Tabla de contenido

[CAPITULO 1 6](#_Toc26825323)

[SISTEMA DE CONTROL DE MASCOTAS 6](#_Toc26825324)

[1 introducción 6](#_Toc26825325)

[1.2 Antecedentes 6](#_Toc26825326)

[1.2.1 Misión. – 6](#_Toc26825327)

[1.2.2 Visión. – 6](#_Toc26825328)

[1.3 Definición del Problema 7](#_Toc26825329)

[1.4 Identificación del problema 7](#_Toc26825330)

[1.4.1 Formulación del problema 8](#_Toc26825331)

[1.5 Objeto de estudio 8](#_Toc26825332)

[1.6 Objetivos 9](#_Toc26825333)

[1.6.1 Objetivo general 9](#_Toc26825334)

[1.6.2 Objetivos específicos 9](#_Toc26825335)

[1.7 Alcance temático 9](#_Toc26825336)

[1.8 Alcance geográfico 10](#_Toc26825337)

[1.9 Alcance temporal 10](#_Toc26825338)

[1.10 Justificación 10](#_Toc26825339)

[1.10.1 Justificación Social 10](#_Toc26825340)

[1.10.2 Justificación Técnica 10](#_Toc26825341)

[1.10.3 Justificación Académica 11](#_Toc26825342)

[1.11 Técnicas de recolección de información 11](#_Toc26825343)

[1.12 Población de objeto de estudio 11](#_Toc26825344)

[CAPITULO 2 12](#_Toc26825345)

[ANTECEDENTES DEL CENTRO 12](#_Toc26825346)

[2.1 INTRODUCCION 12](#_Toc26825347)

[2.2 MISION 12](#_Toc26825348)

[2.3 VISION 12](#_Toc26825349)

[2.4 CONTROL DE ANIMALES: 13](#_Toc26825350)

[2.4.1 Centro Zoonosis 13](#_Toc26825351)

[CAPITULO 3 14](#_Toc26825352)

[MARCO TEÓRICO 14](#_Toc26825353)

[3.1 INTRODUCCIÓN 14](#_Toc26825354)

[3.2 CICLO DE VIDA 14](#_Toc26825355)

[3.3 ELEMENTOS DEL CICLO DE VIDA 14](#_Toc26825356)

[3.3.1 PROCESO BÁSICO DEL CICLO DE VIDA DE UN SISTEMA 15](#_Toc26825357)

[3.3.1.6 Mantenimiento 17](#_Toc26825358)

[3.4 HERRAMIENTA DE MODELAMIENTO DE LAS FASES DE ANALISIS Y DISEÑO DEL PROYECTO: 17](#_Toc26825359)

[3.4.1 Diagrama de Caso de Uso. 17](#_Toc26825360)

[3.4.1.1Simbolos de Caso de Uso. 18](#_Toc26825361)

[3.4.1.2 Relaciones de Caso de Uso 20](#_Toc26825362)

[3.4.2 DIAGRAMA DE CLASES 20](#_Toc26825363)

[3.4.2.1 Componentes Básicos de Diagrama de Clases 21](#_Toc26825364)

[3.4.2.2 Interacciones 21](#_Toc26825365)

[3.5.3 DIAGRAMA DE COMPONENTES 23](#_Toc26825366)

[3.5.3.1 Descripcion 23](#_Toc26825367)

[3.5.3.2 Notacion 23](#_Toc26825368)

[3.5.3.3 Componentes 23](#_Toc26825369)

[3.5.3.4 Interfaz 24](#_Toc26825370)

[3.5.3.5 Paquete 24](#_Toc26825371)

[3.5.3.6 Relacion de Dependencia 24](#_Toc26825372)

[Ejemplo 24](#_Toc26825373)

[3.6.4 DIAGRAMA DE DESPLIEGUE 25](#_Toc26825374)

[3.6.4.1 Características 25](#_Toc26825375)

[3.6.4.2 Los Elementos que lo Componen son: 26](#_Toc26825376)

[CAPITULO 4 28](#_Toc26825377)

[FASE DE ANALISIS Y DISEÑO 28](#_Toc26825378)

[4 INTRODUCCIÓN: 28](#_Toc26825379)

[4.1 HISTORIAS DE USUARIO DEL PROYECTO 28](#_Toc26825380)

[4.2 MODELO DE CASOS DE USO 29](#_Toc26825381)

[4.2.1 IDENTIFICACIÓN DE ACTORES 29](#_Toc26825382)

[4.2.1.1 Administrador, 30](#_Toc26825383)

[4.2.1.2 Personal, 30](#_Toc26825384)

[Diagrama de caso de uso Sistema Zoonosis 30](#_Toc26825385)

[Caso de Uso Autentificar Usuario 31](#_Toc26825386)

[Caso de Uso Gestion Datos Centro 31](#_Toc26825387)

[4.2.2 DESCRIPCIÓN CASO DE USO: 33](#_Toc26825388)

[Personal 34](#_Toc26825389)

[4.3 Diagrama de Secuencia 35](#_Toc26825390)

[4.3.1 Diagrama de Secuencia Módulo Gestión de Usuario 35](#_Toc26825391)

[4.4 DIAGRAMA DE CLASES 37](#_Toc26825392)

[4.4.1 ULTIMA VERSION DEL DIAGRAMA DE CLASES: 37](#_Toc26825393)

[4.5 DIAGRAMA ENTIDAD RELACION. 38](#_Toc26825394)

[4.6 DIAGRAMA ACTIVIDADES 39](#_Toc26825395)

[4.7 FASE DE CONSTRUCCION 40](#_Toc26825396)

[4.7.1 Herramientas De Implementación 40](#_Toc26825397)

[Servidor Web Apache 40](#_Toc26825398)

[XAMPP: 40](#_Toc26825399)

[4.7.2 Herramientas de Diseño 41](#_Toc26825400)

[Php 41](#_Toc26825401)

[4.7.3 Entorno de Desarrollo 41](#_Toc26825402)

[Visual studio code: 42](#_Toc26825403)

[Sqlyog: 42](#_Toc26825404)

[Sublime Text: 42](#_Toc26825405)

[Adobe Photoshop CS6; 43](#_Toc26825406)

[Navegador Google Chrome; 43](#_Toc26825407)

[Dia. - 44](#_Toc26825408)

[4.8 Implementación De Acceso Al Sistema 44](#_Toc26825409)

[4.8.1 Prueba de caja negra de Acceso al Sistema 45](#_Toc26825410)

[4.8.2 IMPLEMENTACIÓN DEL MÓDULO GESTIÓN DE USUARIOS 47](#_Toc26825411)

[4.8.3 Prueba de caja negra módulo Gestión de Usuarios 48](#_Toc26825412)

[ANEXOS 50](#_Toc26825413)

[ANEXO A – ENCUESTA 50](#_Toc26825414)

**INSTITUTO TECNOLÓGICO “INFOCAL” COCHABAMBA  
CENTRO TUPURAYA  
Carrera de Sistemas Informáticos**

****

**DESARROLLO DE UNA PAGINA WEB DE GESTION DE REGISTRO DE MASCOTAS**

**CASO DE ESTUDIO: CENTRO DE ACOJO DE MASCOTAS**

**ZOONOSIS.**

PERFIL DE PROYECTO DE GRADO

Postulante: Albaro Ticacolque Anguela

Docente: María del Rosario Claure Ch.

Cochabamba – Bolivia

Diciembre , 2019

# CAPITULO 1

# SISTEMA DE CONTROL DE MASCOTAS

# 1 introducción

Las innovaciones tecnológicas, la mayor expectativa de vida, el nivel de información de la sociedad en su conjunto, y por lo tanto la mayor exigencia, justifican la elaboración del presente proyecto que busca implementar el manejo de la información que respalde la documentación manejada en los centros de acojo de mascotas. El mejor aliado para una empresa que se precie de ser la mejor que las demás es siempre estar a la vanguardia en toda herramienta que beneficie a la empresa.

Hoy en día los problemas relacionados con control y registro de animales han causado algunos conflictos en la sociedad como ser mascotas abandonadas, extraviadas, robadas y animales peligrosos, y así también el ataque de canes hacia la población.

El presente proyecto busca obtener diseño e implementación de un sistema web

de gestión y registro para el centro ZOONOSIS perteneciente al municipio de SACABA, dicho sistema ayudara a tener un mejor manejo y seguridad al momento de poder acoger a estas mascotas.

# 1.2 Antecedentes

La perrera Municipal de Sacaba ZOONOSIS, fue creada en septiembre el año 2014, es un centro sin fines de lucro, con personería jurídica propia, ubicada en el barrio Chullpamoko km 8/5 camino a sacaba hacia al sud.

La tarea del centro, es acoger a canes y felinos abandonados, peligrosos y en estados críticos de salud para su posterior cuidado u revisión en caso de ataques a personas estos realizan acciones más drásticas con los animales.

* + 1. Misión. –Evitar la transmisión de las enfermedades zoonóticas al hombre, actualizando y difundiendo las normas de atención de rabia para lograr una óptima vigilancia epidemiológica del programa y pueda ser manejada en todos los niveles de atención de los servicios de salud y así cortar la circulación y control de virus rábicos en la población de sacaba.
    2. Visión. –Controlar la circulación del virus rábico y de los parásitos hacia el ser humano y sensibilizando a la población y brindando a todos los habitantes de la población y mejorando hacia el bienestar y calidad de vida de la población.1

**Tabla 1:**

|  |  |
| --- | --- |
| Título: | Sistema Web del CENTRO DE ZOONOSIS MUNICPAL DEL MUNICIPIO DE CERCADO |
| Autor: | JAVIER RODRIGUEZ |
| Año: | 2010 |
| Centro | ZOONOSIS DEL MUNICIPIO DE CERCADO |
| Descripción: | El proyecto “Sistema Web de Registro y censo de cercado ayuda a su municipio a poder tener un mejor manejo de este centro teniendo registros de las mascotas y sus respectivos reportes ayudando a la población tener una mejor calidad de vida. |

**Fuente:** Elaboración propia

# 1.3 Definición del Problema

Después de haber realizado un análisis de la toma de datos con entrevistas al personal del centro involucrado con las mascotas, se pudo evidenciar los siguientes problemas.

# 1.4 Identificación del problema

Después de realizar un análisis en la empresa utilizando técnicas de recolección de datos se pudo evidenciar los siguientes sub-problemas:

* No existe un sistema de registro digitalizado, lo que lleva a perder el tiempo en cuando se trata de buscar alguna mascota registrada manualmente de registros pasados y no tiene un buen registro.
* Falta de evidencia de la documentación registrada para canes y felinos, lo que lleva a no poder tener una capitulación y la transparencia del centro.
* No existe reportes de las mascotas de acuerdo a la (Raza, color y otros), por lo cual no saben cómo separarlos.

**Figura 1**: Árbol de Problemas.

(Ref. Elaboración Propia)

EFECTOS

PROBLEMAS

¿De qué manera se puede mejorar la eficacia del centro de acojo de mascotas Zoonosis sacaba perrera municipal, sí actualmente utilizan medios clásicos está llevando al centro a saturarse y la falta de documentación en el trabajo?

CAUSAS

No existe un sistema de registro digitalizado.

Falta de evidencia de la documentación registrada para canes y felinos

No existe reportes de las mascotas de acuerdo a la (Raza, color y otros).

Lo que lleva a perder el tiempo en cuando se trata de buscar alguna mascota registrada manualmente de registros pasados y no tiene un buen registro.

Lo que lleva a no poder tener una capitulación y la transparencia del centro.

Por lo cual no saben cómo separarlos.

## 1.4.1 Formulación del problema

¿De qué manera se puede mejorar la eficacia del centro de acojo de mascotas Zoonosis sacaba perrera municipal, sí actualmente utilizan medios clásicos está llevando al centro a saturarse y la falta de documentación en el trabajo?

# 1.5 Objeto de estudio

El Objeto de estudio es el Proceso de registro de mascotas del centro de Zoonosis.

# 1.6 Objetivos

## 1.6.1 Objetivo general

Desarrollo de una página web de gestión de registro de mascotas la cual mejore la eficacia y seguridad al momento de presentar informes de todas las tareas realizadas en el centro.

## 1.6.2 Objetivos específicos

* Hacer un análisis de diagnóstico sobre los procesos fundamentales para identificar los principales procesos en el centro zoonosis y de esa manera obtener los requerimientos.
* Identificar la tecnología y herramientas y adecuadas para el desarrollo del proyecto de software de acuerdo a los requerimientos.
* Bosquejar e Implementar los módulos necesarios para cumplir los requerimientos de automatizar el proceso de registro.
* efectuar las pruebas pertinentes para validar el proyecto de software.

# 1.7 Alcance temático

Para el presente proyecto se elaborarán los siguientes módulos:

* **Módulo de Control de Usuarios, mediante uso de privilegios en la gestión de usuarios autorizados.**

Este módulo ayudará al control de accesos de usuario y administrador permitidos por el sistema, el administrador podrá manejar la configuración básica de todos los campos importantes, tanto el registro, respaldos y reportes y el usuario tendrá tareas específicas.

* **Módulo de registro de animales.**

Ayudará al registro de toda clase de rasas para canes y felinos, también los tipos de adopción.

* **Módulo para Respaldo de información mediante una base de datos dedicada al sistema.**

El sistema tendrá un respaldo mediante una base de datos (registro, tipos de mascotas, adopción, reportes).

* **Módulo de usuarios y contraseñas.**

Se tendrá que asignar tipos usuarios y contraseñas de acuerdo al usuario con la cual podrá loquear al sistema.

* **Modulo para reportes específicos de usuarios.**

Este módulo de reportes nos ayudara a sacar tipos de reportes como ser (reporte por fecha, reporte por raza, reporte nombre del dueño, reporte por año, reporte de vacunas, reporte por nombre de la mascota).

# 1.8 Alcance geográfico

Para la demostración del proyecto, se lo realizara en un servidor local, pero se tiene previsto la implantación futura en un servidor web donde se estará alojado, y el servicio estará disponible para todos los centros de zoonosis de la población de sacaba.

## 1.9 Alcance temporal

El proyecto tendrá un lapso de 8 meses, el tiempo que se realizará el análisis de la investigación y desarrollo de la misma.

# 1.10 Justificación

## 1.10.1 Justificación Social

El planteamiento del proyecto de grado responde a una necesidad del centro de acogimiento de mascotas ZOONOSIS de Sacaba. Se mejorará el tipo de manejo de registro de mascotas, dejando de esta manera más eficiente y eficaz en las tareas que realiza el centro así también contará con el apoyo a la población y la responsabilidad para poder criar mascotas. Y que el interesado que adoptara pueda contar con espacios requeridos de acuerdo a la raza y tenga el criterio de realizar sus registros y cuidado de salud del can o felino.

## 1.10.2 Justificación Técnica

Se cuenta con la tecnología necesaria a disposición para realizar todo el desarrollo del proyecto y su posterior implementación utilizando la tecnología Web para el desarrollo y la implementación del sistema en general así mismo se integrará diferentes herramientas complementarias para el desarrollo del proyecto.

