colonia" value="<?php echo
        $colonia ?>" placeholder="Escribe la colonia" type="text">
    </div>
</div>
<div class="form-group row">
    <label class="col-sm-3 col-md-2 col-form-label">Código
    Postal</label>
    <div class="col-sm-9 col-md-4">
        <input class="form-control" name="codigo_p" value="<?php echo
        $codigo_p ?>" placeholder="Escribe el código postal" type="text">
    </div>
</div>

```

```

<label class="col-sm-3 col-md-2 col-form-label">Municipio</label>
<div class="col-sm-9 col-md-4">
  <input class="form-control" name="municipio" value="<?php echo
    $municipio ?>" placeholder="Escribe el municipio" type="text">
</div>
</div>
<div class="form-group row">
  <label class="col-sm-3 col-md-2 col-form-label">Estado</label>
  <div class="col-sm-9 col-md-4">
    <input class="form-control" name="estado" value="<?php echo
      $estado ?>" placeholder="Escribe el estado" type="text">
  </div>
  <label class="col-sm-3 col-md-2 col-form-label">Teléfono *</label>
  <div class="col-sm-9 col-md-4">
    <input class="form-control" name="telefono" value="<?php echo
      $telefono ?>" placeholder="Escribe el teléfono" type="text"
      required="">
  </div>
</div>
<div class="form-group row">
  <label class="col-sm-3 col-md-2 col-form-label">Correo
    electrónico</label>
  <div class="col-sm-9 col-md-4">
    <input class="form-control" name="correo" value="<?php echo $correo ?>"
      placeholder="Escribe el correo electrónico" type="email">
  </div>
</div>
<div class="form-group text-center">
  <input type="submit" class="btn btn-primary" value="Registrar"
    id="submit" onclick="registrar_propietario.submit()"></button>
</div>

```

```

        </form>
        <!-- Default Basic Forms End -->
    </div>
</div>
</div>

<?php include ('views/include/footer.php'); ?>
<!-- js -->
<script src="vendors/scripts/core.js"></script>
    <script src="vendors/scripts/script.min.js"></script>
    <script src="vendors/scripts/process.js"></script>
    <script src="vendors/scripts/layout-settings.js"></script>
</body>
</html>

```

Código para consultar el listado de todos los registros guardados.

Para consulta en la base de datos a través del servidor:

```

<?php session_start();
require 'sesion.php';
require 'admin_config.php';
require 'admin_funciones.php';

$conexion = conexion($bd_config);
if (!$conexion) {
    header('Location: admin_error.php');
}
$pacientes = listaPacientes($conexion);
require 'views/ver_pacientes.view.php';
?>

```

Código que se le mostrará en pantalla al usuario:

```
<?php include ('views/include/head.php'); ?>
<body>
<?php include('views/include/header.php'); ?>
<?php include('views/include/sidebar.php'); ?>
<?php include ('right-sidebar.php'); ?>
<div class="main-container">
  <div class="pd-ltr-20 xs-pd-20-10">
    <div class="min-height-200px">
      <div class="page-header">
        <div class="row">
          <div class="col-md-6 col-sm-12">
            <div class="title">
              <h4>Pacientes</h4>
            </div>
            <nav aria-label="breadcrumb" role="navigation">
              <ol class="breadcrumb">
                <li class="breadcrumb-item"><a href="inicio.php">Inicio</a></li>
                <li class="breadcrumb-item active" aria-current="page">Ver todos
                los pacientes</li>
              </ol>
            </nav>
          </div>
          <div class="col-md-6 col-sm-12 text-right">
            <div class="dropdown">
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="card-box mb-30">
    <div class="pd-20">
```

```

    <h4 class="text-blue h4">Lista de pacientes</h4>
</div>
<div class="pb-20">
<table class="table hover multiple-select-row data-table-export nowrap">
  <thead>
    <tr>
      <th class="table-plus datatable-nosort">Nombre</th>
      <th>Especie</th>
      <th>Raza</th>
      <th>Sexo</th>
      <th>Nacimiento</th>
      <th>Propietario</th>
      <th class="datatable-nosort">Opciones</th>
    </tr>
  </thead>
  <tbody>
    <?php foreach ($pacientes as $pac): ?>
      <tr>
        <td class="table-plus"><?php echo $pac['nombre_pac']; ?> <?php
echo $pac['apellidos_prop']; ?></td>
          <td><?php echo $pac['especie']; ?></td>
          <td><?php echo $pac['raza']; ?></td>
          <td><?php echo $pac['sexo_pac']; ?></td>
          <td>
            <?php $date = new DateTime($pac['nacimiento']); echo $date-
>format('d/m/Y'); ?></td>
          <td><?php echo $pac['nombres_prop']; ?> <?php
echo $pac['apellidos_prop']; ?></td>
          <td>
            <div class="dropdown">
              <a class="btn btn-secondary dropdown-toggle" href="#"

```

```

role="button" data-toggle="dropdown">
  <i class="fa fa-ellipsis-h"></i>
</a>
<div class="dropdown-menu dropdown-menu-right">
  <a class="dropdown-item"
href="editar_paciente.php?id_paciente=<?php
echo $pac['id_paciente']; ?>"><i class="dw dw-
eye"></i> Editar</a>
  <a class="dropdown-item"
href="lista_consultas.php?id_paciente=<?php
echo $pac['id_paciente']; ?>"><i class="dw dw-folder-
158"></i> Consultas</a>
  <a class="dropdown-item"
href="lista_recetas.php?id_paciente=<?php
echo $pac['id_paciente']; ?>"><i class="dw dw-
notebook"></i> Recetas</a>
  <a class="dropdown-item"
href="lista_notas.php?id_paciente=<?php
echo $pac['id_paciente']; ?>"><i class="dw dw-
notepad-2"></i> Notas</a>
</div>
</div>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>
<!-- Export Datatable End -->
</div>

```



```

        <?php include ('views/include/footer.php'); ?>
    </div>
</div>
<!-- js -->
<script src="vendors/scripts/core.js"></script>
<script src="vendors/scripts/script.min.js"></script>
<script src="vendors/scripts/process.js"></script>
<script src="vendors/scripts/layout-settings.js"></script>
<script src="src/plugins/datatables/js/jquery.dataTables.min.js"></script>
<script src="src/plugins/datatables/js/dataTables.bootstrap4.min.js"></script>
<script src="src/plugins/datatables/js/dataTables.responsive.min.js"></script>
<script src="src/plugins/datatables/js/responsive.bootstrap4.min.js"></script>
<!-- buttons for Export datatable -->
<script src="src/plugins/datatables/js/dataTables.buttons.min.js"></script>
<script src="src/plugins/datatables/js/buttons.bootstrap4.min.js"></script>
<script src="src/plugins/datatables/js/buttons.print.min.js"></script>
<script src="src/plugins/datatables/js/buttons.html5.min.js"></script>
<script src="src/plugins/datatables/js/buttons.flash.min.js"></script>
<script src="src/plugins/datatables/js/pdfmake.min.js"></script>
<script src="src/plugins/datatables/js/vfs_fonts.js"></script>
<!-- Datatable Setting js -->
<script src="vendors/scripts/datatable-setting.js"></script></body>
</html>

```

Archivo para consultar el calendario para visualizar las vistas del mes:

```

<link rel="stylesheet" type="text/css" href="src/plugins/fullcalendar/fullcalendar.css">
<?php include('views/include/head.php'); ?>
<body>
<?php include('include/header.php'); ?>
<?php include('include/sidebar.php'); ?>
<?php include ('right-sidebar.php'); ?>

```

```

<div class="main-container">
  <div class="pd-ltr-20 xs-pd-20-10">
    <div class="min-height-200px">
      <div class="page-header">
        <div class="row">
          <div class="col-md-12 col-sm-12">
            <div class="title">
              <h4>Citas del mes</h4>
            </div>
            <nav aria-label="breadcrumb" role="navigation">
              <ol class="breadcrumb">
                <li class="breadcrumb-item">
                  <a href="inicio.php">Inicio</a></li>
                <li class="breadcrumb-item active" aria-current="page">Citas del
                  mes</li>
              </ol>
            </nav>
          </div>
        </div>
      </div>
    <div class="pd-20 card-box mb-30">
      <h4 class="text-center mb-30"><?php echo $mes_actual . ' ' .
        $ano_sistema; ?></h4>
      <div class="calendar-wrap">
        <div id='calendar1'></div>
      </div>
    </div>
  </div>
  <?php include ('views/include/footer.php'); ?>
</div>
</div>

```

```

<!-- js -->
<script src="vendors/scripts/core.js"></script>
<script src="vendors/scripts/script.min.js"></script>
<script src="vendors/scripts/process.js"></script>
<script src="vendors/scripts/layout-settings.js"></script>
<script src="src/plugins/fullcalendar/fullcalendar.min.js"></script>
<script src="vendors/scripts/calendar-setting.js"></script>
<script>
    jQuery(document).ready(function(){
    jQuery("#add-event").submit(function(){
        alert("Submitted");
        var values = {};
        $.each($("#add-event").serializeArray(), function(i, field) {
            values[field.name] = field.value;
        });
        console.log(
            values
        );
    });
});
(function () {
    'use strict';
    // ----- //
    // Calendar
    // ----- //
    jQuery(function() {
        // page is ready
        jQuery("#calendar1").fullCalendar({
            themeSystem: 'bootstrap4',
            // emphasizes business hours
            businessHours: false,

```

```

defaultView: 'month',
// event dragging & resizing
editable: true,
// header
header: {
  left: 'title',
  center: 'month,agendaWeek,agendaDay',
  right: 'today prev,next'
},
events: [
  <?php foreach ($las_citas as $citas): ?>
    {
      title: '<?php echo $citas['nombre_pac']; ?>: <?php
echo $citas['titulo_cita']; ?>',
      url: 'ver_cita.php?id_cita=<?php echo $citas['id_cita']; ?>',
      start: ' <?php echo $citas['fecha_cita'] . 'T' . $citas['hora_cita']; ?>',
      end: ' <?php echo $citas['fecha_cita'] . 'T' . $citas['hora_cita']; ?>'
    },
  <?php endforeach; ?>
],
})
});

})(jQuery);
</script>
</body>
</html>

```

Código para el archivo donde crea consultas, notas, recetas, servicios y productos consumidos por el paciente, así como los historiales de cada uno.

Archivo que hace peticiones a través del servidor hacía la base de datos:

```

<?php session_start();
require 'sesion.php';
require 'admin_config.php';
require 'admin_funciones.php';
$conexion = conexion($bd_config);
if (!$conexion) {
    header('Location: admin_error.php');
}
$id_consulta = id_recibido($_GET['id_consulta']);
$consulta = infoConsulta($conexion, $id_consulta);
$consulta = $consulta[0];
$fecha_nacimiento = new DateTime($consulta['nacimiento']);
$fecha_sistema = new DateTime(date('Y-m-d H:i:s'));
$dateInterval = $fecha_sistema->diff($fecha_nacimiento);
$edad = $dateInterval->format(' %y años ').PHP_EOL;
$notas = notasDeConsulta($conexion, $id_consulta);
$recetas = recetasDeConsulta($conexion, $id_consulta);

$lista_servicios = listaServicios($conexion);
$lista_productos = listaProductos($conexion);
$cotizados = cotizadosConsulta($conexion, $id_consulta);
require 'views/consulta.view.php';
?>

```

Código para la visualización en pantalla del usuario MVZ.

