

**UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO**  
**FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**



**“SISTEMA DE INFORMACIÓN PARA EL REGISTRO GENEALÓGICO  
DE ALPACAS - AJOYANI - 2015”**

**TESIS**

**PRESENTADA POR:**

**Bach. ERNESTO ALANOCA ANCHAPURI**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO ESTADÍSTICO E INFORMÁTICO**

**PUNO – PERÚ**

**2015**



**UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO**  
**FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**



**SISTEMA DE INFORMACIÓN PARA EL REGISTRO GENEALÓGICO DE  
ALPACAS - AJOYANI - 2015**

**TESIS**

Presentada por:

**Bach. ERNESTO ALANOCA ANCHAPURI**

A la Coordinación de Investigación de la Facultad de Ingeniería Estadística e  
Informática de la Universidad Nacional del Altiplano – Puno, para optar el Título  
Profesional de:

**INGENIERO ESTADÍSTICO E INFORMÁTICO**

**APROBADA POR:**

**PRESIDENTE**

:

  
Dr. EDGAR ELOY CARPIO VARGAS

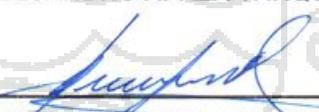
**PRIMER MIEMBRO**

:

  
M.Sc. ALEJANDRO APAZA TARQUI

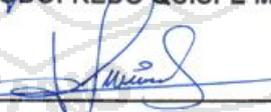
**SEGUNDO MIEMBRO**

:

  
M.Sc. GODOFREDO QUISPE MAMANI

**DIRECTOR DE TESIS**

:

  
Dr. Sc. PERCY HUATA PANCA

**ASESOR DE TESIS**

:

  
Ing. JOHN FERNANDO MAMANI BELLIDO

ÁREA : Informática

TEMA : Sistema de información

## DEDICATORIAS

*A Dios y Señor, por ser nuestro creador,  
amparo, fortaleza y la luz que ilumina  
nuestras vidas cuando más lo  
necesitamos y por hacer palpable su amor  
a través de cada uno de los que nos rodea.*

*Con especial afecto y aprecio a mi  
querida Madre: Paula Segunda, por su  
constante apoyo moral y comprensión  
en todo momento de mi vida que  
impulsaron para la culminación de mi  
profesión y que hoy está al lado de  
nuestro señor.*

*Con especial cariño a mis queridos  
hermanos y hermanas: Teodora,  
Magdalena, Francisco Javier, Hugo y José  
Luis por haberme brindado su apoyo en  
los momentos más difíciles de estudio.*

**Ernesto**

## AGRADECIMIENTOS

### **Mi especial reconocimiento y gratitud:**

A Dios, por bendecirme, guiar mis pasos e iluminar mi camino para seguir adelante en el transcurso de mi vida.

A la Universidad Nacional del Altiplano, alma mater de la Región Puno y a la Escuela Profesional de Ingeniería Estadística e Informática, institución que encamino mi formación profesional.

A los Docentes y Maestros de la Escuela Profesional de Ingeniería Estadística e Informática, quienes contribuyeron en la preparación y culminación de mi carrera profesional los cuales me brindaron sus sabios conocimientos y experiencia profesional en mi formación como Ingeniero.

A mis familiares, amigos y compañeros, por haberme apoyado en todo momento, y sus alientos que hacen merecedor de mis agradecimientos más sinceros.

## ÍNDICE

DEDICATORIAS.....	i
AGRADECIMIENTOS .....	ii
ÍNDICE .....	iii
RESUMEN.....	ix
ABSTRACT .....	x
INTRODUCCIÓN.....	xi
<b>CAPÍTULO I PLAN DE INVESTIGACIÓN .....</b>	<b>1</b>
1.2 EL PROBLEMA.....	1
1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN .....	3
1.4 OBJETIVO GENERAL Y ESPECÍFICOS .....	3
1.5 HIPÓTESIS.....	4
1.6 LIMITACIONES DE LA INVESTIGACIÓN .....	4
<b>CAPÍTULO II MARCO TEÓRICO .....</b>	<b>5</b>
2.1 ANTECEDENTES DE LA INVESTIGACIÓN .....	5
2.2 BASE TEÓRICA.....	6
2.3 DEFINICIÓN DE TÉRMINOS BÁSICOS .....	45
2.4 OPERACIONALIZACIÓN DE VARIABLES .....	48
<b>CAPÍTULO III MATERIALES Y MÉTODOS.....</b>	<b>49</b>
3.1 POBLACIÓN .....	49
3.3 DISEÑO DE LA MUESTRA.....	52
3.4 MÉTODO DE RECOPIACIÓN DE DATOS.....	52
3.5 MÉTODO DE ANÁLISIS DE DATOS .....	53
3.6 MATERIAL EXPERIMENTAL.....	53



**CAPÍTULO IV RESULTADOS Y DISCUSIÓN** ..... 62

4.1 RESULTADOS Y DISCUSIÓN..... 62

4.2 PRUEBAS DEL SOFTWARE ..... 75

**CONCLUSIONES**..... 79

**RECOMENDACIONES Y SUGERENCIAS**..... 80

**BIBLIOGRAFÍA**..... 81

**BIBLIOGRAFÍA WEB** ..... 83

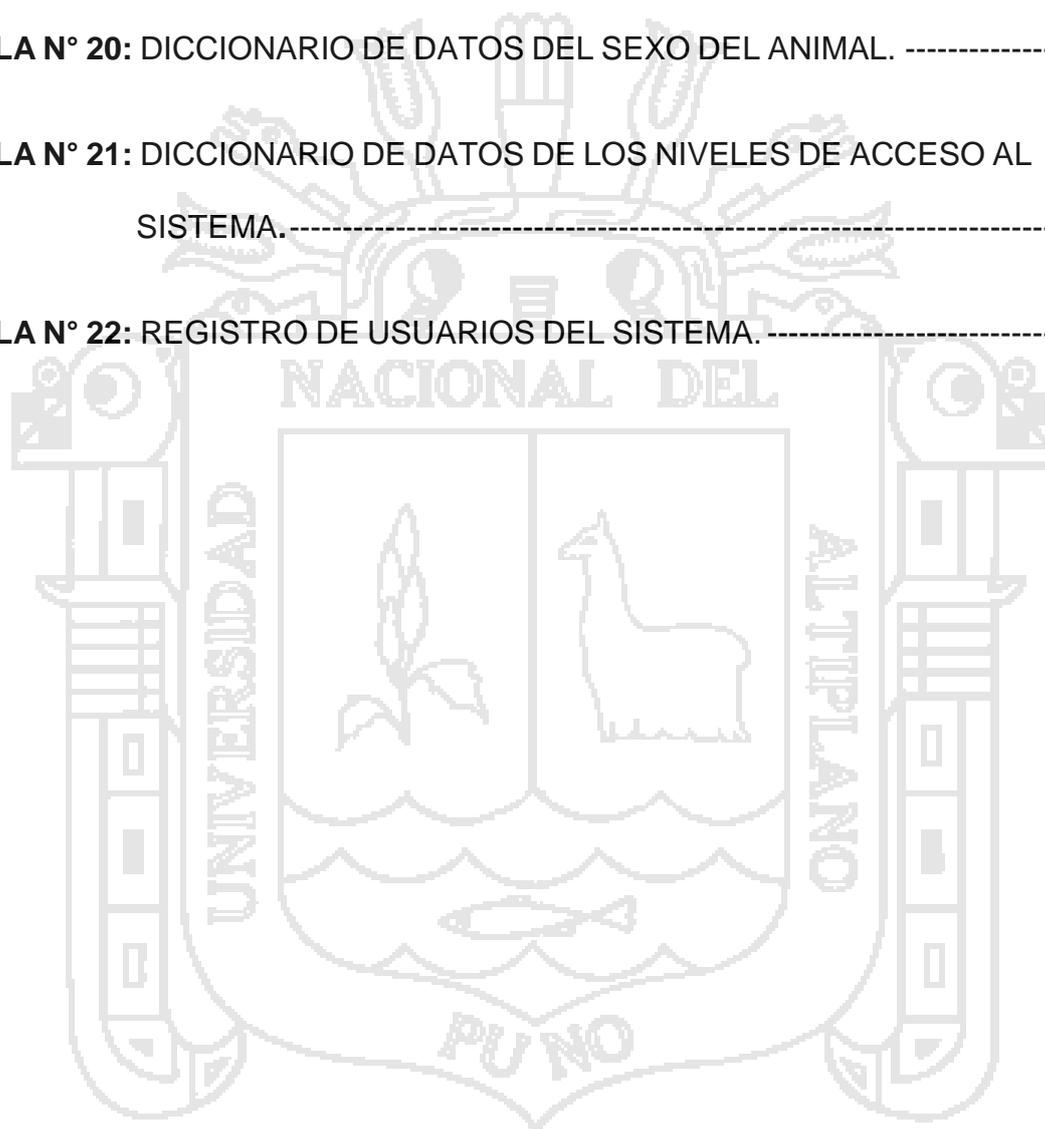
**ANEXOS** ..... 84



## ÍNDICE DE TABLAS

<b>TABLA N° 1:</b> TABLA DE SIN NIVEL DE FORMALIZACIÓN. ....	22
<b>TABLA N° 2:</b> TABLA DE PRIMER NIVEL DE FORMALIZACIÓN. ....	23
<b>TABLA N° 3:</b> TABLA DE SEGUNDO NIVEL DE FORMALIZACIÓN. ....	24
<b>TABLA N° 4:</b> TABLA DE TERCER NIVEL DE FORMALIZACIÓN. ....	25
<b>TABLA N° 5:</b> TABLA DE CUARTO NIVEL DE FORMALIZACIÓN. ....	27
<b>TABLA N° 6:</b> OPERACIONALIZACION DE VARIABLES. ....	48
<b>TABLA N° 7:</b> TABLA FICHAS ISO - 9126. ....	77
<b>TABLA N° 8:</b> TABLA DE COMPARACIÓN DE TIEMPOS ....	78
<b>TABLA N° 9:</b> FICHA DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO ....	85
<b>TABLA N° 10:</b> USUARIOS ACTIVOS E INACTIVOS. ....	87
<b>TABLA N° 11:</b> REGISTRO DE DATOS DE ALPACAS. ....	87
<b>TABLA N° 12:</b> DICCIONARIO DE DATOS DE CALIDAD. ....	87
<b>TABLA N° 13:</b> DICCIONARIO DE DATOS DE CALIDAD CALIFICATIVO POR CABEZA. ....	88
<b>TABLA N° 14:</b> DICCIONARIO DE DATOS DE COLOR DE FIBRA. ....	88
<b>TABLA N° 15:</b> DICCIONARIO DE DATOS DE DENSIDAD DE FIBRA. ....	89
<b>TABLA N° 16:</b> DICCIONARIO DE DATOS DE ESTADO DE PREÑEZ. ....	89

<b>TABLA N° 17:</b> DICCIONARIO DE DATOS DE ESTADO DE EDAD.....	90
<b>TABLA N° 18:</b> DICCIONARIO DE DATOS DE ESTADO DE FINURA.....	90
<b>TABLA N° 19:</b> DICCIONARIO DE DATOS DE ESTADO DE SECTORES DEL DISTRITO DE AJOYANI. ....	90
<b>TABLA N° 20:</b> DICCIONARIO DE DATOS DEL SEXO DEL ANIMAL. ....	91
<b>TABLA N° 21:</b> DICCIONARIO DE DATOS DE LOS NIVELES DE ACCESO AL SISTEMA.....	91
<b>TABLA N° 22:</b> REGISTRO DE USUARIOS DEL SISTEMA.....	92



## ÍNDICE DE GRÁFICOS

<b>GRÁFICO N° 1:</b> PROCESO PRINCIPALES DEL CICLO DE VIDA CLÁSICO. ....	38
<b>GRÁFICO N° 2:</b> DIAGRAMA DE CASOS DE USO.....	40
<b>GRÁFICO N° 3:</b> GRÁFICO DE DIAGRAMA DE SECUENCIAS.....	41
<b>GRÁFICO N° 4:</b> DIAGRAMA DE COLABORACIONES. ....	42
<b>GRÁFICO N° 5:</b> DIAGRAMA DE CLASES.....	43
<b>GRÁFICO N° 6:</b> PROCESO DE PROGRAMACIÓN XTREMA. ....	53
<b>GRÁFICO N° 7:</b> PROCESO PRINCIPALES DEL CICLO DE VIDA CLÁSICO. ....	55
<b>GRÁFICO N° 8:</b> CICLO DE VIDA DEL SOFTWARE DEL SISTEMA.....	62
<b>GRÁFICO N° 9:</b> CICLO DE VIDA DEL SOFTWARE DEL SISTEMA.....	64
<b>GRÁFICO N° 10:</b> DIAGRAMA LÓGICO DE COMPONENTES. ....	65
<b>GRÁFICO N° 11:</b> DIAGRAMA LÓGICO DE SECUENCIAS. ....	66
<b>GRÁFICO N° 12:</b> DIAGRAMA LÓGICO DE ESTADOS. ....	67
<b>GRÁFICO N° 13:</b> MODELADO CONCEPTUAL PARA EL DISEÑO DEL SOFTWARE. .....	68
<b>GRÁFICO N° 14:</b> ESTRUCTURACIÓN DE MODELO LÓGICO. ....	69
<b>GRÁFICO N° 15:</b> MODELO FÍSICO DE BASE DE DATOS. ....	70
<b>GRÁFICO N° 16:</b> VENTANA DE EDITOR DE DREAM WEABER.....	71
<b>GRÁFICO N° 17:</b> USO DE LENGUAJE DE PROGRAMACIÓN. ....	72
<b>GRÁFICO N° 18:</b> USO DE GESTOR DE BASE DE DATOS. ....	73
<b>GRÁFICO N° 19:</b> CONECTIVIDAD ABIERTA A BASE DE DATOS. ....	74
<b>GRÁFICO N° 20:</b> VENTANA PRINCIPAL DE ENTRADA. ....	94
<b>GRÁFICO N° 21:</b> VENTANA DE LOGUIN GENERAL DEL SISTEMA. ....	95
<b>GRÁFICO N° 22:</b> VENTANA DE LOGUIN ADMINISTRADOR. ....	96
<b>GRÁFICO N° 23:</b> VENTANA DE LOGUIN REGISTRADOR. ....	96
<b>GRÁFICO N° 24:</b> ADMINISTRADOR DEL SISTEMA EN BALANCE GENERAL. ...	97

<b>GRÁFICO N° 25:</b> ADMINISTRADOR DEL SISTEMA EN USUARIOS. ....	97
<b>GRÁFICO N° 26:</b> ADMINISTRADOR DEL SISTEMA EN EDICIÓN DE USUARIOS.	98
<b>GRÁFICO N° 27:</b> VENTANA DE BÚSQUEDA NOMBRE DE PROPIETARIO. ....	99
<b>GRÁFICO N° 28:</b> VENTANA DE BÚSQUEDA POR DNI. ....	99
<b>GRÁFICO N° 29:</b> REPORTES DEL SISTEMA EN FORMATO PDF DE DIMENSIONES. ....	100
<b>GRÁFICO N° 30:</b> REPORTES DEL SISTEMA EN FORMATO PDF DE FINURA.	100
<b>GRÁFICO N° 31:</b> REPORTES DEL SISTEMA EN FORMATO PDF DE GENOTIPO. .....	101
<b>GRÁFICO N° 32:</b> VENTANA DE REGISTRO DE NUEVOS DATOS. ....	102
<b>GRÁFICO N° 33:</b> VENTANA DE EDICIÓN DE DATOS. ....	103



## RESUMEN

El principal objetivo de la presente Tesis se basó en la implementación de un sistema de información, para el registro genealógico de alpacas, específicamente para el seguimiento y control de la producción de alpacas en diferentes sectores, con la finalidad de lograr el mejor desempeño y agilizar el proceso de registro y emisión de reportes, garantizando un mejor manejo de la información de las alpacas en el distrito de Ajoyani.

La tesis estuvo enmarcado en el tipo de investigación proyectiva, fundamentada a nivel comprensivo con un diseño de fuente mixta (documental y campo). Se emplearon una serie de técnicas e instrumentos de recolección de datos, específicamente el análisis de fuentes documentales, la observación directa y las entrevistas no estructuradas. Se utilizó como guía una estructura operativa mixta, combinando la metodología de Reingeniería de Procesos y Programación Extrema. Adicionalmente, para el desarrollo de la aplicación se utilizaron diversas tecnologías como el lenguaje de programación PHP, el sistema manejador de base de datos "MySQL". y el servidor Web Apache.

De esta manera se pudo concluir que con la implantación del nuevo sistema se genera un mejor desempeño de las labores en la gerencia de desarrollo agropecuario en cuanto a la realización de los procesos en forma automatizada.

**Palabras Claves:** Sistema de Información, registro genealógico, alpaca, lograr, desempeño, investigación proyectiva, tecnología, Reingeniería de proceso, PHP, MySQL.

## ABSTRACT

The main objective of this thesis is based on the implementation of an information system for the genealogical register of alpacas, specifically for the monitoring and control of alpaca production in different sectors, in order to achieve the best performance and speed the registration process and issuance of reports, ensuring better management of information alpacas in the district of Ajoyani.

The thesis was framed in the type of investigation projective, based to understanding level with a design of mixed source (documentary and field). They employed a series of technicians and instruments of collection of data, specifically the analysis of documentary sources, the direct observation and the no structured interviews. It used as it guides a mixed operative structure, combining the methodology of Reengineering of Processes and Extreme Programming. Additionally, for the development of the application used diverse technologies like the programming language PHP, the system the handler system database MySQL. And the server Web Apache.

Thus it can be concluded that with the implementation of the new system better performance of the tasks in the management of agricultural development regarding the implementation of automated processes is generated.

**Keywords:** Information System, generations, alpaca, achieve performance, projective research, technology, process reengineering, PHP, MySQL.

## INTRODUCCIÓN

El avance de la tecnología, principalmente en el campo de la informática trae como consecuencia descubrimientos y adelantos muy significativos para la humanidad. Por lo que el hombre no puede estar al margen de ello, menos la Municipalidad de Distrital de Ajoyani, la cual debe estar dentro de las exigencias técnicas, científicas y modernidad. En realidad, las aplicaciones del sistema de información tienen su origen en casi todas las áreas de un proyecto y están relacionados con todos los problemas de la organización en la vida real. Los sistemas de información ayudan a los administradores y al personal a analizar un problema visualizar cuestiones complejas y toma de decisiones. Un sistema de información no solo es un programa informático sino es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa.

En la actualidad con el proceso de registro de datos y con el avance de la tecnología se tiene tendencias a nuevos métodos de registro para agilizar el proceso de emisión de datos genealógicos, así como también al uso de diversas herramientas para el mismo fin.

En el Primer capítulo de la tesis se plantea toda la problemática del registro genealógico de alpacas en la Municipalidad de Distrital de Ajoyani, para lo cual el objetivo principal es la implementación de un sistema de información que debe estar dentro de las exigencias técnicas, científicas y modernidad. Recalcando que las aplicaciones del sistema de información tienen su origen en casi todas las áreas de un proyecto y están relacionados con todos los problemas de la organización en la vida real.

En el segundo capítulo se explica de los antecedentes, y la base teórica empleada, dando énfasis en que los sistemas de información ayudan a los administradores y al personal a analizar un problema visualizar cuestiones complejas y toma de decisiones. Un sistema de información no solo es un programa informático sino es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa.

En el tercer capítulo se emplearon una serie de técnicas e instrumentos de recolección de datos, específicamente el análisis de fuentes documentales, la observación directa y las entrevistas no estructuradas.

Cuarto capítulo se obtiene los resultados y la discusión del sistema de información automatizada para el registro, selección, clasificación y emisión de resultados en tiempo real, donde se concluye que todo criador de camélidos sudamericanos del distrito de Ajoyani tiene conocimiento de las bases de empadre controlado y registro genealógico. Para ser partícipes del proceso de selección de animales con alto potencial en finura tanto en carne y fibra.

## CAPÍTULO I

### PLAN DE INVESTIGACIÓN

#### 1.1 EL PROBLEMA

La continua evolución de la tecnología informática al servicio de la humanidad y mejorar la calidad de vida y factibilidad de intercambio de información y por ello, el **SISTEMA DE INFORMACIÓN PARA EL REGISTRO GENEALÓGICO DE ALPACAS DEL DISTRITO DE AJOYANI - 2015**, les dará soluciones productivas y efectivas para los administrativos y comuneros de dicha municipalidad.

Se tiene conocimiento que todo funcionario del distrito de Ajoyani debe saber manipular el software y con una previa capacitación le será muy fácil manejar el software implementado. El presente proyecto es de suma importancia porque será utilizado para modernizar la ganadería en el distrito, por ser importante y necesario implementar el sistema integral de registro de datos como finura, calce, cabeza, riso, preñez, densidad y raza el sistema es una herramienta principal de trabajo para la emisión inmediata de certificados de los datos registrados.

En el distrito de Ajoyani no existe un sistema de registro integral de datos de características genealógicas de alpacas, que servirá como una herramienta principal para los administrativos de la oficina de desarrollo agropecuario de su gobierno local para una eficiente respuesta y tiempo de respuesta de registro automático.

En los últimos años el crecimiento de la población de alpacas en el distrito de Ajoyani ha sido notorio y como consecuencia trae una mayor demanda en la atención administrativa. Esto origina el incremento del gasto administrativo durante su apertura y causa problemas al momento de la atención personalizada pues se tendrán que registrar las características genealógicas de las alpacas.

Actualmente en el distrito de Ajoyani afronta problemas a la hora de realizar los registros y reportes de certificados provocando así aglomeraciones e incomodidades a los trabajadores de la oficina de desarrollo agropecuario debido a que todo el proceso de registro de finura, calce, cabeza, riso, preñez, densidad y raza se realiza de forma manual.

El registro genealógico es un proceso de almacenamiento de datos característicos propios de aspectos genéticos, fechas de nacimiento, árbol genealógico entre otros. Este proceso en la actualidad viene tomando más fuerza porque es necesario para determinar la ubicación y su potencialidad en rasgos genéticos dentro del distrito y de esta manera tomar decisiones para la elaboración de cualquier proyecto agrícola - ganadero.

En la actualidad el distrito de Ajoyani opta por automatizar todo el sistema de información de alpacas para así de esa forma brindar una mejor atención a los usuarios frente a esto se plantea la siguiente ¿En qué medida el desarrollo del “sistema de información para el registro genealógico de alpacas - ajoyani - 2015” mejoró el manejo de registro y reporte genealógico de alpacas, como finura, calce, cabeza, riso, preñez, densidad, raza en el distrito de Ajoyani?.

## 1.2 JUSTIFICACIÓN DE LA INVESTIGACIÓN

La ganadería en el distrito de Ajoyani es la principal actividad económica por ende es base la fuente de ingreso de toda la población, por alcanzar una mejor calidad de vida, la municipalidad distrital de Ajoyani debe implementar el sistema de información de registro de alpacas, de esa manera alcanzar los estándares nacionales e internacionales, por ello se esmera por brindar una mejor calidad en la atención a la población rural, actualmente no cuenta con ningún sistema de registro, se ha notado una deficiencia en cuanto a los registros y solicitudes de documentos.

Mediante la implementación de sistema genealógico permitirá a la institución ser más eficiente, ahorrar tiempo y dinero para los usuarios también se les acortará el tiempo y el molesto trámite en espera.

## 1.3 OBJETIVO GENERAL Y ESPECÍFICOS

### **OBJETIVO GENERAL:**

Implementar un sistema de información para el registro genealógico de datos de alpacas para la Municipalidad Distrital de Ajoyani.

