



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**  
**UNIDAD DE ESTUDIOS A DISTANCIA**  
**MODALIDAD SEMIPRESENCIAL**  
**CARRERA SISTEMAS**



**TESIS DE GRADO**

**Sistema de información con aplicación móvil para el control administrativo de la empresa “Servicios pecuarios Véliz”, del cantón Quevedo.**

**AUTORA**

**KARLA ELIZABETH VÉLIZ CASTRO**

**DIRECTORA**

**ING. LUCRECIA LLERENA GUEVARA MSC.**

**QUEVEDO - LOS RÍOS - ECUADOR**

**2011**

**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**  
**UNIDAD DE ESTUDIOS A DISTANCIA**  
**CARRERA SISTEMAS**

**SISTEMA DE INFORMACIÓN CON APLICACIÓN MÓVIL PARA EL CONTROL ADMINISTRATIVO DE LA EMPRESA “SERVICIOS PECUARIOS VÉLIZ”, DEL CANTÓN QUEVEDO.**

Tesis de Grado presentada al Honorable Comité Técnico Académico Administrativo de la Unidad de Estudios a Distancia como requisito previo para la obtención del título de:

**INGENIERO EN SISTEMAS**

**MIEMBROS DEL TRIBUNAL**

Ing. Nancy Rodríguez Gavilanes MSc .....  
**PRESIDENTE DEL TRIBUNAL**

Ing. Ramiro Rivera García MSc .....  
**MIEMBRO DEL TRIBUNAL**

Ing. Stalin Carreño Sandoya MSc .....  
**MIEMBRO DEL TRIBUNAL**

Ing. Lucrecia Llerena Guevara MSc .....  
**DIRECTORA DE TESIS**

## **DECLARACIÓN**

Yo, **Karla Elizabeth Véliz Castro**, declaro bajo juramento que el trabajo aquí descrito es de mí autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo el derecho de propiedad intelectual correspondiente a este trabajo a la Unidad de Estudios a Distancia de la Universidad Técnica Estatal de Quevedo, según lo establecido por la ley de propiedad intelectual por su reglamento y por la normatividad institucional vigente.

.....  
**Karla Elizabeth Véliz Castro**  
**AUTORA**

## CERTIFICACIÓN

**Ing. Lucrecia Llerena Guevara**, Docente de la Unidad de Estudios a Distancia de la Universidad Técnica Estatal de Quevedo

**Certifica:**

Que la egresada **KARLA ELIZABETH VELIZ CASTRO**, realizó la investigación propuesta y ha cumplido a cabalidad con los requisitos de desarrollo de software y el correspondiente informe técnico de la Tesis de Grado titulada **“SISTEMA DE INFORMACIÓN CON APLICACIÓN MÓVIL PARA EL CONTROL ADMINISTRATIVO DE LA EMPRESA “SERVICIOS PECUARIOS VÉLIZ” DEL CANTÓN QUEVEDO**, bajo mi dirección, acatando las disposiciones reglamentarias establecidas para el efecto.

---

**Ing. Lucrecia Llerena Guevara MSc**  
**DIRECTORA DE TESIS**

## **AGRADECIMIENTO**

El agradecimiento en primer lugar a Dios, que es el creador de la tierra y que dió la vida por nosotros, y estuvo conmigo en todo para culminar una de mis metas proyectadas.

A mis padres que son la guía constante de mi vida y apoyo incondicional para los proyectos que tengo a cumplir.

A la Universidad Técnica Estatal de Quevedo con su Unidad de Estudios a Distancia, que gracias al apoyo y guía constante sirvió de gran ayuda para nuestro desarrollo personal y profesional.

A los docentes de la Unidad de Estudios a Distancia por los conocimientos adquiridos en cada uno de los módulos vistos.

A los miembros del Tribunal y a todas aquellas personas que de una u otra manera contribuyeron para la elaboración del presente trabajo investigativo.

***Karla Elizabeth Véliz Castro***

***Autora.***

## **DEDICATORIA**

Este trabajo está dedicado con mucho amor y sacrificio para Dios y a mis Padres, quienes son el apoyo constante en mis proyectos a cumplir.

**Karla**

## ÍNDICE GENERAL

<b>I.</b>	<b>INTRODUCCIÓN .....</b>	<b>1</b>
1.1.	Planteamiento del Problema .....	2
1.2.	Formulación del Problema .....	2
1.3.	Delimitación del Problema .....	3
1.4.	Justificación y Factibilidad.....	3
1.5.	Objetivos .....	4
1.6.	Hipótesis .....	4
<b>II.</b>	<b>REVISIÓN DE LA LITERATURA .....</b>	<b>5</b>
2.1.	Sistemas de Información.....	5
2.1.1.	Definición de Sistema de Información.....	5
2.2.	Sistema Informático .....	6
2.2.1.	Diferencia entre Sistema Informático y Sistema de Información.....	6
2.3.	Administración .....	8
2.3.1.	Objetivos de la Administración.....	8
2.3.2.	Importancia de la Administración .....	9
2.3.3.	Características de la Administración .....	9
2.3.4.	Control Administrativo.....	10
2.3.5.	Gestión Administrativa .....	11
2.4.	WIFI .....	11
2.4.1.	POCKET PC .....	12
2.4.2.	NET FRAMEWORK .....	13
2.4.3.	La Librería de Clase Base (BCL) .....	14
2.5.	Base de Datos .....	15
2.5.1.	Definición de Base de Datos.....	15
2.5.2.	Características .....	16

2.5.3.	Sistema de Gestión de Base de Datos (SGBD).....	16
2.5.4.	Ventajas de las Bases de Datos .....	16
2.5.5.	Control sobre la Redundancia de Datos .....	18
2.5.6.	Consistencia de Datos .....	18
2.5.7.	Compartición de Datos.....	18
2.5.8.	Mantenimiento de Estándares .....	19
2.5.9.	Mejora en la Integridad de Datos .....	19
2.5.10.	Mejora en la Seguridad .....	19
2.5.11.	Mejora en la Accesibilidad a los Datos.....	19
2.5.12.	Mejora en la Productividad.....	20
2.5.13.	Mejora en el Mantenimiento.....	20
2.5.14.	Aumento de la Concurrencia.....	21
2.5.15.	Mejora en los Servicios de Copias de Seguridad.....	21
2.5.16.	Desventajas de las Bases de Datos.....	21
2.5.17.	Coste del Equipamiento Adicional.....	21
2.5.18.	Vulnerable a los Fallos.....	22
2.5.19.	Tipos de Campos.....	22
2.5.20.	Tipos de Base de Datos .....	23
2.6.	Access.....	23
2.6.1.	Concepto de Base de Datos.....	23
2.6.2.	Características de Access.....	24
2.6.3.	Elementos Fundamentales.....	25
2.7.	MySQL.....	26
2.8.	Sql Server .....	27
2.8.1.	Introducción a las Bases de Datos de SQL SERVER.....	27
2.8.2.	Almacenamiento de los Datos .....	27
2.8.3.	Archivos de Bases de Datos de SQL Server .....	27
2.8.4.	Tipos de archivos de las Bases de Datos .....	28
2.9.	Visual Studio .NET .....	28
2.9.1.	Introducción a Visual Studio .NET.....	28

2.9.2.	Microsoft .Net Framework .....	29
2.9.3.	La Plataforma Microsoft .Net.....	30
2.9.4.	MARCO .Net.....	30
2.9.5.	Componentes del Marco .Net .....	30
2.9.6.	Common Language Runtime (CLR).....	30
2.9.7.	Tipos de Ejecutables.....	31
2.9.8.	¿Qué es un Namespace (o espacio de nombres)?.....	32
2.9.9.	¿Qué es un Assembly (o ensamblado)? .....	32
2.9.10.	Creación de Componentes .NET .....	33
2.9.11.	Creación de Archivos XML.....	33
2.9.12.	Lenguajes en .NET .....	34
2.10.	Ingeniería de Software .....	34
2.10.1.	¿Qué es la Ingeniería de Software?.....	34
2.10.2.	Definición de Ingeniería del Software.....	34
2.10.3.	Definición: Ingeniería .....	35
2.10.4.	Definición de Software .....	36
2.10.5.	Fases de la Ingeniería de Software.....	36
2.10.6.	Metodología de Desarrollo de Software .....	38
2.10.7.	Metodología de Desarrollo Agiles .....	38
2.11.	Metodología Crystal Clear (CC) .....	39
2.11.1.	Desarrollo de la Metodología .....	39
2.11.2.	Qué es Crystal Clear.....	39
2.11.3.	Definiciones .....	41
2.11.4.	Características .....	41
2.12.	Empresas Pecuarías.....	48
2.13.	Atención al Cliente .....	49
<b>III.</b>	<b>MATERIALES Y MÉTODOS .....</b>	<b>52</b>
3.1.	Localización y Duración de la Investigación.....	52
3.2.	Materiales y Equipos.....	52

3.3.	Tipo de Investigación .....	53
3.3.1.	Exploratoria.....	53
3.3.2.	Descriptiva .....	53
3.4.	Métodos .....	53
3.4.1.	Analítico .....	53
3.4.2.	Sintético .....	54
3.4.3.	Fuentes Primarias .....	54
3.4.4.	Fuentes Secundarias .....	54
3.5.	Técnicas e Instrucciones de Evaluaciones .....	54
3.5.1.	Entrevista .....	54
3.5.2.	Encuesta .....	55
3.6.	Población y Muestra.....	55
3.6.1.	Población .....	55
3.6.2.	Muestra .....	55
3.7.	Procedimiento Metodológico .....	56
<b>IV.</b>	<b>RESULTADOS .....</b>	<b>60</b>
4.1.	Situación actual de la Empresa.....	60
4.2.	Resultados de las Encuestas.....	61
4.2.1.	Encuesta a los Clientes.....	61
4.2.1.1.	Inconvenientes en la empresa por la atención.....	61
4.2.1.2.	Molestias por la atención manual en la empresa .....	61
4.2.1.3.	Control de tratamiento de su mascota .....	62
4.2.1.4.	Mejoras en la empresa.....	63
4.2.1.5.	Implementar un sistema para una mejor atención .....	63
4.2.2.	Encuesta a los Proveedores .....	64
4.2.2.1	Productos que adquiere la empresa a los proveedores .....	64
4.2.2.2	Formas de pagos a proveedores .....	65
4.2.2.3	La atención de la empresa.....	65
4.2.2.4	Tiempo que la empresa Servicios Pecuarios Véliz realiza sus pedidos .....	66

4.2.2.5	Empresa automatizada con un Sistema Informático.....	66
4.2.3.	Resultado de la Entrevista .....	67
4.2.3.1	Entrevista al Gerente Propietario .....	67
4.2.4.	Demostración de Hipótesis .....	68
<b>V.</b>	<b>DISCUSIÓN .....</b>	<b>70</b>
<b>VI.</b>	<b>PROPUESTA.....</b>	<b>72</b>
6.1.	Sistema de Información con Aplicación Móvil para el Control Administrativo de la Empresa Servicios Pecuarios Veliz .....	72
6.2.	Justificación .....	72
6.3.	Documentación del Sistema .....	74
6.3.1.	Documentos de Requerimientos.....	74
6.2.2.	Caso de Uso .....	75
6.2.2.2.	Casos de uso Expandido .....	76
6.2.3.	Descripción del Caso de Uso Expandido .....	79
6.2.4.	Diagramas de Actividad .....	86
6.2.5.	Diagramas de Secuencia .....	90
6.2.6.	Diagrama Físico de la Base de Datos.....	93
6.2.7.	Diagrama Entidad Relación .....	94
6.2.8.	Diagrama de Clases.....	95
<b>VII.</b>	<b>CONCLUSIONES .....</b>	<b>96</b>
<b>VIII.</b>	<b>RECOMENDACIONES.....</b>	<b>97</b>
<b>IX.</b>	<b>RESUMEN .....</b>	<b>98</b>
<b>X.</b>	<b>SUMARY .....</b>	<b>99</b>
<b>XI.</b>	<b>BIBLIOGRAFIA .....</b>	<b>100</b>

## ÍNDICE CUADROS

Cuadro 1. Inconveniente con el dueño de la empresa o colaborador por la atención. ....	61
Cuadro 2. Malestar en los clientes sobre la atención manual que reciben en la empresa.....	62
Cuadro 3. A los clientes si se le facilita el control de tratamiento a su animal. .	62
Cuadro 4. Que se debería mejorar en la empresa.....	63
Cuadro 5. Implementación de un sistema en la empresa. ....	64
Cuadro 6. Los proveedores llevan el registro de adquisición de sus productos	64
Cuadro 7. Que forma de pago le brinda a la empresa Servicios Pecuarios Veliz. ....	65
Cuadro 8. Cómo califica la atención a la empresa Servicios Pecuarios Veliz...	65
Cuadro 9. Que plazo de tiempo la empresa Servicios Pecuarios Véliz realiza sus pedidos.....	66
Cuadro 10. Está de acuerdo que se automatice con un Sistema Informático la empresa Servicios Pecuarios Veliz.....	66
Cuadro 11. Caso de Uso Inicio de Sesión .....	79
Cuadro 12. Caso de Uso Cambiar Clave .....	80
Cuadro 13. Caso de Uso Clientes.....	81
Cuadro 14. Caso de Uso Mascota .....	82
Cuadro 15. Caso de Uso Medicamentos .....	83
Cuadro 16. Caso de Uso Usuario .....	84
Cuadro 17. Caso de Uso Realizar Control de Mascota .....	85

## INDICE DE FIGURAS

Figura 1. Caso de Uso General .....	75
Figura 2. Caso de Uso Expandido Inicio de Sesión .....	76
Figura 3. Caso de Uso Expandido Cambiar Clave.....	76
Figura 4. Caso de Uso Expandido Registrar Clientes.....	77
Figura 5. Caso de Uso Expandido Registrar Mascota .....	77
Figura 6. Caso de Uso Expandido Registrar Medicamentos .....	77
Figura 7. Caso de Uso Expandido Usuario.....	78
Figura 8. Caso de Uso Expandido Registrar Control de Mascota.....	78
Figura 9. Diagrama de Actividad Inicio de Sesión.....	86
Figura 10. Diagrama de Actividad Cambiar Clave .....	86
Figura 11. Diagrama de Actividad Registrar Cliente .....	87
Figura 12. Diagrama de Actividad Registrar Mascota.....	87
Figura 13. Diagrama de Actividad Registrar Medicamentos .....	88
Figura 14. Diagrama de Actividad Crear Usuario.....	88
Figura 15. Diagrama de Actividad Registrar Control de Mascota .....	89
Figura 16. Diagrama de Secuencia Inicio de Sesión .....	90
Figura 17. Diagrama de Secuencia Cambiar Clave .....	90
Figura 18. Diagrama de Secuencia Registrar Mascota.....	91
Figura 19. Diagrama de Secuencia Registrar Medicamentos .....	91
Figura 20. Diagrama de Secuencia Crear Usuario .....	92
Figura 21. Diagrama de Secuencia Realizar Control .....	92
Figura 22. Diagrama Físico de la Base de Datos.....	93
Figura 23. Diagrama Entidad Relación .....	94
Figura 24. Diagrama de Clases .....	95

## I. INTRODUCCIÓN

Los Sistemas Informáticos como todo sistema, es el conjunto de partes interrelacionadas, que permite almacenar y procesar información, consisten en procesadores, memoria, sistemas de almacenamiento externo, etc. siendo especialmente importante los sistemas de gestión de bases de datos. Un sistema informático normal emplea un sistema que utiliza dispositivos que se usan para programar y almacenar programas y datos.

Los asistentes personales digitalizados como es el Pocket Pc son una plataforma de gran potencial, ya que poseen capacidad de procedimiento poderoso a un precio bajo, en el área de desarrollo de aplicaciones móviles. Este fenómeno está ligado al hecho de que los detalles, documentación y herramientas de desarrollo estén disponibles en el momento oportuno, permitiendo a las empresas experimentar con un dispositivo móvil de fácil manejo para ingresar datos, sin necesidad de estar frente al computador.

Actualmente en el cantón Quevedo, Provincia de Los Ríos se encuentra la empresa Servicios Pecuarios Véliz, que oferta servicios de veterinaria como tratamientos en: caninos, bovinos, porcinos y emergencias, atendiendo en la salud de los animales aplicando vacunas y vitaminas a las mascotas, en el área administrativa existen falencias por la carencia de un sistema informático que permita obtener información oportuna para cumplir con eficiencia y eficacia en la atención a los clientes.

Por esta razón se propone desarrollar un sistema de información con aplicación móvil para el control administrativo, capaz de controlar ingresos y egresos de productos, registrar clientes y mascotas, registro y movimientos de medicamentos, consultas, reportes de clientes, medicamentos y mascotas, aplicando una tecnología avanzada con el dispositivo móvil que le permitirá al gerente llevar los datos de manera ordenada al concluir la jornada laboral.

## **1.1. Planteamiento del Problema**

La empresa Servicios Pecuarios Véliz carece de un sistema informático, los datos se manejan de forma manual lo que genera pérdida de tiempo y malestar en sus clientes por no mantener información adecuada disponible.

La historia clínica de las mascotas son llenados en carpetas de forma manual en donde constan las fichas médicas, muchas veces el veterinario no recuerda la fecha del periodo de vacunación, por la inmensa cantidad de mascotas por atender, y la poca disponibilidad de tiempo para encontrar la información.

El veterinario realiza visitas diarias a domicilio a los clientes cuyas mascotas estén en tratamiento, se traslada con un cuaderno para anotar las indicaciones respectivas de las mascotas, al llegar al consultorio vuelve a escribir la misma información en la ficha médica.

No hay un control de inventario del ingreso y egreso de productos, la falta de medicamentos ocasiona que las mascotas no sean tratadas de la manera correcta en el momento requerido.

Con la implementación de un sistema se dará solución al problema, el mismo que permitirá mantener actualizada la información de los procesos y servicios de la empresa, para facilitar y agilizar las funciones administrativas del gerente propietario.

## **1.2. Formulación del Problema**

¿Como incide la carencia de un Sistema de Información en la empresa Servicios Pecuarios Véliz, en la atención al cliente?

### **1.3. Delimitación del Problema**

**Campo:** Administrativo

**Área:** Informática

**Objeto:** Desarrollo de Aplicación

**Espacio:** Empresa Servicios Pecuarios Véliz del cantón Quevedo

**Año:** 2011

### **1.4. Justificación y Factibilidad**

En la actualidad los dispositivos móviles aportan diferentes ventajas que sirven como medio de comunicación, son portables con mayores recursos, con acceso a red y por último, es posible desarrollar aplicaciones de acuerdo a las necesidades de cada usuario y cargarlas en los mismos.

La presente tesis consiste en la implementación de un sistema de información con aplicación móvil para el control administrativo de la empresa Servicios Pecuarios Véliz, será de vital importancia por cuanto el gerente utilizará la sincronización de datos en sus consultas a domicilio, los servicios estarán enfocados a las funcionalidades y capacidades de dispositivo móvil.

Con la implementación de este sistema se obtendrá un mejor servicio logrando un impacto que contará con una aplicación novedosa que beneficiará a los clientes con un sistema automatizado y acorde a las necesidades de la empresa, mejorará el procesamiento de datos, brindando agilidad en la atención a los clientes, mostrando un diseño propio que le ayudará a solucionar los problemas y proyectarse al futuro.

La elaboración de este sistema es factible, por cuanto la empresa facilitó toda la información necesaria y requerida.

## **1.5. Objetivos**

### **General**

- Crear un Sistema de Información con Aplicación Móvil para el Control Administrativo de la empresa “Servicios Pecuarios Veliz”, del cantón Quevedo.

### **Específicos**

- Analizar los procesos de flujo de información utilizados actualmente en la empresa.
- Identificar los procesos de flujo de información que se utilizará para el nuevo sistema.
- Determinar el DBMS óptimo para almacenar la información.
- Desarrollar el sistema para el Control Administrativo de la empresa aplicando la herramienta más adecuada.

## **1.6. Hipótesis**

La implementación de un Sistema de Información con Aplicación Móvil para el control administrativo de la empresa Servicios Pecuarios Véliz mejorará la atención a los clientes.

## II. REVISIÓN DE LA LITERATURA

### 2.1. Sistemas de Información

**Fernández (2006).** En la actualidad el desarrollo tecnológico es constante, y por estas variaciones dinámicas en el ambiente en el que se desarrollan las organizaciones modernas, se plantea la necesidad de adaptarse a transformaciones de manera inmediata.

La Alta Dirección empresarial durante muchos años, han reconocido que la información contable tradicional utilizada para el cálculo de utilidades ha tenido un valor limitado para el control. Sin embargo existen muchas empresas donde estos son los únicos datos que se recopilan periódicamente.

La empresa en cada una de las áreas de funcionamiento se ve en la necesidad de tomar decisiones, desde la más insignificante hasta aquella que tiene que ver con su existencia misma. Con tal responsabilidad se necesita una base de datos lo más certera posible sobre la cual descansen esas decisiones.

Como es de suponer, esta información necesaria para tomar las decisiones no aparece casualmente, sino que es recopilada mediante un sistema informativo que abarca su recepción, elaboración, registro y transmisión.

#### 2.1.1. Definición de Sistema de Información

**Fernández (2006).** Un Sistema de Información (SI) es el conjunto formal de procesos de análisis, que operando sobre una colección de datos estructurados de acuerdo a las necesidades de la empresa, recopila elabora y distribuye, la información necesaria para realizar las operaciones básicas y la toma de decisión en la empresa, que nos sirva para desempeñar las funciones de negocio de la empresa de acuerdo con sus estrategias.

De esta manera se ve la necesidad de aprovechar los adelantos informáticos y tecnológicos para automatizar el funcionamiento de los Sistemas de Información y de esta manera disminuir su complejidad para los usuarios de dichos sistemas.

## **2.2. Sistema Informático**

**Alegsa (2010).** Un sistema informático es un conjunto de partes que funcionan relacionándose entre sí con un objetivo preciso. Sus partes son: hardware, software y las personas que lo usan.

Por ejemplo, una computadora, sus dispositivos periféricos y la persona que la maneja, pueden constituir un sistema informático.

Un sistema informático puede formar parte de un sistema de información; en este último la información, uso y acceso a la misma, no necesariamente está informatizada. Por ejemplo, el sistema de archivo de libros de una biblioteca y su actividad en general es un sistema de información. Si dentro del sistema de información hay computadoras que ayudan en la tarea de organizar la biblioteca, entonces ese es un sistema informático.

### **2.2.1. Diferencia entre Sistema Informático y Sistema de Información**

**Alegsa (2010).** En un sistema informático se utilizan computadoras para almacenar, procesar y/o acceder a información.

- En un sistema de información se pueden utilizar computadoras, pero no es necesario. El acceso a la información puede ser físico (por ejemplo, una persona se encarga de buscar en un archivador).
- Tanto el sistema informático como el sistema de información, incluyen a las personas que acceden o producen información dentro del sistema. Las

personas tienen que capacitarse para entender el funcionamiento y procedimientos que soporta sistema.

- Ambos Sistemas tienen un propósito. Por ejemplo, gestionar el acceso y distribución de libros una biblioteca, administrar la entrada/salida de mercadería, personal y otros recursos de un comercio, etc.
- Sistema Informático es un conjunto de elementos interconectados o relacionados para el tratamiento de información.
- Estructura de los Sistemas Informáticos tenemos lo siguiente: Sistema Central las funciones son las siguientes:

**Alegsa (2010).** Las Funciones son las siguientes:

**Unidad de Control.** Dirige todas las actividades del ordenador, generando señales de control para el resto de las unidades según el código de operación de la instrucción. La ejecución de la instrucción supone realizar una serie de operaciones elementales a través de la activación de una serie de señales de control.

**Unidad Aritmético-Lógica.** Ejecuta las operaciones aritméticas lógicas que le señala la instrucción residente en la unidad de control.

**Memoria Principal.** Almacena las instrucciones de los programas y los datos que ha de procesar y los resultados que obtiene al ejecutar los programas. Está constituida por celdas o elementos capaces de almacenar 1 bit de información (0 o 1). Se organiza en conjuntos de elementos de un tamaño determinado, denominados palabras.

**Sistema de Entrada/Salida:** Interacciona directamente con el subsistema central a través de los procesadores de entrada/salida. El procesador de entrada/salida gestiona el acceso a la memoria del subsistema central y la

interacción con el controlador o procesador de periféricos. La conexión es a través de un canal por el que se envían datos, señales de control, etc. (bus). El procedimiento de intercambio de información y los protocolos utilizados están estandarizados, permitiendo que controladores de distintos periféricos utilicen el mismo interface.

**Sistema de Comunicaciones:** También son elementos físicos del Sistema Informático los componentes del sistema de comunicaciones: procesadores y líneas de comunicaciones, módems, concentradores, terminales, etc.

## 2.3. Administración

**Álava (2010).** Es un proceso muy particular consistente en las actividades de planeación, organización, ejecución y control, desempeñadas para determinar y alcanzar los objetivos señalados con el uso de seres humanos y otros recursos.

### 2.3.1. Objetivos de la Administración

**Álava (2010).** Para que exista un sentido de satisfacción debe existir un objetivo, lo que da un propósito al esfuerzo; además el objetivo debe tener un significado y valor; así que la definición de objetivo es: “Un objetivo administrativo es una meta que se fija, que requiere de un campo de acción definido y que sugiera la orientación para los esfuerzos de un dirigente”, en esta definición hay cuatro elementos que son:

- Meta
- Campo de acción
- Definición de la Acción
- Orientación

Los Objetivos son importantes para llegar a los resultados deseados; la falta de objetivos hace que la administración sea innecesariamente difícil, si es que se

puede hablar en rigor de administración; así que, los objetivos básicos son un prerrequisito para determinar cualquier curso de acción y deben ser definidos con claridad para que los comprendan todos los miembros de la empresa.

### **2.3.2. Importancia de la Administración**

**Álava (2010).** La administración es un proceso universal ya que no solo se da en los países capitalistas, sino que también en los países socialistas o de cualquier tipo que sean, la administración es importante tanto en las pequeñas como en las grandes empresas.

Otro hecho importante es que por medio de la administración se puede elevar la productividad y los niveles de vida en los países en vías de desarrollo.

La administración imparte efectividad a los esfuerzos humanos. Ayuda a obtener mejor personal, equipo, materiales, dinero y relaciones humanas.

Se mantiene al frente de las condiciones cambiantes y proporciona previsión y creatividad.

### **2.3.3. Características de la Administración**

**Álava (2010).** Dentro de las características de la administración tenemos las siguientes:

**Universalidad:** La administración se da donde quiera que existe un organismo social (estado, ejército, empresas, iglesias, familia, etc.), porque en él tiene siempre que existir coordinación sistemática de medios.

**Especificidad:** La administración tiene sus propias características las cuales son inconfundibles con otras ciencias, aunque va acompañada siempre de ellas (funciones económicas, contables, productivas, mecánicas, jurídicas, etc.), son completamente distintas.

**Unidad Temporal:** Aunque se distingan etapas, fases y elementos del proceso administrativo, éste es único y, por lo mismo, en todo momento de la vida de una empresa se están dando, en mayor o menor grado, todos o la mayor parte de los elementos administrativos.

**Unidad Jerárquica:** Todos cuantos tienen carácter de jefes en un organismo social, participan en distintos grados y modalidades, de la misma administración.

**Valor Instrumental:** La administración es un instrumento para llegar a un fin, ya que su finalidad es eminentemente práctica y mediante ésta se busca obtener resultados determinados previamente establecidos.

**Flexibilidad:** La administración se adapta a las necesidades particulares de cada organización.

**Amplitud de Ejercicio:** Esta se aplica en todos los niveles jerárquicos de una organización.

La administración es importante porque se aplica en cualquier tipo de organización con deseos de aumentar su productividad y el éxito, dependiendo para esto del elemento humano y material.

#### **2.3.4. Control Administrativo**

**Álava (2010).** Control es la función administrativa que consiste en medir y corregir el desempeño de los subordinados para asegurar que los objetivos de la empresa sean alcanzados.

La finalidad del control es garantizar que los resultados de lo que se planeó, organizó y distribuyó se ajusten lo máximo posible a los objetivos preestablecidos, tarea de la que deben estar muy pendientes los ejecutivos que

se encargaron de la planeación y organización. El control es un proceso que guía la actividad ejecutada para alcanzar un fin determinado de antemano. Como proceso, el control presenta fases que deben tomarse en cuenta en todo momento. Los puntos estrictamente necesarios en cualquier sistema de control son:

- Un objetivo, un fin predeterminado, un plan, una línea de acción, un estándar, una norma, una regla decisoria, un criterio, una unidad de medida.
- Un medio para medir la actividad desarrollada.
- Un procedimiento para comparar tal actividad con el criterio fijado.
- Algún mecanismo que corrija la actividad en curso, para alcanzar los resultados deseados

### **2.3.5. Gestión Administrativa**

**Álava (2010).** La gestión administrativa en una empresa se encarga de realizar estos procesos recién mencionados utilizando todos los recursos que se presenten en una empresa con el fin de alcanzar aquellas metas que fueron planteadas al comienzo de la misma. Se trata de un proceso para realizar las tareas básicas de una empresa sistemáticamente. Un proceso como la gestión administrativa es mucho más fácil de comprender descomponiéndolo en partes, identificando aquellas relaciones básicas; este tipo de modalidad de descripción se denomina modelo y se utiliza para representar aquellas relaciones complejas.

### **2.4. WIFI**

**Carriel (2006).** WiFi, es la sigla para WirelessFidelity (Wi-Fi), que literalmente significa Fidelidad inalámbrica. Es un conjunto de redes que no requieren de cables y que funcionan en base a ciertos protocolos previamente establecidos.

Si bien fue creado para acceder a redes locales inalámbricas, hoy es muy frecuente que sea utilizado para establecer conexiones a Internet.

WiFi es una marca de la compañía Wi-Fi Alliance que está a cargo de certificar que los equipos cumplan con la normativa vigente (que en el caso de esta tecnología es la IEEE 802.11).

Esta nueva tecnología surgió por la necesidad de establecer un mecanismo de conexión inalámbrica que fuera compatible entre los distintos aparatos. En busca de esa compatibilidad fue que en 1999 las empresas 3com, Airones, Intersil, Lucent Technologies, Nokia y Symbol Technologies se reunieron para crear la Wireless Ethernet CompabilityAliance (WECA), actualmente llamada Wi-Fi Alliance.

Al año siguiente de su creación la WECA certificó que todos los aparatos que tengan el sello WiFi serán compatibles entre sí ya que están de acuerdo con los criterios estipulados en el protocolo que establece la norma IEEE 802.11. En concreto, esta tecnología permite a los usuarios establecer conexiones a Internet sin ningún tipo de cables y puede encontrarse en cualquier lugar que se haya establecido un "punto caliente" o hotspotWiFi.

#### **2.4.1. POCKET PC**

**Alegsa (2007).**P/PC, PPC. Es una especificación para computadorasPDA, que ejecutan alguna versión del sistema operativoWindows CE. Pueden tener la capacidad de ejecutar un sistema operativo alternativo como NetBSD o Linux. Estas minicomputadoras tienen muchas de las capacidades de las computadoras de escritorio modernas.

Según el propio Microsoft, las Pocket PC son un "dispositivo de mano que permite a los usuarios almacenar y recibir e-mails, contactar personas, ejecutar archivos multimedia, juegos, intercambiar mensajes de texto con MSN, navegar por la web y más".

Cualquier dispositivo clasificado como Pocket PC debería:

- \* Ejecutar un Windows para dispositivos móviles.
- \* Venir empaquetado con una suite específica de aplicaciones en memoria ROM.
- \* Incluir una pantalla táctil.
- \* Incluir un touchpad.
- \* Estar basado en una CPU ARM versión 4, Intel XScale, MIPS o SH3.

#### **2.4.2. NET FRAMEWORK**

**Arteaga (2005)**. Para aprovechar al máximo las características de .Net es necesario entender la arquitectura básica en la que esta implementada esta tecnología y así beneficiarse de todas las características que ofrece esta nueva plataforma.

.NET Framework es el corazón de la tecnología **.NET**. Es el marco de trabajo y ejecución **común** a toda la tecnología **.NET**. Es por lo tanto un elemento indispensable dentro de la tecnología **.NET**.

El **.NET Framework** es un componente de software que se instala en el sistema operativo. Podemos descargar el **.NET Framework** desde el sitio web de Microsoft:

.NET Framework 1.1

.NET Framework 2.0

Dentro del **.NET framework** están integrados los lenguajes **.NET (C#, VB.NET, J# )**, el **CRL**, el **CLS**, **ADO.NET**, **ASP.NET ...**

.NET soporta varios lenguajes de programación, siendo los más populares:

**C#**

**VB.NET**

**C++** controlado

**J#**

**Delphi**

Todos ellos trabajan perfectamente integrados con el .NET framework, siendo **C#** el lenguaje principal, ya que es el único que ha sido desarrollado específicamente para .NET. Gran parte de la biblioteca de clases que componen el framework están escritas en **C#**.

**.NET Framework** es el conjunto de espacios de trabajo y clases que sirven de base a los lenguajes .NET. Es la evolución de las API de Windows y de la fundación de clases de Microsoft MFC hacia la completa encapsulación. Cualquier programa que desarrollemos utilizando la tecnología .NET, independientemente del lenguaje utilizado, estará basado en .NET Framework.

### **2.4.3. La Librería de Clase Base (BCL)**

**Arteaga (2005)**. La Librería de Clase Base (BCL) es una librería incluida en el .NET Framework formada por cientos de tipos de datos que permiten acceder a los servicios ofrecidos por el CLR y a las funcionalidades más frecuentemente usadas a la hora de escribir programas. Además, a partir de estas clases prefabricadas el programador puede crear nuevas clases que mediante herencia extiendan su funcionalidad y se integren a la perfección con el resto de clases de la BCL. Por ejemplo, implementando ciertos interfaces podemos crear nuevos tipos de colecciones que serán tratadas exactamente igual que cualquiera de las colecciones incluidas en la BCL.

Esta librería está escrita en MSIL, por lo que puede usarse desde cualquier lenguaje cuyo compilador genere MSIL. A través de las clases suministradas en ella es posible desarrollar cualquier tipo de aplicación, desde las

tradicionales aplicaciones de ventanas, consola o servicio de Windows NT hasta los novedosos servicios Web y páginas ASP.NET. Es tal la riqueza de servicios que ofrece que incluso es posible crear lenguajes que carezcan de librería de clases propia y sólo se basen en la BCL -como C#. Dada la amplitud de la BCL, ha sido necesario organizar las clases en ella incluida en espacios de nombres que agrupen clases con funcionalidades similares.

## 2.5. Base de Datos

### 2.5.1. Definición de Base de Datos

**Charle (2008).** Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. A continuación te presentamos una guía que te explicará el concepto y características de las bases de datos.

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una **base de datos** se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada.

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

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

Base de Datos es el conjunto de informaciones almacenadas en un soporte legible por ordenador y organizadas internamente por registros y campos.

### 2.5.2. Características

**Charte (2008).** Entre las principales características de los sistemas de base de datos podemos mencionar:

Independencia lógica y física de los datos.

Redundancia mínima.

Acceso concurrente por parte de múltiples usuarios.

Integridad de los datos.

Consultas complejas optimizadas.

Seguridad de acceso y auditoría.

Respaldo y recuperación.

Acceso a través de lenguajes de programación estándar.

### 2.5.3. Sistema de Gestión de Base de Datos (SGBD)

**Charte (2008).** Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Los Sistemas de Gestión de Base de Datos permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

### 2.5.4. Ventajas de las Bases de Datos

**Charte (2008).** Las ventajas de los sistemas de bases de datos, son entre otras las siguientes:

**Obtener más información de la misma cantidad de data.-** La base de datos facilita al usuario obtener más información debido a la facilidad que provee esta estructura para proveer datos a los usuarios (si se tiene el privilegio).

**Compartir los Datos.-** Usuarios de distintas oficinas pueden compartir datos si están autorizados. Esto implica que si un dato cambia de contenido como por ejemplo la dirección de un cliente, todos los usuarios que pueden acceder ese dato, verán inmediatamente el cambio efectuado.

**Balance de Requerimientos Conflictivos.-** Para que la Base de Datos trabaje apropiadamente, necesita de una persona o grupo que se encargue de su funcionamiento.

**Se refuerza la estandarización** - Debido a lo que se mencionó previamente, es más fácil estandarizar procesos, formas, nombres de datos, formas, etc.

**Charte (2008).** Entre otras características que tienen las ventajas de datos, a continuación se detalla las siguientes:

**Redundancia controlada.-** Debido al sistema tradicional de archivos independientes, los datos se duplicaban constantemente lo cual creaba mucha duplicidad de datos y creaba un problema de sincronización cuando se actualizaba un dato en un archivo en particular.

**Consistencia.-** Al controlarse la redundancia, cuando actualizas un dato, todos los usuarios autorizados de la Base de Datos pueden ver el cambio independientemente de que estén trabajando en distintos sistemas.

**Integridad.-** La base de datos tiene la capacidad de validar ciertas condiciones cuando los usuarios entran datos y rechazar entradas que no cumplan con esas condiciones. El DBA (Data Base Administrator) es responsable de establecer esas validaciones.

**Seguridad** - El DBA al tener control central de los Datos, la Base de Datos le provee mecanismos que le permiten crear niveles de seguridad para distintos tipos de Usuarios.

### **2.5.5. Control sobre la Redundancia de Datos**

**Charte (2008).** Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos.

En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos.

Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

### **2.5.6. Consistencia de Datos**

**Charte (2008).** Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente.

Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes.

### **2.5.7. Compartición de Datos**

**Charte (2008).** En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

En la empresa que se va a aplicar el software la base de datos solo va a ser compartida por el gerente propietario.

### **2.5.8. Mantenimiento de Estándares**

**Charte (2008).** Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

La empresa mantiene un estándar de documentación y de actualización para de esta manera llevar el control diario que necesita.

### **2.5.9. Mejora en la Integridad de Datos**

**Charte (2008).** La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar.

Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargarse de mantenerlas.

### **2.5.10. Mejora en la Seguridad**

**Charte (2008).** La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

En la empresa se mantiene la protección y seguridad necesaria para que no se escape ninguna información, previo a la autorización del propietario.

### **2.5.11. Mejora en la Accesibilidad a los Datos**

**Charte (2008).** Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de

consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

El software de la empresa va ser diseñado para que tenga un reporte o historial de los clientes y mascotas atendidos diariamente.

#### **2.5.12. Mejora en la Productividad**

**Charte (2008).**El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación.

El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

#### **2.5.13. Mejora en el Mantenimiento**

**Charte (2008).**En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan.

Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones.

Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

#### **2.5.14. Aumento de la Concurrencia**

**Charte (2008).**En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad.

La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

#### **2.5.15. Mejora en los Servicios de Copias de Seguridad**

**Charte (2008).**Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

#### **2.5.16. Desventajas de las Bases de Datos**

**Charte (2008).**Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

#### **2.5.17. Coste del Equipamiento Adicional**

**Charte (2008).**Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para

alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD.

Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

#### **2.5.18. Vulnerable a los Fallos**

**Charte (2008).** El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse.

Es por ello que deben tenerse copias de seguridad (Backup).

#### **2.5.19. Tipos de Campos**

**Charte (2008).** Cada Sistema de Base de Datos posee tipos de campos que pueden ser similares o diferentes. Entre los más comunes podemos nombrar:

**Numérico:** entre los diferentes tipos de campos numéricos podemos encontrar enteros “sin decimales” y reales “decimales”.

**Booleanos:** poseen dos estados: Verdadero “Si” y Falso “No”.

**Memos:** son campos alfanuméricos de longitud ilimitada. Presentan el inconveniente de no poder ser indexados.

**Fechas:** almacenan fechas facilitando posteriormente su explotación. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.

**Alfanuméricos:** contienen cifras y letras. Presentan una longitud limitada (255 caracteres).

**Autoincrementables:** son campos numéricos enteros que incrementan en una unidad su valor para cada registro incorporado. Su utilidad resulta: Servir de identificador ya que resultan exclusivos de un registro.

Es necesario aplicar estos tipos de campos para la realización de las bases de datos.

## **2.5.20. Tipos de Base de Datos**

**Charte (2008).** Entre los diferentes tipos de base de datos, podemos encontrar los siguientes:

**MySql (2007):** es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.

**PostgreSql y Oracle:** Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.

**Access:** Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa Access, el cual crea un archivo .mdb con la estructura ya explicada.

**Microsoft SQL Server:** es una base de datos más potente que Access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones.

## **2.6. Access**

### **2.6.1 Concepto de Bases de Datos**

Una base de datos de Microsoft Access es un conjunto de información relacionada con un tema o propósito particular.