## 1.10.3 Justificación Académica

Para cumplir este campo se pretende elaborar el proyecto de factibilidad para el centro de acojo de mascotas usando las herramientas como base de datos, programación web, programación orientada a objetos, otros. Y a la vez poner en práctica los conocimientos adquiridos y el documento resultante también servirá de documento de consulta para otros proyectos del área.

# 1.11 Técnicas de recolección de información

* + - Entreviste con preguntas a los propietarios del centro zoonosis.
    - También con preguntas se entrevistó al ing. Rita Rocha Machado encargado del centro.

# 1.12 Población de objeto de estudio

* + - Beneficio hacia la población de Sacaba.
    - Beneficio para el centro.

# CAPITULO 2

# ANTECEDENTES DEL CENTRO

* 1. INTRODUCCION

El centro ZOONOSIS de sacaba del distrito 4.

En nuestro país como en otros existen el amor a los perros, pero la diferencia con los otros países que en Bolivia llego hacer un negocio lucrativo y en particular con los canes.

El centro Zoonosis actualmente atiende, a todos los distritos en la provincia de sacaba. La labor desplegada del centro Zoonosis es constante en toda sacaba, para la atención de control y tenencia de mascotas, control de canes, control de crianzas de animales, y control de vectores.

Es el área encargada de desarrollar actividades de vigilancia, prevención y control de enfermedades zoonóticas.

Al centro de Zoonosis realiza diferentes actividades, no solo trabaja como “perrera municipal”.

Realiza diferentes actividades, a nivel departamental como ser.

* Realiza control y tenencia y protección de mascotas.
* Casos de crueldad contra los animales
* Abusos a los animales
* Abandono por parte de los dueños
* Adopciones
* Atención veterinaria

# 2.2 MISION

Proteger la salud de la población previniendo la transmisión de las enfermedades de los animales al hombre (zoonosis) las enfermedades transmitidas a través de los alimentos (ETAs), así como la prevención y control del impacto nocivo de los roedores y animales ponzoñosos, con criterio técnico y científico.

# 2.3 VISION

Proporcionar los conocimientos de las ciencias veterinarias a fin de proteger la salud del hombre previniendo el impacto nocivo de las enfermedades (zoonosis), así como preservar el ambiente del efecto de la tenencia de animales.

# 2.4 CONTROL DE ANIMALES:

Las actividades de prevención y de control de las enfermedades animales por los Servicios Veterinarios en todo el mundo representan un Bien Público Mundial. Estas actividades aportan beneficios considerables para la producción agrícola y la seguridad y la inocuidad alimentarias, la salud pública, el bienestar animal, el acceso a los mercados y la reducción de la pobreza rural. La eficacia de las políticas de prevención radica en la buena gobernanza y la calidad de los Servicios Veterinarios, cuya conformidad con las normas de la OIE es un requisito previo indispensable.

## **2.4.1 Centro Zoonosis**

Desde años el hombre y el perro tiene un vínculo muy grande, y no tomar a este animal enserio puede traer consecuencias.

Se conoce como zoonosis, a las enfermedades de los animales que se transmiten al hombre.

El centro de zoonosis, está encargada de la prevención y control de estas enfermedades, Zoonóticas que superan un número mayor de 160, como ser:

# CAPITULO 3

# MARCO TEÓRICO

# INTRODUCCIÓN

Es una de las fases más importantes de un trabajo de investigación, consiste en desarrollar la teoría que va a fundamentar el proyecto con base al planteamiento del problema que se ha realizado. Existen numerosas posibilidades para elaborarlo, la cual depende de la creatividad del investigador. Una vez que se ha seleccionado el tema objeto de estudio y se han formulado las preguntas que guíen la investigación, el siguiente paso consiste en realizar una revisión de la literatura sobre el tema. Esto consiste en buscar las fuentes documentales que permitan detectar, extraer y recopilar la información de interés para construir el marco teórico pertinente al problema de investigación planteado.

# 3.2 CICLO DE VIDA

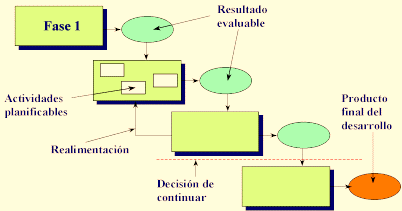
Todo proyecto de ingeniería tiene unos fines ligados a la obtención de un producto, proceso o servicio que es necesario generar a través de diversas actividades. Algunas de estas actividades pueden agruparse en fases porque globalmente contribuyen a obtener un producto intermedio, necesario para continuar hacia el producto final y facilitar la gestión del proyecto. Al conjunto de las fases empleadas se le denomina “ciclo de vida”.

De la misma forma, la práctica acumulada en el diseño de modelos de ciclo de vida para situaciones muy diversas permite que nos beneficiemos de la experiencia adquirida utilizando el enfoque que mejor se adapte a nuestros requerimientos.

# 3.3 ELEMENTOS DEL CICLO DE VIDA

Un ciclo de vida para un proyecto se compone de fases sucesivas compuestas por tareas panificables. Según el modelo de ciclo de vida, la sucesión de fases puede ampliarse con bucles de realimentación, de manera que lo que conceptualmente se considera una misma fase se pueda ejecutar más de una vez a lo largo de un proyecto, recibiendo en cada pasada de ejecución aportaciones de los resultados intermedios que se van produciendo (realimentación).

**Fig. 2** Ciclo de vida



(**Ref.** [http://www.spw.cl/proyectos/apuntes2/images/cvida1.gif](http://www.spw.cl/proyectos/apuntes2/images/cvida1.gif%20VISITA:%2008/SEP/2018))

## 3.3.1 PROCESO BÁSICO DEL CICLO DE VIDA DE UN SISTEMA

Si bien, es cierto que existen diversas metodologías y formas de desarrollar software, la realidad es que hay modelos tan antiguos que ya son como básicos al momento del ciclo de vida de un software. Un ejemplo de esto, es el modelo en cascada para el proceso de desarrollo de un sistema, con el cuál veremos a ciencia cierta el proceso básico, del cual muchos modelos más se empezarán a desarrollar.

**3.3.1.1**  Planificación. El primer punto importante en el ciclo de vida de software, es analizar brevemente los requerimientos que el cliente pide para la elaboración del sistema que necesita. Esta etapa requiere se cierto conocimiento para poder entender la idea que el cliente propone, además de que regularmente debes tomar nota con cada uno de los puntos importantes que se te solicitan, de este modo puedes hacer una planificación al momento y llegar incluso a determinar los tiempos de desarrollo que te llevará, antes de proceder a entregar el producto final. Un punto importante por el cual la planificación siempre debe estar en los ciclos de vida del software. Es porque el cliente se imagina su producto final de una forma tan abstracta, que necesitas hacer que ponga los pies en el suelo para obtener resultados que se acerquen más a la realidad.

**3.3.1.2** Implementación**.** Una vez que hemos platicado con el cliente y tenemos lo que es un análisis de requerimientos, necesidades y funcionalidades por parte de una aceptación en ambas partes, entonces procedemos con lo que es el ciclo de vida de desarrollo de software. Para este punto, existen una infinidad de metodologías de desarrollo de software, que nos ofrecen la posibilidad de trabajar de distintas formas. Más adelante hablaremos más específicamente de ellas, sin embargo, la implementación, es básicamente la parte donde los programadores empiezan a codificar o desarrollar el sistema que se necesita, básicamente se trata del ciclo de vida del desarrollo de sistemas, sin importar el lenguaje de programación mediante el cual se vayan a elaborar.

**3.3.1.3** Pruebas**.** Una vez que el sistema se va desarrollando, es importante para el ciclo de vida del desarrollo del software, que se realicen ciertas pruebas conforme se vaya avanzando. La idea es que no se termine el desarrollo para poder hacer pruebas, si no que mucho antes, durante el proceso de creación, estas ya se puedan ir ejecutando. Las pruebas nos van a permitir ver si el sistema que se está desarrollando es funcional, si tiene algunos errores, si le faltan ciertas cosas para funcionar correctamente, pues básicamente para avanzar al siguiente punto del ciclo de desarrollo de software, será necesario haber pasado las pruebas correctamente.

**3.3.1.4** Documentación**.** Muchas metodologías de lo que es el ciclo de vida software, van creando documentación, conforme se va avanzando en el desarrollo del sistema. Sin embargo, algunas otras prefieren no hacer la documentación hasta el final. Ahora sí que sea cual sea la metodología que elijas, la documentación siempre será importante, pues considera que no siempre vas a estar tú y tu equipo disponibles y cuando otro equipo llegue a programar lo que ustedes hicieron, será indispensable que haya una documentación de la cual se puedan basar, para poder empezar a desarrollar nuevamente el sistema incompleto.

**3.3.1.5** Despliegue**.** Ya casi llegando a lo que son las últimas etapas del desarrollo de software, nos encontramos con el Despliegue. Este no es otra cosa, más que el momento en que el sistema ya está terminado y ha sido aprobado para que se elabore el producto final. Ahora será el momento de distribuirlo y celebrar, pues gracias al equipo de trabajo es como se habrá llegado a esta fase. Lamentablemente, de las etapas de desarrollo de software, esta es a la cual muchos nunca llegan. Pues una gran cantidad de software incompleto se queda en el camino debido a distintos puntos o motivos. Puede ser que el equipo no se unió, el cliente declinó, el proyecto no fue funcional, etc. Así que, de haber llegado a esta fase de desarrollo de software, tu como tu equipo deberá sentirse orgullosos y es momento de volver a desarrollar un proyecto más.

* + - 1. Mantenimiento**.** La última de las fases del desarrollo de software, es el mantenimiento. Que creías, que nunca más verías al software que hicieron, terminaron y distribuyeron. Pues claro que, si lo volverías a ver, pues es momento de darle mantenimiento. Acá además se pueden agregar lo que son las actualizaciones, dependiendo del tipo de desarrollo. Si el equipo siguió trabajando con el software desarrollado y encontraron formas de hacerle mejoras, entonces parte del mantenimiento será actualizarlo a la versión final en todo momento.

# 3.4 HERRAMIENTA DE MODELAMIENTO DE LAS FASES DE ANALISIS Y DISEÑO DEL PROYECTO:

**Lenguaje Unificado de Modelado – UML**

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

## 3.4.1 Diagrama de Caso de Uso.

“Los diagramas de caso de uso son uno de los cinco tipos de diagramas en UML para modelar aspectos dinámicos de sistemas (diagramas de actividad, diagramas de estados, diagramas de secuencia y diagramas de colaboración son otros cuatro tipos de diagramas en UML para modelar los aspectos dinámicos de un sistema). Los diagramas de casos de uso son importantes para modelar el comportamiento de un sistema, un subsistema o una clase. Cada uno muestra un conjunto de casos de uso, actores y sus relaciones. ” [[1]](#footnote-2)

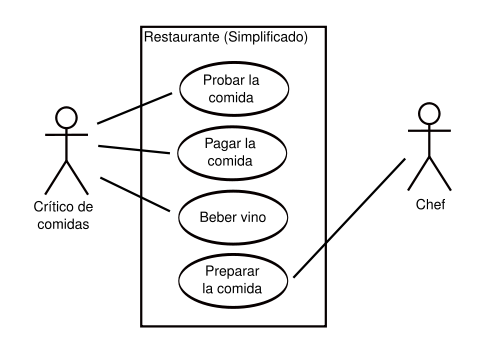
Se aplican los diagramas de casos de uso para modelar las vistas de casos de uso de un sistema. Para la mayor parte, esto involucra el modelado el contexto de un sistema, subsistema, o clase, o modelar las necesidades del comportamiento de esos elementos.

Los diagramas de casos de uso son importantes para visualizar, especificar, y documentar el comportamiento de un elemento. Ellos hacen sistemas, subsistemas, y clases entendibles para presentar una vista exterior de cómo estos elementos pueden ser usados dentro del contexto.

### 3.4.1.1Simbolos de Caso de Uso.

**3.4.1.1.1 Sistemas:**El rectángulo representa los límites del sistema que contiene los casos de uso. Los actores se ubican fuera de los límites del Sistema.

**Fig**. 8 símbolos de caso de uso

**[](https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/3.png)**

( **Ref** .<https://upload.wikimedia.org/wikipedia/commons/thumb/8/80/UML_diagrama_caso_de_uso.svg/747px-UML_diagrama_caso_de_uso.svg.png> )

**3.4.1.1.2 Caso de Uso:** Se representan con óvalos. La etiqueta en el óvalo indica la función del sistema.

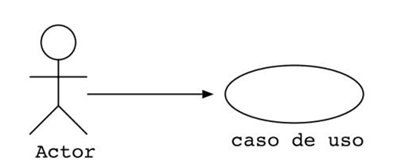
**Fig**. 9 Casos de Uso

**[](https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/4.png)**

**(Ref**.<https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/5.png>)

#### **3.4.1.1.3 Actor:** Un diagrama de caso de uso contiene los símbolos del actor y del caso de uso, junto con líneas conectoras. Los actores son similares a las entidades externas; existen fuera del sistema. El término actor se refiere a un rol específico de un usuario del sistema.

**Fig**. 10 Diagrama que contiene al actor y el caso de uso

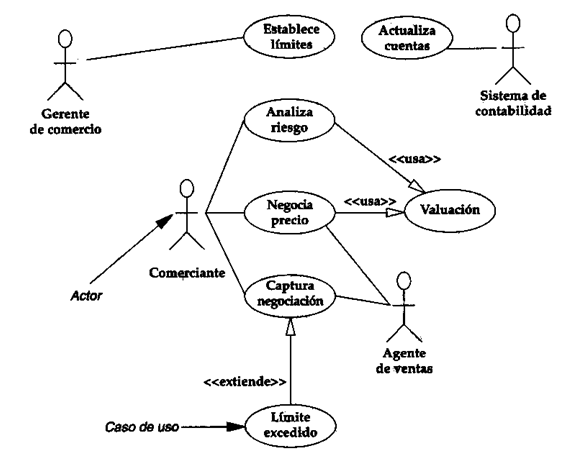
**[](https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/5.png)**

(**Ref**.<https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/5.png> )

#### **3.4.1.1.4 Relaciones**

Las relaciones entre un actor y un caso de uso, se dibujan con una línea simple. Para relaciones entre casos de uso, se utilizan flechas etiquetadas “incluir” o “extender.” Una relación “incluir” indica que un caso de uso es necesitado por otro para poder cumplir una tarea. Una relación “extender” indica opciones alternativas para un cierto caso de uso.

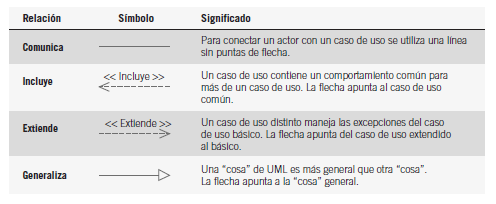
**Fig**. 11 Ejemplo de Casos de Uso

**[](https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/esta.png)**(**Ref**.<https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/esta.png?w=646>)

## 3.4.1.2 Relaciones de Caso de Uso

Las relaciones activas se conocen como relaciones de comportamiento y se utilizan principalmente en los diagramas de casos de uso. Hay cuatro tipos básicos de relaciones de comportamiento: comunica, incluye, extiende y generaliza.

