

**SISTEMA DE MONITOREO REMOTO DEL CRECIMIENTO Y DESARROLLO EN  
CONEJOS PARA LA PRODUCCIÓN CUNÍCULA MEDIANTE PROCESAMIENTO DE  
IMÁGENES**

**Autor**

**MOISES ABUABARA ANGARITA**

**Directora**

**LUZ MARINA SANTOS JAIMES**

**Doctora en Ciencias de la Computación y Matemática Computacional**

**INGENIERÍA DE SISTEMAS**

**INGENIERÍA**

**FACULTAD DE INGENIERIAS Y ARQUITECTURAS**



**UNIVERSIDAD DE PAMPLONA**

**PAMPLONA, 20 noviembre del 2019**

## **Agradecimientos**

Quiero agradecer primeramente a Dios por guiarme en el camino y fortalecerme espiritualmente para emprender el camino que me guía a la realización de mis sueños. Agradecer a mis padres quienes me dieron la vida y me enseñaron que la perseverancia y la constancia es el pilar del éxito, en especial a mi madre Betzi quien siempre ha creído en mí, que con sus consejos y su apoyo incondicional me empuja día a día para que logre mis metas; así mismo, a mi amigo Jeison que me brindo su ayuda cuando más la necesité.

Muestro mis más sinceros agradecimientos a mis tutores de proyecto, quienes con su conocimiento y guía fueron una pieza clave para desarrollar mi proyecto y todas aquellas personas que estuvieron presentes en la realización de este sueño que es tan importante para mí.

Y por último a mis hijos Richard y Victoria, quienes son el motor de mi vida, verlos crecer a mi lado es el empuje que me hace querer hacer más, alcanzar las estrellas para colocarlas en sus manos, soy la persona más afortunada por contar con seres que me aprecian y me quieren, a todos ellos muchas gracias, esto no hubiese sido posible sin su apoyo y compañía.

## **RESUMEN**

La cunicultura es una de las producciones que más necesita la interacción humana, debido a esto es una de las menos practicadas en Colombia; ya que, se necesita el doble de esfuerzo, mano de obra, cuidado y dedicación que cualquier otra actividad como la ganadería o porcicultura, entre otras. Para facilitar esta labor se creó un sistema de información el cual integra automatización en la obtención de los datos y la ejecución de acciones. Por tanto, se crean algoritmos orientados al procesamiento de imágenes que brinden al criador una disminución de esfuerzo humano con respecto a las tareas básicas de esta actividad. El proyecto genera un informe detallado de la vida de los conejos por medio de un sistema de información y monitoreo de fácil acceso, con una interfaz agradable y de fácil entendimiento, la cual automatiza también las labores de alimentación de los conejos y control de crecimiento y desarrollo, optimizando la producción Cunícola.

### **Palabras clave**

*Cunícola, procesamiento de imágenes, monitoreo, automatización.*

## **ABSTRACT**

Cuniculture is one of the productions that human interaction most needs, because of this it is one of the least practiced in Colombia; since, it takes twice as much effort, labor, care and dedication as any other activity such as livestock or pig farming, among others. To facilitate this work, an information system was created which integrates automation in obtaining data and executing actions. Therefore, it is intended to create algorithms oriented to image processing that provide the breeder with a decrease in human effort with respect to the basic tasks of this activity. The project generates a detailed report of the life of the rabbits through an easy-to-access information and monitoring system, with a pleasant and easily understood interface, which also automates the works of rabbits feeding and growth control and development, optimizing the production Cunícola.

### **KEYWORDS:**

Cunícola, image processing, monitoring, automation.

## Índice de Contenido

<b>INTRODUCCIÓN .....</b>	<b>9</b>
<b>1. PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>10</b>
1.1 JUSTIFICACIÓN .....	11
1.3 DELIMITACIÓN .....	11
1.3.1 Objetivo general .....	11
1.3.2 Objetivos específicos .....	11
1.4 METODOLOGÍA .....	12
1.5 ESTRUCTURA DEL DOCUMENTO.....	12
<b>2. MARCO TEÓRICO Y ESTADO DEL ARTE.....</b>	<b>13</b>
2.1 MARCO TEÓRICO .....	13
2.1.1 AUTOMATIZACIÓN Y CONTROL.....	13
2.1.2 Herramientas de automatización .....	17
2.1.3 Actividades .....	22
2.2 ESTADO DEL ARTE.....	23
2.2.1 Internacional .....	23
2.2.2 Nacional .....	24
2.2.3 Locales .....	25
<b>3 SISTEMA DE INFORMACIÓN PARA DAR RESPUESTA A LAS NECESIDADES DE LA PRODUCCIÓN CUNÍCOLA.....</b>	<b>25</b>
3.1 MODELADO DEL CASO DE USO.....	26
3.2 DIAGRAMA MYSQL DEL PROYECTO.....	33
3.3 REQUERIMIENTOS DEL SISTEMA .....	33
<b>4 AUTOMATIZACIÓN DEL PROCESO DE MONITOREO DE CRECIMIENTO Y DESARROLLO EN CONEJOS .....</b>	<b>37</b>
4.1 MODELAMIENTO DE LOS PROCESOS DE LA PRODUCCIÓN CUNÍCOLA.....	37
4.2 TOMA DE IMÁGENES .....	41
4.3 AUTOMATIZACIÓN DEL PROCESO DE ALIMENTACIÓN DEL PROTOTIPO .....	46
4.3.1 Dispensador de alimentos.....	46
4.3.2 Control de alimentación .....	47
4.3.3 Diseño para recolector de desechos biológicos .....	48
4.3.4 Selección de la instrumentación a utilizar.....	48
4.4 DISEÑO E IMPLEMENTACIÓN DE LA AUTOMATIZACIÓN .....	56
4.4.1 Sistema de alimentación automatizada .....	56
4.4.2 Sistema de recolección de desechos .....	56
4.4.3 Esquema general.....	57
<b>5 VALIDACIÓN Y ANÁLISIS DE LA AUTOMATIZACIÓN.....</b>	<b>58</b>
5.1 DESARROLLO DEL ALGORITMO PARA LA OBTENCIÓN DE LOS DATOS A PARTIR DE LAS IMÁGENES.....	58
5.2 ANÁLISIS DE DATOS (DATOS REALES Y DATOS EXPERIMENTALES) .....	61

5.3 DESARROLLO DEL ALGORITMO DE AUTOMATIZACIÓN PARTIENDO DE LOS DATOS OBTENIDO EN EL TRATAMIENTO DE IMÁGENES.....	62
5.4 COMPARACIÓN DEL PROTOTIPO VS CUNICULTURA TRADICIONAL.....	65
5.5 ANÁLISIS DE LOS DATOS TENIENDO EN CUENTA LA CONVERSIÓN ALIMENTO CONSUMIDO, PESO GANADO CONEJO, SEGÚN ESTUDIOS Y OPINIONES DE EXPERTOS .....	70
<b>CONCLUSIONES.....</b>	<b>72</b>
<b>BIBLIOGRAFÍA.....</b>	<b>74</b>

## Tablas de Ilustraciones

<i>Ilustración 1 Caso de uso Login</i> .....	28
<i>Ilustración 2 Caso de uso Registrar</i> .....	29
<i>Ilustración 3 Caso de uso Actualizar</i> .....	31
<i>Ilustración 4 Caso de uso Monitoreo</i> .....	32
<i>Ilustración 5 Diagrama entidad relación</i> .....	33
<i>Ilustración 6 Diagrama general</i> .....	38
<i>Ilustración 7 Diagrama de desechos</i> .....	38
<i>Ilustración 8 Diagrama de alimentación</i> .....	39
<i>Ilustración 9 Diagrama tratamiento de imagen</i> .....	40
<i>Ilustración 10 Diagrama de monitoreo</i> .....	41
<i>Ilustración 11 Imagen tratada con VIA</i> .....	42
<i>Ilustración 12 Json generado por VIA</i> .....	43
<i>Ilustración 13 Entrenamiento del modelo MaskRCNN</i> .....	43
<i>Ilustración 14 Clasificación de la imagen a tratar</i> .....	44
<i>Ilustración 15 Refinamiento de la imagen a tratar</i> .....	44
<i>Ilustración 16 Máscara de imagen tratada</i> .....	45
<i>Ilustración 17 Contorno de las máscara y área correspondiente</i> .....	46
<i>Ilustración 18 Dispensador de agua</i> .....	47
<i>Ilustración 19 Dispensador de alimento sólido</i> .....	47
<i>Ilustración 20 Diagrama de conexión para el dispensador de alimento</i> .....	56
<i>Ilustración 21 Sistema de recolección de desechos</i> .....	57
<i>Ilustración 22 Esquema general de conexión</i> .....	57
<i>Ilustración 23 Crecimiento de conejos automático</i> .....	65
<i>Ilustración 24 Crecimiento de conejos tradicional</i> .....	66
<i>Ilustración 25 Login de ingreso al sistema de información</i> .....	67
<i>Ilustración 26 Pantalla principal del sistema de información</i> .....	68
<i>Ilustración 27 Vista de registro crecimiento en el sistema de información</i> .....	68
<i>Ilustración 28 Sección de informes en el sistema de información</i> .....	69
<i>Ilustración 29 Monitoreo opción de informe</i> .....	70
<i>Ilustración 30 Logo empresa</i> .....	86

## Lista de Tablas

<i>Tabla 1 001 – Login del Sistema.....</i>	<i>27</i>
<i>Tabla 2 002 – Registrar .....</i>	<i>29</i>
<i>Tabla 3 003 – Actualizar.....</i>	<i>30</i>
<i>Tabla 4 004 – Monitoreo.....</i>	<i>32</i>
<i>Tabla 5 plantilla Stakeholders.....</i>	<i>34</i>
<i>Tabla 6 Requerimientos no funcionales.....</i>	<i>35</i>
<i>Tabla 7 Requerimientos funcionales.....</i>	<i>36</i>
<i>Tabla 8 Microcontroladores.....</i>	<i>50</i>
<i>Tabla 9 Controlador Principal.....</i>	<i>52</i>
<i>Tabla 10 Sensores y módulos .....</i>	<i>54</i>
<i>Tabla 11 Motores.....</i>	<i>55</i>
<i>Tabla 12 Cámaras.....</i>	<i>55</i>
<i>Tabla 13 Datos reales y datos experimentales .....</i>	<i>61</i>



## **Introducción**

La búsqueda del ser humano por la calidad en los procesos cotidianos, lo han llevado a incursionar en nuevas formas de realizarlos, ya sean manuales o automáticos, el presente documento es evidencia de esto, permitiendo incorporar un prototipo para la automatización y monitorización remota del crecimiento de conejos, donde el control de variables como el área, peso, ración alimenticia, entre otras, es fundamental para una buena producción. El desarrollo del proyecto se divide en cuatro grandes etapas, la primera consiste en el análisis, desarrollo, pruebas y despliegue del sistema de información, donde estratégicamente se tuvo en cuenta al momento de la creación de la base de datos la tabla crecimiento, para posteriormente acoplar con el monitoreo del crecimiento del conjunto de individuos que están dispuestos en una jaula. La segunda etapa conforma la selección de la tecnología en aspectos de software y hardware, para llevar a cabo el proceso de automatización lo cual está directamente relacionado con un algoritmo de inteligencia artificial encargado de determinar el peso y ración de alimento por jaula. En la tercera etapa se realiza la instalación de la instrumentación a la jaula, acople del algoritmo de inteligencia artificial al microcontrolador, y este a su vez a los actuadores y sensores, para finalmente integrarlo al sistema de información. Y en la cuarta etapa se expone un plan de negocios orientado a la creación de una empresa de automatización de granjas cunícolas.

## **1. Planteamiento del problema**

En la vida del campo, el llevar el control de los animales a criar (Conejos) es una labor que requiere gran cantidad de tiempo y esfuerzo humano, al momento de saber ¿cuántos conejos hay?, ¿el peso que han de tener?, y ¿cuál es el momento exacto para que estos alcancen el tamaño ideal para mantener una producción beneficiosa?, son los principales interrogantes al realizar actividades de monitoreo en la cría y venta de conejos para consumo. Los cunicultores normalmente realizan sus actividades a decisión y criterio de tanteo aleatorio, dicho de forma coloquial “a ojo”, no llevan un orden o control de datos para la producción. Utilizando las TICs (Tecnologías de la información y comunicación) y llevando un riguroso control de datos, los cunicultores que viven de esta actividad económica podrían tener una mejor producción, lo cual, al tener estos datos de manera ordenada pueden ser utilizados para aumentar y mejorar la producción manteniendo un margen costo beneficio, disminuyendo el esfuerzo humano y el tiempo requerido en las actividades. Según (Leonart, 1987) considera que las adecuaciones de nuevas tecnologías permiten mejorar la producción cunícola.

## **1.1 Justificación**

En la actualidad hay sistemas enfocados a grandes especies como el ganado, y podría decirse que son utilizados como base de datos (sólo para obtener información), pero no para la automatización de actividades. La propuesta de este proyecto fue más allá, de la obtención de datos y variables a analizar, puesto que se realizaron de manera automática el cálculo de tamaño y peso de los especímenes (conejos), mediante el análisis de imágenes y algoritmos. Los resultados obtenidos disminuyen el esfuerzo humano, maximizan la producción, disminuyen los gastos en contratación y evitan el desperdicio de alimentos.

## **1.3 Delimitación**

### **1.3.1 Objetivo general**

- Implementar un sistema de información y automatización, para el monitoreo remoto de crecimiento y desarrollo en conejos (teniendo en cuenta las variables: Peso, edad, ración alimenticia) para la producción cunícola en una granja en el municipio de Mutiscua mediante tratamiento de imágenes.

### **1.3.2 Objetivos específicos**

- Diseñar un sistema de información que dé respuesta a las necesidades de la producción cunícola (peso, edad, ración alimenticia, e información básica de los conejos)
- Automatizar el proceso de monitoreo de crecimiento y desarrollo en conejos
- Obtención y análisis de datos mediante un algoritmo de automatización, para la producción cunícola mediante el tratamiento de imágenes
- Implementación y validación de funcionamiento del sistema de monitoreo remoto.

## **1.4 Metodología**

Dentro de los proyectos de investigación se pueden encontrar varias clasificaciones de estudios dependiendo la necesidad, según (Hernandez roberto, Fernandez carlos, & Baptista pilar, 2006) hay exploratorios, descriptivos, correlaciones y explicativos, también se pueden encontrar combinaciones de estos. El proyecto inicia con un estudio exploratorio teniendo en cuenta que el tema es poco conocido, ajustándose a lo mencionado por (Hernandez roberto, Fernandez carlos, & Baptista pilar, 2006) pág. 101 este tipo de investigación es cuando el tema es poco estudiado y se tienen muchas dudas. Como segundo, el proyecto es descriptivo experimental incluyendo variables cuantitativas, donde experimentalmente se obtienen resultados que fueron mejorados para la presentación de un prototipo totalmente ajustado a las necesidades de la explotación cunícola, según (Hernandez roberto, Fernandez carlos, & Baptista pilar, 2006) Pág. 102 “Con frecuencia, la meta del investigador consiste en describir fenómenos, situaciones, contextos y eventos.”

## **1.5 Estructura del documento**

El desarrollo del proyecto está conformado por 7 capítulos, Capítulo 1 el cual contiene como base fundamental la introducción al proyecto incluyendo justificación, delimitación, entre otros. El capítulo 2 está conformando básicamente por el marco conceptual y el estado de arte, encargado de contextualizar e implementar las bases de proyecto. En el capítulo 3 se encuentra el desarrollo del sistema de información, plasmando los casos de uso, diagrama MySQL, entre otros. Capítulo 4 se encuentra el proceso de monitoreo e implementación del proceso de automatización sobre la jaula prototipo. Capítulo 5 se realiza la obtención y el análisis, incluyendo el modelado matemático para el cálculo del peso y la creación del algoritmo inteligente. El capítulo 6 contiene la validación del sistema, los cuales se encuentran acoplados al sistema de información. Capítulo 7 conclusiones del proyecto.

## **2. Marco Teórico Y Estado Del Arte**

La importancia de este capítulo, nace como la teoría fundamental que sustenta la base del proyecto, también permite al lector comprender la problemática planteada y sus principales aspectos, igualmente se presenta una revisión histórica de proyectos relacionados que de alguna manera contribuyen sustancialmente al desarrollo del prototipo, mostrando los avances que se tienen hasta el momento.

### **2.1 Marco teórico**

#### **2.1.1 Automatización y control**

- **Sistema de información**

En (Laudon & Laudon, 2012) lo definen como “un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores del conocimiento a analizar problemas, visualizar temas complejos y crear nuevos productos.

Los sistemas de información contienen información sobre personas, lugares y cosas importantes dentro de la organización, o en el entorno que la rodea. Por información nos referimos a los datos que se han modelado en una forma significativa y útil para los seres humanos. Por el contrario, los datos son flujos de elementos en bruto que representan los eventos que ocurren en las organizaciones o en el entorno físico antes de ordenarlos e interpretarlos en una forma que las personas puedan comprender y usar.

Hay tres actividades en un sistema de información que producen los datos necesarios para que las organizaciones tomen decisiones, controlen las operaciones, analicen problemas y creen nuevos productos o servicios. Estas actividades son: entrada, procesamiento y salida

La entrada captura o recolecta los datos en crudo desde el interior de la organización o a través de su entorno externo. El procesamiento convierte esta entrada en bruto en un formato significativo. La salida transfiere la información procesada a las personas que harán uso de ella, o a las actividades para las que se utilizará. Los sistemas de información también requieren retroalimentación: la salida que se devuelve a los miembros apropiados de la organización para ayudarles a evaluar o corregir la etapa de entrada”.

- **Sistemas embebidos.**

Según (Cristina Arilla, Laura Arribas, 2009), un sistema embebido consiste en un sistema de computación cuyo hardware y software están específicamente diseñados y optimizados para resolver un problema concreto eficientemente. El término "embebido" (también se le conoce como “empotrado”) hace referencia al hecho que la electrónica o el sistema electrónico de control es una parte integral del sistema en que se encuentra. La característica principal que diferencia a los “embebidos” de los demás sistemas electrónicos es que, por estar insertados dentro del dispositivo que controlan, están sujetos en mayor medida a cumplir requisitos de tamaño, fiabilidad, consumo y coste, y su existencia puede no ser aparente. Algunos ejemplos de sistemas embebidos son los sistemas de información integrados en automóviles, trenes o aviones, y controladores de procesos en sistemas de producción industrial.

En (Linares, 2005) se afirma que, “un sistema embebido en principio estaría formado por un microprocesador y un software que se ejecute sobre este. Sin embargo, este software necesitará sin duda un lugar donde poder guardarse para luego ser ejecutado por el procesador. Esto podría tomar la forma de memoria RAM o ROM, Todo sistema embebido necesitará en alguna medida una cierta cantidad de memoria, la cual puede incluso encontrarse dentro del mismo chip del procesador. Además de esto normalmente un sistema embebido contará con una serie de salidas y entradas necesarias para comunicarse con el mundo exterior”.

## ▪ **Sistema automatizado.**

Según (MARTÍNEZ, 2017), “son sistemas y procesos en los cuales se requiere la mínima intervención del ser humano, además debe de ser un sistema flexible el cual se debe ajustar de distintas maneras a los posibles cambios en momentos puntuales.

La automatización se divide en tres tipos:

- La automatización fija (productos con gran índice de demanda y volumen). Este tipo de producción tiene un alto costo y se ha de tener en cuenta que su ciclo de vida es el mismo que el del producto que se fabrica. Ejemplo: Líneas mecanizadas de ensamblaje y líneas de transferencia de maquinado
- La automatización programable (productos con bajo índice de demanda y gran diversidad de productos). En este caso los equipos de producción se diseñan para poder adaptarse a las variantes de los distintos tipos de productos que se fabrican. Estas adaptaciones se realizan por medio de programación (software). Ejemplo: Robots industriales, control numérico, PLCs, relés programables, etc.
- La automatización flexible (productos con un índice medio de demanda). Estos sistemas combinan tanto elementos de la automatización fija como programada. Suele estar constituidas por diferentes estaciones de trabajo interconectadas entre sí y controladas por un ordenador.”

## ▪ **Sistemas de Control**

Según (Gaviño, 2010), establece que “Un sistema de control automático es una interconexión de elementos que forman una configuración denominada sistema, de tal manera que el arreglo resultante es capaz de controlarse por sí mismo. Un sistema o componente del sistema susceptible de ser controlado, al cual se le aplica una señal  $r(t)$  a manera de entrada para obtener una respuesta

o salida  $y(t)$ , puede representarse mediante bloques. Los sistemas de control se clasifican en sistemas de lazo abierto (o no automático) y sistemas de lazo cerrado (retroalimentados o automáticos). Se define acción de control como la cantidad dosificada de energía que afecta al sistema para producir la salida o la respuesta deseada.