**OBJETIVOS ESPECÍFICOS:**

- 1 Diseñar un sistema multiplataforma para el registro de datos genealógicos de alpacas para la Municipalidad Distrital de Ajoyani.
- 2 Implementar componentes del sistema para el registro de datos genealógico de alpacas de gestión, comunicación y control.
- 3 Evaluar la eficiencia del sistema en el proceso de registro de datos y emisión de documentos veterinarios.

**1.4 HIPÓTESIS**

La implementación de un Sistema de registro de datos genealógicos mejora la rapidez para determinar con la ubicación, precio y aspectos genéticos de animal, así hacer la emisión de documentos y reporte de datos requeridos de sus animales por ganaderos que hayan registrado sus ganados en la Municipalidad Distrital de Ajoyani.

**1.5 LIMITACIONES DE LA INVESTIGACIÓN**

- Falta de laptops para el registro de datos en los fundos alejados.
- El sistema sólo hace reporte de fichas de datos genealógicos como, finura, calce, cabeza, rizo, preñez, densidad y raza.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1 ANTECEDENTES DE LA INVESTIGACIÓN

Habiendo realizado la búsqueda de información acerca de antecedentes respecto a nuestra investigación en esta municipalidad no se encontró ningún estudio de esta índole.

Sin embargo, existen varios sistemas de registro de alpacas y se encontraron los siguientes:

- ✓ Contreras R., Mario (2007). Diseño e implementación de un sistema llamado SISRAL (Sistema de Registro de Alpacas) de sistemas de registro de datos, tiene el objetivo de registrar los datos como finura, cabeza, calce se observa que al implementar el sistema las zonas afectadas por el beneficio están mejorando su calidad de ganados.
- ✓ Ricardo A., Cuevas (2008). Sistema Alpaquero - Universidad de Cerro de Pasco. Es un sistema experimental que funcionó para su centro experimental se concluye que al utilizar el sistema hay un control efectivo de árbol genealógico.

## 2.2 BASE TEÓRICA

Según: Bertalanffy, Ludwing. Teoría general de sistemas. Hoy en día las tecnologías de información y comunicaciones (TICs) se han convertido en asistente inseparable del hombre, permitiéndole tener una visión diferente de cómo se deben de hacer las cosas. En el campo de trabajo institucional el administrativo o personal o funcionario puede contar con diferentes mecanismos de acceso a sus datos sin recurrir exclusivamente a la oficina y haciendo largas esperas. En este punto el sistema registro de datos ofrece como una gran herramienta que cumple la tarea fundamental que desarrolla un funcionario de la oficina de desarrollo agropecuario y sirve consulta inmediata para el personal trabajador en la oficina y el poblador la diferencia de que este genera reportes instantáneos en vez de varios días que se demora en emitir dichos documentos, además de darle estas posibilidades también es capaz de establecer una comunicación o un reporte de datos requeridos por un funcionario de la oficina de desarrollo agropecuario de la Municipalidad Distrital de Ajoyani.

Este sistema de registro de datos es una propuesta de alta calidad de servicio apoyada en las TICs; el cual estimula el trabajo instantáneo y eficiencia en el sistema informático de la institución pública lo cual esta con fundamentación científica. En el presente trabajo se encontró algunos aspectos relacionados con la implementación del sistema de registro de datos, dividido en secciones donde se dan a conocer los elementos teóricos, conceptuales y tecnológicos que sirvieron de base para el desarrollo del proyecto; así como las etapas del proceso de desarrollo.

Las razones establecidas para esta investigación nos motivan a responder los siguientes aspectos teóricos:

### **2.2.1 Sistema**

Un sistema es un conjunto de funciones, virtualmente referenciada sobre ejes, bien sean estos reales o abstractos. Se le puede definir también como un conjunto de elementos dinámicamente relacionados formando una actividad para alcanzar un objetivo operando sobre datos, energía o materia para proveer información.

### **2.2.2 Gestión**

Generalmente una persona que gestiona es aquella que mueve todos los hilos necesarios para que ocurra determinada cosa o para que se logre determinado propósito.<sup>1</sup>

### **2.2.3 Software**

Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático, comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

---

1.-Capacidad de llegar al éxito, administrando adecuadamente los recursos.

#### **2.2.4 Ingeniería de Software**

Es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos (**software**). Esta disciplina trasciende la actividad de programación, que es la actividad principal a la hora de crear un software. El ingeniero de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado y con el presupuesto previsto.

Pressman presupone que el sistema de software a realizar cumpla con la misión encomendada, satisfaga las expectativas del usuario, etc. Si comparamos esta definición con la ingeniería del sistema interior, vemos que la Pressman está enfocada al proceso de desarrollo y no tanto a la tecnología con la que se desarrolla; para él es más una ingeniería de proceso y no tanto de producto de su análisis se desprende que la ingeniería de sistemas de software es una especialización de la ingeniería de sistemas cuya creciente importancia está ligada a la de los sistemas de software en nuestra sociedad. (Gonzalo LEON SERRANO, "Ingeniería de sistemas de software" 4ta Edición, mayo - 1996).

#### **2.2.5 Tecnología de software**

Definimos tecnología de software como un conjunto integrado de notaciones herramientas y métodos, basados en unos sólidos fundamentos que permiten el desarrollo de un producto software en un contexto organizativo. (Alvaro PEÑA GONZALES, "Ingeniería del software en entornos de SL" 1ta Edición, marzo - 2005).

### 2.2.6 Orientación de objetos

La orientación de objetos es un paradigma más de programación en el que un sistema se expresa como un conjunto de objetos que interactúan entre ellos. Un paradigma de programación nos proporciona una abstracción del sistema real a algo que podemos programar y ejecutar, y puede decirse que el tipo de abstracción está directamente relacionada con los problemas que puede resolver o al menos con la facilidad con que podremos resolverlos. Mientras que el lenguaje ensamblador es una abstracción del procesador, podríamos decir que otros lenguajes de programación como Basic o C, son abstracciones del propio lenguaje ensamblador. E. KENDALL Kenneth y E. KENDALL julie, "Análisis y Diseño de Sistemas" 6ta Edición, 2005).

### 2.2.7 Análisis Y Diseño De Sistemas Orientado A Objetos

Pressman, Roger S. (2003). Análisis y diseño orientado a objetos (ADOO) es un enfoque de la ingeniería de software que modela un sistema como un grupo de objetos que interactúan entre sí. Este enfoque representa un dominio en términos de conceptos compuestos por verbos y sustantivos, clasificados de acuerdo a su dependencia funcional. En este método de análisis y diseño se crea un conjunto de modelos utilizando una notación acordada como, por ejemplo, el lenguaje unificado de modelado (UML). ADOO aplica técnicas de modelado de objetos para analizar los requerimientos para un contexto.

Por ejemplo, un sistema de negocio, un conjunto de módulos de software y para diseñar una solución para mejorar los procesos involucrados. Las metodologías

de análisis y diseño más modernas son casos de uso guiados a través de requerimientos, diseño, implementación, pruebas, y despliegue.

### 2.2.8 Sistemas De Información Web (SIW)

Los SIW son una clase diferente de sistema de información que requiere que la gente piense en ellos en forma distinta a como lo hacen de los sistemas tradicionales ya que requieren de nuevos enfoques para su diseño y desarrollo. Por otra parte, los SIW son primeramente sistemas de información y luego sistemas Web, por lo que su desarrollo debería usar la misma disciplina requerida para construir sistemas de información no Web exitosos y de calidad. De hecho, muchos de los aspectos de los SIW son producto de aplicar a las Web estrategias de ingeniería de software estándar.

Aunque la Web ha simplificado la liberación de información, el desarrollo de SIW es igualmente; sino más, desafiante que el desarrollo de sistemas de información. Los administradores o desarrolladores pueden confundirse fácilmente por herramientas que dan la impresión de que el verse bien es la única importancia de las páginas Web. Sin la suficiente planificación y diseño, las organizaciones pueden, rápidamente, llegar a tener una pesadilla en lugar de un SIW. (Chacón N. "Integración de software heterogéneo a través de sistemas de información Web", 1999).

### 2.2.9 World Wide Web

En informática, la **World Wide Web (WWW)** o **Red informática mundial** es un sistema de distribución de información basada en hipertexto o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario

visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces. ([http://es.wikipedia.org/wiki/World\\_Wide\\_Web](http://es.wikipedia.org/wiki/World_Wide_Web)).

La Web fue creada alrededor de 1989 por el inglés Tim Berners-Lee con la ayuda del Belga Robert Cailliau mientras trabajaban en el CERN en Ginebra, Suiza, y publicado en 1992. Desde entonces, Berners-Lee ha jugado un papel activo guiando el desarrollo de estándares.

#### **2.2.10 Sistema de Información**

Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema. Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

##### **a. Entrada de Información**

Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas. Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de

diskette, los códigos de barras, los escáneres, la voz, los monitores sensibles al tacto.

#### **b. Almacenamiento de Información**

El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

#### **c. Procesamiento de Información**

Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

#### **d. Salida de Información**

La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son

las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficados y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo en este caso, también existe una interface automática de salida.

### 2.2.11 Bases de datos

Es un almacén que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

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

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

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

### 2.2.12 Definición De Base De Datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Características:

- a. Independencia lógica y física de los datos.

- b. Redundancia mínima.
- c. Acceso concurrente por parte de múltiples usuarios.
- d. Integridad de los datos.
- e. Consultas complejas optimizadas.
- f. Seguridad de acceso y auditoría.
- g. Respaldo y recuperación.

### **2.2.13 Sistema de Gestión de Base de Datos (SGBD)**

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.

#### **Ventajas de las bases de datos**

##### **i. Control sobre la redundancia de datos:**

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.

**ii. Consistencia de datos:**

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. ([www.uprb.edu/profesor/.../base\\_de\\_datosventajasdesventajas.html](http://www.uprb.edu/profesor/.../base_de_datosventajasdesventajas.html))

**iii. Compartición de datos:**

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.

**iv. Mantenimiento de estándares:**

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.

**v. Mejora en la integridad de datos:**

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 encargar de mantenerlas.

**vi. Mejora en la seguridad:**

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.

**vii. Mejora en la accesibilidad a los datos:**

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.

**viii. Mejora en la productividad:**

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.

**ix. Mejora en el mantenimiento:**

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.

**x. Aumento de la concurrencia:**

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.

**xi. Mejora en los servicios de copias de seguridad:**

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.

## **Desventajas de las bases de datos**

### **xii. Complejidad:**

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.

### **xiii. Coste del equipamiento adicional:**

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.

### **xiv. Vulnerable a los fallos:**

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).

### **xv. Tipos de Campos**

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).
- ✓ **Auto incrementables:** 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.

#### 2.2.14 Tipos de Base de Datos

Entre los diferentes tipos de base de datos, podemos encontrar los siguientes:

**PostgreSQL, Oracle, Access, Microsoft SQL Server**

**MySql:** 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.

#### **MySql**

Es un sistema administrativo relacional de bases de datos (RDBMS por sus

siglas en inglés Relational Database Management System). Este tipo de bases de datos puede ejecutar desde acciones tan básicas, como insertar y borrar registros, actualizar información ó hacer consultas simples, hasta realizar tareas tan complejas como la aplicación lo requiera.

MySQL es un servidor multi-usuarios muy rápido y robusto de ejecución de instrucciones en paralelo, es decir, que múltiples usuarios distribuidos a lo largo de una red local o Internet podrá ejecutar distintas tareas sobre las bases de datos localizadas en un mismo servidor.

Utiliza el lenguaje SQL (Structured Query Language) que es el estándar de consulta a bases de datos a nivel mundial.

#### **2.2.15 Estructura de una Base de Datos**

Una base de datos, a fin de ordenar la información de manera lógica, posee un orden que debe ser cumplido para acceder a la información de manera coherente. Cada base de datos contiene una o más tablas, que cumplen la función de contener los campos.

Por consiguiente, una base de datos posee el siguiente orden jerárquico:

- ✓ Tablas
- ✓ Campos
- ✓ Registros
- ✓ Lenguaje SQL

El **lenguaje SQL** es el más universal en los sistemas de base de datos. Este

lenguaje nos permite realizar consultas a nuestras bases de datos para mostrar, insertar, actualizar y borrar datos.

### **2.2.16 Normalización de base de datos**

Uno de los factores más importantes en la creación de páginas web dinámicas es el diseño de las Bases de Datos (BD). Si tus tablas no están correctamente diseñadas, te pueden causar un montón de dolores de cabeza cuando tengas de realizar complicadísimas llamadas SQL en el código PHP para extraer los datos que necesitas. Si conoces como establecer las relaciones entre los datos y la normalización de estos, estarás preparado para comenzar a desarrollar tu aplicación en PHP. Si trabajas con MySQL o con Oracle, debes conocer los métodos de normalización del diseño de las tablas en tu sistema de BD relacional. Estos métodos pueden ayudarte a hacer tu código PHP más fácil de comprender, ampliar, y en determinados casos, incluso hacer tu aplicación más rápida.

Básicamente, las reglas de Normalización están encaminadas a eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas. Los cinco pasos progresivos para normalizar, tienes que tener en cuenta que debes crear una BD funcional y eficiente. También se detalla los tipos de relaciones que tu estructura de datos puede tener. Digamos que queremos crear una tabla con la información de usuarios, y los datos a guardar son el nombre, la empresa, la dirección de la empresa y algún e-mail, o bien URL si las tienen. En principio comenzarías definiendo la estructura de una tabla como esta:

### a. Formalización cero

**TABLA N° 1:** Tabla de sin nivel de formalización.

Usuarios				
Nombre	Empresa	Direccion_Empresa	url1	url2
Joe	ABC	1 Work Lane	abc.com	xyz.com
Jill	XYZ	1 Job Street	abc.com	xyz.com

**Fuente:** Wikipedia.com/tablas.

Diríamos que la anterior tabla está en nivel de Formalización Cero porque ninguna de nuestras reglas de normalización ha sido aplicada. Observa los campos url1 y url2 -- ¿Qué haremos cuando en nuestra aplicación necesitemos una tercera url? ¿Quieres tener que añadir otro campo/columna a tu tabla y tener que reprogramar toda la entrada de datos de tu código PHP? Obviamente no, tú quieres crear un sistema funcional que pueda crecer y adaptarse fácilmente a los nuevos requisitos.

### b. Primer nivel de formalización/normalización. (F/N)

1. Eliminar los grupos repetitivos de las tablas individuales.
2. Crear una tabla separada por cada grupo de datos relacionados.
3. Identificar cada grupo de datos relacionados con una clave primaria.

¿Ves que estamos rompiendo la primera regla cuando repetimos los campos url1 y url2? ¿Y qué pasa con la tercera regla, la clave primaria? La regla tres básicamente significa que tenemos que poner un campo tipo contador auto incrementable para cada registro. De otra forma, ¿Qué pasaría si tuviéramos dos usuarios llamados Joe y queremos diferenciarlos? Una vez que aplicáramos el primer nivel de F/N nos encontraríamos con la siguiente tabla:

**TABLA N° 2:** Tabla de primer nivel de formalización.

Usuarios				
userId	nombre	empresa	direccion_empresa	url
1	Joe	ABC	1 Work Lane	abc.com
1	Joe	ABC	1 Work Lane	xyz.com
2	Jill	XYZ	1 Job Street	abc.com
2	Jill	XYZ	1 Job Street	xyz.com

**Fuente:** [Wikipedia.com/tablas/formalización](http://Wikipedia.com/tablas/formalización).

Ahora diremos que nuestra tabla está en el primer nivel de F/N. Hemos solucionado el problema de la limitación del campo url. Pero sin embargo vemos otros problemas...Cada vez que introducimos un nuevo registro en la tabla usuarios, tenemos que duplicar el nombre de la empresa y del usuario. No sólo nuestra BD crecerá muchísimo, sino que será muy fácil que la BD se corrompa si escribimos mal alguno de los datos redundantes. Aplicaremos pues el segundo nivel de F/N.

### c. Segundo nivel de formalización/normalización. (F/N)

1. Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
2. Relacionar estas tablas mediante una clave externa.

Hemos separado el campo url en otra tabla, de forma que podemos añadir más

Usuarios			
userId	Nombre	Empresa	direccion_empresa
1	Joe	ABC	1 Work Lane
2	Jill	XYZ	1 Job Street

en  
el

futuro si tener que duplicar los demás datos. También vamos a usar nuestra clave primaria para relacionar estos campos:

**TABLA N° 3:** Tabla de segundo nivel de formalización.

Urls		
urlId	relUserId	url
1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com

**Fuente:** [sikipedia.com/tablas/formalizacion2](http://sikipedia.com/tablas/formalizacion2).

Vale, hemos creado tablas separadas y la clave primaria en la tabla usuarios, user Id, está relacionada ahora con la clave externa en la tabla urls, rel UserId. Esto está mejor. ¿Pero qué ocurre cuando queremos añadir otro empleado a la empresa ABC? ¿o 200 empleados? Ahora tenemos el nombre de la empresa y su dirección duplicándose, otra situación que puede inducirnos a introducir errores en nuestros datos. Así que tendremos que aplicar el tercer nivel de F/N.

#### **d. Tercer nivel de formalización/normalización. (f/n)**

Eliminar aquellos campos que no dependan de la clave. Nuestro nombre de empresa y su dirección no tienen nada que ver con el campo user Id, así que tienen que tener su propia empresa Id.

**TABLA N° 4:** Tabla de tercer nivel de formalización.

<b>Empresas</b>		
<b>emprld</b>	<b>empresa</b>	<b>direccion_empresa</b>
1	ABC	1 Work Lane
2	XYZ	1 Job Street

<b>Usuarios</b>		
<b>userid</b>	<b>nombre</b>	<b>relEmpresald</b>
1	Joe	1
2	Jill	2

<b>Urls</b>		
<b>urlld</b>	<b>RelUserId</b>	<b>url</b>
1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com

Fuente: [Wikipedia.com/tablas/formalizacion3](http://Wikipedia.com/tablas/formalizacion3).

Ahora tenemos la clave primaria `emprld` en la tabla `empresas` relacionadas con la clave externa `recEmpresald` en la tabla `usuarios`, y podemos añadir 200 usuarios mientras que sólo tenemos que insertar el nombre 'ABC' una vez. Nuestras tablas de `usuarios` y `urls` pueden crecer todo lo que quieran sin

duplicación ni corrupción de datos. La mayoría de los desarrolladores dicen que el tercer nivel de F/N es suficiente, que nuestro esquema de datos puede manejar fácilmente los datos obtenidos de una cualquier empresa en su totalidad, y en la mayoría de los casos esto será cierto. Pero echemos un vistazo a nuestro campo urls - ¿Ves duplicación de datos? Esto es perfectamente aceptable si la entrada de datos de este campo es solicitada al usuario en nuestra aplicación para que teclee libremente su url, y por lo tanto es sólo una coincidencia que Joe y Jill teclearon la misma url. ¿Pero qué pasa si en lugar de entrada libre de texto usáramos un menú desplegable con 20 o incluso más urls predefinidas? Entonces tendríamos que llevar nuestro diseño de BD al siguiente nivel de F/N, el cuarto, muchos desarrolladores lo pasan por alto porque depende mucho de un tipo muy específico de relación, la relación 'varios-con-varios', la cual aún no hemos encontrado en nuestra aplicación.

**e. Cuarto nivel de formalización/normalización. (F/N)**

Antes de definir el cuarto nivel de F/N, veremos tres tipos de relaciones entre los datos: uno-a-uno, uno-con-varios y varios-con-varios. Mira la tabla usuarios en el Primer Nivel de F/N del ejemplo de arriba. Por un momento imaginamos que ponemos el campo url en una tabla separada, y cada vez que introducimos un registro en la tabla usuarios también introducimos una sola fila en la tabla urls. Entonces tendríamos una relación uno-a-uno: cada fila en la tabla usuarios tendría exactamente una fila correspondiente en la tabla urls. Para los propósitos de nuestra aplicación no sería útil la normalización.

Ahora mira las tablas en el ejemplo del Segundo Nivel de F/N. Nuestras tablas permiten a un sólo usuario tener asociadas varias urls. Esta es una relación uno-

con-varios, el tipo de relación más común, y hasta que se nos presentó el dilema del Tercer Nivel de F/N. la única clase de relación que necesitamos.

La relación varios-con-varios, sin embargo, es ligeramente más compleja. Observa en nuestro ejemplo del Tercer Nivel de F/N que tenemos a un usuario relacionado con varias urls. Como dijimos, vamos a cambiar la estructura para permitir que varios usuarios estén relacionados con varias urls y así tendremos una relación varios-con-varios. Veamos como quedarían nuestras tablas antes de seguir con este planteamiento.

**TABLA N° 5:** Tabla de cuarto nivel de formalización.

<b>Usuarios</b>		
<b>userId</b>	nombre	RelEmpresald
1	Joe	1
2	Jill	2
<b>Empresas</b>		
<b>emprld</b>	empresa	direccion_empresa
1	ABC	1 Work Lane
2	XYZ	1 Job Street
<b>Urls</b>		
<b>urlld</b>	url	
1	abc.com	
2	xyz.com	
<b>url_relations</b>		
<b>relationld</b>	relatedUrld	RelatedUserId
1	1	1
2	1	2
3	2	1
4	2	2

Fuente: [Wikipedia.com/tablas/formalizacion4](http://Wikipedia.com/tablas/formalizacion4).

Para disminuir la duplicación de los datos (este proceso nos llevará al Cuarto Nivel de F/N), hemos creado una tabla que sólo tiene claves externas y primarias `url_relations`. Hemos sido capaces de remover la entrada duplicada en la tabla `urls` creando la tabla `url_relations`. Ahora podemos expresar fielmente la relación que ambos Joe and Jill tienen entre cada uno de ellos, y entre ambos, las urls. Así que veamos exactamente qué es lo que el Cuarto Nivel de F/N. supone:

En las relaciones varios-con-varios, entidades independientes no pueden ser almacenadas en la misma tabla.

Ya que sólo se aplica a las relaciones varios-con-varios, la mayoría de los desarrolladores pueden ignorar esta regla de forma correcta. Pero es muy útil en ciertas situaciones, tal como está. Hemos optimizado nuestra tabla `urls` eliminado duplicados y hemos puesto las relaciones en su propia tabla.

Un ejemplo práctico, ahora podemos seleccionar todas las urls de Joe realizando la siguiente instrucción SQL:

```
SELECT nombre, url FROM usuarios, urls, url_relations WHERE  
url_relations.relatedUserId = 1 AND usuarios.userId = 1 AND urls.urlId =  
url_relations.relatedUrlId
```

Y si queremos recorrer todas las urls de cada uno de los usuarios, haríamos algo así:

```
SELECT nombre, url FROM usuarios, urls, url_relations WHERE usuarios.userId  
= url_relations.relatedUserId AND urls.urlId = url_relations.relatedUrlId
```

#### **f. Quinto nivel de formalización/normalización. (F/N)**

Existe otro nivel de normalización que se aplica a veces, pero es de hecho algo esotérico y en la mayoría de los casos no es necesario para obtener la mejor funcionalidad de nuestra estructura de datos o aplicación. Su principio sugiere:

La tabla original debe ser reconstruida desde las tablas resultantes en las cuales ha sido troceada.

Los beneficios de aplicar esta regla aseguran que no has creado ninguna columna extraña en tus tablas y que la estructura de las tablas que has creado sea del tamaño justo que tiene que ser. Es una buena práctica aplicar esta regla, pero a no ser que estés tratando con una extensa estructura de datos probablemente no la necesitarás. (Dr. E.F.Codd en su escrito de 1972, "Further Normalization of the Data Base Relational Model" (Referente a la normalización de las Bases de Datos Relacionales)) .