***Fig****. 12 Relaciones entre los Casos de Uso*

[](https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/8.png)

(**Ref**.<https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/8.png> )

## 3.4.2 DIAGRAMA DE CLASES

“[En ingeniería de software](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software), un **diagrama de** es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.”[[2]](#footnote-3)

Los diagramas de clases ofrecen una serie de beneficios para toda organización. Usa los diagramas de clases UML para:

* Ilustrar modelos de datos para sistemas de información, sin importar qué tan simples o complejos sean.
* Comprender mejor la visión general de los esquemas de una aplicación.
* Expresar visualmente cualesquier necesidades específicas de un sistema y divulgar esa información en toda la empresa.
* Crear diagramas detallados que resalten cualquier código específico que será necesario programar e implementar en la estructura descrita.
* Ofrecer una descripción independiente de la implementación sobre los tipos empleados en un sistema que son posteriormente transferidos entre sus componentes.

### 3.4.2.1 Componentes Básicos de Diagrama de Clases

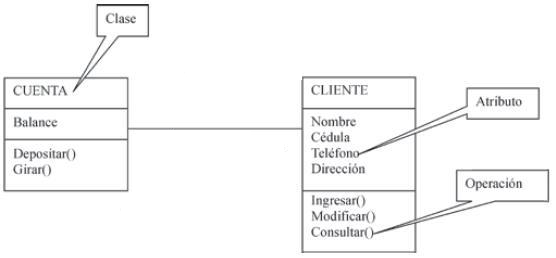
El diagrama de clases estándar está compuesto por tres partes:

**3.4.2.1.1 Sección superior:** Contiene el nombre de la **clase**. Esta sección siempre es necesaria, ya sea que estés hablando del clasificador o de un objeto.

**3.4.2.1.2** **Sección central:** Contiene los **atributos** de la clase. Usa esta sección para describir cualidades de la clase. Esto solo es necesario al describir una instancia específica de una clase.

**3.4.1.1.3** **Sección inferior:** Incluye **operaciones** de clases (métodos). Esto está organizado en un formato de lista. Cada operación requiere su propia línea. Las operaciones describen cómo una clase puede interactuar con los datos.

**Fig**. 13 ejemplo de clases



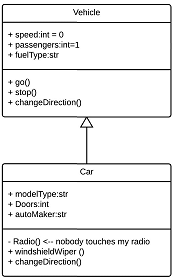
(**Ref**. Elaboración Propia)

### 3.4.2.2 Interacciones

El término "interacciones" se refiere a múltiples relaciones y enlaces que pueden existir en diagramas de objetos y de clases. Algunas de las interacciones más comunes incluyen:

3.4.2.2.1 HERENCIA: El proceso en el que una subclase o clase derivada recibe la funcionalidad de una superclase o clase principal, también se conoce como "generalización". Se simboliza mediante una línea de conexión recta con una punta de flecha cerrada que señala a la superclase.

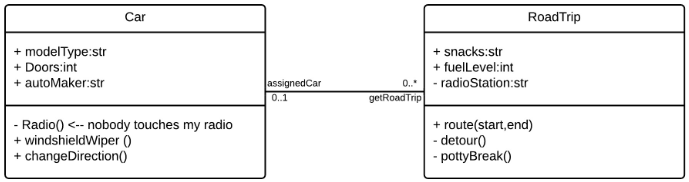
**Fig**. 14 Ejemplo de herencia



(**Ref**.https://d2slcw3kip6qmk.cloudfront.net/marketing/pages/chart/uml/class-diagram/class-diagram-inheritance-175x279.PNG)

3.4.2.2.2 ASOCIACION BIDIRECCIONAL La relación predeterminada entre dos clases. Ambas clases están conscientes una de la otra y de la relación que tienen entre sí. Esta asociación se representa mediante una línea recta entre dos clases.

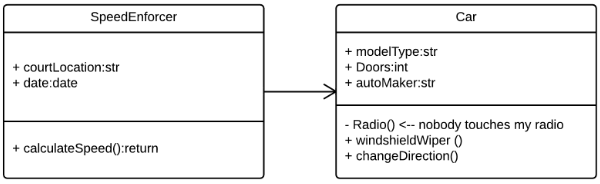
**Fig**. 15 Ejemplo de asociación bidireccional



(**Ref**.https://d2slcw3kip6qmk.cloudfront.net/marketing/pages/chart/uml/class-diagram/class-diagram-bi-directional-association-689x182.PNG)

3.4.2.2.3 ASOCIACION UNIDIRECCIONAL: Una relación un poco menos común entre dos clases. Una clase está consciente de la otra e interactúa con ella. La asociación unidireccional se dibuja con una línea de conexión recta que señala una punta de flecha abierta desde la clase "knowing" a la clase "known".

**Fig**. 16 Ejemplo Asociación unidireccional



(**Ref**.https://d2slcw3kip6qmk.cloudfront.net/marketing/pages/chart/uml/class-diagram/class-diagram-unidirectional-association-600x184)

## 3.5.3 DIAGRAMA DE COMPONENTES

El diagrama de componentes proporciona una visión física de la construcción del sistema de información. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos.

### 3.5.3.1 Descripcion

Como ya se ha indicado, los elementos de estos diagramas son los componentes software y las dependencias entre ellos.

“Un componente es un módulo de software que puede ser código fuente, código binario, un ejecutable, o una librería con una interfaz definida. Una interfaz establece las operaciones externas de un componente, las cuales determinan una parte del comportamiento del mismo. Además se representan las dependencias entre componentes o entre un componente y la interfaz de otro, es decir uno de ellos usa los servicios o facilidades del otro.”[[3]](#footnote-4)

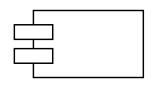
Estos diagramas pueden incluir paquetes que permiten organizar la construcción del sistema de información en subsistemas y que recogen aspectos prácticos relacionados con la secuencia de compilación entre componentes, la agrupación de elementos en librerías, etc.

### 3.5.3.2 Notacion

### 3.5.3.3 Componentes

Un componente se representa como un rectángulo, con dos pequeños rectángulos superpuestos perpendicularmente en el lado izquierdo.

**Fig**. 17 Componente rectangular



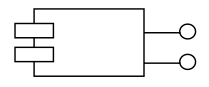
(**Ref**.http://manuel.cillero.es/wp-content/uploads/2013/11/componente.png?x58486 )

Para distinguir distintos tipos de componentes se les puede asignar un estereotipo, cuyo nombre estará dentro del símbolo: << ... >>

### 3.5.3.4 Interfaz

Se representa como un pequeño círculo situado junto al componente que lo implementa y unido a él por una línea continua. La interfaz puede tener un nombre que se escribe junto al círculo. Un componente puede proporcionar más de una interfaz.

**Fig**. 18 Interfaz representada con un pequeño circulo



(**Ref**. http://manuel.cillero.es/wp-content/uploads/2013/11/interfaz.png?x58486 )

### 3.5.3.5 Paquete

Un paquete se representa con un icono de carpeta.

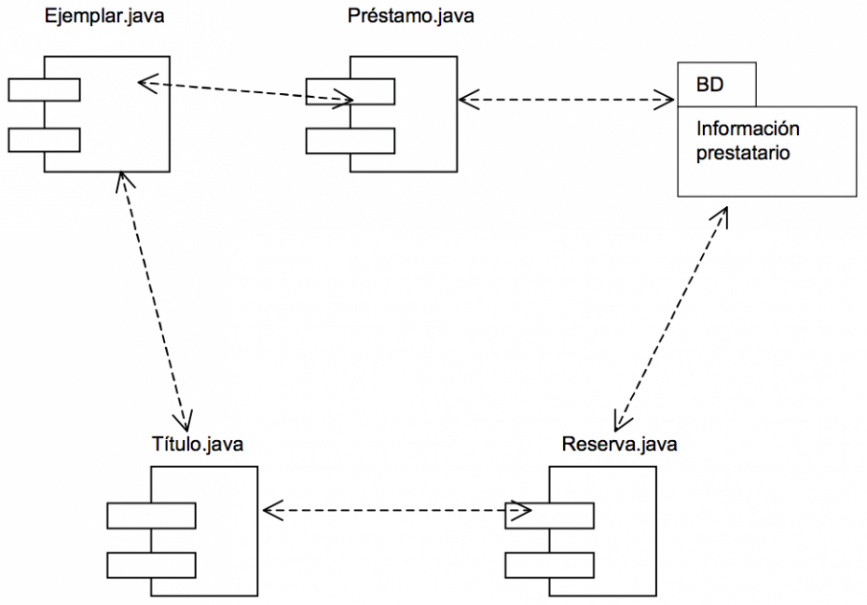
### 3.5.3.6 Relacion de Dependencia

Una relación de dependencia se representa mediante una línea discontinua con una flecha que apunta al componente o interfaz que provee del servicio o facilidad al otro. La relación puede tener un estereotipo que se coloca junto a la línea, entre el símbolo: <<...>>

### Ejemplo

Sistema encargado de la gestión de los préstamos y reservas de libros y revistas en una biblioteca. El lenguaje de desarrollo será Java, y los accesos a la información del prestatario serán mediante un paquete de Base de Datos.

**Fig**. 19 Esta notación es la más habitual, pero MÉTRICA Versión 3 no exige su utilización

[](http://manuel.cillero.es/wp-content/uploads/2013/11/ejemplo-diagrama-de-componentes.png?x58486)

(**Ref**.http://manuel.cillero.es/wp-content/uploads/2013/11/ejemplo-diagrama-de-componentes.png?x58486

## 3.6.4 DIAGRAMA DE DESPLIEGUE

“El diagrama de despliegue permite mostrar la arquitectura en tiempo de ejecución del sistema respecto al hardware y software. Este se utiliza en el diseño y la implementación y se pueden distinguir componentes y nodos, así como las relaciones entre estos.” [[4]](#footnote-5)

* La arquitectura lógica del diagrama de despliegue es la siguiente:
* Capa logica de presentacion y de aplicaciones en la computadora del cliente, en su almacenamiento o en su servicio.
* La presentacion en la computadora del cliente, la logica de aplicaciones en un servidor de la aplicación y el almacenamiento en un servidor de los subsistemas se ira realizando de manera distribuida cada vez mas.

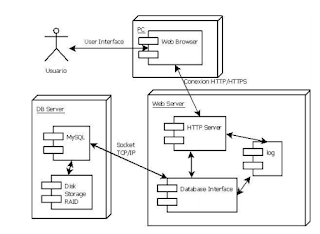
### 3.6.4**.1 Características**

* Utilizando en las etapas del diseño de proceso.
* Se refinan a lo largo del proceso de desarrollo.
* Utiles en el proceso de instalacion del sistema.

Este solo presenta la estructura del sistema solo en tiempos de ejecucion pero no en tiempo de desarrollo o compilacion. Se utiliza mucho para la aplicación distribuidas u online.

* Modela aspectos fisicos de un sistema.
* Modela la vista de despliegue estatica de un sistema.
* Modela una configuracion de nodos y los componentes que residen en ellos.
* Modela la topologia del hardware donde se ejecuta el sistema.

**Fig**. 20 Ejemplo diagrama de despliegue



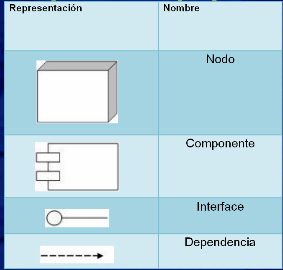
(**Ref**.http://3.bp.blogspot.com/-pIskqASgsbw/VbXaR3ixboI/AAAAAAAAA9I/fVwToVHpVQY/s320/CAS.png

### 3.6.4.2 **Los Elementos que lo Componen son:**

* Relaciones de dependencia, generalizando,asociaacion y realizacion.
* Pueden contener los componentes que residen en los nodos.

**NODOS:** Representan objetos físicos existentes en tiempo de ejecución, sirven para modelar recursos que tiene la memoria y capacidad de proceso. Los nodos se representan mediante paralelepípedos rectangulares.   
  
**3.6.4.3** Relaciones de Dependencia: Entre los nodos se establecen relaciones que significan que existen comunicación entre estos. Se representan mediante líneas continuas y se puede hacer con un estereotipo que indica el tipo de comunicación.  
  
**3.6.4.4** Componente: Se puede ejecutar si se utilizan los recursos de un nodo que pueden estar contenidos en este.

**Fig**. 21 Elementos que lo componen



(**Ref**.http://3.bp.blogspot.com/-bLyqMRurVaw/VbXZ4E8gbfI/AAAAAAAAA9A/MzP-7YHUzIU/s1600/M.png

# CAPITULO 4

# FASE DE ANALISIS Y DISEÑO

# 4 INTRODUCCIÓN:

El análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorar con métodos y procedimientos más adecuados. El desarrollo de sistemas tiene dos componentes.

Es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema.

En este capítulo se muestra todo el proceso de la fase de análisis y diseño mediante la recolección de información previa utilizando las técnicas de recolección de información tales como:

* Encuestas
* Entrevistas

Se utilizo el diagrama DE UML que están a continuación.

* Historias de usuario.
* Descripción de actores.
* Diagrama casos uso
* Diagrama de Secuencia
* Diagrama de clases
* Diagrama entidad relación
* Diagrama de actividades

# 4.1 HISTORIAS DE USUARIO DEL PROYECTO

Son descripciones rotas y esquemáticas, que resumen la necesidad concreta de un usuario al utilizar un producto o servicio como también a la solución que lo satisface.

Su función principal es identificar problemas percibidos y proponer soluciones y de esa manera estimar los esfuerzos necesarios para las ideas propuestas.

Mediante el cuestionario realizado a las personas involucradas se sacarán historias de usuario concretas que nos ayudarán a determinar los pasos a seguir en el desarrollo del sistema.

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **CODIGO: 1** |
|
| **ROL DE USUARIO: Administrador** | **FECHA:** |
| **DESCRIPCION: Y**o como usuario(**Administrador**), me gustaría un sistema de ingreso de datos de las mascotas de entrada y salida del centro y reportes. | |
|
|
| **OBSERVACION: N**inguna | |
|
|  |  |
| **HISTORIA DE USUARIO** | **CODIGO: 2** |
|
| **ROL DE USUARIO:  Personal** | **FECHA:** |
| **DESCRIPCION: Y**o como (**Personal**)necesito tener un sistema donde me facilite las diferentes tareas que designan en cuantos a los datos de los animales. | |
|
|
| **OBSERVACION: N**inguna | |
|
|  |  |

# 4.2 MODELO DE CASOS DE USO

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto, los casos de uso determinan los requisitos funcionales del sistema basados en las historias de usuarios descritos nteriormente.

## 4.2.1 IDENTIFICACIÓN DE ACTORES

Una vez obtenido los requerimientos del sistema también se identifica a los actores que interactuan con él mismo, donde la identificación de los actores del modelo de casos de uso permite.