- Sistema de control de lazo abierto. Es aquel sistema en el cual la acción de control es, en cierto modo, independiente de la salida. Este tipo de sistemas por lo general utiliza un regulador o actuador con la finalidad de obtener la respuesta deseada, la capacidad que tales sistemas tienen para ejecutar una acción con exactitud depende de su calibración. En general, los sistemas de lazo abierto están regulados por base de tiempo. Como ejemplo de dichos sistemas se citan los tostadores de pan, las lavadoras, los hornos de microondas y los semáforos convencionales.
- Sistema de control de lazo cerrado. Es aquel sistema en el cual la acción de control depende de la salida. Dicho sistema utiliza un sensor que detecta la respuesta real para compararla, entonces, con una referencia a manera de entrada. Por esta razón, los sistemas de lazo cerrado se denominan sistemas retroalimentados. El término retroalimentar significa comparar; en este caso, la salida real se compara con respecto al comportamiento deseado, de tal forma que si el sistema lo requiere se aplica una acción correctora sobre el proceso por controlar.”

#### ▪ **Procesamiento de Imágenes**

En (Aguilera, 2011) establece que “El procesamiento de imágenes tiene que ver con la adquisición, transmisión, procesamiento y representación de las imágenes. Las técnicas de proceso de imágenes se utilizan para mejorar la apariencia visual de las imágenes para un observador y para preparar convenientemente el contenido fotográfico de cara a la percepción por parte de máquinas. El proceso digital de imágenes se puede dividir en las siguientes áreas:

- Adquisición o captura que se ocupa de los diferentes caminos para la obtención de imágenes; por ejemplo, utilizando cámaras digitales o digitalizando imágenes analógicas (fotografías).



- Realce y mejora son las técnicas que se usan para mejorar la apariencia visual de las imágenes o para recuperar o restaurar las imágenes degradadas.
- Segmentación que se ocupa de la división de las imágenes en regiones o áreas significativas.
- Extracción de características que se ocupa de la detección y localización de entidades geométricas simples y complejas. Desde entidades simples como líneas y puntos hasta geometrías complejas como curvas y cuádricas”.

### 2.1.2 Herramientas de automatización

#### ▪ Servidor Web

En (Ramos Martín & Ramos Martín, 2011) definen servidor web “como un programa diseñado para aceptar peticiones HTTP del navegador y servir las páginas web que tiene alojadas. Para navegar a un sitio web se necesita saber la dirección web exacta del sitio. Cada sitio web tiene una dirección única conocida como URL (*Uniform Resource Locator*). En una URL se puede distinguir tres partes: el protocolo, el nombre de dominio del ordenador servidor, y la ruta en el servidor, El protocolo que se utiliza para servir páginas web es el HTTP, el nombre de dominio indica el ordenador de internet que nos va a dar información solicitada y la ruta indica la ubicación del archivo solicitado dentro del servidor”.

#### ▪ Arduino

Según (Herrador, 13 de noviembre de 2009), “Arduino es una plataforma de prototipos electrónica de código abierto (*open-source*) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos. Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el lenguaje de programación de Arduino y el ambiente de desarrollo de Arduino. Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo, con

*Flash, Processing, MaxMSP, etc.*). Las placas se pueden ensamblar a mano o encargarlas pre ensambladas, el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia *open-source*, que permite adaptarlas a las necesidades”.

- **Raspberry Pi**

En (TECNOLOGÍA, ABC, 2013), “Raspberry Pi, es un ordenador de tamaño de tarjeta de crédito que se conecta a su televisor y un teclado, una placa que soporta varios componentes necesarios en un ordenador común. Es un pequeño ordenador que puede ser utilizado por muchas de las cosas que su PC de escritorio hace, como hojas de cálculo, procesadores de texto y juegos. También reproduce vídeo de alta definición. Posee un *System on Chip* que contiene un procesador ARM que corre a 700 MHz, un procesador gráfico *VideoCore IV* y hasta 512 MG de memoria RAM. Es posible instalar sistemas operativos libres a través de una tarjeta SD”.

- **Cámara térmica**

Según (Electrónica, 2018), “Una cámara termográfica o cámara térmica es una cámara que muestra en pantalla una imagen de la radiación calorífica que emite un cuerpo. Todos los cuerpos por encima del cero absoluto (-273°C) emiten radiación infrarroja (calor). En general, cuanto mayor es la radiación emitida, mayor es la temperatura del cuerpo. Esta radiación es invisible al ojo humano y su rango en el espectro electromagnético se sitúa entre la luz visible y la radiación de microondas. En concreto, la longitud de onda de los infrarrojos se sitúa entre las 0,7 y las 1000 micras. Dentro de este amplio margen, las cámaras térmicas trabajan en un rango conocido como infrarrojo térmico, que es donde se encuentran las temperaturas más habituales en la superficie terrestre, entre las 8 y las 14 micras, que equivale aproximadamente entre los -20 y 350 ° C.”.

- **MaskRCNN**

Es un modelo de una red neuronal que implementa Keras y TensorFlow para el reconocimiento de objetos proporcionados por Karol Majek y Matterport, el modelo genera cajas y máscaras de

segmentación para cada instancia del objeto en la imagen. Este modelo puede ser descargado desde sus repositorios en Github (matterport, 2019). El código proporcionado da explicación al funcionamiento y como se debe entrenar para un conjunto de imágenes u objetos para los que se quiera utilizar.

- ***VGG Image Annotator (VIA)***

Es un software diseñado por la Universidad de Oxford para realizar anotaciones manualmente en una imagen el cual se ejecuta en un navegador web, este genera un documento en formato JSON con las coordenadas del área de la anotación y el nombre de la imagen.

- **Regresión Lineal**

Los autores (Molina & Rodrigo, 2010) argumentan que “los modelos predictivos o de regresión son la representación de la relación entre dos (o más) variables a través de un modelo formal, supone contar con una expresión lógico-matemática que, aparte de resumir cómo es esa relación, va a permitir realizar predicciones de los valores que tomará una de las dos variables (la que se asuma como variable de respuesta, dependiente, criterio Y) a partir de los valores de la otra (la que se asuma como variable explicativa, independiente, predictora o X).

El modelo de regresión lineal es el más utilizado a la hora de predecir los valores de una variable cuantitativa a partir de los valores de otra variable explicativa también cuantitativa (modelo de regresión lineal simple). Una generalización de este modelo, el de regresión lineal múltiple, permite considerar más de una variable explicativa cuantitativa. Por otra parte, es también posible incluir variables explicativas categóricas en un modelo de regresión lineal si se sigue una determinada estrategia en la codificación de los datos conocida como codificación ficticia.

En concreto, según el modelo de regresión lineal simple, las puntuaciones de los sujetos en dos variables, una de ellas considerada como variable predictora (X) y la otra como variable de respuesta (Y), vienen representadas (modeladas) por la ecuación de una línea recta:

$$Y = \beta_0 + \beta_1 * X_1$$

Cuando hay más de una variable explicativa (modelo de regresión lineal múltiple), se utiliza un subíndice para cada una de ellas, por ejemplo, para el caso de dos variables explicativas:

$$Y = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2''$$

## ▪ Máquinas de estado finito

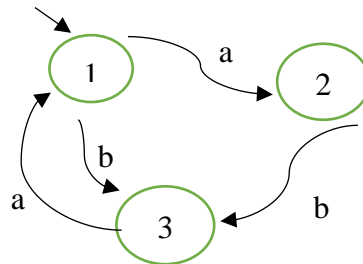
Las máquinas de estado finito definidas como “una herramienta muy útil para especificar aspectos relacionados con tiempo real, dominios reactivos o autónomos, computación reactiva, protocolos, circuitos, arquitecturas de software, etc.

El modelo de FSM (*Finite State Machine*) es un modelo que posee sintaxis y semántica formales y que sirve para representar aspectos dinámicos que no se expresan en otros diagramas” (Máquinas de Estado Finito, 2019).

### Sintaxis

Las máquinas de estado finito se definen como una tupla,  $\langle S, \Sigma, A, \exists S * \Sigma^* S, sk \rangle$  donde:

- $S = \{s_1, s_2, \dots, s_m\}$
- $\Sigma$  : es un alfabeto finito de etiquetas.
- $A$ : es un conjunto finito de aristas etiquetadas que unen nodos.
- $sk \in S$ : es el estado inicial.



- $S = \{1,2,3\}$
- $\Sigma = \{a, b\}$
- $A = \{(1, a, 2), (2, b, 3), (3, a, 1), (1, b, 3)\}$
- $sk = 1$

## ▪ Python

Python es un “lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes, de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables” (python.org, 2009).

### 2.1.3 Actividades

#### ▪ Cunicultura

En (Solano, 2008), define la cunicultura como: “ciencia que trata de la cría, manejo y explotación racional de los conejos, con el propósito de obtener el máximo rendimiento con el mínimo de gastos.

La importancia económica del conejo radica en los siguientes tres aspectos:

- Como productor de pelo, el cual se utiliza como materia prima en la elaboración del fieltro.
- Como productor de pieles, que son usadas para la confección de abrigos, calzados, guantes y adornos.
- Como productor de carne para alimentación humana.”

## 2.2 Estado del Arte

### 2.2.1 Internacional

- **Desarrollo de una base de datos con imágenes termográficas para uso en algoritmos de visión e inteligencia artificial** (Pérez Sáenz, Duarte de la Cruz, Márquez Olivera, & Sánchez García, 2017).

El principal objetivo de este proyecto es presentar como a partir de imágenes extraídas de una cámara termográfica pueden ser interpretadas a través de un algoritmo de tratamiento de imágenes, para luego modelar los datos y proceder a su incorporación dentro de una base de datos para su posterior uso. Este proyecto sirve de ejemplo para la recolección de información a través de las cámaras termográficas otorgando unas pautas básicas que podrían ser utilizados en proyectos similares.

- **Detección de ovinos preñados usando algoritmos de inteligencia y visión artificial** (Sáenz Pérez, et al., 2018).

Ya que la preñez es uno de los estados más importantes en el ciclo de vida de la especie ovina muestra como es monitoreada a través de imágenes termográficas para la detección temprana según los patrones térmicos si la oveja se encuentra en gestación, este proyecto muestra una serie de pautas e interpretación de las imágenes obtenidas para la toma de decisiones en este campo pecuario, muestran el uso del algoritmo supervisado de segmentación *K-means* para los intervalos de temperatura presentes en la imagen, y la implementación del algoritmo *Haar-Cascade* propuesto por Viola y Jones, obteniendo una asertividad del 80%, estos algoritmos pueden servir de ejemplo para la posterior interpretación de imágenes en el presente proyecto.

- **Monitoreo de la producción animal MPA** (Loredo, 2014).

Plantea un sistema de información dedicado a la ganadería de tres especies bovinos, ovinos y porcinos utilizando como modelos el Cascada, Modelo en V, Modelo iterativo, Modelo de desarrollo incremental, Modelo en espiral, Modelo de prototipo, utilizando finalmente el modelo de ciclo de vida, haciendo un sistema de fácil acceso y entendimiento para la toma de decisiones la cual “le permite a la ganadería crecer”.

### 2.2.2 Nacional

- **Implementación de un sistema de identificación ganadero de bajo costo para el control de eventos productivos en el sector pecuario, mediante dispositivos móviles** (Dimas López & Barroso Benítez, 2015).

Este proyecto implemento la utilización de un sistema embebido, mediante el cual se daba gestión de información de usuarios y animales registrados en el sistema, donde se envía al usuario el control realizado de los animales y la trazabilidad de sus productos, la importancia de este proyecto yace en la gestión de algunas variables y procedimientos que son importantes para llevar un control de la información de los animales registrados, para ser mostrados a los usuarios y “mejorar su producción”.

- **Sistema de visión artificial para la estimación del peso de ganado bovino** (Hernández, 2004).

Desarrolla un sistema para calcular el peso de bovinos mediante la toma de imágenes, aplicando redes neuronales para la estimación del peso, e implementado binarización de la imagen, y disminución de ruido de forma manual, la obtención de la información en la imagen se hace a partir de las partes del cuerpo del animal. Aunque hay cierta similitud en el proyecto de Hernández y el presente, también convergen en la implementación y extracción de información, en el proyecto de Hernández se implementan redes neuronales para la estimación del peso, el tratamiento de la imagen es manual. En el presente proyecto las redes neuronales se implementan para la detección del animal, las estimaciones del peso se dan por medio un



algoritmo de regresión lineal y el tratamiento de imágenes se realiza de manera totalmente automática.

### 2.2.3 Locales

- **Aplicación de sistemas embebidos para la automatización de cultivos urbanos (BECERRA, 2018)**

Este proyecto desarrolla un sistema embebido para cultivos urbanos realizando toma de datos remotamente. El cual se relaciona con el proyecto de monitoreo de los conejos por sus sistemas embebidos y la implementación de IoT (Internet de las cosas), el proyecto presentado por Becerra difiere del presente proyecto en la implementación del monitoreo ya que Becerra planteo el acceso a los datos mediante internet, al contrario del presente proyecto que se encarga del análisis de los mismos para una posterior verificación.

- **Automatización integrada del cultivo hidropónico y el lombricultivo componentes de la granja cunícola autosustentable en el municipio de Mutiscua (CAÑAS, 2019).**

En este documento, se realizó la implementación de un cultivo hidropónico bajo un invernadero, el cual permite controlar factores externos como la humedad, temperatura, entre otras; las cuales son vitales para la buena producción. Igualmente se realizó la automatización de un lombricultivo, este componente es el encargado de procesar los desperdicios producidos por los conejos, por lo cual el proyecto realizado acopla al prototipo que se plantea para el monitoreo de los conejos, teniendo en cuenta que este prototipo realiza la recolección de los desperdicios depositándolos directamente en el lombricultivo, también automatiza la alimentación correspondiente a líquido y al grano (concentrado).

## 3 Sistema de información para dar respuesta a las necesidades de la producción Cunícola

En el presente capítulo se lleva a cabo la construcción del sistema, donde se presentan los casos de uso para el sistema de información, también se hace constancia de requerimientos generales que debe cumplir el sistema, teniendo en cuenta los *Stakeholders* que intervienen en la administración del sistema de información.

### 3.1 Modelado del caso de uso

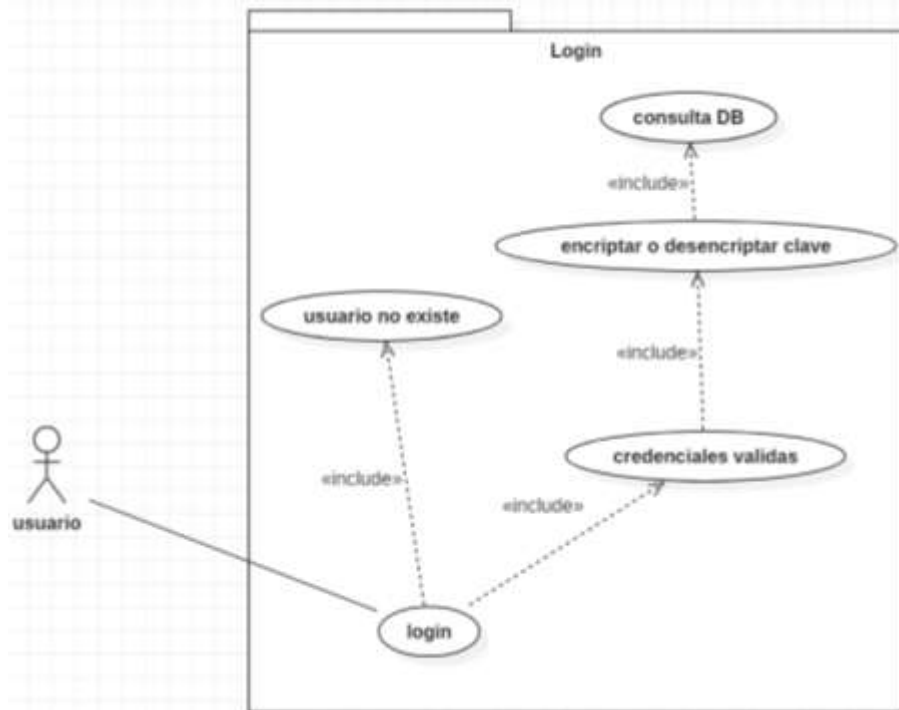
#### ▪ Caso de uso ingresar al sistema (Login)

En la Tabla 1, se presenta la información referente al caso de uso correspondiente al login del sistema de información, como complemento de este se muestra la Ilustración 1, donde se aprecian las relaciones del actor hacia el sistema.

<b>Caso de uso</b>	001 – Login del Sistema
<b>Actores</b>	Administrador
<b>Tipo propósito</b>	Flujo básico
<b>Resumen</b>	El administrador puede ingresar al sistema de información, por medio de Login, ingresando correo electrónico y clave, este a su vez valida que el usuario este registrado, para poder realizar la lectura de la clave la cual esta encriptada en la base de datos (BD), si el usuario no está registrado o alguno de los dos elementos falla (correo y/o clave) emite la notificación del usuario no existe o fallo de credenciales.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>● El usuario debe de abrir el navegador e ingresar a la página web: <a href="http://lizabu.com/">http://lizabu.com/</a> .</li> <li>● El usuario ya debió haber realizado el proceso de registro, donde asigno clave y correo</li> </ul>
<b>Flujo principal</b>	<ul style="list-style-type: none"> <li>● El usuario ingresa correo electrónico y la contraseña.</li> <li>● Presiona el botón de LOGIN en el formulario y este envía los datos a la base de datos, quien le permite el ingreso o muestra</li> </ul>

	una alerta.
<b>Flujo Alternativo</b>	<p><b>FA-1.</b> Al enviar el formulario, el sistema valida si el correo de usuario existe, sí no existe, informa mediante cuadro de dialogo el texto: “El usuario o la contraseña no coinciden”.</p> <p><b>FA-2.</b> Sí al enviar el formulario, el nombre de usuario existe, pero la contraseña no coincide con la guardada y encriptada en la Base de Datos. El sistema informa mediante cuadro de dialogo el texto: “El usuario o la contraseña no coinciden”.</p> <p><b>FA-3.</b> Sí al enviar el formulario, el nombre de usuario existe, y contraseña coincide con la guardada y encriptada en la Base de Datos. El sistema redirige al usuario al home de la aplicación.</p>
<b>Excepciones</b>	<p><b>E-1.</b> Si en el formulario no se ingresan todos los datos, mostrar mensaje de alerta indicando que todos los campos son requeridos. Igualmente, la cantidad de dígitos.</p>

Tabla 1 001 – Login del Sistema  
Fuente propia



*Ilustración 1 Caso de uso Login*  
*Fuente propia*

▪ **Caso de uso registrar**

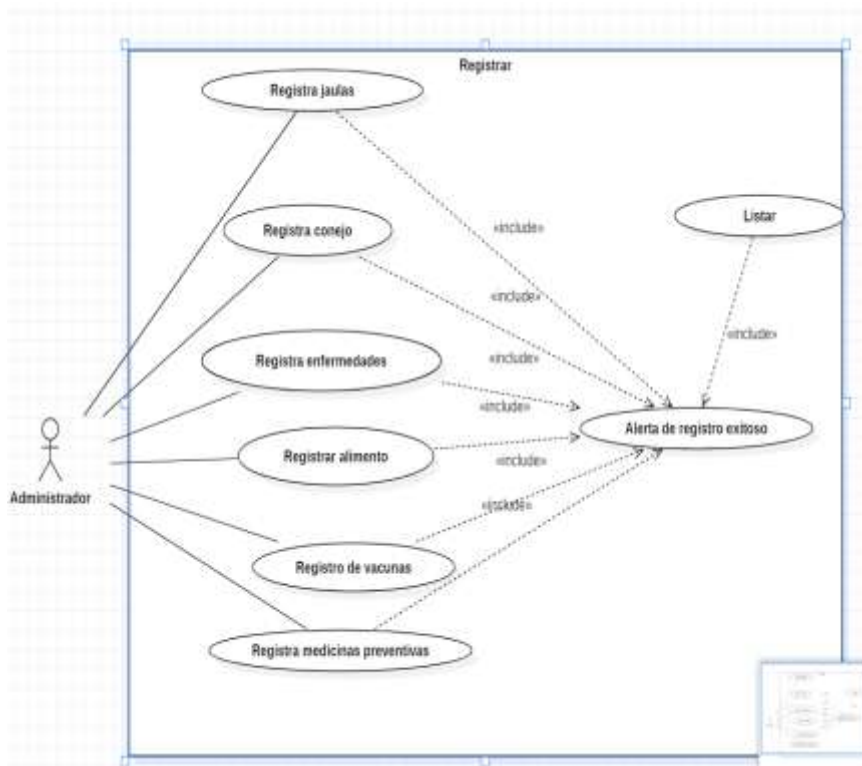
En la tabla 2, se presenta la información referente al caso de uso correspondiente al registro de datos en el sistema de información, como complemento de este se muestra la Ilustración 2, donde se aprecian las relaciones del actor hacia el sistema.