```

<?php include('views/include/head.php'); ?>
<script src="ckeditor/ckeditor.js"></script>
<body>
<?php include('include/header.php'); ?>
<?php include('include/sidebar.php'); ?>
<?php include ('right-sidebar.php'); ?>

```

```

<div class="main-container">
  <div class="pd-ltr-20 xs-pd-20-10">
    <div class="min-height-200px">
      <div class="page-header">
        <div class="row">
          <div class="col-md-6 col-sm-12">
            <div class="title">
              <h4>Pacientes</h4>
            </div>
            <nav aria-label="breadcrumb" role="navigation">
              <ol class="breadcrumb">
                <li class="breadcrumb-item"><a
                  href="inicio.php">Inicio</a></li>
                <li class="breadcrumb-item active" aria-
                  current="page">Registrar paciente</li>
              </ol>
            </nav>
          </div>
          <div class="col-md-6 col-sm-12 text-right">
            <div class="dropdown">
              </div>
            </div>
          </div>
        </div>
      </div>
      <!-- Default Basic Forms Start -->
      <div class="pd-20 card-box mb-30">
        <div class="clearfix">
          <div class="pull-left">
            <h4 class="text-blue h4">Registrar paciente</h4>
            <p class="mb-30">Complete los campos.</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

</div>
<form enctype="multipart/form-data" id="registrar_paciente" action="<?php
echo htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="POST"
name="registrar_paciente" >
  <?php if(!empty($errores)): ?>
    <div class="error">
      <ul>
        <?php echo $errores; ?>
      </ul>
    </div>
  <?php endif; ?>
  <div class="form-group row">
    <label class="col-sm-3 col-md-2 col-form-label">Propietario:</label>
    <div class="col-sm-9 col-md-4">
      <select name="Propietario_idprop" class="selectpicker form-control"
data-size="5" data-style="btn-outline-info">
        <option value="">Elegir una opcion...</option>
        <?php foreach ($propietarios as $prop): ?>
          <option value="<?php echo $prop['id_propietario'];
?>"><?php echo $prop['nombres_prop']; ?> <?php
echo $prop['apellidos_prop']; ?></option>
        <?php endforeach; ?>
      </select>
    </div>
    <label class="col-sm-3 col-md-2 col-form-label">Fotografía
(opcional):</label>
    <div class="col-sm-9 col-md-4">
      <input class="form-control" name="foto_pac" type="file">
    </div>
  </div>
  <div class="form-group row">

```

```

<label class="col-sm-3 col-md-2 col-form-label">Nombre:</label>
<div class="col-sm-9 col-md-4">
  <input class="form-control" name="nombre_pac" value="<?php echo
  $nombre_pac ?>" placeholder="Nombre del paciente." type="text"
  required="">
</div>
<label class="col-sm-3 col-md-2 col-form-label">Especie:</label>
<div class="col-md-4 col-sm-9">
  <select name="especie" required="" class="selectpicker form-
  control" data-size="5" data-style="btn-outline-info">
    <option value="">Elegir una opción...</option>
    <option value="Canino">Canino</option>
    <option value="Felino">Felino</option>
  </select>
</div>
</div>

<div class="form-group row">
  <label class="col-sm-3 col-md-2 col-form-label">Raza:</label>
  <div class="col-sm-9 col-md-4">
    <input class="form-control" name="raza" value="<?php echo $raza
    ?>" placeholder="Escribe la raza." type="text" required="">
  </div>
  <label class="col-sm-3 col-md-2 col-form-label">Sexo:</label>
  <div class="col-md-4 col-sm-9">
    <select name="sexo_pac" class="selectpicker form-control" data-
    size="5" data-style="btn-outline-info" required="">
      <option value="">Elegir una opcion...</option>
      <option value="Hembra">Hembra</option>
      <option value="Hembra Castrada">Hembra Castrada</option>
      <option value="Macho">Macho</option>
    </select>
  </div>
</div>

```



```

        <option value="Macho Castrado">Macho Castrado</option>
    </select>
</div>
</div>
<div class="form-group row">
    <label class="col-sm-3 col-md-2 col-form-label">Fecha de
    nacimiento:</label>
    <div class="col-sm-9 col-md-4">
        <input class="form-control" name="nacimiento" value="<?php echo
        $nacimiento ?>" type="date" required="">
    </div>
    <label class="col-sm-3 col-md-2 col-form-label">Peso:</label>
    <div class="col-sm-9 col-md-4">
        <input class="form-control" name="peso_pac" value="<?php echo
        $peso_pac ?>" placeholder="Escribe el peso." type="text">
    </div>
</div>
<div class="form-group row">
    <label class="col-sm-3 col-md-2 col-form-label">Color:</label>
    <div class="col-sm-9 col-md-4">
        <input class="form-control" name="color_pac" value="<?php echo
        $color_pac ?>" placeholder="Escribe el color." type="text">
    </div>
</div>
<div class="col-sm-12 col-md-12">
    <h4 class="h4 text-blue">Observaciones adicionales: </h4>
    <textarea name="observaciones_pac" id="obs_adi" rows="15"
    cols="80" required=""><?php echo $observaciones_pac ?></textarea>
    <!-- CK Editor -->
    <script>
        CKEDITOR.replace( 'obs_adi', {

```

```

        language: 'es',
        uiColor: '#9AB8F3'
    });
</script>
</div>
<br>
<div class="form-group text-center">
    <input type="submit" class="btn btn-primary" value="Registrar"
    id="submit" onclick="registrar_paciente.submit()"></button>
</div>
</form>
<!-- Default Basic Forms End -->
</div>
</div>
</div>
<?php include ('views/include/footer.php'); ?>
<!-- js -->
<script src="vendors/scripts/core.js"></script>
    <script src="vendors/scripts/script.min.js"></script>
    <script src="vendors/scripts/process.js"></script>
    <script src="vendors/scripts/layout-settings.js"></script>
</body>
</html>

```

Este código es similar al de registrar solo que aquí es para actualización de datos. Para la comunicación entre la base de datos y el servidor.

```

<?php session_start();
require 'sesion.php';
require 'admin_config.php';
require 'admin_funciones.php';
$conexion = conexion($bd_config);

```

```

if (!$conexion) {
    header('Location: error.php');
}
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    //Tomar las variables
    $id_lista_precios = $_POST['id_lista_precios'];
    $estudio_servicio = $_POST['estudio_servicio'];
    $existencias = $_POST['existencias'];
    $costo_prod = limpiarPrecio($_POST['costo_prod']);
    $precio = limpiarPrecio($_POST['precio']);
    $minimo = $_POST['minimo'];
    $maximo = $_POST['maximo'];
    //Actualizar información
    $statement = $conexion->prepare('UPDATE lista_precios SET estudio_servicio =
:estudio_servicio, existencias = :existencias, costo_prod = :costo_prod, precio = :precio,
minimo = :minimo, maximo = :maximo WHERE id_lista_precios = :id_lista_precios');
    $statement->execute(array(':estudio_servicio' => $estudio_servicio, ':existencias' =>
$existencias, ':costo_prod' => $costo_prod, ':precio' => $precio, ':minimo' => $minimo,
':maximo' => $maximo, ':id_lista_precios' => $id_lista_precios));
    //Alert para avisar que ya quedó
    echo '<script type="text/javascript">
        alert("Modificación realizada");
    </script>';
    //Redirección
    echo "<script type='text/javascript'>
        window.location='editar_producto.php?id=$id_lista_precios';
    </script>";
}
$id_lp = id_recibido($_GET['id']);
//Info del usuario a editar
$paraEditar = verListaEdicion($conexion, $id_lp);

```

```
$paraEditar = $paraEditar[0];  
require 'views/editar_producto.view.php';  
?>
```

Código para la vista del usuario MVZ

```
<?php include('views/include/head.php'); ?>  
<body>  
<?php include('include/header.php'); ?>  
<?php include('include/sidebar.php'); ?>  
<?php include ('right-sidebar.php'); ?>  
<div class="main-container">  
  <div class="pd-ltr-20 xs-pd-20-10">  
    <div class="min-height-200px">  
      <div class="page-header">  
        <div class="row">  
          <div class="col-md-6 col-sm-12">  
            <div class="title">  
              <h4>Productos</h4>  
            </div>  
            <nav aria-label="breadcrumb" role="navigation">  
              <ol class="breadcrumb">  
                <li class="breadcrumb-item">  
                  <a href="inicio.php">Inicio</a></li>  
                <li class="breadcrumb-item">  
                  <a href="ver_productos.php">Ver productos</a></li>  
                <li class="breadcrumb-item active" aria-current="page">Editar producto</li>  
              </ol>  
            </nav>  
          </div>  
          <div class="col-md-6 col-sm-12 text-right">
```

```

        <div class="dropdown">
        </div>
    </div>
</div>
</div>
</div>
<!-- Default Basic Forms Start -->
<div class="pd-20 card-box mb-30">
    <div class="clearfix">
        <div class="pull-left">
            <h4 class="text-blue h4">Editar la información.</h4>
            <p class="mb-30">Complete los campos.</p>
        </div>
    </div>
    <form id="registrar_usuario" action="<?php echo
    htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="POST"
    name="registrar_usuario" >
        <input type="hidden" name="id_lista_precios" value="<?php echo
        $paraEditar['id_lista_precios']; ?>">
        <div class="form-group row">
            <label class="col-sm-3 col-md-2 col-form-label">Nombre:</label>
            <div class="col-sm-9 col-md-4">
                <input class="form-control" name="estudio_servicio" value="<?php
                echo $paraEditar['estudio_servicio']; ?>" placeholder="Escribe el
                nombre del producto." type="text" required="">
            </div>
        </div>
        <div class="form-group row">
            <label class="col-sm-3 col-md-2 col-form-label">Costo
            bruto:</label>
            <div class="col-sm-9 col-md-4">
                <input class="form-control" name="costo_prod" value="<?php echo

```

```

$paraEditar['costo_prod']; ?>" placeholder="Escribe el costo bruto."
type="text" required="">
</div>
<label class="col-sm-3 col-md-2 col-form-label">Precio
venta:</label>
<div class="col-sm-9 col-md-4">
<input class="form-control" name="precio" value="<?php echo
$paraEditar['precio']; ?>" placeholder="Escribe el precio de venta al
público." type="text" required="">
</div>
</div>

<div class="form-group row">
<label class="col-sm-2 col-md-1 col-form-
label">Existencias:</label>
<div class="col-sm-8 col-md-3">
<input class="form-control" name="existencias" value="<?php echo
$paraEditar['existencias']; ?>" placeholder="Escribe las existencias"
type="number" required="">
</div>
<label class="col-sm-2 col-md-1 col-form-label">Mínimo:</label>
<div class="col-sm-8 col-md-3">
<input class="form-control" name="minimo" value="<?php echo
$paraEditar['minimo']; ?>" placeholder="Escribe el mínimo"
type="number" required="">
</div>
<label class="col-sm-2 col-md-1 col-form-label">Máximo:</label>
<div class="col-sm-8 col-md-3">
<input class="form-control" name="maximo" value="<?php echo
$paraEditar['maximo']; ?>" placeholder="Escribe el máximo"
type="number" required="">

```

```

        </div>
    </div>
    <div class="form-group text-center">
        <input type="submit" class="btn btn-primary" value="Actualizar"
            id="submit" onclick="registrar_usuario.submit()"></button>
    </div>
</form>
<!-- Default Basic Forms End -->
</div>
</div>
</div>
<?php include ('views/include/footer.php'); ?>
<!-- js -->
<script src="vendors/scripts/core.js"></script>
    <script src="vendors/scripts/script.min.js"></script>
    <script src="vendors/scripts/process.js"></script>
    <script src="vendors/scripts/layout-settings.js"></script>
</body>
</html>

```

Código para subir archivos de imágenes o fotos.

```

<?php
    // Trabajar con archivo
    $fileTmpPath = $_FILES['foto_pac']['tmp_name'];
    $fileName = $_FILES['foto_pac']['name'];
    $fileSize = $_FILES['foto_pac']['size'];
    $fileType = $_FILES['foto_pac']['type'];

    $fileNameCmps = explode(".", $fileName);
    $fileExtension = strtolower(end($fileNameCmps));

```

```

$newFileName = md5(time() . $fileName) . '.' . $fileExtension;
$allowedfileExtensions = array('jpg', 'jpeg', 'JPG', 'JPEG', 'png', 'pdf');
if (in_array($fileExtension, $allowedfileExtensions)) {
    // directory in which the uploaded file will be moved
    $uploadFileDir = './archivos/mascotas/';
    $dest_path = $uploadFileDir . $newFileName;
    if(move_uploaded_file($fileTmpPath, $dest_path))
    {
        $foto_pac = $newFileName;
    }
    else
    {
        $message = 'There was some error moving the file to upload directory. Please
        make sure the upload directory is writable by web server.';
    }
} else {
    $dest_path = "Sin archivo";
    $foto_pac = $dest_path;
}
?>

```

Código para rehabilitar.

```

<?php session_start();
require '../admin_config.php';
require '../admin_funciones.php';

$conexion = conexion($bd_config);
if (!$conexion) {
    header('Location: admin_error.php');
}

```



```

$id_lista_precios = id_recibido($_GET['id_lista_precios']);
$activado_producto = 'Publicado';
//Inhabilitar
$statement = $conexion->prepare('UPDATE lista_precios SET activado_producto =
:activado_producto WHERE id_lista_precios = :id_lista_precios');
$statement->execute(array(':activado_producto' => $activado_producto,
':id_lista_precios' => $id_lista_precios));
//Alert para avisar que ya quedó
echo '<script type="text/javascript">
    alert("Producto rehabilitado");
</script>';
//Redirección
echo "<script type='text/javascript'>
    window.location='../ver_productos.php';
</script>";
if (isset($_SESSION['usuario'])) {
    require 'views/agregar_producto.view.php';
} else {
    header('Location: index.php');
}
?>

```

Código para inhabilitar.

```

<?php session_start();
require '../admin_config.php';
require '../admin_funciones.php';

$conexion = conexion($bd_config);
if (!$conexion) {
    header('Location: admin_error.php');
}

```

```

$id_lista_precios = id_recibido($_GET['id_lista_precios']);
$activado_producto = 'Inhabilitado';
//Inhabilitar
$statement = $conexion->prepare('UPDATE lista_precios SET activado_producto =
:activado_producto WHERE id_lista_precios = :id_lista_precios');
$statement->execute(array(':activado_producto' => $activado_producto,
':id_lista_precios' => $id_lista_precios));
//Alert para avisar que ya quedó
echo '<script type="text/javascript">
    alert("Producto inhabilitado");
</script>';
//Redirección
echo "<script type='text/javascript'>
    window.location='../ver_productos.php';
</script>";
//Revisar si tiene una sesión y si no lo regresa al login/registro
if (isset($_SESSION['usuario'])) {
    require 'views/agregar_producto.view.php';
} else {
    header('Location: index.php');
}
?>

```

Código en javascript para que el usuario MVZ realice su firma.