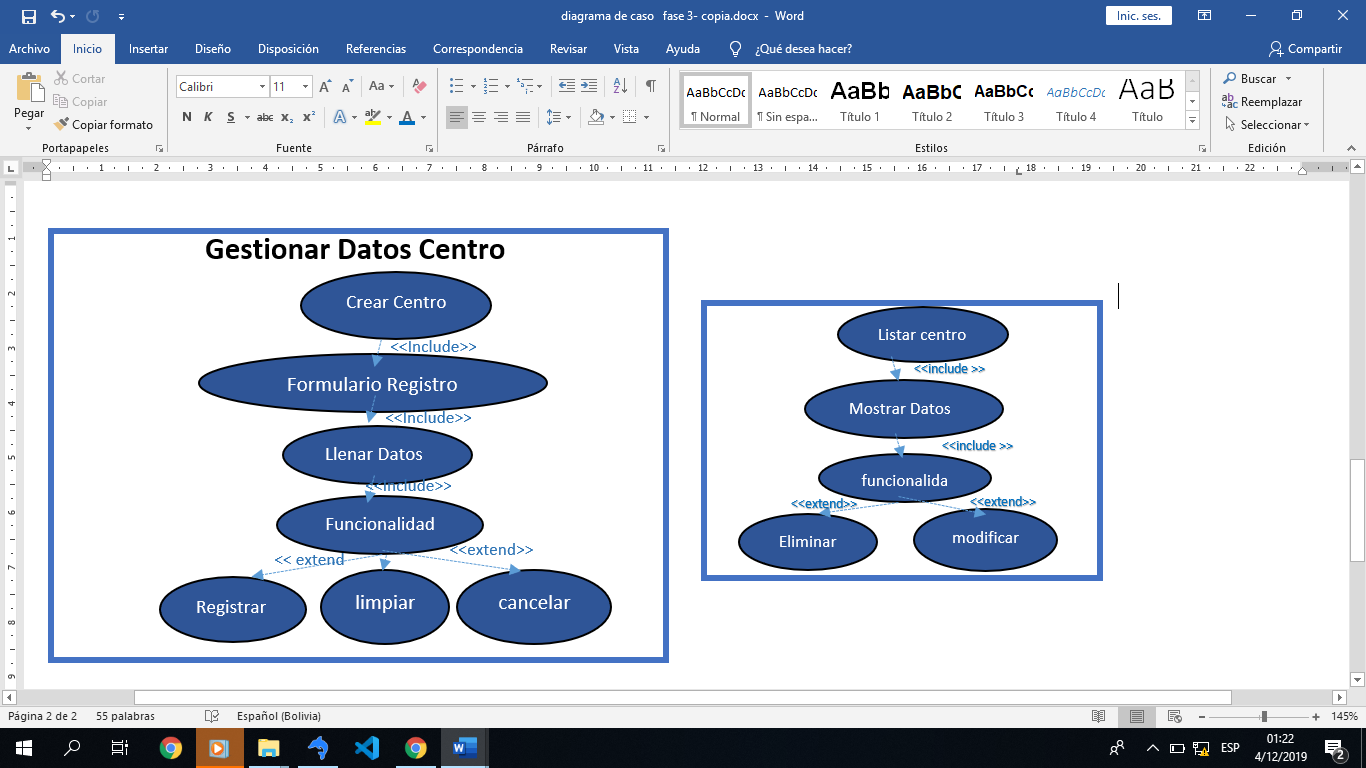
Los actores del sistema son los siguientes:

* + - 1. Administrador, es la persona encargada del manejo sistema y es el que verifica la documentacion y la recepciona.
      2. Personal, es el individuo que es el encargado de controlar la entrada y salida de datos de las mascotas. Asimismo, es el que registra las tareas en el sistema.

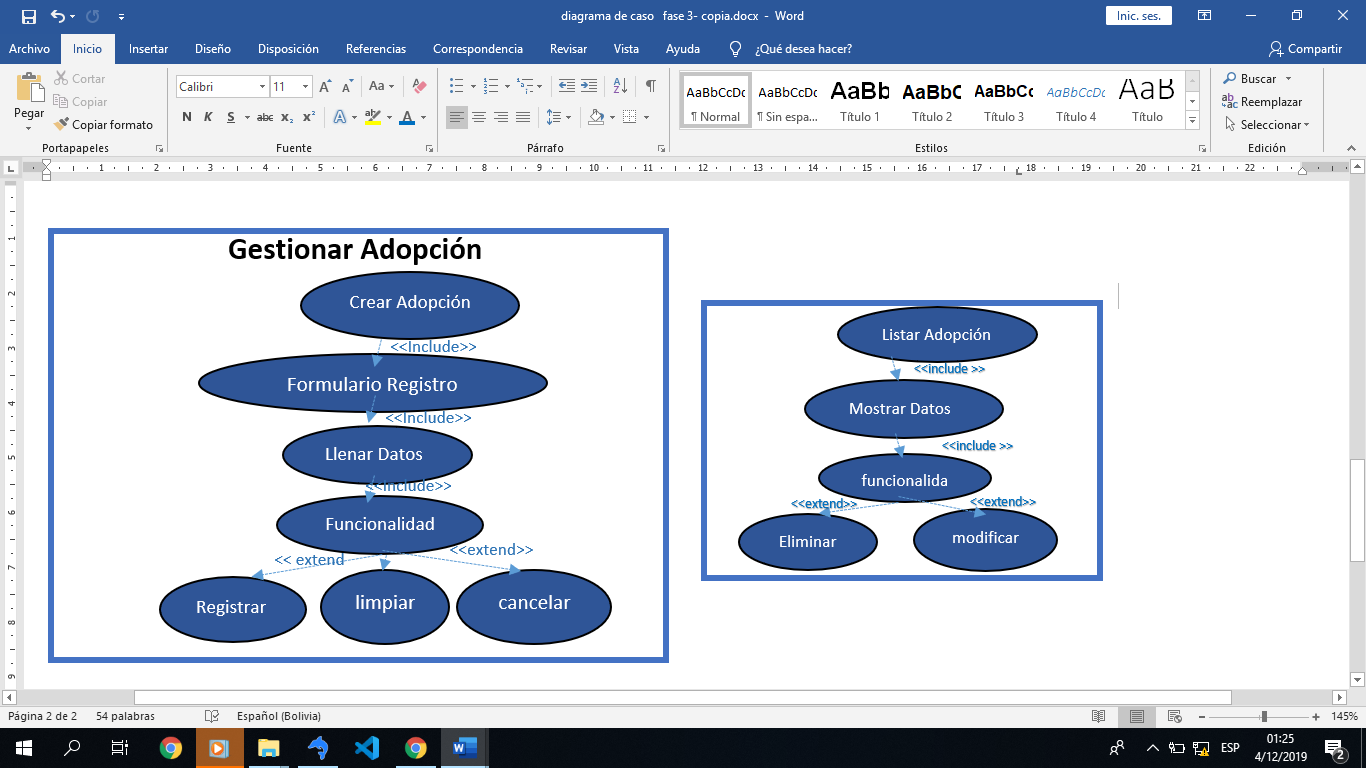
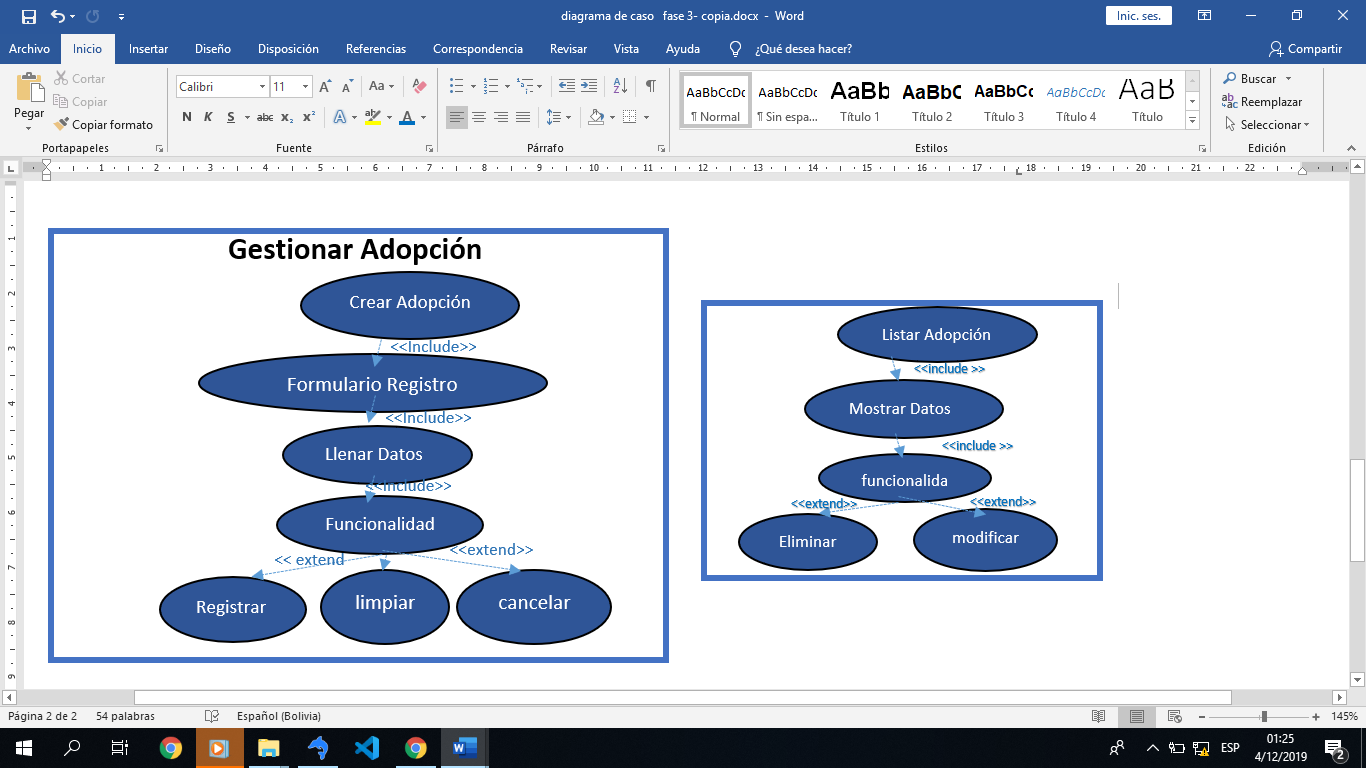
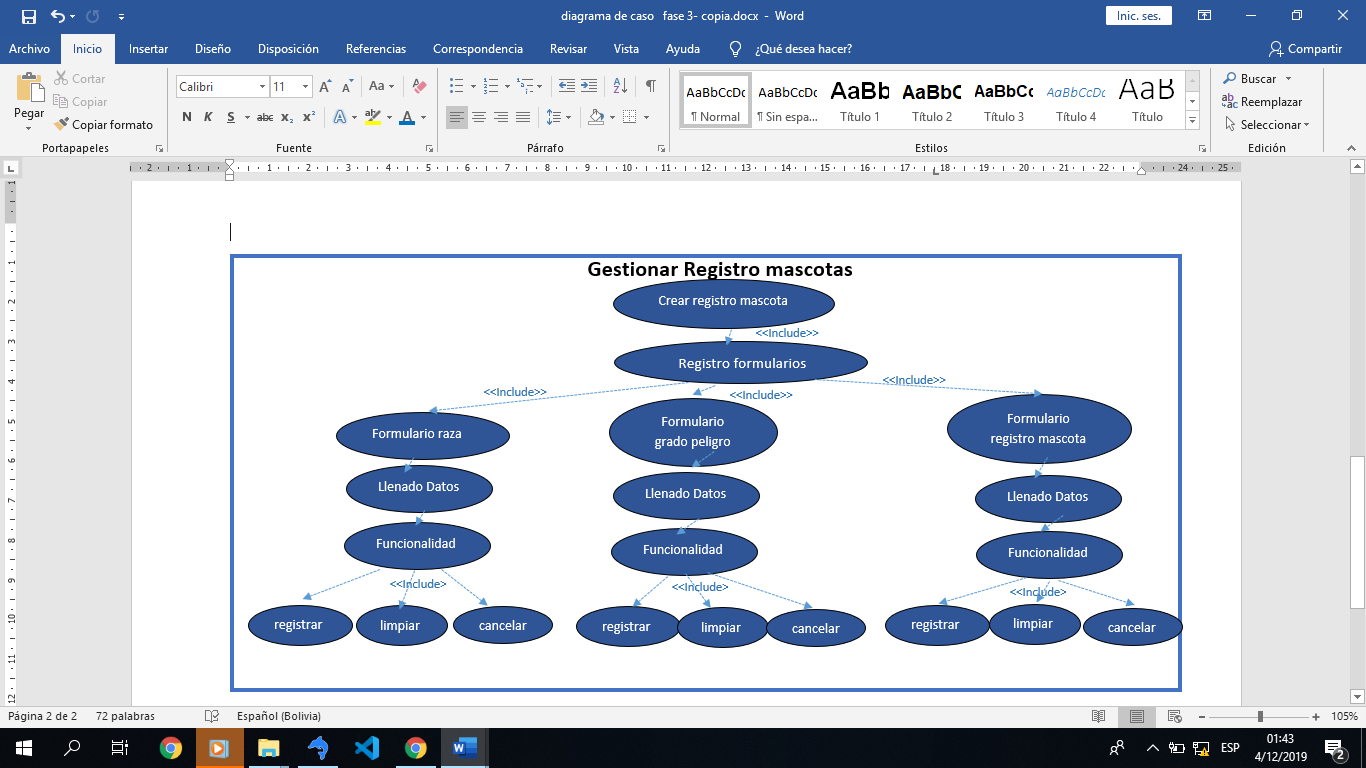
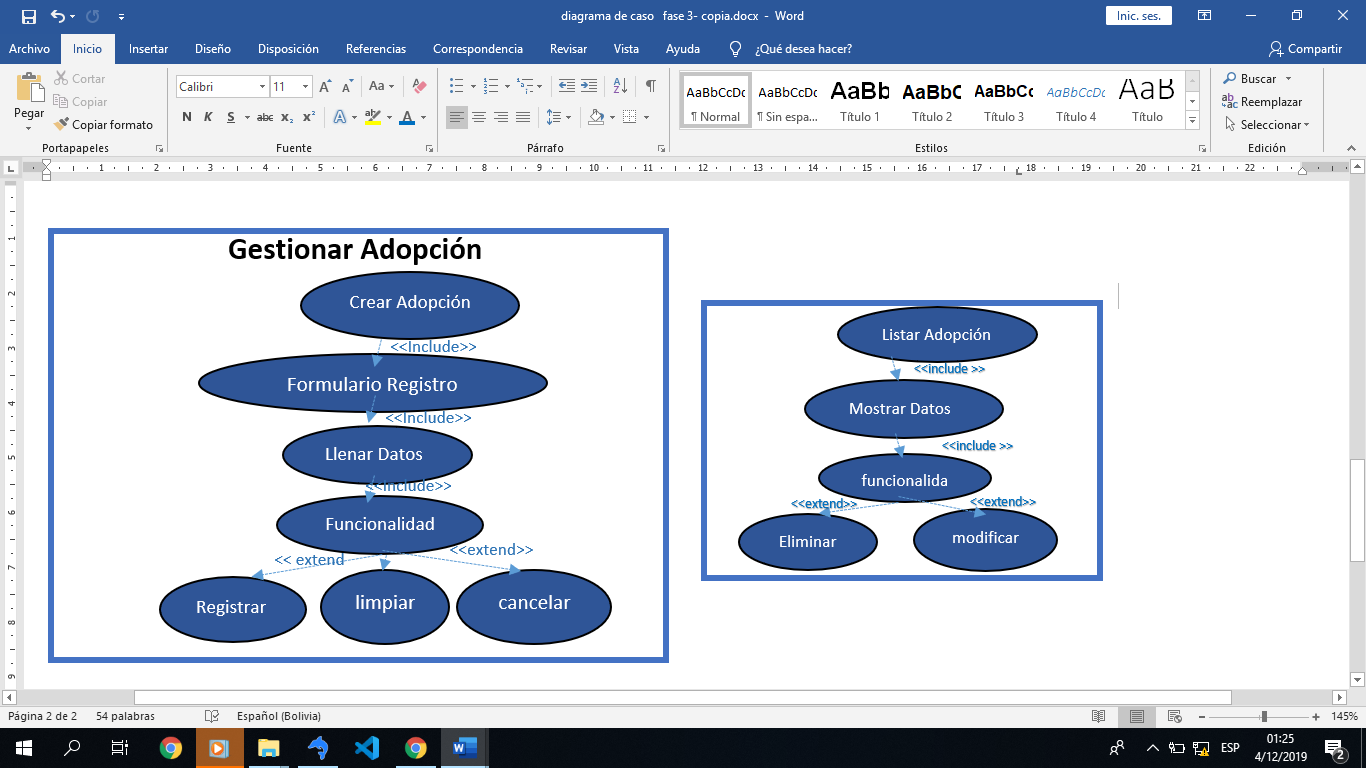
## Diagrama de caso de uso Sistema Zoonosis