<b>Caso de uso</b>	002 – Registrar (RF 001 - Login del Sistema )
<b>Actores</b>	Administrador
<b>Tipo propósito</b>	Flujo básico
<b>Resumen</b>	Como el sistema de información está dirigido a un solo usuario, este será el encargado de realizar cualquier acción dentro del sistema, dentro de sus funcionalidades está el permitir realizar registros, como lo es registrar jaula, conejos, enfermedades, alimento, vacunas y medicina preventiva. Para llevar a cabo esta acción solo basta con ingresar al tipo de elemento que requiere registrar y seleccionar <b>Add New</b> , esta opción es la encargada de dirigir el usuario a la página de ingreso de datos para su posterior registro.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>● El usuario debe tener una sesión activa en la aplicación. CU 001</li> </ul>
<b>Flujo principal</b>	<ul style="list-style-type: none"> <li>● Selecciona el tipo de elemento (jaula, conejos, etc), que el administrador desea agregar, esta lista los artículos de ese elemento</li> <li>● Selecciona los Add New para registrar un nuevo artículo, quien lo redirige al formulario</li> <li>● Presiona un botón de guardar en el formulario y este envía los</li> </ul>

	datos a registrar en la BD y muestra el listado, donde se encuentra el artículo acabado de registrar
<b>Flujo Alternativo</b>	No aplica
<b>Excepciones</b>	<i>E-1.</i> Sí en el formulario no se ingresan todos los datos, mostrar mensaje de alerta indicando que todos los campos son requeridos.

*Tabla 2 002 – Registrar*

*Fuente propia*



*Ilustración 2 caso de uso Registrar*

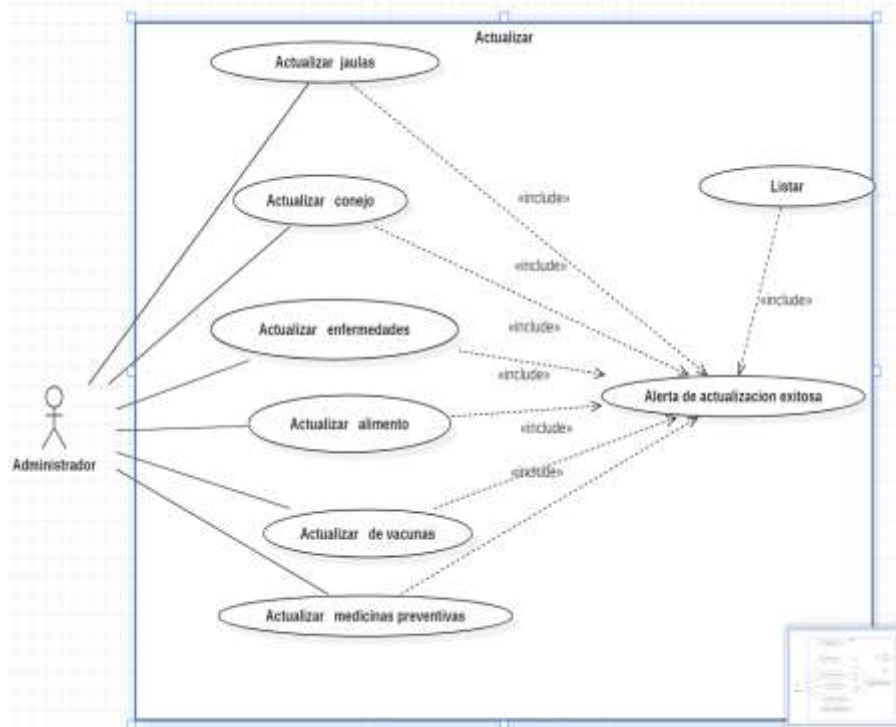
*Fuente propia*

▪ **Caso de uso Actualizar**

En la Tabla 3, se presenta la información referente al caso de uso correspondiente a la actualización de datos en el sistema de información, como complemento de este se muestra la Ilustración 3, donde se aprecian las relaciones del actor hacia el sistema.

<b>Caso de uso</b>	003 – Actualizar (RF 001 – Login del Sistema)
<b>Actores</b>	Usuario Inventarios
<b>Tipo propósito</b>	Flujo básico
<b>Resumen</b>	Para realizar el proceso de actualización, se debe tener en cuenta que el usuario debe estar logueado en el sistema, los procesos que están permitidos actualizar son: jaula, conejos, enfermedades, alimento, vacunas y medicina preventiva. Para llevar a cabo este proceso, el usuario debe ingresar la tabla que contiene el elemento (ejemplo jaula) y de la lista seleccionar el registro deseado y proceder a realizar la actualización. para los procesos que incluyen eliminar y ver información, se realiza de misma forma, solo basta con seleccionar la casilla que desea teniendo en cuenta las 3 acciones (actualizar, eliminar y ver)
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El usuario debe tener una sesión activa en la aplicación. CU 001</li> </ul>
<b>Flujo principal</b>	<ul style="list-style-type: none"> <li>• El Administrador selecciona el elemento general que sea actualizar, (ejemplo jaula), lo cual muestra la lista de todos los artículos</li> <li>• <i>Selecciona el artículo que desea actualizar, esta opción redirige al usuario al formulario del artículo seleccionado</i></li> <li>• Ingresa los datos a actualizar y envía el formulario, este envió registra los cambios en la BD</li> </ul>
<b>Flujo Alternativo</b>	<i>No aplica</i>
<b>Excepciones</b>	<i>E-1.</i> Sí en el formulario no se ingresan todos los datos, mostrar mensaje de alerta indicando que todos los campos son requeridos

*Tabla 3 003 – Actualizar  
Fuente propia*



*Ilustración 3 Caso de uso Actualizar  
Fuente propia*

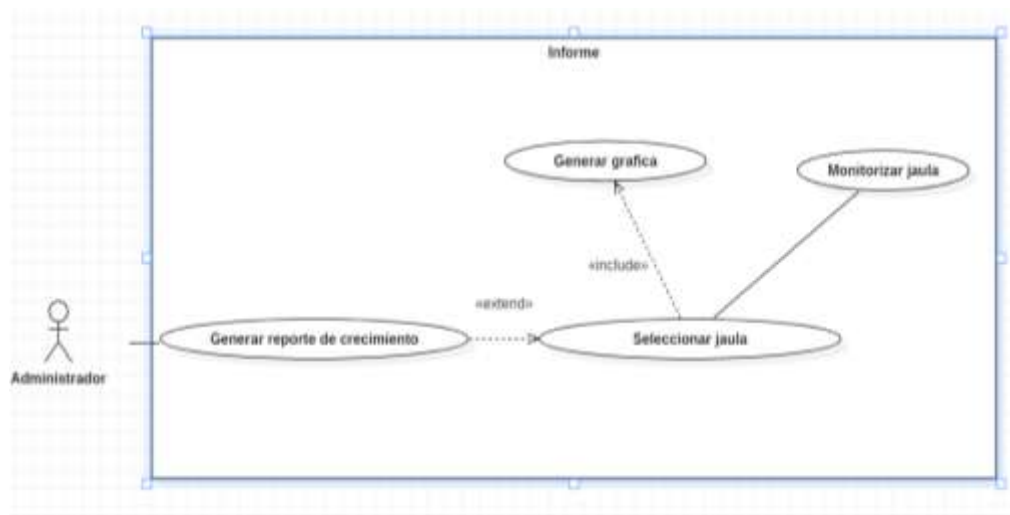
▪ **Caso de uso monitoreo**

En la Tabla 4, se presenta la información referente a el caso de uso correspondiente al monitoreo de los conejos en el sistema de información, como complemento de este se muestra la Ilustración 4, donde se aprecian las relaciones del actor hacia el sistema.

<b>Caso de uso</b>	<b>004 – Monitoreo (RF 001 – Login del Sistema)</b>
<b>Actores</b>	Usuario Inventarios
<b>Tipo propósito</b>	Flujo básico
<b>Resumen</b>	Como los casos anteriores el usuario debe estar logueado en el sistema, para poder realizar el monitorio el cual se encuentra dividido en dos opciones, por medio de la gráfica que deja ver el crecimiento a través del tiempo, la segunda opción permite monitorizar en vivo el estado de la jaula por medio de la cámara.

<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>El usuario debe tener una sesión activa en la aplicación. CU 001</li> </ul>
<b>Flujo principal</b>	<ul style="list-style-type: none"> <li>El Administrador ingresa a registro de crecimiento donde selecciona la jaula a monitorizar. Esta lo envía directo a las gráficas de crecimiento</li> <li><i>En la parte inferior se encuentra la opción de monitoreo en vivo, la cual muestra el video en tiempo real de la jaula.</i></li> </ul>
<b>Flujo Alternativo</b>	<p><b>FA-1.</b> El administrador se puede ver el reporte del crecimiento y el alimento consumido</p> <p><b>FA-2.</b> Al administrador realiza el monitoreo en tiempo real como soporte extra a las gráficas, para ver el estado real de los animales</p>
<b>Excepciones</b>	<p><b>E-1.</b> Sí en el formulario no se ingresan todos los datos, mostrar mensaje de alerta indicando que todos los campos son requeridos</p>

*Tabla 4 004 – Monitoreo  
Fuente propia*



*Ilustración 4 Caso de uso Monitoreo*



Fuente propia

### 3.2 Diagrama MySQL del proyecto

El diseño de la base de datos, se realiza por medio del gestor de base de datos MySQL, donde se presenta un modelo relacional, el modelo fue diseñado de forma general para la granja, aunque cabe mencionar que para el proyecto las tablas más importantes son **users**, **jails**, **increases**, **foods** y **animal**; las cuales tienen contacto directo con el proceso de automatización y el tratamiento de imagen, de las cuales depende el informe que genera el sistema de información, se puede apreciar el diagrama correspondiente en la Ilustración 5, presentada a continuación .

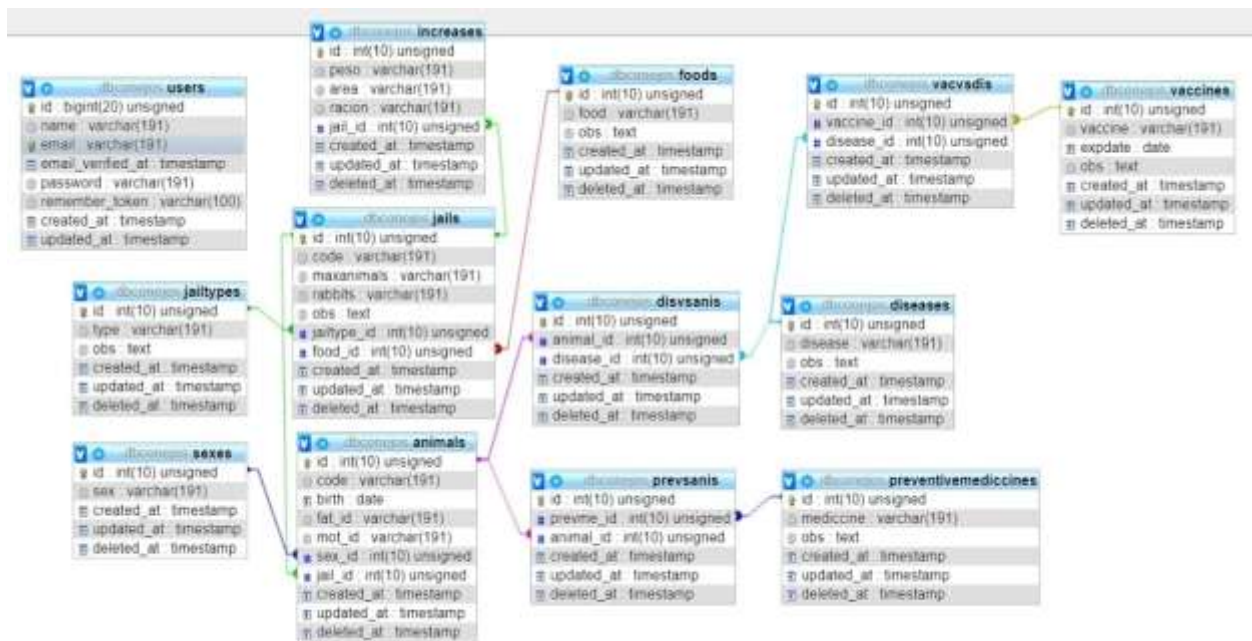


Ilustración 5 Diagrama entidad relación  
Fuente propia

### 3.3 Requerimientos del sistema

La Tabla 5, presentada a continuación hace referencia a la plantilla de *Stakeholders (Interesados)* en el sistema de información.

<i>Granja Cunicola</i>				
<i>Plantilla Stakeholders</i>				
<i>Código</i>	<i>Rol o cargo</i>	<i>Descripción del rol o interés</i>	<i>Nivel de influencia</i>	<i>Interés en el proyecto</i>
<i>01</i>	Administrador	Encargado de gestionar el sistema de información en general, teniendo en cuenta que es el único usuario del sistema	<b>Alto</b> Por ser el único rol, es el encargado del sistema en general.	El mayor interés es la reducción de costos y la disponibilidad de la información, en cuanto a control de crecimiento y monitoreo de los conejos

*Tabla 5 plantilla Stakeholders*  
*Fuente propia*

▪ **Requerimientos no funcionales**

Se presentan los requerimientos no funcionales del sistema de información en la Tabla 6 y los funcionales en la Tabla 7.

<i>Código</i>	<i>Nombre</i>	<i>Descripción</i>
<i>RNF 01</i>	Portabilidad	El sistema de información puede ser ejecutado o iniciado en cualquier dispositivo electrónico con acceso a internet, el cual por medio del navegador le permite hacer uso del sistema informático
<i>RNF 02</i>	Eficiencia	El sistema de información, manifiesta una latencia mínima para realizar las acciones permitidas, igualmente en las gráficas generadas como reporte del crecimiento y la monitorización en tiempo real.
<i>RNF 03</i>	Usabilidad	El sistema de información presenta una facilidad para realizar las diferentes acciones, permitiendo realizar una navegación fácil e

		intuitiva, con menú sencillo donde encuentra las actividades a realizar.
<i>RNF 04</i>	Seguridad	El sistema está orientado a uno solo usuario, por lo cual el registro solo está disponible una vez, además las credenciales como la clave, es encriptada en la base de datos por medio de Hash, permitiendo implementar una serie de medidas preventivas y reactivas, donde se busca proteger la confidencialidad, disponibilidad e integridad de los datos.

*Tabla 6 Requerimientos no funcionales*  
*Fuente propia*

### **Requerimientos Funcionales**

Se presentan los requerimientos funcionales del sistema de información en la tabla 7.

<i>Código</i>	<i>Nombre</i>	<i>Descripción</i>
<i>RF 01</i>	usuario	El sistema permite realizar el registro para el primer usuario, permitiendo ingresar los datos que posteriormente utiliza para ingresar el sistema de información, cabe denotar que el usuario no permite actualizar o eliminarse a sí mismo.
<i>RF 02</i>	Alimento	El sistema de información permite realizar las cuatro acciones básicas para una tabla, dependiendo de la tabla seleccionada, permite insertar, actualizar, listar o mostrar y eliminar cada registro que el usuario necesite.
<i>RF 03</i>	Enfermedades	
<i>RF 04</i>	Conejos	
<i>RF 05</i>	Jaulas	
<i>RF 06</i>	Vacunas	
<i>RF 07</i>	Medicina preventiva	
<i>RF 08</i>	Reporte	Los reportes de crecimientos, están contemplados en graficas donde se muestre a través del tiempo el crecimiento de los conejos, tomados como uno solo en la jaula, además permite ver el alimento consumido
<i>RF 09</i>	Monitoreo	El monitoreo en tiempo real de la jaula, se realiza gracias a una cámara instalada en la jaula, para poder realizar el dicho

		monitoreo, cuando el administrador lo solicite por medio de la página web
<i>RF 10</i>	Login	El sistema permite el ingreso del administrador que se registró previamente, haciendo uso del correo electrónico y clave, siempre y cuando estos datos coincidan con los almacenados en la tabla usuario el administrador podrá ingresar al sistema de información.

*Tabla 7 Requerimientos funcionales  
Fuente propia*

## **4 Automatización del proceso de monitoreo de crecimiento y desarrollo en conejos**

Los proyectos en general tienen un eje fundamental en su desarrollo, con el fin de dar inicio a la implementación del mismo, se describe la tarea de automatización en el proceso de monitoreo como parte esencial de este proyecto, para la cual se debe conocer las fases en que se divide, haciendo más fácil la implementación y la recolección de información. Como proyecto de automatización, basado fundamentalmente en controladores autómatas, el desarrollo de este proyecto se planteó mediante el uso de Python como lenguaje de programación por su versatilidad y compatibilidad entre plataformas, haciendo este las veces de mediador entre el sistema de información y el sistema de automatización (parte física), donde se implementan el tratamiento de imágenes, la extracción de información, actualización de información, programación de las actividades y ejecución de estas.

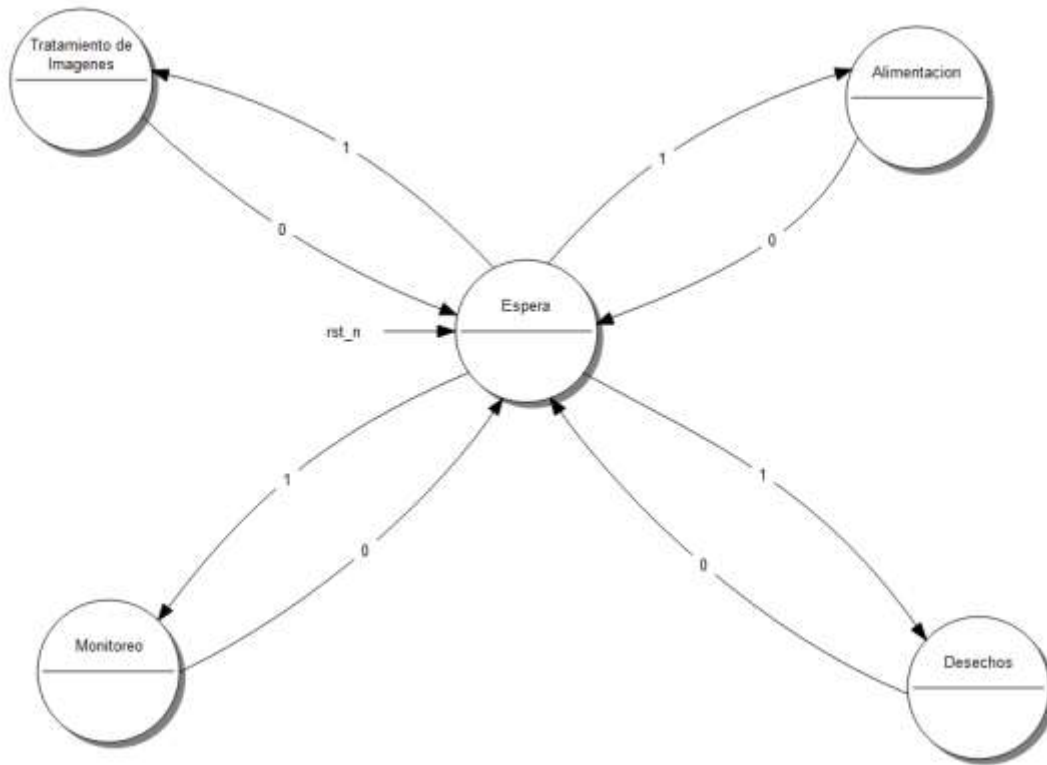
### **4.1 Modelamiento de los procesos de la producción Cunicola**

- **Diagramas de máquinas de estado finito para el sistema de automatización**

Se plantea para el sistema de automatización el modelamiento de sus procesos por medio de gráficos o diagramas aplicando máquinas de estado finito, por su afinidad con los microcontroladores haciendo de este un excelente método de representación.

- **Diagrama general**

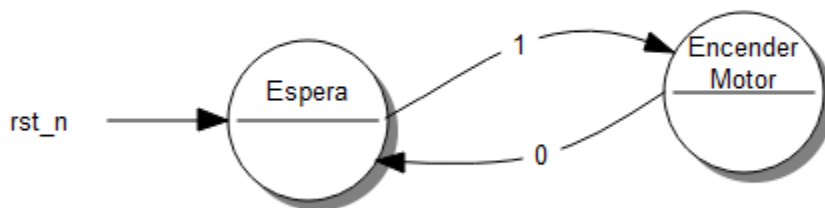
Se muestra un esquema general de los procesos principales del sistema de automatización en la Ilustración 6, para lo cual se establecen cuatro procesos: Tratamiento de imágenes, alimentación, monitoreo y desechos.