```

/* Variables de Configuración */
var idCanvas='canvas';
var idForm='formCanvas';
var inputImagen='imagen';
var estiloDelCursor='crosshair';
var colorDelTrazo='#555';
var colorDeFondo='#fff';

```

```

var grosorDelTrazo=2;

/* Variables necesarias */
var contexto=null;
var valX=0;
var valY=0;
var flag=false;
var imagen=document.getElementById(inputImagen);
var anchoCanvas=document.getElementById(idCanvas).offsetWidth;
var altoCanvas=document.getElementById(idCanvas).offsetHeight;
var pizarraCanvas=document.getElementById(idCanvas);

/* Esperamos el evento load */
window.addEventListener('load',IniciarDibujo,false);

function IniciarDibujo(){
  /* Creamos la pizarra */
  pizarraCanvas.style.cursor=estiloDelCursor;
  contexto=pizarraCanvas.getContext('2d');
  contexto.fillStyle=colorDeFondo;
  contexto.fillRect(0,0,anchoCanvas,altoCanvas);
  contexto.strokeStyle=colorDelTrazo;
  contexto.lineWidth=grosorDelTrazo;
  contexto.lineJoin='round';
  contexto.lineCap='round';
  /* Capturamos los diferentes eventos */
  pizarraCanvas.addEventListener('mousedown',MouseDown,false);
  pizarraCanvas.addEventListener('mouseup',MouseUp,false);
  pizarraCanvas.addEventListener('mousemove',MouseMove,false);
  pizarraCanvas.addEventListener('touchstart',TouchStart,false);
  pizarraCanvas.addEventListener('touchmove',TouchMove,false);
}

```

```

pizarraCanvas.addEventListener('touchend', TouchEnd, false);
pizarraCanvas.addEventListener('touchleave', TouchEnd, false);
}
function MouseDown(e){
    flag=true;
    contexto.beginPath();
    valX=e.pageX-posicionX(pizarraCanvas); valY=e.pageY-
posicionY(pizarraCanvas);
    contexto.moveTo(valX, valY);
}
function MouseUp(e){
    contexto.closePath();
    flag=false;
}
function MouseMove(e){
    if(flag){
        contexto.beginPath();
        contexto.moveTo(valX, valY);
        valX=e.pageX-posicionX(pizarraCanvas); valY=e.pageY-
posicionY(pizarraCanvas);
        contexto.lineTo(valX, valY);
        contexto.closePath();
        contexto.stroke();
    }
}
function TouchMove(e){
    e.preventDefault();
    if (e.targetTouches.length == 1) {
        var touch = e.targetTouches[0];
        MouseMove(touch);
    }
}

```

```

}
function TouchStart(e){
  if (e.targetTouches.length == 1) {
    var touch = e.targetTouches[0];
    MouseDown(touch);
  }
}
function TouchEnd(e){
  if (e.targetTouches.length == 1) {
    var touch = e.targetTouches[0];
    MouseUp(touch);
  }
}
function posicionY(obj) {
  var valor = obj.offsetTop;
  if (obj.offsetParent) valor += posicionY(obj.offsetParent);
  return valor;
}
function posicionX(obj) {
  var valor = obj.offsetLeft;
  if (obj.offsetParent) valor += posicionX(obj.offsetParent);
  return valor;
}
/* Limpiar pizarra */
function LimpiarTrazado() {
  contexto=document.getElementById(idCanvas).getContext('2d');
  contexto.fillStyle=colorDeFondo;
  contexto.fillRect(0,0,anchoCanvas,altoCanvas);
}
/* Enviar el trazado */
function GuardarTrazado() {

```