### **2.2.17 PHP**

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la web. Dentro de una página Web puede incrustar código PHP que se ejecutara cada vez que se visite una página. El código PHP es interpretado en el servidor web y genera código HTML y otro contenido que el visitante verá, PHP fue conocido en 1994 y es fruto del trabajo de un hombre, Rasmus Lerdori. Ha sido adoptado por otras personas de talento y ha experimentado cuatro importantes transformaciones hasta convertirse en el producto actual. (LikeWelling, Laura thomson. "Desarrollo Web con PHP y MySQL "3ra Edicion , 2005 )

### **2.2.18 Ciclo de vida clásico para la ingeniería del software**

La metodología de ciclo de vida clásico para la Ingeniería del Software. Algunas veces llamado **“Modelo en cascada”** exige un enfoque sistemático y secuencial del desarrollo del software que comienza en el nivel del sistema y progresa a través del Análisis de Requerimientos, Diseño, Construcción/Implementación, Prueba y Mantenimiento, modelizado a partir del ciclo convencional de una ingeniería, el paradigma del ciclo de vida abarca las siguientes actividades. (Roger S. Pressman, 1997: 25 a 29) [www.uprb.edu/profesor/.../base\\_de\\_datosdesventajasdesventajas.htm](http://www.uprb.edu/profesor/.../base_de_datosdesventajasdesventajas.htm)

✓ **Ingeniería y Análisis del Sistema.** Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

Este planteamiento del sistema es esencial cuando el software debe interrelacionarse con otros elementos, tales como el hardware, personas y base de datos. La ingeniería y el análisis del sistema abarcan los requisitos globales a nivel del sistema con una pequeña cantidad de análisis y de diseño a un nivel superior.

✓ **Análisis de los requisitos del software.** El proceso de recopilación de los requisitos se centra e intensifica especialmente para el software. Para comprender la naturaleza de los programas que hay que construir, el ingeniero de software (“Analista”) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas. Los requisitos, tanto del sistema como del software, se documentan y se revisan con el cliente.

- ✓ **Diseño del Sistema.** El diseño del software es realmente un proceso multipaso que se enfoca sobre cuatro atributos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental, y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecido de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos el diseño se documenta y forma parte de la configuración del software.
- ✓ **Construcción/Implementación.** El diseño debe traducirse en una forma legible para la máquina. El paso de implementación utilizando los lenguajes de programación y las herramientas de software adecuados realizan esta tarea de implementación. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.
- ✓ **Prueba.** Una vez que se ha generado el código y se ha producido la implementación del software, comienza la prueba del programa. La prueba se centra en la lógica interna del software, asegurando que todos los módulos se han probado, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.
- ✓ **Implantación y Aceptación.** Esta fase tiene como objetivo principal la entrega y aceptación del sistema en su totalidad, y la realización de todas las actividades necesarias para el paso a producción del mismo. Conviene señalar la participación del usuario de operación en las pruebas de implantación, del usuario final en las pruebas de aceptación, y del responsable del mantenimiento. Las actividades previas al inicio de la producción incluyen la preparación de la infraestructura necesaria para configurar el entorno, la instalación de los

componentes, la activación de los procedimientos manuales y automáticos asociados y, cuando proceda, la migración o carga inicial de datos. Para ello se toma como punto de partida los productos software probado.

✓ **Mantenimiento.** El software, indudablemente sufrirá cambios después de que se entregue al cliente. Los cambios ocurrirán debido a que se hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo. El mantenimiento del software aplica cada uno de los pasos precedentes del ciclo de vida de un programa existente en vez de a uno nuevo.

El ciclo de vida es el paradigma más ampliamente usado en la ingeniería del software, pese a sus falencias, esta metodología tiene un lugar importante y definido dentro del trabajo realizado en ingeniería del software. Suministra una plantilla en la que pueden colocarse los métodos para el análisis, diseño, construcción, prueba, implantación y mantenimiento. Además, vemos que los pasos del paradigma clásico del ciclo de vida son muy similares a los pasos genéricos aplicables a todos los paradigmas de ingeniería del software. El ciclo de vida clásico sigue siendo el modelo más ampliamente usado por los ingenieros del software. A pesar de sus inconvenientes, es significativamente mejor que desarrollar el software sin guías.

### 2.2.19 Metodología Ágil

Las Metodologías Ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de e-projects que las metodologías convencionales (ISO-9000, CMM, etc) debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

### **2.2.20 Metodología aplicada XP (EXTREME PROGRAMMING)**

Es una metodología centrada en potenciar las relaciones interpersonales como clave para un éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Xp se define como especialmente adecuada para proyectos muy cambiantes.

### **2.2.21 Sistema de registro de datos de alpacas**

Este sistema recopila toda la información existente dentro de la institución o sector agropecuario, el cual se usa para el uso y la administración de dicha información real para una municipalidad, rancho, establecimiento universitario, instituto o colegio en el cual se desarrolle el sistema.

El sistema de Información de registro de datos es una aplicación que automatiza todo el proceso de gestión agropecuaria de una municipalidad, mediante ágiles opciones que permiten registrar y compartir información de animales como alpacas y la planificación de cada periodo en ejecución.

El sistema de información genealógico de alpacas incluye:

- Definición de parámetros del sistema
- Planificación Agropecuaria
- Control de peso

- Control de finura del vellón
- Registro de nuevas alpacas
- Control de empadre controlado
- Información histórica de alpacas
- **Planificación agropecuaria**

Permite establecer el ciclo del curso de vida de un animal durante su periodo de vida desde el nacimiento, su reproducción, su muerte de cada animal. Adicionalmente el sistema incluye un módulo que administra los recursos de usuarios con diferentes niveles. El sistema cuenta con varios reportes que nos permiten controlar los datos antes y después de su inicio de registro.

- **Control de registro de datos de alpacas**

Estas opciones facilitan el registro de información referente a cada alpaca, permitiendo que la unidad de desarrollo agropecuario pueda realizar el centro de cada alpaca mensualmente.

Algunas de las operaciones disponibles son:

- ✓ Modificación de datos registrados
- ✓ Ingreso de datos registrados
- ✓ Actualización de datos
- ✓ Adición de datos
- ✓ Bloqueo y desbloqueo de Usuarios.

### 2.2.22 Estándares de calidad.

#### Norma de Calidad iso-9126:

ISO 9126 es un estándar internacional para la evaluación del Software, fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad del software.

La normativa define seis características de la aplicación, estas seis características son divididas en: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de sub características que permiten profundizar en la evaluación de la calidad de productos de software.

#### 2.2.23 Pruebas de software verificación y validación

##### Verificación

“¿Estamos construyendo el producto correctamente?”.

El software debería ajustarse a su especificación

##### Validación

“¿estamos construyendo el producto correcto?”.

El software debería hacer lo que el cliente realmente reclama.

La prueba V &V debe aplicarse en cada etapa del software.

#### Tiene dos objetivos principales

- ✓ El descubrimiento de defectos en el sistema

- ✓ La evaluación de si el sistema es útil y utilizable en una situación operacional o no.

### **Normas de Calidad asociadas a las pruebas de software**

- ✓ Desarrollo del software-Validación y Verificación del software Norma ISO/IEC12207, Norma ISO/IEC17025 (<http://www.lysconsultores.com/Descargar/NT006.pdf>)
- ✓ Mantenimiento del software–NormaISO14764.
- ✓ Software Testing Norma ISO/IEC29119(<http://softwaretestingstandard.org/>).

### **2.2.24 Tipos de pruebas**

#### **Pruebas de defectos**

- ✓ Pruebas diseñadas para descubrir defectos en el sistema.
- ✓ Una prueba de defectos exitosa es aquella que revela la presencia de defectos en un sistema.

#### **Pruebas de validación**

- ✓ Previsto para mostrar que el software cumple sus requerimientos.
- ✓ Una prueba con éxito es aquella que muestra que un requerimiento se ha implementado correctamente.

#### **Prueba de aceptación**

El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.<sup>2</sup>

Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario. La validación del sistema se consigue mediante la realización de pruebas de caja negra que demuestran la conformidad con los requisitos y que se recogen en el plan de pruebas, el cual define las verificaciones a realizar y los casos de prueba asociados.

### **Pruebas de unidad**

La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software (Módulo).

Aquí se prueban los caminos de control importantes, con el fin de descubrir errores dentro del ámbito de un módulo. Estas pruebas se pueden hacer desde etapas tempranas de desarrollo como pruebas estáticas.

### **Errores Más Comunes Durante La Prueba De Unidad**

- ✓ Procedencia aritmética incorrecta mal aplicada
- ✓ Operaciones de modo mezcladas.
- ✓ Inicializaciones incorrectas.

---

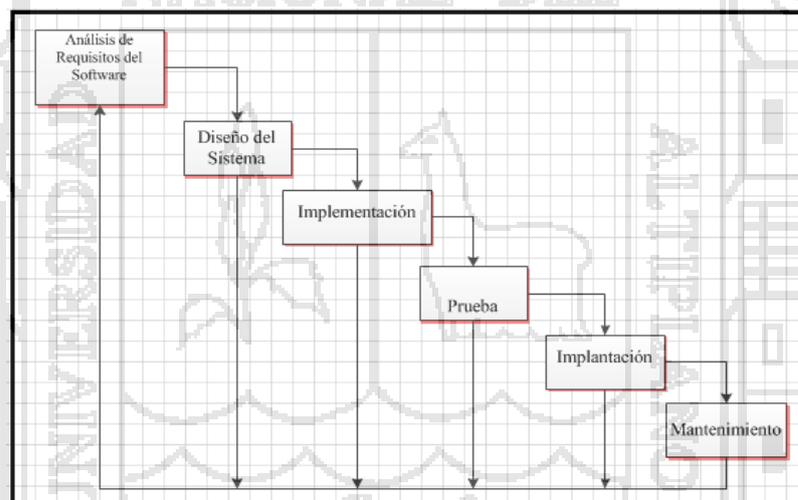
2.- Prueba del software por el usuario y aceptación definitiva.

- ✓ Falta de precisión.
- ✓ Representación incorrecta de una expresión.

### 2.2.25 Diagrama o modelo Entidad-Relación

Un diagrama o modelo entidad-relación (a veces denominado por sus siglas en inglés, *E-R* "Entityrelationship", o del español *DER* "Diagrama de Entidad Relación") es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

**GRÁFICO N° 1:** Proceso principales del Ciclo de Vida Clásico.



Fuente: Teoría entidad Relación

### 2.2.26 Lenguaje del Modelado Unificado (UML).

UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Los diagramas del UML son la representación gráfica de una colección de elementos con sus relaciones, ofreciendo así una

vista del sistema a modelar. Para poder representar de forma correcta un sistema, el lenguaje presenta una amplia variedad de diagramas para así visualizar el sistema desde diversas perspectivas.

Entre esos diagramas se encuentran:

A. Diagramas de Casos de Uso

B. Diagramas de Secuencia

C. Diagramas de Actividades

D. Diagramas de clases.

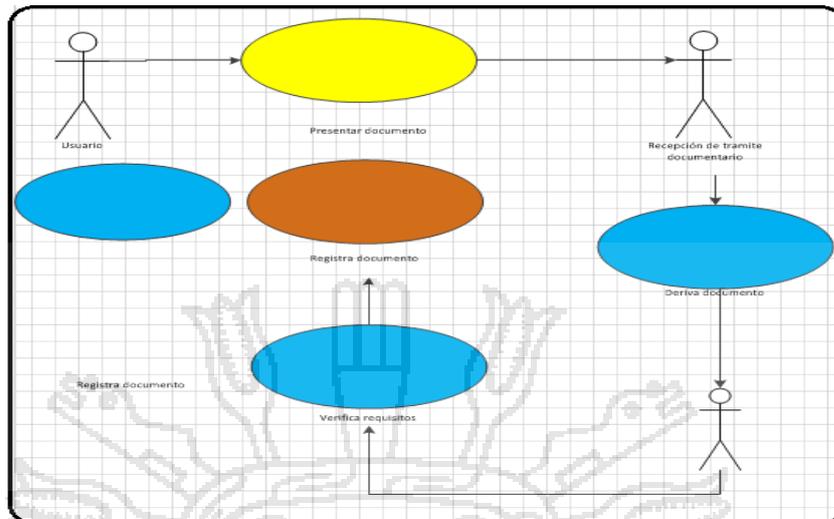
E. Diagrama Colaboración.

E. Diagrama Componentes.

**a). Diagramas de casos de uso.**

Los Casos de Uso son parte del análisis (qué). De forma que al ser parte del análisis nos ayudan a describir qué es lo que el sistema debe hacer. Los Casos de Uso son qué hace el sistema desde el punto de vista del usuario. Es decir, describimos el uso del sistema y cómo este interactúa con el usuario.

**GRÁFICO N° 2:** Diagrama de casos de uso.

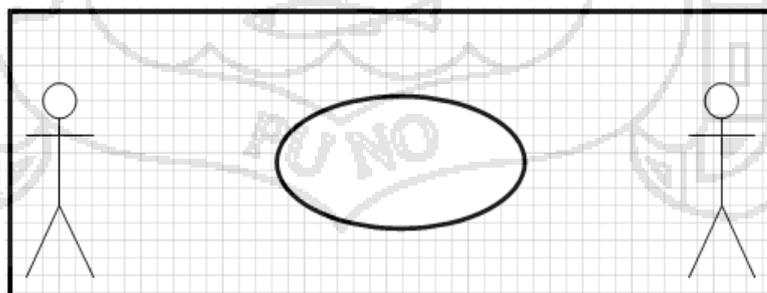


Fuente: Elaboración Propia

**Actor:**

Una definición previa, es que un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

**Gráfico N° 03:** Interacción de actores.



Fuente: Elaboración Propia.

**Caso de Uso:**

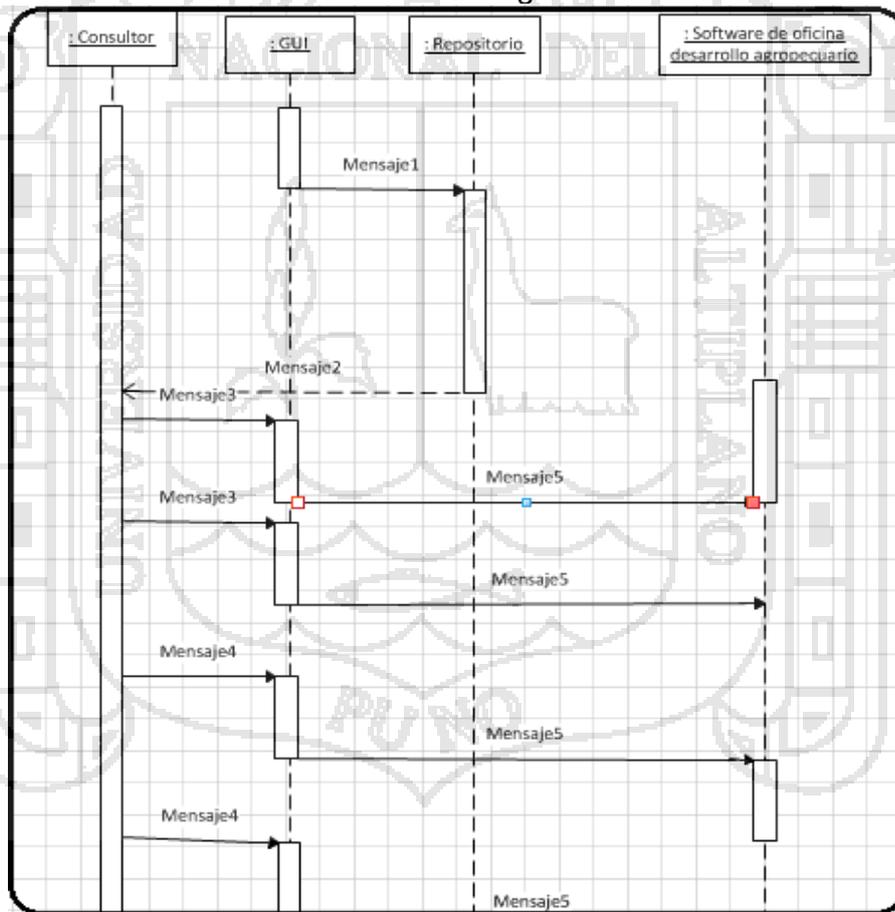
Es una operación/tarea específica que se realiza tras una orden de algún agente

externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

**b). Diagramas de secuencias.**

Los mensajes pueden ser bien síncronos, el tipo normal de llamada del mensaje donde se pasa el control a objeto llamado hasta que el método finalice, o asíncronos donde se devuelve el control directamente al objeto que realiza la llamada. Los mensajes síncronos tienen una caja vertical en un lateral del objeto invocante que muestra el flujo del control del programa.

**GRÁFICO N° 3:** Gráfico de diagrama de secuencias.



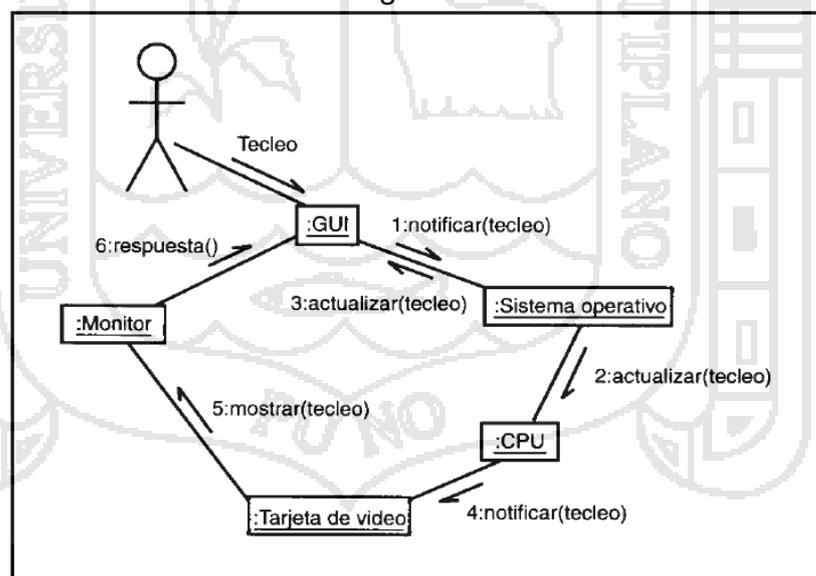
Fuente: Elaboración Propia

**c). Diagramas de colaboración.**

En estos diagramas de colaboración se muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

En los diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa.

**GRÁFICO N° 4:** Diagrama de colaboraciones.



**Fuente:** Diagrama de colaboración.

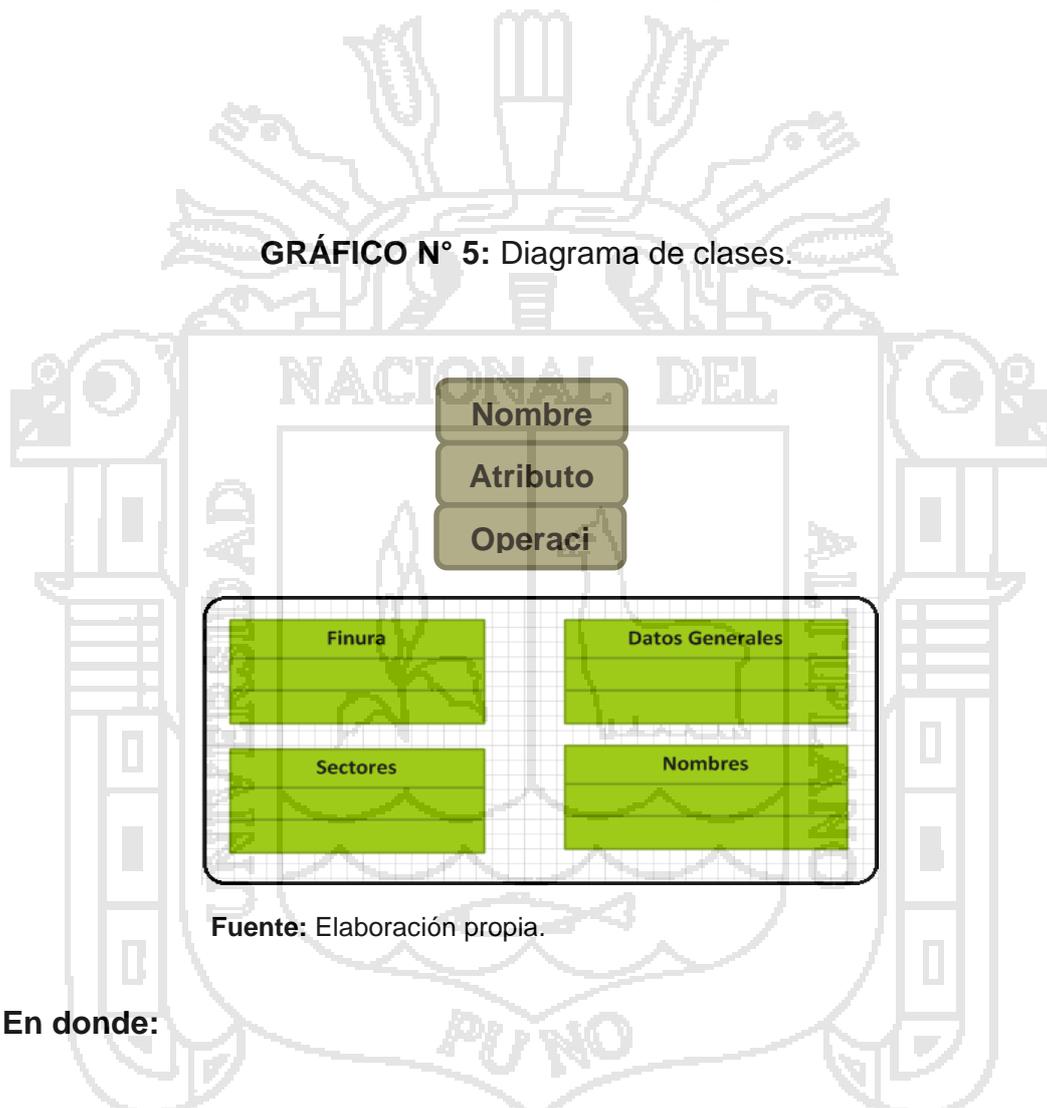
#### d). Diagramas de clases.

En el diagrama de clases se muestran las diferentes clases que componen un

sistema y cómo se relacionan unas con otras. Además, los diagramas de clases son diagramas «estáticos» porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases «conocen» a qué otras clases o qué clases «son parte» de otras clases.

En UML, una clase es representada por un rectángulo que posee tres divisiones:

**GRÁFICO N° 5:** Diagrama de clases.



Fuente: Elaboración propia.

En donde:

**Superior:** Contiene el nombre de la Clase (Datos generales, datos de longitud fibra, datos de pesos y localización)

**Intermedio:** Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).

**Inferior:** Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

### **2.2.27 Arquitectura cliente – servidor.**

Arquitectura hardware y software adecuada para el proceso distribuido, en el que la comunicación se establece de uno a varios. Un proceso es un programa en ejecución. Proceso cliente es el que solicita el servicio. Proceso servidor es capaz de proporcionar un servicio.

Un proceso cliente se puede comunicar con varios procesos servidores y un servidor se puede comunicar con varios clientes. Los procesos se pueden ejecutar en la misma máquina o en distintas máquinas comunicadas a través de su red, por lo general, la parte de la aplicación correspondiente al cliente se optimiza para la interacción con el usuario, ejecutándose en su propia máquina, mientras que la parte correspondiente al servidor proporciona la funcionalidad multiusuario centralizada y se ejecuta en la máquina remota.