El primer paso que debe seguir para la creación de bases de datos consiste en crear tablas. Cualquiera que sea la información que tenga, necesita una o más tablas en la base de datos para almacenar dicha información. Una vez que tenga sus tablas, puede crear consultas, formularios, informes y otros objetos que le ayuden a usar sus datos. También es posible modificar la apariencia o el funcionamiento de un objeto cambiando sus propiedades, así como utilizar los Asistentes y las herramientas de Microsoft Access para crear y modificar objetos.

Sólo puede tener abierta una B.D. de Microsoft Access a la vez, sin embargo, es posible abrir varias tablas al mismo tiempo en una B.D.

### **2.6.2 Características de ACCESS.**

Microsoft Access es un sistema interactivo de administración de bases de datos para Windows. Access tiene la capacidad de organizar, buscar y presentar la información resultante del manejo de sus bases de datos. Entre sus principales características se encuentran:

Access es gráfico, por lo que aprovecha al máximo la potencia gráfica de Windows, ofreciendo métodos usuales de acceso a los datos y proporcionando métodos simples y directos de trabajar con la información.

Access facilita la administración de datos, ya que sus posibilidades de consulta y conexión le ayudan a encontrar rápidamente la información deseada, cualquiera que sea su formato o lugar de almacenamiento.

Con Access es posible producir formularios e informes sofisticados y efectivos, así como gráficos y combinaciones de informes en un solo documento.

Access permite lograr un considerable aumento en la productividad mediante el uso de los asistentes y las macros. Estos permiten automatizar fácilmente muchas tareas sin necesidad de programar.

### 2.6.3 Elementos fundamentales

Una B.D. de Access no es sólo una tabla de datos, sino que es un conjunto de objetos. Access le permite crear formularios, informes y otros objetos que le ayudan a presentar sus datos tal como lo desee, pero la información propiamente dicha, se almacena en tablas.

#### **Tablas**

Es un conjunto de datos acerca de un tema específico. Los datos de la tabla se representan en columnas (llamadas campos) y filas (llamadas registros). En una tabla, un campo es una categoría o tipo de información.

#### **Consultas**

Es una pregunta que Ud. plantea acerca de la información contenida en su base de datos, como por ejemplo “¿Qué productos tienen proveedores australianos?”. Los datos que responden a la pregunta pueden provenir de una tabla o de varias; la consulta reúne la información solicitada.

#### **Formularios**

Un **formulario** es, por lo general, un buen diseño para introducir, cambiar y ver los registros de su base de datos. Al abrir un formulario, Access recupera los datos deseados de las tablas y los presenta de acuerdo con su diseño ya sea en la pantalla o en formato impreso. En un formulario se muestran determinados registros, con un diseño personalizado.

#### **Informes**

Se utiliza para representar los datos en una página impresa y para mostrar los subtotales y totales correspondientes a todo un conjunto de registros. Los informes se pueden presentar con un diseño personalizado.

## 2.7. MySQL

**MySQL** es un gestor de base de datos sencillo de usar y increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

Las características principales de MySQL son:

- Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.
- Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.
- Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.
- Existe una gran cantidad de software que la usa.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

## **2.8. Sql Server**

### **2.8.1. Introducción a las Bases de Datos de SQL SERVER**

**Charte (2008).** Cuando se crea una base de datos, se configura la estructura de almacenamiento de los datos. Esta estructura incluye al menos un archivo de datos y un archivo de registro de transacciones. Antes de crear una base de datos, debe entender cómo almacena los datos Microsoft SQL Server, así como la función del registro de transacciones.

### **2.8.2. Almacenamiento de los Datos**

**Charte (2008).** Cuando cree una base de datos, tendrá una perspectiva mejor de los planes de la capacidad, la integridad de los datos y el rendimiento si comprende cómo almacena los datos Microsoft SQL Server.

### **2.8.3. Archivos de Bases de Datos de SQL Server**

**Charte (2008).** Las bases de datos de SQL Server se almacenan en archivos. Estos archivos se generan automáticamente cuando se crea una base de datos. Existe más de un tipo de archivo y cada base de datos utiliza siempre al menos dos archivos.

## 2.8.4. Tipos de archivos de las Bases de Datos

**Charte (2008).** Los archivos de bases de datos pueden ser:

Archivo de datos principal. Este archivo constituye el punto de partida de la base de datos. Todas las bases de datos deben tener un archivo de base de datos principal y sólo uno. Los nombres de archivos de datos principales normalmente tienen la extensión.

Archivos de registro. Estos archivos contienen toda la información de registro de transacciones que se utiliza para recuperar la base de datos. Todas las bases de datos tienen al menos un archivo de registro. Los nombres de archivos de registro normalmente tienen la extensión.

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

## 2.9. Visual Studio .NET

### 2.9.1. Introducción a Visual Studio .NET

**Charte (2006).** Visual Studio .NET es la Herramienta Rápida de Desarrollo (RAD) de Microsoft para la siguiente generación de Internet que son los Servicios Web XML. Esta herramienta permite la creación de aplicaciones usando el Marco .NET, es decir usando el CLR, la Librería de Clases, ADO .NET, ASP .NET, etc.

Es un software que brinda las herramientas necesarias para crear, distribuir, administrar y dar mantenimiento a aplicaciones Web distribuidas que usan Servicios Web XML, todo esto con una gran facilidad, rapidez y bajo costo.

Se puede crear aplicaciones Web directamente usando el Framework .NET y algún programa editor, por ejemplo el Bloc de Notas, pero el tiempo que

llevaría el desarrollo no justificaría el ahorro de costos, en cambio, si se utiliza una herramienta como Visual Studio .NET el tiempo de desarrollo se reduciría enormemente.

Visual Studio .NET permite también la integración y el uso cruzado de lenguajes de programación: Visual Basic .NET, Visual C# .NET, Visual C++ .NET y JScript .NET

A diferencia de la versión anterior no existe Visual Interdev, ni Visual J++, además Visual Foxpro .NET no comparte las características unificadas del Marco .NET

Visual Studio.net es un software que brinda herramientas necesarias para crear, distribuir, administrar y dar mantenimiento a aplicaciones Web, con una gran facilidad, rapidez y bajo costo.

## **2.9.2. Microsoft .Net Framework**

**Charte (2006).** Lo que se dice en el eBook Microsoft .NET Framework. ".NET Framework es un entorno para construir, instalar y ejecutar servicios Web y otras aplicaciones. Se compone de tres partes principales: el CommonLanguageRuntime, las clases Framework y ASP.NET", lo que dice el MSDN Library.

El .NET Framework es un entorno multi-lenguaje para la construcción, distribución y ejecución de Servicios Webs y aplicaciones."

El .NET Framework es el corazón de .NET, cualquier cosa que queramos hacer en cualquier lenguaje .NET debe pasar por el filtro cualquiera de las partes integrantes del .NET Framework.

El .NET Framework es una nueva plataforma diseñada para simplificar el desarrollo de aplicaciones en el entorno distribuido de Internet.

### **2.9.3. La Plataforma Microsoft .Net**

**Charte (2006).** La plataforma Microsoft .NET es un conjunto de tecnologías diseñadas para transformar el Internet, a que este sea completamente escalable. Esto proporciona nuevas formas de construir aplicaciones a partir de las colecciones de Servicios Web.

La plataforma Microsoft .NET está compuesta por:

- El Marco .Net
- Los Bloques de Servicios .Net
- Visual Studio .Net
- Los Servidores Empresariales .Net

### **2.9.4. MARCO .Net**

**Charte (2006).** Es la parte central de la plataforma Microsoft .Net y consiste en un compilador común llamado CLR (CommonLanguageRuntime).

### **2.9.5. Componentes del Marco .Net**

- ASP .Net: Formularios Web Servicios Web
- Windows Form: Interfaz de usuario
- ADO .Net: Datos y XML
- Librería de clases del Marco .Net

### **2.9.6. Common Language Runtime (CLR)**

**Charte (2006).** Conocido como el lenguaje común en tiempo de ejecución. Es el Runtime común que soportan todos los lenguajes del Marco .Net y que realiza la mayor parte de las funciones para que corran las aplicaciones .Net. Es el que se encarga de compilar y ejecutar una aplicación. Principales tareas del CLR:

Cargar las clases y administrar la meta data  
Convierte el lenguaje intermedio Microsoft a código nativo  
Administra la ejecución del código  
Soporte de Thread, control de errores, etc.

El CommonLanguageRuntime (CLR) es una serie de librerías dinámicas (DLLs), también llamadas assemblies, que hacen las veces de las DLLs del API de Windows así como las librerías Runtime de Visual Basic o C++. Cualquier ejecutable depende de una forma u otra de una serie de librerías, ya sea en tiempo de ejecución como a la hora de la compilación. Se acabó eso de que existan dos tipos de ejecutables: los que son autosuficientes y no dependen de librerías externas o los que necesitan de librerías en tiempo de ejecución para poder funcionar, tal es el caso de las versiones anteriores de Visual Basic. Por otro lado, la librería de clases de .NET Framework proporciona una jerarquía de clases orientadas a objeto disponibles para cualquiera de los lenguajes basados en .NET, incluido el Visual Basic. También permite crear Threads o hilos o tramas de ejecución.

El CLR es una serie de librerías usadas en tiempo de ejecución para que nuestros ejecutables o cualquiera basado en .NET puedan funcionar.

### **2.9.7. Tipos de Ejecutables**

**Charte (2006).** Con Visual Basic .NET puedes crear básicamente estos dos tipos de ejecutables: de consola, no gráfico, al estilo del viejo MS-DOS, y gráficos, como los que normalmente estamos acostumbrados a ver en Windows.

Existen otros tipos de aplicaciones que se pueden crear con Visual Basic .NET: aplicaciones ASP.NET, (realmente no es una aplicación o ejecutable, sino un compendio de distintos tipos de elementos...), servicios Web, servicios Windows, etc.

### 2.9.8. ¿Qué es un Namespace (o espacio de nombres)?

**Charte (2006).** Un espacio de nombres es un esquema lógico de nombres para tipos en el que un nombre de tipo simple, como MiTipo, aparece precedido por un nombre jerárquico separado por puntos. [...]"

Un Namespace, es una forma de agrupar clases, funciones, tipos de datos, etc. que están relacionadas entre sí. Por ejemplo, entre los Namespaces que podemos encontrar en el .NET Framework encontramos uno con funciones relacionadas con Visual Basic: Microsoft. Visual Basic. Si te fijas, Microsoft y Visual Basic están separados por un punto, esto significa que Microsoft a su vez es un Namespace que contiene otros "espacios de nombres", tales como el mencionado Visual Basic, CSharp y Win32 con el cual podemos acceder a eventos o manipular el registro del sistema.

Por regla general se deberían agrupar en un Namespace funciones o clases que estén relacionadas entre sí. De esta forma, será más fácil saber que estamos trabajando con funciones relacionadas entre sí.

Un Namespace, o espacio de nombre, es un término informático que se utiliza en programación, se refiere a una colección de nombres de entidad definidos por el programador, se puede resumir un Namespace cómo un conjunto de nombres en el cual todos los nombres son únicos.

### 2.9.9. ¿Qué es un Assembly (o ensamblado)?

**Charte (2006).** Un ensamblado es el bloque constructivo primario de una aplicación de .NET Framework. Se trata de una recopilación de funcionalidad que se construye, versiona e instala como una única unidad de implementación (como uno o más archivos). [...]"

Un ensamblado o Assembly puede estar formado por varios ficheros DLLs y EXEs, pero lo más importante es que todos los ensamblados contienen un manifiesto (o manifest), gracias al cual se evitan muchos de los quebraderos de

cabeza a los que Windows nos tiene acostumbrados, al menos en lo referente a las distintas versiones de las librerías y ejecutables, seguramente habrás oído hablar de las DLL Hell (o librerías del demonio) expresión que se usa cuando hay incompatibilidad de versiones entre varias librerías que están relacionadas entre sí.

**Librerías de Clases del Marco .Net**

Conjunto de clases comunes. Unifica los diferentes marcos existentes: APIs, MFC, WFC, VB y VBA. La clase base principal se llama System.

**ADO .Net: Datos y XML**

ADO .Net proporciona un amplio soporte XML.

**ASP .Net: Formularios y Servicios Web**

ASP .Net es un marco de programación, incorporable en el CLR y puede construir aplicaciones Web potentes.

Un Assembly o Ensamblado es una librería dinámica (DLL) en la cual pueden existir distintos espacios de nombres.

#### **2.9.10. Creación de Componentes .NET**

**Charte (2006).** Crear componentes o controles de usuario es tan simple como crear un formulario, ya que usando la herencia se puede pasar todas las características de un objeto a otro, esto está presente en todos los objetos creados en Visual Studio .NET, sean visuales o no.

#### **2.9.11. Creación de Archivos XML**

**Charte (2006).** Con el diseñador de XML, crear un archivo XML es más fácil que nunca, ya que se muestra de colores el código y se auto completan los Tags que uno va escribiendo. Este maneja 3 vistas: XML, esquemas y datos.

### 2.9.12. Lenguajes en .NET

**Charre (2006).** En Visual Studio .NET vienen los siguientes Lenguajes de Programación:

- Visual Basic .NET
- Visual C# .NET
- Visual C++ .NET
- Visual JScript .NET.

## 2.10. Ingeniería de Software

### 2.10.1. ¿Qué es la Ingeniería de Software?

**Quispe (2007).** La Ingeniería del Software es una disciplina o área de la informática o ciencias de la computación, que ofrece método y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo.

Hoy día es cada vez más frecuente la consideración de la Ingeniería del Software como una nueva área de la ingeniería, y el Ingeniero del Software comienza a ser una profesión implantada en el mundo laboral internacional, con derechos, deberes y responsabilidades que cumplir, junto a una, y reconocida consideración social en el mundo empresarial y, por suerte, para esas personas con brillante futuro.

### 2.10.2. Definición de Ingeniería del Software

**Quispe (2007).** Ingeniería del Software es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas de software.

Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar y operar (funcionar) y mantenerlos. Así como también desarrollo de software o producción de software.

La Ingeniería del Software es el establecimiento y uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales.

La Ingeniería de Software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema hasta el mantenimiento de este después que se utiliza.

### **2.10.3. Definición: Ingeniería**

**Quispe (2007).** La ingeniería es el estudio y la aplicación de las distintas ramas de la tecnología. El profesional en este ámbito recibe el nombre de ingeniero.

La actividad del ingeniero supone la concreción de una idea en la realidad. Esto quiere decir que, a través de técnicas, diseños y modelos, y con el conocimiento proveniente de las ciencias, la ingeniería puede resolver problemas y satisfacer necesidades humanas.

La ingeniería también supone la aplicación de la inventiva y del ingenio para desarrollar una cierta actividad. Esto, por supuesto, no implica que no se utilice el método científico para llevar a cabo los planes.

Ingeniería es disponer de la ciencia para analizar un problema, diseñar la solución y trabajar para construirla.

#### **2.10.4. Definición de Software**

**Quispe (2007).** Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

Un sistema de software consiste en diversos programas independientes, archivos de configuración que se utilizan para ejecutar estos programas, un sistema de documentación que describe la estructura del sistema, la documentación para el usuario que explica cómo utilizar el sistema y sitios web que permitan a los usuarios descargar la información de productos recientes.

El software de computadora es el producto que los ingenieros de software construyen y después mantienen en el largo plazo. El software se forma con:

- Las instrucciones (programas de computadora) que al ejecutar se proporcionan las características, funciones y el grado de desempeño deseados.
- Las estructuras de datos que permiten que los programas manipulen información de manera adecuada.
- Los documentos que describen la operación y uso de los programas.

El software no son solo programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta.

#### **2.10.5. Fases de la Ingeniería de Software**

**Quispe (2007).** La Ingeniería de Software requiere llevar a cabo numerosas tareas, dentro de etapas como las siguientes:

**Análisis de Requerimientos.-** Extraer los requisitos y requerimientos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requerimientos incompletos, ambiguos o contradictorios. El resultado del análisis de requerimientos con el cliente se plasma en el documento ERS,

**Especificación de Requerimientos del Sistema,** cuya estructura puede venir definida por varios estándares, tales como CMMI.

**Especificación.-** La Especificación de Requisitos describe el comportamiento esperado en el software una vez desarrollado. Gran parte del éxito de un proyecto de software radicará en la identificación de las necesidades del negocio (definidas por la alta dirección), así como la interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requisitos del software.

**Quispe (2007).** A continuación se detalla las siguientes fases de la Ingeniería de Software consideradas como informarles.

**Arquitectura.-** El Arquitecto de Software es la persona que añade valor a los procesos de negocios gracias a su valioso aporte de soluciones tecnológicas. La Arquitectura de Sistemas en general, es una actividad de planeación, ya sea a nivel de infraestructura de red y hardware, o de Software. La Arquitectura de Software consiste en el diseño de componentes de una aplicación (entidades del negocio), generalmente utilizando patrones de arquitectura.

**Programación.-** Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo y ni la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.

**Prueba.-** Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica el que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un área de pruebas; sin perjuicio de lo anterior el programador debe hacer sus propias pruebas.

**Documentación.-** Todo lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, pasando por modelaciones (UML), diagramas, pruebas, manuales de usuario, manuales técnicos, etc; todo con el propósito de eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones al sistema.

**Mantenimiento.-** Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo inicial del software. Alrededor de 2/3 de toda la ingeniería de software tiene que ver con dar mantenimiento. Una pequeña parte de este trabajo consiste en arreglar errores, o *bugs*.

#### **2.10.6. Metodología de Desarrollo de Software**

**Quispe (2007). Metodología:** Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software.

#### **2.10.7. Metodología de Desarrollo Agiles**

**Quispe (2007).** Las metodologías ágiles, se han comenzado a desarrollar hace muy poco tiempo, dentro de estas encontramos la Metodología Cristal la cual identifica con colores diferentes cada método, y su elección debe ser consecuencia del tamaño y criticidad del proyecto, de forma que los de mayor tamaño, o aquellos en los que la presencia de errores o desbordamiento de

agendas implique consecuencias graves, deben adoptar metodologías más pesadas. De esta forma se pretende obtener mayor rentabilidad en el desarrollo de proyectos de software, Los métodos Crystal no prescriben prácticas concretas, porque están en continuo cambio.

## **2.11. Metodología Crystal Clear (CC)**

**Quispe (2007).** Crystal es una metodología de desarrollo de Software ágil, más que una metodología se la considera una familia de metodologías, debido a que se subdivide en varios tipos de metodologías en función a la cantidad de persona que vayan a estar en el proyecto. Es una metodología que ha sido creada por una persona en particular (Alistair Cockburn ) el cuál llevo la creó en base al análisis de distintos proyectos de desarrollo de SW y su propia experiencia, lo cual fusionando ambos aspectos dio lugar a una metodología bastante interesante, la cual se presenta a continuación.

### **2.11.1. Desarrollo de la Metodología**

#### **2.11.2. Antecedentes**

**Quispe (2007).** En los inicios de 1990, en un estudio realizado en IBM se llegó a los siguientes acuerdos (Cockburn, 2001).

Los equipos exitosos enfatizaban que no habían seguido métodos formales ni herramientas CASE y que habían estimulado la comunicación y los test.

Los equipos con problemas no entendían sus fallas o si habían cumplido con métodos formales.

#### **2.11.3. Qué es Crystal Clear**

**Quispe (2007).** Crystal Clear no es una metodología en si misma sino una familia de metodologías con un “código genético” común.

El nombre Crystal deriva de la caracterización de los proyectos según 2 dimensiones, tamaño y complejidad (como en los minerales, color y dureza).

Por ejemplo.

- Clear es para equipos de hasta 8 personas o menos.
- Amarillo para equipos entre 10 a 20 personas.
- Naranja para equipos entre 20 a 50 persona.
- Roja para equipos entre 50 a 100 personas.
- Azul para equipos entre 100 a 200 personas.

**Quispe (2007).** Crystal Clear puede ser usado en proyectos pequeños y como casi todos los otros métodos, Crystal Clear consiste en valores, técnicas y procesos.

En primera instancia se especifican los antecedentes de la metodología, continuando con definiciones que ayudan a estructurar la fundamentación teórica y se termina con las características esenciales de los diferentes tipos de Crystal.

La familia de metodologías Crystal comparten con la XP una orientación humana, pero esta centralización en Alistair también pone mucho peso en las revisiones al final de la iteración, animando al proceso a aplicar técnicas de mejoramiento continuo en forma automática.

Su aserción es que el desarrollo iterativo esta para encontrar los problemas temprano, y entonces permitir corregirlos. Esto pone más énfasis la gente se hace de una manera diferente. Alistair considera que las personas encuentran difícil seguir un proceso disciplinado, así que más que seguir la alta disciplina de la XP, Alistair explora la metodología menos disciplinada que aun podría tener éxito, intercambiando conscientemente productividad por facilidad de ejecución. El considera que aunque Crystal es menos productivo que la XP, mas personas serán capaces de seguirlo.

#### 2.11.4. Definiciones

**Quispe (2007).** Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos.

El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas.

Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros).

#### 2.11.5. Características

**Quispe (2007).** Las personas, como dispositivos activos, tienen modos de éxito y modos de fallo. Los siguientes son los principales:

- Cuando el número de personas aumenta, también aumenta la necesidad de coordinar.
- Cuando el potencial de daños se incrementa, la tolerancia a variaciones se ve afectada.
- La sensibilidad del tiempo en que se debe estar en el mercado varía: a veces este tiempo debe acortarse al máximo y se toleran defectos, otras se enfatiza la auditoría, contabilidad, protección legal, entre otros.
- Las personas se comunican mejor cara a cara, con la pregunta y la respuesta en el mismo espacio de tiempo.

- El factor más significativo es “comunicación”.

Los métodos se llaman Crystal evocando las facetas de una gema: cada faceta es otra versión del proceso, y todas se sitúan en torno a un núcleo idéntico. Hay cuatro variantes de metodologías: Crystal Clear para equipos de 8 o menos integrantes; Amarillo, para 8 a 20; Naranja, para 20 a 50; Rojo, para 50 a 100.

#### **2.11.6. Crystal Clear (CC).**

**Quispe (2007).** La más exhaustivamente documentada es Crystal Clear (CC), y es la que se ha de describir a continuación:

Crystal Clear puede ser usado en proyectos pequeños. El otro método elaborado en profundidad es el Naranja, apto para proyectos de duración estimada en 2 años. Los otros dos aún se están desarrollando. Como casi todos los otros métodos, Crystal Clear consiste en valores, técnicas y procesos.

Los siete valores o propiedades de Crystal Clear son:

**1) Entrega Frecuente.** Consiste en entregar software a los clientes con frecuencia, no solamente en compilar el código. La frecuencia dependerá del proyecto, pero puede ser diaria, semanal o mensual.

**2) Comunicación Osmótica.** Todos juntos en el mismo cuarto. Una variante especial es disponer en la sala de un experto diseñador senior y discutir respecto del tema que se trate.

**3) Mejora Reflexiva.** Tomarse un pequeño tiempo (unas pocas horas cada o una vez al mes) para pensar bien qué se está haciendo, cotejar notas, reflexionar, discutir.

**4) Seguridad Personal.** Hablar con los compañeros cuando algo molesta dentro del grupo.

**5) Foco.** Saber lo que se está haciendo y tener la tranquilidad y el tiempo para hacerlo.

**6) Fácil.** Acceso a usuarios expertos. Tener alguna comunicación con expertos desarrolladores.

**Quispe (2007).** Crystal Clear no requiere ninguna estrategia o técnica, pero siempre es útil tener unas cuantas a mano para empezar. Las estrategias comunes a otras Metodologías Ágiles, son:

**1) Exploración de 360°.** Verificar o tomar una muestra del valor de negocios del proyecto, los requerimientos, el modelo de dominio, la tecnología, el plan del proyecto y el proceso.

**2) Victoria Temprana.** Es mejor buscar pequeños triunfos iniciales que aspirar a una gran victoria tardía.

**3) Esqueleto Ambulante.** Es una transacción que debe ser simple pero completa.

**4) Re Arquitectura Incremental.** Se ha demostrado que no es conveniente interrumpir el desarrollo para corregir la arquitectura. Más bien la arquitectura debe evolucionar en etapas, manteniendo el sistema en ejecución mientras ella se modifica.

**5) Radiadores de Información.** Es una lámina pegada en algún lugar que el equipo pueda observar mientras trabaja o camina. Tiene que ser comprensible para el observador casual, entendida de un vistazo y renovada periódicamente para que valga la pena visitarla.

**Quispe (2007).** En cuanto a las técnicas de Crystal Clear, se favorecen:

**a) Entrevistas de Proyectos.** Se suele entrevistar a más de un responsable para tener visiones más ricas.

**b) Talleres de Reflexión.** El equipo debe detenerse treinta minutos o una hora para reflexionar sobre sus convenciones de trabajo, discutir inconvenientes y mejoras y planear para el período siguiente.

**c) Planeamiento Blitz.** Una técnica puede ser el Juego de Planeamiento de XP. En este juego, se ponen tarjetas indexadas en una mesa, con una historia de usuario o función visible en cada una. El grupo finge que no hay dependencias entre tarjetas, y las alinea en secuencias de desarrollo preferidas. Los programadores escriben en cada tarjeta el tiempo estimado para desarrollar cada función. El patrocinador del usuario escribe la secuencia de prioridades, teniendo en cuenta los tiempos referidos y el valor de negocio de cada función. Las tarjetas se agrupan en períodos de tres semanas llamados iteraciones que se agrupan en entregas, usualmente no más largas de tres meses.

**d) Estimación Delphi con Estimaciones de Pericia.** En el proceso Delphi se reúnen los expertos responsables y proceden como en un remate para proponer el tamaño del sistema, su tiempo de ejecución, la fecha de las entregas según dependencias técnicas y de negocios y para equilibrar las entregas en paquetes de igual tamaño.

**e) Encuentros Diarios de Pie.** La palabra clave es “brevedad”, cinco a diez minutos como máximo.

**f) Miniatura de Procesos.** Una forma de presentar Crystal Clear puede consumir entre 90 minutos y un día. La idea es que la gente pueda “de gustar” la nueva metodología.

**g) Gráficos de Quemado.** Su nombre viene de los gráficos de quemado de calorías de los regímenes dietéticos; se usan también en Scrum.

Se trata de una técnica de graficación para descubrir demoras y problemas tempranamente en el proceso, evitando que se descubra demasiado tarde que todavía no se sabe cuánto falta. Para ello se hace una estimación del tiempo faltante para programar lo que resta al ritmo actual, lo cual sirve para tener dominio de proyectos en los cuales las prioridades cambian bruscamente y con frecuencia. Esta técnica se asocia con algunos recursos ingeniosos, como la Lista Témpana, llamada así porque se refiere al agregado de ítems con alta prioridad en el tope de las listas de trabajos pendientes, esperando que los demás elementos se hundan bajo la línea deflotación; los elementos que están sobre la línea se entregarán en la iteración siguiente, los que están por debajo en las restantes. En otras Metodologías Ágiles la Lista Témpana no es otra cosa que un gráfico de retraso. Los gráficos de quemado ilustran la velocidad del proceso, analizando la diferencia entre las líneas proyectadas y efectivas de cada entrega.

**h) Programación lado a lado.** Mucha gente siente que la programación en pares de XP involucra una presión excesiva; la versión de Crystal Clear establece proximidad, pero cada quien se enfoca a su trabajo asignado, prestando un ojo a lo que hace su compañero, quien tiene su propia máquina. Esta es una ampliación de la Comunicación Osmótica al contexto de la programación.

### **Roles en la Metodología Crystal Clear**

Hay ocho roles nominados en Crystal Clear que son los siguientes:

**Quispe (2007):** Patrocinador, Usuario Experto, Diseñador Principal, DiseñadorProgramador, Experto en Negocios, Coordinador, Verificador, Escritor.

A continuación se describen los artefactos de los que son responsables los roles de Crystal Clear:

**Patrocinador.** Produce la Declaración de Misión con Prioridades de Compromiso (Tradeoff). Consigue los recursos y define la totalidad del proyecto.

**Usuario Experto.** Junto con el Experto en Negocios produce la Lista de Actores Objetivos y el Archivo de Casos de Uso y Requerimientos. Debe familiarizarse con el uso del sistema, sugerir atajos de teclado, modos de operación, información a visualizar simultáneamente, navegación.

**Diseñador Principal.** Produce la Descripción Arquitectónica. Se supone que debe ser al menos un profesional de Nivel 3. En Metodologías Ágiles se definen tres niveles de experiencia:

Nivel 1 es capaz de “seguir los procedimientos”.

Nivel 2 es capaz de “apartarse de los procedimientos específicos” y encontrar otros distintos

Nivel 3 es capaz de manejar con fluidez, mezclar e inventar procedimientos.

El Diseñador Principal tiene roles de coordinador, arquitecto, mentor y programador más experto.

**Diseñador Programador.** Produce, junto con el Diseñador Principal, los Borradores de Pantallas, el Modelo Común de Dominio, las Notas y Diagramas de Diseño, el Código Fuente, el Código de Migración, las Pruebas y el Sistema Empaquetado. Un programa en Crystal Clear es “diseño y programa”; sus programadores son diseñadores programadores. En Crystal Clear un diseñador que no programe no tiene cabida.

**Experto en Negocios.** Junto con el Usuario Experto produce la Lista de Actores Objetivos y el Archivo de Casos de Uso y Requerimientos. Debe conocer las reglas y políticas del negocio.

**Coordinador.** Con la ayuda del equipo, produce el Mapa de Proyecto, el Plan de Entrega, el Estado del Proyecto, la Lista de Riesgos, el Plan y Estado de Iteración y la Agenda de Visualización.

**Verificador.** Produce el Reporte de Bugs. Puede ser un programador en tiempo parcial, o un equipo de varias personas.

**Escritor.** Produce el Manual de Usuario. El Equipo como Grupo es responsable de producir la Estructura y Convenciones del Equipo y los Resultados del Taller de Reflexión.

**El Código Genético:** Este modelo ve al desarrollo de software como una serie de partidos que consisten en inventar y comunicar. Cada partido es diferente y tiene como objetivo entregar software y prepararse para el siguiente juego. Esto permite al equipo trabajar concentrado y en forma efectiva con un objetivo claro cada vez.

**Propiedades** Frecuencia en las Entregas: entregar al usuario funcionalidad "usable" con una frecuencia de entre 2 semanas y no más de un mes.

## **Principios**

**Quispe (2007).** En esta metodología se detallan principios que son los siguientes:

- El grado de detalle necesario en documentar requerimientos, diseño, planeamiento, etc., varía según el proyecto.
- Es imposible eliminar toda documentación pero puede ser reducida logrando un modo de comunicación más accesible, informal y preciso que pueda ser accedido por todos los miembros del equipo.

- El equipo ajusta constantemente su forma de trabajo para lograr que cada personalidad encaje con los otros miembros, con el entorno y las particularidades de cada asignación.

## **Estrategias**

**Quispe (2007).** – A continuación una breve explicación de las estrategias en esta metodología:

Ni las estrategias ni las técnicas son mandatorias para Crystal Clear. Pero es bueno tener en cuenta alguna de ellas al momento de empezar a trabajar.

Tres de las estrategias que están más relacionadas son las de apuntar a tener "Victorias Tempranas", arrancar el desarrollo de lo que se denomina un "Esqueleto que Camine" y pensar siempre en hacer "Re arquitectura Incremental" van de la mano.

### **2.12. Empresas Pecuarias**

La empresa ganadera es una unidad económica de producción que combina factores y servicios productivos (pienso, mano de obra, etc.) a fin de producir bienes (leche, carne, huevos, etc.) que destina al mercado; es por lo tanto, una unidad de control y de toma de decisiones de la producción animal.

El empresario ganadero es un ente lógico y racional que debe saber tomar sus decisiones, ya sea de modo propio o a través de asesoramiento externo, a fin de efectuar una óptima política empresarial.

La Producción Animal analiza el conjunto de sistemas de explotación y técnicas de producción; es decir, la relación existente entre el producto y una serie de factores, en tanto que la "Economía Pecuaria" estudia la asignación más conveniente de recursos escasos para la obtención más conveniente de un

conjunto ordenado de objetivos", apoyándose en las técnicas de gestión pecuaria.

En el ámbito general de la "Producción Animal" y el particular de la "Economía Ganadera", se entiende por **Gestión**:

"El conjunto de conocimientos y técnicas, de naturaleza productiva y económica, que permiten la utilización y aplicación óptima, en el corto y largo plazo de los insumos disponibles, ya sean endógenos o exógenos a la explotación".

### **2.13. Atención al Cliente**

**Gaitner (1983).** El tener un enfoque bien definido hacia el cliente y una orientación hacia el servicio constituye un reto para toda organización, el servicio que se ofrece a los clientes es importante siempre y actualmente constituye un elemento básico para el éxito o fracaso de muchas empresas, ya que una relación cordial entre el proveedor y el cliente, y sobre todo una relación comprensiva ante cualquier tipo de reclamo conforman una ventaja competitiva.

También puede ser considerada como las actividades secundarias que realiza una empresa para optimizar la satisfacción que reciba el cliente de sus actividades principales.

Para ello debemos preguntar que servicios se ofrecerán y como lo vamos a hacer, la primera se refiere en determinar cuáles son los servicios que el cliente exige o demanda mediante el cuestionario, observación, grupos focales.

La segunda pregunta ¿nivel de servicios que vamos a ofrecer? Se refiere al hecho de que una vez realizada las encuestas se debe analizar la cantidad y calidad del servicio que requiere el consumidor estableciendo una línea al

consumidor 1-800, buzones de sugerencias esto ayudará a conocer las fallas de nuestros servicios, muchas de las veces se dice que una queja de un cliente es un buen regalo si sabemos manejarlos y corregirlos.

La última pregunta que se refiera a cómo lo vamos a ofrecer, qué actitudes van a tomar nuestros colaboradores, precio, si se le va a ofrecer un servicio gratuito por un año o determinado tiempo por la compra de un determinado producto, un elemento clave del servicio al cliente se encuentra en la capacitación humana de todo el personal con énfasis a quienes se involucren directamente con el cliente pues de ellos depende que el cliente regrese o no.

Tanto el cliente interno (colaboradores) como externo deben sentirse cómodos en la organización en el caso de que un cliente interno sea tratado mal como podemos esperar que ellos traten bien a nuestros clientes externos.

Como lo hemos señalado anteriormente el servicio enfocado al cliente debe:

- Centrar todos sus procesos al cliente.
- Orientar toda la gestión al concepto de creación de valor para los clientes (los clientes, siempre, han comprado valor y sólo valor).
- Incorporar una cultura de servicio adquiriendo así la empresa una mejora constante en sus servicios.
- Utilizar el servicio como distintivo de la empresa convirtiéndose en un elemento diferenciador acompañado de una fuerte ventaja competitiva.
- Desarrollar nuevas medidas cuyo objetivo será el de establecer nuevos criterios y sistemas para la medición del desempeño de la empresa, sus

diferentes áreas y personas, permitiendo así relacionar directamente la satisfacción de los clientes con los objetivos financieros y las mediciones operativas.

- Manejar efectivamente las tecnologías.- Aprovechando al máximo las Tecnologías que se nos presente con el objeto de mejorar las relaciones con nuestros clientes, entre ellos diseños de páginas web, correo electrónico, CRM (Customer Relationship Management, se entiende como la Gestión de Relación con clientes).

El servicio al cliente radica en la filosofía, las actitudes y los comportamientos de cada uno de los empleados que van desde el portero hasta el presidente.

### **Importancia Atención al Cliente**

La atención al cliente puede convertirse en un elemento principal para el éxito o el fracaso de muchas empresas, las empresas que están orientadas hacia el cliente son las que tienen más probabilidad de éxito en el futuro.

Atraer un nuevo cliente es aproximadamente seis veces más caro que mantener uno, por lo que las empresas han optado por dar mayor énfasis en cuanto al servicio al cliente. Se ha observado que los clientes son sensibles al servicio que reciben, algunas personas dejan de comprar un producto o servicio por las fallas en la información que se les proporciona por parte del personal encargado de atender a los compradores, ante esta realidad, se hace necesario que la atención al cliente sea de la mas alta calidad, así como también la información que se brinda, y el trato humano y técnico con el que va a establecer en una relación comercial.

### III. MATERIALES Y MÉTODOS

#### 3.1. Localización y Duración de la Investigación

El presente trabajo se realizó en la empresa Servicios Pecuarios “Véliz” la misma que se dedica a la atención de mascotas y tiene como gerente al Dr. Veterinario Abraham Véliz Pincay, ubicado en Malecón Eloy Alfaro, entre la 19 y 20 ava, Cdla. San José, Quevedo, Provincia de Los Ríos.

La duración de la investigación fué de 180 días.

#### 3.2. Materiales y Equipos

Materiales, Equipos, Software y Hardware que se utilizaron en la investigación:

<b>Detalle</b>	<b>Cantidad</b>
<b>Materiales</b>	
Resmas de papel	4
Lápiz	2
Cuadernos	2
<b>Software</b>	
Sistema Operativo Windows 7	1
Microsoft Office 2007	1
Audiovox Pocket Pc	1
Microsoft Visual Estudio. Net	1
<b>Hardware</b>	
Computadora Core 2 Duo	1
Cámara Fotográfica Pentax 1 GB	1
Impresora Lexmark X5270	1
Memory Flash 2 Gb	1
Pocket Pc	1

### **3.3. Tipo de Investigación**

Los tipos de investigación utilizados fueron Exploratoria y Descriptiva.

#### **3.3.1. Exploratoria**

El objetivo de la investigación exploratoria fue recopilar la información preliminar que ayudó a definir problemas y a sugerir hipótesis.

Este tipo de investigación permitió obtener información de la empresa Servicios Pecuarios Véliz para el desarrollo e investigación del proyecto.

#### **3.3.2. Descriptiva**

Comprende la descripción, registros, análisis e interpretación de la naturaleza actual y la composición o procesos de los fenómenos. El enfoque se hace sobre conclusiones dominantes o sobre cómo una persona, grupo o cosa se conduce o funciona en el presente.

La investigación descriptiva en esta tesis estudió los hechos en el área veterinaria.

### **3.4. Métodos**

#### **3.4.1. Analítico**

En este trabajo se aplicó el método analítico que consiste en la extracción de las partes de un todo, con el objeto de estudiarlas y examinarlas por separado. Por ejemplo las relaciones entre las mismas, en este caso se tomó en cuenta los criterios de todos los clientes y proveedores de la empresa Servicios Pecuarios “Véliz”.

### **3.4.2. Sintético**

Es un proceso mediante el cual se relacionaron hechos aparentemente aislados y se formula una teoría que unifica los diversos elementos, que nos ayudó en el desarrollo del sistema para llevar un mejor registro de los clientes y mascotas en la empresa.

## **3.5. Fuentes**

### **3.5.1. Fuentes Primarias**

Se obtuvo la información proporcionada por el Gerente propietario de la Empresa Servicios Pecuarios “Véliz”.

### **3.5.2. Fuentes Secundarias**

Libros, módulos, catálogos, folletos e internet, sirvieron como fuente de apoyo para la elaboración del software de la empresa.

## **3.6. Técnicas e Instrucciones de Evaluaciones**

Las técnicas e instrumentos utilizados para la elaboración del Sistema de Información, fueron la encuesta y entrevista.

### **3.6.1. Entrevista**

La entrevista se aplicó al Gerente Propietario Dr. Veterinario Abraham Véliz Pincay, para recopilar información para el sistema de información que se desarrolló en la empresa.

(Ver anexo 1).

### 3.6.2. Encuesta

En este caso se tomaron en cuenta los criterios de los clientes y proveedores que acuden a la empresa Servicios Pecuarios Veliz, quienes proporcionaron la información requerida. (Ver anexo 2).

## 3.7. Población y Muestra

### 3.7.1. Población

Está conformada por la persona que labora y trabaja en la empresa Servicios Pecuarios Veliz, que es el Dr. Veterinario Abraham Véliz Pincay, cuenta con 10 proveedores, personas que se dedican a la distribución de productos e insumos para el consumo agro-veterinario. Esta empresa tiene 200 personas que tienen sus mascotas en tratamientos y control sanitario.

### 3.7.2. Muestra

La muestra a calcular en los clientes es:

$$n = \frac{Z^2 \cdot p \cdot q \cdot N}{E^2(N-1) + p \cdot q \cdot Z^2}$$

$$n = \frac{(1,96)^2(0,50)(0,50)(200)}{(0,05)^2(200-1) + (0,50)(0,50)(1,96)^2}$$

$$n = \frac{192,08}{0,4975 + 0,9604}$$

$$n = \frac{192,08}{1,4579}$$

$$n = 132 \text{ clientes}$$

n= tamaño de muestra

N= población a investigar

Z= nivel de confianza =1,96

p= variabilidad positiva =50 %

q= variabilidad negativa =50%

E= precisión o error=0,05

### 3.8. Procedimiento Metodológico

Para el desarrollo del sistema propuesto en esta tesis se utilizó la metodología de desarrollo de software orientado a objetos, la metodología ágil Crystal Clear, la cual se centra en tener una arquitectura ejecutable a medida que se va diseñando y construyendo iterativamente un sistema.