*Ilustración 6 Diagrama general  
Fuente propia*

▪ **Diagrama de Desechos**

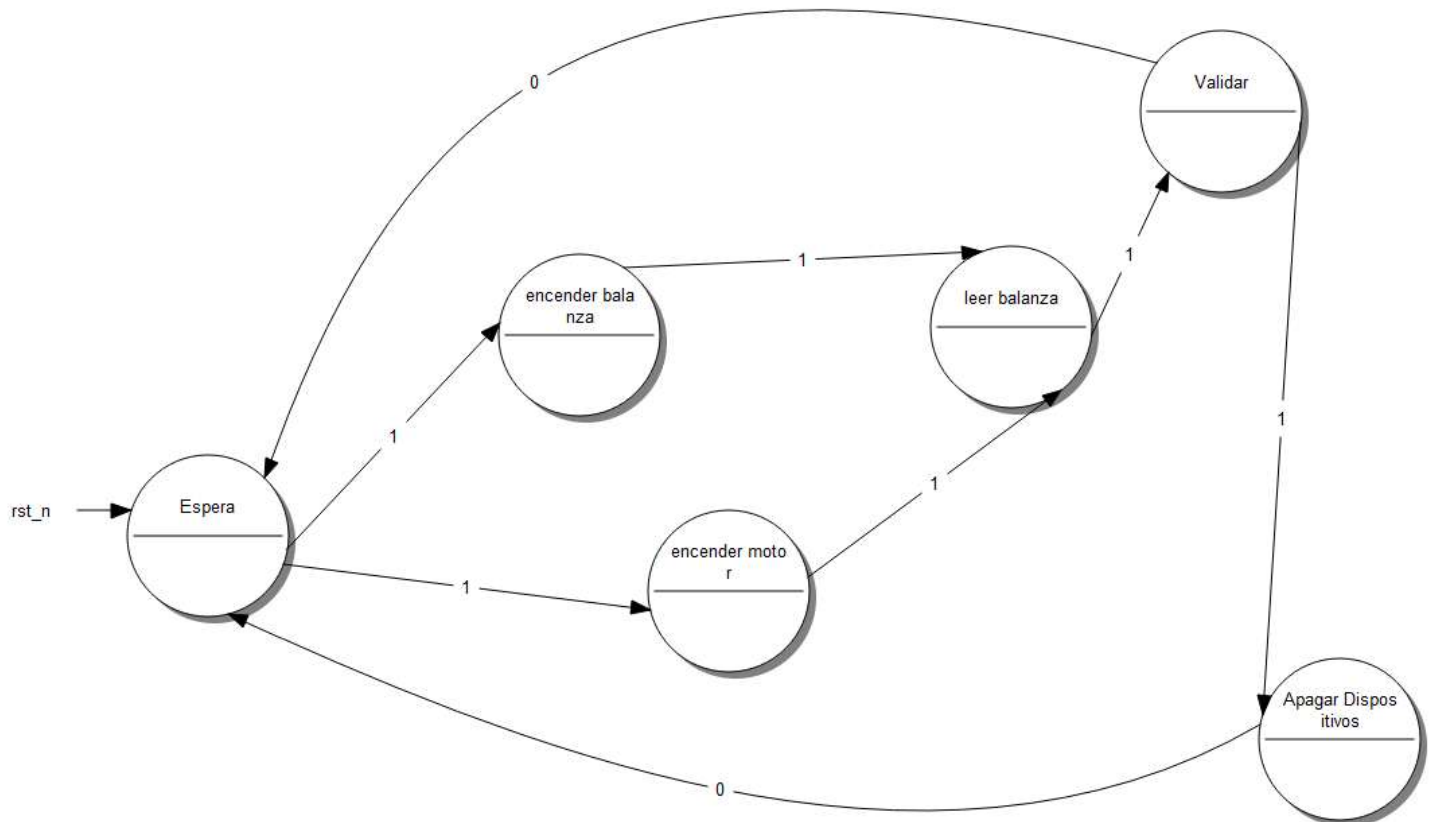
El siguiente diagrama en la Ilustración 7 representa el proceso mediante el cual se da limpieza al material biológico en la jaula, este sólo consta de una acción, donde se enciende un motor para mover los residuos fecales a través de un tubo.



*Ilustración 7 Diagrama de desechos  
Fuente propia*

### ▪ Diagrama de Alimentación

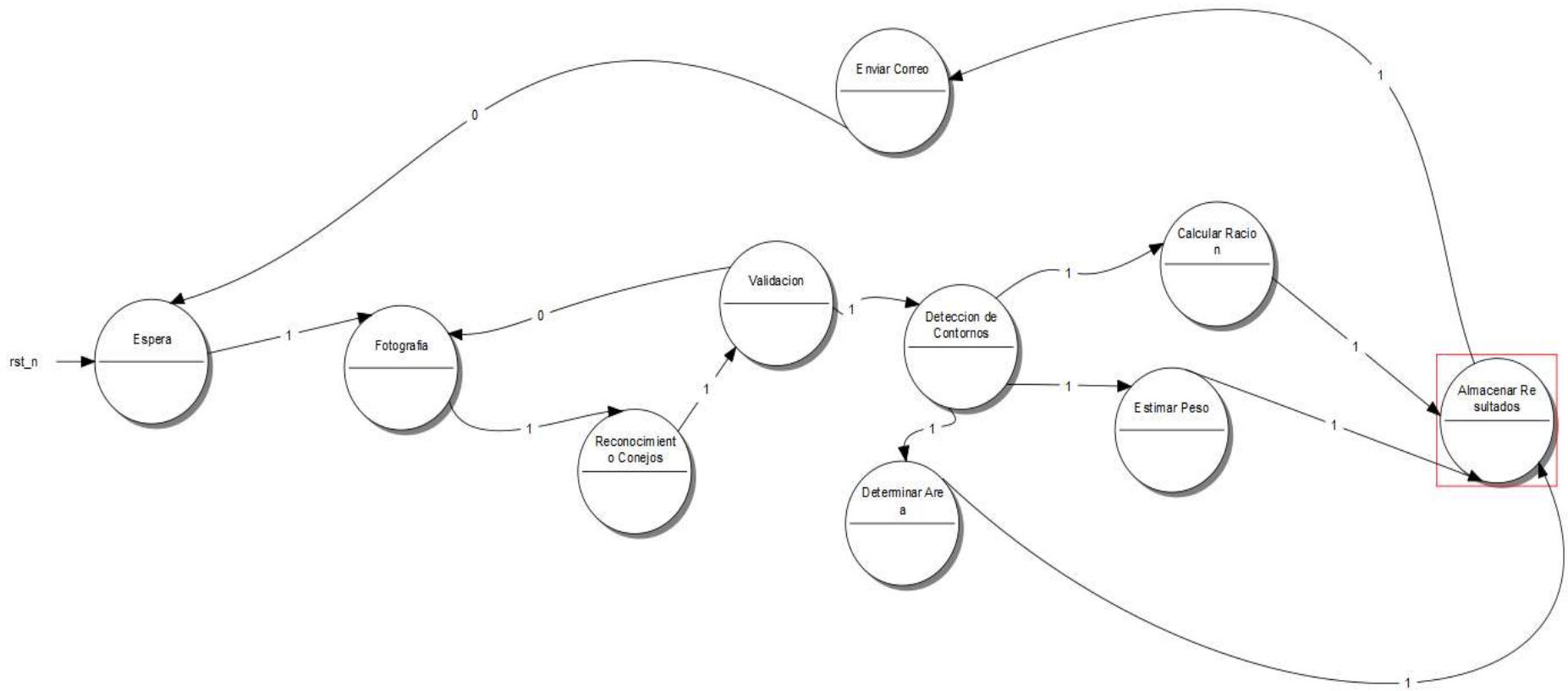
En representación de los procesos de alimentación para los conejos dentro de una jaula dada en la Ilustración 8, consta de varias acciones como lo son el encender los dispositivos, realizar la validación de los datos, estas son de gran importancia para que el sistema funcione adecuadamente.



*Ilustración 8 Diagrama de alimentación  
Fuente propia*

### ▪ Diagrama Tratamiento de Imágenes

Este proceso se clasifica como el más complejo del sistema donde se empieza por la toma de imágenes, procesa para reconocer a los individuos y se valida, luego se determina el área, se calcula el peso y la ración correspondiente, se hace el almacenamiento de la información, y se envía correo de información generada, se muestra en la Ilustración 9.

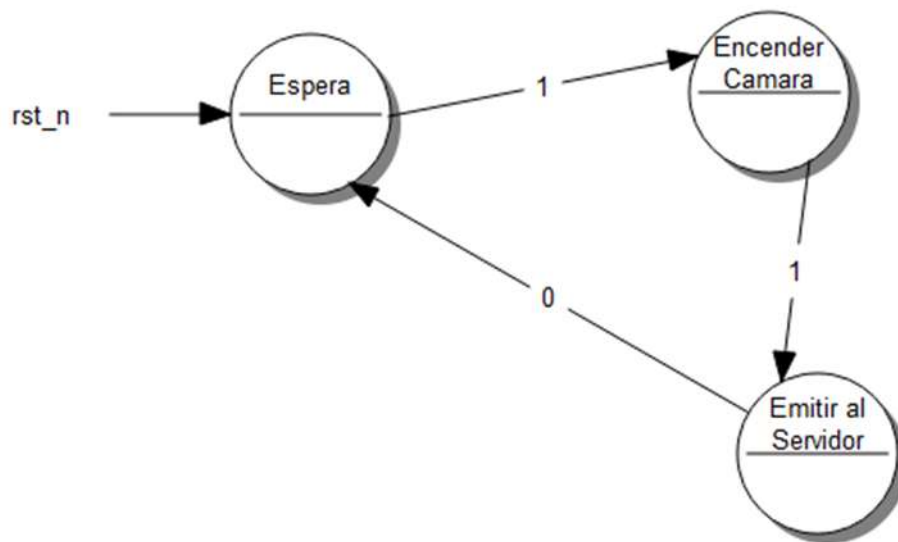


*Ilustración 9 Diagrama tratamiento de imagen  
Fuente propia*



### ▪ Diagrama Monitoreo

El diagrama de monitoreo representado en la Ilustración 10, solo consta de una cámara y realizar el *streaming* de video, representándose de la siguiente manera.



*Ilustración 10 Diagrama de monitoreo  
Fuente propia*

### 4.2 Toma de imágenes

En este proceso de toma de imágenes se usa una cámara en la parte superior del prototipo de la jaula, para este desarrollo se usó un celular conectado a Linux por medio de la red WIFI, donde a través de Python se realiza un script el cual accede a la cámara importando la librería OpenCV que permite realizar la captura de la foto, posteriormente se realiza el procesamiento de imágenes usando TensorFlow, Keras y basado en un modelo para reconocimiento de objetos llamado MaskRCNN el cual segmenta las máscaras en una imagen, para luego proceder a la detección del área de los individuos a analizar y obtener un peso estimado aproximado, con los datos obtenidos se actualiza la base de datos.

Para el proceso de entrenamiento se procede a generar un *dataset* que consta de varias imágenes

de los objetos a identificar (en este caso conejos), dos carpetas para validación y entrenamiento, las cuales mediante *VGG Image Annotator (VIA)* se generan anotaciones, como se muestra en la Ilustración 11.



*Ilustración 11 Imagen tratada con VIA*  
*Fuente propia*

Luego de procesar todas las imágenes en las carpetas de validación y entrenamiento se genera automáticamente un documento que contiene una estructura similar a la siguiente presentada en la Ilustración 12.

```

# Load annotations
# VGG Image Annotator saves each image in the form:
# { 'filename': '28503151_5b5b7ec140_b.jpg',
#   'regions': {
#     '0': {
#       'region_attributes': {},
#       'shape_attributes': {
#         'all_points_x': [...],
#         'all_points_y': [...],
#         'name': 'polygon'}}},
#   ... more regions ...
# },
# 'size': 100202
# }

```

*Ilustración 12 Json generado por VIA  
Fuente (github, 2018)*

Se procede el entrenamiento del modelo el cual es un proceso algo lento que requiere gran capacidad de rendimiento, este proceso varía dependiendo la capacidad de una máquina u otra, al final de este proceso tendremos un archivo generado en formato h5, el cual es el modelo entrenado para su uso en la detección de los conejos, Ilustración 13, como ejemplo del proceso de entrenamiento.

```

Epoch 2/10
5/100 [>.....] - ETA: 2:29 - loss: 0.2036 - rpn_class_
loss: 6.2893e-04 - rpn_bbox_loss: 0.0113 - mrcnn_class_loss: 0.0038 - mrcnn_bbox
6/100 [>.....] - ETA: 2:28 - loss: 0.1979 - rpn_class_
loss: 6.1877e-04 - rpn_bbox_loss: 0.0125 - mrcnn_class_loss: 0.0033 - mrcnn_bbox
7/100 [=>.....] - ETA: 2:38 - loss: 0.1909 - rpn_class_
loss: 6.0774e-04 - rpn_bbox_loss: 0.0130 - mrcnn_class_loss: 0.0041 - mrcnn_bbox
8/100 [=>.....] - ETA: 2:32 - loss: 0.1844 - rpn_class_
loss: 5.9591e-04 - rpn_bbox_loss: 0.0127 - mrcnn_class_loss: 0.0037 - mrcnn_bbox
9/100 [=>.....] - ETA: 2:29 - loss: 0.2390 - rpn_class_
loss: 7.2536e-04 - rpn_bbox_loss: 0.0222 - mrcnn_class_loss: 0.0048 - mrcnn_bbox
10/100 [==>.....] - ETA: 2:30 - loss: 0.2704 - rpn_class_
loss: 8.2870e-04 - rpn_bbox_loss: 0.0291 - mrcnn_class_loss: 0.0049 - mrcnn_bbox
11/100 [==>.....] - ETA: 2:32 - loss: 0.2855 - rpn_class_
loss: 9.1230e-04 - rpn_bbox_loss: 0.0339 - mrcnn_class_loss: 0.0065 - mrcnn_bbox
12/100 [==>.....] - ETA: 2:37 - loss: 0.2945 - rpn_class_
loss: 9.8031e-04 - rpn_bbox_loss: 0.0368 - mrcnn_class_loss: 0.0065 - mrcnn_bbox
13/100 [==>.....] - ETA: 2:37 - loss: 0.3075 - rpn_class_
loss: 0.0010 - rpn_bbox_loss: 0.0383 - mrcnn_class_loss: 0.0068 - mrcnn_bbox
14/100 [==>.....] - ETA: 2:32 - loss: 0.3144 - rpn_class_
loss: 0.0011 - rpn_bbox_loss: 0.0391 - mrcnn_class_loss: 0.0068 - mrcnn_bbox
15/100 [==>.....] - ETA: 2:35 - loss: 0.3143 - rpn_class_
loss: 0.0011 - rpn_bbox_loss: 0.0395 - mrcnn_class_loss: 0.0069 - mrcnn_bbox

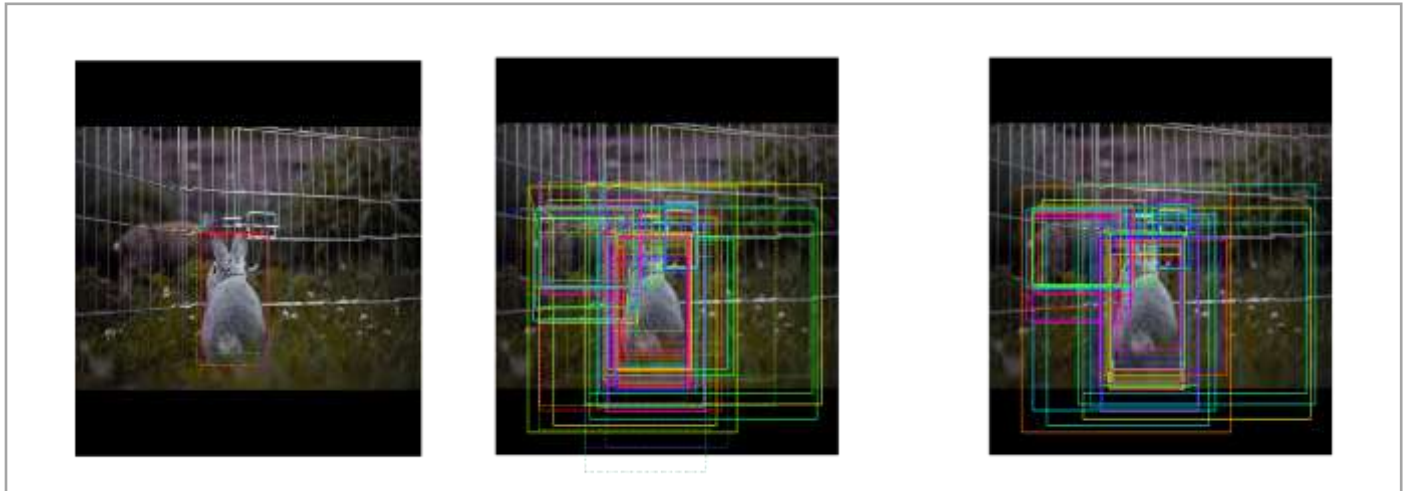
```

*Ilustración 13 entrenamiento del modelo MaskRCNN  
Fuente propia*

Validando el funcionamiento del modelo se realizan las siguientes actividades:

- **Clasificación**

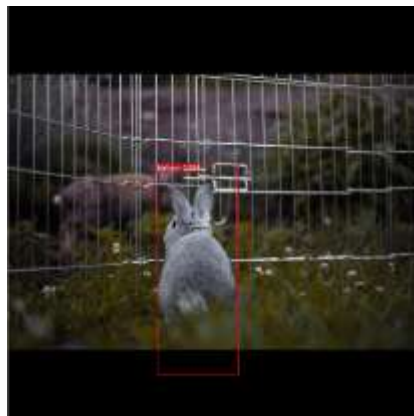
Visualiza la región propuesta de la muestra anclajes positivos y negativos junto con el refinamiento de la caja de anclaje, ver Ilustración 14.



*Ilustración 14 clasificación de la imagen a tratar  
Fuente propia*

- **Refinamiento del cuadro delimitador**

Representado en la Ilustración 15, dando como resultado un sólo cuadro delimitador.



*Ilustración 15 Refinamiento de la imagen a tratar  
Fuente propia*

- **Generacion de máscaras en la imagen**

El cual se representa en la Ilustración 16, luego de haber realizado las operaciones correspondientes de detección en la imagen.



*Ilustración 16 Máscara de imagen tratada*  
*Fuente propia*

Estando la máscara generada se procede a la detección del área de las máscaras usando OpenCV, ya que la imagen es binaria se realiza el dibujo del contorno según la forma del área, guardando esa información en variables, se consulta la base de datos y se compara si la cantidad de máscaras es diferente a la cantidad de conejos en la jaula, si este valor es diferente se repite nuevamente todo el proceso, dando como resultado una imagen similar a la Ilustración 17.



*Ilustración 17 Contorno de las mascara y área correspondiente  
Fuente propia*

Sí el proceso culmina exitoso, es decir la cantidad de conejos es igual a las máscaras se actualiza el peso de los conejos con relación al área en la base de datos y se envía las imágenes al correo del cliente, este proceso se realiza cada semana de manera automática gracias a la librería de Python crontab, la cámara se encuentra a 1.06 Metros de la base de la jaula.

### **4.3 Automatización del proceso de alimentación del prototipo**

#### **4.3.1 Dispensador de alimentos**

- **Agua**

El agua será de libre consumo para los conejos por esta razón se utiliza un bebedero manual el cual los conejos puedan activar por sus propios medios cuando tengan la necesidad, similar a la Ilustración 18.



*Ilustración 18 Dispensador de agua  
Fuente propia*

- **Alimento sólido**

Se realiza un sistema automático por medio de una tarjeta Arduino conectada a WIFI la cual mediante un servidor web básico recibe la cantidad de comida que debe suministrar, automáticamente enciende el motor que empuja la comida por un tubo utilizando un sinfín de resorte hasta que empieza a caer en un plato, como se muestra en la Ilustración 19.



*Ilustración 19 Dispensador de alimento solido  
Fuente propia*

#### **4.3.2 Control de alimentación**

Controlado también por Arduino, el plato donde cae el alimento posee debajo una balanza, la cual mide la cantidad de comida y finaliza el proceso de empuje del alimento hasta que alcanza el gramaje indicado por el sistema, Python se encarga de consultar la ración, en la base de datos, repitiendo este proceso tres veces al día, usando crontab.

#### **4.3.3 Diseño para recolector de desechos biológicos**

El mecanismo está controlado por una placa Arduino aparte, la cual tiene un motor que se enciende cada hora por medio de una banda de anjeo la cual transporta los desechos sólidos a una bolsa, ya que la banda es de anjeo los desechos líquidos pasan directo a un tubo, de esta manera se separan los desechos sin intervención humana.

#### **4.3.4 Selección de la instrumentación a utilizar**

- 2 Placas Arduino
- 1 Celda de Carga 50 kg
- 1 Módulo HX711
- 2 Motoreductores 10 kg a 6 rpm
- 1 Placa Raspberry Pi 4
- Celular Android cámara 8 Mpx
- 2 Módulos esp8266
- Webserver en DigitalOcean
- 2 Relee de 5v a 12v



## Microcontroladores

La tabla 8, muestran las alternativas que se pueden implementar para microcontroladores, buscando analogías que se ajusten a la necesidad, aunque no se incluye en esta tabla, la Raspberry también es una alternativa a utilizar.