```
imagen.value=document.getElementById(idCanvas).toDataURL('image/png');  
document.forms[idForm].submit();  
}
```

3.3 Realización del sistema web (interfaces).

En este capítulo se mostrarán las interfaces del sistema finalizado, cada una de ellas tiene su respectiva descripción que indica su funcionalidad.

Para que el usuario MVZ inicie sesión, son con las claves de acceso que se le proporcionaron por medio del formulario que se muestra en la figura 50.

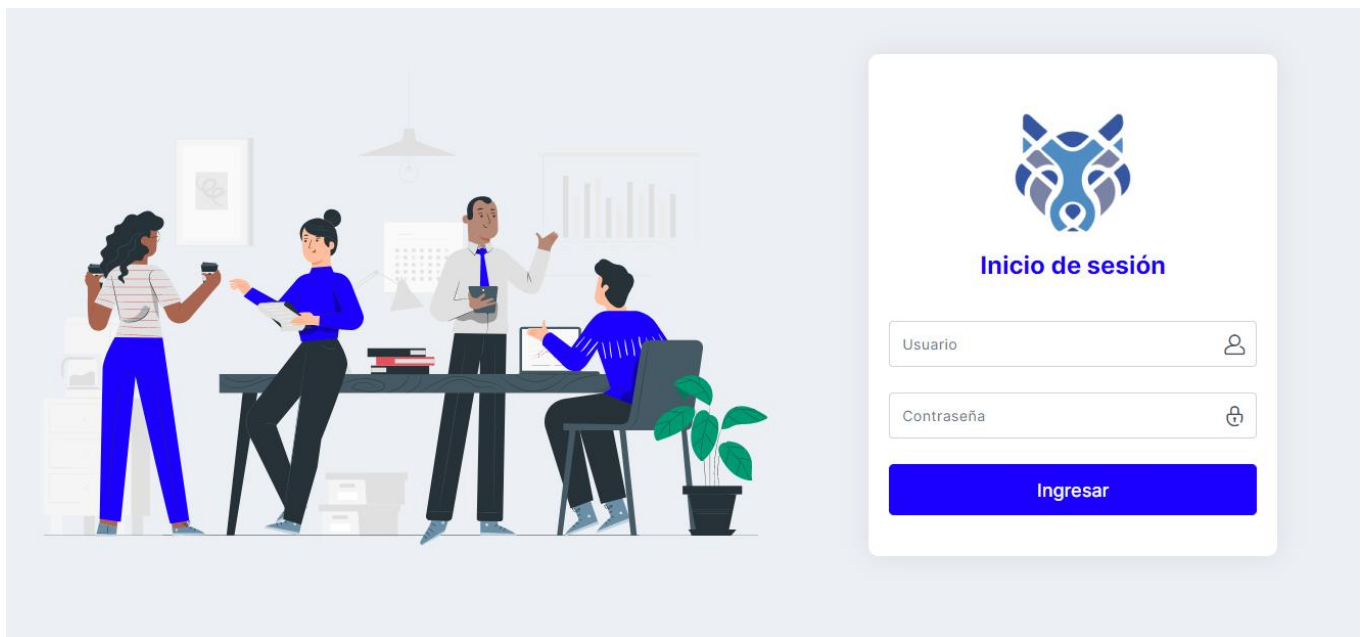


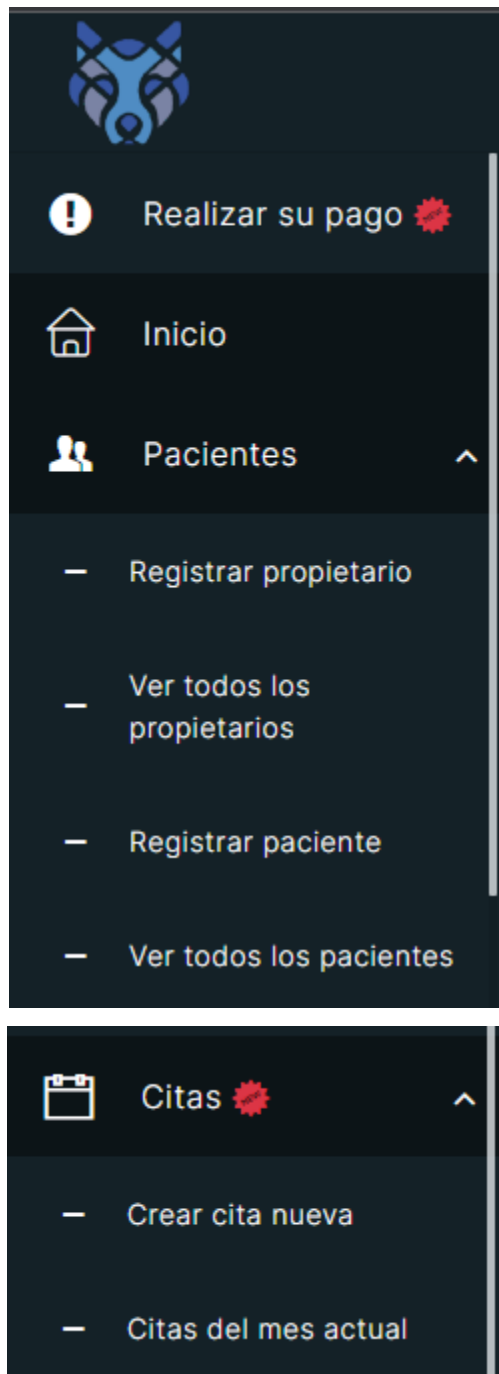
Figura 50. Pantalla de login.

Una vez ingresados los datos serán validados como se observa en la figura 51.



Figura 51. Indica que está validando los datos del usuario.

En la figura 52 se observa el menú que siempre será visible en el sistema para los veterinarios.



-  Consultas 
 - Nueva consulta
 - Ver consultas
-  Notas 
 - Ver todas las notas
-  Recetas 
 - Ver todas las recetas
-  Almacén y servicios 
 - Registrar producto
 - Ver productos
 - Registrar servicio
 - Ver servicios

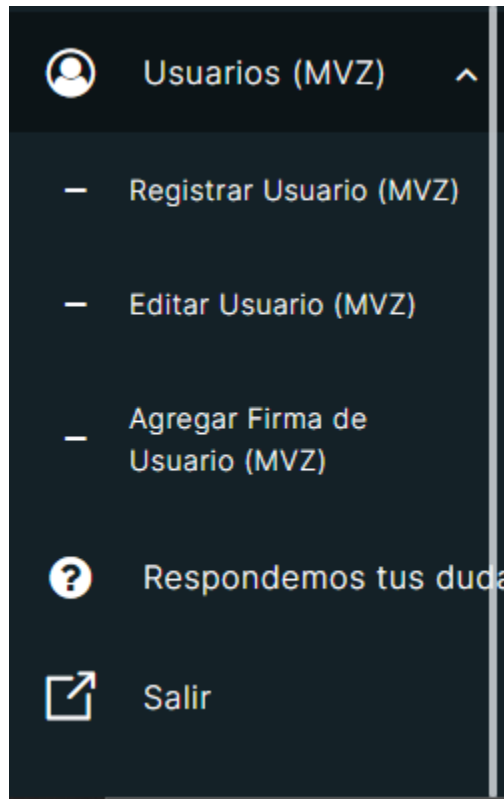


Figura 52. Menú que siempre se mostrará en el uso del sistema.

Cuando se haya iniciado la sesión, se tendrá la vista general de cuantos pacientes han sido atendidos por medio de las gráficas y todos los accesos en el menú para redirigirse a cada uno de los formularios o consultar los datos guardados, como se observa en la figura 53.



Figura 53. Una vez validado los datos, se prosigue con el acceso del sistema en general.

El veterinario podrá registrar al propietario en la sección de “pacientes” y seleccionando la opción de “registrar propietario”, con el formulario que se muestra en la figura 54.

The screenshot shows a web interface for a veterinary system. At the top right, there is a settings gear icon and a user profile icon with the email address 'contacto@lupware.com'. Below this is a breadcrumb trail: 'Inicio > Registrar propietario'. The main section is titled 'Registrar propietario' and includes a note: '* Los campos en asterísco son obligatorios.' The form consists of the following fields:

Nombre(s) *	<input type="text" value="Escribe el nombre(s) del propietario"/>	Apellidos *	<input type="text" value="Escribe el apellido paterno"/>
Calle y número	<input type="text" value="Escribe la calle y número"/>	Colonia	<input type="text" value="Escribe la colonia"/>
Código Postal	<input type="text" value="Escribe el código postal"/>	Municipio	<input type="text" value="Escribe el municipio"/>
Estado	<input type="text" value="Escribe el estado"/>	Teléfono *	<input type="text" value="Escribe el teléfono"/>
Correo electrónico	<input type="text" value="Escribe el correo electrónico"/>		

At the bottom center of the form is a blue button labeled 'Registrar'.

Figura 54. Formulario para registrar los datos del propietario.

Para consultar la lista de todos los propietarios se hará en la tabla en la sección del menú “pacientes” y buscar la opción de ver todos los propietarios para que se muestre el contenido, como se observa en la figura 55.

The screenshot shows a web application interface with a header containing a settings icon and a user profile for 'contacto@lupware.com'. Below the header is a section titled 'Lista propietarios' with buttons for 'Copiar', 'CSV', 'PDF', and 'Imprimir'. A search bar labeled 'Buscar' is positioned on the right. The main content is a table with the following data:

Nombre	Correo	Teléfono	Dirección	Editar
Alicia Perez Jardin 1		55 4810 9284	Del llano 333, colonia San Juan Tlalpan, CP: 14378. Tlalpan, CDMXX.	Editar
Alma Angélica García Lascurain		55 8580 8574	Retorno 16 No 8, colonia Avante, CP: 04460. Coyoacán, CDMX.	Editar
Andrea Equihua	pivi_zhety_vet@hotmail.com	5528956775	, colonia , CP: 09830. , CDMX.	Editar
Carmen De La Viãa		5552170903	Xochicaltitla 62 , colonia Villa Coyoacán, CP: 04010. Coyoacán, CDMX.	Editar
Claudia Lastiri	clau.lastiri@gmail.com	5516775193	Calvario 17, int 2, colonia Tlalpan centro, CP: 14000. Tlalpan, CDMX.	Editar
Daniel Mercado Hernández		5531404904	Rancho Miradores 56 , colonia Los Girasoles III , CP: 04920. Coyoacán, CDMX.	Editar
Diego Ricardo Luna		538847499383	, colonia , CP: . . .	Editar
Emiliano Ibarra		5559098298	Cascada 251, colonia Jardines del Pedregal , CP: . . .	Editar
Enrique Ortiz		5554323605	Heriberto Frãas 112, colonia Narvarte, CP: 03020. Benito Juárez , CDMX.	Editar
Fany Valdivieso	fannyv@gmail.com	55 15 74 22 85	Nieve 9, colonia El pedregall, CP: 58969. Naucalpan, Estado de México.	Editar

At the bottom right of the table, there is a pagination control showing page 1 of 3, with buttons for navigation.

Figura 55. Muestra del listado de todos los propietarios que están registrados en el sistema.

Para la actualización de uno o más datos de los propietarios se realizará después de haber consultado la tabla de ver todos los propietarios, seleccionando la última columna Editar para abrir el formulario que se observa en la figura 56.

The screenshot shows a web application interface. At the top right, there is a settings icon, a user profile icon, and the email address "contacto@lupware.com". Below this is a header section titled "Pacientes" with a breadcrumb "Inicio > Editar propietario". The main content area is titled "Editar propietario" and contains a note: "* Los campos en asterisco son obligatorios." The form consists of several input fields arranged in two columns. The first column contains: "Nombre(s) *" with the value "Claudia", "Calle y número" with "Calvario 17, int 2", "Código Postal" with "14000", "Estado" with "CDMX", and "Correo electrónico" with "clau.lastiri@gmail.com". The second column contains: "Apellidos *" with "Lastiri", "Colonia" with "Tlalpan centro", "Municipio" with "Tlalpan", and "Teléfono *" with "5516775193". A blue "Actualizar" button is positioned below the form fields. At the bottom of the page, it says "Desarrollo por Lupware".

Figura 56. Formulario con los campos de los datos existentes para realizar alguna modificación.

El veterinario podrá registrar a un nuevo paciente mediante el menú de pacientes, seleccionar registrar paciente para que se abra un formulario, como se observa en la figura 57.

The screenshot displays a web interface for patient registration. At the top right, there is a user profile icon and the email address 'contacto@lupware.com'. The main content area is titled 'Pacientes' and includes a breadcrumb 'Inicio > Registrar paciente'. Below this, the section 'Registrar paciente' prompts the user to 'Complete los campos.' The form consists of several input fields: 'Propietario' (a dropdown menu with 'Elegir una opción...'), 'Nombre' (text input with placeholder 'Nombre del paciente.'), 'Raza' (text input with placeholder 'Escribe la raza.'), 'Fecha de nacimiento' (date picker with placeholder 'dd/---/aaaa'), 'Color' (text input with placeholder 'Escribe el color.'), 'Fotografía (opcional)' (file upload button 'Seleccionar archivo' and status 'Sin archivos seleccionados'), 'Especie' (dropdown menu with 'Elegir una opción...'), 'Sexo' (dropdown menu with 'Elegir una opción...'), and 'Peso' (text input with placeholder 'Escribe el peso.'). Below the form is a section for 'Observaciones adicionales:' with a rich text editor toolbar. A blue 'Registrar' button is positioned at the bottom center of the form. The footer of the page reads 'Desarrollo por Lupware'.

Figura 57. Formulario para llenar los datos del paciente.

El veterinario desde el menú podrá ver el listado de todos sus pacientes con diferentes opciones, cómo se observa en la figura 58.

Nombre	Especie	Raza	Sexo	Nacimiento	Propietario	Opciones
Dunca García Lascurain	Canino	Mestizo	Macho	26/05/2021	Alma Angélica García Lascurain	...
Kira Silva Martínez	Canino	BÃ³xer	Hembra	12/08/2019	Martha Silva Martínez	...