**Metodología Crystal.** Es un conjunto de diferentes metodologías que podemos seleccionar en función de la adecuación al proyecto que se va a desarrollar en este caso en la empresa Servicios Pecuarios Véliz, se implantó un sistema de información con aplicación para el control administrativo en el cual se lleva un control de los clientes y mascotas, consultas, egresos e ingresos de medicamentos y reporte de consultas.

**Planificación por Etapas.** Básicamente consiste en la planificación del siguiente incremento del sistema. La misma que finalizó con una versión ejecutable en la empresa a modo de pruebas para que el gerente se sienta satisfecho con el sistema. Se seleccionaron los requisitos implementados en el incremento y se planificaron cada una de las etapas del sistema.

**Relevamiento de Información.** Entrevista y diálogo con el gerente para conocer el funcionamiento actual y cuáles son las falencias de la empresa y dejar en claro las especificaciones que debía llevar en el desarrollo del sistema.

**Análisis del Problema.** Una vez realizado el relevamiento de información, se analizaron los datos para empezar con el diseño y desarrollo del sistema.

**Diseñar el Software.** Una vez definido el análisis se procedió a diseñar la base de datos utilizando SQL Server 2008 como sistema administrador de bases de datos, diagramas de caso de uso y diagramas de secuencia.

**Realizar la Experimentación.** Una vez desarrolla la fase del diseño se procedió a realizar la experimentación que tiene como objeto, cumplir con dos propósitos principales:

El sistema sea capaz de explicar el comportamiento en razón de sus políticas y estructuras orientadas.

El sistema se convierta en un instrumento que permita estudiar el cambio de las estructuras y políticas de la organización-

Por lo tanto se debe pasar por tres subprocesos:

**Verificación.** La verificación se la realizó en base a la estabilidad y a las respuestas del sistema.

El sistema se sometió a un gran número de pruebas en base a diferentes parámetros dando resultados esperados en función a las propuestas. Manteniendo al estabilidad aún siendo simulado en eventos muy críticos.

**Análisis.** Se procedió a realizar la retroalimentación basada en los resultados de los eventos, por lo que luego de varias pruebas se procedió a dar por culminado el sistema.

**Validación.** Se realizó la validación de todos los datos ingresados y se verificó que todo funcione a la perfección y que no existan errores, en caso de haberlos se verificó y se corrigió para que se ejecute con normalidad y se obtenga los resultados esperados.

Crystal Clear es una metodología ágil, permitió reducir el riesgo de una mala construcción. Crystal Clear fue seleccionado porque hace hincapié en:

- Eficiencia.** El objetivo de utilizar Crystal Clear fué minimizar los residuos, centrándose al mismo tiempo en las características importantes del equipo, es decir la conexión desde el servidor con la Pocket PC para que los datos sean importados y exportados de manera correcta.

- Habitabilidad.** Crystal Clear está diseñado para ser cómodo de usar, de modo que los equipos sean utilizados de manera correcta.

- Seguridad.** Se decidió desarrollar este proyecto con Crystal Clear, porque es un metodología ágil, fácil de utilizar y se adhirió correctamente a SQL server y visual studio.net, herramientas que se resolvió utilizar para el desarrollo de la aplicación móvil.

Esta metodología cuenta con ocho roles que a continuación se detallan:

- Patrocinador.** Se analizaron las necesidades y prioridades para realizar el sistema en la empresa Servicios Pecuarios Véliz con el compromiso de automatizarla y beneficiando al gerente y para satisfacción de los clientes, consiguiendo los recursos adecuados y así definir la totalidad del proyecto.

- Usuario Experto.** Se elaboraron la lista de objetivos que se necesitaban, diseñando los casos de uso y requerimientos del sistema de información con aplicación móvil para el control administrativo de la empresa Servicios Pecuarios Véliz.

- Diseñador Principal.** Se analizaron los lenguajes de programación y se decidió por las ventajas presentadas utilizar la base de datos SQL, Visual

Studio 2008 estos deben ser compatibles con el dispositivo móvil Pocket PC que fue la aplicación que se va a utilizar en el desarrollo del sistema.

**Diseñador Programador.** Este se utilizó junto con el diseñador principal, para diseñar los Borradores de Pantallas, el Modelo Común de Dominio, las Notas y Diagramas de Diseño, el Código Fuente, el Código de Migración, las Pruebas y el Sistema Empaquetado.

**Experto en Negocios.** Junto con el usuario experto produce la lista de Actores Objetivos y el Archivo de Casos de Uso y Requerimientos .para que el gerente realice las pruebas correspondientes para el buen uso del sistema aplicando el dispositivo móvil, debiendo conocer los respectivos pasos para la conexión como es servidor y Pocket PC, utilizando los implementos necesarios incluido el router.

**Verificador.** Se utilizó para el desarrollo del reporte de pruebas a utilizar para que el sistema funcione bien y no tenga ningún error en el momento de ejecutar desde el servidor o del dispositivo móvil Pocket PC, ingresando los datos de varios clientes.

**Escritor.** Se empleó para elaborar los manuales:

- Manual desde el Servidor PC,
- Conexión utilizando el router, el dispositivo móvil (Pocket Pc)
- Manual de Usuario desde el Pocket Pc.

## **IV. RESULTADOS**

### **4.1. Situación actual de la Empresa**

Se desarrolló la investigación en una empresa dedicada a la atención de clientes con sus mascotas en el cantón Quevedo y sus alrededores, cuenta con una infraestructura amplia como es el consultorio e instalaciones adecuadas para la visita de los clientes que llegan a diario hacer tratar a sus mascotas.

Al realizar un análisis y estudio de la empresa se observó que cuenta con dos áreas: consultorio y administrativa, en el área del consultorio es donde recibe a los clientes con sus mascotas para el tratamiento según la historia clínica que este tenga, quiere decir aplicación de vacunas, refuerzos, desparasitantes, anabólicos, vitaminas, balanceado pro can.

En el área administrativa se notó deficiencia por cuanto no cuenta con disponibilidad de información oportuna, es decir, la persona encargada que es el gerente propietario debe buscar la carpeta de historia clínica de la mascota de manera manual, causando malestar a los clientes por la pérdida de tiempo.

Los medicamentos que utiliza para el tratamiento de las mascotas no lleva un control de ingreso y egreso, por lo que se le dificulta al momento de hacer una compra o venta, y tiene que acudir a la compra de los mismos.

El veterinario realiza visitas domiciliarias a sus clientes esto implica que tiene que llevar la carpeta de las mascotas que va a atender para poder anotar que productos se le aplica y vende en el momento de visita.

## 4.2. Resultados de las Encuestas

### 4.2.1. Encuesta a los Clientes

Se realizaron 132 encuestas a los clientes de la Empresa Servicios Pecuarios Véliz en el año 2011 en la ciudad de Quevedo, Provincia de Los Ríos.

#### 4.1.1.1. Inconvenientes en la empresa por la atención

Los resultados del cuadro # 1 indican que el 72% de los clientes encuestados califica la atención de la empresa como deficiente, los procesos administrativos son manuales y carece de un control secuencial de la atención, tratamientos de las mascotas que atiende, el 28% opinaron que están conformes con el proceso que esta brinda.

En la encuesta realizada se comprobó la necesidad que la empresa mejore su calidad en los procesos administrativos que manejan, considerando un factor importante para la automatización de la empresa.

#### Cuadro 1. Inconveniente por la atención.

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	95	72%
NO	37	28%
<b>TOTAL</b>	<b>132</b>	<b>100%</b>

Fuente: Clientes Empresa Servicios Pecuarios Véliz  
Elaborado por: La Autora

#### 4.1.1.2. Molestias por la atención manual en la empresa

Los resultados en el cuadro # 2 reflejan que el 68% de los clientes manifiestan que el servicio que brinda la empresa les genera pérdida de tiempo, ya que la empresa carece de un archivo ordenado de los clientes que acuden a requerir el servicio, y el 32% dicen estar conforme con el servicio.

En los actuales momentos y en un mundo globalizado toda empresa debe utilizar un sistema informático que genere información ordenada para el buen funcionamiento de la misma y crear un ambiente de satisfacción en los clientes.

**Cuadro 2. Malestar en los clientes sobre la atención manual.**

<b>ALTERNATIVAS</b>	<b>FRECUENCIA</b>	<b>PORCENTAJE</b>
SI	90	68%
NO	42	32%
<b>TOTAL</b>	<b>132</b>	<b>100%</b>

Fuente: Clientes Empresa Servicios Pecuarios Véliz  
Elaborado por: La Autora

**4.1.1.3. Control de tratamiento de su mascota**

Los resultados del cuadro # 3 indican que el 95% de los clientes encuestados manifiestan que no se les proporciona el control de tratamiento de su mascota, mientras que el 5% se sienten conformes que se lleve el control de tratamiento.

De las personas encuestadas un alto porcentaje desconocen los procesos de control y tratamiento de sus mascotas, es decir el botiquín veterinario no cuenta con un sistema que le permita llevar las historias clínicas de cada mascota atendida.

**Cuadro 3. Facilidad para el control de tratamiento a la mascota.**

<b>ALTERNATIVAS</b>	<b>FRECUENCIA</b>	<b>PORCENTAJE</b>
SI	7	5%
NO	125	95%
<b>TOTAL</b>	<b>132</b>	<b>100%</b>

Fuente: Clientes Empresa Servicios Pecuarios Véliz  
Elaborado por: La Autora

#### 4.1.1.4. Mejoras en la empresa

Los resultados del cuadro # 4 indican que el 67% de los usuarios encuestados no se hallan conformes con la prescripción médica realizada por el veterinario ya que no es actualizada, y para diferentes síntomas es la misma prescripción y el 11% consideran que se sienten conforme con el servicio.

Los clientes consideran que el gerente de la empresa debería automatizar la base de datos de medicamentos para las recetas de los controles médicos que provee de la mascota estén acorde al tratamiento.

**Cuadro 4. Mejoras para la empresa.**

<b>ALTERNATIVAS</b>	<b>FRECUENCIA</b>	<b>PORCENTAJE</b>
COSTOS DE		
MEDICAMENTOS	67	51%
ATENCION AL CLIENTE	15	11%
SERVICIOS	50	38%
<b>TOTAL</b>	<b>132</b>	<b>100%</b>

Fuente: Clientes Empresa Servicios Pecuarios Véliz  
Elaborado por: La Autora

#### 4.1.1.5. Implementar un sistema para una mejor atención

Los resultados del cuadro # 5 el 91% de los clientes indican que se debería implementar un sistema para automatizar los registros de datos que genera el control de las mascotas, y el 9% consideran que se sienten conformes con el proceso manual que brinda la empresa.

Esta respuesta, identifica la necesidad de implantar un sistema informático administrativo para agilizar los diferentes procesos que realiza y brindar un servicio de calidad a los clientes mejorando la imagen de la empresa y la generación de nuevos ingresos.

### **Cuadro 5. Implementación de un Sistema en la empresa.**

<b>ALTERNATIVAS</b>	<b>FRECUENCIA</b>	<b>PORCENTAJE</b>
SI	120	91%
NO	12	9%
<b>TOTAL</b>	<b>132</b>	<b>100%</b>

Fuente: Clientes Empresa Servicios Pecuarios Véliz  
Elaborado por: La Autora

#### **4.2.2. Encuesta a los Proveedores**

Se realizaron encuestas a los 10 proveedores de la Empresa Servicios Pecuarios Véliz en el año 2011 en la ciudad de Quevedo, Provincia de Los Ríos.

##### **4.1.2.1 Productos que adquiere la empresa a los proveedores**

En referencia al cuadro # 6 el 90% de los proveedores indicaron que la empresa a la que proveen de productos no cuenta con un registro de compras ocasionando duplicidad en los pedidos que realiza mensualmente y el 10% manifiesta no tener inconvenientes con pagos y pedidos.

El resultado indica que la empresa no lleva un registro de los productos que adquiere es decir no existe un control de ingresos y egresos económico, ni medicamentos.

### **Cuadro 6. Control de productos que adquiere la empresa a los proveedores.**

<b>ALTERNATIVAS</b>	<b>FRECUENCIA</b>	<b>PORCENTAJE</b>
SI	9	90%
NO	1	10%
<b>TOTAL</b>	<b>10</b>	<b>100%</b>

Fuente: Clientes Empresa Servicios Pecuarios Véliz  
Elaborado por: La Autora

#### 4.1.2.2 Formas de pagos a proveedores

Según el resultado del cuadro # 7 el 80% de los proveedores indicaron que la forma de pago es al contado, pero no lo realiza a tiempo por falta de control y registro de las facturas de los productos adquiridos y el 20% manifiestan es a crédito y no tienen inconvenientes para cobrar dichas facturas.

**Cuadro 7. Inconvenientes con la forma de pago a los proveedores**

<b>ALTERNATIVAS</b>	<b>FRECUENCIA</b>	<b>PORCENTAJE</b>
CONTADO	8	80%
CREDITO	2	20%
<b>TOTAL</b>	<b>10</b>	<b>100%</b>

Fuente: Clientes Empresa Servicios Pecuarios Véliz  
Elaborado por: La Autora

#### 4.1.2.3 La atención de la empresa

El resultado del cuadro # 8 el 60% los proveedores manifiestan que la empresa no brinda una buena atención en la solicitud de productos por la falta de un sistema que contribuya a la actualización del stock existente de medicamentos, y el 10% expresan que no tienen inconvenientes con el servicio que ofrece.

**Cuadro 8. Cómo califica la atención en esta empresa.**

<b>ALTERNATIVAS</b>	<b>FRECUENCIA</b>	<b>PORCENTAJE</b>
MUY BUENA	1	10%
BUENA	3	30%
REGULAR	6	60%
<b>TOTAL</b>	<b>10</b>	<b>100%</b>

Fuente: Clientes Empresa Servicios Pecuarios Véliz  
Elaborado por: La Autora

#### 4.1.2.4 Tiempo que la empresa Servicios Pecuarios Véliz realiza sus pedidos

El resultado del cuadro # 09 indica que el 60% de los encuestados realiza los pedidos de forma semanal de manera indiscriminada por que no tiene un control de los medicamentos que solicita, y el 20% manifiestan que los pedidos los realiza a diario.

**Cuadro 9. Tiempo que la empresa realiza sus pedidos.**

<b>ALTERNATIVAS</b>	<b>FRECUENCIA</b>	<b>PORCENTAJE</b>
MENSUAL	3	30%
SEMANAL	6	60%
DIARIO	2	20%
<b>TOTAL</b>	<b>10</b>	<b>100%</b>

Fuente: Clientes Empresa Servicios Pecuarios Véliz  
Elaborado por: La Autora

#### 4.1.2.5 Empresa automatizada con un Sistema Informático.

En referencia al cuadro # 10 el 80% los encuestados respondieron que están de acuerdo que la empresa se automatice esto ayudaría a mejorar los procedimientos que actualmente utiliza la empresa para la atención a sus usuarios, y el 20% opinan que están conformes con el proceso actual.

**Cuadro 10. Automatización de la empresa con un Sistema Informático.**

<b>ALTERNATIVAS</b>	<b>FRECUENCIA</b>	<b>PORCENTAJE</b>
SI	8	80%
NO	2	20%
<b>TOTAL</b>	<b>10</b>	<b>100%</b>

Fuente: Clientes Empresa Servicios Pecuarios Véliz  
Elaborado por: La Autora

### **4.2.3. Resultado de la Entrevista**

#### **4.1.3.1 Entrevista al Gerente Propietario**

Se realizó la entrevista al gerente propietario de la empresa Servicios Pecuarios Véliz Dr. Abraham Véliz Pincay con el objetivo de conocer la situación actual de la empresa.

##### **1. ¿Qué tipo de animales atiende usted?**

Dentro de esta empresa se da servicio médico veterinario a usuarios que poseen mascotas por ejemplo a caninos, mientras que en un mínimo porcentaje a los gatos, cerdos, y vacunos.

##### **2. ¿Qué datos del cliente registra en su archivo?**

Entre los datos que se registran esta la historia clínica de la mascota con los stickers de cada vacuna, que ha recibido para controlar la revacunación, prescripción medica de vitaminas, desparasitaros y datos personales del cliente como: nombre, dirección y teléfono.

##### **3. ¿Cómo realiza el registro de los clientes que acuden a la empresa con sus mascotas?**

El registro se lo realiza en carpetas (historia clínica) que es el control sanitario de las mascotas, los tratamientos son de 3 a 5 días sin ingreso o mas días dependiendo el cuadro sintomatológico del animal.

##### **4. ¿Lleva un registro de los productos y medicamentos que aplica a las mascotas?**

El registro es con las facturas de compra que se realiza en la Agropecuaria "Don Nico", "Agro-Fuerte" y "Súper Tienda Quevedo", que de acuerdo a lo que

se vende si es diario, semanal o mensual, a medida que se egresa los productos se realiza la compra nuevamente en un tiempo determinado.

#### **5. ¿Cuáles son los productos utilizados en el tratamiento de la mascota?**

Vacunas para la prevención de las enfermedades, vitaminas y antiofídicos de acuerdo al cuadro sintomatológico que presente el animal.

Vacunas para caninos: Parvo virosis, Hepatitis, Moquillo, Antirrábica.

Desparasitantes: Ivomec, Ibermectina, Bolfo, Canisul.

Vitaminas: Vitamina K, Complejo B, Tabletas Calcio Gluconato con vitamina D, Complejo B (gotas).

#### **6. ¿Cuáles son sus proveedores de su empresa?**

Las casas comerciales que me proveen de los productos para el tratamiento de las mascotas en mi empresa son: Agropecuaria “Don Nico” ubicado en la parroquia San Camilo, Agro Fuerte y Súper Tienda Quevedo en el cantón Quevedo.

#### **4.2.4. Demostración de Hipótesis**

En la actualidad la empresa realiza de manera manual la historia clínica de los datos de las mascotas de cada cliente, esto provoca malestar a los clientes ya que lleva tiempo en la búsqueda para saber que tipo de tratamiento o control lleva la mascota, no lleva un control de ingreso y egreso de mercadería que tiene en vitrina.

Con el resultado de las encuestas que se efectuó en un tiempo determinado a los clientes, proveedores y veterinario de la empresa Servicios Pecuarios Veliz

en la ciudad de Quevedo, y haciendo el análisis respectivo se llegó a la conclusión, que el sistema que la implementación del sistema para la automatización mejorará considerablemente ya que se realizó las pruebas respectivas y en los actuales momentos esta en funcionamiento y satisface las diferentes necesidades de los clientes que visitan la empresa.

Este software permitió la modernización de la empresa, el gerente propietario brindará un óptimo servicio a sus clientes y la eficiencia de este le dará una mayor efectividad en la utilización del mismo ya que cuenta con una Aplicación Móvil que podrá manipular desde el dispositivo la información que necesita y realizar la actualización de datos diariamente en el sistema.

## V. DISCUSIÓN

En base al trabajo de campo realizado en la empresa Servicios Pecuarios Véliz se comprueba que es necesaria que la implementación de un sistema de información con aplicación móvil, lo cual permitirá registrar clientes y mascotas, medicamentos, consultas de manera adecuada y organizada en función a sus actividades y desarrollo laboral, tal como lo manifiesta **Fernández (2006)** un Sistema de Información (SI) es el conjunto formal de procesos de análisis, que operando sobre una colección de datos estructurados de acuerdo a las necesidades de la empresa, recopila, elabora y distribuye, la información necesaria para realizar las operaciones básicas y la toma de decisión en la empresa, que sirva para desempeñar las funciones de negocio de la empresa de acuerdo con sus estrategias.

La entrevista efectuada demuestra las falencias del área administrativa al no proyectar un enfoque real en los procesos de trabajo, es así que un sistema de información con aplicación móvil permitirá llevar un mejor control en ingresos y egresos de datos, de productos que adquiere, de acuerdo a lo que expresa **Álava (2010)** la administración es un proceso particular consistente en las actividades de planeación, organización, ejecución y control, desempeñadas para determinar y alcanzar los objetivos señalados.

Al modernizar la empresa con un sistema de información en el cual se le agrega una aplicación móvil ésta permitirá llevar un control más ágil y personalizado que ayudará al gerente propietario en sus actividades diarias de visita a sus clientes y llevar la información en el dispositivo móvil, (Pocket Pc) realizar las consultas planificadas y al llegar a su domicilio lo que hará es la exportación de la información al servidor mediante la conexión de una red inalámbrica utilizando un router y así se le haría fácil llevar un control más rápido, como lo señala **Alegsa (2007)** la aplicación móvil permite el acceso desde un dispositivo móvil a toda la información ya sea corporativa, o de uso personal que quiera movilizar, tal como agendas de clientes, catálogos de productos y precios. Como otros ordenadores, los Pocket PC o PPC son

ordenadores de carácter general, no obstante se han diseñado aplicaciones especialmente adaptadas a sus características, como por ejemplo lectores de códigos de barras o receptores GPS para el control de flotas.

La programación de aplicaciones puede realizarse mediante varios lenguajes, y generalmente desde un computador de escritorio, entre los cuales C Sharp, Visual Basic y Léxico son buenos ejemplos.

## **VI. PROPUESTA**

### **6.1. Sistema de Información con Aplicación Móvil para el Control Administrativo de la Empresa Servicios Pecuarios Veliz**

#### **6.2. Justificación**

La ausencia de un sistema de información imposibilita cuestionar los resultados conseguidos en cuanto a ventas, abastecimientos, y utilidades, por eso se hace necesaria y urgente la implementación de este sistema, ya que permite la planeación, organización, coordinación, dirección y control de todas y cada una de las actividades desarrolladas en esta empresa.

Con el objeto de facilitar el trabajo en el área administrativa se justifica un sistema de información con aplicación móvil para mejorar y transformar el trabajo manual al trabajo automático encaminado al desarrollo y automatización de la empresa.

Para el desarrollo de la aplicación se utilizó una metodología orientada a objetos como es Crystal Clear la cual permite poder reconocer de la aplicación los objetos a diseñar, de estos objetos sus propiedades, su valor y sus operaciones a realizar, para poder diseñar y desarrollar las distintas tareas a desempeñar por la aplicación, esta metodología reduce la complejidad de los problemas ya que divide la aplicación en problemas pequeños y permite obtener soluciones con mejor rendimiento.

La Base de datos es el conjunto de informaciones almacenadas en un soporte legible por el ordenador y organizadas internamente por registro y campos. SQL Server se utiliza para manejar grandes volúmenes de informaciones.

Visual Studio 2008, permitió la creación de ambientes gráficos de manera fácil y sencilla, permitiendo el uso de herramientas propias de la aplicación, como es

la adaptación de la Pocket PC. Es un lenguaje de programación que permite el desarrollo de las aplicaciones para internet, para móviles y aplicaciones de propósito general.

Una aplicación, es un tipo de programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Los dispositivos móviles son aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente a una red, con memoria limitada, diseñados específicamente para una función pero que pueden llevar a cabo otras funciones más generales.

Para tener conexión y sincronización con el dispositivo Pocket PC se requiere la instalación en la PC del software de sincronización de archivos active Sync que se encuentra en el disco drivers de la Pocket PC o en la página oficial de Microsoft.

Active Sync de Microsoft, es el centro de control para el intercambio de datos entre el PC y Windows Mobile / Pocket PC. Apoya la sincronización de Contactos, Calendario, Correo Electrónico, tareas y notas con MS Outlook.

En caso de no contar con un dispositivo conectado a la PC para realizar las pruebas correspondientes, se puede trabajar con el emulador incluido en la plataforma de desarrollo Visual Studio. Los emuladores son máquinas virtuales que corren en la PC como si fuera el dispositivo mismo.

Existen diversos emuladores que dependen de la plataforma de desarrollo y el tipo de framework que se utiliza al programar (Windows CE 2002, Windows Mobile 2003, Smartphone, etc.).

### 6.3. Documentación del Sistema

La metodología Crystal Clear y la diversidad de alternativas propuestas por el lenguaje de Modelado Unificado (UML) se considera la elección metodológica para desarrollar este trabajo de investigación: **Sistema de Información con Aplicación Móvil para el Control Administrativo de la empresa “Servicios Pecuarios Véliz”**.

#### 6.3.1. Documentos de Requerimientos

REQUERIMIENTOS ID	DESCRIPCION	CASO DE USO
RQ01	Inicio de sesión	CU01
RQ02	Cambiar clave	CU02
RQ03	Registrar Clientes	CU03
RQ04	Registrar Mascota	CU04
RQ05	Registrar Medicamento	CU05
RQ06	Registrar Usuario	CU06
RQ07	Registrar Control de Mascota	CU07

## 6.2.2. Caso de Uso

### 6.2.2.1. Caso de Uso General

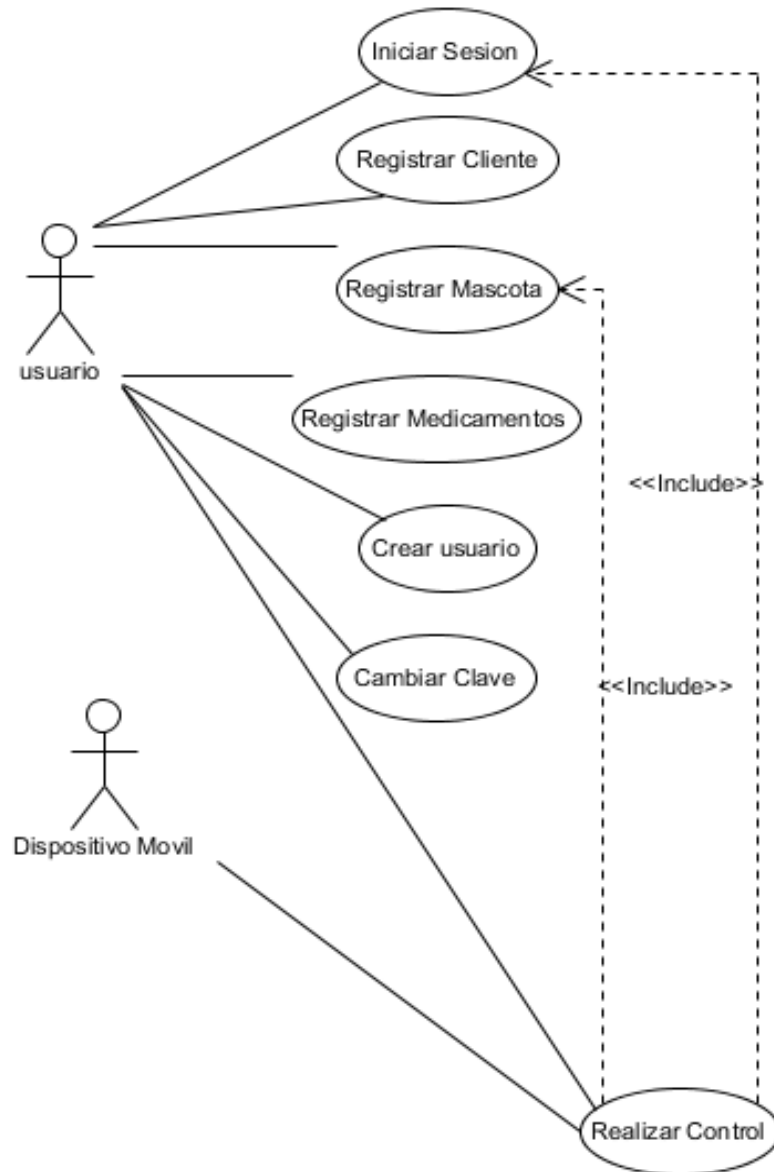


Figura 1. Caso de Uso General

## 6.2.2.2. Casos de uso Expandido

### 6.2.2.2.1. Inicio de Sesión

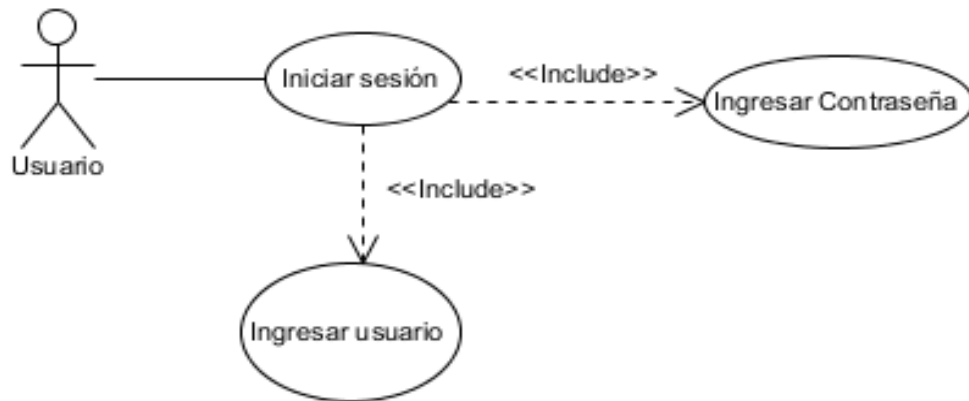


Figura 2. Caso de Uso Expandido Inicio de Sesión

### 6.2.2.2.2. Cambiar Clave

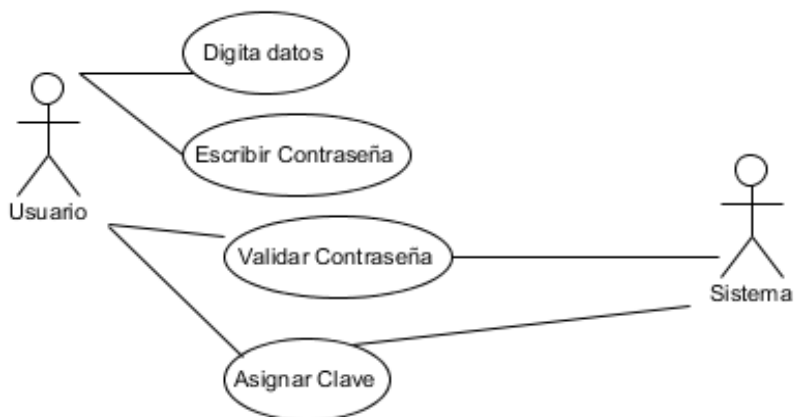


Figura 3. Caso de Uso Expandido Cambiar Clave

### 6.2.2.2.3. Registrar Clientes

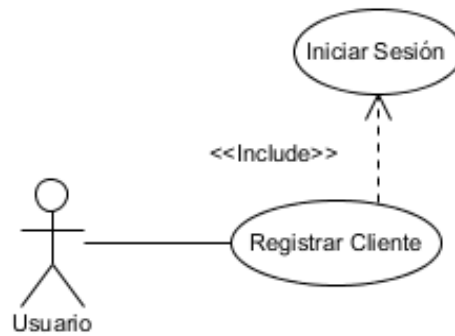


Figura 4. Caso de Uso Expandido Registrar Clientes

### 6.2.2.2.4. Registrar Mascota

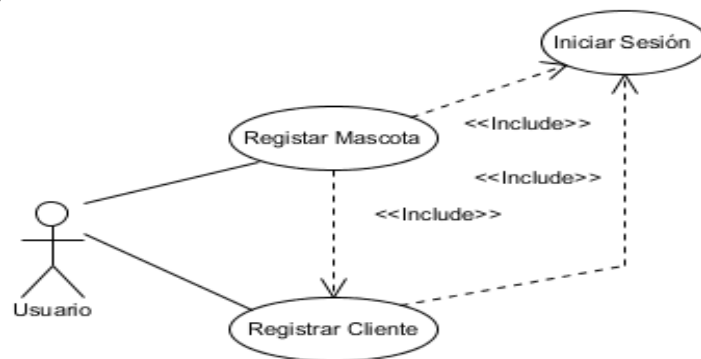


Figura 5. Caso de Uso Expandido Registrar Mascota

### 6.2.2.2.5. Registrar Medicamentos

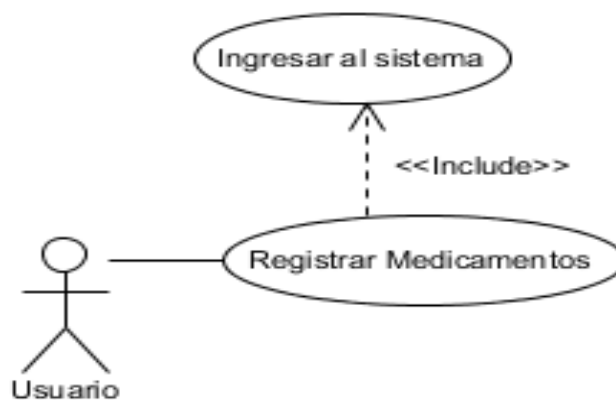


Figura 6. Caso de Uso Expandido Registrar Medicamentos

### 6.2.2.2.6. Registrar Usuario

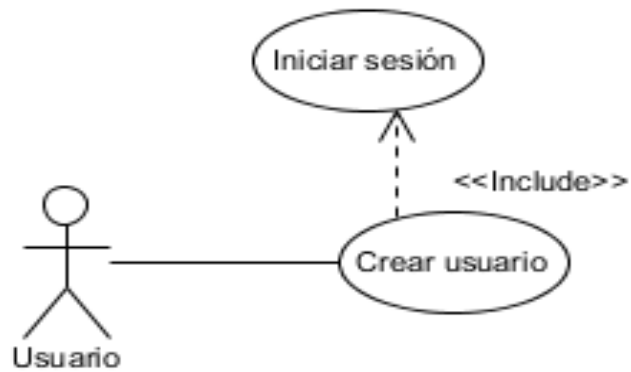


Figura 7. Caso de Uso Expandido Usuario

### 6.2.2.2.7. Registrar Control de Mascota



Figura 8. Caso de Uso Expandido Registrar Control de Mascota

### 6.2.3. Descripción del Caso de Uso Expandido

#### 6.2.3.1. Inicio de Sesión

**Cuadro 11. Caso de Uso Inicio de Sesión**

<b>Caso de Uso</b>	Iniciar Sesión				
<b>Identificador</b>	CU001				
<b>Importancia</b>	Primario	<b>Implementación</b>	Real	<b>Tipo</b>	Básico
<b>Propósito</b>	Controlar el ingreso de los usuarios al sistema				
<b>Resumen</b>	Usuario ingresa nombre y contraseña para acceder al sistema.				
<b>Pre- Condiciones</b>	El sistema no está iniciado				
<b>Flujo Normal de Eventos</b>	<u>Acciones del Actor</u> 1.- Este caso de uso inicia cuando el usuario decide ingresar al sistema. 3.-Ingresa usuario y contraseña		<u>Respuesta del Sistema</u> 2.- Muestra campos de usuario y contraseña a ingresar. 4.-Válida la información ingresada. 5.-Ingresa al sistema.		
<b>Flujo alterno</b>	<u>Acciones del Actor</u>  F4: Vuelva a escribir el usuario y contraseña		<u>Respuestas del Sistema</u> F4: Si el usuario y contraseña son incorrectos, ingresar datos correctos		
<b>Post- condición</b>	Inicializa el sistema				
<b>Operaciones</b>	Ingresar al sistema				
<b>Validaciones</b>	Usuario y Clave correctos				

### 6.2.3.2. Cambiar Clave

**Cuadro 12. Caso de Uso Cambiar Clave**

<b>Caso de Uso</b>	Cambiar Clave				
<b>Identificador</b>	CU002				
<b>Importancia</b>	Primario	<b>Implementación</b>	Real	<b>Tipo</b>	Básico
<b>Propósito</b>	Cambiar la clave del usuario				
<b>Resumen</b>	Usuario realiza el cambio de su clave.				
<b>Pre- Condiciones</b>	El sistema está iniciado				
<b>Flujo Normal de Eventos</b>	<u>Acciones del Actor</u>		<u>Respuesta del Sistema</u>		
	1.- El usuario necesita registrar una nueva clave.		2.- Muestra campos a ingresar.		
	3.-Ingresa datos para la nueva clave		4.-Válida la información ingresada.		
			5.-Registra la nueva clave.		
<b>Flujo alternativo</b>	<u>Acciones del Actor</u>		<u>Respuestas del Sistema</u>		
			F4: La información no es correcta		
	F4: vuelva a introducir la información del dato.				
<b>Post-condición</b>	Registra la nueva clave en el sistema				
<b>Operaciones</b>	Registrar				
<b>Validaciones</b>	Clave correcta				

### 6.2.3.3. Registrar Clientes

**Cuadro 13. Caso de Uso Clientes**

<b>Caso de Uso</b>	Registrar Clientes				
<b>Identificador</b>	CU003				
<b>Importancia</b>	Primario	<b>Implementación</b>	Real	<b>Tipo</b>	Básico
<b>Propósito</b>	Contar con una lista de los clientes				
<b>Resumen</b>	Añadir un nuevo cliente a la lista, se especifica el nombre, apellido, dirección, etc.				
<b>Pre- Condiciones</b>	El sistema está iniciado y el cliente no está registrado				
<b>Flujo Normal de Eventos</b>	<u>Acciones del Actor</u>		<u>Respuesta del Sistema</u>		
	1.- El usuario debe registrar un nuevo cliente.		2.- Muestra campos del cliente a ingresar.		
	3.-Ingresa datos del cliente		4.-Válida la información ingresada.		
			5.-Registra al cliente.		
<b>Flujo alternativo</b>	<u>Acciones del Actor</u>		<u>Respuestas del Sistema</u>		
			F4: La información no completa falta un dato		
	F4: vuelva a introducir la información del dato.				
<b>Post- condición</b>	Registra el nuevo Cliente en el sistema				
<b>Operaciones</b>	Nuevo Guardar Actualizar Eliminar Salir				
<b>Validaciones</b>	IdCliente correcto				

### 6.2.3.4. Registrar Mascota

**Cuadro 14. Caso de Uso Mascota**

<b>Caso de Uso</b>	Registrar Mascota				
<b>Identificador</b>	CU004				
<b>Importancia</b>	Primario	<b>Implementación</b>	Real	<b>Tipo</b>	Básico
<b>Propósito</b>	Contar con una lista de Mascotas				
<b>Resumen</b>	Añadir una nueva mascota a la base de datos, debe especificar el idMascota, IdCliente, Nombre, raza, sexo, Tipo, FechaNac				
<b>Pre- Condiciones</b>	El sistema está iniciado y el cliente no está registrado anteriormente				
<b>Flujo Normal de Eventos</b>	<u>Acciones del Actor</u>		<u>Respuesta del Sistema</u>		
	1.- El usuario debe registrar una nueva mascota.		2.- Muestra campos de la mascota a ingresar.		
	3.-Ingresa datos de la mascota		4.-Válida la información ingresada.		
			5.-Registra la mascota.		
<b>Flujo alternativo</b>	<u>Acciones del Actor</u>		<u>Respuestas del Sistema</u>		
			F4: La información no completa falta un dato		
	F4: vuelva a introducir la información del dato.				
<b>Post- condición</b>	Registra la nueva mascota en el sistema				
<b>Operaciones</b>	Nuevo Guardar Actualizar Eliminar Salir				
<b>Validaciones</b>	IdMascota correcto				

### 6.2.3.5. Registrar Medicamentos

**Cuadro 15. Caso de Uso Medicamentos**

<b>Caso de Uso</b>	Registrar Medicamento				
<b>Identificador</b>	CU005				
<b>Importancia</b>	Primario	<b>Implementación</b>	Real	<b>Tipo</b>	Básico
<b>Propósito</b>	Contar con una lista de medicamentos				
<b>Resumen</b>	Añadir un nuevo medicamento a la lista, en la cual se debe especificar el Id, nombre, descripción, observación.				
<b>Pre- Condiciones</b>	El sistema está iniciado y el medicamento no está registrado				
<b>Flujo Normal de Eventos</b>	<u>Acciones del Actor</u>		<u>Respuesta del Sistema</u>		
	1.- El usuario debe registrar un medicamento.		2.- Muestra campos del medicamento a ingresar.		
	3.-Ingresa datos del medicamento		4.-Válida la información ingresada.		
			5.-Registra el medicamento.		
<b>Flujo alternativo</b>	<u>Acciones del Actor</u>		<u>Respuestas del Sistema</u>		
			F4: La información no completa falta un dato		
	F4: vuelva a introducir la información del dato.				
<b>Post- condición</b>	Registra el nueva medicamento en el sistema				
<b>Operaciones</b>	Nuevo Guardar Actualizar Eliminar Salir				
<b>Validaciones</b>	Id Medicamento correcto				