Dispositivo	Características	Precio	Selección	Criterio de selección
<b>Arduino</b>	<ul style="list-style-type: none"> <li>• Microcontrolador: ATmega2560</li> <li>• Voltaje Operativo: 5V</li> <li>• Voltaje de Entrada: 7-12V</li> <li>• Voltaje de Entrada(límites): 20V</li> <li>• Pines digitales de Entrada/Salida: 54 (de los cuales 15 proveen salida PWM)</li> <li>• Pines análogos de entrada: 16</li> <li>• Corriente DC por cada Pin Entrada/Salida: 40 mA</li> <li>• Corriente DC entregada en el Pin 3.3V: 50 mA</li> <li>• Memoria Flash: 256 KB (8KB usados por el bootloader)</li> <li>• SRAM: 8KB</li> <li>• EEPROM: 4KB</li> <li>• Clock Speed: 16 MHz</li> </ul>	\$45.000	x	El precio, las especificaciones ofrecidas por la placa son suficientes para los procesos de automatización, por otro lado la durabilidad de un Arduino tiende a ser mayor a la de una placa BeagleBone
<b>BeagleBone</b>	<ul style="list-style-type: none"> <li>• Procesador AM335x ARM A8 a 1GHz.</li> <li>• 512MB de memoria RAM DDR3.</li> <li>• Gráfica SGX530 para procesar gráficos en 3D.</li> <li>• Salida de audio y vídeo a través del puerto mini-HDMI.</li> <li>• Tarjeta de red Fast Ethernet 10/100Mbps.</li> <li>• 1 puerto USB 2.0.</li> <li>• 2GB de memoria interna (para el Sistema Operativo) y la posibilidad de ampliarla</li> </ul>	\$271.000		

	mediante una memoria USB o una micro-SD. <ul style="list-style-type: none"><li>• Dispone de botones de encendido, reset y boot.</li></ul>			
--	---	--	--	--

*Tabla 8 Microcontroladores*

*Fuente propia*

## Controlador principal

Se presentan en la tabla 9, las alternativas y características de controladores principales, los cuales pueden ser implementados en el proceso de automatización.

Dispositivo	Características	Precio	Selección	Criterio de selección
Raspberry Pi	<ul style="list-style-type: none"> <li>• Sistema en un chip: Broadcom BCM2711</li> <li>• CPU: Procesador de cuatro núcleos a 1,5 GHz con brazo Cortex-A72</li> <li>• GPU: VideoCore VI</li> <li>• Memoria: 1/2/4GB LPDDR4 RAM</li> <li>• Conectividad: 802.11ac Wi-Fi / Bluetooth 5.0, Gigabit Ethernet</li> <li>• Vídeo y sonido: 2 x puertos micro-HDMI que admiten pantallas de 4K@60Hz a través de HDMI 2.0, puerto de pantalla MIPI DSI, puerto de cámara MIPI CSI, salida estéreo de 4 polos y puerto de vídeo compuesto.</li> <li>• Puertos: 2 x USB 3.0, 2 x USB 2.0</li> <li>• Alimentación: 5V/3A vía USB-C, 5V vía cabezal GPIO</li> <li>• Expansión: Cabezal GPIO de 40 pines</li> </ul>	\$320.000	X	El principal motivo de selección de esta placa se fundamenta en que la granja ya contaba con este dispositivo para otras funciones, solo sería necesario anexar a sus procesos la nueva funcionalidad.
RockPi 4	<ul style="list-style-type: none"> <li>• Chip : Allwinner H6</li> <li>• CPU : procesador Quad-Core, basado en Arm Cortex A53</li> <li>• GPU : Mali-T720 MP2 de doble núcleo</li> <li>• Memoria : 1/2 / 3GB LPDDR3</li> <li>• Almacenamiento : módulo flash eMMC, tarjeta micro SD (de arranque), flash SPI de 128 Mbit</li> <li>• Vídeo : salida HDMI 2.0a</li> <li>• Audio : auricular estéreo de 3,5 mm / conector de micrófono</li> <li>• Conectividad : Gigabit Ethernet, Wi-Fi, Bluetooth</li> <li>• USB : 2 x host USB 2.0, 1 x host USB 3.0</li> </ul>	\$450.000		

Computador convencional	<input type="checkbox"/> Procesador AMD Dual-Core serie A4 <input type="checkbox"/> Memoria RAM: 4 GB <input type="checkbox"/> Disco Duro: 64 GB <input type="checkbox"/> Pantalla HD 14" pulgadas <input type="checkbox"/> Gráficos: AMD Radeon R3 <input type="checkbox"/> Peso: 1.51Kg <input type="checkbox"/> WIFI, Bluetooth <input type="checkbox"/> USB, HDMI, RJ-45, Web cam incorporada	\$700.000		
-------------------------	--	-----------	--	--

*Tabla 9 Controlador Principal*

*Fuente propia*

## Sensores y módulos

En la tabla 10, se dan a conocer las características de algunos sensores y módulos que se utilizan en el sistema de información.

Dispositivo	Características	Precio	Selección	Criterio de selección
<b>HX711</b>	<ul style="list-style-type: none"> <li>• Digitalización completa en el chip HX711</li> <li>• Conexión a protoboard o tarjeta con header estándar</li> <li>• 2 canales diferenciales de entrada</li> <li>• Interfaz de salida digital serial</li> <li>• Regulador integrado en chip para fuente analógica</li> <li>• Voltaje de operación: 2.7 – 5 volts</li> <li>• Corriente de operación : &lt; 1.5mA</li> <li>• Corriente en espera: &lt;1 uA</li> <li>• Operación seleccionable: 80 y 10 muestras por segundo</li> <li>• Funciona con Arduino, Microcontroladores PIC y de otras marcas</li> </ul>	\$22.000	X	Incluido con la balanza, se toma como única opción, ya que por lo general las balanzas electrónicas utilizan este módulo y/o sistema, siendo el único a la venta en el mercado colombiano.
<b>Esp8266</b>	<ul style="list-style-type: none"> <li>• 32-bit RISC CPU: Tensilica Xtensa LX106 corriendo a 80 MHz (que puede ser overclokeado a 160MHz si se requiere)</li> <li>• 64 KiB de RAM para instrucciones y 96 KiB de RAM para datos</li> <li>• IEEE 802.11 b/g/n Wi-Fi</li> <li>• 16 pines GPIO</li> <li>• SPI e I2C</li> <li>• UART en los pines dedicados (usada para la programación del chip)</li> <li>• Un convertidor Analógico-Digital (ADC) de 10 bit</li> </ul>	\$16.000	X	El precio, y las características son suficientes para establecer una comunicación inalámbrica por medio de la transmisión wifi

<b>Esp32</b>	<ul style="list-style-type: none"> <li>• Procesador dual core <b>Xtensa® LX6</b> de 32 bits</li> <li>• Es compatible Arduino con el pluggin adecuado (<i>Enseguida vamos</i>)</li> <li>• Velocidad de reloj : Entre 160 Mhz y 240 Mhz</li> <li>• 520 Kb de RAM</li> <li>• <b>Wifi</b> integrado: Acces point &amp; Station</li> <li>• Bluetooth 4.2 2.4 Ghz; BT 2.0 y 4.0 BLE</li> <li>• 36 GPIO pins.</li> <li>• 16 x <b>Analog-to-Digital Converter (ADC)</b> de 12 bits de resolución y se pueden programar con límite de entrada a 1V, 2 V y 4V</li> <li>• 2 x Digital to Analog converter DAC de 8 bits.</li> <li>• Pueden definirse hasta 16 canales de PWM.</li> <li>• 2 x UART o puertas serie</li> <li>• 2 x I2C channels y 4 x SPI channels.</li> <li>• La cosa sigue pero mejor lo dejamos aquí para no asustar.</li> </ul>			
--------------	--	--	--	--

*Tabla 10 Sensores y módulos*

*Fuente propia*

## Motores

En la tabla 11, se muestran las características de alternativas de motores para su implementación en el sistema, aunque según la necesidad se presenta el motor, por lo general es de fácil uso el Motorreductor además la fuerza que este tiene es lo bastante buena analizando precio, calidad y fuerza.

<b>Dispositivo</b>	<b>Características</b>	<b>Precio</b>	<b>Selección</b>	<b>Criterio de selección</b>
<b>Motorreductor</b>	<ul style="list-style-type: none"> <li>• 12v</li> <li>• 15 kg Fuerza</li> </ul>	\$25.000	X	Se presenta como solución al presentar la mayor fuerza entre

				los tipos de motor.
<b>Stepper</b>	<ul style="list-style-type: none"> <li>• 12v</li> <li>• 2 kg fuerza</li> </ul>	\$20.000		
<b>Servomotor</b>	<ul style="list-style-type: none"> <li>• 12v</li> <li>• 360 grados de giro</li> <li>• 2 kg fuerza</li> </ul>	\$22.000		
<b>Motor eléctrico</b>	<ul style="list-style-type: none"> <li>• 5v</li> <li>• 1 kg fuerza</li> </ul>	\$16.000		

*Tabla 11 Motores*

*Fuente propia*

## Cámara

La tabla 12, hace referencia a algunas de las características mas importantes presentes en las alternativas para cámaras.

<b>Dispositivo</b>	<b>Características</b>	<b>Precio</b>	<b>Selección</b>	<b>Criterio de selección</b>
<b>Celulares</b>	Conexión inalámbrica	\$250.000	X	Se contaba con estos elementos a disposición, no estaban en uso, y se llevó a cabo la reutilización
<b>Webcam</b>	Conexión alámbrica/inalámbrica	\$65.000		
<b>Cámara digital</b>	Conexión alámbrica	\$320.000		

*Tabla 12 Cámaras*

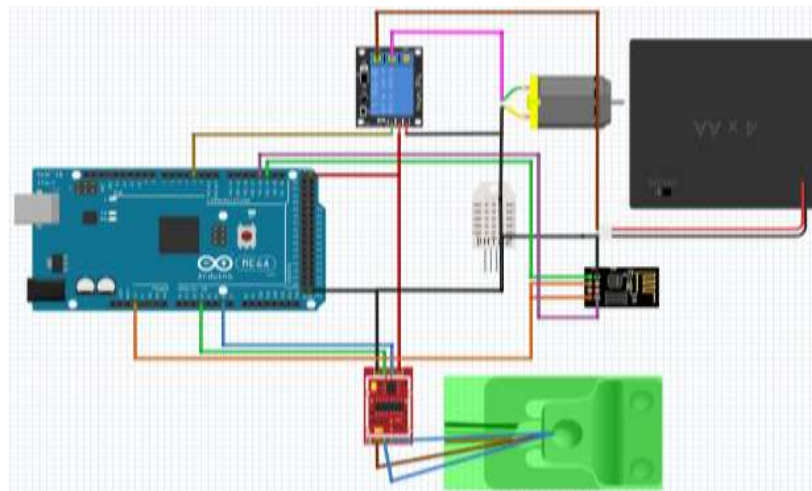
*Fuente propia*

La razón de peso para la selección de la tecnología a utilizar, fue el poco consumo de energía, el bajo costo, la calidad de los implementos según la necesidad, la vida útil y resistencia a polvo; se utiliza una Raspberry pi 4 comparado con otra tecnología, porque ofrece posibilidad de instalar Linux como sistema operativo facilitando la ejecución de los scripts en Python, y la conexión con la cámara del celular por medio de la red WIFI y sus especificaciones en cuanto a la RAM y procesador son similares a las de un computador de gama media-baja.

## 4.4 Diseño e implementación de la automatización

### 4.4.1 Sistema de alimentación automatizada

Se muestra la representación de los dispositivos y las conexiones presentes en el sistema de alimentación, hacen parte de la lista de dispositivos una placa Arduino, que mantiene comunicación directa con la Raspberry mediante conexión WIFI mediante un módulo esp8266 conectada a los pines seriales de la placa Rx1 y Tx1, el sistema cuenta también con un relé para controlar el voltaje necesario para el funcionamiento del motorreductor conectado en el pin digital 3, un módulo Hx711 el cual es el encargado de traducir la señal enviada de la celda de carga (Balanza) a la placa Arduino conectada a los pines análogos 6 y 3, una fuente de alimentación de 12v conectada al relé, representado en la Ilustración 20.

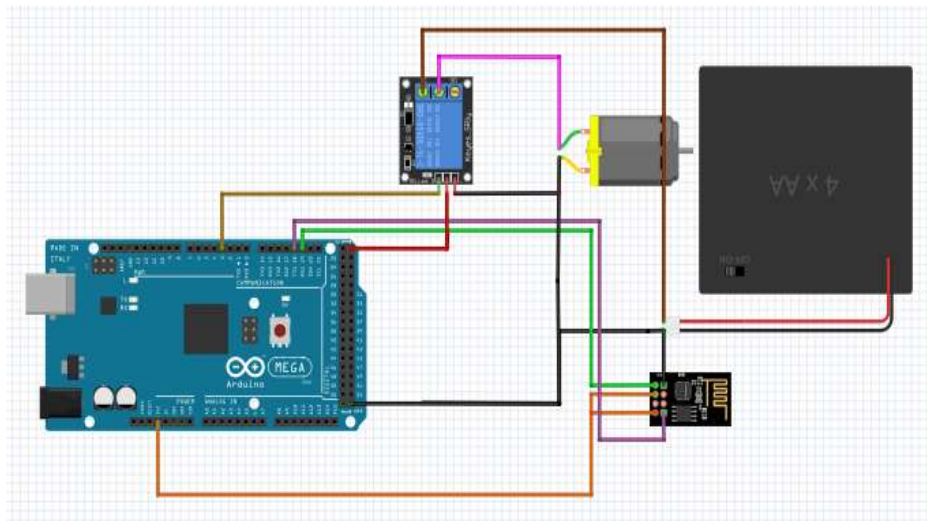


*Ilustración 20 Diagrama de conexión para el dispensador de alimento  
Fuente propia*

### 4.4.2 Sistema de recolección de desechos

El siguiente esquema muestra las conexiones en el sistema de eliminación de desechos del prototipo, el cual es muy similar al de alimentación pero eliminando las conexiones de los módulos correspondientes a la balanza, este sistema consta de una placa arduino, un módulo esp8266, un relé, un motorreductor y fuente de alimentación de 12v, el cual se muestra en la Ilustración 21.

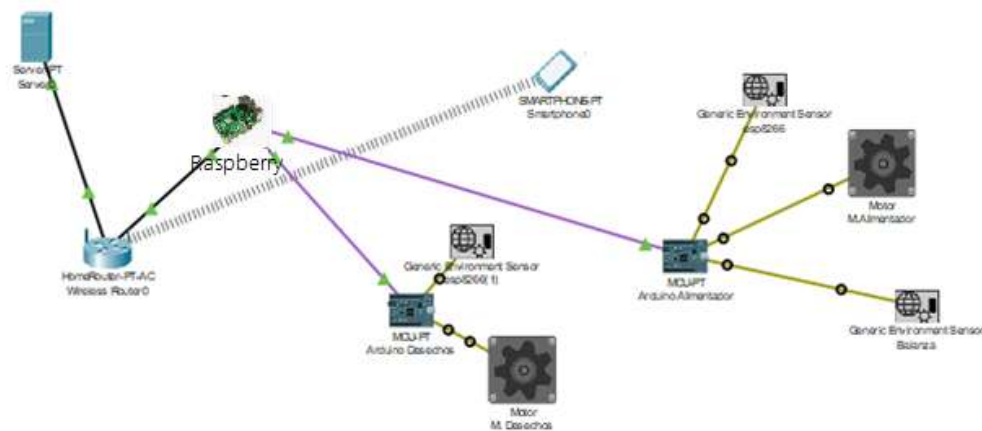




*Ilustración 21 Sistema de recolección de desechos*  
Fuente propia

#### 4.4.3 Esquema general

Para el sistema, las comunicaciones entre los dispositivos es indispensable para su correcto funcionamiento. El siguiente diagrama, Ilustración 22, muestra un esquema de dichas conexiones, se establecen a través de un router WIFI el cual esta conectado a internet, la placa Raspberry como dispositivo principal es el encargado de enviar las señales a las placas arduino correspondientes de la alimentación y la eliminación de desechos.



*Ilustración 22 Esquema general de conexión*  
Fuente propia

## 5 Validación y análisis de la automatización

En este capítulo se muestran los algoritmos implementados para la automatización de los procesos en la granja cunícola, entre estos procesos se encuentran el tratamiento de imágenes, alimentación y desechos.

### 5.1 Desarrollo del algoritmo para la obtención de los datos a partir de las imágenes

A partir de los datos generados (Área) en el código se realiza el muestreo tomando varios conejos de diferente tamaño, este muestreo consistió en tomar fotos en las mismas condiciones establecidas para el experimento inicial, la distancia de la cámara a la base de la jaula (1.06 m), el tipo de jaula (Jaula redonda), tipo de cámara (cámara celular de 8 mpx); Luego de la toma de fotografías se procede a tomar el peso de los animales, para así generar una tabla que relaciona el área con el peso, para así, generar una tercera variable (Ración), la cual fue calculada mediante la información nutricional proporcionada por las empresas de concentrados y agricultores de la granja cunícola.

```
def Main(img):
//Se carga el modelo;
    results = model.detect([img], verbose=0)
    r = results[0]

//Procesamiento de imagen por MaskRCNN;
    frame = display_instances(
        img, r['rois'], r['masks'], r['class_ids'], class_names, r['scores']
    )

//Conversión de la imagen a blanco y negro exceptuando la máscara;
    splash = color_splash(img, r['masks'])
    original = splash
    img = original.copy()

    cv2.imshow('all', original)
    img[:, :, 2] = 0
//Selección de banda de colores;
    im_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
//Obteniendo colores reflejados en azul según la banda;
```

```

low_blue = np.array([100, 50, 50])
high_blue = np.array([130, 255, 255])

mask = cv2.inRange(im_hsv, low_blue, high_blue)
umbral = cv2.threshold(mask, 25, 255, cv2.THRESH_BINARY)[1]

# Dilatación del umbral para tapar agujeros
umbral = cv2.dilate(umbral, None, iterations=2)

# Copia del umbral para detectar los contornos
contornosimg = umbral.copy()

contours, _ = cv2.findContours(contornosimg, cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
areas = []
# Dibujar los contornos
for c in contours:
    area = cv2.contourArea(c)
    if area > 1000 and area < 100000:
        cv2.drawContours(original, [c], 0, (0, 255, 0), 2, cv2.LINE_AA)

        momentos = cv2.moments(c)
        cx = int(momentos['m10'] / momentos['m00'])
        cy = int(momentos['m01'] / momentos['m00'])

        # Dibujar el centro y calcular el área del contorno
        cv2.circle(original, (cx, cy), 3, (0, 255, 0), -1)
        area2 = cv2.contourArea(c)
        cm = area2 * 2.54 / 96
        areas.append(cm)
        # inserción de las coordenadas del centro y área en la imagen
        cv2.putText(original, "(x: " + str(cx) + ", y: " + str(cy) + ", Area: " + str(cm) + ")",
            (cx + 10, cy + 10), font, 0.5, (255, 255, 255), 1)

# se muestra la imagen final
cv2.imshow('Final', original)

while (1):
    tecla = cv2.waitKey(5) & 0xFF
    if tecla == 27:
        break

cv2.destroyAllWindows()

return int(r['rois'].shape[0]), areas, original

```

```

def calcP(Area):
    # Lectura del csv y obtención de columnas
    df = pd.read_csv(r'tconejos.csv')
    x = df['Area'].values
    y = df['Peso'].values
    z = df['Racion'].values

    x = x.reshape(-1, 1)

    # Se crea la regresión
    lin = LinearRegression()

    lin.fit(x, y)

    poly = PolynomialFeatures(degree=3)
    X_poly = poly.fit_transform(x)

    poly.fit(X_poly, y)
    lin2 = LinearRegression()
    lin2.fit(X_poly, y)

    Peso = lin2.predict(poly.fit_transform([[Area]]))

    suma = (df['Peso'].values)

    dataX2 = pd.DataFrame()
    dataX2["Area"] = df['Area'].values
    dataX2["suma"] = suma
    XY_train = np.array(dataX2)
    z_train = df['Racion'].values

    # Creamos un nuevo objeto de Regresión Lineal
    regr2 = LinearRegression()

    # Entrenamos el modelo, esta vez, con 2 dimensiones
    regr2.fit(XY_train, z_train)

    poly2 = PolynomialFeatures(degree=3)
    X_poly2 = poly2.fit_transform(XY_train)

    poly2.fit(X_poly2, z_train)
    lin3 = LinearRegression()
    lin3.fit(X_poly2, z_train)
    Racion = lin3.predict(poly2.fit_transform([[Area, Peso]]))

    print(Peso)

```

`print(Racion)`

`return` Peso, Racion

## 5.2 Análisis de datos (Datos reales y datos experimentales)

Se muestra a continuación la Tabla 8, donde se ven las relaciones obtenidas a partir del censo de animales, la cual se implementa en una regresión polilínea para la correcta estimación de las variables.

Área (Cm2)	Peso (g)	Ración (g)
100	110	25
158	300	30
185	600	35
200	900	50
240	1200	63
297	1700	70
338	2000	80
440	2600	85
528	2997	97
580	3405	108
850	3750	140
1000	4000	170
1300	4600	175
1590	5100	180

*Tabla 13 Datos reales y datos experimentales  
Fuente propia*

- **Regresión lineal**

El sentido de la regresión lineal es obtener una relación entre variables, para predecir el comportamiento de una variable cuantitativa, este algoritmo se implementa en este trabajo tomando el área como variable independiente y el peso como dependiente, dando este algoritmo una función lineal simple que se ajuste a los valores censados en la tabla anterior, teniendo en cuenta estas dos variables se hace una segunda regresión lineal multivariable que tome en cuenta la ración para estimar la cantidad de comida a ingerir por los conejos, procurando la mejor aproximación de acuerdo a su peso.