Laika Ibarra	Canino	Yorkshire terrier	Hembra Castrada	15/10/2010	Emiliano Ibarra	...
Leitha Ibarra	Canino	Yorkshire terrier	Hembra Castrada	15/10/2010	Emiliano Ibarra	...
Leonidas De La Viãza	Felino	DPC	Macho Castrado	23/02/2017	Carmen De La Viãza	...

Figura 58. Listado de todos los pacientes registrados en el sistema.

Para agendar citas, desde el listado de todos los pacientes elegirá a uno de ellos en la última columna de la tabla donde se elegirá para crear la cita, como se aprecia en la figura 59.

contacto@lupware.com

Citas

Inicio > [Crear cita nueva](#)

Selecciona un paciente para agendar una cita

Mostrando 10 resultados

Nombre	Propietario	Especie	Seleccionar
Dunca García Lascurain	Alma Angélica García Lascurain	Canino	<input checked="" type="radio"/> Elegir
Kira Silva Martínez	Martha Silva Martínez	Canino	<input checked="" type="radio"/> Elegir
Laika Ibarra	Emiliano Ibarra	Canino	<input checked="" type="radio"/> Elegir
Leitha Ibarra	Emiliano Ibarra	Canino	<input checked="" type="radio"/> Elegir
Leonidas De La Viã±a	Carmen De La Viã±a	Felino	<input checked="" type="radio"/> Elegir
Leono Ibã±ez Garcia	Paola Ibã±ez Garcia	Felino	<input checked="" type="radio"/> Elegir
Luca Escutia	Perla Escutia	Canino	<input checked="" type="radio"/> Elegir
Lucy Mancilla	Valeria Mancilla	Felino	<input checked="" type="radio"/> Elegir
Ludwika Ibarra	Emiliano Ibarra	Canino	<input checked="" type="radio"/> Elegir

Figura 59. Listado de todos los pacientes para poder crearles su respectiva cita.

Seleccionado el paciente, se mostrará un formulario para el llenado y así crear una cita, como se observa en la figura 60.



Citas

Inicio > [Crear cita nueva](#)

Crear cita nueva










Complete los campos.

MVZ asignado:

Fecha:  Hora: 

Título cita:

Descripción cita (opcional):

B I |  |  |  |  |  |  |  |  | 

[Registrar](#)

Figura 60. Formulario para crear la cita nueva del paciente.

Cómo se observa en la figura 61, se tiene un calendario del mes actual donde el veterinario podrá revisar las consultas agendadas de los pacientes.

The screenshot displays a web interface for a veterinary appointment calendar. At the top right, there is a user profile icon and the email address 'contacto@lupware.com'. Below this, the page title is 'Citas del mes' with a breadcrumb link 'Inicio > Citas del mes'. The main content area is titled 'Marzo 2022' and 'March 2022'. It features a navigation bar with 'mes', 'semana', and 'día' buttons, and a 'Hoy' button with left and right arrows. The calendar grid shows days from Sunday to Saturday. A blue box highlights a specific appointment on Monday, March 7th, at 12:20p, labeled 'Dunca: vacuna'. The footer indicates 'Desarrollo por Lupware'.

Figura 61. Calendario del mes actual que muestra a los usuarios MVZ que consultas tienen durante ese mes.

Se puede seleccionar el día y actualizar datos de la cita en caso de que el propietario no pueda asistir en el día o la hora por lo que se abrirá un formulario, como se observa en la figura 62.

The screenshot shows a web application interface for managing appointments. At the top right, there is a user profile icon and the email address 'contacto@lupware.com'. The main content area is divided into two sections: 'Cita' and 'Recordatorio'.

Cita

Paciente:

MVZ asignado:

Fecha: Hora:

Título cita:

Descripción cita (opcional):

Recordatorio

¿Quieres enviar un **recordatorio a Alma Angélica García Lascurain** sobre la cita que tienen programada?

Elige una opción y sigue los pasos que se te indiquen.

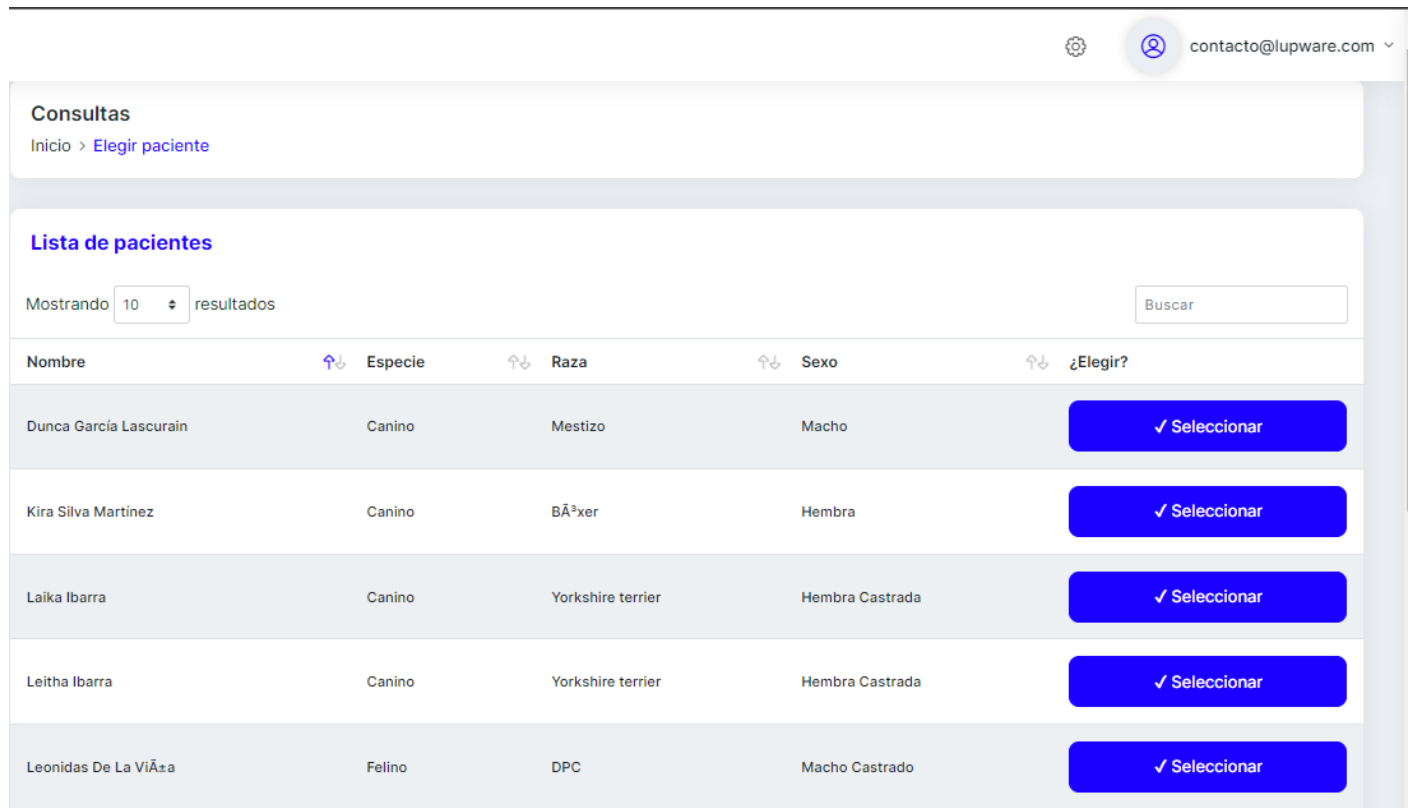
1) Whatsapp:

2) Correo:

***Nota:** Si no has registrado el correo o el número celular del propietario, los recordatorios no podrán realizarse.

Figura 62. Interfaz para editar los datos de la cita

Para la creación de consultas, igualmente se podrá seleccionar a uno de ellos por medio del listado que se aprecia en la figura 63.



The screenshot displays a web application interface for patient selection. At the top right, there is a user profile icon and the email address 'contacto@lupware.com'. Below this, the page title is 'Consultas' with a breadcrumb trail 'Inicio > Elegir paciente'. The main section is titled 'Lista de pacientes' and includes a filter for 'Mostrando 10 resultados' and a search box labeled 'Buscar'. A table lists five patients with columns for 'Nombre', 'Especie', 'Raza', 'Sexo', and '¿Elegir?'. Each row has a blue button with a checkmark and the text 'Seleccionar'.

Nombre	Especie	Raza	Sexo	¿Elegir?
Dunca García Lascurain	Canino	Mestizo	Macho	✓ Seleccionar
Kira Silva Martinez	Canino	BÃ³xer	Hembra	✓ Seleccionar
Laika Ibarra	Canino	Yorkshire terrier	Hembra Castrada	✓ Seleccionar
Leitha Ibarra	Canino	Yorkshire terrier	Hembra Castrada	✓ Seleccionar
Leonidas De La ViÃ±a	Felino	DPC	Macho Castrado	✓ Seleccionar

Figura 63. Interfaz para elegir uno de todos los pacientes para crearle su respectiva consulta.

Para el llenado de información general, se mostrará un formulario que se deberá completar, como se aprecia en la figura 64.

Paciente: Dunca
Inicio > Nueva consulta

Dunca
Canino

Propietario
Alma Angélica García Lascurain

Teléfono
55 8580 8574

Correo electrónico

Fechas

Nacimiento
26/05/2021

Edad
0 años

Costos

Total acumulado:
\$0.00

General Nueva nota Historial notas Receta Historial recetas Agregar servicio Agregar producto

Información general del paciente

Fecha consulta: 18/03/2022 07:05 pm Cambiar fecha (respetar formato): 2022-03-18 19:05:43

Tipo de consulta: Escribe el tipo de consulta.

Motivo de consulta: Escribe el motivo de la consulta.

Observaciones:

B I | [Rich text editor icons] ?

Ninguna

Guardar

Figura 64. Interfaz para las diferentes secciones, aquí se muestra la general donde el MVZ llenara el tipo y motivo de consulta.

Para añadir notas respecto a la cita del paciente se hará a través de la pestaña “notas” para llenar el formulario que se observa en la figura 65.

The screenshot shows a web application interface for creating a note for a patient named Dunca. The interface is divided into two main sections: a patient profile sidebar on the left and a main form area on the right.

Patient Profile (Left Sidebar):

- Paciente:** Dunca
- Inicio > Nueva consulta**
- Profile Image:** A cartoon dog icon.
- Nombre:** Dunca
- Especie:** Canino
- Propietario:** Alma Angélica García Lascurain
- Teléfono:** 55 8580 8574
- Correo electrónico:** (link)
- Fechas:**
 - Nacimiento:** 26/05/2021
 - Edad:** 0 años
- Costos:**
 - Total acumulado:** \$0.00

Main Form Area (Right):

- Navigation:** General, **Nueva nota** (active), Historial notas, Receta, Historial recetas, Agregar servicio, Agregar producto
- Form Title:** Nueva nota
- Título:** Titulo de la nota (input field)
- Nota:** A rich text editor with a toolbar containing icons for Bold (B), Italic (I), Text Color, Background Color, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and Help (?).
- Imagen:** Seleccionar archivo | Sin archivos seleccionados (file upload field)
- Guardar:** A blue button to save the note.

Figura 65. Interfaz del formulario para crear una nota.

Para ver el historial de todas las notas que creó el veterinario se consultarán en la pestaña “historial de notas”, como se observa en la figura 66.

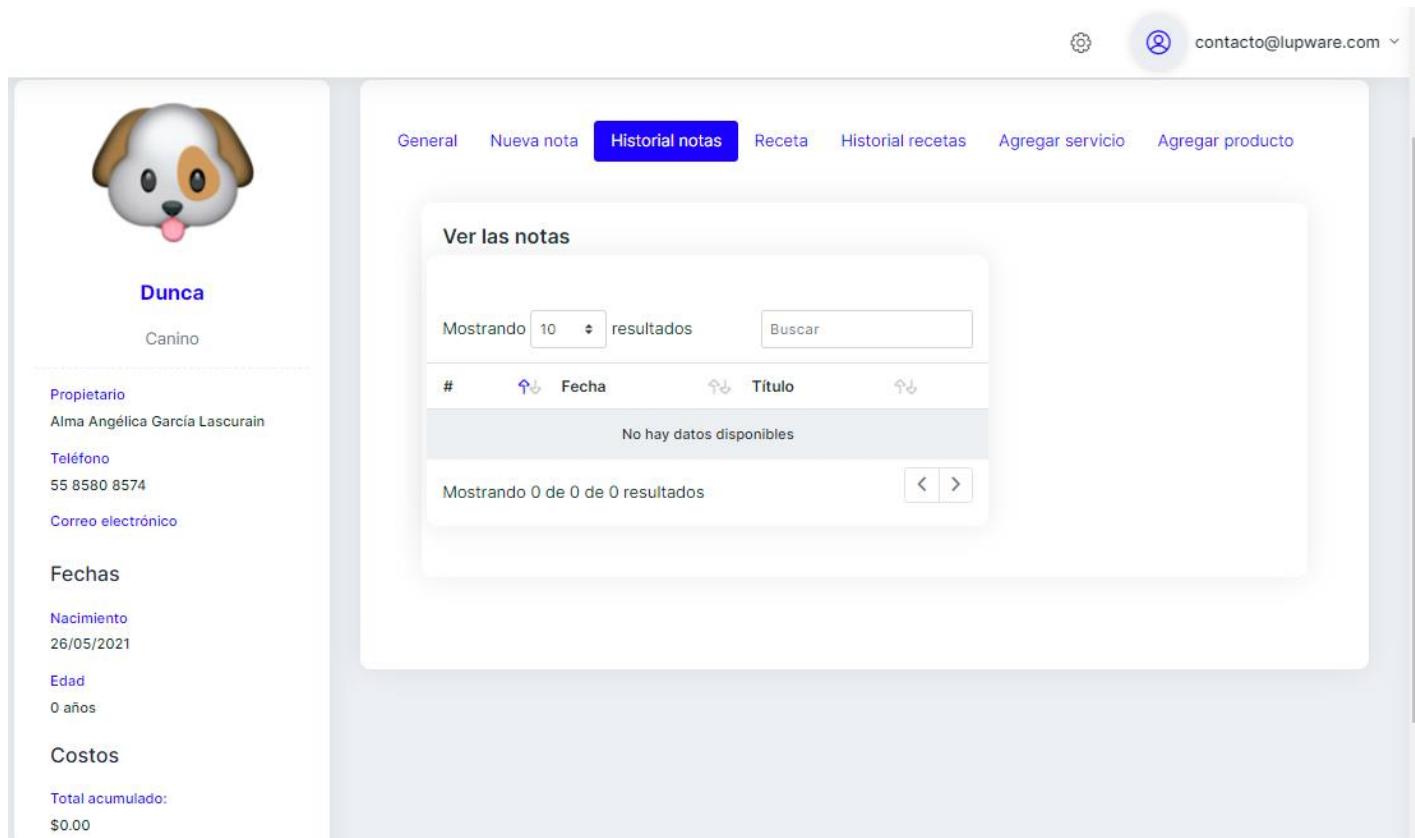


Figura 66. Interfaz para ver todas las notas que se han registrado del paciente.

Para la creación de recetas se hará en la pestaña donde dice “receta” la cual, abrirá un formulario que deberá completar, como se observa en la figura 67.

The screenshot shows a web application interface for creating a prescription. On the left, there is a profile card for a patient named Dunca, a dog, with the following details:

- Propietario:** Alma Angélica García Lascurain
- Teléfono:** 55 8580 8574
- Correo electrónico:** (link)
- Fechas:**