### 6.2.3.6. Registrar Usuario

**Cuadro 16. Caso de Uso Usuario**

<b>Caso de Uso</b>	Registrar Usuario				
<b>Identificador</b>	CU006				
<b>Importancia</b>	Primario	<b>Implementación</b>	Real	<b>Tipo</b>	Básico
<b>Propósito</b>	Registrar los usuarios en el sistema				
<b>Resumen</b>	Registrar los usuarios que manipularan el sistema.				
<b>Pre- Condiciones</b>	El sistema está iniciado y el usuario no está registrado				
<b>Flujo Normal de Eventos</b>	<u>Acciones del Actor</u>		<u>Respuesta del Sistema</u>		
	1.- Iniciar el sistema para registrar un usuario.		2.- Muestra datos del usuario a ingresar.		
	3.-Ingresa datos del usuario		4.-Válida la información ingresada		
			5.-Registra usuario.		
<b>Flujo alterno</b>	<u>Acciones del Actor</u>		<u>Respuestas del Sistema</u>		
			F4: La información no completa falta un dato		
	F4: vuelva a introducir la información del dato.				
<b>Post-condición</b>	Registra el nueva medicamento en el sistema				
<b>Operaciones</b>	Nuevo Guardar Actualizar Eliminar Salir				
<b>Validaciones</b>	Idusuario correcto				

### 6.2.3.7. Registrar Realizar Control de Mascota

**Cuadro 17. Caso de Uso Realizar Control de Mascota**

<b>Caso de Uso</b>	Registrar Usuario				
<b>Identificador</b>	CU007				
<b>Importancia</b>	Primario	<b>Implementación</b>	Real	<b>Tipo</b>	Básico
<b>Propósito</b>	Realizar un control del estado de salud de las mascotas				
<b>Resumen</b>	Llevar un control de la evolución de cada mascota.				
<b>Pre- Condiciones</b>	El sistema está iniciado				
<b>Flujo Normal de Eventos</b>	<u>Acciones del Actor</u>		<u>Respuesta del Sistema</u>		
	1.- Este caso inicia cuando el usuario registra el control que se realiza a cada mascota.		2.- Muestra datos a ingresar.		
	3.- Escoje Archivo xml enviado por el dispositivo movil		4.-Procesa datos.		
	5.-Ingresa datos para realizar el control de cada mascota		6.-Verifica que los datos ingresados sean correctos		
			7.- Guarda control de mascota		
<b>Flujo alterno</b>	<u>Acciones del Actor</u>		<u>Respuestas del Sistema</u>		
	F3:No existe archivo		F3: No muestra información alguna		
	F3: Ingresar datos		F5: Datos Incorrectos		
	F4: Ingresa nuevos datos				
<b>Post- condición</b>	Registra el nueva medicamento en el sistema				
<b>Validaciones</b>	Cliente, Mascotas y Medicamentos registrados				

## 6.2.4. Diagramas de Actividad

### 6.2.4.1. Iniciar Sesión

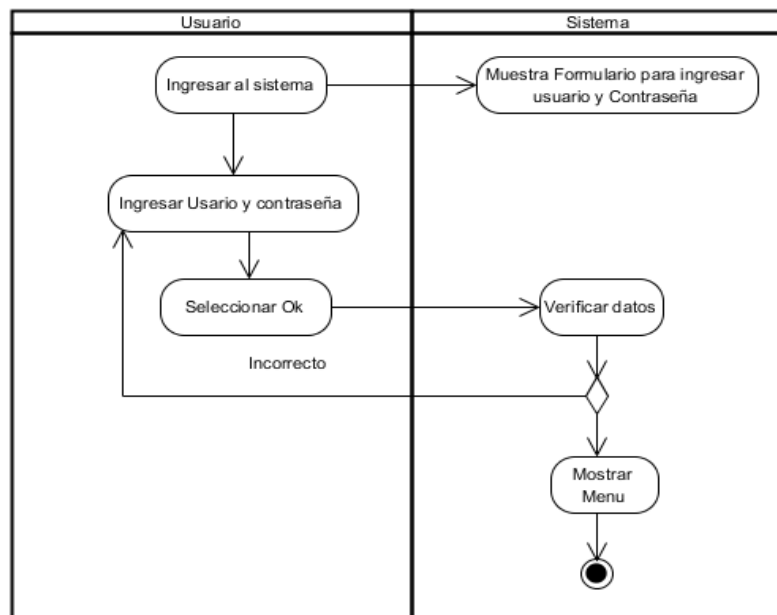


Figura 9. Diagrama de Actividad Inicio de Sesión

### 6.2.4.2. Cambiar Clave

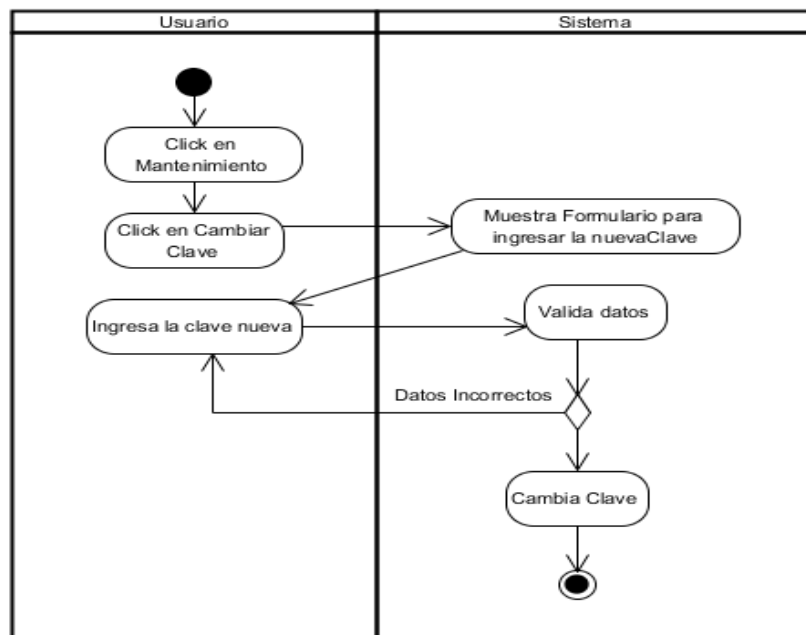


Figura 10. Diagrama de Actividad Cambiar Clave

### 6.2.4.3. Registrar Cliente

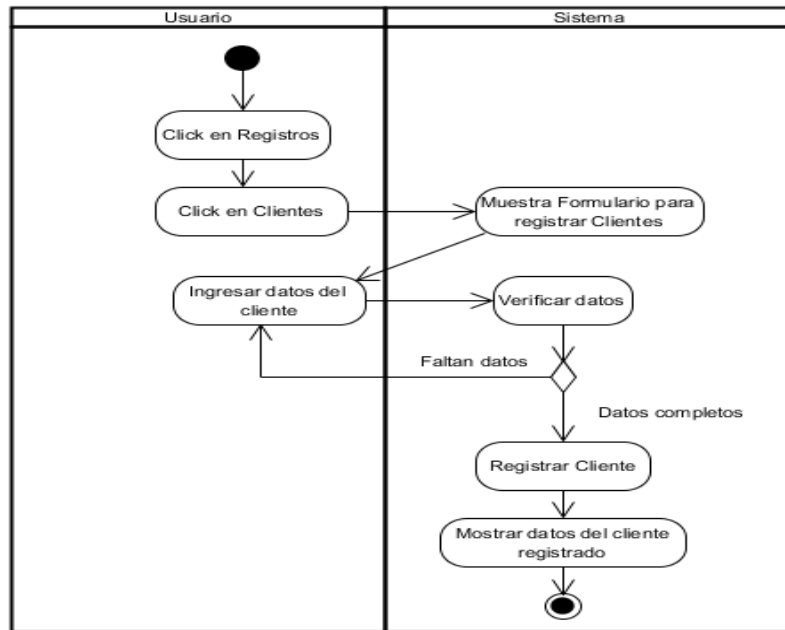


Figura 11. Diagrama de Actividad Registrar Cliente

### 6.2.4.4. Registrar Mascota

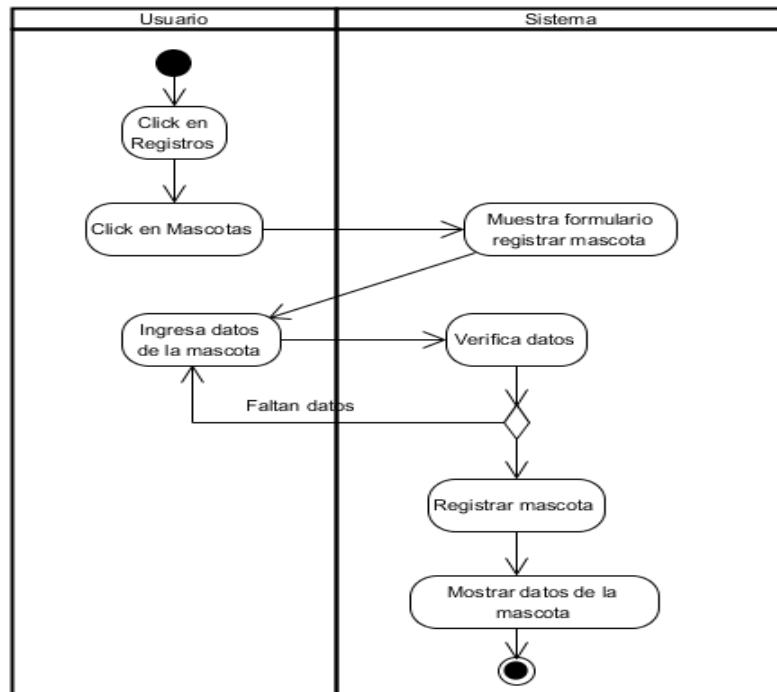


Figura 12. Diagrama de Actividad Registrar Mascota

### 6.2.4.5. Registrar Medicamentos

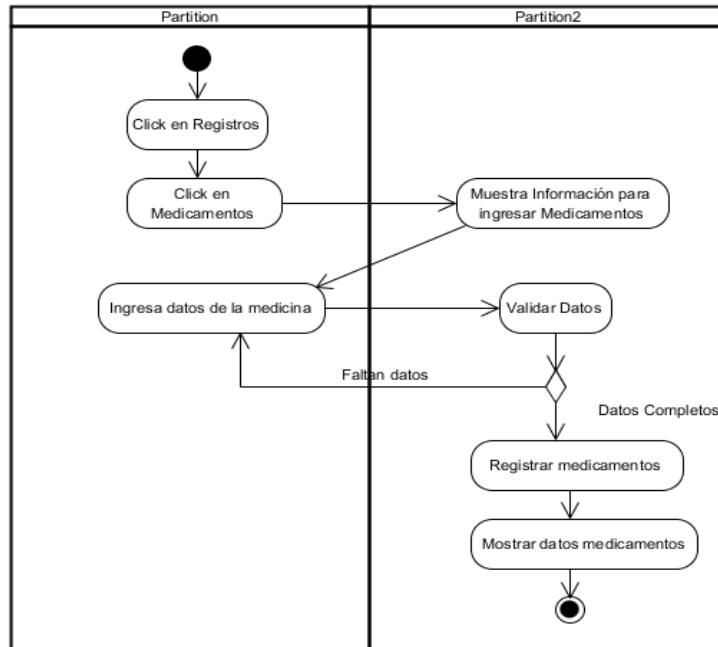


Figura 13. Diagrama de Actividad Registrar Medicamentos

### 6.2.4.6. Crear Usuario

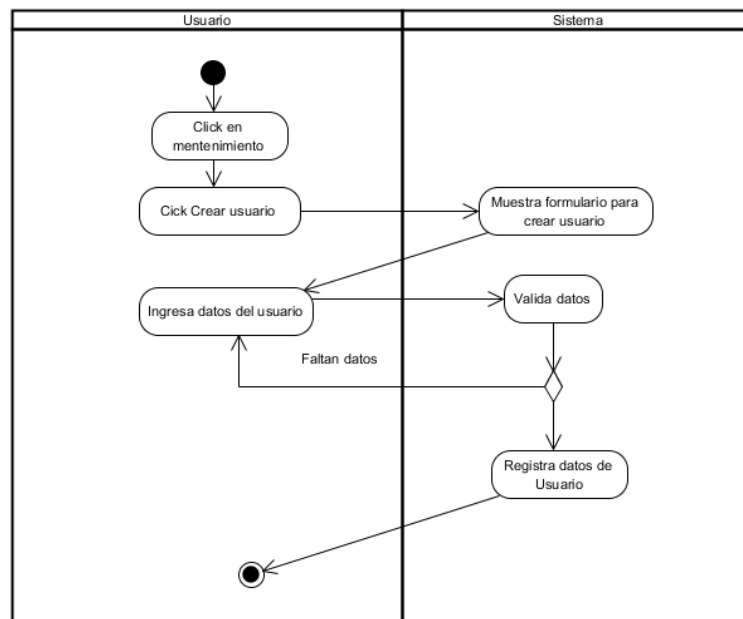
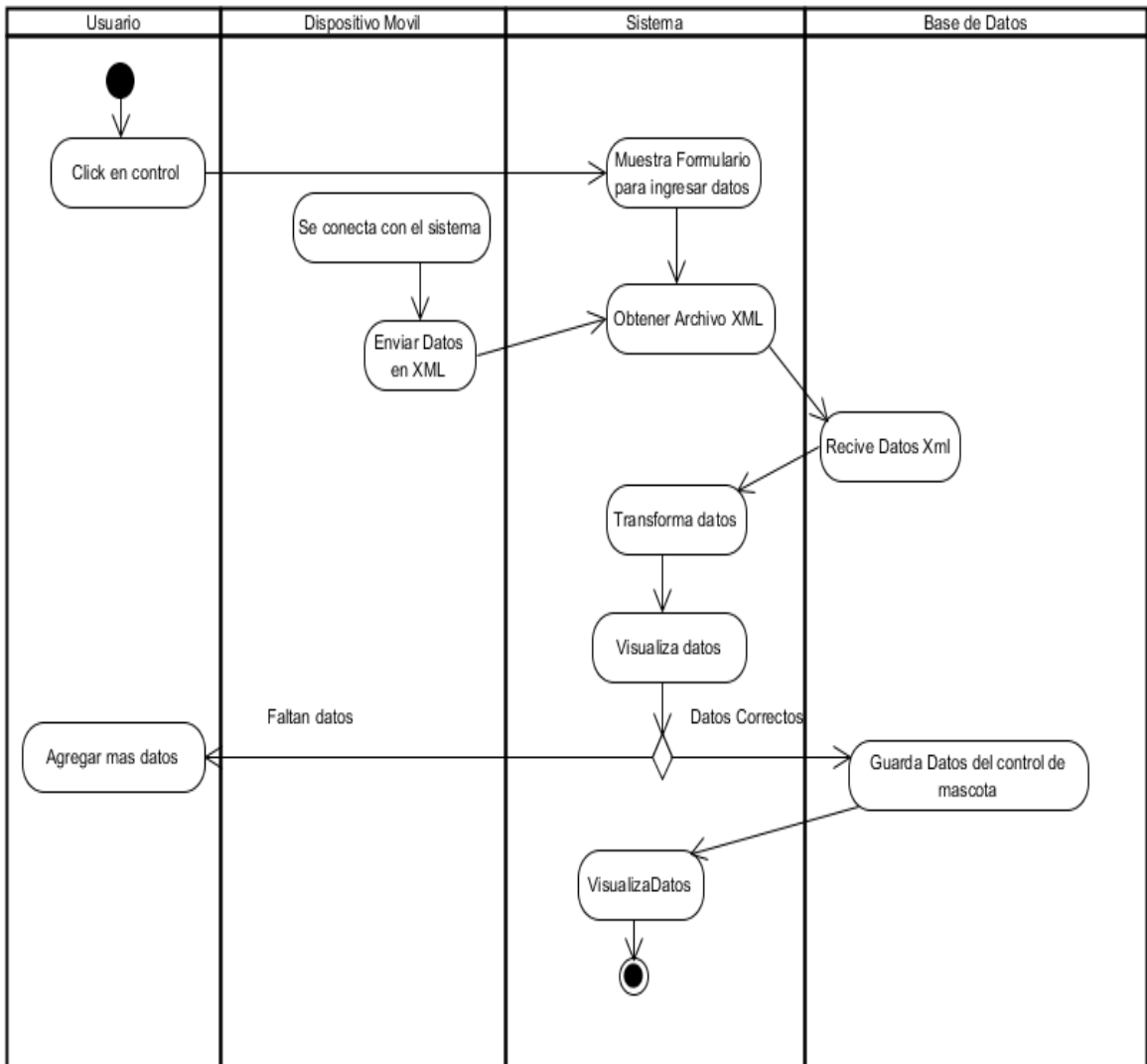


Figura 14. Diagrama de Actividad Crear Usuario

### 6.2.4.7. Realizar Control de Mascota



**Figura 15. Diagrama de Actividad Registrar Control de Mascota**

## 6.2.5. Diagramas de Secuencia

### 6.2.5.1. Inicio de Sesión

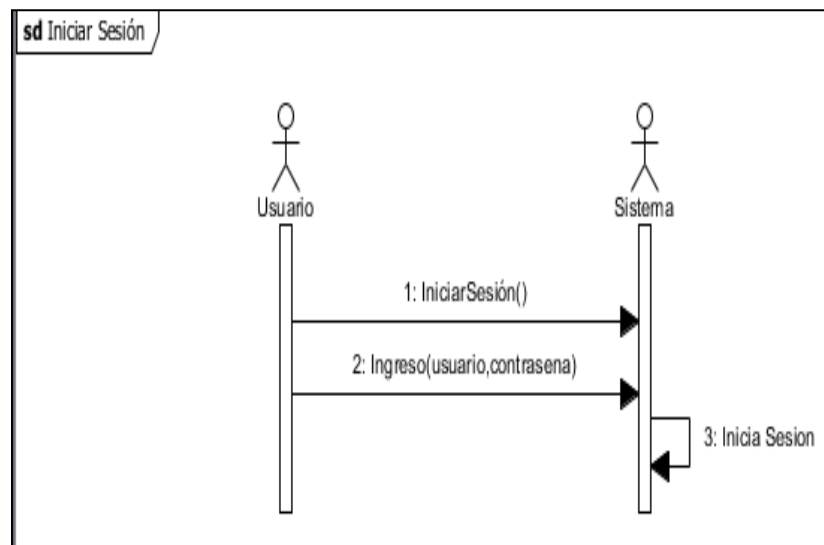


Figura 16. Diagrama de Secuencia Inicio de Sesión

### 6.2.5.2. Cambiar Clave

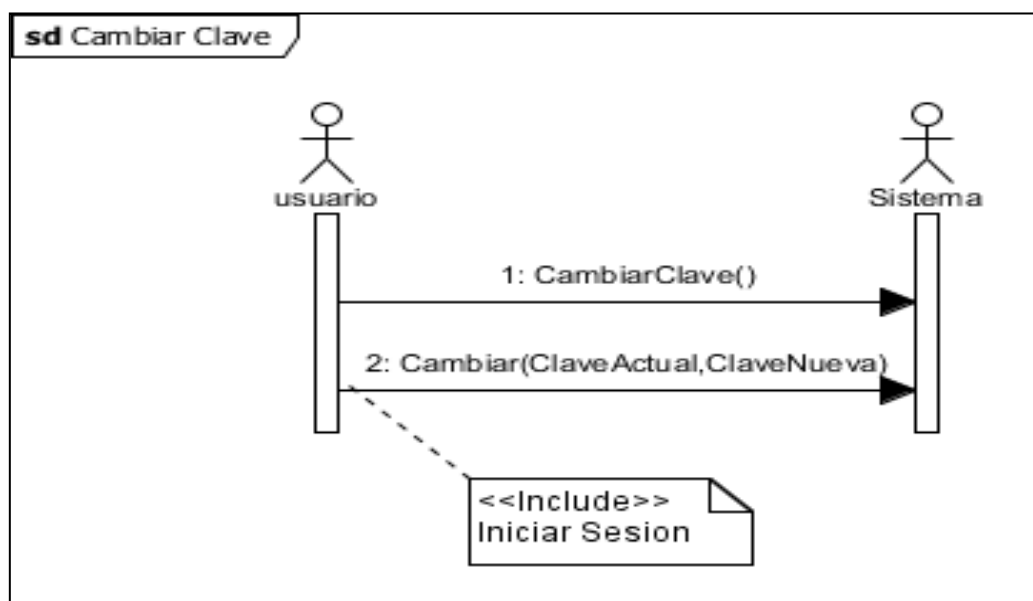


Figura 17. Diagrama de Secuencia Cambiar Clave

### 6.2.5.3. Registrar Mascota

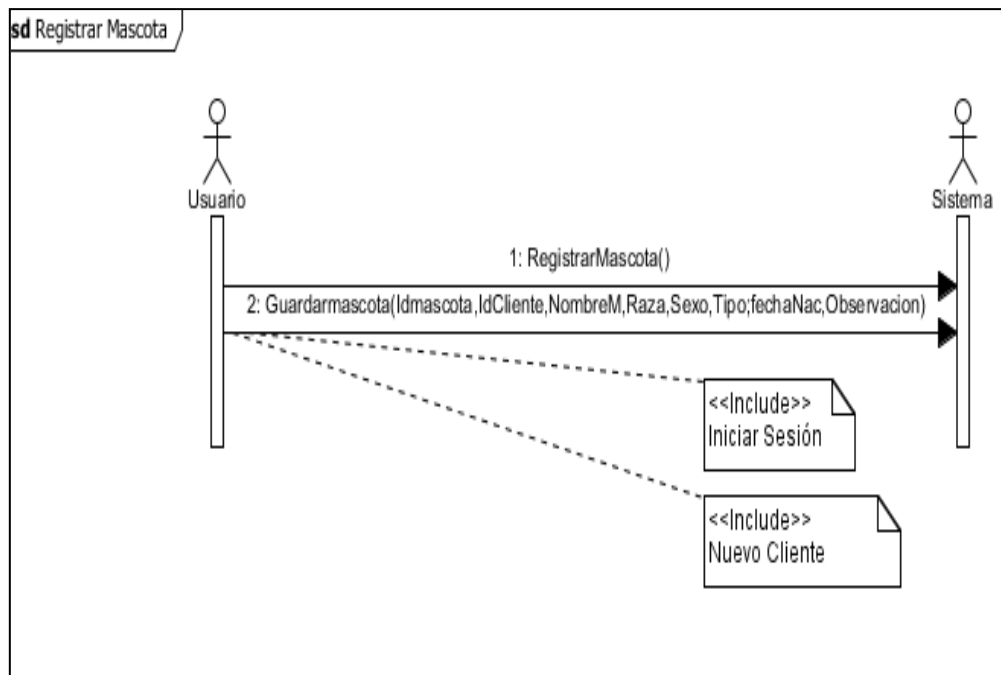


Figura 18. Diagrama de Secuencia Registrar Mascota

### 6.2.5.4. Registrar Medicamentos

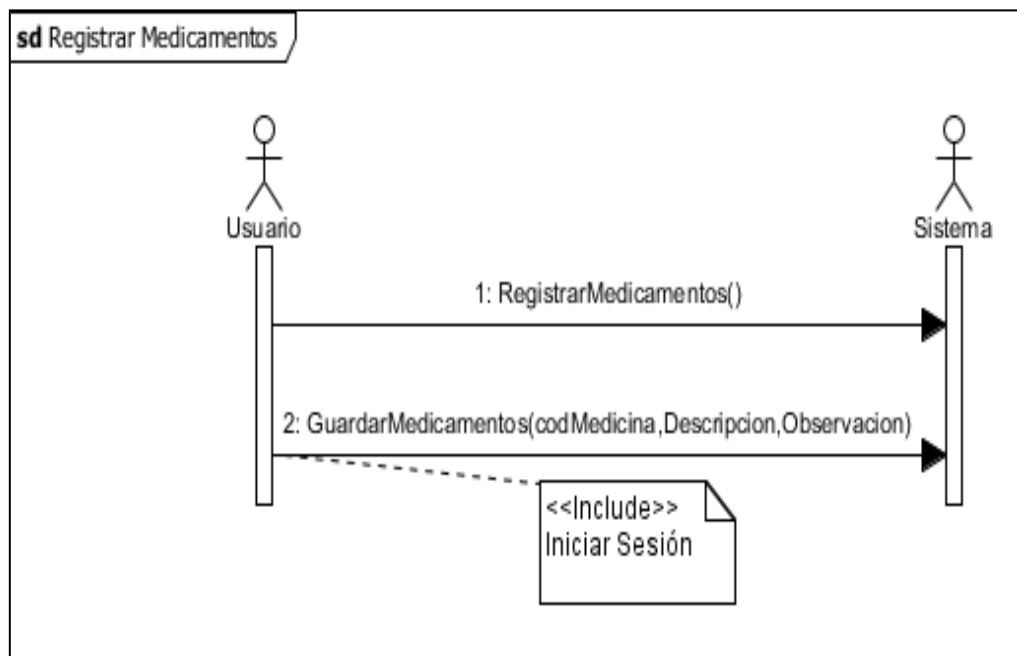


Figura 19. Diagrama de Secuencia Registrar Medicamentos

### 6.2.5.5. Crear Usuario

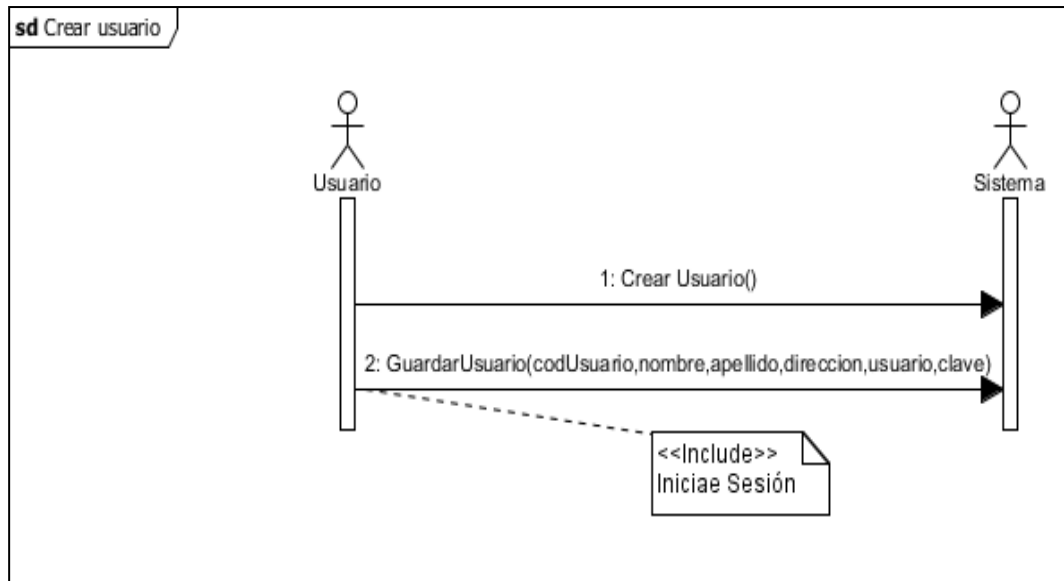


Figura 20. Diagrama de Secuencia Crear Usuario

### 6.2.5.6. Realizar Control

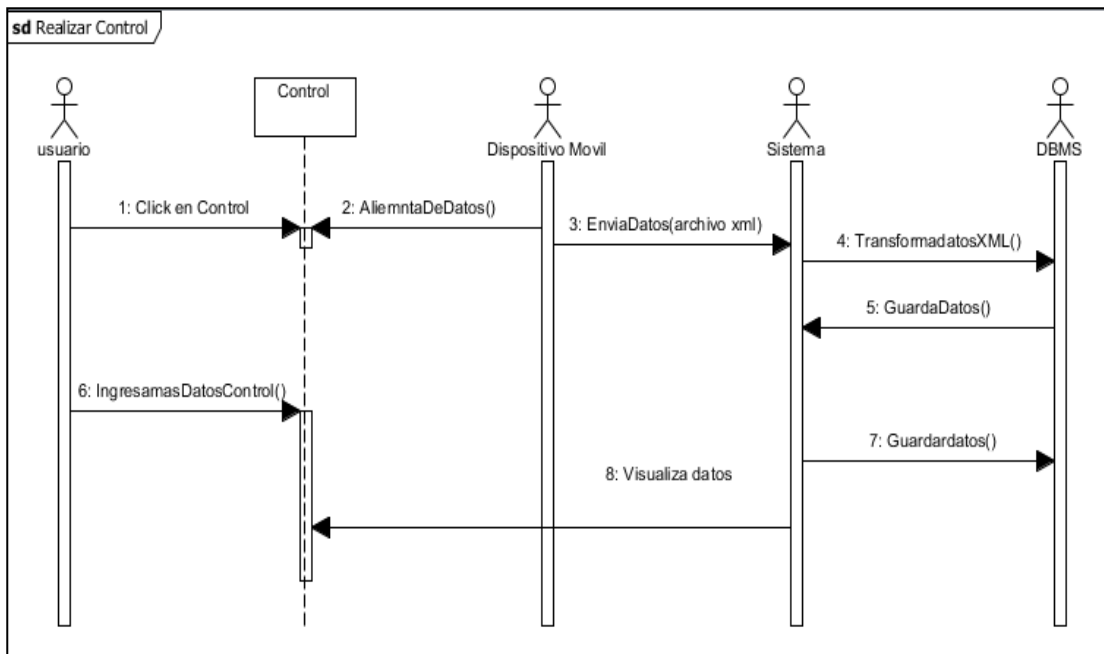


Figura 21. Diagrama de Secuencia Realizar Control

### 6.2.6. Diagrama Físico de la Base de Datos

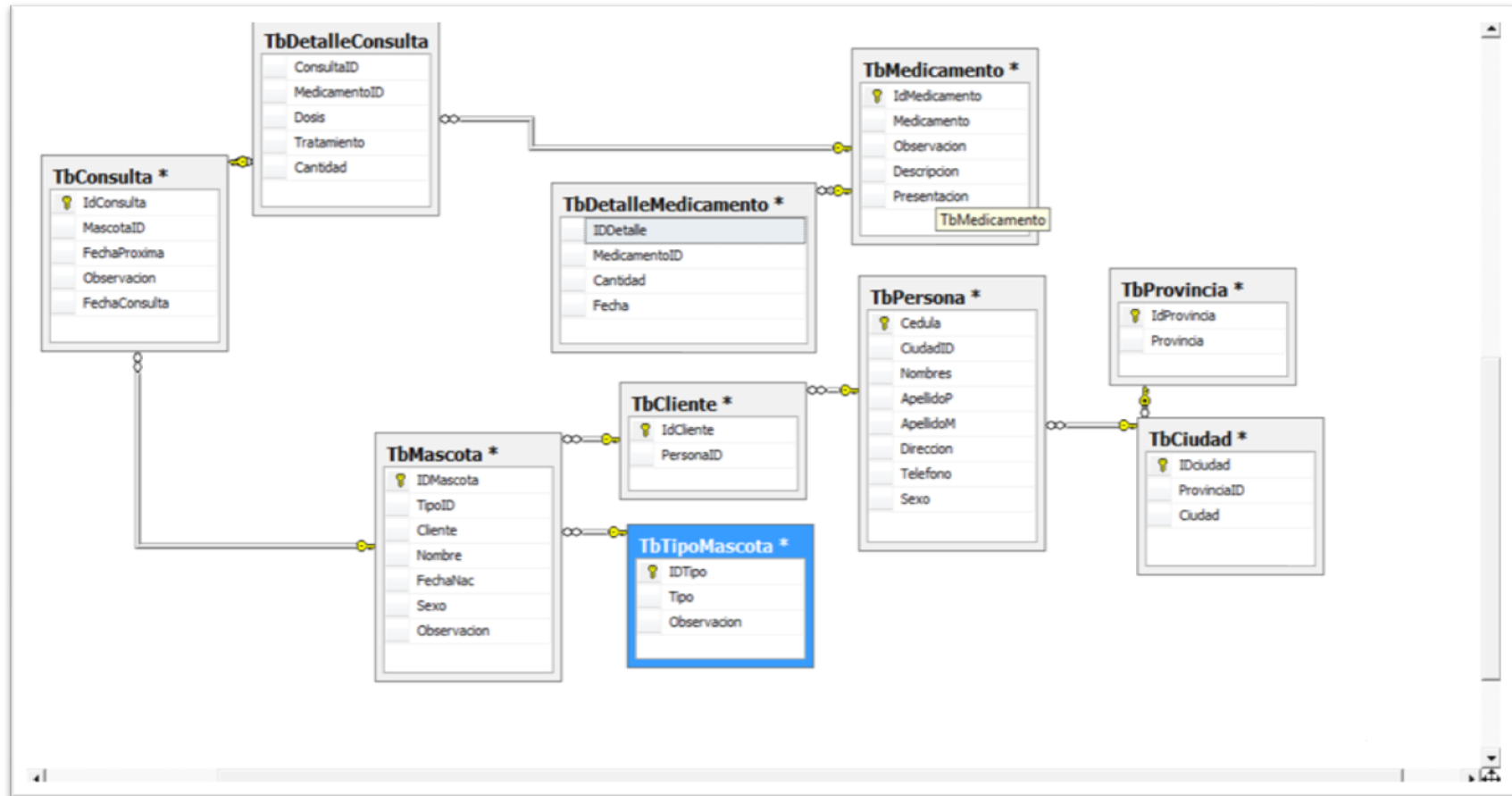


Figura 22. Diagrama Físico de la Base de Datos

### 6.2.7. Diagrama Entidad Relación

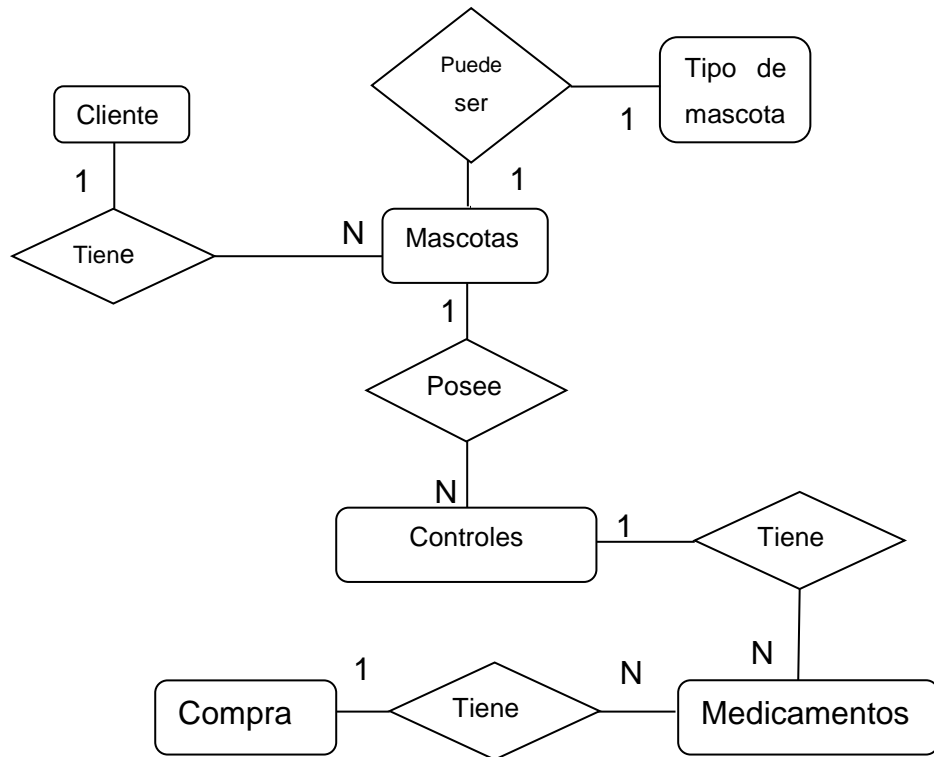


Figura 23. Diagrama Entidad Relación

## 6.2.8. Diagrama de Clases

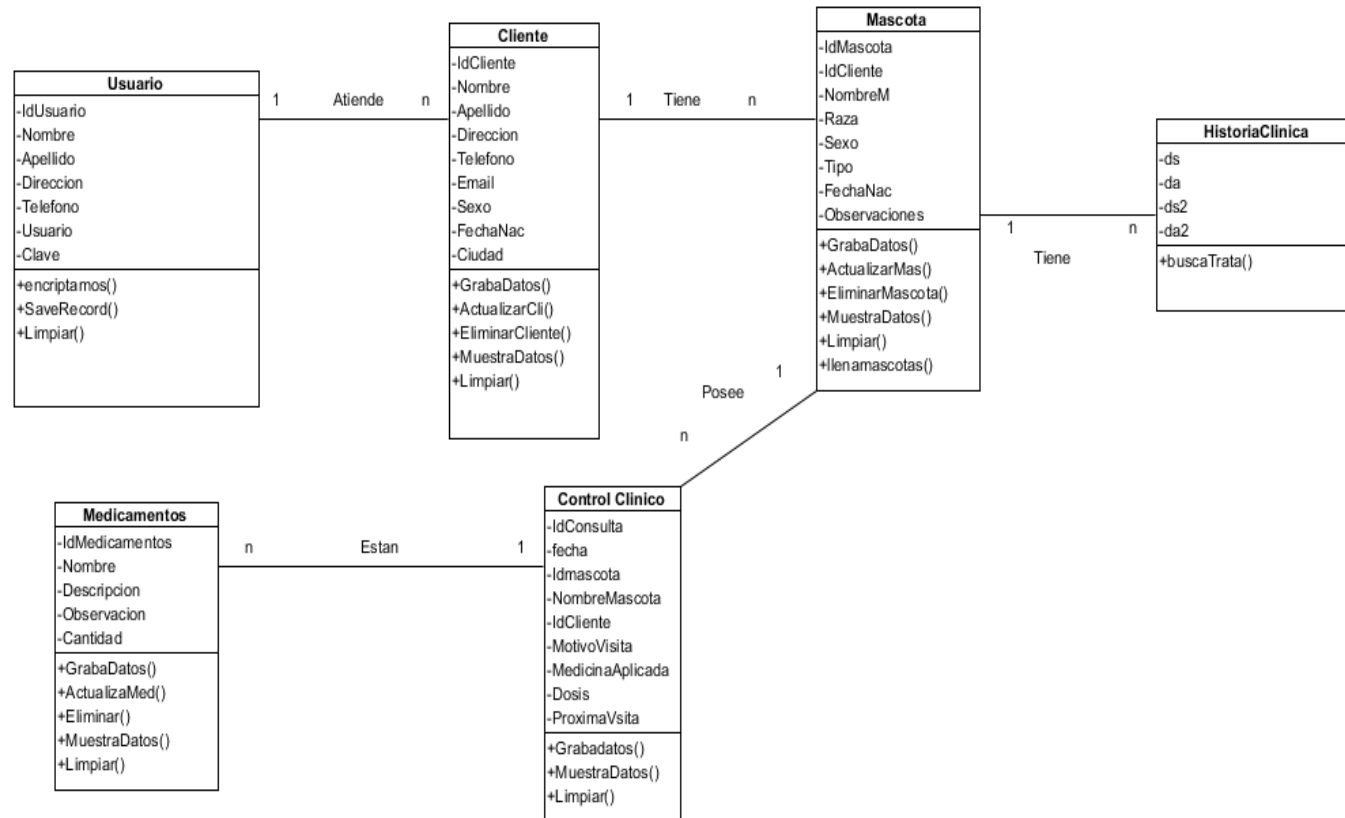


Figura 24. Diagrama de Clases

## VII. CONCLUSIONES

De acuerdo a los objetivos planteados y a los resultados obtenidos en la investigación, se obtuvieron las siguientes conclusiones:

Con el análisis de los procesos de flujo de información utilizados actualmente se demostró que la atención a los clientes y despacho de medicamentos son lentos debido a que todo se maneja manualmente provocando pérdida de tiempo en la ejecución de cada uno de ellos, por no tener la documentación en regla en el momento oportuno.

Se identificaron los procesos de flujo de información referentes a la atención de clientes y mascotas, ingresos y egresos de medicamentos; mismos que están considerandos en el nuevo Sistema de Información con Aplicación Móvil para el Control Administrativo de la empresa Servicios Pecuarios Véliz, para lo cual la aplicación permite la sincronización de datos tanto en servidor como el dispositivo móvil Pocket PC.

Se determinó que el DBMS más óptimo para el desarrollo de la Base de Datos del sistema es SQL Server debido a que utiliza grandes volúmenes de información, se encarga de la administración de datos, procesamiento de transacciones y consultas e integridad de los datos.

La herramienta más adecuada para el desarrollo del sistema fue la aplicación del lenguaje de programación Visual Studio .Net 2008 considerando que es un software bastante útil para el desarrollo de aplicaciones, el cual brindó las facilidades necesarias, proporcionando herramientas de diseño y creación de aplicaciones no solo para PC sino también para Pocket PC.

## VIII. RECOMENDACIONES

Las recomendaciones se las formuló con el propósito de alcanzar resultados favorables para la empresa Servicios Pecuarios Véliz.