### 5.3 Desarrollo del algoritmo de automatización partiendo de los datos obtenidos en el tratamiento de imágenes

```
#include "HX711.h"
#define DEBUG_HX711

// Parámetro para calibrar el peso y el sensor
#define CALIBRACION 1940.0

// Pin de datos y de reloj
byte pinData = A1;
byte pinClk = A0;

// Objeto HX711
HX711 bascula;

#define LED 52
#define DEBUG true

HardwareSerial & esp8266 = Serial1;
// El TX del módulo al terminal 1 del Arduino.
// El RX del módulo al terminal 1 del Arduino.
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(115200);
  esp8266.begin(115200); // Importante la velocidad del módulo.
  pinMode(LED, OUTPUT);
  Serial.println("[HX711] Inicio del sensor HX711");
  // Iniciar sensor
  bascula.begin(pinData, pinClk);
  // Aplicar la calibración
  bascula.set_scale(CALIBRACION);
```

```

// Iniciar la tara
// No tiene que haber nada sobre el peso
bascula.tare();
digitalWrite(LED, LOW);

sendData("AT+RST\r\n", 2000, DEBUG); // Borra la configuración que tenía el módulo
sendData("AT+CWJAP=\"jlizcano\", \"jlizcano_2018_@.\">\r\n", 2000, DEBUG);
delay(5000); // Espera un poco que conecte con el Router.
sendData("AT+CWMODE=3\r\n", 1000, DEBUG); // Modo de cliente y servidor.
sendData("AT+CIFSR\r\n", 1000, DEBUG); // En el Serial Monitor aparece la IP de cliente y
servidor.
sendData("AT+CIPMUX=1\r\n", 1000, DEBUG); // Multiples conexiones.
sendData("AT+CIPSERVER=1,80\r\n", 1000, DEBUG); // El Puerto web es el 80
}

void loop()
{
  if (esp8266.available()) // Consulta si el módulo está enviando algún mensaje
  {

    if (esp8266.find("+IPD,"))
    {
      delay(1000);

      int connectionId = esp8266.read() - 48;
      // Aquí la construcción de la página con dos formularios.

      String webpage = "<head></head>";
      webpage += "<h1>Cunicola Farm</h1>";
      webpage += "<form method=\"get\" action=\"/\">";
      webpage += "<p>Peso: <input type=\"text\" name=\"peso\" size=\"40\"></p>";
      webpage += "<button type=\"submit\">Send</button></form>";
      webpage += "Estado del LED13: ";
      int p13 = digitalRead(LED_BUILTIN);
      webpage += p13;

      String cipSend = "AT+CIPSEND=";
      cipSend += connectionId;
      cipSend += ",";
      cipSend += webpage.length();
      cipSend += "\r\n";

      sendData(cipSend, 1000, DEBUG);
      sendData(webpage, 1000, DEBUG);

      // Cierra la conexión

```

```

String closeCommand = "AT+CIPCLOSE=";
closeCommand += connectionId;
closeCommand += "\r\n";

sendData(closeCommand, 3000, DEBUG);
}
}
}

// Función para Enviar datos al Servidor.
String sendData(String command, const int timeout, boolean debug)
{
String response = "";
esp8266.print(command); // Envía la información de command al servidor
long int time = millis();

while ( (time + timeout) > millis())
{
while (esp8266.available())
{
// A response van los datos que regresan al servidor.
char c = esp8266.read(); // Va leyendo carácter a carácter.
response += c;
}

}

//obtención de variable enviada por el cliente en el formulario
if (debug)
{
Serial.println(response);
if (response.indexOf("GET /?peso=") > 0) {
float val;
val = response.substring(response.indexOf("peso=") + 5,
response.indexOf("%3B")).toFloat();
digitalWrite(LED_BUILTIN, HIGH);
Serial.print("El peso es : ");
Serial.println(val);

float current_weight = (bascula.get_units());
Serial.println(current_weight, 3);
//validación de ración según valor ingresado en el formulario;
bool f = 0;
while (f == 0) {
float current_weight = (bascula.get_units());
if (current_weight < val) {

```



```

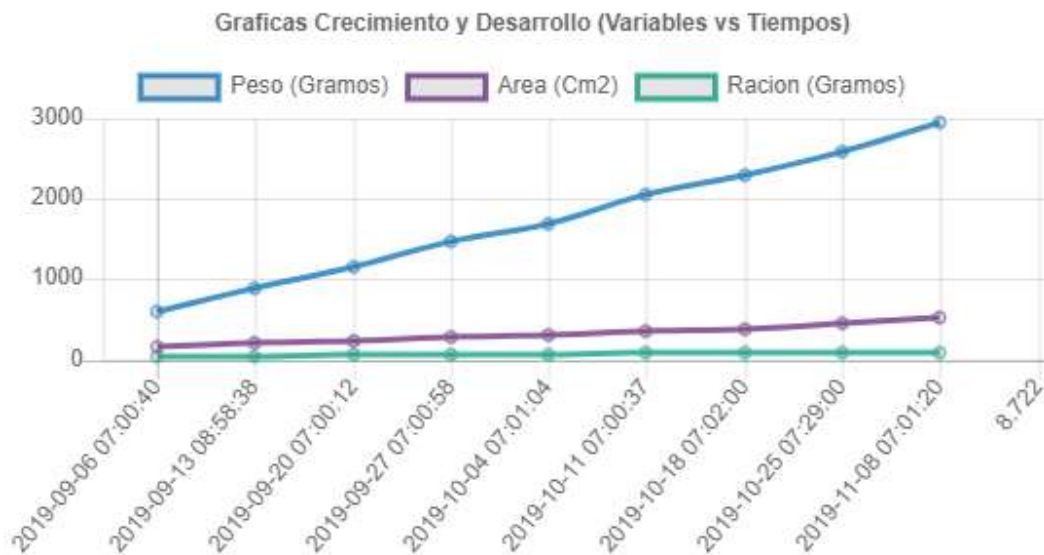
Serial.println(current_weight, 3);
delay(500);
digitalWrite(LED, HIGH);
}
else {
Serial.println(current_weight, 1);
digitalWrite(LED, LOW);
f = 1;
break;
}
}
digitalWrite(LED, LOW);
Serial.println(current_weight, 1);

}
}
return response;
}

```

#### 5.4 Comparación del prototipo vs cunicultura tradicional

La siguiente imagen, Ilustración 23, hace referencia a los datos registrados por el sistema automatizado y generados en la opción de reporte del sistema de información.



*Ilustración 23 Crecimiento Conejos Automático*

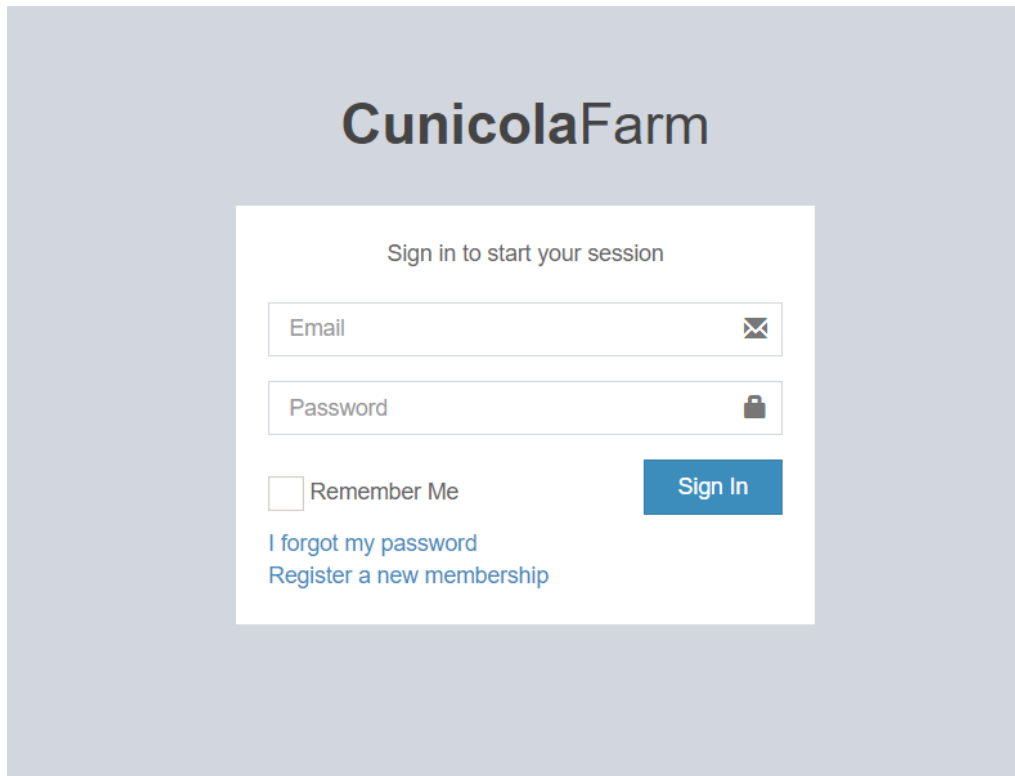
*Fuente propia*

La Ilustración 24, hace referencia a la gráfica de datos de una producción cunícola tradicional con la misma raza de conejos y el mismo tiempo de monitoreo, donde se puede apreciar el aumento en la ración alimenticia, cabe destacar que el 40% de la ración alimenticia es de concentrado Soya, el 60% restante es suministrado por la huerta incluido zanahoria, hortalizas, etc.



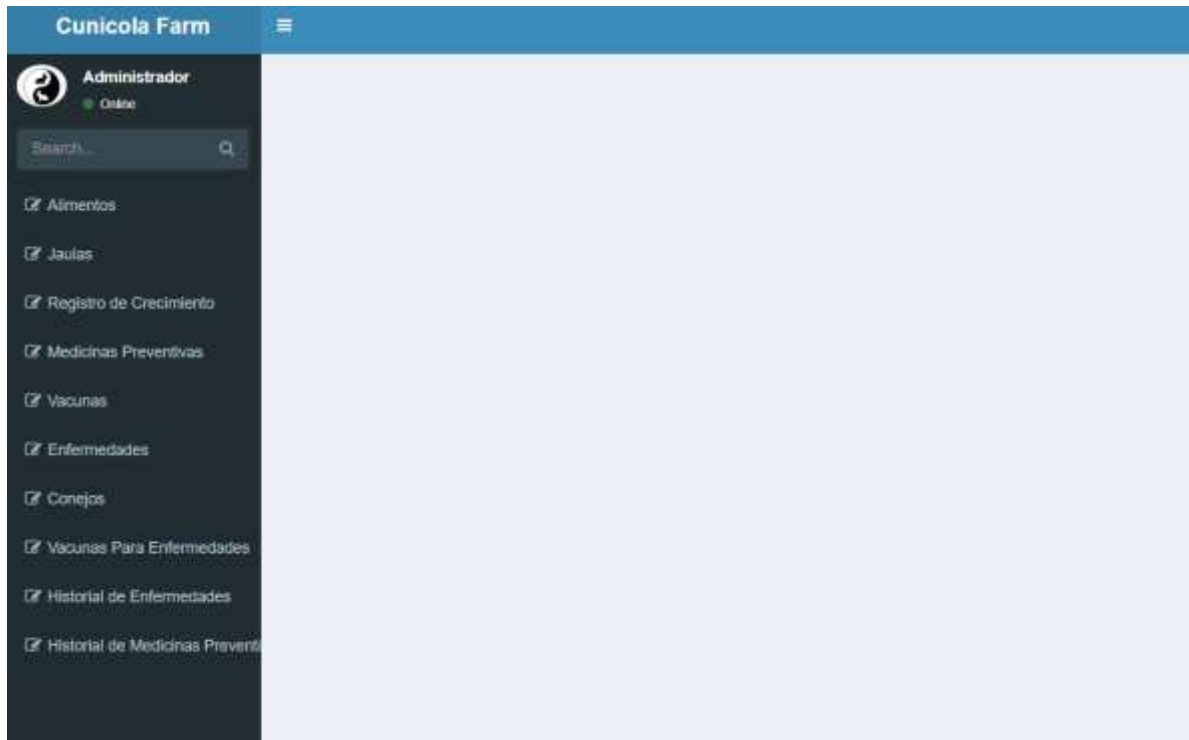
*Ilustración 24 Crecimiento de Conejos Tradicional*  
*Fuente propia*

En las dos Ilustraciones anteriores (23 y 24), a pesar que son la misma raza de conejos con la misma edad y tiempo de monitoreo, en el sistema tradicional de cunicultura presenta más gasto, aproximadamente un 13 kg en concentrado y 20 kg en hortalizas, para dar una ganancia aproximada de 2.900 gr, además no se está teniendo en cuenta los gastos generados por el cunicultor. El prototipo demostró utilidad en cuanto a la cunicultura ya disminuye los gastos en cría de estos, con el proceso de alimentación y monitoreo automáticos, junto a esto una ración adecuada a su tamaño, evitando sobre costos, lo cual es el principal objetivo de la cunicultura.



*Ilustración 25 Login de ingreso al sistema de información  
Fuente propia*

En la Ilustración 25, se observa la pantalla de login que permite el ingreso al sistema de información luego de que se valide el usuario en la base de datos, estando correctas sus credenciales de inicio, ingresa a la pantalla principal del sistema, la cual se puede evidenciar en la Ilustración 26, en la ventana principal (*home*), se puede verificar en la parte izquierda un panel de opciones donde el usuario interactúa e ingresa dependiendo de su necesidad a las opciones permitidas.

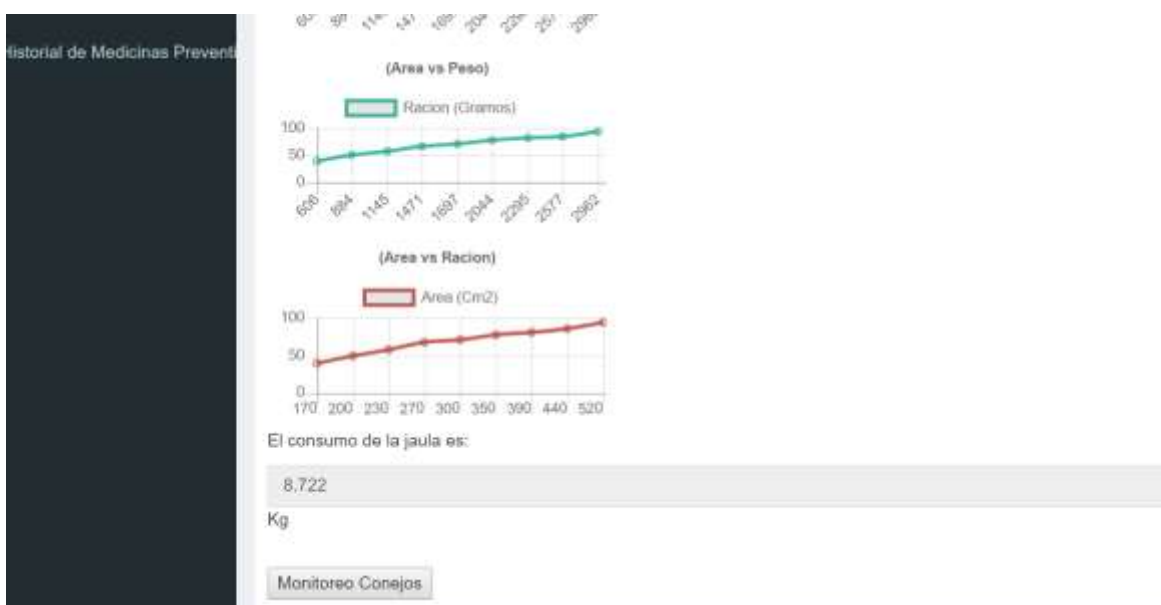


*Ilustración 26 Pantalla principal del sistema de información  
Fuente propia*

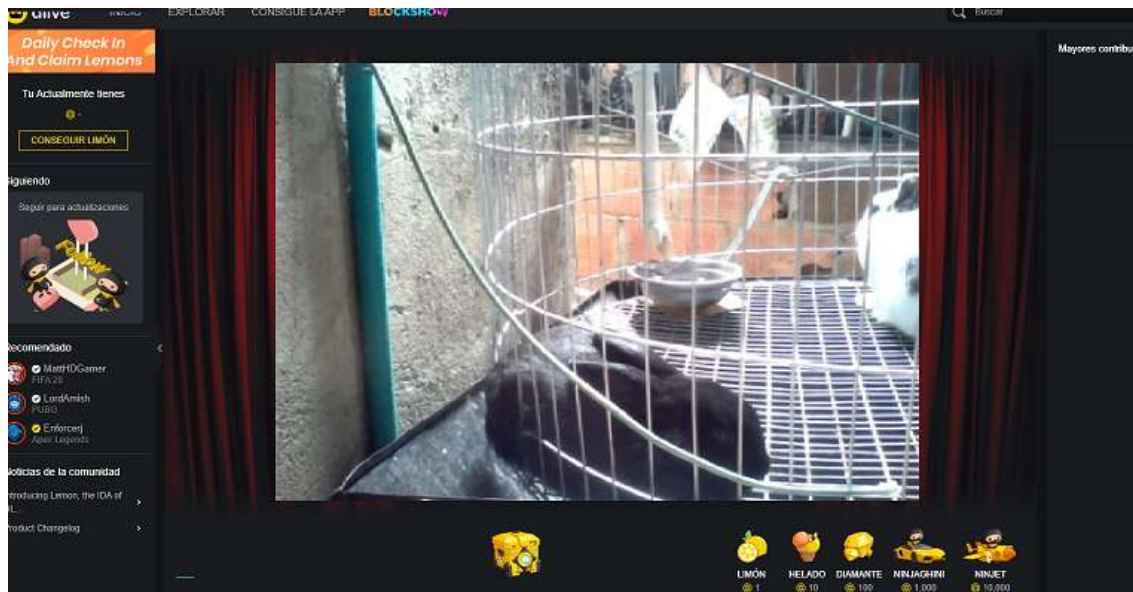
Peso	Area	Racion	Ji
606	170	40	1
884	200	50	1
1145	230	58	1
1471	270	67	1
1697	300	71	1
2044	350	77	1

*Ilustración 27 Vista de registro crecimiento en el sistema de información  
Fuente Propia*

Entre las opciones de vista permitidas para el usuario se encuentra la opción de registro de crecimiento, Ilustración 27, en donde se encuentra una lista de los registros pertinentes a las jaulas, en la parte de arriba de la página se encuentra a disposición del usuario, la opción de buscar y seleccionar la jaula a la cual quiera acceder el usuario, haciendo uso de la opción de seleccionar en la ilustración 28, se puede observar la forma del informe generada para el usuario, mostrando entre sus opciones un botón llamado monitoreo de conejos.



*Ilustración 28 Sección de informes en el sistema de información  
Fuente propia*



*Ilustración 29 Monitoreo opción de informe  
Fuente propia*

Al dar *click* en el botón monitoreo de conejos, esta acción redirecciona a un sitio web donde se puede ver en tiempo real la actividad en la jaula, la cual se puede evidenciar en la Ilustración 29 presentada anteriormente.

### **5.5 Análisis de los datos teniendo en cuenta la conversión alimento consumido, peso ganado conejo, según estudios y opiniones de expertos**

En lo expresado en el ítem anterior, se tiene: para producir 3 kg, por el medio automatizado se puede obtener con aproximadamente 9 kg de alimento seco en conejo nueva Zelanda, cabe denotar que la raza es muy importante con la ganancia de peso vs alimentación, esto debido a que los conejos que fueron criados de forma artesanal siendo de raza mestiza o criolla tardaron más tiempo, además la alimentación fue de 33 kg, dividida en 13 kg de concentrado y 20 kg en desperdicio de la huerta, este alimento es aprovechado en un 65%. Para la obtención de este dato se determinó experimentalmente, al colocar 1000 gramos de alimento como hojas de brócoli, coliflor, etc; sobre la jaula a primera hora de la mañana, para posteriormente en la tarde recolectar un desperdicio de 350 gramos, el proceso no es totalmente certero, ya que influye que el alimento sea de gusto para el conejo, el estado de ánimo y la necesidad de alimento que este tenga.

Según (José Fabio, 2006), para calcular el alimento que debe comer un conejo para producir cierta cantidad de carne, se debe tener en cuenta la raza y el tipo, siendo coneja paridora, padrote, levante otros autores tienen en cuenta los conejos de reserva. Para raza como nueva Zelanda en engorde, según el autor consumen desde el destete al sacrificio un aproximado de 3 a 3.5 kg de alimento por kilo de peso vivo, aunque también va en el desarrollo del gazapo y la buena alimentación de la madre. En los conejos utilizados en el prototipo, pesando cada uno aproximadamente 1.5 kg vivo.

$$(3kg * 3.2kg) = 9.6kg$$

Se esperaba que el experimento se consumiesen 10 kg de alimento para producir 3 kg de peso vivo, dando como resultado un consumo mínimo de lo esperado siendo favorable para la producción cunícola.

El proceso de crianza artesanal, aumento considerablemente en su costo de alimentación, esto debido al desperdicio que se presentó, además la raza de conejo contribuyo en el aumento, siendo muy poco rentable la cría de conejos de forma artesanal, sin ningún control ni medida de alimento según el peso del conejo. Al momento de realizar el sacrificio el valor de este, solo cubre los gastos de alimentación, aunque la cría de conejo artesanal generalmente es para garantizar carne a la familia mas no para una explotación cunícola.