 - Nacimiento:** 26/05/2021
 - Edad:** 0 años
- Costos:**
 - Total acumulado:** \$0.00

The main content area is titled "Receta" and contains the following elements:

- Navigation tabs: General, Nueva nota, Historial notas, **Receta**, Historial recetas, Agregar servicio, Agregar producto
- Weight field: "Peso:" with a value of 20
- Description field: "Descripción:" with a rich text editor toolbar (Bold, Italic, Underline, Bulleted list, Numbered list, Indent, Outdent, Link, Unlink, Image) and a large text input area.
- Save button: "Guardar"

Figura 67. Interfaz donde el MVZ realizará la receta para el paciente.

En la pestaña “historial recetas”, se consultará todas las recetas que se han generado para el paciente como se observa en la figura 68.

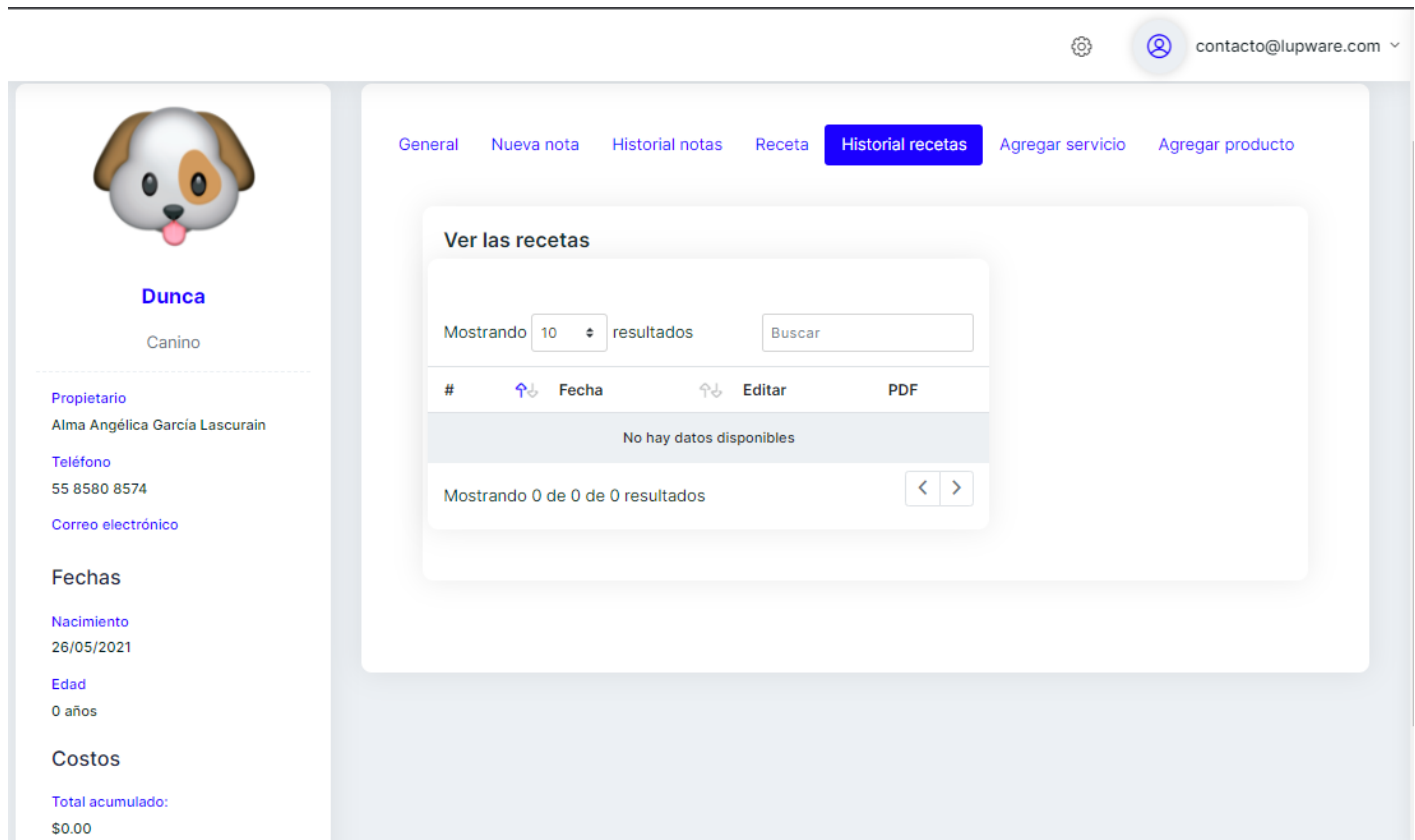


Figura 68. Interfaz que muestra todas las recetas que le han sido generadas al paciente.

En la pestaña “agregar servicio” se muestra todos los servicios cotizados y los que brinda la veterinaria, como se aprecia en la figura 69.

The screenshot shows a user interface for a veterinary management system. At the top right, there is a settings gear icon and a user profile icon with the email address 'contacto@lupware.com'. The main header area displays 'Paciente: Dunca' and a breadcrumb trail 'Inicio > Nueva consulta'. Below this, on the left, is a profile card for 'Dunca', a dog, with the name 'Dunca' and the breed 'Canino'. The owner's name is 'Alma Angélica García Lascurain' and the phone number is '55 8580 8574'. The main content area has several tabs: 'General', 'Nueva nota', 'Historial notas', 'Receta', and 'Historial recetas'. Below these are two buttons: 'Agregar servicio' (highlighted in blue) and 'Agregar producto'. A modal window titled 'Servicios cotizados:' is open, showing a search interface with 'Mostrando 10 resultados' and a search box containing 'Buscar'. Below the search box is a table header with columns: 'Nombre', 'Cantidad', 'Total', and 'Eliminar'. The table body is currently empty, displaying the message 'No hay datos disponibles'.

Figura 69. Interfaz donde se muestran todos los servicios contratados para el paciente.

Del listado de los servicios que se muestran, el veterinario buscará el que contratará el propietario seleccionando “agregar” de la última columna de la tabla, como se observa en la figura 70.

Mostrando 0 de 0 de 0 resultados

Agregar nuevo servicio a la consulta:

Mostrando 10 resultados

Nombre	Precio	Agregar
4 DX (Dirofilaria immitis, Ehrlichia canis, Borrelia)	\$1,000.00	+
Acidos biliares	\$450.00	+
Acidos biliares (basal+posprandial)	\$800.00	+
ACTH	\$1,000.00	+
Amoniaco (Entre 12:30 y 3:00)	\$500.00	+
An?lisis bacteriológico general c/antibiograma	\$600.00	+
An?lisis bacteriológico y micológico para otitis 1 oido	\$900.00	+

Figura 70. Continuación de la pestaña de agregar servicio que está abajo del listado de los que ya se agregaron en la consulta del paciente.

El veterinario una vez que selecciona uno de los servicios, podrá agregar cierta cantidad como se aprecia en la figura 71.

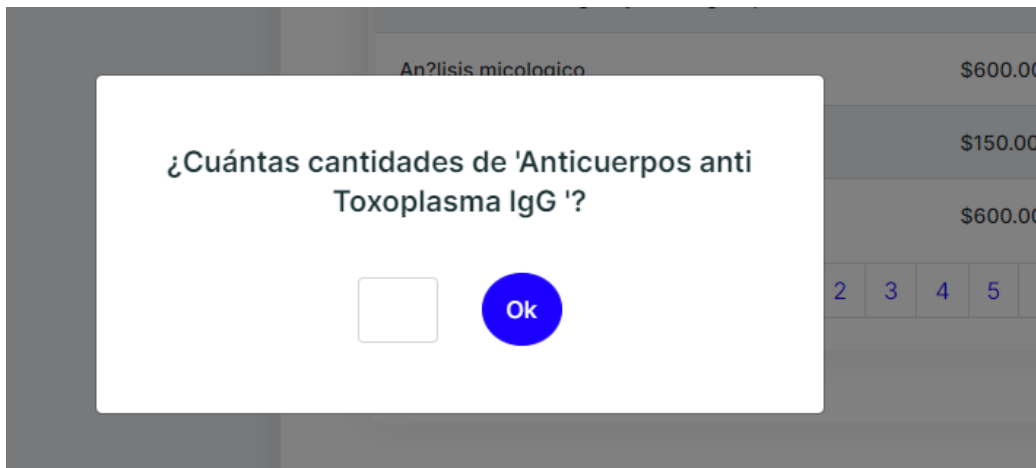


Figura 71. Interfaz para solicitar cuantas cantidades se necesitan de ese servicio.

Al igual que los servicios, las veterinarias ofrece sus productos los cuales también estarán en un listado, cómo se observa en la figura 72 y así solamente el veterinario podrá agregarlos a su historial del paciente.

The screenshot shows a web interface for a veterinary clinic. At the top right, there is a settings gear icon and a user profile icon with the email address "contacto@lupware.com". The main header area displays "Paciente: Dunca" and a breadcrumb trail "Inicio > Nueva consulta". On the left side, there is a profile card for the patient "Dunca", a dog emoji, and the species "Canino". Below this, contact information for the owner "Alma Angélica García Lascurain" is listed, including a phone number "55 8580 8574" and a link for "Correo electrónico". The main content area has a navigation bar with options: "General", "Nueva nota", "Historial notas", "Receta", and "Historial recetas". Below this, there are two buttons: "Agregar servicio" and "Agregar producto", with the latter being highlighted in blue. The "Agregar producto" button is active, showing a modal window titled "Productos cotizados:". This modal contains a search bar with "Mostrando 10 resultados" and a "Buscar" input field. Below the search bar is a table header with columns: "Nombre", "Cantidad", "Total", and "Eliminar", each with a sort icon. The table body is currently empty, displaying the message "No hay datos disponibles". At the bottom of the modal, it shows "Mostrando 0 de 0 de 0 resultados" and navigation arrows.

Figura 72. Interfaz para agregar uno o varios productos de la consulta del paciente.

En la última columna de la tabla, como se observa en la figura 73 se podrá agregar el producto que seleccione el veterinario.

The screenshot shows a web application interface. At the top right, there is a settings gear icon, a user profile icon, and the email address 'contacto@lupware.com'. On the left, a sidebar contains the following information:

- Fechas**
- Nacimiento**: 26/05/2021
- Edad**: 0 años
- Costos**
- Total acumulado:** \$0.00

The main content area is titled 'Agregar nuevo producto a la consulta:'. It features a search bar with the text 'Mostrando 10 resultados' and a 'Buscar' button. Below this is a table with the following columns: 'Nombre', 'Precio', and 'Agregar'. The table lists four products:

Nombre	Precio	Agregar
Producto A	\$150.00	+
Producto Bb	\$2,500.00	+
Vacuna	\$450.00	+
Vitaminas	\$65.00	+

At the bottom of the table, it says '1-4 de 4 envíos' and includes a pagination control with a blue box containing the number '1' and arrows for navigation.

Figura 73. Interfaz que esta abajo del listado de productos que han solicitado para el paciente.

En la sección de “consultas” es para redireccionarse a donde está todo el listado de los pacientes, como se aprecia en la figura 74 para realizar la consulta más rápida y de la forma más sencilla.

Ver consultas
Inicio > [Elegir paciente](#)

Elige un paciente para ver sus consultas
Selecciona una opción.

Mostrando 10 resultados

Nombre	Especie	Raza	Sexo	Seleccionar
Dunca García Lascurain	Canino	Mestizo	Macho	✓ Elegir
Kira Silva Martínez	Canino	BÃ³xer	Hembra	✓ Elegir
Laika Ibarra	Canino	Yorkshire terrier	Hembra Castrada	✓ Elegir
Leitha Ibarra	Canino	Yorkshire terrier	Hembra Castrada	✓ Elegir
Leonidas De La ViÃ±a	Felino	DPC	Macho Castrado	✓ Elegir

Figura 74. Interfaz que le muestra al usuario el listado de todos pacientes para ver sus consultas.

Seleccionado el paciente se mostrará todo el historial de las consultas que ha tenido, como se observa en la figura 75.

Ver consultas
Inicio > [Elegir paciente](#)

Lista de consultas
Elegir una consulta

Mostrando 10 resultados

Fecha	Tipo de consulta	Motivo	Costo	Seleccionar
04/03/2022 08:33 pm			\$0.00	✓ Elegir
07/03/2022 10:00 pm			\$0.00	✓ Elegir
11/11/2021 01:09 pm			\$1,200.00	✓ Elegir
12/10/2021 09:06 pm			\$0.00	✓ Elegir
12/10/2021 09:23 pm			\$0.00	✓ Elegir

Figura 75. Interfaz que muestra el listado de sus consultas del paciente seleccionado.

Al igual que las consultas se podrá ver el listado de los pacientes, pero aquí se consultarán todas las notas, como se aprecia en la figura 76.

Ver notas
Inicio > [Elegir paciente](#)

Elige un paciente para ver sus notas
Selecciona una opción.

Mostrando 10 resultados

Nombre	Especie	Raza	Sexo	Seleccionar
Dunca García Lascurain	Canino	Mestizo	Macho	<input checked="" type="checkbox"/> Elegir
Kira Silva Martínez	Canino	BÃ³xer	Hembra	<input checked="" type="checkbox"/> Elegir
Laika Ibarra	Canino	Yorkshire terrier	Hembra Castrada	<input checked="" type="checkbox"/> Elegir