Una aplicación Cliente/Servidor típica es un servidor de base de datos al que varios usuarios realizan consultas simultáneamente. El proceso cliente realiza una consulta, el proceso servidor le envía las tablas resultantes de la consulta y el proceso cliente las interpreta y muestra el resultado en pantalla.

## 2.3 DEFINICIÓN DE TÉRMINOS BÁSICOS

**2.3.1 ASP (Active Server Pages)** Herramienta que conforman páginas con contenido.

**2.3.2 CGI (Common Gateway Interface)** es una norma para establecer comunicación entre un servidor web y un programa, de tal modo que este último pueda interactuar con internet.

**2.3.3 LINK** establece la asociación entre una página que puede ser: cliente servidor.

**2.3.4 SIW (Sistema De Información Web)** comprenden sistemas de información que utilizan la tecnología web para la interacción con los usuarios.

**2.3.5 TIC (Tecnologías De La Información Y La Comunicación)** conjunto de herramientas que combinan los recursos informáticos para lograr un uso más eficiente de la información y la comunicación.

**2.3.6 TCP/IP (Transfer Information Protocol/ Internet Protocol).** -Es el protocolo común utilizado por todos los ordenadores conectados a internet, de manera que estos puedan comunicarse.

**2.3.7 INGENIERÍA DEL SOFTWARE.** - Es el enfoque sistemático para el desarrollo, operación, mantenimiento y eliminación del software.

**2.3.8 DIAGRAMA DE FLUJO DE DATOS.** - Tiene una serie de propiedades para asegurar la información que son completas y directas.

**2.3.9 ALGORITMO.** - Conjunto de reglas y procedimientos lógicos que, ejecutados en un orden determinado resuelve un problema.

**2.3.10 BASE DE DATOS.** - Organización y conservación de datos e información a través de un control central, el cual permite al usuario el acceso a los datos de acuerdo a sus atributos.

**2.3.11 API ( Application Programming Interface)** Es un conjunto de rutinas que provee acceso a funciones de un determinado software.

**2.3.12 DISEÑO.** - Delineación de una figura organizacional (planear, reemplazar y completar) interactuando el diseño lógico (análisis de sistema).

**2.3.13 SISTEMA.** -Es un conjunto de componentes que se interactúan entre sí, para lograr un cierto objetivo.

**2.3.14 INFORMÁTICA.** -Campo de conocimientos que abarca todos los campos de diseño y el uso de las computadoras y el tratamiento automático de la información.

**2.3.15 HARDWARE.** -Dispositivos de una computadora o parte física en ella.

**2.3.16 REPORTE.** - Un reporte es un informe o una noticia. Este tipo de documento(que puede ser impreso, digital, audiovisual, etc.) pretende transmitir una información, aunque puede tener diversos objetivos. Existen reportes divulgativos, persuasivos y de otros tipos.

**2.3.17 GESTIÓN.** - El término gestión, por lo tanto, implica al conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto.

**2.3.18 DIGITAR.** - Incorporar datos a la computadora utilizando el teclado.

**2.3.19 OPTIMIZAR.** - Buscar la mejor manera de realizar una actividad.

**2.3.20 USUARIO.** - Es la persona que hace uso de los servicios proporcionados por una instancia o institución.

**2.3.21 CALCE.** - parte de la alpaca el cual se ingresa a la base de datos.

**2.3.22 DELEGAR REPORTES:** Permite delegar los reportes o la firma de la misma a la cualquier dependencia de la municipalidad.

2.4 OPERACIONALIZACIÓN DE VARIABLES

TABLA N° 6:Operacionalizacion de Variables.

Variables	Dimensión	Indicadores	Escala o Medición
Sistema de Información	Diagramas UML	Diseño de la Interfaz del sistema	- Muy bueno
			- Bueno
			- Regular
		Amigabilidad del Interfaz del sistema	- Amigable
			- Poco Amigable
			- No es amigable
		Accesibilidad al Sistema	- Fácil
			- Medianamente Fácil
			- Difícil
		Servicio que ofrece el Sistema	- Eficiente
			- Medianamente eficiente
			- Deficiente
Confiabilidad de los reportes	- Confiable		
	- Poco confiable		
	- No es confiable		
Mejora la Atención a los pobladores y Productores	Reduce el tiempo para la atención a los Productores y trabajadores	Tiempo en la atención	- Más de 5 minutos
			- 4 minutos
			- Menos de 3 minutos

Fuente: Elaboración propia.

## CAPÍTULO III

### MATERIALES Y MÉTODOS

#### 3.1 POBLACIÓN

La población en un proyecto de investigación puede estar constituida por personas, objetos, materiales, servicios y otros. En nuestro caso la población está constituida por todos propietarios ganaderos criadores de alpacas del distrito de Ajoyani que hicieron un registro dentro de la oficina de desarrollo agropecuario el cual consta con un total de 1478 criadores con un total de 24000 alpacas.

#### 3.2 MUESTRA

Para determinar el marco muestral se utilizó el diseño de Muestreo Aleatorio Simple (MAS), donde cada elemento de la población tiene igual probabilidad de que sea incluido en la muestra, se aplicó este diseño porque el muestreo probabilístico garantiza la representatividad de la muestra. El tipo de muestreo empleado fue el Muestreo Aleatorio Simple, ya que la población donde los sujetos de estudio están de forma homogénea y de donde se seleccionarían mediante el procedimiento aleatorio simple.

La muestra piloto se realizó a 72 criadores suponiendo que el comportamiento de los sujetos es proveniente de una población con una distribución normal.

Dado que se conoce la cantidad total de elementos que constituyen la población a estudiar, se utilizó la fórmula para calcular muestras de poblaciones finitas, así:

### CALCULO DE TAMAÑO DE MUESTRA

$$n_0 = \frac{Z_{\alpha/2}^2 PQ}{E^2}$$

$$n = \frac{n_0}{1 + n_0/N}$$

$$f = \frac{n_0}{N}$$

Dónde:

$n_0$  : Primera aproximación del tamaño de la muestra.

$n$  : Tamaño de la muestra definitivo.

$f$  : Fracción de muestreo.

$N$  : Tamaño de la población.

$Q$  : Proporción de opinión de que con la implementación de un Sistema de información no mejorara la eficiencia en agilización de los trámites (0.2).

$p$  : proporción de pobladores que opinaron de que con la implementación del Sistema (0.8).

$Z_{\alpha/2}^2$  : Desviación estándar (para un nivel de confianza de 95% es de 1.96).

$E$  : Error absoluto aceptado (0.09).

## Definición del tamaño de muestra

Dónde:

$$Z_{\alpha/2} = 1.96$$

$$E = 0.09$$

$$P = 0.8$$

$$Q = 1 - P = 1 - 0.80 = 0.20$$

$$N = 1478$$

Calculamos primera aproximación de la muestra

$$n_0 = \frac{Z_{\alpha/2}^2 PQ}{E^2}$$

$$n_0 = \frac{1.96 * 1.96 * 0.8 * 0.2}{0.09 * 0.09} = 75.31$$

Comprobando el factor de corrección es mayor del 5%

$$f = \frac{75.31}{1478} = 0.051$$

Por lo tanto, el tamaño definitivo de la muestra será

$$n = \frac{n_0}{1 + n_0/n}$$

$$n = \frac{75.88}{1 + 75.88/1478} = 72.17 = 73$$

### 3.3 DISEÑO DE LA MUESTRA

Realizando el redondeo al entero inmediato se obtuvo  $n = 73$  pobladores, el cual fue un tamaño de muestra para el trabajo de investigación. El marco muestral se obtuvo a partir de los pobladores que fueron censados en el distrito de Ajoyani-Carabaya - 2015.

### 3.4 MÉTODO DE RECOPIACIÓN DE DATOS

Se utilizó los procedimientos manuales o digitales tradicionales (encuestas, empadronamiento, archivos, recibos, copias, listado de pagos) que nos permitió clasificar la información proporcionados por el personal administrativo de la oficina de desarrollo agropecuario. Se hizo uso de toda esta información para poder utilizarla en el nuevo diseño del sistema de información.

Se introdujo un nuevo sistema de recopilación de datos para el control de registros genealógicos en base al software (MySQL, PHP y HTML) y para ello se utilizó:

- ✓ Encuestas
- ✓ Observación
- ✓ Análisis de contenido
- ✓ Estudio de casos

Para la recopilación de la información fue necesario realizar una encuesta a los operadores del sistema, información que fue necesaria para conocer la utilidad de dicho software.

**Encuesta:** La encuesta se aplicó a 73 comuneros que viene hacer una representación de la población que está inscrita en el distrito de Ajoyani.

**Entrevista:** Con la entrevista se recopiló información, mediante una conversación, donde el entrevistador realizo la pregunta y el entrevistado colaboro con la respuesta.

### 3.5 MÉTODO DE ANÁLISIS DE DATOS

El tratamiento de datos se realizó utilizando el ciclo de vida del software, lo que nos permitió analizar, diseñar, desarrollar e implantar un sistema de información.

La metodología que se usó es el “Modelo XP” que exige un enfoque sistemático y secuencial del desarrollo del software que comienza en el nivel del sistema y progresa a través del Análisis de Requerimientos, diseño, construcción/Implementación, prueba y mantenimiento.

### 3.6 MATERIAL EXPERIMENTAL

#### PROGRAMACIÓN EXTREMA

GRÁFICO N° 6: Proceso de programación xtrema.



Fuente: Teoría Programación xtrema

La Programación Extrema (XP) es posiblemente el método ágil más conocido y ampliamente utilizado. El nombre fue acuñado por Beck (Beck, 2000) debido a que el enfoque fue desarrollado utilizando buenas prácticas reconocidas, como el desarrollo iterativo, y con la participación del cliente en niveles “extremos”.

En la programación extrema, todos los requerimientos se expresan como escenarios (llamados historias de usuario), los cuales se implementan directamente como una serie de tareas.

Los programadores trabajan en parejas y desarrollan pruebas para cada tarea antes de escribir el código. Todas las pruebas se deben ejecutar satisfactoriamente cuando el código nuevo se integre al sistema. Existe un pequeño espacio de tiempo entre las entregas del sistema.

La programación extrema implica varias prácticas, que se ajustan a los principios de los métodos ágiles:

El desarrollo incremental se lleva a cabo a través de entregas del sistema pequeñas y Frecuentes y por medio de un enfoque para la descripción de requerimientos basado en las historias de cliente o escenarios que pueden ser la base para el proceso de Planificación.

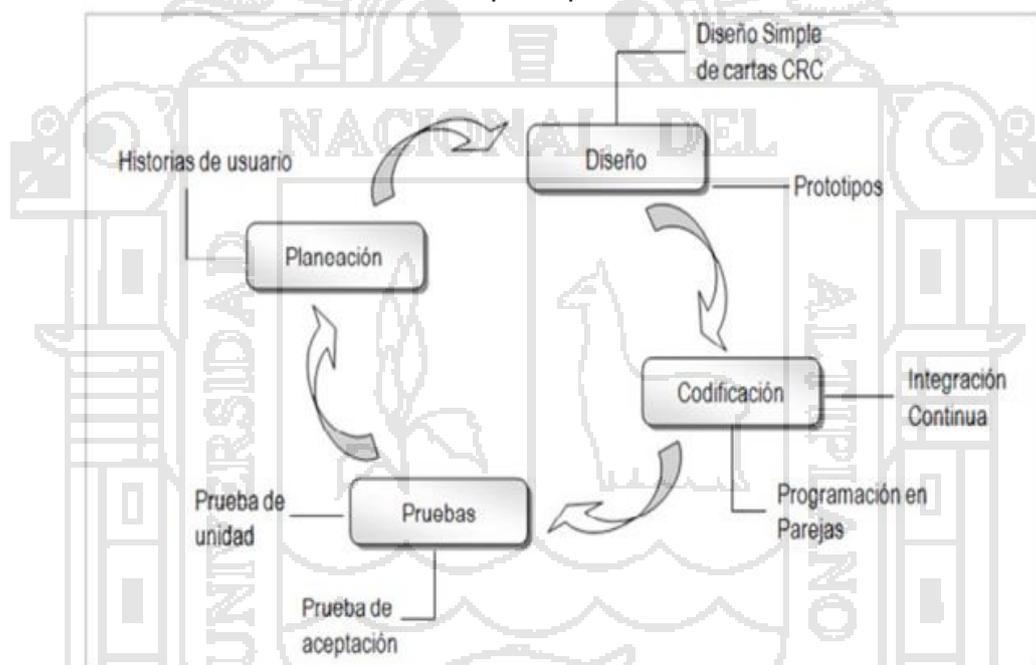
La participación del cliente se lleva a cabo a través del compromiso a tiempo completo del cliente en el equipo de desarrollo. Los representantes de los clientes participan en el desarrollo y son los responsables de definir las pruebas de aceptación del sistema.

El interés en las personas, en vez de en los procesos, se lleva a cabo a través de la programación en parejas, la propiedad colectiva del código del sistema, y

un proceso de desarrollo sostenible que no implique excesivas jornadas de trabajo.

El cambio se lleva a cabo a través de las entregas regulares del sistema, un desarrollo previamente probado y la integración continua. El mantenimiento de la simplicidad se lleva a cabo a través de la refactorización constante para mejorar la calidad del código y la utilización de diseños sencillos que no prevén cambios futuros en el sistema.

**GRÁFICO N° 7:** Proceso principales del Ciclo de Vida Clásico.



**Fuente:**Wikipedia La Programación Extrema (XP)

En un proceso de la XP, los clientes están fuertemente implicados en la especificación y establecimiento de prioridades de los requerimientos del sistema. Los requerimientos no se especifican como una lista de funciones requeridas del sistema. Más bien, los clientes del sistema son parte del equipo de desarrollo y discuten escenarios con otros miembros del equipo. Desarrollan conjuntamente una «tarjeta de historias» que recoge las necesidades del cliente.

Una vez que se han desarrollado las tarjetas de historias, el equipo de desarrollo las divide en tareas y estima el esfuerzo y recursos requeridos para su implementación. El cliente establece entonces la prioridad de las historias a implementar, eligiendo aquellas historias que pueden ser utilizadas inmediatamente para entregar un apoyo útil al negocio. Por supuesto, cuando los requerimientos cambian, las historias sin implementar también cambian o se pueden descartar. Si se requieren cambios en un sistema que ya se ha entregado, se desarrollan nuevas tarjetas de historias y, de nuevo, el cliente decide si estos cambios tienen prioridad sobre nuevas funcionalidades.

La programación extrema adopta un enfoque «extremo» para el desarrollo iterativo. Se pueden construir varias veces al día nuevas versiones del software y los incrementos se entregan al cliente cada dos meses aproximadamente. Cuando un programador construye el sistema para crear una versión nueva, debe ejecutar todas las pruebas automatizadas existentes además de las pruebas para las funcionalidades nuevas. El nuevo software generado solamente se acepta si se ejecutan satisfactoriamente todas las pruebas. Un precepto fundamental de la ingeniería del software tradicional es que se debe diseñar para el cambio. Esto es, hay que prever los cambios futuros en el software y diseñar éste de forma que tales cambios se puedan implementar fácilmente.

Sin embargo, la programación extrema ha descartado este principio partiendo del hecho de que diseñar para el cambio es a menudo un esfuerzo inútil. Con frecuencia los cambios previstos nunca se materializan y realmente se efectúan peticiones de cambios completamente diferentes.

El problema con la implementación de cambios imprevistos es que tienden a degradar la estructura del software, por lo que los cambios se hacen cada vez más difíciles de implementar. La programación extrema aborda este problema sugiriendo que se debe factorizar constantemente el software. Esto significa que el equipo de programación busca posibles mejoras del software y las implementa inmediatamente. Por lo tanto, el software siempre debe ser fácil de entender y cambiar cuando se implementen nuevas historias.

En conclusión, podríamos destacar varios puntos a favor y algunos en contra sobre este tipo de programación, por ejemplo, el implementar un equipo de trabajo y además éste sea organizado de forma que las jornadas laborales sean cortas pero aprovechadas de la mejor forma, logra mejores resultados tanto en eficiencia en el equipo como en el proyecto en el que se esté trabajando “cuatro ojos miran mejor que dos”. Una de sus desventajas más sobresalientes podría llegar a ser el hecho de que el cliente pertenece al equipo de desarrollo en tiempo completo, lo que podría llegar a ser un obstáculo quizás para el cliente, disponibilidad de horarios para trabajar sobre el proyecto que se esté realizando.

Otro contratiempo encontrado para la programación XP, se encuentra en el tipo de proyecto al que se le aplica esta programación, ya que es recomendable que se aplique sólo a proyectos a corto plazo, asimismo el hecho de que imposible prever todo antes de programar que es algo de lo que se intenta hacer en este método prever algún tipo de error antes de programar.

Sin más podemos decir que se puede considerar un tipo de programación bastante óptimo ya que reduce la tasa de errores.

## PASOS DE DESARROLLO DE CÓDIGO EN MODO XP

### Disponibilidad del cliente

Uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto. No solamente como apoyo a los desarrolladores, sino formando parte del grupo. El involucramiento del cliente es fundamental para que pueda desarrollarse un proyecto con la metodología XP.

Al comienzo del proyecto, el cliente debe proporcionar las historias de usuarios.

Pero, dado que estas historias son expresamente cortas y de “alto nivel”, no contienen los detalles necesarios para realizar el desarrollo del código. Estos detalles deben ser proporcionados por el cliente, y discutidos con los desarrolladores, durante la etapa de desarrollo. No se requieren de largos documentos de especificaciones, sino que los detalles son proporcionados por el Cliente, en el momento adecuado, “cara a cara” a los desarrolladores.

Si bien esto parece demandar del cliente recursos por un tiempo prolongado, debe tenerse en cuenta que en otras metodologías este tiempo es insumido por el cliente en realizar los documentos detallados de especificación.

Adicionalmente, al estar el cliente en todo el proceso, puede prevenir a tiempo de situaciones no deseables, o de funcionamientos que no eran los que en realidad se deseaban. En otras metodologías, estas situaciones son detectadas en forma muy tardía del ciclo de desarrollo, y su corrección puede llegar a ser muy complicada.

### **Programación dirigida por las pruebas**

En las metodologías tradicionales, la fase de pruebas, incluyendo la definición de los tests, es usualmente realizada sobre el final del proyecto, o sobre el final del desarrollo de cada módulo. La metodología XP propone un modelo inverso, en el que, lo primero que se escribe son los test que el sistema debe pasar. Luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas.

### **Programación en pares**

XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Si bien parece que ésta práctica duplica el tiempo asignado al proyecto (y por ende, los costos en recursos humanos), al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas. El producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales.

Programación en pares tiene un sobre costo aproximado de 15%, y no de un 100% como se puede pensar a priori. Este sobre costo es rápidamente pagado por la mejor calidad obtenida en el producto final.

Adicionalmente, la programación en pares tiene las siguientes ventajas:

- ✓ La mayoría de los errores se descubren en el momento en que se codifican, ya que el código es permanentemente revisado por dos personas.
- ✓ La cantidad de defectos encontrados en las pruebas es estadísticamente menor.

- ✓ Los diseños son mejores y el código más cortó.
- ✓ El equipo resuelve problemas en forma más rápida.
- ✓ Las personas aprenden significativamente más, acerca del sistema y acerca de desarrollo de software.

### **Integraciones permanentes**

Todos los desarrolladores necesitan trabajar siempre con la “última versión”. Realizar cambios o mejoras sobre versiones antiguas causan graves problemas, y retrasan al proyecto. Es por eso que XP promueve publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores. Idealmente, todos los días deben existir nuevas versiones publicadas para evitar errores, solo una pareja de desarrolladores puede integrar su código a la vez.

### **Propiedad colectiva del código**

En un proyecto XP, todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto. Asimismo, cualquier pareja de programadores puede cambiar el código que sea necesario para corregir problemas, agregar funciones o recodificar.

En otras metodologías, este concepto puede parecer extraño. Muchas veces se asume que, si hay algo de propiedad colectiva, la responsabilidad también es colectiva. Y que “todos sean responsables”, muchas veces significa que “nadie es responsable”. Ward Cunningham explica en una entrevista con Bill Veners, que este razonamiento no es correcto cuando se trabaja con la metodología de

XP. En este caso, quienes encuentran un problema, o necesitan desarrollar una nueva función, pueden resolverlo directamente, sin necesidad de “negociar” con el “dueño” o autor del módulo (ya que, de hecho, este concepto no existe en XP). Muchas veces, explica Cunningham, una solución pasa por la recodificación de varios módulos, que atraviesan de forma horizontal una determinada jerarquía vertical. Si es necesario dialogar y convencer al encargado de cada módulo, posiblemente la solución no se pueda implementar, por lo menos en tiempos razonables. En XP, se promueve la recodificación, en aras de generar códigos más simples y adaptados a las realidades cambiantes. Cualquier pareja de programadores puede tomar la responsabilidad de este cambio. Los testeos permanentes deberían de asegurar que los cambios realizados cumplen con lo requerido, y además, no afectan al resto de las funcionalidades.

### **Ritmo sostenido**

La metodología XP indica que debe llevarse un ritmo sostenido de trabajo. Anteriormente, ésta práctica se denominaba “Semana de 40 horas”. Sin embargo, lo importante no es si se trabajan, 35, 40 o 42 horas por semana. El concepto que se desea establecer con esta práctica es el de planificar el trabajo de manera de mantener un ritmo constante y razonable, sin sobrecargar al equipo.

Cuando un proyecto se retrasa, trabajar tiempo extra puede ser más perjudicial que beneficioso. El trabajo extra desmotiva inmediatamente al grupo e impacta en la calidad del producto. En la medida de lo posible, se debería renegociar el plan de entregas (“Release Plan”), realizando una nueva reunión de planificación con el cliente, los desarrolladores y los gerentes.

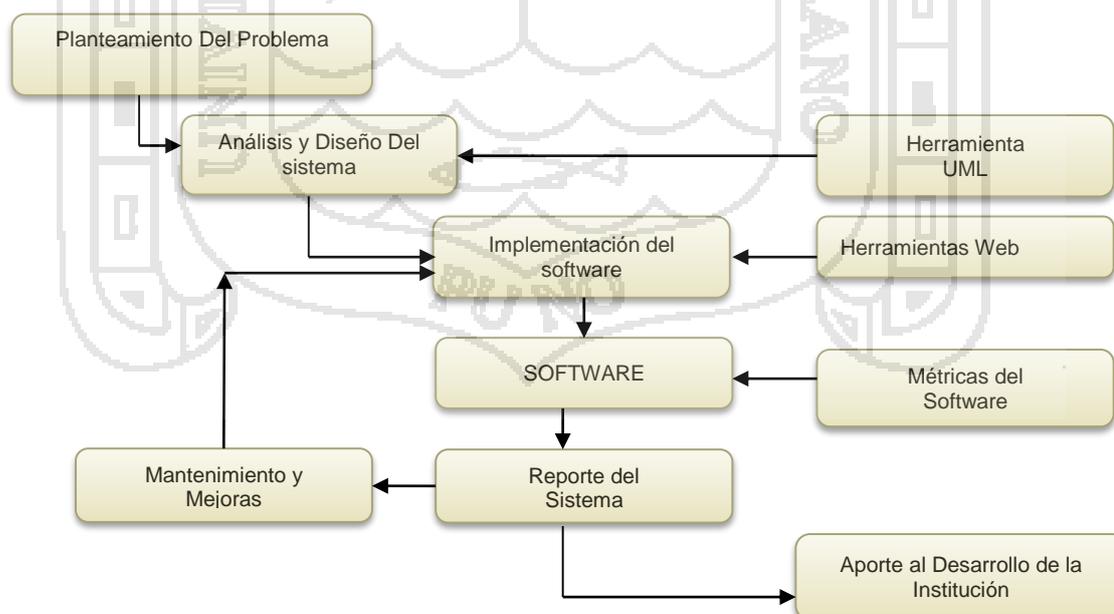
## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

#### 4.1 RESULTADOS Y DISCUSIÓN

Para el adecuado desarrollo del software del sistema, se tiene presente una secuencia de procedimientos, que permitirá llevar en forma ordenada para un mejor entendimiento de las diversas etapas, de la ejecución del presente trabajo de investigación, bajo las especificaciones de Ingeniería de Software Orientado a Objetos, y con la ayuda de Lenguaje Unificado de Modelado (UML) presta todo su apoyo al ciclo de vida completo de software.