Se debe respaldar la información para optimizar el proceso si se tiene cambios importantes de los datos, con la finalidad de asegurar la confiabilidad del sistema y realizar un cronograma de mantenimiento para la PC y el Pocket PC, para no tener inconvenientes a futuro.

Los usuarios se deben capacitar sobre el programa y el buen funcionamiento del sistema con aplicación móvil (Pocket PC), para mejorar la actualización de los datos de los clientes y mascotas que son atendidas en la empresa.

Se debe ampliar el sistema de información con aplicación móvil para que sea utilizado en otras empresas permitiendo la automatización y buen funcionamiento, para satisfacer las necesidades de los usuarios.

Se recomienda leer el Manual de Usuario para seguir las instrucciones que hay en cada formulario del sistema, para un buen manejo de la Aplicación Móvil (Pocket Pc), conjuntamente con la aplicación desde el servidor.

## IX. RESUMEN

En el siguiente trabajo se ha elaborado un Sistema de Información con Aplicación Móvil para el Control Administrativo de la empresa Servicios Pecuarios Véliz, cuyo campo de acción es el área administrativa por lo que se hizo un estudio sobre los principales campos problemáticos que afronta la empresa en el proceso de su información de registro de clientes y mascotas.

Se determinó la necesidad de construir un software que se pueda obtener un mejor control de cada uno de los clientes que visitan la empresa para el respectivo tratamiento de sus mascotas.

Se utilizó la metodología Crystal Clear que es un proceso de iteraciones de desarrollo de software, esta metodología ha sido el pilar fundamental para desarrollar cada una de las etapas por el cual esta conformado el software, también se desarrolla un análisis del contexto que se realizó al iniciar el proceso de la investigación de campo esto permitió conocer los procesos y flujos de la información actual de la empresa.

El sistema de información ha sido desarrollado usando las herramientas Microsoft Visual Studio.Net 2008 y Base de Datos SQL Server, permitiendo al sistema realizar las siguientes funciones: ingreso y registro de datos de los clientes y mascotas, genera la historia clínica y control de la mascota, reporte del tratamiento de la mascota, reporte de los medicamentos en existencia.

En resumen, este sistema de información con aplicación móvil permitirá al gerente de la empresa llevar un mejor Control Administrativo, y mantener actualizada la información de los clientes y sus mascotas.

## X. SUMMARY

In the following work has developed an information system with Mobile Application for the Administrative Control of the Livestock Services company Véliz, whose field of action is the administrative area so it became a study on the major problem areas facing the company in the process of your registration information to clients and pets.

Is also made and identified the need to build a software that will be able to get a better control of each of the customers who visit the company for the respective treatment of your pets.

Used the methodology Crystal Clear that is a process of iterations of development of software, this methodology has been the pillar fundamental to develop each one of the stages by which this conformed the software, also develops an analysis of the context that realized when initiating the process of the investigation of field perdition know the processes and flows of the current information of the company.

The system of information has been developed using the tools Microsoft Visual Studio.Net 2008 And Base of Data SQL Server, allowing to the system realize the following functions: entry and register of data of the clients and pets, generates the clinical history and control of the pet, report of the treatment of the pet, report of the medicines in existence.

In summary, this system of information with mobile application will allow to the gerent of the company carry a better Administrative Control, and maintain updated the information of the clients and his pets.

## XI. BIBLIOGRAFIA

AGILES Metodología Crystal Clear, disponible en <http://www.agileshift.cl/Tutorial/DesarrolloAgilParte2.pdf>

ALAVA Francisco (2010), Administración, Conceptos Básicos, segunda edición Pp 20-25.

ALEGSA Sistema Informático (2010), disponible en <http://www.alegsa.com.ar/Dic/sistema%20informatico.php>

ALEGSA (2007), disponible en <http://www.alegsa.com.ar/Dic/pocket%20pc.php>

ARTEAGA (2005), disponible en <http://www.devjoker.com/contenidos/conceptosgenerales-NET/95/common-lenguaje-specification-CLS.aspx>

ANAYA MULTIMEDIA (2003), Programación en Access

CARRIEL (2006), disponible en <http://www.misrespuestas.com/que-es-wifi.html>

CHARTE Ojeda Francisco (2008), Programación, Base de Datos con Microsoft Visual Basic.Net, Anaya multimedia año 2008

CHARTE Ojeda Francisco (2008), SQL SERVER, disponible en: <http://www.casadellibro.com/libros/charteojedafrancisco/charte2oje da32francisco>

CHARTE Ojeda Francisco (2006), VISUAL STUDIO.NET, disponible en: <http://www.casadellibro.com/libros/charteojedafrancisco/charte2oje da32francisco>

DUBOIS, PAUL Ed. Anaya Multimedia (2005), Primera edición, Español

EDITORIAL Anaya multimedia (2007), Microsoft Paso a Paso, Base de Datos con SQL Server 2005, disponible en: [www.anayamultimedia.es](http://www.anayamultimedia.es)

FERNÁNDEZ Alarcón Vicente (2006), Desarrollo de sistemas de información, Edicions UPC, Pp 12-18.

MI TECNOLÓGICO (2010), Definición de Administración, disponible en <http://www.mitecnologico.com/Main/DefinicionAdministracionYConceptosBasicos>

PRESSMAN Roger S. (2005), Ingeniería del Software: Un Enfoque Práctico, Sexta Edición Pp 15-34.

PEREZ Damián (2007), Base de Datos, disponible en: <http://www.maestrodelaWeb.com>

QUISPE Rodolfo (2007), Ingeniería de software, disponible en: <http://www.rodolfoquispe.org/blog/queeslaingenieriadesoftware.php>

SERVICIOS Pecuarios Véliz, (2011) Dr. Veterinario. Abraham Véliz Pincay, Gerente Propietario



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO  
UNIDAD DE ESTUDIOS A DISTANCIA  
CARRERA SISTEMAS



**Encuesta a Clientes de la Empresa Servicios Pecuarios Véliz**

Encuesta realizada a los clientes de la empresa antes mencionada

**1. ¿Ha tenido usted inconvenientes con el dueño de la empresa o sus colaboradores por la atención?**

SI ( )      NO ( )

**2. ¿Le resulta molesto que el dueño de la empresa revise sus datos en carpetas ya que de esta manera no es rápida la atención?**

SI ( )      NO ( )

**3. ¿La empresa facilita el control de tratamiento de su animal?**

SI ( )      NO ( )

**4. ¿Qué considera usted que se debería mejorar en la empresa?**

COSTOS ( )    ATENCION AL CLIENTE ( )    SERVICIOS ( )

**5. ¿Le gustaría que esta empresa implemente un Sistema para una mejor atención?**

SI ( )      NO ( )



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO  
UNIDAD DE ESTUDIOS A DISTANCIA  
CARRERA SISTEMAS



**Encuesta a los Proveedores**

Encuesta realizada a los Proveedores de la empresa Servicios Pecuarios Véliz

**1. ¿Lleva un registro de los productos que adquiere la empresa Servicios Pecuarios Veliz?**

SI ( )      NO ( )

**2. ¿Usted qué forma de pago le brinda a la empresa Servicios Pecuarios Véliz?**

CONTADO ( )      CREDITO ( )

**3. ¿Cómo califica la atención en esta empresa?**

EXCELENTE ( )      BUENA ( )      MALA ( )

**4. ¿En qué plazo de tiempo la Empresa Servicios Pecuarios Veliz realiza sus pedidos?**

DIARIO ( )      SEMANAL ( )      MENSUAL ( )

**5. ¿Está de acuerdo que la empresa se automatice con un Sistema Informático?**

SI ( )      NO ( )



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO  
UNIDAD DE ESTUDIOS A DISTANCIA  
CARRERA SISTEMAS



## **Entrevista**

Entrevista al Gerente Propietario de la empresa Servicios Pecuarios Véliz

- 1. ¿Qué tipo de animales atiende usted?**
- 2. ¿Qué datos del cliente registra su archivo?**
- 3. ¿Cómo realiza el registro de todos sus clientes que acuden a su empresa con sus mascotas?**
- 4. ¿Lleva un registro de los productos y medicamentos que aplica a las mascotas?**
- 5. ¿Cuáles son los productos utilizados en el tratamiento de la mascota?**
- 6. ¿Cuáles son sus proveedores de su empresa?**

# MANUAL DE USUARIO

## Introducción

Este documento cuenta con información adecuada para el manejo del Sistema de Información con Aplicación Móvil para el Control Administrativo de la empresa SERVICIOS PECUARIOS VELIZ para su buen funcionamiento.

El sistema de Información con Aplicación Móvil fue creado con el objetivo de brindar facilidades en los procesos que realiza la empresa. El administrador del sistema podrá registrar datos de los clientes y sus mascotas que acuden a realizar el respectivo tratamiento de su animal.

Es de gran importancia consultar este manual antes y/o durante la utilización del sistema, y lo guiará paso a paso en el manejo de las funciones. Con el fin de facilitar la comprensión del manual, se incluye gráficos ilustrativos.

## Objetivo del Manual

- Guía para acceder al Sistema de Información con Aplicación Móvil para el Control Administrativo.
- Conocer la utilización del sistema, mediante una descripción detallada e ilustrada de las opciones, es decir familiarizarse con el sistema para un buen funcionamiento del mismo.

El manual está orientado a los usuarios involucrados en la etapa de Operación del Sistema de Información con Aplicación Móvil para el Control Administrativo.

## Lo que debe Conocer

Los conocimientos mínimos que deben tener las personas que operarán el sistema y utilizarán el manual son:

- Conocimientos básicos acerca de Programas Utilitarios.
- Conocimiento básico de Windows.

### **Especificaciones Técnicas**

Para la Implementación del software se requiere de un equipo con las siguientes características:

<b>Hardware</b>	<b>Software</b>
Pc Intel Core 2 Duo	Sistema Operativo Windows 7
Impresora Lexmark	Visual Studio 2008
Pocket Pc	SQL Server 2008

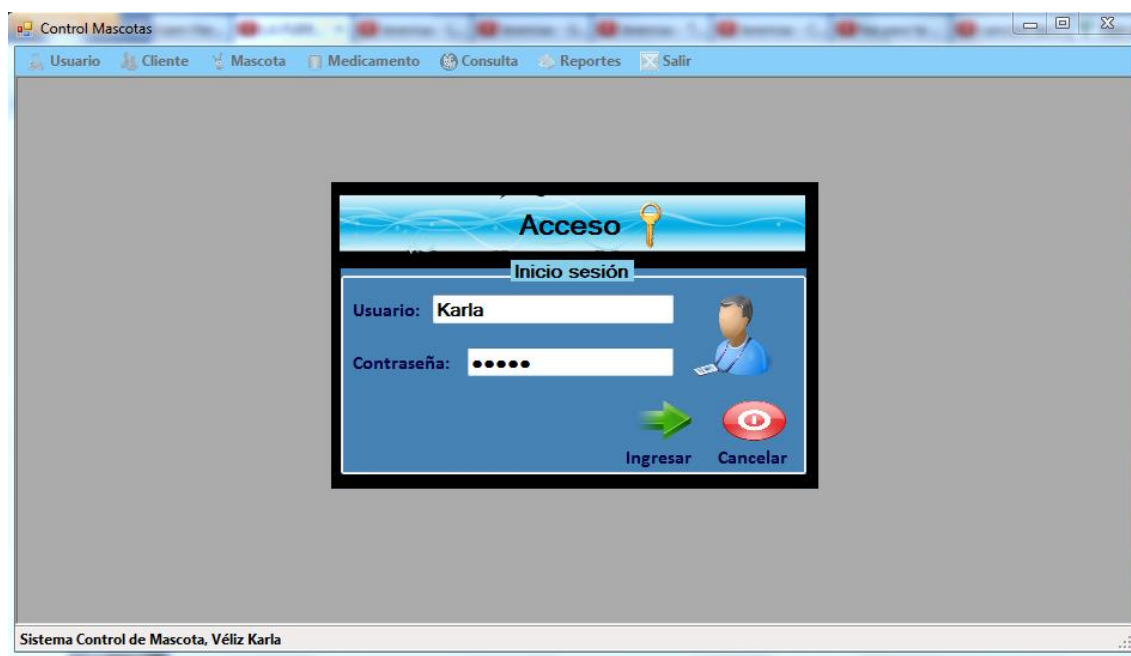
## Manual Usuario para el Sistema de Control de Máscotas desde el Servidor

### Ingreso al Sistema

Para acceder al Sistema de Información con Aplicación de escritorio para el Control Administrativo de la empresa SERVICIOS PECUARIOS VELIZ.

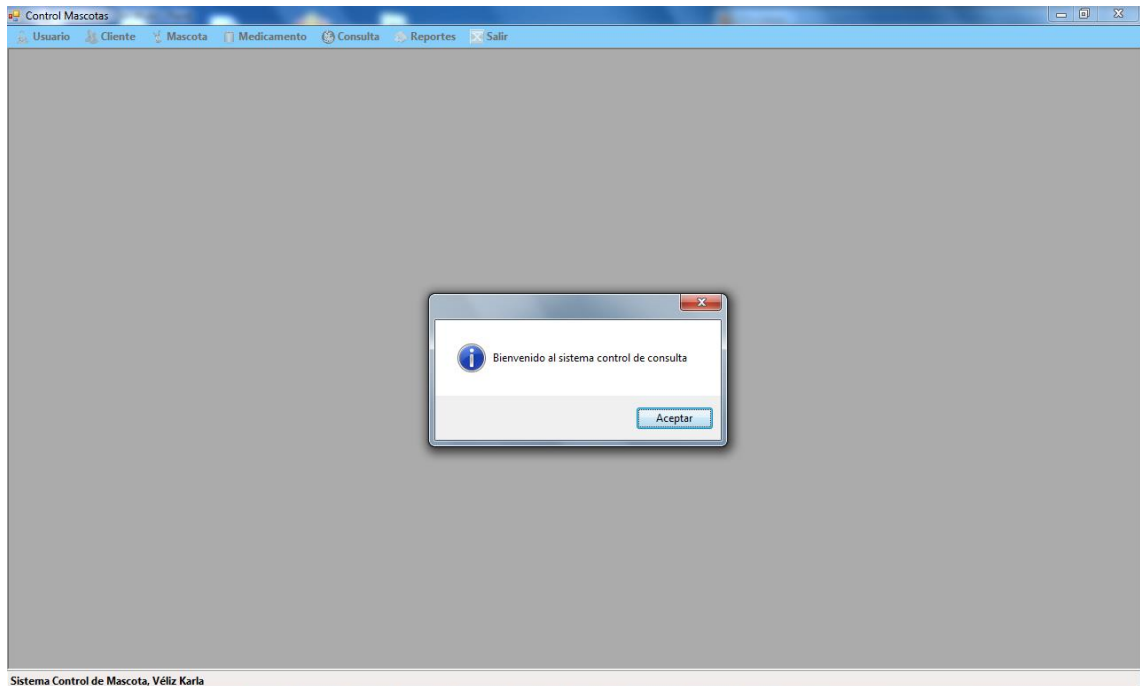
La pantalla inicial permite ingresar al sistema.

### MENU INGRESO AL SISTEMA



**Figura Nro. 25 (Ingreso al Sistema)**

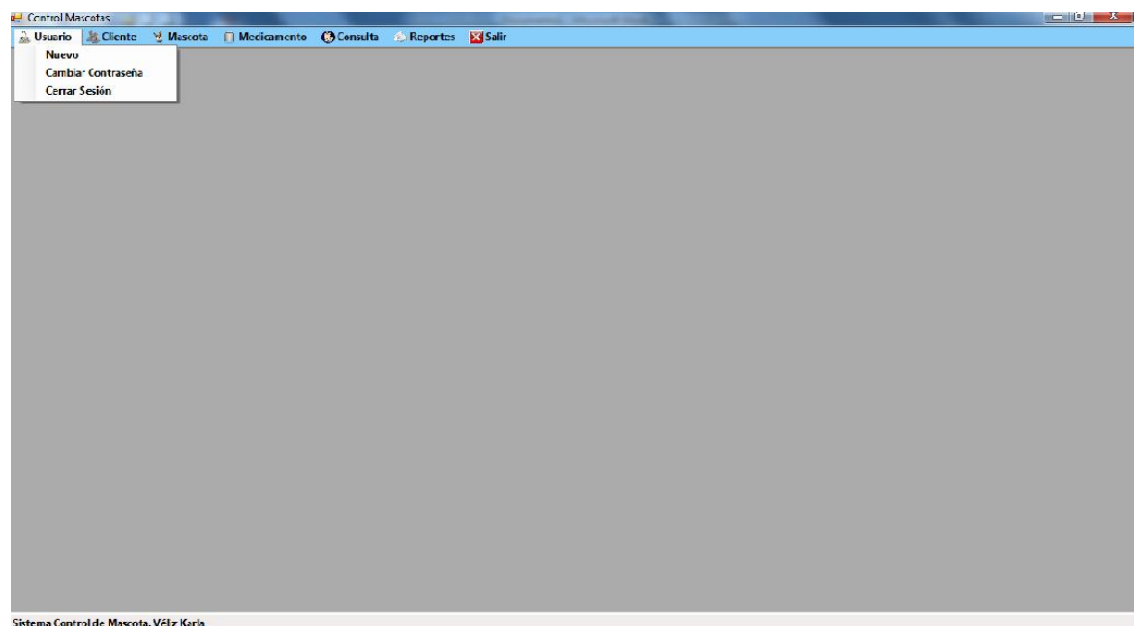
En la pantalla de Inicio Fig.25 se visualizará un formulario que solicita se digite el usuario y contraseña. Al ingresar el usuario y contraseña correctos el sistema presentará un mensaje informando el correcto ingreso al sistema Fig. Nro. 26.



**Figura Nro. 26 (Mensaje Bienvenido al Sistema)**

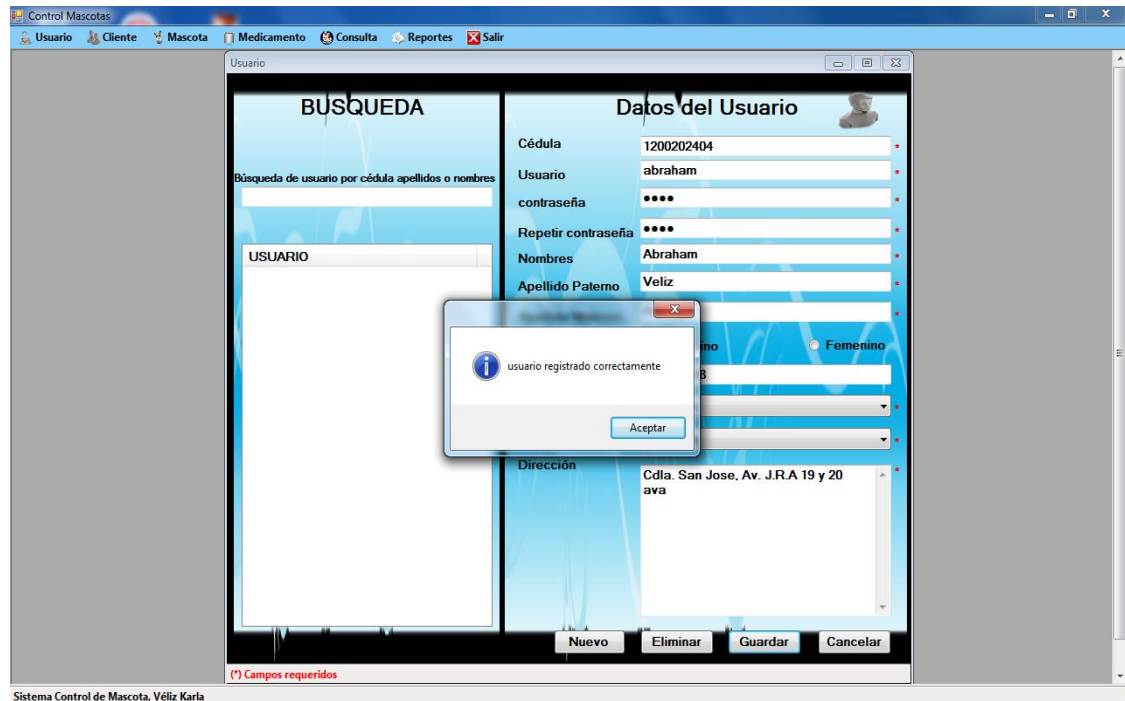
Al hacer clic en Aceptar de la Fig. Nro. 26 se visualizará las opciones del sistema para su manipulación: Usuario, Clientes, Mascotas, Medicamentos, Consulta, Reportes y Salir. En el Menu Usuario se encuentra las opciones de Nuevo, Cambiar Contraseña y Cerrar Sesión.

## **MENÚ USUARIO**



**Figura Nro. 27 (Menú Usuario)**

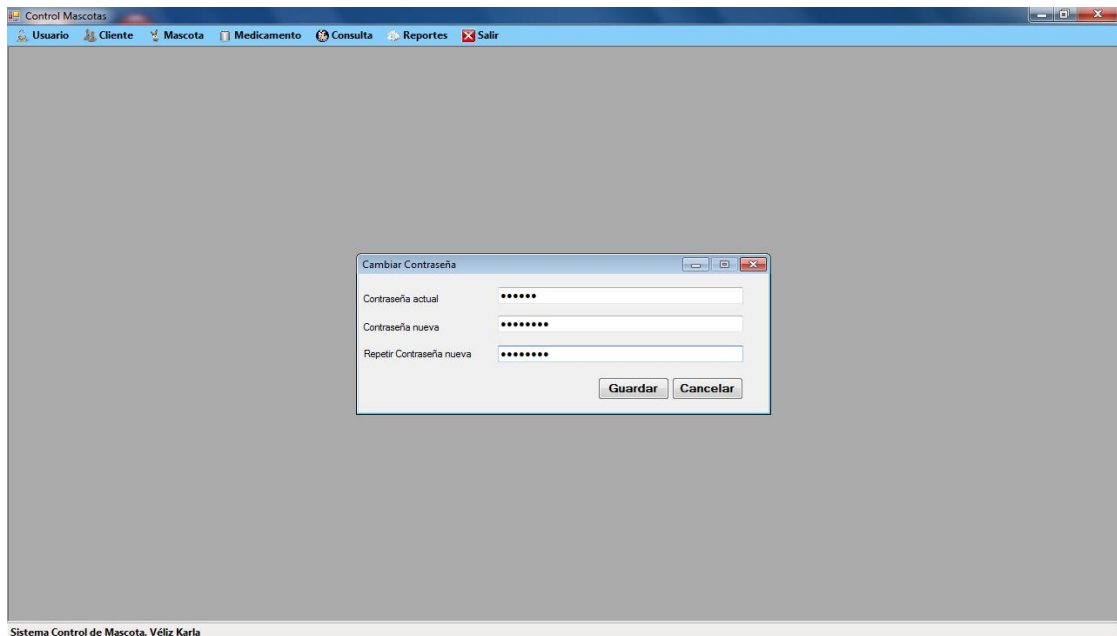
Al seleccionar la opción Nuevo del Menú Usuario Fig. Nro. 28 aparece la siguiente ventana que permitirá ingresar los datos respectivos de un usuario como: cédula, usuario, contraseña, apellidos y nombres, dirección. También se realiza la búsqueda de los usuarios existentes y poder modificar los datos.



**Figura Nro. 28 (Opción Nuevo)**

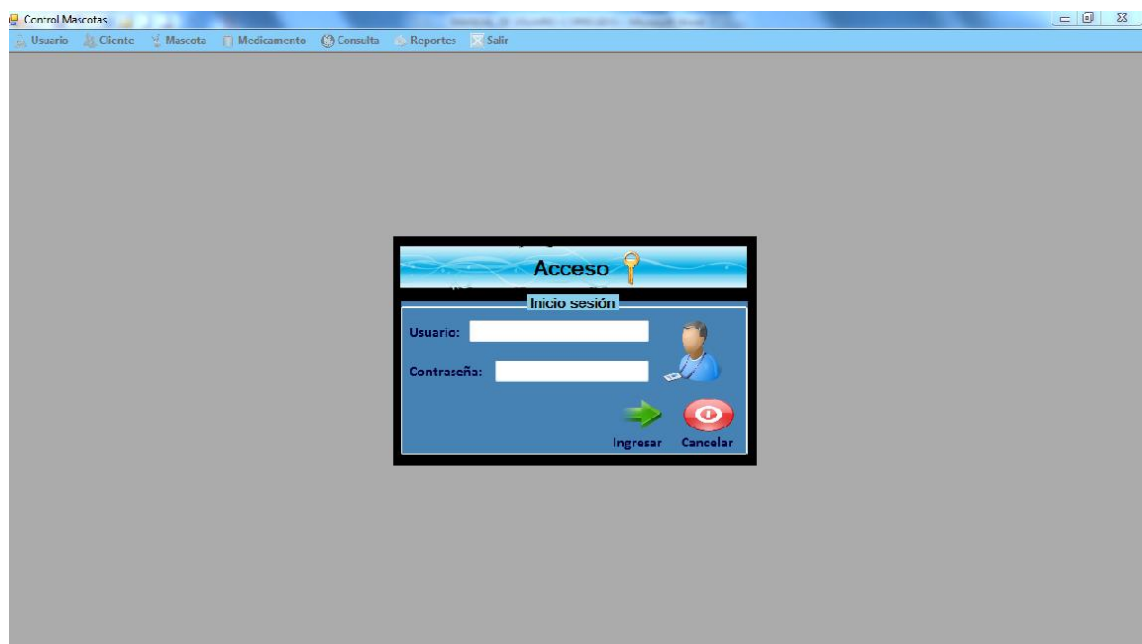
Al seleccionar el botón Guardar se generará un mensaje informando que el registro del nuevo usuario se ha realizado correctamente como se puede ver en la figura Nro. 28.

Al dar clic en la opción cambiar Contraseña del Menú Usuario se generará una nueva ventana para cambiar la contraseña del usuario que ha iniciado la sesión. Esta ventana solicitará la contraseña actual y la nueva contraseña.



**Figura Nro. 29 (Cambiar contraseña del menú Usuario)**

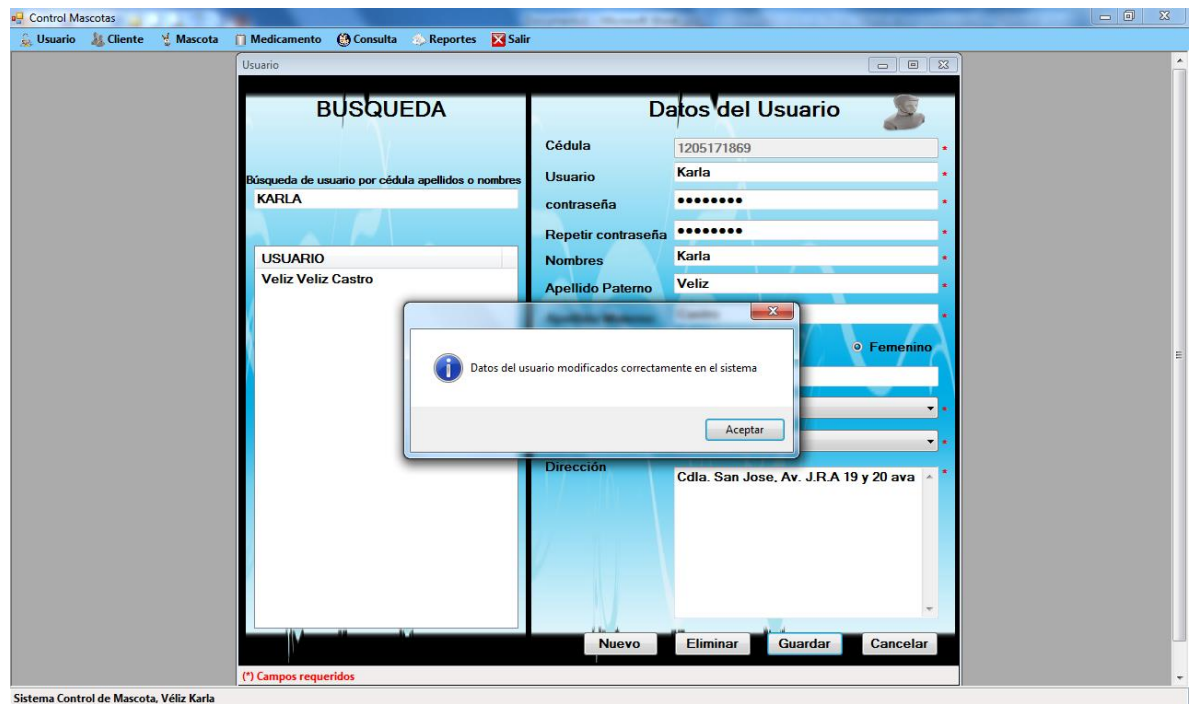
Al dar clic en cerrar sesión aparece la pantalla de inicio de sesión Fig. Nro. 30 se selecciona cancelar para salir del sistema o se podrá ingresar al sistema con otro usuario.



**Figura Nro. 30 (Inicio de Sesión)**

## MENU CLIENTE

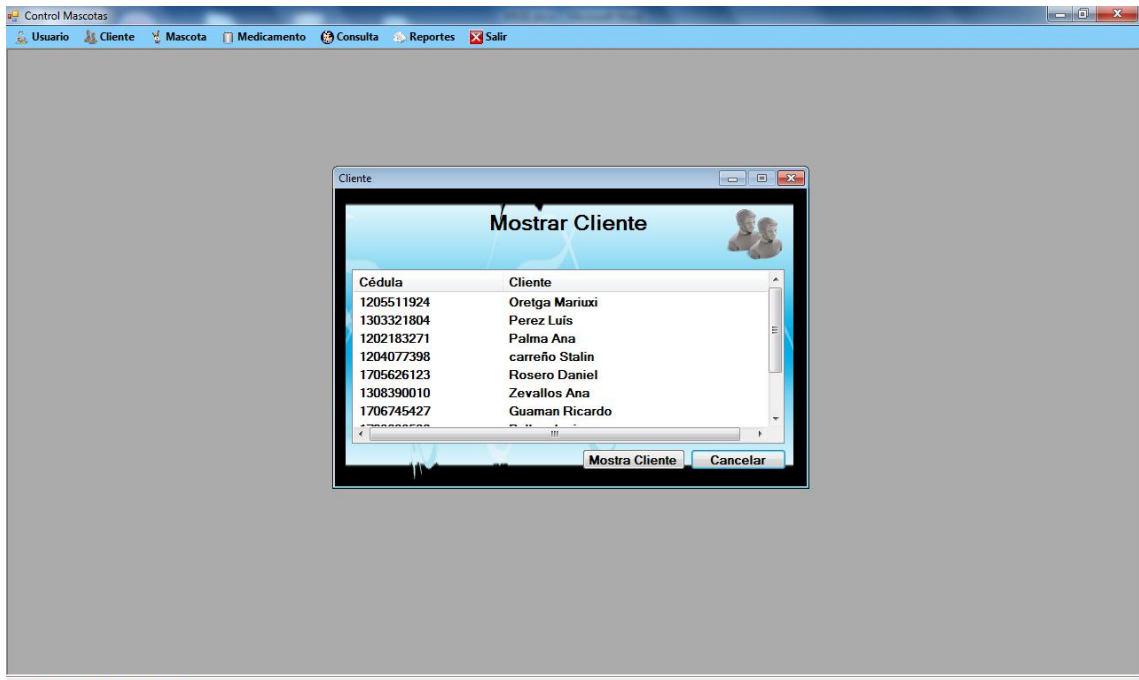
La opción Nuevo del Menú Cliente permite registrar un cliente, como se visualiza en la siguiente pantalla se digita datos necesarios como cédula, nombres y apellidos, también modificar los datos si el usuario lo requiere.



**Figura Nro. 31 (Menú Cliente)**

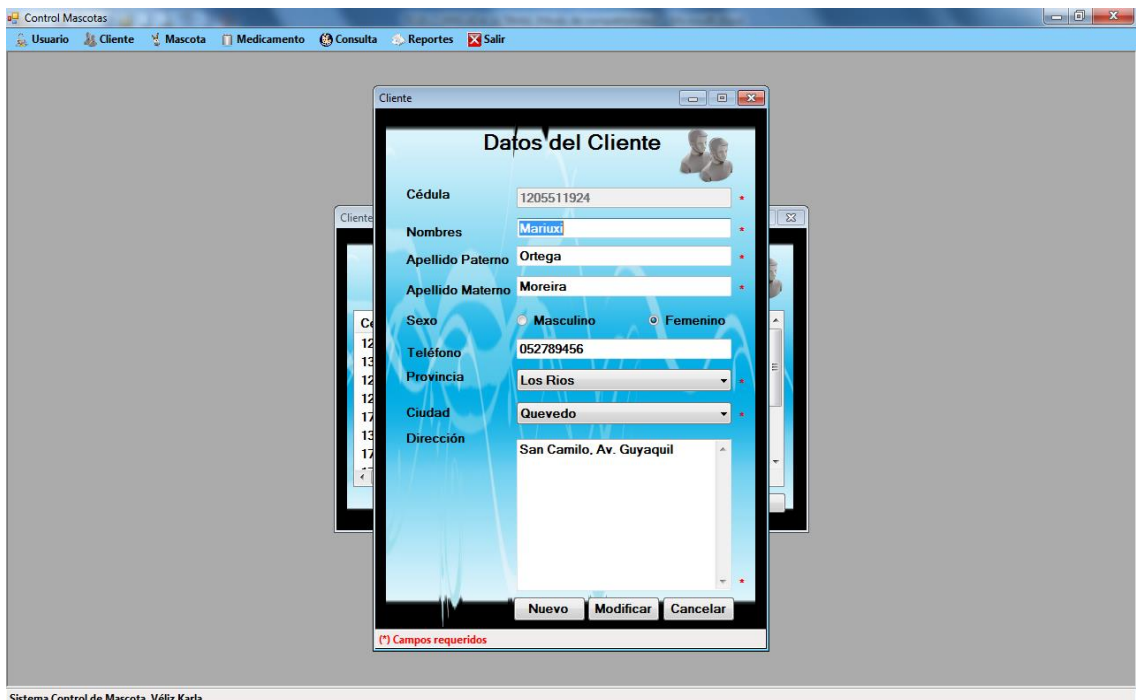
Al seleccionar el botón Guardar Fig. Nro. 31 presentará un mensaje informando al usuario el correcto registro del cliente en el sistema, tal como se observa en la figura anterior.

La opción Mostrar en el Menú Cliente Fig. Nro. 32 genera una nueva interfaz que permitirá visualizar la información de los clientes como cédula y apellidos.



**Figura Nro. 32 (Mostrar Cliente)**

Para observar los datos de un usuario Fig. Nro. 32 se deberá seleccionar el usuario que se desea ver los datos y a continuación presionar clic en Mostrar Cliente para poder visualizar todos los datos que corresponden a dicho usuario como son: Cédula, Nombres, Apellidos y Dirección, también permite modificar los datos. Fig. Nro. 33.



**Figura Nro. 33 (Datos del Cliente)**

## MENU MASCOTA

La opción Nueva del Menú Mascota generará una nueva interfaz que permitirá registrar los datos de una nueva mascota como: Dueño, Tipo de Mascota, Nombre, Sexo, Fecha de Nacimiento, Observación, como se indica en la Fig. Nro. 34.



**Figura Nro. 34 (Registro de datos de Mascotas)**

En el botón guardar generará un mensaje informando que la mascota se ha registrado correctamente como indica la figura anterior.

Al presionar clic en la opción Mostrar del Menú Mascota se generará una nueva interfaz que permitirá ver los datos de la(s) mascota(s) que tiene el cliente.



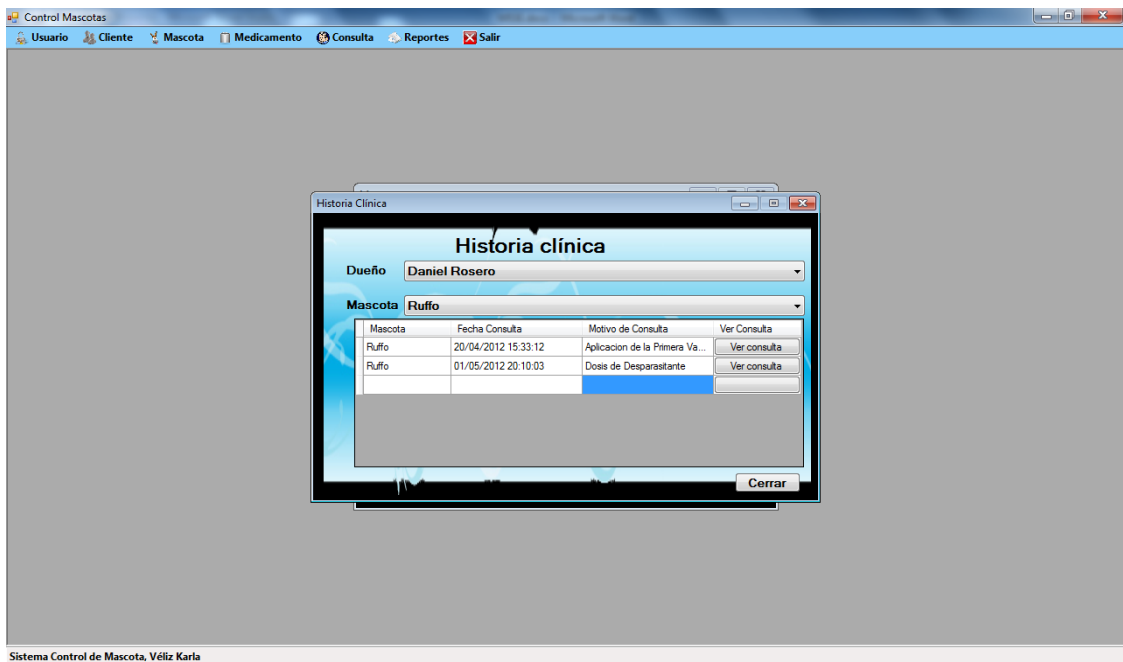
**Figura Nro. 35 (Mostrar Mascota)**

Al seleccionar la mascota y dar clic en el botón Ver Mascota se podrá ver los datos registrados y también se podrá modificar los datos en la Fig. Nro. 36.



**Figura Nro. 36 (Mascotas Registradas)**

Al presionar en el botón Ver Historial aparecerá una nueva ventana que permite ver la Historia Clínica de una mascota presentando en primera instancia datos de interés como fecha y motivo de consulta respectivos a la mascota seleccionada en la Fig. Nro. 37.



**Figura Nro. 37 (Historia Clínica de la Mascota)**

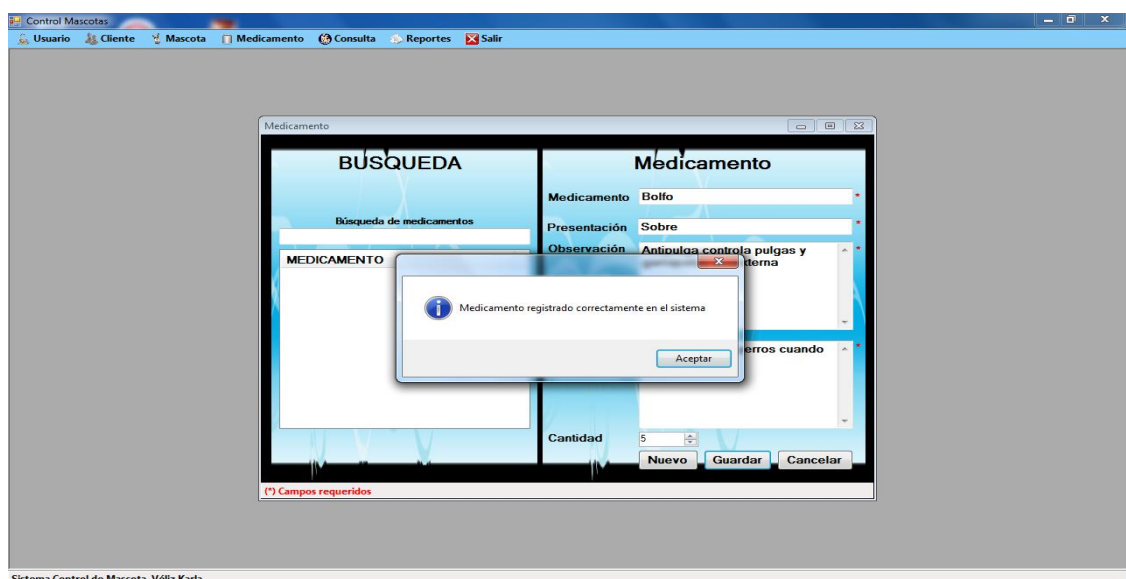
En el menú Mascota en la opción Nuevo Tipo al presionar clic permitirá registrar un Nuevo Tipo de mascota digitando datos como: Tipo y Observación, para dar paso al nuevo registro en el sistema como se lo indica en la Fig. 38.