Leitha Ibarra	Canino	Yorkshire terrier	Hembra Castrada	<input checked="" type="checkbox"/> Elegir
Leonidas De La ViÃ±a	Felino	DPC	Macho Castrado	<input checked="" type="checkbox"/> Elegir

Figura 76. Interfaz del listado de todos los pacientes para ver todas las notas de cada uno de ellos.

Por cada paciente se consultará su respectiva receta seleccionando en la última columna elegir, como se observa en la figura 77.

Ver recetas
Inicio > [Elegir paciente](#)

Elige un paciente para ver sus recetas
Selecciona una opción.

Mostrando 10 resultados

Nombre	Especie	Raza	Sexo	Seleccionar
Dunca García Lascurain	Canino	Mestizo	Macho	<input checked="" type="checkbox"/> Elegir
Kira Silva Martínez	Canino	BÃ³xer	Hembra	<input checked="" type="checkbox"/> Elegir
Laika Ibarra	Canino	Yorkshire terrier	Hembra Castrada	<input checked="" type="checkbox"/> Elegir
Leitha Ibarra	Canino	Yorkshire terrier	Hembra Castrada	<input checked="" type="checkbox"/> Elegir
Leonidas De La ViÃ±a	Felino	DPC	Macho Castrado	<input checked="" type="checkbox"/> Elegir

Figura 77. Interfaz para consultar las recetas por paciente.

Al consultar las recetas, se tendrán las opciones de descargar en formato PDF o copiar el enlace, como se observa en la figura 78 para enviárselo al propietario y lo descarguen en el momento que ellos lo deseen.

contacto@lupware.com

Ver recetas

Inicio > Elegir nota

Lista de recetas

Elegir una consulta

Mostrando 10 resultados

Fecha	Consulta relacionada	URL	Seleccionar
11/11/2021 03:49 pm	#73	https://vetapp.lupware.com/miconsultorio Copiar	<input type="radio"/> Elegir
19/12/2021 10:20 pm	#80	https://vetapp.lupware.com/miconsultorio Copiar	<input type="radio"/> Elegir

Figura 78. Interfaz donde muestra todas las recetas por paciente.

El veterinario podrá agregar sus nuevos productos desde la sección de “almacén y servicios”, el cual abrirá un formulario como se observa en la figura 79.

The image shows a web application interface for adding a new product. At the top right, there is a settings gear icon and a user profile icon with the email address 'contacto@lupware.com'. Below this, a breadcrumb trail reads 'Inicio > Agregar producto'. The main heading is 'Agregar producto' in blue, followed by the instruction 'Completa los campos.' The form contains several input fields: 'Nombre:' with a placeholder 'Escribe el nombre del producto.'; 'Costo bruto:' with a placeholder 'Escribe el costo bruto.'; 'Precio venta:' with a placeholder 'Escribe el precio de venta al público.'; 'Existencias:' with a placeholder 'Escribe las existencias'; 'Mínimo:' with a placeholder 'Escribe el mínimo'; and 'Peso:' with a placeholder 'Escribe el máximo'. A blue 'Registrar' button is positioned at the bottom center of the form.

Figura 79. Interfaz del formulario para agregar un nuevo producto.

Al consultar el listado de todos los productos que se han registrado en el sistema, en la última columna de “acciones”, como se aprecia en la figura 80, se tendrán las opciones de editar (para actualizar uno o más datos), inhabilitarlo o deshabilitarlo para que se agregue a la receta del paciente.

The screenshot shows a web application interface for managing products. At the top right, there is a user profile icon and the email address "contacto@lupware.com". Below this, the page title is "Productos" with a breadcrumb "Inicio > Ver productos". The main section is titled "Ver productos" and contains a toolbar with buttons for "Copiar", "CSV", "PDF", and "Imprimir", along with a search box labeled "Buscar". Below the toolbar is a table with the following columns: "Producto", "Costo neto", "Precio venta", "Existencias", "Mínimo", "Máximo", "¿Activo en sistema?", and "Acciones". The table contains three rows of product data. A dropdown menu is open over the "Acciones" column for the second row, showing three options: "Editar", "Inhabilitar", and "Rehabilitar".

Producto	Costo neto	Precio venta	Existencias	Mínimo	Máximo	¿Activo en sistema?	Acciones
Producto A	100.00	150.00	-29	10	100	Publicado	...
Producto Bb	2,000.00	2,500.00	240	80	60	Publicado	Editar Inhabilitar Rehabilitar
Vacuna	96.00	450.00	85	10	5	Publicado	...

Figura 80. Interfaz donde se muestran todos los productos registrados para poder aplicar una de las tres acciones.

Para realizar el registro de un nuevo servicio, en la sección de “almacén y servicios” se realiza con el formulario que se observa en la figura 81.

contacto@lupware.com

Catálogo

Inicio > Registrar servicio

Registrar nuevo servicio

Completa los campos.

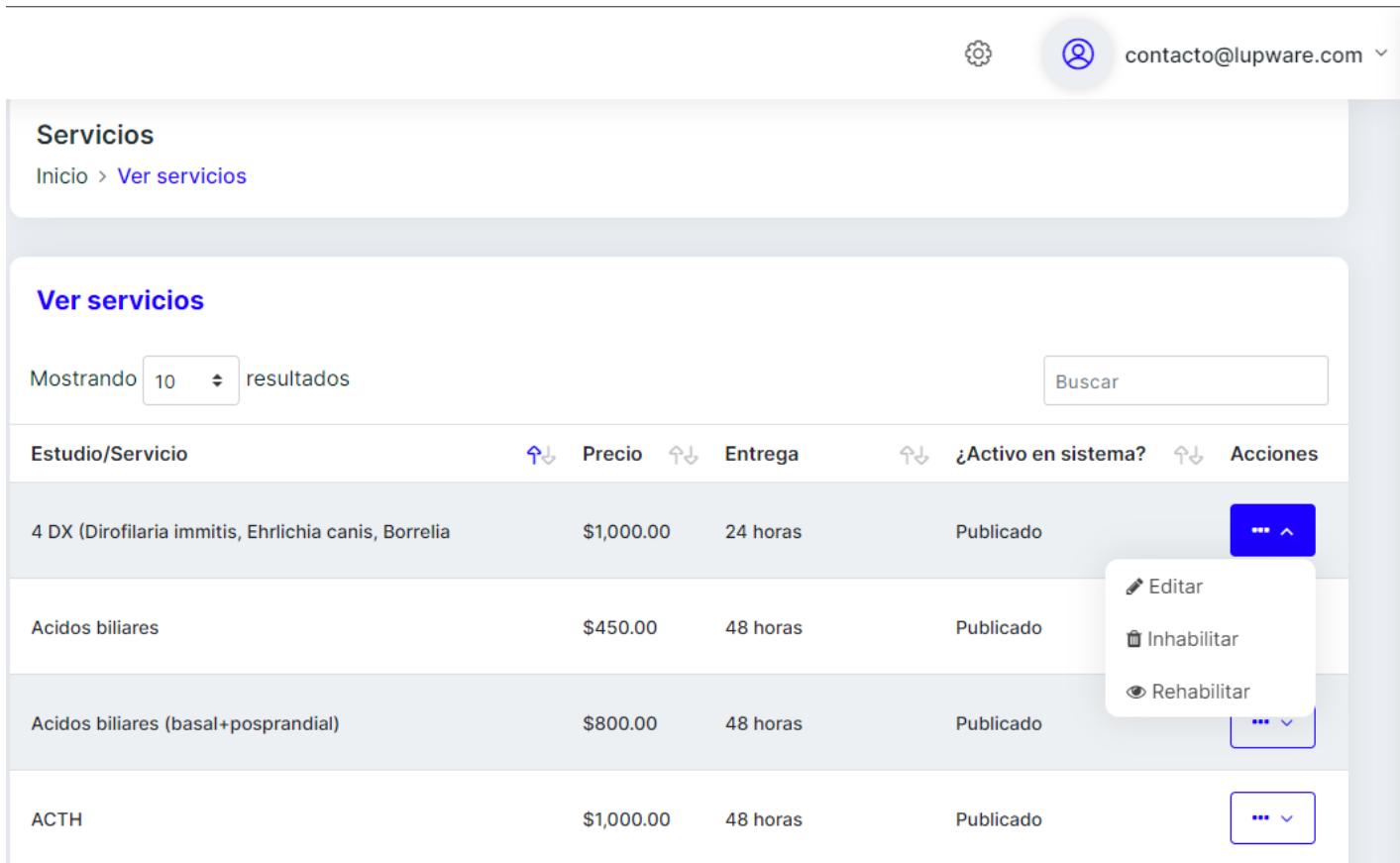
Nombre del servicio	<input type="text" value="Escribe el nombre del estudi"/>	Precio	<input type="text" value="Indica el precio."/>	Entrega	<input type="text" value="N/A"/>
---------------------	---	--------	--	---------	----------------------------------

[Registrar](#)

Desarrollo por [Lupware](#)

Figura 81. Interfaz del formulario para registrar un nuevo servicio.

En el listado de todos los servicios guardados también tiene las opciones de editar (para actualizar uno o más datos), inhabilitar o deshabilitar para acceder en el listado cuando se agrega en la receta, como se aprecia en la figura 82.



The screenshot shows a web application interface for managing services. At the top right, there is a settings gear icon and a user profile icon with the email 'contacto@lupware.com'. Below this is a header section titled 'Servicios' with a breadcrumb 'Inicio > Ver servicios'. The main content area is titled 'Ver servicios' and includes a filter 'Mostrando 10 resultados' and a search box labeled 'Buscar'. A table lists services with columns for 'Estudio/Servicio', 'Precio', 'Entrega', '¿Activo en sistema?', and 'Acciones'. A dropdown menu is open over the 'Acciones' column, showing options: 'Editar', 'Inhabilitar', and 'Rehabilitar'.

Estudio/Servicio	Precio	Entrega	¿Activo en sistema?	Acciones
4 DX (Dirofilaria immitis, Ehrlichia canis, Borrelia)	\$1,000.00	24 horas	Publicado	...
Acidos biliares	\$450.00	48 horas	Publicado	...
Acidos biliares (basal+posprandial)	\$800.00	48 horas	Publicado	...
ACTH	\$1,000.00	48 horas	Publicado	...

Figura 82. Interfaz donde se muestran todos los servicios para realizar una de las tres acciones.

En el sistema se administran los usuarios que podrán acceder, para dar de alta se hace en la sección de “Usuarios (MVZ)” para abrir el formulario que se observa en la figura 83.

The screenshot shows a web interface for user registration. At the top right, there is a settings icon and a user profile icon with the email address 'contacto@lupware.com'. The main content area is titled 'Usuarios (MVZ)' and includes a breadcrumb 'Inicio > Registrar usuario (MVZ)'. Below this is the 'Registrar usuario (MVZ)' section with the instruction 'Completar los campos.' The form consists of five input fields: 'Nombre(s) *' with placeholder 'Nombre(s) del usuario', 'Apellidos *' with placeholder 'Apellido(s) del usuario', 'Usuario (correo electrónico):' with placeholder 'Escribe el usuario (correo electrónico)', 'Contraseña:' with placeholder 'Escribe la contraseña', and 'Confirmar contraseña:' with placeholder 'Confirma la contraseña'. A blue 'Registrar' button is positioned below the fields. At the bottom of the page, it says 'Desarrollo por [Lupware](#)'.

Figura 83. Interfaz del formulario para registrar un nuevo usuario MVZ.

También se puede consultar a todos los usuarios que se tienen registrados en el sistema, como se observa en la figura 84.

contacto@lupware.com

Usuarios (MVZ)

Inicio > [Ver todos los usuarios \(MVZ\)](#)

Lista usuarios (MVZ)

Copiar CSV PDF Imprimir

Nombre	Usuario	¿Activo?	Editar
Betzabeth Mercado	mvz.betzamh@gmail.com	Activo	[Editar](#)
Fanny1 Sandoval1	desarrollos1@lupware.com	Activo	[Editar](#)
Izchel Palacios	izchel.palacios@gmail.com	Activo	[Editar](#)
j k	sistemas@berlingo.com.mx	Activo	[Editar](#)