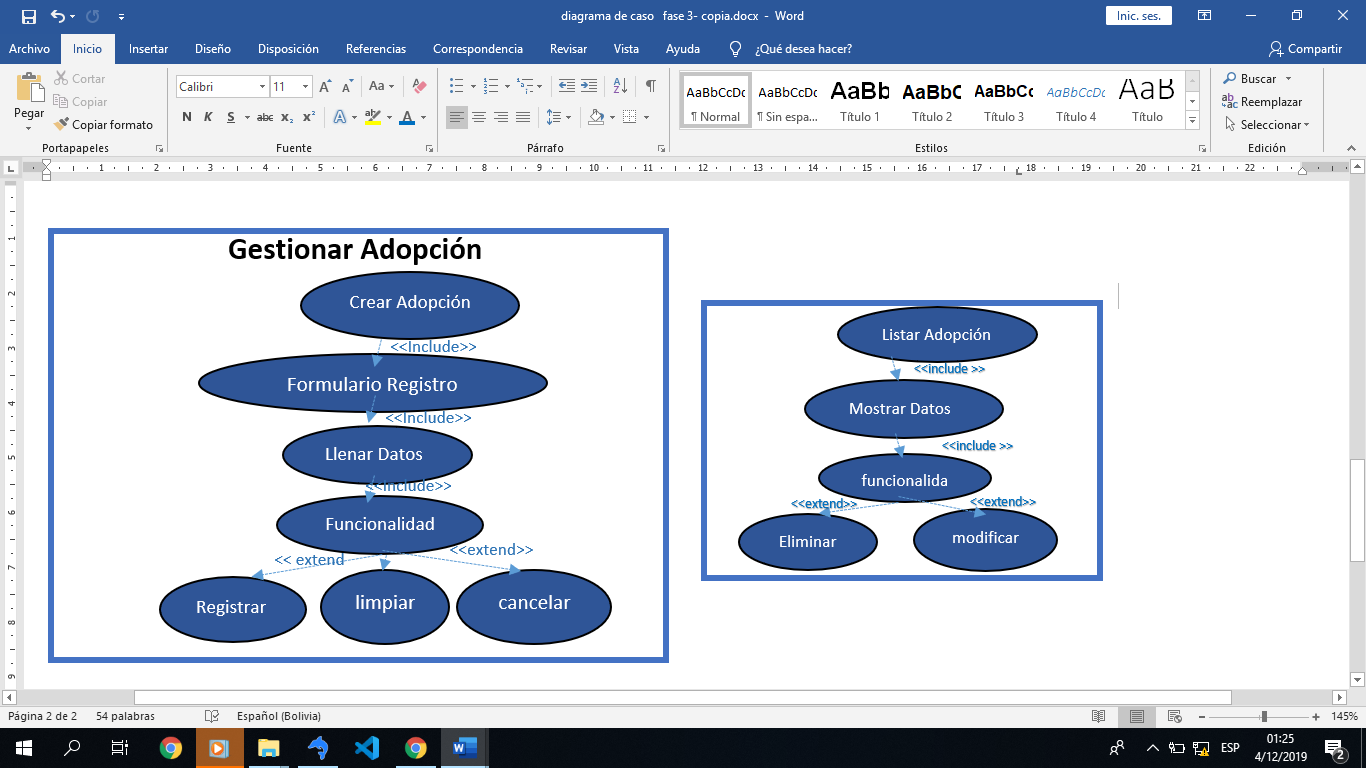
## Caso de Uso Autentificar Usuario

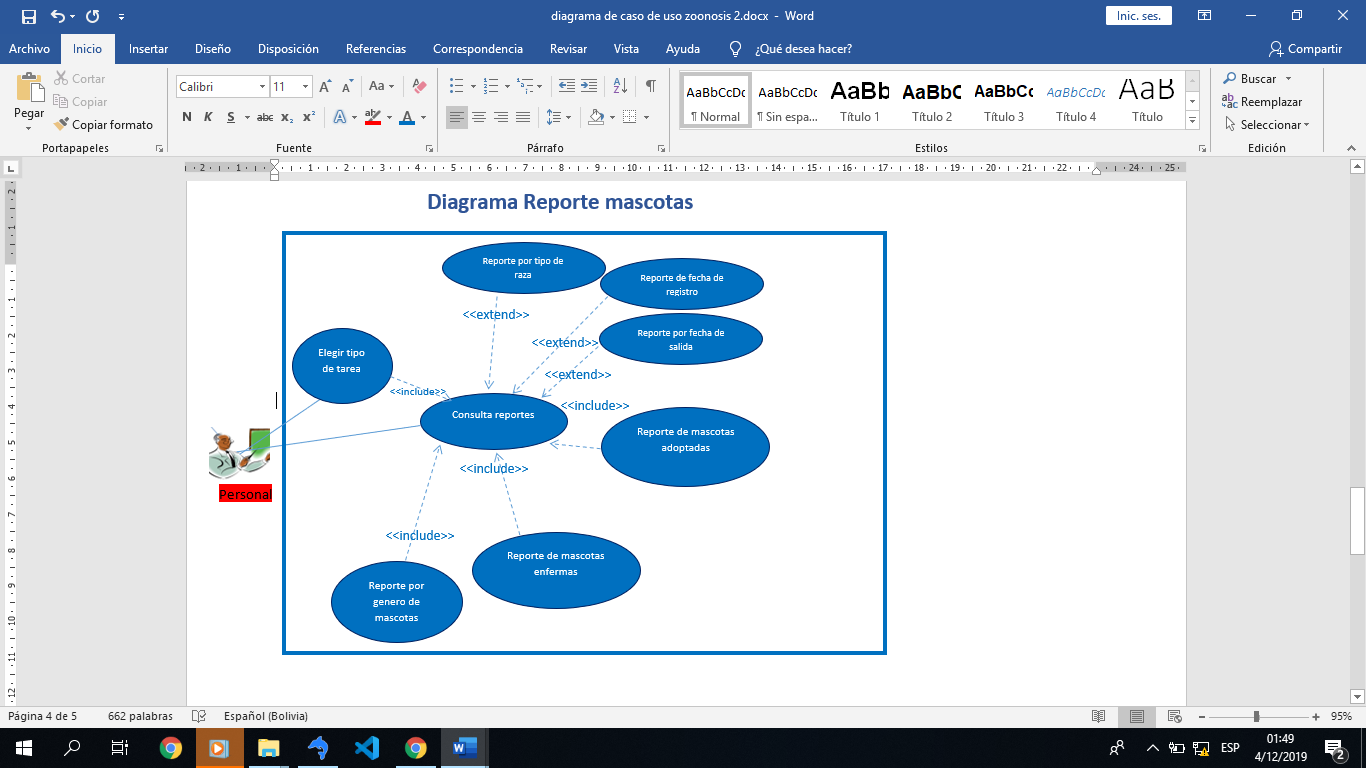
## Caso de Uso Gestion Datos Centro



**Caso de uso gestión de registro mascotas**



**Caso de Uso Gestion de adopcion**

Reportes

4.2.2 DESCRIPCIÓN CASO DE USO: A continuación, se describirán cada uno de los escenarios más importantes del sistema ZoonoSiS.

#### 4.2.2.1 Escenario 1: Reportes de Mascotas

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Reporte de Mascotas** | 1 |
| Actores | 1. administrador | |
| **Tipo** | **Primario** | |
| Referencias | Encargado ingresa al sistema con su usuario e imprime información detallada sobre la mascota. | |
| **Propósito** | | |
| Generar reportes completos inherentes a la mascota. | | |
| **Descripción** | | |
| 1º El Encargado ingresa el Gmail del usuario en el sistema.  2º El sistema verifica los datos del usuario y solicita el tipo de reporte que necesita.  3º El Encargado elige el tipo de reporte:   * Reporte del número de las mascotas * Reporte documentación de solicitud de adopción * Reporte por tipo de raza   4º El sistema genera los informes requeridos en el sistema.  5º El sistema imprime el reporte para el solicitante. | | |

#### 4.2.2.2 Escenario 2: Reporte de Personal

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Reporte de Personal** | **1** |
| Actores | Personal | |
| **Tipo** | **Secundario** | |
| Referencias | Asistente ingresa al sistema con su usuario e imprime información detallada sobre la información que tiene permitida. | |
| **Propósito** | | |
| Generar reportes específicos de cada mascota. | | |
| **Descripción** | | |
| 1º El Asistente ingresa al sistema con su Gmail de usuario  2º El sistema verifica los datos del usuario y solicita el tipo de reporte que necesita.  El Asistente elige el tipo de reporte:   * Reporte de fecha de registro * Reporte por fecha de salida de la mascota   3º Genera los informes requeridos en el sistema.  4º Imprime el reporte para el solicitante. | | |

# 4.3 Diagrama de Secuencia

El diagrama de secuencia es un tipo de diagrama de interacción ordenada que tiene por objetivo describir el comportamiento dinámico que tiene el sistema de información haciendo enfasis en la secuencia de mensajes intercambiados por los objetos [Jacob, 2000].

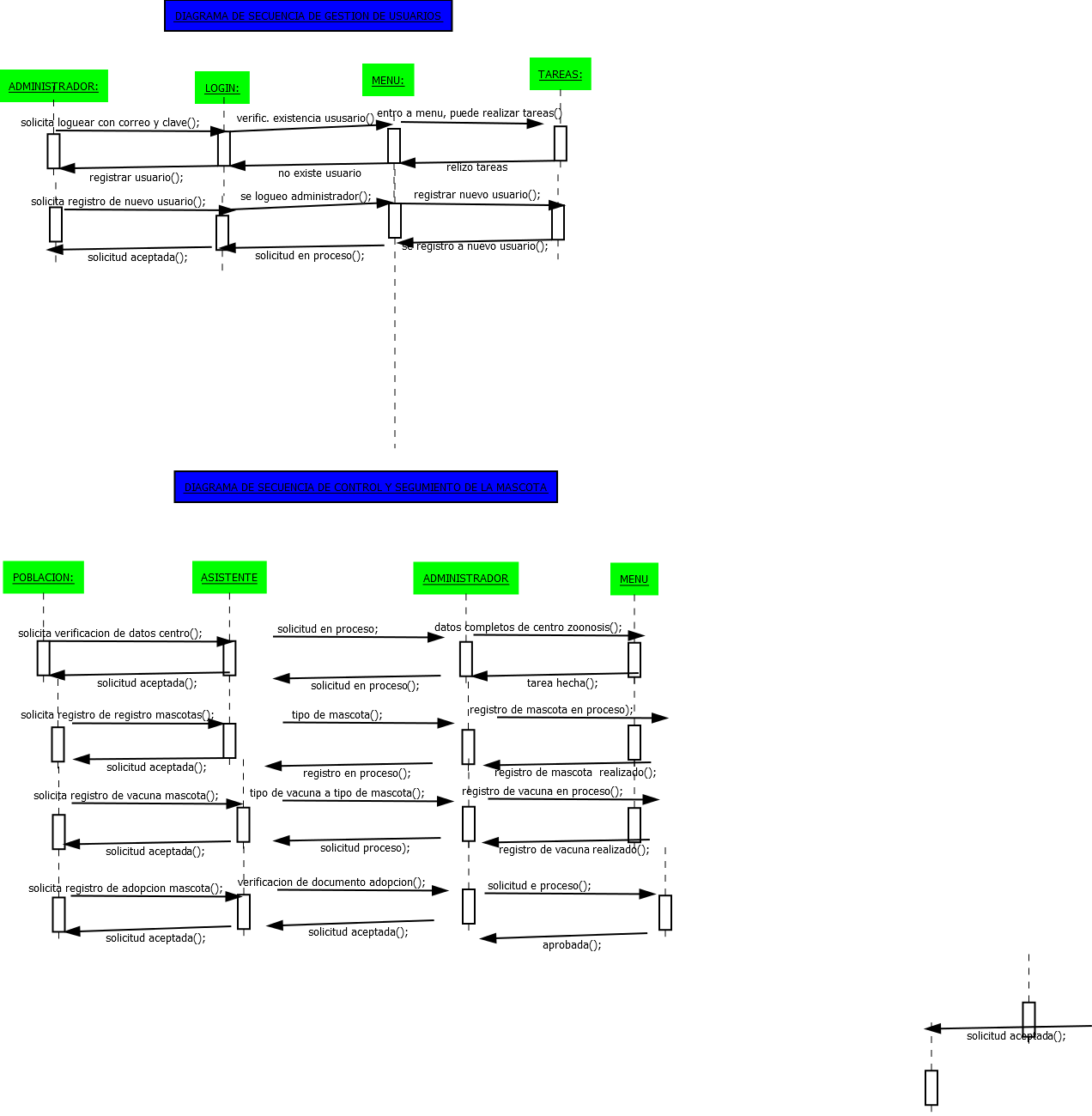
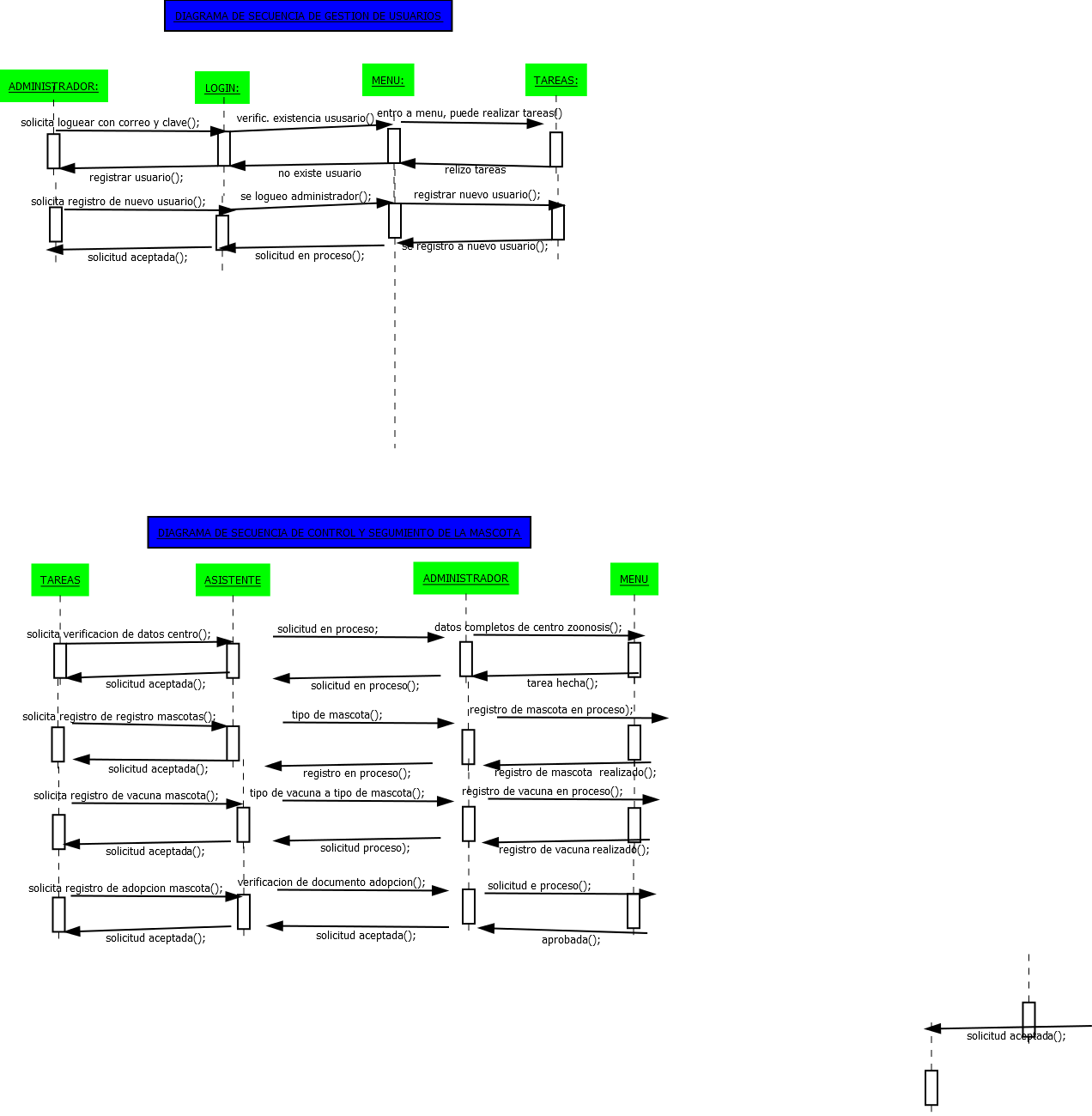
Dentro de estos diagramas se indicaran los módulos o clases que forman parte del programa y las llamadas que hacen en cada uno de ellos para realizar una tarea determinada, los diagramas de secuencia muestran:

* Los objetos participando de la interacción.
* La secuencia de mensajes intercambiados.

## 4.3.1 Diagrama de Secuencia Módulo Gestión de Usuario

Como ya se vió anteriormente en los diagramas de casos de uso, a este módulo solo puede acceder el usuario administrador del sistema a través de su respectivo control de autentificación y validación de datos.

En la siguiente figura se puede observar como el usuario administrador del sistema interactúa con el sistema a través de la interfaz gráfica que hace de vista y recibe del respectivo controlador la información necesaria para poder gestionar los datos del usuario dentro la base de datos como se muestra en la siguiente figura:



**Ref**.:( Elaboración Propia)

Diagrama17: Diagrama de Secuencia Módulo Gestión de Usuarios

Fuente: Elaboración Propia

**Ref**.:( Elaboración Propia)

**Imagen 3.3.1: Diagrama de secuencia modulo Gestionar Usuario**

**Fuente: Elaboración Propia**

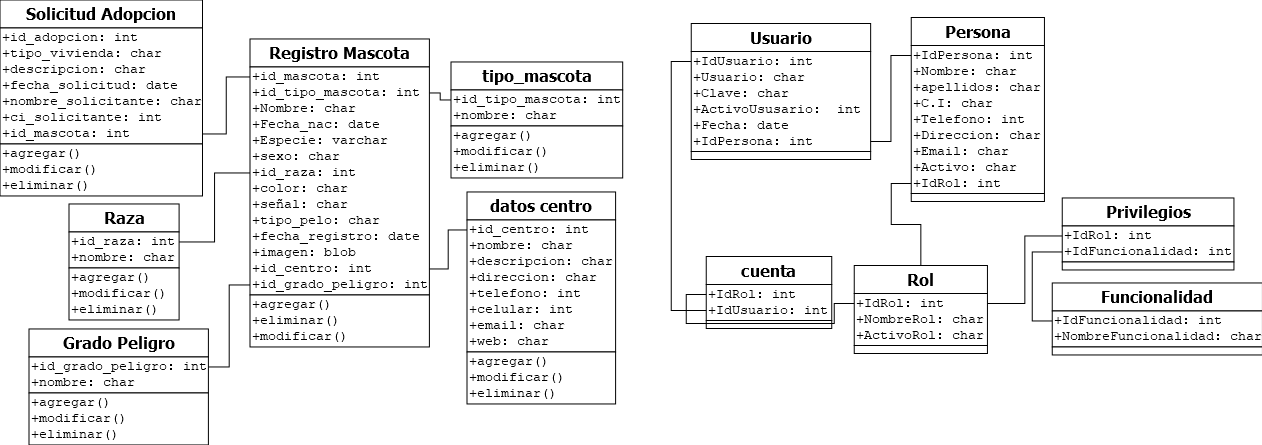
# 4.4 DIAGRAMA DE CLASES

Un diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Nos sirve para visualizar las relaciones entre las clases que involucran el sistema, también nos permitirá

representar gráficamente y de manera estática la estructura general de un sistema, mostrando cada una de las clases y sus interacciones (como herencias, asociaciones, etc.)

## 4.4.1 ULTIMA VERSION DEL DIAGRAMA DE CLASES:

Una vez revisado los requerimientos del usuario se modifico el diagrama de clases 4 veces, donde finalmente se tiene el seguimiente diagrama de clases.



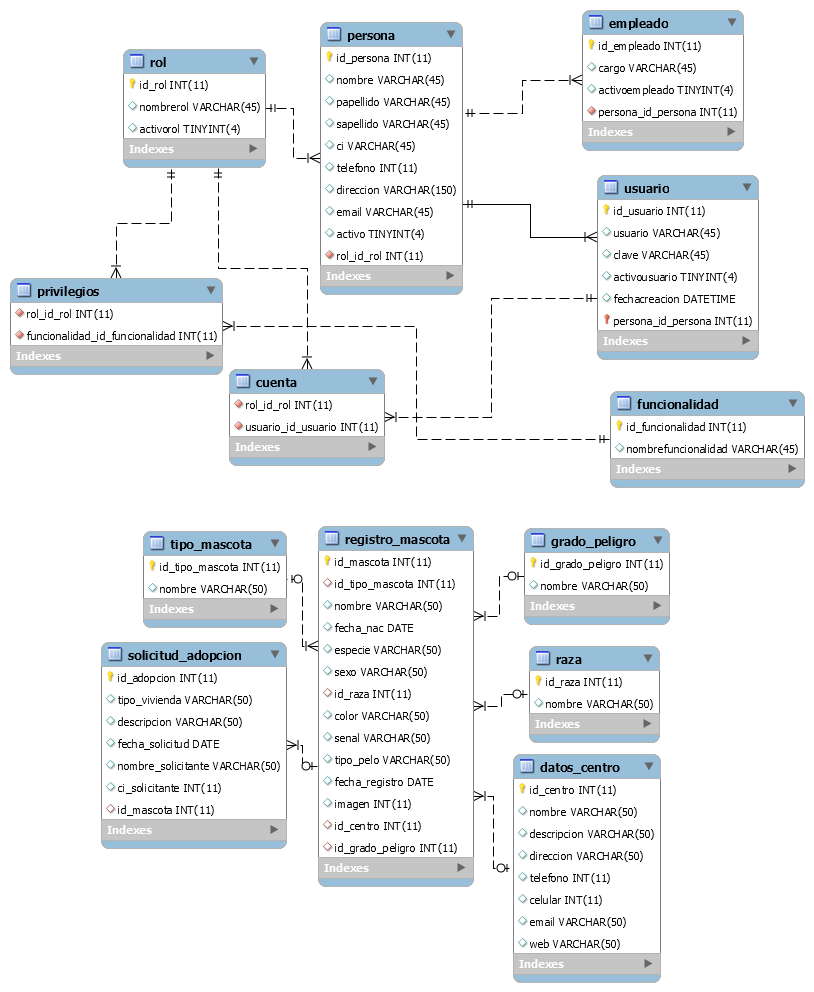
**Fig**. 24 Versión Final de la base de datos

**Ref**.:( Elaboración Propia)

# 4.5 **DIAGRAMA ENTIDAD RELACION.**

**Diagrama entidad relación**

**Fig**.25 Diagrama Entidad relación



**Ref**. Elaboración Propia)

## 4**.6 DIAGRAMA ACTIVIDADES**

**Ref**.:( Elaboración Propia)

# 4.7 FASE DE CONSTRUCCION

Dentro de esta fase de construcción se realizarán los métodos, lógica de programación del sistema y los artefactos del flujo de trabajo, además también se realizarán las pruebas correspondientes a cada resultado de la implementación de cada módulo con el fin de conocer y corregir la existencia de errores, obviamente teniendo en cuenta el patrón para desarrollo de software MVC.

## 4.7.1 Herramientas De Implementación

Servidor Web Apache**.** -El servidor Apache HTTP, también llamado Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento.

“Su misión es crítica, ya que es el encargado de aceptar las peticiones de páginas (o recursos en general) que provienen de los visitantes que acceden a nuestro sitio web y gestionar su entrega o denegación, de acuerdo a las políticas de seguridad establecidas. Esto, que puede parecer simple, implica muchas facetas y funcionalidades que debe cubrir, como pueden ser: Atender de manera eficiente, ya que puede recibir un gran número de peticiones HTTP, restricciones de acceso a los ficheros que no se quieran ‘exponer’, gestión de autentificaciones de usuarios o filtrado de peticiones según el origen de éstas, manejar los errores por páginas no encontradas, informando al visitante y/o redirigiendo a páginas predeterminadas o gestión de la información a transmitir en función de su formato e informar adecuadamente al navegador que está solicitando dicho recurso.”[[5]](#footnote-6)

XAMPP:  es un servidor independiente de plataforma de código libre. Permite instalar de forma sencilla Apache propio ordenador, sin sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito.

XAMPP incluye además servidores de bases de datos como MySQL y SQLite con sus respectivos gestores phpMyAdmin y phpSQLiteAdmin. Incorpora también el intérprete de PHP, el intérprete de Perl, servidores de FTP como ProFTPD o FileZilla FTP Serve, etc. entre muchas cosas más.

“Si alguna vez has intentado instalar Apache, sabes que no es una tarea fácil, sin embargo, con XAMPP todo es diferente. Una de las ventajas de usar XAMPP es que su instalación es de lo más sencilla, basta descargarlo, extraerlo y comenzar a usarlo.”[[6]](#footnote-7)

En si es una herramienta de desarrollo trabajo (páginas web o programación, por ejemplo) en tu propio ordenador sin necesidad de tener que Accesar a internet.

## 4.7.2 Herramientas de Diseño

Para la implementación se utilizó el lenguaje de programación PHP de código de lado del servidor. Para el diseño de la base de datos seusó*“PowerDesingner 16.1 que* permite el manejo de archivos “.sql”, donde se pueden realizar los modelos conceptuales, modelos lógicos y también los modelos físicos de nuestra BD.

Php**:** “Acrónimo recursivo en inglés de PHP Hypertext Preprocessor(preprocesador de hipertexto), es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [de propósito general](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_de_prop%C3%B3sito_general) de [código del lado del servidor](https://es.wikipedia.org/wiki/Script_del_lado_del_servidor) originalmente diseñado para el [desarrollo web](https://es.wikipedia.org/wiki/Desarrollo_web) de [contenido dinámico](https://es.wikipedia.org/wiki/Contenido_din%C3%A1mico). Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento [HTML](https://es.wikipedia.org/wiki/HTML) en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante.”[[7]](#footnote-8)

También se utilizó la herramienta case de UML *“Vicio-2010t”* la cual permite la realización y maquetación de diagramas UML para la posterior implementación del producto de software, otra herramienta que sirvió para el diseño del sistema es “Adobe Photoshop” una poderosa herramienta para diseñar las interfaces particularmente web.

## 4.7.3 Entorno **de Desarrollo**

Para el desarrollo del actual sistema se utilizó el entorno de trabajo *“IDE”* denominado “*Sublime\_Text”* el cual es un programa informático compuesto por un conjunto de herramientas de programación, se caracteriza por ser uneditor de texto multiplataforma y ligero que soporta varias ventanas abiertas al momento.

Visual studio code: Es un editor de código fuente desarrollado por [Microsoft](http://www.ecured.cu/Microsoft) para [Windows](http://www.ecured.cu/Windows) , [Linux](http://www.ecured.cu/Linux) y [macOS](http://www.ecured.cu/MacOS) . Incluye soporte para depuración , control de Git integrado, resaltado de sintaxis , finalización de código inteligente , fragmentos de código y refactorización de código . También es personalizable, de modo que los usuarios pueden cambiar el tema del editor, los métodos abreviados de teclado y las preferencias. Es gratuito y de código abierto.