**Figura Nro. 38 (Registrar Tipo de Mascota)**

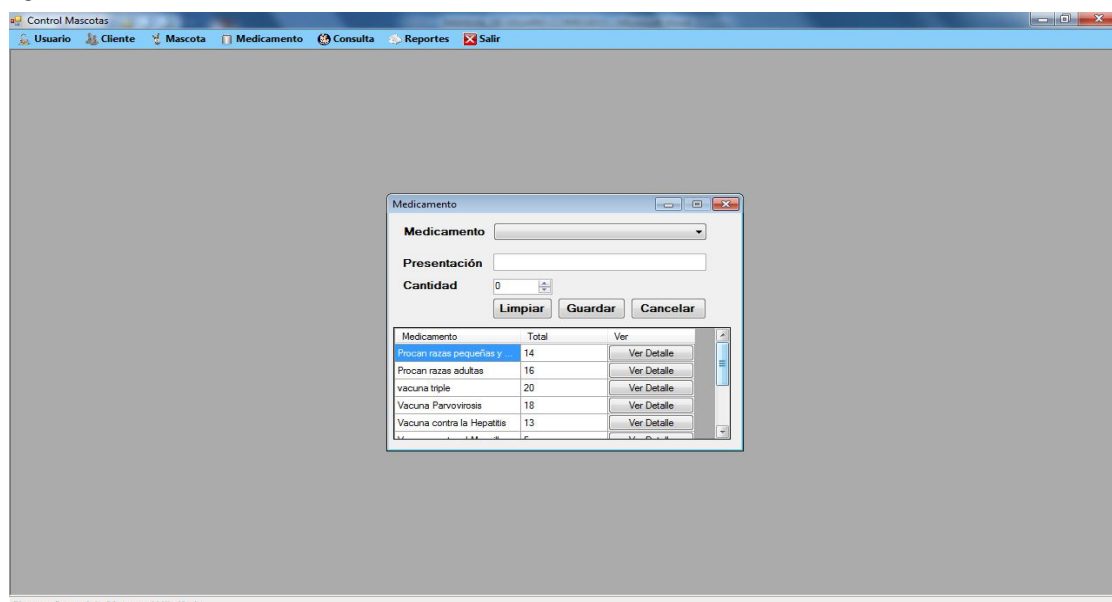
## MENU MEDICAMENTOS

En el menú Medicamentos se encuentra la opción Nuevo, al presionar se genera una nueva interfaz para llenar los campos Medicamento, Observación y Descripción y se puede ingresar un nuevo registro en el sistema. También esta opción permite modificar los datos. Ver Fig. Nro. 39.



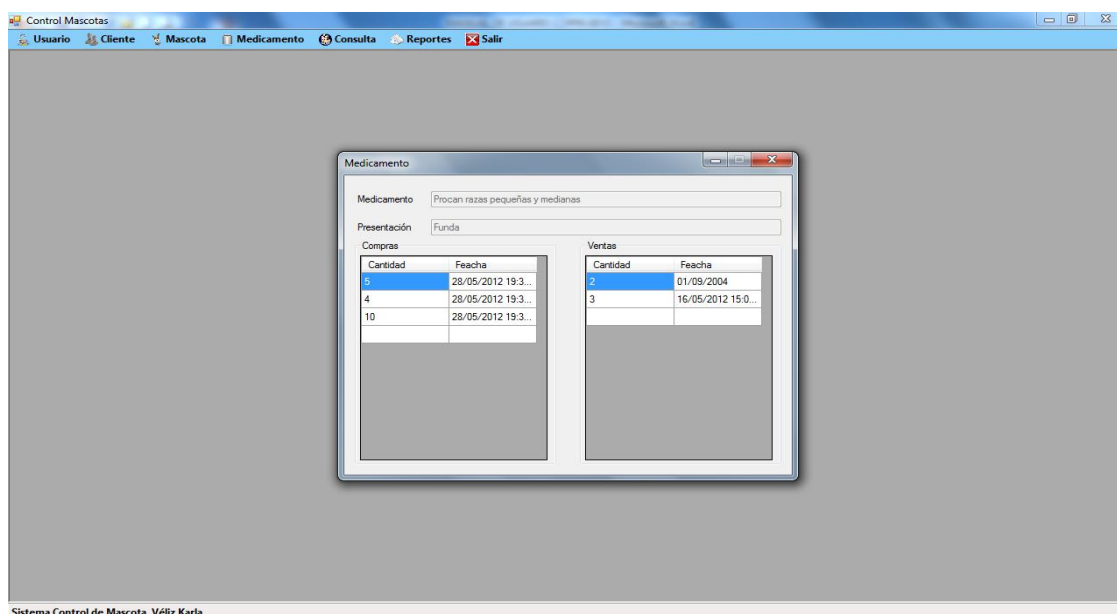
**Figura Nro. 39 (Registro de Medicamentos)**

En el Menú Medicamentos se encuentra la opción comprar, el listado de los medicamentos existentes en la empresa con sus respectivos nombres, presentación, cantidad, realizar compras de mas medicamentos. Ver Fig. Nro. 40.



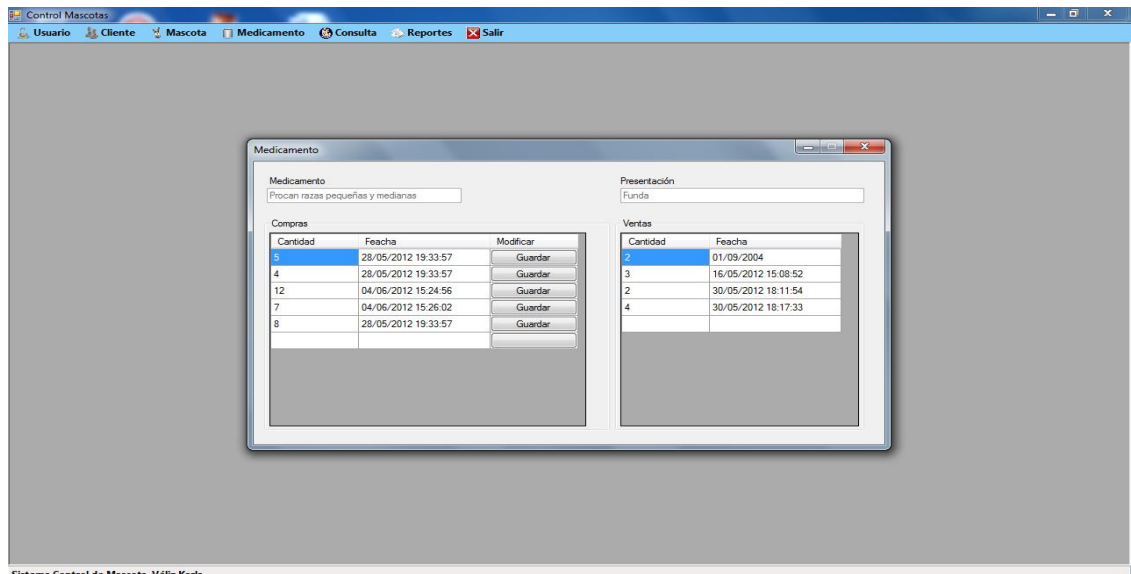
**Figura Nro. 40 (Compra de Medicamentos)**

Al presionar clic en medicamento del listado existente a lado se encuentra la columna Ver Detalle dar clic y aparece otra pantalla donde se encuentra el nombre del medicamento, presentación, Compras y Ventas, ahí se detalla la cantidad de compras y ventas con sus respectivas fechas. Ver Fig. Nro. 41.



**Figura Nro. 41 (Detalle de Compra y Venta de Medicamentos)**

Al presionar clic en los Medicamentos del listado existente, Fig. Nro. 42 a lado se encuentra la columna Ver Detalle hacer clic y aparecerá otra pantalla donde se especifica el nombre del medicamento, presentación, Compras y Ventas, ahí se detalla la cantidad de compras y ventas con sus respectivas fechas, se podrá modificar la compra y se guardará respectivamente.



Sistema Control de Mascota, Véliz Karla

**Figura Nro. 42 (Modificar compra de Medicamento)**

## MENU CONSULTA

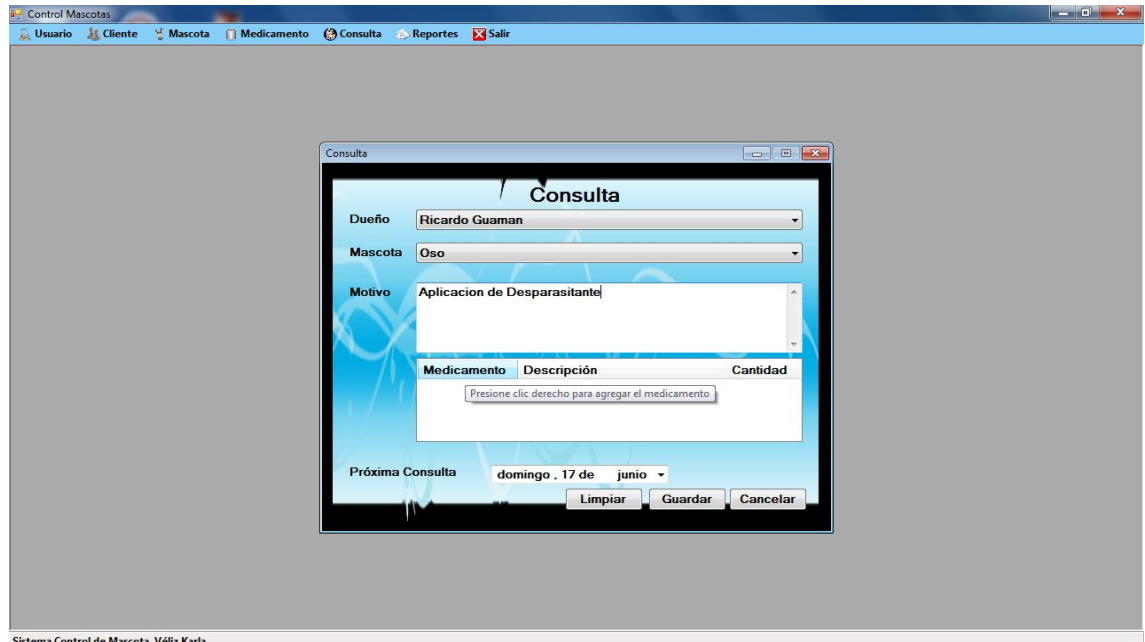
Al presionar clic en la opción Nueva del Menú Consulta Fig. Nro. 43 se presentará una nueva ventana que permite realizar una nueva consulta a una mascota seleccionando el dueño de la mascota se llenarán los datos como Motivo, Medicamento y Fecha de la próxima consulta.



Sistema Control de Mascota, Véliz Karla

**Figura Nro. 43 (Consulta de Mascota)**

En el campo Motivo del Menu Consulta Figura Nro. 44 se encuentra la opción de agregar medicamento, hacer clic en medicamento aparecerá un mensaje que dice presione clic derecho para agregar el medicamento.



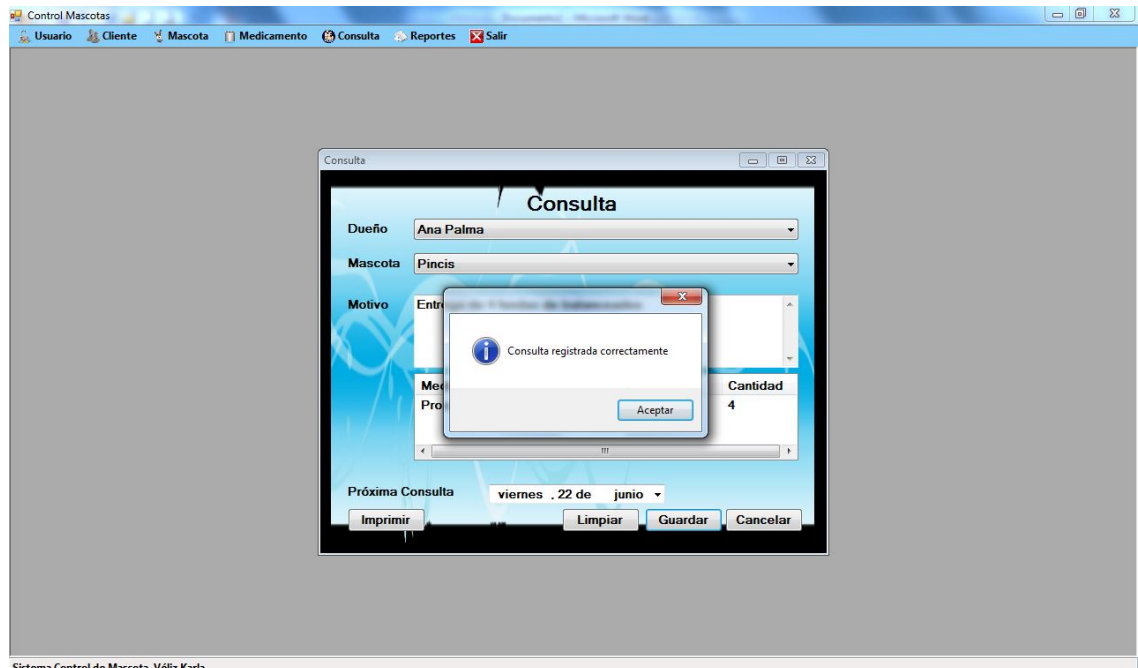
**Figura Nro. 44 (Agregar Medicamento)**

Presionar clic en la opción que dice en la Figura 44, aparece otra pantalla donde se escojerá el medicamento, llenar la dosis y cantidad del medicamento a utilizar.



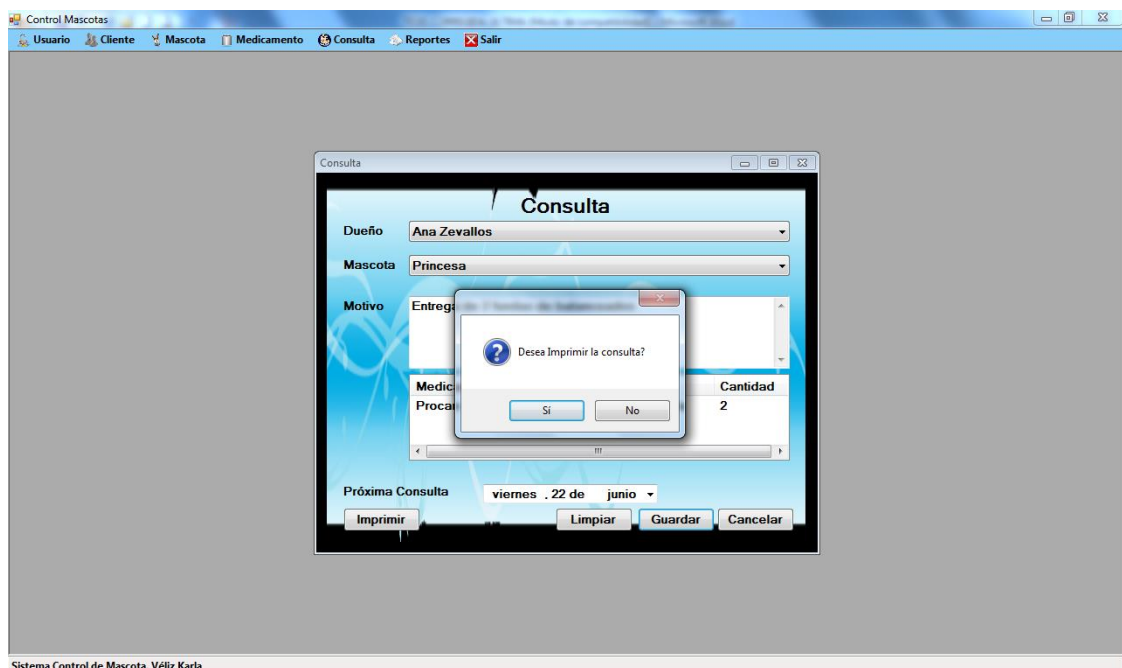
**Figura Nro. 45 (Listado de Medicamentos para agregar a la Consulta)**

Al seleccionar clic en el botón Guardar se presentará un mensaje informando el correcto registro de la consulta.



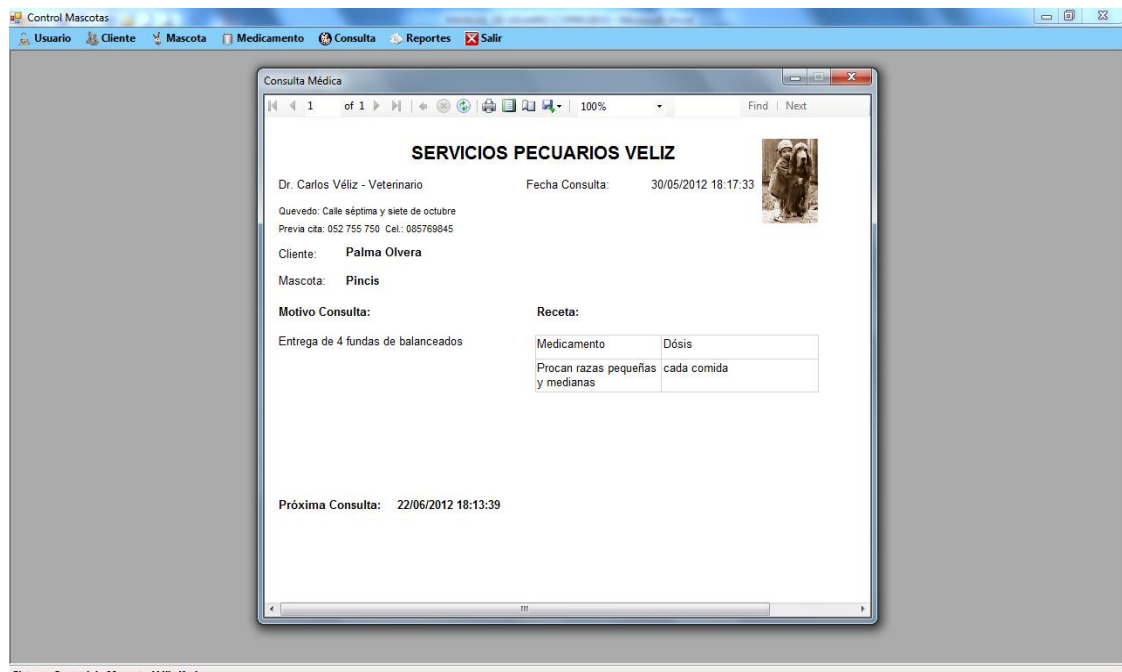
**Figura Nro. 46 (Consulta registrada correctamente)**

Presionar clic en aceptar aparecerá otra ventana donde dice si Desea Imprimir la Consulta. Figura Nro. 47.



**Figura Nro. 47 (Imprimir Consulta)**

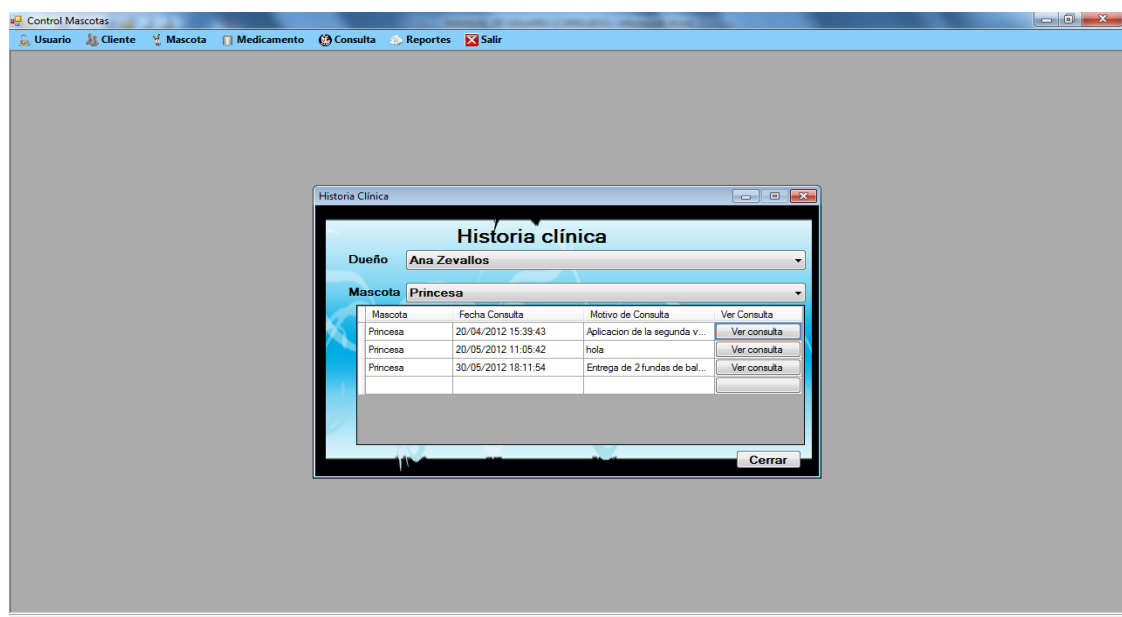
Presionar clic en si y aparece la Consulta Figura Nro. 48 donde se detalla el nombre de la empresa, propietario, dirección, fecha de la consulta, nombre del cliente y la mascota, motivo de la consulta, la receta y fecha de la próxima consulta.



Sistema Control de Mascota. Véliz Karla

**Figura Nro. 48 (Impresión de Consulta)**

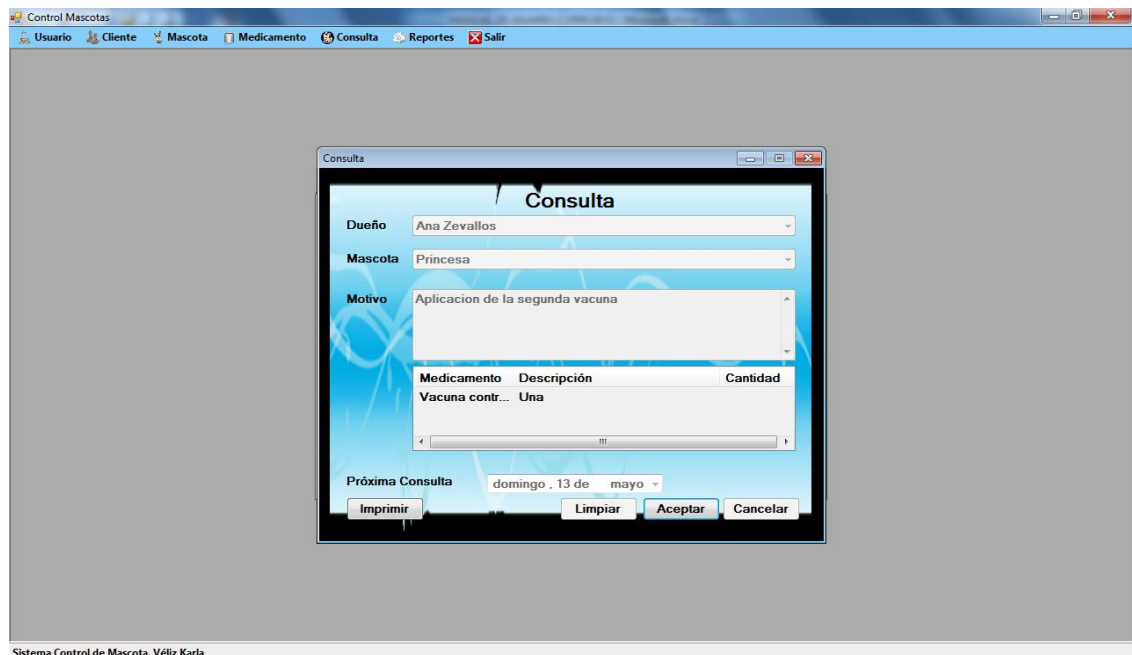
En la opción Mostrar del Menú Consulta Figura Nro. 49 se podrá visualizar la ventana Historia Clínica seleccionar el Dueño y la Mascota y aparecerá el detalle de la mascota como fecha de la consulta, motivo de consulta y ver la consulta.



Sistema Control de Mascota. Véliz Karla

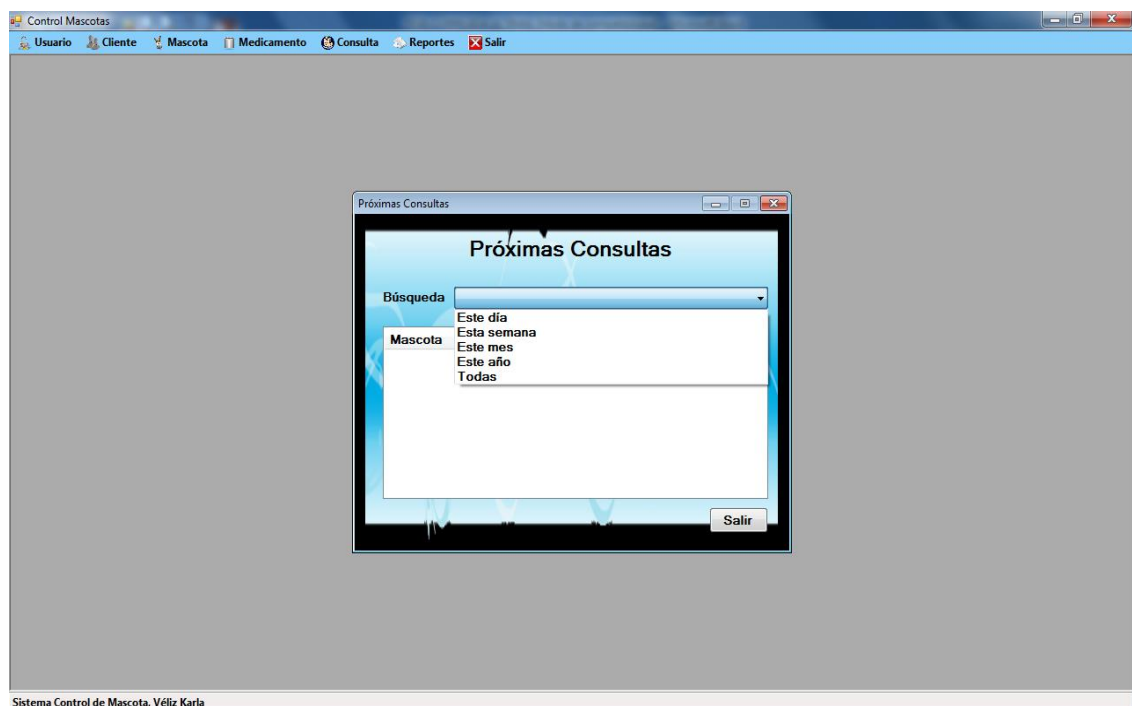
**Figura Nro. 49 (Historia Clínica)**

Al seleccionar Ver Consulta aparecerá la interfaz realizada con sus respectivos datos. Figura Nro. 50.

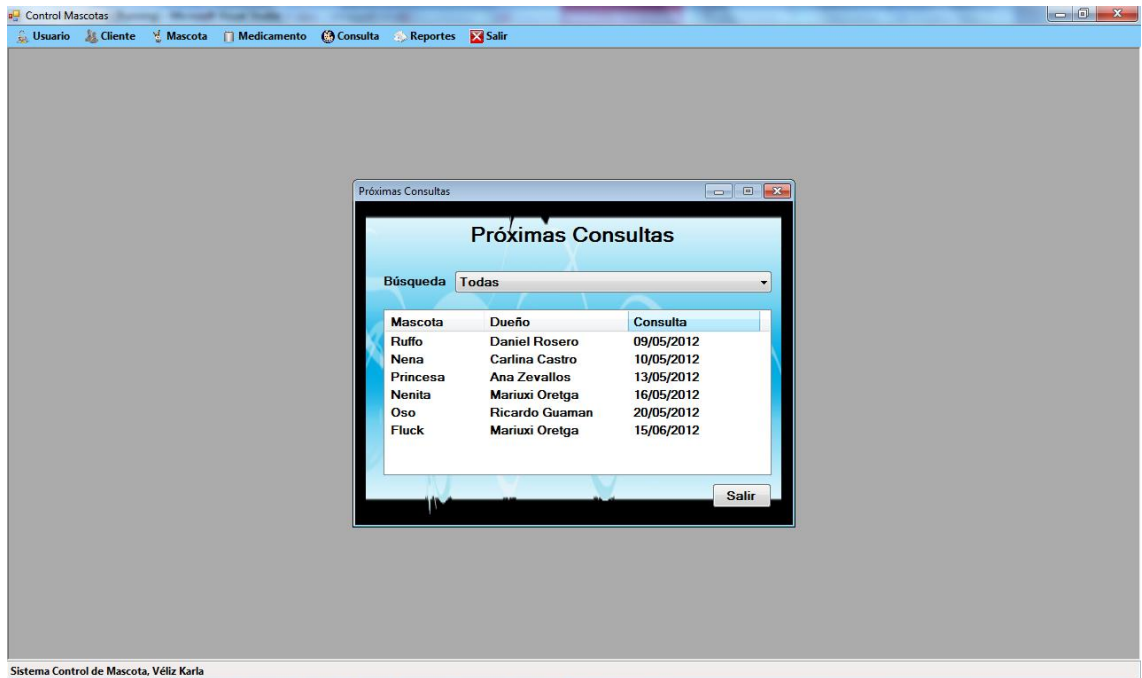


**Figura Nro. 50 (Consulta)**

En la opción Próximas del Menú Consulta Figura Nro. 51 al seleccionar clic aparecerá una ventana donde se podrá ver las consultas próximas clasificadas por: Este Día, Esta Semana, Este Mes, Este Año, Todas. Figura Nro. 51.



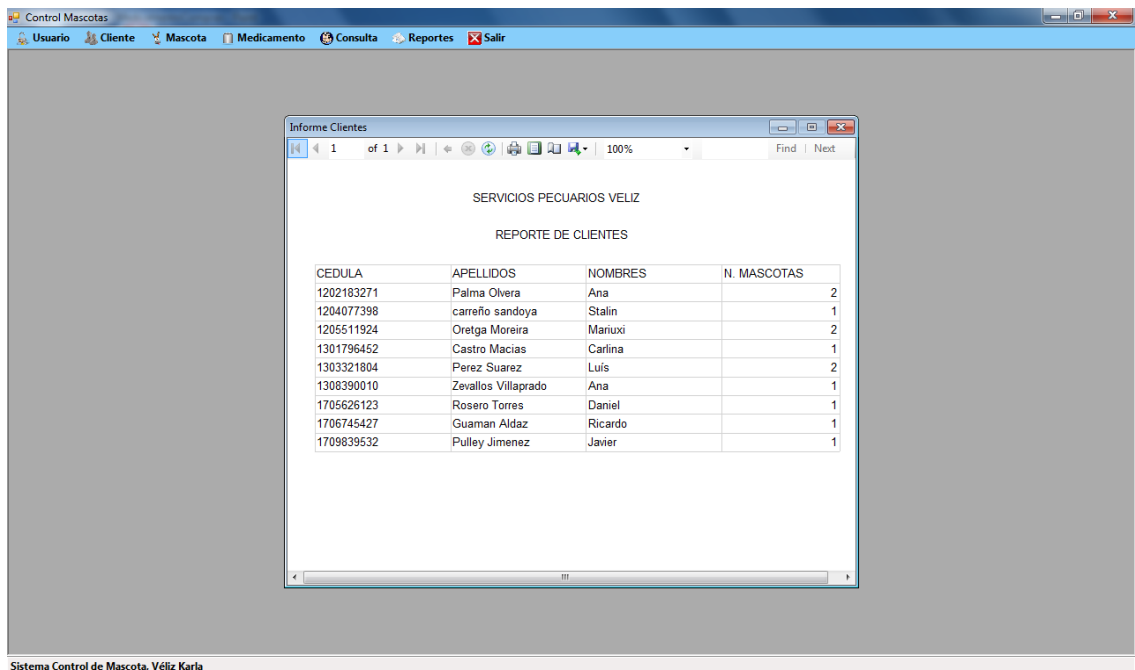
**Figura Nro. 51 (Próxima Consultas)**



**Figura Nro. 52 (Todas las Consultas)**

## MENU REPORTES

En la opción Reportes se podrá observar las ventanas de Reportes de Clientes, Mascotas y Medicamentos. Ver figura Nro. 53.



**Figura Nro. 53 (Reporte de Clientes)**

Control Mascotas

Usuario Cliente Mascota Medicamento Consulta Reportes Salir

Informe Mascotas

Ciente: -TODAS-

1 of 1

100%

Find | Next

**SERVICIOS PECUARIOS VELIZ**

**INFORME DE MASCOTAS**

Ciente	Mascota	Tipo de Mascota	Fecha Nacimiento
Oretga	Fluck	Canino	13/11/2011 17:11:02
Perez	Lili	Canino	02/01/2012 9:51:55
Oretga	Nenita	Canino	16/01/2012 17:23:24
Palma	Pincis	Canino	04/09/2004 0:00:00
Perez	Lulu	Canino	17/09/2004 0:00:00
carreño	Guille	Canino	05/01/2012 17:36:09
Rosero	Ruffo	Canino	20/03/2012 15:17:13
Zevallos	Princesa	Canino	13/03/2012 15:37:03
Guaman	Oso	Canino	20/01/2012 16:15:22

Sistema Control de Mascota, Véliz Karla

Figura Nro. 54 (Reporte de Mascotas)

Control Mascotas

Usuario Cliente Mascota Medicamento Consulta Reportes Salir

Informe Medicamento

1 of 1

100%

Find | Next

**SERVICIOS PECUARIOS VELIZ**

**INFORME DE MEDICAMENTOS**

MEDICAMENTO	PRESENTACION	TOTAL
Vacuna Antirrabica	frasco	18
Vacuna contra el Moquillo	frasco	6
Vacuna contra la Hepatitis	frasco	3
Vacuna contra la Sarna	frasco	7
Vacuna Parvovirus	frasco	8
vacuna triple	frasco	11
Procan razas adultas	Funda	24
Procan razas pequeñas y medianas	Funda	23
Desparasitantes	pastillas	42

Sistema Control de Mascota, Véliz Karla

Figura Nro. 55 (Reporte de Medicamentos)

## MANUAL DE CONEXIÓN UTILIZANDO EL ROUTER Y EL DISPOSITIVO MÓVIL (POCKET PC)

Para conectar los equipos informáticos a la red interna, únicamente será preciso tener un router inalámbrico que le proporcione una dirección IP.



Figura Nro. 56 (Router)

### Conexión a la Red mediante el Ordenador

Comprobar que la tarjeta inalámbrica del ordenador está activada pulsando sobre el botón Inicio --> Panel de control --> Centro de redes y recursos compartidos --> Cambiar configuración del Adaptador donde el ícono de Conexión de red inalámbrica no debe estar desactivado. En caso de estar desactivado debe dar clic derecho sobre el ícono y dar clic en la opción Activar.

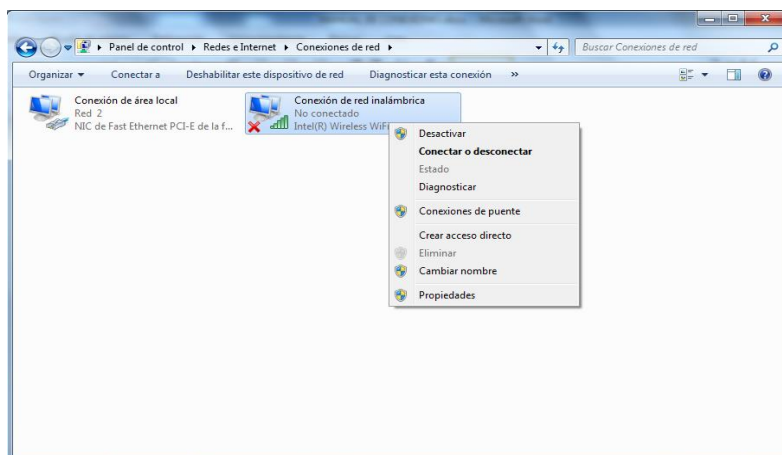
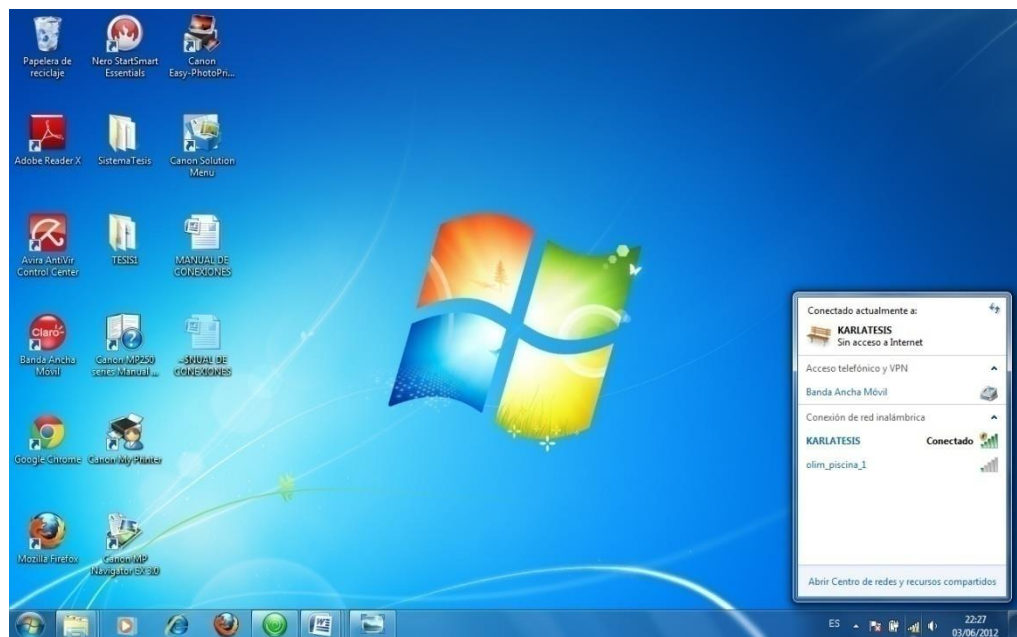


Figura Nro. 57 (Conexión a la Red-Ordenador)

Una vez conectado el router nuestro computador y dispositivo móvil detectarán automáticamente la red inalámbrica y se procederá a conectar a ella siguiendo los pasos a continuación:

En la barra de tareas en la parte derecha el sistema operativo Windows Seven detectará automáticamente la red inalámbrica permitiendo al usuario poder conectarse sin inconvenientes con solo un clic.

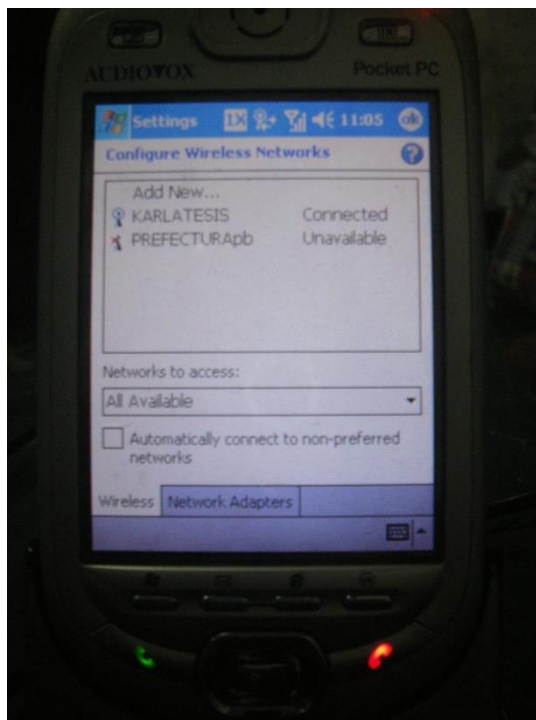
Se debe seleccionar la red inalámbrica con la que se desea establecer la conexión. Tras hacerlo, en el caso de contar con una conexión protegida por contraseña (caso habitual), deberá digitar la clave de acceso correspondiente y pulsar sobre el botón conectar para establecer la conexión.