< 1 >

Figura 84. Interfaz donde muestra a todos los usuarios MVZ registrados en el sistema.

Se selecciona a un usuario de todos los registros para actualizar uno o más datos a través del formulario que se aprecia en la figura 85.

contacto@lupware.com

Usuarios (MVZ)

Inicio > [Editar usuario \(MVZ\)](#)

Editar usuario (MVZ)

Completar los campos

Nombre(s)*:	<input type="text" value="Fanny"/>	Apellido paterno*:	<input type="text" value="Sandoval"/>
Usuario (correo electrónico):	<input type="text" value="desarrollos1@lupware.com"/>	¿Activo?	<input type="text" value="Activo"/>
Contraseña (Vacío si no se va a cambiar):	<input type="password" value="....."/>	Confirmar contraseña:	<input type="password" value="....."/>

Figura 85. Interfaz del formulario para editar los datos del usuario MVZ.

Las recetas salen personalizadas con la firma del veterinario desde la sección de “agregar firma de usuario (MVZ)”, donde se apreciará si ya cuenta con una para su modificación o en su caso desea crearla, como se observa en la figura 86.

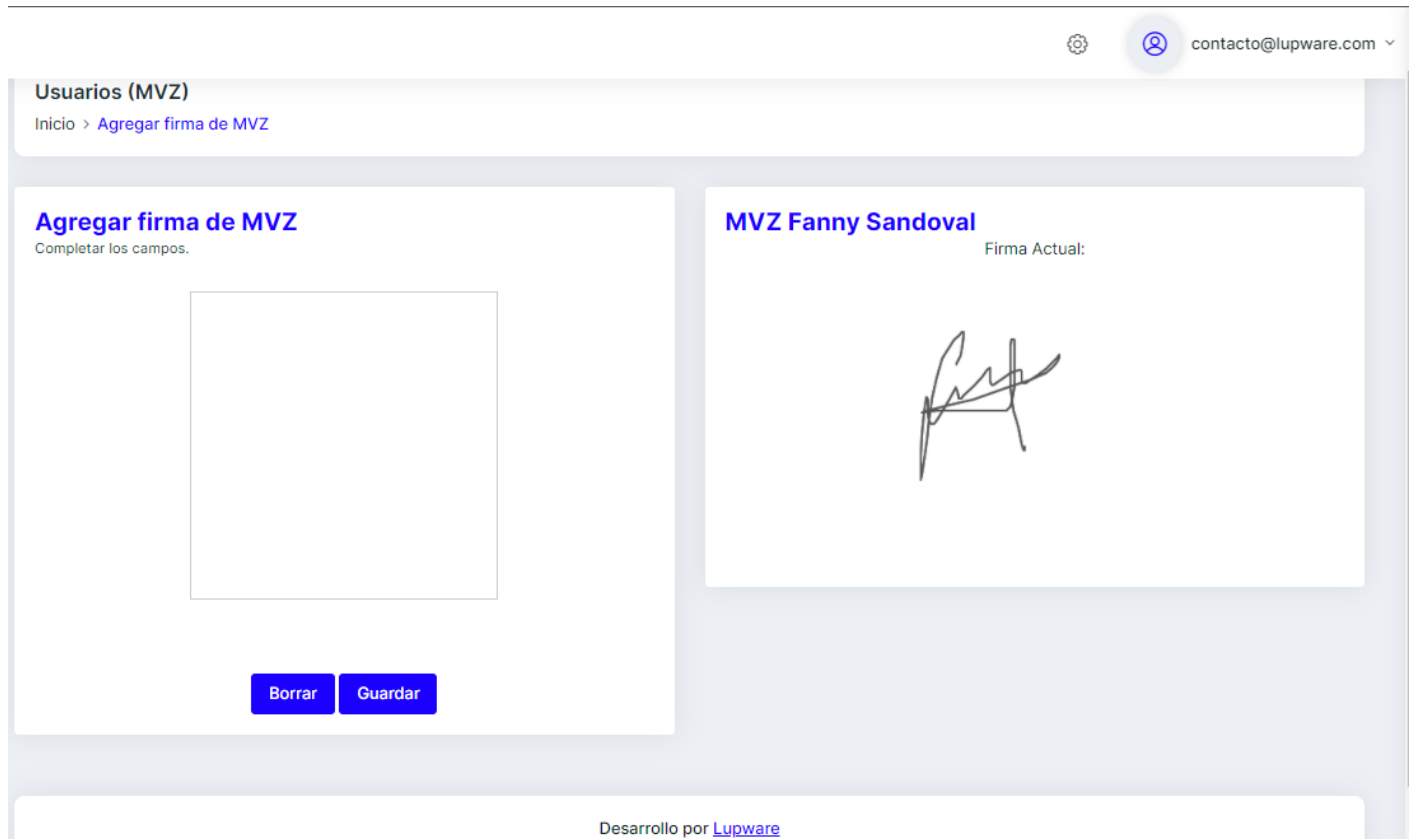


Figura 86. Interfaz para que el usuario MVZ realice y guarde su firma.

Para usar el sistema, el veterinario tendrá que realizar su pago correspondiente en la sección “realizar pago” y se muestre la pantalla que se aprecia en la figura 87.

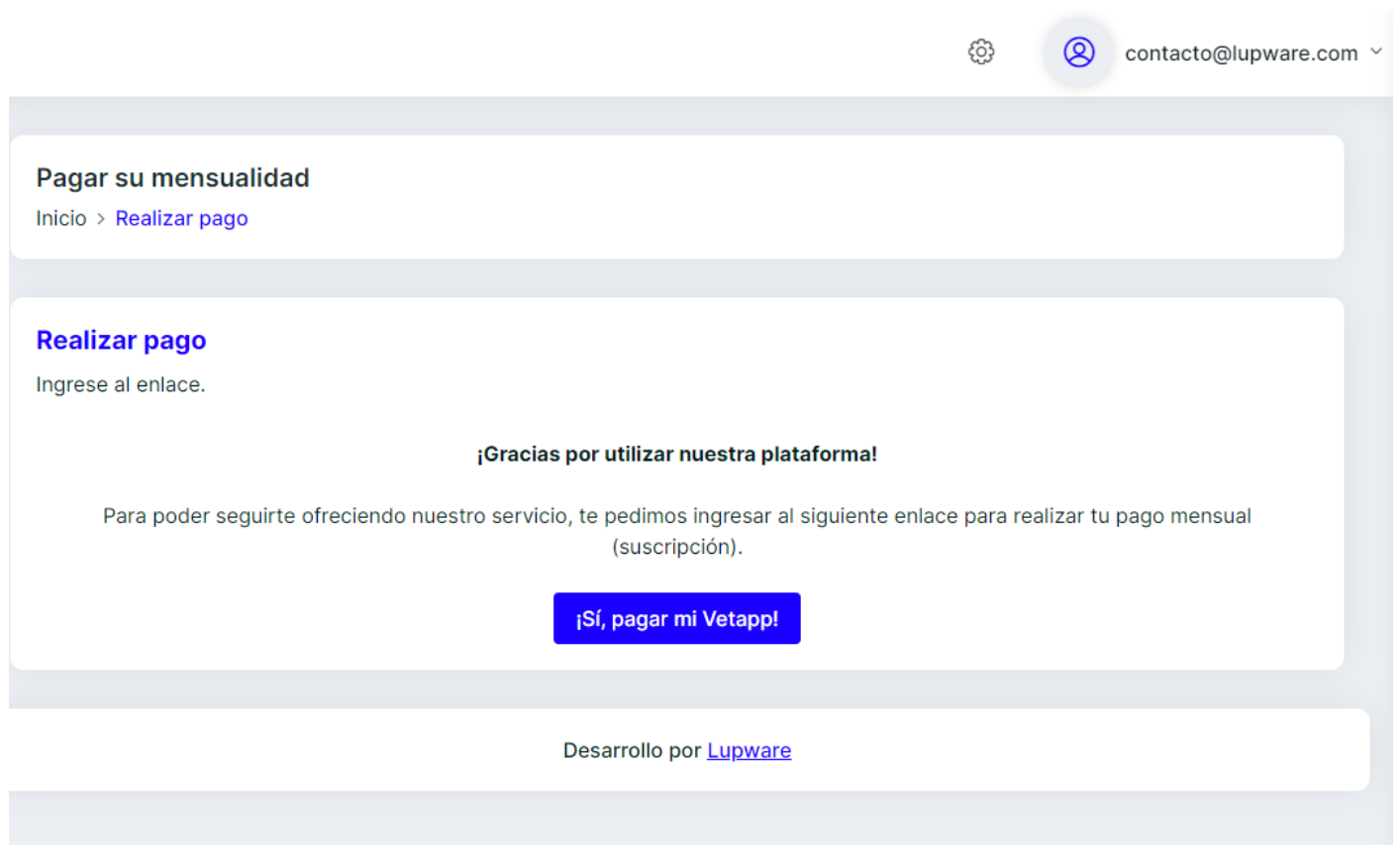


Figura 87. Interfaz para que el usuario MVZ realice su pago para uso del sistema.

En la sección de “respondemos tus dudas”, el veterinario podrá consultarlas por medio de los videos o el manual de usuario que están incluidos en esta sección, como se aprecia en la figura 88.

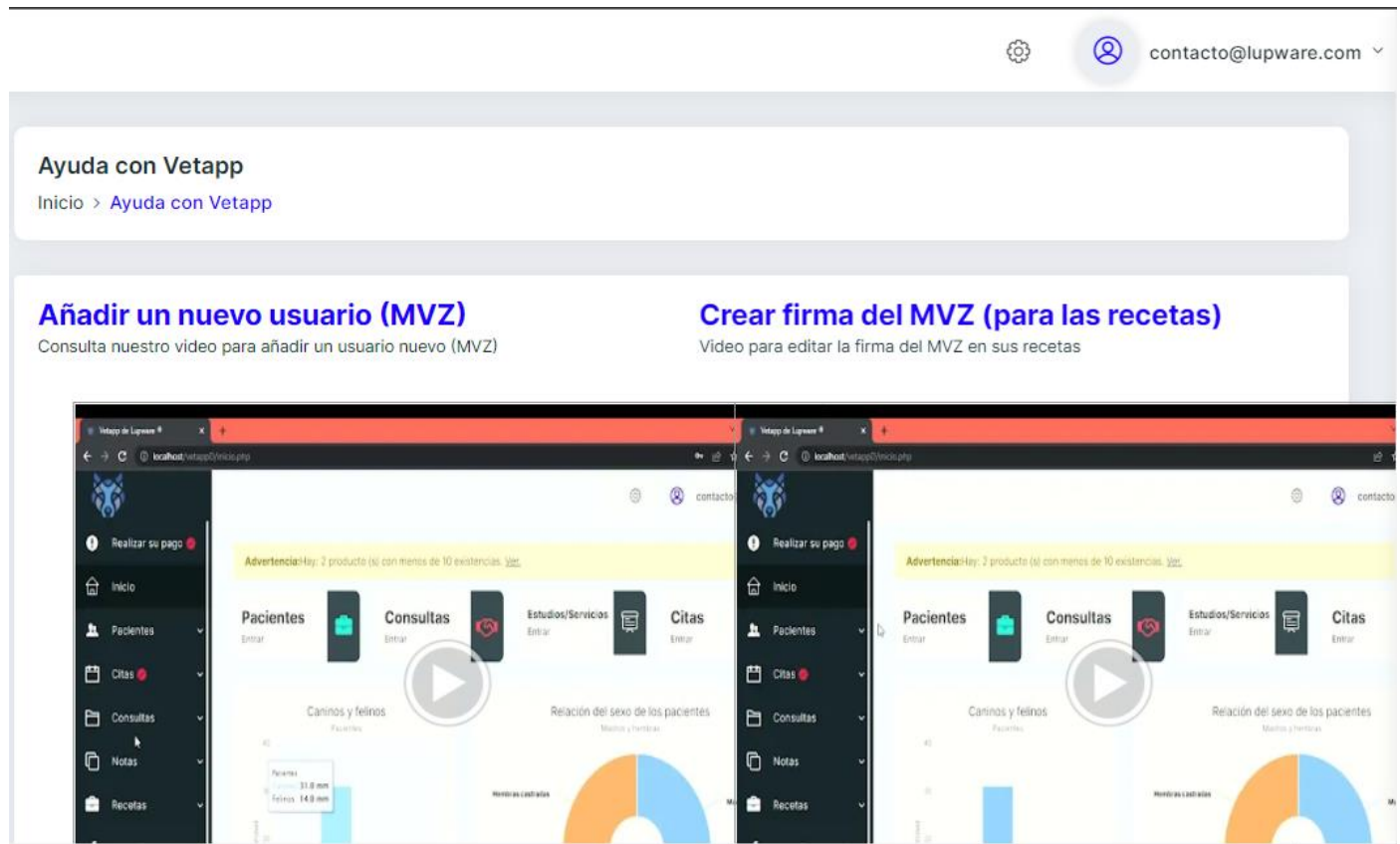


Figura 88. Interfaz para resolver dudas al usuario MVZ de cómo usar el sistema.

CONCLUSIONES.

Con este trabajo damos por concluido que el informático también puede enfocarse en el desarrollo de software en distintos lenguajes de programación después de que tiene una amplia variedad de todas las ramas en las que se puede dedicar, solo es cuestión de buscar acorde a lo que en realidad quiere enfocarse para que así se vaya adquiriendo la experiencia necesaria para el ámbito laboral.

Con esta experiencia se puede percatar que cuando una empresa va iniciando, los colaboradores son un pilar para ellos y se tiene más carga de trabajo, ya que, no son varios y en este caso se debió tener varios roles al mismo tiempo bajo el esquema de trabajador de FreeLancer de poder trabajar en el horario que se quisiera, pero con la mirada hacia enfrente de tener el trabajo en tiempo y forma para evitar conflictos con los

clientes.

Cada que se realiza una página web se van solicitando los requerimientos y uno va adecuando de acuerdo con las necesidades en una de las opciones que se le muestra al cliente para que sea aprobada y en caso de que no, tiene que volverse a hacer siempre y cuando el director haya negociado nuevamente el proyecto para así seguir trabajando en una propuesta.

Se rediseña las interfaces, se ordenan los códigos y se van incluyendo todo el material que se nos envía para darle más forma a la página.

Cuando son sistemas web hay que hacer sus respectivos diagramas, hacer su base de datos en el software, diseñar las interfaces, conectar a la base de datos, programarlas y hacer pruebas para validar que todo esté funcionando y guardando correctamente.

De acuerdo con el plan de estudio de la carrera de informática necesarios para poder desarrollarme en el ámbito profesional he usado los conocimientos que a continuación se describen:

- Análisis y diseño de sistemas I (tercer semestre).
- Análisis y diseño de sistemas II (cuarto semestre).
- Programación IV programación de interfaces (cuarto semestre).
- Introducción a las bases de datos (quinto semestre).
- Desarrollo de aplicaciones de bases de datos (sexto semestre).
- Laboratorio de sistemas de información (séptimo semestre).
- Seminario de programación en internet I (optativa).
- Seminario de programación en internet II (optativa).
- Seminario de comercio electrónico I (optativa).
- Seminario de comercio electrónico II (optativa).

Con lo que se mencionó anteriormente, se comprueba que la carrera está ofreciendo las herramientas necesarias para poder desarrollarse profesionalmente y que con el paso del tiempo y experiencia las va fortaleciendo más y mejor para que así el egresado de informática se interese por esta área de desarrollador web o de sistemas ya que hay variedad de lenguajes con los cuales puede ir formándose para profundizar en sus conocimientos.

BIBLIOGRAFÍA.

- Méndez, P. (26 de marzo de 2020). *¡Senior vs Junior! (Niveles de programadores)*. EDteam. <https://normas-apa.org/referencias/citar-pagina-web/>
- Sánchez, J. (2004). *Diseño conceptual de Bases de Datos guía de aprendizaje*. Creative Commons. [Jorge Sanchez, profesor de informática. Manuales, ejercicios y documentos sobre cursos de informática](#)
- Senn, J. (2000). *Análisis y diseño de sistemas de información*. Mc Graw Hill
- Mateu, C. (2004). *Desarrollo de aplicaciones web*. Universidad de Catalunya.
- Cornejo, A. (2012). *Manual de HTML*. Feedbooks.
- Silberschatz, A., Korth, H. y Sudarshan, S. (2002). *Fundamentos de bases de datos. Cuarta edición*. McGRAW-HILL Inc.
- Corcuera, P. *Bases de Datos Modelo Relacional*. [diapositiva 14]. Dpto. Matemática aplicada y ciencias de la computación. Universidad de Cantabria. https://personales.unican.es/corcuerp/BD/Slides/BD_ModeloRelacional.pdf
- Deyimar, A. (3 de junio de 2022). *¿Qué es Bootstrap? – Una guía para principiantes*. Hostinger. Recuperado el 11 de septiembre de 2022 de <https://www.hostinger.mx/tutoriales/que-es-bootstrap>
- MDN, C. (4 de agosto de 2022). *JavaScript*. MDN Web Docs. Recuperado el 11 de septiembre de 2022 de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- MDN, C. (5 de septiembre de 2022). *Fundamentos de JavaScript*. MDN Web Docs. Recuperado el 11 de septiembre de 2022 de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- Robledano, A. (24 de septiembre de 2019). *Qué es MySQL: Características y ventajas*. OpenWebinars. <https://openwebinars.net/blog/que-es-mysql/>
- Bodnar, D. (28 de enero de 2021). *¿Qué es un navegador web?*. Avast Academy. Recuperado el 11 de septiembre de 2022 de <https://www.avast.com/es-es/c-what-is-a-web-browser>
- Vega, F. (2017). *Qué es un navegador web*. Platzi. Recuperado el 11 de septiembre de 2022 de <https://platzi.com/clases/2042-prework-windows/32460-que-es-el-navegador/>

- MDN, C. (13 de agosto de 2022). *Generalidades del protocolo HTTP*. MDN Web Docs. Recuperado el 11 de septiembre de 2022 de <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>
- E.U. Informática en Segovia, Departamento de informática. (5 de marzo de 2006). *Tema 2: El modelo cliente / servidor*. Universidad de Valladolid. https://www.infor.uva.es/~fdiaz/sd/2005_06/doc/SD_TE02_20060305.pdf
- Microsoft. (2022). Visual Studio Code. <https://code.visualstudio.com/>
- Support, A. (8 de junio de 2022). *Apache HTTP SERVER PROJECT*. La Fundación de Software Apache. Recuperado el 12 de septiembre de 2022 de <https://httpd.apache.org/>
- The Apache Software Foundation. (2022). *Desarrollo liderado por la comunidad "the apache way"*. <https://www.apache.org/>
- VMWare. (2022). *XAMPP Apache + MariaDB + PHP + Perl*. <https://www.apachefriends.org/es/index.html>
- Becker, A. (2017). *HeidiSQL*. <https://www.heidisql.com/>
- Lucidchart. (s.f.). *Tutorial de diagrama de clases UML*. <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>
- Kimmel, P. (2008). *Manual de UML*. Mc Graw Hill.
- Schmuller, J. (1999). *Aprendiendo UML en 24 horas*. Prentice Hall.

Descarga de plantillas para la realización de sistemas:

- <https://bootstrapmade.com/>
- https://themewagon.com/theme_tag/bootstrap-4-templates/