## Conclusiones

La implementación de un alimentador por medio de un tubo pvc haciendo uso de un sinfín y un motorreductor, es bastante eficiente para realizar el proceso de transporte hacia la jaula correspondiente, además este puede contener una gran cantidad de alimento que ayuda a independizar el proceso de, la supervisión del cunicultor, el buen funcionamiento de este instrumento puede ser evidenciado por medio del monitoreo remoto implementado en el sistema de información. Cabe resaltar que este presenta un bajo costo a comparación de otros alimentadores existentes en el mercado.

En primera instancia para el cálculo correspondiente del peso del conejo se presentó como propuesta el uso de una regresión lineal simple, el cual no fue descartado, pero debido a inconvenientes a la hora del cálculo presentaba desfases muy grandes y fue necesario ser mejorado, convirtiéndose en una regresión polinomial multivariable, la cual se ajusta mejor a la curva correspondiente a los datos reales tomados, generando un margen de error mínimo con una variación de 0 % a 10% correspondiente al peso real del animal.

Factores que intervienen en el proceso de identificación del área del animal varía dependiendo del tipo de cámara o píxeles soportados por ésta, la raza del animal y la distancia de la cámara a la base de la jaula. Con respecto a la raza del animal se debe tener presente que el área de estas presenta discrepancias con el peso real, y estos deben ser ajustados dependiendo de la raza.

El desarrollo del algoritmo implementado para determinar el área de los conejos fue diseñado en dos dimensiones utilizando una cámara en la parte superior de la jaula, el cual realiza su labor de predicción del área de manera eficiente y acertada, pero esta podría ser mejorada siendo realizada en tres dimensiones, con la inclusión de otras cámaras en diferentes ángulos proporcionando una mayor precisión en la estimación del peso del conejo haciendo una relación entre ellas, las cuales generarían un volumen, dando una estimación aun mas precisa, ya que se tiene en cuenta toda la forma del conejo.



Para la detección de la forma del conejo se propuso el algoritmo de detección de objetos MaskRCNN, teniendo en cuenta que esta genera un área sombreada correspondiente a la máscara de los conejos se hizo más fácil el cálculo del peso de estos, generando más precisión a la hora de detectar los contornos y no se tendría que someter la imagen a extensos procesos de filtrado, para mejorar este proceso, se propone la implementación y utilización de una cámara térmica la cual genera un área aun mas precisa ignorando las texturas y objetos en la imagen que puedan distorsionar la precisión en la detección de los conejos, haciendo este proceso un poco más rápido ya que no se tendría que realizar la detección del objeto por medio del algoritmo MaskRCNN, el cual realiza cálculos mediante una configuración con redes neuronales, haciendo el proceso temporalmente más extenso.

## Bibliografía

- Aguilera, d. G. (2011). *Procesamiento de imágenes*. Obtenido de universidad de salamanca:  
<http://ocw.usal.es/enseñanzas-tecnicas/procesamiento-avanzado-de-imagenes-digitales/contenidos/tema2.pdf/view>
- Cabeza, l. M. (2004). *El sistema integrado de información ganadera*. E-cooperación en la administración pública.
- Dimas lópez, a., & barroso benítez, b. (2015). *Implementacion de un sistema de identificacion ganadero de bajo costo para el control de eventos productivos en el sector pecuario, mediante dispositivos móviles*. Lórica,colombia: universidad de córdoba.
- Electrónica, p. (10 de julio de 2018). *Promax electrónica*. Obtenido de  
<http://www.promax.es/esp/noticias/400/camara-termografica-como-funcionan-y-por-que-pueden-ser-necesarias>
- Herrador, r. E. (13 de noviembre de 2009). *Guía de usuario de arduino*. Córdoba, españa: creative commons.
- Laudon, k., & laudon, j. (2012). *Sistemas de informacion gerencial*. Aucalpan de juárez, estado de méxico: pearson educación.
- Loredo, r. S. (2014). Monitoreo de la producción animal mpa. *Revista iberoamericana de las ciencias biológicas y agropecuarias*.
- Pérez sáenz, r., duarte de la cruz, d., márquez olivera, m. V., & sánchez garcía, o. (2017). *Desarrollo de una base de datos con imágenes termográficas para uso en algoritmos de visión e inteligencia artificial*. Celaya,méxico: tecnológico nacional de méxico.
- Ramos martín, a., & ramos martín, j. (2011). *Aplicaciones web*. Madrid, españa: editorial paraninfo.
- Sáenz pérez, r., márquez olivera, m. V., sánchez garcía, o. J., duarte de la cruz, d., hernández herrera, v. G., & marrujo garcía, l. G. (2018). *Detección de ovinos preñados usando algoritmos de inteligencia y visión artificial*. Celaya, méxico: tecnológico nacional.

*Tecnología, abc.* (madrid, españa de 2013). Obtenido de que es raspberry pi y para qué sirve:  
<https://www.abc.es/tecnologia/informatica-hardware/20130716/abci-raspberry-como-201307151936.html>

Viola, p., & jones, m. (2001). Rapid object detection using a boosted cascade of simple features. *Computer vision and pattern recognition, 2001. Cvpr 2001. Proceedings of the 2001 ieee computer society conference on*, 1.

*Algoritmo Regresión lineal.* (2018, Mayo). Retrieved from *aprendemachinelearning*:

<https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/>  
BECERRA, J. J. (2018). *Aplicación de sistemas embebidos para la automatización de cultivos urbanos.* . Pamplona: Universidad de Pamplona.

CAÑAS, J. J. (2019). *AUTOMATIZACIÓN INTEGRADA DEL CULTIVO HIDROPÓNICO Y EL LOMBRICULTIVO COMPONENTES DE LA GRANJA CUNICOLA AUTOSUSTENTABLE EN EL MUNICIPIO DE MUTISCUA.* Pamplona: Universidad de Pamplona.

Daniel. (2011). *“LA IMPORTANCIA DE LA AUTOMATIZACIÓN EN INVERNADEROS PARA EL CULTIVO DE JITOMATE EN EL ESTADO DE MORELOS.* UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, Mexico MDF.

García, B. (2018, 06 11). *Inteligencia artificial y automatización, el futuro del ser humano en peligro.* Retrieved from <https://blogthinkbig.com/automatizacion-del-empleo>

*github.* (2018, Junio). Retrieved from <https://user-images.githubusercontent.com/35247113/40924949-10e51e80-684b-11e8-87d2-3a68355b0949.png>

Hernandez roberto, Fernandez carlos, & Baptista pilar. (2006). *Metodologia de la investigacion 4 edición.* Mexico. D. F.: McGraw-Hili \_ Interamericana .

HERNANDEZ, N. A. (2004). *SISTEMA DE VISIÓN ARTIFICIAL PARA LA ESTIMACIÓN DEL PESO DE GANADO BOVINO.* BUCARAMANGA: UNIVERSIDAD INDUSTRIAL DE SANTANDER.

José Fabio, A. (2006, 09 20). *Alimentos para Conejos. Aspectos básicos de alimentación para la producción intensiva.* Retrieved from Departamento de Nutrición Animal. Corporación PIPASA: <https://www.engormix.com/cunicultura/articulos/alimentos-conejos-aspectos-basicos-t26578.htm>

Luis, V. H. (2019, 05 12). *Curso de programacion arduino ProgramarFacil.* Retrieved from #114 Sigfox, Arduino MKRFOX1200 y un medidor de radiación UV:  
<https://programarfacil.com/podcast/sigfox-arduino-mkrfox1200-radiacion-uv/>

*Máquinas de Estado Finito.* (2019, Junio). Retrieved from Unicauca:  
<ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/SEDS/Material%20Auxiliar/FM.pdf>

matterport. (2019, Junio). *Github.* Retrieved from Algoritmos para MaskRCNN.:  
[https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN)

Molina, G., & Rodrigo, M. F. (2010). *Estadística descriptiva en Psicología.* Valencia: Universidad de Valencia.

*python.org.* (2009, Septiembre). Retrieved from  
<http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>

## Anexo 1.

Código referente al procesamiento de imágenes, en el sistema de información y monitoreo, basado en el código de implementación procesamiento de imágenes planteado por (matterport, 2019), en este código se realiza la detección del área, y se calcula el peso y ración por medio de una regresión lineal.

```
# Importamos librerías
import cv2
import numpy as np
import sys
import math
import smtplib
import mimetypes
from os import remove
import os
import sys
sys.path.append(os.path.join("../samples/coco/")) # To find local version
import coco
import mrcnn.model as modellib
from mrcnn import utils
from mrcnn.model import log
from PIL import Image
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import mysql.connector
from mysql.connector import Error
import time
import datetime
import skimage.draw

import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
#get_ipython().run_line_magic('matplotlib', 'inline')
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
```

```

plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score

# Importamos los módulos necesarios
from email.mime.multipart import MIMEMultipart
from email.mime.image import MIMEImage
from email.mime.text import MIMEText

# Cargamos una fuente de texto
font = cv2.FONT_HERSHEY_SIMPLEX

def display_images(images, titles=None, cols=4, cmap=None, norm=None,
                    interpolation=None):
    titles = titles if titles is not None else [""] * len(images)
    rows = len(images) // cols + 1
    plt.figure(figsize=(14, 14 * rows // cols))
    i = 1
    for image, title in zip(images, titles):
        plt.subplot(rows, cols, i)
        plt.title(title, fontsize=9)
        plt.axis('off')
        plt.imshow(image.astype(np.uint8), cmap=cmap,
                   norm=norm, interpolation=interpolation)
        i += 1
    plt.show()

def color_splash(image, mask):
    """Apply color splash effect.
    image: RGB image [height, width, 3]
    mask: instance segmentation mask [height, width, instance count]

    Returns result image.
    """
    # Make a grayscale copy of the image. The grayscale copy still
    # has 3 RGB channels, though.
    gray = skimage.color.gray2rgb(skimage.color.rgb2gray(image)) * 255
    # Copy color pixels from the original color image where mask is set
    if mask.shape[-1] > 0:
        # We're treating all instances as one, so collapse the mask into one layer
        mask = (np.sum(mask, -1, keepdims=True) >= 1)
        splash = np.where(mask, image, gray).astype(np.uint8)
    else:
        splash = gray.astype(np.uint8)

```

```
return splash
```

```
def apply_mask(image, mask, color, alpha=0.5):  
    dimensions = img.shape  
    height = img.shape[0]  
    width = img.shape[1]  
    print('Image Dimension   : ', dimensions)  
    print('Image Height      : ', height)  
    print('Image Width       : ', width)  
    size = (height, width)  
    imw = np.ones(size)  
    for n, c in enumerate(color):  
        image[:, :, n] = np.where(  
            mask == 1,  
            image[:, :, n] * (1 - alpha) + alpha * c,  
            image[:, :, n]  
        )  
    return image
```

```
def random_colors(N):  
    np.random.seed(1)  
    colors = [(255,0,0) for _ in range(N)]  
    return colors
```

```
def display_instances(image, boxes, masks, ids, names, scores):  
    n_instances = boxes.shape[0]  
    colors = random_colors(n_instances)  
    print(str(colors))  
    if not n_instances:  
        print('NO INSTANCES TO DISPLAY')  
    else:  
        assert boxes.shape[0] == masks.shape[-1] == ids.shape[0]  
  
    for i, color in enumerate(colors):  
        if not np.any(boxes[i]):  
            continue  
  
        y1, x1, y2, x2 = boxes[i]  
        label = names[ids[i]]  
        score = scores[i] if scores is not None else None  
        caption = '{} {:.2f}'.format(label, score) if score else label  
        mask = masks[:, :, i]  
  
        image = apply_mask(image, mask, color)  
  
return image
```

```

def Main(img):
    results = model.detect([img], verbose=0)
    r = results[0]

    frame = display_instances(
        img, r['rois'], r['masks'], r['class_ids'], class_names, r['scores']
    )

    splash = color_splash(img, r['masks'])
    original = splash
    img = original.copy()

    cv2.imshow('all', original)
    img[:, :, 2] = 0

    im_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

    low_blue = np.array([100, 50, 50])
    high_blue = np.array([130, 255, 255])

    mask = cv2.inRange(im_hsv, low_blue, high_blue)
    umbral = cv2.threshold(mask, 25, 255, cv2.THRESH_BINARY)[1]

    # Dilatamos el umbral para tapar agujeros
    umbral = cv2.dilate(umbral, None, iterations=2)

    # Copiamos el umbral para detectar los contornos
    contornosimg = umbral.copy()

    contours, _ = cv2.findContours(contornosimg, cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
    # cv2.drawContours(img, contours, -1, (0,255,0), 2)
    areas = []
    for c in contours:
        area = cv2.contourArea(c)
        if area > 1000 and area < 100000:
            cv2.drawContours(original, [c], 0, (0, 255, 0), 2, cv2.LINE_AA)

        momentos = cv2.moments(c)
        cx = int(momentos['m10'] / momentos['m00'])
        cy = int(momentos['m01'] / momentos['m00'])

        # Dibujar el centro
        cv2.circle(original, (cx, cy), 3, (0, 255, 0), -1)
        area2 = cv2.contourArea(c)

```

```

cm = area2 * 2.54 / 96
areas.append(cm)
# Escribimos las coordenadas del centro
cv2.putText(original, "(x: " + str(cx) + ", y: " + str(cy) + ", Area: " + str(cm) + ")",
             (cx + 10, cy + 10), font, 0.5, (255, 255, 255), 1)

# Mostramos la imagen final
cv2.imshow('Final', original)

while (1):
    tecla = cv2.waitKey(5) & 0xFF
    if tecla == 27:
        break

cv2.destroyAllWindows()

return int(r['rois'].shape[0]), areas, original
def consulta(id):
    try:
        mySQLConnection = mysql.connector.connect(host='107.180.44.154',
                                                  database='dbconejos',
                                                  user='nunlamedcaf',
                                                  password='8246Asdf//')

        cursor = mySQLConnection.cursor(buffered=True)
        sql_select_query = """select * from jails where id = %s """
        cursor.execute(sql_select_query, (id,))
        record = cursor.fetchone()
        return int(record[3])

    except mysql.connector.Error as error:
        print("Failed to get record from MySQL table: {}".format(error))

    finally:
        if (mySQLConnection.is_connected()):
            cursor.close()
            mySQLConnection.close()
            print("MySQL connection is closed")

def Mail(Peso,Racion):

    # Creamos objeto Multipart, quien será el recipiente que enviaremos
    msg = MIMEMultipart()
    msg['From'] = "abuabara1112@gmail.com"
    msg['To'] = "mabuabaraa@gmail.com"

```



```
msg['Subject'] = "Imágenes Informe"
msg.attach(MIMEText("Script Ejecutado Mostrando las siguiente informacion Peso:
"+str(Peso)+" Racion: "+str(Racion)+"."))
```

```
# Adjuntamos Imagen
```

```
file = open("Informe.jpg", "rb")
attach_image = MIMEImage(file.read())
attach_image.add_header('Content-Disposition', 'attachment; filename = "Informe"')
msg.attach(attach_image)
```

```
# Autenticamos
```

```
mailServer = smtplib.SMTP('smtp.gmail.com', 587)
mailServer.ehlo()
mailServer.starttls()
mailServer.ehlo()
mailServer.login("@gmail.com", "admin123")
```

```
# Enviamos
```

```
mailServer.sendmail("cunicolafarm@gmail.com", "mabuabaraa@gmail.com",
msg.as_string())
```

```
# Cerramos conexión
```

```
mailServer.close()
remove("Informe.jpg")
```

```
def shot():
```

```
# 1.creating a video object
```

```
video = cv2.VideoCapture(0)
```

```
# 2. Variable
```

```
a = 0
```

```
# 3. While loop
```

```
a = a + 1
```

```
check, frame = video.read()
```

```
img = frame
```

```
video.release()
```

```
# Mostramos la imagen final
```

```
cv2.imshow('F', img)
```

```
while (1):
```

```
    tecla = cv2.waitKey(5) & 0xFF
```

```
    if tecla == 27:
```

```
        break
```

```
cv2.destroyAllWindows()
```

```
return img
```

```
def insertReg(peso, racion, idjail, date, sumt):
```

```
    mydb = mysql.connector.connect(  
        host="107.180.44.154",  
        user="nunlamedcaf",  
        passwd="8246Asdf/",  
        database="dbconejos"  
    )
```

```
    mycursor = mydb.cursor()
```

```
    sql = "INSERT INTO increases (peso, area, racion, jail_id, created_at) VALUES (%s, %s, %s,  
%s, %s)"
```

```
    val = [  
        (peso, sumt, racion, idjail, date)  
    ]
```

```
    mycursor.executemany(sql, val)
```

```
    mydb.commit()
```

```
    print(mycursor.rowcount, "was inserted.")
```

```
def getJA(id):
```

```
    try:
```

```
        mySQLConnection = mysql.connector.connect(host='107.180.44.154',  
            database='dbconejos',  
            user='nunlamedcaf',  
            password='8246Asdf/')
```

```
        cursor = mySQLConnection.cursor(buffered=True)
```

```
        sql_select_query = """select * from increases where jail_id = %s ORDER BY id DESC"""
```

```
        cursor.execute(sql_select_query, (id,))
```

```
        record = cursor.fetchone()
```

```
        return int(record[2])
```

```
except mysql.connector.Error as error:
```

```
    print("Failed to get record from MySQL table: {}".format(error))
```

```
finally:
```

```
    if mySQLConnection.is_connected():
```

```
        cursor.close()
```

```
        mySQLConnection.close()
```

```
        print("MySQL connection is closed")
```

```

def calcP(Area):
    data = pd.read_csv("./tconejos.csv", delimiter=',')

    filtered_data = data[(data['Area'] <= 3500) & (data['Peso'] <= 8000)]

    f1 = filtered_data['Area'].values
    f2 = filtered_data['Peso'].values

    # filtered_data.describe()
    dataX = filtered_data[["Area"]]
    X_train = np.array(dataX)
    y_train = filtered_data['Peso'].values

    # Creamos el objeto de Regresión Linear
    regr = linear_model.LinearRegression()

    # Entrenamos nuestro modelo
    regr.fit(X_train, y_train)

    # Prediccion
    y_pred = regr.predict(X_train)

    # Cuánto "Peso" hay en Area, según el modelo:
    print('Peso Para un Area de ', Area, ' : ')

    Peso = regr.predict([[Area]])
    print(int(Peso))

    suma = (filtered_data['Peso'])

    dataX2 = pd.DataFrame()
    dataX2["Area"] = filtered_data["Area"]
    dataX2["suma"] = suma
    XY_train = np.array(dataX2)
    z_train = filtered_data['Racion'].values

    # Creamos un nuevo objeto de Regresión Lineal
    regr2 = linear_model.LinearRegression()

    # Entrenamos el modelo con 2 dimensiones
    regr2.fit(XY_train, z_train)

    # Prediccion
    z_pred = regr2.predict(XY_train)

```

```
print('Racion Para un Area de ', Area, ' y Peso de', Peso, ' : ')
Racion = regr2.predict([[Area, Peso]])
print(int(Racion))
```

```
return Peso, Racion
```

```
if __name__ == '__main__':
    crabs = consulta(1)
    img = shot()

    ts = time.time()
    timestamp = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')

    ROOT_DIR = os.getcwd()
    MODEL_DIR = os.path.join(ROOT_DIR, "logs")
    COCO_MODEL_PATH = os.path.join(ROOT_DIR, "../mask_rcnn_rabbit_0010.h5")
    if not os.path.exists(COCO_MODEL_PATH):
        utils.download_trained_weights(COCO_MODEL_PATH)

    class InferenceConfig(coco.CocoConfig):
        GPU_COUNT = 1
        IMAGES_PER_GPU = 1

    config = InferenceConfig()
    config.display()

    model = modellib.MaskRCNN(
        mode="inference", model_dir=MODEL_DIR, config=config
    )
    model.load_weights(COCO_MODEL_PATH, by_name=True)
    class_names = [
        'BG', 'rabbit']

    nm, areas, total= Main(img)
    print(str(nm)+" , ")

    f=0

    while f == 0:
        ArJa = getJA(1)
        li = ArJa
        ls = li + 200
        img = shot()
        nm, areas, total = Main(img)
```

```
print(str(nm) + ", ")
sumt = 0
for x in areas:
    sumt += x
print(sumt)

# if nm != crabs :
#   f=0
if li < sumt and sumt < ls and nm == crabs:
    print("ahsdsadhkjasdhlas")
    f = 1
    break
```

Peso, Racion = calcP(sumt)

```
cv2.imwrite("Informe.jpg", total)
print(Racion)
insertReg(int(Peso), int(Racion[0]), 1, timestamp, round(sumt))
Mail(int(Peso), int(Racion[0]))
sys.exit()
```

## **Anexo 2**

2 Automatización como plan de negocio para una empresa

### **AUTOMATIZACIÓN E INTELIGENCIA ARTIFICIAL APLICADA A LA AGRICULTURA LIZABU**

#### **LOGO**



Ilustración 30 Logo empresa

#### **PLAN DE NEGOCIO**

**Correo Electrónico**

**[lizabuautomatizaciones@gmail.com](mailto:lizabuautomatizaciones@gmail.com)**

**Página Web**

**<http://lizabu.com/>**

**CÓDIGO POSTAL: 543050**

**TELÉFONO: 3173291860**

**Moisés Abuabara Angarita**

## **2.1 Introducción como empresa**

Automatización e Inteligencia Artificial Aplicada a la Agricultura **Lizabu**, es una empresa desarrolladora de sistemas automatizados, los cuales están orientados mayormente a la agricultura, aunque también se pueden realizar proyectos en cualquier otra actividad, donde se requiera implementar la automatización y/o la inteligencia artificial. Al ser una empresa con base tecnológica, hace que constantemente esté aplicando nuevas metodologías o procedimientos para aumentar la eficiencia en el desarrollo de proyectos, por lo cual una de sus principales virtudes es la innovación, continuamente se están probando nuevas tecnologías para implementar en los sistemas automatizados.