**Figura Nro. 58 (Conexión Ordenador y Pocket Pc)**

### **Conexión a la Red mediante el Dispositivo Móvil**

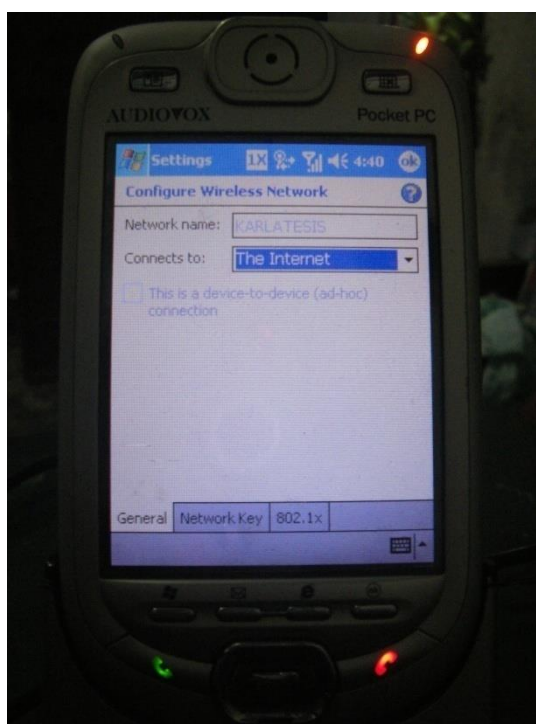
Cuando el dispositivo móvil detecte la red inalámbrica automáticamente aparecerá en la lista de conexiones disponibles y se debe seleccionar una red a conectarse.



**Figura Nro. 59 (Conexión a la Red mediante la Pocket Pc)**

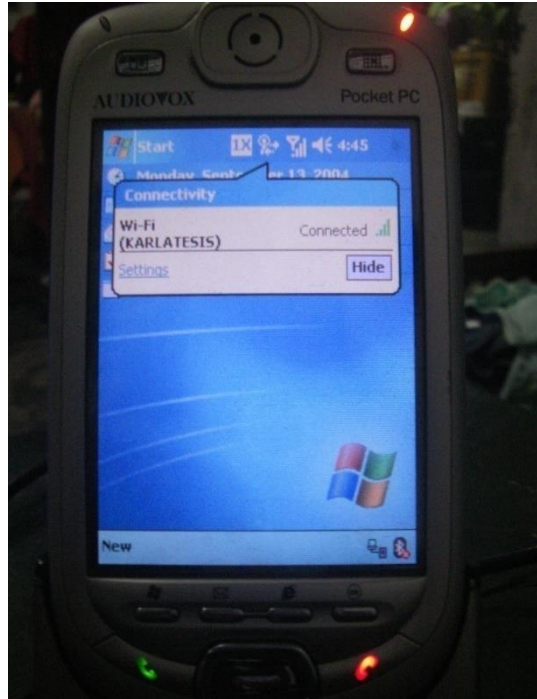
Si el dispositivo no detecta la red inalámbrica, revisar las redes disponibles siguiendo los pasos a continuación:

Pulsar sobre el botón Inicio --> Sttings --> Conections. En esta pantalla aparecerán las redes disponibles donde debe pulsar sobre la red a la que desea conectarse y elegir la opción Conectt.



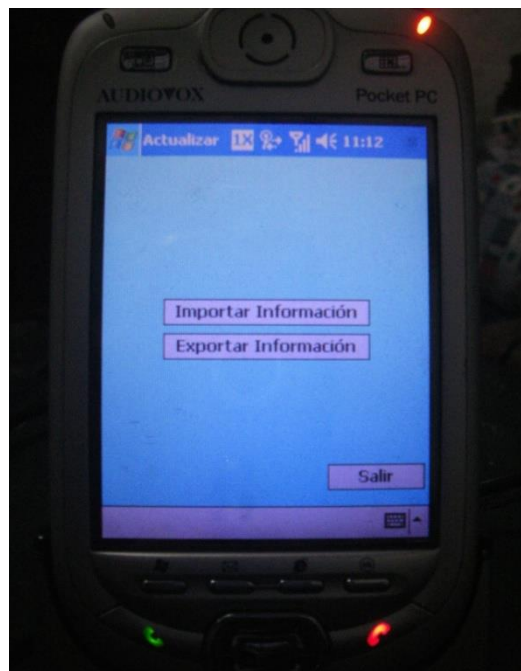
**Figura Nro. 60 (Pocket Pc conectado al Ordenador)**

Para verificar que nuestro dispositivo móvil está conectado a la red deberá pulsar sobre el ícono de conexiones ubicado en la barra de tareas del sistema operativo donde procederá a informar a que red se encuentra conectada.



**Figura Nro. 61 (Red Conectada)**

Para poder realizar las transacciones correspondientes del sistema como lo es Importar Información y Exportar Información que son opciones del aplicativo móvil, deberán ambos equipos (Servidor y Dispositivo móvil) estar en la misma red.

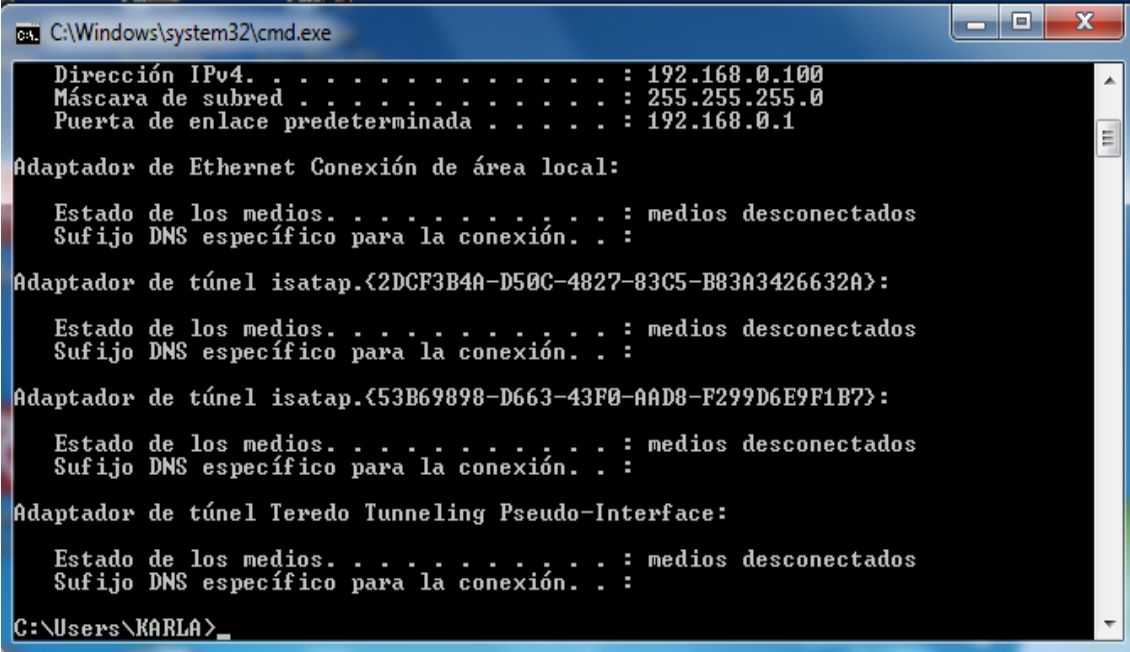


**Figura Nro. 62 (Importacion y Exportacion de datos Pocket Pc-Servidor)**

Para poder verificar que ambos equipos pertenezcan a la misma red debe seguir los pasos que se presentan a continuación.

En las ventanas de comandos a la cuál se llega si pulsa el un botón Inicio --> todos los programas --> Accesorios --> Símbolo del sistema, se ejecutará los siguientes comandos.

Ipconfig.- Este comando muestra la información relativa a los parámetros de nuestra configuración **IP** actual como muestra la Figura Nro. 63.



```
C:\Windows\system32\cmd.exe
Dirección IPv4. . . . . : 192.168.0.100
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.0.1

Adaptador de Ethernet Conexión de área local:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de túnel isatap.{2DCF3B4A-D50C-4827-83C5-B83A3426632A}:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de túnel isatap.{53B69898-D663-43F0-AAD8-F299D6E9F1B7}:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de túnel Teredo Tunneling Pseudo-Interface:

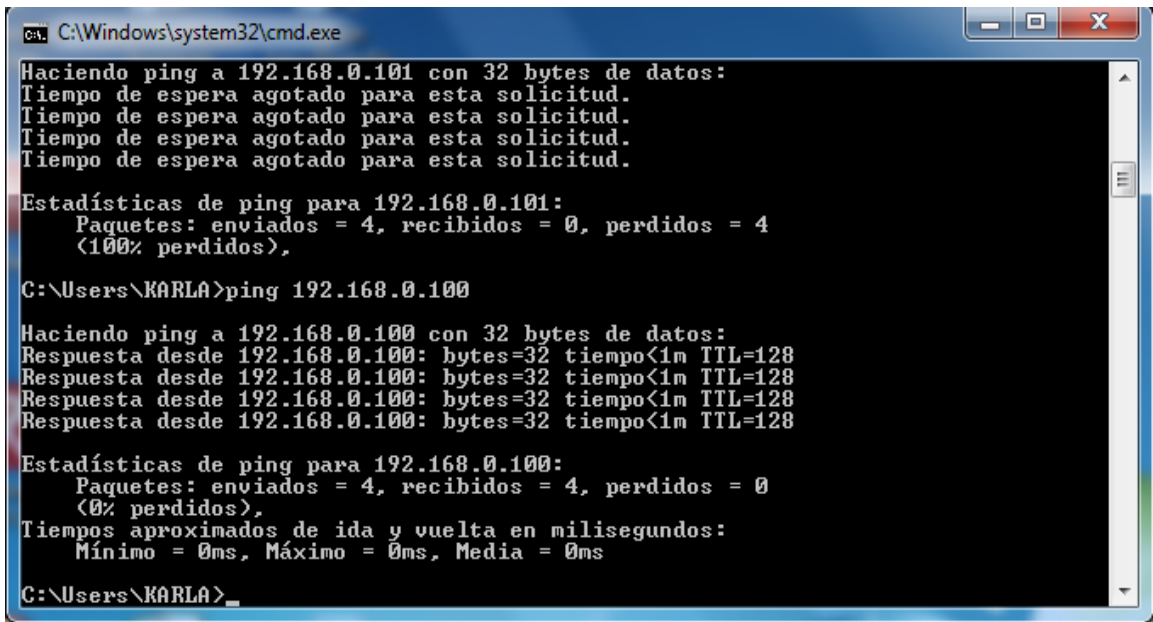
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

C:\Users\KARLA>
```

**Figura Nro. 63 (Verificación de equipos conectados)**

**Ping.-** Cuando hace ping a un equipo (al aplicativo móvil) o a una dirección IP lo que hace el sistema es enviar a esa dirección una serie de paquetes (normalmente cuatro) de un tamaño total de 64 bytes (salvo que se modifique) y queda en espera del reenvío de estos (eco), por lo que se utiliza para medir la latencia o tiempo que tardan en comunicarse dos puntos remotos.

Una de las ventajas de ejecutar este comando es que los paquetes se envían atacando directamente la IP a la que se dirige el ping, lo que hace que una de sus utilidades es comprobar la conectividad de nuestro Pc con el dispositivo móvil.



```
ca: C:\Windows\system32\cmd.exe
Haciendo ping a 192.168.0.101 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.0.101:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
      (100% perdidos),

C:\Users\KARLA>ping 192.168.0.100

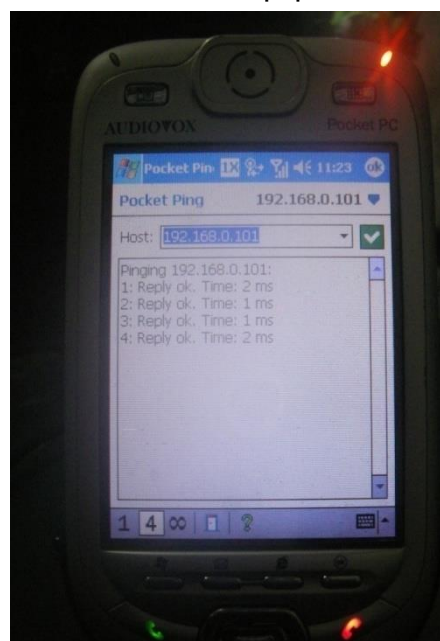
Haciendo ping a 192.168.0.100 con 32 bytes de datos:
Respuesta desde 192.168.0.100: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.100: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.100: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.100: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.0.100:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
      (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
      Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\KARLA>
```

**Figura Nro. 64 (Servidor conectado al Pocket Pc)**

Igualmente se realizan estos pasos en el dispositivo móvil, la diferencia es que el sistema operativo Windows Mobile que la Pocket PC no tiene la ventana de comando, razón por la cual hay que descargar desde internet un aplicativo que permita verificar la conectividad a otros equipos.



**Figura Nro. 65 ( Pocket Pc conectado al Servidor)**

Al realizar estas configuraciones el sistema móvil se comunicará sin inconvenientes con el servidor y logrará realizar las transacciones respectivas de Importar y Exportar información con éxito.

## **Manual Usuario para el Sistema de Control de Mascotas en el Dispositivo Móvil**

### **Ingreso al Sistema**

Como acceder al Sistema de Información con Aplicación Móvil para el Control Administrativo de la empresa SERVICIOS PECUARIOS VELIZ.

A continuación aparecerá la pantalla del Sistema que permite ingresar al sistema.



**Figura Nro. 66 ( Ingreso al Sistema)**

**Usuario:** Ingrese aquí el nombre del usuario para acceder al sistema

**Clave:** Ingrese aquí la contraseña correspondiente al usuario ingresado para acceder al sistema

Al ingresar el usuario y contraseña correctos se visualizará las opciones del sistema para su manipulación que son: Clientes (Nuevo, Mostrar) y Macostas (Nuevo, Mostrar, Consultas, Agregar Medicamentos).



**Figura Nro. 67 (Sistema de Control de Mascota)**

**Próximas Consultas:** Visualiza las siguientes consultas a realizar de cada una de las mascotas

**Registrar Consulta:** Permite almacenar la consulta de la mascota que ha sido tratada

**Medicamentos:** Permite registrar nuevos Medicamentos

**Salir:** Salir completamente del sistema

### **Menú Cliente**

**Nuevo:** Permite crear, Editar y Eliminar clientes

**Mostrar:** Visualiza los clientes registrados en la base de datos.

### **Menú Mascotas**

**Nuevo:** Permite crear, Editar y Eliminar Mascotas

**Mostrar:** Visualiza mascotas que han sido registradas en la base de datos.

**Tipo:** Permite crear nuevo tipo de mascota (Ejemplo: Canino, Felino)



**Figura Nro. 68 (Datos del Cliente)**

**Nuevo:** Visualiza la ventana para registrar nuevo cliente

**Salir:** salir de la ventana de clientes para ir al menu principal

**Ver Cliente:** Visualiza los datos toda la informacion del cliente seleccionado en una nueva ventana

**Mostrar Mascotas:** Visualiza las mascotas por cliente en una nueva ventana

## Mascotas

En la siguiente ventana se visualiza las mascotas de un cliente seleccionado con las opciones de nueva mascota, ver consulta y salir.



**Figura Nro. 69 (Datos de la Mascota)**

**Nueva mascota:** Visualiza una nueva ventana que permite registrar nuevas mascotas del cliente seleccionado

**Ver Mascota:** Visualiza toda la información de la mascota seleccionada

**Historia Clínica:** Visualiza en una nueva ventana el historial clínico de una mascota seleccionada

**Nueva Consulta:** Permite registrar los datos de la mascota seleccionada para un nuevo tratamiento

**Salir:** salir de la ventana de clientes para ir al menu principal

### **Cientes**

Se puede Visualizar los datos de la mascota como es su dueño, nombre raza, tipo, fecha de nacimiento, sexo y observación.



The image shows a screenshot of a mobile application interface for entering pet data. The form is titled "Nueva Mascota" and includes the following fields and options:

- Dueño:** A dropdown menu with the value "0928604313".
- Nombre:** A text input field containing "Pirulis".
- Raza:** A text input field containing "Sakicha".
- Tipo:** A text input field containing "Canino".
- Fecha Nac.:** A date picker showing "15/07/11".
- Sexo:** Radio buttons for "Macho" (selected) and "Hembra".
- Observación:** A text area containing "Mascota en Observación Normal".

At the bottom of the form are two buttons: "Guardar" and "Cancelar".

**Figura Nro. 70 (Datos del la Mascota)**

**Cedula:** Ingrese aquí la cedula del cliente

**Nombre:** Ingrese aquí los nombres del cliente

**Apellido Paterno:** Ingrese aquí el apellido paterno del cliente

**Apellido Materno:** Ingrese aquí el apellido materno del cliente

**Sexo:** seleccione el sexo del cliente

**Provincia:** Seleccione la provincia donde vive el cliente

**Ciudad:** Seleccione la ciudad donde vive el cliente

**Telefono:** Ingrese aquí el teléfono del cliente

**Provincia:** Ingrese aquí el celular del cliente

**Guardar:** almacena la información en el sistema

**Cancelar:** Cancela el ingreso y nos vamos a la ventana principal

### **Consultas**

Si se selecciona la opción mostrar Consulta se podran visualizar en una nueva ventana el historial y control de consultas que ha tenido la mascota que se ha seleccionado y a que dueño pertenece.



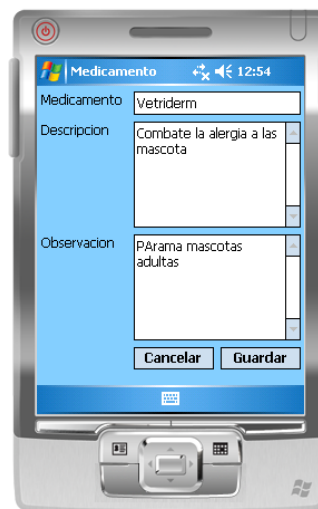
**Figura Nro. 71 (Control de Consulta de la Mascota)**

**Nueva Consulta:** Permite registrar nuevas consultas para fechas futuras de la mascota seleccionada

**Salir:** Salir de la ventana de clientes para ir al menu principal

## **Medicamentos**

Permite registrar medicamentos (nombre del medicamento, una descripción y una observación.)



**Figura Nro. 72 (Registrar Medicamentos)**

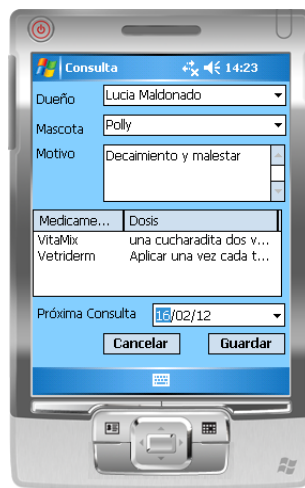
**Medicamento:** Ingrese aquí el nombre del medicamento a registrar

**Descripcion:** Ingrese aquí la descripción del medicamento

**Observacion:** Ingrese aquí una observación del medicamento

## **Consultas**

Permite registrar, seleccionar Dueño, Mascota motivo y fecha de la próxima consulta.



**Figura Nro. 73 (Registro de Consulta)**

**Dueño** : Seleccione el dueño de la mascota

**Mascota**: Seleccione la mascota del dueño

**Motivo**: Ingrese aquí motivo de consulta

**Proxima Consulta**: Ingrese aquí la fecha de la proxima consulta

**Guardar**: Almacena la información en el sistema

**Cancelar**: Cancela el ingreso y nos vamos a la ventana principal

### **Tipo de Mascota**

Permite registrar los tipos de mascotas(ejemplo aves,felinos,caninos).



**Figura Nro. 74 (Registro Tipo de Mascota)**

**Tipo de mascota:** Ingrese aquí el tipo de mascota

**Observacion:** Ingrese aquí la observacion del tipo de mascota ingresado

**Guardar:** Almacena la información en el sistema

**Cancelar:** Cancela el ingreso y nos vamos a la ventana principal

**MATRIZ DE DATOS DE LA EMPRESA SERVICIOS PECUARIOS VELIZ**

FECHA	ACTIVIDADES	TAREA	
		MANUAL	SISTEMA
01/05/2012	Ingreso de datos de tratamiento de mascotas para control sanitario	10"	4"
02/05/2012	Ingreso de datos de control sanitario (aplicación de vacunas)	12"	5"
03/05/2012	Visita domiciliaria (aplicación de desparasitantes)	9"	5"
04/05/2012	Ingreso de datos de mascotas nuevas para tratamiento	11"	5"
05/05/2012	Ingreso de datos de control sanitario (aplicación de vitaminas)	10"	5"
06/05/2012	Ingreso de datos de control sanitario (aplicación de vacunas anabólicas)	10"	5"
07/05/2012	Ingreso de datos de control sanitario (aplicación de refuerzos de vacunas)	11"	5"
08/05/2012	Ingreso de datos de tratamiento de mascotas para control sanitario	12"	3"
09/05/2012	Ingreso de datos de tratamiento de mascotas para control sanitario	9"	2"
10/05/2012	Ingreso de datos de control sanitario (aplicación de vitaminas)	12"	4"
11/05/2012	Ingreso de datos de control sanitario (aplicación de vacunas)	11"	3"
12/05/2012	Ingreso de datos de control sanitario (aplicación de desparasitantes)	13"	4"
13/05/2012	Ingreso de datos de control sanitario (aplicación de refuerzos de vacunas)	12"	4"
14/05/2012	Ingreso de datos de mascotas nuevas para tratamiento	9"	4"
15/05/2012	Ingreso de datos de control sanitario (aplicación de vacunas)	9"	2"
16/05/2012	Ingreso de datos de control sanitario (aplicación de vacunas)	10"	4"
17/05/2012	Ingreso de datos de control sanitario (aplicación de desparasitantes)	11"	5"
18/05/2012	Ingreso de datos de control sanitario (aplicación de desparasitantes)	13"	4"
19/05/2012	Visita domiciliaria (aplicación de vacunas)	9"	5"
20/05/2012	Visita domiciliaria (aplicación de vacunas)	10"	4"

21/05/2012	Ingreso de datos de mascotas nuevas para tratamiento	8"	3"
22/05/2012	Ingreso de datos de control sanitario (aplicación de vacunas anabólicas)	8"	4"
23/05/2012	Visita domiciliaria (aplicación de vacunas)	9"	3"
24/05/2012	Visita domiciliaria (aplicación de vitaminas y desparasitantes)	10"	5"
25/05/2012	Ingreso de datos de mascotas nuevas para tratamiento	10"	3"
26/05/2012	Ingreso de datos de control sanitario (aplicación de vacunas)	12"	6"
27/05/2012	Visita domiciliaria (aplicación de desparasitantes)	10"	4"
28/05/2012	Visita domiciliaria (aplicación de vitaminas y desparasitantes)	9"	4"
29/05/2012	Ingreso de datos de control sanitario (aplicación de refuerzos de vacunas)	10"	3"
30/05/2012	Ingreso de datos de control sanitario (aplicación de vacunas anabólicas)	9"	2"
31/05/2012	Visita domiciliaria (aplicación de vitaminas y desparasitantes)	11"	4"

# CODIFICACIÓN DEL SISTEMA DE INFORMACIÓN CON APLICACIÓN MÓVIL PARA EL CONTROL ADMINISTRATIVO DE LA EMPRESA SERVICIOS PECUARIOS VELIZ DEL CANTON QUEVEDO

## SISTEMA DE ESCRITORIO

### CAPA DE DATOS

Imports System.Data.SqlClient

Public Class Conexion

#Region "ATRIBUTOS"

Public Cnn As SqlConnection

Private Cmd As SqlCommand

Private Dreader As SqlDataReader

Private Transaccion As SqlTransaction

Private Transac As Boolean

Private MostrarMSj As Boolean

#End Region

#Region "CONSTRUCTOR"

Public Sub New()

Cnn = New SqlConnection("Data Source=KARLA-PC\SQLEXPRESS;Initial Catalog=Veterinaria; User ID=KarlaTesis; Password=12345")

End Sub

#End Region

#Region "METODOS"

Public Function BoolConectar() As Boolean

BoolConectar = False

Try

If Cnn.State = Data.ConnectionState.Closed Then

Cnn.Open()

End If

```

        BoolConectar = True
    Catch ex As SqlException
        If MostrarMSj Then    MessageBox.Show(ex.Message,    NS,
    MessageBoxButtons.OK,    MessageBoxIcon.Asterisk,
    MessageBoxDefaultButton.Button1)
    End Try
End Function

```

```

Public Function BoolCerrar() As Boolean
    BoolCerrar = False
    Try
        If Cnn.State = Data.ConnectionState.Open Then
            Cnn.Close()
            BoolCerrar = True
        End If
    Catch ex As Exception
        If MostrarMSj Then    MessageBox.Show(ex.Message,    NS,
    MessageBoxButtons.OK,    MessageBoxIcon.Asterisk,
    MessageBoxDefaultButton.Button1)
    End Try

End Function

```

```

Public Function BoolConsultaParametros(ByRef lector As SqlDataReader,
ByVal Procedimiento As String, ByVal VcParametros() As SqlParameter) As
Boolean
    BoolConsultaParametros = False
    Try
        Cmd = New SqlCommand(Procedimiento, Cnn)
        Cmd.CommandType = CommandType.StoredProcedure
        If VcParametros.Length > 0 Then
            Cmd.Parameters.AddRange(VcParametros)
        End If
    End Try

```

```

        lector = Cmd.ExecuteReader()
        BoolConsultaParametros = True
    Catch ex As SqlException
        If MostrarMSj Then MessageBox.Show(ex.Message, NS,
        MessageBoxButtons.OK, MessageBoxIcon.Asterisk,
        MessageBoxDefaultButton.Button1)
    End Try
End Function

```

```

Public Function IntEjecutarParametros(ByVal Procedimeiento As String,
ByVal VcParametros() As SqlParameter) As Integer
    IntEjecutarParametros = -1
    Try
        Cmd = New SqlCommand(Procedimeiento, Cnn)
        Cmd.CommandType = CommandType.StoredProcedure
        If VcParametros.Count > 0 Then
            Cmd.Parameters.AddRange(VcParametros)
        End If
        Dr = Cmd.ExecuteReader()
        If Dr.Read() Then
            If GetValorInt(Dr(0)) = 1 Then
                IntEjecutarParametros = 1
            End If
        End If
        Dr.Close()
    Catch ex As SqlException
        If MostrarMSj Then MessageBox.Show(ex.Message, NS,
        MessageBoxButtons.OK, MessageBoxIcon.Asterisk,
        MessageBoxDefaultButton.Button1)
    End Try
End Function

```

```

Public Function BoolConsulta(ByRef dr As SqlDataReader, ByVal sentencia
As String) As Boolean
    BoolConsulta = False
    Try
        Cmd = New SqlCommand(sentencia, Cnn)
        dr = Cmd.ExecuteReader()
        BoolConsulta = True
    Catch ex As SqlException
        If MostrarMSj Then    MessageBox.Show(ex.Message,    NS,
MessageBoxButtons.OK,    MessageBoxIcon.Asterisk,
MessageBoxDefaultButton.Button1)
    End Try
End Function

```

```

Public Function IntEjecutar(ByVal sentencia As String) As Integer
    Try
        Cmd = New SqlCommand(sentencia, Cnn)
        IntEjecutar = Cmd.ExecuteNonQuery()
    Catch ex As SqlException
        If MostrarMSj Then    MessageBox.Show(ex.Message,    NS,
MessageBoxButtons.OK,    MessageBoxIcon.Asterisk,
MessageBoxDefaultButton.Button1)
        IntEjecutar = -1
    End Try
End Function

```

```

Public Sub LLenarComboBox(ByRef Cb As ComboBox, ByVal Procedimiento
As String, ByVal VcP() As SqlParameter)

    With Cb
        .Visible = False : .Items.Clear() : .Enabled = False
    End With

```

```

Try
    If Not BoolConsultaParametros(Dreader, Procedimiento, VcP) Then
Cb.Visible = True : Exit Sub

    Cb.DisplayMember = "PTexto"

    While Dreader.Read
        Cb.Items.Add(New ElementoCB(Convert.ToString(Dreader("Texto")),
Convert.ToInt64(Dreader("ID"))))

    End While

    Catch ex As Exception
        MsgBox("Error al Llenar La Lista. Desc: " & ex.Message,
MsgBoxStyle.Critical, "Error de LlenarComboBox")

    Finally
        Cb.Visible = True : Cb.Enabled = True
    End Try

    If Not Dreader.IsClosed Then Dreader.Close()
    Dreader = Nothing
End Sub

Public Sub LLenarComboBoxStr(ByRef Cb As ComboBox, ByVal
Procedimiento As String, ByVal VcP() As SqlParameter)

    With Cb
        .Visible = False : .Items.Clear() : .Enabled = False
    End With

    Try

```

```
If Not BoolConsultaParametros(Dreader, Procedimiento, VcP) Then  
Cb.Visible = True : Exit Sub
```

```
    Cb.DisplayMember = "PTexto"
```

```
    While Dreader.Read
```

```
        Cb.Items.Add(New ElementoCB(Convert.ToString(Dreader("Texto")),  
Convert.ToString(Dreader("ID"))))
```

```
    End While
```

```
    Catch ex As Exception
```

```
        MsgBox("Error al Llenar La Lista. Desc: " & ex.Message,  
MsgBoxStyle.Critical, "Error de LlenarComboBox")
```

```
    Finally
```

```
        Cb.Visible = True : Cb.Enabled = True
```

```
    End Try
```

```
    If Not Dreader.IsClosed Then Dreader.Close()
```

```
    Dreader = Nothing
```

```
End Sub
```

```
Public Function EmpezarTransaccion() As Boolean
```

```
    EmpezarTransaccion = False
```

```
    Try
```

```
        Transaccion = Cnn.BeginTransaction
```

```
        Transac = True
```

```
        EmpezarTransaccion = True
```

```
    Catch ex As SqlException
```

```
        If MostrarMSj Then MessageBox.Show("Se ha producido el Siguiete  
Error" & Environment.NewLine & ex.Message)
```

```
    End Try
```

```
End Function
```

```

Public Function ConfirmarTransaccion() As Boolean
    ConfirmarTransaccion = False
    Try
        Transaccion.Commit()
        ConfirmarTransaccion = True
        Transac = False
    Catch ex As SqlException
        If MostrarMSj Then MessageBox.Show("Se ha producido el Siguiente
Error" & Environment.NewLine & ex.Message)
    End Try
End Function

```

```

Public Function DeshacerTransaccion() As Boolean
    DeshacerTransaccion = False
    Try
        Transaccion.Rollback()
        DeshacerTransaccion = True
        Transac = False
    Catch ex As SqlException
        If MostrarMSj Then MessageBox.Show("Se ha producido el Siguiente
Error" & Environment.NewLine & ex.Message)
    End Try
End Function

```

```

Public Function GetValor(ByRef valor As Object) As String
    If IsDBNull(valor) Then valor = ""
    Return valor
End Function

```

```

Public Function GetValorInt(ByRef valor As Object) As Integer
    If IsDBNull(valor) Then valor = 0
    Return valor
End Function

```

```
Public Sub LiberarRecursos()
```

```
    Cnn.Dispose() : Cnn = Nothing
```

```
    If Not IsNothing(Cmd) Then Cmd.Dispose() : Cmd = Nothing
```

```
    If Not IsNothing(Transaccion) Then Transaccion.Dispose() : Transaccion =  
Nothing
```

```
End Sub
```

```
Public Function SigID(ByVal campo As String, ByVal tabla As String) As  
Integer
```

```
    SigID = -1
```

```
    Try
```

```
        Cmd = New SqlCommand("SELECT MAX([" & campo & "]) As Max  
FROM " & "[" & tabla & "]", Cnn)
```

```
        Dreader = Cmd.ExecuteReader()
```

```
        While Dreader.Read
```

```
            SigID = GetValorInt(Dreader("Max")) + 1
```

```
        End While
```

```
    Catch ex As SqlException
```

```
        If MostrarMSj Then MessageBox.Show(ex.Message, NS,  
MessageBoxButtons.OK, MessageBoxIcon.Asterisk,  
MessageBoxDefaultButton.Button1)
```

```
    End Try
```

```
End Function
```

```
Public Function SigIDParametros(ByVal campo As String, ByVal tabla As  
String) As Integer
```

```
    SigIDParametros = -1
```

```
    Try
```

```
        Cmd = New SqlCommand("SiguienteID", Cnn)
```

```
        Cmd.CommandType = CommandType.StoredProcedure
```

```
        Cmd.Parameters.Add(New SqlParameter("@Campo", campo))
```

```

    Cmd.Parameters.Add(New SqlParameter("@Tabla", tabla))

Dreader = Cmd.ExecuteReader()
If (Dreader.Read) Then
    If IsDBNull(Dreader("IDSiguiente")) Then
        SigIDParametros = 1
    Else
        SigIDParametros = GetValorInt(Dreader("IDSiguiente")) + 1
    End If
End If
Dreader.Close()

    If    MostrarMSj    Then    MessageBox.Show(ex.Message,    NS,
    MessageBoxButtons.OK,                                MessageBoxIcon.Asterisk,
    MessageBoxDefaultButton.Button1)
    End Try
End Function

#End Region

End Class

```

## **CAPA DE NEGOCIOS**

### **CLASE CLIENTE**

```

Imports System.Data.SqlClient

Public Class Cliente : Inherits Persona
    Private VcParametros() As SqlParameter
    Public IDCliente As Integer

    Public Sub New()

```

End Sub

Public Function GetClientes() As List(Of List(Of String))

    GetClientes = New List(Of List(Of String))

    Dim lista As List(Of String)

    Array.Resize(VcParametros, 0)

    If ObjDatos.BoolConsultaParametros(Dr, "S\_Clientes", VcParametros)

Then

        While Dr.Read()

            lista = New List(Of String)

            lista.Add(ObjDatos.GetValor(Dr("ID")))

            lista.Add(ObjDatos.GetValor(Dr("TEXTO")))

            lista.Add(ObjDatos.GetValor(Dr("CEDULA")))

            GetClientes.Add(lista)

        End While

        Dr.Close()

    End If

End Function

Public Function SaveCliente() As Boolean

    SaveCliente = False

    IDCliente = ObjDatos.SigIDParametros("IDCliente", "TbCliente")

    If IDCliente > 0 Then

        If SavePersona() Then

            Array.Resize(VcParametros, 2)

            VcParametros(0) = New SqlParameter("@IDCliente", IDCliente)

            VcParametros(1) = New SqlParameter("@Cedula", cedula)

            If ObjDatos.IntEjecutarParametros("I\_Cliente", VcParametros) Then

SaveCliente = True

        End If

    End If

End Function

```

Public Sub LLenarCBCliente(ByRef cb As ComboBox)
    Array.Resize(VcParametros, 0)
    ObjDatos.LLenarComboBox(cb, "S_ClienteNombres", VcParametros)
End Sub
End Class

```

## **CLASE CONSULTA**

```
Imports System.Data.SqlClient
```

```
Public Class Consulta
```

```
    Public IDConsulta As Integer
```

```
    Public ProximaConsulta As Date
```

```
    Public Motivo As String
```

```
    Public Fecha As Date
```

```
    Public IDMascota As Integer
```

```
    Public Detalles As New List(Of DetalleConsulta)
```

```
    Private VCPParametros() As SqlParameter
```

```
    Private ObjDetalle As New DetalleConsulta
```

```
    Public clienteID As Integer
```

```
    Public Function SaveConsulta() As Boolean
```

```
        SaveConsulta = False
```

```
        IDConsulta = ObjDatos.SigIDParametros("IDConsulta", "TbConsulta")
```

```
        If IDConsulta > 0 Then
```

```
            Array.Resize(VCPParametros, 4)
```

```
            VCPParametros(0) = New SqlParameter("@ID", IDConsulta)
```

```
            VCPParametros(1) = New SqlParameter("@MascotaID", IDMascota)
```

```
            VCPParametros(2) = New SqlParameter("@FechaProxima",
```

```
ProximaConsulta)
```

```
            VCPParametros(3) = New SqlParameter("@Observacion", Motivo)
```

```
            If ObjDatos.IntEjecutarParametros("I_Con consulta", VCPParametros) > -1
```

```
Then
```

```
        For Each item As DetalleConsulta In Detalles
```

```
            item.IDConsulta = IDConsulta : item.SaveDetalle()
```

```

        Next
        SaveConsulta = True
    End If
End If
End Function

```

```

Public Function getListaConsulta(ByVal IDMC As Integer) As List(Of List(Of
String))

```

```

    getListaConsulta = New List(Of List(Of String))
    Dim lista As List(Of String)
    Array.Resize(VCPParametros, 1)
    VCPParametros(0) = New SqlParameter("@IDMC", IDMC)
    If ObjDatos.BoolConsultaParametros(Dr, "S_ConсультаMascota",
VCPParametros) Then
        While Dr.Read()
            lista = New List(Of String)
            lista.Add(ObjDatos.GetValor(Dr("FechaConsulta")))
            lista.Add(ObjDatos.GetValor(Dr("Observacion")))
            lista.Add(ObjDatos.GetValor(Dr("IDconsulta")))
            getListaConsulta.Add(lista)
        End While
        Dr.Close()
    End If
End Function

```

```

Public Function GetConsulta(ByVal ID As Integer) As Boolean
    GetConsulta = False
    Array.Resize(VCPParametros, 1)
    VCPParametros(0) = New SqlParameter("@IDConsulta", ID)
    If ObjDatos.BoolConsultaParametros(Dr, "S_Conсульта", VCPParametros)
Then
        While Dr.Read
            ProximaConsulta = ObjDatos.GetValor(Dr("FechaProxima"))

```

```

        Motivo = ObjDatos.GetValor(Dr("Observacion"))
        Fecha = ObjDatos.GetValor(Dr("FechaConsulta"))
        IDMascota = ObjDatos.GetValorInt(Dr("MascotaID"))
        clienteID = ObjDatos.GetValorInt(Dr("IDCLiente"))
    End While
    Dr.Close()
    Detalles = ObjDetalle.GetDetalles(ID) : GetConsulta = True
End If
End Function
End Class

```

### **CLASE DETALLE CONSULTA**

```

Imports System.Data.SqlClient
Public Class DetalleConsulta
    Private VCPParametros() As SqlParameter
    Public IDConsulta As Integer
    Public Medicamento As Integer
    Public Tratamiento As String
    Public Dosis As String
    Public Sub New()

    End Sub
    Public Sub New(ByVal medi As Integer, ByVal trata As String, ByVal doss As
String)
        Medicamento = medi : Tratamiento = trata : Dosis = doss
    End Sub
    Public Function SaveDetalle() As Boolean
        SaveDetalle = False
        Array.Resize(VCPParametros, 4)
        VCPParametros(0) = New SqlParameter("@IDConsulta", IDConsulta)
        VCPParametros(1) = New SqlParameter("@MedicamentoID",
Medicamento)
        VCPParametros(2) = New SqlParameter("@Dosis", Dosis)

```

```

    VCPParametros(3) = New SqlParameter("@Tratamiento", Tratamiento)
    If ObjDatos.IntEjecutarParametros("I_Detalle", VCPParametros) > -1 Then
SaveDetalle = True
    End Function

```

```

    Public Function GetDetalles(ByVal IDC As Integer) As List(Of
DetalleConsulta)
    GetDetalles = New List(Of DetalleConsulta)
    Array.Resize(VCPParametros, 1)
    VCPParametros(0) = New SqlParameter("@IDConsulta", IDC)
    If ObjDatos.BoolConsultaParametros(Dr, "S_DetalleConsulta",
VCPParametros) Then
        While Dr.Read()
            GetDetalles.Add(New
DetalleConsulta(ObjDatos.GetValorInt(Dr("MedicamentoID")),
ObjDatos.GetValor(Dr("Tratamiento")), ObjDatos.GetValor(Dr("Dosis"))))
        End While
        Dr.Close()
    End If
    End Function
End Class

```

### **CLASE ELEMENTOCB**

```

Public Class ElementoCB
    Private Texto As String
    Private IDObj As Object
    Private IDLong As Long
    Private IDSTr As String

    Public Sub New(ByVal _Texto As String, ByVal _IDObj As Object)
        Texto = _Texto : IDObj = _IDObj
    End Sub

```

```
Public Sub New(ByVal _Texto As String, ByVal _IDLong As Long)
    Texto = _Texto : IDLong = _IDLong
```

```
End Sub
```

```
Public Sub New(ByVal _Texto As String, ByVal _IDStr As String)
```

```
    Texto = _Texto : IDStr = _IDStr
```

```
End Sub
```

```
Public Property PTexto() As String
```

```
    Get
```

```
        Return Texto
```

```
    End Get
```

```
    Set(ByVal value As String)
```

```
        Texto = value
```

```
    End Set
```

```
End Property
```

```
Public Property IDString() As String
```

```
    Get
```

```
        Return IDStr
```

```
    End Get
```

```
    Set(ByVal value As String)
```

```
        IDStr = value
```

```
    End Set
```

```
End Property
```

```
Public Property IDEntero() As Long
```

```
    Get
```

```
        Return IDLong
```

```
    End Get
```

```
    Set(ByVal value As Long)
```

```
        IDLong = value
```

```
    End Set
```

```
End Property
```

```

Public Property IDObjeto() As Object
    Get
        Return IDObj
    End Get
    Set(ByVal value As Object)
        IDObj = value
    End Set
End Property
End Class