**GRÁFICO N° 8:** Ciclo de vida del Software del Sistema.



**Fuente:** Elaboración Propia

#### 4.1.1 Análisis y Diseño del Sistema

En los primeros pasos de una metodología orientada a objetos, en particular usaremos el proceso: concepción, elaboración y construcción, que contiene a la obtención de un proceso preciso, comprensible, conciso y correcto del mundo real.

Las fases en este proceso son:

- ✓ Diseño de los datos.
- ✓ Diseño Arquitectónico.
- ✓ Diseño de la Interfaz.
- ✓ Diseño de procedimientos.

#### 4.1.2 Definición del Problema

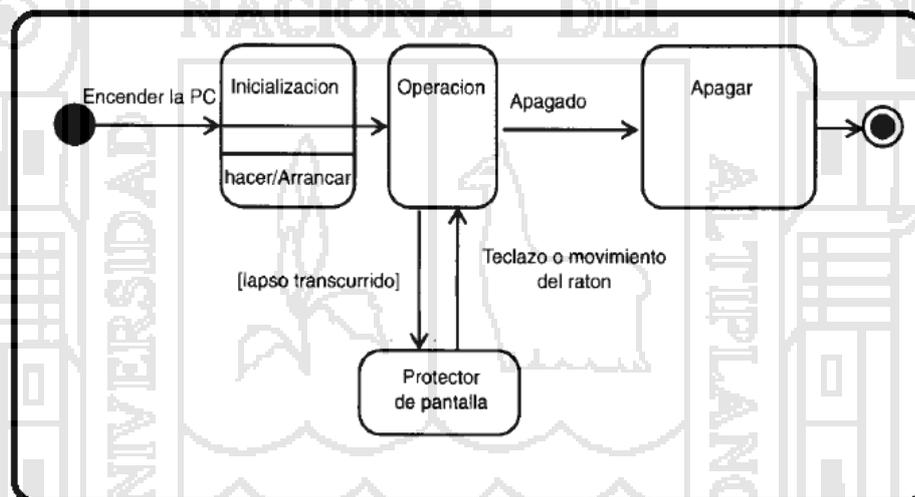
Desarrollar un software para dar soporte a la automatización de la Oficina de desarrollo agropecuario de la Municipalidad Distrital de Ajoyani; debido a muchas causas: como el tiempo reducido, la gran cantidad de solicitudes que se ejecuta a cada registro y petición de registros de árbol genealógico y la dificultad de procedimientos en forma manual; para ello necesitamos desarrollar una herramienta computacional capaz de realizar los requerimientos de control y seguimiento en formas eficiente y rápida, utilizando las más adecuada técnica en su proceso de elaboración de información en forma concreta, precisa y en tiempo real.

### 4.1.3 Análisis de Requerimientos

Para iniciar la construcción del software determinamos los requerimientos dando a conocer su necesidad, meta y exigencias del desarrollo del software del sistema, los cuales fueron ya desarrollados en la fase de recolección de información; ya que generalmente se les presenta problemas en tratamiento de la información, para presentar información oportuna, confiable y con resultados exactos.

### 4.1.4 Implementación del Software del Sistema

**GRÁFICO N° 9:** Ciclo de vida del Software del Sistema.



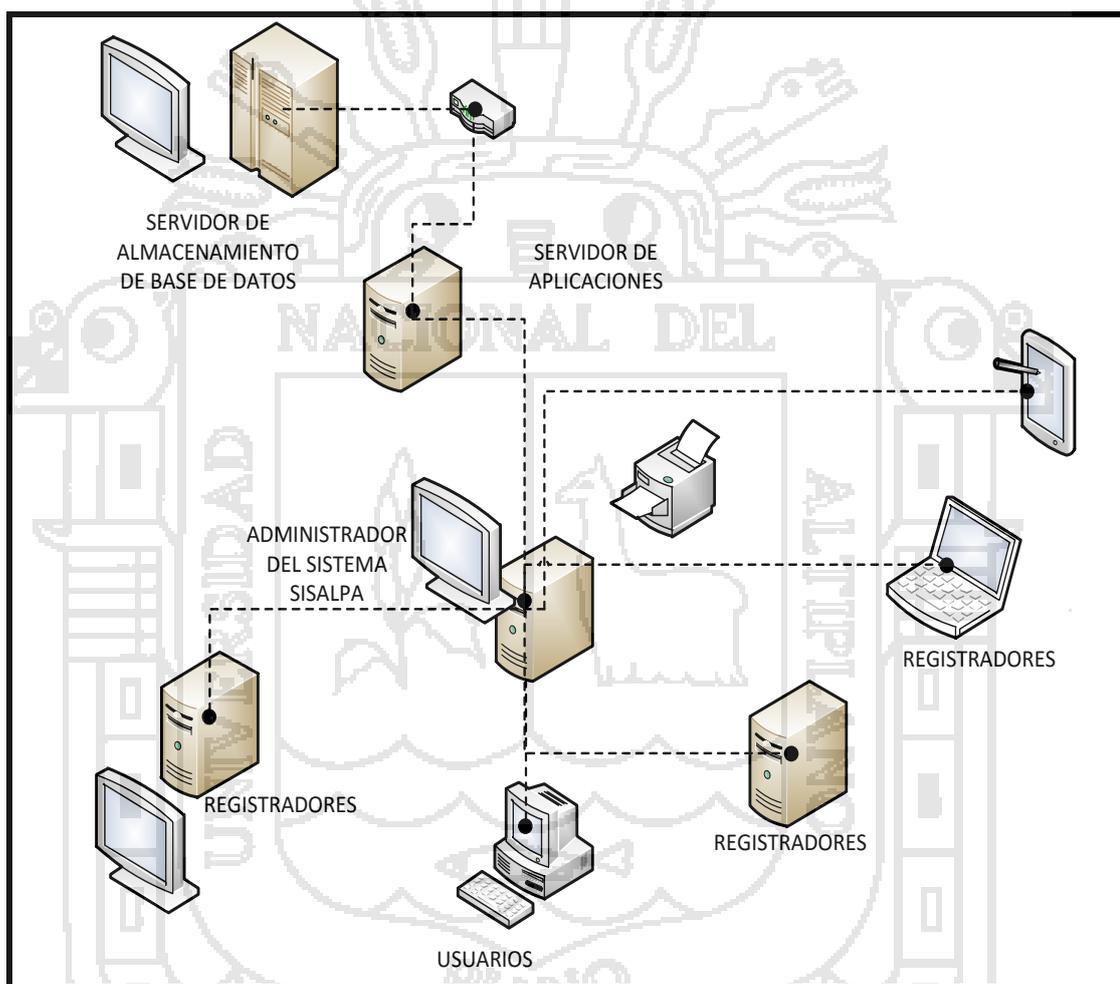
**Fuente:** Ciclo de vida del software.

En la fase de diseño lo fundamental es, saber qué es lo que necesitamos hacer, primero desde un nivel elevado y después empleando niveles cada vez más detallados.

### Diagrama de Componentes

Diagrama que muestra la interacción entre componentes en el proceso de funcionamiento del sistema de Registro Genealógico para la Municipalidad Distrital de Ajoyani.

**GRÁFICO N° 10:** Diagrama lógico de componentes.

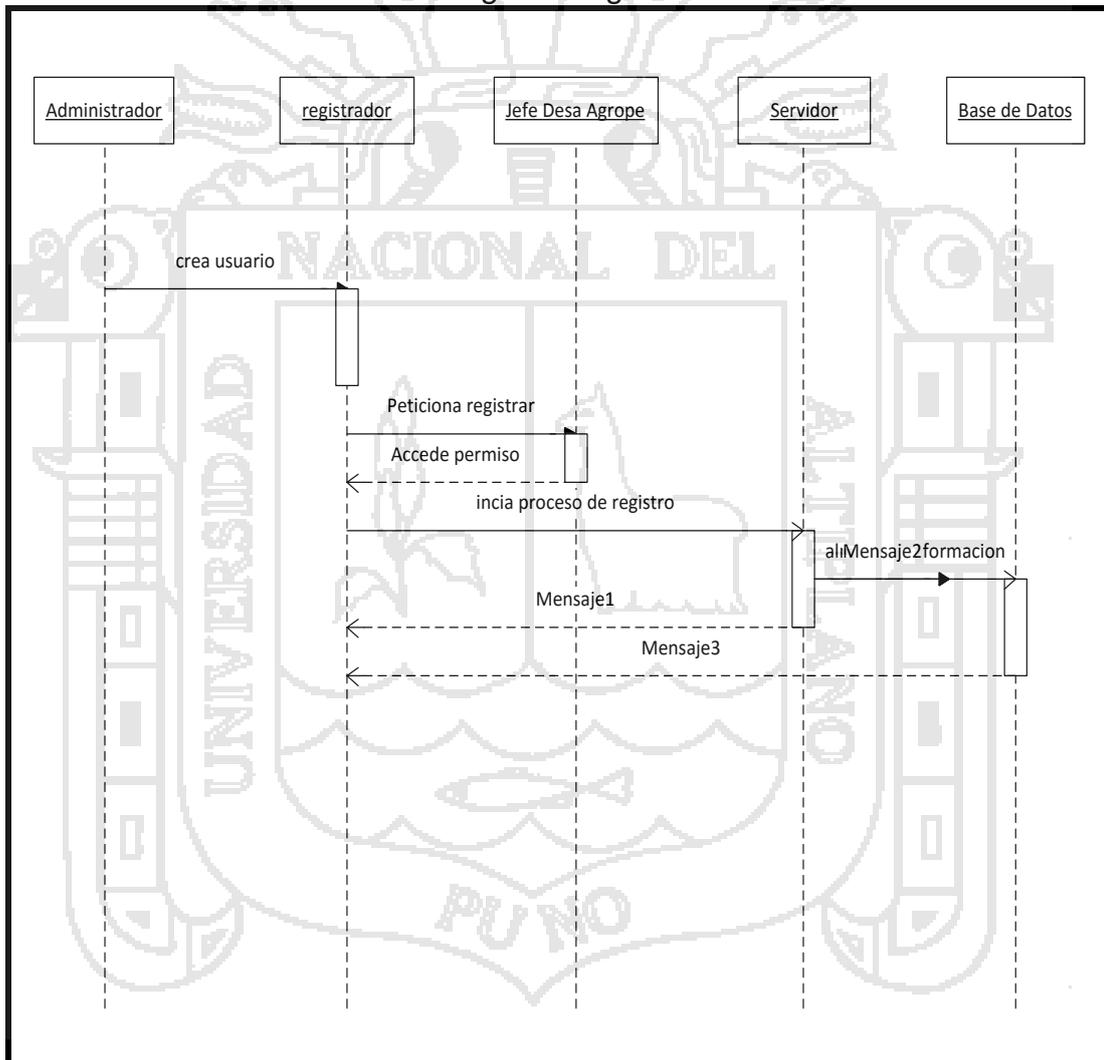


Fuente: Elaboración propia.

**Diagrama de Secuencias**

Diagrama que muestra la interacción entre componentes en el proceso de funcionamiento del sistema de Registro Genealógico para la Municipalidad Distrital de Ajoyani donde muestra la secuencia consecutiva de los procesos a seguir para el registro de datos de un determinado animal en la base de datos.

**GRÁFICO N° 11:** Diagrama lógico de secuencias.

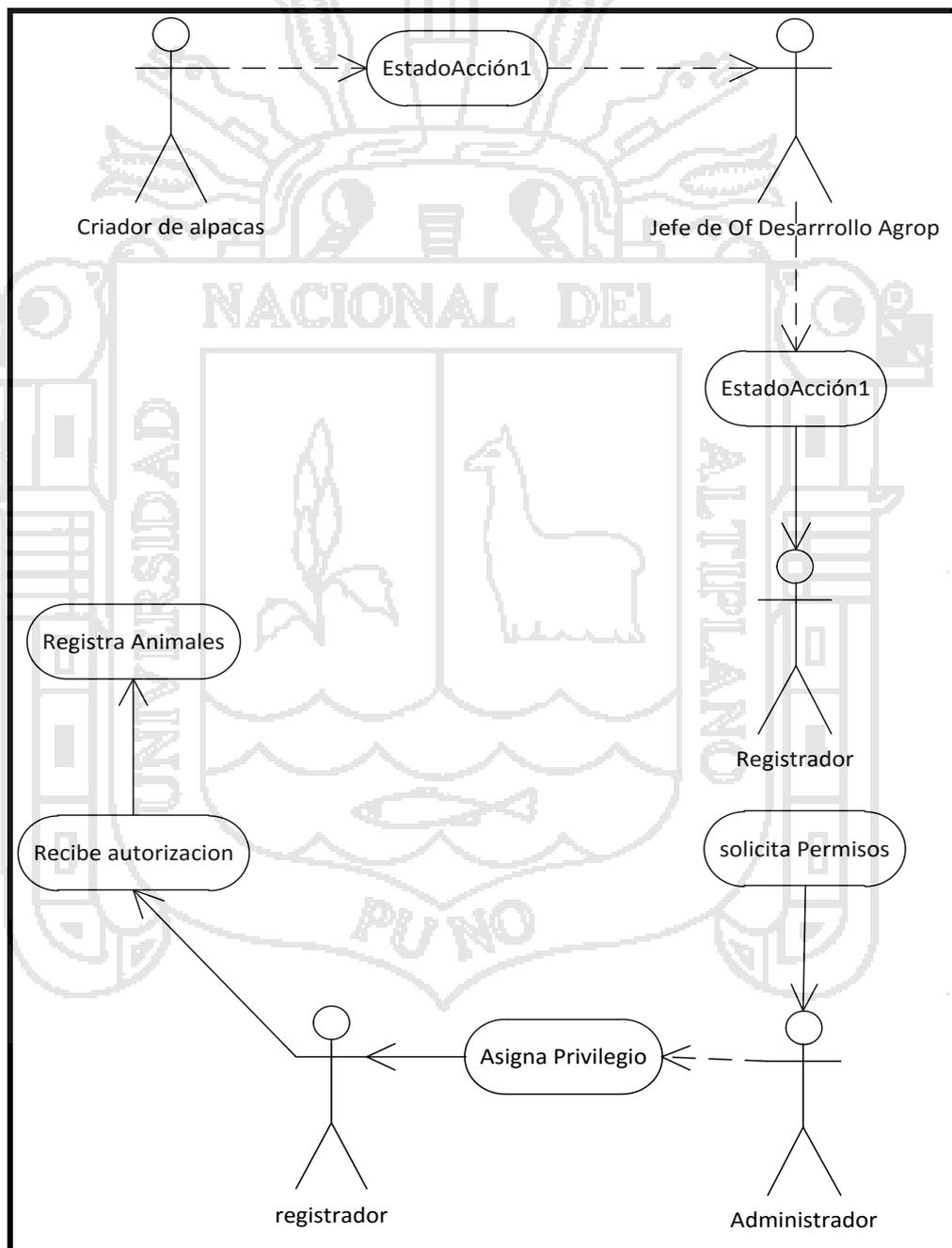


**Fuente:** Elaboración Propia

**Diagrama De Estados**

Diagrama que muestra la interacción entre componentes en el proceso de funcionamiento del sistema de Registro Genealógico para la Municipalidad Distrital de Ajoyani

**GRÁFICO N° 12:** Diagrama lógico de estados.

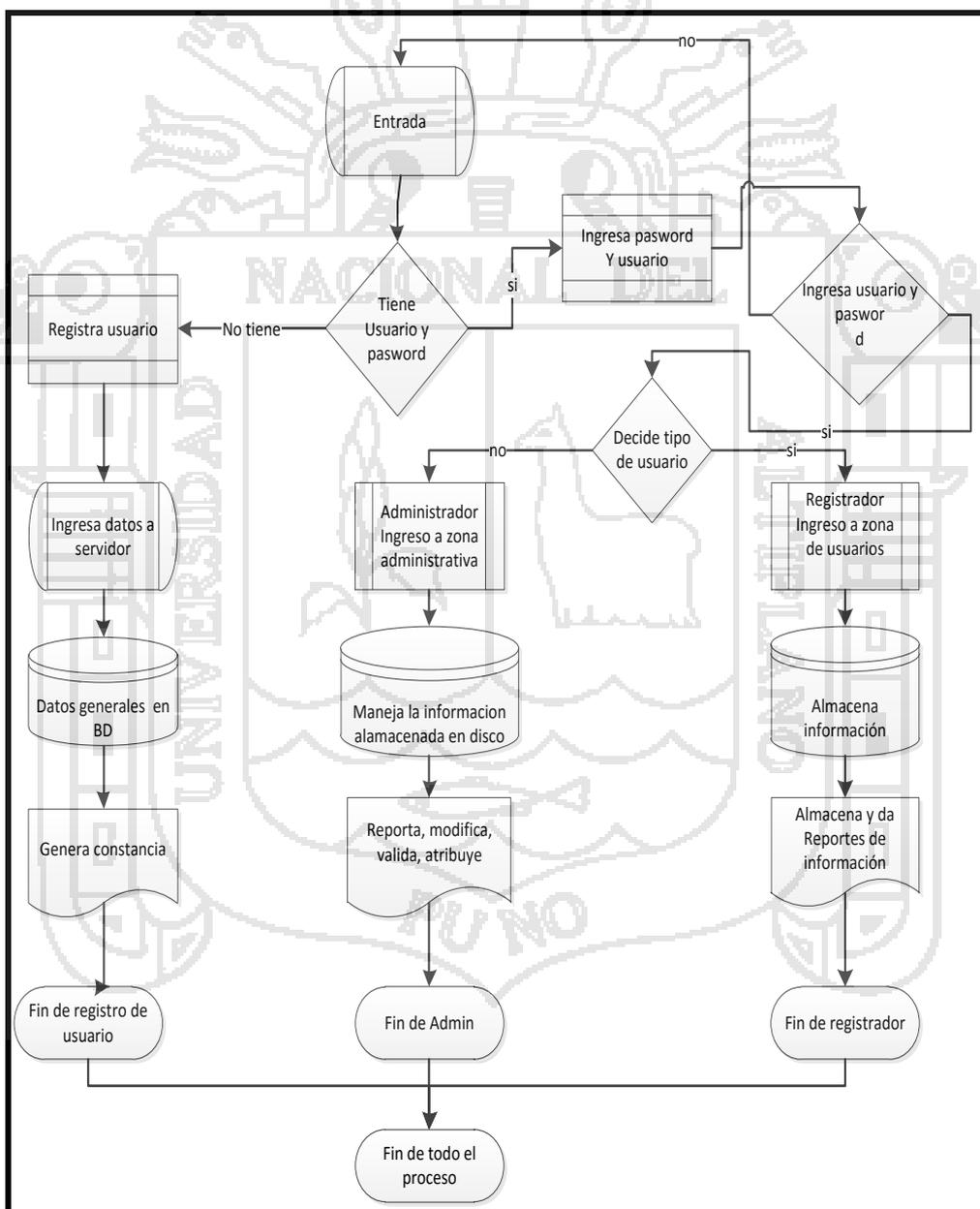


Fuente: Elaboración propia

### 4.1.5 Modelamiento De Base De Datos

Se utilizan para representar la realidad a un alto nivel de abstracción. Mediante los modelos conceptuales se puede construir la descripción de la realidad fácil de entender.

**GRÁFICO N° 13:** Modelado conceptual para el diseño del Software.

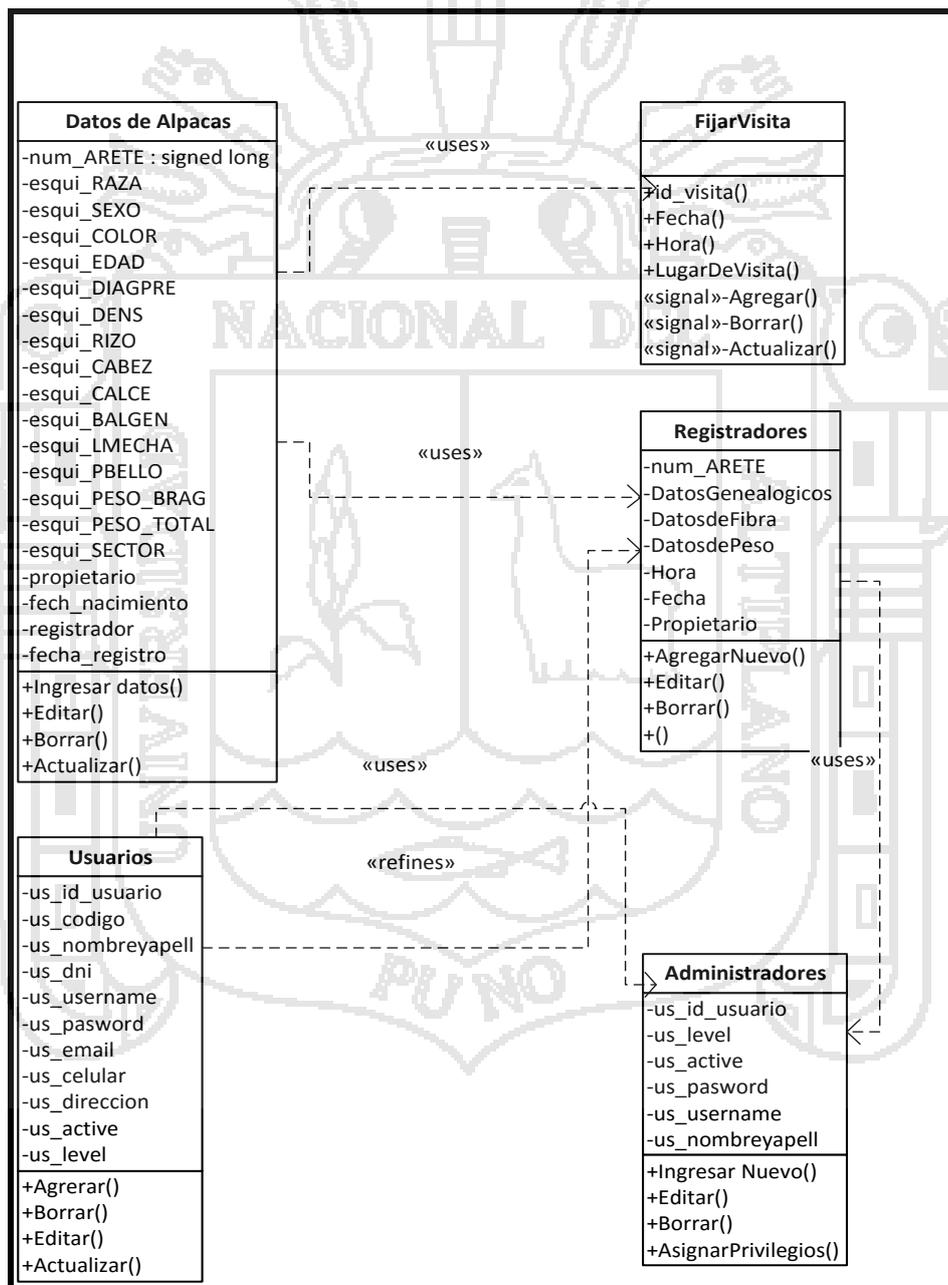


**Fuente:** Elaboración propia.

**Estructuración de Modelo Lógico.**

Diagrama que muestra la interacción entre clases en el proceso de funcionamiento del sistema de Registro Genealógico para la Municipalidad Distrital de Ajoyani.