Sqlyog: “Es una excelente interfaz gráfica diseñada especialmente para trabajar de forma más rápida y cómoda con el servidor de base de datos [MySQL](https://www.ecured.cu/MySQL). El programa está orientado a usuarios que ya tienen ciertos conocimientos de [SQL](https://www.ecured.cu/SQL) y necesitan un intérprete gráfico sin excesivas florituras, ágil y funcional. SQLyog te permite administrar usuarios y permisos, y realizar múltiples peticiones a base de datos.” [[8]](#footnote-9) También puedes insertar fácilmente, mediante plantillas, peticiones que realices con frecuencia, como por ejemplo creación de tablas. Otras opciones son la exportación de datos en formato [CSV](https://www.ecured.cu/CSV), [HTML](https://www.ecured.cu/HTML) y [XML](https://www.ecured.cu/XML), la optimización de bases de datos, la posibilidad de guardar [scripts](https://www.ecured.cu/Scripts) de [SQL](https://www.ecured.cu/SQL) como favoritos. Soporte Unicode/UTF8 completo, Productividad desarrollador / usuario, Documentación de esquema [HTML](https://www.ecured.cu/HTML), Atajos para generar sentencias [SQL](https://www.ecured.cu/SQL) [DML](https://www.ecured.cu/index.php?title=DML&action=edit&redlink=1) desde definición de esquema, Editor de consultas con pestañas múltiples y editor de resultados, Ejecución múltiple de consultas, Ejecución de consultas multihilo - Posibilidad de detener consultas grandes, Plantillas [SQL](https://www.ecured.cu/SQL), Interfaz de cuadrícula estilo [Excel](https://www.ecured.cu/Excel) para ver/actualizar los resultados, Editor de Blob multiformato, Se pueden ver datos en modo cuadrícula o texto, Exportación de Resultado/Datos a [CSV](https://www.ecured.cu/CSV) / [XML](https://www.ecured.cu/XML) / [HTML](https://www.ecured.cu/HTML) / [Excel](https://www.ecured.cu/Excel).

## **Sublime Text:**

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

[Sublime Text](http://www.sublimetext.com/) permite tener varios documentos abiertos mediante pestañas, e incluso emplear varios paneles para aquellos que utilicen más de un monitor. Dispone de modo de pantalla completa, para aprovechar al máximo el espacio visual disponible de la pantalla.

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

“Este programa dispone de auto-guardado, muchas opciones de personalización especializada, cuenta con un buen número de herramientas para la edición del código y automatización de tareas. Soporta macros, Snippets y auto completar, entre otras funcionalidades. Algunas de sus características son ampliables mediante plugins.” [[9]](#footnote-10)

## **Adobe Photoshop CS6;**

“Adobe Photoshop es un [editor de gráficos rasterizados](https://es.wikipedia.org/wiki/Editor_de_gr%C3%A1ficos_rasterizados) desarrollado por [Adobe Systems Incorpórate](https://es.wikipedia.org/wiki/Adobe_Systems_Incorporated). Usado principalmente para el [retoque](https://es.wikipedia.org/wiki/Retoque_fotogr%C3%A1fico) de [fotografías](https://es.wikipedia.org/wiki/Fotograf%C3%ADa) y gráficos, su nombre en español significa literalmente ‘taller de fotos’. Es líder mundial del mercado de las [aplicaciones](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) de edición de imágenes y domina este sector de tal manera que su nombre es ampliamente empleado como sinónimo para la edición de imágenes en general.“[[10]](#footnote-11)

Muchas empresas (en especial empresas web), utilizan en este programa para realizar los logotipos con los que se identificarán.

## **Navegador Google Chrome;**

Google Chrome es un [navegador web](http://www.alegsa.com.ar/Dic/navegador.php) de [código abierto](http://www.alegsa.com.ar/Dic/codigo%20abierto.php) desarrollado por [Google](http://www.alegsa.com.ar/Dic/google.php).  
“Este navegador está constituido por componentes de otros navegadores de código abierto como WebKit y Mozilla. El objetivo de Google Chrome es lograr un navegador con una interfaz sencilla, y una muy buena estabilidad, velocidad y seguridad.”**[[11]](#footnote-12)**

## **Dia. -**

“Dia es una [aplicación informática](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) de propósito general para la creación de diagramas, creada originalmente por Alexander Larsson[1](https://es.wikipedia.org/wiki/Dia_(programa)#cite_note-1)​, y desarrollada como parte del proyecto [GNOME](https://es.wikipedia.org/wiki/GNOME) .” [[12]](#footnote-13) Está concebido de forma modular, con diferentes paquetes de formas para diferentes necesidades.

# 4.8 Implementación De Acceso Al Sistema

Para el acceso y como cierto grado de seguridad del sistema se implementó el controlador ***“zoonosis/controlador/ctrl\_login.php”***, este archivo está encargado de controlar tanto a las vistas como los modelos en este caso orientados al módulo de autentificación de usuarios, el controlador de este módulo fue desarrollado de la siguiente forma:

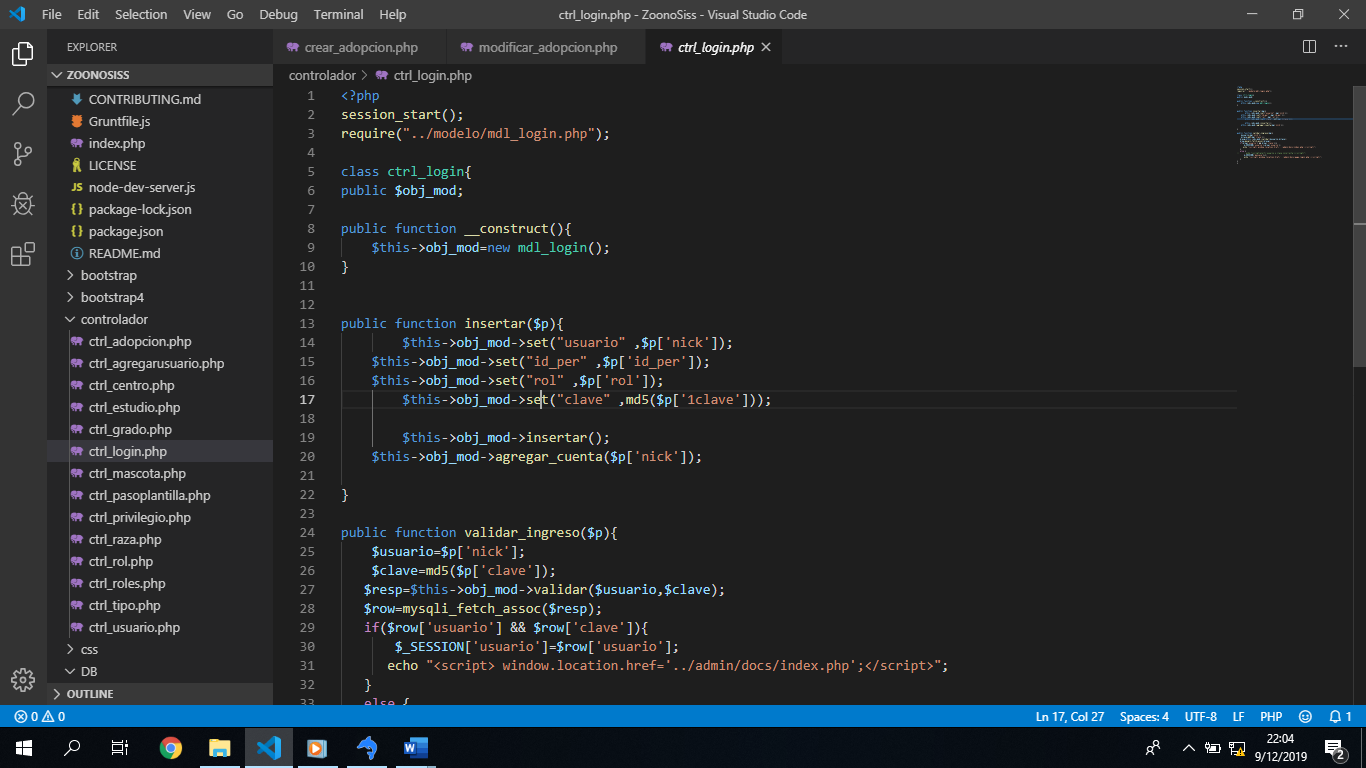


Figura 27: Código del Controlador ctrl\_login.php

Fuente: Elaboración Propia

## 4.8.1 Prueba de caja negra de Acceso al Sistema

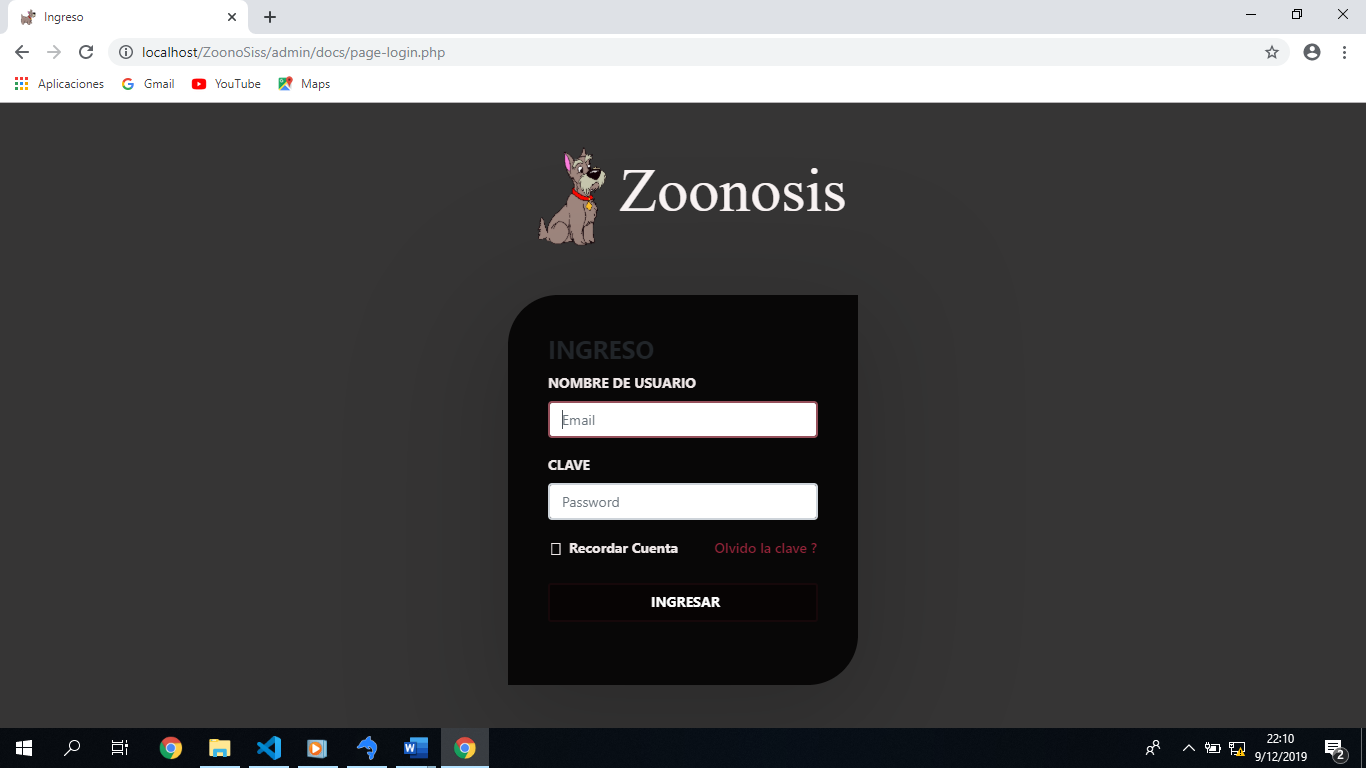


Figura 30: Vista Generada por el archivo login.php

Fuente: Elaboración Propia

Para la documentación del actual módulo y los posteriores, realizaremos las pruebas de caja negra para poder verificar el funcionamiento y la existencia de errores en el sistema.

|  |  |
| --- | --- |
| **No. Caso de Prueba** | 1 |
| **Usuario** | Administrador, perspnal |
| **Nombre de Caso de uso** | Acceso al sistema |
| **Nombre del Caso de Prueba** | Acceso a la cuenta como usuario |
| **Entradas** | Usuario: admin@admin.com  Clave: admin |
| **Salidas** | Ingreso a los módulos permitidos |
| **Descripción** | Ingreso al sistema como usuario cualquiera en este caso administrador. |
| **Procedimiento de Prueba** | 1. Ingresar a la opción acceso 2. Ingrese los datos indicados en entradas. 3. Presione botón enviar para verificar los datos en la base de datos. |
| **Resultados Esperados** | El sistema valida la información ingresada y da acceso a su cuenta del usuario. |