```

### **CLASE MASCOTA**

```
Imports System.Data.SqlClient
```

```
Public Class Mascota
```

```
    Public Nombre As String
```

```
    Public Observacion As String
```

```
    Public FechaNac As Date
```

```
    Public Macho As Boolean
```

```
    Public IDMascota As Integer
```

```
    Public IDTipo As Integer
```

```
    Public IDCliente As Integer
```

```
    Private VcParametros() As SqlParameter
```

```
    Public Tipo As String
```

```
    Public Sub New()
```

```
End Sub
```

```
    Public Sub LlenarCbMascota(ByVal cb As ComboBox, ByVal IDCliente As Integer)
```

```
        Array.Resize(VcParametros, 1)
```

```
        VcParametros(0) = New SqlParameter("@IDCliente", IDCliente)
```

```
        ObjDatos.LLenarComboBox(cb, "S_MascotaCliente", VcParametros)
```

```

End Sub

Public Function SaveTipo()
    SaveTipo = False
    IDTipo = ObjDatos.SigIDParametros("IDTipo", "TbTipoMascota")
    If IDTipo > 0 Then
        Array.Resize(VcParametros, 3)
        VcParametros(0) = New SqlParameter("@IDTipo", IDTipo)
        VcParametros(1) = New SqlParameter("@Tipo", Tipo)
        VcParametros(2) = New SqlParameter("@Observacion", Observacion)
        If ObjDatos.IntEjecutarParametros("I_TIPO", VcParametros) > 0 Then
SaveTipo = True
        End If
    End Function

Public Function GetMascota(ByVal idmsc As Integer) As Boolean
    GetMascota = False
    Array.Resize(VcParametros, 1)
    VcParametros(0) = New SqlParameter("@IDMascota", idmsc)
    If ObjDatos.BoolConsultaParametros(Dr, "S_Mascota", VcParametros)
Then
        While Dr.Read()
            IDMascota = ObjDatos.GetValorInt(Dr("IDMascota"))
            IDCliente = ObjDatos.GetValorInt(Dr("Cliente"))
            IDTipo = ObjDatos.GetValorInt(Dr("TipoID"))
            Nombre = ObjDatos.GetValor(Dr("Nombre"))
            Observacion = ObjDatos.GetValor(Dr("Observacion"))
            FechaNac = ObjDatos.GetValor(Dr("FechaNac"))
            Macho = ObjDatos.GetValor(Dr("sexo"))
        End While
        Dr.Close() : GetMascota = True
    End If
End Function

```

```
Public Sub New(ByVal idm As Integer, ByVal idc As Integer, ByVal idt As Integer, ByVal name As String, ByVal fn As Date, ByVal mac As Boolean, ByVal Obs As String)
```

```
    IDMascota = idm : IDCliente = idc : IDTipo = idt : Nombre = name : FechaNac = fn : Macho = mac : Observacion = Obs  
End Sub
```

```
Public Sub LLenarCbTipo(ByRef cb As ComboBox)
```

```
    Array.Resize(VcParametros, 0)
```

```
    ObjDatos.LLenarComboBox(cb, "S_TipoMascota", VcParametros)
```

```
End Sub
```

```
Public Function SaveMascota() As Boolean
```

```
    SaveMascota = False
```

```
    IDMascota = ObjDatos.SigIDParametros("IdMascota", "TbMascota")
```

```
    If IDMascota > 0 Then
```

```
        Array.Resize(VcParametros, 7)
```

```
        VcParametros(0) = New SqlParameter("@IDMascta", IDMascota)
```

```
        VcParametros(1) = New SqlParameter("@IDCliente", IDCliente)
```

```
        VcParametros(2) = New SqlParameter("@IDTipo", IDTipo)
```

```
        VcParametros(3) = New SqlParameter("@Nombre", Nombre)
```

```
        VcParametros(4) = New SqlParameter("@FechaNac", FechaNac)
```

```
        VcParametros(5) = New SqlParameter("@sexo", If(Macho, 1, 0))
```

```
        VcParametros(6) = New SqlParameter("@Observacion", Observacion)
```

```
        If ObjDatos.IntEjecutarParametros("I_Mascota", VcParametros) > 0
```

```
Then SaveMascota = True
```

```
    End If
```

```
End Function
```

```
Public Function GetListaMascota(ByVal IDC As Integer) As List(Of Mascota)
```

```
    GetListaMascota = New List(Of Mascota)
```

```

    Array.Resize(VcParametros, 1)
    VcParametros(0) = New SqlParameter("@IDCLiente", IDC)
    If ObjDatos.BoolConsultaParametros(Dr, "S_Mascotas", VcParametros)
Then
    While Dr.Read()

        GetListaMascota.Add(New
Mascota(ObjDatos.GetValorInt(Dr("IDMASCOTA")),
ObjDatos.GetValorInt(Dr("CLIENTE")),    ObjDatos.GetValorInt(Dr("TIPOID")),
ObjDatos.GetValor(Dr("NOMBRE")),
CType(ObjDatos.GetValor(Dr("FECHANAC")),    Date),    CType(Dr("SEXO"),
Boolean), ObjDatos.GetValor(Dr("OBSERVACION"))))
        End While
    End If
End Function

End Class

```

### **CLASE MEDICAMENTO**

```

Imports System.Data.SqlClient
Public Class Medicamento
    Public IDMedicamento As Integer
    Public Medicamento As String
    Public Observaion As String
    Public Descripcion As String
    Private VCPParametros() As SqlParameter
    Public Sub New()

End Sub

Public Sub LlenarCBMedicamento(ByVal cb As ComboBox)
    Array.Resize(VCPParametros, 0)
    ObjDatos.LLenarComboBox(cb, "S_Medicamentos", VCPParametros)

```

```

End Sub
Public Function SaveMedicamento()
    SaveMedicamento = False
    IDMedicamento = ObjDatos.SigIDParametros("IDMedicamento",
    "TbMedicamento")
    If IDMedicamento > 0 Then
        Array.Resize(VCPParametros, 4)
        VCPParametros(0) = New SqlParameter("@IDMedicamento",
IDMedicamento)
        VCPParametros(1) = New SqlParameter("@Medicamento",
Medicamento)
        VCPParametros(2) = New SqlParameter("@Observacion", Observaion)
        VCPParametros(3) = New SqlParameter("@Descripcion", Descripcion)
        If ObjDatos.IntEjecutarParametros("I_Medicamento", VCPParametros) >
0 Then SaveMedicamento = True
        End If
    End Function

End Class

```

## **CLASE PERSONA**

```

Imports System.Data.SqlClient
Public Class Persona
    Public cedula As String
    Public nombres As String
    Public ApellidoP As String
    Public ApellidoM As String
    Public Masculino As Boolean
    Public fono As String
    Public IDciudad As Integer
    Public IdProvincia As Integer
    Public Direccion As String

```

```

Private VcParametro() As SqlParameter

Public Sub New()

End Sub

Public Function ExistePersonaUsuario(ByVal CD As String) As Boolean
    ExistePersonaUsuario = False
    Array.Resize(VcParametro, 1)
    VcParametro(0) = New SqlParameter("@Cd", CD)
    If ObjDatos.BoolConsultaParametros(Dr, "S_Persona", VcParametro)
Then
        If Dr.HasRows Then ExistePersonaUsuario = True
        Dr.Close()
    End If
End Function

Public Function GetCliente(ByVal cd As String) As Boolean
    GetCliente = False
    Array.Resize(VcParametro, 1)
    VcParametro(0) = New SqlParameter("@Cd", cd)
    If ObjDatos.BoolConsultaParametros(Dr, "S_Persona", VcParametro)
Then
        While Dr.Read()
            cedula = ObjDatos.GetValor(Dr("Cedula"))
            nombres = ObjDatos.GetValor(Dr("Nombres"))
            ApellidoM = ObjDatos.GetValor(Dr("ApellidoM"))
            ApellidoP = ObjDatos.GetValor(Dr("ApellidoP"))
            Masculino = ObjDatos.GetValor(Dr("Sexo"))
            fono = ObjDatos.GetValor(Dr("Telefono"))
            IDciudad = ObjDatos.GetValor(Dr("CiudadID"))
            Direccion = ObjDatos.GetValor(Dr("Direccion"))
        End While
        Dr.Close() : GetCliente = True
    End If
End Function

```

```
End If
End Function
```

```
Public Function SavePersona() As Boolean
    SavePersona = False
    Array.Resize(VcParametro, 8)
    VcParametro(0) = New SqlParameter("@Cedula", cedula)
    VcParametro(1) = New SqlParameter("@Nombres", nombres)
    VcParametro(2) = New SqlParameter("@ApellidoP", ApellidoP)
    VcParametro(3) = New SqlParameter("@ApellidoM", ApellidoM)
    VcParametro(4) = New SqlParameter("@Sexo", Masculino)
    VcParametro(5) = New SqlParameter("@Fono", fono)
    VcParametro(6) = New SqlParameter("@Ciudad", IDciudad)
    VcParametro(7) = New SqlParameter("@Direccion", Direccion)
    If ObjDatos.IntEjecutarParametros("I_Persona", VcParametro) > 0 Then
SavePersona = True
    End Function
```

```
Public Function BoolCeedula(ByVal CI As String) As Boolean
    BoolCeedula = False
    If CI.Trim.Length = 10 Then
        Dim a(9) As Integer
        For i As Integer = 0 To 9
            a(i) = Integer.Parse(CI(i).ToString())
        Next
        For cont As Integer = 0 To 9 Step 2
            a(cont) = a(cont) * 2
        Next
        For conta As Integer = 1 To 9 Step 2
            a(conta) = a(conta) * 1
        Next
        For ii As Integer = 0 To 9
            If a(ii) >= 10 Then
```

```

        a(ii) = (a(ii) - 10) + 1
    End If
Next
Dim contaaa As Integer = a(0) + a(1) + a(2) + a(3) + a(4) + a(5) + a(6) +
a(7) + a(8)
For s As Integer = 0 To 90 Step 10
    If contaaa > (s) And contaaa <= (s + 10) Then
        contaaa = (s + 10) - contaaa
    End If
Next
If contaaa = a(9) Then
    BoolCeedula = True
End If
End If
End Function
End Class

```

## **CLASE PROVINCIA**

```

Imports System.Data.SqlClient
Public Class Provincia
    Private VcParametro() As SqlParameter

    Public Sub LLenarCBP(ByRef cb As ComboBox)
        Array.Resize(VcParametro, 0)
        ObjDatos.LLenarComboBox(cb, "S_Provincia", VcParametro)
    End Sub

    Public Sub LLenarCBC(ByRef cb As ComboBox, ByVal idp As Integer)
        Array.Resize(VcParametro, 1)
        VcParametro(0) = New SqlParameter("@IDProvincia", idp)
        ObjDatos.LLenarComboBox(cb, "S_Ciudad", VcParametro)
    End Sub

```

End Class

## CLASE USUARIO

Imports System.Data.SqlClient

Public Class Usuario : Inherits Persona

Public IDUsuario As Integer

Public Usuario As String

Public clave As String

Private VcParametro() As SqlParameter

Public Function ComprobarUsuario() As Boolean

ComprobarUsuario = False

Array.Resize(VcParametro, 2)

VcParametro(0) = New SqlParameter("@Usr", Usuario)

VcParametro(1) = New SqlParameter("@clave", clave)

If ObjDatos.BoolConsultaParametros(Dr, "S\_ComprobarUsr",  
VcParametro) Then

If Dr.Read() Then

ComprobarUsuario = True

IDUSr = ObjDatos.GetValorInt(Dr("IDUSUARIO"))

End If

Dr.Close()

End If

End Function

Public Function CambiarClave(ByVal idusr As Integer, ByVal clave As String)  
As Boolean

CambiarClave = False

Array.Resize(VcParametro, 2)

VcParametro(0) = New SqlParameter("@idUsr", idusr)

VcParametro(1) = New SqlParameter("@clave", clave)

```

        If ObjDatos.IntEjecutarParametros("CambiarClvaeUSR", VcParametro) > 0
Then CambiarClave = True
    End Function
    Public Function SaveUSuario() As Boolean
        SaveUSuario = True
        IDUsuario = ObjDatos.SiglIDParametros("IDUsuario", "TbUsuario")
        If IDUsuario > -1 Then
            If SavePersona() Then
                Array.Resize(VcParametro, 4)
                VcParametro(0) = New SqlParameter("@ID", IDUsuario)
                VcParametro(1) = New SqlParameter("@Persona", cedula)
                VcParametro(2) = New SqlParameter("@Nick", Usuario)
                VcParametro(3) = New SqlParameter("@Clave", clave)
                If ObjDatos.IntEjecutarParametros("I_Usuario", VcParametro) > 0
Then SaveUSuario = True
                    End If
                End If
            End Function
        End Class

```

## **SISTEMA DE DISPOSITIVO MOVIL**

### **CAPA DE DATOS**

#### **CLASE CONEXIÓN LOCAL**

```

Imports System.Data.SqlClient
Imports System.Data
Public Class ConexionServer

#Region "ATRIBUTOS"
    Private Cnn As SqlConnection
    Private Cmd As SqlCommand
    Private Dreader As SqlDataReader

```

```

Private Transaccion As SqlTransaction
Private Transac As Boolean
Public MostrarMSj As Boolean
Private DA As SqlDataAdapter
Private DSet As DataSet
#End Region

#Region "CONSTRUCTOR"

Public Sub New()

    Cnn = New SqlConnection("Data Source=KARLA-PC\SQLEXPRESS;Initial
Catalog=Veterinaria; User ID=KarlaTesis; Password=12345")
End Sub

#End Region

#Region "METODOS"
Public Function IntEjecutar(ByVal sentencia As String) As Integer
Try
    If Transac Then
        Cmd = New SqlCommand(sentencia, Cnn, Transaccion)
    Else
        Cmd = New SqlCommand(sentencia, Cnn)
    End If

    IntEjecutar = Cmd.ExecuteNonQuery()
Catch ex As SqlException
    IntEjecutar = -1
End Try
End Function

```

```
Public Function boolCrearDataTable(ByRef DT As DataTable, ByVal nombre  
As String, ByVal SetenciaSQL As String) As Boolean
```

```
Try
```

```
DA = New SqlDataAdapter(SetenciaSQL, Cnn)
```

```
DSet = New DataSet()
```

```
DA.Fill(DSet)
```

```
DT = DSet.Tables(0).Copy
```

```
DT.TableName = nombre
```

```
DA.Dispose() : DA = Nothing
```

```
boolCrearDataTable = True
```

```
Catch ex As SQLException
```

```
    If MostrarMSj Then MsgBox("Error SQL: " & ex.Message,  
MsgBoxStyle.Critical, "Error SQL")
```

```
    boolCrearDataTable = False
```

```
End Try
```

```
End Function
```

```
Public Function BoolConectar() As Boolean
```

```
    BoolConectar = False
```

```
Try
```

```
    If Cnn.State = Data.ConnectionState.Closed Then
```

```
        Cnn.Open()
```

```
        BoolConectar = True
```

```
    End If
```

```
Catch ex As SQLException
```

```
    End Try
```

```
End Function
```

```
Public Function BoolCerrar() As Boolean
```

```

BoolCerrar = False
Try
    If Cnn.State = Data.ConnectionState.Open Then
        Cnn.Close()
        BoolCerrar = True
    End If
Catch ex As Exception

End Try

```

```

End Function

```

```

Public Function BoolConsulta(ByRef dr As SqlDataReader, ByVal sentencia
As String) As Boolean
    BoolConsulta = False
    Try
        Cmd = New SqlCommand(sentencia, Cnn)
        dr = Cmd.ExecuteReader()
        BoolConsulta = True
    Catch ex As SqlException
    End Try
End Function

```

```

Public Function EmpezarTransaccion() As Boolean
    EmpezarTransaccion = False
    Try
        Transaccion = Cnn.BeginTransaction
        Transac = True
        EmpezarTransaccion = True
    Catch ex As SqlException
        If MostrarMSj Then MessageBox.Show("Se ha producido el Siguiente
Error" & Environment.NewLine & ex.Message)
    End Try
End Function

```

```
End Try
End Function
```

```
Public Function ConfirmarTransaccion() As Boolean
    ConfirmarTransaccion = False
    Try
        Transaccion.Commit()
        ConfirmarTransaccion = True
        Transac = False
    Catch ex As SqlException
        If MostrarMSj Then MessageBox.Show("Se ha producido el Siguiete
Error" & Environment.NewLine & ex.Message)
    End Try
End Function
```

```
Public Function DeshacerTransaccion() As Boolean
    DeshacerTransaccion = False
    Try
        Transaccion.Rollback()
        DeshacerTransaccion = True
        Transac = False
    Catch ex As SqlException
        If MostrarMSj Then MessageBox.Show("Se ha producido el Siguiete
Error" & Environment.NewLine & ex.Message)
    End Try
End Function
```

```
Public Function GetValor(ByRef valor As Object) As String
    If IsDBNull(valor) Then valor = ""
    Return valor
End Function
```

```
Public Function GetValorInt(ByRef valor As Object) As Integer
```

```
    If IsDBNull(valor) Then valor = 0
    Return valor
End Function
```

```
Public Sub LiberarRecursos()
```

```
    If Not IsNothing(Cnn) Then Cnn.Dispose() : Cnn = Nothing
```

```
    If Not IsNothing(Cmd) Then Cmd.Dispose() : Cmd = Nothing
```

```
    If Not IsNothing(Transaccion) Then Transaccion.Dispose() : Transaccion =
Nothing
```

```
End Sub
```

```
Public Function SigID(ByVal campo As String, ByVal tabla As String) As
Integer
```

```
    SigID = -1
```

```
    Try
```

```
        Cmd = New SqlCommand("SiguienteID", Cnn)
```

```
        Cmd.CommandType = CommandType.StoredProcedure
```

```
        Cmd.Parameters.Add(New SqlParameter("@Campo", campo))
```

```
        Cmd.Parameters.Add(New SqlParameter("@Tabla", tabla))
```

```
        Dreader = Cmd.ExecuteReader()
```

```
        If (Dreader.Read) Then
```

```
            If IsDBNull(Dreader("IDSiguiente")) Then
```

```
                SigID = 1
```

```
            Else
```

```
                SigID = GetValorInt(Dreader("IDSiguiente")) + 1
```

```
            End If
```

```
        End If
```

```
    Catch ex As SqlException
```

```
    End Try
```

```
End Function
```

#End Region

End Class

### **CLASE CONEXION SERVIDOR**

Imports System.Data.SqlClient

Imports System.Data

Public Class ConexionServer

#Region "ATRIBUTOS"

Private Cnn As SqlConnection

Private Cmd As SqlCommand

Private Dreader As SqlDataReader

Private Transaccion As SqlTransaction

Private Transac As Boolean

Public MostrarMSj As Boolean

Private DA As SqlDataAdapter

Private DSet As DataSet

#End Region

#Region "CONSTRUCTOR"

Public Sub New()

Cnn = New SqlConnection("Data Source=KARLA-PC\SQLEXPRESS;Initial  
Catalog=Veterinaria; User ID=KarlaTesis; Password=12345")

End Sub

#End Region

#Region "METODOS"

Public Function IntEjecutar(ByVal sentencia As String) As Integer

```

Try
    If Transac Then
        Cmd = New SqlCommand(sentencia, Cnn, Transaccion)
    Else
        Cmd = New SqlCommand(sentencia, Cnn)
    End If

    IntEjecutar = Cmd.ExecuteNonQuery()
Catch ex As SqlException
    IntEjecutar = -1
End Try
End Function

```

```

Public Function boolCrearDataTable(ByRef DT As DataTable, ByVal nombre
As String, ByVal SetenciaSQL As String) As Boolean

```

```

Try
    DA = New SqlDataAdapter(SetenciaSQL, Cnn)
    DSet = New DataSet()

    DA.Fill(DSet)
    DT = DSet.Tables(0).Copy
    DT.TableName = nombre
    DA.Dispose() : DA = Nothing
    boolCrearDataTable = True

Catch ex As SqlException
    If MostrarMSj Then MsgBox("Error SQL: " & ex.Message,
MsgBoxStyle.Critical, "Error SQL")
    boolCrearDataTable = False
End Try

End Function

```

```

Public Function BoolConectar() As Boolean
    BoolConectar = False
    Try
        If Cnn.State = Data.ConnectionState.Closed Then
            Cnn.Open()
            BoolConectar = True
        End If
    Catch ex As SqlException
    End Try

```

End Function

```

Public Function BoolCerrar() As Boolean
    BoolCerrar = False
    Try
        If Cnn.State = Data.ConnectionState.Open Then
            Cnn.Close()
            BoolCerrar = True
        End If
    Catch ex As Exception
    End Try

```

End Function

```

Public Function BoolConsulta(ByRef dr As SqlDataReader, ByVal sentencia
As String) As Boolean
    BoolConsulta = False
    Try
        Cmd = New SqlCommand(sentencia, Cnn)
        dr = Cmd.ExecuteReader()
        BoolConsulta = True
    Catch ex As SqlException
    End Try

```

End Function

```

Public Function EmpezarTransaccion() As Boolean
    EmpezarTransaccion = False

```

```

Try
    Transaccion = Cnn.BeginTransaction
    Transac = True
    EmpezarTransaccion = True
Catch ex As SqlException
    If MostrarMSj Then MessageBox.Show("Se ha producido el Siguiete
Error" & Environment.NewLine & ex.Message)
End Try
End Function

```

```

Public Function ConfirmarTransaccion() As Boolean
    ConfirmarTransaccion = False
    Try
        Transaccion.Commit()
        ConfirmarTransaccion = True
        Transac = False
    Catch ex As SqlException
        If MostrarMSj Then MessageBox.Show("Se ha producido el Siguiete
Error" & Environment.NewLine & ex.Message)
    End Try
End Function

```

```

Public Function DeshacerTransaccion() As Boolean
    DeshacerTransaccion = False
    Try
        Transaccion.Rollback()
        DeshacerTransaccion = True
        Transac = False
    Catch ex As SqlException
        If MostrarMSj Then MessageBox.Show("Se ha producido el Siguiete
Error" & Environment.NewLine & ex.Message)
    End Try
End Function

```

```

Public Function GetValor(ByRef valor As Object) As String
    If IsDBNull(valor) Then valor = ""
    Return valor
End Function

```

```

Public Function GetValorInt(ByRef valor As Object) As Integer
    If IsDBNull(valor) Then valor = 0
    Return valor
End Function

```

```

Public Sub LiberarRecursos()

```

```

    If Not IsNothing(Cnn) Then Cnn.Dispose() : Cnn = Nothing

```

```

    If Not IsNothing(Cmd) Then Cmd.Dispose() : Cmd = Nothing

```

```

    If Not IsNothing(Transaccion) Then Transaccion.Dispose() : Transaccion =
Nothing

```

```

End Sub

```

```

Public Function SigID(ByVal campo As String, ByVal tabla As String) As
Integer

```

```

    SigID = -1

```

```

    Try

```

```

        Cmd = New SqlCommand("SiguienteID", Cnn)

```

```

        Cmd.CommandType = CommandType.StoredProcedure

```

```

        Cmd.Parameters.Add(New SqlParameter("@Campo", campo))

```

```

        Cmd.Parameters.Add(New SqlParameter("@Tabla", tabla))

```

```

        Dreader = Cmd.ExecuteReader()

```

```

    If (Dreader.Read) Then

```

```

        If IsDBNull(Dreader("IDSiguiente")) Then

```

```

            SigID = 1

```

```

        Else

```

```

            SigID = GetValorInt(Dreader("IDSiguiente")) + 1

```

```

        End If

```

```

        End If
        Catch ex As SqlException
        End Try
    End Function
#End Region

End Class

```

## **CAPA DE NEGOCIOS**

### **CLASE CLIENTE**

```
Public Class Cliente : Inherits Persona
```

```
#Region "Atributos"
```

```
    Public IDCliente As Integer
```

```
#End Region
```

```
#Region "Constructor"
```

```
    Public Sub New()
```

```
    End Sub
```

```
    Public Sub New(ByVal id As Integer, ByVal ci As String, ciud As integer,
    ByVal name As String, ByVal app As String, ByVal apm As String, ByVal dir As
    String, ByVal fono As String,ByVal sx As Boolean)
```

```
        IDCliente = id : Cedula = ci : IDCiudad = ciud : Nombre = name : ApellidoP
    = app : ApellidoM = apm : Direccion = dir : Telefono = fono : Masculino = sx
```

```
    End Sub
```

```
#End Region
```

```
#Region "Métodos"
```

```
    Public Function GetCliente(ByVal ID As Integer) As Boolean
```

```
        If ObjDatos.BoolConsulta(Dr, "SELECT PERSONAID FROM TBCLIENTE
    WHERE IDCLIENTE=" & ID & ";") Then
```

```
            While Dr.Read
```

```
                Cedula = ObjDatos.GetValor(Dr("PERSONAID"))
```

```

        End While
        Dr.Close()
        If GetPersona() Then GetCliente = True
    End If

End Function

Public Function GetListCliente() As List(Of Cliente)
    GetListCliente = New List(Of Cliente)
    If ObjDatos.BoolConsulta(Dr, "SELECT TbCliente.IDCliente,
TbPersona.Cedula, TbPersona.CiudadID, TbPersona.Nombres, " _
        & "TbPersona.ApellidoP, TbPersona.ApellidoM,
TbPersona.Direccion, TbPersona.Telefono, TbPersona.Sexo " _
        & "FROM TbCliente INNER JOIN TbPersona ON
TbCliente.PersonalID = TbPersona.Cedula") Then
        While Dr.Read
            GetListCliente.Add(New Cliente(Dr("IDCliente"), Dr("Cedula"),
Dr("CiudadID"), Dr("Nombres"), Dr("ApellidoP"), Dr("ApellidoM"),
Dr("Direccion"), Dr("Telefono"), If(Dr("Sexo") = "TRUE", True, False)))
        End While
        Dr.Close()
    End If
End Function

Public Function SaveCliente() As Boolean

    IDCliente = ObjDatos.SigID("IDCLIENTE", "TBCLIENTE")
    If IDCliente > 0 Then
        If ObjDatos.EmpezarTransaccion() Then
            If SavePersona() Then
                If ObjDatos.IntEjecutar("INSERT INTO TBCLIENTE VALUES(" &
IDCliente & ", " & Cedula & ", 1);") Then
                    ObjDatos.ConfirmarTransaccion()
                    SaveCliente = True
                End If
            End If
        End If
    End If
End Function

```

```

        Else
            ObjDatos.DeshacerTransaccion()
        End If
    Else
        ObjDatos.DeshacerTransaccion()
    End If
End If
End If
End Function
#End Region
End Class

```

## **CLASE CONSULTA**

```

Public Class Consulta
    Public IDConsulta As Integer
    Public MascotaID As Integer
    Public fecha As Date
    Public fechanext As Date
    Public Motivo As String
    Public ClientelID As Integer
    Public Detalles As List(Of DetalleConsulta)

    Public Sub New()

    End Sub

    Public Function GetConsulta(ByVal ID As Integer) As Boolean
        Dim ObjDetalle As New DetalleConsulta()
        If ObjDatos.BoolConsulta(Dr, "SELECT TbConsulta.IDConsulta,
TbConsulta.MascotaID, TbConsulta.FechaProxima, TbConsulta.Observacion, "
_
        & "TbConsulta.FechaConsulta, TbMascota.Cliente FROM
TbConsulta INNER JOIN " _

```

```

        & "TbMascota ON TbConsulta.MascotalD =
TbMascota.IDMascota WHERE (TbConsulta.IDConsulta =" & ID & ")") Then
        'If ObjDatos.BoolConsulta(Dr, "SELECT TbConsulta.IDConsulta,
TbConsulta.MascotalD, TbConsulta.FechaProxima, " _
        '
        ' & "TbConsulta.Observacion,
TbConsulta.FechaConsulta, TbMascota.Cliente, TbCliente.PersonalID() " _
        '
        ' & "FROM TbCliente INNER JOIN TbMascota ON
TbCliente.IDCliente = TbMascota.Cliente INNER JOIN " _
        '
        ' & "TbPersona ON TbCliente.PersonalID =
TbPersona.Cedula CROSS JOIN TbConsulta() WHERE " _
        '
        ' & "(TbConsulta.IDConsulta =" & ID & ")") Then
        While Dr.Read()
            IDConsulta = ObjDatos.GetValorInt(Dr("IDConsulta"))
            ClienteID = ObjDatos.GetValorInt(Dr("Cliente"))
            MascotalD = ObjDatos.GetValorInt(Dr("MascotalD"))
            fecha = CType(Dr("FechaConsulta"), Date)
            fechanext = CType(Dr("FechaProxima"), Date)
            Motivo = ObjDatos.GetValor(Dr("Observacion"))
        End While
        Dr.Close()
        Detalles = ObjDetalle.GetDetalles(IDConsulta)
        GetConsulta = True
    Else
        MessageBox.Show("Error al consultar datos", NS,
MessageBoxButtons.OK, MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button1)
    End If
End Function

Public Function SaveConsulta() As Boolean
    IDConsulta = ObjDatos.SigID("idconsulta", "Tbconsulta")
    If IDConsulta > -1 Then
        ObjDatos.EmpezarTransaccion()

```

```

        If ObjDatos.IntEjecutar("INSERT INTO TBCONSULTA VALUES(" &
IDConsulta & "," & MascotaID _
                & "," & fechanext.ToString("MM/dd/yyyy") & "," & Motivo
& "," & fecha.ToString("MM/dd/yyyy") & ",1);") > -1 Then
            For Each det As DetalleConsulta In Detalles
                If Not det.SaveDetalle(IDConsulta) Then
                    ObjDatos.DeshacerTransaccion() : Exit Function
                End If
            Next
            ObjDatos.ConfirmarTransaccion()
            SaveConsulta = True
        Else
            ObjDatos.DeshacerTransaccion()
        End If
    End If
End Function
End Class

```

### **CLASE DETALLE CONSULTA**

```

Public Class DetalleConsulta
    Public IDMedic As Integer
    Public MedicamentoNombre As String
    Public Dosis As String
    Public Tratamiento As String

    Public Sub New(ByVal med As Integer, ByVal dos As String, ByVal trata As
String, ByVal medic As String)
        IDMedic = med : Dosis = dos : Tratamiento = trata : MedicamentoNombre
= medic
    End Sub

    Public Sub New()

```

End Sub

```
Public Function SaveDetalle(byval IDConsulta As integer) As Boolean
    If ObjDatos.IntEjecutar("INSERT INTO TBDETALLECONSULTA
VALUES(" & IDConsulta _
        & "," & IDMedic & "," & Dosis & "," & Tratamiento &
",1);") > -1 Then SaveDetalle = True
End Function
```

```
Public Function GetDetalles(ByVal IDConsulta As Integer) As List(Of
DetalleConsulta)
    GetDetalles = New List(Of DetalleConsulta)
    If ObjDatos.BoolConsulta(Dr, "SELECT TbDetalleConsulta.ConsultaID,
TbDetalleConsulta.MedicamentoID, " _
        & "TbDetalleConsulta.Dosis,
TbDetalleConsulta.Tratamiento, TbMedicamento.Medicamento " _
        & "FROM TbDetalleConsulta INNER JOIN TbMedicamento
ON TbDetalleConsulta.MedicamentoID = TbMedicamento.IDMedicamento " _
        & "WHERE (TbDetalleConsulta.ConsultaID = " &
IDConsulta & ")") Then
        While Dr.Read()
            GetDetalles.Add(New
DetalleConsulta(ObjDatos.GetValorInt(Dr("MedicamentoID")),
ObjDatos.GetValor(Dr("Dosis")), ObjDatos.GetValor(Dr("Tratamiento")),
ObjDatos.GetValor(Dr("Medicamento"))))
        End While
        Dr.Close()
    End If
End Function
```

End Class

## CLASE MASCOTA

Public Class Mascota

#Region "Atributos"

Public IdMascota As Integer

Public clienteld As Integer

Public Tipold As Integer

Public Nombre As String

Public macho As Boolean

Public observacion As String

Public fechaNac As Date

#End Region

#Region "cONSTRUCTORES"

Public Sub New()

End Sub

Public Sub New(ByVal ID As Integer, ByVal CLIENT As Integer, ByVal TIPO As Integer, ByVal name As String, ByVal SEXO As Boolean, ByVal OBS As String, ByVal F As Date)

IdMascota = ID : clienteld = CLIENT : Tipold = TIPO : Nombre = name : macho = SEXO : observacion = OBS : fechaNac = F

End Sub

#End Region

#Region "Metodos"

Public Function SaveMascota() As Boolean

IdMascota = ObjDatos.SigID("IDMASCOTA", "TBMASCOTA")

If IdMascota > 0 Then

If ObjDatos.IntEjecutar("INSERT INTO TBMASCOTA VALUES(" & IdMascota & "," \_

& Tipold & "," & clienteld & "," & Nombre & "," \_

& fechaNac.ToString("MM/dd/yyyy") & "," & IIf(macho, 1,

0) & "," & observacion & ",1);") > -1 Then SaveMascota = True

```

End If
End Function

Public Function GetMascota(ByVal Id As Integer) As Boolean
    If ObjDatos.BoolConsulta(Dr, "SELECT * FROM TBMASCOTA WHERE
IDMASCOTA=" & Id & ";") Then
        While Dr.Read()
            clienteId = ObjDatos.GetValorInt(Dr("CLIENTE"))
            Tipoid = ObjDatos.GetValorInt(Dr("TIPOID"))
            Nombre = ObjDatos.GetValor(Dr("NOMBRE"))
            macho=CType(Dr("SEXO"),Boolean)
            observacion = ObjDatos.GetValor(Dr("OBSERVACION"))
            fechaNac = CType(ObjDatos.GetValor(Dr("FECHANAC")), Date)
        End While
        Dr.Close()
        GetMascota = True
    End If
End Function

Public Function GETIISTMASC(ByVal ID As Integer) As List(Of Mascota)
    GETIISTMASC = New List(Of Mascota)
    Dim SENTENCIA As String
    SENTENCIA = "SELECT * FROM TBMASCOTA"
    If ID <> 0 Then
        SENTENCIA &= " WHERE CLIENTE=" & ID & ";"
    End If
    If ObjDatos.BoolConsulta(Dr, SENTENCIA) Then
        While Dr.Read
            GETIISTMASC.Add(New
Mascota(ObjDatos.GetValorInt(Dr("IDMASCOTA")),
ObjDatos.GetValorInt(Dr("CLIENTE")),    ObjDatos.GetValorInt(Dr("TIPOID")),
ObjDatos.GetValor(Dr("NOMBRE")),        CType(Dr("SEXO"),    Boolean),

```

```

ObjDatos.GetValor(Dr("OBSERVACION")),
CType(ObjDatos.GetValor(Dr("FECHANAC")), Date)))
    End While
    Dr.Close()
End If
End Function

```

```

#End Region
End Class

```

### **CLASE MEDICAMENTO**

```

Public Class Medicamento

```

```

#Region "Atributos"

```

```

    Public IDMedicamento As Integer

```

```

    Public Medicamento As String

```

```

    Public Observacion As String

```

```

    Public Descripcion As String

```

```

#End Region

```

```

#Region "Cosnstructor"

```

```

#End Region

```

```

#Region "Metodos"

```

```

    Public Function GetMedicamento() As Boolean

```

```

        If ObjDatos.BoolConsulta(Dr, "") Then

```

```

            While Dr.Read()

```

```

                Medicamento = ObjDatos.GetValor(Dr("Medicamento"))

```

```

                Observacion = ObjDatos.GetValor(Dr("Observacion"))

```

```

                Descripcion = ObjDatos.GetValor(Dr("Descripcion"))

```

```

            End While

```

```

            Dr.Close()

```

```

            GetMedicamento = True

```

```

        End If

```

```

End Function

Public Function SaveMedicamento() As Boolean
    IDMedicamento = ObjDatos.SigID("IDMEDICAMENTO",
"TBMEDICAMENTO")
    If IDMedicamento > 0 Then
        If ObjDatos.IntEjecutar("INSERT INTO TBMEDICAMENTO VALUES(" &
IDMedicamento & "," _
& Medicamento & "," & Observacion & "," & Descripcion
& ",1);") > -1 Then SaveMedicamento = True
    End If
End Function

#End Region
End Class

```

## **CLASE PERSONA**

```

Public Class Persona
#Region "Atributos"
    Public Cedula As String
    Public Nombre As String
    Public ApellidoP As String
    Public ApellidoM As String
    Public Direccion As String
    Public Telefono As String
    Public IDCiudad As Integer
    Public Masculino As Boolean
#End Region
#Region "Constructores"
    Public Sub New(ByVal ID As String)

    End Sub
    Public Sub New()

```

```

End Sub
#End Region

#Region "Métodos"

Public Function GetPersona() As Boolean
    If ObjDatos.BoolConsulta(Dr, "SELECT * FROM TBPERSONA WHERE
CEDULA="" & Cedula & "";"") Then
        While Dr.Read
            Nombre = ObjDatos.GetValor(Dr("NOMBRES"))
            ApellidoP = ObjDatos.GetValor(Dr("APELLIDOP"))
            ApellidoM = ObjDatos.GetValor(Dr("APELLIDOM"))
            Direccion = ObjDatos.GetValor(Dr("DIRECCION"))
            Telefono = ObjDatos.GetValor(Dr("TELEFONO"))
            Masculino = CType(Dr("SEXO"), Boolean)
        End While
        Dr.Close()
        GetPersona = True
    End If
End Function

Public Function BoolCedula(ByVal CI As String) As Boolean
    If CI.Trim.Length = 10 Then
        Dim a(9) As Integer
        For i As Integer = 0 To 9
            a(i) = Integer.Parse(CI(i).ToString())
        Next
        For cont As Integer = 0 To 9 Step 2
            a(cont) = a(cont) * 2
        Next
        For conta As Integer = 1 To 9 Step 2
            a(conta) = a(conta) * 1
        Next
    End If
End Function

```

```

For ii As Integer = 0 To 9
    If a(ii) >= 10 Then
        a(ii) = (a(ii) - 10) + 1
    End If
Next
Dim contaaa As Integer = a(0) + a(1) + a(2) + a(3) + a(4) + a(5) + a(6) +
a(7) + a(8)
For s As Integer = 0 To 90 Step 10
    If contaaa > (s) And contaaa <= (s + 10) Then
        contaaa = (s + 10) - contaaa
    End If
Next
If contaaa = a(9) Then
    BoolCedula = True
End If
End If
End Function

```

```

Public Function SavePersona() As Boolean
    If ObjDatos.IntEjecutar("INSERT INTO TBPERSONA VALUES('" & Cedula
& "," & IDCiudad & "," & Nombre & "," _
        & ApellidoP & "," & ApellidoM & "," & Direccion & "," _
        & Telefono & "," & IIf(Masculino, 1, 0) & ",1);") > -1 Then
SavePersona = True
    End Function
#End Region
End Class

```

## **CLASE PROVINCIA**

```

Public Class Provincia
#Region "Atributos"
    Public IDProvincia As Integer

```

```

    Public Provincia As String
#End Region
#Region "Constructor"
    Public Sub New()

        End Sub
#End Region
#Region "Métodos"
    Public Function GetProvincia() As Boolean

        End Function

    Public Function GetTotalProvincia() As List(Of List(Of String))
        GetTotalProvincia = New List(Of List(Of String))

        End Function

#End Region
End Class

```

### **CLASE DE TIPO DE MASCOTA**

```

Public Class TipoMascota
#Region "Atributos"
    Private IDTipo As Integer
    Public Tipo As String
    Public Observacion As String
#End Region

#Region "Metodos"
    Public Function SaveTipo() As Boolean
        IDTipo = ObjDatos.SigID("IDTIPO", "TBTIPOMASCOTA")
        If IDTipo > 0 Then

```

```

        If ObjDatos.IntEjecutar("INSERT INTO TBTIPOMASCOTA VALUES(" &
IDTipo _
                & "," & Tipo & "," & Observacion & ",1);") > -1 Then
SaveTipo = True
        End If
    End Function
#End Region
End Class

```

### **CLASE USUARIO**

```

Public Class Usuario : Inherits Persona
#Region "Atributos"
    Private Usuario As String
    Private Password As String
#End Region

#Region "Constructor"
    Public Sub New(ByVal usr As String, ByVal clave As String)
        Usuario = usr : Password = clave
    End Sub
#End Region

#Region "METODOS"
    Public Function ComprobarUSr() As Boolean
        If ObjDatos.BoolConsulta(Dr, "SELECT * FROM TBUSUARIO WHERE
NICK=" & _
                & Usuario & " AND CLAVE =" & Password & ";;") Then
            If Dr.Read Then ComprobarUSr = True
            Dr.Close()
        End If
    End Function
#End Region
End Class

```

## FOTOS DEL GERENTE PROPIETARIO