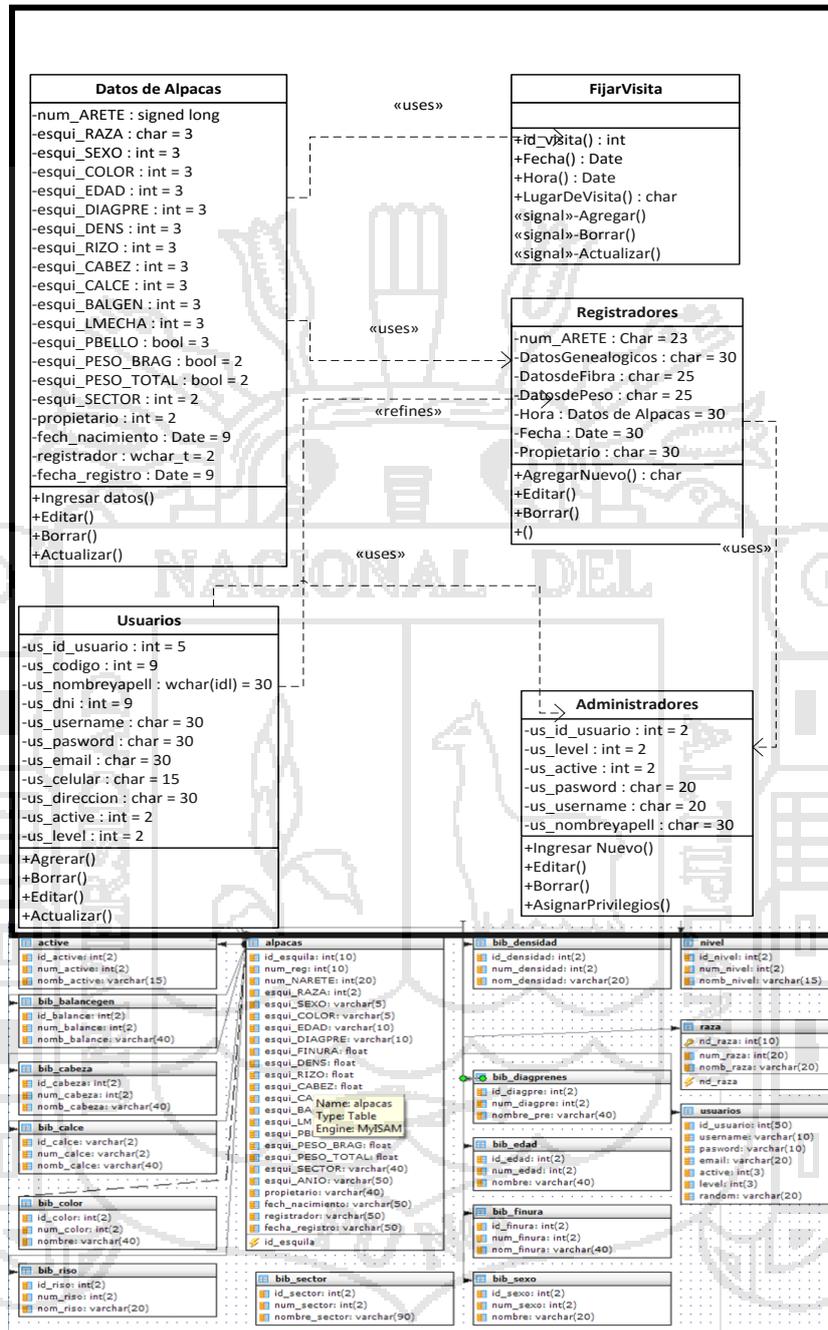
**GRÁFICO N° 14:** Estructuración de modelo lógico.



Fuente: Elaboración Propia.

Implementación de Modelo Físico de base de datos.

GRÁFICO N° 15: Modelo Físico de base de datos.

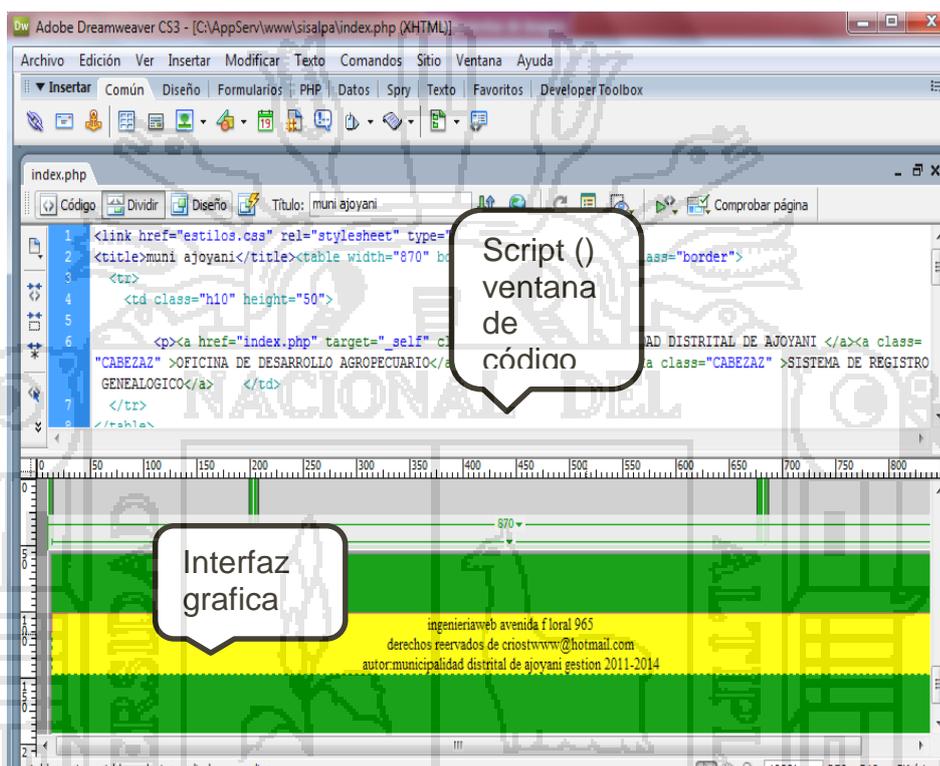


Fuente: Elaboración propia.

#### 4.1.6 Desarrollo del Software del Sistema

El desarrollo del software se realizó con el lenguaje de programación PHP y el HTML y como editor el DreamWeaver.

**GRÁFICO N° 16:** Ventana de editor de Dream Weaber.



**Fuente:** Elaboracion propia

Esta ventana muestra dos áreas una es la parte del código el otro es donde muestra la interfaz la cual está siendo diseñada en la parte del código donde se hace las modificaciones lógicas del software en PHP o en HTML y en la parte de la interfaz se puede hacer también modificaciones en mayor cantidad de código puede ser insertando botones, radio botón, área de texto y todo lo necesario para la creación del software.

## Uso del Lenguaje de Programación

Una aplicación PHP es una aplicación dirigida por eventos. El usuario controla la aplicación que se ejecutara provocando eventos. Por ejemplo: cada acción que realiza el usuario como abrir una aplicación provoca un evento. El lenguaje utilizado para escribir procedimientos se denomina JavaScript pueden incluir sentencias, órdenes y funciones. A continuación, se presenta un procedimiento para la ejecución de un script el cual sucede cada que ejecute una aplicación o un script PHP y que pida datos a la base de datos.

**GRÁFICO N° 17:** Uso de lenguaje de Programación.

```
<?php
$p=$_GET['p'];
$u=$_GET['u'];
?><?php
include("../conexion.php");
$link=conectarse();
$p=$_GET['p'];
$cod_datper=$_GET['p'];
$cod_datos=$_GET['cod_datos'];
$nombres_datos=$_GET['nombres_datos'];
$apellidos_datos=$_GET['apellidos_datos'];
$dni_datos=$_GET['dni_datos'];
$edad_datos=$_GET['edad_datos'];
$lugnac=$_GET['lugnac'];
$direccion=$_GET['direccion'];
$celular=$_GET['celular'];
$correo=$_GET['correo'];
$especialidad=$_GET['especialidad'];
$resultado=mysql_query("INSERT INTO
datospersonales
```

**Fuente:** Elaboración propia.

## Uso del Gestor de Base de Datos

Una aplicación MySQL es una aplicación dirigida por eventos. El usuario controla la aplicación que se ejecutara provocando eventos. Por ejemplo: cada acción que realiza el usuario como abrir una aplicación provoca un evento. El lenguaje utilizado para escribir procedimientos se denomina JavaScript pueden incluir sentencias, órdenes y funciones. A continuación, se presenta un procedimiento para la ejecución de un script el cual sucede cada que ejecute una aplicación o un script PHP y que pida datos a la base de datos.

**GRÁFICO N° 18:** Uso de gestor de Base de Datos.



```
<?php
function conectarse()
{
  if
  (!(($link=mysql_connect("localhost","root","minibanco"))))
  {
    echo "error conectando a la base de dato";
    exit();
  }
  if(!mysql_select_db("sisalpa",$link))
  {
    echo "error seleccionando la base de datos";
    exit();
  }
  return $link;
}
?>
```

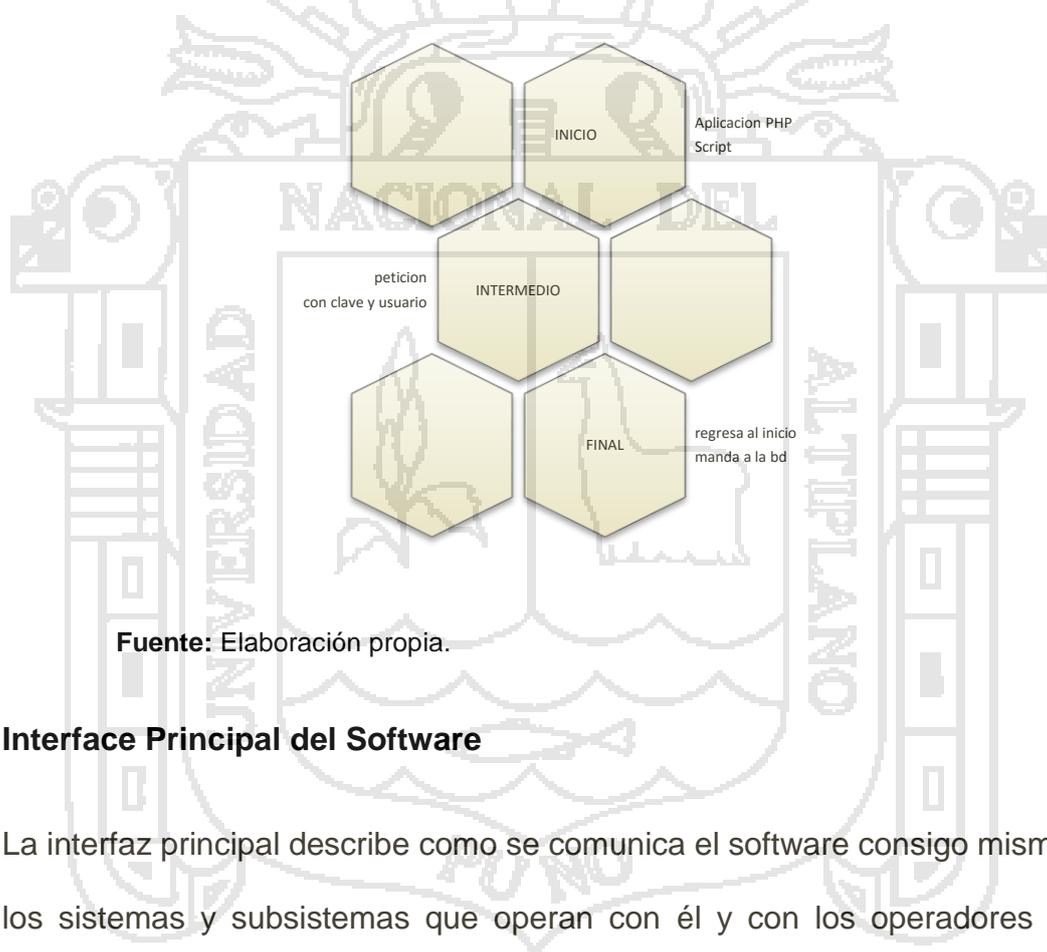
Gráfico 01: Ciclo de vida del Software del Sistema

**Fuente:** Elaboración propia.

### Conectividad Abierta de Base de Datos

El sistema al ejecutar el strip anterior ya hace conexión con la base de datos para así poder obtener los datos necesario para sus respectivos usos como Php Utiliza el Script para tener acceso al Sistema.

**GRÁFICO N° 19:** Conectividad abierta a Base de Datos.



**Fuente:** Elaboración propia.

### Interface Principal del Software

La interfaz principal describe como se comunica el software consigo mismo, con los sistemas y subsistemas que operan con él y con los operadores que lo emplean. Una interfaz implica un flujo de información (por ejemplo datos y/o controles). El diseño de interfaz se concentra en dos áreas importantes:

## 4.2 PRUEBAS DEL SOFTWARE

### 4.2.1 Diseño de las Interfaces Internas del Programa

Este depende de los datos que debe fluir entre los módulos y las características del lenguaje de programación en el que se implementa el software. En general el proceso de análisis contiene mucha información requerida para el diseño de interfaces.

El diagrama de flujo de datos describe como se transforman los objetos de datos al moverse a través del sistema de software, las transformaciones de diagrama de flujo de datos se convierten en módulos dentro de la estructura del programa, por lo tanto, las flechas que fluyen deben convertirse en un diagrama para la interfaz de modulo que corresponda a esa transformación.

### 4.2.2 Diseño de la Interface Hombre Maquina

Las categorías de diseño de interfaz hombre – máquina (IHM) a usar son: Interacción general, visualización de información y la entrada de datos.

#### ➤ **Interacción general**

La interfaz mostrará un formato para la selección de menú, visualización de datos, ofrece respuestas significativas que garanticen la comunicación en los dos sentidos (entre el operador y la interfaz). Reduce la cantidad de información que se debe memorizar para las operaciones.

➤ **Visualización de la información**

No se confunde al operador en la entrada de datos, se utiliza un formato de visualización que le permite una rápida asimilación de la información. Se usó etiquetas consistentes, abreviadas estándares colores predecibles para haber podido apreciar mejor el significado obvio de la información. Se usó ventanas para diferenciar varios tipos de información que han sido requeridos.

➤ **Entrada de datos**

- ✓ Se eliminó las entradas innecesarias (información que puede adquirirse automáticamente o calcularse inmediatamente en el programa).
- ✓ Se dejará al usuario controlar el flujo interactivo de entrada de datos.
- ✓ Minimizar el número de acciones de entrada de datos que necesita el usuario sobre todo reducir cantidad de escritura en la entrada de datos.

### 4.2.3 Prueba De Software Según ISO - 9126

#### PROMEDIO GENERAL DE LAS FICHAS DE EVALUACIÓN ISO – 9126

**TABLA N° 7:** Tabla Fichas ISO - 9126.

Clasificación	Intervalo	Decisión
A) Inaceptable	[ 27 - 54 >	
B) Mínimamente aceptable	[ 54 – 81 >	
C) Aceptable	[ 81 – 95 >	
D) Cumple los requisitos	[ 95 – 122 >	109
E) Excede los requisitos	[ 122 – 135 ]	

**Fuente:** Tablas de validación de software.

Según los resultados el promedio de 109, 110, 108, 112, 110, 107, 122, 90, 110, 112 nos resultó con un promedio de 109, indicando que cumple con los requisitos según ISO – 9126. Todo esto se muestra en el ANEXO.

#### **Decisión:**

De acuerdo a los resultados de la calidad del software se concluyó que el sistema de Registro Genealógico de Alpacas para la Municipalidad Distrital de Ajoyani, cumple los requisitos con un promedio de 109 puntos del total de 135 puntos que se considera en el cuadro de decisiones ISO – 9126.

## CUADRO COMPARATIVO SEGÚN EL TIEMPO EN MINUTOS

TABLA N° 8: Tabla de comparación de tiempos

CONDICIÓN	CON SISTEMA	SIN SISTEMA
Registro de datos	1	30
Búsqueda de datos	2	40
Validación de registro de datos	0.5	20
Impresión de Reportes	0.5	5
TOTAL	4	95

Fuente: Tablas de validación de software

**Decisión:**

De acuerdo al cuadro comparativo expresado en minutos de una inscripción con sistema y sin sistema se aprecia que utilizando un sistema de registro demora 4 minutos. Por animal mientras que sin el sistema de inscripción es de 95 minutos. Generando demasiada pérdida de tiempo.

## CONCLUSIONES

Concluido el diseño, implementación y las pruebas aplicadas en forma preliminar el Sistema de registro genealógico, en este último capítulo se concluye con los resultados obtenidos durante la investigación.

Este trabajo de Software es un trabajo de ingeniería, el cual ayudara a la solución de problemas en la gerencia de desarrollo agropecuario de la municipalidad distrital de ajoyani.

Se acuerdo a los resultados obtenidos se concluye:

- El diseño del sistema de información para el registro de datos genealógicos de las alpacas para la Municipalidad Distrital de Ajoyani permitió la funcionalidad en un sistema multiplataforma.
- La Implementación del sistema de información para el registro de datos genealógicos para la Municipalidad Distrital de Ajoyani, es adecuado y de fácil uso que permite la comunicación y el control de los procesos de producción tales como la calidad seguimiento de empadre controlado y rasgos genéticos de las alpacas.
- Al Evaluar la eficiencia del sistema de registro de datos genealógicos cumple con los estándares de calidad y es un aporte significativo, que causara un impacto positivo en la administración de datos de los trabajadores en la oficina de desarrollo agropecuario y para los productores alpaqueros (en cuanto a rapidez, calidad y precisión de la información que se genera).

## RECOMENDACIONES Y SUGERENCIAS

### Para el Sistema

- Se recomienda adoptar la tecnología cliente/servidor en futuros desarrollos de sistemas de información dentro de la Municipalidad distrital de Ajoyani.
- Se recomienda adquisición de un servidor, con prestaciones mayores en lo que respecta velocidad de proceso, prevención contra apagones, dispositivos de tapes backups, almacenamiento mayor a (1000Gb a más).
- Se recomienda realizar compras de dispositivos de protección de servidor ups, supresor de picos, instalación de puesta a tierra y otros depósitos de protección a la parte del hardware del sistema.
- Es recomendable que en posteriores investigaciones similares realizar o aplicar métricas del software para garantizar la mejor calidad del software, como también conocer su funcionalidad entregada por la aplicaron, de modo que se pueda establecer comparaciones con los resultados obtenidos en el presente trabajo.

### Para los usuarios

- Se recomienda realizar capacitaciones al personal encargado del registro de sistemas.
- Se recomienda capacitar y sensibilizar a cerca de la existencia a los pobladores y comuneros de la zona.

**BIBLIOGRAFÍA**

- Agarwal & Kumar, (2001). *Estimating Software projects. Software engineering Notes*, 26(4), pp. 60 – 67.
- Albrecht, A. (1983). *Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation*. IEEE Trans. Software Eng., Vol. SE-9, No. 6.
- Allison, (2011). *La Biblia de MySQL y Apache para Desarrolladores Web y CMS*, IEEE Trans. O'Reilly.
- Bielak, J. (2000). *Improving Size Estimates Using Historical Data*. IEEE Software, vol.17, no.6, pp. 27-35, Nov./Dec.
- Boehm, B. (1981). *Software Engineering Economics*. PrenticeHall, Upper Saddle River, N.J.
- Boehm, B.; Clark, B.; Horowitz, E.; Westland, C.; Madachy, R.; Selby, R. (1995). *Cost Models for Future Software Life Cycle Process: COCOMO 2.0*. Ann. Of Software J.D. Arther and S. M. Henry, eds., Science Publishers, Amsterdam, The Netherlands, Vol. 1, pp. 45-60.
- Bozoki, G.J. (1991). *Performance Simulation of SSM (Software Sizing Model)*, Proc. 13th Conf., Int' l Soc. of Parametric Analysts, Int' l Soc. Of Parametric Analysts, New Orleans, pp. CM –14.
- Britos, P. (2008). *Procesos de Explotación de Información Basados en Sistemas Inteligentes*. Tesis Doctoral. Facultad de Informática. Universidad Nacional de la Plata.

- Elmasri / Navate (1995) *Sistema de base de datos, Conceptos fundamentales*  
Segunda Edición Addison Wesley Iberoamericana.
- Herrera, L. (2007). *Proyecto de innovación en tutorías*, Proyecto, Facultad de  
Educación y Humanidades, Universidad de Granada, Melilla
- Laboratorio Nacional de Calidad del Software, (2009). *Ingeniería del software:  
metodologías y ciclos de vida, Guía de Ingeniería del Software.*
- Narro, J. (2012). *Sistema institucional de tutoría UNAM*, Proyecto, Universidad  
Nacional Autónoma, México.
- Portugal. R. *Sistemas de información, Teoría e implementación.*
- Pressman R. S. (2000). *Diseño de base de datos relacionales*, Adoración de Miguel  
Castaño AlfaOmega Grupo Editor.
- Pressman R. S. (2001). *Ingeniería de software*, Un enfoque práctico Quinta Edición  
Mc Graw-Hill Interamericana.
- Rumbaugt, J. (2000). *El lenguaje de unificado del modelado*, Manual de referencia,  
Primera Edición Addison Wesley.
- Schmuller J. (2000). *Aprendiendo uml en 24 horas*, Primera Edición Prentice Hall.
- Viaña, J. M. (2010). *Estándares para la acreditación de carreras profesionales  
universitarias de medicina, enfermería y obstetricia*, Modelo de Calidad para la  
Acreditación del CONEAU, Tomo V.

## BIBLIOGRAFÍA WEB

Sistemas de Información [On-Line]. accedido 08 de septiembre del 2014. Disponible en <http://www.econlink.com.ar/sistemas-informacion/definicion>

Wikipedia La Enciclopedia Libre [On-Line]. Accedido 12 de septiembre del 2014. Disponible en <http://es.wikipedia.org/wiki/Turismo>

Modelado de Sistemas de Información [On-Line]. Accedido 20 de septiembre del 2014. Disponible en <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html>.

Wikipedia La Enciclopedia Libre [On-Line]. Accedido 25 de septiembre del 2014. Disponible en.

Desarrollo en Espiral [On-Line]. Accedido el 14 de noviembre del 2014. Disponible en <http://desarrollo-en-espiral.jimdo.com/modelo-en-espiral/>.

Blogdiario.com [On-Line]. Accedido 08 de septiembre del 2014. Disponible en <http://scruz334.blogspot.es/1193169600/el-modelo-en-espiral/>

[http://es.wikipedia.org/wiki/An%C3%A1lisis\\_y\\_dise%C3%B1o\\_orientado\\_a\\_objetos](http://es.wikipedia.org/wiki/An%C3%A1lisis_y_dise%C3%B1o_orientado_a_objetos).

Computación e Informática [On-Line]. Accedido 28 de septiembre del 2014. Disponible en <http://www.rodolfoquispe.org/blog/que-es-la-ingenieria-de-software.php>.