Tabla 12: Prueba de Acceso de usuario

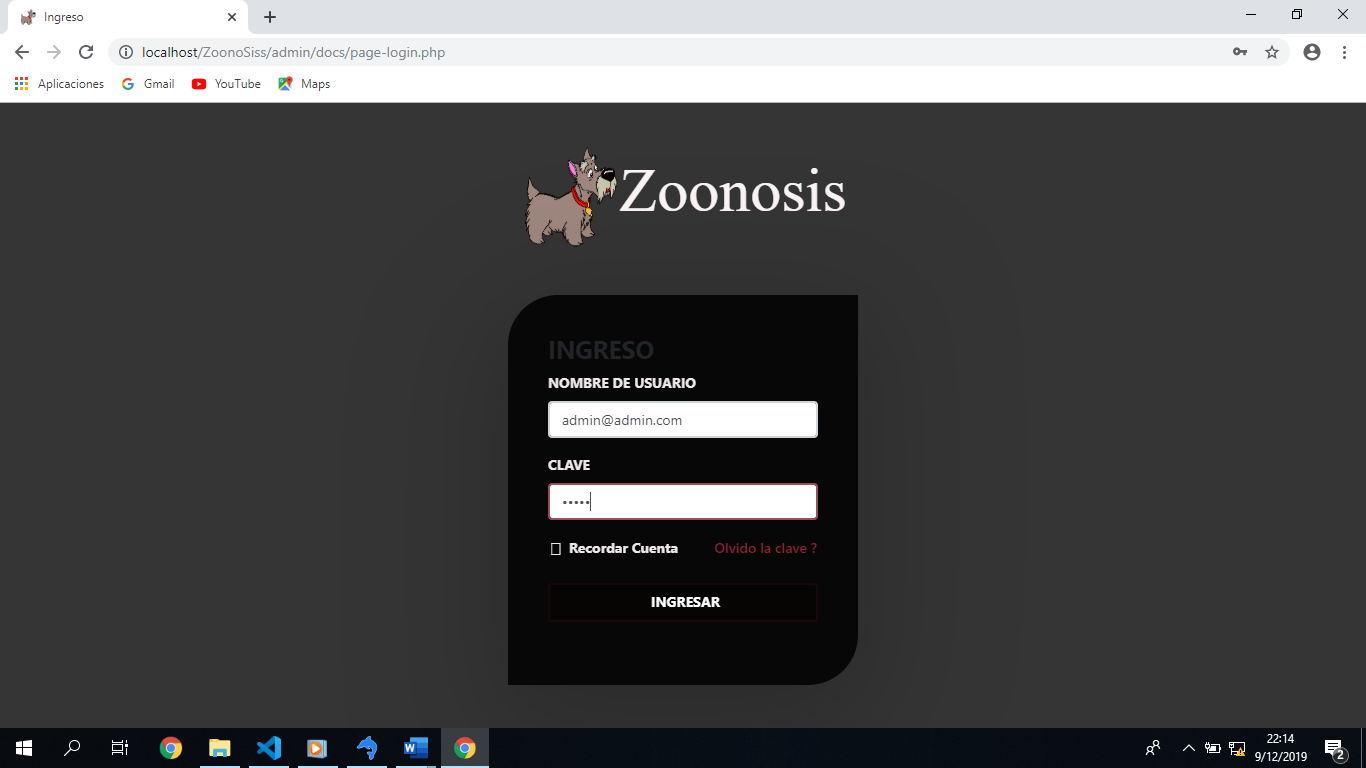
**Fuente:** Elaboración Propia

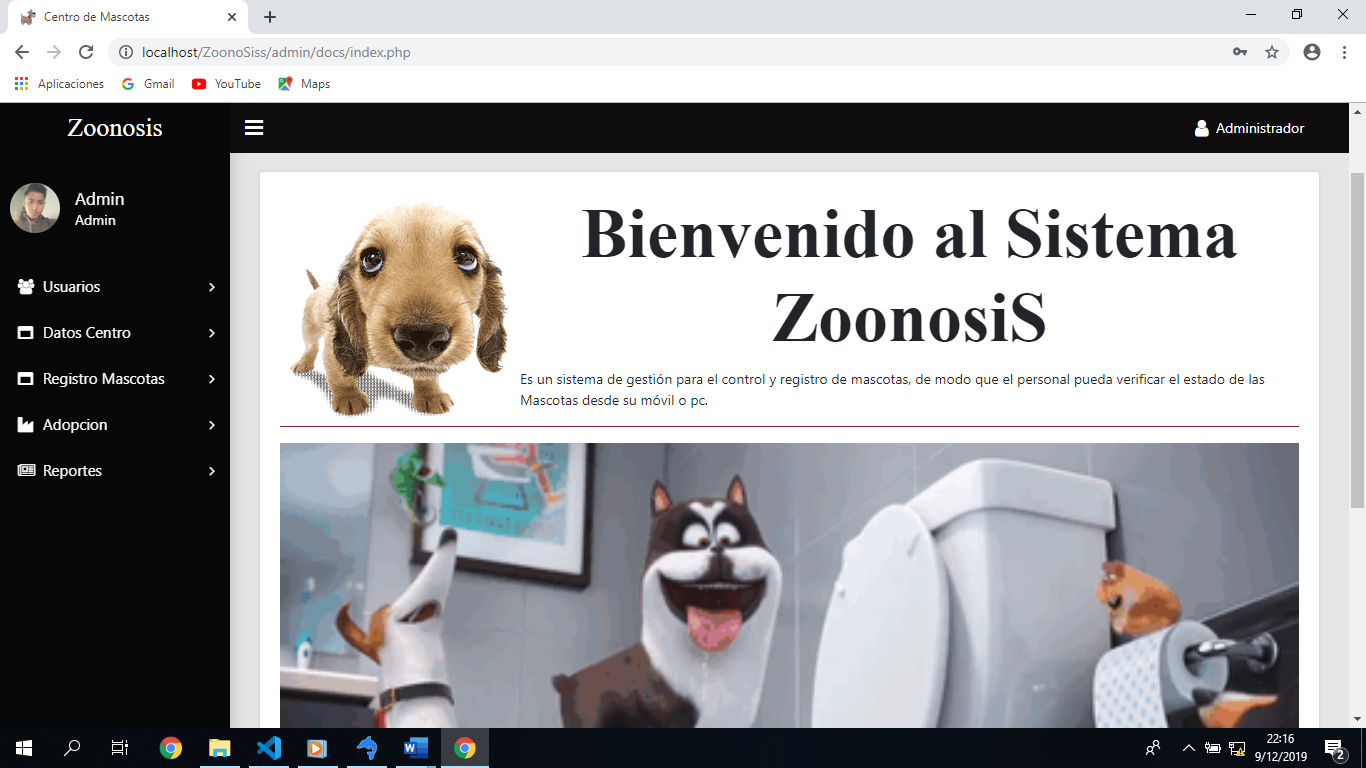
Ilustración 32: Prueba de Caja Negra de Acceso al Sistema con datos Incorrectos

Fuente: Elaboración Propia

La prueba de este módulo consiste en, primero se debe iniciar el sistema y por defecto se invocara al archivo ***“login.php”*** donde se imprimirá en pantalla el formulario del autentificación.

Ilustración 31: Prueba de Caja Negra de Acceso al Sistema con datos Correctos

Fuente: Elaboración Propia



## 4.8.2 IMPLEMENTACIÓN DEL MÓDULO GESTIÓN DE USUARIOS

La función Save es un método del modo controlador que realiza el paso de captura de parámetros de la interfaz del usuario y enviando los a la base de datos para ser registrados.

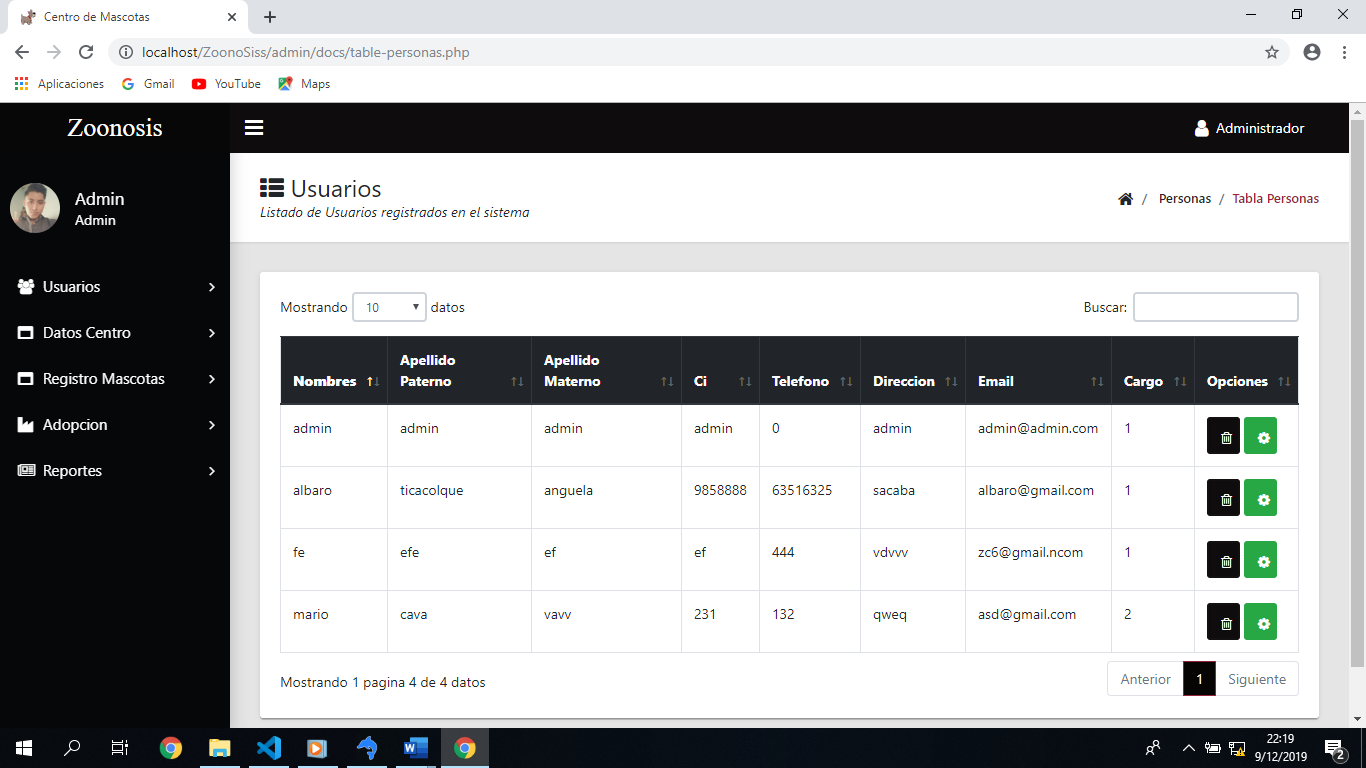
A continuación, en la siguiente figura se muestra la interfaz de gestión de usuarios del sistema.

Ilustración 35: Método de registro de un usuario en la base de datos

Fuente: Elaboración Propia

## 4.8.3 Prueba de caja negra módulo Gestión de Usuarios

Se realizará las pruebas de caja negra para el registro y borrado de usuarios.

|  |  |
| --- | --- |
| **No. Caso de Prueba** | **2** |
| **Usuario** | **Administrador** |
| **Nombre de Caso de uso** | **Gestión Usuarios** |
| **Nombre del Caso de Prueba** | Gestión usuario |
| **Entradas** | * Registro del formulario principal de llenado de datos generales de los usuarios |
| **Salidas** | Registro en la base de datos. - Validación de error |
| **Descripción** | Se puede realizar el registro de nuevo usuario, modificación de datos, y dar de baja a un usuario. |
| **Procedimiento de Prueba** | 1. Ingresar a la opción usuario 2. Ingrese los datos indicados en entradas. 3. Presione botón enviar para verificar los datos en la base de datos. |
| **Resultados Esperados** | El sistema valida la información ingresada y registra los datos en la base de datos. |

Tabla 13: Prueba de Gestión usuarios

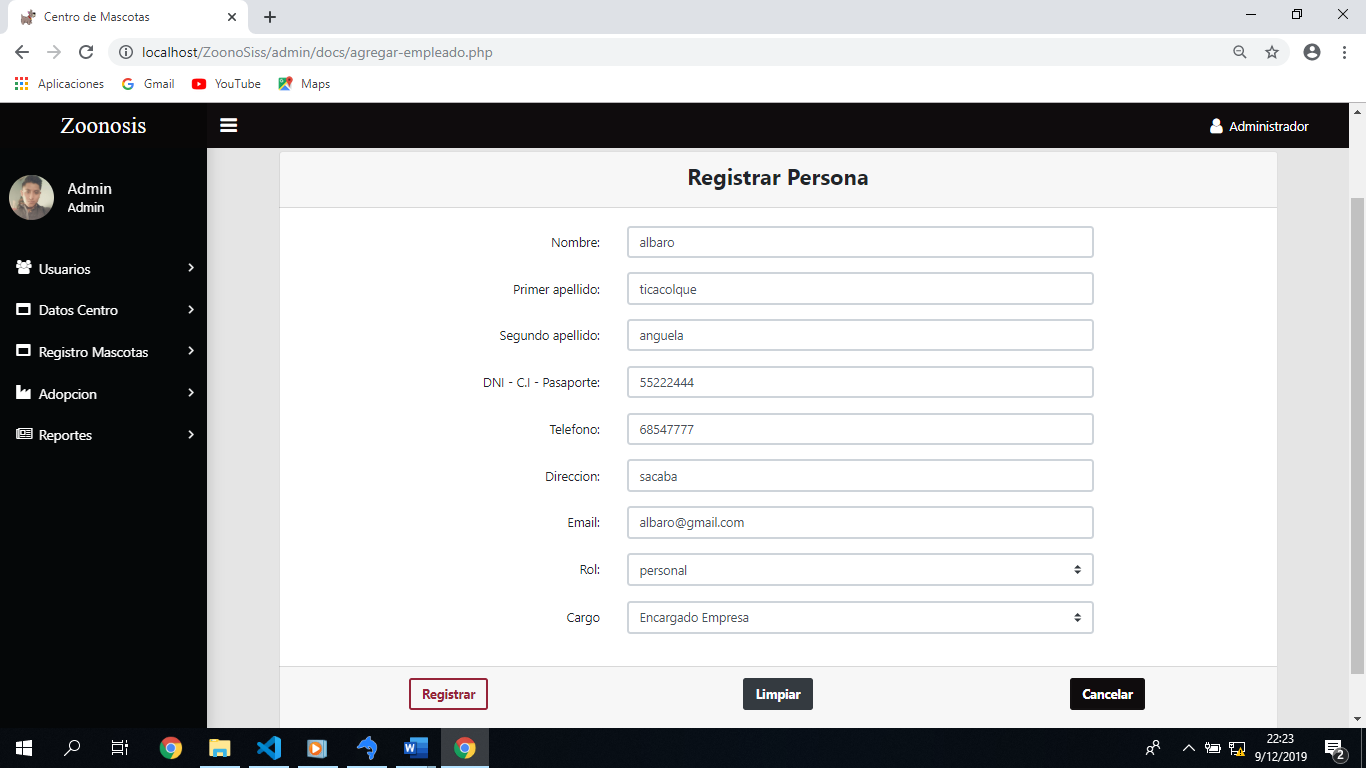
**Fuente:** Elaboración Propia

Ilustración 37: Prueba de Caja Negra Registrar Usuario

Fuente: Elaboración Propia

# ANEXOS

## ANEXO A – ENCUESTA

Con este cuestionario se espera poder definir si el centro zoonosis cuenta con estas tareas para el registro y gestión de canes y felinos.

Lee con atención y responde las siguientes preguntas:

1. ¿zoonosis cuenta con algún sistema?

Sí

No

2. ¿Zoonosis tiene un registro de estas mascotas?

* Sea en Word
* Excel
* Actas
* ninguno

3. ¿Qué características positivas destaca el centro hacia la ayuda de las mascotas?

* socorro a perros callejeros
* canes abandonados
* avisos para adopción
* vacunas
* parasitación
* otros

4. ¿Tiene avisos en páginas web?

Sí

No

5. ¿Tiene algún tipo de reporte para poder buscar a una mascota?

* raza
* color
* sexo
* ninguno

6. ¿Tiene algún tipo registro de mascotas peligrosa?

Sí

No

7. ¿En que afectan los perros callejeros?

* virus o enfermedades rábicas
* peligro a la población
* atropello de la población a estos canes
* ninguno

1. <https://www.monografias.com/trabajos94/modelado-sistemas-uml/modelado-sistemas-uml.shtml>. [↑](#footnote-ref-2)
2. <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml> [↑](#footnote-ref-3)
3. <https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-componentes/> [↑](#footnote-ref-4)
4. <http://ingsoftware-luiszambrano.blogspot.com/2015/07/diagrama-de-despliegue.html> [↑](#footnote-ref-5)
5. <https://www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf/> [↑](#footnote-ref-6)
6. <http://myu-charly.blogspot.com/> [↑](#footnote-ref-7)
7. <https://es.wikipedia.org/wiki/PHP> [↑](#footnote-ref-8)
8. [https://www.ecured.cu/SQLyog](https://www.ecured.cu/SQLyog%2030/VII/2018)  [↑](#footnote-ref-9)
9. [https://es.wikipedia.org/wiki/Sublime\_Text](https://es.wikipedia.org/wiki/Sublime_Text%2030/VII/2018)  [↑](#footnote-ref-10)
10. <https://es.wikipedia.org/wiki/Adobe_Photoshop> [↑](#footnote-ref-11)
11. [https://es.wikipedia.org/wiki/Sublime\_Tex](https://es.wikipedia.org/wiki/Sublime_Text%2030/VII/2018) [↑](#footnote-ref-12)
12. <https://es.wikipedia.org/wiki/Google_Chrome> [↑](#footnote-ref-13)