La empresa está ubicada en el municipio de Pamplona Norte de Santander (sede principal), este proyecto nace de un proyecto de investigación como trabajo de grado.

## **2.2 Antecedentes**

### **2.2.1 Nombre de la empresa**

Automatización e Inteligencia Artificial Aplicada a la Agricultura Lizabu

### **2.2.2 Concepto del negocio**

Automatización e Inteligencia Artificial Aplicada a la Agricultura Lizabu, como producto principal ofrece la automatización inteligente ya sea para la industria o para el agro, de este producto se derivan servicios como asesoría, mantenimiento, actualización de sistemas automáticos, etc. El producto y los servicios ofrecidos tienen gran pedido, aunque se debe tener en cuenta que mientras se da a conocer la empresa los proyectos son pocos.

### **2.2.3 Descripción de la empresa**

Automatización e Inteligencia Artificial Aplicada a la Agricultura Lizabu, está impulsada por la base tecnológica y la necesidad del mercado colombiano en la implementación de sistemas automatización con soporte en inteligencia artificial, la empresa se encuentra en Pamplona Norte de Santander donde nació. Dentro de los servicios que la empresa ofrece se encuentran:

Análisis de sistemas de información para la automatización

Análisis de sistemas automáticos

Diseño de sistemas automáticos (diseño de circuitos, diseño de software, hardware, etc.)

Desarrollo de sistemas automáticos inteligentes

Asesoría sobre equipos para automatizar

Mantenimiento de sistemas automatizados

### **2.2.4 Que tiene de innovadora la idea de negocio**

La inclusión de inteligencia artificial en la automatización utilizada en la agroindustria, incluyendo procedimientos como el análisis de patrones en imágenes, redes neuronales, etc. Al mismo modo la utilización de la última tecnología en instrumentación (software y hardware), con una constante capacitación en la implementación de nuevas herramientas, metodologías de trabajo, etc.

### **2.2.5 Beneficios del producto o servicio que ofrece**

Dentro de los principales beneficios podemos encontrar la calidad del producto y los servicios ofrecidos, si el producto no cumple con los requerimientos previamente pactados entre las dos partes, la empresa se encarga de realizar la devolución del dinero con intereses. Como segundos beneficios se tiene la garantía total de la funcionalidad y la usabilidad del producto a largo plazo, ya que cuenta con los mejores dispositivos para garantizar la durabilidad. Al momento de realizar la automatización, se requiere ser específico en cuanto a la recolección de la información y los procesos a automatizar, por lo cual la mayoría de los procesos se tecnifican, mejorando la calidad de vida del cliente y dependiendo del tipo de automatización, la producción aumenta y los gastos se minimizan.



### **2.2.6 Ventajas competitivas**

Una de las principales ventajas de la empresa es la innovación, el estar constantemente investigando sobre nuevas tecnologías, inteligencia artificial y otros medios que le permitan mayor eficiencia en los procesos de automatización. Otra de las ventajas competitivas es la incursión en un mercado colombiano el cual está poco desarrollado en este ámbito, ya que la inteligencia artificial en procesos de automatización en el agro o en otros campos es muy poco explorado, hay casos específicos que la misma empresa es la encargada de realizar las implementaciones de inteligencia artificial pero no comercializan el producto, además es un mercado donde la tecnología cada día es más utilizada y donde la inteligencia artificial transformará el futuro.

### **2.2.7 Características de los clientes potenciales**

El producto y servicios van orientados a cualquier individuo natural o jurídico, que desee implementar un sistema de automatización o desee conocer sobre el campo de la automatización inteligente, por lo cual la variedad de los clientes potenciales es amplia, siendo beneficioso para la empresa.

### **2.2.8 Contexto donde se va a desarrollar el negocio**

La empresa da sus inicios en Pamplona Norte de Santander, aunque físicamente se encuentra en Pamplona, las ventas están orientadas a todo el territorio nacional colombiano, donde el principal laboratorio de pruebas y prototipos está en Mutiscua.

### **2.2.9 Estudio de factibilidad**

#### **Financiero**

La empresa Automatización e Inteligencia Artificial Aplicada a la Agricultura Lizabu, demanda un costo inicial de \$10.000.000 (diez millones de pesos colombianos) para la compra de equipos para poder realizar proyectos que requieran la construcción de piezas, entre otros.

### **Social**

La aceptación por la comunidad Pamplonesa y la sociedad en general es positiva, ya que le empresa genera progreso para el municipio, también ayuda al crecimiento económico teniendo en cuenta que para un futuro puede generar empleo y ayudar a mejorar la calidad de vida de muchos individuos al automatizar procesos aumentando las ganancias reflejadas en la producción.

### **Medio ambiente**

La implementación de automatización con inteligencia artificial, hace que los procesos sean efectivos, donde se puede observar mayor impacto ambiental es en el agro, ayuda a minimizar los agentes contaminantes, ahorro de agua, etc. Según (Daniel, 2011) la automatización de un invernadero ayuda minimizar un 80 % de agua. Analizando este ejemplo son muchos los beneficios que le trae al medio ambiente, siempre y cuando la utilización sea responsable.

## **La proyección financiera**

Proyección financiera a un año.

### **Pasivo**

Mensuales

**Varios 300.000**

### **Activos**

Mensuales

Ganancias (\$500.000)

Inversión realizada con las ganancias de la empresa

Inversión a la empresa mensual que se cuenta como gastos es de \$100.000

## **2.2.10 La conveniencia de invertir en el negocio (rentabilidad)**

La inversión en la empresa Automatización e Inteligencia Artificial Aplicada a la Agricultura Lizabu, es una apuesta al éxito ya que el agro colombiano requiere transformaciones que mejoren la producción, implementando aspectos como la automatización y la inteligencia artificial, también se debe tener en cuenta que se crea sobre la necesidad actuales del mercado, proyectadas a las necesidades del futuro.

Según (García, 2018) la automatización y el uso de inteligencia artificial es un campo que cada día está siendo más explorado, al igual las necesidades aumentan cada día, aunque la inclusión de robots causa desempleo. Según un reciente estudio del MIT, la pérdida de empleos causada por la robótica y la inteligencia artificial tendrá un impacto mucho mayor en las ciudades más pequeñas. Según diferentes estudios, este proceso se llevará a cabo en tres grandes oleadas. Por lo cual es rentable para las empresas encargadas de realizar automatización e inteligencia artificial, pero aun futuro afecta el trabajo del ser humano, el cual será reemplazado por máquinas inteligentes.

### **2.2.11 La forma de llevarlo a cabo**

Se implementan dos fases fundamentales para llevar a cabo la realización de la empresa:

#### **Fase de exploración**

Está orientada a investigar y explorar la temática relacionada con inteligencia artificial y automatización, al mismo tiempo se realiza investigaciones del mercado, precios, costos, derechos y deberes según la ley, inscripción ante la cámara de comercio, publicidad, entre otros aspectos relacionados con la empresa.

#### **Implementación**

La implementación en el mercado como primer punto es darse a conocer como empresa, por ello se identificaron en la fase exploratoria los mejores medios de comunicación, los cuales sean efectivos, veraces y económicos para transmitir información, del mismo modo se definen políticas internas para la interacción con el cliente y socios.

En términos generales se plantean ideas, las cuales son evaluadas en la primera fase y según lo determinado en esta fase, al analizar todos los parámetros que influyen en la eficiencia de la idea,

pasa a la fase de implementación de ser prometedora, de lo contrario es desechada. Las siguientes son ítems a analizar como primer paso:

Ventas

Compras

Clientes

Beneficios

Calidad

Promociones

Publicidad

### **2.2.12 Análisis del sector al que ingresa la empresa**

En el mercado colombiano se puede encontrar diferentes empresas tecnológicas, las cuales se encargan de desarrollar proyectos de automatización como: INGEACOL Ingeniería y Automatización de Colombia, Isateck – Automatización, Halud Ingeniería - Equipos de Automatización Indus, etc. Se debe tener en cuenta que la mayoría de estas empresas no tiene sitio web oficial donde se demuestren proyectos anteriores, historia, objetivos, información sobre su constitución legal, entre otros. La particularidad de estas empresas es que no incluyen la inteligencia artificial como producto a utilizar en los procesos de automatización ya sea en el agro o en procesos industriales, además son muy poco conocidos en el mercado, por lo cual la inclusión de una nueva a la empresa legalmente constituida con calidad y usabilidad en los proyectos realizados generarían impacto directo en el mercado de la automatización.

## **2.3 Información del plan de negocio**

### **2.3.1 Misión**

La incursión en nuevas tecnologías en el área de la automatización junto con los avances tecnológicos para el uso de la inteligencia artificial mejoran los procesos de automatización, en aspectos como la precisión y resolución de problemas imprevistos de una forma lógica, ayudando

a la tecnificación de los procesos, con productos que van dirigidos a control de procesos industriales, agrícolas, ganaderos, piscícola, en general cualquier actividad que se puede automatizar.

### **2.3.2 Visión**

La empresa se proyecta como una empresa innovadora en los procesos de automatización, conocida como una empresa responsable y con un compromiso en el ámbito de la capacitación y las aplicaciones de nuevas tecnologías y/o metodologías en el ámbito de la empresa. Entre los aspectos más importante es el compromiso con los proyectos adquiridos garantizando proyectos eficientes con usabilidad. Para el 2020 ser conocida en los departamentos Norte de Santander y Santander por la realización de proyectos con gran impacto y eficiencia, para el año 2035 ser conocida a nivel nacional incursionando en el mercado internacional.

### **2.3.3 Objetivos**

#### **Objetivo general**

Diseñar sistemas automáticos basados en inteligencia artificial, para la automatización de procesos en la producción agrícola.

#### **Objetivos Específicos**

Analizar sistemas orientados a la automatización

Diseñar sistemas orientados a la automatización

Desarrollar sistemas automáticos

Innovar en sistemas automáticos con el uso de inteligencia artificial

Generar investigaciones sobre inteligencia artificial en la automatización

Generar empleos en disciplinas como sistemas, electrónica y eléctrica.

### **2.3.4 Metas**

Anticiparse las necesidades de los clientes para proponer mejores proyectos de automatización

Incorporación de la empresa en el mercado colombiano

Perfeccionamiento de las metodologías, modelos, etc. Implementados en la empresa

Desarrollar nuevos proyectos orientados a la automatización

Aumentar la calidad de los proyectos

Realizar nuevos proyectos basados en la innovación en temas como la inteligencia artificial y la automatización

Disminución de los gastos empleados

Aumentar la eficiencia en tiempos de desarrollo por medio de metodologías ágiles

### **2.3.5 Políticas**

#### **Personal de trabajo:**

Los empleados no pueden involucrar sus problemas familiares con los de la empresa.

Se ocasionará un adelanto de sueldo a los empleados solo cuando tenga pruebas válidas para este mismo ser requerido.

Los empleados deben portar sus uniformes y tomar las medidas de riesgo y seguridad que la empresa le otorga.

La empresa está en la obligación de prestar seguros de vida y salud sin costo alguno a sus empleados, después que estos lleven mínimo 3 años en la empresa.

Ya que un empleado no lleve 3 años en la empresa debe cancelar \$50.000 pesos mensualmente para sus seguros de vida y salud.

#### **Privacidad:**

Los servicios, datos e información del cliente, no es cedida a otras empresas ya sea que presten el mismo servicio o no.

La empresa no se hace responsable del uso de la información o servicios ofrecidos.

sólo recoge los datos estrictamente necesarios para ofrecer los servicios derivados de su actividad y demás prestaciones, atribuidas por la Ley colombiana.

Los Usuarios estarán obligados a hacer un uso razonable de los servicios, bajo el principio de buena fe y con respeto a la legalidad vigente, a la moral, al orden público, a las buenas costumbres, todo ello según las posibilidades y fines para los que están concebidos.

#### 2.4.1 Control financiero

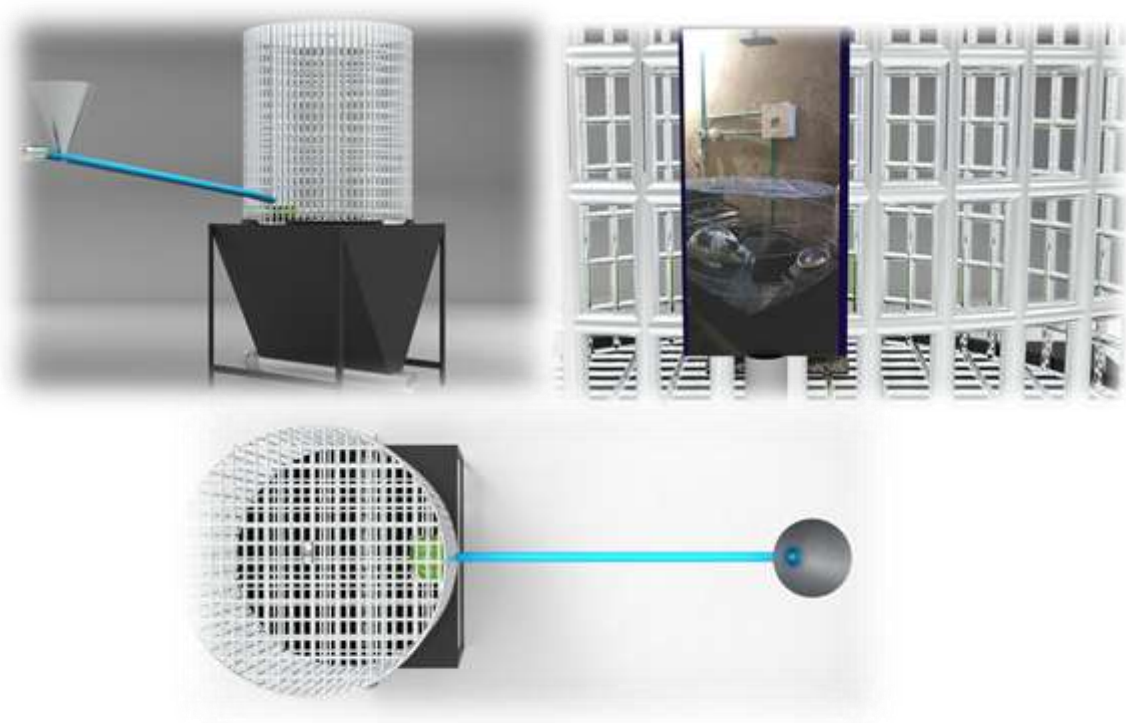
En la siguiente tabla se presentan los costos de producción del prototipo funcional.

<b>COSTO PRODUCCIÓN PROTOTIPO</b>			
<b>CANTIDAD</b>	<b>ÍTEM</b>	<b>PRECIO UNITARIO</b>	<b>PRECIO TOTAL</b>
1	Alambre acerado 3 mm	45,000	45,000
1	Lamina zinc lisa 3mm	50,000	50,000
2	Cámara celular	-	-
2	Placas Arduino	45,000	90,000
1	Módulo HX711	7,000	7,000
1	Placa Raspberry Pi 4	320,000	320,000
2	Módulos esp8266	16,000	32,000
2	Relee de 5v a 12v	7,000	14,000
1	Celda de Carga 50 kg	15,000	15,000
2	Motorreductores 10 kg a 6 rpm	25,000	50,000
1	Web Server en DigitalOcean	22,000	22,000
1	Estaño y váquelas	10,000	10,000
1	Tubo PVC	15,000	15,000
<b>TOTALES</b>		<b>577,000</b>	<b>670,000</b>

La tabla a continuación hace referencia a los costos de producción de el modelo de jaula final.

<b>COSTO PRODUCCIÓN PRODUCTO FINAL</b>			
<b>CANTIDAD</b>	<b>ÍTEM</b>	<b>PRECIO UNITARIO</b>	<b>PRECIO TOTAL</b>
1	Alambre acerado 3 mm	45,000	45,000
1	Lamina zinc lisa 3mm	50,000	50,000
2	Cámara resistente al agua y polvo 8mp	95,000	190,000
2	Protectores para Arduino	35,000	70,000
2	Placas Arduino	45,000	90,000
1	Módulo HX711	7,000	7,000
1	Placa Raspberry Pi 4	320,000	320,000
2	Módulos esp8266	16,000	32,000
2	Relee de 5v a 12v	7,000	14,000
1	Celda de Carga 50 kg	15,000	15,000
2	Motorreductores 10 kg a 6 rpm	25,000	50,000
1	Web Server en DigitalOcean	22,000	22,000
1	Estaño y váquelas	10,000	10,000
1 mts	Tuvo PVC	15,000	15,000
<b>TOTALES</b>		<b>707,000</b>	<b>930,000</b>
<b>PRECIO DE VENTA</b>		<b>1,100,000</b>	

Finalmente se presenta como propuesta las imágenes renderizadas de la apariencia física del producto a comercializar.





Para el análisis de rentabilidad del producto a comercializar, se hace necesario tener en cuenta los costos variables que se generan en la empresa, a continuación, se muestra una estimación financiera para una empresa legalmente constituida.

<b>COSTOS FIJOS</b>	
<b>INFRAESTRUCTURA</b>	
Arriendo local ubicado cra 5 #2-31 san Agustín-pamplona	350,000
Internet banda ancha 10mb	95,000
<b>TOTALES</b>	<b>445,000</b>
<b>MANO DE OBRA</b>	
Ensamblador	1,500,000
Recepción y despacho de producto	950,000
<b>TOTALES</b>	<b>2,450,000</b>
<b>COSTOS INDIRECTOS</b>	
Servicio energía	75,000
Acueducto	35,000
<b>TOTAL COSTOS INDIRECTOS</b>	<b>110,000</b>
<b>TOTAL COSTOS FIJOS</b>	<b>3,005,000</b>

Al analizar los costos de producción y costos fijos de la empresa, se estima un punto de equilibrio el cual se muestra en la tabla a continuación.

<b>PUNTO DE EQUILIBRIO</b>	
p.e unidades	17.67647
p.e\$	19,444,118

Teniendo en cuenta la información anterior para que el producto a producir sea rentable se estima que deben ser vendidas en promedio 18 unidades del producto mensualmente.

## 2.4.2 Rentabilidad para el cunicultor

Teniendo en cuenta compra del producto final para un cunicultor orientado a una pequeña empresa, se realiza una estimación para conocer si es rentable la inversión en el producto, realizando un comparativo de los gastos y ganancias entre una producción tradicional y una producción automatizada haciendo uso del producto, dando como resultado la información presentadas en las tablas a continuación para un total de 300 conejos y la adquisición de 50 jaulas, suponiendo que no exista pérdida de alimento para una producción tradicional representado con la ración correspondiente a la producción automatizada.

<b>TRADICIONAL</b>		
<b>COSTOS</b>		
<b>CONEJOS</b>	COMIDA 70 DIAS POR CONEJO	16,333
	UNIDADES	300
	<b>TOTAL</b>	<b>4,899,900</b>
<b>TRABAJADOR</b>	70 DIAS	<b>1,400,000</b>
	OTROS COSTOS	<b>200,000</b>
	<b>TOTAL COSTOS</b>	<b>6,499,900</b>
<b>VENTA</b>		
	PRECIO KILO CONEJO	14,000
	CONEJOS DE 2 KILOS	600
	<b>TOTAL VENTA</b>	<b>8,400,000</b>
	<b>GANANCIA</b>	<b>1,900,100</b>

<b>AUTOMATIZACIÓN</b>		
<b>COSTOS</b>		
<b>CONEJOS</b>	COMIDA 70 DIAS POR CONEJO	16,333
	UNIDADES	300
	<b>TOTAL</b>	<b>4,899,900</b>
	OTROS COSTOS	<b>200,000</b>
	<b>TOTAL COSTOS</b>	<b>5,099,900</b>
<b>VENTA</b>		
	PRECIO KILO CONEJO	14,000
	CONEJOS DE 2 KILOS	600
	<b>TOTAL VENTA</b>	<b>8,400,000</b>
	<b>GANANCIA</b>	<b>3,300,100</b>

Realizando la adquisición de 50 unidades del producto representada en una inversión equivalente a \$55.000.000 de pesos, teniendo en cuenta el precio unitario del producto por un valor de \$1.100.000 de pesos. Se calcula el tiempo de recuperación de una inversión con esta magnitud, representada en la siguiente tabla.

<b>Periodos Anuales</b>
5.2
<b>Recuperación Anual</b>
17,207,664.29
<b>Recuperación Inversión</b>
3.20