MySQL [On-Line]. Accedido 05 de octubre del 2014. Disponible en [www.mysql.com](http://www.mysql.com).



## ANEXO 01 ESTÁNDAR ISO 9126.

**TABLA N° 9:** Ficha de evaluación de la calidad del producto

INDICADORES	PUNTUACIÓN				
	1	2	3	4	5
<b>1. FUNCIONALIDAD</b>					
<b>Adecuación:</b> la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.					
<b>Exactitud:</b> la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.					
<b>Interoperabilidad:</b> la capacidad del producto software para interactuar con uno o más sistemas especificados					
<b>Seguridad:</b> referido a la capacidad del producto software para proteger la información y los datos					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad					
<b>2. FIABILIDAD</b>					
<b>Madurez:</b> la capacidad del producto software para evitar fallos provocados por errores en el software.					
<b>Tolerancia a fallos:</b> la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.					
<b>Recuperabilidad:</b> la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.					
<b>3. USABILIDAD</b>					
<b>Comprensibilidad:</b> la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.					
<b>Facilidad de aprendizaje:</b> la capacidad del producto software para permitir al usuario aprender su aplicación.					
<b>Atracción:</b> la capacidad del producto software para atraer al usuario.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.					
<b>Operabilidad:</b> la capacidad del producto software para permitir que el usuario lo opere y lo controle.					
<b>4. EFICIENCIA</b>					
<b>Comportamiento temporal:</b> la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.					
<b>Utilización de recursos:</b> la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.					
<b>5. MANTENIBILIDAD</b>					



<b>Analizabilidad:</b> Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.					
<b>Cambiabilidad:</b> Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación.					
<b>Estabilidad:</b> Capacidad del producto software de evitar los efectos inesperados de las modificaciones.					
<b>Facilidad de prueba:</b> Capacidad del producto software de permitir validar las partes modificadas.					
<b>Conformidad:</b> Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.					

INDICADORES	PUNTUACIÓN				
	1	2	3	4	5
<b>6. PORTABILIDAD</b>					
<b>Adaptabilidad:</b> la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.					
<b>Facilidad de instalación:</b> la capacidad del producto software para ser instalado en un ambiente determinado.					
<b>Coexistencia:</b> la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.					
<b>Reemplazabilidad:</b> la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.					
<b>SUB TOTALES</b>					
<b>TOTAL</b>					

Indicador Cualitativo	Valor
Deficiente	1
Malo	2
Regular	3
Bueno	4
Muy bueno	5

Clasificación	Intervalo	Decisión
A) Inaceptable	[ 27 - 54 >	
B) Mínimamente aceptable	[ 54 - 81 >	
C) Aceptable	[ 81 - 95 >	
D) Cumple los requisitos	[ 95 - 122 >	109
E) Excede los requisitos	[ 122 - 135 ]	

**Tabla :iso-9126**

Fuente: tablas de validación de software.

## ANEXO 02

### TABLAS DE LA BASE DE DATOS

#### TABLA N° 10: Usuarios activos e inactivos.

Servidor: localhost ▶ Base de datos: bdalpacos ▶ Tabla: active

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id_active	int(2)			No		auto_increment	[Iconos de acción]
<input type="checkbox"/>	num_active	int(2)			No			[Iconos de acción]
<input type="checkbox"/>	nomb_active	varchar(15)	utf8_general_ci		No			[Iconos de acción]

#### TABLA N° 11: Registro de datos de alpacas.

Servidor: localhost ▶ Base de datos: bdalpacos ▶ Tabla: alpacas

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id_alpacas	int(5)			No		auto_increment	[Iconos de acción]
<input type="checkbox"/>	num_reg	int(10)			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	num_NARETE	int(20)			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_RAZA	int(2)			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_SEXO	varchar(5)	utf8_general_ci		Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_COLOR	varchar(5)	utf8_general_ci		Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_EDAD	varchar(10)	utf8_general_ci		Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_DIAGPRE	varchar(10)	utf8_general_ci		Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_FINURA	float			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_DENS	float			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_RIZO	float			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_CABEZ	float			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_CALCE	float			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_BALGEN	float			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_LMECHA	float			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_PBELLO	float			Sí	NULL		[Iconos de acción]
<input type="checkbox"/>	esqui_PESO_BRAG	float			Sí	NULL		[Iconos de acción]

#### TABLA N° 12: Diccionario de datos de calidad.

Servidor: localhost ▶ Base de datos: bdalpacos ▶ Tabla: bib\_balancegen

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id_balance	int(2)			No		auto_increment	[Iconos de acción]
<input type="checkbox"/>	num_balance	int(2)			No			[Iconos de acción]
<input type="checkbox"/>	nomb_balance	varchar(40)	utf8_general_ci		No			[Iconos de acción]

	←T→	id_balance	num_balance	nomb_balance
<input type="checkbox"/>			1	1 MUY BUENA
<input type="checkbox"/>			2	2 BUENA
<input type="checkbox"/>			3	3 MEDIA
<input type="checkbox"/>			4	4 REGULAR
<input type="checkbox"/>			5	5 MALA

**TABLA N° 13:** Diccionario de datos de calidad calificativo por cabeza.

Servidor: localhost ▶ Base de datos: bdalpacos ▶ Tabla: bib\_cabeza

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id_cabeza	int(2)			No		auto_increment	
<input type="checkbox"/>	num_cabeza	int(2)			No			
<input type="checkbox"/>	nomb_cabeza	varchar(40)	utf8_general_ci		No			

	←T→	id_cabeza	num_cabeza	nomb_cabeza
<input type="checkbox"/>			1	1 MUY BUENA
<input type="checkbox"/>			2	2 BUENA
<input type="checkbox"/>			3	3 MEDIA
<input type="checkbox"/>			4	4 REGULAR
<input type="checkbox"/>			5	5 MALA

**TABLA N° 14:** Diccionario de datos de color de fibra.

Servidor: localhost ▶ Base de datos: bdalpacos ▶ Tabla: bib\_color

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id_color	int(2)			No		auto_increment	
<input type="checkbox"/>	num_color	int(2)			No			
<input type="checkbox"/>	nombre	varchar(40)	utf8_general_ci		No			

			id_color	num_color	nombre
<input type="checkbox"/>			1	1	blanco
<input type="checkbox"/>			2	2	café
<input type="checkbox"/>			3	3	lf1
<input type="checkbox"/>			4	4	lf2
<input type="checkbox"/>			5	5	lf3
<input type="checkbox"/>			6	6	lfx
<input type="checkbox"/>			7	7	gris
<input type="checkbox"/>			8	8	gris oscuro
<input type="checkbox"/>			9	9	negro
<input type="checkbox"/>			10	10	cafex
<input type="checkbox"/>			11	11	cafey
<input type="checkbox"/>			12	12	cafez

**TABLA N° 15:** Diccionario de datos de densidad de fibra.

Servidor: localhost ▶ Base de datos: bdalpacas ▶ Tabla: bib\_densidad

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_densidad	int(2)			No		auto_increment	
<input type="checkbox"/> num_densidad	int(2)			No			
<input type="checkbox"/> nom_densidad	varchar(20)	utf8_general_ci		No			

			id_densidad	num_densidad	nom_densidad
<input type="checkbox"/>			1	1	MUY ALTA
<input type="checkbox"/>			2	2	ALTA
<input type="checkbox"/>			3	3	MEDIA
<input type="checkbox"/>			4	4	BAJA
<input type="checkbox"/>			5	5	MUY BAJA

**TABLA N° 16:** Diccionario de datos de estado de preñez.

Servidor: localhost ▶ Base de datos: bdalpacas ▶ Tabla: bib\_diagprenes

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_diagpre	int(2)			No		auto_increment	
<input type="checkbox"/> num_diagpre	int(2)			No			
<input type="checkbox"/> nombre_pre	varchar(40)	utf8_general_ci		No			

			id_diagpre	num_diagpre	nombre_pre
<input type="checkbox"/>			1	1	PREÑADA
<input type="checkbox"/>			2	2	VACIA

**TABLA N° 17:** Diccionario de datos de estado de edad.

Servidor: localhost ▶ Base de datos: bdalpacas ▶ Tabla: bib\_edad

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id_edad	int(2)			No		auto_increment	
<input type="checkbox"/>	num_edad	int(2)			No			
<input type="checkbox"/>	nombre	varchar(40)	utf8_general_ci		No			

			id_edad	num_edad	nombre
<input type="checkbox"/>			1	1	DLM
<input type="checkbox"/>			2	2	DL
<input type="checkbox"/>			3	3	2D
<input type="checkbox"/>			4	4	4D
<input type="checkbox"/>			5	5	6D
<input type="checkbox"/>			7	7	BLL

**TABLA N° 18:** Diccionario de datos de estado de finura.

Servidor: localhost ▶ Base de datos: bdalpacas ▶ Tabla: bib\_finura

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id_finura	int(2)			No		auto_increment	
<input type="checkbox"/>	num_finura	int(2)			No			
<input type="checkbox"/>	nom_finura	varchar(40)	utf8_general_ci		No			

			id_finura	num_finura	nom_finura
<input type="checkbox"/>			1	1	EXRAFINA
<input type="checkbox"/>			2	2	FINA
<input type="checkbox"/>			3	3	SEMI FINA
<input type="checkbox"/>			4	4	GRUESA
<input type="checkbox"/>			5	5	MUY GRUESA

**TABLA N° 19:** Diccionario de datos de estado de sectores del distrito de Ajoyani.

Servidor: localhost ▶ Base de datos: bdalpacas ▶ Tabla: bib\_sector

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id_sector	int(2)			No		auto_increment	
<input type="checkbox"/>	num_sector	int(2)			No			
<input type="checkbox"/>	nombre_sector	varchar(90)	utf8_general_ci		No			

←T→	id_sector	num_sector	nombre_sector
<input type="checkbox"/>	1	1	callpuyo
<input type="checkbox"/>	2	2	puerto arturo
<input type="checkbox"/>	3	3	añoccaya
<input type="checkbox"/>	4	4	ajoyani
<input type="checkbox"/>	5	5	alto alianza
<input type="checkbox"/>	6	6	alianza
<input type="checkbox"/>	7	7	fundo los andes

**TABLA N° 20:** Diccionario de datos del Sexo del Animal.

Servidor: localhost ▶ Base de datos: bdalpacas ▶ Tabla: bib\_sexo

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_sexo	int(2)			No		auto_increment	
<input type="checkbox"/> num_sexo	int(2)			No			
<input type="checkbox"/> nombre	varchar(20)	utf8_general_ci		No			

←T→	id_sexo	num_sexo	nombre
<input type="checkbox"/>	1	1	macho
<input type="checkbox"/>	2	2	hembra

**TABLA N° 21:** Diccionario de datos de los niveles de acceso al sistema.

Servidor: localhost ▶ Base de datos: bdalpacas ▶ Tabla: nivel

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_nivel	int(2)			No		auto_increment	
<input type="checkbox"/> num_nivel	int(2)			No			
<input type="checkbox"/> nomb_nivel	varchar(15)	utf8_general_ci		No			

←T→	id_nivel	num_nivel	nomb_nivel
<input type="checkbox"/>	1	1	administrador
<input type="checkbox"/>	2	2	sub administrad
<input type="checkbox"/>	3	3	usuario
<input type="checkbox"/>	4	4	nivel bajo

**TABLA N° 22:** Registro de Usuarios del sistema.

Servidor: localhost ▶ Base de datos: bdalpacas ▶ Tabla: usuarios

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id_usuario	int(3)			No		auto_increment	
<input type="checkbox"/>	username	varchar(10)	utf8_general_ci		No			
<input type="checkbox"/>	password	varchar(10)	utf8_general_ci		No			
<input type="checkbox"/>	email	varchar(20)	utf8_general_ci		No			
<input type="checkbox"/>	active	int(3)			No			
<input type="checkbox"/>	level	int(3)			No			
<input type="checkbox"/>	random	varchar(20)	utf8_general_ci		No			

	id_usuario	username	password	email	active	level	random
<input type="checkbox"/>	1	Ernesto Al	123	ernestoalanoca@hotmail.com	1	1	random
<input type="checkbox"/>	2	walcheber	12345	walche@gmail.com	0	4	7
<input type="checkbox"/>	3	edgar	123	edgar@hotmail.com	0	2	12313546



**Anexo N°03**



**MUNICIPALIDAD DISTRITAL DE AJOYANI – CARABAYA**

# GERENCIA DE DESARROLLO SOCIAL Y AGROPECUARIO

La presente encuesta se realiza con un fin de investigar, por lo se ruega pueda brindar los datos reales que usted vea conveniente por lo cual se le ruega marcar con una (x) la alternativa que vea conveniente.

- Usted cree que, con la implementación de un Sistema de Registro Genealógico para agilizar los procesos de empadre controlado y verificación de rasgos genealógicos, mejore la gestión Administrativa del Municipio.
  - Sí.
  - No.
- Usted cree que, con la implementación de un Sistema de Registro genealógico para agilizar los procesos de empadre y rasgos genéticos, mejore la gestión Administrativa del Municipio.
  - Sí.
  - No.

3. ¿Cómo Ud. considera el servicio de registro de rasgos genéticos en la oficina de desarrollo agropecuario de la Municipalidad distrital de Ajoyani?
- Eficiente
  - Medianamente Eficiente
  - Deficiente

c.c.Arch.  
c.Alc.  
GDAMD.avc

La Municipalidad distrital de Ajoyani agradece su colaboración



## Anexo N°04

MUNICIPALIDAD DISTRITAL DE AJOYANI – CARABAYA



# GERENCIA DE DESARROLLO SOCIAL Y AGROPECUARIO

La presente encuesta se realiza con un fin de investigar para implementar un Sistema Informático de Registro Genealógico de alpacas (SRA) que tenga la finalidad de agilizar cualquier trámite documentario, por lo se ruega pueda brindar los datos reales que usted vea conveniente por lo cual se le ruega marcar con una (x) la alternativa que vea conveniente.

- ¿Cómo considera Ud. ¿Los servicios que ofrece el Sistema de Registro Genealógico de alpacas?
  - Eficiente
  - Medianamente eficiente
  - Deficiente
- ¿Cómo considera Ud. La interacción con el Sistema del “SRA”?
  - Muy Fácil.
  - Fácil.
  - Difícil
- ¿Cómo considera Ud. el diseño de la interface SRA?

- a) Muy bueno
- b) Bueno
- c) Regular

4. ¿Cree Ud. Que los reportes del SRA son confiables?

- a) Si
- b) No

La Municipalidad distrital de Ajoyani agradece su colaboración

c.c.Arch.  
c.Alc.  
GDAMD.avc

## Anexo N°04

### Manual de Usuario

#### Ventana principal de entrada

Para iniciar a usar el sistema se tiene que iniciar primeramente entrando a un link con un navegador libre en HTML escribiendo el siguiente URL en la pestaña de navegación localhost/sisalpa/luego de escribir se sigue los link de preferencia a la que se quiere ingresar

**GRÁFICO N° 20:** Ventana principal de entrada.

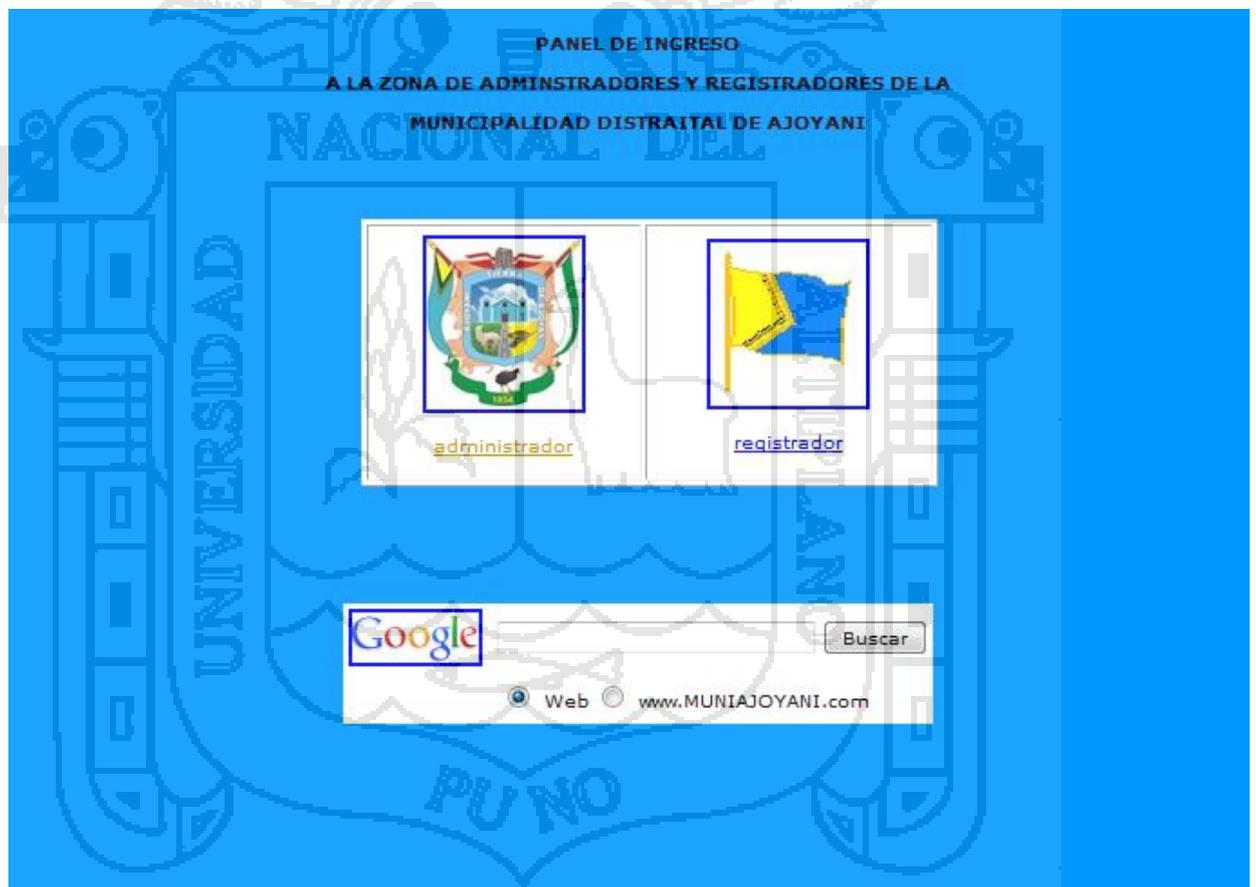


**Ventana de loguin administrador**

Para iniciar con un loguin y entrar a diferentes zonas autorizadas se tiene que iniciar primeramente entrando a un link con un navegador libre en HTML escribiendo el siguiente URL en la pestaña de navegación localhost/sisalpa/ luego para que aparezca la venta del loguin se sigue los siguientes pasos.

1. Ingresar a la ventana principal
2. Hacer click en “ENTRAR Y LOGUEARSE”
3. Hacer clic en **administrador**
4. Escribir el pasword y usuario proporcionado
5. Realizar diferentes tareas de su conveniencia

**GRÁFICO N° 21:** Ventana de loguin general del sistema.



**GRÁFICO N° 22:** Ventana de login Administrador.

Panel de Login del Administrador

Usuario:	
Password:	
	<input type="button" value="ingresar"/>

### Ventana de loginregistrator

Para iniciar con un login registrator y entrar a diferentes zonas autorizadas se tiene que iniciar primeramente entrando a un link con un navegador libre en HTML escribiendo el siguiente URL en la pestaña de navegación localhost/sisalpa/ luego para que aparezca la venta del login se sigue los siguientes pasos.

1. Ingresar a la ventana principal
2. Hacer click en “ENTRAR Y LOGUEARSE”
3. Hacer click en **registrator**
4. Escribir el password y usuario proporcionado
5. Realizar diferentes tareas de su conveniencia

**GRÁFICO N° 23:** Ventana de login registrator.

Panel de Login del  
REGISTRADOR

Usuario:	
Password:	
	<input type="button" value="ingresar"/>

### Ventanas de acceso a Administrador del sistema

Para iniciar como administrador del sistema y entrar a diferentes zonas ADMINISTRATIVAS se tiene que iniciar primeramente entrando a un link con un navegador libre en HTML escribiendo el siguiente URL en la pestaña de navegación localhost/sisalpa/ se sigue los siguientes pasos.

1. Ingresar a la ventana principal
2. Hacer click en “ENTRAR Y LOGUEARSE” con usuario de administrador

3. Escribir el password y usuario proporcionado
4. Realizar diferentes tareas de su conveniencia con funciones de "ADMINISTRACIÓN"

**GRÁFICO N° 24:** Administrador del sistema en balance general.

**MUNICIPALIDAD DISTRITAL DE AJOYANI  
OFICINA DE DESARROLLO AGROPECUARIO  
SISTEMA DE REGISTRO GENEALOGICO**

**VENTANA PRINCIPAL PARA EL ADMINISTRADOR**

salir active balancegen **cabeza** calce color densidad edad riso nivel raza sector usuarios

Balance General

**Bib\_balancegen - 1 a 5 de 5**

editar borrar 1 ▼ añadir nuevo

<< < > >>

[Mostrar todo registros](#) [Mostrar filtro](#)

<input type="checkbox"/>	No.	Num_balance ▲	Nomb_balance		
<input type="checkbox"/>	1	1	MUY BUENA	editar	borrar
<input type="checkbox"/>	2	2	BUENAs	editar	borrar
<input type="checkbox"/>	3	3	MEDIA	editar	borrar

**GRÁFICO N° 25:** Administrador del sistema en usuarios.

**MUNICIPALIDAD DISTRITAL DE AJOYANI  
OFICINA DE DESARROLLO AGROPECUARIO  
SISTEMA DE REGISTRO GENEALOGICO**

**VENTANA PRINCIPAL PARA EL ADMINISTRADOR**

salir active balancegen cabeza calce color densidad edad riso nivel raza sector **usuarios**

**Usuarios - 1 a 10 de 34**

editar borrar

<< < > >>

<input type="checkbox"/>	No.	Us_codigo ▲	Us_nombreyapell	Us	Us_username	Us_pasword	Us_email	Us
<input type="checkbox"/>	1	30838	ALANOCA ANCHAPURI, E...	42203637	alanoca	alanoca	alanoca@hotmail.com	951
<input type="checkbox"/>	2	30839	PUMA VILLALBA, edgar	42203638	edgar	edgar	edagr@hotmail.com	951
<input type="checkbox"/>	3	30841	CHAMBI CONDORI, Walc...	422036310	walcheber	walcheber	chambiw@hotmail.com	951
<input type="checkbox"/>	4	30842	ALEJO MAMANI Pedro	42203630	alejo	alejo	alejo@hotmail.com	951
<input type="checkbox"/>	5	30843	CHAMBI TOLEDO, Inoce...	45573293	chambi	chambi	chambi@hotmail.com	951
<input type="checkbox"/>	6	30844	MASIAS MAMANI, Calix...	40222825	masias	masias	masias@hotmail.com	951
<input type="checkbox"/>	7	30845	AQUINO CHIPANA Serap...	40414788	aquino	aquino	aquino@hotmail.com	951
<input type="checkbox"/>	8	30846	MAMANI CHOQUECALLATA,	41699858	mamani	mamani	mamani@hotmail.com	951
<input type="checkbox"/>	9	30847	FONSECA MACIAS Joel	42642284	fonseca	fonseca	fonseca@hotmail.com	951

**GRÁFICO N° 26:** Administrador del sistema en edición de usuarios.

**MUNICIPALIDAD DISTRITAL DE AJOYANI  
OFICINA DE DESARROLLO AGROPECUARIO**

**SISTEMA DE REGISTRO GENEALOGICO**

**VENTANA PRINCIPAL PARA EL ADMINISTRADOR**

salir active balancegen cabeza calce color densidad edad riso nivel raza sector usuarios

**Actualizar Usuarios**

Us_codigo:	30839
Us_nombreyapell:	PUMA VILLALBA, edgar
Us:	42203638
Us_username:	edgar
Us_password:	edgar
Us_email:	edagr@hotmail.com
Us_celular:	951235682
Us_direccion:	av. Panamahhhh
Us_active:	1

### Ventana de búsqueda de datos por código de nombre de propietario

Para realizar búsquedas por código, nombre de propietario también se puede realizar similar al anterior el cual puede realizarla un doctor, técnico o cualquier persona que dese ingresar al sistema y entrar a la zona de búsquedas se tiene que iniciar primeramente entrando a un link con un navegador libre en HTML conectados con el servidor escribiendo el siguiente URL en la pestaña de navegación localhost/sisalpa/ se sigue los siguientes pasos.

1. Ingresar a la ventana principal
2. Hacer click en "ENTRAR Y LOGUEARSE" con usuario de administrador
3. Escribir el pasword y usuario proporcionado
4. Hacer clic en buscador por código o nombre de propietario.
5. Ingresar el código del arete
6. Realizar diferentes tareas de su conveniencia con funciones búsqueda de datos de alpacas, propietarios, empadres, etc.

**GRÁFICO N° 27:** Ventana de búsqueda nombre de propietario.

MUNICIPALIDAD DISTRITAL DE AJOYANI OFICINA DE DESARROLLO AGROPECUARIO				
SISTEMA DE REGISTRO GENEALOGICO				
Ventana Principal para el registrador				
salir	busca nombre	buscar por DNI	registrar	administracion
Codigo		ale		
CODIGO	NOMBRE	DNI	DIRECCION	CELULAR
30842	ALEJO MAMANI Pedro	42203630	JR. alaves	951235689
30849	TORRES MAMANI Valeriano	40379204	inambaria	951235696
30852	ALEGRA YUCRA Nicodemo E.	71706837	viluyo	951235699
30853	ALEGRE QUECARA Nicolas	01558105	Jr alamos	951235710

**Ventana de búsqueda de datos por código de nombre de propietario**

Para realizar búsquedas por código, nombre de propietario también se puede realizar similar al anterior el cual puede realizarla un doctor, técnico o cualquier persona que dese ingresar al sistema y entrar a la zona de búsquedas se tiene que iniciar primeramente entrando a un link con un navegador libre en HTML conectados con el servidor escribiendo el siguiente URL en la pestaña de navegación localhost/sisalpa/ se sigue los siguientes pasos.

1. Ingresar a la ventana principal
2. Hacer click en “ENTRAR Y LOGUEARSE” con usuario de administrador
3. Escribir el pasword y usuario proporcionado
4. Hacer clic en buscador por código o nombre de propietario.
5. Ingresar el código del arete
6. Realizar diferentes tareas de su conveniencia con funciones búsqueda de datos de alpacas, propietarios, empadres, etc.

**GRÁFICO N° 28:** Ventana de búsqueda por DNI.

MUNICIPALIDAD DISTRITAL DE AJOYANI OFICINA DE DESARROLLO AGROPECUARIO				
SISTEMA DE REGISTRO GENEALOGICO				
Ventana Principal para el registrador				
salir	busca nombre	buscar por DNI	registrar	administracion
Codigo		4		
« Previous 1 2 Next »				
CODIGO	NOMBRE	DNI	DIRECCION	CELULAR
30839	PUMA VILLALBA, edgar	42203638	av. Panama	951235682
30840	CHUCUYA GOMEZ, hugo	42203639	av. Inambari	951235681
30841	CHAMBI CONDORI, Walcheber	422036310	JR. alaves	951235690
30842	ALEJO MAMANI Pedro	42203630	JR. alaves	951235689
30843	CHAMBI TOLEDO, Inocencio Nazar	45573293	JR. alaves	951235690
30844	MASIAS MAMANI, Calixto	40222825	JR. alaves	951235691
30845	AQUINO CHIPANA Serapio	40414788	inambaria	951235692
30846	MAMANI CHOQUECALLATA	41600000	inambaria	951235693

**Ventanas de reportes del sistema**





**GRÁFICO N° 32:** Ventana de registro de nuevos datos.

nuevo

SISTEMA DE REGISTRO GENEALOGICO  
DE DATOS DE LOS ANIMALES A REGISTRAR

DATOS PUNTALES DEL DUEÑO DEL ANIMAL

DATOS GENERALES

ARETE	RAZA	SEXO	COLOR	EDAD	DIAGN PREÑEZ	FINURA	DENSIDAD
	suri	macho	blanco	DLM	PREÑADA	EXRAFINA	MUY ALTA

DATOS GENERALES 2

RIZO	CABEZA	CALCE	BALANZE GENERAL	LONG MECHA	LONG BELLO	PESO BRAGA	PESO TOTAL
MUY BUENA	MUY BUENA	MUY BUENA	MUY BUENA	kg	cm	kg	kg

DATOS DE PROCEDENCIA

SECTOR	PROPIETARIO	ORFECHA DE NACIMIENTO	REGISTRAD
calpuyo	ALEJO MAMANI Pedro	/ /	PUMA VILLALBA, edgar

FECHAS

RECHA DE REGISTRO			GRABAR
/ /	<input type="button" value="buscar"/>		<input type="button" value="GRABAR"/>

nuevo    buscar nom    eliminar    modificar

### Ventana de edición de datos

Para de datos se puede realizar similar al anterior el cual puede realizarla un doctor, técnico o cualquier persona que dese ingresar al primeramente se tiene que iniciar entrando a un link con un navegador libre en HTML conectados con el servidor escribiendo el siguiente URL en la pestaña de navegación localhost/sisalpa/ se sigue los siguientes pasos.

1. Ingresar a la ventana principal
2. Hacer clic en "ENTRAR Y LOGUEARSE" con usuario de administrador o personal autorizado
3. Escribir el password y usuario proporcionado
4. Hacer clic edición de datos.
5. Editar los datos.
6. Hacer clic en grabar datos

**GRÁFICO N° 33:** Ventana de edición de datos.

nuevo

SISTEMA DE REGISTRO GENEALÓGICO  
DE DATOS DE LOS ANIMALES A REGISTRAR

**DATOS PUNTUALES DEL DUEÑO DEL ANIMAL**

<b>APELLIDO PATERNO</b>	<b>APELLIDO MATER</b>	<b>NOMBRES</b>				<b>BUSQUEDA CODIGO</b>
						buscar

**DATOS GENERALES**

<b>DISTRITO</b>	<b>SECTOR</b>	<b>SEXO</b>	<b>RAZA</b>	<b>ARETE</b>		
<b>AÑO</b>	<b>MES</b>	<b>DÍA</b>		<b>REGISTRADOR</b>		

**REMUNERACIONES**

<b>BALANCE</b>	<b>CABEZA</b>	<b>CALCE</b>	<b>COLOR</b>	<b>DENSIDAD</b>		

**DESCUENTOS**

<b>FINURA</b>	<b>RISO</b>	<b>DIAG-PRENEZ</b>	<b>EDAD</b>			
<b>OBSERVACIONES</b>						
						GRABAR

nuevo    buscar non    eliminar    modificar



## Anexo N°05

Código fuente de los diferentes scripts  
Código fuente de ingreso de datos

```

<?php
include('.././conexion.php');
$link=conectarse();
?>

<link href="estilos.css" rel="stylesheet" type="text/css">
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" /><center><input type="button"
value="nuevo" ></center>
<form name="form" action="insertarv.php" method="get">
<table width="860" border="1" align="center"
class="cuadro">
<tr height="17">
<td colspan="12" class="encabezado"><div
align="center" class="Estilo1">
<div align="center" class="h3">SISTEMA DE
REGISTRO GENEALOGICO</div>
<div align="center" class="h3">DE DATOS DELOS
ANIMALES A REGISTRAR</div>
</div></td>
</tr>
<tr height="17">
<td colspan="9"
class="cabeceracajas"><strong>DATOS PUNTUALES
DELDUE&Ntilde;O DEL ANIMAL</strong></td>
</tr>
<tr height="17">
<td colspan="9"
class="cabeceracajas"><strong>DATOS
GENERALES</strong></td>
</tr>
<tr>
<td width="103" class="CUERPO"><span
class="h3">ARETE</span></td>
<td width="101" class="CUERPO"><span
class="h3">RAZA</span></td>
<td class="CUERPO"><span
class="h3">SEXO</span></td>
<td class="CUERPO"><span
class="h3">COLOR</span></td>
<td width="100" class="CUERPO"><span
class="h3">EDAD</span></td>
<td class="CUERPO"><span class="h3">DIAGN
PRE&Ntilde;EZ</span></td>

```

```

        <td class="CUERPO"><span
class="h3">FINURA</span></td>
        <td class="CUERPO"><span
class="h3">DENSIDAD</span></td>
    </tr>
    <tr height="17">
        <td class="CUERPO"><input name="num_ARETE"
class="hhh1" id="c3424" value="" size="15" /></td>
        <td class="CUERPO"><select class="hhh1"
name="esqui_RAZA" >
            <?php
            $dg1="SELECT
            rs.`nd_raza`,
            rs.`nomb_raza`

            FROM
            raza as rs
            ";
            $jedg1=mysql_query($dg1,$link);
            while($row=mysql_fetch_array($jedg1)){
            echo "<option
            value=".$row['nd_raza'].">".$row['nomb_raza']. "</option>";
            }?>
            </select></td>
        <td class="CUERPO"><select class="hhh1"
name="esqui_SEXO" >
            <?php
            $dg1="SELECT
            sx.`id_sexo`,
            sx.`num_sexo`,
            sx.`nombre`

            FROM
            bib_SEXO as sx
            ";
            $jedg1=mysql_query($dg1,$link);
            while($row=mysql_fetch_array($jedg1)){
            echo "<option
            value=".$row['id_sexo'].">".$row['nombre']. "</option>";
            }?>
            </select></td>
        <td class="CUERPO"><select class="hhh1"
name="esqui_COLOR" >
            <?php
            $dg1="SELECT
            cr.`num_color`,
            cr.`nombre`

            FROM
            bib_color as cr

```

```

";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=".$row['num_color'].>".$row['nombre']."</option>";
}?>
</select></td>
<td class="CUERPO"><select class="hhh1"
name="esqui_EDAD" >
<?php
$dg1="SELECT
ed.`num_edad`,
ed.`nombre`

FROM
bib_edad as ed
";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=".$row['num_edad'].>".$row['nombre']."</option>";
}?>
</select></td>
<td class="CUERPO"><select class="hhh1"
name="esqui_DIAGPRE" >
<?php
$dg1="SELECT
dp.`num_diagpre`,
dp.`nombre_pre`

FROM
bib_diagprenes as dp
";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=".$row['num_diagpre'].>".$row['nombre_pre']."</option>";
}?>
</select></td>
<td class="CUERPO"><select class="hhh1"
name="esqui_FINURA" >
<?php
$dg1="SELECT
fn.`num_finura`,
fn.`nom_finura`

FROM
bib_finura as fn
";
$ejedg1=mysql_query($dg1,$link);

```

```

while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=".$row['num_finura'].">".$row['nom_finura']."</option>";
}?>
</select></td>
<td class="CUERPO"><select class="hhh1"
name="esqui_DENS" >
<?php
$dg1="SELECT
dn.`num_densidad`,
dn.`nom_densidad`

FROM
bib_densidad as dn
";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=".$row['num_densidad'].">".$row['nom_densidad']."</optio
n>";
}?>
</select></td>
</tr>
<tr>
<td colspan="9"
class="cabeceracajas"><strong>DATOS GENERALES
2</strong></td>
</tr>
<tr height="17">
<td class="CUERPO"><span
class="h3">RIZO</span></td>
<td class="CUERPO"><span
class="h3">CABEZA</span></td>
<td class="CUERPO"><span
class="h3">CALCE</span></td>
<td class="CUERPO"><span class="h3">BALANZE
GENERAL</span></td>
<td class="CUERPO"><span class="h3">LONG
MECHA</span></td>
<td class="CUERPO"><span class="h3">LONG
BELLO</span></td>
<td class="CUERPO"><span class="h3">PESO
BRAGA</span></td>
<td class="CUERPO"><span class="h3">PESO
TOTAL</span></td>
</tr>
<tr height="17">
<td class="CUERPO"><select class="hhh1"
name="esqui_RIZO" >
<?php

```

```

$dg1="SELECT
rz.`num_riso`,
rz.`nom_riso`

FROM
bib_riso as rz
";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=".$row['num_riso'].>".$row['nom_riso'].</option>";
}?>
</select></td>
<td class="CUERPO"><select class="hhh1"
name="esqui_CABEZ" >
<?php
$dg1="SELECT
cbz.`num_cabeza`,
cbz.`nomb_cabeza`

FROM
bib_cabeza as cbz
";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=".$row['num_cabeza'].>".$row['nomb_cabeza'].</option>"
;
}?>
</select></td>
<td class="CUERPO"><select class="hhh1"
name="esqui_CALCE" >
<?php
$dg1="SELECT
clz.`num_calce`,
clz.`nomb_calce`

FROM
bib_calce as clz
";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=".$row['num_calce'].>".$row['nomb_calce'].</option>";
}?>
</select></td>
<td class="CUERPO"><select class="hhh1"
name="esqui_BALGEN" >
<?php
$dg1="SELECT

```

```

bgn.`num_balance`,
bgn.`nomb_balance`

FROM
bib_balancegen as bgn
";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=".$row['num_balance'].>".$row['nomb_balance'].</option
>";
}??>
</select></td>
<td class="CUERPO"><input name="esqui_LMECHA"
class="hhh1" id="c3427" value="          kg" size="15" /></td>
<td class="CUERPO"><input name="esqui_PBELLO"
class="hhh1" id="c3428" value="          cm" size="15" /></td>
<td class="CUERPO"><input
name="esqui_PESO_BRAG" class="hhh1" id="c3428" value="
kg" size="15" /></td>
<td class="CUERPO"><input
name="esqui_PESO_TOTAL" class="hhh1" id="c" value="
kg" size="15" /></td>
</tr>
<tr>
<td colspan="9"
class="cabeceracajas"><strong>DATOS DE
PROCEDENCIA</strong></td>
</tr>
<tr height="17">
<td class="CUERPO" colspan="2"><span
class="h3">SECTOR</span></td>

<td class="CUERPO" colspan="2"><span
class="h3">PROPIETARIO</span></td>

<td class="CUERPO" colspan="2"><span
class="h3">ORFECHA DE NACIMIENTO</span></td>

<td class="CUERPO" colspan="2"><span
class="h3">REGISTRAD</span></td>
</tr>
<tr height="17">
<td class="CUERPO" colspan="2"><select
class="hhh123456" name="esqui_SECTOR" >
<?php
$dg1="SELECT
sec.`num_sector`,
sec.`nombre_sector`

```

```

FROM
bib_sector as sec
";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=" . $row['num_sector'] . ">" . $row['nombre_sector'] . "</option>"
;
}??>
</select></td>

<td class="CUERPO" colspan="2"><select
class="hhh123456" name="propietario" >
<?php
$dg1="SELECT
us.`us_codigo`,
us.`us_nombreyapell`

FROM
usuarios as us
where
(
us.`us_level`=4
)
";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo "<option
value=" . $row['us_codigo'] . ">" . $row['us_nombreyapell'] . "</option>"
";
}??>
</select></td>

<td class="CUERPO" colspan="2"><input
name="fech_nacimiento" class="hhh1" id="c2" value=" / / "
size="15" /></td>

<td class="CUERPO" colspan="2"><select
class="hhh123456" name="esqui_SECTOR2" >
<?php
$dg1="SELECT
us.`us_codigo`,
us.`us_nombreyapell`,
us.`us_level`

FROM
usuarios as us

where

```

```

(
us.`us_level`=3
)
";
";
$ejedg1=mysql_query($dg1,$link);
while($row=mysql_fetch_array($ejedg1)){
echo " <option
value=".$row['us_codigo'].>".$row['us_nombreapellido'].</option>
";
}??>
</select></td>
</tr>
<tr>
<td colspan="9"
class="cabeceracajas"><strong>FECHAS</strong></td>
</tr>
<tr height="17">
<td class="CUERPO" colspan="2"><span
class="h3">RECHA DE REGISTRO</span></td>
<td class="CUERPO">&nbsp;</td>
<td class="CUERPO">&nbsp;</td>
<td class="CUERPO">&nbsp;</td>
<td class="CUERPO">&nbsp;</td>
<td class="CUERPO">&nbsp;</td>
<td class="CUERPO">&nbsp;</td>
</tr>
<tr height="17">
<td class="CUERPO" colspan="2"><input
name="fecha_registro" class="hhh1" id="c3" value=" / / "
size="15" /></td>
<td class="CUERPO"><input name="button22222"
type="reset" class="BOTON"
onclick="location.href('buscadorporcodigo.php')" value=" buscar "
/></td>
<td class="CUERPO">&nbsp;</td>
<td class="CUERPO">&nbsp;</td>
<td class="CUERPO">&nbsp;</td>
<td class="CUERPO">&nbsp;</td>
<td class="CUERPO"><input name="button22222"
type="submit" class="BOTON" value=" GRABAR " /></td>
</tr>
</table>
</form>

<center>
<table width="200" border="1" align="center"
class="CUERPO">
<tr>
<td>

```

```
<div align="center">
  <input name="button222" type="reset"

  <td>
    <div align="center">
      <input name="button222" type="reset"
class="BOTON" value="eliminar ">
    </div>
  </td> <td>
    <div align="center">
      <input name="button222" type="reset"
class="BOTON"
onclick="location.href('../..../buscador/buscadormodificar/buscador.
php')" value="modificar ">
    </div>
  </td>
</tr>
</table>
```



## Anexo N°06

Código fuente de ingreso de validación de datos

```

<?php
include('../conexion.php');
$link=conectarse();
$p=$_GET['p'];
$num_ARETE=$_GET['num_ARETE'];
$esqui_RAZA=$_GET['esqui_RAZA'];

$resultado=mysql_query("INSERT INTO alpacas
(num_ARETE,esqui_RAZA,esqui_SEXO,esqui_COLOR,esqui_EDAD,esqui_D
IAGPRE,esqui_FINURA,esqui_DENS,esqui_RIZO,esqui_CABEZ,esqui_CALC
E,esqui_BALGEN,esqui_LMECHA,esqui_PBELLO,esqui_PESO_BRAG,esqui
_PESO_TOTAL,esqui_SECTOR,propietario,fech_nacimiento,registrador,fech
a_registro)
VALUES('$num_ARETE','$esqui_RAZA','$esqui_SEXO','$esqui_COLOR','$es
qui_EDAD','$esqui_DIAGPRE','$esqui_FINURA','$esqui_DENS','$esqui_RIZO
','$esqui_CABEZ','$esqui_CALCE','$esqui_BALGEN','$esqui_LMECHA','$esq
ui_PBELLO','$esqui_PESO_BRAG','$esqui_PESO_TOTAL','$esqui_SECTOR'
','$propietario','$fech_nacimiento','$registrador','$fecha_registro')",$link);
if($resultado)
{
header("location: cursos.php?p=$p");
}
else
{
<input name="button222" type="reset" class="BOTON"
onclick="location.href('insertar.php')" value=" nuevo " />
</div></td>
<td><div align="center">
<input name="button222" type="reset" class="BOTON"
onclick="location.href('../buscador/buscadoreeditar/buscador.php')" value="
buscar nom" />
</div></td>
<td><div align="center">
<input name="button222" type="reset" class="BOTON" value="eliminar
" />
</div></td>
<td><div align="center">
<div align="center">
<input name="button2223" type="reset" class="BOTON"
onclick="location.href('../buscador/buscadormodificar/buscador.php')"
value="modificar " />
</div>
</div></td>
</tr>
</table>
</center>

```

## Anexo N°07

Código de consultas de datos según manejador de base de datos MySQL  
Maestro

### CÓDIGO DE CONSULTA DE CARACTERÍSTICAS BÁSICAS DEL ANIMAL

```

SELECT
al.`num_ARETE`,
al.`esqui_RAZA`,
rs.`nomb_raza`,
al.`esqui_SEXO`,
sx.`nombre`,
al.`esqui_EDAD`,
ed.`nombre`,
al.`esqui_DIAGPRE`,
pre.`nombre_pre`,
al.`propietario`,
usr.`us_nombreyapell`
FROM
alpacas as al,
raza as rs,
bib_sexo as sx,
bib_edad as ed,
bib_diagprenes as pre,
usuarios as usr
where
(
al.`propietario`='30841' and
al.`esqui_RAZA` = rs.`num_raza` and
al.`esqui_SEXO` = sx.`num_sexo` and
al.`esqui_EDAD` = ed.`num_edad` and
al.`esqui_DIAGPRE` = pre.`num_diagpre` and
al.`propietario` = usr.`us_codigo`
)

```

### CONSULTA DE BASE DE DATOS PARA ASPECTOS GENEALÓGICOS DE FIBRA

```

SELECT
al.`esqui_COLOR`,
col.`nombre`,
al.`esqui_DENS`,
den.`nom_densidad`,
al.`esqui_RIZO`,
ris.`nom_riso`,

```

```

al.`esqui_CABEZ`,
cab.`nomb_cabeza`,
al.`esqui_CALCE`,
cal.`nomb_calce`,
al.`esqui_BALGEN`,
blg.`nomb_balance`,
al.`esqui_LMECHA`,
al.`esqui_PBELLO`
FROM
alpacas as al,
bib_color as col,
bib_densidad as den,
bib_riso as ris,
bib_cabeza as cab,
bib_calce as cal,
bib_balancegen as blg
where
(
al.`propietario`='30842' and
al.`esqui_COLOR` = col.`num_color` and
al.`esqui_DENS` = den.`num_densidad` and
al.`esqui_RIZO` = ris.`num_riso` and
al.`esqui_CABEZ` = cab.`num_cabeza` and
al.`esqui_CALCE` = cal.`num_calce` and
al.`esqui_BALGEN` = blg.`num_balance`
)

```

#### CONSULTA DE DATOS GENERALES ACERCA SOBRE EL PESO

```

select
al.`esqui_peso_brag`,
al.`esqui_peso_total`,
al.`esqui_sector`,
al.`propietario`,
usrs.`us_nombreyapell`,
al.`fecha_nacimiento`,
al.`registrador`,
al.`fecha_registro`
from
alpacas as al,
usuarios as usrs

where
(
al.`propietario`='30842' and
al.`propietario` = usrs.`us_codigo`
)

```

## CONSULTA DATOS GENERALES

```
SELECT
al.`esqui_RAZA`,
rs.`nomb_raza`,
al.`esqui_SEXO`,
sx.`nombre`,
al.`esqui_EDAD`,
ed.`nombre`,
al.`esqui_DENS`,
den.`nom_densidad`,
al.`propietario`,
usr.`us_nombreyapell`
FROM
alpacas as al,
raza as rs,
bib_sexo as sx,
bib_edad as ed,
bib_densidad as den,
usuarios as usr
where
(
al.`propietario`='30846' and
al.`esqui_RAZA` = rs.`num_raza` and
al.`esqui_SEXO` = sx.`num_sexo` and
al.`esqui_EDAD` = ed.`num_edad` and
al.`esqui_DENS` = den.`num_densidad` and
al.`propietario` = usr.`us_codigo`
)
```